

UNIVERSIDAD DE
EL SALVADOR
FACULTAD DE CIENCIAS NATURALES Y
MATEMÁTICA
ESCUELA DE FÍSICA



Trabajo de graduación
**Estudio de las propiedades ópticas
de puntos cuánticos de carbono grafeno**

Presenta:

Jorge Alberto Cuadra Aparicio CA14010

Para optar por el grado de:

Licenciado en Física

Docente Director:

Prof. PhD (Eur) Carlos Ernesto Rudamas Flores

Fecha:

15 de octubre de 2020

Agradecimientos

La realización de esta tesis solo fue posible gracias al apoyo incondicional de mi familia, la cual siempre confió en mi sin importar las circunstancias, y el grupo de investigación del Dr. Calos Rudamas, quienes me apoyaron en todas las discusiones concerniente a mi “tierrita”, revisión de mis textos científicos con inclinaciones poéticas/coloquiales y cumplimientos de *deadlines* a último minuto. Mis más sinceros agradecimientos a todos ustedes por estar conmigo durante todo este recorrido y permitirme alcanzar mis metas.

Lista de Publicaciones

El presente trabajo ha sido parcialmente presentado en las siguientes publicaciones o congresos.

- [1] Jorge Cuadra, Hamilton Ponce y Carlos Rudamas. “Puntos cuánticos de grafeno-carbono: Propiedades ópticas”. En: *I Central American Research Workshop*. 2019.
- [2] Jorge Cuadra, Hamilton Ponce y Carlos Rudamas. “IEEE 2018 38th Central America and Panama Convention (CONCAPAN XXXVIII) - San Salvador, El Salvador - Optical Properties of Graphene Carbon Quantum Dots for Solar Cells Applications”. En: 2018. ISBN: 978-1-5386-6122-2. DOI: 10.1109/CONCAPAN.2018.8596611. URL: <https://ieeexplore.ieee.org/document/8596611>.
- [3] Jorge Cuadra, Hamilton Ponce y Carlos Rudamas. “OPTICAL TRANSITION STUDY ON GRAPHE-NE CARBON QUANTUM DOTS”. En: *XXVIII International Materials Research Congress, IMRC 2019*. Materials Research Society, 2019.
- [4] Jorge Cuadra, Hamilton Ponce y Carlos Rudamas. “Electronic response function in multi-layered graphene carbon quantum dots”. En: *Semana internacional de la nanotecnología*. 2019.

Índice general

Agradecimientos	II
Lista de Publicaciones	III
Índice de general	VI
Índice de tablas	VII
Índice de figuras	X
Resumen	I
Introducción	III
1. Fundamento teórico	1
1.1. Interacción de la radiación electromagnética con la materia	1
1.1.1. Absorción	1
1.1.1.1. Ley de Beer-Lambert-Bouguer	1
1.1.1.2. Modelo de oscilador armónico forzado	2
1.1.1.3. Fuerza del oscilador	3
1.1.2. Luminiscencia	3
1.2. Puntos cuánticos	4
1.2.1. Puntos cuánticos semiconductores	6
1.2.2. Descripción teórica de los puntos cuánticos	6
1.2.2.1. Pozo cuadrado de potencial finito	6
1.2.2.2. Problema de muchos cuerpos	7
1.3. Estado del conocimiento	8
1.3.1. Propiedades electrónicas del carbono	8
1.3.2. Puntos cuánticos de carbono	9
2. Metodología	14
2.1. Síntesis de muestras	14
2.2. Medición de espectros de absorción	16
2.3. Realización de cálculos	16
2.3.1. Teoría del funcional de densidad	16
2.3.1.1. Teorema, Hamiltoniano y ecuaciones de Kohn y Sham	16
2.3.1.2. Funcional de energía de intercambio-correlación	19
2.3.1.2.1. Aproximación de la densidad local	19
2.3.1.2.2. Aproximación del gradiente general	20
2.3.1.2.3. Aproximación de Van der Walls	20

2.3.1.3. Realización de cálculos de campo autoconsistente	21
2.3.2. Dibujo de estructuras	23
2.3.3. Cálculos de optimización geométrica	24
2.3.4. Simulación de espectros de absorción y respuesta electrónica	25
2.3.5. Cálculos de densidad de estados	25
2.3.6. Método de Numerov	26
3. Resultados y discusión	29
3.1. Síntesis	29
3.2. Optimización geométrica	32
3.3. Espectros de absorción	36
3.4. Origen de las transiciones a partir de DoS	41
3.5. Efecto del radio y número de placas de las estructuras en los espectros de absorción	44
4. Conclusiones y recomendaciones	55
4.1. Recomendaciones	55
4.2. Trabajo a futuro	55
4.3. Conclusiones	56
Referencias bibliográficas	59
Anexos	64
.1. Anexo A: Archivos <i>input</i> para cálculos de optimización geométrica (<i>relaxation calculations</i>)	64
.2. Anexo B: Archivos <i>input</i> para cálculos de DoS y PDoS	65
.3. Anexo C: Archivos <i>input</i> para simulaciones de espectros de absorción cálculos de respuesta electrónica	68
.4. Anexo D: Archivos <i>input</i> para lazar cálculos de optimización geométrica, espectros de absorción, DoS, PDoS y respuesta electrónica en Torque	70
.5. Anexo E: Código de Python3 para apilar placas de grafeno	78
.6. Anexo F: Código de Python3 para post-procesado de los resultados de DoS y PDoS	80
.7. Anexo G: Código de Python3 para resolución de modelo semi-empírico	86

Índice de cuadros

2.1. Etiquetas para muestras sintetizadas por pirólisis	15
2.2. Parámetros para las estructuras de GCQD utilizadas	23
2.3. Porcentaje de átomos para las estructuras de GCQD utilizadas	23
3.1. Valores obtenidos para estados bases y excitados por DFT y el método semi-empírico propuesto	52

Índice de figuras

1.1. Diagrama de los principales fenómenos físicos que ocurren debido a la interacción entre radiación electromagnética y la materia.	1
1.2. Esquema de la ley de Beer-Lambert-Bouguer. Radiación con intensidad I_0 incide en una muestra espesor b y superficie S . La Radiación I , la cual a penetrado una distancia x en la muestra, cumple con una atenuación exponencial.	2
1.3. Principio de Franck-Condon y su efecto en los espectros de absorción y luminiscencia. (a) Transiciones verticales de absorción y luminiscencia favorecidas. (b) Stokes Shift entre el espectro de absorción y luminiscencia.	4
1.4. Representación de un material y la densidad de estados correspondiente para diferentes niveles de restricción espacial. En (a) y (e) se observa el caso para <i>bulk</i> . Las restricciones espaciales sobre una y dos dimensiones se muestran en (b) y (f), junto con (c) y (g) describen los casos de <i>Quantum Wells</i> y <i>Quantum Wires</i> correspondientemente. Mientras el caso de restricción espacial total, esto sería un QDs, se observa en (d) y (h).	5
1.5. Dependencia de los espectros de absorción (a) y luminiscencia (b) respecto al tamaño de la nanoestructura para QDs de CdSe [44].	6
1.6. Modelo de pozo de potencial aplicado para explicar las propiedades de los QDs de PbSe. (a) esquema de niveles de energía y funciones de onda para modelar el confinamiento de QDs de diferentes tamaños. (c) espectros de absorción para QDs de PbSe y su dependencia con el tamaño de la estructura. (b) comportamiento de la energía del band edge respecto al cambio de tamaño en QDs de PbSe. [28].	7
1.7. Diagrama de las múltiples interacciones en un sistema de muchos cuerpos. Por simplicidad solo se muestran la interacción de los núcleos atómicos I y J con las nubes electrónicas i y j	8

1.8.	Representaciones de (a) la hibridación SP^2 , (b) del solapamiento en un enlace σ , (c) de solapamiento en un enlace π , y (d) y la formacion de un enlace doble de etileno por medio de los enlaces σ y π [45].	9
1.9.	Representacion de las celdas unitarias para el diamante y grafito. La direccion de los ejes X, Y y Z estan dadas por las flechas roja, verde y azul respectivamente. Todas las estructuras han sido creadas utilizando Avogadro [47].	10
1.10.	Representacion de la red cristalina del diamante y grafito. La direccion de los ejes X, Y y Z estan dadas por las flechas roja, verde y azul respectivamente. Todas las estructuras han sido creadas utilizando Avogadro [47].	10
1.11.	Estructura de bandas electrónicas para (a) grafeno y (b) grafito. La linea roja indica la energia de Fermi. Ambas estructuras de bandas fueron calculadas utilizando DFT bajo utilizando parámetros similares [46].	11
1.12.	Geometrías presente en los GCQDs deducidas a partir de diferentes técnicas experimentales. (a) Geometría esférica observada en GCQDs. (b) Geometría cilindrica observada en GCQDs [48].	11
1.13.	Mejora en el espectro de absorción de los GCQDs al ser dopados con nitrógeno. Se compara una muestra de GCQDs sin dopar contra una muestra de GCQDs dopados con N [7].	12
1.14.	Diferentes tipos de estructuras utilizadas para la descripción de GCQDs por diferentes autores.	13
1.15.	Comparación entre espectros de absorción experimentales y teórico para GCQDs. La estructura para los cálculos utilizo una sola placa de grafeno, sin dopar y obviando interacciones de VdW. A pesar de cierta concordancia entre los espectros, diferencias importantes son apreciadas[15].	13
2.1.	Proceso utilizado para la síntesis de GCQDs. (a) Se prepara una solución de AC. (b) dicha solución es colocada en un proceso de pirólisis. (c) Se extraen muestras GCQDs a diferentes tiempos.	15
2.2.	Diagrama del arreglo experimental para la medición de espectros de absorción en muestras de GCQDs.	16
2.3.	Esquematización de las fuerzas de VdW en un sistema de dos capas de grafeno. Las flechas indican regiones típicas donde ocurren fluctuaciones de carga. La elipse punteada indica las regiones de interés que describen los funcionales de XC LDA y GGA [17].	21
2.4.	Esquema típico del algoritmo utilizado para cálculos de SCF en DFT[57].	22
2.5.	Proceso para el dibujo de las estructuras de GCQDs utilizadas. (a) El proceso inicia con el dibujo de una placa de grafeno amplia. (b) Una placa circular de grafeno del radio deseado es cortada. (c) la placa es dopada con diferentes tipos de átomos dependiendo del tipo de estructura que desea dibujarse. (d) un número de estas placas son apiladas para obtener la altura deseada.	24
2.6.	Procesos utilizados para los cálculos de optimización geométrica realizados. (a) proceso utilizado y (b) algoritmo de optimización utilizando el método BFGS [66].	25
2.7.	Proceso para el cálculo de espectros de absorción	26
2.8.	Proceso para la simulación de DoS y PDoS.	26
3.1.	Muestras de GCQD con y sin dopaje de nitrógeno obtenidas para un tiempo de pirólisis de 7 min. (a) y (b) muestra de GCQD sin dopaje expuesta bajo luz VIS y UV. (c) y (d) muestra de GCQD sin dopaje expuesta bajo luz VIS y UV.	30

3.2.	Espectros de absorción de diferentes autores, los cuales han realizado mediciones de diámetros a los GCQD sintetizado. Se comparan los espectros de absorción, previamente mencionados, con los espectros medidos de la muestra Sam N-GCQD 01 [7, 70, 71]. En la inserción se puede observar un acercamiento a la banda asociada con transiciones $\pi - \pi^*$ (CC).	30
3.3.	Tamaño (Diámetro) de los GCQD en función del tiempo de pirólisis. (a) dependencia del tamaño contra el tiempo obtenidas mediante mediciones de TEM [6]. (b) Intepolación lineal para la dependencia ente tamaño y tiempo para los primeros 20 <i>min</i> de pirólisis.	31
3.4.	Espectros de absorción para muestras sintetizadas sin (a) y con (b) dopamiento de nitrógeno. Los comportamientos de las bandas de absorción se mantienen para ambas síntesis.	31
3.5.	Comparación entre espectros Raman reportados para CA, grafeno, GCQD por otros autores y el espectro medido para la muestra Sam. N-GCQD 01 [74, 75].	32
3.6.	Energía total del sistema en función del numero de iteraciones (tiempo de cálculo) para la estructura <i>SGN R2H2</i> . Se puede observar un comportamiento asintótico aproximadamente en $-1,0911 \text{ KeV}$ ($-801,95 \text{ Ry}$).	33
3.7.	Dependencia de la energía total del sistema en función del número de iteraciones (tiempo de cálculo). En (a) se puede observar como la energía total del sistema muestra una tendencia a aumentar con cada iteración, mientras que en (b) se puede observar que la energía de total del sistema disminuye. Ambos cálculos de optimización fueron realizados en la estructura <i>SGN R2H4</i> .	33
3.8.	Celda unitaria utilizada para los cálculos de optimización geométrica para la estructura <i>SGO R3H4</i> . Estos parámetros de dimensiones fueron obtenidos con las ecuaciones 3.2, 3.3 y 3.4.	36
3.9.	Comparación de la variación de los espectros de absorción respecto al número de iteraciones Lanczos. Se mantiene constante el ensanchamiento de las bandas de absorción en 68 meV (5 mRy) y se varia el número de iteraciones Lanczos en 4000, 6000, 8000 y 10000 iteraciones. El OS para dichos cálculos es mostrado por una línea punteada morada.	37
3.10.	Comparación de la variación de los espectros de absorción respecto al ensanchamiento de las bandas. Se mantiene constante el número de iteraciones Lanczos en 10000, y se varia el ensanchamiento de las bandas de absorción en 68 meV (5 mRy), 136 meV (10 mRy), 272 meV (20 mRy) y 544 meV (40 mRy). El OS para dichos cálculos es mostrado por una línea punteada morada.	38
3.11.	Comparación de espectros de absorbancia simulados para la estructura <i>SGN R2H4</i> . En (a), (b) y (c) se comparan los espectros de absorbancia simulados variando el ensanchamiento para las transiciones utilizando los funcionales de XC LDA, GGA y VdW, respectivamente. Los ensanchamientos utilizados fueron 68 meV (5 mRy), 136 meV (272 mRy) y 68 meV (5 mRy). Todas las simulaciones utilizaron 8000 iteraciones de Lanczos.	39
3.12.	Comparación de espectros de absorbancia simulados para la estructura <i>SGN R2H4</i> . Los espectros simulados utilizando los diferentes funcionales de XC se comparan bajo un mismo ensanchamiento y numero de iteraciones de Lanczos. Los ensanchamientos utilizados (de arriba hacia abajo) fueron 68 meV (5 mRy), 136 meV (272 mRy) y 68 meV (5 mRy). Todas las simulaciones utilizaron 10000 iteraciones de Lanczos.	40
3.13.	Comparación entre espectros de absorbancia experimentales y teóricos. En la inserción se puede observar un acercamiento a la banda asociada con las transiciones $\pi - \pi^*$ (CC). Los espectros simulados utilizaron 8000 iteraciones Lanczos.	41
3.14.	DoS y PDoS para la estructura <i>SGN R2H2</i> utilizando funcional de XC VdW. Los DoS y PDoS han sido desplazados en energía de tal manera que la energía de Fermi E_F se encuentra colocada en el origen del eje de las energías, esto es $E_F = 0$.	42

3.15. DoS y PDoS para estructura <i>SGN R2H2</i> obtenidos utilizando funcional de XC VdW. (a), (c) y (e) muestran los estados de valencia y conducción para las transiciones electrónicas que podrían dar origen a las bandas de absorción observadas, experimental y teóricamente, centradas alrededor de (d) 5,0 eV, (d) 3,7 eV y (f) 2,7 eV, respectivamente.	43
3.16. Bandas de absorción, DoS y PDoS para la estructura <i>SGO R3H2</i> relacionada con la transición $n - \pi^*$ (OC) obtenida utilizando GGA. (a) muestra los estados de valencia y conducción para la transición electrónica $\pi - \pi^*$ (CC). Se puede que los estados de valencia están dominados por estados π (C) y seguido de estados n (O). (b) muestra la banda de absorción relacionada con la transición electrónica señalada en su respectiva DoS.	44
3.17. Comparación entre espectros de absorción simulados utilizando funcional de XC VdW bajo cambios en los parámetros geométricos de la estructura. (a) comparación de los espectros simulados para las estructuras <i>SGN R2H2</i> y <i>R3H2</i> . (b) acercamiento en la comparación de la banda de absorción $\pi - \pi^*$ (CC) entre las estructuras <i>SGN R2H2</i> y <i>R3H2</i> . El QCE puede ser observado para esta banda al variar el radio de la estructura. (c) comparación de los espectros simulados para las estructuras <i>SGN R2H1</i> , <i>R2H2</i> , <i>R2H3</i> y <i>R2H4</i> . (d) acercamiento en la comparación de la banda de absorción $\pi - \pi^*$ (CC) entre las estructuras <i>SGN R2H1</i> , <i>R2H2</i> , <i>R2H3</i> y <i>R2H4</i> . Se puede observar una dependencia entre el centro de la banda de absorción $\pi - \pi^*$ (CC) y el número de placas de la estructura. La banda de absorción relacionada a la transición $\pi - \pi^*$ (CC) es señalada por la zona resaltada en naranja.	45
3.18. Respuesta electrónica obtenidas utilizando funcional de XC VdW para estructuras <i>SGN R2H3</i> . Respuesta electrónica para las estructuras <i>SGN</i> con (a) 1 placa (<i>R2H1</i>), (b) 2 placas (<i>R2H2</i>), (c) 3 placas (<i>R2H3</i>) y (d) 4 placas (<i>R2H4</i>) de grafeno dopado.	47
3.19. Bandas de absorción, DoS y PDoS para las estructuras <i>SGN R2H1</i> , <i>R2H2</i> y <i>R2H3</i> relacionadas con la transición $\pi - \pi^*$ (CC) obtenidas utilizando el funcional de XC VdW. (a), (a) y (e) muestran la contribución de los orbitales π (C) que cada capa de grafeno aporta a los estados de valencia y conducción para la transición electrónica $\pi - \pi^*$ (CC). (b), (d) y (f) muestran las bandas de absorción relacionadas con la transición electrónica $\pi - \pi^*$ (CC) para cada estructura.	48
3.20. Bandas de absorción, DoS y PDoS para la estructura <i>SGN R2H4</i> relacionadas con la transición $\pi - \pi^*$ (CC) obtenidas utilizando el funcional de XC VdW. (a) y (c) muestran la contribución de los orbitales π (C) que cada capa de grafeno aporta a los estados de valencia y conducción para la transición electrónica $\pi - \pi^*$ (CC). (b) y (d) muestran las bandas de absorción relacionadas con cada transición electrónica $\pi - \pi^*$ (CC) mencionada previamente.	49
3.21. Modelo propuesto para explicar el comportamiento de los espectros de absorción con el número de placas de la estructura. (a) Diagrama de energías para el modelo semi-empírico propuesto. En este se observa como la superposición de las energías potenciales de las placas de grafeno individuales genera una energía potencial modificada para toda la estructura. (b) Funciones semi-empíricas para las energías potenciales entre las placas de grafeno propuestas por Chen et. al. [67].	50
3.22. Modelo semi-empírico aplicado a la estructura <i>R2H1</i> . Transición del último estado π (C) ocupado (HOMO) al primer estado excitado π^* (C) (LUMO).	51
3.23. Resultados del modelo semi-empírico propuesto para la estructura <i>R2H2</i>	52
3.24. Resultados modelos semi-empírico para estructura <i>R2H3</i>	53
3.25. Resultados modelos semi-empírico para estructura <i>R2H4</i>	54
3.26. Niveles de energía obtenidos con el modelo semi-empírico para las estructuras <i>R2HY</i> . . .	54

Resumen

Los puntos cuánticos de carbono de grafeno (GCQD) son nanoestructuras de carbono, conformados por varias capas de grafeno apiladas [1-8]. Los GCQDs han atraído mucha atención debido a sus propiedades únicas y diversas aplicaciones tecnológicas, como por ejemplo: baterías, etiquetas biológicas, condensadores, celdas solares, diodos emisores de luz *Light Emitting Diodes* (LEDs), entre otros [7, 8]. Todas estas tecnologías podría beneficiarse de la baja toxicidad, estabilidad química y física de estas nanoestructuras [8, 9].

A pesar de muchas investigaciones realizadas en GCQDs, muchas propiedades optoelectrónicas aun no son comprendidas por completo [8, 10-12]. Entre estas propiedades, la absorción y la fotoluminiscencia podrían ser mencionadas. Los espectros de absorción de los GCQDs muestran diferentes bandas. El origen de estas bandas ha sido obtenido a través de métodos semi-empíricos, como la regla de Woodward-Fieser, o modelos de primeros principios, muchas veces referidos como modelos *ab-initio*, en hidrocarburos policíclicos aromáticos, conocidos también por su nombre en ingles *Polycyclic Aromatic Hydrocarbons* (PAHs) [13-15]. Estos modelos no toman en cuenta características relevantes de estas nanoestructuras como lo son el dopamiento de átomos a las placas de grafeno, la naturaleza multi-capas de estas nanoestructuras y las interacciones de Van der Waals (VdW) que ocurren en este tipo de sistemas.

Con el propósito de estudiar el origen de estas bandas de absorción y el efecto que producen las interacciones de VdW y el número de placas en la estructura sobre la absorción, se han estudiado por medio de espectroscopia óptica muestras de GCQDs. Dichas muestras han sido sintetizadas por medio de la pirólisis de Ácido Cítrico, *Citric Acid* (CA), obteniéndose estructuras con un radio de aproximado de 1 nm y una altura de 1,4 nm (lo cual equivale alrededor de 4 placas de grafeno apiladas) [1-4, 6, 7]. Para ayudar al comprender las propiedades de absorción de los GCQDs, se realizaron cálculos utilizando la teoría del funcional de densidad, mejor conocida por su nombre en ingles *Density Functional Theory* (DFT) bajo 3 diferentes aproximaciones de funcionales de intercambio-correlación, también conocido como *eXchange-Correlation* (XC). Dichos funcionales de XC son la aproximación local de densidad, *Local Density Aproximation* (LDA), aproximación de gradiente generalizada, *General Gradient Aproximation* (GGA) y el funcional de densidad Van der Waals (VdW) los cuales son aproximaciones que tratan de reproducir interacciones locales, semi-locales y no-locales, respectivamente [16-18]. Los cálculos de DFT realizados en este trabajo utilizaron el paquete computacional de *Quantum-Espresso* [19, 20]. El origen de las bandas de absorción se estudio a partir de cálculos de DoS, PDoS y simulaciones de los espectros de absorción de los GCQDs. Para el estudio del efecto que las múltiples capas de grafeno producen en las propiedades de absorción en estas nanoestructuras, se realizaron cálculos de respuesta electrónica y se analizo la contribución que cada placa de grafeno aporta a la DoS y PDoS total de la estructura. Para corroborar los resultados computacionales se planteo un modelo semi-empírico para explicar la transiciones electrónicas en las estructuras de GCQDs tomando en cuenta las interacciones de VdW presentes entre las placas de grafeno.

Los espectros experimentales, obtenidos de la muestras sintetizadas, concordaban con los espectros simulados. Las simulaciones utilizando el funcional XC de VdW son los resultados que mejor concuerdan con los experimentos. Entre los acuerdos entre experimento/teoría podemos mencionar la existencia de tres bandas de absorción, centradas alrededor de 2,7 eV (459 nm), 3,7 eV (335 nm) y 5,0 eV (248 nm), y la intensidad relativa entre estas bandas de absorción. Los resultados de DoS y PDoS podrían ser interpretados como transiciones $\sigma - \pi^*$ (NC), $\sigma - \pi^*$ (OC), $\pi - \pi^*$ (CC). A partir de los resultados del modelo semi-empírico, propuesto en este trabajo, junto con los cálculos de respuesta electrónica y DoS por placa, se podrían concluir que las interacciones de VdW alteran las propiedades de absorción de la banda asociada a la transición $\pi - \pi^*$ (CC).

Introducción

Los puntos cuánticos, también referidos como *Quantum Dots* (QDs), son estructuras formadas por una colección de átomos en diversas geometrías las cuales rondan las dimensiones de varios o cientos de nanómetros [21]. Estas estructuras presentan propiedades electrónicas y ópticas diferentes a sus estados tridimensionales [22]. Debido a sus propiedades especiales, los QDs han sido aplicados en diversos dispositivos, entre los cuales se pueden mencionar: baterías, bio-marcadores, celdas solares y sensores electrónicos u ópticos [23-26].

Muchas de las novedosas aplicaciones de los QDs, han utilizado átomos metálicos o semiconductores para la síntesis de estas estructuras [26-28]. Sin embargo, los QDs metálicos o semiconductores presentan una gran toxicidad [29]. Por lo tanto, sería deseable obtener nano-estructuras que posean las propiedades electrónicas y ópticas características de los QDs metálicos o semiconductores pero sin su alta toxicidad. Los puntos cuánticos de carbono, también referidos como *Carbon Quantum Dots*(CQDs), son una alternativa que podría solventar la problemática de la toxicidad. Esto se debe, a que los CQDs presentan propiedades electrónicas y ópticas similares a los QDs metálicos o semiconductores pero sin su alta toxicidad [9, 30]. A pesar de las ventajas que poseen los CQDs, el avance de sus aplicaciones se ve dificultado experimentalmente en su proceso de síntesis, así como por propiedades únicas las cuales no están completamente explicadas [10-12]. Entre estas propiedades sin comprender podemos mencionar, las múltiples bandas de absorción observadas y los confusos mecanismos de fotoluminiscencia que presentan estas estructuras [8, 10-12]

Para aportar a la discusión de las propiedad de absorción de los GCQDs, se estudiaran muestras sintetizadas a partir de ácido cítrico por medio de espectroscopia óptica. Además para ayudar a la interpretación de los resultados experimentales, se realizaron cálculos utilizando la teoría del funcional de densidad y la teoría funcional de densidad dependiente del tiempo, a las cuales también se les nombra como *Density Functional Theory*(DFT) y *Time Dependent Density Functional Theory*(TDDFT), respectivamente [31-37]. Para la realización de estos cálculos se utilizó el paquete computacional *open-Source Package for Research in Electronic Structure, Simulation, and Optimization*, cuyo acrónimo es *Quantum ESPRESSO* (QE) [19, 20, 38-43]. A partir de estos experimentos y cálculos se espera comprender el efecto de las interacciones de VdW los átomos dopantes y el número de placas de la estructura sobre las propiedades de absorción de los GCQDs.

El presente trabajo está dividido en 4 capítulos. En el primer capítulo se describirá la teoría básica relacionada a los GCQDs. En el capítulo 2, se introducirá toda la metodología para la síntesis y realización de los cálculos presentados en este trabajo. La presentación de los resultados y su respectiva discusión se realizará en el capítulo 3. Finalmente en el capítulo 4, se sugerirán recomendaciones, se propondrán posibles trabajos a futuro y se darán las conclusiones de esta investigación.

Capítulo 1

Fundamento teórico

En el presente capítulo se introduce de manera resumida la teoría básica utilizada para la realización de este trabajo. Primeramente, se detallarán los fenómenos de absorbancia y luminiscencia en la materia. Luego se describirán las estructuras conocidas como puntos cuánticos. Finalmente, se abordará la descripción del material a estudiar y los últimos avances obtenidos en el estudio de los puntos cuánticos de carbono.

1.1. Interacción de la radiación electromagnética con la materia

Cuando una onda electromagnética incide sobre la materia diferentes fenómenos físicos pueden ocurrir. Entre tales fenómenos físicos se pueden mencionar la absorción, la luminiscencia y la dispersión. En la figura 2.2 se puede observar un diagrama de los fenómenos mencionados.

1.1.1. Absorción

Existen varios modelos para la descripción de la absorción de la radiación electromagnética en la materia. Entre dichos modelos se puede mencionar la ley de Beer-Lambert-Bouguer, el modelo de oscilador forzado y la fuerza del oscilador.

1.1.1.1. Ley de Beer-Lambert-Bouguer

La ley de Beer-Lambert-Bouguer es un enfoque fenomenológico muy utilizado en muestras líquidas y gaseosas de bajas concentraciones u muestras sólidas de poco espesor.

Esta ley propone que una onda electromagnética al penetrar en la materia disminuirá su intensidad de manera exponencial en función de la distancia penetrada dentro de la muestra y la concentración de esta. En la figura 1.2 se detalla un esquema para explicar la relación. En dicho esquema un haz de radiación electromagnética con intensidad I_0 incide sobre una muestra de espesor b y superficie S . Al atravesar la muestra el haz de radiación electromagnética ha disminuido su intensidad a I .

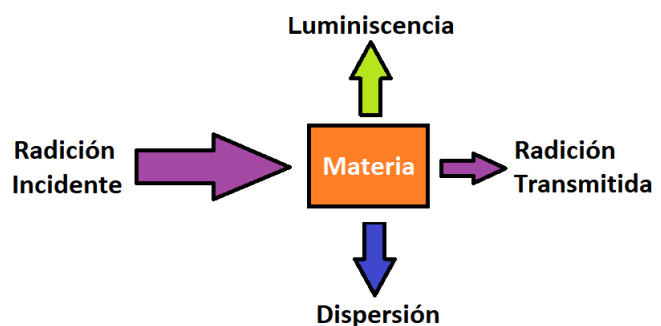


Figura 1.1: Diagrama de los principales fenómenos físicos que ocurren debido a la interacción entre radiación electromagnética y la materia.

Esta relación de decrecimiento exponencial puede ser convertida en una ecuación introduciendo un constante de decrecimiento. Tal constante de decrecimiento, muchas veces referida como sección eficaz $\sigma(\lambda)$, modula que tanta radiación electromagnética, de una longitud de onda determinada, es absorbida por la muestra. Con la introducción de la sección eficaz $\sigma(\lambda)$ la ley de Beer-Lambert-Bouguer puede ser escrita como:

$$I(\lambda, c, x) = I_0 e^{-\sigma(\lambda)cx} \quad (1.1)$$

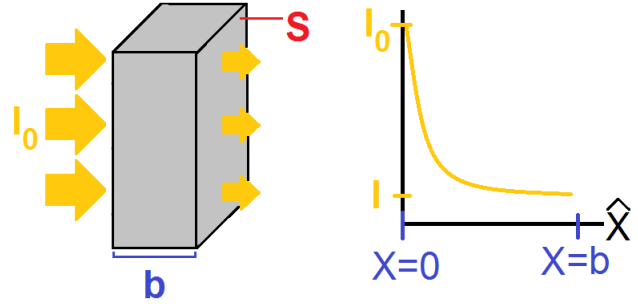


Figura 1.2: Esquema de la ley de Beer-Lambert-Bouguer. Radiación con intensidad I_0 incide en una muestra espesor b y superficie S . La Radiación I , la cual a penetrado una distancia x en la muestra, cumple con una atenuación exponencial.

1.1.1.2. Modelo de oscilador armónico forzado

Este modelo consiste en suponer que un haz de radiación electromagnética impulsa a los electrones, del material irradiado, a una oscilación armónica forzada descrita por la ecuación:

$$0 = m \frac{d^2x}{dt^2} + b \frac{dx}{dt} + Dx - qE_0 e^{i\omega t} \quad (1.2)$$

donde la solución para la amplitud del movimiento descrita por la ecuación 1.2 viene dada por:

$$x_0 = \frac{qE_0}{m(\omega_0^2 - \omega^2 + i\gamma\omega)} \quad (1.3)$$

Las constantes introducidas γ y ω_0 están definida mediante las igualdades $\gamma = \frac{b}{m}$ y $\omega_0 = \frac{D}{m}$. Tal movimiento origina pequeños dipolos dentro del material. Según la teoría electromagnética, la polarización \vec{P} de un material esta relacionada con la constante dieléctrica relativa ϵ . A su vez, la constante dieléctrica relativa ϵ esta relacionada con el índice de refracción n de un material. Utilizando en estas relaciones mencionadas es posible relacionar la amplitud de la oscilación armónica forzada x_0 , dada en la ecuación 1.3, con el índice de refracción n . Dicha relación nos permite escribir al índice de refracción n como una cantidad compleja dada por:

$$n = n' + i\kappa \quad (1.4)$$

$$n' = \frac{Nq^2}{2\epsilon_0 m} \frac{\gamma\omega}{(\omega_0^2 - \omega^2)^2 + \gamma^2\omega^2} \quad (1.5)$$

$$\kappa = 1 + \frac{Nq^2}{2\epsilon_0 m} \frac{(\omega_0^2 - \omega^2)}{(\omega_0^2 - \omega^2)^2 + \gamma^2\omega^2} \quad (1.6)$$

La parte real de la constante de refracción n es relacionada con la dispersividad del material. Mientras la parte imaginaria de la constante de refracción n es relacionada la absorción del material. Debido a esto las ecuaciones 1.4, 1.5 y 1.6 son llamadas relaciones de dispersión. El modelo descrito previamente ha sido un enfoque clásico, sin embargo es posible realizar los cálculos previos utilizando la mecánica cuántica y obtenerse relaciones muy similares al caso clásico.

1.1.1.3. Fuerza del oscilador

En un sistema, ya sea un átomo, molécula o sólido, existen una cantidad de estados electrónicos ocupados y desocupados. En donde los estados ocupados poseen energías mas bajas que lo estados ocupados. Cuando el sistema es excitado, por ejemplo al ocurrir la absorción de un fotón, no todas las transiciones electrónicas de un estado ocupado a un estado desocupado son posibles. Para cuantificar la posibilidad que una transición en particular ocurra se ha definido, utilizando el modelo del oscilador armónico forzado, la cantidad conocida como Fuerza del Oscilador, o por sus termino en ingles de *Oscillator Strength* (OS).

El OS se define a partir de la siguiente manera: En un sistema el cual ha absorbido energía, se define el OS como la probabilidad f_{ik} de que un sistema sufra una transición de un estado electrónico inicial ocupado de energía E_i a un estado final desocupado E_k . Por lo tanto si sumamos las probabilidades de todas las posibles transiciones f_{ik} se obtendría:

$$\sum_k f_{ik} = 1 \quad (1.7)$$

Modificando las relaciones de dispersión, ecuaciones 1.4, 1.5 y 1.6, a partir de la definición de OS, ecuación 1.7, se obtienen las siguientes relaciones de dispersión modificadas:

$$n_i = n'_i + \nu\kappa_i \quad (1.8)$$

$$n'_i = \frac{N_i q^2}{2\epsilon_0 m} \frac{\gamma f_{ik} \omega_{ik}}{(\omega_{ik}^2 - \omega^2)^2 + \gamma_{ik}^2 \omega^2} \quad (1.9)$$

$$\kappa_i = 1 + \frac{N_i q^2}{2\epsilon_0 m} \frac{(\omega_{ik}^2 - \omega^2) f_{ik}}{(\omega_{ik}^2 - \omega^2)^2 + \gamma_{ik}^2 \omega^2} \quad (1.10)$$

Donde N_i donde es la densidad de átomos por volumen que se encuentran en el estado energético E_i . Cerca de una línea de absorción, $|w_0 - w| \ll w_0$, ecuaciones 1.8, 1.9 y 1.10 se reducen a las ecuaciones 1.4, 1.5 y 1.6. El termino γ_{ik}^2 representa el ensanchamiento de las líneas espectrales, y en espectroscopia recibe el nombre de Ancho Completo en la Mitad del Máximo, o también llamado *Full Width at Half Maximum* (FWHM) por su terminología en ingles.

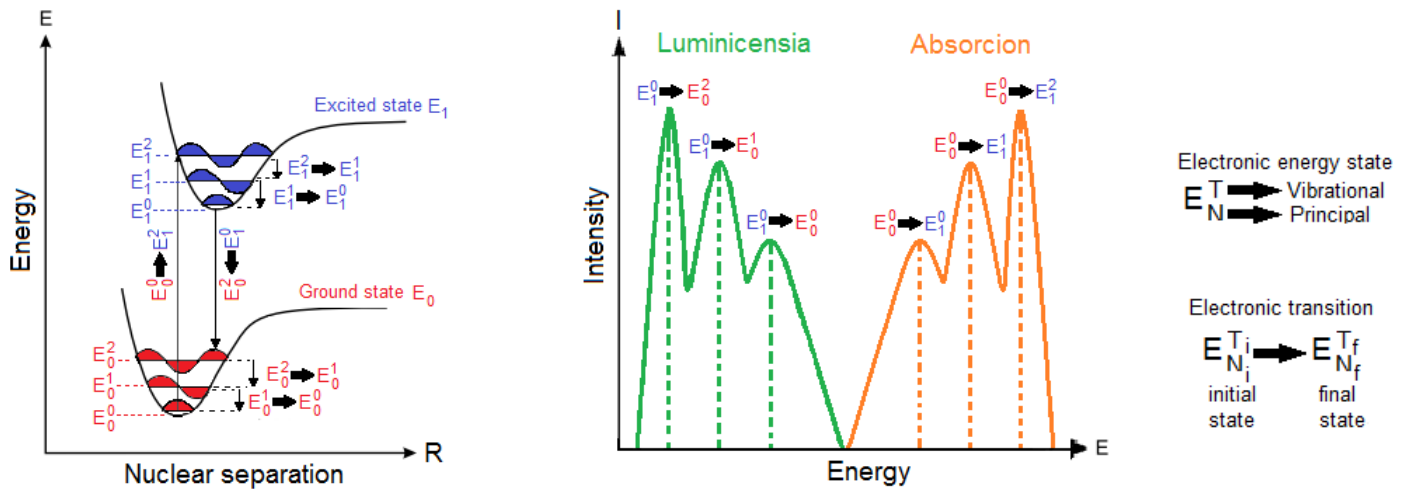
1.1.2. Luminiscencia

La luminiscencia ocurre debido a una transición electrónica de un estado de mayor energía E_k a un estado de menor energía E_i . Dicha transición ocurre luego de transcurrido un tiempo t conocido como tiempo de vida del estado excitado t_τ . Dependiendo del sistema, diferentes procesos físicos pueden ocurrir durante dicha transición electrónica. Durantes tales procesos la diferencia de energía $\Delta E = E_k - E_i$ puede ser transformada en energía calorífica, energía vibracional y radiación electromagnética.

Para sistemas monoatómicos, en general, las bandas de los espectros de absorción y luminiscencia coinciden. Dependiendo de las características del sistema monoatómico, ciertas transiciones son mas probables, mientras otras son menos probables. Además, existen ciertas transiciones tan poco probables que han sido denominadas transiciones prohibidas. Dependiendo que tan probable sea una transición, así aumentara o disminuirá la intensidad de una banda de luminiscencia respecto a su respectiva banda de absorción. Por otra parte, en los sistemas moleculares y cristalinos de muchos átomos los procesos de luminiscencia son mas complejos.

El principio de Franck-Condon, el cual es esquematizado en la figura 1.3, es un modelo para explicar las transiciones verticales de absorción y luminiscencia para dichos sistemas de muchos átomos. En la

figura 1.3 (a) se observa el cambio de la energía potencial electrostática respecto a la separación nuclear en un sistemas de varios átomos. Estos estados energéticos principales E_0 y E_1 representan el estado base y el primer estado excitado respectivamente. Para cada estado principal E_N , donde N designa el estado principal 1 u 2, se muestran por simplicidad tres estados vibracionales E_N^T , donde T designa los estados vibracionales E_N^0 , E_N^1 y E_N^2 . Debido al solapamiento entre las funciones de onda del estado E_0^0 y E_1^2 , la transición vertical de excitación (absorción) $E_0^0 \rightarrow E_1^2$ es favorecida. Sin embargo una vez en el estado E_1^2 , el sistema tiene la opción de relajarse mediante vibraciones originando así las transiciones oscuras $E_1^2 \rightarrow E_1^1$ y $E_1^1 \rightarrow E_1^0$. Estas transiciones son llamadas oscuras debido a que se emite un fonón en lugar de un fotón.



(a) Transiciones verticales bajo el principio de Franck-Condon

(b) Espectro de absorción y luminiscencia para el estado base y primer estado excitado.

Figura 1.3: Principio de Franck-Condon y su efecto en los espectros de absorción y luminiscencia. (a) Transiciones verticales de absorción y luminiscencia favorecidas. (b) Stokes Shift entre el espectro de absorción y luminiscencia.

Una vez que el sistema se ha relajado al estado E_1^0 al no haber mas estados vibracionales disponibles para el estado principal E_1 en los cuales perder energía, el sistema lleva a cabo una desexcitación (luminiscencia) al estado principal E_0 emitiendo un foón en el proceso. Dependiendo del solapamiento entre el estado inicial E_1^0 y el estado final E_0^T , la intensidad de las bandas de luminiscencia se vera afectada. En la figura 1.3 (b) se observan la intensidad de las bandas de absorción y luminiscencia contra la energía de las transiciones verticales que las originan, mostradas previamente en la figura 1.3 (a). El espectro de luminiscencia se ha desplazado a energías menores respecto al espectro de absorción. Esto es originado por la perdida de energía en la creación de los fonones en el sistema. Este fenómeno es conocido como el desplazamiento de Stokes, o también llamado *Stokes Shift* (SS) por su terminología en ingles.

1.2. Puntos cuánticos

Se denominan puntos cuánticos, también referidos en inglés como *Quantum Dots* (QDs), a estructuras multi-atómicas en las cuales los portadores de carga han sido confinados en sus tres dimensiones espaciales [21, 22, 27]. El confinamiento espacial de los portadores de carga origina una redistribución de la densidad de estados, también referida en inglés como *Density Of States* (DOS). La DOS de un material indica cuantos portadores de carga pueden permanecer simultáneamente a una energía determinada.

En la figura 1.4 se puede observar el proceso de confinamiento espacial sobre un material. Un material tridimensional sin restricciones espaciales, al cual también se le refiere por su nombre en inglés *bulk*, es mostrado con su DOS en las figuras 1.4 (a) y 1.4 (e) respectivamente. Para el caso *bulk* los portadores de carga pueden encontrarse en un continuo de energías. Esto es debido a que los portadores de carga no poseen ninguna restricción espacial. Para los casos de 1 y 2 dimensiones confinadas, figuras 1.4 (b) y 1.4 (c) respectivamente, se aprecia una mayor prohibición en las energías permitidas para los portadores de carga, figuras 1.4 (f) y 1.4 (g) respectivamente, debido a un mayor confinamiento espacial. Al ocurrir una restricción espacial completa, figura 1.4 (d), se obtiene un QD. En la figura 1.4 (h), se puede apreciar que para un QD la DOS consta de estados energéticos discretos. Dicho comportamiento de la DOS en los QDs es similar a la DOS en los átomos.

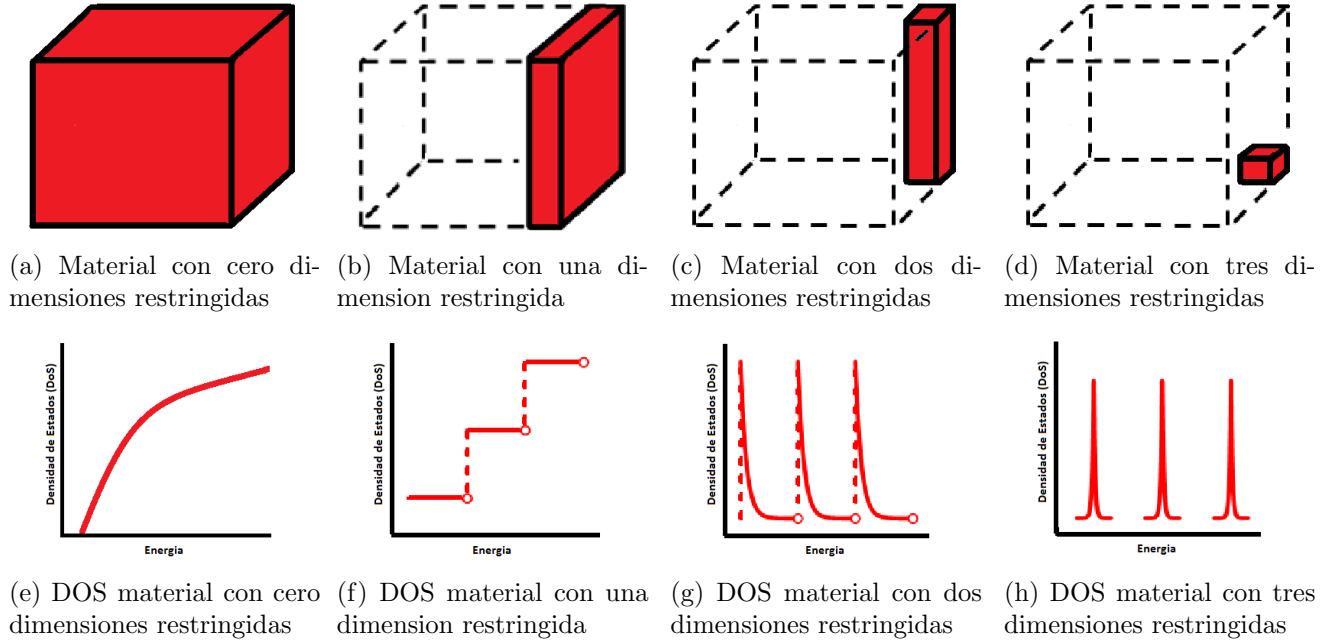


Figura 1.4: Representación de un material y la densidad de estados correspondiente para diferentes niveles de restricción espacial. En (a) y (e) se observa el caso para *bulk*. Las restricciones espaciales sobre una y dos dimensiones se muestran en (b) y (f), junto con (c) y (g) describen los casos de *Quantum Wells* y *Quantum Wires* correspondientemente. Mientras el caso de restricción espacial total, esto sería un QDs, se observa en (d) y (h).

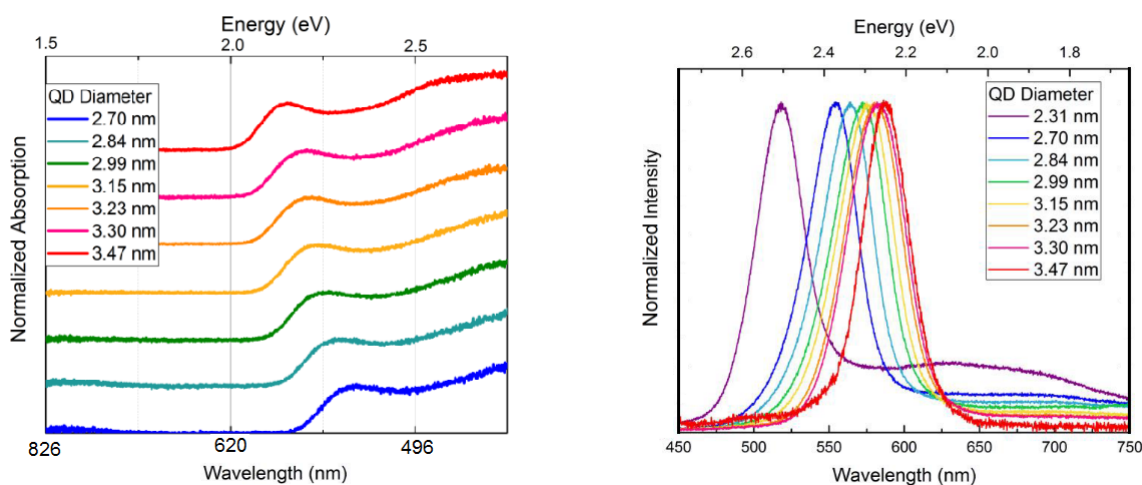
Como consecuencia de un mayor confinamiento de los portadores de carga, los QDs presentan propiedades diferentes que en su estado *bulk*. Los QDs presentan la capacidad de alterar sus propiedades electrónicas y ópticas a través de cambios en el tamaño de la estructura [22, 27]. Sin embargo las propiedades electrónicas y ópticas de los QDs no solo se ven modificadas por el tamaño de la estructura. Estas propiedades también se ven alteradas dependiendo de los átomos que conforman la estructura, el método de síntesis utilizado, los enlaces formados y los defectos de red presentes en la superficie de QD [21, 27]. Históricamente, los primeros QDs sintetizados fueron a partir de elementos metálicos o semiconductores. Posteriormente fueron descubiertos QDs formados por otro tipo de elementos, como es carbono.

Existe un gran interés, tanto teórico como práctico, por estudiar las propiedades de los QDs. Desde la perspectiva de la física fundamental, los QDs presentan fenómenos únicos los cuales nos permitirán mejorar la comprensión actual de los sistemas de bajas dimensiones. En cambio desde un punto de física aplicada, las propiedades electrónicas y ópticas únicas de los QDs podrían permitir mejorar diversas tecnologías existentes. Entre las aplicaciones tecnológicas de los QDs pueden mencionarse: baterías, bio-marcadores, celdas solares, LEDs, sensores, entre otras [23-26].

1.2.1. Puntos cuánticos semiconductores

Los QDs metálicos o semimetálicos son sintetizados a partir de una gran variedad de elementos, entre los cuales podemos mencionar Cadmio, Plomo, Selenio y Telurio. Los QDs metálicos y semiconductores poseen una alta toxicidad, debido a los elementos utilizados para su síntesis [29]. Por lo tanto una de las problemáticas a resolver para las tecnologías con base en QDs metálicos o semiconductores, son los riesgos a la salud que estas estructuras presentan.

A pesar de los riesgos ambientales que estas estructuras presentan, los QDs metálicos o semiconductores ha sido objeto de mucho estudio por sus atractivas propiedades electrónicas y ópticas. Dentro de dichas propiedades, se puede mencionar la dependencia de las propiedades ópticas de los QDs con el tamaño de la estructura. En la figura 1.5, se puede observar un corrimiento en los del espectros de absorción y luminiscencia a medida que la estructura aumenta de tamaño. Este comportamiento indica que la absorción y luminiscencia de los QDs puede ser modificada controlando el tamaño de la estructura [23-26].



(a) Dependencia del espectro de absorción con el tamaño de la estructura.

(b) Dependencia del espectro de luminiscencia con el tamaño de la estructura.

Figura 1.5: Dependencia de los espectros de absorción (a) y luminiscencia (b) respecto al tamaño de la nanoestructura para QDs de CdSe [44].

1.2.2. Descripción teórica de los puntos cuánticos

Los QDs presentan características distintas a sus materiales en bulk. por la tanto se han planteado diversos modelos para la descripción teórica de estas nanoestructuras. Dos enfoques muy utilizados para la descripción teórica de los QDs son: el pozo cuadrado de potencial finito y el la problema de muchos cuerpos.

1.2.2.1. Pozo cuadrado de potencial finito

Uno de los modelos mas utilizado para la descripción teórica de los QDs es el modelo de una partícula atrapada en un pozo de potencial finito [28]. En este modelo podemos simular el efecto del QD sobre los portadores de carga como un potencial ligante cuadrado finito. La modificación del tamaño de la estructura se puede comprender como un cambio en la anchura del pozo de potencial. Esto obliga que los niveles de energía permitidos para el sistema se encuentren mas separados entre menor sea el tamaño de la estructura. A su vez la función de onda que representa al portador de carga se deslocalizada a medida que el tamaño de la estructura disminuye. A este fenómeno se le nombro Efecto de Confinamiento

Cuántico, también referido en inglés como *Quantum Confinement Effect* (QCE). En la figura 1.6 se muestra una representación esquemática del QCE. Tres pozos de la misma profundidad de potencial son comparados cada uno con diferente anchura y a medida que el pozo disminuye de tamaño la función de onda del portador de carga se deslocaliza mas. Esto se encuentra representado en la figura 1.6 (a).

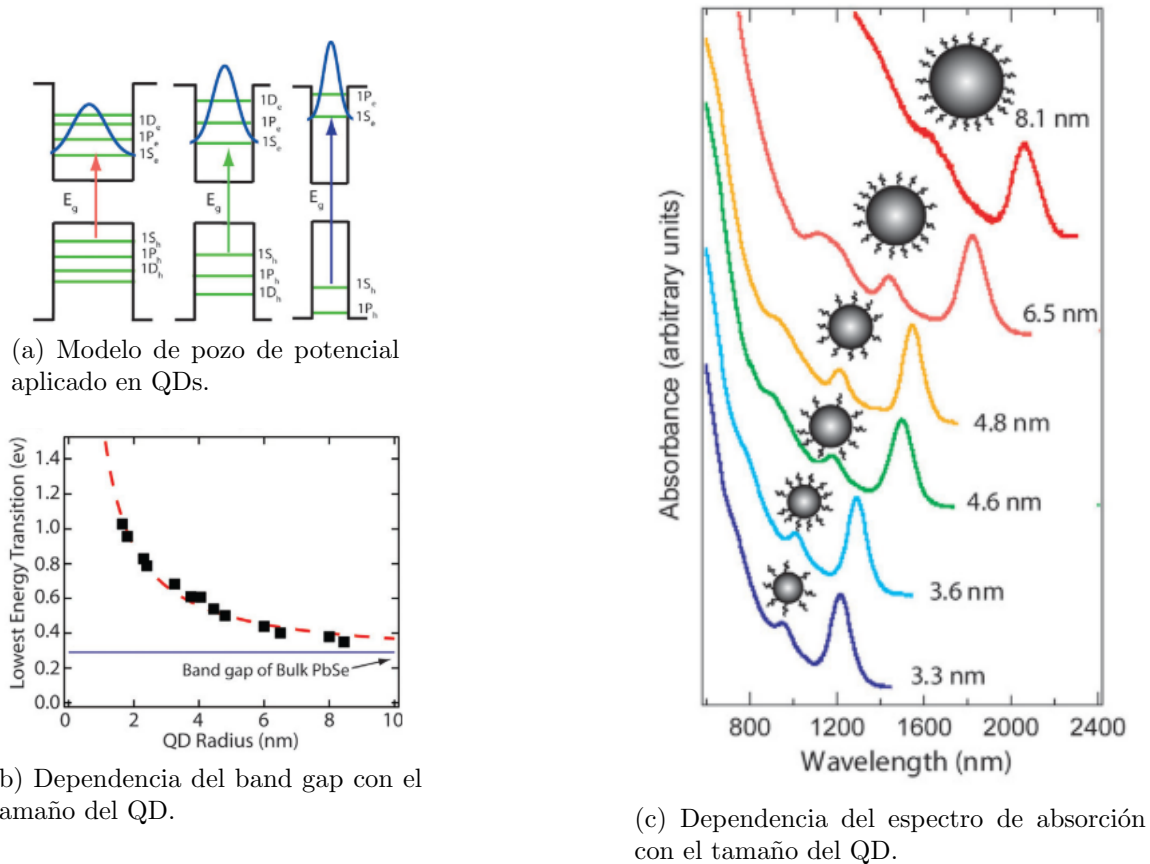


Figura 1.6: Modelo de pozo de potencial aplicado para explicar las propiedades de los QDs de PbSe. (a) esquema de niveles de energía y funciones de onda para modelar el confinamiento de QDs de diferentes tamaños. (c) espectros de absorción para QDs de PbSe y su dependencia con el tamaño de la estructura. (b) comportamiento de la energía del band edge respecto al cambio de tamaño en QDs de PbSe. [28].

Debido al QCE que sufren los portadores de carga, las propiedades electrónicas de la estructura son alteradas al modificar el tamaño del QD [22, 27, 28]. Una de las propiedades modificadas mediante un cambio de tamaño es el espectro de absorción [28]. En la figura 1.6 (c) se puede observar como los espectros de absorción para QDs de PbSe dependen del radio de la estructura. A medida que el radio de los QDs disminuye, la banda de absorción se corre a menores longitudes de onda menores o su vez a mayores energías. Además los valores del band gap medido en los QDs difiere del valor medido para en su estado bulk, como puede ser observado para el caso de los QDs de PbSe en la figura 1.6 (b).

1.2.2.2. Problema de muchos cuerpos

El modelo de pozo cuadrado de potencial finito es muy útil para describir de manera cualitativa los QDs. Sin embargo, el modelo no es muy adecuado para obtener una descripción cuantitativa de estas nanoestructuras. Una de las principales características del modelo de pozo cuadrado fue tomar el potencial ligante como un pozo cuadrado finito. En la realidad un sistemas de muchos átomos no poseería un potencial ligante tan sencillo.

Por lo tanto se necesita un modelo que permita obtener resultados cuantitativos que concuerden con las mediciones experimentales. Para mejorar la descripción de los QDs se debería de tomar en cuenta los potenciales ligantes que ejerce cada núcleo a todos los electrones del sistema. Además, se debe de tomar en cuenta la repulsión núcleo-núcleo y electrón-electrón presentes en el sistema. En la figura 1.7 se muestra un esquema de la interacción entre dos núcleos, etiquetados con I y J , y dos electrones, etiquetados con i y j .

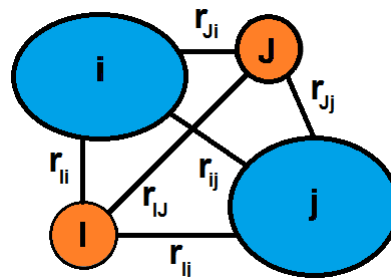


Figura 1.7: Diagrama de las múltiples interacciones en un sistema de muchos cuerpos. Por simplicidad solo se muestran la interacción de los núcleos atómicos I y J con las nubes electrónicas i y j .

Entre mayor sea el número de átomos, por ende de número de núcleos y electrones, más interacciones entre las diversas partículas debe de tomarse en cuenta. Un groso estimado indica que el número de interacciones ronda alrededor de $2*(N-1)! + N!$ donde N es el número de átomos en el sistema. Debido a la facilidad con la que la complejidad de estos cálculos incrementa al crecer el número de partículas analizadas, diferentes aproximaciones y técnicas computacionales han sido diseñadas para resolver de manera aproximada este tipo de interacciones. Entre el repertorio de opciones para resolver problemas de muchos cuerpo se encuentran la aproximación $K\dot{P}$, la aproximación del enlace fuerte también referida en inglés como *Tight-Binding* (TB), método autoconsistente Hartree-Fock (HF) y Teoría del Funcional de Densidad también referida en inglés como *Density Functional Theory* (DFT). Cada uno de estos diferentes acercamientos a la problemática de muchos cuerpos posee sus ventajas y desventajas para la resolución de dicho problema.

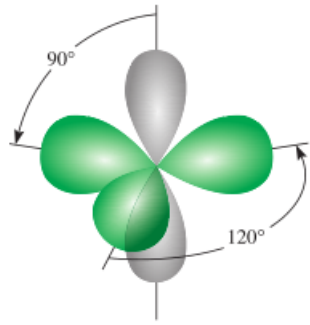
1.3. Estado del conocimiento

En esta sección se presentará el estado del conocimiento actual sobre los puntos cuánticos de carbono grafeno. Primero, se detallarán las propiedades electrónicas del carbono y la red cristalina en la cual se organiza dicho elemento. Luego, se describirá el sistema a estudiar, sus aplicaciones y el *gap* en el conocimiento.

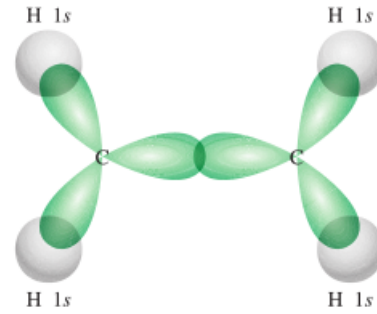
1.3.1. Propiedades electrónicas del carbono

El carbono (C) es el sexto elemento de la tabla periódica, con una masa de $12,0111u$, un radio atómico promedio de $70 pm$ y un radio de Van der Waals de $170 pm$. La configuración electrónica del C es: $[He] 2S^2 2P^4$. Tal configuración electrónica le permite al C hibridarse en las configuraciones SP^1 , SP^2 y SP^3 . Para formar un enlace entre dos átomos de C a partir de estas hibridaciones, se debe de dar un solapamiento entre las funciones de onda que describen cada hibridación [45]. Si el solapamiento es frontal, al enlace se le denomina enlace σ . Mientras si el solapamiento es lateral, al enlace se le denomina enlace π . En la figura 1.8 se puede observar como se forma un enlace σ y un enlace π para los carbonos de la molécula de etileno (C_2H_4). Cada carbono se encuentra en una hibridación SP^2 , utilizando sus orbitales $2S$ y $2p_x$ y $2p_y$ para el enlace σ y su orbital restante $2p_z$ para el enlace π .

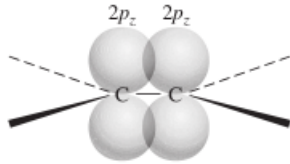
Los formas sólidas, también referidas como alótropos, más comunes del C son el grafito y el diamante. En la figura 1.9 se muestran las celdas unitarias de ambos cristales. El diamante, con hibridación SP^3 , posee una estructura cristalina cúbica con un parámetro de red de $3,57 \text{ \AA}$. Mientras, el grafito, con hibridación SP^2 , ocurre en dos fases cristalinas: hexagonal (α), también conocida como fase Bernal, y



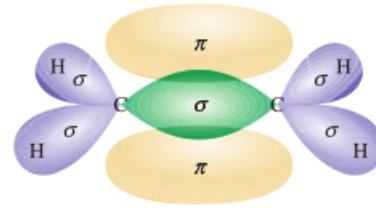
(a) Hibridación SP^2 del C .



(b) Enlace σ entre átomos de C .



(c) Enlace π entre átomos de C .



(d) Enlace doble formado por el enlace σ y π .

Figura 1.8: Representaciones de (a) la hibridación SP^2 , (b) del solapamiento en un enlace σ , (c) de solapamiento en un enlace π , y (d) y la formación de un enlace doble de etileno por medio de los enlaces σ y π [45].

romboédrica (β). Ambas con un parámetro de red de $1,42 \text{ \AA}$ y una separación entre planos de $3,35 \text{ \AA}$ [46]. La diferencia entre ambas fases es el apilamiento de los diferentes planos de carbono. La fase α posee un apilamiento ABA , mientras el apilamiento en la fase β es ABC [46].

Un plano individual de grafito es conocido como *grafeno*. En la figura 1.10 se comparan las redes cristalinas del grafito y grafito β . Por otra parte, en la figura 1.11 se comparan las estructura de bandas para el grafito y grafito β cerca del punto K .

1.3.2. Puntos cuánticos de carbono

A partir de mediciones experimentales utilizando diferentes técnicas como Microscopía de fuerza atómica (AFM), por sus siglas en inglés *Atomic Force Microscopy*, Microscopía Electrónica de Transmisión (TEM), por sus siglas en inglés *Transmission Electronic Microscopy*, Microscopía Electrónica de Barrido (SEM), por sus siglas en inglés *Scanning Electronic Microscopy*, Cristalografía de rayos X (XRD), por sus siglas en inglés *X-Ray Diffraction*, entre otras, se ha obtenido información de la geometría presente en los CQDs [6, 7, 48, 49]. Generalmente se presentan dos tipos de estructuras en los CQDs. La primera posee geometría esférica, mientras la segunda posee geometría cilíndrica [6, 48, 49]. El ordenamiento de ambas geometrías presentes en los CQDs puede ser explicado a partir del apilamiento de placas de grafito [6, 7, 48, 49].

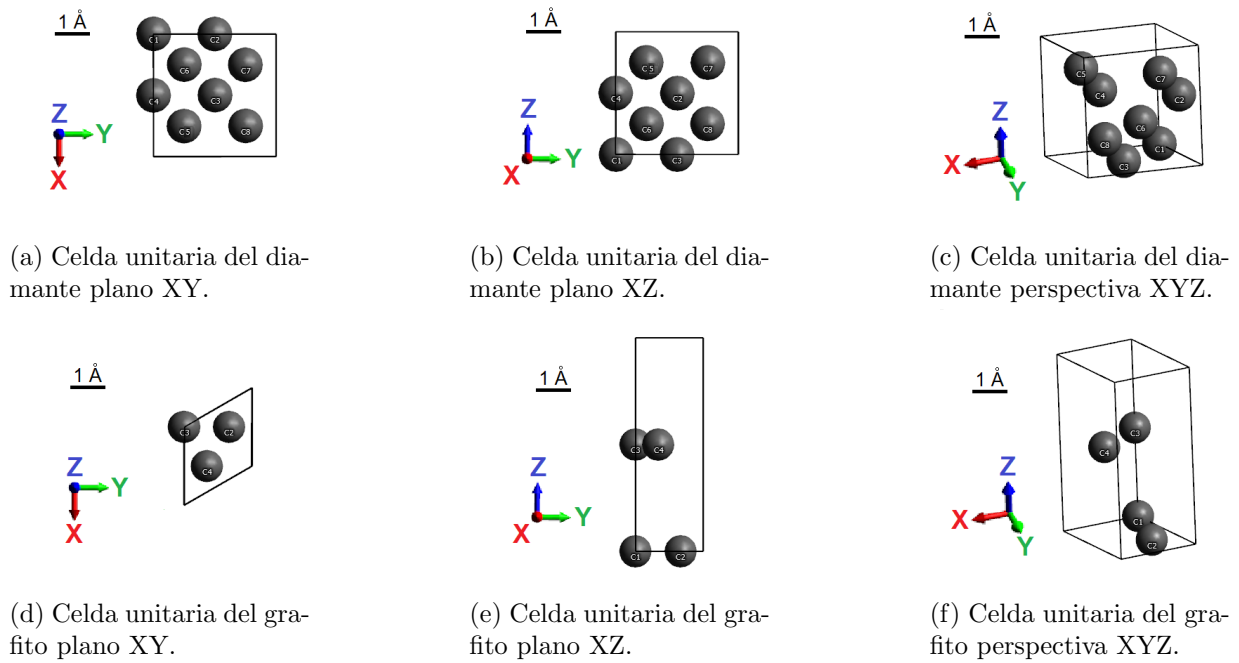


Figura 1.9: Representación de las celdas unitarias para el diamante y grafito. La dirección de los ejes X, Y y Z están dadas por las flechas roja, verde y azul respectivamente. Todas las estructuras han sido creadas utilizando Avogadro [47].

En la figura 1.12 se puede observar representaciones de ambas geometrías observadas en los CQDs.

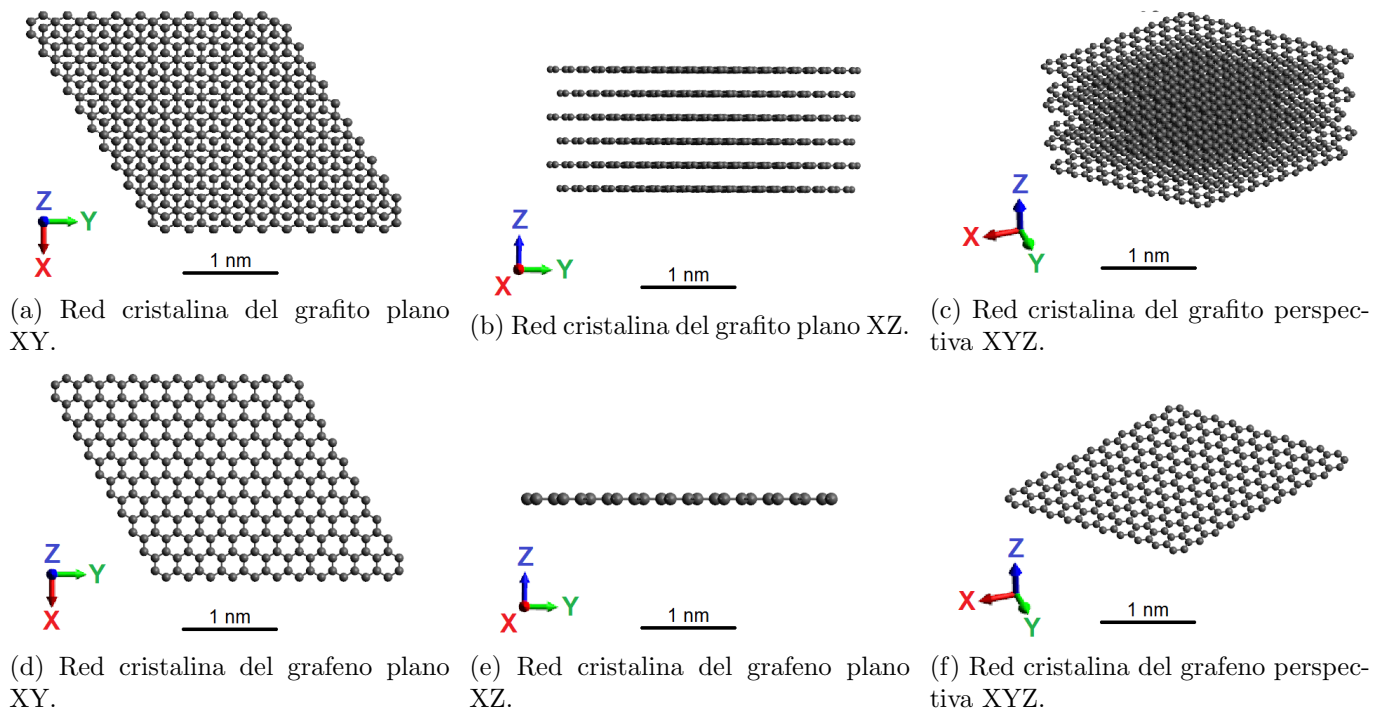
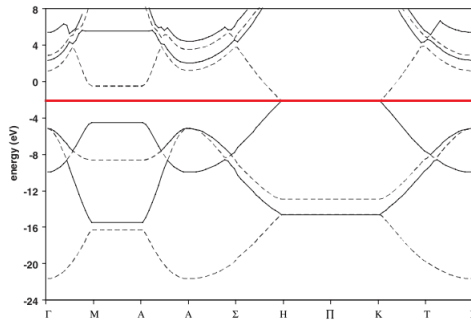
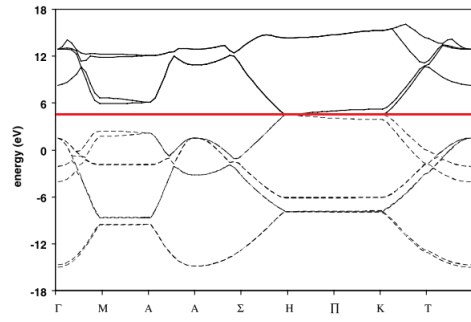


Figura 1.10: Representación de la red cristalina del diamante y grafito. La dirección de los ejes X, Y y Z están dadas por las flechas roja, verde y azul respectivamente. Todas las estructuras han sido creadas utilizando Avogadro [47].

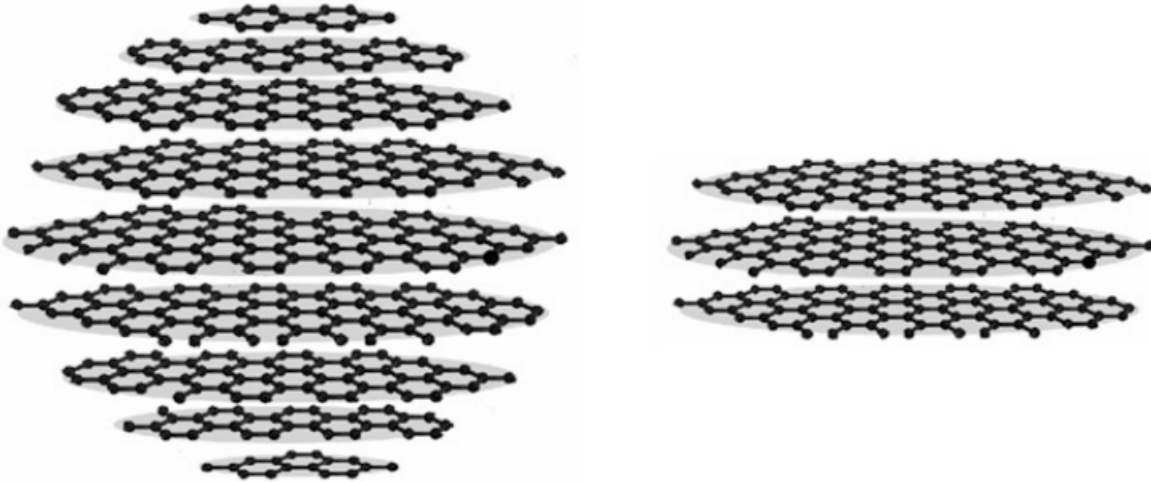


(a) Estructura de bandas electrónicas para el grafeno.



(b) Estructura de bandas electrónicas para el grafito.

Figura 1.11: Estructura de bandas electrónicas para (a) grafeno y (b) grafito. La línea roja indica la energía de Fermi. Ambas estructuras de bandas fueron calculadas utilizando DFT bajo utilizando parámetros similares [46].



(a) Representación de la geometría esférica de los CQDs. (b) Representación de la geometría cilíndrica de los GCQDs.

Figura 1.12: Geometrías presente en los CQDs deducidas a partir de diferentes técnicas experimentales. (a) Geometría esférica observada en CQDs. (b) Geometría cilíndrica observada en GCQDs [48].

En la figura 1.12a se observa que para la formación de una geometría esférica es necesario el apilamiento de un número de placas de grafeno con diferentes radios. Mientras, la geometría mostrada en la figura 1.12b presenta un apilamiento de placas de grafeno de radios similares. Este apilamiento es muy similar al encontrado en el grafito [50]. Por lo cual a los CQDs que presentan geometría cilíndrica se les ha denominado Puntos Cuánticos de Carbono Grafeno, también referidos en inglés como Graphene Carbon Quantum Dots (GCQDs).

Los GCQDs presentan propiedades electrónicas y ópticas similares a los QDs metálicos o semiconductores pero sin su alta toxicidad [9, 29, 30]. La mayoría de métodos de síntesis para la obtención de GCQDs puede ser clasificado en una de las dos categorías siguientes: método abajo hacia arriba, también referido en inglés como *bottom-up method*, y método arriba hacia abajo, también referido en inglés como *top-down method* [6, 7, 48, 49].

En los métodos *bottom-up* se forman GCQDs a partir de moléculas relativamente sencillas, como por ejemplo moléculas de ácido cítrico o glucosa, que servirán como fuente de átomos de carbono. En cambio, los métodos *top-down* utilizan estructuras macroscópicas complejas, como por ejemplo grafito o

celulosa vegetal, como fuentes de carbono para la síntesis de GCQDs. Ambos tipos de métodos presentan sus respectivas ventajas y desventajas, sin embargo ambas poseen una dificultad en común: síntesis con alto control sobre los tamaños, radio y número de placas, de los GCQDs sintetizados. Este es uno de los principales desafíos experimentales que deben de ser superados en este tipo de nanoestructuras [48].

Otro de los desafíos relevantes que enfrentan los GCQDs es su limitada absorción en el visible. Una de las aplicaciones mas atractivas de los QDs es su potencial para mejorar la tecnología actual de paneles solares. Los GCQDs son una alternativa mas amigable con el medio ambiente a comparación de los QDs metálicos o semiconductores. Sin embargo, los GCQDs tienen una absorción muy baja en el rango visible del espectro electromagnético.

Una solución a esta problemática ha sido aprovechar la versatilidad del *C* para formar enlaces, y dopar a la estructura del GCQD con diversos elementos o moléculas [7, 48, 49, 51]. Con los diferentes dopajes en los GCQDs se ha logrado alterar de manera controlada algunas de sus propiedades electrónicas y ópticas. El uso de átomos de nitrógeno (*N*) y potasio (*K*) han mostrado ser efectivos en incrementar el rango de absorción para los GCQDs [7, 52, 53]. En la figura 1.13 se muestra la comparación entre GCQDs con y sin dopaje de *N*. Es posible observar un incremento en la absorción de los GCQDs dopados con *N* respecto a los GCQDs sin dopaje [7].

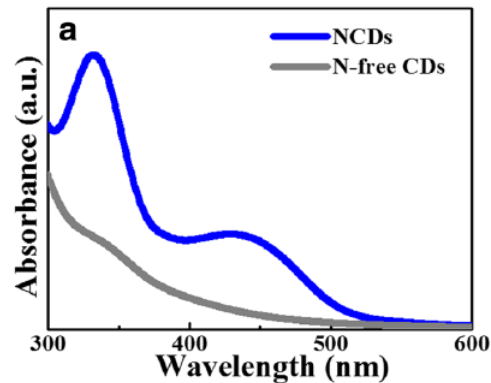
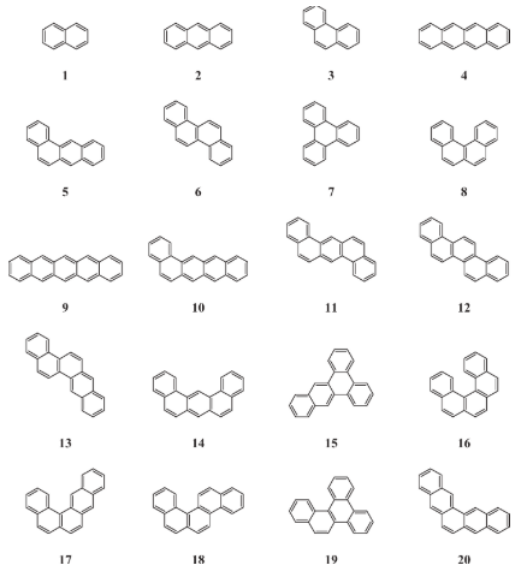


Figura 1.13: Mejora en el espectro de absorción de los GCQDs al ser dopados con nitrógeno. Se compara una muestra de GCQDs sin dopar contra una muestra de GCQDs dopados con *N* [7].

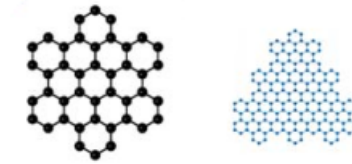
El dopamiento de GCQDs también permite ajustar la bio-compatibilidad de las estructuras para su utilización como bio-marcadores o bio-detectores [7, 49, 51]. A pesar de todos los estudios que se han realizado en GCQDs, existen muchos fenómenos que todavía no se explican por completo. Ejemplo de estos fenómenos se pueden mencionar: el origen de las bandas de absorción y mecanismo de la fotoluminiscencia en los GCQDs [5, 8].

Hasta la actualidad, se utilizan modelos teóricos-computacionales los cuales no toman en cuenta ciertas características de los GCQDs. Por ejemplo, la mayoría de cálculos de GCQDs utilizan estructuras de moléculas planas hidrocarburos policíclicos aromáticos, también referidos en inglés como *Polycyclic Aromatic Hydrocarbons* (PAHs), o estructuras conformados por una sola placa de grafeno [13-15, 54]. En la figura 1.14 se puede observar distintas estructuras utilizadas para la descripción teórica de los GCQDs.

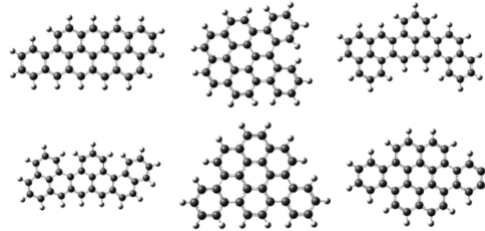
El dopaje de átomos, como *N* y *K*, en las estructuras de grafeno para cálculos de GCQDs generalmente es ignorado, tomando a lo sumo la presencia de átomos de hidrógeno (H). La presencia de estos átomos dopantes afecta significativamente las propiedades de los GCQDs como se observo previamente en la figura 1.13. Otra característica que ignoran los cálculos actuales en GCQDs son las interacciones de larga distancia de Van der Waals (VdW).



(a) Estructuras PAHs [13].



(b) Estructuras de grafeno [15].



(c) Estructuras de grafeno dopadas [14].

Figura 1.14: Diferentes tipos de estructuras utilizadas para la descripción de GCQDs por diferentes autores.

Se sabe que en sistemas con múltiples átomos de carbono formando enlaces σ y π , las propiedades electrónicas se ven influenciadas apreciablemente debido a las interacciones de VdW [16-18]. Por lo tanto, obviar dichas interacciones de VdW es una aproximación muy burda. En la figura 1.15 se puede observar una comparación entre espectros de absorción experimental contra teórica para GCQDs. Para tal cálculo, se utilizó una sola placa de grafeno, fueron obviadas las interacciones de VdW y no se tomó en cuenta el dopaje. Se puede observar que los espectros experimental y teórico concuerdan en la existencia de algunas bandas de absorción. Sin embargo, la teoría menosprecia la intensidad de algunas bandas y otras no es capaz de reproducirlas.

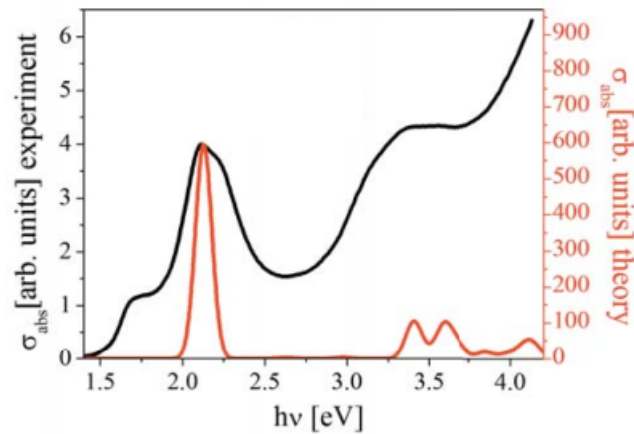


Figura 1.15: Comparación entre espectros de absorción experimentales y teórico para GCQDs. La estructura para los cálculos utilizó una sola placa de grafeno, sin dopar y obviando interacciones de VdW. A pesar de cierta concordancia entre los espectros, diferencias importantes son apreciadas [15].

Capítulo 2

Metodología

En el presente capítulo se describe la metodología utilizada para el estudio de las propiedades de absorción de GCQDs con y sin dopamiento de nitrógeno. La metodología de la investigación constará de dos fases, una experimental y una teórica. En la parte experimental se sintetizarán las muestras de GCQDs y se realizarán las mediciones de absorción. Por el contrario, en la parte teórica se realizarán simulaciones de absorción en GCQDs utilizando modelos computacionales para comprender la física que ocurre en los procesos de absorción de dichas estructuras.

2.1. Síntesis de muestras

Para la síntesis de muestras de GCQDs, se utilizaron marchas químicas previamente reportadas [2, 6, 7]. Dicha síntesis utiliza ácido cítrico, comúnmente abreviada por su nombre en inglés *Citric Acid* (CA), cuya fórmula molecular es $C_6H_8O_7$, y amoníaco, cuya fórmula molecular es NH_3 , como fuentes de carbono y nitrógeno, respectivamente. El proceso de síntesis inicia con la preparación de una solución que utiliza agua destilada como solvente, H_2O . Las cantidades de AC, amoníaco y agua utilizadas dentro de este trabajo fueron 12 g, 9 g y 30 mL, respectivamente. En la figura 2.1 (a) se muestra las estructuras moleculares de todos los compuestos involucrados en dicha solución. Mientras en 2.1 (b) y 2.1 (c) se muestran diagramas del proceso de síntesis utilizado.

Una vez preparada la solución de CA, amoníaco y agua, esta se colocará en contacto con una fuente de calor para realizar una pirólisis. En este trabajo dicha fuente de calor fue una placa calefactora, o también referido por su nombre en inglés, *hotplate*. La solución fue calentada hasta llegar a una temperatura de 200 °C, a un ritmo de calentamiento de $10 \frac{°C}{min}$. Se utilizará un termómetro para medir la temperatura de la solución durante este proceso de calentamiento. En este trabajo se utilizó un vaso de precipitado de borosilicato Pyrex. Una vez que la solución alcanza una temperatura de 200 °C, se mantiene a esta temperatura durante 5 min. Una vez transcurridos estos 5 min, se extraen muestras de 1 mL cada minuto por los próximos 5 min. Cada muestra extraída es diluida en 10 mL de agua destilada a temperatura ambiente. Cada muestra extraída y diluida será centrifugada por 30 min a 5000 rpm. Luego se separa por decantación la solución resultante, la cual contiene los GCQDs, y el soluto precipitado. En la tabla 2.1 se resumen las etiquetas colocadas a las diferentes muestras sintetizadas en el presente trabajo.

Cuadro 2.1: Etiquetas para muestras sintetizadas por pirólisis

Tiempo de pirólisis	GCQD sin dopaje de nitrógeno	GCQD con dopaje de nitrógeno
5 min	Sam. GCQD 01	Sam. N-GCQD 01
6 min	Sam. GCQD 02	Sam. N-GCQD 02
7 min	Sam. GCQD 03	Sam. N-GCQD 03
8 min	Sam. GCQD 04	Sam. N-GCQD 04
9 min	Sam. GCQD 04	Sam. N-GCQD 05
10 min	Sam. GCQD 06	Sam. N-GCQD 06

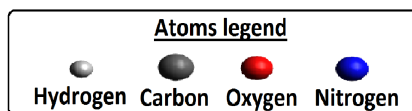
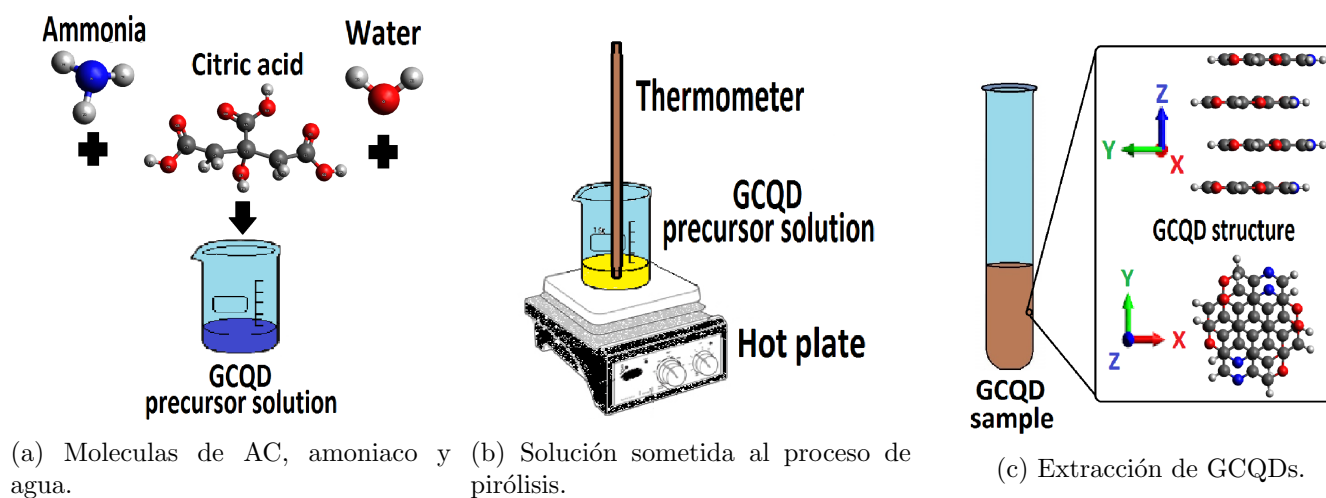


Figura 2.1: Proceso utilizado para la síntesis de GCQDs. (a) Se prepara una solución de AC. (b) dicha solución es colocada en un proceso de pirólisis. (c) Se extraen muestras GCQDs a diferentes tiempos.

2.2. Medición de espectros de absorción

Las muestras sintetizadas fueron caracterizadas mediante espectroscopia de absorción. Se utilizó un espectrógrafo *UV-VIS USB-4000 Ocean Optics* para las mediciones de los espectros. Como fuente de luz continua, se utilizó una lámpara de deuterio-halógeno *DT-Mini Ocean Optics*. Cada espectro de absorción fue obtenido con un tiempo de integración de 650 ms y un total de 40 escaneos. Como referencia se utilizó agua destilada.

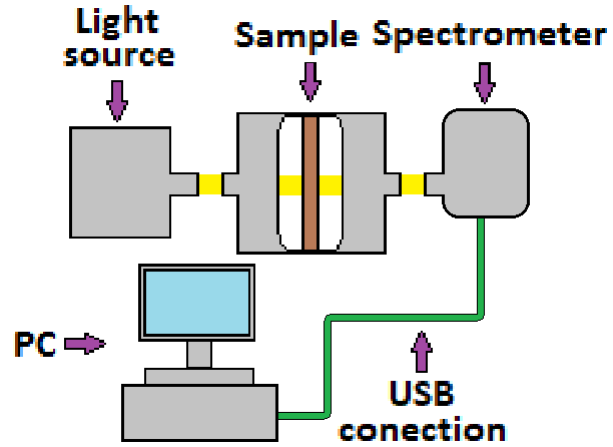


Figura 2.2: Diagrama del arreglo experimental para la medición de espectros de absorción en muestras de GCQDs.

2.3. Realización de cálculos

Para la descripción teórica de las estructuras se utilizaron en conjunto los software: opEn-Source Package for Research in Electronic Structure, Simulation, and Optimization (Quantum Espresso) (QE) 6.2.0, X-window CRYistalline Structure and DENsities (XCRYSDen) 1.6.2, Visualization for Electronic and STructural Analysis (VESTA) 3.0.0, Avogadro 1.90.0 [19, 20, 47, 55, 56].

Las estructuras de los GCQDs fueron dibujadas utilizando Avogadro. Mientras, XCRYSDen y VESTA fueron utilizados para la visualización de las simulaciones realizadas de los GCQDs. Por otra parte, QE fue utilizado para los cálculos de optimización geométrica de las estructuras, las simulaciones de los espectros de absorción y las DoS/PDoS de los GCQDs.

2.3.1. Teoría del funcional de densidad

2.3.1.1. Teorema, Hamiltoniano y ecuaciones de Kohn y Sham

El método se basa en la resolución de la ecuación de Schrödinger utilizando una cantidad auxiliar denominada densidad electrónica $\rho(\vec{r})$. Además para garantizar la unicidad de la solución obtenida por DFT se utiliza el teorema dado por Hohenberg and Kohn (HK) en 1964 y posteriormente reformulado para aplicaciones prácticas por Kohn y Sham (KS) en 1965. Dicho teorema puede ser escrito como [31, 32]:

Teorema 1 *Dado una densidad electrónica del estado base $\rho_0(\vec{r})$ de un sistema, existe solamente un potencial externo V_{ext} el cual es capaz de reproducir dicha configuración de carga.*

Para resolver la ecuación de Schrödinger en un sistema de muchos átomos, DFT primero modifica el Hamiltoniano \hat{H} del sistema de una manera conveniente. Luego utiliza la densidad electrónica $\rho(\vec{r})$ y el teorema de KS para la obtención de las soluciones del sistema. El Hamiltoniano para un sistema de N electrones y M núcleos sería el siguiente:

$$\hat{H} = \hat{T}_{\text{Núcleo}} + \hat{T}_{\text{Electrón}} + \hat{V}_{\text{Núcleo-Núcleo}} + \hat{V}_{\text{Electrón-Electrón}} + \hat{V}_{\text{Núcleo-Electrón}} \quad (2.1)$$

El Hamiltoniano dado en la Ecu. (2.1) puede ser simplificado aplicando la aproximación adiabática de Born-Oppenheimer [31, 32]. Con esta aproximación la energía cinética de los M núcleos atómicos puede ser despreciada. Mientras la repulsión núcleo-núcleo puede ser remplazada por una constante E_{NN} la cual para los cálculos electrónicos es ignorada. Dichos cambios transforman el Hamiltoniano de la Ecu. (2.1) en la siguiente expresión:

$$\hat{H} = \hat{T}_{\text{Electrón}} + \hat{V}_{\text{Electrón-Electrón}} + \hat{V}_{\text{Núcleo-Electrón}} \quad (2.2)$$

En la Ecu. (2.2) los operadores $\hat{T}_{\text{Electrón}}$ y $\hat{V}_{\text{Electrón-Electrón}}$ representa la energía cinética y la energía potencial para el sistema de N electrones que interactúan entre ellos. Mientras el operador $\hat{V}_{\text{Núcleo-Electrón}}$ representa la energía potencial de los M núcleos a los N electrones. Para simplificar la notacion se introducen los siguientes cambios en los operadores:

$$\begin{aligned} \hat{T}_{\text{Electrón}} &\rightarrow \hat{T}_{SI} \\ \hat{V}_{\text{Electrón-Electrón}} &\rightarrow \hat{T}_{SI} \\ \hat{V}_{\text{Núcleo-Electrón}} &\rightarrow \hat{V}_{Ext} \\ \hat{H} &= \hat{T}_{SI} + \hat{V}_{SI} + \hat{V}_{Ext} \end{aligned} \quad (2.3)$$

El subíndice SI significa Si Interactúan, e indican que los operadores describen a un sistema en el cual los electrones interaccionan entre ellos. Sin embargo la forma exacta de los operadores \hat{T}_{SI} y \hat{V}_{SI} así como su efecto sobre la densidad de carga $\rho(\vec{r})$ es desconocido. Para superar dicho inconveniente se mapea el sistema de N electrones que interaccionan entre ellos a un sistema de N electrones los cuales no interaccionan entre ellos. Los operadores pertenecientes a un sistema de electrones que no interaccionan entre ellos seán representados por el subíndice NI , que significa No Interactúan. La realización del mapeo se basara suponiendo que los operadores para un sistema de electrones SI tiene que ser igual a los operadores mapeados a un sistema de electrones NI mas un operador adicional para corregir el ignorar la interacción electrón-electrón. Ademas se tomara que la interacción electrón-electrón en un sistema de electrones NI es igual a la interacción clásica de dos nubes de carga. Las ideas previas resultan en el siguiente par de ecuaciones para los operadores \hat{T}_{SI} y \hat{V}_{SI} :

$$\begin{aligned} \hat{T}_{SI} &= \hat{T}_{NI} + \hat{T}_C \\ &= \sum_j^N \frac{\hat{P}_j}{2m} + \hat{T}_C \end{aligned} \quad (2.4)$$

$$\begin{aligned} \hat{V}_{SI} &= \hat{V}_{NI} + \hat{V}_C \\ &= \hat{V}_{Columb} + \hat{V}_C \\ &= \sum_j \frac{K_e}{\hat{r}_{ij}} + \hat{V}_C \end{aligned} \quad (2.5)$$

El subíndice C indica los operadores de corrección. El mapeo del Hamiltoniano de la Ecu. (2.3) puede ser escrito utilizando las Ecu. (2.4) y (2.5) como:

$$\begin{aligned} \hat{H} &= \hat{T}_{SI} + \hat{V}_{SI} + \hat{V}_{Ext} \\ &= \left[\hat{T}_{NI} + \hat{T}_C \right] + \left[\hat{V}_{NI} + \hat{V}_C \right] + \hat{V}_{Ext} \\ &= \sum_i \frac{\hat{P}_i}{2m} + \sum_I \frac{K_e}{\hat{r}_{iI}} + \sum_{j \neq i}^N \frac{K_e}{\hat{r}_{ij}} + \left[\hat{T}_C + \hat{V}_C \right] \\ \hat{H}_{KS} &= \sum_i \frac{\hat{P}_i}{2m} + \sum_I \frac{K_e}{\hat{r}_{iI}} + \sum_{j \neq i}^N \frac{K_e}{\hat{r}_{ij}} + \hat{E}_{XC} \end{aligned} \quad (2.6)$$

Se ha introducido en la Ecu. (2.6) el la energía de intercambio-correlación \hat{E}_{XC} , donde el subíndice XC representa la expresión en ingles *eXchange-Correlation* (XC). Dicho operador se origina a partir de las correcciones introducidas al realizar el mapeo de un sistema electrónico SI a uno NI:

$$\hat{E}_{XC} = \hat{T}_C + \hat{V}_C \quad (2.7)$$

El Hamiltoniano encontrado en la Ecu. (2.6) es conocido como el Hamiltoniano de Kohn-Sham(KS) [31, 32]. La energía total del sistema vendría dada por:

$$E_{Tot} = \langle \Psi | \hat{H}_{KS} | \Psi \rangle \quad (2.8)$$

Donde $|\Psi\rangle$ debe de ser un conjunto de estados antisimetricos y ortogonales para cumplir con el principio de exclusión de Pauli [31, 32]. Llegados a este punto, es introducida la densidad de carga $\rho(\vec{r})$. En vez de resolver directamente Ecu. (2.8), se introducen N multiplicadores de Lagrange en la Ecu. (2.8). Los multiplicadores de Lagrange introducirán N constricciones que mantendrán los estados energéticos ortogonales entre si. Ademas la constricción introducida, sujeta a que la carga electrónica total en el sistema sea igual a N . Matemáticamente dichas constricciones son:

$$\langle \psi_i | \psi_j | \psi_i | \psi_j \rangle = \sigma_{ij} \quad (2.9)$$

$$N = \sum_i^N \langle \psi_i | \psi_i | \psi_i | \psi_i \rangle \quad (2.10)$$

$$\rho(\vec{r}) = \sum_i^N \psi_i^* \psi_i \quad (2.11)$$

Por lo tanto dependiendo de la densidad de carga $\rho(\vec{r})$ utilizada el valor de la energía total del sistema variará. Con la introducción de la densidad de carga $\rho(\vec{r})$ la energía del sistema se ha convertido en un funcional dependiente de esta. Matemáticamente esto se expresa como:

$$E_{KS}[\rho(\vec{r})] = \langle \Psi | \hat{H}_{KS} | \Psi \rangle \quad (2.12)$$

Si se varia la densidad de carga $\rho(\vec{r})$ del sistema, la energía de este cambiara. Utilizando el primer teorema de KS, se sabe que solo existe una densidad de carga $\rho_0(\vec{r})$ la cual pueda reproducir el potencial externo \hat{V}_{Ext} del sistema. Esto implica que la densidad de carga $\rho_0(\vec{r})$ minimiza la energía del sistema [31, 32]. Por lo tanto, cualquier otra configuración de densidad de carga $\rho(\vec{r})$ sera menos estable. Las afirmaciones anteriores son obtenidas a partir del teorema de KS y el principio de variación [31, 32]. En la terminología de funcionales dada por Ecu. (2.12), se obtendría la siguiente desigualdad concerniendo la energía total del sistema dada por las densidades electrónicas $\rho_0(\vec{r})$ y $\rho(\vec{r})$:

$$E_{KS}[\rho(\vec{r})] > E_{KS}[\rho_0(\vec{r})] \quad (2.13)$$

La Ecu. (2.13) nos indica que utilizando una densidad de carga arbitraria $\rho(\vec{r})$, es posible aproximarse la densidad de carga real del sistema $\rho_0(\vec{r})$ a través de una serie de variaciones en la densidad de carga arbitraria $\rho(\vec{r})$. La variación de la densidad de carga arbitraria $\rho(\vec{r})$ se lleva a cabo mediante métodos de autoconsistencia, también denominados *Self-Consistent Field(SCF) Methods* [19, 31, 32, 57].

Utilizando las constricciones planteadas en las Ecu. (2.9), (2.10) y (2.11) junto con el hamiltoniano KS dado en la Ecu. (2.6) es posible definir un conjunto de N ecuaciones para describir el comportamiento de cada electrón [19, 31, 32]:

$$0 = \left[\frac{\hat{P}_i}{2m} + \sum_I \frac{K_e}{\hat{r}_{iI}} + \sum_j \left\langle \psi_j \left| \frac{K_e}{\hat{r}_{ij}} \right| \psi_j \right\rangle + \hat{V}_{XC} - \epsilon_i \right] |\psi_i\rangle \quad (2.14)$$

Este conjunto de N ecuaciones electrónicas es llamado ecuaciones de Kohn-Sham, o solamente ecuaciones KS. En dichas ecuaciones ha sido introducido el potencial de intercambio-correlación $V_{XC}[\rho(\vec{r})]$, el cual se define como:

$$V_{XC}[\rho(\vec{r})] = \frac{\delta E_{XC}[\rho(\vec{r})]}{\delta \rho(\vec{r})} \quad (2.15)$$

2.3.1.2. Funcional de energía de intercambio-correlación

En principio las ecuaciones de KS, Ecu. (2.14), describen exactamente a un sistema multielectrónico. Sin embargo el potencial de intercambio-correlación $V_{XC}[\rho(\vec{r})]$ es desconocido [31, 32, 34, 58, 59]. Esto ocurre debido a que todas las correcciones introducidas, al realizar el mapeo de electrones SI a NI, fueron agrupadas dentro de la energía de intercambio-correlación $E_{XC}[\rho(\vec{r})]$. Como la forma exacta de dichas correcciones introducidas es desconocida, tanto la energía de intercambio-correlación $E_{XC}[\rho(\vec{r})]$ como el potencial de intercambio-correlación $V_{XC}[\rho(\vec{r})]$, son desconocidos [31, 32, 34, 58, 59]. Para superar dicha dificultad se recurren a aproximaciones de la energía de intercambio-correlación $E_{XC}[\rho(\vec{r})]$ y del potencial de intercambio-correlación $V_{XC}[\rho(\vec{r})]$ [31, 32, 34, 58, 59]. Dichas aproximaciones son obtenidas estudiando sistemas sencillos en los cuales pueda ser obtenida una expresión para la energía de intercambio-correlación $E_{XC}[\rho(\vec{r})]$ y el potencial de intercambio-correlación $V_{XC}[\rho(\vec{r})]$ [31, 32, 34, 58]. Posteriormente se extrapolarían las expresiones obtenidas a sistemas mas complejos.

2.3.1.2.1 Aproximación de la densidad local

Kohn y Sham propusieron en 1965 utilizar un gas de electrones como sistema electrónico de prueba, con el fin posterior de extrapolar los resultados obtenidos del potencial de intercambio-correlación $V_{XC}[\rho(\vec{r})]$ a sistemas electrónicos mas complejos [31, 59, 60]. Hubieron dos motivaciones principales para utilizar el gas de electrones como punto de partida para una aproximación general. La primera motivación era que se contaban con soluciones teóricas y numéricas que describían al gas de electrones de manera razonablemente precisa [31, 32]. La segunda motivación fue suponer que el aporte de la densidad electrónica $\rho(\vec{r})$ sobre el funcional de intercambio-correlación $V_{XC}[\rho(\vec{r})]$ era altamente local. Debido a esta suposición este método para calcular el funcional de intercambio-correlación $V_{XC}[\rho(\vec{r})]$ fue nombrado aproximación de la densidad local, o también conocido por *Local Density Approximation*(LDA).

La forma para la energía de intercambio-correlación $E_{XC}[\rho(\vec{r})]$ y el potencial de intercambio-correlación $V_{XC}[\rho(\vec{r})]$ propuestos por Kohn y Sham fueron los siguientes [31, 59, 60]:

$$E_{XC}[\rho(\vec{r})] = \int \epsilon_{XC}[\rho(\vec{r})] \rho(\vec{r}) d\vec{r} \quad (2.16)$$

$$V_{XC}[\rho(\vec{r})] = \left(\epsilon_{XC}[\rho(\vec{r})] + \rho(\vec{r}) \frac{d\epsilon_{XC}[\rho(\vec{r})]}{d\rho(\vec{r})} \right) \quad (2.17)$$

Donde el termino $\epsilon_{XC}[\rho(\vec{r})]$ representa la energía de intercambio-correlación por partícula. Una expresión exacta de esta energía es desconocida [31, 32, 59]. Sin embargo se puede obtener un aproximación de la energía de intercambio-correlación por partícula $\epsilon_{XC}[\rho(\vec{r})]$ utilizando un gas de electrones como sistema de prueba.

Para aproximar la energía de intercambio-correlación por partícula $\epsilon_{XC}[\rho(\vec{r})]$ se utiliza un conjunto de ondas planas, las cuales describen a un gas de electrones de Dirac [31, 59, 61, 62]. Seguido, se realiza un calculo de auto consistencia por medio de técnicas de Monte Carlo y se estima numéricamente la energía de intercambio-correlación por partícula $\epsilon_{XC}[\rho(\vec{r})]$ [31, 59, 61, 62]. Unos de los resultados numéricos

mas utilizados para la energía de intercambio-correlación por partícula $\epsilon_{XC} [\rho(\vec{r})]$ mediante este método descrito es [59]:

$$\epsilon_{XC} [\rho(\vec{r})] = \begin{cases} -\frac{0,4582}{r_s[\rho(\vec{r})]} - \frac{0,1423}{(1+1,0529\sqrt{r_s[\rho(\vec{r})]+0,3334r_s[\rho(\vec{r})]})} & (a) \\ -\frac{0,4582}{r_s[\rho(\vec{r})]} - 0,0480 + 0,0311 \ln r_s [\rho(\vec{r})] - 0,0116r_s [\rho(\vec{r})] + 0,0020r_s \ln r_s [\rho(\vec{r})] & (b) \end{cases} \quad (2.18)$$

$$r_s [\rho(\vec{r})] = \left(\frac{3}{4\pi\rho(\vec{r})} \right)^{\frac{1}{3}} \quad (2.19)$$

Donde (a) es utilizada si $r_s [\rho(\vec{r})] > 1$ se cumple, mientras (b) $r_s [\rho(\vec{r})] \leq 1$ es aplicada en caso contrario. $r_s [\rho(\vec{r})]$ esta dado en unidades atómicas.

2.3.1.2.2 Aproximación del gradiente general

Cálculos de DFT utilizando LDA lograron reproducir varias propiedades electrónicas de varios sistemas homogéneos y algunos sistemas inhomogéneos[31, 32, 59]. Sin embargo en algunos sistemas, especialmente sistemas atómicos, moleculares e inhomogéneos, los cálculos de DFT utilizando LDA sobrestimaban muchas propiedades electrónicas[31, 32, 59]. Debido a la discrepancia entre los resultados experimentales y teóricos se desarrollo una nueva aproximación para el funcional de intercambio-correlación el cual ahora dependería de la densidad electrónica $\rho(\vec{r})$ y el gradiente de la densidad electrónica $\nabla\rho(\vec{r})$. Debido a la dependencia de dicha aproximación con el gradiente de la densidad electrónica $\nabla\rho(\vec{r})$ esta fue nombrada aproximación del gradiente general, también referida por su nombre en ingles *General Gradient Approximation*(GGA).

La aproximación de GGA supone que la energía de intercambio-correlación por partícula ϵ_{XC} ahora depende de la densidad electrónica $\rho(\vec{r})$ y el gradiente de la densidad electrónica $\nabla\rho(\vec{r})$ [31, 32, 59]. Por lo tanto las expresiones para la energía de intercambio-correlación $E_{XC} [\rho(\vec{r}), \nabla\rho(\vec{r})]$ y el potencial de intercambio-correlación $V_{XC} [\rho(\vec{r}), \nabla\rho(\vec{r})]$ toman la forma siguiente[59]:

$$E_{XC} [\rho(\vec{r}), \nabla\rho(\vec{r})] = \int \epsilon_{XC} [\rho(\vec{r}), \nabla\rho(\vec{r})] \rho(\vec{r}) d\vec{r} \quad (2.20)$$

$$V_{XC} [\rho(\vec{r}), \nabla\rho(\vec{r})] = \left(\epsilon_{XC} [\rho(\vec{r}), \nabla\rho(\vec{r})] + \rho(\vec{r}) \frac{d\epsilon_{XC} [\rho(\vec{r}), \nabla\rho(\vec{r})]}{d\rho(\vec{r})} \right) \quad (2.21)$$

$$\epsilon_{XC} [\rho(\vec{r})] = \rho(\vec{r})^{\frac{1}{3}} F_{XC}^{GGA} [\rho(\vec{r}), \nabla\rho(\vec{r})] \quad (2.22)$$

Una de las primeras aproximaciones de GGA propuso tomar la energía de intercambio-correlación por partícula $\epsilon_{XC} [\rho(\vec{r}), \nabla\rho(\vec{r})]$ como una relación directa con la densidad electrónica $\rho(\vec{r})$ modulada por un factor de incremento de intercambio-correlación $F_{XC}^{GGA} [\rho(\vec{r}), \nabla\rho(\vec{r})]$ [59, 63]. La energía de intercambio-correlación por partícula $\epsilon_{XC} [\rho(\vec{r}), \nabla\rho(\vec{r})]$ y el factor de incremento de intercambio-correlación $F_{XC}^{GGA} [\rho(\vec{r}), \nabla\rho(\vec{r})]$ fueron modelados de la siguiente manera[59, 63]:

$$F_{XC}^{GGA} [\rho(\vec{r}), \nabla\rho(\vec{r})] = (1 + 1,296s [\rho(\vec{r}), \nabla\rho(\vec{r})]^2 + 14s [\rho(\vec{r}), \nabla\rho(\vec{r})]^4 + 0,2s [\rho(\vec{r}), \nabla\rho(\vec{r})]^6)^{\frac{1}{15}} \quad (2.23)$$

$$s [\rho(\vec{r}), \nabla\rho(\vec{r})] = \frac{1}{(13\pi^2)^{\frac{1}{3}}} \frac{\nabla\rho(\vec{r})}{\rho(\vec{r})^{\frac{4}{3}}} \quad (2.24)$$

2.3.1.2.3 Aproximación de Van der Walls

LDA y GGAA han tenido varios éxitos describiendo diferentes sistemas, como moléculas aisladas o material en bulk [16-18, 31, 32, 59]. Sin embargo, estas dos aproximaciones originan resultados pocos

congruentes con los experimentos al momento de describir sistemas de interacción débil, también referidos como *sparse matter* en inglés, como por ejemplo sistemas multicapa, cristales policristalinos y orgánicos moleculares [16-18]. Esto se puede explicar debido a las regiones de interés donde dichas aproximaciones fueron obtenidas. En la figura 2.3 se muestra el potencial para dos placas de grafeno. Las interacciones de VdW origina que las fluctuaciones de la densidad electrónica en una región del sistema, alrededor de z_1 (z_2), afecte la densidad electrónica en otra región del sistema, alrededor de z_2 (z_1).

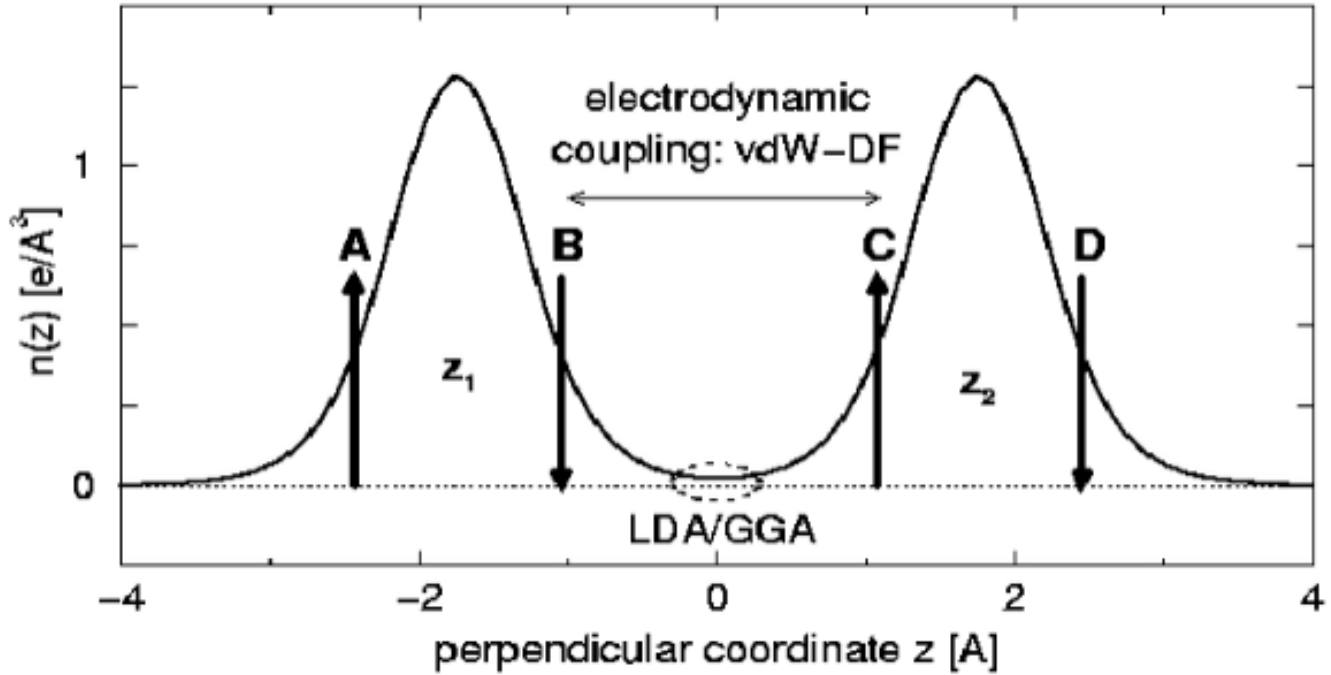


Figura 2.3: Esquemización de la fuerzas de VdW en un sistema de dos capas de grafeno. Las flechas indican regiones típicas donde ocurren fluctuaciones de carga. La elipse punteada indica las regiones de interés que describen los funcionales de XC LDA y GGA [17].

Este esta interaccion de caracter no local es ignorada por LDA y GGA los cual se enfocan en interacciones locales o semi-locales, tal como muestra la figura 2.3. LDA y GGA aproximan la interacción en este tipos de sistemas como el promedio encerrado por la elipse punteada. Diversos autores proponen diferentes metodologías para incorporar dentro de los cálculos de DFT estas interacciones no locales. Una de esta metodologías es incorporar un factor extra que sera el que contribuirá con la interacción no local al sistema [16, 17]. Dicho propuesta es:

$$E_{XC}[\rho(\vec{r}), \nabla\rho(\vec{r})] = E_X^{revPBE}[\rho(\vec{r}), \nabla\rho(\vec{r})] + E_C^{LDA}[\rho(\vec{r}), \nabla\rho(\vec{r})] + E_c^{nl}[\rho(\vec{r}), \nabla\rho(\vec{r})] \quad (2.25)$$

En la ecuación 2.25, el funcional de XC ha sido dividido en un parte semilocal, mediante el uso de LDA y GGA, y una parte no local la cual es modulada por $E_c^{nl}[\rho(\vec{r}), \nabla\rho(\vec{r})]$. En este trabajo, se utilizara el funcional de XC VdW-DF, referido a partir de ahora solamente como funcional de XC VdW, el cual esta basado en la ecuación 2.25. Detalles de la obtención del funcional XC de VdW pueden ser conseguidos en la literatura [16-18].

2.3.1.3. Realización de cálculos de campo autoconsistente

Un ejemplo del flujograma ha seguir para la realización de un cálculo de SCF en DFT se da en la figura (2.4). En dicho flujograma, el cálculo inicia tomando una densidad electrónica de entrada

$\rho_{\text{in}(j)}(\vec{r})$. Luego, se calculan los aportes energéticos de cada interacción y se resuelven las ecuaciones KS para el sistema. Si la densidad electrónica de entrada $\rho_{\text{in}(j)}(\vec{r})$ fuese la densidad electrónica base de dicho sistema, la densidad electrónica de salida $\rho_{\text{out}(j)}(\vec{r})$, obtenida a partir de la solución de las ecuaciones KS, debería de reproducir la densidad electrónica de entrada $\rho_{\text{in}(j)}(\vec{r})$. Si esto ocurre se ha encontrado la solución al estado base del sistema. De no ser este el caso, se inicia otro ciclo de cálculos con una densidad electrónica de entrada modificada $\rho_{\text{in}(j+1)}(\vec{r})$. Este bucle de cálculos continua hasta que se ha llegue a una convergencia entre las densidades electrónicas de entrada $\rho_{\text{in}(j+1)}(\vec{r})$ y salida $\rho_{\text{out}(j+1)}(\vec{r})$. En la practica el cálculo de SCF se detiene cuando una tolerancia aceptable en la convergencia de las densidades electrónicas ha sido obtenida. Para evitar modificaciones arbitrarias a las densidades de entrada $\rho_{\text{in}(j+1)}(\vec{r})$ y una convergencia lo mas eficiente y rápida, diferentes métodos de modificación de la densidad de carga de entrada han sido implementados [19, 20].

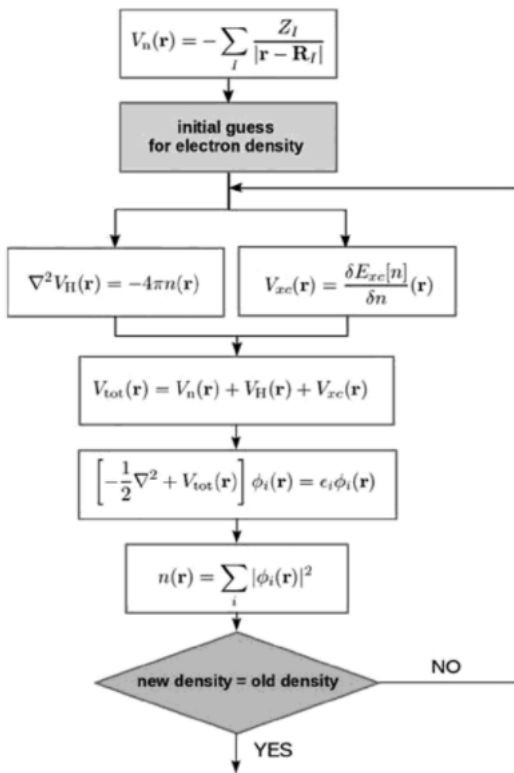


Figura 2.4: Esquema típico del algoritmo utilizado para cálculos de SCF en DFT[57].

QE utiliza funciones de onda planas, llamadas por su nombre en ingles *Plane Waves* (PW), como conjunto de base de entrada para los cálculos de SCF. De manera similar, los cálculos de campo no auto-consistentes, llamados también por sus nombre en ingles *Non Self-Consistent Filed* (NSCF), son cálculos de densidades electrónicas los cuales utilizan una densidad electrónica previamente calculada por medio de un SCF. A diferencia de los SCF, los cálculos de NSCF no se les impone la restricción de que autoconsistencia. Los NSCF generan a partir de la densidad electrónica obtenida del SCF, una densidad utilizan con una malla mas fina. Dicha malla es utilizada para la convergencia en calculos de bandas o DoS. Además de esto, QE utiliza la aproximación de *frozen core* para los electrones internos de cada átomo. Esta aproximación de los electrones internos introducen el uso de pseudo potenciales, llamadas por su nombre en ingles *Pseudo Potentials* (PP), los cuales intentan describir la atracción disminuida percibida por los electrones de valencia debida al acorazamiento de los electrones internos.

Un uso correcto de PP y un conjunto adecuado de PW deben de ser utilizados para un cálculo de DFT. Otros autores han trabajados estructuras de grafeno y grafito utilizando PP obtenidos usando PBE, el cual es un tipo de GGA, por lo cual en el presente trabajo se utilizaran PP de carácter similar. Para los cálculos de SCF y NSCF se utilizaron como los siguientes parametros: la energía de corte fue de 35 Ry o 476 eV para estructuras con menos de 100 átomos y de 25 Ry o 340 eV para estructuras con mas de 100 átomos. Las simulaciones realizadas en este trabajo utilizaron dos conjuntos diferentes de PP [64, 65].

Cuadro 2.2: Parámetros para las estructuras de GCQD utilizadas

Etiqueta	Radio (nm)	Altura (nm)	Numero de átomos	Etiqueta	Radio (nm)	Altura (nm)	Numero de átomos
R2H1	0.47	0.13	15	R3H1	0.71	0.13	30
R2H2	0.47	0.20	30	R3H2	0.71	0.20	60
R2H3	0.47	0.36	60	R3H3	0.71	0.36	120
R2H4	0.47	0.69	90	R3H4	0.71	0.69	180

Cuadro 2.3: Porcentaje de átomos para las estructuras de GCQD utilizadas

Etiqueta	Porcentaje elemental				Etiqueta	Porcentaje elemental				Etiqueta	Porcentaje elemental			
	C	H	O	N		C	H	O	N		C	H	O	N
SGH R2H1	24	6	0	0	SGO R2H1	20	6	8	0	SGN R2H1	18	6	4	2
SGH R2H2	48	12	0	0	SGO R2H2	40	12	16	0	SGN R2H2	72	24	16	4
SGH R2H3	72	18	0	0	SGO R2H3	80	18	24	0	SGN R2H3	108	18	24	6
SGH R2H4	96	24	0	0	SGO R2H4	100	24	32	0	SGN R2H4	144	24	32	8
SGH R3H1	56	6	0	0	SGO R3H1	42	6	14	0	SGN R3H1	36	6	14	6
SGH R3H2	112	12	0	0	SGO R3H2	84	12	28	0	SGN R3H2	72	12	28	12
SGH R3H3	168	18	0	0	SGO R3H3	126	18	42	0	SGN R3H3	108	18	42	18
SGH R3H4	224	24	0	0	SGO R3H4	168	24	56	0	SGN R3H4	144	24	56	24

2.3.2. Dibujo de estructuras

En este trabajo se realizaron simulaciones de 18 estructuras diferentes de GCQDs. Cada una de estas estructuras consta de un número H de placas circulares de radio R . Dichas placas de grafeno son apiladas con una geometría de grafito. Una estructura de un radio $R = X$ y con un número de placas $H = Y$ se etiqueta con la siguiente nomenclatura $RXH Y$. Dependiendo de los átomos que acompañen a las estructuras de carbono se utilizan las siguientes tres etiquetas:

1. Las estructuras que posean átomos de carbono e hidrógeno serán etiquetadas como *Stacking Graphite Hydrogenized (SGH)*.
2. Las estructuras que posean átomos de carbono, hidrógeno y oxígeno serán etiquetadas como *Stacking Graphite Oxygenized (SGO)*.
3. Las estructuras que posean átomos de carbono, hidrógeno, oxígeno y nitrógeno serán etiquetadas como *Stacking Graphite Nitrogenized (SGN)*.

Los parámetros (número de placas, radio, altura) de las estructuras utilizadas dentro de este trabajo son resumidas en la tabla 2.2. Mientras el porcentaje de cada elemento para cada estructura es resumido en la tabla 2.3.

Para dibujar una estructura $RXH Y$ se utilizó el siguiente proceso. Mediante el software Avogadro, se dibujaba una placa de grafeno lo suficientemente amplia para contener un círculo de radio $R = X$. Si la estructura a dibujar será dopada con algún elemento, los átomos de carbono del perímetro son sustituidos por átomos del elemento dopante. La estructura luego es pasiva, para ello los átomos de carbono subcoordinados en el perímetro de la estructura son enlazados con átomos de hidrógeno. Finalmente, un número de placas $H = Y$ son apiladas siguiendo un patrón de apilamiento *ABA* similar al del grafito. El proceso de dopamiento y apilamiento fue realizado a través de un script codificado en

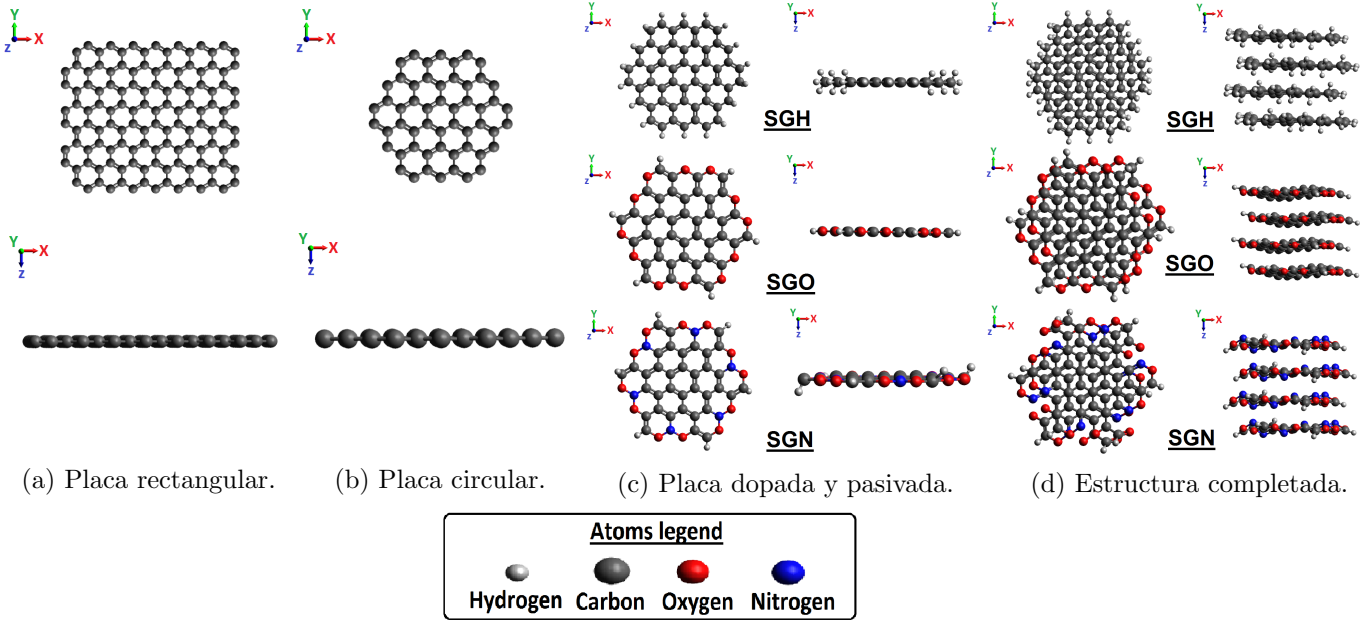


Figura 2.5: Proceso para el dibujo de las estructuras de GCQDs utilizadas. (a) El proceso inicia con el dibujo de una placa de grafeno amplia. (b) Una placa circular de grafeno del radio deseado es cortada. (c) la placa es dopada con diferentes tipos de átomos dependiendo del tipo de estructura que desea dibujarse. (d) un número de estas placas son apiladas para obtener la altura deseada.

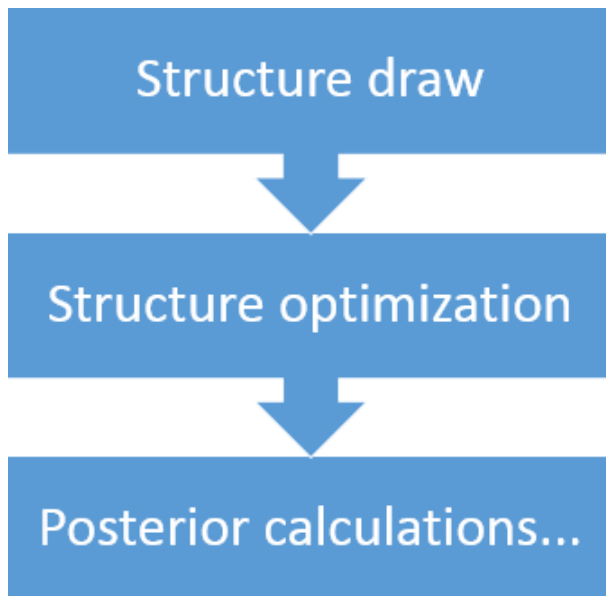
Python 3.6, los cuales han sido añadidos en los anexos .5. En la figura 2.5 se puede observar el proceso para construcción de las estructuras.

2.3.3. Cálculos de optimización geométrica

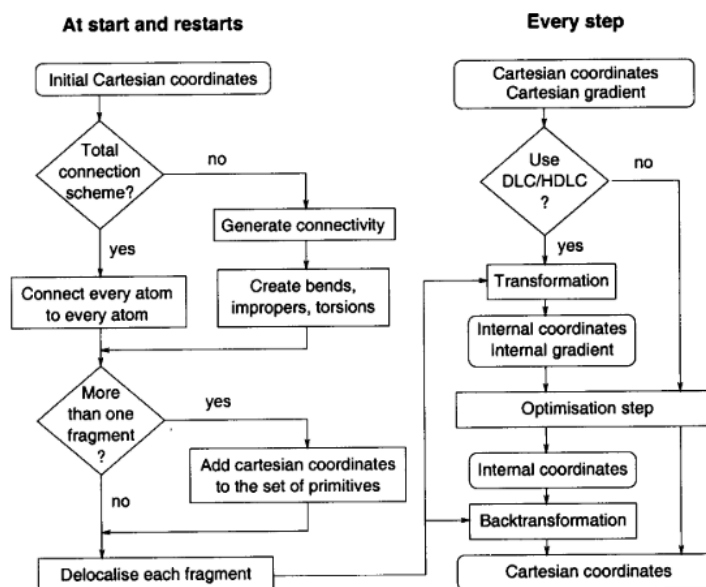
Las propiedades electrónicas y ópticas de un sistema se estudian a partir del estado base de dicho sistema. Por esto en simulaciones en sistemas de muchos cuerpos, primero se realiza una optimización geométrica del sistema a simular. Dicha optimización geométrica busca encontrar las coordenadas espaciales $C(x, y, z)$ para cada átomo que minimice la energía total del sistema. Dentro del marco de DFT un proceso general para una optimización geométrica es el siguiente [19, 66]. Primero se introduce para el sistema un conjunto de coordenadas espaciales iniciales $C_{I,0}(x, y, z)$ y un conjunto de funciones electrónicas iniciales $\rho_{I,0}(x, y, z)$. Luego se realiza un cálculo de SCF, obteniendo así un conjunto de funciones electrónicas finales $\rho_{F,0}(x, y, z)$. A partir de estas funciones electrónicas finales $\rho_{F,0}(x, y, z)$ se optimiza la posición geométrica de los átomos. Obteniendo así una energía total del sistema E_0 y conjunto de coordenadas espaciales finales $C_{F,0}(x, y, z)$.

Para asegurar que se ha obtenido un conjunto de coordenadas que minimizan la energía total del sistema, se realizan varias iteraciones de estos cálculos. Se considera que se ha obtenido una configuración de mínima energía para el sistema, cuando la energía total obtenida en la iteración previa E_{i-1} difiere de la energía total de la iteración actual E_i igual o menor que un factor establecido $|E_i - E_{i-1}| \leq \delta E$. Estas iteraciones utilizan como coordenadas espaciales $C_{I,i}(x, y, z)$ y densidades electrónicas $\rho_{I,i}(x, y, z)$ iniciales las coordenadas espaciales $C_{F,i-1}(x, y, z)$ y densidades electrónicas $\rho_{F,i-1}(x, y, z)$ finales de la iteración previa. En la figura 2.6 se puede observar un esquema resumido del proceso completo de optimización geométrica, así como el algoritmo para una iteración de este proceso [66].

Para esta optimización geométrica, también conocido en la documentación de QE como cálculo de relajación o *relaxation calculation* en el idioma inglés, se utilizó el método de optimización de Newton, en particular se utilizó un algoritmo de Broyden–Fletcher–Goldfarb–Shanno (BFGS) [19, 66]. Se utilizó



(a) Proceso para la optimización geométrica.



(b) Algoritmo para la optimización geométrica [66].

Figura 2.6: Procesos utilizado para los cálculos de optimización geométrica realizados. (a) proceso utilizado y (b) algoritmo de optimización utilizando el método BFGS [66].

el paquete computacional de QE *pw.x* para realizar los cálculos optimización geométrica. de Para este trabajo se utilizo una diferencia de energías de $\delta E = 1 \times 10^{-9}$. Otros variables de interés son mostradas en los archivos de entrada agregados en los anexos .1 y su archivo correspondiente para ser ejecutado en *Torque* se encuentra en los anexos .4.

2.3.4. Simulación de espectros de absorción y respuesta electrónica

Para la realización de las simulaciones de los espectros de absorción el proceso es el siguiente. Se utiliza las coordenadas $C(x, y, z)$ de una estructura, previamente optimizada geoméricamente, para un cálculo de SCF con alta precisión alrededor del punto Γ . Seguido de esto se calcula el tensor de polarización $\alpha_{ij}(\omega)$ utilizando el método de Liouville–Lanczos [20, 41, 42]. Este método consiste en reescribir las ecuaciones de HF en función de la frecuencia de oscilación ω bajo la forma cuántica de la ecuación de Liouville L [41, 42]. Luego el sistema es sometido a una pequeña perturbación para obtener su respuesta a un campo electrico exterior y simular el proceso de absorción. Para la resolución de este conjunto de ecuaciones se utiliza el algoritmo de bi-ortogonalización de Lanczos [41, 42].

2.3.5. Cálculos de densidad de estados

Para la realización de los cálculos de DoS y PDoS el proceso es el siguiente. Se utiliza las coordenadas $C(x, y, z)$ de una estructura, previamente optimizada geoméricamente, para un cálculo de SCF con alta precisión en una malla 3D. Luego se prosigue con un calculo NSCF con alta precisión en una malla 3D mas fina. Una vez termino estos cálculos de auto-consistencia se procede a calcular la DoS para el sistema con las densidades electrónicas $\rho_{I,i}(x, y, z)$ previamente encontrada. Para el cálculos de las PDoS se descovoluciona la DoS calcula usando los orbitales atómicos y densidades electrónicas $\rho_{I,i}(x, y, z)$ de entrada. Finalmente, se procede a ensanchar las DoS y PDoS obtenidas según se cree necesario.

Con las ecuaciones resueltas y con el tensor de polarización $\alpha_{ij}(\omega)$ calculado, se procede a calcular el OS del sistema. Finalmente, para reproducir los resultados experimentales se introduce un ensanchamiento del OS. En la figura 2.7 se puede observar un diagrama de este proceso. Además con los coeficientes de Lanczos obtenidos, se puede hacer un cálculo de como responderá la densidad electrónica a un campo eléctrico externo de una energía determina. Esto nos puede dar información de como se comporta la nube electrónica por la presencia de un foton. Para el cálculo de SCF se utilizo *pw.x*, mientras que para los cálculos de perturbación, OS y ensanchamiento se utilizo turboTDDFT (*turbo_lanczos.x* y *turbo_spectrum.x*). Todos estos paquetes computacionales incluidos en la distribución de QE. Para el cálculo de SCF se utilizo una tolerancia de $\delta E = 1 \times 10^{-12}$ y un total de 10000 términos para el algoritmo de bi-ortogonalización de Lanczos. Otras variables de interés son mostradas en los archivos de entrada agregados en los anexos .3 y su archivo correspondiente para ser ejecutado en *Torque* se encuentra en .4 para los espectros de absorción y .4 para respuesta electrónica.

Para los cálculos de SCF y NSCF se utilizo *pw.x*, mientras los cálculos de DoS, PDoS y ensanchamiento se utilizo *projwfc.x*. Ambos paquetes computacionales incluidos en la distribución de QE. Para el cálculo de SCF y NSCF se utilizo una tolerancia de $\delta E = 1 \times 10^{-12}$. La malla para los cálculos de SCF y NSCF fueron de $5 \times 5 \times 5 \times 10 \times 10 \times 10 \times$, respectivamente. Otras variables de interés son mostradas en los archivos de entrada agregados en los anexos .2 y su archivo correspondiente para ser ejecutado en *Torque* se encuentra en .4. Para el post-procesamiento de sumar los diferentes orbitales y contribucion de cada placa se utilizo un script hecho en Python 3.6 que se muestra en los anexos .6.

2.3.6. Método de Numerov

Para apoyar los resultados computacionales, se plateo un modelo para explicar el QCE en un sistema de varias placas de grafeno. En este modelo, se utiliza una función semi-empírica reportada para la energía potencial entre placas de grafeno [67]. La ecuación de Schrödingerde resultante de este planteamiento sera resuelta utilizando el método de Numerov [68, 69]. Numerov presenta una algoritmo sencillo, manteniendo una presición igual o superior a estos algoritmo. Por ejemplo, Numerov es mas sencillo que como Runge-kutta orden 4, el cual es un métodos popular para resolución de , y a pesar de

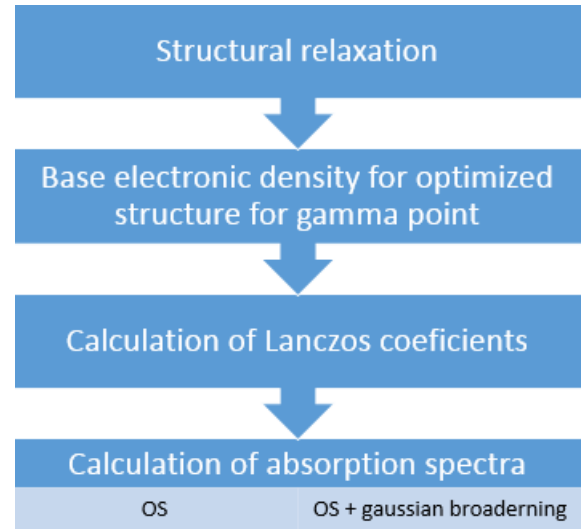


Figura 2.7: Proceso para el calculo de espectros de absorción

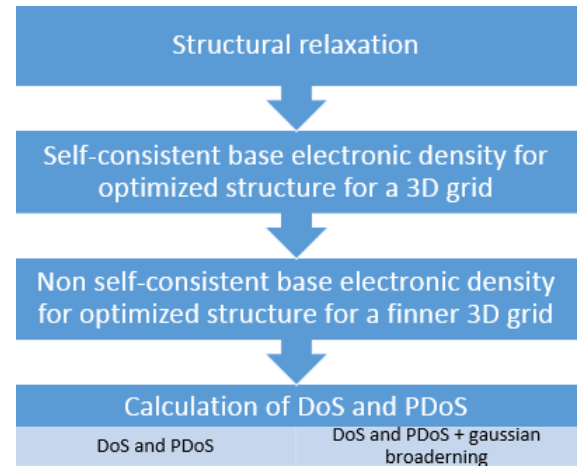


Figura 2.8: Proceso para la simulación de DoS y PDoS.

esto se obtienen resultados iguales o mejores que este [68].

Debido a que no se conoce las *eigen*-energías del sistema a resolver. Se creara una malla de energías, comenzando con la energía mínima a la energía máxima permitida por la función de energía potencial. Luego el problema de una *eigen*-energía desconocida se transformara al problema de encontrar la raíz de una función. Para esto, barrera la malla de energía creada, cuando el signo al final de la función prueba a una energía determinada cambie de signo indicara la presencia de una posible raíz (en el problema original la presencia de una *eigen*-energía del sistema). Para encontrar el valor exacto de la raíz (*eigen*-energía) se utilizara el método de bisección. En el anexo .7 se muestra el código en Python3 utilizado para resolver este problema.

Capítulo 3

Resultados y discusión

3.1. Síntesis

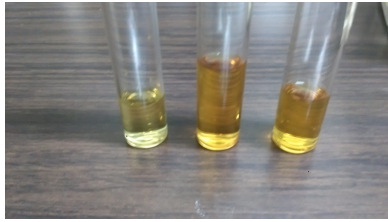
La solución de CA al momento de preparación traslucida. Una vez comenzada la pirólisis, esto es cuando la solución de CA alcanza los $200\text{ }^{\circ}\text{C}$, la solución se mantuvo traslucida por 5 min . Luego de 5 min de pirólisis, la solución se torno de color amarillo claro. A medida que el tiempo de pirólisis aumentaba el color de la muestra se torno de amarillo a marón y luego a café oscuro. Otros autores han observado este mismo comportamiento, y se lo atribuyen a la formación de GCQDs [6]. Este fenómeno fue observado para las soluciones con y sin amoniaco. Sin embargo, la solución con amoniaco se torno café oscuro en los primeros 10 min de pirólisis. Contrariamente, la solución sin amoniaco tardo alrededor de 15 min de pirólisis en obtener esa misma coloración. Esto puede ser explicado de la siguiente manera. El ritmo de formación de GCQD respecto al tiempo en ambas soluciones, con y sin amoniaco, debería ser similar o un poco menor para la solución con amoniaco. Por lo tanto, las propiedades ópticas de GCQD dopados y sin dopamiento de nitrógeno debería de ser la causa de la coloración mas apresurada. Además, varios autores reportan que la absorción en el visible mejora para los GCQD dopados con nitrógeno respecto a los GCQD sin dopar [7]. En la figura 3.1 se puede observar una comparación del comportamiento bajo luz VIS y UV entre las muestras con y sin dopaje de nitrógeno.

Debido a que no se cuenta con acceso a un equipo de TEM o SEM para caracterizar el tamaño de los GCQD sintetizados, se utilizo medidas de TEM, SEM, AFM y espectros de absorción de otros autores para estimar el tamaño (altura y diámetro) de las muestras sintetizadas en este trabajo.

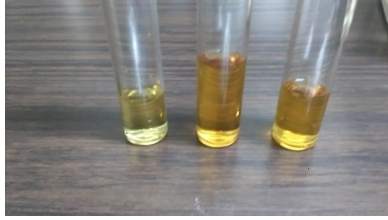
Todos los espectros muestran 3 bandas de absorción en las energías de $5,0\text{ eV}$ o 230 nm , $3,5\text{ eV}$ o 350 nm y $2,7\text{ eV}$ o 450 nm . El origen de estas bandas han sido asociado a transiciones $\pi-\pi^*$ (CC), $n-\pi^*$ (OC) y $n-\pi^*$ (NC), respectivamente. La banda de absorción más energética es asociada con el excitón libre de la nanoestructura de carbono. Debido a que el excitón libre se ve afectado por el CQE, la energía de esta banda esta relacionada con el tamaño de la estructura. En cambio, las bandas de absorción de menores energía no presentan dependencia del tamaño de la estructura. Dado que el QCE no se observa para estas bandas de absorción, estas transiciones pueden ser analizadas como transiciones moleculares o transiciones de defecto introducidas por átomos dopantes en la estructura. Las pequeñas diferencias entre las estas bandas de absorción menos enérgicas son explicadas debido a ciertas diferencias en la síntesis de la muestra (reactivos utilizados, pequeñas diferencias de temperatura y tiempo de pirólisis, etc).

En la figura 3.2, se puede observar que la banda de absorción para el excitón libre $\pi-\pi^*$ (CC) de la muestra Sam. N-GCQD 01 posee valores similares a los reportados por otros autores. Por lo tanto podemos estimar que las muestras sintetizadas, tanto muestras con y sin dopamiento de nitrógeno, constan de estructuras con un diámetro alrededor de 1 nm y entre 2 a 6 placas de grafeno apiladas

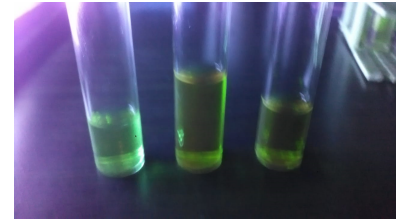
Reportes previos que estudian el proceso de formación de GCQD a partir de CA sin dopaje, también



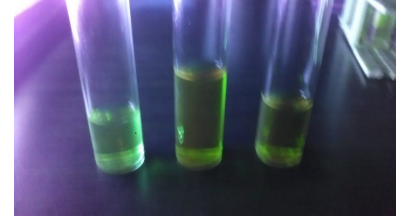
(a) Muestra sin dopaje de nitrógeno bajo VIS.



(c) Muestra con dopaje de nitrógeno bajo VIS.



(b) Muestra sin dopaje de nitrógeno bajo UV.



(d) Muestra con dopaje de nitrógeno bajo UV.

Figura 3.1: Muestras de GCQD con y sin dopaje de nitrógeno obtenidas para un tiempo de pirólisis de 7 *min.* (a) y (b) muestra de GCQD sin dopaje expuesta bajo luz VIS y UV. (c) y (d) muestra de GCQD con dopaje expuesta bajo luz VIS y UV.

En la figura 3.2 se muestra una comparación entre los espectros de absorción de GCQD dopados con N reportados por varios autores y los espectros de absorción de GCQD dopados con N medidos en este trabajo [7, 70-72]. Los espectros presentados en la figura 3.2 han sido obtenidos de la pirólisis de CA, o compuestos de CA, bajo condiciones similares, exceptuando Wang et al. el cual utilizó 30 *min* de pirólisis. Los intervalos de diámetros reportados por Jian Ju et al., Wang et al. y Xiaohui Lui et al. son 2,1 *nm* a 3,8 *nm*, 1,8 *nm* a 2,48 *nm*, 7,0 *nm* a 15,0 *nm* y 1,0 *nm* a 5,0 *nm*, respectivamente. Mientras, Los intervalos de alturas reportados por Jian Ju et al., Wang et al. y Xiaohui Lui et al. son 0,5 *nm* a 1,5 *nm*, 1,0 *nm* a 3,0 *nm* y 0,5 *nm* a 2,0 *nm*, respectivamente. Zhi Yang et al. no reporta alturas para sus mediciones [71]. Debido a que los GCQD presentan una morfología similar al grafito, podemos estimar el número de placas para estas síntesis utilizando la separación entre placas de grafito (0,34 *nm*).

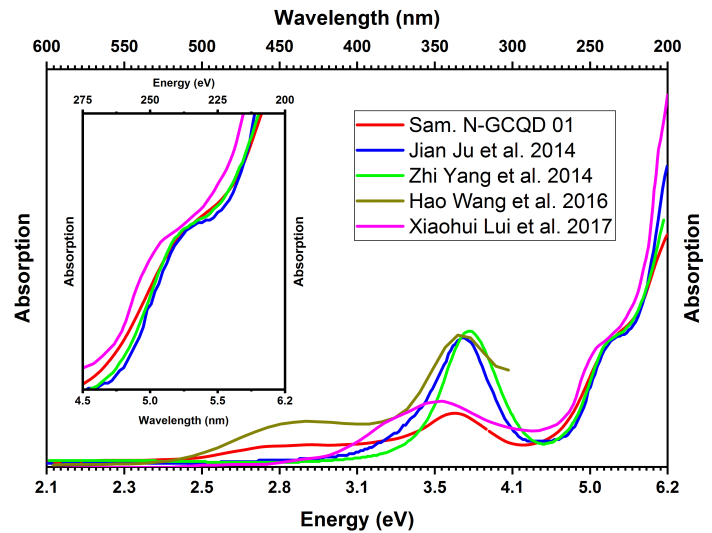
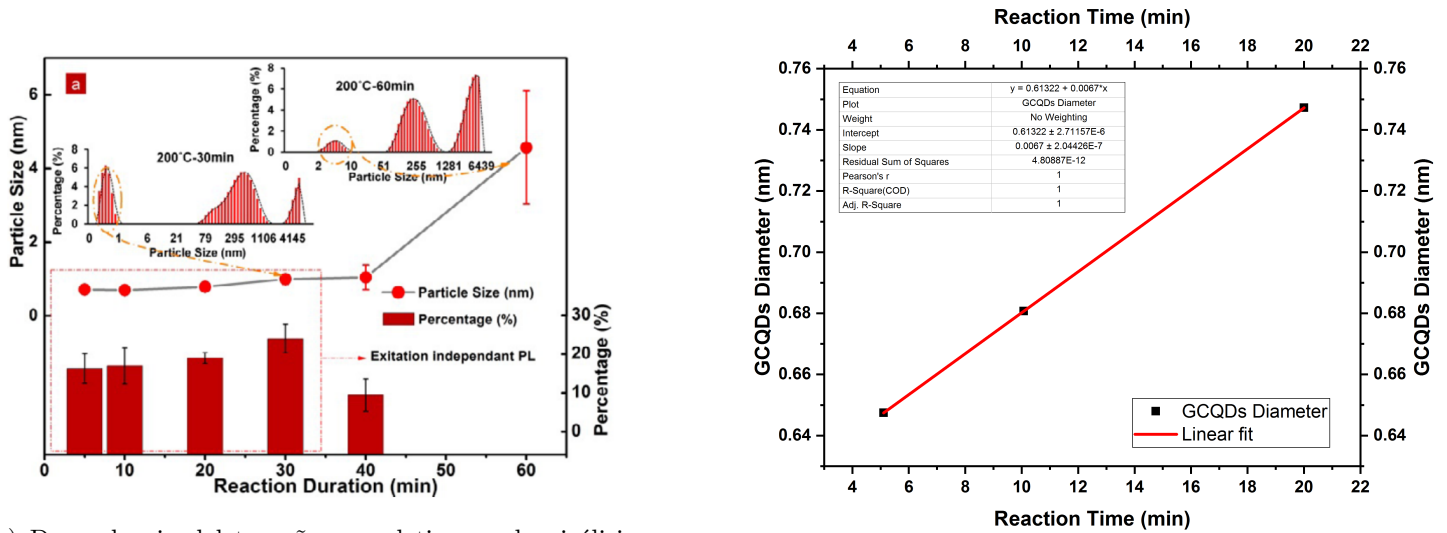


Figura 3.2: Espectros de absorción de diferentes autores, los cuales han realizado mediciones de diámetros a los GCQD sintetizado. Se comparan los espectros de absorción, previamente mencionados, con los espectros medidos de la muestra Sam N-GCQD 01 [7, 70, 71]. En la inserción se puede observar un acercamiento a la banda asociada con transiciones $\pi - \pi^*$ (CC).

reportan tamaños alrededor a 1 *nm* para tiempos de pirólisis entre 5 *min* y 10 *min* [6, 73]. Se puede realizar una estimación mas fina de los diámetros basándose en una interpolación utilizando los datos reportados de mediciones TEM [6]. En la figura 3.3 podemos observar los histogramas de diámetros medidos con TEM para muestras de GCQDs sintetizadas en diferentes tiempos de pirólisis. De la figura 3.3 (a) pueden obtenerse un total de 3 puntos que pueden ser usados para la interpolación dentro del

rango de tiempos de pirólisis utilizados en este trabajo. La interpolación resultante puede ser observada en la figura 3.3 (b).

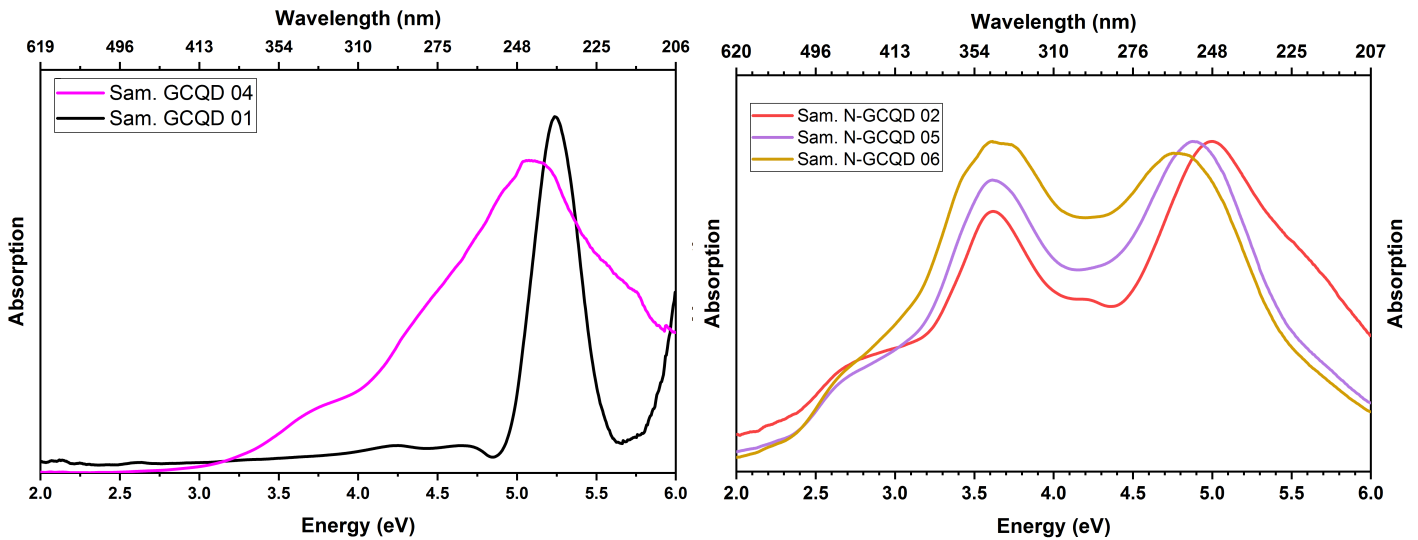


(a) Dependencia del tamaño con el tiempo de pirólisis [6].

(b) Interpolación lineal de la dependencia del tamaño con el tiempo de pirólisis.

Figura 3.3: Tamaño (Diametro) de los GCQD en función del tiempo de pirólisis. (a) dependencia del tamaño contra el tiempo obtenidas mediante mediciones de TEM [6]. (b) Intepolación lineal para la dependencia ente tamaño y tiempo para los primeros 20 *min* de pirólisis.

Si la estimación del incremento de los tamaños respecto al tiempo de pirólisis es valida, se esperaría ver un desplazamiento al rojo de la banda de absorción para el excitón libre $\pi - \pi^*$ (CC). En la figura 3.4 se puede observar un desplazamiento al rojo para las diferentes muestras sintetizadas a medida que los tiempos de pirólisis aumentan. En contraste, las bandas de absorción relacionadas a los átomos dopantes (O y N) se mantiene fija en energía (2,7 eV y 3,5 eV).



(a) QCE observado en muestras GCQD 01 y 04.

(b) QCE observado en muestras N-GCQD 01, 03 y 05.

Figura 3.4: Espectros de absorción para muestras sintetizadas sin (a) y con (b) dopamiento de nitrógeno. Los comportamientos de las bandas de absorción se mantienen para ambas síntesis.

En la figura 3.5 se compara el espectro Raman para la Sam. N-GCQD 01 contra el espectro Raman del CA, grafeno y GCQD reportado por otros autores [74, 75]. El espectro Raman de dicha muestra ha sido cortesía de H. Ponce y O. Deodanes. De este espectro podemos observar que hay bandas características tanto de CA como de GCQD. Lo cual nos indica que hay residuos de CA debido a una reacción incompleta para ese tiempo de pirólisis. Las bandas Raman que la muestra Sam. N-GCQD 01 comparte tanto con el grafeno como con los GCQD indican la formación de estas estructuras en síntesis realizadas.

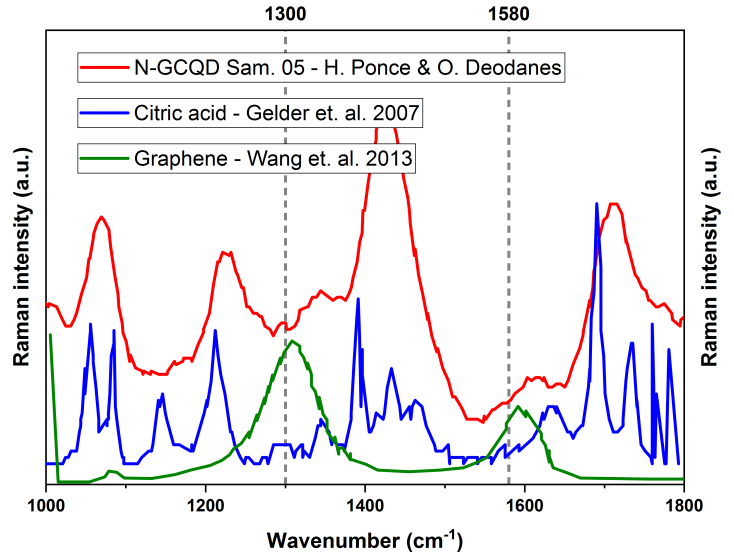


Figura 3.5: Comparación entre espectros Raman reportados para CA, grafeno, GCQD por otros autores y el espectro medido para la muestra Sam. N-GCQD 01 [74, 75].

3.2. Optimización geométrica

En la parte de las simulaciones, el primer paso es discutir los resultados de las optimizaciones geométricas. Por cada estructura y cada funcional XC se realizó uno de estos cálculos. Un cálculo de optimización es considerado aceptable cuando cada iteración disminuye la energía total del sistema. Debido a que todos los cálculos solo proporcionan una aproximación a la densidad electrónica y energía del sistema, se fija un límite de tolerancia. Este límite de tolerancia indica que una vez superado, cada iteración realizada disminuirá la energía total del sistema de manera tan poco apreciable que el trabajo computacional y tiempo invertido para realizar el cálculo es demasiado comparado con mejora que se obtiene en la energía total del sistema. En la figura 3.6 se puede observar un gráfico del número de iteraciones y el tiempo invertido en ellas contra la energía total del sistema para la estructura *SGN R2H2*.

Todas las estructuras se consideraron satisfactoriamente optimizadas geométricamente, si y solo si se observaba este comportamiento asintótico para la energía total del sistema contra el número de iteraciones (tiempo de cálculo). Como era de esperar las estructuras con mayor números de átomos, y número de átomos dopantes, necesitaron tiempos mayores de cálculo. En particular, las estructuras *SGO* y *SGN* con las dimensiones *R2H3*, *R2H4*, *R3H3* y *R3H4* tuvieron que utilizar un algoritmo diferente de mezclado de densidades electrónicas para evitar que la energía total del sistema divergiera. En la figura 3.7 es posible observar este fenómeno.

Estos resultados de optimización geométrica pueden ser comprendidos de la siguiente manera. El modo de mezclado de densidades electrónicas *local-TF*, es utilizado obtener un convergencia de la densidad electrónica y energía total del sistema para estructuras con poca homogeneidad en la densidad electrónica. Las estructuras *SGO* y *SGN* presentan átomos dopantes que perturban los enlaces deslocalizados π que se forman en las placas de grafeno, lo cual disminuye la homogeneidad de la densidad electrónica del sistema. Si además apilamos varias placas de grafeno con alta inhomogeneidad en la densidad electrónica, esto incrementaría aún más dicha inhomogeneidad de la la densidad electrónica. Esto explicaría porque las estructuras con mas átomos y mayor numero de placas (*R2H3*, *R2H4*, *R3H3* y *R3H4*) son las que presentaron este fenómeno. Mientras otras estructuras a pesar de tener un numero

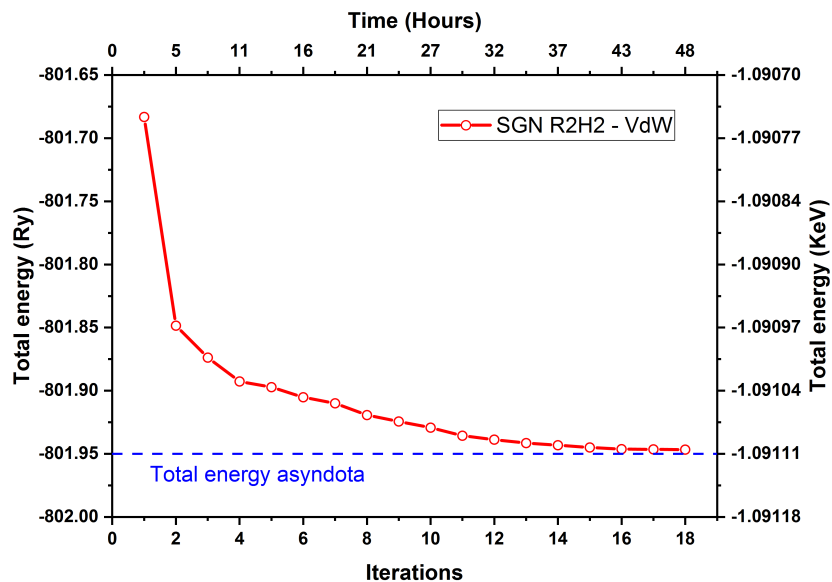
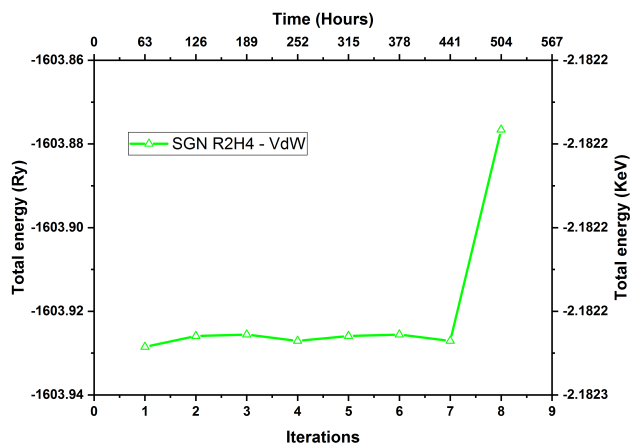
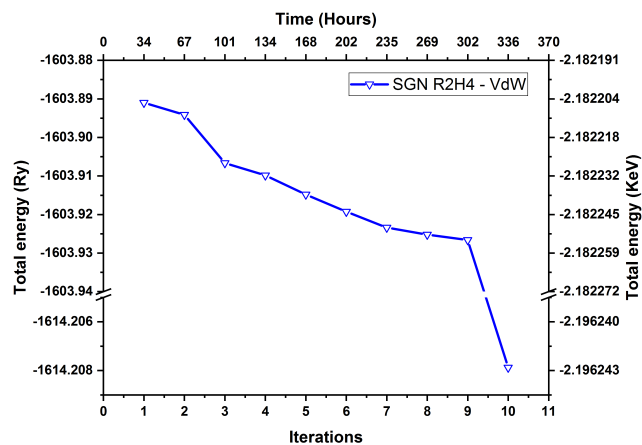


Figura 3.6: Energía total del sistema en función del número de iteraciones (tiempo de cálculo) para la estructura *SGN R2H2*. Se puede observar un comportamiento asintótico aproximadamente en $-1,0911 \text{ KeV}$ ($-801,95 \text{ Ry}$).

Una vez superada iteración número 15 (37 h), cada nuevo cálculo de optimización geométrica disminuye muy poco la energía total del sistema. Matemáticamente, se puede analizar este comportamiento como una asíndota en la relación de estas variables. Se realizaron varios ensayos de optimización geométrica con las estructuras *SGH R2H1*, *R2H2*, *R2H3* y *R2H4*. A partir de estos, se observó que las estructuras mostraban un comportamiento asintótico para la relación energía total del sistema contra el número de iteraciones (tiempo de cálculo) utilizado el valor de $1,0 \times 10^{-9} \text{ Ry}$ o $1,3 \times 10^{-8} \text{ eV}$. En vista de los buenos resultados de optimización geométrica obtenidos para las estructuras de prueba, se decidió utilizar este límite de tolerancia para la convergencia en la energía total del sistema para todas las estructuras estudiadas en este trabajo.



(a) Energía total del sistema obtenida para modo de mezclado *plain*.



(b) Energía total del sistema obtenida para modo de mezclado *local-TF*.

Figura 3.7: Dependencia de la energía total del sistema en función del número de iteraciones (tiempo de cálculo). En (a) se puede observar como la energía total del sistema muestra una tendencia a aumentar con cada iteración, mientras que en (b) se puede observar que la energía de total del sistema disminuye. Ambos cálculos de optimización fueron realizados en la estructura *SGN R2H4*.

similar de átomos pero al estar conformados por menos placas no mostraron este fenómeno (*R3H1* y *R3H2*). A partir de esto se podría afirmar que las estructuras *SGO* y *SGN* poseen inhomogeneidad en la densidad electrónica debido a los átomos dopantes del sistemas y esta toma especial importancia cuando el número suficiente de átomos y placas es alcanzado.

Otra cantidad relacionada con la homogeneidad u inhomogeneidad de la densidad electrónica es el tiempo que se necesita para realizar los cálculos de SCF. Esto se debe a que, el mezclado de densidades electrónicas permiten disminuir (aumentar) el tiempo de cálculo para la convergencia de la densidad

electrónica y la energía total del sistema si este es escogido acorde (no acorde) a la simetría del sistema. Recordando que el tiempo t para realizar cálculos de DFT es proporcional al cubo del número de átomos N del sistema (esto es $t \propto N^3$), cualquier fenómeno que dificulte la convergencia en los cálculos de SCF vera su efecto acentuado para sistemas que tenga un gran número de átomos ó el algoritmo de mezclado sea particularmente inadecuado. Este incremento en tiempo de cálculo debido a escoger un mal algoritmo de mezcla de las densidades electrónicas puede verse reflejado en la figura 3.7. En esta figura se puede observar como incrementa (disminuye) la energía total del sistema cuando se utiliza el algoritmo de mezclado *plain* (*local-TF*) para la estructura *SGN R2H4*. Debido a la inhomogeneidad, el algoritmo de mezclado *plain* falla en obtener una convergencia para la energía total de la estructura. En cambio, el algoritmo de mezclado *local-TF* no solo logro converger la energía total de la estructura sino que ademas utilizo una menor cantidad de tiempo de cálculo. Aproximadamente, el algoritmo *local-TF* necesito un 33,4% menos de tiempo de cálculo respecto al algoritmo *plain*.

Otros parámetros de importancia cuyos valores fueron obtenidos de manera empírica fueron el volumen de la celda unitaria, el número de puntos de la malla 3D y la energía de corte para las funciones de ondas plana. El volumen de la celda unitaria se refiere a un parámetro que muchos códigos de simulación orientados a física del estado solido, QE siendo uno de ellos, utilizan. Estos códigos computacionales toman la celda unitaria del sistema a estudiar, para luego generar un numero muy grande de estas. La razón de este enfoque es debido a que de esta manera se trata de simular la interacción entre celdas unitarias vecinas dentro de un sólido cristalino. Aunque este enfoque es útil para sólidos, no es el adecuado para simular QDs. Para las simulaciones de sistemas “finitos”, como moléculas o QDs, primero se simula el sistema aislado de las interacciones del entorno, esto es se simula la molécula o QD en el vacío. Por lo tanto si se utilizan este tipo de cogidos computacionales, para evitar la interacción entre celdas unitarias vecinas se realiza lo siguiente. Se escoge una celda unitaria con un volumen lo suficientemente grandes para que las densidades electrónicas ρ calculadas no se solaparan de manera relevante con el vecino inmediato. Como solo se desea que las densidades electrónicas no se solapen, la geometría de la celda unitaria no es relevante. Sin embargo, debido a que era mas sencillo generar los códigos de entrada utilizando celdas unitarias rectangulares esta geometría fue la utilizada.

El aumentar el volumen de la celda unitaria implica un costo mayor en el número de ondas planas y puntos a utilizar en la malla 3D necesarias para simular la estructura. Un aumento de estos factores incrementan los tiempo necesarios para realizar los cálculos. Por lo tanto, se tiene que encontrar valores para el volumen rectangular de la celda unitaria, número de puntos de la malla 3D y valores de corte de las funciones de onda planas que mantengan cierto rigor en la simulación y permitan tiempos de cálculo factibles. Los parámetros que mejor equilibrio entre precisión y rapidez, obtenidos como resultado de los cálculos en las estructuras de prueba, de los cálculos fueron los siguientes:

- El número de puntos que mejor funciono para la malla 3D fueron $5 \times 5 \times 5$ para cálculos de optimización geométrica, SCF para simulaciones de espectros de absorción y cálculo de DoS, mientras que para cálculos *NSCF* de DoS fueron utilizadas $10 \times 10 \times 10$.
- Para los cortes de energía en la funciones de onda plana, se utilizo un valor base de $15 Ry$ ($204 eV$) para todas las estructuras. Por cada aumento de radio o por cada capa de grafeno introducida a la estructura se incrementaba la energía de corte por un factor de $5 Ry$ ($68 eV$). La ecuación 3.1 resume este procedimiento:

$$ecutwfc = 15 + 5RX + 5HY \quad (3.1)$$

donde *ecutwfc*, *RX* y *HY* son la energía de corte para las ondas planas, el número del radio y placas utilizadas para la estructura. Cada termino en la ecuación 3.1 tiene unidades de *Ry*.

- Recordando que las estructuras eran colocadas con los planos de grafeno paralelos al plano *XY* y siendo el eje *Y* la dirección donde se desplazaba la siguiente placa de grafeno apilada, se resumen en

las siguientes formulas los valores obtenidos para las dimensiones de las celdas unitarias de cada estructura:

$$l_x = R + (1,130 \text{ nm}) \quad (3.2)$$

$$l_y = R + (1,200 \text{ nm}) \quad (3.3)$$

$$l_z = H + (0,800 \text{ nm}) \quad (3.4)$$

Donde l_x , l_y y l_z son los lados de la celda unitaria rectangular, H y R son los valores de altura y radio de la estructura dibujada. Las variables l_x , l_y , l_z , H , R y factores independientes (2,0, 3,4 y 2,0) en las ecuaciones 3.2, 3.3 y 3.4 se encuentra en nm . En la figura 3.8 se observa una celda unitaria con dimensiones de $(1,8 \text{ nm}) \times (1,9 \text{ nm}) \times (1,5 \text{ nm})$ utilizada para la estructura *SGO R3H4*.

La razón de porque las estructuras *SGH R2H1*, *R2H2*, *R2H3* y *R2H4* fueron escogidas como estructuras de prueba para determinar los parámetros a utilizar en las demás estructuras se debido a la simpleza de estas. La sencillez de las estructuras *SGH R2HY* se debe a que estas no presentan átomos dopantes (oxígeno y nitrógeno), por lo tanto son un sistema computacionalmente menos demandante. Se intento trabajar con estructuras constituidas solamente de átomos de carbono, sin átomos de hidrógeno, oxígeno u nitrógeno, pero estas estructuras no eran estables bajo los funcionales XC de LDA y GGA, para el funcional de VdW no se hicieron pruebas. La inestabilidad de un estructura se observa en los resultados de optimización geométrica por dos comportamientos:

1. Se observa un aumento de la energía total del sistema a medida que las iteraciones (tiempo de cálculo) aumentan, similar al comportamiento observado en la figura 3.7 (a).
2. Los átomos de la estructura se comienza a alejar de tal manera que ocurren uno o más de los siguientes comportamientos:
 - a) Los átomos de carbono se alejan tanto entre ellos que se pueden considerar “ átomos aislados”.
 - b) Las átomos se colocan en celdas unitarias vecinas.
 - c) Las placas de grafeno se fragmentan en pequeñas cadenas de carbono.

Una optimización geométrica es adecuada porque los demás cálculos utilizan las coordenadas *XYZ* de los átomos en la estructura optimizada como *input*. Sino se utilizan estructura debidamente optimizadas los cálculos posteriores (simulación de absorción, DoS, etc.) tienen muchos problemas en converger, y si lo hacen generalmente conllevan a resultados que no son físicamente aceptables (Espectros de absorción simulados con valores negativos, por ejemplo). Todas las estructuras optimizadas con los parámetros mencionados no presentaron problemas en los cálculos posteriores.

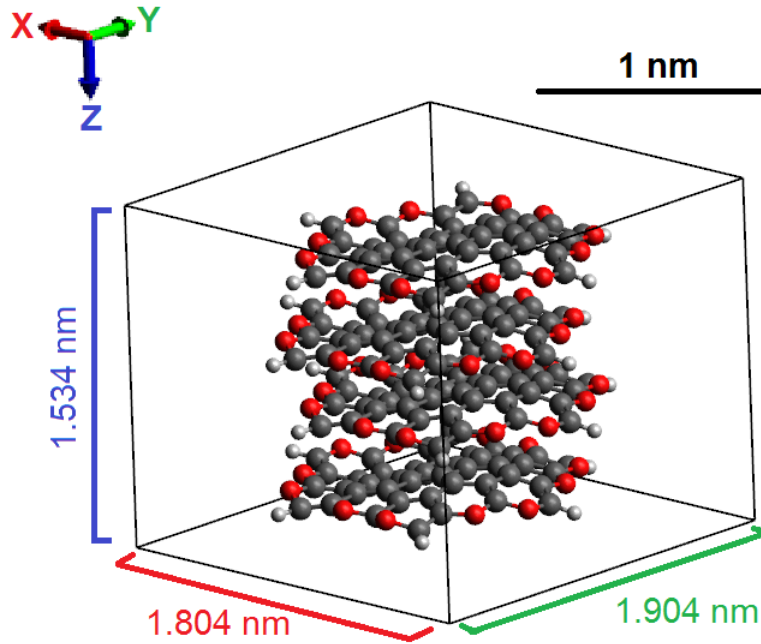


Figura 3.8: Celda unitaria utilizada para los cálculos de optimización geométrica para la estructura *SGO R3H4*. Estos parámetros de dimensiones fueron obtenidos con las ecuaciones 3.2, 3.3 y 3.4.

3.3. Espectros de absorción

Dos de los parámetros que afectan el espectro de absorción simulado son el número de iteraciones Lanczos y el ensanchamiento de las bandas de absorción. El número de iteraciones Lanczos permite obtener mejores aproximaciones del OS para el sistema, y por ende una mejor aproximación al espectro de absorción [41, 42]. Por otra parte, el ensanchamiento de las bandas de absorción intenta realizar una compensación para asemejar el espectro teórico con el experimental.

En las figuras 3.9 y 3.10 podemos observar simulaciones de espectros de absorción para la estructura *SGN R2H2*. Las simulaciones mostradas en 3.9 se realizaron manteniendo constante el ensanchamiento de las bandas de absorción y variando el número de iteraciones de Lanczos. Mientras, las simulaciones mostradas en 3.10 se realizaron manteniendo constante el número de iteraciones de Lanczos y variando el ensanchamiento de las bandas de absorción.

Al realizar simulaciones de espectros de absorción ciertas características espectrales, como bandas de absorción, aparecen en los resultados. A medida que ciertos parámetros son modificados dichas características espectrales disminuyen o desaparecen. En la figura 3.9 se observan como pequeñas bandas de absorción disminuyen en intensidad de manera apreciables a medida que el número de iteraciones de Lanczos aumenta. De manera similar se puede observar en la figura 3.10, como ciertas bandas que poseían una intensidad pequeña son absorbidas y desaparecen debido al ensanchamiento de las bandas más prominentes. En ambas figuras se puede observar el OS para dicha simulación del espectro de absorción. El paquete de QE, *turboTDDFT*, obtiene el OS a partir de la diagonalizar la ecuación de Liouville la cual describe la respuesta de la densidad electrónica a una perturbación externa al sistema [41, 42]. La diagonalización de esta matriz es computacionalmente demandante, por ello *turboTDDFT* la resuelve aproximadamente por medio del algoritmo de Lanczos. En el proceso de calcular el espectro a partir del OS, *turboTDDFT* simplifica este procedimiento utilizando directamente los coeficientes de Lanczos. Durante este último paso es donde se incluye en los cálculos el ensanchamiento de las bandas de absorción. Este enfoque evita calcular de manera directa el OS y permite obtener de eficiente los espectros de absorción. Por lo tanto, para recuperar el OS y compararlo contra las simulaciones reali-

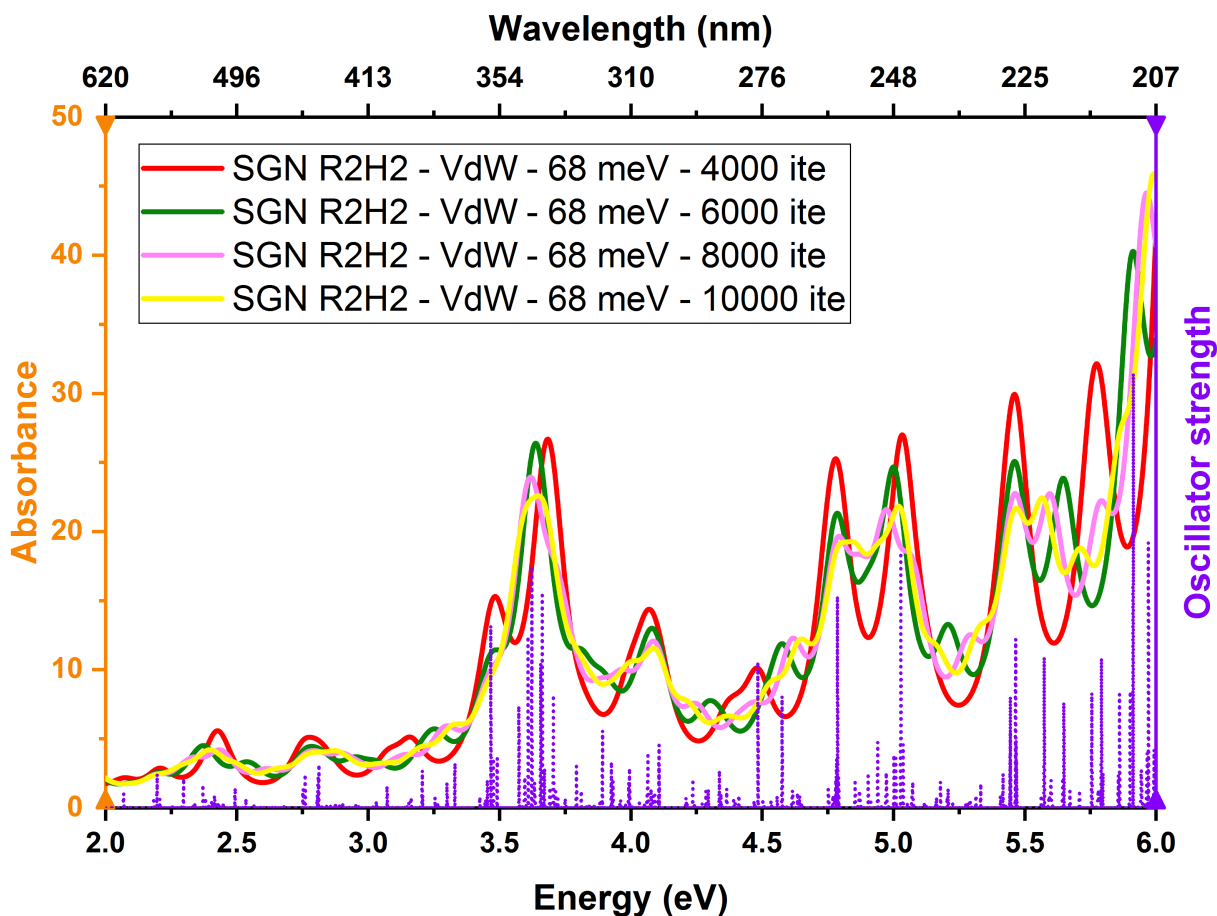


Figura 3.9: Comparación de la variación de los espectros de absorción respecto al número de iteraciones Lanczos. Se mantiene constante el ensanchamiento de las bandas de absorción en 68 meV (5 mRy) y se varía el número de iteraciones Lanczos en 4000, 6000, 8000 y 10000 iteraciones. El OS para dichos cálculos es mostrado por una línea punteada morada.

zadas se realizó el siguiente procedimiento. Se cálculo un espectro de absorción con un ensanchamiento sumamente pequeño, 1 meV ($0,1 \text{ mRy}$), y con alta resolución energética de $3000 \frac{\text{pixels}}{\text{eV}}$. Para motivos de comparación los espectros normales de absorción poseían una resolución energética de $1334 \frac{\text{pixels}}{\text{eV}}$. Una mayor resolución era necesaria porque al disminuir el ensanchamiento de las bandas, estas se hacían tan delgadas que si no se tenía una malla lo suficientemente fina en energías ciertas bandas serían omitidas. Esto conllevaría a obtener un OS el cual no podría describir ciertas bandas de absorción que si se observarían en otros espectros simulados con un mayor ensanchamiento.

En la figura 3.11 se observan los espectros de absorción simulados para la estructura *SGN R2H4* utilizando diferentes funcionales de XC. Todos los espectros simulados muestran las siguientes características similares. Se observan un conjunto de bandas de absorción alrededor de $5,0 \text{ eV}$ (248 nm), $3,7 \text{ eV}$ (335 nm) y $2,5 \text{ eV}$ (496 nm). A medida que se ensanchan los diferentes conjuntos de bandas mencionados, estas forman tres diferentes bandas de absorción definidas. La presencia de estas tres bandas de absorción esta de acuerdo con los reportes experimentales previos [1-3, 6, 7]. Por otra parte, la simulación utilizando el VdW funcional de XC muestra un mayor número de bandas alrededor de $5,0 \text{ eV}$ respecto a las simulaciones utilizando los funcionales de XC LDA y GGA. Los OS y espectros de absorción simulados con un ensanchamiento de 68 meV (5 mRy) muestran este comportamiento con mayor claridad. Otra diferencia importante entre las simulaciones utilizando el funcional de XC VdW y las obtenidas usando los funcionales de XC LDA y GGA ocurre con la banda de absorción de menor

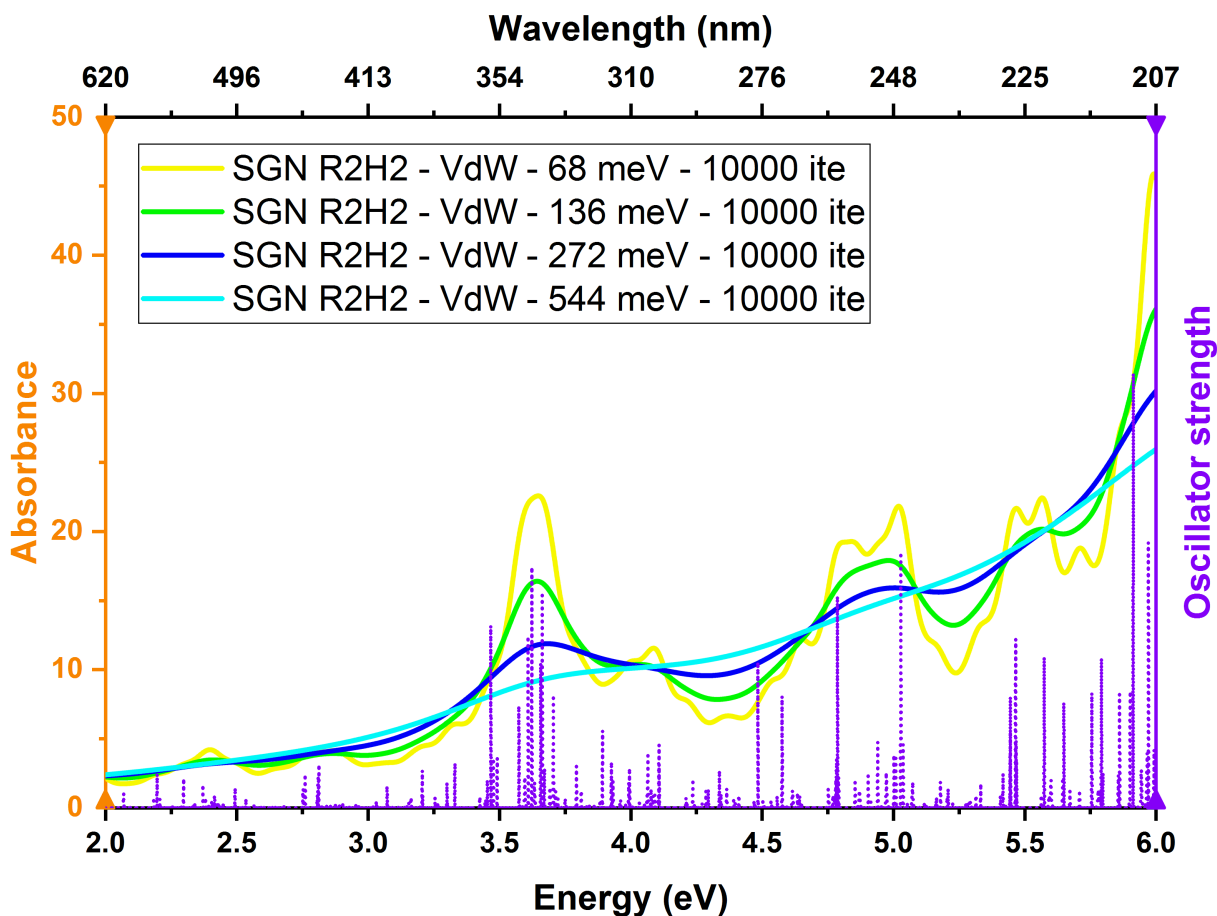


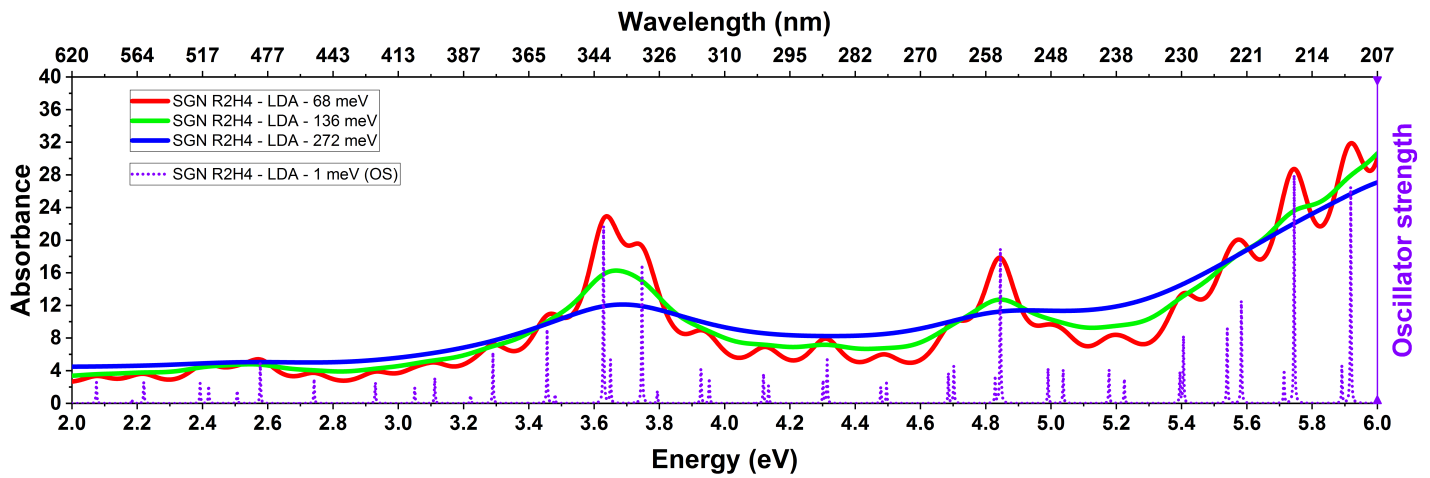
Figura 3.10: Comparación de la variación de los espectros de absorción respecto al ensanchamiento de las bandas. Se mantiene constante el número de iteraciones Lanczos en 10000, y se varia el ensanchamiento de las bandas de absorción en 68 meV (5 mRy), 136 meV (10 mRy), 272 meV (20 mRy) y 544 meV (40 mRy). El OS para dichos cálculos es mostrado por una línea punteada morada.

energía, centrada alrededor de $2,5 \text{ eV}$.

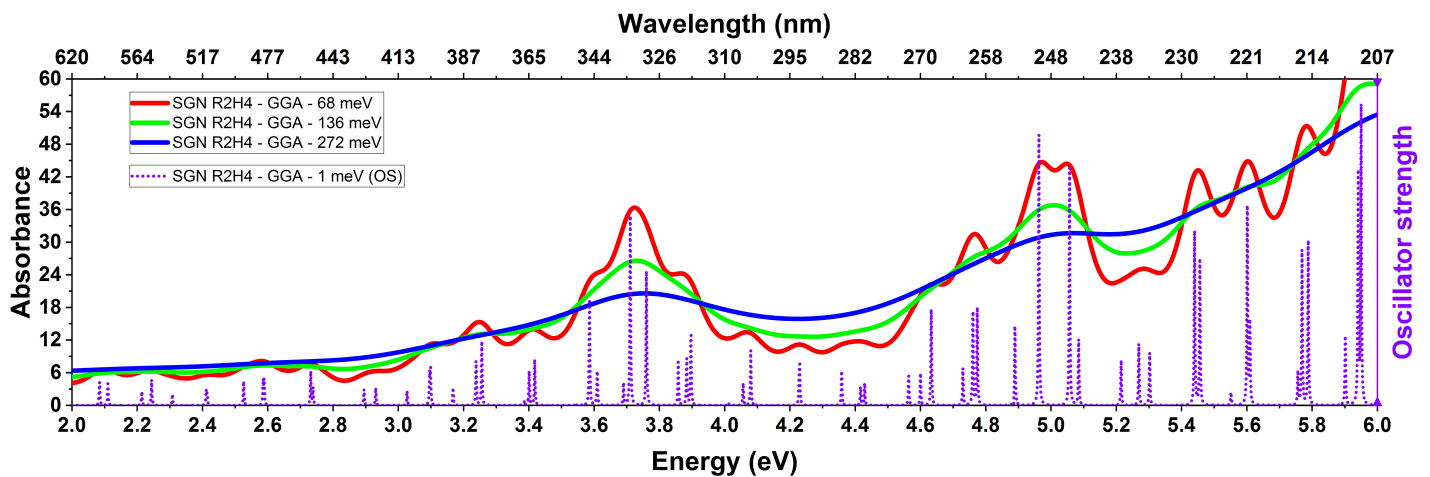
En esta banda de menor energía, los espectros de absorción simulados utilizando VdW muestran que la intensidad de esta banda es apreciable en comparación, a pesar de ser menor, a la intensidad de las otras dos bandas de mayor energía. En cambio, los espectros simulados utilizando LDA y GGA no logran definir de manera apreciable esta banda a menor energía. Los espectros simulados con un ensanchamiento de 272 meV graficados en la figura 3.11 muestran este comportamiento claramente.

En la figura 3.12 podemos observar una comparación directa entre espectros de absorción utilizando diferentes funcionales de XC para la estructura *SGN R2H4*. En esta comparación podemos observar como al ensancharse las bandas el cociente entre el máximo de absorción para las bandas centradas en $5,0 \text{ eV}$ y $3,7 \text{ eV}$ varia. Los espectros simulados utilizando LDA y GGA poseen un cociente entre estas dos bandas aproximado a 1. En el caso de las simulaciones con LDA este cociente es ligeramente menor a 1, esto es la absorción en la banda de $3,7 \text{ eV}$ es ligeramente mayor a la absorción de la banda de $5,0 \text{ eV}$. En cambio, el espectro simulado utilizando el funcional de XC VdW muestra el comportamiento contrario. El cociente entre el máximo absorción para las bandas centradas en $5,0 \text{ eV}$ y $3,7 \text{ eV}$ es mayor a 1 para esta simulación. Esto indica que la estructura absorbe mas fuertemente en la región UV del espectro electromagnético que en el VIS.

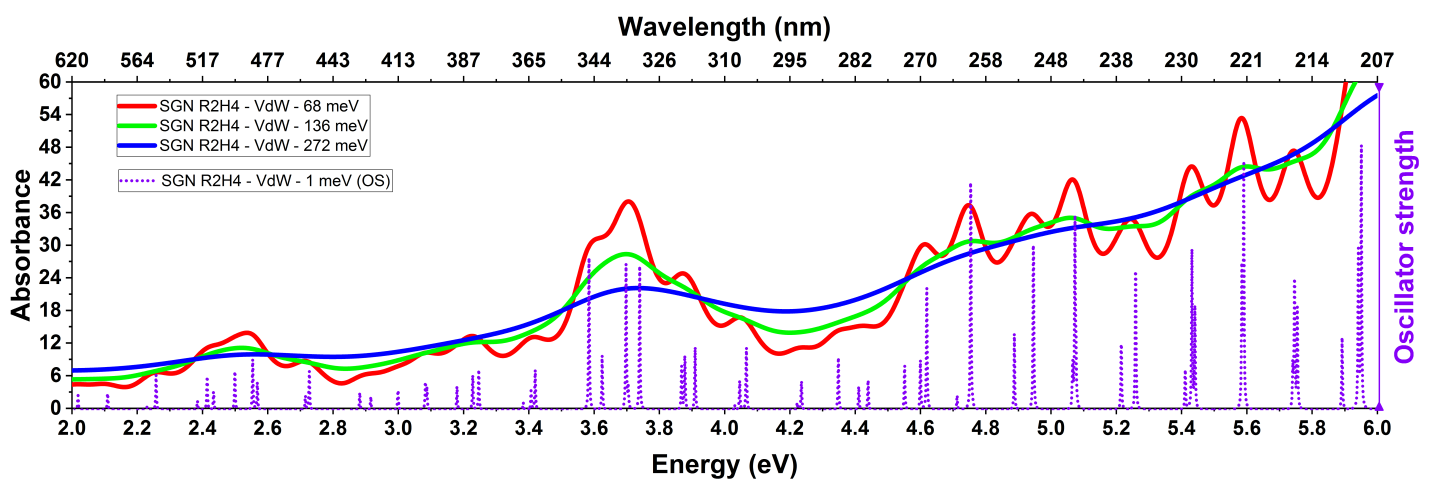
El comportamiento del cociente entre los máximos de absorción de las bandas centradas en $5,0 \text{ eV}$ y $3,7 \text{ eV}$ puede ser explicado de la siguiente manera. Los funcionales de XC LDA y GGA no pueden simular



(a) Espectros de absorción utilizando LDA.



(b) Espectros de absorción utilizando GGA.



(c) Espectros de absorción utilizando VdW.

Figura 3.11: Comparación de espectros de absorbancia simulados para la estructura *SGN R2H4*. En (a), (b) y (c) se comparan los espectros de absorbancia simulados variando el ensanchamiento para las transiciones utilizando los funcionales de XC LDA, GGA y VdW, respectivamente. Los ensanchamientos utilizados fueron 68 meV (5 mRy), 136 meV (272 mRy) y 68 meV (5 mRy). Todas las simulaciones utilizaron 8000 iteraciones de Lanczos.

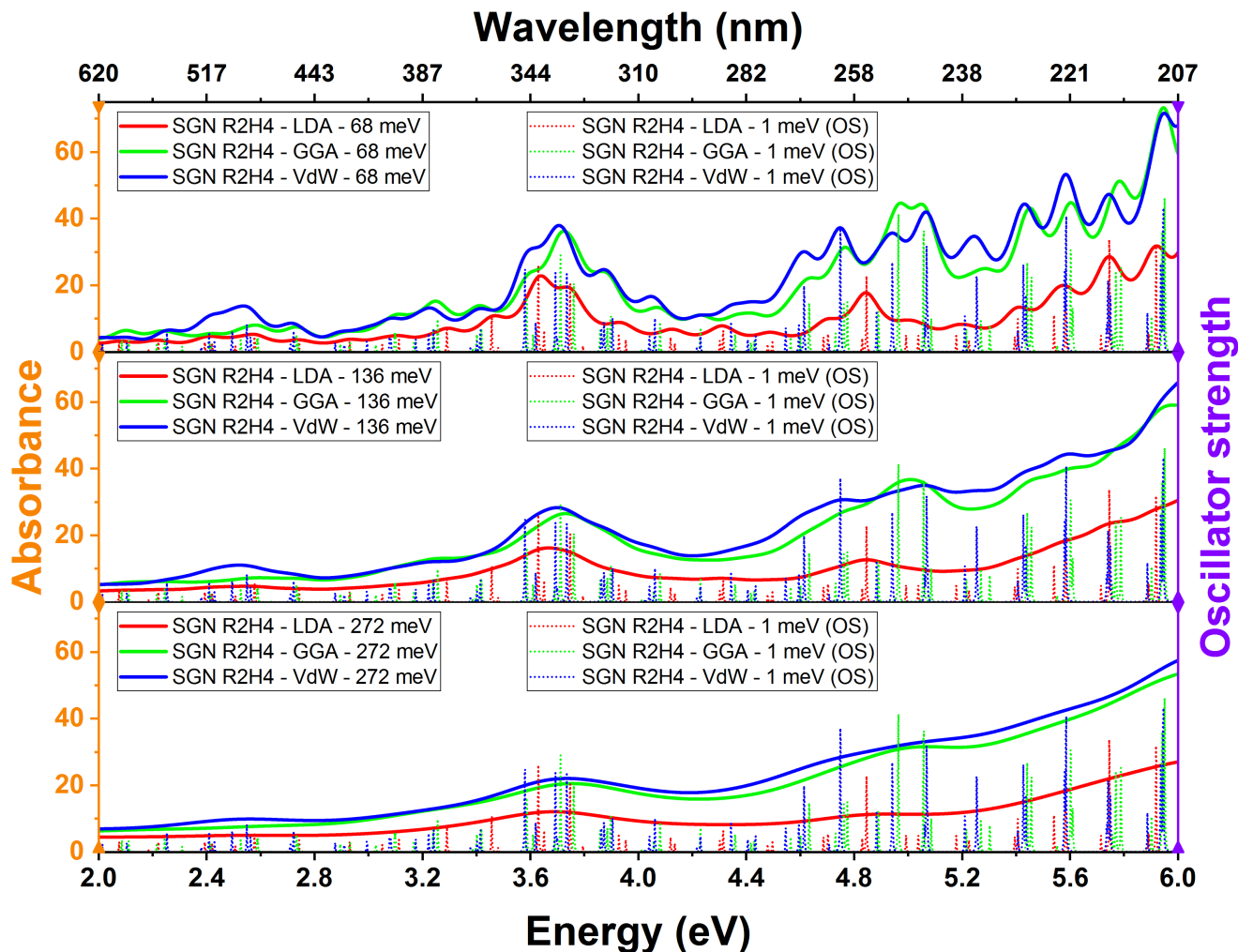


Figura 3.12: Comparación de espectros de absorbancia simulados para la estructura *SGN R2H4*. Los espectros simulados utilizando los diferentes funcionales de XC se comparan bajo un mismo ensanchamiento y número de iteraciones de Lanczos. Los ensanchamientos utilizados (de arriba hacia abajo) fueron 68 meV (5 mRy), 136 meV (272 mRy) y 68 meV (5 mRy). Todas las simulaciones utilizaron 10000 iteraciones de Lanczos.

muy bien interacciones no locales. Por ejemplo, LDA (y GGA de manera un poco menos marcada) subestiman la interacción de enlaces deslocalizados $\pi - \pi^*$ (ejemplo enlaces *CC* en el grafito) mientras sobrestiman la interacción de tipo de enlaces (ejemplo enlaces *CO* y *NO* en las moléculas de CO_2 y NO_2). Debido a que los GCQDs conforman un sistema con enlaces deslocalizados $\pi - \pi^*$ (*CC*) y además poseen enlaces *CO*, se puede comprender porque las simulaciones utilizando funcionales de XC LDA y GGA resultan en una sobrestimación de absorción mayor para la banda centrada $3,5 \text{ eV}$, la cual es asociada a transiciones $n - \pi$ (*OC*). En cambio, las simulaciones utilizando el funcional VdW generan un número mayor de sub-bandas de absorción alrededor de $5,0 \text{ eV}$, la cual es asociada a transiciones $\pi - \pi$ (*CC*). Al ensancharse estas sub-bandas, conforman una banda de absorción alrededor de 5 eV mucho más intensa que la banda de absorción alrededor de $3,7 \text{ eV}$.

En la figura 3.13 podemos observar una comparación entre los espectros de absorción experimentales y teóricos. En esta figura se puede observar como los espectros teóricos son capaces de simular los espectros experimentales. Tanto la presencia de tres bandas de absorción como la intensidad relativa entre estas bandas pueden ser reproducidas por la simulaciones utilizando el funcional de XC VdW.

Las simulaciones utilizando los funcionales XC de LDA y GGA no son capaces de reproducir todos

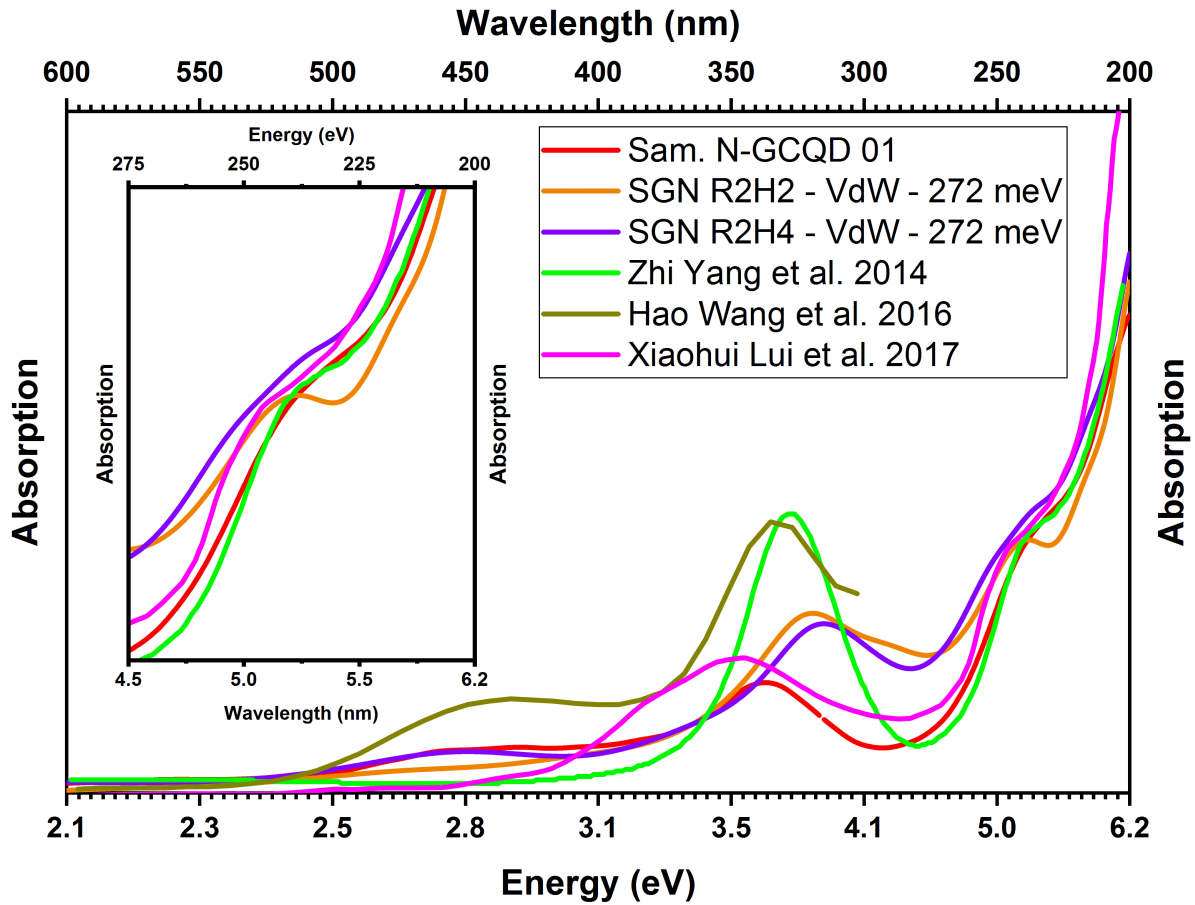


Figura 3.13: Comparación entre espectros de absorbancia experimentales y teóricos. En la inserción se puede observar un acercamiento a la banda asociada con las transiciones $\pi - \pi^*$ (CC). Los espectros simulados utilizaron 8000 iteraciones Lanczos.

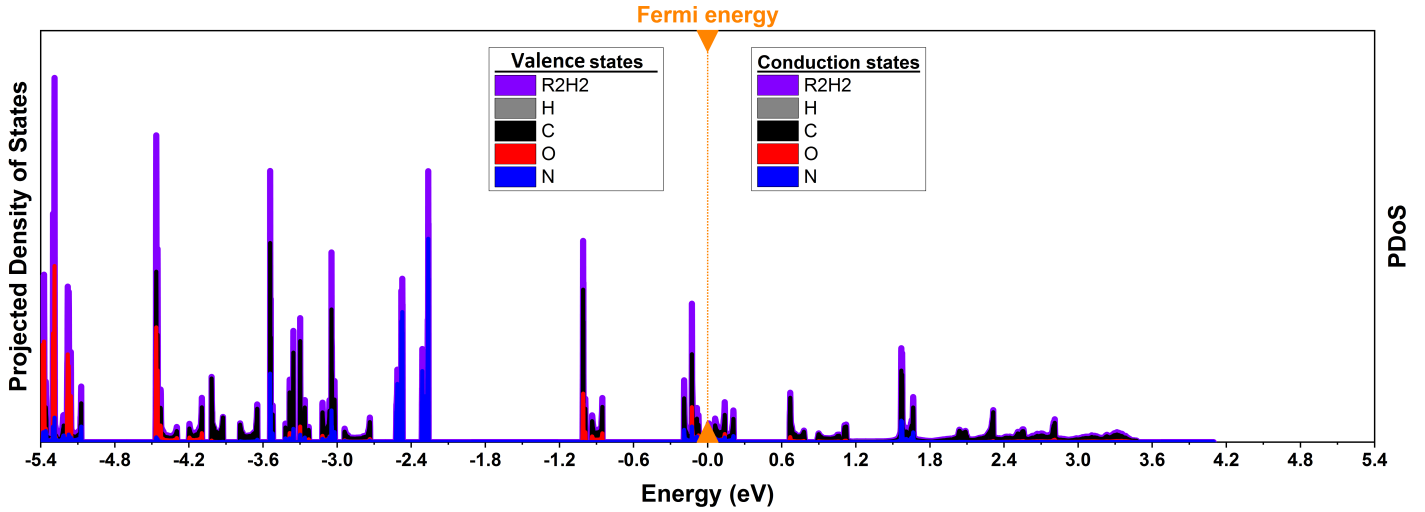
los comportamiento experimentales como el cociente entre los máximos de absorción y la presencia de la banda de energía centrada en $3,5 \text{ eV}$. Por ello, el funcional XC de VdW realiza una mejor descripción para las estructuras de GCQDs que LDA y GGA.

Finalmente, se puede observar la necesidad de ensanchar las bandas de absorción en los espectros teóricos dado que sin este ensanchamiento los espectros simulados no podrían reproducir el experimento. Los espectros de absorción teóricos, aparte de un ensanchamiento, también han sido desplazados al azul por una cantidad de $0,2 \text{ eV}$ para lograr una mejor descripción del experimento.

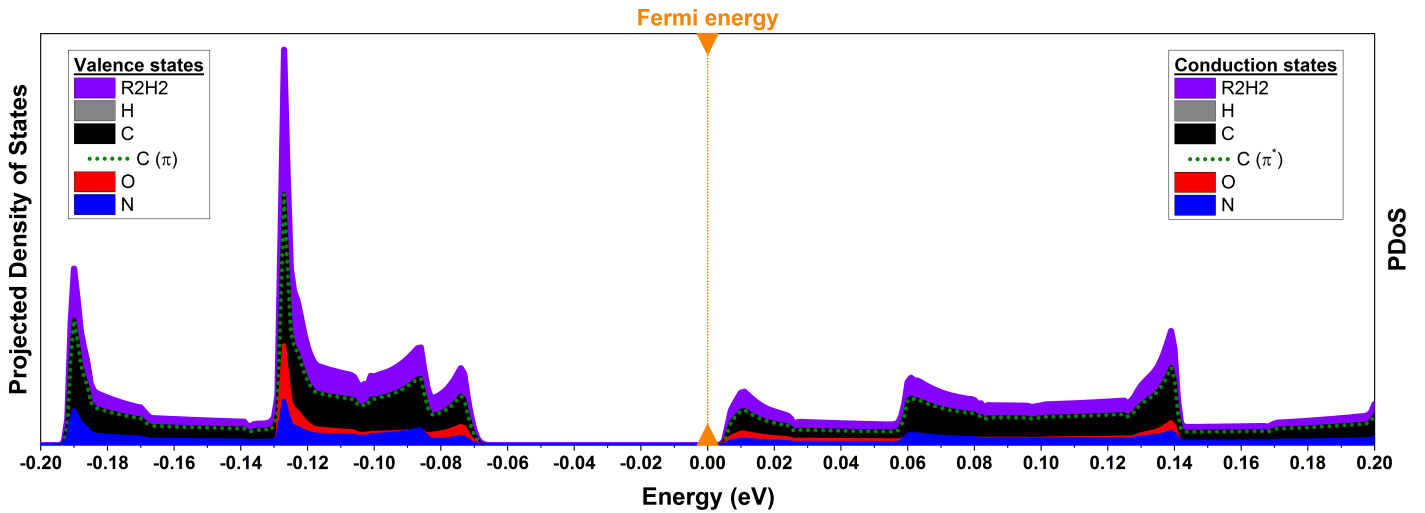
3.4. Origen de las transiciones a partir de DoS

El origen de las bandas de absorción observadas, tanto experimentalmente como teóricamente, puede ser estudiado a partir de los resultados obtenidos en los cálculo de DoS y PDoS. Debido a que los espectros de absorción son mejor simulados utilizando el funcional de XC VdW, el origen de las bandas de absorción sera estudiado utilizando cálculos de DoS y PDoS bajo el funcional de XC VdW principalmente. En la figura 3.15 se muestran secciones de la DoS y PDoS para la estructura *SGN R2H2*, tanto para estados por debajo (estados de valencia) como por encima (estados de conducción) de la energía de Fermi E_F .

Los análisis de las PDoS mostrados en la figura 3.15 han sido obtenidos bajo las siguientes consideraciones. El paquete *projwfc.x* origina un resultado de PDoS utilizando una nomenclatura atómica



(a) DoS y PDoS completo calculado para estructura *SGN R2H2*.



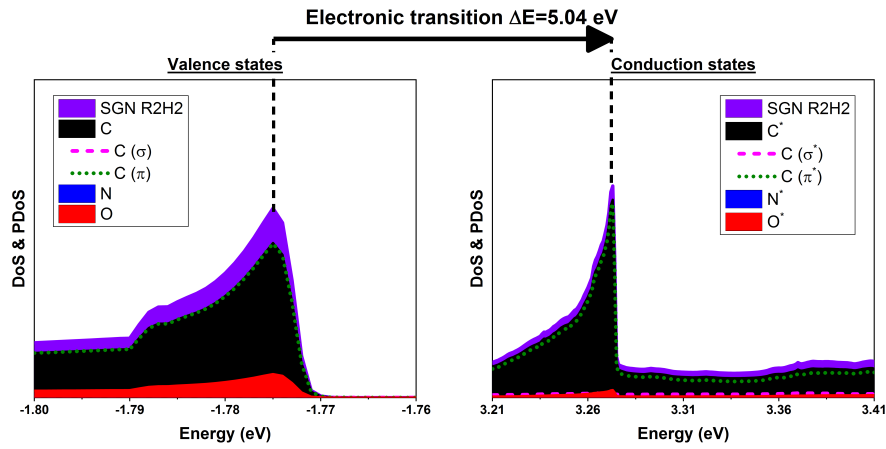
(b) Acercamiento alrededor de la energía de Fermi E_F .

Figura 3.14: DoS y PDoS para la estructura *SGN R2H2* utilizando funcional de XC VdW. Los DoS y PDoS han sido desplazados en energía de tal manera que la energía de Fermi E_F se encuentra colocada en el origen del eje de las energías, esto es $E_F = 0$.

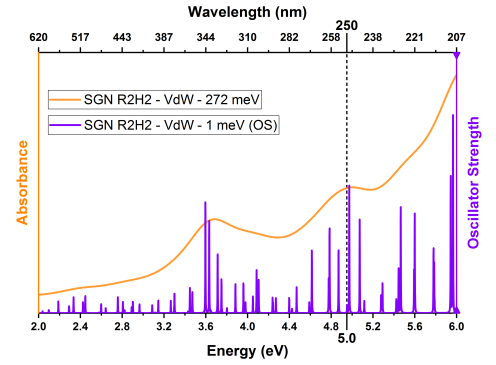
(s , p_z , p_y y p_x). Basados en la estructuras utilizadas en este trabajo, los átomos de C , O y N se han hibridado en sp^2 , sp y sp^2 , respectivamente. Por lo tanto, para los átomos de C se han tomado como estados π (pi^*) la suma de todos los orbitales p_z (p_z^*) que se encontraban debajo (encima) de la energía de Fermi. En cambio para los átomos de N y O se tomaron los orbitales no ligantes n como la suma de los orbitales s y s con p_z , respectivamente.

La diferencia de energía entre los estados de valencia y conducción mostrados en la figura 3.15 poseen un valor cercano a las tres bandas de absorción observadas, centradas alrededor de $2,7 \text{ eV}$, $3,7 \text{ eV}$ y $5,0 \text{ eV}$. En 3.15 (a) se observan que los estados de valencia y conducción están dominados por estados π (C) y π^* (C), respectivamente. A partir de estos resultados, la banda de absorción observada alrededor $5,0 \text{ eV}$, señalada en la figura 3.15 (d), podría ser asociada a una transición electrónica entre estados $\pi - \pi^*$ (CC).

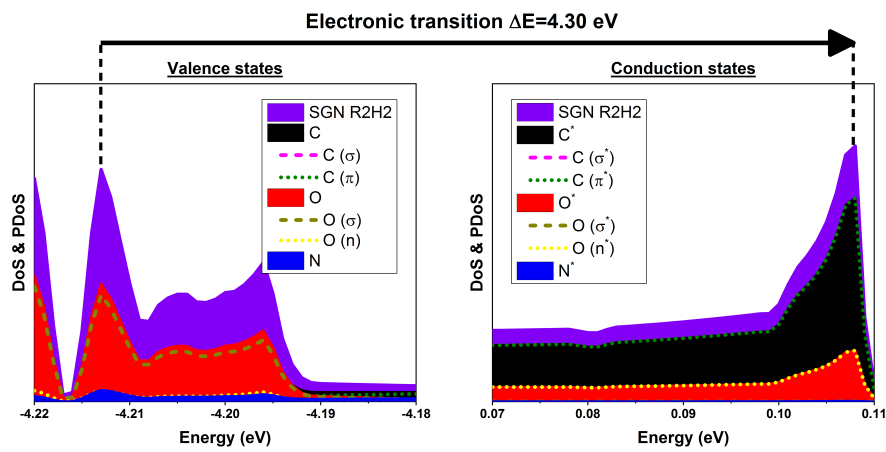
Para las bandas de absorción observada alrededor de $3,7 \text{ eV}$ y $2,7 \text{ eV}$, señaladas en las figuras 3.15 (d) y 3.15 (f), se puede realizar un análisis similar al hecho para la banda mas energética. En 3.15 (c) se observan que los estados de valencia y conducción están dominados por estados σ (O) y π^*



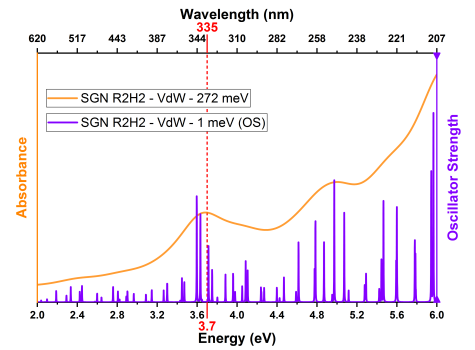
(a) DoS y PDoS para transición electrónica CC



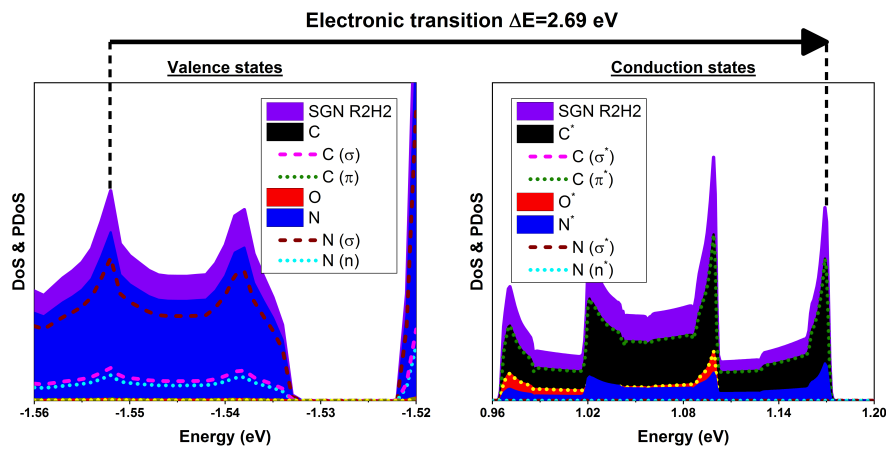
(b) Banda de absorción asociada a transición CC .



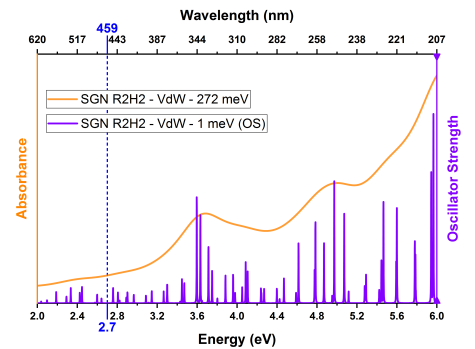
(c) DoS y PDoS para transición electrónica OC



(d) Banda de absorción asociada a transición OC .



(e) DoS y PDoS para transición electrónica NC

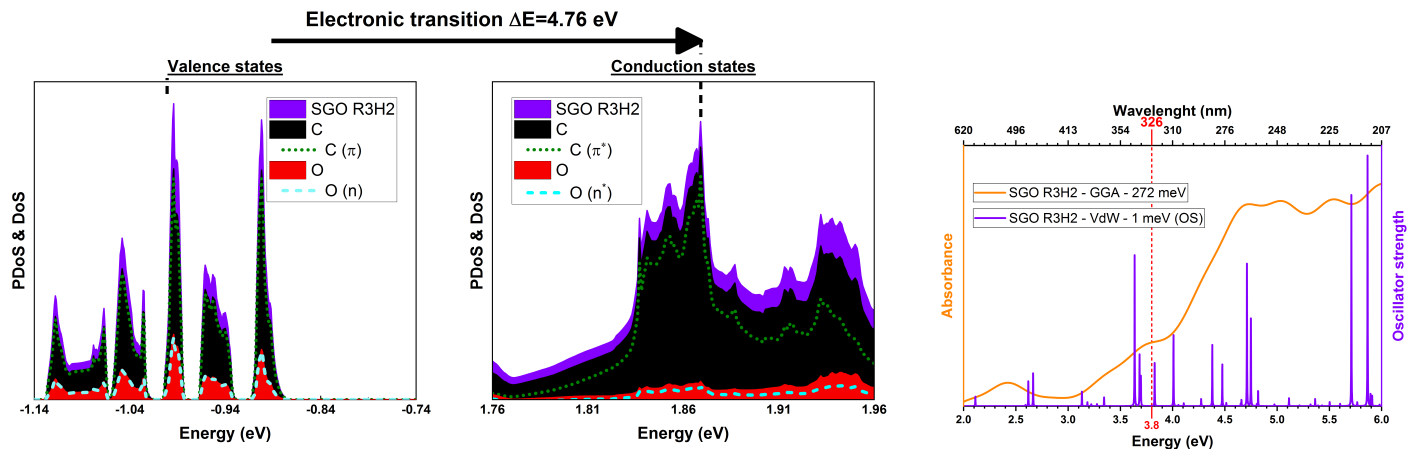


(f) Banda de absorción asociada a transición NC .

Figura 3.15: DoS y PDoS para estructura *SGN R2H2* obtenidos utilizando funcional de XC VdW. (a), (c) y (e) muestran los estados de valencia y conducción para las transiciones electrónicas que podrían dar origen a las bandas de absorción observadas, experimental y teóricamente, centradas alrededor de (d) 5,0 eV, (d) 3,7 eV y (f) 2,7 eV, respectivamente.

(C), respectivamente. Mientras, en 3.15 (e) se observan que los estados de valencia y conducción están dominados por estados σ (N) y π^* (C), respectivamente. Por lo tanto, las bandas de absorción observadas alrededor $3,7 \text{ eV}$ y $2,7 \text{ eV}$ podrían ser asociadas a transiciones electrónicas entre estados $\sigma - \pi^*$ (OC) y $\sigma - \pi^*$ (NC). Todas las estructuras estudiadas en este trabajo muestran este mismo comportamiento para el origen de las tres bandas de absorción observadas.

Surgen dos diferencias con los resultados obtenidos mediante las transiciones asociadas a OC y NC mediante los resultados de DoS en este trabajo y los diversos reportes de otros autores. Primeramente la energía de la banda asociada a OC es sobreestimada por $0,60 \text{ eV}$. Como segundo punto, la transición asignadas por otros autores a las bandas de OC y NC son estados $n - \pi^*$ (OC) y $n - \pi^*$ (NC), lo cual difiere con los resultados de este trabajo. Dicha diferencias en el origen de las bandas podría deberse a utilizar PP basados en GGA a pesar de que se utilizo un funcional de XC VdW. Otra alternativa es que el funcional XC VdW no pueda simular de manera correcta esta transición. Sin embargo, los funcionales LDA y GGA asocian ambas de estas bandas de absorción (OC y NC) a transiciones $\pi - \pi^*$ (CC), por lo tanto ambos son superados en este aspecto por VdW. En la figura 3.16 se puede observar como GGA asocia de manera incorrecta el origen de la transición OC para la estructura SGO R3H2. Todas las estructuras presenta utilizando LDA o GGA muestran este mismo comportamiento para la asignación del origen de la transiciones.

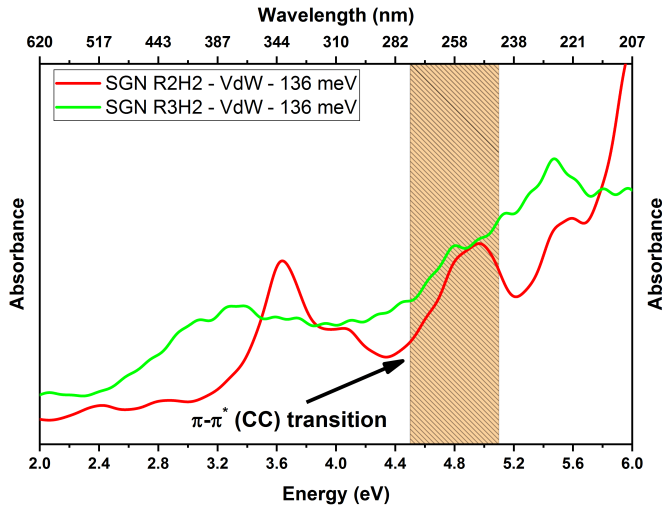


(a) DoS y PDoS para transición electrónica CC en estructura SGO R3H2. (b) Banda de absorción asociada a transición CC en estructura SGO R3H2.

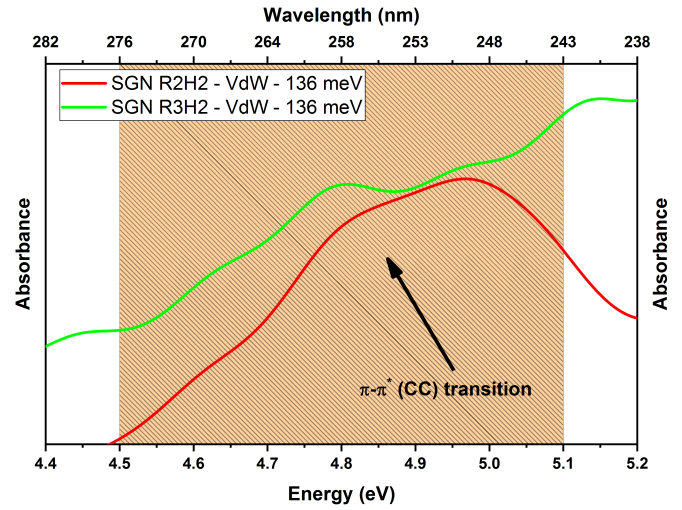
Figura 3.16: Bandas de absorción, DoS y PDoS para la estructura SGO R3H2 relacionada con la transición $n - \pi^*$ (OC) obtenida utilizando GGA. (a) muestra los estados de valencia y conducción para la transición electrónica $\pi - \pi^*$ (CC). Se puede que los estados de valencia estan dominados por estados π (C) y seguido de estados n (O). (b) muestra la banda de absorción relacionada con la transición electrónica señalada en su respectiva DoS.

3.5. Efecto del radio y número de placas de las estructuras en los espectros de absorción

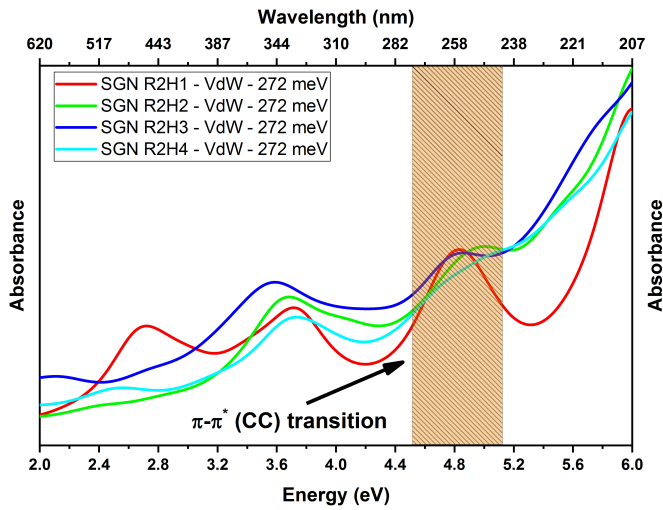
En la figura 3.17 se muestran comparaciones de los espectros de absorción simulados, utilizando el funcional de XC VdW, para estructuras con distintos parámetros geométricos. En todos ellos se observan cambios en los espectros de absorción, en particular en la banda de absorción asociada a la transición $\pi - \pi^*$ (CC). En 3.17 (a) y 3.17 (b) se muestran los espectros simulados para las estructuras SGN R2H2 y R3H2.



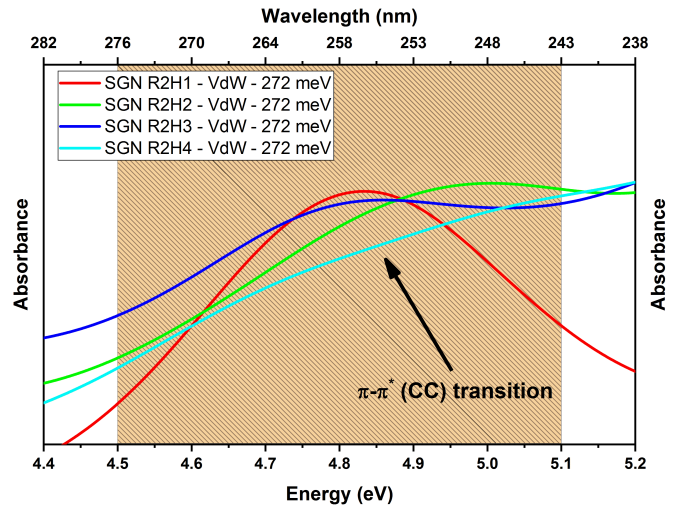
(a) Comparación de espectros de absorción simulados variando el radio de la estructura.



(b) Acercamiento alrededor de la banda de absorción asociada a la transición $\pi - \pi^*$ (CC) para el caso de radio variable.



(c) Comparación de espectros de absorción simulados variando el número de placas de la estructura.



(d) Acercamiento alrededor de la banda de absorción asociada a la transición $\pi - \pi^*$ (CC) para el caso de número de placas variable.

Figura 3.17: Comparación entre espectros de absorción simulados utilizando funcional de XC VdW bajo cambios en los parámetros geométricos de la estructura. (a) comparación de los espectros simulados para las estructuras *SGN R2H2* y *R3H2*. (b) acercamiento en la comparación de la banda de absorción $\pi - \pi^*$ (CC) entre las estructuras *SGN R2H2* y *R3H2*. El QCE puede ser observado para esta banda al variar el radio de la estructura. (c) comparación de los espectros simulados para las estructuras *SGN R2H1*, *R2H2*, *R2H3* y *R2H4*. (d) acercamiento en la comparación de la banda de absorción $\pi - \pi^*$ (CC) entre las estructuras *SGN R2H1*, *R2H2*, *R2H3* y *R2H4*. Se puede observar una dependencia entre el centro de la banda de absorción $\pi - \pi^*$ (CC) y el número de placas de la estructura. La banda de absorción relacionada a la transición $\pi - \pi^*$ (CC) es señalada por la zona resaltada en naranja.

Se puede observar un desplazamiento al rojo en la banda de absorción $\pi - \pi^*$ (CC) para la estructura *SGN R2H3* respecto a la estructura *SGN R2H2*. Esto puede ser explicado a partir del QCE. La banda de absorción $\pi - \pi^*$ (CC) es relacionada con un excitón libre en la estructura [11, 12]. Por lo tanto, al aumentar el radio de la estructura el confinamiento de los portadores de carga disminuye y como consecuencia la banda de absorción relacionada con esta quasi-partícula se desplaza a menores energías o longitudes de onda mayores.

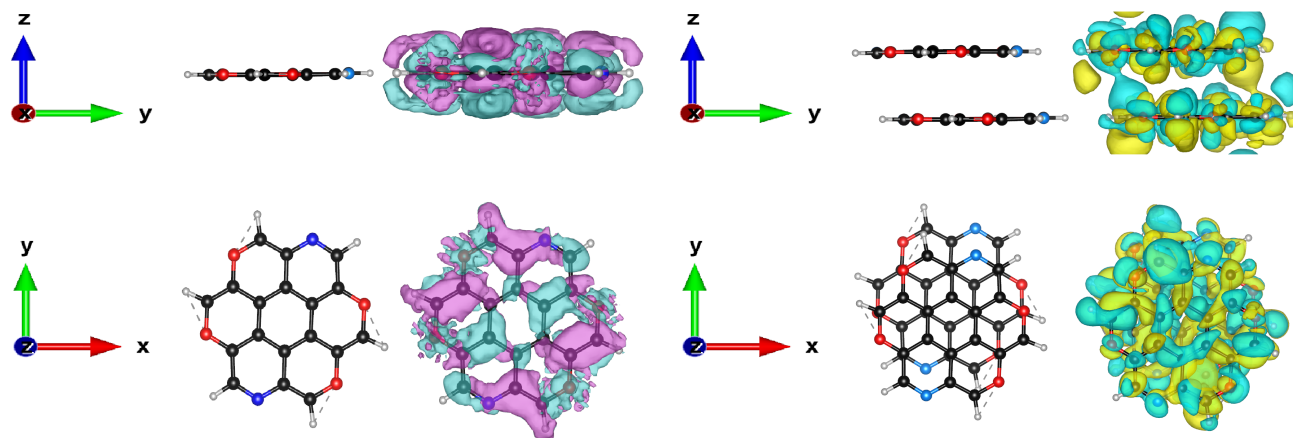
En 3.17 (c) y 3.17 (d) se muestran los espectros simulados para las estructuras *SGN R2H1*, *R2H2*, *R2H3* y *R2H4*. En estas figuras, se puede observar como la banda de absorción $\pi - \pi^*$ (CC) de las estructuras *SGN R2H2* y *R2H4* se ha desplazado a mayores energías respecto a la banda de absorción $\pi - \pi^*$ (CC) de las estructuras *SGN R2H1* y *R2H3*. El desplazamiento de estas bandas ronda los 200 meV. Se sabe debido a cálculos y experimentos, han encontrado que la energías relacionadas con las interacciones de VdW rondan el orden de los cientos de meV [16-18]. Debido que las simulaciones presentadas en 3.17 tratan de simular las interacciones de VdW en la estructura del GCQD, se plantea la posibilidad que estos desplazamientos en la banda de absorción $\pi - \pi^*$ (CC) sean originados por interacciones de VdW entre las placas de grafeno que conforman el GCQDs [4].

Con la intención de seguir estudiando una posible interacción entre placas de grafeno en los GCQDs, se han realizado cálculos de la respuesta electrónica del sistema para una perturbación que coincida con la energía de la transición $\pi - \pi^*$ (CC). Resultados de estos cálculos para estructuras *SGN R2HY* son mostrados en la figura 3.18. Para una sola placa de grafeno, la respuesta electrónica se localiza principalmente por debajo y encima de la placa como se observa en la figura 3.18 (a). Esta distribución de la densidad electrónica es muy similar a los enlaces deslocalizados π a los cuales se les asocia la banda de absorción centrada alrededor de 5,0 eV. En las figuras 3.18 (b), 3.18 (c) y 3.18 (d) se puede observar una densidad electrónica compartida entre las diferentes placas para las estructuras *SGN R2H2*, *R2H3* y *R2H4*. Estos resultados pueden ser interpretados como una interacción entre los orbitales π^* (C) de las placas de grafeno presentes en la estructura.

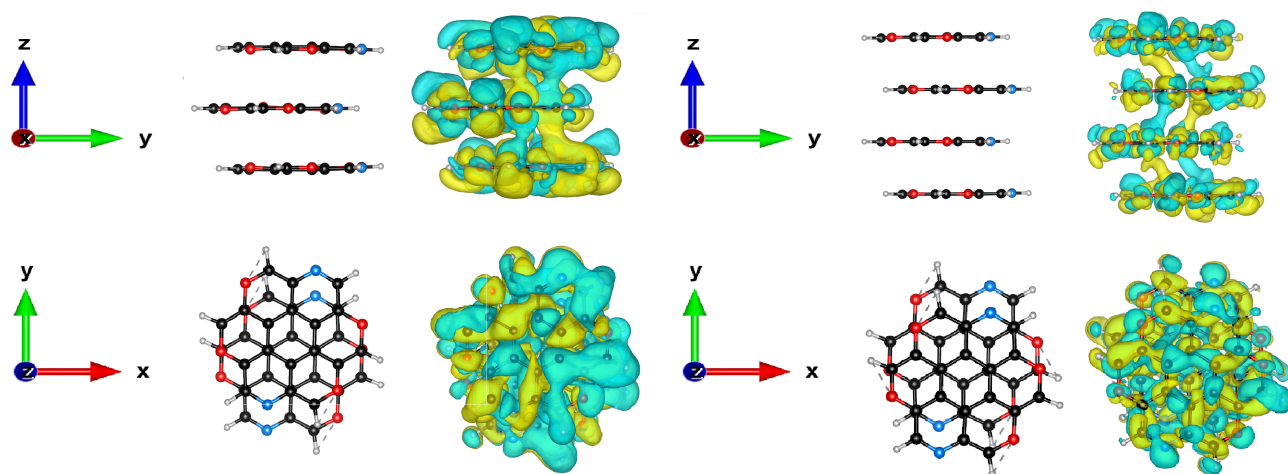
Esta ultima afirmación puede ser respaldada a través de los cálculos de DoS y PDoS. En la figura 3.19 se muestra la contribución de cada placa de grafeno a la DoS y PDoS total de las diversas estructuras *SGN R2HY*. Además también se muestra la banda de absorción que dicha transición electrónica origina. Como fue discutido previamente, el resultado que los estados de valencia y conducción sean dominados por estados π (C) y π^* (C), en todas estas estructuras, podría ser asociado con una transición electrónica $\pi - \pi^*$ (CC). En la figura 3.19 también muestra las contribuciones que cada placa de grafeno aporta a los estados de valencia (conducción) π (π^*) totales de la estructuras. De las figuras 3.19 (a) como era de esperarse en la estructura *SGN R2H1* tanto los estado de valencia como de conducción son debido a la única placa de grafeno existente en la estructura. Por otro lado, en la figura 3.19 (c) observamos que los estados de valencia y conducción son proporcionados de manera equitativa por ambas placas de grafeno presentes en la estructura *SGN R2H2*. Finalmente, en la figura 3.19 (e) observamos que para la estructura *SGN R2H3* los estados de valencia están dominados por los estados π de la primera L_1 y tercera placa L_3 , mientras los estados de conducción están dominados por estados π^* de la segunda placa L_2 .

La estructura *SGN R2H4* muestra similitudes a las otras estructuras *SGN R2HY*, pero también comportamientos muy característicos de esta. Algo muy particular de la estructura *SGN R2H4*, es que la banda de absorción asociada a la transición $\pi - \pi^*$ (CC) muestra un hombro, lo cual incrementa la anchura de esta banda. Analizando los resultados de DoS y PDoS para esta estructura, se encontraron dos transiciones electrónicas posibles, una cercana al hombro de la banda y la otra al máximo de esta. Por lo tanto en la figura 3.20 se muestran la DoS y PDoS, así como las bandas de absorción asociadas, a estas dos transiciones electrónicas $\pi - \pi^*$ (CC) para la estructura *SGN R2H4*. La primera transición tiene una diferencia de energía muy similar a la observada en las estructuras *SGN R2H1* y *R2H3*, mientras la segunda transición tiene una diferencia de energía más cercana con la estructura *SGN R2H2*. Debido a las características espectrales observadas en el espectro de absorción de la estructura *SGN R2H4* alrededor de estas energías y junto con los resultados de la DoS, podríamos concluir que hay dos bandas de absorción superpuestas en esta zona. Una de estas bandas, la centrada en 4,76 eV, posee una probabilidad de transición menor que la otra, la centrada en 4,97 eV. La superposición de estas genera la banda de absorción junto con el hombro que observamos en el espectro de absorción.

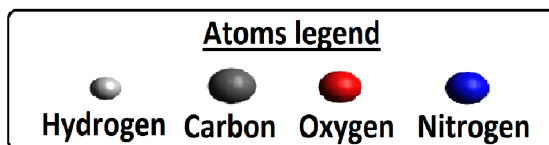
Chen et. al. propone dos funciones de energía potencial, una para placas vecinas en apilamiento AB (ϕ_{AB}) y otra para placas en apilamiento AA (ϕ_{AA}). Ambas funciones son mostradas en 3.21 (b). Para



(a) Respuesta electrónica para estructuras *SGN R2H1*. (b) Respuesta electrónica para estructuras *SGN R2H2*.



(c) Respuesta electrónica para estructuras *SGN R2H3*. (d) Respuesta electrónica para estructuras *SGN R2H4*.



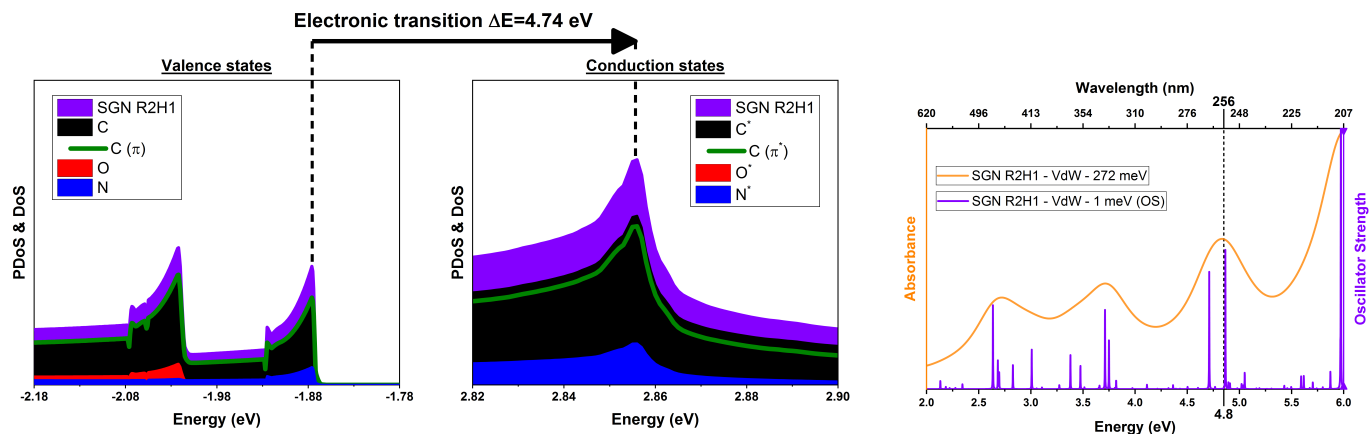
(e) Leyenda para estructuras *SGN R2HY* mostradas en cálculos de respuesta electrónica.

Figura 3.18: Respuesta electrónica obtenidas utilizando funcional de XC VdW para estructuras *SGN R2H3*. Respuesta electrónica para las estructuras *SGN* con (a) 1 placa (*R2H1*), (b) 2 placas (*R2H2*), (c) 3 placas (*R2H3*) y (d) 4 placas (*R2H4*) de grafeno dopado.

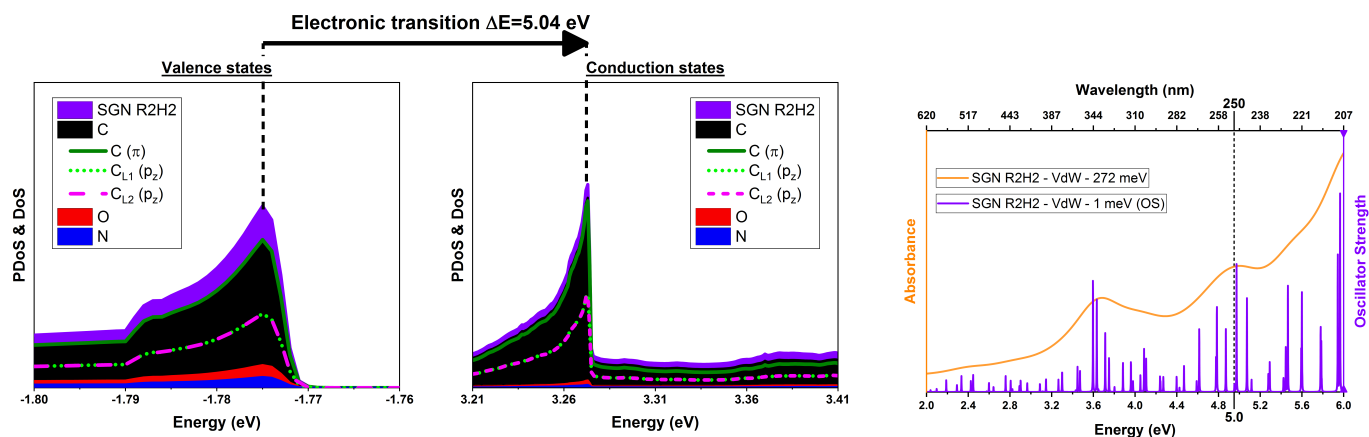
nuestro modelos debido a que habrán interacciones entre placas con apilamiento AA y AB se utilizara un superposición entre ambas energías potenciales (ϕ_{Mix}). La superposición constara de la siguiente suma pesada las energías potenciales ϕ_{AA} y ϕ_{AB} :

$$\phi_{Mix} = 0,25\phi_{AA} + 0,025\phi_{AB} \quad (3.5)$$

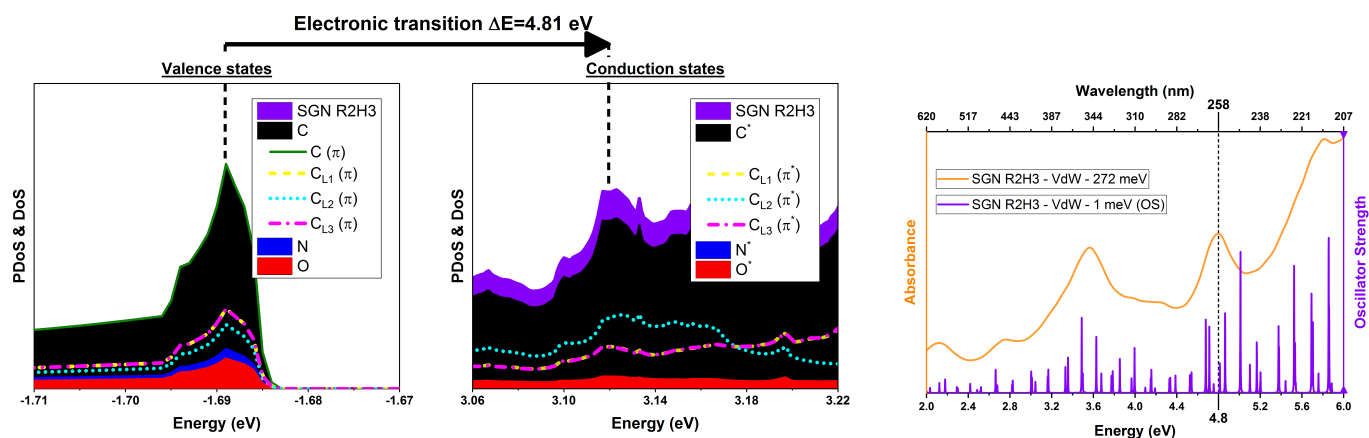
Los valores de superposición pesados han sido escogidos de tal manera que ϕ_{AA} domine cerca de la



(a) DoS y PDoS para transición electrónica CC en estructura $SGN R2H1$. (b) Banda de absorción asociada a transición CC en estructura $SGN R2H1$.

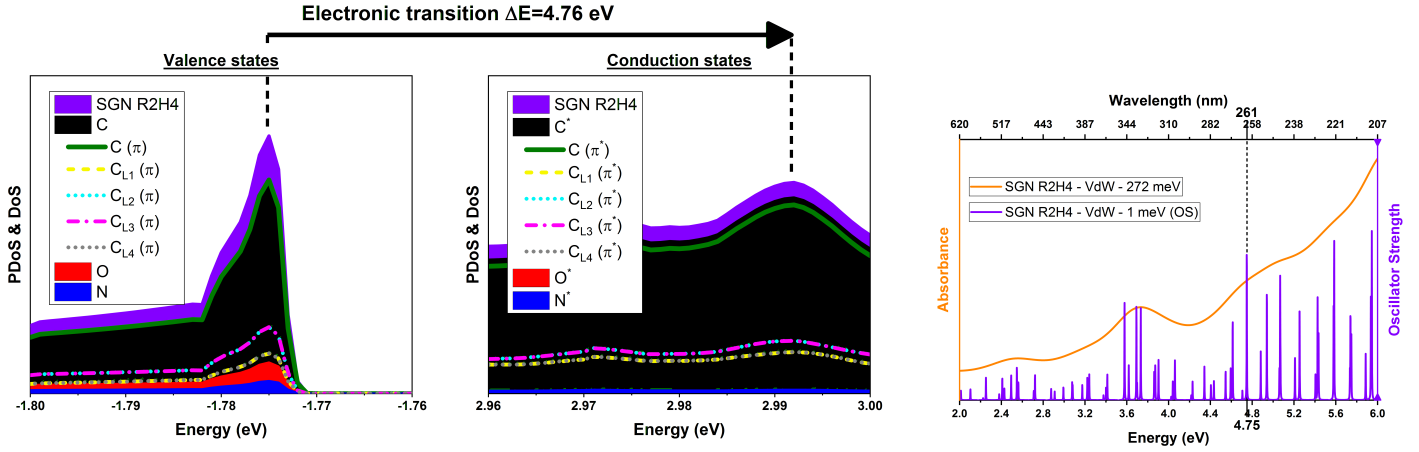


(c) DoS y PDoS para transición electrónica CC en estructura $SGN R2H2$. (d) Banda de absorción asociada a transición CC en estructura $SGN R2H2$.

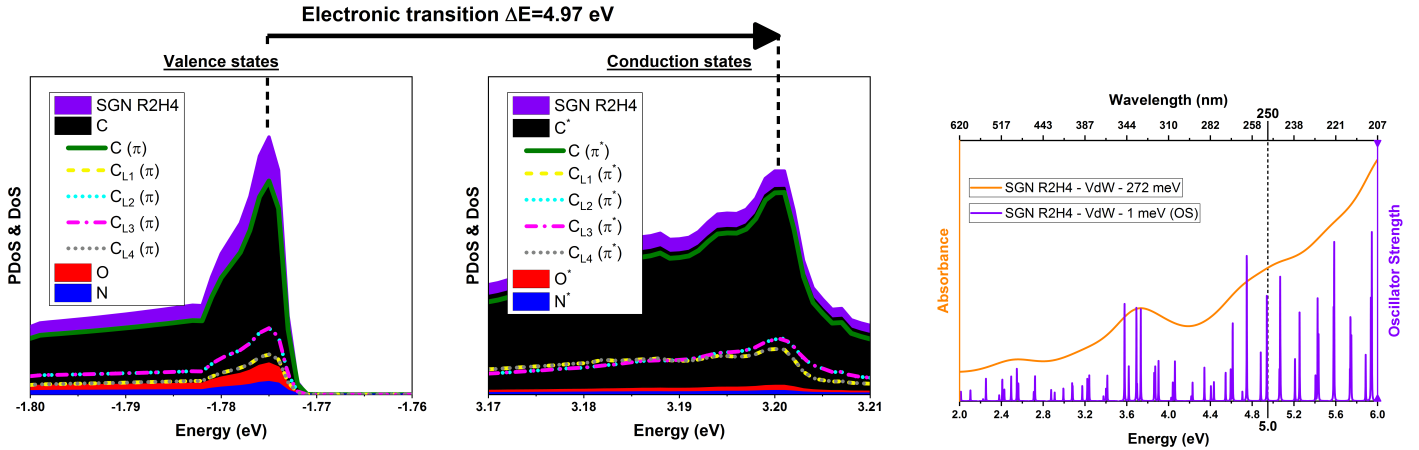


(e) DoS y PDoS para transición electrónica CC en estructura $SGN R2H3$. (f) Banda de absorción asociada a transición CC en estructura $SGN R2H3$.

Figura 3.19: Bandas de absorción, DoS y PDoS para las estructuras $SGN R2H1$, $R2H2$ y $R2H3$ relacionadas con la transición $\pi - \pi^*$ (CC) obtenidas utilizando el funcional de XC VdW. (a), (a) y (e) muestran la contribución de los orbitales π (C) que cada capa de grafeno aporta a los estados de valencia y conducción para la transición electrónica $\pi - \pi^*$ (CC). (b), (d) y (f) muestran las bandas de absorción relacionadas con la transición electrónica $\pi - \pi^*$ (CC) para cada estructura.



(a) DoS y PDoS para transición electrónica CC menos intensa en estructura $SGN R2H4$.

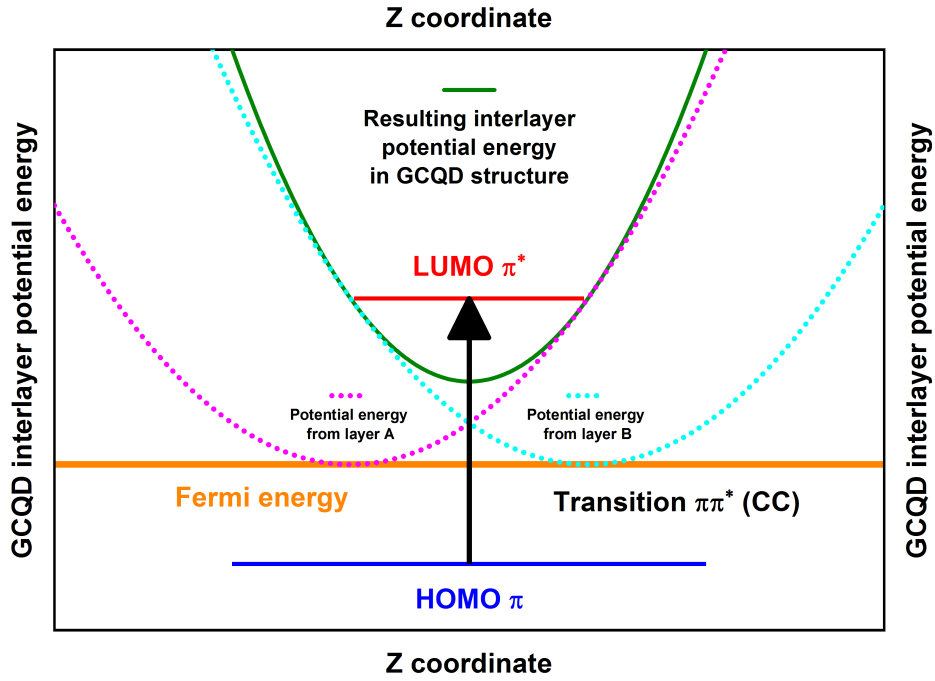


(c) DoS y PDoS para transición electrónica CC más intensa en estructura $SGN R2H4$.

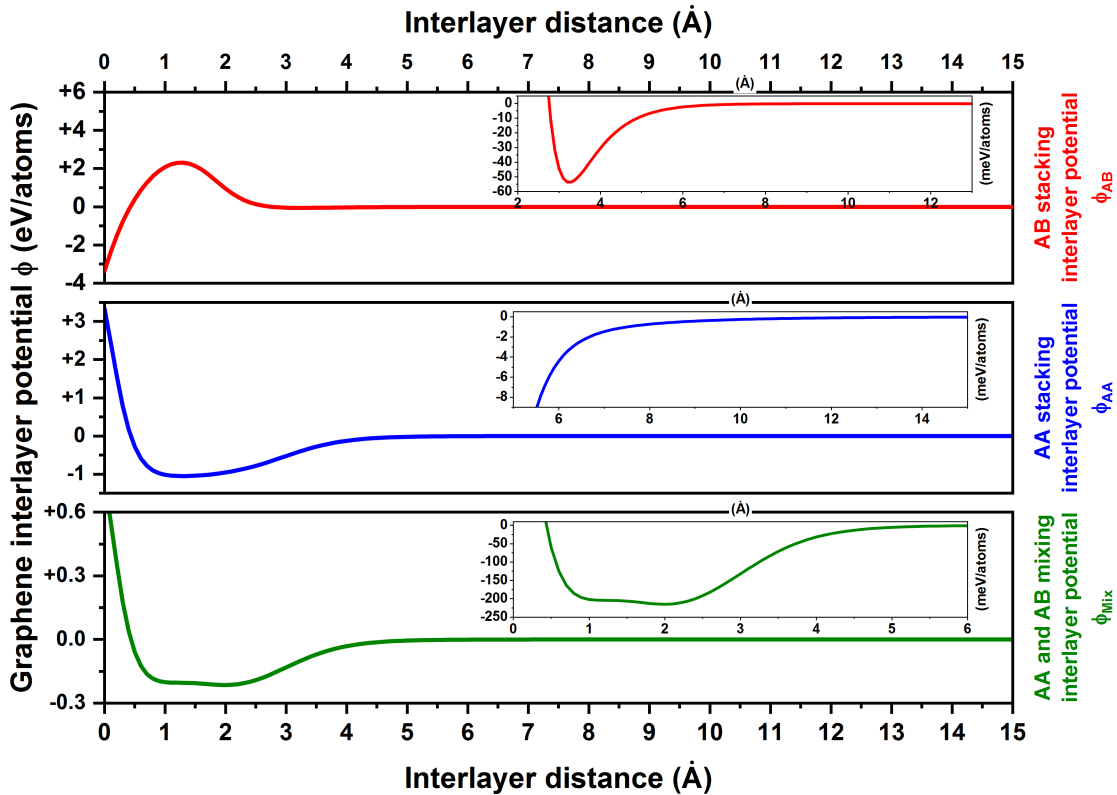
Figura 3.20: Bandas de absorción, DoS y PDoS para la estructura $SGN R2H4$ relacionadas con la transición $\pi - \pi^*$ (CC) obtenidas utilizando el funcional de XC VdW. (a) y (c) muestran la contribución de los orbitales π (C) que cada capa de grafeno aporta a los estados de valencia y conducción para la transición electrónica $\pi - \pi^*$ (CC). (b) y (d) muestran las bandas de absorción relacionadas con cada transición electrónica $\pi - \pi^*$ (CC) mencionada previamente.

placa de grafeno. Esto es porque la densidad electrónica de los enlaces σ no permiten a los electrones excitados en estados π^* acercarse demasiado a la placa de grafeno. En este modelo, cada placa de grafeno poseerá una energía potencial ϕ_{Mix} con su origen desplazado a la posición de la placa. Se considera que tanto por encima como por debajo de la placa se generara la misma energía potencial. El mínimo de La energía potencial resultante de la superposición de las energías potenciales individuales las placas de grafeno sera colocado en la energía de Fermi E_F , la cual se ha tomado con valor igual a cero ($E_F = 0$). Estas consideraciones también son mostradas en la figura 3.21. Se ha tomado como el valor de HOMO para este modelo el promedio de las energías de los estados de valencia obtenidos para las transiciones $\pi - \pi^*$ (CC) obtenidas por DFT. En la tabla 3.1 se resume los valores de las energías para valencia y conduccion obtenidas por DFT usando VdW y también se muestran los valores de energía de HOMO usado y los valores del primero (LUMO), segundo y tercer estado excitado obtenidos por este método.

Estos dos niveles cercanos permiten que en una estructura $R2H4$ sean permitidas dos transiciones $\pi -$



(a) Diagrama de energías para la transición $\pi\pi^*$ (CC) en el modelo propuesto.



(b) Funciones semi-empíricas para energías potenciales entre placas de grafeno [67].

Figura 3.21: Modelo propuesto para explicar el comportamiento de los espectros de absorción con el número de placas de la estructura. (a) Diagrama de energías para el modelo semi-empírico propuesto. En este se observa como la superposición de las energías potenciales de las placas de grafeno individuales genera una energía potencial modificada para toda la estructura. (b) Funciones semi-empíricas para las energías potenciales entre las placas de grafeno propuestas por Chen et. al. [67].

Una vez planteada la energía potencial de la estructura, se resolverá la ecuación de Schrödinger de manera numérica por medio del método de Numerov. En la figura 3.22 se observan el potencial resultante para una estructura *R2H1* sin dopamiento alguno. De esta podemos apreciar la energía potencial debida a una placa de grafeno individual y la simetría de esta por encima y debajo de la placa. También es importante notar que la probabilidad del electrón en este estado excitado es mayor a una distancia simétrica por encima y debajo del plano de la placa de grafeno. La solución de Schrödinger obtenida para el modelo semi-empírico en *R2H1* solo genero un estado excitado.

En las figuras 3.23, 3.24 y 3.25 se muestran de manera similar, las energías potenciales superpuestas obtenidas para las estructuras *R2H2*, *R2H3* y *R2H4*. También se presentan las probabilidades obtenidas para los primeros dos estados excitados de todas estas estructuras mencionadas. Un diagrama de los niveles de energía obtenidos numéricamente para todas las funciones de onda presentadas es mostrado en la figura 3.26.

A partir de toda esta información presentada, planteamos las siguientes interpretaciones. El modelo semi-empírico muestra que para la transición del estado base (HOMO) al primer estado excitado (LUMO) las estructuras *R2H1* con *R2H3* resultan en una diferencia de energía muy similar. Este mismo comportamiento fue observado para los espectros de absorción obtenidos con VdW para las estructuras *SGN R2H1* y *R2H3*, los cuales presentaban a la misma energía la banda asociada a la transición $\pi - \pi^*$ (CC).

De manera similar a los resultados de la DoS, PDoS y espectros de absorción para la estructura *SGN R2H3* por DFT, el método semi-empírico propuesto indica la existencia de dos niveles excitados cercanos en energía para la estructura *R2H4*, esto puede observarse en las figuras 3.26 y 3.25.

π^* (CC) con energías ligeramente diferentes. La primera transición para *R2H4*, esta sería del estado base (HOMO) al primer estado excitado (LUMO), posee una diferencia de energía similar a las transiciones observadas para *R2H1* y *R2H3*. Mientras, la segunda transición para *R2H4*, esta sería del estado base (HOMO) al segundo estado excitado, posee una diferencia de energía similar a la transición observada para *R2H2*. Los resultados obtenidos por DFT usando VdW para las estructuras *SGN R2H2* y *R2H4*, concuerdan con el modelo semi-empírico y la existencia de una transición $\pi - \pi^*$ (CC) con energía similares entre estas estructuras.

De las funciones de probabilidad obtenidas por este modelo semi-empírico, podemos observar donde se localiza la densidad electrónica para estos estados excitados. Por ejemplo, para la *R2H2* el primer estado excitado se localiza entre las dos placas de grafeno como se observa en la figura 3.23. Este comportamiento puede explicarse de la siguiente manera. Un electrón al ser promovido a un estado excitado, experimentara dos potenciales ligantes debido a la presencia de las dos placas de grafeno.

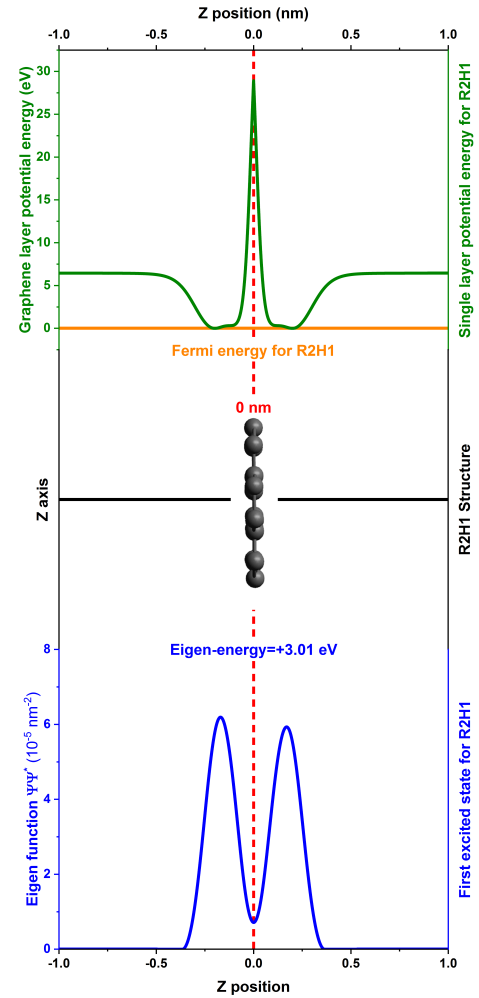
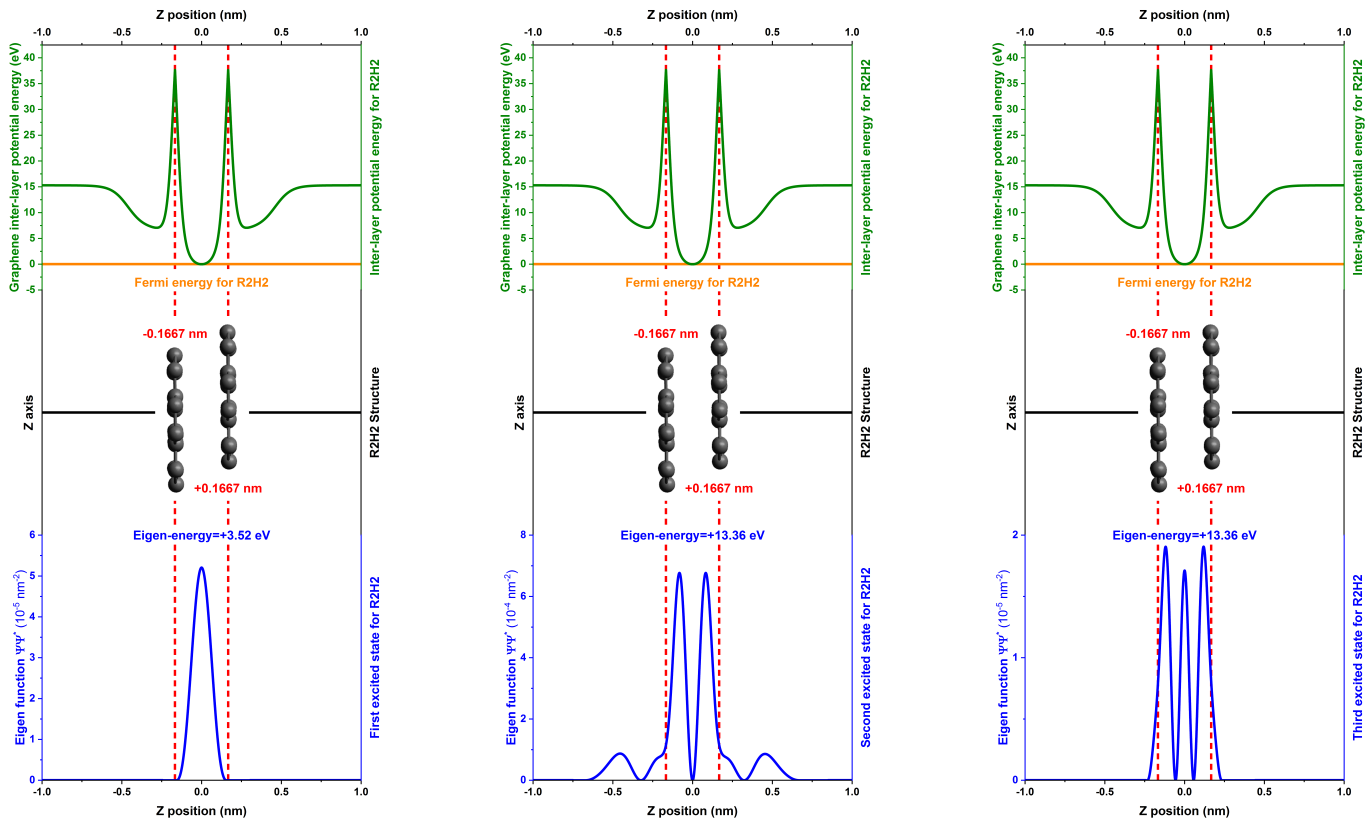


Figura 3.22: Modelo semi-empírico aplicado a la estructura *R2H1*. Transición del ultimo estado π (C) ocupado (HOMO) al primer estado excitado π^* (C) (LUMO).

Cuadro 3.1: Valores obtenidos para estados bases y excitados por DFT y el método semi-empírico propuesto

Resultados DFT				
Etiqueta	Energía de enlace por placa (eV)	Energía base (eV)	Estado excitado primero (eV)	Estado excitado segundo (eV)
SGN R2H1	—	-01.880	+02.860	—
SGN R2H2	+00.630	-01.770	+03.270	—
SGN R2H3	+01.150	-01.690	+03.120	—
SGN R2H4	+01.740	-01.775	+02.990	+03.200

Resultados Modelo semi-empírico					
Etiqueta	Energía de enlace por placa (eV)	Energía base HOMO (eV)	Estado excitado primero LUMO (eV)	Estado excitado segundo (eV)	Estado excitado tercero (eV)
R2H1	—	-01.775	+03.019	—	—
R2H2	+03.200	-01.775	+03.522	+13.361	+27.807
R2H3	+03.200	-01.775	+03.041	+12.637	+14.712
R2H4	+03.200	-01.775	+02.796	+03.627	+12.329

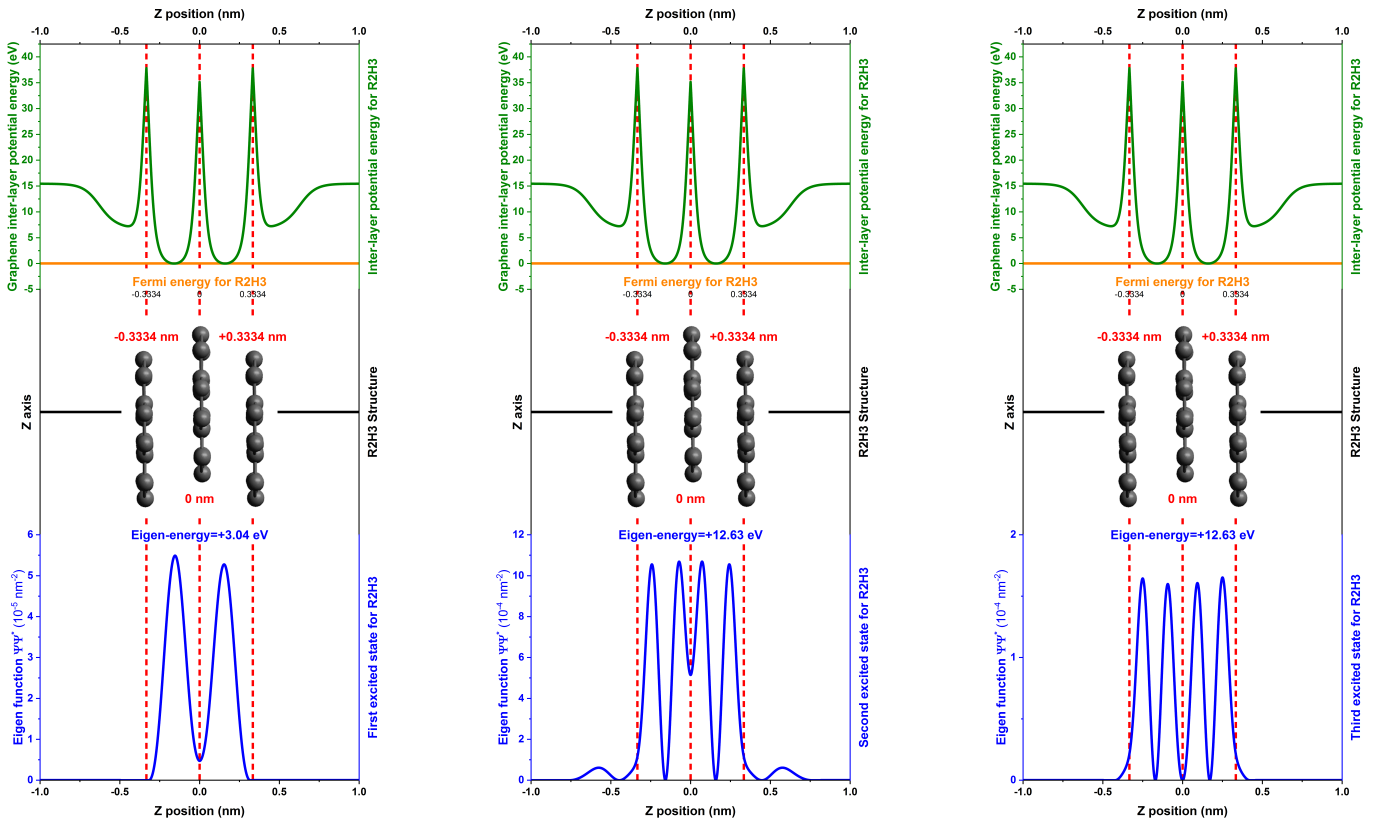


(a) Transición del ultimo estado π (C) ocupado al primer estado excitado π^* (C).

(b) Transición del ultimo estado π (C) ocupado al segundo estado excitado π^* (C).

(c) Transición del ultimo estado π (C) ocupado al tercer estado excitado π^* (C).

Figura 3.23: Resultados del modelo semi-empírico propuesto para la estructura *R2H2*.



(a) Transición del ultimo estado π (C) ocupado al primer estado excitado π^* (C).

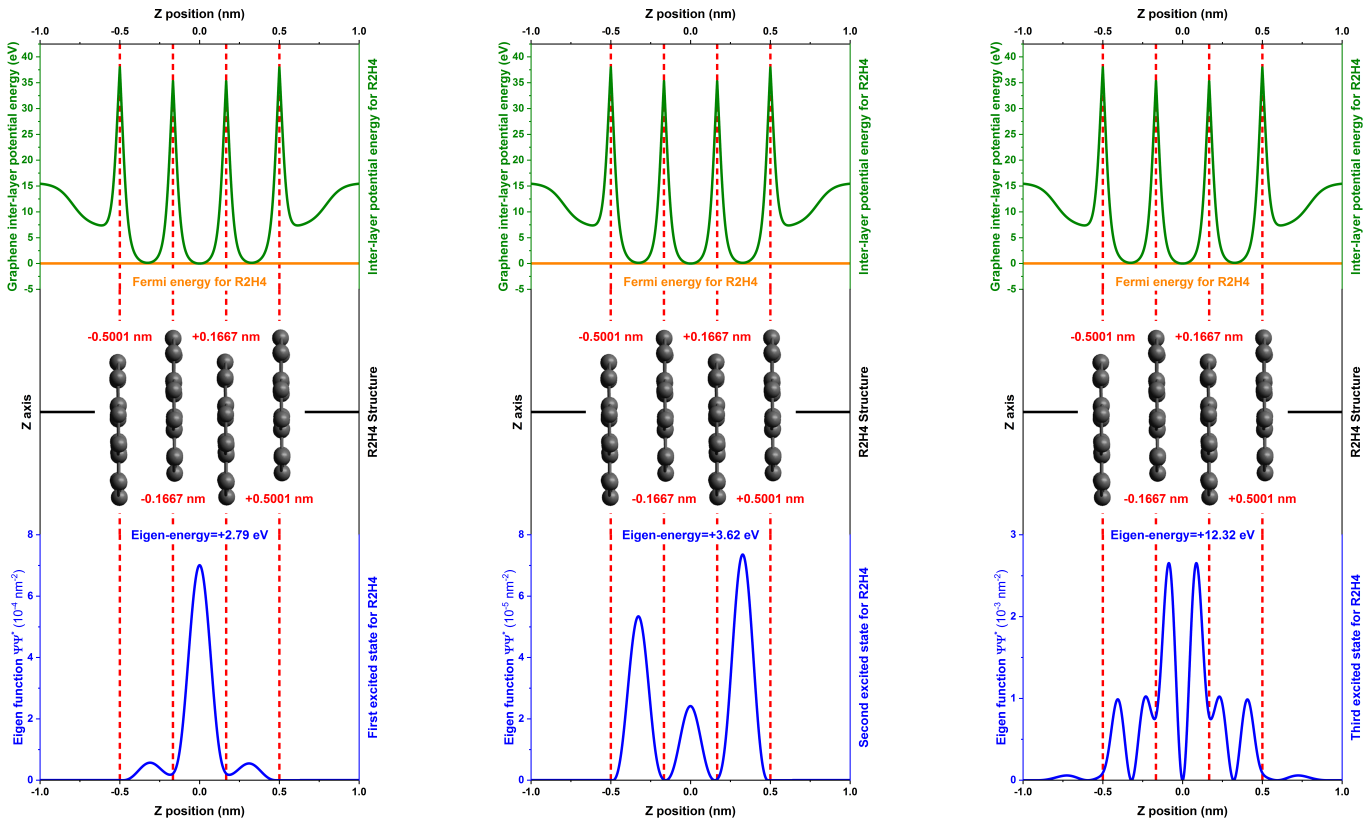
(b) Transición del ultimo estado π (C) ocupado al segundo estado excitado π^* (C).

(c) Transición del ultimo estado π (C) ocupado al tercer estado excitado π^* (C).

Figura 3.24: Resultados modelos semi-empírico para estructura $R2H3$.

Estos potenciales generan un mínimo de energía en medio de ambas placa, lo que obliga al electrón promovido a mantenerse en esta zona. A su vez, el obligar al electrón promovido a mantenerse en los confines de las dos placas de grafeno aumenta el confinamiento del electrón respecto a casa de una placa aislada de grafeno. Por lo tanto, la presencia de una segunda placa aumenta el confinamiento a pesar de incrementar el tamaño de la estructura. Bajo estas condiciones, el desplazamiento al azul observado para las estructuras $SGN R2H2$ y $R2H4$ están de acuerdo con los resultados esperados del QCE. De manera similar, vemos que al añadir otra placa de grafeno a la estructura generando así $R2H3$, se observa en la figura 3.24 que se genera una función de probabilidad similar a la de una una placa de grafeno aislada como el de la figura 3.22. Por lo tanto tenemos un confinamiento similar ambos lados de la placa central (L_2) en la estructura $R2H3$. Como el confinamiento es similar al de una estructura $R2H1$, se esperaria una transición similar a esta estructura por parte de $R2H3$. De esta manera se explica el desplazamiento al rojo observado por la estructura $SGN R2H3$ en el espectro de absorción al aumentar las dimensiones de la estructura.

Para finalizar, se puede observar que la diferencia de energías entre el primer estado excitado y el segundo estado excitado para todas las estructuras obtenidos por medio del modelo semi-empírico poseen una diferencia apreciable, alrededor de $10 eV$, excepto para $R2H4$ donde su diferencia es pequeña. Por lo cual, se plantea la posibilidad que estos dos estados sean el resultado de un desdoblamiento de niveles energéticos ocasionados por la interacciones entre placas. Otros modelos teóricos o semi-empíricos sugieren comportamientos similares para sistemas de una sola placa de grafeno [15, 76]. Por lo cual mas estudios son necesarios para verificar dicho fenómeno de desdoble es responsable por las características de $R2H4$.



(a) Transición del ultimo estado π_1 (C) ocupado al primer estado excitado π^* (C).

(b) Transición del ultimo estado π_1 (C) ocupado al segundo estado excitado π^* (C).

(c) Transición del ultimo estado π_1 (C) ocupado al tercer estado excitado π^* (C).

Figura 3.25: Resultados modelos semi-empírico para estructura *R2H4*.

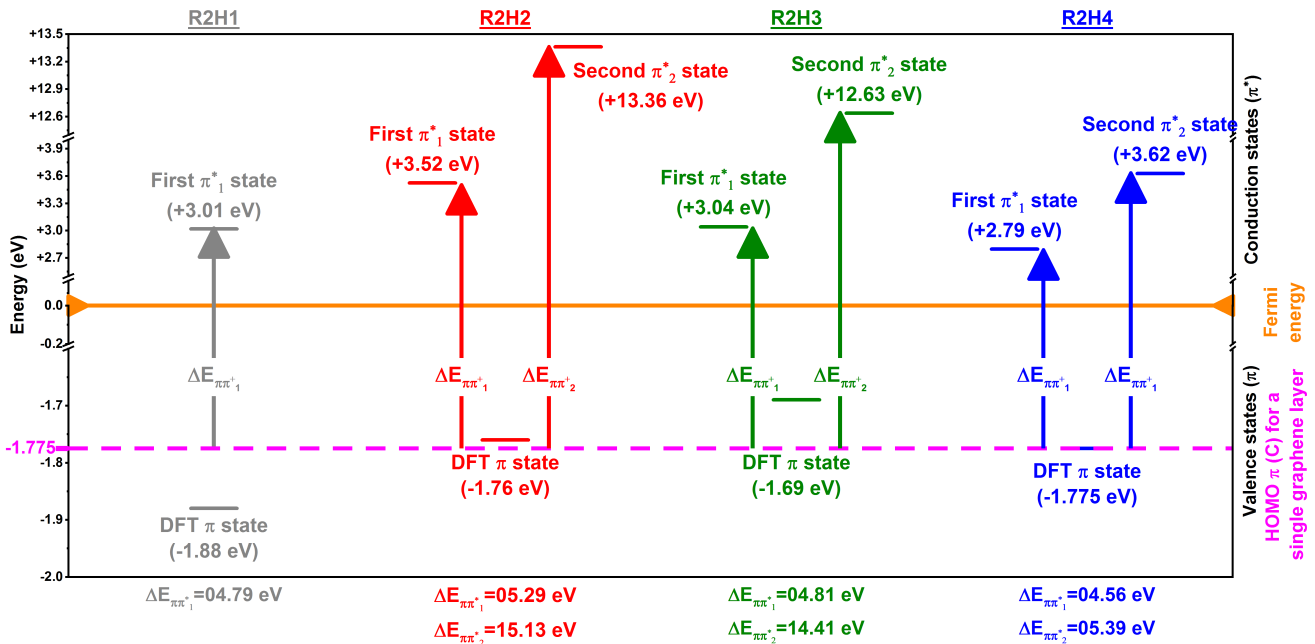


Figura 3.26: Niveles de energía obtenidos con el modelo semi-empírico para las estructuras *R2HY*.

Capítulo 4

Conclusiones y recomendaciones

en este capítulo, se presentan algunas recomendaciones a tomar en cuenta para investigaciones posteriores. También se detallan trabajos a futuros planeados para dar continuidad a los resultados obtenidos en esta investigación. Finalmente, se plantean las conclusiones respectivas a este trabajo.

4.1. Recomendaciones

Algunas recomendaciones, tanto experimentales como teóricas, para trabajos similares al realizado.

- La solución debe de estar contenida en un recipiente el cual soporte tanto la temperatura máxima de pirólisis, así como el incremento de temperatura que se utilizara.
- Una apropiada centrifugación del producto carbogeno final debe ser realizada para evitar sobrantes de la reacción y eliminar productos no deseados como óxido de grafito.
- Para tener un mejor control sobre la temperatura de la solución de CA en pirólisis, se recomienda emplear la técnica de baño de María, utilizando aceite u otro compuesto cuyo punto de ebullición supere los 300 °C.
- Debido a que tanto el tiempo como la temperatura de pirólisis alteran la distribución de tamaños de los GCQDs, el uso de uno o varios electropares para controlar los cambios de temperatura, en uno o varios puntos de la solución, es preferido a usar solamente un termómetro o un indicador digital de equipo de calentamiento.
- Los GCQD poseen una gran absorción en el UV, para obtener medidas con poco ruido debería de utilizarse camino ópticos cortos y fuentes de UV con alta intensidad.
- Para estructuras grandes el corte de energía de las ondas planas puede relajarse a menores energías
- El uso de scripts computacionales, para ejecutar de manera automática la cadena de cálculos computacionales, así como el análisis posterior de los resultados, es recomendable.

4.2. Trabajo a futuro

Para dar continuidad a los resultados obtenidos en este trabajo, se plantean investigaciones posteriores que podrían permitirnos comprender mejor las propiedades de los GCQDs.

- Sintetizar muestras de GCQDs con mayores tamaños (radios y número de placas) y con diferentes dopamientos, Na, Cl, S, etc, para estudiar sus propiedades de absorción y fotoluminiscencia.

- Lograr separar, por medio de centrifugación, filtrado o cromatografía, estructuras con diferentes radios y números de placas para observar como afectan las variaciones de estas dimensiones a las propiedades de absorción y fotoluminiscencia.
- Realizar cálculos de GCQDs utilizando otros funcionales XC que tomen en cuenta interacciones de VdW, como por ejemplo vdW-DF2 y vdW-DF3. El uso de funcionales XC híbridos podría ser otra alternativa viable.
- Simular propiedades de absorción y fotoluminiscencia para estructuras con radios y número de placas mayores a los utilizados en el presente trabajo
- Considerar efectos de borde, dopamiento, grupos funcionales y la naturaleza de 3D del sistema formado por el GCQD en el modelo semi-empírico propuesto en este trabajo.
- Realizar cálculos de GCQDs utilizando solventes, debido a que ciertas propiedades ópticas, como la absorción, PL, entre otras, se ven afectadas por el tipo de solvente y las propiedades de este, como PH, conductividad, temperatura. Por lo cual, realizar simulaciones bajo diferentes solventes podría aportar a la discusión de los datos experimentales obtenidos para estas nanoestructuras.

4.3. Conclusiones

En el presente trabajo se han estudiado, tanto de manera experimental como teórica, las propiedades de absorción de los GCQDs. Se lograron sintetizar muestras de GCQDs, Los tamaños de estas estructuras han sido estimados en $1,0\text{ nm}$ de radio y $1,4\text{ nm}$ de altura (alrededor de 4 placas). Estos valores han sido interpolados con datos medidos por otros autores. Los espectros de absorción de las muestras sintetizadas en este trabajo, poseen espectros características similares a otras estructuras con tamaños comparables. Espectros de absorción para GCQDs, con tamaños similares a los sintetizados, han sido simulados a partir de cálculos de DFT bajo diferentes funcionales de XC. Se han logrado reproducir de manera teórica las tres bandas observadas de absorción observadas experimentalmente. De los tres funcionales utilizados en este trabajo, pz (LDA), PBE (GGA) y vdW-DF (VdW), las simulaciones utilizando el funcional de XC VdW es el que logra reproducir de mejor manera la posición y la intensidad relativa entre los máximos de las diferentes bandas de absorción.

Ademas de simulaciones de espectros de absorción, se realizaron cálculos de DoS, PDoS y respuesta electrónica. Utilizando los resultados de la DoS y PDoS se ha asociado una posible transición electrónica a cada de una de estas bandas. Para las bandas da absorción observadas alrededor de $2,7\text{ eV}$ (459 nm), $3,7\text{ eV}$ (335 nm) y 248 nm , a partir de los resultados de este trabajo a dichas bandas se les podría asociar las transiciones $\sigma - \pi^*$ (NC), $\sigma - \pi^*$ (OC) y $\pi - \pi^*$ (CC), respectivamente. Comparaciones contra otros modelos semi-empíricos o de primero principios, como las reglas de Woodward-Fieser o simulaciones de DFT en PAHs, sugieren que dichas bandas se deben a transiciones $n - \pi^*$ (NC), $n - \pi^*$ (OC) y $\pi - \pi^*$ (CC), respectivamente para las tres bandas observadas experimentalmente. Los cálculos realizados en este trabajo concuerdan por completo con el origen de la banda de absorción mas energética. Para las otras dos bandas observadas, los cálculos presentados concuerda con otros modelos en la los átomos involucrados en la transición, NC y OC, pero no el los orbitales de N y O que participan en esta transición. Más estudios son necesarios para determinar si el origen de esta discrepancia es debido a que el funcional VdW-DF no logra simular por completo las interacciones en la estructura del GCQD o si el origen de dichas transiciones es diferente al aceptado actualmente.

Los cálculos de respuesta electrónica mostraban una densidad electrónica compartida entre las placas de grafeno que constituían la estructura. De manera similar, tanto las simulaciones de espectros de absorción como DoS y PDoS mostraban pequeños desplazamientos en la banda de absorción relacionada

con la transición $\pi\pi^*$ (CC) dependiendo del número de placas de la estructura. Por ejemplo *RXH2* (*RXH1*) y *RXH4* (*RXH3*) poseían desplazamientos similares entre estas estructuras. Debido a que los desplazamientos se encontraban en el orden de los meV , y las interacciones de VdW rondan estos mismos ordenes de magnitud, se planteo la posibilidad que esta banda de absorción era modulada debido a las interacciones de VdW entre placas de grafeno. Para apoyar dicho planteamiento, se propuso un modelo semi-empírico el cual toma a los estados excitados π^* como estados electrónicos originados entre los potenciales de atracción de las diferentes placas de grafeno presentes en la estructura. A pesar de la simplicidad del modelo, el cual no toma en cuenta efectos de borde, dopamiento y solo considera una dimensión ($1D$), se obtienen resultados que están de acuerdo con los resultados obtenidos por medio de cálculos mas elaborados por medio de DFT.

Para finalizar, en este trabajo se plantea una explicación de como el número de placas y las fuerzas de VdW afectan las propiedades de absorción de los GCQDs. Si las propiedades de absorción son alteradas, existe una gran probabilidad de que las propiedades de fotoluminiscencia también se vean afectadas. Actualmente, existe una fuerte discusión sobre las propiedades de fotoluminiscencia de estas nano-estructuras. Sin embargo, hay resultados mixtos entre los resultados experimentales y los modelos teóricos propuestos para los GCQD. Hasta el momento, no tenemos conocimiento de que alguno de estos modelos tome en cuenta la interacción entre placas de los GCQD en sus propiedades de fotoluminiscencia. el tomar en cuenta esta interacción podría ayudar a esclarecer el comportamiento de fotoluminiscencia en los GCQDs.

Referencias bibliográficas

- [1] Jorge Cuadra, Hamilton Ponce y Carlos Rudamas. “Puntos cuánticos de grafeno-carbono: Propiedades ópticas”. En: *I Central American Research Workshop*. 2019.
- [2] Jorge Cuadra, Hamilton Ponce y Carlos Rudamas. “IEEE 2018 38th Central America and Panama Convention (CONCAPAN XXXVIII) - San Salvador, El Salvador - Optical Properties of Graphene Carbon Quantum Dots for Solar Cells Applications”. En: 2018. ISBN: 978-1-5386-6122-2. DOI: 10.1109/CONCAPAN.2018.8596611. URL: <https://ieeexplore.ieee.org/document/8596611>.
- [3] Jorge Cuadra, Hamilton Ponce y Carlos Rudamas. “OPTICAL TRANSITION STUDY ON GRAPHENE CARBON QUANTUM DOTS”. En: *XXVIII International Materials Research Congress, IMRC 2019*. Materials Research Society, 2019.
- [4] Jorge Cuadra, Hamilton Ponce y Carlos Rudamas. “Electronic response function in multi-layered graphene carbon quantum dots”. En: *Semana internacional de la nanotecnología*. 2019.
- [5] Zhipan Zhang y col. “Graphene quantum dots: An emerging material for energy-related applications and beyond”. En: *Energy and Environmental Science* 5.10 (2012), págs. 8869-8890. ISSN: 17545692. DOI: 10.1039/c2ee22982j.
- [6] Shujun Wang y col. “Structural evolution of graphene quantum dots during thermal decomposition of citric acid and the corresponding photoluminescence”. En: *Carbon* 82.C (2015), págs. 304-313. ISSN: 00086223. DOI: 10.1016/j.carbon.2014.10.075.
- [7] Hao Wang y col. “Nitrogen-doped carbon dots for “green” quantum dot solar cells”. En: *Nanoscale Research Letters* 11.1 (2016), págs. 1-18. ISSN: 1556276X. DOI: 10.1186/s11671-016-1231-1.
- [8] P. Tian y col. “Graphene quantum dots from chemistry to applications”. En: *Materials Today Chemistry* 10 (2018), págs. 221-258. ISSN: 24685194. DOI: 10.1016/j.mtchem.2018.09.007. URL: <https://doi.org/10.1016/j.mtchem.2018.09.007>.
- [9] A. N. Emam y col. “Cyto-toxicity, biocompatibility and cellular response of carbon dots-plasmonic based nano-hybrids for bioimaging”. En: *RSC Advances* 7.38 (2017), págs. 23502-23514.
- [10] Mahasin Alam Sk y col. “Revealing the tunable photoluminescence properties of graphene quantum dots”. En: *J. Mater. Chem. C* 2.34 (2014), págs. 6954-6960. ISSN: 2050-7526. DOI: 10.1039/C4TC01191K.
- [11] Sung Kim y col. “Anomalous Behaviors of Visible Luminescence from Graphene Quantum Dots: Interplay between Size and Shape”. En: *ACS Nano* 6 (9 sep. de 2012). DOI: 10.1021/nn302878r.
- [12] Shoujun Zhu y col. “The photoluminescence mechanism in carbon dots (graphene quantum dots, carbon nanodots, and polymer dots): current state and future perspective”. En: *Nano Research* 8.2 (2015), págs. 355-381. ISSN: 19980000. DOI: 10.1007/s12274-014-0644-3.
- [13] Adelio R. Matamala y Alejandro A. Alarcón. “A Simple Model for the Calculation of HOMO and LUMO Energy Levels of Benzocatafusenes”. En: *I. J. Q. Chem.* 106 (2006), págs. 3190-3207. DOI: 10.1002/qua.

- [14] Sharma S R K C Yamijala, Madhuri Mukhopadhyay y Swapan K. Pati. “Linear and nonlinear optical properties of graphene quantum dots: A computational study”. En: *J. Phys. Chem. C* 119.21 (2015), págs. 12079-12087. ISSN: 19327455. DOI: 10.1021/acs.jpcc.5b03531.
- [15] Isil Ozfidan y col. “Theory of optical properties of graphene quantum dots”. En: *Physica Status Solidi - Rapid Research Letters* 10.1 (2016), págs. 102-110. ISSN: 18626270. DOI: 10.1002/pssr.201510335.
- [16] T. Thonhauser y col. “Van der Waals density functional: Self-consistent potential and the nature of the van der Waals bond”. En: *Physical Review B - Condensed Matter and Materials Physics* 76.12 (2007), págs. 1-11. ISSN: 10980121. DOI: 10.1103/PhysRevB.76.125112. arXiv: 0703442 [cond-mat].
- [17] D. C. Langreth y col. “A density functional for sparse matter”. En: *Journal of Physics Condensed Matter* 21.8 (2009). ISSN: 09538984. DOI: 10.1088/0953-8984/21/8/084203.
- [18] Berland Kristian y col. “van der Waals forces in density functional theory: a review of the vdW-DF method”. En: *Reports on Progress in Physics* 78.6 (2015), pág. 66501. ISSN: 00344885. DOI: 10.1088/0034-4885/78/6/066501. arXiv: 1412.6827.
- [19] Paolo Giannozzi y col. “QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials”. En: *Journal of Physics: Condensed Matter* 21.39 (2009), 395502 (19pp).
- [20] P Giannozzi y col. “Advanced capabilities for materials modelling with QUANTUM ESPRESSO”. En: *Journal of Physics: Condensed Matter* 29.46 (2017), pág. 465901.
- [21] John Sinclair. “An Introduction to Quantum Dots: Confinement, Synthesis, Artificial Atoms and Applications”. En: 401.3 (2009), págs. 273-294. ISSN: 15213889. DOI: 10.1002/andp.19314010303. eprint: 0402594v3.
- [22] Vasilios N. Stavrou. *Quantum Dots Theory and Applications*. 2015. Cap. 1, págs. 3-26. ISBN: 9789535121558.
- [23] Runwei Mo y col. “Facile Synthesis of Anatase TiO₂ Quantum-Dot/Graphene-Nanosheet Composites with Enhanced Electrochemical Performance for Lithium-Ion Batteries”. En: *Advanced Materials* 26 (13 abr. de 2014). DOI: 10.1002/adma.201304338.
- [24] Timothy Jamieson y col. “Biological applications of quantum dots”. En: *Biomaterials* 28 (31 2007). DOI: 10.1016/j.biomaterials.2007.07.014.
- [25] A. J. Nozik y col. “Semiconductor Quantum Dots and Quantum Dot Arrays and Applications of Multiple Exciton Generation to Third-Generation Photovoltaic Solar Cells”. En: *Chemical Reviews* 110 (11 2010). DOI: 10.1021/cr900289f.
- [26] S. Coe-Sullivan y col. “Quantum Dots for LED Downconversion in Display Applications”. En: *ECS Journal of Solid State Science and Technology* 2 (2 nov. de 2012). DOI: 10.1149/2.012302jss.
- [27] Vasilios N. Stavrou. *Quantum Dots Theory and Applications*. 2015. Cap. 5, págs. 3-26. ISBN: 9789535121558.
- [28] Octavi E. Semonin, Joseph M. Luther y Matthew C. Beard. “Quantum dots for next-generation photovoltaics”. En: *Materials Today* 15.11 (2012), págs. 508-515. ISSN: 13697021. DOI: 10.1016/S1369-7021(12)70220-1.
- [29] Nan Chen y col. “The cytotoxicity of cadmium-based quantum dots”. En: *Biomaterials* 33 (5 2012).
- [30] Raz Jelinek. *Carbon Quantum Dots Synthesis, Properties and Applications*. 2016, págs. 1-3. ISBN: 978-3-319-43909-9. DOI: 10.1007/978-3-319-43911-2_1.

- [31] Libero J. Bartolotti y Ken Flurchick. “An Introduction to Density Functional Theory”. En: *Reviews in Computational Chemistry* January (2014), págs. 187-216. ISSN: 00029505. DOI: 10.1002/9780470125847.ch4.
- [32] W. Kohn, A. D. Becke y R. G. Parr. “Density Functional Theory of Electronic Structure”. En: *Journal of Physical Chemistry* 100 (31 1996). DOI: 10.1021/jp9606691.
- [33] James F. Lutsko. “[Advances in Chemical Physics] Advances in Chemical Physics Volume 144 — Recent Developments in Classical Density Functional Theory”. En: vol. 10.1002/9780470564318. 2010. ISBN: 9780470564318,9780470547861. DOI: 10.1002/9780470564318.ch1.
- [34] Kieron Burke. “Perspective on density functional theory”. En: *The Journal of Chemical Physics* 136 (15 2012). DOI: 10.1063/1.4704546.
- [35] E. K. U. Gross y Walter Kohn. “Local density-functional theory of frequency-dependent linear response”. En: *Physical Review Letters* 55 (26 dic. de 1985). DOI: 10.1103/PhysRevLett.55.2850.
- [36] M.A.L. Marques y E.K.U. Gross. “Time-DEPENDENT DENSITY FUNCTIONAL THEORY”. En: *Annual Review of Physical Chemistry* 55 (1 2004). DOI: 10.1146/annurev.physchem.55.091602.094449.
- [37] M.E. Casida y M. Huix-Rotllant. “Progress in Time-Dependent Density-Functional Theory”. En: *Annual Review of Physical Chemistry* 63 (1 mayo de 2012). DOI: 10.1146/annurev-physchem-032511-143803.
- [38] Andrea Dal Corso. “A Pseudopotential Plane Waves Program (PWSCF) and some Case Studies”. En: *Lecture Notes in Chemistry* 67 (1996).
- [39] Jesse Noffsinger y col. “EPW: A program for calculating the electron-phonon coupling using maximally localized Wannier functions”. En: *Computer Physics Communications* 181.12 (2010), págs. 2140-2148. ISSN: 00104655. DOI: 10.1016/j.cpc.2010.08.027. eprint: 1005.4418. URL: <http://dx.doi.org/10.1016/j.cpc.2010.08.027>.
- [40] Dario Alfè. “PHON: A program to calculate phonons using the small displacement method”. En: *Computer Physics Communications* 180.12 (2009), págs. 2622-2633. ISSN: 00104655. DOI: 10.1016/j.cpc.2009.03.010. eprint: arXiv:1011.1669v3. URL: <http://dx.doi.org/10.1016/j.cpc.2009.03.010>.
- [41] Osman Bariş Malcioğlu y col. “TurboTDDFT - A code for the simulation of molecular spectra using the Liouville-Lanczos approach to time-dependent density-functional perturbation theory”. En: *Computer Physics Communications* 182.8 (2011), págs. 1744-1754. ISSN: 00104655. DOI: 10.1016/j.cpc.2011.04.020.
- [42] Dario Rocca y col. “Turbo charging time-dependent density-functional theory with Lanczos chains”. En: *Journal of Chemical Physics* 128.15 (2008). ISSN: 00219606. DOI: 10.1063/1.2899649. eprint: 0801.1393.
- [43] Arash A. Mostofi y col. “wannier90: A tool for obtaining maximally-localised Wannier functions”. En: *Computer Physics Communications* 178 (9 2008). DOI: 10.1016/j.cpc.2007.11.016.
- [44] Sandra Alvarenga y col. “Proceedings of the 2nd International Conference of Theoretical and Applied Nanoscience and Nanotechnology (TANN’18) - Niagara Falls, Canada - Changes on the Stokes Shift in Large CdSe Colloidal Quantum Dots by a Ligand Exchange”. En: 140. 2018, págs. 1-5. DOI: 10.11159/tann18.140.
- [45] R. Chang y col. *Química*. McGraw-Hill, 2010, págs. 437-438. ISBN: 978-007-351109-2.

- [46] Newton Ooi, Asit Rairkar y James B. Adams. "Density functional study of graphite bulk and surface properties". En: *Carbon* 44.2 (2006), págs. 231-242. ISSN: 00086223. DOI: 10.1016/j.carbon.2005.07.036.
- [47] Marcus D. Hanwell y col. "Avogadro: an advanced semantic chemical editor, visualization, and analysis platform". En: *Journal of Cheminformatics* 4 (1 dic. de 2012).
- [48] Raz Jelinek. *Carbon Quantum Dots Synthesis, Properties and Applications*. 2016, págs. 5-27. ISBN: 978-3-319-43909-9. DOI: 10.1007/978-3-319-43911-2_2.
- [49] Fengyi Du y col. "Nitrogen-doped carbon dots with heterogeneous multi-layered structures". En: *RSC Advances* 4 (71 ago. de 2014). DOI: 10.1039/c4ra06818a.
- [50] A. K. Geim y K. S. Novoselov. "The rise of graphene". En: *Nature Materials* 6.3 (2007), págs. 183-191. ISSN: 14761122. DOI: 10.1038/nmat1849. URL: <http://www.nature.com/articles/nmat1849>.
- [51] Xiaohui Gao y col. "Carbon quantum dot-based nanoprobe for metal ion detection". En: *Journal of Materials Chemistry C* 4.29 (2016), págs. 6927-6945. ISSN: 20507526. DOI: 10.1039/c6tc02055k.
- [52] Shahriar Shafiee y Erkan Topal. "When will fossil fuel reserves be diminished?" En: *Energy Policy* 37.1 (ene. de 2009), págs. 181-189. ISSN: 03014215.
- [53] M. Asif y T. Muneer. "Energy supply, its demand and security issues for developed and emerging economies". En: *Renewable and Sustainable Energy Reviews* 11.7 (2007), págs. 1388-1413. ISSN: 13640321. DOI: 10.1016/j.rser.2005.12.004.
- [54] Woosung Kwon y col. "Control of Photoluminescence of Carbon Nanodots via Surface Functionalization using Para-substituted Anilines". En: *Scientific Reports* 5. February (2015), págs. 1-10. ISSN: 20452322. DOI: 10.1038/srep12604. URL: <http://dx.doi.org/10.1038/srep12604>.
- [55] Anton Kokalj. "Computer graphics and graphical user interfaces as tools in simulations of matter at the atomic scale". En: *Computational Materials Science* 28 (2 2003). DOI: 10.1016/S0927-0256(03)00104-6.
- [56] "VESTA 3 for three-dimensional visualization of crystal, volumetric and morphology data". En: *Journal of Applied Crystallography* 44 (6 2011). DOI: 10.1107/S0021889811038970.
- [57] Arthur Marronnier. "Anharmonicity and Instabilities in Halide Perovskites for Last Generation Solar Cells". Tesis doct. Jun. de 2018.
- [58] Aron J. Cohen, Paula Mori-Sánchez y Weitao Yang. "Challenges for Density Functional Theory". En: *Chemical Reviews* 112 (1 2012). DOI: 10.1021/cr200107z.
- [59] Paolo Giannozzi. *DENSITY FUNCTIONAL THEORY FOR ELECTRONIC STRUCTURE CALCULATIONS*. Lecture Notes per il Corso di Struttura della Materia. 2002.
- [60] W. Kohn y L. J. Sham. "Self-Consistent Equations Including Exchange and Correlation Effects". En: *Physical Review* 385.1951 (1965). ISSN: 0031-899X. DOI: 10.1103/PhysRev.140.A1133. eprint: [PhysRev.140.A1133](https://arxiv.org/abs/10.1103/PhysRev.140.A1133) (10.1103).
- [61] J. P. Perdew. "Comments on the metal surface from a simple analytic model". En: *Physical Review B* 21.2 (1980), pág. 5.
- [62] Paulo H. Acioli. "Review of quantum Monte Carlo methods and their applications". En: *Journal of Molecular Structure: THEOCHEM* 394.2-3 (1997), págs. 75-85. ISSN: 01661280. DOI: 10.1016/S0166-1280(96)04821-X.
- [63] John P. Perdew y Wang Yue. "Accurate and simple density functional for the electronic exchange energy: Generalized gradient approximation". En: *Physical Review B* 33.12 (1986), págs. 8800-8802. ISSN: 01631829. DOI: 10.1103/PhysRevB.33.8800.

- [64] *Los espectros de absorción simulados y los cálculos de optimización geométrica utilizaron los siguientes PP: C.pbe-n-rrkjus_psl.1.0.0.UPF, H.pbe-rrkjus_psl.1.0.0.UPF, O.pbe-n-rrkjus_psl.1.0.0.UPF and N.pbe-n-rrkjus_psl.1.0.0.UPF. Todos los PP mencionados fueron tomados de la base de datos de PP de QE <http://www.quantum-espresso.org/pseudopotentials>.*
- [65] *Los cálculos de DoS y PDoS utilizaron los siguientes PP: C.pbe-n-kjpaw_psl.1.0.0.UPF, H.pbe-kjpaw_psl.1.0.0.UPF, O.pbe-n-kjpaw_psl.1.0.0.UPF and N.pbe-n-kjpaw_psl.1.0.0.UPF. Todos los PP mencionados fueron tomados de la base de datos de PP de QE <http://www.quantum-espresso.org/pseudopotentials>.*
- [66] Salomon R. Billeter, Alexander J. Turner y Walter Thiel. “Linear scaling geometry optimisation and transition state search in hybrid delocalised internal coordinates”. En: *Physical Chemistry Chemical Physics* 2.10 (2000), págs. 2177-2186. ISSN: 14639076. DOI: 10.1039/a909486e.
- [67] Xiaobin Chen y col. “Interlayer interactions in graphites”. En: *Scientific Reports* 3 (2013), págs. 1-5. ISSN: 20452322. DOI: 10.1038/srep03046.
- [68] D.G. Kanhere. “Bound State of One Dimensional Potential by Numerov Method”. En: *Physics Education* 23.1 (2006), págs. 55-60.
- [69] Mohandas Pillai, Joshua Goglio y Thad G. Walker. “Matrix Numerov method for solving Schrödinger’s equation”. En: *American Journal of Physics* 80.11 (2012), págs. 1017-1019. ISSN: 0002-9505. DOI: 10.1119/1.4748813.
- [70] Jian Ju y col. “Nitrogen-doped graphene quantum dots-based fluorescent probe for the sensitive turn-on detection of glutathione and its cellular imaging”. En: *RSC Advances* 4.94 (2014), págs. 52583-52589. ISSN: 20462069. DOI: 10.1039/c4ra10601f.
- [71] Zhi Yang y col. “Nitrogen-doped, carbon-rich, highly photoluminescent carbon dots from ammonium citrate”. En: *Nanoscale* 6.3 (2014), págs. 1890-1895. ISSN: 20403364. DOI: 10.1039/c3nr05380f.
- [72] Xiaohui Liu y col. “Structure and photoluminescence evolution of nanodots during pyrolysis of citric acid: From molecular nanoclusters to carbogenic nanoparticles”. En: *Journal of Materials Chemistry C* 5.39 (2017), págs. 10302-10312. ISSN: 20507526. DOI: 10.1039/c7tc03429f.
- [73] Pichitchai Pimpang, Rattiphorn Sumang y Supab Choopun. “Effect of concentration of citric acid on size and optical properties of fluorescence graphene quantum dots prepared by tuning carbonization degree”. En: *Chiang Mai Journal of Science* 45.5 (2018), págs. 2005-2014. ISSN: 01252526.
- [74] Peijie Wang y col. “The SERS study of graphene deposited by gold nanoparticles with 785 nm excitation”. En: *Chemical Physics Letters* 556 (2013), págs. 146-150. ISSN: 0009-2614. DOI: 10.1016/j.cplett.2012.11.018.
- [75] Joke De Gelder y col. “Reference database of Raman spectra of biological molecules”. En: *Journal of Raman Spectroscopy* 38.April (2007), págs. 1133-1147. DOI: 10.1002/jrs.
- [76] Paweł Bugajny y col. “Optical properties of geometrically optimized graphene quantum dots”. En: *Phys. E Low-Dimensional Syst. Nanostructures* 85 (2017), págs. 294-301. ISSN: 13869477. DOI: 10.1016/j.physe.2016.08.030.

Anexos

.1. Anexo A: Archivos *input* para cálculos de optimización geométrica (*relaxation calculations*)

Archivo para lanzar optimización geométrica usando *pw.x* en *SGN R2H1 VdW - calculation_rlx_1_pw_rlx.in*

```
&CONTROL
calculation='relax'
title='Optimization calculation'
prefix='CQD_SGN_R2H2_ST1_VDW-DF'
verbosity='high'
pseudo_dir='/home/jcuadra/QE_Pseudo_Potentials/'
outdir='/home/jcuadra/Calculations_Data/CQD_Stacking_Graphite_Nitrogenized/CQD_SGN_R2H2_ST1_VDW-DF/results_rlx'
restart_mode='from_scratch'
wf_collect=.True.
nstep=300
max_seconds=763200
tstress=.True.
tprnfor=.True.
etot_conv_thr=1.0D-3
forc_conv_thr=1.0D-3
disk_io='low'
/

&SYSTEM
ibrav=8
a=13.15549
b=14.57183
c=7.348
nat=60
ntyp=4
ecutwfc=30
ecutrho=120
ecutfock=120
nosym=.True.
occupations='smearing'
degauss=1.0D-3
smearing='gaussian'
input_dft='vdw-df'
/

&ELECTRONS
electron_maxstep=300
conv_thr=1.0D-9
mixing_mode='plain'
mixing_beta=6.5D-1
mixing_ndim=10
diagonalization='david'
diago_david_ndim=6
startingpot='atomic'
startingwfc='atomic'
/

&IONS
ion_dynamics='bfgs'
ion_positions='default'
pot_extrapolation='atomic'
wfc_extrapolation='atomic'
bfgs_ndim=1
upscale=100
trust_radius_max=7.0D-1
trust_radius_min=1.0D-2
trust_radius_ini=1.0D-1
/

K_POINTS {automatic}
5 5 4 0 0 0

ATOMIC_SPECIES
H 01.00794 H.pbe-rrkjus_psl.0.1.UPF
C 12.01070 C.pbe-n-rrkjus_psl.0.1.UPF
N 14.00670 N.pbe-n-rrkjus_psl.0.1.UPF
```

O 15.99940 0.pbe-n-rrkjus_psl.0.1.UPF

```
ATOMIC_POSITIONS {angstrom}
N 5.34974 4.44998 2.00000
C 4.12175 5.15896 2.00000
C 4.12174 6.57694 2.00000
O 2.89375 7.28592 2.00000
C 2.89374 8.70390 2.00000
C 7.80574 4.44998 2.00000
C 6.57775 5.15896 2.00000
C 6.57774 6.57694 2.00000
C 7.80575 7.28592 2.00000
C 7.80574 8.70390 2.00000
C 6.57775 9.41288 2.00000
C 6.57774 10.83085 2.00000
C 5.34975 7.28592 2.00000
C 5.34974 8.70390 2.00000
C 4.12175 9.41288 2.00000
O 4.12174 10.83085 2.00000
C 5.34975 11.53983 2.00000
C 10.26175 7.28592 2.00000
O 10.26174 8.70390 2.00000
O 9.03375 5.15896 2.00000
C 9.03374 6.57694 2.00000
C 9.03375 9.41288 2.00000
C 9.03374 10.83085 2.00000
N 7.80575 11.53983 2.00000
H 3.22801 4.64296 2.00000
H 2.00000 9.21990 2.00000
H 7.80574 3.41798 2.00000
H 5.34975 12.57183 2.00000
H 11.15549 6.76992 2.00000
H 9.92748 11.34685 2.00000
N 5.34974 3.03200 5.34800
C 4.12175 3.74098 5.34800
C 4.12174 5.15896 5.34800
O 2.89375 5.86794 5.34800
C 2.89374 7.28592 5.34800
C 7.80574 3.03200 5.34800
C 6.57775 3.74098 5.34800
C 6.57774 5.15896 5.34800
C 7.80575 5.86794 5.34800
C 7.80574 7.28592 5.34800
C 6.57775 7.99490 5.34800
C 6.57774 9.41287 5.34800
C 5.34975 5.86794 5.34800
C 5.34974 7.28592 5.34800
C 4.12175 7.99490 5.34800
O 4.12174 9.41287 5.34800
C 5.34975 10.12185 5.34800
C 10.26175 5.86794 5.34800
O 10.26174 7.28592 5.34800
O 9.03375 3.74098 5.34800
C 9.03374 5.15896 5.34800
C 9.03375 7.99490 5.34800
C 9.03374 9.41287 5.34800
N 7.80575 10.12185 5.34800
H 3.22801 3.22498 5.34800
H 2.00000 7.80192 5.34800
H 7.80574 2.00000 5.34800
H 5.34975 11.15385 5.34800
H 11.15549 5.35194 5.34800
H 9.92748 9.92887 5.34800
```

.2. Anexo B: Archivos *input* para cálculos de DoS y PDoS

Archivo para lanzar SCF usando *pw.x* en
SGN R2H1 VdW - calculation_del_1_pw_scf.in

```
&CONTROL
calculation='scf'
title='Self consistent calculation'
prefix='CQD_SGN_R2H2_ST1_VDW-DF'
verbosity='high'
pseudo_dir='/home/jcuadra/QE_Pseudo_Potentials/'
outdir='/home/jcuadra/Calculations_Data/CQD_Stacking_Graphite_Nitrogenized/CQD_SGN_R2H2_ST1_VDW-DF/results_del/'
restart_mode='from_scratch'
wf_collect=.True.
nstep=1
max_seconds=763200
tstress=.False.
tprnfor=.False.
etot_conv_thr=1.0D-3
forc_conv_thr=1.0D-3
disk_io='low'
/

&SYSTEM
ibrav=8
```

```

a=13.15549
b=14.57183
c=7.348
nat=60
ntyp=4
ecutwfc=30
ecutrho=120
ecutfock=120
nosym=.True.
occupations='tetrahedra_opt' !'fixed' 'smearing'
degauss=1.0D-3
smearing='gaussian'
input_dft='vdw-df'
/

&ELECTRONS
electron_maxstep=300
conv_thr=1.0D-10
mixing_mode='plain'
mixing_beta=6.5D-1
mixing_ndim=10
diagonalization='david'
diago_david_ndim=6
startingpot='atomic'
startingwfc='atomic'
/

K_POINTS {automatic}
5 5 5 0 0 0

ATOMIC_SPECIES
H 01.00794 H.pbe-rrkjus_psl.1.0.0.UPF
C 12.01070 C.pbe-n-rrkjus_psl.1.0.0.UPF
N 14.00670 N.pbe-n-rrkjus_psl.1.0.0.UPF
O 15.99940 O.pbe-n-rrkjus_psl.1.0.0.UPF

ATOMIC_POSITIONS {angstrom}
N 5.306702879 4.411472309 1.946869334
C 4.157494607 5.122344275 1.973971176
C 4.099477012 6.545387078 1.967481047
O 2.874536425 7.240894441 1.975380549
C 2.899385686 8.682564084 2.001495963
C 7.761563723 4.440705582 1.947259144
C 6.512694325 5.083975480 1.932627409
C 6.528744061 6.538857944 1.933275246
C 7.737567599 7.245494686 1.931073601
C 7.758520779 8.661197420 1.935723964
C 6.552122096 9.393943968 1.943011506
C 6.612845502 10.842561259 1.993242466
C 5.309194557 7.271017511 1.946484675
C 5.337488389 8.695543860 1.954886880
C 4.086510496 9.394937370 1.985408471
O 4.156651884 10.819583546 1.986860475
C 5.377764471 11.498982987 2.039192747
C 10.175985190 7.265520638 1.964869530
O 10.162526196 8.667224900 1.973546138
O 8.986084951 5.175583073 1.973241426
C 8.995362957 6.558964335 1.951150784
C 8.968555240 9.374532671 1.968631853
C 8.958267394 10.794837062 1.991052349
N 7.821011394 11.510829467 2.013935136
H 3.207235548 4.559603771 2.004805706
H 1.880070044 9.091514582 2.003103393
H 7.955101644 3.358559857 1.963253526
H 5.213963872 12.585558974 2.034087887
H 11.197816632 6.860592504 1.993564057
H 9.928222511 11.323488956 2.003050593
N 5.334436566 3.060990042 5.334053836
C 4.197258555 3.777009391 5.356949539
C 4.186816677 5.197280031 5.379361763
O 2.993658494 5.905105862 5.374296578
C 2.979441173 7.306267759 5.383119657
C 7.777750582 3.072824689 5.308789239
C 6.542643561 3.729294466 5.354737145
C 6.603393318 5.177888552 5.404975781
C 7.817958965 5.876255524 5.393096093
C 7.846302320 7.300826073 5.401503674
C 6.626703746 8.032959487 5.414721782
C 6.642773901 9.487868778 5.415376099
C 5.396920843 5.910659807 5.412278468
C 5.417936657 7.326364408 5.416925799
C 4.160129663 8.012795431 5.396851847
O 4.169536054 9.396359742 5.375110408
C 5.393903407 10.131094300 5.400743043
C 10.256066432 5.889300506 5.346502814
O 10.281007518 7.330356886 5.372631970
O 8.998415593 3.752332084 5.361050319
C 9.069030955 5.176881310 5.362576101
C 9.055948671 8.026477405 5.380513553
C 8.997944161 9.449503778 5.374029463
N 7.848784524 10.160365077 5.401132144
H 3.227250987 3.248332281 5.344946789
H 1.957686697 7.711258315 5.354450562
H 7.941512444 1.986290690 5.313897106
H 5.200376719 11.213268173 5.384742283
H 11.275400672 5.480344847 5.344898517
H 9.948242083 10.012229715 5.343200596

```

Archivo para lanzar NSCF usando *pw.x* en *SGN R2H1 VdW* - calculation_del_2_pw_nscf.in

```
&CONTROL
calculation='nscf'
title='Non self consistent calculation'
prefix='CQD_SGN_R2H2_ST1_VDW-DF'
verbosity='high'
pseudo_dir='/home/jcuadra/QE_Pseudo_Potentials/'
outdir='/home/jcuadra/Calculations_Data/CQD_Stacking_Graphite_Nitrogenized/CQD_SGN_R2H2_ST1_VDW-DF/results_del/'
restart_mode='from_scratch'
wf_collect=.True.
nstep=1
max_seconds=763200
tstress=.False.
tprnfor=.False.
etot_conv_thr=1.0D-3
forc_conv_thr=1.0D-3
disk_io='low'
/

&SYSTEM
ibrav=8
a=13.15549
b=14.57183
c=7.348
nat=60
ntyp=4
ecutwfc=30
ecutrho=120
ecutfock=120
nosym=.True.
occupations='tetrahedra_opt' !'fixed' 'smearing'
degauss=1.0D-3
smearing='gaussian'
input_dft='vdw-df'
/

&ELECTRONS
electron_maxstep=300
conv_thr=1.0D-10
mixing_mode='plain'
mixing_beta=6.5D-1
mixing_ndim=10
diagonalization='david'
diago_david_ndim=6
startingpot='atomic'
startingwfc='atomic'
/

K_POINTS {automatic}
10 10 10 0 0 0

ATOMIC_SPECIES
H 01.00794 H.pbe-rrkjus_psl.1.0.0.UPF
C 12.01070 C.pbe-n-rrkjus_psl.1.0.0.UPF
N 14.00670 N.pbe-n-rrkjus_psl.1.0.0.UPF
O 15.99940 O.pbe-n-rrkjus_psl.1.0.0.UPF

ATOMIC_POSITIONS {angstrom}
N 5.306702879 4.411472309 1.946869334
C 4.157494607 5.122344275 1.973971176
C 4.099477012 6.545387078 1.967481047
O 2.874536425 7.240894441 1.975380549
C 2.899385686 8.682564084 2.001495963
C 7.761563723 4.440705582 1.947259144
C 6.512694325 5.083975480 1.932627409
C 6.528744061 6.538857944 1.933275246
C 7.737567599 7.245494686 1.931073601
C 7.758520779 8.661197420 1.935723964
C 6.552122096 9.393943968 1.943011506
C 6.612845502 10.842561259 1.993242466
C 5.309194557 7.271017511 1.946484675
C 5.337488389 8.695543860 1.954886880
C 4.086510496 9.394937370 1.985408471
O 4.156651884 10.819583546 1.986860475
C 5.377764471 11.498982987 2.039192747
C 10.175985190 7.265520638 1.964869530
O 10.162526196 8.667224900 1.973546138
O 8.986084951 5.175583073 1.973241426
C 8.995362957 6.558964335 1.951150784
C 8.968555240 9.374532671 1.968631853
C 8.958267394 10.794837062 1.991052349
N 7.821011394 11.510829467 2.013935136
H 3.207235548 4.559603771 2.004805706
H 1.880070044 9.091514582 2.003103393
H 7.955101644 3.358559857 1.963253526
H 5.213963872 12.585558974 2.034087887
H 11.197816632 6.860592504 1.993564057
H 9.928222511 11.323488956 2.003050593
N 5.334436566 3.060990042 5.334053836
C 4.197258555 3.777009391 5.356949539
C 4.186816677 5.197280031 5.379361763
O 2.993658494 5.905105862 5.374296578
C 2.979441173 7.306267759 5.383119657
C 7.777750582 3.072824689 5.308789239
```


C	6.542643561	3.729294466	5.354737145
C	6.603393318	5.177888552	5.404975781
C	7.817958965	5.876255524	5.393096093
C	7.846302320	7.300826073	5.401503674
C	6.626703746	8.032959487	5.414721782
C	6.642773901	9.487868778	5.415376099
C	5.396920843	5.910659807	5.412278468
C	5.417936657	7.326364408	5.416925799
C	4.160129663	8.012795431	5.396851847
O	4.169536054	9.396359742	5.375110408
C	5.393903407	10.131094300	5.400743043
C	10.256066432	5.889300506	5.346502814
O	10.281007518	7.330356886	5.372631970
O	8.998415593	3.752332084	5.361050319
C	9.069030955	5.176881310	5.362576101
C	9.055948671	8.026477405	5.380513553
C	8.997944161	9.449503778	5.374029463
N	7.848784524	10.160365077	5.401132144
H	3.227250987	3.248332281	5.344946789
H	1.957686697	7.711258315	5.354450562
H	7.941512444	1.986290690	5.313897106
H	5.200376719	11.213268173	5.384742283
H	11.275400672	5.480344847	5.344898517
H	9.948242083	10.012229715	5.343200596

Archivo para lanzar cálculo de DoS y PDoS usando *projwfc.x* en *SGN R2H1 VdW* - calculation_del_3_projwfc_plt.in

```
&PROJWFC
prefix='CQD_SGN_R2H2_ST1_VDW-DF'
outdir='/home/jcuadra/Calculations_Data/CQD_Stacking_Graphite_Nitrogenized/CQD_SGN_R2H2_ST1_VDW-DF/results_del'
ngauss=
degauss=
Emin=
Emax=
DeltaE=
/
```

.3. Anexo C: Archivos *input* para simulaciones de espectros de absorción cálculos de respuesta electrónica

Archivo para lanzar SCF usando *pw.x* en *SGN R2H1 VdW* - calculation_sab_1_pw_scf.in

```
&CONTROL
calculation='scf'
title='Self consistent calculation'
prefix='CQD_SGN_R2H2_ST1_VDW-DF'
verbosity='high'
pseudo_dir='/home/jcuadra/QE_Pseudo_Potentials/'
outdir='/home/jcuadra/Calculations_Data/CQD_Stacking_Graphite_Nitrogenized/CQD_SGN_R2H2_ST1_VDW-DF/results_sab/'
restart_mode='from_scratch'
wf_collect=.True.
nstep=1
max_seconds=763200
tstress=.False.
tprnfor=.False.
etot_conv_thr=1.0D-2
forc_conv_thr=1.0D-2
disk_io='low'
/

&SYSTEM
ibrav=8
a=13.15549
b=14.57183
c=7.348
nat=60
ntyp=4
ecutwfc=30
ecutrho=120
ecutfock=120
nosym=.True.
occupations='fixed' !'tetrahedra_opt' !'smearing'
degauss=1.0D-3
smearing='gaussian'
input_dft='vdw-df'
/
```

```

&ELECTRONS
electron_maxstep=300
conv_thr=1.0D-9
mixing_mode='plain'
mixing_beta=6.5D-1
mixing_ndim=10
diagonalization='david' !'cg'
diago_david_ndim=6
startingpot='atomic'
startingwfc='atomic'
/

K_POINTS {gamma}

ATOMIC_SPECIES
H 01.00794 H.pbe-rrkjus_psl.1.0.0.UPF
C 12.01070 C.pbe-n-rrkjus_psl.1.0.0.UPF
N 14.00670 N.pbe-n-rrkjus_psl.1.0.0.UPF
O 15.99940 O.pbe-n-rrkjus_psl.1.0.0.UPF

ATOMIC_POSITIONS {angstrom}
N 5.306702879 4.411472309 1.946869334
C 4.157494607 5.122344275 1.973971176
C 4.099477012 6.545387078 1.967481047
O 2.874536425 7.240894441 1.975380549
C 2.899385686 8.682564084 2.001495963
C 7.761563723 4.440705582 1.947259144
C 6.512694325 5.083975480 1.932627409
C 6.528744061 6.538857944 1.933275246
C 7.737567599 7.245494686 1.931073601
C 7.758520779 8.661197420 1.935723964
C 6.552122096 9.393943968 1.943011506
C 6.612845502 10.842561259 1.993242466
C 5.309194557 7.271017511 1.946484675
C 5.337488389 8.695543860 1.954886880
C 4.086510496 9.394937370 1.985408471
O 4.156651884 10.819583546 1.986860475
C 5.377764471 11.498982987 2.039192747
C 10.175985190 7.265520638 1.964869530
O 10.162526196 8.667224900 1.973546138
O 8.986084951 5.175583073 1.973241426
C 8.995362957 6.558964335 1.951150784
C 8.968555240 9.374532671 1.968631853
C 8.958267394 10.794837062 1.991052349
N 7.821011394 11.510829467 2.013935136
H 3.207235548 4.559603771 2.004805706
H 1.880070044 9.091514582 2.003103393
H 7.955101644 3.358559857 1.963253526
H 5.213963872 12.585558974 2.034087887
H 11.197816632 6.860592504 1.993564057
H 9.928222511 11.323488956 2.003050593
N 5.334436566 3.060990042 5.334053836
C 4.197258555 3.777009391 5.356949539
C 4.186816677 5.197280031 5.379361763
O 2.993658494 5.905105862 5.374296578
C 2.979441173 7.306267759 5.383119657
C 7.777750582 3.072824689 5.308789239
C 6.542643561 3.729294466 5.354737145
C 6.603393318 5.177888552 5.404975781
C 7.817958965 5.876255524 5.393096093
C 7.846302320 7.300826073 5.401503674
C 6.626703746 8.032959487 5.414721782
C 6.642773901 9.487868778 5.415376099
C 5.396920843 5.910659807 5.412278468
C 5.417936657 7.326364408 5.416925799
C 4.160129663 8.012795431 5.396851847
O 4.169536054 9.396359742 5.375110408
C 5.393903407 10.131094300 5.400743043
C 10.256066432 5.889300506 5.346502814
O 10.281007518 7.330356886 5.372631970
O 8.998415593 3.752332084 5.361050319
C 9.069030955 5.176881310 5.362576101
C 9.055948671 8.026477405 5.380513553
C 8.997944161 9.449503778 5.374029463
N 7.848784524 10.160365077 5.401132144
H 3.227250987 3.248332281 5.344946789
H 1.957686697 7.711258315 5.354450562
H 7.941512444 1.986290690 5.313897106
H 5.200376719 11.213268173 5.384742283
H 11.275400672 5.480344847 5.344898517
H 9.948242083 10.012229715 5.343200596

```

Archivo para lanzar calcular coeficientes de *lanz*os usando *turbo_lanzos.x*
 en
SGN R2H1 VdW - calculation_sab_2_turbo_lanzos_coe.in

```

&l_r_input
prefix='CQD_SGN_R2H2_ST1_VDW-DF'
outdir='/home/jcuadra/Calculations_Data/CQD_Stacking_Graphite_Nitrogenized/CQD_SGN_R2H2_ST1_VDW-DF/results_sab/'
restart=False
restart_step=1000

```

```
lr_verbosity=1
/

&lr_control
itermax=4000
ipol=4
/

&lr_post
plot_type=2
/
```

Archivo para lanzar cálculo del espectro de absorción usando *turbo_spectrum.x*
en
SGN R2H1 VdW - calculation_sab_3_turbo_spectrum_sab.in

```
&lr_input
prefix='CQD_SGN_R2H2_ST1_VDW-DF'
outdir='/home/jcuadra/Calculations_Data/CQD_Stacking_Graphite_Nitrogenized/CQD_SGN_R2H2_ST1_VDW-DF/results_sab/'
itermax0=4000
itermax=
extrapolation=
epsil=
units=1
start=1.0
end=13.0
increment=1.D-3
ipol=4
td='lanczos'
/
```

Archivo para lanzar cálculo de la respuesta electrónica usando *turbo_lanczos.x*
en
SGN R2H1 VdW - calculation_sab_4_turbo_lanczos_res.in

```
&lr_input
prefix='CQD_SGN_R2H2_ST1_VDW-DF'
outdir='/home/jcuadra/Calculations_Data/CQD_Stacking_Graphite_Nitrogenized/CQD_SGN_R2H2_ST1_VDW-DF/results_sab/'
restart=False.
restart_step=1000
lr_verbosity=1
/

&lr_control
itermax=4000
ipol=4
charge_response=1
/

&lr_post
omeg=
epsil=
w_T_npol=3
plot_type=3
/
```

.4. Anexo D: Archivos *input* para lanzar cálculos de optimización geométrica, espectros de absorción, DoS, PDoS y respuesta electrónica en Torque

Archivo para lanzar cálculo de optimización geométrica en
SGN R2H1 VdW - CQD_SGN_R2H1_ST1_VDW-DF_RLX.pbs

```
#!/bin/bash
#PBS -N RLX_CQD_SGN_R2H1_ST1_VDW-DF
```

```

#PBS -M jacuadraa@yaxchilan.larcad.mx
#PBS -l nodes=2:ppn=8,walltime=212:00:00
#PBS -j eo
#PBS -k eo
#PBS -m abe
#PBS -q mctp

#####
##Variables#####
#####
NumPro=16 ##NumPro=Number Processors
StrLab=CQD_SGN_R2H1_ST1_VDW-DF ##StrLab=Structure Label
StrTyp=CQD_Stacking_Graphite_Nitrogenized ##StrTyp=Structure Type
DirInp=/home/jcuadra/Calculations_Inputs/$StrTyp/$StrLab ##DirInp=Directory Inputs
DirDup=/home/jcuadra/Calculations_Outputs/$StrTyp/$StrLab ##DirDup=Directory Outputs
DirDat=/home/jcuadra/Calculations_Data/$StrTyp/$StrLab/results_rlx ##DirDat=Directory Data
CalRel=calculation_rlx_1_pw_rlx ##CalRel=Calculation Relaxation
CalRelEndCon="End of BFGS Geometry Optimization" ##CalRelEndCon=Calculation Relaxation End Condition
HisFilLab=$CalRel".his" ##HisFilLab=History File Label

#####
##Calculations#####
#####
module load CondensedmSW/QEspresso-6.3
if [ ! -f $DirDup/$HisFilLab ]; then
  echo "*****" > $DirDup/$HisFilLab
  echo "Relaxation calculations for $StrLab started at $(date)" > $DirDup/$HisFilLab
  echo "*****" > $DirDup/$HisFilLab
else
  echo "*****" > $DirDup/$HisFilLab
  echo "Relaxation calculations for $StrLab started at $(date)" > $DirDup/$HisFilLab
  echo "*****\n" > $DirDup/$HisFilLab
fi

##Perform calculation < relaxation >*****
echo "Initiating relaxation calculation for $StrLab ..." >> $DirDup/$HisFilLab
if [ ! -f $CalRel".out" ]; then
  echo "Relaxation calculation for $StrLab started at $(date)" >> $DirDup/$HisFilLab
  mpirun --map-by core:pe=1 -np $NumPro -machinefile $PBS_NODEFILE pw.x < $DirInp/$CalRel".in" > $DirDup/$CalRel".out"
  CalRelEnd=$( grep -F -o "$CalRelEndCon" $DirDup/$CalRel".out" ) ##CalRelEnd=Calculation Relaxation Ended
  if [ "$CalRelEnd" != "$CalRelEndCon" ]; then
    sed -i 's/from_scratch/restart/' $DirInp/$CalRel".in"
    echo "Relaxation incomplete calculation for $StrLab finished at $(date) \n" >> $DirDup/$HisFilLab
  else
    echo "Relaxation calculation for $StrLab finished at $(date) \n" >> $DirDup/$HisFilLab
  fi
else
  CalRelEnd=$( grep -F -o "$CalRelEndCon" $DirDup/$CalRel".out" ) ##CalRelEnd=Calculation Relaxation Ended
  if [ "$CalRelEnd" != "$CalRelEndCon" ]; then
    echo "Relaxation calculation for $StrLab started again at $(date)" >> $DirDup/$HisFilLab
    mpirun --map-by core:pe=1 -np $NumPro -machinefile $PBS_NODEFILE pw.x < $DirInp/$CalRel".in" > $DirDup/$CalRel".out"
    CalRelEnd=$( grep -F -o "$CalRelEndCon" $DirDup/$CalRel".out" ) ##CalRelEnd=Calculation Relaxation Ended
    if [ "$CalRelEnd" != "$CalRelEndCon" ]; then
      echo "Relaxation incomplete calculation for $StrLab finished at $(date) \n" >> $DirDup/$HisFilLab
    else
      echo "Relaxation calculation for $StrLab finished at $(date) \n" >> $DirDup/$HisFilLab
    fi
  else
    echo "Relaxation calculation for $StrLab was already completed!!! \n" >> $DirDup/$HisFilLab
  fi
fi

```

Archivo para lanzar cálculo de DoS y PDoS en SGN R2H1 VdW - CQD_SGN_R2H1_ST1_VDW-DF_DEL.pbs

```

#!/bin/bash
#PBS -N DEL_CQD_SGN_R2H1_ST1_VDW-DF
#PBS -M jacuadraa@yaxchilan.larcad.mx
#PBS -l nodes=2:ppn=8,walltime=212:00:00
#PBS -j eo
#PBS -k eo
#PBS -m abe
#PBS -q mctp

#####
##Variables#####
#####
NumPro=16 ##NumPro=Number Processors
NumProSafe=8 ##NumPro=Number Processors
StrLab=CQD_SGN_R2H1_ST1_VDW-DF ##StrLab=Structure Label
StrTyp=CQD_Stacking_Graphite_Nitrogenized ##StrTyp=Structure Type
DirHom=/home/jcuadra ##DirHom=Directory Home
DirInp=/home/jcuadra/Calculations_Inputs/$StrTyp/$StrLab ##DirInp=Directory Inputs
DirDup=/home/jcuadra/Calculations_Outputs/$StrTyp/$StrLab ##DirDup=Directory Outputs
DirDat=/home/jcuadra/Calculations_Data/$StrTyp/$StrLab/results_del ##DirDat=Directory Data
CalSCF=calculation_del_1_pw_scf ##CalSCF=Calculation Self Consistent Field
CalNCSF=calculation_del_2_pw_nscf ##CalNCSF=Calculation Lanczos
CalDEL=calculation_del_3_projwfc_plt ##CalDEL=Calculation Density of States Electronic
HisFilLab=$StrLab"_DEL".his" ##HisFilLab=History File Label

DEIFol="_DEL" ##DEIFol=Density of States Electronic Folder
DEIFer=-1.4297 ##DEIFer=Density of States Electronic Fermi Energy (eV)
DEIInt=5 ##DEIInt=Density of States Electronic Interval (eV) (DEIFer-DEIInt,DEIFer+DEIInt)

```

```

DElPoi=10000 ##DElPoi=Density of States Electronic Points
DElSeaPro=".pdos_atm#" ##DElSeaPro=Density of States Electronic Search Projected
DElSeaTot=".pdos_tot" ##DElSeaTot=Density of States Electronic Search Total
CalSCFEndCon="End of self-consistent calculation" ##CalSCFEndCon=Calculation Self Consistent End Condition
CalNSCFEndCon="JOB DONE" ##CalNSCFEndCon=Calculation Non-Self Consistent End Condition
CalDElEndCon="JOB DONE" ##CalDElEndCon=Calculation Density of States Electronic End Condition
ComNga="ngauss=" ##ComNga=Command ngauss
ComDga="degauss=" ##ComDga=Command degauss
ComEmi="Emin=" ##ComEmi=Command Emin
ComEma="Emax=" ##ComEma=Command Emax
ComDel="DeltaE=" ##ComDel=Command DeltaE
VarNga="0 1 -1 -99" ##VarNga=Variable ngauss
VarDga="0.00000000 0.008819834 0.018374654 0.036749309 0.073498618" ##VarDga=Variable degauss

#####
##Calculations#####
#####
module load CondensedSW/QEspresso-6.3
if [ ! -f $DirOup/$HisFillab ]; then
  echo "#####" > $DirOup/$HisFillab
  echo "Electronic DoS calculations for $StrLab started at $(date)" > $DirOup/$HisFillab
  echo "#####" > $DirOup/$HisFillab
else
  echo "#####" > $DirOup/$HisFillab
  echo "Electronic DoS calculations for $StrLab started at $(date)" > $DirOup/$HisFillab
  echo "#####" > $DirOup/$HisFillab
fi

##Perform calculation < self consistent filed scf >#####
echo "Initiating SCF calculation for $StrLab ..." > $DirOup/$HisFillab
if [ ! -f $DirOup/$CalSCF.out ]; then
  echo "SCF calculation for $StrLab started at $(date)" > $DirOup/$HisFillab
  mpirun --map-by core:pe=1 -np $NumPro -machinefile $PBS_NODEFILE pw.x < $DirInp/$CalSCF.in > $DirOup/$CalSCF.out
  CalSCFEnd=$( grep -F -o "$CalSCFEndCon" $DirOup/$CalSCF.out ) ##CalSCFEnd=Calculation SFC Ended
  if [ "$CalSCFEnd" = "$CalSCFEndCon" ]; then
    ##sed -i 's/from_scratch/restart/' $DirInp/$CalSCF.in
    echo "SCF incomplete calculation for $StrLab finished at $(date)" > $DirOup/$HisFillab
  else
    echo "SCF calculation for $StrLab finished at $(date)" > $DirOup/$HisFillab
  fi
else
  CalSCFEnd=$( grep -F -o "$CalSCFEndCon" $DirOup/$CalSCF.out ) ##CalSCFEnd=Calculation SCF Ended
  if [ "$CalSCFEnd" != "$CalSCFEndCon" ]; then
    echo "SCF calculation for $StrLab started again at $(date)" > $DirOup/$HisFillab
    mpirun --map-by core:pe=1 -np $NumPro -machinefile $PBS_NODEFILE pw.x < $DirInp/$CalSCF.in > $DirOup/$CalSCF.out
    CalSCFEnd=$( grep -F -o "$CalSCFEndCon" $DirOup/$CalSCF.out ) ##CalSCFEnd=Calculation SFC Ended
    if [ "$CalSCFEnd" != "$CalSCFEndCon" ]; then
      echo "SCF incomplete calculation for $StrLab finished at $(date)" > $DirOup/$HisFillab
    else
      echo "SCF calculation for $StrLab finished at $(date)" > $DirOup/$HisFillab
    fi
  else
    echo "SCF calculation for $StrLab was already completed!!!" > $DirOup/$HisFillab
  fi
fi

##Perform calculation < non-self consistent filed scf >#####
echo "Initiating NSCF calculation for $StrLab initiating..." > $DirOup/$HisFillab
CalNSCFEnd=$( grep -F -o "$CalNSCFEndCon" $DirOup/$CalNSCF.out ) ##CalNSCFEnd=Calculation SCF Ended
if [ "$CalNSCFEnd" = "$CalNSCFEndCon" ]; then
  if [ ! -f $DirOup/$CalNSCF.out ]; then
    echo "NSCF calculation for $StrLab started at $(date)" > $DirOup/$HisFillab
    mpirun --map-by core:pe=1 -np $NumPro -machinefile $PBS_NODEFILE pw.x < $DirInp/$CalNSCF.in > $DirOup/$CalNSCF.out
    CalNSCFEnd=$( grep -F -o "$CalNSCFEndCon" $DirOup/$CalNSCF.out ) ##CalNSCFEnd=Calculation Non-self Consistent Field Ended
    if [ "$CalNSCFEnd" != "$CalNSCFEndCon" ]; then
      sed -i 's/False/True/' $DirInp/$CalNSCF.in
      echo "NSCF incomplete calculation for $StrLab finished at $(date)" > $DirOup/$DirOup/$HisFillab
    else
      echo "NSCF calculation for $StrLab finished at $(date)" > $DirOup/$HisFillab
    fi
  else
    CalNSCFEnd=$( grep -F -o "$CalNSCFEndCon" $DirOup/$CalNSCF.out ) ##CalNSCFEnd=Calculation Non-self Consistent Field Ended
    if [ "$CalNSCFEnd" != "$CalNSCFEndCon" ]; then
      echo "NSCF calculation for $StrLab started again at $(date)" > $DirOup/$HisFillab
      mpirun --map-by core:pe=1 -np $NumPro -machinefile $PBS_NODEFILE pw.x < $DirOup/$CalNSCF.in > $DirOup/$CalNSCF.out
      CalNSCFEnd=$( grep -F -o "$CalNSCFEndCon" $DirOup/$CalNSCF.out ) ##CalNSCFEnd=Calculation Non-self Consistent Field Ended
      if [ "$CalNSCFEnd" != "$CalNSCFEndCon" ]; then
        echo "NSCF incomplete calculation for $StrLab finished at $(date)" > $DirOup/$HisFillab
      else
        echo "NSCF calculation for $StrLab finished at $(date)" > $DirOup/$HisFillab
      fi
    else
      echo "NSCF calculation for $StrLab was already completed!!!" > $DirOup/$HisFillab
    fi
  fi
fi

##Perform calculation < projected electronic density of states >#####
echo "Initiating projected electronic density of states calculation for $StrLab ..." > $DirOup/$HisFillab
CalSCFEnd=$( grep -F -o "$CalSCFEndCon" $DirOup/$CalSCF.out ) ##CalSCFEnd=Calculation SCF Ended
CalNSCFEnd=$( grep -F -o "$CalNSCFEndCon" $DirOup/$CalNSCF.out ) ##CalNSCFEnd=Calculation Non-self Consistent Field Ended
if [ "$CalSCFEnd" = "$CalSCFEndCon" ]; then
  if [ "$CalNSCFEnd" = "$CalNSCFEndCon" ]; then
    DElDir=$DirOup/$StrLab$DElFol ##DElDir=Density of States Electronic Directory
    if [ ! -d $DElDir ]; then
      echo "Directory $DElDir do not exist. The directory will be created." > $DirOup/$HisFillab
      mkdir $DElDir
    else
      echo "Directory $DElDir do exist. The directory will be cleaned." > $DirOup/$HisFillab
      rm -rf $DElDir
      mkdir $DElDir
    fi
  fi
fi

```

```

var_emi=$(echo $DELPer - $DELInt | bc -l) ##var_emi=variable Emin
var_ema=$(echo $DELPer + $DELInt | bc -l) ##var_ema=variable Emax
var_del=$(echo $var_ema - $var_emi | bc -l) ##var_Del=variable DeltaE
var_del=$(echo $var_del / $DELPOI | bc -l) ##var_Del=variable DeltaE
for i_var_nga in $VarNga ; do ##i_var_nga=i-counter type ngauss
for i_var_dga in $VarDga ; do ##i_var_dga=i-counter type degauss
var_nga="" #var_nga=variable ngauss
if [ "$i_var_nga" = "0" ]; then
var_nga="Simple-Gaussian"
fi
if [ "$i_var_nga" = "1" ]; then
var_nga="Methfessel-Paxton"
fi
if [ "$i_var_nga" = "-1" ]; then
var_nga="Marzari-Vanderbilt"
fi
if [ "$i_var_nga" = "-99" ]; then
var_nga="Fermi-Dirac"
fi
CalDElItelab="" "$var_nga" "$i_var_dga" "$var_del" "$var_emi" "$var_ema" ##CalDElItelab=Calculation Density of States Electronic Iteration Label
CalDElIte=$CalDEl$CalDElItelab ##CalDElIte=Calculation Density of States Electronic Iteration
echo "Initiating projected electronic density of states calculation for $CalDElIte ..." >> $DirOup/$HisFillLab
cp $DirInp/$CalDEl.in $DirInp/$CalDElIte.in
sed -i "s/$ComNga/$ComNga$i_var_nga/" $DirInp/$CalDElIte.in"
sed -i "s/$ComDga/$ComDga$i_var_dga/" $DirInp/$CalDElIte.in"
sed -i "s/$ComEmi/$ComEmi$var_emi/" $DirInp/$CalDElIte.in"
sed -i "s/$ComEma/$ComEma$var_ema/" $DirInp/$CalDElIte.in"
sed -i "s/$ComDel/$ComDel$var_del/" $DirInp/$CalDElIte.in"
if [ ! -f $DirOup/$CalDElIte.out ]; then
echo "Projected electronic density of states calculation for $CalDElIte started at $(date)" >> $DirOup/$HisFillLab
mpirun --map-by core:pe=1 -np $NumPro -machinefile $PBS_NODEFILE projwfc.x < $DirInp/$CalDElIte.in > $DirOup/$CalDElIte.out"
CalDElEnd=$( grep -F -o "$CalDElEndCon" $DirOup/$CalDElIte.out" ) ##CalDElEnd=Calculation Density of states Electronic Ended
if [ "$CalDElEnd" != "$CalDElEndCon" ]; then
echo "projected electronic density of states incomplete calculation for $CalDElIte finished at $(date)" >> $DirOup/$HisFillLab
else
echo "projected electronic density of states calculation for $CalDElIte finished at $(date)" >> $DirOup/$HisFillLab
fi
else
CalDElEnd=$( grep -F -o "$CalDElEndCon" $DirOup/$CalDElIte.out" ) ##CalDElEnd=Calculation Density of states Electronic Ended
if [ "$CalAbsEnd" != "$CalDElEnd" ]; then
echo "projected electronic density of states calculation for $CalDElIte started again at $(date)" >> $DirOup/$HisFillLab
mpirun --map-by core:pe=1 -np $NumPro -machinefile $PBS_NODEFILE projwfc.x < $DirInp/$CalDElIte.in > $DirOup/$CalDElIte.out"
CalDElEnd=$( grep -F -o "$CalDElEndCon" $DirOup/$CalDElIte.out" ) ##CalDElEnd=Calculation Density of states Electronic Ended
if [ "$CalDElEnd" != "$CalDElEndCon" ]; then
echo "projected electronic density of states incomplete calculation for $CalDElIte finished at $(date)" >> $DirOup/$HisFillLab
else
echo "projected electronic density of states calculation for $CalDElIte finished at $(date)" >> $DirOup/$HisFillLab
fi
else
echo "projected electronic density of states calculation for $CalDElIte was already completed!!!" >> $DirOup/$HisFillLab
fi
fi
rm $DirInp/$CalDElIte.in"
CalDElEnd=$( grep -F -o "$CalDElEndCon" $DirOup/$CalDElIte.out" ) ##CalDElEnd=Calculation Density of states Electronic Ended
if [ "$CalDElEnd" = "$CalDElEndCon" ]; then
echo "Organizing projected electronic density of states for $CalDElIte..." >> $DirOup/$HisFillLab
DELlIteDir=$DELDir/"$CalDElIte" #DELlIteDir=Density of States Electronic Iteration Directory
if [ ! -d $DELlIteDir ]; then
mkdir $DELlIteDir
cp $StrLab$DELSeaPro* $DELlIteDir
cp $StrLab$DELSeaTot* $DELlIteDir
echo "projected electronic density of states for $CalDElIte organized at $(date)" >> $DirOup/$HisFillLab
else
rm -rf $DELlIteDir
mkdir $DELlIteDir
cp $StrLab$DELSeaPro* $DELlIteDir
cp $StrLab$DELSeaTot* $DELlIteDir
echo "projected electronic density of states for $CalDElIte organized at $(date)" >> $DirOup/$HisFillLab
fi
else
echo "Organizing projected electronic density of states for $CalDElIte could not be performed." >> $DirOup/$HisFillLab
fi
rm $DirOup/$CalDElIte.out"
done
done
else
echo "NSCF has not been perform yet... The PDoS calculation will not be perform." >> $DirOup/$HisFillLab
fi
else
echo "SCF has not been perform yet... The PDoS calculation will not be perform." >> $DirOup/$HisFillLab
fi

```

Archivo para lanzar simulación de espectro de absorción en SGN R2H1 VdW - CQD_SGN_R2H1_ST1_VDW-DF_SAB.pbs

```

#!/bin/bash
#PBS -N SAB_CQD_SGN_R2H1_ST1_VDW-DF
#PBS -M jacuadraa@yaxchilan.larcad.mx
#PBS -l nodes=2:ppn=8,walltime=212:00:00
#PBS -j eo
#PBS -k eo
#PBS -m abe

```

```

#PBS -q mctp

##*****
##Variables*****
##*****
NumPro=16 ##NumPro=Number Processors
NumProSafe=8 ##NumPro=Number Processors
StrLab=CQD_SGN_R2H1_ST1_VDW-DF ##StrLab=Structure Label
StrTyp=CQD_Stacking_Graphite_Nitrogenized ##StrTyp=Structure Type
DirHom=/home/jcquadra ##DirHom=Directory Home
DirInp=/home/jcquadra/Calculations_Inputs/StrTyp/StrLab ##DirInp=Directory Inputs
DirOut=/home/jcquadra/Calculations_Outputs/StrTyp/StrLab ##DirOut=Directory Outputs
DirDat=/home/jcquadra/Calculations_Data/StrTyp/StrLab/results_sab ##DirDat=Directory Data
CalSCF=calculation_sab_1_pv_scf ##CalSCF=Calculation Self Consistent Field
Callan=calculation_sab_2_turbo_lanczos_coe ##Callan=Calculation Lanczos
CalAbs=calculation_sab_3_turbo_spectrum_sab ##CalAbs=Calculation Absorbance
HisFillLab=StrLab"_ABS.his" ##HisFillLab=History File Label

AbsFol="_SAB" ##AbsFol=Absorbance Folder
AbsLab="_plot_chi.dat" ##AbsLab=Absorbance Label
AbsSpeSea="S(E)" ##AbsSpeSea=Absorbance Spectrum Search
AbsSpeHeaLab="Energy Absorbance" ##AbsSpeHeaLab=Absorbance Spectrum Header Labels
AbsSpeHeaUni="eV" ##AbsSpeHeaUni=Absorbance Spectrum Header Units
AbsSpeDel=" S(E) = " ##AbsSpeDel=Absorbance Spectrum Delete
AbsBacLab="_Chi" ##AbsBacLab=Absorbance Backup Label
AbsSavTyp=".dat" ##AbsSavTyp=Absorbance Save Type
CalSCFEndCon="End of self-consistent calculation" ##CalSCFEndCon=Calculation Self Consistent End Condition
CallanEndCon="End of Lanczos iterations" ##CallanEndCon=Calculation Lanczos End Condition
CalAbsEndCon="JOB DONE" ##CalAbsEndCon=Calculation Absorbance End Condition
ComExt="extrapolation=" ##ComExt=Command Extrapolation
ComIte="itermax=" ##ComIte=Command Itermax
ComEps="epsil=" ##ComEps=Command Epsil
VarExt="'constant' 'osc'" ##VarExt=Variable Extrapolation
VarIte="10000 12000 14000" ##VarIte=Variable Itermax
VarEps="1.0D-4 0.5D-2 1.0D-2 2.0D-2 4.0D-2" ##VarEps=Variable Epsil

##*****
##Calculations*****
##*****
module load CondensedmSW/QEspresso-6.3
if [ ! -f $DirOut/HisFillLab ]; then
echo "*****" > $DirOut/HisFillLab
echo "Absorbance spectrum calculations for $StrLab started at $(date)" > $DirOut/HisFillLab
echo "*****" > $DirOut/HisFillLab
else
echo "*****" >> $DirOut/HisFillLab
echo "Absorbance spectrum calculations for $StrLab started at $(date)" >> $DirOut/HisFillLab
echo "*****" >> $DirOut/HisFillLab
fi
##Perform calculation < self consistent field scf >*****
echo "Initiating SCF calculation for $StrLab ..." >> $DirOut/HisFillLab
if [ ! -f $DirOut/CalSCF ]; then
echo "SCF calculation for $StrLab started at $(date)" >> $DirOut/HisFillLab
mpirun -np $NumPro --map-by core -machinefile $PBS_NODEFILE pw.x < $DirInp/CalSCF.in > $DirOut/CalSCF.out
CalSCFEnd=$( grep -F -o "$CalSCFEndCon" $DirOut/CalSCF.out ) ##CalSCFEnd=Calculation SFC Ended
if [ "$CalSCFEnd" != "$CalSCFEndCon" ]; then
sed -i 's/from_scratch/restart/' $DirInp/CalSCF.in
echo "SCF incomplete calculation for $StrLab finished at $(date)" >> $DirOut/HisFillLab
else
echo "SCF calculation for $StrLab finished at $(date)" >> $DirOut/HisFillLab
fi
else
CalSCFEnd=$( grep -F -o "$CalSCFEndCon" $DirOut/CalSCF.out ) ##CalSCFEnd=Calculation SCF Ended
if [ "$CalSCFEnd" != "$CalSCFEndCon" ]; then
echo "SCF calculation for $StrLab started again at $(date)" >> $DirOut/HisFillLab
mpirun -np $NumPro --map-by core -machinefile $PBS_NODEFILE pw.x < $DirInp/CalSCF.in > $DirOut/CalSCF.out
CalSCFEnd=$( grep -F -o "$CalSCFEndCon" $DirOut/CalSCF.out ) ##CalSCFEnd=Calculation SFC Ended
if [ "$CalSCFEnd" != "$CalSCFEndCon" ]; then
echo "SCF incomplete calculation for $StrLab finished at $(date)" >> $DirOut/HisFillLab
else
echo "SCF calculation for $StrLab finished at $(date)" >> $DirOut/HisFillLab
fi
else
echo "SCF calculation for $StrLab was already completed!!!" >> $DirOut/HisFillLab
fi
fi
##Perform calculation < lanczos coefficients >*****
echo "Lanczos coefficients calculation for $StrLab initiating..." >> $DirOut/HisFillLab
CalSCFEnd=$( grep -F -o "$CalSCFEndCon" $DirOut/CalSCF.out ) ##CalSCFEnd=Calculation SCF Ended
if [ "$CalSCFEnd" = "$CalSCFEndCon" ]; then
if [ ! -f $DirOut/Callan ]; then
echo "Lanczos coefficients calculation for $StrLab started at $(date)" >> $DirOut/HisFillLab
mpirun -np $NumPro --map-by core -machinefile $PBS_NODEFILE turbo_lanczos.x < $DirInp/Callan.in > $DirOut/Callan.out
CallanEnd=$( grep -F -o "$CallanEndCon" $DirOut/Callan.out ) ##CallanEnd=Calculation Lanczos Ended
if [ "$CallanEnd" != "$CallanEndCon" ]; then
sed -i 's/False/True/' $DirInp/Callan.in
echo "Lanczos coefficients incomplete calculation for $StrLab finished at $(date)" >> $DirOut/HisFillLab
else
echo "Lanczos coefficients calculation for $StrLab finished at $(date)" >> $DirOut/HisFillLab
fi
else
CallanEnd=$( grep -F -o "$CallanEndCon" $DirOut/Callan.out ) ##CallanEnd=Calculation Lanczos Ended
if [ "$CallanEnd" != "$CallanEndCon" ]; then
echo "Lanczos coefficients calculation for $StrLab started again at $(date)" >> $DirOut/HisFillLab
mpirun -np $NumPro --map-by core -machinefile $PBS_NODEFILE turbo_lanczos.x < $DirInp/Callan.in > $DirOut/Callan.out
CallanEnd=$( grep -F -o "$CallanEndCon" $DirOut/Callan.out ) ##CallanEnd=Calculation Lanczos Ended
if [ "$CallanEnd" != "$CallanEndCon" ]; then
echo "Lanczos coefficients incomplete calculation for $StrLab finished at $(date)" >> $DirOut/HisFillLab
else

```

```

        echo "Lanczos coefficients calculation for $StrLab finished at $(date)" >> $DirOup/$HisFillLab
    fi
else
    echo "Lanczos coefficients calculation for $StrLab was already completed!!!" >> $DirOup/$HisFillLab
fi
fi
fi
#Perform calculation < absorbance spectrum >*****
echo "Initiating absorbance spectrum calculation for $StrLab ..." >> $DirOup/$HisFillLab
CalSCFEnd=$( grep -F -o "$CalSCFEndCon" $DirOup/$CalSCF.out ) ##CalSCFEnd=Calculation SCF Ended
CallanEnd=$( grep -F -o "$CallanEndCon" $DirOup/$Callan.out ) ##CallanEnd=Calculation Lanczos Ended
if [ "$CalSCFEnd" = "$CalSCFEndCon" ]; then
if [ "$CallanEnd" = "$CallanEndCon" ]; then
AbsDir=$DirOup/$StrLab$AbsFol
if [ ! -d $AbsDir ]; then
    echo "Directory $AbsDir do not exist. The directory will be created." >> $DirOup/$HisFillLab
    mkdir $AbsDir
else
    echo "Directory $AbsDir do exist. Any new directory will be created." >> $DirOup/$HisFillLab
fi
for i_var_ext in $VarExt ; do ##i_ttyp_ext=i-counter type extrapolation
for i_var_ite in $VarIte ; do ##i_ttyp_ite=i-counter type itermax
for i_var_eps in $VarEps ; do ##i_ttyp_eps=i-counter type epsilon
CalAbsIteLab="$i_var_ext"$i_var_ite"$i_var_eps" ##CalAbsIteLab=Calculation Absorbance Iteration Label
CalAbsIte=$CalAbs$CalAbsIteLab ##CalAbsIte=Calculation Absorbance Iteration
echo "Initiating absorbance spectrum calculation for $CalAbsIte ..." >> $DirOup/$HisFillLab
cp $DirInp/$CalAbs.in $DirInp/$CalAbsIte.in
sed -i "s/$ComExt/$ComExt$i_var_ext/" $DirInp/$CalAbsIte.in
sed -i "s/$ComIte/$ComIte$i_var_ite/" $DirInp/$CalAbsIte.in
sed -i "s/$ComEps/$ComEps$i_var_eps/" $DirInp/$CalAbsIte.in
if [ ! -f $DirOup/$CalAbsIte.out ]; then
    echo "Absorbance spectrum calculation for $CalAbsIte started at $(date)" >> $DirOup/$HisFillLab
    mpirun -np $NumPro --map-by core -machinefile $PBS_NODEFILE turbo_spectrum.x < $DirInp/$CalAbsIte.in > $DirOup/$CalAbsIte.out
    CalAbsEnd=$( grep -F -o "$CalAbsEndCon" $DirOup/$CalAbsIte.out ) ##CalAbsEndCon=Calculation Absorbance Ended
    if [ "$CalAbsEnd" != "$CalAbsEndCon" ]; then
        echo "Absorbance spectrum incomplete calculation for $CalAbsIte finished at $(date)" >> $DirOup/$HisFillLab
    else
        echo "Absorbance spectrum calculation for $CalAbsIte finished at $(date)" >> $DirOup/$HisFillLab
    fi
else
    CalAbsEnd=$( grep -F -o "$CalAbsEndCon" $DirOup/$CalAbsIte.out ) ##CalAbsEndCon=Calculation Absorbance Ended
    if [ "$CalAbsEnd" != "$CalAbsEndCon" ]; then
        echo "Absorbance spectrum calculation for $CalAbsIte started again at $(date)" >> $DirOup/$HisFillLab
        mpirun -np $NumPro --map-by core -machinefile $PBS_NODEFILE turbo_spectrum.x < $DirInp/$CalAbsIte.in > $DirOup/$CalAbsIte.out
        CalAbsEnd=$( grep -F -o "$CalAbsEndCon" $DirOup/$CalAbsIte.out ) ##CalAbsEndCon=Calculation Absorbance Ended
        if [ "$CalAbsEnd" != "$CalAbsEndCon" ]; then
            echo "Absorbance spectrum incomplete calculation for $CalAbsIte finished at $(date)" >> $DirOup/$HisFillLab
        else
            echo "Absorbance spectrum calculation for $CalAbsIte finished at $(date)" >> $DirOup/$HisFillLab
        fi
    else
        echo "Absorbance spectrum calculation for $CalAbsIte was already completed!!!" >> $DirOup/$HisFillLab
    fi
fi
rm $DirInp/$CalAbsIte.in
CalAbsEnd=$( grep -F -o "$CalAbsEndCon" $DirOup/$CalAbsIte.out ) ##CalAbsEndCon=Calculation Absorbance Ended
if [ "$CalAbsEnd" = "$CalAbsEndCon" ]; then
    echo "Initiating absorbance spectrum $AbsSavTyp file creation..." >> $DirOup/$HisFillLab
    if [ ! -f $CalAbsIte$AbsSavTyp ]; then
        echo "Absorbance spectrum $AbsSavTyp file creation started at $(date)" >> $DirOup/$HisFillLab
        cp $DirHom/$StrLab$AbsLab $AbsDir/$StrLab$CalAbsIteLab$AbsBacLab$AbsSavTyp
        rm $DirHom/$StrLab$AbsLab
        grep -F "$AbsSpeSea" $AbsDir/$StrLab$CalAbsIteLab$AbsBacLab$AbsSavTyp > $AbsDir/$StrLab$CalAbsIteLab"$SAB"$AbsSavTyp
        sed -i id $AbsDir/$StrLab$CalAbsIteLab"$SAB"$AbsSavTyp
        sed -i "1s/~/AbsSpeHeaLab\n/" $AbsDir/$StrLab$CalAbsIteLab"$SAB"$AbsSavTyp
        sed -i "2s/~/AbsSpeHeaUni\n/" $AbsDir/$StrLab$CalAbsIteLab"$SAB"$AbsSavTyp
        sed -i "s/$AbsSpeDel/~/ $AbsDir/$StrLab$CalAbsIteLab"$SAB"$AbsSavTyp
        echo "Absorbance spectrum $AbsSavTyp file creation finished at $(date)" >> $DirOup/$HisFillLab
    else
        echo "Absorbance spectrum $AbsSavTyp file creation was already completed!!!" >> $DirOup/$HisFillLab
    fi
else
    echo "Absorbance spectrum $AbsSavTyp file creation could not be performed." >> $DirOup/$HisFillLab
fi
rm $DirOup/$CalAbsIte.out
done
done
done
fi
fi

```

Archivo para lanzar cálculo de respuesta electrónica en SGN R2H1 VdW - CQD_SGN_R2H1_ST1_VDW-DF_REL.pb

```

#!/bin/bash
#PBS -N REL_CQD_SGN_R2H1_ST1_VDW-DF
#PBS -M jacuadraa@yaxchilan.larcad.mx
#PBS -l nodes=1:ppn=8,walltime=212:00:00
#PBS -j eo
#PBS -k eo
#PBS -m abe
#PBS -q mctp

```



```

#####
##Variables#####
#####
NumPro=8 ##NumPro=Number Processors
NumProSafe=8 ##NumPro=Number Processors
StrLab=CQD_SGN_R2H1_ST1_VDW-DF ##StrLab=Structure Label
StrTyp=CQD_Stacking_Graphite_Nitrogenized ##StrTyp=Structure Type
DirHom=/home/jcuadra ##DirHom=Directory Home
DirInp=/home/jcuadra/Calculations_Inputs/$StrTyp/$StrLab ##DirInp=Directory Inputs
DirOut=/home/jcuadra/Calculations_Outputs/$StrTyp/$StrLab ##DirOut=Directory Outputs
DirDat=/home/jcuadra/Calculations_Data/$StrTyp/$StrLab/results_sab ##DirDat=Directory Data
CalSCF=calculation_sab_1_pw_scf ##CalSCF=Calculation Self Consistent Field
Callan=calculation_sab_2_turbo_lanczos_coe ##Callan=Calculation Lanczos
CalRes=calculation_sab_4_turbo_lanczos_res ##CalAbs=Calculation Response
HisFillLab=$StrLab".REL.his" ##HisFillLab=History File Label

ResFol="_REL" ##ResFol=Response Folder
ResLab=".plot_chi.dat" ##ResLab=Response Label
ResSpeSea="S(E)" ##ResSpeSea=Response Spectrum Search
ResSpeHeaLab="Energy Absorbance" ##ResSpeHeaLab=Response Spectrum Header Labels
ResSpeHeaUni="eV" ##ResSpeHeaUni=Response Spectrum Header Units
ResSpeDel="S(E)=" ##ResSpeDel=Response Spectrum Delete
ResBacLab="_Chi" ##ResBacLab=Response Backup Label
ResSavTyp=".dat" ##ResSavTyp=Response Save Type
CalSCFEndCon="End of self-consistent calculation" ##CalSCFEndCon=Calculation Self Consistent End Condition
CallanEndCon="End of Lanczos iterations" ##CallanEndCon=Calculation Lanczos End Condition
CalResEndCon="JOB DONE" ##CalPolEndCon=Calculation Response End Condition
ComOme="omeg=" ##ComIte=Command Itermax
ComEps="epsil=" ##ComEps=Command Epsil
VarOme="0.3674930882" ##VarIte=Variable Itermax. Rydberg units
VarEps="0 1.0D-3 1.0D-2 1.0D-1" ##VarEps=Variable Epsil

#####
##Calculations#####
#####
module load CondensedMW/QEspresso-6.3
if [ ! -f $DirOut/$HisFillLab ]; then
echo "#####" > $DirOut/$HisFillLab
echo "Response charge calculations for $StrLab started at $(date)" > $DirOut/$HisFillLab
echo "#####" > $DirOut/$HisFillLab
else
echo "#####" > $DirOut/$HisFillLab
echo "Response charge calculations for $StrLab started at $(date)" > $DirOut/$HisFillLab
echo "#####" > $DirOut/$HisFillLab
fi
##Perform calculation < self consistent filed scf >#####
echo "Initiating SCF calculation for $StrLab ..." > $DirOut/$HisFillLab
if [ ! -f $DirOut/$CalSCF".out" ]; then
echo "SCF calculation for $StrLab started at $(date)" > $DirOut/$HisFillLab
mpirun -np $NumPro --map-by core -machinefile $PBS_NODEFILE pw.x < $DirInp/$CalSCF".in" > $DirOut/$CalSCF".out"
CalSCFEnd=$( grep -F -o "$CalSCFEndCon" $DirOut/$CalSCF".out" ) ##CalSCFEnd=Calculation SFC Ended
if [ "$CalSCFEnd" != "$CalSCFEndCon" ]; then
sed -i 's/from_scratch/restart/' $DirInp/$CalSCF".in"
echo "SCF incomplete calculation for $StrLab finished at $(date)" > $DirOut/$HisFillLab
else
echo "SCF calculation for $StrLab finished at $(date)" > $DirOut/$HisFillLab
fi
else
CalSCFEnd=$( grep -F -o "$CalSCFEndCon" $DirOut/$CalSCF".out" ) ##CalSCFEnd=Calculation SCF Ended
if [ "$CalSCFEnd" != "$CalSCFEndCon" ]; then
echo "SCF calculation for $StrLab started again at $(date)" > $DirOut/$HisFillLab
mpirun -np $NumPro --map-by core -machinefile $PBS_NODEFILE pw.x < $DirInp/$CalSCF".in" > $DirOut/$CalSCF".out"
CalSCFEnd=$( grep -F -o "$CalSCFEndCon" $DirOut/$CalSCF".out" ) ##CalSCFEnd=Calculation SFC Ended
if [ "$CalSCFEnd" != "$CalSCFEndCon" ]; then
echo "SCF incomplete calculation for $StrLab finished at $(date)" > $DirOut/$HisFillLab
else
echo "SCF calculation for $StrLab finished at $(date)" > $DirOut/$HisFillLab
fi
else
echo "SCF calculation for $StrLab was already completed!!" > $DirOut/$HisFillLab
fi
fi
##Perform calculation < lanczos coefficients >#####
echo "Lanczos coefficients calculation for $StrLab initiating..." > $DirOut/$HisFillLab
CalSCFEnd=$( grep -F -o "$CalSCFEndCon" $DirOut/$CalSCF".out" ) ##CalSCFEnd=Calculation SCF Ended
if [ "$CalSCFEnd" = "$CalSCFEndCon" ]; then
if [ ! -f $DirOut/$Callan".out" ]; then
echo "Lanczos coefficients calculation for $StrLab started at $(date)" > $DirOut/$HisFillLab
mpirun -np $NumPro --map-by core -machinefile $PBS_NODEFILE turbo_lanczos.x < $DirInp/$Callan".in" > $DirOut/$Callan".out"
CallanEnd=$( grep -F -o "$CallanEndCon" $DirOut/$Callan".out" ) ##CallanEnd=Calculation Lanczos Ended
if [ "$CallanEnd" != "$CallanEndCon" ]; then
sed -i 's/False/True/' $DirInp/$Callan".in"
echo "Lanczos coefficients incomplete calculation for $StrLab finished at $(date)" > $DirOut/$DirOut/$HisFillLab
else
echo "Lanczos coefficients calculation for $StrLab finished at $(date)" > $DirOut/$HisFillLab
fi
else
CallanEnd=$( grep -F -o "$CallanEndCon" $DirOut/$Callan".out" ) ##CallanEnd=Calculation Lanczos Ended
if [ "$CallanEnd" != "$CallanEndCon" ]; then
echo "Lanczos coefficients calculation for $StrLab started again at $(date)" > $DirOut/$HisFillLab
mpirun -np $NumPro --map-by core -machinefile $PBS_NODEFILE turbo_lanczos.x < $DirOut/$Callan".in" > $DirOut/$Callan".out"
CallanEnd=$( grep -F -o "$CallanEndCon" $DirOut/$Callan".out" ) ##CallanEnd=Calculation Lanczos Ended
if [ "$CallanEnd" != "$CallanEndCon" ]; then
echo "Lanczos coefficients incomplete calculation for $StrLab finished at $(date)" > $DirOut/$HisFillLab
else
echo "Lanczos coefficients calculation for $StrLab finished at $(date)" > $DirOut/$HisFillLab
fi
else
fi
else
fi

```

```

echo "Lanczos coefficients calculation for $StrLab was already completed!!!" >> $DirOup/$HisFillab
fi
fi
fi
##Perform calculation < polarization electronic density >*****
echo "Initiating response charge calculation for $StrLab ..." >> $DirOup/$HisFillab
CalSCFEnd=$( grep -F -o "$CalSCFEndCon" $DirOup/$CalSCF.out ) ##CalSCFEnd=Calculation SCF Ended
CallanEnd=$( grep -F -o "$CallanEndCon" $DirOup/$Callan.out ) ##CallanEnd=Calculation Lanczos Ended
if [ "$CalSCFEnd" = "$CalSCFEndCon" ]; then
if [ "$CallanEnd" = "$CallanEndCon" ]; then
ResDir=$DirOup/"$StrLab"$ResFol
if [ ! -d $ResDir ] ; then
echo "Directory $ResDir do not exist. The directory will be created." >> $DirOup/$HisFillab
mkdir $ResDir
else
echo "Directory $ResDir do exist. Any new directory will be created." >> $DirOup/$HisFillab
fi
for i_var_ome in $VarOme ; do ##i_typ_ext=i-counter variable omega
for i_var_eps in $VarEps ; do ##i_typ_eps=i-counter variable epsilon
CalResIteLab="$ComOme" "$i_var_ome" "$ComEps" "$i_var_eps" ##CalResIteLab=Calculation Response Iteration Label
CalResIte=$CalRes$CalResIteLab ##CalResIte=Calculation Response Iteration
echo "Initiating response charge calculation for $CalResIteLab ..." >> $DirOup/$HisFillab
cp $DirInp/$CalRes.in $DirInp/$CalResIte.in
sed -i "s/$ComOme/$ComOme$i_var_ome/" $DirInp/$CalResIte.in"
sed -i "s/$ComEps/$ComEps$i_var_eps/" $DirInp/$CalResIte.in"
if [ ! -f $DirOup/$CalResIte.out ] ; then
echo "Response charge calculation for $CalResIte started at $(date)" >> $DirOup/$HisFillab
mpirun -np $NumPro --map-by core -machinefile $PBS_NODEFILE turbo_lanczos.x < $DirInp/$CalResIte.in > $DirOup/$CalResIte.out"
CalResEnd=$( grep -F -o "$CalResEndCon" $DirOup/$CalResIte.out ) ##CalResEnd=Calculation Response Ended
if [ "$CalResEnd" != "$CalResEndCon" ]; then
echo "Response charge incomplete calculation for $CalResIte finished at $(date)" >> $DirOup/$HisFillab
else
echo "Response charge complete calculation for $CalResIte finished at $(date)" >> $DirOup/$HisFillab
ResDirAct=$ResDir/$CalResIte
if [ ! -d $ResDirAct ] ; then
echo "Directory $ResDirAct do not exist. The directory will be created." >> $DirOup/$HisFillab
mkdir $ResDirAct
else
echo "Directory $ResDirAct do exist. Any new directory will be created." >> $DirOup/$HisFillab
fi
mv $StrLab* $ResDirAct
fi
else
CalResEnd=$( grep -F -o "$CalResEndCon" $DirOup/$CalResIte.out ) ##CalResEnd=Calculation Response Ended
if [ "$CalResEnd" != "$CalResEndCon" ]; then
echo "Response charge calculation for $CalResIte started again at $(date)" >> $DirOup/$HisFillab
mpirun -np $NumPro --map-by core -machinefile $PBS_NODEFILE turbo_lanczos.x < $DirInp/$CalResIte.in > $DirOup/$CalResIte.out"
CalResEnd=$( grep -F -o "$CalResEndCon" $DirOup/$CalResIte.out ) ##CalResEnd=Calculation Response Ended
if [ "$CalResEnd" != "$CalResEndCon" ]; then
echo "Response charge incomplete calculation for $CalResIte finished at $(date)" >> $DirOup/$HisFillab
else
echo "Response charge complete calculation for $CalResIte finished at $(date)" >> $DirOup/$HisFillab
ResDirAct=$ResDir/$CalResIte
if [ ! -d $ResDirAct ] ; then
echo "Directory $ResDirAct do not exist. The directory will be created." >> $DirOup/$HisFillab
mkdir $ResDirAct
else
echo "Directory $ResDirAct do exist. Any new directory will be created." >> $DirOup/$HisFillab
fi
mv $StrLab* $ResDirAct
fi
else
echo "Response charge calculation for $CalResIte was already completed!!!" >> $DirOup/$HisFillab
fi
fi
rm $DirInp/$CalResIte.in"
##CalResEnd=$( grep -F -o "$CalResEndCon" $DirOup/$CalResIte.out ) ##CalResEnd=Calculation Response Ended
##if [ "$CalResEnd" = "$CalResEndCon" ]; then
## echo "Initiating absorbance spectrum $ResSavTyp file creation..." >> $DirOup/$HisFillab
## if [ ! -f $CalResIte$ResSavTyp ] ; then
## echo "Response charge $ResSavTyp file creation started at $(date)" >> $DirOup/$HisFillab
## cp $DirHom/$StrLab$ResLab $ResDir/$StrLab$CalResIteLab$ResBacLab$ResSavTyp
## rm $DirHom/$StrLab$ResLab
## grep -F "$ResSpeSea" $ResDir/$StrLab$CalResIteLab$ResBacLab$ResSavTyp > $ResDir/$StrLab$CalResIteLab"_RES"$ResSavTyp
## sed -i 1d $ResDir/$StrLab$CalResIteLab"_SAB"$ResSavTyp
## sed -i "1s/~/ $ResDir/$StrLab$CalResIteLab"_SAB"$ResSavTyp
## sed -i "2s/~/ $ResDir/$StrLab$CalResIteLab"_SAB"$ResSavTyp
## sed -i "s/~/ $AbsDir/$StrLab$CalResIteLab"_SAB"$ResSavTyp
## echo "Response charge $ResSavTyp file creation finished at $(date)" >> $DirOup/$HisFillab
## else
## echo "Response charge $ResSavTyp file creation was already completed!!!" >> $DirOup/$HisFillab
## fi
##else
## echo "Response charge $ResSavTyp file creation could not be performed." >> $DirOup/$HisFillab
##fi
##rm $DirOup/$CalResIte.out"
done
done
fi
fi

```

5. Anexo E: Código de Python3 para apilar placas de grafeno

Archivo *input* y principal para apilamiento de placas de grafeno: Graphite_Layer_Stacking.py

```
from numpy import zeros
from RaW_TXT import RNLTF #RNLTF=Read Number Lines Text File.
from RaW_TXT import RDSTF #RDSTF=Read Data as String form Text File.
from RaW_TXT import RDFTF #RDFTF=Read Data as Float form Text File.
from RaW_TXT import WDSTF #WDSTF=Write Data as String form Text File.

#PFNW='./Periodicity_Layer/' #PFNW=Path File Name written.
PFNW='./Periodicity_Graphite/' #PFNW=Path File Name written.
RS=3 #RS=Radius Structure.
FNR='./Unit_Layer_Relaxed/CQD_SLN_R'+str(RS)+'H1.xyz' #FNR=File Name Read.
FNW='CQD_SGN_R'+str(RS)+'H' #FNW=File Name written.

TSP='.xyz' #TSP=Type Save File.

DP="%0.5f" #DP=Decimal Precision.

IL=2 #IL=Ignored Lines.
IC=1 #IC=Ignored Columns.

MinNumLay=1 #Minimum Number of Layers.
MaxNumLay=6 #MaxNumLay=Maximum Number of Layers.
StpNumLay=1 #StpNumLay=Step Number of Layers.

MaxNumLay=MaxNumLay+1

#Periodicity==1 means that the layers will not be displaced.
PX=1 #PX=Periodicity X.
PY=2 #PY=Periodicity Y.

BX=2.0 #BX=Box X.
BY=2.0 #BY=Box Y.
BZ=2.0 #BZ=Box Z.

NoD=0.5 #NoD=Number of Decimals.
DX=1.417984 #DX=Delta X. Units anstrogn.
DY=1.417984 #DY=Delta Y. Units anstrogn.
DZ=3.348000 #DZ=Delta Z. Units anstrogn.
GLS=True #GLS=Graphite Like Stacking.

#CrCS=Create Carbon Structure.
#noal=number of atoms. integer variable. This variable indicate the total number of atoms per layer.
#nols=number of layers structure. integer variable. This variable indicate the number of layers to stack.
#ael=atomic elements layer. string array dimensions [noal][1].
#apl=atomic positions layer. float array dimensions [noal][3].
#tysta=type stacking. bool variable. This indicate the type of layer staking.
def CrCS(noal,nols,ael,apl,tysta):
    noas=noal*nols #noas=number of atoms structure.
    aes=zeros((noas,1),dtype=str) #aes=atomic elements structure.
    aps=zeros((noas,3),dtype=float) #aps=atomic positions structure.
    if tysta==True:
        inoas=0 #inoas=index number of atoms structure.
        for inols in range(0,nols,1): #inols=index number of layers structure.
            for iapl in range(0,noal,1): #inoal=index number of atoms. integer variable. This variable indicate the total number of atoms per layer.
                aes[inoas][0]=ael[iapl][0]
                if inols%PX==0 and PX!=1: aps[inoas][0]='%'+str(NoD)+'f'%(apl[iapl][0]+DX)
                else: aps[inoas][0]='%'+str(NoD)+'f'%(apl[iapl][0])
                if inols%PY==0 and PY!=1: aps[inoas][1]='%'+str(NoD)+'f'%(apl[iapl][1]+DY)
                else: aps[inoas][1]='%'+str(NoD)+'f'%(apl[iapl][1])
                aps[inoas][2]='%'+str(NoD)+'f'%(apl[iapl][2]+inols*DZ)
                inoas=inoas+1
            else:
                inoas=0 #inoas=index number of atoms structure.
                for inols in range(0,nols,1): #inols=index number of layers structure.
                    for iapl in range(0,noal,1): #inoal=index number of atoms. integer variable. This variable indicate the total number of atoms per layer.
                        aes[inoas][0]=ael[iapl][0]
                        aps[inoas][0]='%'+str(NoD)+'f'%(apl[iapl][0])
                        aps[inoas][1]='%'+str(NoD)+'f'%(apl[iapl][1])
                        aps[inoas][2]='%'+str(NoD)+'f'%(apl[iapl][2]+inols*DZ)
                        inoas=inoas+1
                    return aes,aps

#CeCS=Centering Carbon Structure.
#aps=atomic positions structure. integer variable. This variable indicates the cartesian positions of the atoms in the structure.
#noas=number of atoms structure. integer variable. This variable indicate the total number of atoms in the structure.
def CeCS(noas,aps):
    newaps=zeros((noas,3),dtype=float) #newaes=new number atomic positions structure.
    box=zeros((3,),dtype=float)
    limx,limy,limz=zeros((2,),dtype=float),zeros((2,),dtype=float),zeros((2,),dtype=float) #limx=limits x. limy=limits y. limz=limits z.
    limx[0],limy[0],limz[0]=aps[0][0],aps[0][1],aps[0][2]
    limx[1],limy[1],limz[1]=aps[0][0],aps[0][1],aps[0][2]
    for inoas in range(0,noas,1): #inoas=index number of atoms structure.
        if aps[inoas][0]<limx[0]: limx[0]=aps[inoas][0]
        if aps[inoas][0]>limx[1]: limx[1]=aps[inoas][0]
        if aps[inoas][1]<limy[0]: limy[0]=aps[inoas][1]
        if aps[inoas][1]>limy[1]: limy[1]=aps[inoas][1]
        if aps[inoas][2]<limz[0]: limz[0]=aps[inoas][2]
        if aps[inoas][2]>limz[1]: limz[1]=aps[inoas][2]
    disx,disy,disz=BX-limx[0],BY-limy[0],BZ-limz[0] #disx=displacement x. disy=displacement y. disz=displacement z.
```

```

maxbx,maxby,maxbz=2*Bx+limx[1]-limx[0],2*By+limy[1]-limy[0],2*Bz+limz[1]-limz[0] #maxbx=maximum box x. #maxby=maximum box y. #maxbz=maximum box z.
for inoas in range(0,noas,1): #inoas=index number of atoms structure.
    newaps[inoas][0]=aps[inoas][0]+disx
    newaps[inoas][1]=aps[inoas][1]+disy
    newaps[inoas][2]=aps[inoas][2]+disz
box[0],box[1],box[2]='%'+str(NoD)+'f' %(maxbx),('%'+str(NoD)+'f' %(maxby),('%'+str(NoD)+'f' %(maxbz)
return newaps,box

#CLEPS=Create List Elements and Positions Structure.
#noal=number of atoms. integer variable. This variable indicate the total number of atoms per layer.
#nols=number of layers structure. integer variable. This variable indicate the number of layers to stack.
#ael=atomic elements layer. string array dimensions [noal][1].
#apl=atomic positions layer. float array dimensions [noal][3].
def CLEPS(noal,nols,aes,aps):
    aeps=[]*(noal*nols+IL) #aeps=atomic elements and positions structure.
    for iaeps in range(0,noal*nols+IL,1): aeps[iaeps]=' '*4 #iaeps=index atomic elements and positions structure.
    aeps[0][0]=str(noal*nols)
    for ic in range(0,4,1): #ic=index column.
        if ic==0:
            for iL in range(IL,noal*nols+IL,1): aeps[iL][ic]=aes[iL-IL][0] #iL=index Lines.
        else:
            for iL in range(IL,noal*nols+IL,1): aeps[iL][ic]='%'+str(NoD)+'f' %(aps[iL-IL][ic-1]) #iL=index Lines.
    return aeps

print('Reading number of atoms in unit layer.')
NoA=RNLTF(FNR)-IL #NL=Number of Atoms.
print('Number of atoms in unit layer is',NoA)
print('Initiated reading of atomic positions and elements for unit layer.')
AEL=RDSTF(FNR,NoA+IL,1,IL,0) #APL=Atomic Elements Layer.
APL=RDFTF(FNR,NoA+IL,4,IL,IC) #APL=Atomic Positions Layer.
print('Completed reading of atomic positions and elements for unit layer.')
for iANumLay in range(MinNumLay,MaxNumLay,StpNumLay):
    SNL=iANumLay #SNL=Structure Number of Layers.
    print('*****')
    print('Initiated work on carbon structure of '+str(SNL)+' layers.')
    print('*****')
    print('Initiated creation on carbon structure of '+str(SNL)+' layers.')
    AES,APS=CrcCS(NoA,SNL,AEL,APL,GLS) #APS=Atomic Elements Structure. APS=Atomic Positions Structure.
    print('Completed creation on carbon structure of '+str(SNL)+' layers.')
    print('Initiated centering on carbon structure of '+str(SNL)+' layers.')
    APS,BxLim=CeCS(NoA*SNL,APS) #BxLim=Box Limits.
    print('Completed centering on carbon structure of '+str(SNL)+' layers.')
    print('Initiated written on carbon structure of '+str(SNL)+' layers.')
    AEPS=CLEPS(NoA,SNL,AES,APS) #AEPS=Atomic Elements and Positions Structure.
    WDSTF(PFNW+FNW+str(SNL)+' .xyz',NoA*SNL+IL,4,AEPS,BxLim)
    print('Complete written on carbon structure of '+str(SNL)+' layers.')
print('Job done. Have a nice day.')

```

Archivo lectura y escritura de resultados - Graphite_Layer_Stacking.py

```

from numpy import zeros

#RNLTF=Read Number Lines Text File.
#variables:
#fn=file name. string variable.
def RNLTF(fn):
    with open(fn) as of: #of=open file.
        for i, l in enumerate(of):
            pass
    return i+1

#RDSTF=Read Data as String form Text File.
#variables:
#fnr=file name to read. string variable.
#nl=number lines. integer variable. This variable indicate the number of lines to read. The total number of lines in the files is equal to nl+igl.
#nc=number columns. integer variable. This variable indicate the number of columns to read. The total number of columns in the files is equal to nc+igc.
#igl=ignored lines. integer variable. This variable indicate the number of lines to ignore. The total number of lines in the files is equal to nl+igl.
#igc=ignored lines. integer variable. This variable indicate the number of columns to ignore. The total number of columns in the files is equal to nc+igc.
def RDSTF(fn,nl,nc,igl,igc):
    rd=zeros(shape=(nl-igl,nc-igc),dtype=str) #rd=read data.
    of=open(fn,'r')#of=open file.
    al=of.readlines() #al=all lines.
    of.close()
    for il in range(igl,nl,1): #il=index line.
        sli=al[il].split() #sc=splited line i.
        for ic in range(igc,nc,1): #ic=index column.
            rd[il-igl][ic-igc]=sli[ic]
    return rd

#RDFTF=Read Data as Float form Text File.
#variables:
#fnr=file name to read. string variable.
#nl=number lines. integer variable. This variable indicate the number of lines to read. The total number of lines in the files is equal to nl+igl.
#nc=number columns. integer variable. This variable indicate the number of columns to read. The total number of columns in the files is equal to nc+igc.
#igl=ignored lines. integer variable. This variable indicate the number of lines to ignore. The total number of lines in the files is equal to nl+igl.
#igc=ignored lines. integer variable. This variable indicate the number of columns to ignore. The total number of columns in the files is equal to nc+igc.
def RDFTF(fn,nl,nc,igl,igc):
    rd=zeros(shape=(nl-igl,nc-igc),dtype=float) #rd=read data.
    of=open(fn,'r')#of=open file.
    al=of.readlines() #al=all lines.
    of.close()
    for il in range(igl,nl,1): #il=index line.

```

```

sli=al[il].split() #sc=splited line i.
for ic in range(igc,nc,1): #ic=index column.
    rd[il-ig1][ic-igc]=float(sli[ic])
return rd

#WDSTF=Write Data as String form Text File.
#variables:
#fnw=file name to write. string variable.
#nl=number lines. integer variable. This variable indicate the number of lines to write.
#nc=number columns. integer variable. This variable indicate the number of columns to write.
#db=data base. float array dimensions [nl][nc].
#bdb=box data base. float array dimensions [3].
def WDSTF(fnw,nl,nc,db,bdb):
    cf=open(fnw,'w') #cf=created file.
    for inl in range(0,nl,1): #inl=index number line.
        ltw=' ' #ltw=line to write.
        for inc in range(0,nc,1): #inc=index number column.
            ltw=ltw+str(db[inl][inc])+ ' '
        cf.write(ltw+'\n')
    ltwd='Dimension Box => '+x+'+str(bdb[0])+'\t'+y+'+str(bdb[1])+'\t'+z+'+str(bdb[2])+'\t'
    cf.write(ltwd+'\n')
    cf.close()
    return

```

.6. Anexo F: Código de Python3 para post-procesado de los resultados de DoS y PDoS

Archivo *input* para post-procesamiento - DOS_Ele_Grouping_Input.py

```

IgnLinFilXYZSta=2 #IgnLinFilXYZSta=Ignored Lines in File XYZ Start.
IgnLinFilXYZEnd=1 #IgnLinFilXYZEnd=Ignored Lines in File XYZ End.
IgnLinFilDoS=1 #IgnLinFilDoS=Ignored Lines in File DoS.

MasDatAna=False #MasDatAna=Massive Analysis.
AddSavFil='VDW' #AddSavFil=Addition to Save File.
SavLab='FD_36D-3' #SavLab=Save Label.
BasNamSavFil='CQD_SGN_R2H1_ST1_VDW-DF' #BasNamSav=Base Name Saved File.
BasNamQDFil='CQD_SGN_R2H1_ST1_VDW-DF' #BasNamQDFil=Base Name QD File.
DirAtmFilDOS='/home/jorge/UES_Cluster/Calculations_Outputs_LARCAD/CQD_Stacking_Graphite_Nitrogenized/CQD_SGN_R2H1_ST1_VDW-DF/CQD_SGN_R2H1_ST1_VDW-DF_DEL/
calculation_del_3_projwfc_plt_Simple-Gaussian_0.036749309_001000000000000000000_-6.4297_3.5703/' #Directory Atomic Files DOS.
DirStuFilXYZ='/home/jorge/UES_Cluster/Calculations_Inputs/CQD_Stacking_Graphite_Nitrogenized/CQD_SGN_R2H1_ST1_VDW-DF/' #DirStuFilXYZ=Directory Structure File XYZ.

DirSavPltDOS='./figures_dos/' #DirSavPltDOS=Directory Save Plots DOS.
DirSavFilDOS='./files_dos/' #DirSavFilDOS=Directory Save File DOS.

StuNamXYZ='CQD_SGN_R2H1_ST1.xyz' #StuNamXYZ=Structure Name XYZ.
StuLayNum=1 #StuLayNum=Structure Layers Number.
GenAtmNamDOS=BasNamQDFil+'.pdos_atm#' #GenAtmNamDOS=Generic Atomic Name DOS.
GenTotNamDOS=BasNamQDFil+'.pdos_tot' #GenTotNamDOS=Generic Total Name DOS.
GenNamProEleDOS=BasNamSavFil+'_'+AddSavFil #GenNamProEleDOS=Generic Name Processed Elemental DOS.

MaxDec=5 #MaxDec=Maximun Decimals.
MaxCha='U50' #Maximun Characters.
SavpLyDoS=True #SavpLyDoS=Save per Layer DOS.
SavIndDOS=False #SavIndDOS=Save Individual DOS.
SavGroDOS=True #SavGroDOS=Save Grouping DOS.
SavFigDOS=True #SavFigDOS=Save Figures DOS.
SavFilDOS=True #SavFilDOS=Save Files DOS.
ShwFilPro=True #ShwFilPro>Show File Processing.
ShwOnlTotDOS=False #ShwOnlTotDOS>Show Only Total DOS.
ShwFigDOS=False #ShwFigDOS>Show Figures DOS.

GenOrb=['s','p','d','f'] #GenOrb=Generic Orbitals.
GenOrbDim=[1,3,5,7] #GenOrbDim=Generic Orbitals Dimentions.
GenOrbDimLab=['s','pz','px','py','dz2','dxx','dzy','dx2-y2','dxy'] #GenOrbDimLab=Generic Orbitals Dimentions Labels.
AtmOrb=[['C',2],
        ['H',1],
        ['O',2],
        ['N',2],
        ['Cd',1],
        ['Se',2]
        ] #AtomOrb=Atomic Orbitals.

```

Archivo *input* para procesamiento masivo - DOS_Ele_Grouping_List.py

```

DifStcNum=3 #DifStcNum=Diference Structure Numbers.

DifStc_AddSavFil=['VDW','VDW','VDW'] #DifStc_AddSavFil=Diference Structure Addition to Save File.

DifStc_StuLayNum=[2,3,4] #DifStc_StuLayNum=Diferent Structure Structure Layers Number.

```

```

DifStc_StuNamXYZ=[
'CQD_SGN_R2H2_ST1_xyz',
'CQD_SGN_R2H3_ST1_xyz',
'CQD_SGN_R2H4_ST1_xyz',
]

DifStc_BasNamSavFil=[
'CQD_SGN_R2H2_ST1_VDW-DF',
'CQD_SGN_R2H3_ST1_VDW-DF',
'CQD_SGN_R2H4_ST1_VDW-DF',
] #DifStc_BasNamSavFil=Diferent Structure Base Name Saved File.

DifStc_BasNamQDFil=[
'CQD_SGN_R2H2_ST1_VDW-DF',
'CQD_SGN_R2H3_ST1_VDW-DF',
'CQD_SGN_R2H4_ST1_VDW-DF',
] #DifStc_BasNamQDFil=Diferent Structure Base Name QD File.

DifStc_GenAtmNamDOS=[] #GenAtmNamDOS=Diferent StructureGeneric Atomic Name DOS.
DifStc_GenTotNamDOS=[] #GenTotNamDOS=Diferent StructureGeneric Total Name DOS.
DifStc_GenNamProEleDOS=[] #GenNamProEleDOS=Diferent StructureGeneric Name Processed Elemental DOS.

for i_Stc in range(0,DifStcNum,1): #i_Stc=i-counter Structure.
    DifStc_GenAtmNamDOS.append(DifStc_BasNamQDFil[i_Stc]+'_pdos_atm#')
    DifStc_GenTotNamDOS.append(DifStc_BasNamQDFil[i_Stc]+'_pdos_tot')
    DifStc_GenNamProEleDOS.append(DifStc_BasNamSavFil[i_Stc]+'_'+DifStc_AddSavFil[i_Stc])

Difstc_DirStuFilXYZ=[
'/home/jorge/UES_Cluster/Calculations_Inputs/CQD_Stacking_Graphite_Nitrogenized/CQD_SGN_R2H2_ST1_VDW-DF/',
'/home/jorge/UES_Cluster/Calculations_Inputs/CQD_Stacking_Graphite_Nitrogenized/CQD_SGN_R2H3_ST1_VDW-DF/',
'/home/jorge/UES_Cluster/Calculations_Inputs/CQD_Stacking_Graphite_Nitrogenized/CQD_SGN_R2H4_ST1_VDW-DF/',
]

DifStc_DirAtmFoldDOS=[
'/home/jorge/UES_Cluster/Calculations_Outputs_LARCAD/CQD_Stacking_Graphite_Nitrogenized/CQD_SGN_R2H2_ST1_VDW-DF/CQD_SGN_R2H2_ST1_VDW-DF_DEL/',
'/home/jorge/UES_Cluster/Calculations_Outputs_LARCAD/CQD_Stacking_Graphite_Nitrogenized/CQD_SGN_R2H3_ST1_VDW-DF/CQD_SGN_R2H3_ST1_VDW-DF_DEL/',
'/home/jorge/UES_Cluster/Calculations_Outputs_LARCAD/CQD_Stacking_Graphite_Nitrogenized/CQD_SGN_R2H4_ST1_VDW-DF/CQD_SGN_R2H4_ST1_VDW-DF_DEL/',
] #DifStc_DirAtmFoldDOS=Diferent Structure Directory Atomic Folder DOS.

DifStc_DirAtmFilDOS=[
[
['calculation_del_3_projwfc_plt_Fermi-Dirac_0.00000000_00100000000000000000_-6.5031_3.4969','FD_OD-3'],
['calculation_del_3_projwfc_plt_Fermi-Dirac_0.008819834_00100000000000000000_-6.5031_3.4969','FD_8D-3'],
['calculation_del_3_projwfc_plt_Fermi-Dirac_0.018374654_00100000000000000000_-6.5031_3.4969','FD_18D-3'],
['calculation_del_3_projwfc_plt_Fermi-Dirac_0.036749309_00100000000000000000_-6.5031_3.4969','FD_36D-3'],
['calculation_del_3_projwfc_plt_Marzari-Vanderbilt_0.00000000_00100000000000000000_-6.5031_3.4969','MV_OD-3'],
['calculation_del_3_projwfc_plt_Marzari-Vanderbilt_0.008819834_00100000000000000000_-6.5031_3.4969','MV_8D-3'],
['calculation_del_3_projwfc_plt_Marzari-Vanderbilt_0.018374654_00100000000000000000_-6.5031_3.4969','MV_18D-3'],
['calculation_del_3_projwfc_plt_Marzari-Vanderbilt_0.036749309_00100000000000000000_-6.5031_3.4969','MV_36D-3'],
['calculation_del_3_projwfc_plt_Marzari-Vanderbilt_0.073498618_00100000000000000000_-6.5031_3.4969','MV_73D-3'],
['calculation_del_3_projwfc_plt_Methfessel-Paxton_0.00000000_00100000000000000000_-6.5031_3.4969','MP_OD-3'],
['calculation_del_3_projwfc_plt_Methfessel-Paxton_0.008819834_00100000000000000000_-6.5031_3.4969','MP_8D-3'],
['calculation_del_3_projwfc_plt_Methfessel-Paxton_0.073498618_00100000000000000000_-6.5031_3.4969','MP_73D-3'],
['calculation_del_3_projwfc_plt_Simple-Gaussian_0.00000000_00100000000000000000_-6.5031_3.4969','SG_OD-3'],
['calculation_del_3_projwfc_plt_Simple-Gaussian_0.008819834_00100000000000000000_-6.5031_3.4969','SG_8D-3'],
['calculation_del_3_projwfc_plt_Simple-Gaussian_0.018374654_00100000000000000000_-6.5031_3.4969','SG_18D-3'],
['calculation_del_3_projwfc_plt_Simple-Gaussian_0.036749309_00100000000000000000_-6.5031_3.4969','SG_36D-3'],
['calculation_del_3_projwfc_plt_Simple-Gaussian_0.073498618_00100000000000000000_-6.5031_3.4969','SG_73D-3']
],
[
['calculation_del_3_projwfc_plt_Fermi-Dirac_0.00001000_00100000000000000000_-6.3876_3.6124','FD_OD-3'],
['calculation_del_3_projwfc_plt_Fermi-Dirac_0.008819834_00100000000000000000_-6.3876_3.6124','FD_8D-3'],
['calculation_del_3_projwfc_plt_Fermi-Dirac_0.036749309_00100000000000000000_-6.3876_3.6124','FD_36D-3'],
['calculation_del_3_projwfc_plt_Marzari-Vanderbilt_0.00001000_00100000000000000000_-6.3876_3.6124','MV_OD-3'],
['calculation_del_3_projwfc_plt_Marzari-Vanderbilt_0.008819834_00100000000000000000_-6.3876_3.6124','MV_8D-3'],
['calculation_del_3_projwfc_plt_Marzari-Vanderbilt_0.018374654_00100000000000000000_-6.3876_3.6124','MV_18D-3'],
['calculation_del_3_projwfc_plt_Marzari-Vanderbilt_0.036749309_00100000000000000000_-6.3876_3.6124','MV_36D-3'],
['calculation_del_3_projwfc_plt_Marzari-Vanderbilt_0.073498618_00100000000000000000_-6.3876_3.6124','MV_73D-3'],
['calculation_del_3_projwfc_plt_Methfessel-Paxton_0.00001000_00100000000000000000_-6.3876_3.6124','MP_OD-3'],
['calculation_del_3_projwfc_plt_Methfessel-Paxton_0.018374654_00100000000000000000_-6.3876_3.6124','MP_18D-3'],
['calculation_del_3_projwfc_plt_Methfessel-Paxton_0.036749309_00100000000000000000_-6.3876_3.6124','MP_36D-3'],
['calculation_del_3_projwfc_plt_Simple-Gaussian_0.00001000_00100000000000000000_-6.3876_3.6124','SG_OD-3'],
['calculation_del_3_projwfc_plt_Simple-Gaussian_0.018374654_00100000000000000000_-6.3876_3.6124','SG_18D-3'],
['calculation_del_3_projwfc_plt_Simple-Gaussian_0.036749309_00100000000000000000_-6.3876_3.6124','SG_36D-3'],
['calculation_del_3_projwfc_plt_Simple-Gaussian_0.073498618_00100000000000000000_-6.3876_3.6124','SG_73D-3']
],
[
['calculation_del_3_projwfc_plt_Fermi-Dirac_0.00000000_00100000000000000000_-6.3876_3.6124','FD_OD-3'],
['calculation_del_3_projwfc_plt_Fermi-Dirac_0.008819834_00100000000000000000_-6.3876_3.6124','FD_8D-3'],
['calculation_del_3_projwfc_plt_Fermi-Dirac_0.018374654_00100000000000000000_-6.3876_3.6124','FD_18D-3'],
['calculation_del_3_projwfc_plt_Fermi-Dirac_0.036749309_00100000000000000000_-6.3876_3.6124','FD_36D-3'],
['calculation_del_3_projwfc_plt_Fermi-Dirac_0.073498618_00100000000000000000_-6.3876_3.6124','FD_73D-3'],
['calculation_del_3_projwfc_plt_Marzari-Vanderbilt_0.00000000_00100000000000000000_-6.3876_3.6124','MV_OD-3'],
['calculation_del_3_projwfc_plt_Marzari-Vanderbilt_0.008819834_00100000000000000000_-6.3876_3.6124','MV_8D-3'],
['calculation_del_3_projwfc_plt_Marzari-Vanderbilt_0.018374654_00100000000000000000_-6.3876_3.6124','MV_18D-3'],
['calculation_del_3_projwfc_plt_Marzari-Vanderbilt_0.036749309_00100000000000000000_-6.3876_3.6124','MV_36D-3'],
['calculation_del_3_projwfc_plt_Marzari-Vanderbilt_0.073498618_00100000000000000000_-6.3876_3.6124','MV_73D-3'],
['calculation_del_3_projwfc_plt_Methfessel-Paxton_0.00000000_00100000000000000000_-6.3876_3.6124','MP_OD-3'],
['calculation_del_3_projwfc_plt_Methfessel-Paxton_0.008819834_00100000000000000000_-6.3876_3.6124','MP_8D-3'],
['calculation_del_3_projwfc_plt_Methfessel-Paxton_0.018374654_00100000000000000000_-6.3876_3.6124','MP_18D-3'],
['calculation_del_3_projwfc_plt_Methfessel-Paxton_0.036749309_00100000000000000000_-6.3876_3.6124','MP_36D-3'],
['calculation_del_3_projwfc_plt_Methfessel-Paxton_0.073498618_00100000000000000000_-6.3876_3.6124','MP_73D-3'],
['calculation_del_3_projwfc_plt_Simple-Gaussian_0.00000000_00100000000000000000_-6.3876_3.6124','SG_OD-3'],
['calculation_del_3_projwfc_plt_Simple-Gaussian_0.008819834_00100000000000000000_-6.3876_3.6124','SG_8D-3'],
['calculation_del_3_projwfc_plt_Simple-Gaussian_0.036749309_00100000000000000000_-6.3876_3.6124','SG_36D-3'],
['calculation_del_3_projwfc_plt_Simple-Gaussian_0.073498618_00100000000000000000_-6.3876_3.6124','SG_73D-3']
]
] #DifStc_DirAtmFilDOS=Diferent Structure Directory Atomic Files DOS.

```

Archivo principal - DOS_Ele_Grouping_Core.py

```
import DOS_Ele_Grouping_Input as Inp #Lst=List.
import DOS_Ele_Grouping_List as Lst #Lst=List.
import DOS_Ele_Grouping_RaW_TXT as RaW #RaW=Read and Write.

from numpy import zeros
from matplotlib.pyplot import close,figure,legend,plot,savefig,show,xlabel,ylabel,title

#DetStuEle=Determine Structure Elements.
#Variables:
#num_atm_stu=number atoms structure. integer variable. This variable give the number of atoms in the structure.
#ind_ele_num=index elements number. integer array of dimensions [num_atm_stu]. This variables give the Simbol for each atom in the structure.
def DetStuEle(num_atm_stu,ind_ele_num):
    num_atm_ele=0 #num_atm_ele=number atoms elements.
    uni_atm_ele=[] #uni_atm_ele=unique atoms elements.
    for i_atm_stu in range(0,num_atm_stu,1): #i_atm_stu=i-index atoms structure.
        uni_atm_ele_con=True #uni_atm_ele_con=unique atom element condition.
        for i_uni_atm_stu in range(0,num_atm_ele,1): #i_uni_atm_stu=i-index unique atoms structure.
            if ind_ele_num[i_atm_stu]==uni_atm_ele[i_uni_atm_stu]:
                uni_atm_ele_con=False
                break
        if uni_atm_ele_con==True:
            num_atm_ele=num_atm_ele+1
            uni_atm_ele.append(ind_ele_num[i_atm_stu])
    return num_atm_ele

#GenAtmFilNamDOS=Generate Atomic File Names DOS.
#Variables:
#num_atm_stu=number atoms structure. integer variable. This variable give the number of atoms in the structure.
#ind_ele_num=index elements number. integer array of dimensions [num_atm_stu]. This variables give the Simbol for each atom in the structure.
def GenAtmFilNamDOS(num_atm_stu,ind_ele_num):
    all_ele_atm_fil_nam_dos=[] #all_ele_atm_fil_nam_dos=all element atomic file names dos.
    all_atm_fil_nam_dos=[] #all_atm_fil_nam_dos=all atomic files names dos.
    all_atm_fil_orb_dos=[] #all_atm_fil_orb_dos=all atomic files orbital dos.
    num_atm_fil_nam_dos=0 #num_atm_fil_nam_dos=number atomic file names dos.
    num_atm_ele_dat_bas=len(Inp.AtmOrb) #num_atm_dat_bas=number atom elements data base
    for iAtmStr in range(0,num_atm_stu,1): #iAtmStr=index Atom in Structure.
        ActNumOrb=0 #ActNumOrb=Actual Number of Orbitals.
        for iAtmEle in range(0,num_atm_ele_dat_bas,1):
            if Inp.AtmOrb[iAtmEle][0]==ind_ele_num[iAtmStr]:
                ActNumOrb=Inp.AtmOrb[iAtmEle][1]
                break
        if ActNumOrb==0:
            raise SystemExit('The element \''+ind_ele_num[iAtmStr]+'\' was not found in the database. Please add this element and then repeat the grouping process')
        for iActNumOrb in range(0,ActNumOrb,1):
            act_atm_fil_nam_dos=Inp.GenAtmNamDOS+str(iAtmStr+1)+'('+ind_ele_num[iAtmStr]+'')+'#'+str(iActNumOrb+1)+'('+Inp.GenOrb[iActNumOrb]+'')
            #ActAtmFilNamDOS=Actual Atomic File Name DOS.
            all_ele_atm_fil_nam_dos.append(ind_ele_num[iAtmStr])
            all_atm_fil_nam_dos.append(act_atm_fil_nam_dos)
            all_atm_fil_orb_dos.append(Inp.GenOrbDim[iActNumOrb])
            num_atm_fil_nam_dos=num_atm_fil_nam_dos+1
    all_ele_atm_fil_nam_dos_arr=zeros((num_atm_fil_nam_dos,),dtype=str) #all_ele_atm_fil_nam_dos_arr=all element atomic file names dos array.
    all_atm_fil_nam_dos_arr=zeros((num_atm_fil_nam_dos,),dtype=Inp.MaxCha) #all_atm_fil_nam_dos_arr=all atomic files names dos array.
    all_atm_fil_orb_dos_arr=zeros((num_atm_fil_nam_dos,),dtype=int) #all_atm_fil_orb_dos_arr=all atomic files orbital dos array.
    for inum_atm_fil_nam_dos in range(0,num_atm_fil_nam_dos,1):
        all_ele_atm_fil_nam_dos_arr[inum_atm_fil_nam_dos]=all_ele_atm_fil_nam_dos[inum_atm_fil_nam_dos]
        all_atm_fil_nam_dos_arr[inum_atm_fil_nam_dos]=all_atm_fil_nam_dos[inum_atm_fil_nam_dos]
        all_atm_fil_orb_dos_arr[inum_atm_fil_nam_dos]=all_atm_fil_orb_dos[inum_atm_fil_nam_dos]
    return num_atm_fil_nam_dos,all_ele_atm_fil_nam_dos_arr,all_atm_fil_nam_dos_arr,all_atm_fil_orb_dos_arr

#SumEleDOS=Summation Elemental DOS.
#variables:
#num_atm_ele=number atoms elements. interger variable. This variable give the number of unique elements in the structure.
#num_stp_ene_grd=number steps energy grid. integer variable. This variable indicates the grid values in the energy axis.
#num_atm_fil_nam_dos=number atomic file names dos. integer variable. This variable indicates the number of atomic dos to be read.
#all_atm_fil_lab_dos=all atomic file label dos. string array of dimensions [num_atm_fil_nam_dos]. This variable has the file label of all the atomic dos to be read.
#all_atm_fil_nam_dos=all atomic file names dos. string array of dimensions [num_atm_fil_nam_dos]. This variable has the file names of all the atomic dos to be read.
#all_atm_fil_orb_dos=all atomic files orbital dos. integer array of dimensions [num_atm_fil_nam_dos]. This variable has the number of orbitals of all the atomic dos to be read.
def SumEleDOS(num_atm_ele,num_stp_ene_grd,num_atm_fil_nam_dos,all_atm_fil_lab_dos,all_atm_fil_nam_dos,all_atm_fil_orb_dos):
    mar_ele_dos=zeros(shape=(num_atm_ele+1,),dtype=int) #mar_ele_dos=mark element dos.
    tot_num_orb=0 #tot_num_orb=total number orbitals.
    for i_num_atm_ele in range(0,num_atm_ele+1,1): #i_num_atm_ele=index number atomic element.
        mar_ele_dos[i_num_atm_ele]=tot_num_orb
        if i_num_atm_ele<num_atm_ele:
            for i_num_orb_atm_ele in range(0,Inp.AtmOrb[i_num_atm_ele][1],1): #i_num_orb_atm_ele=index number orbitals atomic element.
                tot_num_orb=tot_num_orb+Inp.GenOrbDim[i_num_orb_atm_ele]
            tot_num_orb=tot_num_orb+1
    ene_grd=zeros(shape=(num_stp_ene_grd,),dtype=float) #ene_grd=energy grid.
    tot_stc_dos=zeros(shape=(num_stp_ene_grd,),dtype=float) #tot_stc_dos=total structure dos.
    tot_ele_dos=zeros(shape=(tot_num_orb+1,num_stp_ene_grd),dtype=float) #tot_ele_dos=total elemental dos.
    tot_lay_dos=zeros(shape=(Inp.StuLayNum+1,num_stp_ene_grd),dtype=float) #tot_lay_dos=total layer dos.
    lay_ele_dos=zeros(shape=(Inp.StuLayNum,tot_num_orb+1,num_stp_ene_grd),dtype=float) #lay_ele_dos=layer elemental dos.
    act_atm_dos=RaW.ReadatFloTxFil(Inp.DirAtmFilDOS+Inp.GenTotNamDOS,num_stp_ene_grd,3,1,num_stp_ene_grd+Inp.IgnLinFilDoS,1,0,3,1) #act_atm_dos=actual atomic density of states.
    for i_num_stp_ene_grd in range(0,num_stp_ene_grd,1): #i_num_stp_ene_grd=index number steps energy grid.
        ene_grd[i_num_stp_ene_grd]=act_atm_dos[i_num_stp_ene_grd][0]
        tot_stc_dos[i_num_stp_ene_grd]=act_atm_dos[i_num_stp_ene_grd][2]
        tot_lay_dos[Inp.StuLayNum-1][i_num_stp_ene_grd]=act_atm_dos[i_num_stp_ene_grd][2]
    i_stc_lay=0 #i_stc_lay=i-counter structure.
    j_num_atm_fil_nam_dos=0 #j_num_atm_fil_nam_dos=j-counter number atomic file names dos.
    num_atm_fil_lay=int(num_atm_fil_nam_dos/Inp.StuLayNum) #num_atm_fil_lay=number atomic files per layer.
    for i_num_atm_fil_nam_dos in range(0,num_atm_fil_nam_dos,1): #i_num_atm_fil_nam_dos=index number atomic file names dos.
        num_stp_ene_grd_act=RaW.ReaNuLinTxFil(Inp.DirAtmFilDOS+all_atm_fil_nam_dos[i_num_atm_fil_nam_dos])-Inp.IgnLinFilDoS #num_stp_ene_grd_act=number step energy grid actual.
```

```

ene_grd_act=Raw.ReadatFloTxFil(Inp.DirAtmFilDOS+all_atm_fil_nam_dos[i_num_atm_fil_nam_dos],num_stp_ene_grd_act,1,1,num_stp_ene_grd_act+Inp.IgnLinFilDoS,1,0,1,1)
#act_atm_dos_pre_rea=actual atoms dos pre read
act_atm_dos_pre_rea=Raw.ReadatFloTxFil(Inp.DirAtmFilDOS+all_atm_fil_nam_dos[i_num_atm_fil_nam_dos],num_stp_ene_grd_act,all_atm_fil_orb_dos[i_num_atm_fil_nam_dos],1,
num_stp_ene_grd_act+Inp.IgnLinFilDoS,1,2,all_atm_fil_orb_dos[i_num_atm_fil_nam_dos]+2,1) #act_atm_dos_pre_rea=actual atoms dos pre read.
i_num_stp_ene_grd=0 #i_num_stp_ene_grd=i-counter number energy grid.
act_atm_dos=zeros(shape=(num_stp_ene_grd,all_atm_fil_orb_dos[i_num_atm_fil_nam_dos]),dtype=float,order='F')
for i_num_stp_ene_grd_act in range(0,num_stp_ene_grd_act,1): #i_num_stp_ene_grd_act=i-counter number step energy grid actual.
    while i_num_stp_ene_grd<num_stp_ene_grd:
        if ene_grd[i_num_stp_ene_grd]==ene_grd_act[i_num_stp_ene_grd_act]:
            act_atm_dos[i_num_stp_ene_grd]=act_atm_dos_pre_rea[i_num_stp_ene_grd_act]
            break
        else:
            i_num_stp_ene_grd=i_num_stp_ene_grd+1
act_atm_ele=0 #act_atm_ele=actual atomic element.
for i_act_atm_ele in range(0,num_atm_ele,1): #i_act_atm_ele=index actual atomic element.
    if all_atm_fil_lab_dos[i_num_atm_fil_nam_dos]==Inp.AtmOrb[i_act_atm_ele][0]:
        act_atm_ele=i_act_atm_ele
        break
if Inp.ShwFilPro==True: print('\t'+>>>Reading and summing '+str(all_atm_fil_orb_dos[i_num_atm_fil_nam_dos])+'+ '+Inp.AtmOrb[act_atm_ele][0]+' orbitals from file
'+all_atm_fil_nam_dos[i_num_atm_fil_nam_dos])
orb_pos=1 #orb_pos=orbital position.
for i_GenOrbDim in range(0,Inp.AtmOrb[act_atm_ele][1],1):
    if Inp.GenOrbDim[i_GenOrbDim]<all_atm_fil_orb_dos[i_num_atm_fil_nam_dos]: orb_pos=orb_pos+Inp.GenOrbDim[i_GenOrbDim]
for i_num_orb_ele in range(0,all_atm_fil_orb_dos[i_num_atm_fil_nam_dos],1): #i_num_orb_ele=index number orbital element.
    for i_num_stp_ene_grd in range(0,num_stp_ene_grd,1): #i_num_stp_ene_grd=index number steps energy grid.
        tot_ele_dos[mar_ele_dos[num_atm_ele]][i_num_stp_ene_grd]=tot_ele_dos[mar_ele_dos[num_atm_ele]][i_num_stp_ene_grd]+act_atm_dos[i_num_stp_ene_grd][i_num_orb_ele]
        tot_ele_dos[mar_ele_dos[act_atm_ele]][i_num_stp_ene_grd]=tot_ele_dos[mar_ele_dos[act_atm_ele]][i_num_stp_ene_grd]+act_atm_dos[i_num_stp_ene_grd][i_num_orb_ele]
        tot_ele_dos[mar_ele_dos[act_atm_ele]+orb_pos+i_num_orb_ele][i_num_stp_ene_grd]=tot_ele_dos[mar_ele_dos[act_atm_ele]+orb_pos+i_num_orb_ele][i_num_stp_ene_grd]
        +act_atm_dos[i_num_stp_ene_grd][i_num_orb_ele]
        lay_ele_dos[i_stc_lay][mar_ele_dos[num_atm_ele]][i_num_stp_ene_grd]=lay_ele_dos[i_stc_lay][mar_ele_dos[num_atm_ele]][i_num_stp_ene_grd]
        +act_atm_dos[i_num_stp_ene_grd][i_num_orb_ele]
        lay_ele_dos[i_stc_lay][mar_ele_dos[act_atm_ele]][i_num_stp_ene_grd]=lay_ele_dos[i_stc_lay][mar_ele_dos[act_atm_ele]][i_num_stp_ene_grd]
        +act_atm_dos[i_num_stp_ene_grd][i_num_orb_ele]
        lay_ele_dos[i_stc_lay][mar_ele_dos[act_atm_ele]+orb_pos+i_num_orb_ele][i_num_stp_ene_grd]=lay_ele_dos[i_stc_lay][mar_ele_dos[act_atm_ele]
        +orb_pos+i_num_orb_ele][i_num_stp_ene_grd]+act_atm_dos[i_num_stp_ene_grd][i_num_orb_ele]
    if j_num_atm_fil_nam_dos==num_atm_fil_lay-1:
        i_stc_lay=i_stc_lay+1
        j_num_atm_fil_nam_dos=0
    else: j_num_atm_fil_nam_dos=j_num_atm_fil_nam_dos+1
if Inp.ShwFilPro==True: print('')
return tot_num_orb,mar_ele_dos,ene_grd,tot_stc_dos,tot_ele_dos,tot_lay_dos,lay_ele_dos

#PltDOS=Plotting DOS.
#variables:
#num_atm_ele=number atoms elements. interger variable. This variable give the number of unique elements in the structure.
#mar_ele_dos=mark element dos. integer array of dimension [tot_num_orb+1]. This variable indicates a marker integer points to differentiate the orbitals for each
element in the array tot_ele_dos.
#ene_grd=energy grid. float array of dimension [num_stp_ene_grd]. This variable contains the points in the energy grid.
#tot_ele_dos=total elemental dos. float array of dimension [num_atm_ele+1][num_stp_ene_grd]. This variable contains the DOS of each individual element and the
superposition on all the atoms present in the structure.
#qe_tot_dos=quantum espresso total dos. float array of dimension [num_stp_ene_grd]. This variable contains the DOS superposition on all the atoms present in the
structure given by Quantum Espresso.
def PltDOS(num_atm_ele,mar_ele_dos,ene_grd,tot_ele_dos,qe_tot_dos):
    for i_num_atm_ele in range(0,num_atm_ele,1): #i_num_atm_ele=index number atomic elements.
        plt_tit=Inp.StuNamXYZ+'_'+Inp.SavLab #plt_tit=plot title.
        if i_num_atm_ele<num_atm_ele: plt_tit=plt_tit+'_PDOS_for_'+Inp.AtmOrb[i_num_atm_ele][0]
        else:
            plt_tit=plt_tit+'_PDOS_total'
            plot(ene_grd,qe_tot_dos,label='QE_'+plt_tit,color='r')
            plot(ene_grd,tot_ele_dos[mar_ele_dos[i_num_atm_ele]],label=plt_tit,color='b')
            xlabel('energy')
            ylabel('PDOS')
            title(plt_tit)
            legend(loc=1)
        if Inp.SavFigDOS==True: savefig(Inp.DirSavPltDOS+plt_tit+'.png') #,bbox_inches='tight'
        if Inp.ShwOnlTotDOS==True:
            if i_num_atm_ele==num_atm_ele: show()
        elif Inp.ShwFigDOS==True: show()
        close()
    return

#SavInvOrbDOS=Saving Individual Orbitals DOS.
#variables:
#num_atm_ele=number atoms elements. interger variable. This variable give the number of unique elements in the structure.
#tot_num_orb=total number orbitals. integer variable. This variable contains the number of orbitals in each atom plus the sum of said orbitals.
#mar_ele_dos=mark element dos. integer array of dimension [tot_num_orb+1]. This variable indicates a marker integer points to differentiate the orbitals for each
element in the array tot_ele_dos.
#num_stp_ene_grd=number steps energy grid. integer variable. This variable indicates the grid values in the energy axis.
#ene_grd=energy grid. float array of dimension [num_stp_ene_grd]. This variable contains the points in the energy grid.
#tot_ele_dos=total elemental dos. float array of dimension [tot_num_orb+1][num_stp_ene_grd]. This variable contains the DOS of each individual element and the
superposition on all the atoms present in the structure.
def SavInvOrbDOS(num_atm_ele,tot_num_orb,mar_ele_dos,num_stp_ene_grd,ene_grd,tot_ele_dos):
    for i_num_atm_ele in range(0,num_atm_ele+1,1): #i_num_atm_ele=index number atomic element.
        sta_orb=mar_ele_dos[i_num_atm_ele] #sta_orb=start orbitals.
        end_orb=0 #end_orb=end orbitals.
        if i_num_atm_ele==num_atm_ele: end_orb=sta_orb+1
        else: end_orb=mar_ele_dos[i_num_atm_ele+1]
        for i_num_orb_ele in range(sta_orb,end_orb,1): #i_num_orb_ele=index number orbital element.
            fil_tit='' #fil_tit=file title.
            if i_num_atm_ele<num_atm_ele: fil_tit=Inp.GenNamProEleDOS+'_'+Inp.AtmOrb[i_num_atm_ele][0]
            else: fil_tit=Inp.GenNamProEleDOS+'_Total'
            if i_num_orb_ele==sta_orb: fil_tit=fil_tit+'_Total' #and i_num_atm_ele<num_atm_ele
            else: fil_tit=fil_tit+'_'+Inp.GenOrbDimLab[i_num_orb_ele-sta_orb-1]
            if Inp.ShwFilPro==True: print('\t'+>>>Saving results in '+fil_tit)
            fil_dos=zeros((num_stp_ene_grd,2),dtype=float) #fil_dos=file dos.
            for i_num_stp_ene_grd in range(0,num_stp_ene_grd,1): #i_num_stp_ene_grd=index number steps energy grid.
                fil_dos[i_num_stp_ene_grd][0]=ene_grd[i_num_stp_ene_grd] #(" "+str(dec)+"f "%
                fil_dos[i_num_stp_ene_grd][1]=tot_ele_dos[i_num_orb_ele][i_num_stp_ene_grd]
            Raw.WriDatStrTxFil(Inp.DirSavFilDOS+fil_tit,num_stp_ene_grd,2,fil_dos,Inp.MaxDec)
            print('')

```



```

return

#SavGroOrbLayDOS=Saving Group Orbitals Layers DOS.
#variables:
#lay_act=actual layer. integer variable. This variable indicates the actual layer been saved.
#num_atm_ele=number atoms elements. integer variable. This variable give the number of unique elements in the structure.
#tot_num_orb=total number orbitals. integer variable. This variable contains the number of orbitals in each atom plus the sum of said orbitals.
#mar_ele_dos=mark element dos. integer array of dimension [tot_num_orb+1]. This variable indicates a marker integer points to differentiate the orbitals for each
element in the array tot_ele_dos.
#num_stp_ene_grd=number steps energy grid. integer variable. This variable indicates the grid values in the energy axis.
#ene_grd=energy grid. float array of dimension [num_stp_ene_grd]. This variable contains the points in the energy grid.
#lay_ele_dos=layer elemental dos. float array of dimension [tot_num_orb+1][num_stp_ene_grd]. This variable contains the DOS of each individual element and the
superposition on all the atoms present in the structure.
def SavGroOrbLayDOS(lay_act,num_atm_ele,tot_num_orb,mar_ele_dos,num_stp_ene_grd,ene_grd,lay_ele_dos):
    fil_dos=zeros((num_stp_ene_grd,mar_ele_dos[num_atm_ele]+2),dtype=float) #fil_dos=file dos.
    for i_num_stp_ene_grd in range(0,num_stp_ene_grd,1): #i_num_stp_ene_grd=index number steps energy grid.
        fil_dos[i_num_stp_ene_grd][0]=ene_grd[i_num_stp_ene_grd]
    fil_tit=Inp.BasNamSavFil+'_Lay_'+str(lay_act)+'_Ele_PDoS'+'+'+Inp.SavLab+'_'+'.dat' #fil_tit=file title.
    fil_heas='Energy' #fil_heas=file header.
    fil_Uni='eV'
    i_orb_ele_col=0 #i_orb_ele_col=i-index orbital element column.
    for i_num_atm_ele in range(0,num_atm_ele+1,1): #i_num_atm_ele=index number atomic element.
        sta_orb=mar_ele_dos[i_num_atm_ele] #sta_orb=start orbitals.
        end_orb=0 #end_orb=end orbitals.
        if i_num_atm_ele==num_atm_ele: end_orb=sta_orb+1
        else: end_orb=mar_ele_dos[i_num_atm_ele+1]
        for i_num_orb_ele in range(sta_orb,end_orb,1): #i_num_orb_ele=index number orbital element.
            fil_Uni=fil_Uni+' '+a.u.'
            if i_num_atm_ele<num_atm_ele: fil_heas=fil_heas+' '+Inp.AtmOrb[i_num_atm_ele][0]
            else: fil_heas=fil_heas+' '+Inp.BasNamSavFil
            if i_num_orb_ele==sta_orb: fil_heas=fil_heas+'_Total' #and i_num_atm_ele<num_atm_ele
            else: fil_heas=fil_heas+' '+Inp.GenOrbDimLab[i_num_orb_ele-sta_orb-1]
            for i_num_stp_ene_grd in range(0,num_stp_ene_grd,1): #i_num_stp_ene_grd=index number steps energy grid.
                fil_dos[i_num_stp_ene_grd][i_orb_ele_col+1]=lay_ele_dos[i_num_orb_ele][i_num_stp_ene_grd]
            i_orb_ele_col=i_orb_ele_col+1
    if Inp.ShwFilPro==True: print('\t'+>>>Saving results in '+fil_tit)
    RaW.WriDatStrTxtFilHea(Inp.DirSavFilDOS+fil_tit,num_stp_ene_grd,mar_ele_dos[num_atm_ele]+2,fil_dos,Inp.MaxDec,fil_heas+'\n'+fil_Uni)
    print('')
    return

#SavGroOrbStcDOS=Saving Group Orbitals Structure DOS.
#variables:
#num_atm_ele=number atoms elements. interger variable. This variable give the number of unique elements in the structure.
#tot_num_orb=total number orbitals. integer variable. This variable contains the number of orbitals in each atom plus the sum of said orbitals.
#mar_ele_dos=mark element dos. integer array of dimension [tot_num_orb+1]. This variable indicates a marker integer points to differentiate the orbitals for each
element in the array tot_ele_dos.
#num_stp_ene_grd=number steps energy grid. integer variable. This variable indicates the grid values in the energy axis.
#ene_grd=energy grid. float array of dimension [num_stp_ene_grd]. This variable contains the points in the energy grid.
#tot_ele_dos=total elemental dos. float array of dimension [tot_num_orb+1][num_stp_ene_grd]. This variable contains the DOS of each individual element and the
superposition on all the atoms present in the structure.
def SavGroOrbStcDOS(num_atm_ele,tot_num_orb,mar_ele_dos,num_stp_ene_grd,ene_grd,tot_ele_dos):
    fil_dos=zeros((num_stp_ene_grd,mar_ele_dos[num_atm_ele]+2),dtype=float) #fil_dos=file dos.
    for i_num_stp_ene_grd in range(0,num_stp_ene_grd,1): #i_num_stp_ene_grd=index number steps energy grid.
        fil_dos[i_num_stp_ene_grd][0]=ene_grd[i_num_stp_ene_grd]
    fil_tit=Inp.BasNamSavFil+'_Ele_PDoS'+'+'+Inp.SavLab+'_'+'.dat' #fil_tit=file title.
    fil_heas='Energy' #fil_heas=file header.
    fil_Uni='eV'
    i_orb_ele_col=0 #i_orb_ele_col=i-index orbital element column.
    for i_num_atm_ele in range(0,num_atm_ele+1,1): #i_num_atm_ele=index number atomic element.
        sta_orb=mar_ele_dos[i_num_atm_ele] #sta_orb=start orbitals.
        end_orb=0 #end_orb=end orbitals.
        if i_num_atm_ele==num_atm_ele: end_orb=sta_orb+1
        else: end_orb=mar_ele_dos[i_num_atm_ele+1]
        for i_num_orb_ele in range(sta_orb,end_orb,1): #i_num_orb_ele=index number orbital element.
            fil_Uni=fil_Uni+' '+a.u.'
            if i_num_atm_ele<num_atm_ele: fil_heas=fil_heas+' '+Inp.AtmOrb[i_num_atm_ele][0]
            else: fil_heas=fil_heas+' '+Inp.BasNamSavFil
            if i_num_orb_ele==sta_orb: fil_heas=fil_heas+'_Total' #and i_num_atm_ele<num_atm_ele
            else: fil_heas=fil_heas+' '+Inp.GenOrbDimLab[i_num_orb_ele-sta_orb-1]
            for i_num_stp_ene_grd in range(0,num_stp_ene_grd,1): #i_num_stp_ene_grd=index number steps energy grid.
                fil_dos[i_num_stp_ene_grd][i_orb_ele_col+1]=tot_ele_dos[i_num_orb_ele][i_num_stp_ene_grd]
            i_orb_ele_col=i_orb_ele_col+1
    if Inp.ShwFilPro==True: print('\t'+>>>Saving results in '+fil_tit)
    RaW.WriDatStrTxtFilHea(Inp.DirSavFilDOS+fil_tit,num_stp_ene_grd,mar_ele_dos[num_atm_ele]+2,fil_dos,Inp.MaxDec,fil_heas+'\n'+fil_Uni)
    print('')
    return

#GroExe=Grouping Execute.
#This function will execute grouping of DOS.
def GroExe():
    print('>Working on '+Inp.StuNamXYZ+' '+Inp.SavLab+' will shortly beging...'+'\n')
    print('>The choosen directory to read the atomic DoS files is: '+Inp.DirAtmFilDOS+'\n')
    print('>The choosen directory to read the structure XYZ file is: '+Inp.DirStuFilXYZ+'\n')
    print('>The choosen directory to write the elemental DoS file is: '+Inp.DirSavPltDOS+'\n')
    print('>The choosen directory to write the elemental DoS plots is: '+Inp.DirSavFilDOS+'\n')
    NumStpEneGrd=RaW.ReaNumLinTxtFil(Inp.DirAtmFilDOS+Inp.GenTotNamDOS)-Inp.IgnLinFilDoS #NumStpEneGrd=Number Steps Energy Grid.
    print('>Generating all indivual atomic orbitals has started.')
    NumAtmStu=RaW.ReaNumLinTxtFil(Inp.DirStuFilXYZ+Inp.StuNamXYZ)-Inp.IgnLinFilXYZSta-Inp.IgnLinFilXYZEnd #NumAtmStu=Number Atoms Structure.
    IndEleNum=RaW.ReaDatStrTxtFil(Inp.DirStuFilXYZ+Inp.StuNamXYZ,NumAtmStu+Inp.IgnLinFilXYZSta,1,Inp.IgnLinFilXYZSta,0) #IndEleNum=Index Element Number.
    NumAtmEle=DetStuEle(NumAtmStu,IndEleNum) #NumAtmEle=Number Atomic Elements.
    NumAtmFilNamDOS,AllEleAtmFilNamDOS,AllAtmFilNamDOS,AllAtmFilOrbDOS=GenAtmFilNamDOS(NumAtmStu,IndEleNum) #NumAtmFilNamDOS=Number Atomic File Names DOS.
    AllEleAtmFilNamDOS=All Element Atomic Files Names DOS. AllAtmFilNamDOS=All Atomic File Names DOS. AllAtmFilOrbDOS=All Atomic File Orbital DOS.
    print('>Generating all indivual atomic orbitals has finished.'+'\n')
    print('>Read and sum of all individual atomic orbitals DOS has started.'+'\n')
    TotNumOrb,MarEleDOS,EneGrd,TotStrDOS,TotEleDOS,TotLayDOS,LayEleDOS=SumEleDOS(NumAtmEle,NumStpEneGrd,NumAtmFilNamDOS,AllEleAtmFilNamDOS,AllAtmFilNamDOS,AllAtmFilOrbDOS)
    #TotNumOrb=Total Number Orbitals. MarEleDOS=Mark Elemental DOS. EneGrd=Energy Grid. TotStrDOS=Total Structure DOS. TotEleDOS=Total Elemental DOS.
    LayEleDOS=Layer Electronic DOS.
    print('>Read and sum of all indivual atomic orbitals DOS finished.'+'\n')
    print('>Save of the sum of all individual atomic orbitals DOS started.'+'\n')
    if Inp.SavIndDOS==True: SavInvOrbDOS(NumAtmEle,TotNumOrb,MarEleDOS,NumStpEneGrd,EneGrd,TotEleDOS)

```

```

if Inp.SavpLyDoS==True:
    for i_StuLay in range(0,Inp.StuLayNum,1): #i_StuLay=i-counter Structure Layer.
        SavGroOrbLayDOS(i_StuLay,NumAtmEle,TotNumOrb,MarEleDOS,NumStpEneGrd,EneGrd,LayEleDOS[i_StuLay])
    if Inp.SavGroDOS==True: SavGroOrbStcDOS(NumAtmEle,TotNumOrb,MarEleDOS,NumStpEneGrd,EneGrd,TotEleDOS)
    print('>Save of the sum of all individual atomic orbitals DOS finished.'+'\n')
    print('>Plot of all individual atomic orbitals DOS finished.'+'\n')
    PltDOS(NumAtmEle,MarEleDOS,EneGrd,TotEleDOS,TotStrDOS)
    print('>Plot of all individual atomic orbitals DOS started.'+'\n')
    print('>All generated atomic orbitals DOS had been read, saved and plotted.'+'\n')

#WelMes=Welcome Message.
#welstr=welcome string. string variable. This variable contains the welcome message to print on screen.
def WelMes(welstr):
    Mar=' ' #Mar=Margin.
    for i_Wel in range(0,len(welstr),1): Mar=Mar+'*' #i_Wel=index Welcome.
    print('\n\n\n')
    print(Mar)
    print(welstr)
    print(Mar)
    print('\n\n\n')
    return

WelMes('*** Welcome to PDOS Grouping 1.1.0 ***')
if Inp.MasDatAna==True:
    for i_Stc in range(0,Lst.DifStcStuLayNum,1): #i_Stc=i-counter Structure.
        Inp.StuLayNum=Lst.DifStcStuLayNum[i_Stc]
        Inp.GenAtmNamDOS=Lst.DifStc_GenAtmNamDOS[i_Stc]
        Inp.GenTotNamDOS=Lst.DifStc_GenTotNamDOS[i_Stc]
        Inp.GenNamProEleDOS=Lst.DifStc_GenNamProEleDOS[i_Stc]
        Inp.StuNamXYZ=Lst.DifStc_StuNamXYZ[i_Stc]
        Inp.DirStuFilXYZ=Lst.DifStc_DirStuFilXYZ[i_Stc]
        Inp.BasNamSavFil=Lst.DifStc_BasNamSavFil[i_Stc]
        BasNamQDFil=Lst.DifStc_BasNamQDFil[i_Stc]
        for i_Sim in range(0,len(Lst.DifStc_DirAtmFilDOS[i_Stc]),1): #i_Sim=i-counter Simulation.
            Inp.DirAtmFilDOS='/home/jorge/UES_Cluster/Calculations_Outputs_LARCAD/CQD_Stacking_Graphite_Nitrogenized/'+Inp.BasNamSavFil+'/'+Inp.BasNamSavFil+'_DEL/'
            '+Lst.DifStc_DirAtmFilDOS[i_Stc][i_Sim][0]+'/'
            Inp.SavLab=Lst.DifStc_DirAtmFilDOS[i_Stc][i_Sim][1]
            GroExe()
    else: GroExe()
WelMes('*** Have a nice day!!! ***')

```

Archivo para lectura y escritura de resultados - DOS_Ele_Grouping_RaW_TXT.py

```

from numpy import zeros

#ReaNumLinTxtFil=Read Number Lines Text File.
#variables:
#fn=file name. string variable.
def ReaNumLinTxtFil(fn):
    with open(fn) as of: #of=open file.
        for i, l in enumerate(of):
            pass
    return i+1

#ReaDatStrTxtFil=Read Data as String form Text File.
#variables:
#fn=file name to read. string variable.
#nl=number lines. integer variable. This variable indicate the number of lines to read. The total number of lines in the files is equal to nl+igl.
#nc=number columns. integer variable. This variable indicate the number of columns to read. The total number of columns in the files is equal to nc+igc.
#igl=ignored lines. integer variable. This variable indicate the number of lines to ignore. The total number of lines in the files is equal to nl+igl.
#igc=ignored lines. integer variable. This variable indicate the number of columns to ignore. The total number of columns in the files is equal to nc+igc.
def ReaDatStrTxtFil(fn,nl,nc,igl,igc):
    rd=['']*((nl-igl)*nc) #rd=read data.
    if nc > 1:
        for il in range(igl,nl,1): #il=index line.
            rd[il]=['']*nc
    of=open(fn,'r')#of=open file.
    al=of.readlines() #al=all lines.
    of.close()
    if nc > 1:
        for il in range(igl,nl,1): #il=index line.
            sli=al[il].split() #sc=splited line i.
            for ic in range(igc,nc,1): #ic=index column.
                rd[il-igl][ic-igc]=str(sli[ic])
    else:
        for il in range(igl,nl,1): #il=index line.
            sli=al[il].split() #sc=splited line i.
            rd[il-igl]=str(sli[0])
    return rd

#ReaDatFloTxtFil=Read Data as Float form Text File.
#variables:
#fnr=file name to read. string variable.
#nltr=number lines to read. integer variable. This variable indicate the number of lines to read.
#nctr=number columns to read. integer variable. This variable indicate the number of columns to read.
#fltr=firts line to read. integer variable. This variable indicate the firts line to be read on the file.
#lltr=last line to read. integer variable. This variable indicate the lasts line to be read on the file.
#sltr=step lines to read. integer variable. This variable indicate the step in line to be read on the file.
#fctr=firts column to read. integer variable. This variable indicate the firts column to be read on the file.
#lctr=last column to read. integer variable. This variable indicate the lasts column to be read on the file.
#sctr=step columns to read. integer variable. This variable indicate the step in columns to be read on the file.
def ReaDatFloTxtFil(fn,nltr,nctr,fltr,lltr,sltr,fctr,lctr,sctr):
    rd=zeros(shape=(nltr,nctr),dtype=float) #rd=read data.

```

```

of=open(fn,'r')#of=open file.
al=of.readlines() #al=all lines.
of.close()
actl=0 #actl=actual line.
for il in range(fltr,lltr,sltr): #il=index line.
    actc=0 #actc=actual column.
    sli=al[il].split() #sc=splited line i.
    for ic in range(fcctr,lctr,sctr): #ic=index column.
        rd[actl][actc]=sli[ic]
        actc=actc+1
    actl=actl+1
return rd

#WriDatStrTxtFil=Write Data as String form Text File.
#variables:
#fnw=file name to write. string variable.
#nl=number lines. integer variable. This variable indicate the number of lines to be write.
#nc=number columns. integer variable. This variable indicate the number of columns to be write.
#db=data base. float array dimensions [nl][nc]. This variable contains the data base to be write.
#dec=decimals. integer variable. This variable indicates the number of decimals to be write.
def WriDatStrTxtFil(fnw,nl,nc,db,dec):
    cf=open(fnw,'w') #cf=created file.
    for inl in range(0,nl,1): #inl=index number line.
        ltw=' ' #ltw=line to write.
        for inc in range(0,nc,1): #inc=index number column.
            ltw=ltw+str(db[inl][inc])+' '
        cf.write(ltw+'\n')
    cf.close()
    return

#WriDatStrTxtFilHea=Write Data as String form Text File with Headers.
#variables:
#fnw=file name to write. string variable.
#nl=number lines. integer variable. This variable indicate the number of lines to be write.
#nc=number columns. integer variable. This variable indicate the number of columns to be write.
#db=data base. float array dimensions [nl][nc]. This variable contains the data base to be write.
#dec=decimals. integer variable. This variable indicates the number of decimals to be write.
#hea=header. string variable. This variable indicates the file header.
def WriDatStrTxtFilHea(fnw,nl,nc,db,dec,hea):
    cf=open(fnw,'w') #cf=created file.
    cf.write(hea+'\n')
    for inl in range(0,nl,1): #inl=index number line.
        ltw=' ' #ltw=line to write.
        for inc in range(0,nc,1): #inc=index number column.
            #act_dat="%. "+str(dec)+"f"%db[inl][inc]
            ltw=ltw+str(db[inl][inc])+' '
        cf.write(ltw+'\n')
    cf.close()
    return

```

7. Anexo G: Código de Python3 para resolución de modelo semi-empírico

Archivo *input* de parámetros del sistema - SGLE_Input.py

```

# =====
# =====
# Stacking Graphene Layers Eigenproblem - Input (SGLE_Input) Version == 1.0.0=====
# Start SGLE_Input description=====
#
# End SGLE_Input description=====
# =====
# =====
# SGLE_Input parameters=====
# Start initialization of parameters for SGLE_Input.=====

Dir_Sav='./Found_Eigen_Fun_Val/' #Dir_Sav=Directory Saving. String variable.
Nrm_Mtd='cua_nrm' #nrm_mtd=normalization method. string variable. It contains the method in which the eigen-functions will be normalized.
#The only allowed values for this parameter are:
#non_nrm=non normalization. No nomalization is performed.
#rct_nrm=rectangular normalization. Integrates using Rectangular method.
#cua_nrm=cuadratic normalization. Integrates using Simpson method.

# *****
# Graphene layers data.
# *****
Lay_Num=4 #lay_num=layer interacting number. Integer variable. Adimensional
Lay_Atm_Num=30 #lay_atm_num=layer atoms number. Integer variable. Adimensional.
Lay_EqS=0.3334 #lay_eqs=layer equilibrium separation. Float variable. Units in mm.
Lay_BEn=[-0.626,-0.626,-1.156,-1.742] #Lay_BEn=Layer Binding Energy. Float variable. Units in eV.
Lay_NPE=False #Lay_NPE=Layer (Re)Normalization Potential Energy. Boolean variable.
# *****

```

```

# *****
# Spatial mesh input.
# *****
Spa_Sta=-1.1000 #ISV=Initial Spatial Value. Units nm.
Spa_End=+1.000 #FSV=Final Spatial Value. Units nm.
Spa_Stp=None #Spa_Stp=Spatial Step. Units nm.
Spa_Stp_Num=5000 #Spa_Grd_Num=Spatial Step Number. Integer variable. Adimensional.
# *****

# *****
# Energy mesh input.
# *****
Ene_Sta=000.00 #IEV=Initial Energy Value. Float variable. Units eV.
Ene_End=100.00 #FEV=Final Energy Value. Float variable. Units eV.
Ene_Stp=None #Ene_Stp=Energy Step. Float variable. Units eV.
Ene_Stp_Num=100 #Ene_Stp_Num=Energy Step Number. Integer variable. Adimensional.
PEn_ACr=True #PEn_ACr=Potential Energy Above Cero. Boolean variable. Adimensional.
# *****

# *****
# Egein function mesh input.
# *****
EEF_LFV=+1E-6 #EEF_LFV=Electronic Eigen-Function Left Firts Value. Float variable. Units in nm-2.
EEF_LSV=-1E-5 #EEF_LSV=Electronic Eigen-Function Left Second Value. Float variable. Units in nm-2.
EEF_RFV=+5*10**3 #EEF_RFV=Electronic Eigen-Function Right Firts Value. Float variable. Units in nm-2.
EEF_RSV=0 #EEF_RLV=Electronic Eigen-Function Right Second Value. Float variable. Units in nm-2.
EEF_Exp=250 #EEF_Exp=Electronic Eigen-Function Extra Points. Integer variable. Adimensional.
EEF_MRT=50 #EEF_MRT=Electronic Eigen-Function Maximum Root Tries. Integer variable. Adimensional.
EEF_Err_Tol=1E-4 #EEF_Tol=Electronic Eigen-Function Error Tolerance. Float variable. Units in eV.
PEF_Err_Tol=1E-3 #PEF_Err_Tol=Potential Energy Function Error Tolerance. Float variable. Units in eV.
# *****

# *****
# Visualizing generated data.
# *****
Wai_Tim=0.5 #wai_tim=waiting time. Float variable. Units in seconds.
Wai_Act=True #wai_act=waiting activate. Boolean variable. Adimensional.
Plt_ATr=True #Plt_ATr=Plot All Tries. Boolean variable. Adimensional.
# *****

# End initialization of parameters for SGLE_Input.=====
# =====
# =====

```

Archivo principal - SGLE_Main.py

```

# # =====
# =====
# Stacking Graphene Layers Eigenproblem - Main (SGLE_Main) Version == 1.0.0=====
# Start SGLE_Main description=====
#
# End SGLE_Main description=====
# =====
# =====

# =====
# =====
# SGLE_Main modules=====
# Start module calling for SGLE_Main.=====
import SGLE_Input as inp
import SGLE_SEILBE as seilbe
import SGLE_Plotter as plt
import SGLE_Numerov_Solver as numsol
import SGLE_Read_Write as raw
# End module calling for SGLE_Main.=====
# =====
# =====

# =====
# =====
# SGLE_Main libraries=====
# Start calling of needed libraries for SGLE_SEILBE functions.=====
from numpy import zeros,sign,longdouble
from time import sleep
# End calling of needed libraries for SGLE_SEILBE functions.=====
# =====
# =====

# =====
# =====
# SGLE_Main parameters=====
# Start initialization of parameters for SGLE_Main.=====

# End initialization of parameters for SGLE_Main.=====
# =====
# =====

```

```

cf_meh2=26.231 #247609304 #26.247 # #*10**6 #Cnv_mh2=Conversion Factor mass electron (me) plack bar constant squared (h^{2}). Units in (eV*nm)^{-1}
spa_num=0 #spa_num=spatial (elements) number.
spe_num=0 #spe_num=spatial (elements) extended number.
spa_stp=0 #spa_stp=spatial step.
ene_num=0 #ene_num=energy (elements) number.
ene_stp=0 #ene_stp=energy step.

spa_sep=zeros(shape=(2),dtype=longdouble,order='F') #spa_sep=spatial start end points.
ene_sep=zeros(shape=(2),dtype=longdouble,order='F') #ene_sep=energy start end points. longdouble
eef_ivl=zeros(shape=(4),dtype=longdouble,order='F') #eef_ivl=electronic eigen-functions initializing values.

spa_sep[0],spa_sep[1]=inp.Spa_Sta,inp.Spa_End
ene_sep[0],ene_sep[1]=inp.Ene_Sta,inp.Ene_End
eef_ivl[0],eef_ivl[1],eef_ivl[2],eef_ivl[3]=inp.EEF_LFV,inp.EEF_LSV,inp.EEF_RFV,inp.EEF_RSV
if inp.Spa_Stp_Num==0 or inp.Spa_Stp_Num==None:
    spa_stp=inp.Spa_Stp
    spa_num=int((spa_sep[1]-spa_sep[0])/spa_stp)
elif inp.Spa_Stp==0 or inp.Spa_Stp==None:
    spa_num=2*inp.Spa_Stp_Num+1
    spa_stp=(spa_sep[1]-spa_sep[0])/spa_num
spe_num=spa_num+2*inp.EEF_Exp

spa_grd=zeros(shape=(spe_num),dtype=longdouble,order='F') #spa_grd=spatial grid.
pen_grd=zeros(shape=(2,spe_num),dtype=longdouble,order='F') #pen_grd=potential energy grid.

for i_spa_num in range(0,inp.EEF_Exp,1): #i_spa_num=i-counter spa_num.
    spa_grd[i_spa_num]=spa_sep[0]-(inp.EEF_Exp-i_spa_num)*spa_stp
    pen_grd[0][i_spa_num]=seilbe.SEILBE(spa_grd[i_spa_num],inp.Lay_Num,inp.Lay_Atm_Num,inp.Lay_EqS)
    pen_grd[1][i_spa_num]=cf_meh2*pen_grd[0][i_spa_num]
for i_spa_num in range(inp.EEF_Exp,inp.EEF_Exp+spe_num,1): #i_spa_num=i-counter spa_num.
    spa_grd[i_spa_num]=spa_sep[0]+(i_spa_num-inp.EEF_Exp)*spa_stp
    pen_grd[0][i_spa_num]=seilbe.SEILBE(spa_grd[i_spa_num],inp.Lay_Num,inp.Lay_Atm_Num,inp.Lay_EqS)
    pen_grd[1][i_spa_num]=cf_meh2*pen_grd[0][i_spa_num]
spa_grd[spa_num+inp.EEF_Exp]=spa_sep[1]
pen_grd[0][spa_num+inp.EEF_Exp]=seilbe.SEILBE(spa_grd[spa_num+inp.EEF_Exp],inp.Lay_Num,inp.Lay_Atm_Num,inp.Lay_EqS)
pen_grd[1][spa_num+inp.EEF_Exp]=cf_meh2*pen_grd[0][spa_num+inp.EEF_Exp]
for i_spa_num in range(spa_num+inp.EEF_Exp+1,spe_num,1): #i_spa_num=i-counter spa_num.
    spa_grd[i_spa_num]=spa_sep[1]+(i_spa_num-spa_num-inp.EEF_Exp)*spa_stp
    pen_grd[0][i_spa_num]=seilbe.SEILBE(spa_grd[i_spa_num],inp.Lay_Num,inp.Lay_Atm_Num,inp.Lay_EqS)
    pen_grd[1][i_spa_num]=cf_meh2*pen_grd[0][i_spa_num]

spa_max,pen_max=0,max(pen_grd[0]) #spa_max=spatial maximum. pen_max=potential energy maximum.
spa_min,pen_min=0,min(pen_grd[0]) #spa_min=spatial minimum. pen_min=potential energy minimum.
if inp.Lay_NPE==True:
    ren_cnt=inp.Lay_BEn[inp.Lay_Num-1]/pen_min #ren_cnt=re normalization constant.
    for i_pen_num in range(0,spe_num,1): #i_pen_num=i-counter potential energy number (elements grid).
        pen_grd[0][i_pen_num]=ren_cnt*pen_grd[0][i_pen_num]
        pen_grd[1][i_pen_num]=ren_cnt*pen_grd[1][i_pen_num]
    pen_max=max(pen_grd[0]) #pen_max=potential energy maximum.
    pen_min=min(pen_grd[0]) #pen_min=potential energy minimum.
#if pen_min<0: pen_max=0
if inp.PEn_ACr==False: pen_max=0
#pen_min=0.5
if inp.Ene_Stp_Num==0 or inp.Ene_Stp_Num==None:
    ene_stp=inp.Ene_Stp
    ene_num=int((ene_sep[1]-ene_sep[0])/ene_stp)
elif inp.Ene_Stp==0 or inp.Ene_Stp==None:
    ene_num=inp.Ene_Stp_Num
    ene_stp=(pen_max-pen_min)/ene_num
eee_grd=zeros(shape=(2,ene_num),dtype=longdouble,order='F') #eee_grd=eigen-energy grid.
for i_ene_num in range(0,ene_num-1,1): #inesgrid=i-counter ene_num.
    eee_grd[0][i_ene_num]=pen_min+i_ene_num*ene_stp
    eee_grd[1][i_ene_num]=cf_meh2*eee_grd[0][i_ene_num]
if inp.Ene_Stp_Num==0 or inp.Ene_Stp_Num==None: eee_grd[0][ene_num-1]=ene_sep[1]
elif inp.Ene_Stp==0 or inp.Ene_Stp==None: eee_grd[0][ene_num-1]=pen_max
eee_grd[1][ene_num-1]=cf_meh2*eee_grd[0][ene_num-1]

for inesgrid in range(0,spe_num,1): #inesgrid=index number elements spatial grid.
    if pen_max==pen_grd[0][inesgrid]:
        spa_max=spa_grd[inesgrid]
        break
for inesgrid in range(0,spe_num,1): #inesgrid=index number elements spatial grid.
    if pen_min==pen_grd[0][inesgrid]:
        spa_min=spa_grd[inesgrid]
        break
print('Potential maximum energy=',pen_max,' at ', 'x=',spa_max)
print('Potential minimum energy=',pen_min,' at ', 'x=',spa_min)
plt.SFP_2D(spa_grd,pen_grd[0],'X position (nm)','Energy (eV)','Potential function',inp.Dir_Sav)
raw.Wri_Dat_Str_Txt(inp.Dir_Sav+'Binding_Potential.dat',[spa_grd,pen_grd[0]],2,spe_num,'w')
input('Start eigen-values search???'+'\\n'+ '\\t'+ '\\t'+ 'Press any key to continue...')

efi_plt=zeros(shape=(3),dtype=int,order='F') #efi_plt=eigen function interval plot.
efi_plt[0],efi_plt[1],efi_plt[2]=inp.EEF_Exp,inp.EEF_Exp+spa_num,spa_num

eef=[] #eef=eigen energies found.
eig_lab=[] #eig_lab=eigen labels.
neef=0 #neef=number eigen energies found.
ntree=0 #ntree=number tries to root eigen energies.
if inp.Plt_ATr==True:
    ene_bel,ene_act,ene_abv=0,0,0 #ene_bel=energy below. ene_act=energy actual. ene_abv=energy above.
    err_bel,err_act,err_abv=0,0,0 #err_bel=error below. err_act=error actual. err_abv=error above.
    rot_sea_con=True #rot_sea_con=root cluing continue.
    rot_cle_fnd=False #rot_cle_fnd=root cleue found.
    aee=False #aee=acceptable eigen energy.
    fig_con,axs_efn,lin_efn,err_txt=plt.Plt_Ini(spa_sep,spa_grd,pen_grd,efi_plt)
    ineegrid=0 #inesgrid=index number elements energy grid.
    while rot_sea_con==True:
        if rot_cle_fnd==False and ineegrid<ene_num: ene_act=eee_grd[0][ineegrid]

```

```

else: ene_act=(ene_bel+ene_abv)/2
#print('Iteration',ineegrid)
#print('Energy report',ene_bel,ene_act,ene_abv)
#print('Error',err_bel,err_act,err_abv)
#print(ene_act,rot_cle_fnd,ineegrid)
pen_spl=numsol.PEN_HCT(spe_num,inp.EEF_Exp,inp.EEF_Exp+spa_num,spa_grd,pen_grd[0],ene_act)
#,inp.PEF_Err_Tol,[inp.Lay_Num,inp.Lay_At_m_Num,inp.Lay_EqS],seilbe.SEILBE) #pen_spl=potential energy spatial limits.
#print('\t'+Solving for energy '+str(ene_act))
if pen_spl==None:
    if ineegrid<ene_num: ineegrid=ineegrid+1
    else: rot_sea_con=False
elif ene_act==0: break
else:
    #print('\t'+Energies (test-root1-root2):',ene_act,pen_grd[0][pen_spl[0]],pen_grd[0][pen_spl[1]])
    #input('???')
    aee,err_act=numsol.QSE_NM_PT(spe_num,pen_spl,efi_plt,inp.EEF_Exp,spa_stp,spa_grd,pen_grd[1],cf_meh2*ene_act,neef,eef_ivl,inp.EEF_Err_Tol,inp.Nrm_Mtd,cf_meh2,
        inp.Dir_Sav,fig_con,axs_efn,lin_efn,err_txt,rot_cle_fnd,plt.Plt_Mea_Man_Upd,raw.Wri_Dat_Str_Txt)
    if inp.Wai_Act==True: sleep(inp.Wai_Tim)
    if aee==True:
        #input('Eigen-energy found!!!. The eigen-energy is e='+str(ene_act)+'...')
        print('\t'+Eigen-energy found!!!. The eigen-energy is e='+str(ene_act)+'...')
        eef.append(ene_act)
        eig_lab.append(neef)
        neef=neef+1
        ineegrid=ineegrid+1
        rot_cle_fnd=False
        ntree=0
        ene_bel,ene_act,ene_abv=0,0,0
        err_bel,err_act,err_abv=0,0,0
    else:
        if rot_cle_fnd==False:
            if sign(err_act)!=sign(err_bel) and err_bel!=0:
                rot_cle_fnd=True
                ene_abv=ene_act
                err_abv=err_act
            else:
                ene_bel=ene_act
                err_bel=err_act
                if ineegrid<ene_num: ineegrid=ineegrid+1
                else: rot_sea_con=False
        else:
            #print('ener...','below=',ene_bel,'above=',ene_abv,'act=',ene_act)
            #print('err...','below=',err_bel,'above=',err_abv,'act=',err_act)
            #input('press any key to continue...')
            if ntree<inp.EEF_MRT: ntree=ntree+1
            else:
                print('\t'+Possible energy found between '+str(ene_bel)+' and '+str(ene_abv))
                eef.append(ene_act)
                eig_lab.append(neef)
                neef=neef+1
                ineegrid=ineegrid+1
                rot_cle_fnd=False
                ntree=0
                ene_bel,ene_act,ene_abv=0,0,0
                err_bel,err_act,err_abv=0,0,0
            if sign(err_act)==sign(err_abv):
                ene_abv=ene_act
                err_abv=err_act
            if sign(err_act)==sign(err_bel):
                ene_bel=ene_act
                err_bel=err_act
    else: print('No plotting implemented...Sorry for the inconvenieces')

raw.Wri_Dat_Str_Txt(inp.Dir_Sav+'Eigen_Val.dat',[eig_lab,eef],2,neef,'w')

print('> A total of '+str(neef)+' eigen-energies had been found!!! ')
print('\t'+Index number'+\t'+Eigen-energy')
for i_neef in range(0,neef,1): #i_neef=i-counter neef.
    print('\t'+str(i_neef+1),eef[i_neef])
print('\n\n'+Have a nice day'+\n\n')

```

Archivo función de energía potencial entre placas de grafeno de iniciación - SGLE_SEULBE.py

```

# # =====
# =====
# SGLE_SEILBE description=====
# Stacking Graphene Layers Eigenproblem - Semi-Empirical Inter-Layer Binding Energies (SGLE_SEILBE) Version == 1.0.0
#
#
# =====
# =====
# =====
# =====
# Start calling of needed libraries for SGLE_SEILBE functions.=====
from numpy import exp,expm1,zeros

```

```

# End calling of needed libraries for SGLE_SEILBE functions.=====
# =====
# =====
# SGLE_SEILBE parameters=====
# Start initialization of needed parameters for SGLE_SEILBE functions.=====
lay_tot_ene=-400.9534751 #lay_tot_ene=layer total energy. Units in Ry.
CF_Ene_Pot_Gra_LLI=0.001 #CF_EneCF_Ene_Pot_Gra_LLI_Pot=Conversion Factor Energy Potential Graphene Layer Layer Interaction. Units meV to eV.
CF_Ene_Pot_Gra_LSI=13.605662285137 #CF_Ene_Pot_Gra_LSI=Conversion Factor Energy Potential Graphene Layer Self Interaction. Units Ry to eV.

lay_inte_typ=[[1],[0],[1]] #lay_inte_typ=layer interaction type. Units only 0 or 1 values.

D0=[-3.975825,-0.028157] #Units meV/atom.
R0=[1,1] #Units Angstrom^-1.
gamma=[1.357722,0.501727] #Adimensional.
a1=[8.768916,8.713450] #Units meV/atom.
a2=[-0.000232,-0.000005] #Units meV/atom.
a3=[0.009122,-1.356305] #Units meV/atom.
b1=[3.245134,5.168542] #Units Angstrom^-1.
b2=[12.409694,12.409694] #Units Angstrom^-1.
b3=[5.069858,1.983866] #Units Angstrom^-1.
c1=[1.786003,0.110757] #Units Angstrom.
c2=[28.113815,28.113815] #Units Angstrom.
c3=[3.289036,2.950554] #Units Angstrom.
rs12_cab=[0.0,0.0] #rs12_cab=RSL2 cut above. Units in Angstrom.
rs12_cbe=[0.0,0.0] #rs12_cbe=stacking cut below. Units in Angstrom.
lay_esm=[0.25,0.25] #[0.65,0.65] #[0.65,0.65] #[0.7,0.7] #lay_esm=layer end structure mix.
cor_sca=[50,800]#[50,900] #[500,500] #cor_sca=correction scale. Adimensional.
#NOTE!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
#Without these values the potential energy functions do not reproduce the paper reported curves!!!!!!!!!!!!
#I dont know why these factors are need. Maybe the author introduced some incorrect values by mistake????!
#NOTE!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

# End initialization of needed parameters for SGLE_SEILBE functions.
# =====
# =====

# =====
# SGLE_SEILBE related functions.=====
# Start SGLE_SEILBE functions.=====

#EPF_Gra_LLI=Energy Potential Function Graphene Layer-Layer Interaction.
#Variables:
#dlv=distance layer value. Float variable. Give the separation between graphene layers. Units in Angstrom.
#lay_atm=layer atoms. It contains the number of atoms per layer. Adimensional.
#lay_typ=layer type. Integer variable. Give the parameter for a AA or AB stacking graphene layer. Only 0 or 1 values.
def EPF_Gra_LLI(dlv,lay_atm,lay_typ):
    epf_gll=0 #epf_gll=energy potential function graphene layer layer. Units in meV.
    epf_gll=epf_gll+D0[lay_typ]*exp(gamma[lay_typ]*(1-dlv/R0[lay_typ]))
    epf_gll=epf_gll+(a1[lay_typ]/(2+expm1(b1[lay_typ]*(dlv-c1[lay_typ])))
    epf_gll=epf_gll+(a2[lay_typ]/(2+expm1(b2[lay_typ]*(dlv-c2[lay_typ])))
    epf_gll=epf_gll+(a3[lay_typ]/(2+expm1(b3[lay_typ]*(dlv-c3[lay_typ])))
    epf_gll=CF_Ene_Pot_Gra_LLI*cor_sca[lay_typ]*lay_atm*epf_gll #Conversion meV ==> eV
    return epf_gll

#LPo_ScE=Layer Position Shift (respect) Center.
#Variables:
#lay_num=layer number. It contains the number of layer in the system.
#lay_act=layer actual. It contains the number of the actual layer. Adimensional.
def LPo_ScE(lay_num,lay_act):
    lay_pos=0 #lay_pos=layer position.
    if lay_num%2==0: lay_pos=lay_act-(lay_num-1)/2
    else: lay_pos=lay_act-(lay_num-1)/2
    return lay_pos

#LIT_NTP=Layer Interaction Neighbour Type.
#Variables:
#lay_num=layer number. It contains the number of layer in the system.
#lay_act=layer actual. It contains the number of the actual layer. Adimensional.
#lay_nei=layer neighbour. It indicates which neighbour interaction is wanted.
def LIT_NTP(lay_num,lay_act):
    inte_ntp=zeros(shape=(lay_num-1),dtype=int,order='F') #inte_ntp=interaction neighbour type.
    i_lay_nei=0 #i_lay_nei=i-counter layer neighbour.
    for lay_nei in range(0,lay_num,1): #lay_nei=layer neighbour.
        if lay_act!=lay_nei:
            if abs(lay_nei-lay_act)%2==0: inte_ntp[i_lay_nei]=1
            i_lay_nei=i_lay_nei+1
    return inte_ntp

hw=10
k=0.5*26.231*hw**2
#SEILBE=Semi-Empirical Inter-Layer Binding Energy.
#variables:
#xv=x value. float variable. it contains the value to evaluate EPF. Units in nm.
#lay_num=layer number. It contains the number of layer in the system.
#lay_atm=layer atoms. It contains the number of atoms per layer. Adimensional.
#lay_sep=layer separation. It contains the separation between layer. Units in nm.
def SEILBE_A(xv,lay_num,lay_atm,lay_sep):
    EPFV=0 #0.5*k*xv**2 #EPFV=Energy Potential Function.
    #if -0.2<xv and xv<0.2: EPFV=-100
    #"""
    if lay_num==1:
        xlc=10*xv #xlc=x layer centered.

```

```

for i_inte_typ in range(0,2,1): #i_inte_typ=i-counter inte_typ.
    epfv_lay=0.0 #epfv_lay=energy potential function layer.
    if rsl2_cab[i_inte_typ]<xlc: epfv_lay=EPF_Gra_LLI(xlc,lay_atm,i_inte_typ)
    elif rsl2_cbe[i_inte_typ]<=xlc and xlc<=rsl2_cab[i_inte_typ]: epfv_lay=EPF_Gra_LLI(rsl2_cab[i_inte_typ],lay_atm,i_inte_typ)
    elif -rsl2_cbe[i_inte_typ]<=xlc and xlc<=rsl2_cbe[i_inte_typ]: epfv_lay=EPF_Gra_LLI(abs(xlc),lay_atm,i_inte_typ)
    elif -rsl2_cab[i_inte_typ]<=xlc and xlc<=-rsl2_cbe[i_inte_typ]: epfv_lay=EPF_Gra_LLI(rsl2_cab[i_inte_typ],lay_atm,i_inte_typ)
    elif xlc<=-rsl2_cab[i_inte_typ]: epfv_lay=EPF_Gra_LLI(-xlc,lay_atm,i_inte_typ)
    EPFV=EPFV+lay_esm[i_inte_typ]*epfv_lay
else:
    for i_ln in range(0,lay_num,1): #i_ln=i-counter layer number.
        xlc=10*(xv-lay_sep*LPO_ScE(lay_num,i_ln)) #xlc=x layer centered.
        inte_ntp=LIt_MTp(lay_num,i_ln) #inte_ntp=interaction neighbour type.
        for i_iln in range(0,lay_num-1,1): #i_iln=i-counter interaction with other layers number.
            inte_typ=inte_ntp[i_iln] #inte_typ=interaction type.
            epfv_lay=0 #epfv_lay=energy potential function layer.
            if rsl2_cab[inte_typ]<xlc: #epfv_lay=EPF_Gra_LLI(xlc,lay_atm,inte_typ)
                if i_iln!=lay_num-1: epfv_lay=EPF_Gra_LLI(xlc,lay_atm,inte_typ)
            else:
                for i_inte_typ in range(0,2,1): #i_inte_typ=i-counter inte_typ.
                    epfv_lay=epfv_lay+lay_esm[i_inte_typ]*EPF_Gra_LLI(xlc,lay_atm,i_inte_typ)
                elif rsl2_cbe[inte_typ]<=xlc and xlc<=rsl2_cab[inte_typ]: epfv_lay=EPF_Gra_LLI(rsl2_cab[inte_typ],lay_atm,inte_typ)
                elif -rsl2_cbe[inte_typ]<=xlc and xlc<=rsl2_cbe[inte_typ]: epfv_lay=EPF_Gra_LLI(abs(xlc),lay_atm,inte_typ)
                elif -rsl2_cab[inte_typ]<=xlc and xlc<=-rsl2_cbe[inte_typ]: epfv_lay=EPF_Gra_LLI(rsl2_cab[inte_typ],lay_atm,inte_typ)
                elif xlc<=-rsl2_cab[inte_typ]: #epfv_lay=EPF_Gra_LLI(-xlc,lay_atm,inte_typ)
                    if i_iln!=0: epfv_lay=EPF_Gra_LLI(-xlc,lay_atm,inte_typ)
                else:
                    for i_inte_typ in range(0,2,1): #i_inte_typ=i-counter inte_typ.
                        epfv_lay=epfv_lay+lay_esm[i_inte_typ]*EPF_Gra_LLI(-xlc,lay_atm,i_inte_typ)
            if inte_typ==0: epfv_lay=0
            #if inte_typ==1: epfv_lay=0
            EPFV=EPFV+epfv_lay
        return EPFV

def SEILBE_B(xv,lay_num,lay_atm,lay_sep):
    EPFV=0 #0.5*k*xv**2 #EPFV=Energy Potential Function.
    if lay_num==1:
        xlc=10*xv #xlc=x layer centered.
        for i_inte_typ in range(0,2,1): #i_inte_typ=i-counter inte_typ.
            epfv_lay=0.0 #epfv_lay=energy potential function layer.
            if rsl2_cab[i_inte_typ]<xlc: epfv_lay=EPF_Gra_LLI(xlc,lay_atm,i_inte_typ)
            elif rsl2_cbe[i_inte_typ]<=xlc and xlc<=rsl2_cab[i_inte_typ]: epfv_lay=EPF_Gra_LLI(rsl2_cab[i_inte_typ],lay_atm,i_inte_typ)
            elif -rsl2_cbe[i_inte_typ]<=xlc and xlc<=rsl2_cbe[i_inte_typ]: epfv_lay=EPF_Gra_LLI(abs(xlc),lay_atm,i_inte_typ)
            elif -rsl2_cab[i_inte_typ]<=xlc and xlc<=-rsl2_cbe[i_inte_typ]: epfv_lay=EPF_Gra_LLI(rsl2_cab[i_inte_typ],lay_atm,i_inte_typ)
            elif xlc<=-rsl2_cab[i_inte_typ]: epfv_lay=EPF_Gra_LLI(-xlc,lay_atm,i_inte_typ)
            EPFV=EPFV+lay_esm[i_inte_typ]*epfv_lay
    else:
        for i_ln in range(0,lay_num,1): #i_ln=i-counter layer number.
            xlc=10*(xv-lay_sep*LPO_ScE(lay_num,i_ln)) #xlc=x layer centered.
            inte_ntp=LIt_MTp(lay_num,i_ln) #inte_ntp=interaction neighbour type.
            for i_iln in range(0,lay_num-1,1): #i_iln=i-counter interaction with other layers number.
                inte_typ=inte_ntp[i_iln] #inte_typ=interaction type.
                epfv_lay=0 #epfv_lay=energy potential function layer.
                if rsl2_cab[inte_typ]<xlc and i_iln!=lay_num-1: epfv_lay=EPF_Gra_LLI(xlc,lay_atm,inte_typ)
                elif rsl2_cbe[inte_typ]<=xlc and xlc<=rsl2_cab[inte_typ]: epfv_lay=EPF_Gra_LLI(rsl2_cab[inte_typ],lay_atm,inte_typ)
                elif -rsl2_cbe[inte_typ]<=xlc and xlc<=rsl2_cbe[inte_typ]: epfv_lay=EPF_Gra_LLI(abs(xlc),lay_atm,inte_typ)
                elif -rsl2_cab[inte_typ]<=xlc and xlc<=-rsl2_cbe[inte_typ]: epfv_lay=EPF_Gra_LLI(rsl2_cab[inte_typ],lay_atm,inte_typ)
                elif xlc<=-rsl2_cab[inte_typ] and i_iln!=0: epfv_lay=EPF_Gra_LLI(-xlc,lay_atm,inte_typ)
                #if inte_typ==0: epfv_lay=0
                #if inte_typ==1: epfv_lay=0
                EPFV=EPFV+epfv_lay
            if i_iln==0 or i_iln==lay_num-1:
                for i_inte_typ in range(0,2,1): #i_inte_typ=i-counter inte_typ.
                    epfv_lay=0 #epfv_lay=energy potential function layer.
                    if i_iln==0:
                        if xlc<=-rsl2_cab[i_inte_typ]: epfv_lay=EPF_Gra_LLI(-xlc,lay_atm,i_inte_typ)
                        elif -rsl2_cbe[i_inte_typ]<=xlc and xlc<=0: epfv_lay=EPF_Gra_LLI(rsl2_cab[i_inte_typ],lay_atm,i_inte_typ)
                    else:
                        if rsl2_cab[i_inte_typ]<xlc: epfv_lay=EPF_Gra_LLI(xlc,lay_atm,i_inte_typ)
                        elif 0<xlc and rsl2_cbe[i_inte_typ]<=xlc: epfv_lay=EPF_Gra_LLI(rsl2_cab[i_inte_typ],lay_atm,i_inte_typ)
                        EPFV=EPFV+lay_esm[i_inte_typ]*epfv_lay
            EPFV=EPFV/lay_num
        return EPFV

def SEILBE(xv,lay_num,lay_atm,lay_sep):
    EPFV=0 #0.5*k*xv**2 #EPFV=Energy Potential Function.
    if lay_num==1:
        xlc=10*xv #xlc=x layer centered.
        for i_inte_typ in range(0,2,1): #i_inte_typ=i-counter inte_typ.
            epfv_lay=0.0 #epfv_lay=energy potential function layer.
            if rsl2_cab[i_inte_typ]<xlc: epfv_lay=EPF_Gra_LLI(xlc,lay_atm,i_inte_typ)
            elif rsl2_cbe[i_inte_typ]<=xlc and xlc<=rsl2_cab[i_inte_typ]: epfv_lay=EPF_Gra_LLI(rsl2_cab[i_inte_typ],lay_atm,i_inte_typ)
            elif -rsl2_cbe[i_inte_typ]<=xlc and xlc<=rsl2_cbe[i_inte_typ]: epfv_lay=EPF_Gra_LLI(abs(xlc),lay_atm,i_inte_typ)
            elif -rsl2_cab[i_inte_typ]<=xlc and xlc<=-rsl2_cbe[i_inte_typ]: epfv_lay=EPF_Gra_LLI(rsl2_cab[i_inte_typ],lay_atm,i_inte_typ)
            elif xlc<=-rsl2_cab[i_inte_typ]: epfv_lay=EPF_Gra_LLI(-xlc,lay_atm,i_inte_typ)
            EPFV=EPFV+lay_esm[i_inte_typ]*epfv_lay
    else:
        for i_ln in range(0,lay_num,1): #i_ln=i-counter layer number.
            xlc=10*(xv-lay_sep*LPO_ScE(lay_num,i_ln)) #xlc=x layer centered.
            for i_inte_typ in range(0,2,1): #i_inte_typ=i-counter inte_typ.
                epfv_lay=0.0 #epfv_lay=energy potential function layer.
                if rsl2_cab[i_inte_typ]<xlc: epfv_lay=EPF_Gra_LLI(xlc,lay_atm,i_inte_typ)
                elif rsl2_cbe[i_inte_typ]<=xlc and xlc<=rsl2_cab[i_inte_typ]: epfv_lay=EPF_Gra_LLI(rsl2_cab[i_inte_typ],lay_atm,i_inte_typ)
                elif -rsl2_cbe[i_inte_typ]<=xlc and xlc<=rsl2_cbe[i_inte_typ]: epfv_lay=EPF_Gra_LLI(abs(xlc),lay_atm,i_inte_typ)
                elif -rsl2_cab[i_inte_typ]<=xlc and xlc<=-rsl2_cbe[i_inte_typ]: epfv_lay=EPF_Gra_LLI(rsl2_cab[i_inte_typ],lay_atm,i_inte_typ)
                elif xlc<=-rsl2_cab[i_inte_typ]: epfv_lay=EPF_Gra_LLI(-xlc,lay_atm,i_inte_typ)
                EPFV=EPFV+lay_esm[i_inte_typ]*epfv_lay
            return EPFV

# End SGLE_SEILBE functions.=====
# =====

```

Archivo método de Numerov - SGLE_Numerov_Solver.py

```
# =====
# =====
# =====
# Stacking Graphene Layers Eigenproblem - Numerov Solver (SGLE_Numerov_Solver) Version == 1.0.0
# This script use Numerov method algorithm to resolve schrodinger equation in one dimension with the form:
#  $d^2/dx^2[y(x)] - [v(x)-E]y(x) = 0$ 
# The script takes a spactial and energy interval, then it create a grid with the step specified by the user.
# Finally it start looking for all possible solutions with a tolerance inside the said spatial and energy interval.
# The user only need to specify:
# the spatial interval
# energy interval
# spatial step
# energy step
# error tolerace
# initial spatial guess
# potential energy function
# =====
# =====
# =====
# =====
# SGLE_Numerov_Solver related libraries.
# =====
# Start calling of needed libraries for SGLE_Numerov_Solver functions.

from numpy import sign,zeros,longdouble
from scipy.integrate import simps

# End calling of needed libraries for SGLE_Numerov_Solver functions.
# =====
# =====

# =====
# SGLE_Plotter functions.=====
# Start SGLE_Numerov_Solver functions.=====

#S_Rect_Int=Simple_Rectangular_Integration.
#np=number points. integer variable. it contains the number points in the function.
#st=step. float variable. it contains the step taken between points.
#iva=independent variable. array float dimension [nf]. it contains the values of the independent variable to plot.
#dva=dependent variable. array float dimension [nf]. it contains the values of the dependtdent variable to plot.
def S_Rect_Int(np,st,iva,dva):
    auf=1 #area under function.
    for inp in range(0,np,1): auf=auf+st*dva[inp] #inp=index number points.
    return auf

#NormF=Normalize Function.
#np=number points. integer variable. it contains the number points in the function.
#st=step. float variable. it contains the step taken between points.
#iva=independent variable. array float dimension [nf]. it contains the values of the independent variable to plot.
#dva=dependent variable. array float dimension [nf]. it contains the values of the dependtdent variable to plot.
#moi=method of integration. string variable. it contain the name of the method to integrate the function.
#moi can only have the following values:
#non_norm=non_normalization.
#sim-rect=simple-rectangular. integrates using rectangular method.
#sim-simp=simple-simpson. integrates using Simpson method.
def NormF(np,st,iva,dva,moi):
    con=1 #con=constant of normalization.
    if moi=='non_nrm': con=1
    elif moi=='rct_nrm': con=S_Rect_Int(np,st,iva,dva)
    elif moi=='cua_nrm': con=simps(y=dva,x=iva,even='avg')
    nf=(1/con)*dva #nf=nomalize function.
    return nf,con

#QSE_NM_PT=Quick Scan for Energies_Numerov Method_Plot Tries.
#Variables:
#poi_grd=points grids. It contains the number of elements in the spatial and potential energy grid.
#poi_plt=points plot.
#ppe_grd=previous and posterior extra grid.
#spa_itv=spatial interval. It contains the number of elements in the spatial grid.
#sst=spatial step. float variable. It contains the value of the spatial step.
#sgrid=spatial grid. float array dimension [nesgrid]. It contains the spatial distribution of points to resolve.
#pgrid=potetial energy grid. It contains the potential energy function evaluated alongside the spatial grid.
#te=test energy. float variable. It contains the test energy to solve schrodinger equation for this iteration.
#neef=number eigen energies found.
#fsp1fv=firts second punultimate last function values. float array dimensions [4]. It contains the value firts, second, punultimate and last values for the
function solutions.
#etol=error tolerace= float variable. It contains the tolerance to accept a solution.
#normme=normalization method. string variable. It contains the method in which the eigen functions will be normalized.
#cf_meh2=
#dir_sav=directory saving.
#fig_con=figure control.
#axs_tef=axis test eigen function.
#lin_tef=line test eigen function.
```

```

#err_txt=error text.
#odd_sol=odd solution.
def QSE_NM_PT(poi_grd,spa_itv,poi_plt,ppe_grd,sst,sgrid,pegrid,te,neef,fsplfv,etol,normme,cf_meh2,dir_sav,fig_con,axs_tef,lin_tef,err_txt,odd_sol,fun_ppd,fun_wri):
    aee=False #aee=acceptable eigen energy.
    spa_sta_grd=spa_itv[0]-ppe_grd #spa_sta_grd=spatial start grid.
    spa_end_grd=spa_itv[1]+ppe_grd #spa_end_grd=spatial end grid.
    gfgrid=te-pegrid #gfgrid=g function grid.
    efts=zeros(shape=(poi_grd),dtype=longdouble,order='F') #eigen funtion for test energy.
    efts[spa_sta_grd],efts[spa_sta_grd+1]=fsplfv[0],fsplfv[1]
    #efts[0],efts[1]=fsplfv[0],fsplfv[1]
    sst2=(sst**2)/12 #sst2=sst square.
    #for inesgrid in range(0,nesgrid+1,1): #inesgrid=index number elements spatial grid. (sgrdh+1)-2=sgrdh-1
    inesgrid=spa_sta_grd #inesgrid=index number elements spatial grid. (sgrdh+1)-2=sgrdh-1
    sol_deq=True #sol_deq=solving differential equation.
    while sol_deq==True:
        c0=1+sst2*gfgrid[inesgrid] #c0=coefficient 0.
        c1=1-5*sst2*gfgrid[inesgrid] #c1=coefficient 1.
        c2=1+sst2*gfgrid[inesgrid+2] #c2=coefficient 2.
        efts[inesgrid+2]=(2*c1*efts[inesgrid+1]-c0*efts[inesgrid])/c2 #2*c1*efts[inesgrid+1]-c0*efts[inesgrid])/c2
        inesgrid=inesgrid+1
        if spa_end_grd<inesgrid+2: sol_deq=False
    #efts,efts_nc=NormF(poi_grd,sst,sgrid,efts,normme) #efts_nc=efts_nc=normalization constant.
    #print('err',spa_itv[1],efts[spa_itv[1]])
    obv_err,oba_err=efts[spa_end_grd],abs(efts[spa_end_grd]) #obv_err=obtained value error. oba_err=obtained absolute error.
    #print('??',spa_sta_grd,efts[spa_sta_grd])
    #print('??',spa_itv[1]-1,efts[spa_itv[1]-1])
    #print('??',spa_itv[1],efts[spa_itv[1]])
    #print('??',spa_itv[1]+1,efts[spa_itv[1]+1])
    #print('??',spa_end_grd,efts[spa_end_grd])
    if oba_err<=etol:
        aee=True
        fun_ppd(dir_sav,sgrid,efts,poi_plt,fig_con,axs_tef,lin_tef,err_txt,'test energy = '+str(te/cf_meh2)+' eV','Error of = '+str(oba_err),'Eigen_Fun_Val_A_'+str(neef),aee)
        fun_wri(dir_sav+'Eigen_Fun_A_'+str(neef)+'.dat',[sgrid,efts],2,poi_grd,'w')
        efts=efts/cf_meh2
        #efts,new_nmc=NormF(poi_grd,sst,sgrid,efts,normme) #new_nmc=new normalization constant.
        #fun_wri(dir_sav+'Eigen_Fun_B_'+str(neef)+'.dat',[sgrid,efts],2,poi_grd,'w')
        fun_ppd(dir_sav,sgrid,efts,poi_plt,fig_con,axs_tef,lin_tef,err_txt,'test energy = '+str(te/cf_meh2)+' eV','Error of = '+str(oba_err),'Eigen_Fun_Val_B_'+str(neef),aee)
    return aee,obv_err

```

#PEN_HCt=Potential Energy Horizontal Cut.

#Variables:

```

#poi_grd=points grids. It contains the number of elements in the spatial and potential energy grid.
#pst_grd=point start grids.
#pfn_grd=points finish grids.
#sst=spatial step. float variable. It contains the value of the spatial step.
#sp_grd=spatial grid. float array dimension [nesgrid]. It contains the spatial distribution of points to resolve.
#pe_grd=potential energy grid. It contains the potential energy function evaluated alongside the spatial grid.
#ene_shf=energy shift. float variable. It contains the test energy to solve schrodinger equation for this iteration.
#ene_tol=energy tolerance. Float variable. It contains the tolerance to find the shifted potential energy roots.
#pe_fpa=potential energy parameters.
#pe_fun=potential energy function.

```

```

def PEN_HCt(poi_grd,pst_grd,pfn_grd,sp_grd,pe_grd,ene_shf): #,lim): #,ene_tol,pe_fpa,pe_fun):
    rot_fnd=[] #rot_fnd=root founds.
    npe_grd=pe_grd-ene_shf #npe_grd=new potential energy grid.
    rot_num=0 #rot_num=root number.
    rot_pos=[] #rot_pos=root positions.
    rot_typ=[] #rot_typ=root type. True for real roots and False for possible root intervals.
    pre_poi_sgn,act_poi_sgn=0,0 #pre_poi_sgn=previous points sign. act_poi_sgn=actual points sign.
    for i_poi_grd in range(pst_grd,pfn_grd-1,1): #i_poi_grd=i-counter poi_grd.
        if npe_grd[i_poi_grd]==0:
            rot_num=rot_num+1
            rot_pos.append(i_poi_grd)
            rot_typ.append(True)
        elif sign(npe_grd[i_poi_grd])!=sign(npe_grd[i_poi_grd+1]):
            rot_num=rot_num+1
            rot_pos.append(i_poi_grd)
            rot_typ.append(False)
    if npe_grd[pfn_grd]==0:
        rot_num=rot_num+1
        rot_pos.append(pfn_grd)
        rot_typ.append(True)
    #if 2*lim+2<=rot_pos[rot_num-1]-rot_pos[0]:
    #    rot_pos[0]=rot_pos[0]+lim
    #    rot_pos[1]=rot_pos[1]-lim
    if rot_num!=0:
        if rot_pos[rot_num-1]-rot_pos[0]==0:
            if rot_pos[rot_num-1]<poi_grd-2: rot_pos[rot_num-1]=rot_pos[rot_num-1]+2
            else: rot_pos[0]=rot_pos[0]-2
        elif rot_pos[rot_num-1]-rot_pos[0]==1:
            if rot_pos[rot_num-1]<poi_grd-3: rot_pos[rot_num-1]=rot_pos[rot_num-1]+1
            else: rot_pos[0]=rot_pos[0]-1
        return [rot_pos[0],rot_pos[rot_num-1]]
    else:
        if 0<=ene_shf: return [pst_grd,pfn_grd]
        else: return None

```

```

def QSE_NM_PT_B(sgrdh,nesgrid,sst,sgrid,pegrid,te,neef,fsplfv,etol,normme,cf_meh2,dir_sav,fig_con,axs_tef,lin_tef,err_txt,odd_sol,fun_ppd,fun_wri):
    aee=False #aee=acceptable eigen energy.
    gfgrid=2*te-pegrid #gfgrid=g function grid.
    efts=zeros(shape=(nesgrid),dtype=float,order='F') #eigen funtion for test energy.
    efts_lr=zeros(shape=(2,sgrdh+1),dtype=float,order='F') #eigen funtion for test energy left right.
    efts[0],efts[1],efts[-2],efts[-1]=fsplfv[0],fsplfv[1],fsplfv[2],fsplfv[3]
    efts_lr[0][0],efts_lr[0][1]=fsplfv[0],fsplfv[1]
    efts_lr[1][0],efts_lr[1][1]=fsplfv[2],fsplfv[3]
    sst2=sst**2 #sst2=sst square.
    for inesgrid in range(0,sgrdh-1,1): #inesgrid=index number elements spatial grid. (sgrdh+1)-2=sgrdh-1
        c0=1+sst2*gfgrid[inesgrid]/12 #c0=coefficient 0.

```

```

c1=1-5*sst2*gfgrid[inesgrid]/12 #c1=coefficient 1.
c2=1+sst2*gfgrid[inesgrid+2]/12 #c2=coefficient 2.
efts_lr[0][inesgrid+2]=(2*c1*efts_lr[0][inesgrid+1]-c0*efts_lr[0][inesgrid])/c2
#print(inesgrid+2,efts_lr[0][inesgrid+2])
efts[inesgrid+2]=efts_lr[0][inesgrid+2]
for inesgrid in range(0,sgrdh-1,1): #inesgrid=index number elements spatial grid. (sgrdh+1)-2=sgrdh-1
c0=1+sst2*gfgrid[inesgrid-inesgrid-1]/12 #c0=coefficient 0.
c1=1-5*sst2*gfgrid[inesgrid-inesgrid-1]/12 #c1=coefficient 1.
c2=1+sst2*gfgrid[inesgrid-(inesgrid+3)]/12 #c2=coefficient 2.
efts_lr[1][inesgrid+2]=(2*c1*efts_lr[1][inesgrid+1]-c0*efts_lr[1][inesgrid])/c2
efts[inesgrid-(inesgrid+3)]=efts_lr[1][inesgrid+2]
efts[sgrdh]=(efts_lr[0][sgrdh]+efts_lr[1][sgrdh])/2
efts,efts_nc=NormF(nesgrid,sst,sgrid,efts,normme) #efts_nc=efts_nc=normalization constant.
efts_lr[0]=efts_lr[0]/efts_nc
efts_lr[1]=efts_lr[1]/efts_nc
obt_err=efts_lr[0][sgrdh]-efts_lr[1][sgrdh] #obt_err=obtained error.
if abs(obt_err)<=etol and odd_sol==True:
aee=True
fun_ppd(dir_sav,sgrid,efts,fig_con,axs_tef,lin_tef,err_txt,'test energy = '+str(te)+' eV','Error of = '+str(abs(obt_err)), 'Eigen_Fun_Val_A_'+str(neef),aee)
fun_wri(dir_sav+'Eigen_Fun_A_'+str(neef)+'.dat',[sgrid,efts],2,nesgrid,'w')
efts=efts/cf_meh2
efts,new_nmc=NormF(nesgrid,sst,sgrid,efts,normme) #new_nmc=new normalization constant.
fun_wri(dir_sav+'Eigen_Fun_B_'+str(neef)+'.dat',[sgrid,efts],2,nesgrid,'w')
elif abs(obt_err)<=etol and odd_sol==False:
slp_ifp=(efts_lr[0][sgrdh-1]-efts_lr[0][sgrdh]-efts_lr[1][sgrdh]+efts_lr[1][sgrdh-1])/sst #slp_ifp=slope inflection point.
if slp_ifp<=etol:
aee=True
fun_ppd(dir_sav,sgrid,efts,fig_con,axs_tef,lin_tef,err_txt,'test energy = '+str(te)+' eV','Error of = '+str(abs(obt_err)), 'Eigen_Fun_Val_A_'+str(neef),aee)
fun_wri(dir_sav+'Eigen_Fun_A_'+str(neef)+'.dat',[sgrid,efts],2,nesgrid,'w')
efts=efts/cf_meh2
efts,new_nmc=NormF(nesgrid,sst,sgrid,efts,normme) #new_nmc=new normalization constant.
fun_wri(dir_sav+'Eigen_Fun_B_'+str(neef)+'.dat',[sgrid,efts],2,nesgrid,'w')
fun_ppd(dir_sav,sgrid,efts,fig_con,axs_tef,lin_tef,err_txt,'test energy = '+str(te)+' eV','Error of = '+str(abs(obt_err)), 'Eigen_Fun_Val_B_'+str(neef),aee)
return aee,obt_err

# End SGLE_Numerov_Solver functions.=====
# =====
# =====

```

Archivo para lectura y escritura resultados - SGLE_Read_Write.py

```

#####
##PyEART Module Read and Write -- Version 1.0 ##
##The funtions on this module are made to read and write data related with the simulation. ##
#####

#Rea_Fil_Txt=Read File Text.
#This function will open an text file and save its contents in a array.
#Parameters.
#txt_fil_nam=text file name. string parameter. This parameter indicates the directory and name of the file to be read.
def Rea_Fil_Txt(txt_fil_nam):
print(txt_fil_nam)
rea_txt_fil=open(file=txt_fil_nam,mode='r') #rea_txt_fil=read text file.
rea_dat_num_lin=0 #rea_dat_num_lin=read data number lines.
rea_dat_num_col=0 #rea_dat_num_col=read data number columns.
rea_dat=[None] #rea_dat=read data x.
for i_lin in rea_txt_fil: #i_lin=i-counter line.
row=rea_txt_fil.readline()
cells=row.split()
if rea_dat_num_lin==0:
rea_dat_num_col=len(cells)
rea_dat=[None]*rea_dat_num_col
for i_col in range(0,rea_dat_num_col,1): #i_col=i-counter column.
rea_dat[i_col]=[]
if len(cells)==rea_dat_num_col:
for i_col in range(0,rea_dat_num_col,1): #i_col=i-counter column.
rea_dat[i_col].append(cells[i_col])
rea_dat_num_lin=rea_dat_num_lin+1
else: break
rea_txt_fil.close()
return rea_dat

#Rep_Fil_Txt=Replace File Text.
#This function will open an text file and save its contents in a array.
#Parameters.
#txt_fil_nam=text file name. String parameter. This parameter indicates the directory and name of the file to be read.
#txt_lin_lab= String parameter. This parameter indicates the line which is going to be replaced.
#txt_lin_rep= String parameter. This parameter indicates the text which is going to be replaced.
def Rep_Fil_Txt(txt_fil_nam,txt_lin_lab,txt_lin_rep):
ori_txt_fil=open(file=txt_fil_nam,mode='r').read() #rep_txt_fil=replace text file.
rep_txt=ori_txt_fil.replace(txt_lin_lab,txt_lin_rep)
rep_ext_fil=open(file=txt_fil_nam,mode='w')
rep_ext_fil.write(rep_txt)
rep_ext_fil.close()
return

#Wri_Dat_Str_Txt=Write Data String Text.
#This function will write the data given in a text file.
#Parameters.
#txt_fil_nam=text file name. string parameter. This parameter indicates the directory and name of the file to be read.
#wri_dat_val=write data values. float array of [wri_dat_col_num][wri_dat_lin_num] dimention. This parameter has all the data to write.
#wri_dat_col_num=write data columns number. integer variable. This parameter indicates the number of columns in the data.
#wri_dat_lin_num=write data lines number. integer variable. This parameter indicates the number of rows in the data.
#wri_dat_mod=write data mode. string parameter. This parameter indicates the type of writing that will be perform.

```

```

def Wri_Dat_Str_Txt(txt_fil_nam,wri_dat_val,wri_dat_col_num,wri_dat_lin_num,wri_dat_mod):
    txt_fil=open(file=txt_fil_nam,mode=wri_dat_mod) #txt_fil=text file.
    for i_wri_dat_lin_num in range(0,wri_dat_lin_num,1): #i_wri_dat_lin_num=i-counter write data lines number.
        line=''
        #print(i_wri_dat_lin_num)
        for i_wri_dat_col_num in range(0,wri_dat_col_num,1): #i_wri_dat_col_num=i-counter write data columns number.
            #print(i_wri_dat_col_num)
            line=line+str(wri_dat_val[i_wri_dat_col_num][i_wri_dat_lin_num])+'\t'
        line=line+'\n'
        txt_fil.write(line)
    txt_fil.close()
    return

```

Archivo para graficar resultados - SGLE_Plotter.py

```

# =====
# =====
# Stacking Graphene Layers Eigenproblem - Plotter (SGLE_Plotter) Version == 1.0.0
# This module will plot the results given by SGLE_Main.
# =====
# =====

# =====
# =====
#Plot related functions.
# =====
# Start calling of needed libraries for SGLE_Plotter functions.

from numpy import zeros
from matplotlib.pyplot import figure,show,plot,xlabel,ylabel,title,legend,savefig

# End calling of needed libraries for SGLE_Plotter functions.
# =====
# =====

# =====
# =====
# SGLE_Plotter functions.=====
# Start SGLE_Plotter functions.

#SFP_2D=Simple Function Plot_2 Dimensions.
#iva=independent variable. array float dimension [n]. it contains the values of the independent variable to plot.
#dva=dependent variable. array float dimension [n]. it contains the values of the dependent variable to plot.
#ivaal=independent variable axis label. string variable. it contains the axis name of the independent variable to plot.
#dvaal=dependent variable axis label. string variable. it contains the axis name of the dependent variable to plot.
#plt=plot label. sting variable. it contains the label of the function to plot.
#pltt=plot title. string variable. it contains the title of the plot.
def SFP_2D(iva,dva,ivaal,dvaal,pltl,pltt,dir_sav):
    figure('Potential energy')
    plot(iva,dva,label=pltl)
    xlabel(ivaal)
    ylabel(dvaal)
    title(pltt)
    legend(loc=1)
    savefig(dir_sav+'Potential_Energy'+'.png')
    show()
    return

#Plt_Rsc_YAx=Plot Rescale Y Axis.
#This function will rescale the Y axis of a given plot.
#Parameters:
#plt_num=plot number. Integer parameter. This parameter indicates de number of individual lines (pairs of x,y values) in a given subplot.
#dat_sfi=data start end interval.
#plt_dat=plot data. Float array of [plt_axes][number of points to plot in plt_axes value] dimation.
#plt_axes=plot axes. Object axes. This parameter gives the matplotlib.pyplot.axes for plotting.
def Plt_Rsc_YAx(plt_num,dat_sfi,plt_dat,plt_axes):
    plt_min,plt_max=[],[] #plt_min=plot minimum. plt_max=plot maximum.
    plt_min_val,plt_max_va=0,0 #plt_min_val=plot minimum value. plt_max_va=plot maximum value.
    lim_dat=zeros(shape=(plt_num,dat_sfi[2]),dtype=float,order='F') #lim_dat=limit data.
    for i_plt_num in range(0,plt_num,1): #i_plt_num=i-counter plt_num.
        for i_dat_poi in range(dat_sfi[0],dat_sfi[1],1): #i_dat_poi=i-counter data points.
            lim_dat[i_plt_num][i_dat_poi-dat_sfi[0]]=plt_dat[i_plt_num][i_dat_poi]
            plt_min.append(min(lim_dat[i_plt_num]))
            plt_max.append(max(lim_dat[i_plt_num]))
    plt_min_val=min(plt_min)
    plt_max_val=max(plt_max)
    plt_axes.set_ylim(bottom=plt_min_val,top=plt_max_val)
    return plt_axes,[plt_min_val,plt_max_val]

#Plt_Ini=Plot Initializer.
#This function will initiate a plot to control progress.
#Parameters:
#itv_ivx=interval independent variable x.
#dat_ivx=data independent variable x.
#dat_tef=data test eigen function.
#dat_sfi=data start end interval.
def Plt_Ini(itv_ivx,dat_ivx,dat_tef,dat_sfi):
    plt_fig_con=figure('Eigen-Energies control panel') #plt_fig_con=plot figure control.
    plt_mea_grd=plt_fig_con.add_gridspec(nrows=1,ncols=1) #plt_mea_grd=plot measurement grid.

```