

AUTORIDADES UNIVERSITARIAS

RECTOR

INGENIERO MARIO ROBERTO NIETO LOVO

VICE-RECTORA ACADÉMICA

MAESTRA ANA MARÍA GLOWER DE ALVARADO

SECRETARIA GENERAL

DOCTORA ANA LETICIA DE AMAYA

FISCAL GENERAL

LICDO. FRANCISCO CRUZ LETONA

FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE

DECANO

LICDO. RAÚL ERNESTO AZCUNAGA LÓPEZ

VICE-DECANO

ING. WILLIAM VIRGILIO ZAMORA GIRÓN

SECRETARIO

LICDO. VÍCTOR HUGO MERINO QUEZADA

JEFA DEL DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA

ING. Ms. SORAYA LISSETTE BARRERA DE GARCÍA

COORDINADOR GENERAL DEL PROCESO DE GRADUACIÓN

ING. MAURICIO ERNESTO GARCÍA EGUIZÁBAL

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



**DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE SEGUIMIENTO
INFORMÁTICO, PARA LOS GRADUADOS(AS) DE POSTGRADO DE LA UNIVERSIDAD DE
EL SALVADOR FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE.**

PRESENTADO POR:

MENA RECINOS, VERÓNICA LIZET.
RAMOS SALAZAR, HÉCTOR MANUEL.
RIVERA DÍAZ, BENJAMÍN ERNESTO.

PARA PODER OPTAR AL GRADO DE:
INGENIERO DE SISTEMAS INFORMÁTICOS.

DOCENTE DIRECTOR:
ING. CARLOS STANLEY LINARES PAULA.

SANTA ANA, SEPTIEMBRE DE 2012.

ÍNDICE.

CONTENIDO	
CAPÍTULO 1.....	7
1.1 INTRODUCCIÓN.....	7
1.2 OBJETIVOS.....	8
1.2.1 OBJETIVO GENERAL.....	8
1.2.2 OBJETIVOS ESPECÍFICOS.....	8
1.3 ANTECEDENTES.....	9
1.4 PLANTEAMIENTO DEL PROBLEMA.....	11
1.5 JUSTIFICACIÓN.....	13
1.6 ALCANCES.....	15
1.7 LIMITANTES.....	16
CAPÍTULO 2.....	17
2.1 MARCO TEÓRICO.....	17
2.2 METODOLOGÍA DE TRABAJO.....	18
2.2.1 FASE DE PLANIFICACIÓN Y RECOPIACIÓN DE INFORMACIÓN.....	18
2.2.2 FASE DE DISEÑO.....	19
2.2.3 FASE DE DESARROLLO.....	19
2.2.4 FASE DE IMPLEMENTACIÓN.....	19
2.3 TÉCNICAS DE INVESTIGACIÓN.....	19
2.3.1 ENCUESTA.....	20
2.3.2 INTERPRETACIÓN DE RESULTADOS.....	21
2.3.3 CONCLUSIONES DE LOS ANÁLISIS DE LOS RESULTADOS.....	26
2.4 SITUACIÓN ACTUAL.....	27
2.4.1 RESULTADO DE ENTREVISTA AL PERSONAL Y JEFATURA DE UNIDAD DE POSTGRADO...	27
2.5 PLANTEAMIENTO DE LA SOLUCIÓN.....	29
CAPÍTULO 3.....	30
3.1 DEFINICIÓN DE REQUERIMIENTOS DEL SISTEMA.....	30
3.1.1 REQUERIMIENTOS DE DESARROLLO.....	30
3.1.1.1 SOFTWARE.....	30
3.1.1.1.1 ASP.NET. ⁶	31
3.1.1.1.2 PHP. ⁷	32
3.1.1.1.3 JAVA. ¹⁰	33
3.1.1.1.4 AJAX. ¹¹	35
3.1.1.1.5 JQUERY. ¹²	36
3.1.1.1.6 JAVASCRIPT. ¹⁴	36
3.1.1.1.7 SERVIDOR HTTP GLASSFISH. ¹⁵	37
3.1.1.1.8 BASE DE DATOS POSTGRESQL. ¹⁶	37
3.1.1.1.9 PRIMEFACES. ¹⁷	38
3.1.1.1.10 HERRAMIENTA DE CONTROL DE VERSIONES GIT. ¹⁸	38
3.1.1.1.11 REQUERIMIENTOS DE DESARROLLO FINALES.....	39
3.1.1.2 HARDWARE.....	40
3.1.1.2.1 REQUISITOS DEL SISTEMA PARA JAVA EE 6. ¹⁹	40

3.1.1.2.2 REQUISITOS DEL SISTEMA PARA GIMP. ²⁰	42
3.1.1.2.3 REQUISITOS DEL SISTEMA PARA NETBEANS. ²¹	42
3.1.1.2.4 REQUISITOS DEL SISTEMA PARA POSTGRESQL. ²²	43
3.1.1.3 USUARIOS.....	44
3.1.1.3.1 USUARIOS FINALES.....	44
3.1.1.3.2 USUARIOS INTERMEDIOS.....	44
3.1.1.3.3 ADMINISTRADORES.....	45
3.1.1.3.4 DESARROLLADORES.....	45
3.2 ANÁLISIS DE FACTIBILIDAD DEL PROYECTO.....	46
3.2.1 FACTIBILIDAD TÉCNICA.....	46
3.2.2 FACTIBILIDAD ECONÓMICA.....	48
3.2.2.1 COSTO DEL PERSONAL.....	48
3.2.2.2 COSTO DE DEPRECIACIÓN.....	48
3.2.2.3 COSTO DE ENERGÍA ELÉCTRICA.....	50
3.2.2.4 OTROS GASTOS.....	51
3.2.3 FACTIBILIDAD OPERACIONAL.....	52
3.2.3.1 ANALISTAS PROGRAMADORES.....	52
3.2.3.2 DOCENTE DIRECTOR.....	52
3.2.3.3 USUARIOS DEL SISTEMA INFORMÁTICO.....	53
3.3 DISEÑO LÓGICO DE SISTEMA.....	53
3.3.1 DIAGRAMAS DE CASO DE USO.....	53
3.4 DISEÑO DE INTERFACES.....	61
3.4.1 INTERFAZ DE ENTRADA DE DATOS.....	61
3.4.2 INTERFAZ DE SALIDA DE DATOS.....	62
3.5 DISEÑO DE BASE DE DATOS.....	63
3.5.1 DESCRIPCIÓN DE LA BASE DE DATOS.....	63
3.5.2 MODELO ENTIDAD-RELACIÓN.....	66
3.5.3 MODELO FÍSICO DE BASE DE DATOS.....	67
3.5.4 DICCIONARIO DE DATOS.....	68
3.6 REQUERIMIENTOS DE SEGURIDAD DEL SISTEMA.....	79
3.6.1 ESPECIFICACIONES DE SEGURIDAD DEL SISTEMA.....	79
3.6.1.1 ENCRIPCIÓN MD5.....	79
3.6.1.2 GENERACIÓN DE CONTRASEÑAS CON GUID. ³¹	80
3.6.1.3 MANEJO DE SESIONES. ³²	80
3.6.1.4 MANEJO DE ROLES POR GLASSFISH.....	81
3.6.1.5 ACCESO CONTROLADO POR LA UNIDAD DE POSTGRADOS.....	81
3.6.2 SEGURIDAD DEL SERVIDOR.....	82
3.6.2.1 EL PROTOCOLO HTTPS.....	82
CAPÍTULO 4.....	83
4.1 OBJETIVOS DEL PLAN DE IMPLEMENTACIÓN.....	83
4.2 PLAN DE IMPLEMENTACIÓN.....	83
4.2.1 ACTIVIDADES DEL PLAN DE IMPLEMENTACIÓN.....	84
4.2.2 DESCRIPCIÓN DE ACTIVIDADES.....	85
4.2.3 CRONOGRAMA DEL PLAN DE IMPLEMENTACIÓN.....	86
4.3 COSTO DE LA IMPLEMENTACIÓN.....	87
CAPÍTULO 5.....	88
5.1 MANUAL DEL USUARIO DEL SISTEMA DE CONTROL DE SEGUIMIENTO INFORMÁTICO, PARA LOS GRADUADOS(AS) DE POSTGRADO (SICSGUP).....	88

5.1.1 GENERALIDADES.....	88
5.2 FUNCIONALIDAD DEL SISTEMA.....	88
5.2.1 MODULO PARA EL USUARIO FINAL.....	88
5.2.1.2 ACCESO AL SISTEMA.....	89
5.2.1.3 CREACIÓN DE CURRÍCULO.....	89
5.2.1.3.1 AGREGAR DATOS PERSONALES.....	90
5.2.1.3.2 AGREGAR LOS ESTUDIOS REALIZADOS.....	91
5.2.1.3.3 AGREGAR SU EXPERIENCIA LABORAL.....	92
5.2.1.4 AGREGAR FOTOGRAFÍA DE PERFIL.....	93
5.2.1.5 PAGINA PRINCIPAL DE PERFIL.....	93
5.2.1.6 VISTA DE UN PERFIL.....	95
5.2.1.7 VER TODOS LOS CONTACTOS.....	95
5.2.1.8 MENSAJES PRIVADOS.....	96
5.2.2 MODULO PARA EL ADMINISTRADOR.....	97
5.2.2.1 PANTALLA PRINCIPAL DEL ADMINISTRADOR.....	97
5.2.2.2 VER TODOS LOS POST HECHOS POR LOS USUARIO.....	98
5.2.2.3 REPORTES Y GRÁFICOS.....	98
5.2.2.4 NOTICIAS.....	99
5.2.2.5 OFERTAS DE TRABAJO.....	100
5.2.2.5.1 BÚSQUEDA DE CANDIDATOS.....	100
5.3 MANUAL DEL ADMINISTRADOR.....	101
CAPÍTULO 6.....	126
6.1 CONCLUSIONES.....	126
6.2 RECOMENDACIONES.....	127
REFERENCIAS BIBLIOGRÁFICAS.....	128
ANEXOS.....	129
ANEXO A – DESARROLLO CAPITULAR.....	129
ANEXO B – ENTREVISTA A PERSONAL ADMINISTRATIVO DE LA UNIDAD DE POSTGRADO DE LA UNIVERSIDAD DE EL SALVADOR FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE.....	131
ANEXO C - DIAGRAMA DE FLUJO DE DATOS (ALMACENAMIENTO DE EXPEDIENTE).....	134
ANEXO D - DIAGRAMA DE FLUJO DE DATOS (MANEJO DE OFERTAS DE TRABAJO).....	135
ANEXO E- DIAGRAMA DE FLUJO DE DATOS (ELABORACIÓN DE INFORMES).....	136
ANEXO F- UNIVERSIDADES SONDEADAS-SITIOS WEB OFICIALES.....	137
ANEXO G- CRONOGRAMA.....	138
ANEXO H- ESPECIFICACIONES PARA EL SERVIDOR DE LA UNIDAD DE POSTGRADOS.....	139
HP PROLIANT ML150 G6 E5504 2.0GHZ QUAD CORE HOT PLUG SAS/SATA TOWER SERVER.....	139
SERVICIO WEB HOSTING.....	140
MAQUINA DE ESCRITORIO VOSTRO 470 DE DELL.....	141
ANEXO I- SCRIPT DE LA BASE DE DATOS.....	142

GENERALIDADES DEL PROYECTO.

CAPÍTULO 1.

1.1 INTRODUCCIÓN.

El trabajo de Graduación es uno de los pilares fundamentales en el proceso de formación de los estudiantes de Ingeniería de Sistemas Informáticos de la UES-FMO, pues permite poner en práctica los conocimientos que se adquirieron a lo largo del desarrollo de la carrera, y garantizar que todas las áreas de conocimiento se encuentran integradas.

Parte fundamental de este trabajo de graduación es la de ofrecer la utilidad del mismo, que sea pertinente, necesario y parte vital de alguna institución, para solventar problemas o una necesidad que esta tenga; Es por esa razón que se ha buscado beneficiar con el presente trabajo de graduación a la Unidad de Postgrados de La Universidad de El Salvador, Facultad Multidisciplinaria de Occidente.

Actualmente la excelencia en la enseñanza y la calidad de los profesionales graduados(as) de Postgrados de la Universidad de El Salvador, Facultad Multidisciplinaria de Occidente es muy conocida, por lo que es común que empresas nacionales soliciten a la universidad la información sobre estos profesionales para cubrir ofertas de trabajo, las cuales resultan en oportunidades perdidas por falta de información de estos; llevándonos al punto de la importancia que el seguimiento de graduados(as) de postgrado llega a tener y de la comunicación que debe existir entre estos y la Unidad de Postgrado de la Universidad. Al igual de importante la comunicación entre los mismos profesionales quienes pueden resultar ser fuente de información sobre ofertas de trabajo externas, sin mencionar la necesidad de la comunicación común necesaria para las actividades académicas que realizan los estudiantes que cursan alguna carrera de Postgrado.

El presente texto expone los objetivos, el planteamiento del problema, los alcances y limitaciones de un tema que ha sido poco explotado hasta el momento y que de ser implementado permitirá la Unidad de Postgrados de La Universidad de El Salvador, Facultad Multidisciplinaria de Occidente la oportunidad de colocación de sus profesionales en el ámbito laboral nacional y un mejor control de la información de sus actuales estudiantes, además de la metodología de trabajo e investigaciones iniciales para el desarrollo de el presente proyecto.

1.2 OBJETIVOS.

1.2.1 OBJETIVO GENERAL.

- ✓ Dotar de un Sistema de Control de Seguimiento Informático para sus Graduados(as), que facilite una comunicación y un contacto profesional con los mencionados, permitiendo dar respuesta a ofertas de trabajo, a la Unidad de Postgrados de La Universidad de El Salvador, Facultad Multidisciplinaria de Occidente.

1.2.2 OBJETIVOS ESPECÍFICOS.

- ✓ Desarrollar un sistema de modo que su uso sea factible para el personal de la Unidad de Postgrados, y que logre beneficiar tanto a la Unidad de Postgrado como a sus estudiantes y los Graduados(as).
- ✓ Garantizar la seguridad del medio, de modo que solo los estudiantes, Graduados(as) y el personal administrativo de la Unidad de Postgrado tengan acceso a los módulos que les corresponden mediante niveles de seguridad personalizados.
- ✓ Garantizar la integridad de la información obtenida mediante informes.
- ✓ Garantizar una forma más exacta y precisa de obtener información como resultado de las búsquedas personalizadas sobre los datos de graduados(as), permitiendo dar respuesta a las ofertas de trabajo que las empresas dirigen a la Unidad de Postgrado.

1.3 ANTECEDENTES.

Actualmente en El Salvador para los profesionales tener una maestría además de un título universitario, ha tomado un auge importante pues les abre las puertas a posibles ascensos laborales o a optar por mejores trabajos en empresas nacionales e internacionales con oficinas en el país; En la década de los noventa optar por una maestría presentaba diversas dificultades para los profesionales ya que no todas las instituciones educativas a nivel nacional podían ofrecer este servicio, agregando a este inconveniente que el costo de una maestría era más alto que hoy; Para un profesional de Santa Ana optar por una maestría de alguna universidad en San Salvador por ejemplo, podía representar un problema debido a horarios laborales y gastos de viaje; Hoy en día La Universidad de El Salvador como otras Universidades en el país ofrece al público profesional la oportunidad de especializarse, actualmente la Facultad Multidisciplinaria de Occidente brinda este servicio desde el año 2005 cuando se creó la Escuela de Postgrados, ofreciendo al público de la Zona Occidental 10 carreras en diversas ramas profesionales como la salud, las finanzas, la docencia, el medio ambiente, entre otras. De esta forma la Universidad de El Salvador se pone a la vanguardia con la calidad de los profesionales que ofrece al mercado laboral del país.

Acudiendo a la importancia que ha tomado la educación en Postgrados por parte de la Universidad de El Salvador FMO, se tiene como finalidad de este proyecto de tesis el desarrollo e implementación de un sistema de seguimiento para graduados(as) de postgrado, que facilite la comunicación entre los estudiantes y la escuela de postgrado así como también el contacto profesional con los estudiantes graduados(as).

Con la finalidad de conocer cuántas universidades en el país realizan un seguimiento a sus graduados(as), se han visitado los sitios web oficiales de varias universidades, y hemos llegado a la conclusión de que actualmente son pocas las universidades a nivel nacional que ofrecen en su sitio un link específico para graduados(as), los sitios sondeados pertenecen a las siguientes universidades (Ver Anexo E):

Universidad	Posee seguimiento a graduados(as).
Universidad Albert Einstein (UAE)	No
Universidad Centroamericana José Simeón Cañas (UCA)	No
Universidad Capitán General Gerardo Barrios (UCGB)	No
Universidad Evangélica de El Salvador (UEES)	Si
Universidad Salvadoreña Alberto Masferrer (USAM)	Si
Universidad Tecnológica de El Salvador (UTEC)	Si
Universidad Francisco Gavidia (UFG)	No
Universidad Autónoma de Santa Ana (UNASA)	No
Universidad Católica de El Salvador (UNICAES)	No
Universidad Don Bosco (UDB)	No

Lista de Universidades sondeadas sobre el seguimiento a Graduados(as)- Estudio Propio ¹

La Universidad Centroamericana José Simeón Cañas (UCA) ofrece en su sitio web una sección de búsqueda de empleo personalizado, una sección para información sobre postgrados y diplomados además de la bolsa de trabajo común. La Universidad Don Bosco (UDB) y la Universidad Capitán General Gerardo Barrios (UCGB) solo manejan una bolsa de trabajo e información de postgrados y diplomados.

Actualmente el seguimiento a graduados(as) solamente es realizado por tres universidades en sus sitios web oficiales: **Universidad Evangélica de El Salvador, Universidad Salvadoreña Alberto Masferrer y Universidad Tecnológica de El Salvador.**

¹ Sondeo realizado en las Páginas Web oficiales de cada Universidad de la lista. - 21 de Marzo de 2012.

1.4 PLANTEAMIENTO DEL PROBLEMA.

La Unidad de Postgrados de la Facultad Multidisciplinaria de Occidente, tiene como misión brindar a sus alumnos una formación académica de acuerdo a los retos de una sociedad dinámica, procurando que logren una evolución tanto personal como profesional coherente a los cambios tecnológicos y sociales. Ofreciendo servicios educativos de calidad, cuyo producto sean profesionales formados con alta calidad que aporten sus conocimientos al desarrollo sostenible e integrarse al campo laboral, haciendo real el aporte al avance del país.

La Unidad de Postgrados de la Facultad padece de la falta de un sistema que se encargue de un seguimiento a graduados(as), lo que dificulta la comunicación entre esta Unidad y las personas graduadas, además esto produce una dificultad para retroalimentarse sobre la información de los graduados(as), para conocer si las expectativas de sus métodos de enseñanza rinden en el mercado laboral nacional, carece de una forma de establecer informes estadísticos más específicos y detallados al final de los periodos de clases, que les permita por ejemplo hacer proyecciones futuras o evaluaciones sobre posibles cambios en sus planes de estudio para lograr que estos se apeguen a las demandas laborales de hoy.

Es un hecho que la unidad puede ser un medio de conexión profesional muy eficaz entre los graduados(as) y las empresas nacionales, ya que las empresas hacen llegar a la unidad ofertas de trabajo que actualmente son oportunidades perdidas, para todos aquellos graduados(as) que por alguna razón no tengan trabajo o tengan un trabajo que no les de la satisfacción profesional que desean; todo esto como resultado de la ausencia de comunicación entre la unidad y los antes mencionados.

El hecho de realizar un informe general sobre el número de personas graduadas por especialidad resulta en la tediosa búsqueda manual de información en los archivos físicos de la Unidad y la actualización manual de la información en el archivo digital que se utiliza para ello, lo que requiere de tiempo y causa un cuello de botella para las operaciones de la asistente administrativa que es la persona encargada de la mayoría de procesos que se realizan en la Escuela de Postgrados, Por lo que cualquier toma de decisión en base a la información de algún informe tomará tiempo. A esto agregamos que cuando se debe saber datos de la población graduada, por ejemplo: cuantos

trabajan en instituciones públicas o privadas, número de graduados(as) por departamentos del país, o número de graduados(as) que trabajan fuera de El Salvador, no se cuenta con los medios para obtener esta información.

Por todo lo anterior se genera la necesidad de una herramienta que resuelva todos estos inconvenientes a la hora de llevar un seguimiento de graduados(as); Se pretende que este Sistema se implemente al inicio en el Servidor de el Departamento de Ingeniería y Arquitectura a nivel de la Unidad de Postgrados y permitir que a futuro también pueda implementarse para la Facultad completa.

1.5 JUSTIFICACIÓN.

La Unidad de Postgrados de la Universidad de El Salvador, Facultad Multidisciplinaria de Occidente, en la actualidad no cuenta con ningún sistema informático para el manejo de la información y seguimiento profesional adecuado para poder realizar contactos profesionales entre sus graduados(as) y empresas que pudieran necesitar de personal capacitado.

Actualmente, se cuenta con la tecnología de red necesaria para poder implementar un sistema informático que cumpla con los requerimientos de la Unidad de Postgrados, pero no existe ningún sistema que se adapte a lo que la Universidad necesita, sin intervenir en sus políticas.

El seguimiento a los graduados(as) es de suma importancia para la Unidad de Postgrados, ya que, como se menciona anteriormente, la retroalimentación que esta puede brindar permitiría a la unidad conocer el impacto social que su metodología de enseñanza logra en el mercado laboral de hoy. De ahí surge la necesidad que el seguimiento a los graduados(as) sea consistente a través del tiempo, y que les permita mantener la información de forma segura y confiable.

La investigación para desarrollar un plan de seguimiento informático, surge de la necesidad de sistematizar el control de los estudiantes graduados(as) de las diferentes carreras que se imparten en la Unidad de postgrados. Al igual, el Ministerio de Educación como ente rectora de la educación exige a todas las Universidades que están en proceso de acreditación contar con un control de alumnos egresados y graduados(as) de acuerdo a los requisitos mediante una Comisión de Acreditación de la Calidad de la Educación Superior (CDA) que evalúa a la institución².

Con el desarrollo de un sistema de seguimiento a graduados(as) se beneficiarían:

- ❖ La Unidad de Postgrados de la Facultad Multidisciplinaria de Occidente. Con el desarrollo de un sistema de seguimiento a graduados(as) se cumplirá con un requisito que el Ministerio de Educación exige para acreditar la calidad de la institución. Además, permitirá tener un control sobre los alumnos matriculados en las maestrías y al mismo tiempo tener un contacto con las personas graduadas de dicha unidad.

- ❖ A los graduados(as). Con el desarrollo de un sistema de seguimiento los alumnos podrán continuar conociendo información que ofrece la Unidad de Postgrados (nuevos cursos, reuniones, etc.), además de poder ser candidatos a ofertas de trabajo que lleguen a la Unidad de postgrados.

1.6 ALCANCES.

- ✓ El sistema permitirá que la Unidad de Postgrados cuente con información actualizada de sus profesionales graduados(as).
- ✓ El sistema se desarrollará según las necesidades y requerimientos de la Unidad de Postgrados de la Universidad de El Salvador, Facultad Multidisciplinaria de Occidente, de tal manera, que permita posterior el crecimiento de éste, para que pueda ser implementado a nivel de la Facultad en un futuro.
- ✓ La Implementación del Sistema de control de seguimiento informático, para los graduados(as) de postgrado de la Universidad de El Salvador Facultad Multidisciplinaria de Occidente se llevará a cabo en el Servidor del Departamento de Ingeniería y Arquitectura.
- ✓ El Sistema cubrirá los siguientes procesos fundamentales:
 - ❖ Manejo de perfil profesional y Currículo Vitae de los estudiantes y graduados(as) de la unidad de postgrado.
 - ❖ Manejo de Contactos profesionales.
 - ❖ Manejo de Ofertas de trabajo.
 - ❖ Búsquedas de datos sobre graduados(as), basadas en criterios específicos o generales.
 - ❖ Manejo de propuestas automatizadas de perfiles finales para ofertas de trabajo.
 - ❖ Manejo de información de Empresas interesadas en la contratación de profesionales.
 - ❖ Seguimiento de profesionales (lugar de trabajo, sector en el trabajan, etc.).
 - ❖ Creación de grupos, programación de actividades académicas entre usuarios.
 - ❖ Manejo de Informes.
- ✓ Se capacitará al personal administrativo de la Unidad de Postgrados de la Universidad que fueren encargados de la administración y mantenimiento del Sistema informático.

1.7 LIMITANTES.

- ✓ Disponibilidad de tiempo que dedique el personal de la Unidad de Postgrados de la Universidad de El Salvador FMO, para resolver las dudas que se presenten durante la recopilación de información y el desarrollo del Sistema Informático.

METODOLOGÍA DE INVESTIGACIÓN.

CAPÍTULO 2.

2.1 MARCO TEÓRICO.

Los programas de posgrado dieron inicio en el año 2005 siendo parte de la coordinación de los proyectos académicos especiales. Iniciando en ese período con la primera generación de Maestría en Profesionalización de la Docencia Superior, Maestría en Métodos y Técnicas de Investigación, y la Maestría en Administración Financiera.

En Enero de 2006, según acuerdo del Consejo Superior Universitario No. 015-2005-2007 (IV-2.4), tomado en sesión Ordinaria celebrada el día 06 de abril de 2006, fue creada la Unidad de Posgrado. Teniendo como dependencia jerárquica el DECANATO de la Facultad Multidisciplinaria de Occidente. En el año 2010 se presentó el proyecto que permitió convertir a la Unidad de Posgrado en la primer Escuela de Posgrado de la Universidad de El Salvador; aunque lamentablemente a la fecha por situaciones de planificación, presupuesto, e inclusive falta de atención a la importancia que esta escuela debería de tener se adolece de grandes falencias entre ellas: la administrativa, financiera y también la poca o nula posibilidad de poder incorporar nuevas tecnologías al servicio de los diferentes programas que se administran.

Hasta la fecha se cuenta con 10 programas de posgrados autorizados para la Facultad Multidisciplinaria de Occidente:

- ✓ Especialidad Médica de Cirugía General.
- ✓ Especialidad Médica de Ginecología.
- ✓ Especialidad Médica de Medicina Interna.
- ✓ Especialidad Médica de Medicina Pediátrica.
- ✓ Maestría en Administración Financiera.
- ✓ Maestría en Consultoría Empresarial.
- ✓ Maestría en Gestión Ambiental.
- ✓ Maestría en Métodos y Técnicas de Investigación Social.
- ✓ Maestría en Profesionalización de la Docencia Superior.
- ✓ Maestría en Salud Pública.

Durante el periodo de 2006- 2011 se han graduado los siguientes profesionales:

- ✓ Maestría en Salud Pública (35 profesionales egresados de la primera generación)
- ✓ Maestría en Administración Financiera (15 graduados(as) de la primera generación, 24 de la segunda en proceso de graduación y 22 de la tercera en proceso de inscripción de trabajos de grado).
- ✓ Maestría en Profesionalización de la Docencia Superior (18 profesionales graduados(as) de la segunda generación, 13 profesionales de la tercera).
- ✓ Maestría en Métodos y Técnicas de Investigación Social (29 profesionales graduados(as) de la primera generación)

Tal y como se expresa en párrafos anteriores la administración de la Escuela de Posgrado recae en esquemas muy similares a cuando se inició como unidad de posgrado; es decir, un jefe (con atribuciones y responsabilidades de jefatura pero con responsabilidad académica a tiempo completo en el departamento académico del cual proviene), una secretaria que atiende toda la parte administrativa de los 10 programas en mención y una persona de servicios de los cuales la única figura contratada de manera oficial es la plaza de Jefatura de la Unidad siendo las otras dos eventuales.

2.2 METODOLOGÍA DE TRABAJO.

En el presente proyecto se realizará una investigación de los procedimientos para lograr obtener una solución satisfactoria al problema establecido. Las fases a realizar en esta investigación se describen a continuación:

2.2.1 FASE DE PLANIFICACIÓN Y RECOPIACIÓN DE INFORMACIÓN.

En esta fase se estableció la organización del grupo y se establecieron los aspectos más importantes a evaluar de la investigación.

Se recopiló información por medio de investigación bibliográfica, entrevistas, encuestas, investigación en la Web entre otros; estableciendo el problema y la posible solución de este.

2.2.2 FASE DE DISEÑO.

En esta fase se llevo a cabo el diseño diversos aspectos del Sistema, como por ejemplo: módulos a desarrollar, requerimientos del sistema, costo de sistema, niveles de permisos, acceso de usuarios, diseño de base de datos, etc.

2.2.3 FASE DE DESARROLLO.

En esta fase se desarrollo el sistema con los módulos, la base de datos y las herramientas que requiera el sistema establecidos en la fase anterior.

2.2.4 FASE DE IMPLEMENTACIÓN.

En esta fase se desarrollo un plan de implementación para el Sistema ya terminado, además de establecer el costo de esta.

2.3 TÉCNICAS DE INVESTIGACIÓN.

✓ Técnica e instrumentos de recolección de datos.

❖ Entrevista.

En un primer momento se entrevisto al Jefe de la Unidad de Posgrado, Ing. Med. Mauricio García Eguizábal, para conocer su perspectiva ante este proyecto; al igual, se le realizó una entrevista a la asistente administrativa que es la persona encargada de las operaciones comunes de la Unidad y finalmente se abordó a los Coordinadores de algunas maestrías para conocer sus inquietudes, sugerencias y aportes para nuestro proyecto, siempre utilizando como técnica la entrevista.

❖ Encuesta.

Dirigida a estudiantes activos de maestrías en curso, para conocer sus opiniones sobre la calidad y medios de comunicación que se utilizan para difundir la información que la Unidad de Postgrados brinda sobre cursos, pagos, reincorporaciones, etc.

❖ **Consultas Bibliográficas:**

La fuente de información para la investigación fueron libros, revistas, tesis que se encuentran en la biblioteca o hemeroteca de la Universidad de El Salvador FMO, y de las cuales se dejará una referencia bibliográfica en el documento finalizado.

❖ **Consultas Web:**

Se recolecto información a lo largo del desarrollo del presente proyecto ya sea para consultar libros digitales, sondeos de sitios web universitarios, investigaciones sobre las herramientas a utilizar para el desarrollo del sistema entre otros.

2.3.1 ENCUESTA.

Dirigida a estudiantes activos de maestrías que están actualmente cursando su último año, para conocer sus opiniones de la forma en como utilizan la tecnología para sus contactos profesionales y académicos.

Las preguntas fueron redactadas en forma cerrada en su mayoría, para obtener datos exactos y una pregunta abierta para facilitar la exposición de los ideales y visiones de los profesionales que podría ayudar a realizar un sistema con mayores alcances.

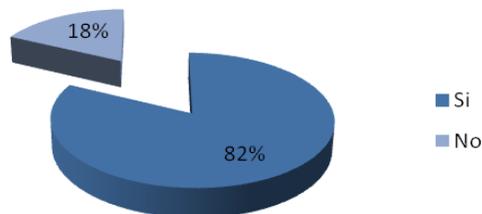
Número de encuestados = 41 Alumnos.

Que es el total de estudiantes que se encuentran cursando su último año de estudio.

2.3.2 INTERPRETACIÓN DE RESULTADOS.

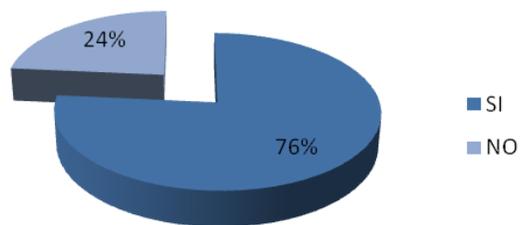
1. ¿Considera usted que al finalizar sus estudios de Postgrado las redes sociales le servirán para mantener contactos profesionales?

El 82 % de los estudiantes de postgrados opinan que las redes sociales podrían ayudar a mantener contactos profesionales, mientras el 18 % opinan que las redes sociales no son una herramienta para mantener contactos profesionales una vez finalizados los estudios de postgrado.



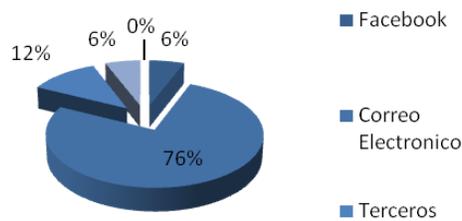
2. ¿Considera usted que las redes sociales que actualmente utiliza hace que ocupe más tiempo del necesario?

El 76.47 % de los estudiantes de postgrados opinan las redes sociales que actualmente utilizan hacen que se utilice más tiempo que el que podría ser necesario, siendo así un factor de distracción.



3. ¿A través de qué manera se entera de las últimas noticias de la Unidad de postgrados?

El medio de comunicación más utilizado para publicar noticias, información, etc., es el correo electrónico, mientras la información que es colocada en los diversos lugares de la Universidad tiene un porcentaje de 0.00% de efectividad para ser difundida. Las redes sociales tienen un impacto similar a las visitas que hacen los estudiantes a la Unidad de Postgrados, tomando el 12% el hecho de que los estudiantes de postgrado quienes difunden la información en las redes sociales ya que la Unidad de postgrado no cuenta con cuentas en las redes sociales.

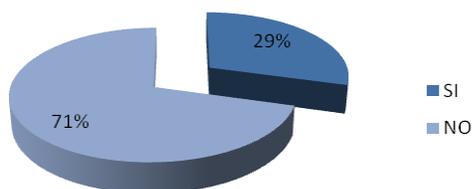


4. ¿Considera que la forma en la que actualmente la unidad de postgrado comparte la información es la adecuada?

Actualmente, poco más de un 70% considera que la forma en la que se difunde la información no es la adecuada, siendo por medio de correo electrónico, el medio por el cuál la mayoría de estudiantes conocen la información se podría explotar alguna otra herramienta que sea de utilidad y en la que la mayoría se encuentren familiarizadas.

Sí: 29.4%

No: 70.60%



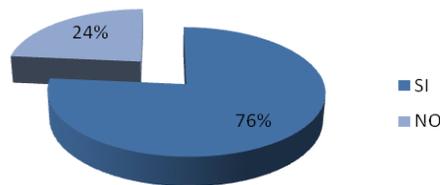
5. ¿Considera usted que una red social especializada para la unidad de postgrado pueda beneficiarle para obtener contactos profesionales y oportunidades laborales?

La tendencia actual que las redes sociales son una herramienta informática que la unidad de postgrados pudiera aprovechar, ya que, más de $\frac{3}{4}$ considera que una red social especializada para la unidad podría traer beneficios, tanto para obtener contactos y oportunidades laborales.

A la vez, buscando un equilibrio entre las noticias y la distracción que a su vez podría causar (véase pregunta 2).

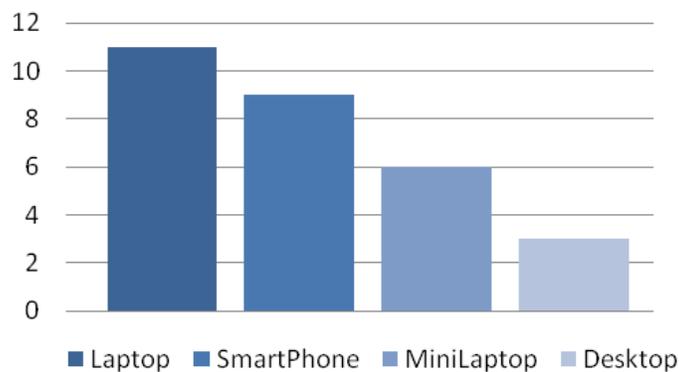
Sí: 76.47%

No: 23.53%



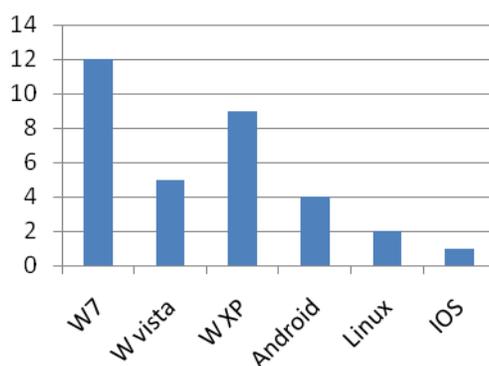
6. ¿Qué Dispositivos utiliza regularmente para navegar por la web?

La utilización de laptops y Smartphone, actualmente se han convertido en una herramienta de uso diario por su portabilidad y su uso cotidiano, mostrando también que la población está a favor de la tecnología y abiertos a los cambios. Mientras el uso de Tablet, muestra una utilización de 0.00%.



7. Seleccione los sistemas operativos que utiliza

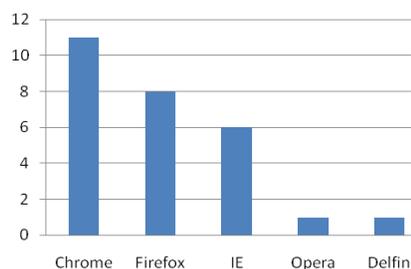
El conocimiento de los sistemas operativos que actualmente utilizan, son un marco referente para determinar la aceptación a los cambios tecnológicos que se dan día a día. Los estudiantes de la Unidad de Postgrados, muestran una gran aceptación a los avances tecnológicos, ya que 12 de 17 utilizan Windows 7, 9 utilizan Windows XP, es decir, la diferencia entre XP y Windows 7 sólo son 3 personas aproximadamente; Mientras que los sistemas operativos de Linux y IOS no son muy utilizados entre los estudiantes de maestrías.



8. ¿Cuáles son los navegadores de su preferencia?

La utilización de navegadores es fundamental en el desarrollo de aplicaciones web, ya que no todos los navegadores web interpretan de igual forma los estándares provistos por la W3C³. (Véase anexo B).

Se muestra una gran aceptación a los navegadores Chrome, Firefox e Internet Explorer. Siendo ellos los navegadores que respetan más los estándares (Internet Explorer desde la versión 9), y muestran según estudios propios compatibilidades bastante aceptables a HTML5⁴ y CSS3⁵ (véase anexo C).



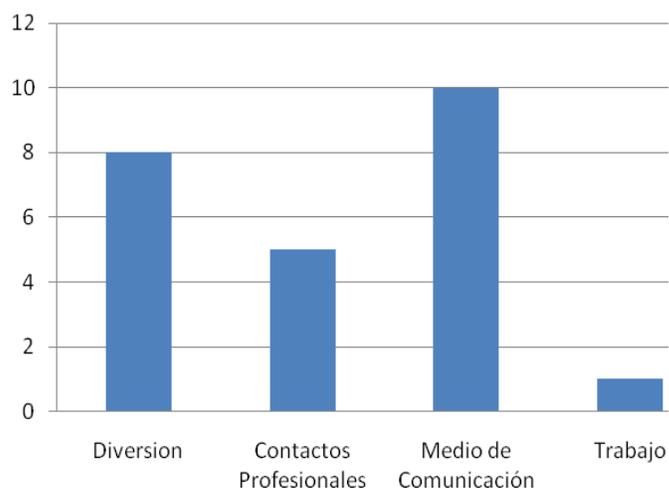
3 Sitio oficial normas W3C para entornos Web - <http://www.w3.org/>

4 Testeo online de navegadores Web para HTML5 - <http://www.html5test.com>

5 Testeo online de navegadores Web para CSS3 - <http://www.css3test.com>

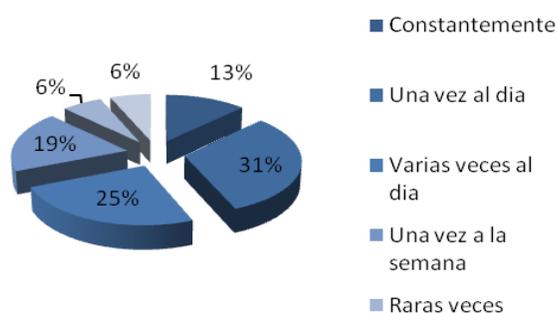
9. ¿Con qué objeto utiliza usted las redes sociales?

Actualmente las redes sociales son consideradas por la mayoría como un medio de comunicación práctico, y en forma secundaria como un medio utilizado para diversión.



10. ¿Con qué frecuencia visita usted las redes sociales?

El 13% de los estudiantes nos confirmaron que visitan las redes sociales en forma constante durante el día, mientras que el 31% solo lo visita una vez al día, el 25% acceden varias veces al día al tener oportunidad, el 19% solo acceden una vez a la semana, el 6% accesa rara vez y el 6% no respondió a la pregunta.



2.3.3 CONCLUSIONES DE LOS ANÁLISIS DE LOS RESULTADOS.

- ✓ Los estudiantes de último año de la Unidad de Postgrados, muestran una gran aceptación a los cambios y a la actualización tecnológica.

- ✓ Los estudiantes de la Unidad de Postgrados utilizan como medio de comunicación primario el correo electrónico para comunicarse entre ellos; al igual, muestran inconformidad con el medio de comunicación actual (tanto entre ellos, como con los medios de información que la unidad de postgrados ofrece).

- ✓ La aceptación tecnológica y la tendencia a la utilización de redes sociales muestran que podría ser un medio de comunicación estable y aceptable entre los estudiantes de último año de las maestrías impartidas por la Unidad de Postgrados. Dicha aceptación permitiría poder dar un seguimiento a los estudiantes una vez graduados(as), así como mantener contacto con ellos para informarle sobre ofertas de trabajo, obtener contactos profesionales, nuevos cursos y demás información que la Unidad de Postgrados puede ofrecer.

2.4 SITUACIÓN ACTUAL

2.4.1 RESULTADO DE ENTREVISTA AL PERSONAL Y JEFATURA DE UNIDAD DE POSTGRADO.

A grandes rasgos, existe una desinformación generalizada sobre los graduados(as) de la Unidad de postgrados, ya que no se conoce el impacto social que ellos causan en la sociedad y/o en el ámbito laboral, ya que hasta el momento, no ha sido posible contactar con un número significativo de ellos. Los coordinadores se muestran dispuestos a hacer uso de herramientas tecnológicas que les faciliten la promoción de las ofertas de Posgrados en esta modalidad, pero concuerdan que para llegar a tomarla realmente en serio deberá de ser sencilla.

Para conocer la situación actual se llevaron a cabo entrevistas con la asistente administrativa, algunos coordinadores de carreras y la jefatura de la Unidad de Postgrado, con lo que se logro conocer los siguientes puntos esenciales:

- ✓ El expediente que la Unidad de Postgrado almacena de los estudiantes es en forma física y contiene la hoja de vida, papeleo que estos solicitan a la Unidad, copias de recibos de pagos mensuales para consultar y verificar en el sistema de cobro de colecturía, la correspondencia que se les envía, entre otros; El expediente de notas es administrado por el departamento de Administración Académica (ver anexo B).
- ✓ No existe actualmente un medio para comunicar en forma masiva al público sobre los cursos, promociones de maestrías y congresos; Por lo que los avisos, las posibles ofertas de trabajo para estudiantes que aun cursan maestría y correspondencias propias de la Unidad de Postgrados para los estudiantes les son comunicados por medio de correos electrónicos, llamadas telefónicas y últimamente por Facebook; procesos que son realizadas por la asistente administrativa y que conlleva tiempo y organización pues debe considerarse que actualmente hay un aproximado de 134 estudiantes de diferentes especialidades.

- ✓ Actualmente no se realiza un seguimiento a graduados(as) por parte de la Unidad de Postgrados por lo que no hay forma de saber el impacto social que los graduados(as) tienen en el mercado laboral, esto para mantener o mejorar la calidad de la metodología de enseñanza y planes de estudio actuales.
- ✓ Las ofertas de trabajo que empresas nacionales hacen llegar a la Unidad de Postgrado por medio de los coordinadores de carreras o por los mismos estudiantes que actúan a veces como medios de comunicación entre las empresas y los coordinadores son comunicadas a la asistente administrativa, quien a su vez las comunica a la comunidad estudiantil por los medios mencionados en los párrafos anteriores. Pero si la oferta de trabajo requiere de un profesional graduado la comunicación de esta queda pendiente pues no se lleva seguimiento de graduados(as) como se mencionó anteriormente (Ver anexo C).
- ✓ La Unidad de Postgrados elabora informes al final de cada ciclo académico, solo maneja un estándar que es el número de alumnos por especialidad internos o externos, este se realiza en formato Excel; modificando y actualizando el archivo que se creó desde el inicio de labores de la Unidad de Postgrados sin guardar un historial digital; hasta el momento no se pueden realizar informes estadísticos más detallados.
- ✓ Para elaborar los informes la asistente administrativa realiza una búsqueda manual en los archivos físicos de los expedientes, para luego modificar y actualizar la información también manualmente en el archivo digital de Excel (Ver anexo D).

2.5 PLANTEAMIENTO DE LA SOLUCIÓN.

Debido a que no existe un sistema informático que permita realizar un seguimiento de graduados(as) y la comunicación entre la Unidad de postgrados y los actuales alumnos se propone:

Una aplicación que permita manejar datos sobre seguimiento de graduados(as), comunicación entre la unidad y los estudiantes, manejo de ofertas de trabajo y generación de reportes por medio de un modulo administrativo que permitirá al encargado del mantenimiento: agregar, modificar, eliminar y consultar datos sobre cualquiera de las acciones mencionadas.

La aplicación contará con un modulo para el usuario Intermedio donde podrá buscar candidatos para ofertas de trabajos mediante filtros personalizados de características requeridas, dando origen a un listado de profesionales que cumplan con los requisitos; también le permitirá realizar actividades como elaboración de informes, comunicación de avisos, publicitar cursos, diplomados o promociones en maestrías, entre otros.

La aplicación contará con un modulo para el acceso del usuario final que le permitirá revisar información académica, comunicación directa con la Unidad de postgrado, formar grupos, participar en discusiones entre otras actividades.

La administración de diferentes tipos de usuarios requiere el desarrollo de diferentes tipos de interfaces que no manejarán los mismos privilegios.

DISEÑO Y DESARROLLO.

CAPÍTULO 3.

3.1 DEFINICIÓN DE REQUERIMIENTOS DEL SISTEMA.

3.1.1 REQUERIMIENTOS DE DESARROLLO.

3.1.1.1 SOFTWARE.

En este proyecto de graduación se desarrollo una Plataforma tecnológica que permita a la Unidad de Postgrados el establecimiento de contactos profesionales entre empresas y graduados, la comunicación directa con la comunidad estudiantil y con los graduados(as), ofrecer masivamente las carreras que actualmente oferta, realizar consultas y obtener los resultados en segundos además de poder contar con informes estadísticos específicos cuando estos sean necesarios. Esta plataforma debe permitir ser utilizada a cualquier hora, tanto dentro de la Universidad como fuera de ella, lo que por medio de una conexión a internet fácilmente puede hacerse mediante una aplicación Web.

La aplicación Web será desarrollada mediante lenguajes de programación que no produzcan inconvenientes y permitan manejar una aplicación de tamaño grande sin problemas de ninguna índole, para determinar el lenguaje se han tomado como base para una investigación los tres lenguajes de programación web del lado del servidor mas utilizados hoy en día, luego de establecer ventajas y desventajas de cada uno se seleccionará el que para nuestra aplicación sea la mejor opción.

Lenguajes evaluados:

- ✓ ASP.Net.
- ✓ PHP.
- ✓ JAVA.

3.1.1.1.1 ASP.NET. ⁶

Microsoft introdujo la tecnología llamada Active Server Pages en diciembre de 1996. Es parte del Internet Information Server (IIS) desde la versión 3.0 y es una tecnología de páginas activas que permite el uso de diferentes scripts y componentes en conjunto con el tradicional HTML para mostrar páginas generadas dinámicamente. ASP.NET es más reciente la versión 1.0 fue liberado el 5 de enero de 2002 como parte de la versión 1.0 del .NET Framework, es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

Ventajas.

- ✓ Los controles son cargados desde el servidor (Textbox, panels, dropdownlist, etc.).
- ✓ Puede detectar el tipo de navegador que hace la petición para evaluar la versión de HTML que este soporta.
- ✓ Es liviano.
- ✓ Realiza conexiones con una base de datos fácilmente.
- ✓ Se puede utilizar en cualquier computadora que esta conectada a la red y tenga un navegador.
- ✓ Es un lenguaje fácil de utilizar en la programación.

Desventajas.

- ✓ Todas las peticiones onclick, onchange, onleave reciben peticiones al lado del servidor lo que pone en riesgo la seguridad.
- ✓ No es de licencia sin costo, es un producto con licencia de Microsoft.

⁶ ASP.NET <http://www.slideshare.net/Helmilpa/estudio-comparativo-de-php-aspnet-y-java> 21 Mayo de 2012.

- ✓ En el desarrollo con ASP tradicional utilizamos lenguajes de scripting no tipeados como VBScript o JScript. Podemos instalar otros motores de scripting que impongan verificación de tipos; sin embargo, no son universalmente conocidos o utilizados como los anteriores.
- ✓ Trabaja con el Servidor Web IIS de Windows, no puede trabajar con otro tipo de servidores Web como por ejemplo Apache.

3.1.1.1.2 PHP.⁷

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Resmas Lerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group, publicado bajo la PHP License, considerada esta licencia como software libre⁸, puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

PHP es un lenguaje de programación interpretado (Lenguaje de alto rendimiento), diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

Ventajas.

- ✓ Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- ✓ El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.

⁷ PHP <http://www.slideshare.net/Helmilpa/estudio-comparativo-de-php-aspnet-y-java> 21 Mayo 2012.

⁸ Free Software Foundation <http://www.fsf.org/> 21 de Mayo de 2012.

- ✓ Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- ✓ Posee una amplia documentación en el sitio web oficial de PHP⁹.
- ✓ Su Licencia es sin costo.
- ✓ Permite la programación orientada a objetos.

Desventajas.

- ✓ La programación orientada a objetos es deficiente si la aplicación es demasiado grande.
- ✓ Como es un lenguaje que se interpreta en ejecución, para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado.
- ✓ Debido a que es un lenguaje interpretado, un script en PHP que es manejado en el servidor solamente, suele funcionar considerablemente más lento cuando las solicitudes al servidor aumentan.

3.1.1.1.3 JAVA.¹⁰

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. La memoria es gestionada mediante un recolector de basura.

⁹ Sitio Oficial de PHP <http://www.php.net/> 21 Mayo 2012.

¹⁰ JAVA <http://www.oracle.com/technetwork/java/index.html> 21 Mayo de 2012.

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrolladas por Sun Microsystems en 1995. En la actualidad la documentación y las herramientas de desarrollo son distribuidas por Oracle Corporation.

Ventajas.

- ✓ La herramienta JDK tienen licencia sin costo.
- ✓ Cuenta con el soporte de Oracle.
- ✓ Las actualizaciones de sus versiones se realizan cada 6 meses para mantener la seguridad.
- ✓ Es independiente del sistema operativo en el que se desarrolle, porque plataforma puede tomarse como IDE.
- ✓ Flexibilidad en la programación orientada a objetos.
- ✓ Fácil acceso a bases de datos independientemente de la plataforma utilizada.
- ✓ Se ha desarrollado una versión para el mercado de dispositivos electrónico como teléfonos móviles.
- ✓ Índice de productividad elevado.

Desventajas.

- ✓ Para manejo a bajo nivel debe utilizarse métodos nativos para evitar problemas de portabilidad.

Después de establecer las ventajas y desventajas de cada uno de estos lenguajes, podemos descartar dos de ellos, empezando por ASP.NET, debido a que su licencia es de paga y representaría un coste para la universidad; También se descarta el uso de PHP ya que la cantidad de información que maneje el sistema y las peticiones que pueda tener el servidor puede hacer que el sistema funcione lentamente o que deje de funcionar en determinados periodos de tiempo.

Finalmente seleccionamos el lenguaje de programación JAVA que permite el buen funcionamiento de aplicaciones grandes, su Licencia no es de paga y cuenta con mucha documentación en la Web entre otras característica.

Otros Lenguajes de programación que se utilizarán:

3.1.1.1.4 AJAX.¹¹

Ajax, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas, estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano; De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Es necesario que las maquinas cuenten con un navegador que permita desplegar correctamente la información que se manejará con AJAX para lo que se han evaluado los navegadores que permitan este lenguaje y los que no, debe tenerse en cuenta que ésta es una lista general, y el soporte de las aplicaciones Ajax dependerá de las características que el navegador permita.

Navegadores que permiten el funcionamiento de AJAX:

- ✓ Navegadores como Mozilla, Mozilla Firefox, SeaMonkey, Camino, K-Meleon, IceWeasel, Flock, Epiphany, Galeon y Netscape versión 7.1 y superiores; Google Chrome de Google, Safari de Apple, Midori y Chromium.
- ✓ Microsoft Internet Explorer para Windows versión 5.0 y superiores, y los navegadores basados en él.
- ✓ Navegadores con el API KHTML versión 3.2 y superiores implementado, incluyendo Konqueror versión 3.2 y superiores y el Web Browser para S60 de Nokia tercera generación y posteriores.
- ✓ Opera versión 8.0 y superiores, incluyendo Opera Mobile Browser versión 8.0 y superiores.

¹¹ AJAX <http://www.w3schools.com/ajax/default.asp> 21 Mayo 2012.

3.1.1.1.5 JQUERY.¹²

jQuery es una biblioteca de componentes para el framework jQuery que le añaden un conjunto de plug-ins, widgets, comportamiento y efectos visuales para la creación de aplicaciones web.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT¹³ y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

- ✓ Es compatible con los navegadores Internet Explorer 6.0, Mozilla Firefox 3, Safari 3.1, Ópera 7.0 en adelante, Google Chrome y sus versiones posteriores.

3.1.1.1.6 JAVASCRIPT.¹⁴

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Para evitar estas incompatibilidades entre navegadores, el World Wide Web Consortium diseñó el estándar Document Object Model (DOM, o Modelo de Objetos del Documento en español), que incorporan Konqueror, las versiones 6 de Internet Explorer y Netscape Navigator, Opera la versión 7, Mozilla Application Suite y Mozilla Firefox desde su primera versión.

12 JQUERY <http://jqueryui.com> 21 Mayo 2012.

13 Licencia MIT: "Se autoriza, de forma gratuita, a cualquier persona que obtenga una copia de este software y archivos de documentación asociados..." <http://www.opensource.org/licenses/mit-license.php> 21 Mayo 2012.

14 JAVASCRIPT <http://www.w3schools.com/js/> 21 Mayo 2012

3.1.1.1.7 SERVIDOR HTTP GLASSFISH. ¹⁵

GlassFish es un servidor de aplicaciones de software libre desarrollado por Sun Microsystems, compañía adquirida por Oracle Corporation, que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. La versión comercial es denominada Oracle GlassFish Enterprise Server (antes Sun GlassFish Enterprise Server). Es gratuito y de código libre, se distribuye bajo un licenciamiento dual a través de la licencia CDDL (Licencia Común de Desarrollo y Distribución) y la GNU GPL (Licencia Pública General).

GlassFish está basado en el código fuente donado por Sun y Oracle Corporation, éste último proporcionó el módulo de persistencia TopLink o API Java Persistence. GlassFish tiene como base al servidor Sun Java System Application Server de Oracle Corporation, un derivado de Apache Tomcat, y que usa un componente adicional llamado Grizzly que usa Java NIO para escalabilidad y velocidad.

3.1.1.1.8 BASE DE DATOS POSTGRESQL. ¹⁶

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD (licencia de software libre permisiva permite el uso del código fuente en software no libre). Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group). PostgreSQL ha tenido una larga evolución, la cual se inicia en 1982 con el proyecto Ingres en la Universidad de Berkeley. Este proyecto, liderado por Michael Stonebraker, fue uno de los primeros intentos en implementar un motor de base de datos relacional.

¹⁵ GlassFish <http://glassfish.java.net/es/> 01 de Junio 2012.

¹⁶ POSTGRESQL <http://www.postgresql.org/> 01 Junio 2012.

Después de haber trabajado un largo tiempo en Ingres y de haber tenido una experiencia comercial con él mismo, Michael decidió volver a la Universidad en 1985 para trabajar en un nuevo proyecto sobre la experiencia de Ingres, dicho proyecto fue llamado post-ingres o simplemente POSTGRES.

El proyecto post-ingres pretendía resolver los problemas con el modelo de base de datos relacional que habían sido aclarados a comienzos de los años 1980. El principal de estos problemas era la incapacidad del modelo relacional de comprender "tipos", es decir, combinaciones de datos simples que conforman una única unidad. Actualmente estos son llamados objetos. Se esforzaron en introducir la menor cantidad posible de funcionalidades para completar el soporte de tipos. Estas funcionalidades incluían la habilidad de definir tipos, pero también la habilidad de describir relaciones - las cuales hasta ese momento eran ampliamente utilizadas pero mantenidas completamente por el usuario. En Postgres la base de datos «comprendía» las relaciones y podía obtener información de tablas relacionadas utilizando reglas. Postgres usó muchas ideas de Ingres pero no su código.

3.1.1.1.9 PRIMEFACES.¹⁷

PrimeFaces es un código abierto implementación de JavaServer Faces (JSF), que cuenta con el Ajax en sus componentes del lado cliente y que facilitan la creación de las aplicaciones web.

3.1.1.1.10 HERRAMIENTA DE CONTROL DE VERSIONES GIT.¹⁸

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

Esto le permite ver el estado anterior del archivo, y comparar los cambios en el tiempo, ver si se modificó algo que podría estar causando problemas, qué es lo produjo el problema y cuando.

¹⁷ Prime Faces <http://primefaces.org/> 01 Junio 2012.

¹⁸ GIT Control de Versiones <http://git-scm.com/>, <http://git-scm.com/book/en/Getting-Started-About-Version-Control> 01 Junio 2012.

3.1.1.1.11 REQUERIMIENTOS DE DESARROLLO FINALES.

Como lenguajes de programación para el desarrollo del sistema:

- ✓ Java Enterprise Edition 6.
- ✓ CSS versión 2 y 3.
- ✓ PrimeFaces.
- ✓ HTML versiones 4 y 5.
- ✓ JQuery.
- ✓ Ajax.
- ✓ JavaScript.

Servidor Http:

- ✓ GlassFish.

Herramientas de Desarrollo:

- ✓ Netbeans.
- ✓ Notepad++.
- ✓ PgAdminIII.

Herramientas para el diseño:

- ✓ Gimp.

Herramientas de Control de Versión:

- ✓ Git.

Herramientas de Testing:

- ✓ JUnit.

Base de Datos:

- ✓ PostgreSQL.

3.1.1.2 HARDWARE.

Los requisitos mínimos de hardware para que la aplicación sea funcional son los siguientes:

3.1.1.2.1 REQUISITOS DEL SISTEMA PARA JAVA EE 6.¹⁹

En Windows:

Plataforma	Versión	Memoria	Navegadores	Espacio en disco
Windows (32 bits)				
Windows x86	Windows 7	128 MB	Chrome IE 7, IE 8, IE 9	98 MB
	Windows Vista	128 MB	Firefox 3.6+, Firefox 4	
	Windows Server 2008	128 MB		
	Windows XP	64 MB		
Windows de 64 bits				
Windows x64 Modo de 64 bits	Windows 7	128 MB	• SO de 64 bits, navegadores de 32 bits: Chrome IE 7, IE 8, IE 9 Firefox 3.6+, Firefox 4	98 MB
	Windows Server 2008	128 MB		
	Windows Vista	128 MB	• SO de 64 bits, navegadores de 64 bits: IE 7, IE 8, IE 9 Firefox 4	
	Windows XP	128 MB		

¹⁹ Requisitos del Sistema para Java 6 (aplica para Java 7) <http://www.java.com/es/download/help/sysreq.xml> -- 21 de Mayo de 2012.

En Linux:

Plataforma	Versión	Memoria	Navegadores	Espacio en disco
Linux (32 bits)				
Linux x86	Oracle Enterprise Linux 5.5+	64 MB	Todos los SO compatibles con estas versiones de Firefox. Firefox 3.6+, Firefox 4	58 MB
	Red Hat Enterprise Linux 6.x	64 MB		
	Red Hat Enterprise Linux 5.5+	64 MB		
	SUSE 10*	64 MB		
	SLES 11*	64 MB		
	Linux (64 bits)			
Linux x64 Modo de 64 bits	Oracle Enterprise Linux 6	64 MB	Todos los SO compatibles con estas versiones de Firefox. Firefox 3.6+, Firefox 4	58 MB
	Oracle Enterprise Linux 5.5+	64 MB		
	Red Hat Enterprise Linux 6.x	64 MB		
	Red Hat Enterprise Linux 5.5+	64 MB		
	SLES 11*	64 MB		
	SLES 10*	64 MB		

* Sin certificación en Oracle VM

3.1.1.2.2 REQUISITOS DEL SISTEMA PARA GIMP. ²⁰

GIMP se ejecuta en la mayoría de sistemas Unix con X11 (X Windows System). Si el sistema puede ejecutar un navegador web grafico, lo más probable es que también puede ejecutar GIMP. Funciona incluso en PC's relativamente antiguos (Pentium 200) o en estaciones de trabajo de edad (dom Ultra1). La cantidad de memoria puede ser un factor importante, especialmente si usted tiene la intención de trabajar en imágenes de gran tamaño. Un mínimo de 128 MB de RAM es recomendable, pero en realidad usted debe tener 256 MB o más.

3.1.1.2.3 REQUISITOS DEL SISTEMA PARA NETBEANS. ²¹

El IDE NetBeans 7.1.2 funciona en sistemas operativos que soportan el Java VM (Virtual Machine), sus requerimientos son los siguientes:

Configuraciones mínimas de hardware
Microsoft Windows XP Professional SP3/Vista SP1/Windows 7 Professional. Procesador: 800 MHz Intel Pentium III o equivalente. Memoria: 512 MB. Espacio en disco: 750 MB de espacio libre en disco.
Ubuntu 9.10. Procesador: 800 MHz Intel Pentium III o equivalente. Memoria: 512 MB. Espacio en disco: 650 MB de espacio libre en disco.

Configuraciones de hardware recomendadas
Microsoft Windows XP Professional SP3/Vista SP1/Windows 7 Professional. Procesador: 2.6 GHz Intel Pentium IV o equivalente. Memoria: 2 GB. Espacio en disco: 1 GB de espacio libre en disco.
Ubuntu 11.04. Procesador: 2.6 GHz Intel Pentium IV o equivalente. Memoria: 2 GB. Espacio en disco: 850 MB de espacio libre en disco.

²⁰ GIMP www.gimp.org/unix/ 01 Junio 2012.

²¹ Requisitos NetBeans <http://netbeans.org/community/releases/71/relnotes.html> 01 Junio 2012

Nota: La resolución de pantalla mínima de IDE es de 1024x768 píxeles.

NetBeans IDE también se ejecutan en las diferentes distribuciones de Linux como Red Hat Enterprise Linux, Fedora 14, y muchos otros.

Netbeans incluye los servidores GlassFish Server Open Source Edition 3.1, y Apache Tomcat 7.0.11.

Servidores de aplicaciones probados en Netbeans:

- ✓ Servidor GlassFish OSE (Open Source Edition) 3.0.1.
- ✓ GlassFish Enterprise Server v2.1.1.
- ✓ Sun Java System Application Server PE 8.2.
- ✓ WebLogic 11g (10.3.3.0).
- ✓ Tomcat 6.0.26.
- ✓ Tomcat 5.5.
- ✓ JBoss 5.0.
- ✓ JBoss 4.2.3 (J2EE 1.4).

3.1.1.2.4 REQUISITOS DEL SISTEMA PARA POSTGRESQL. ²²

No hay requerimientos del sistema específicos para PostgreSQL, lo mínimo con lo que debe contar un equipo es:

- ✓ Espacio en disco duro 70 MB mínimo.
- ✓ Memoria 256 MB.

3.1.1.3 USUARIOS.

El sistema será diseñado de tal forma que maneje diferentes interfaces, las que formarán un ambiente amigable y cómodo para los diferentes tipos de usuarios que deberán utilizarlo; Los usuarios estarán clasificados como:

- ✓ Usuarios Finales.
- ✓ Usuarios Intermedios.
- ✓ Administradores.
- ✓ Diseñadores/Desarrolladores.

3.1.1.3.1 USUARIOS FINALES.

Los usuarios finales del sistema serán los estudiantes y graduados(as) de la Unidad de Postgrado de la Universidad, quienes harán uso del sistema para crear perfiles, currículos en línea, formar grupos con los demás estudiantes o graduados(as), optar a los empleos ofertados, entre otros. Se considera que los usuarios finales deben tener conocimientos básicos en el manejo de navegadores Web como Firefox de Mozilla, Internet Explorer y Google Chrome.

3.1.1.3.2 USUARIOS INTERMEDIOS.

Los Usuarios Intermedios son las personas que trabajan en la Unidad de Postgrado de la Universidad, como por ejemplo los docentes de materias impartidas en las maestrías, estos usuarios tendrán las mismas opciones que los usuarios finales pero serán capaces de administrar grupos y aceptar a sus miembros.

Se considera que los usuarios finales deben tener conocimientos sobre el manejo de navegadores Web como Firefox de Mozilla, Internet Explorer y Google Chrome.

3.1.1.3.3 ADMINISTRADORES.

Los usuarios administradores serán seleccionados por la Unidad de Postgrados para la administración de la aplicación; tendrá todos los privilegios en el sistema, podrá comunicar al público de forma masiva información sobre diplomados y carreras, comunicar a los estudiantes y graduados(as) sobre avisos, ofertas de empleo que las empresas comuniquen a la Unidad, consultar datos, crear reportes, administrar usuarios y privilegios, dar mantenimiento general de la aplicación.

Se considera que los usuarios finales deben tener conocimientos sobre el sistema operativo en el que trabajara el sistema, manejo de navegadores como Firefox de Mozilla, Internet Explorer y Google Chrome, además de los conocimientos del sistema de control de seguimiento de graduados(as).

3.1.1.3.4 DESARROLLADORES.

Los Desarrolladores son los encargados del diseño, desarrollo, depuración e implementación del sistema informático en sí; Los conocimientos que deben tener:

- ✓ Conocimientos del Sistema operativo en el que se trabajará el sistema.
- ✓ Conocimientos en el uso de navegadores como Firefox de Mozilla, Internet Explorer y Google Chrome.
- ✓ Conocimientos de programación en los lenguajes Java, Ajax, JavaScript, JQuery, HTML, CSS etc.
- ✓ Conocimientos en el manejo de Base de Datos PostgreSQL.

3.2 ANÁLISIS DE FACTIBILIDAD DEL PROYECTO.

El término “factibilidad” se refiere a la disponibilidad de los recursos necesarios para llevar a cabo los objetivos o metas señalados.

3.2.1 FACTIBILIDAD TÉCNICA.

Para el desarrollo del sistema se cuenta con el equipo y la tecnología siguiente:

Equipo	Características
Computadora de Escritorio	Pentium D CPU 3.00 GHz 1.50 GHz. 1 GB de RAM DDR2. 512 MB de video GForce 4800 PCI Express.
Laptop	Toshiba Satellite T135-S1305WH. Processor Name: Intel Pentium U4100 Processor Core 2 Duo 1.3 GHz. Memoria RAM: 3 GB.
Mini laptop	Compaq CQ10-500. Procesador Intel® Atom™ CPU N455 @ 1.66 GHz 1.67 GHz. Memoria DDR2 RAM 1.00 GB. Tipo de Sistema: Sistema Operativo 32 bits.
Laptop	HP LA G41350. Procesador: AMD Dual-Core E-450 (1,65 GHz). Memoria: DDR3 de 2 GB (1 x 2 GB). Unidad interna: SATA de 500 GB (5400 rpm). Sistema operativo de 64 bits.
Tecnología a Utilizar.	
Entorno de Desarrollo	GlassFish versión 3.1.2. Netbeans versión 7.1.2. Notepad++. PgAdminIII versión 1.12.3. GIMP versión 2.6. Hibernate versión 4.1.3. JSF versión 2.0 PostgreSQL 9.1.3. JPA versión 2.0. PrimeFaces ultima versión estable. IceFaces ultima versión estable. JUnit versión 2.0. Git versión 1.7.5.4.
Conexión a Internet	Inalámbrico y Modem (1GB).

Además, para la fase de implementación del proyecto, se cuenta con el hardware y tecnología siguiente en la Unidad de Postgrados de la Universidad de El Salvador FMO:

Tecnología/Equipo.	Características.
3 Maquinas de Escritorio.	Windows XP Professional 5.1 español. Vostro 220 serie. Procesador Intel(R) Core(TM)2 Duo CPU E8400 @3.00 GHZ (2CPUS). Memoria 2012 MB RAM. Adaptador de pantalla Intel(R) G45/G43 Express Chipset. Memoria Interna 1024 MB
Tipos de Navegadores	Firefox Mozilla 12. Google Chrome 13.0.772.0. Internet Explorer 7.0.5730.13.
Internet	Si.

Las características del servidor del Departamento de Ingeniería y Arquitectura son las siguientes:

Tecnología/Equipo.	Características.
Servidor	Hp Proliant 150 ml. Procesador: 2 Microprocesadores Intel Xeon 3.00 GHZ 2 núcleos. Memoria 8GB de RAM.
Servicios	Soporte para Java EE 6. Servidor http Glassfish. Servidor PostgreSQL.

3.2.2 FACTIBILIDAD ECONÓMICA.

En este apartado se detallara que tan factible es el proyecto tomando como referencia los puntos detallados a continuación:

3.2.2.1 COSTO DEL PERSONAL.

El coste de personal representa la remuneración del esfuerzo humano que se incorpora al proceso de elaboración del producto o de prestación de servicios.

El trabajo de tesis se desarrollo durante 6 meses, tres meses para el diseño y tres meses más para la programación, para calcular el coste de personal se realizo un sondeo a los empleos que se ofertan en uno de los sitios web nacionales más utilizado en el país: CompuTrabajo²³; Para programadores Java el sueldo que se ofrece oscila entre \$750.00 a \$ 1,200.00; en muchas ocasiones son empleos por proyectos.

Con este dato podemos establecer un costo de personal promedio por proyecto para el trabajo de desarrollo de la etapa de diseño y programación del sistema; Tomando en consideración que no se trabajaban jornadas de 8 horas y que cuando se programó algunas horas eran ocupadas para documentación sobre los lenguajes de programación.

- ✓ Para la etapa de diseño durante tres meses el costo de personal será de \$ 1,200.00, (Etapa realizada por una persona).
- ✓ Para la etapa de programación durante los últimos tres meses el costo de personal del sistema es de \$ 3,000.00 (Etapa desarrollada por tres personas durante 3 meses).

Dando un total de costo de personal total de **\$ 4,200.00 (A)**, el cual será absorbido por el trabajo de graduación en curso y no representará para la Universidad y menos para la Unidad de Postgrados ningún tipo de desembolso.

3.2.2.2 COSTO DE DEPRECIACIÓN.

Según el Art. 30 capitulo IV de la Ley de Impuesto sobre la Renta²⁴; un equipo puede depreciarse un 20% anual.

23 CompuTrabajo <http://www.sv.computrabajo.com/bt-ofertas.htm> 22 Mayo 2012.

24 Ley de Impuesto sobre la Renta

<http://elsalvador.eregulations.org/media/ley%20de%20impuesto%20sobre%20la%20renta.pdf> 22 Mayo 2012.

Calculando Depreciación de equipo.

1 Mini laptop \$400.00, Con un año de depreciación.

$$\$400.00 \times 0.20 = \$ 80.00, \$ 400.00 - \$ 80.00 = \$ \mathbf{320.00}.$$

2 Laptops \$ 1,050.00 (\$ 500.00 y \$ 550.00). Una de ellas con un año de depreciación.

$$\$ 550.00 \times 0.20 = \$110.00, \$ 550.00 - \$ 110.00 = \$ \mathbf{440.00}.$$

Laptop de \$ 500.00 fue adquirida este año.

1 Maquina de Escritorio \$ 500.00 con tres años de depreciación.

$$\$ 500.00 \times 0.20 = \$ 100.00, \$ 500.00 - \$ 100.00 = \$ 400.00$$

$$\$ 400.00 \times 0.20 = \$ 80.00, \$ 400.00 - \$ 80.00 = \$ 320.00.$$

$$\$ 320.00 \times 0.20 = \$ 64.00, \$ 320.00 - \$ 64.00 = \$ \mathbf{256.00}.$$

Total de depreciación:

Equipo	Valor al inicio de proyecto	Depreciación	Meses de trabajo	Total x Maquina
Laptop 1	\$ 500.00	\$ 100.00	6	\$ 49.98
Laptop 2	\$ 440.00	\$ 88.00	6	\$ 43.98
Mini Laptop 1	\$ 320.00	\$ 64.00	6	\$ 31.98
Maquina Escritorio	\$ 256.00	\$ 51.20	6	\$ 25.62
TOTAL				\$ 151.56 B)

3.2.2.3 COSTO DE ENERGÍA ELÉCTRICA.

Precios de cargo sobre energía AES Clesa El Salvador²⁵:

Tarifa residencial (cargo en dólares), menos de 99 KWH al mes: \$ 0.207532.

Cargo por comercialización \$ 0.814817.

Cargo por distribución \$ 0.071818.

Total tarifa por KWH \$ 1.094167.

Tarifa residencial (cargo en dólares), mayor a 99 KWH al mes: \$ 0.199051.

Cargo por comercialización \$ 0.934739.

Cargo por distribución \$ 0.061545.

Total tarifa por KWH \$ 1.195335.

Se ha calculado un promedio en el consumo mensual de 131 KWH de los últimos 6 meses considerando meses de 30 y 31 días.

Calculando el costo promedio por KWH/h:

2 meses de 30 días.

$131 \text{ KWH} / 30 \text{ días} = 4.36 \text{ KWH},$

$4.36 \text{ KWH} / 24 \text{ horas} = 0.181667 \text{ KWH/h.}$

4 meses de 31 días.

$131 \text{ KWH} / 31 \text{ días} = 4.23 \text{ KWH}$

$4.23 \text{ KWH} / 24 \text{ horas} = 0.17625 \text{ KWH/h.}$

²⁵ Tarifa de KWH AES Clesa <http://www.aeselsalvador.com/2009/Portals/0/SitioAES/AvisoCLESA.JPG> 01 Junio de 2012

Durante los dos meses iniciales se ha trabajado por 4 horas diarias.

120 horas en mes de 30 días x 3 equipos= 360 horas.

124 horas en mes de 31 días x 3 equipos= 372 horas.

Durante los siguientes cuatro meses se han establecido 7 horas diarias.

651 horas en 93 días (3 meses de 31 días) x 3 equipos= 1953 horas.

210 horas en el mes de 30 días x 3 equipos= 630 horas.

Calculamos costo total por horas:

(990 horas x 0.181667 KWH/h) x \$ 1.195335 KWH = \$ 214.98 (meses de 30 días).

(2325 horas x 0.17625 KWH/h) x \$ 1.195335 KWH = \$ 489.83 (meses de 31 días).

Total Costo de energía eléctrica: **\$ 704.81 C).**

3.2.2.4 OTROS GASTOS.

Se realizó un aproximado de otros gastos como papelería, transporte, uso de celular, internet y gastos por otros inconvenientes.

Internet:	\$ 226.80. (139 días aproximadamente por 3 personas)
Transporte:	\$ 90.00.
Papelería:	\$ 25.00.
Llamadas y mensajería de celular:	\$ 18.00.
Otros inconvenientes:	\$ 120.00.
TOTAL	\$ 479.80 D).

COSTO TOTAL DEL SISTEMA.	
A - Costo de Personal.	\$ 4,200.00
B - Costo de Depreciación.	\$ 151.56
C - Costo de energía eléctrica.	\$ 704.81
D- Otros gastos.	\$ 479.80
Costos de Implementación	\$ 53.00
Costo de Sistema.	\$ 5,589.17 USD

Calculo de Costo de Sistema después de seis meses de desarrollo (El cual será absorbido por el trabajo de graduación en curso).

3.2.3 FACTIBILIDAD OPERACIONAL.

Para el desarrollo del sistema se necesitará del siguiente recurso humano que a continuación se detalla:

3.2.3.1 ANALISTAS PROGRAMADORES.

Los integrantes del grupo de trabajo de grado, egresados de la carrera de Ingeniería de Sistemas Informático, encargados del análisis, diseño y programación del sistema de control de seguimiento informático de graduados(as) de postgrados de la Universidad de El salvador FMO.

3.2.3.2 DOCENTE DIRECTOR.

Profesional encargado de asesorar, guiar y evaluar todo el proceso de grado en todas sus fases:

- ✓ Fase De Planificación Y Recopilación De Información.
- ✓ Fase De Diseño.
- ✓ Fase De Desarrollo.
- ✓ Fase De Implementación.

Las cuales se establecieron anteriormente (Apartado 2.2).

3.2.3.3 USUARIOS DEL SISTEMA INFORMÁTICO.

Constituido por el personal de la Unidad de Postgrados y demás usuarios que utilizarán el sistema en desarrollo.

3.3 DISEÑO LÓGICO DE SISTEMA.

El diseño de sistema se representa a través de dos fases: el diseño lógico y el diseño físico.

Cuando los analistas formulan un diseño lógico; escriben las especificaciones detalladas del nuevo sistema; esto es, describen sus características: las salidas, entradas, archivos y bases de datos y procedimientos.

El diseño físico, actividad que sigue el diseño lógico, produce programas de software, archivos y un sistema en marcha, las especificaciones del diseño indican a los programadores que debe hacer el sistema ²⁶, para iniciar con el Diseño lógico del sistema se presentan a continuación los diagramas de caso de uso de este proyecto.

3.3.1 DIAGRAMAS DE CASO DE USO.

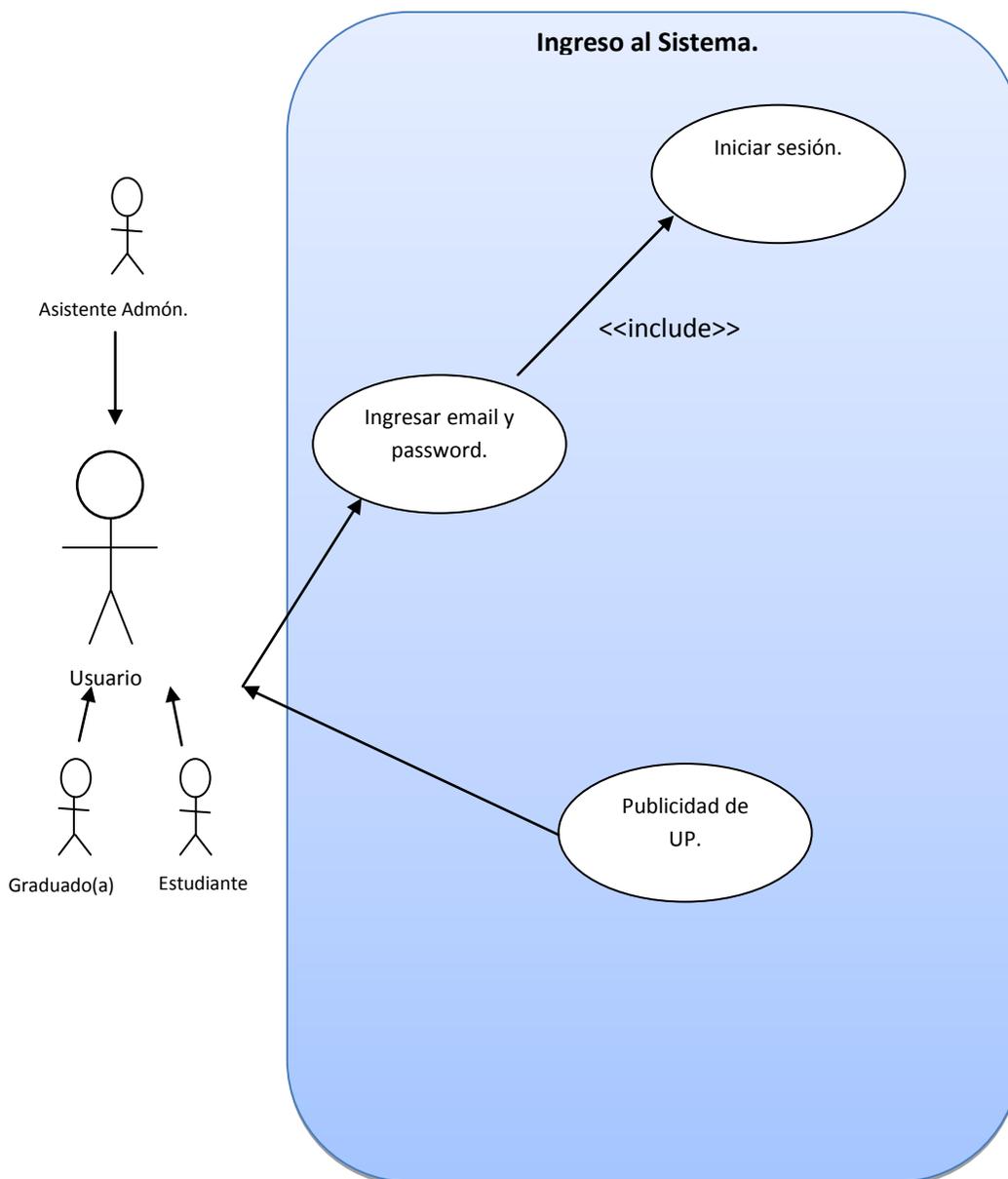
Un diagrama de casos de uso es una especie de diagrama de comportamiento y define una notación gráfica para representar casos de uso. ²⁷

A continuación los diagramas de caso de uso de el Sistema de Control de Seguimiento Informático para los Graduados(as) de Postgrado de la Universidad de El Salvador Facultad Multidisciplinaria de Occidente o SICSGUP, como será llamado el sistema desde ahora en este documento.

26 Análisis y Diseño de Sistemas <http://www.slideshare.net/rolfpinto/analisis-y-diseo-de-sistemas-kendal-kendal-sexta-edicin> 29 de Junio de 2012.

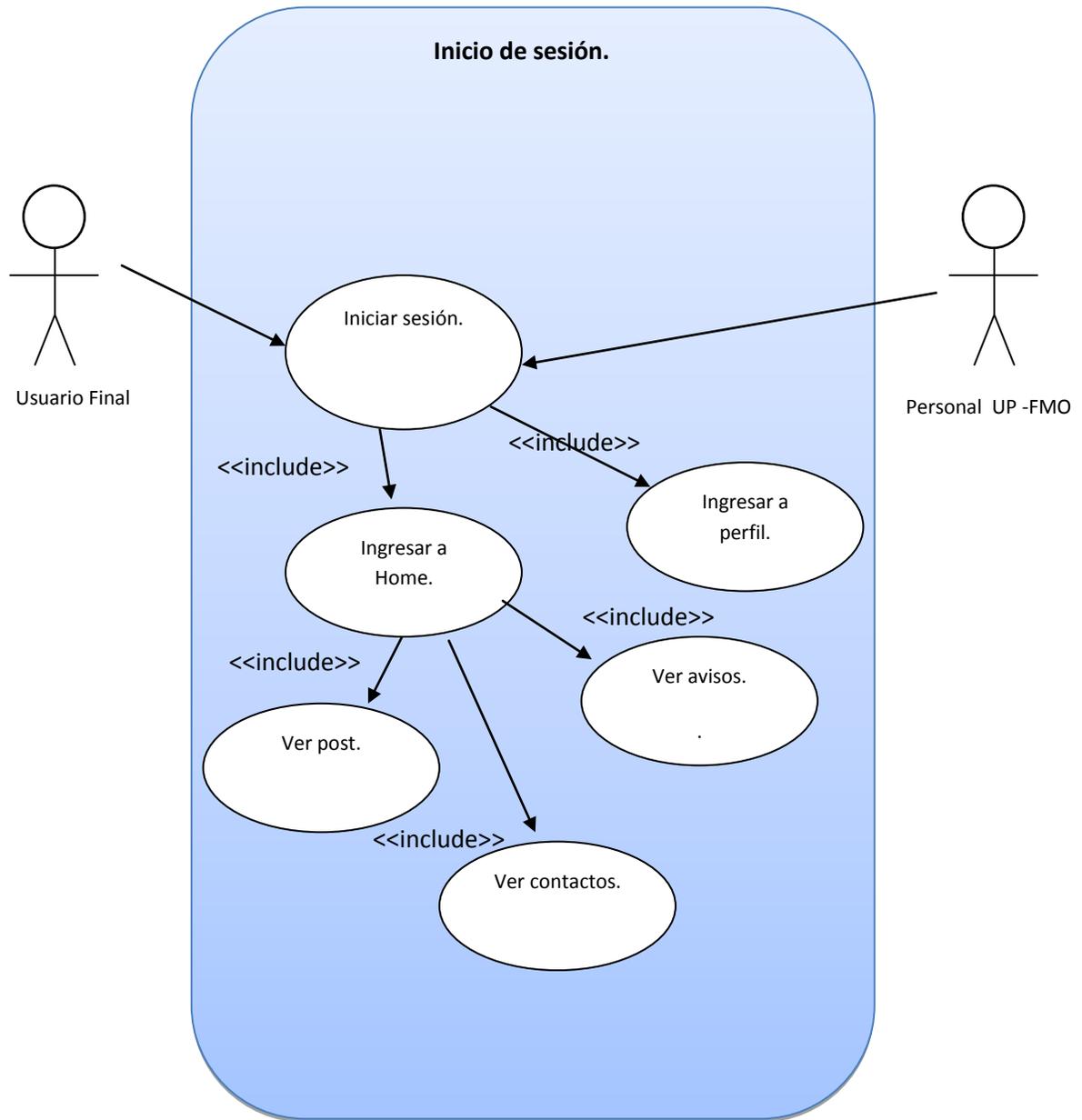
27 Diagramas de Caso de Uso <http://www.slideshare.net/ktyk/uml-casos-de-uso>. 29 de Junio de 2012.

Caso de Uso: **Ingreso al sistema (página de inicio).**



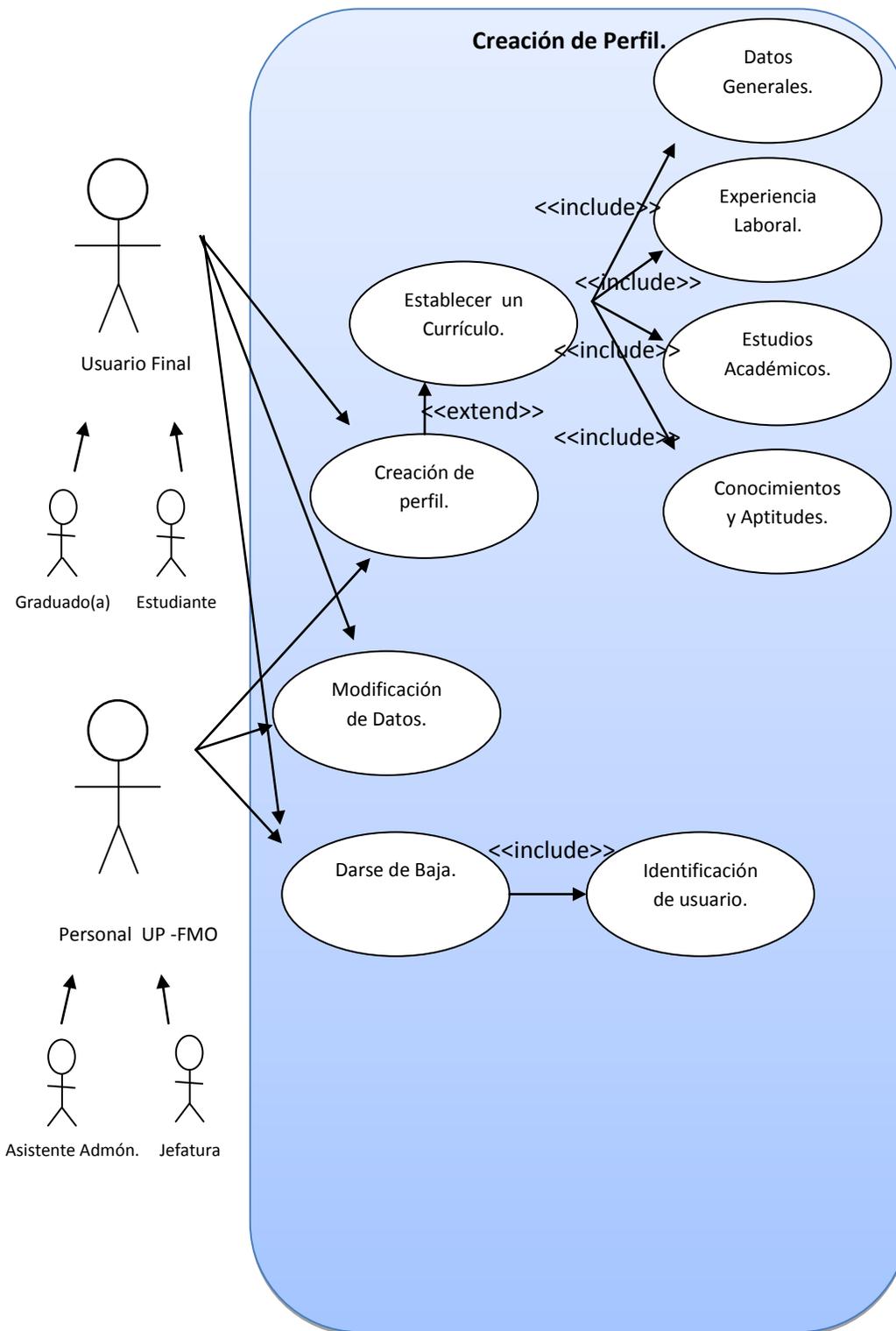
Procedimiento realizado por los usuarios al acceder a la página principal del Sistema SICSGUP.

Caso de uso: **Inicio de sesión.**



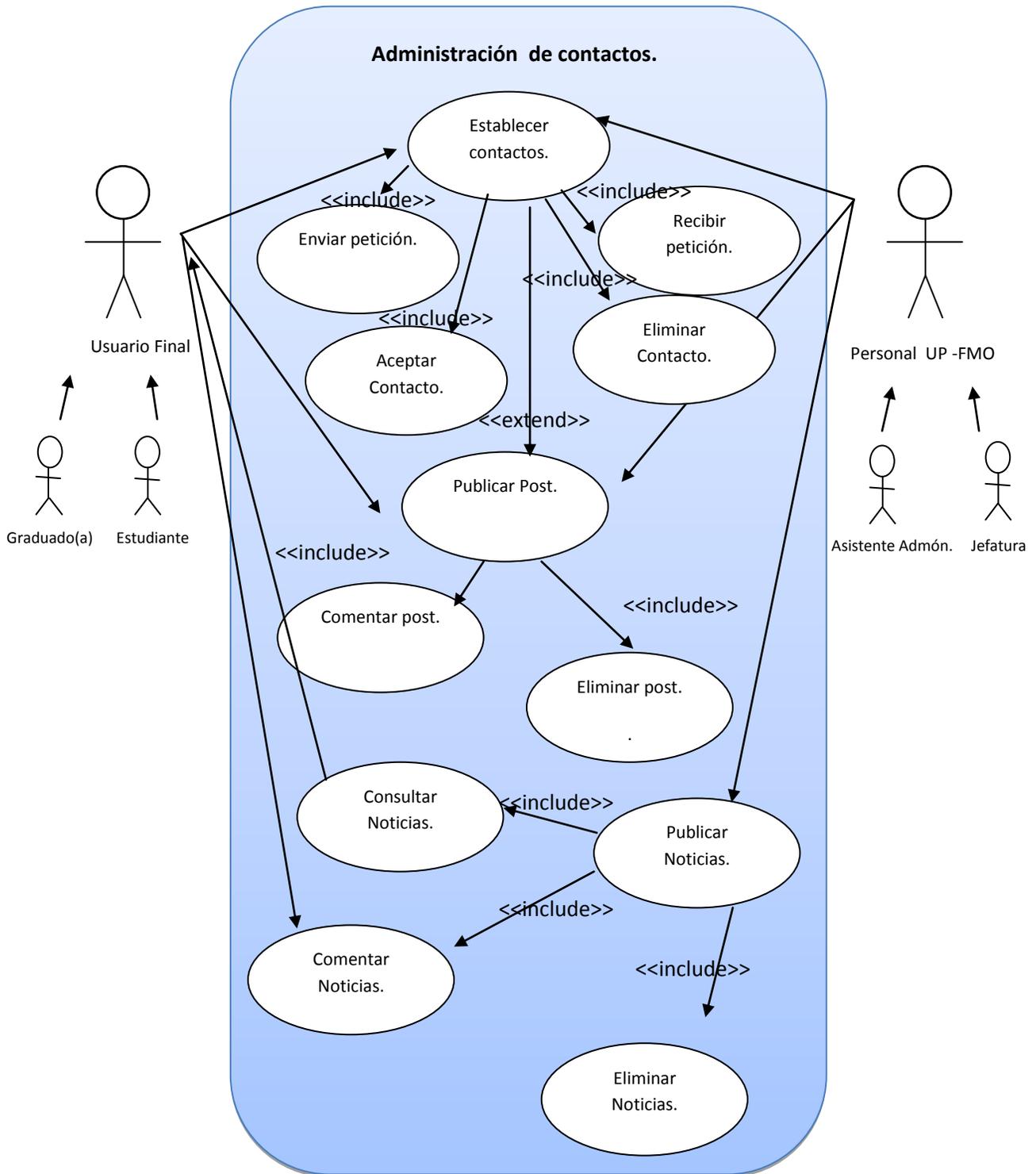
Procedimiento que los usuarios realizarán al ingresar al sitio; deberán registrarse.

Caso de Uso: Creación de Perfil.



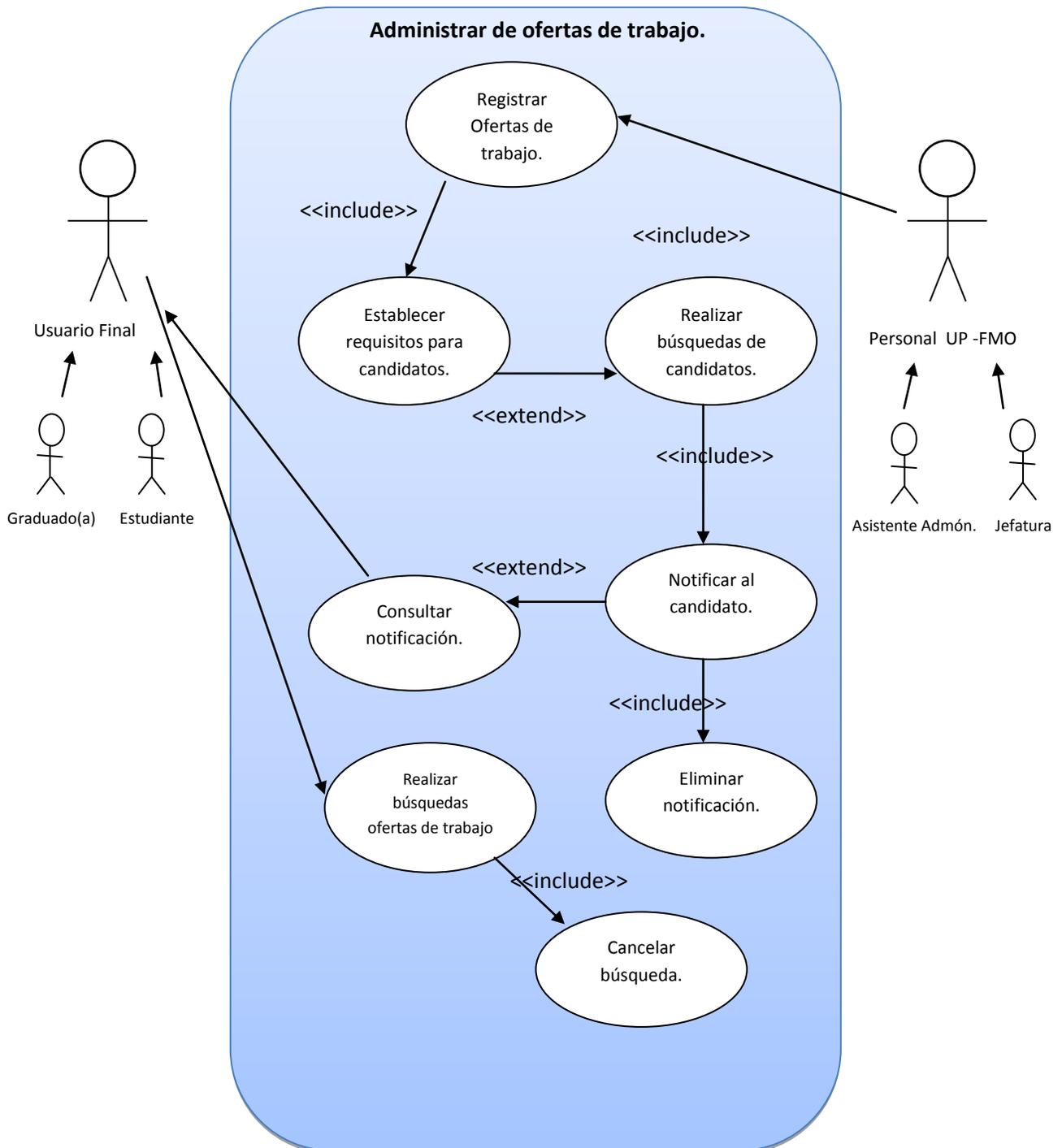
Procedimiento que los usuarios podrán realizar después de iniciar sesión. Deberán crear su perfil si aun no lo han hecho; crear su currículum además de modificar o eliminar información.

Caso de uso: **Administración de contactos.**



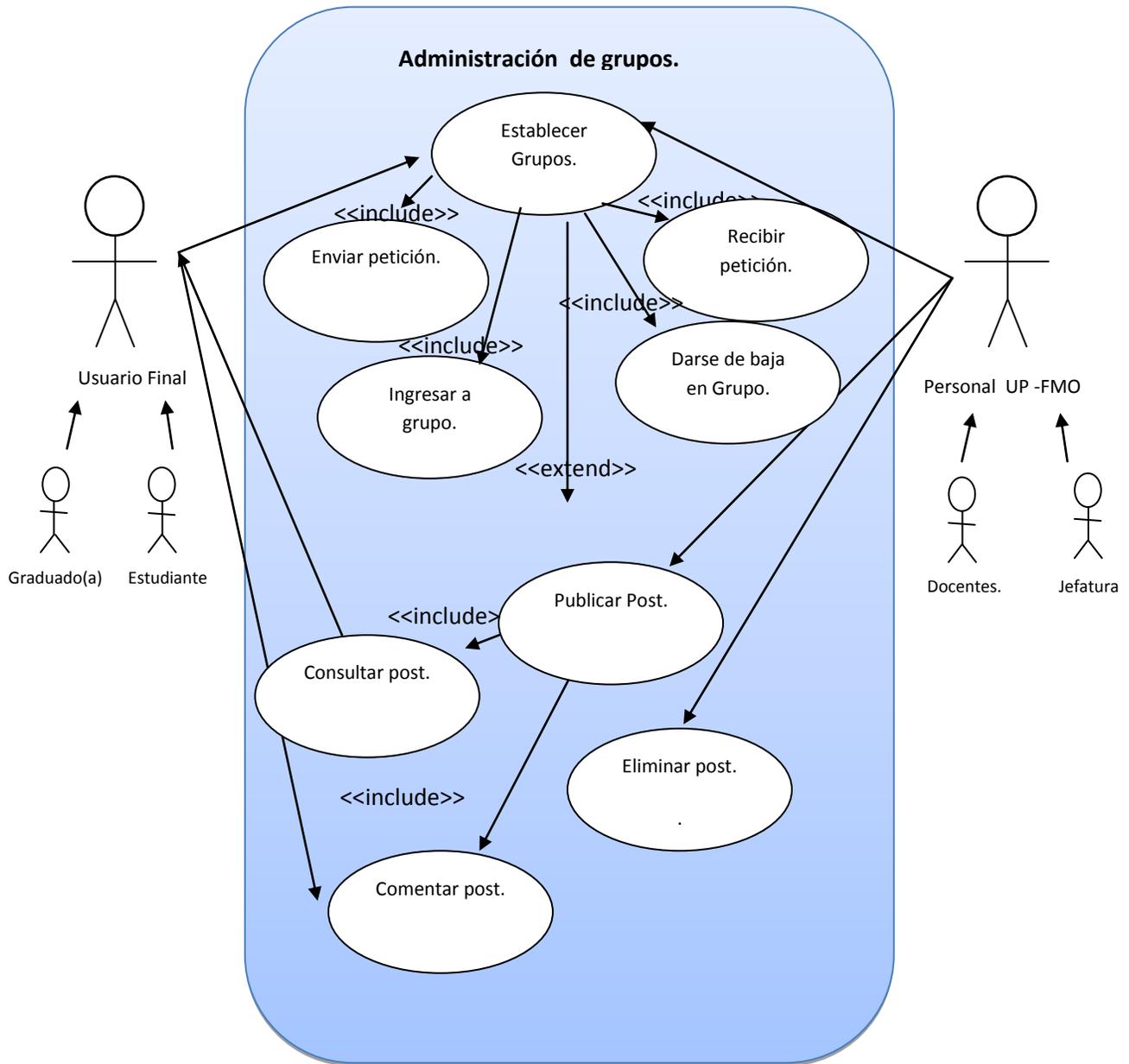
Procedimiento realizado cuando los usuarios deseen crear relación de amistad con otros usuarios; además de poder publicar post, comentarlos, ver noticias y eliminar esta información.

Caso de uso: **Administrar de ofertas de trabajo.**



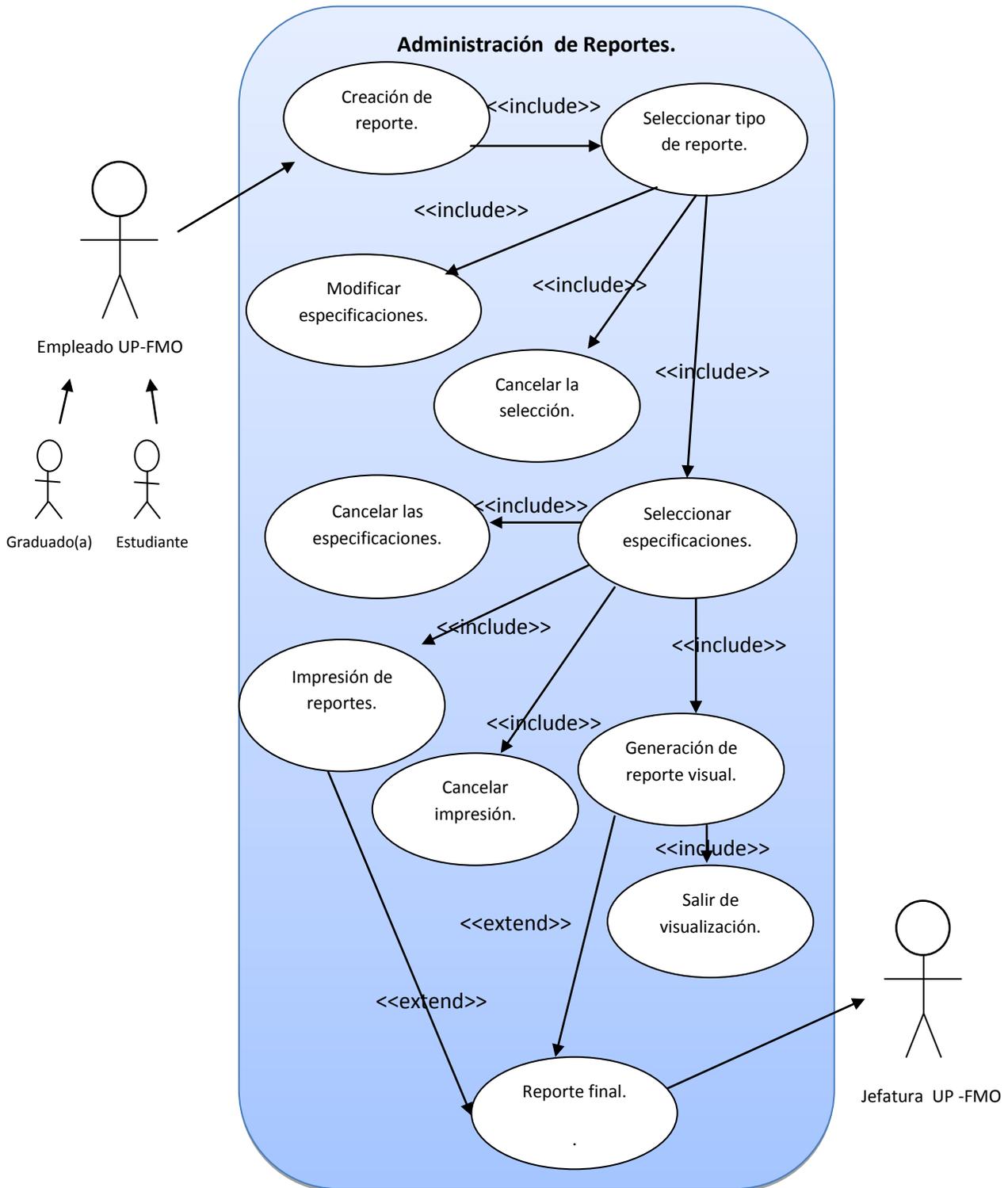
Procedimientos que los usuarios tanto de la Unidad de Postgrados como los usuarios finales realizarán cuando cuenten con información de las ofertas de trabajo.

Caso de uso **Administración de grupos.**



Procedimiento realizado cuando los usuarios deseen pertenecer a grupos administrados por los docentes y jefaturas; establezcan relación de amistad con otros usuarios dentro de los grupos; y puedan observar post, comentarlos, ver noticias.

Caso de uso: **Administración de Reportes.**



Procedimiento realizado por asistente administrativa de la unidad de Postgrado al momento de crear consultas visuales y para impresión; finalizando con la entrega de informes a la jefatura.

3.4 DISEÑO DE INTERFACES.

La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo. Normalmente suelen ser fáciles de entender y fáciles de accionar.²⁸

A continuación se presentan las interfaces que forman parte del sistema SICSGUP:

3.4.1 INTERFAZ DE ENTRADA DE DATOS.

Las interfaces de entrada de datos permiten que un cliente solicite datos a un programa ejecutado en el servidor web; mediante un explorador web.

Ejemplo de interfaz de entrada:

Interfaz de ingreso de datos sobre los estudios académicos que permitirá al usuario ingresar información sobre los estudios académicos que este ha realizado; estos datos pueden ser ingresados al momento de ingresar por primera vez al sistema o pueden ser modificados cuando el usuario lo desee.

Firefox Mozilla Firefox
localhost:9080/UESOCCFM/OPSTGRADOS2012/binf/basicinformation.xhtml DuckDuckGo

INICIO Configuración | Mi Perfil | Mensajes | Cerrar Sesión

Primer nombre: Hector Segundo nombre: Manuel
Apellidos: Ramos Genero: Hombre
Direccion: Colonia Espanya
Continuar

Telefono de casa: Casa Agregar numero

Numeros Telefonicos Tipo			
1234444			
77889999			
5452534			
112331			
2449-1386			
24924234			

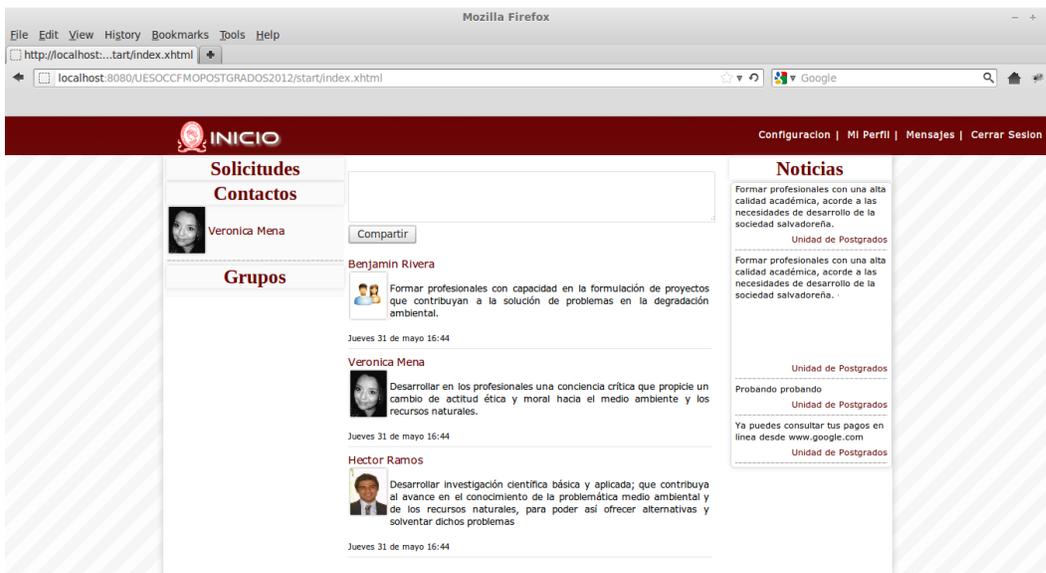
Universidad de El Salvador - Facultad Multidisciplinaria de occidente - Todos los derechos reservados. ©

28 Interfaz de Usuario <http://www.slideshare.net/sanfer08/disenio-de-la-interfaz-de-usuario> 01 Julio de 2012.

3.4.2 INTERFAZ DE SALIDA DE DATOS.

Son aquellas que muestran los resultados e información que proviene de una solicitud previa del sistema.

Un ejemplo de interfaz de salida es la pagina principal que todos los usuarios verán al ingresar a su perfil, en ella se mostrarán todas las noticias que la Unidad de Postgrados como usuario central tiene para toda su comunidad además de información sobre contactos, grupos entre otros.



3.5 DISEÑO DE BASE DE DATOS.

En este apartado se establecen todos aquellos elementos necesarios para el diseño de la base de datos de nuestro proyecto y que será el lugar donde se coleccionaran los datos de una forma interrelacionada; por lo que es necesario definiciones claras y específicas.

3.5.1 DESCRIPCIÓN DE LA BASE DE DATOS.

A continuación se presentan las tablas que conforman la base de datos con su respectiva descripción:

Nombre de Tabla	Descripción
addresses	Esta tabla almacenará la dirección de los usuarios del sistema, el usuario podrá ingresarla en el momento de crear su perfil y podrá modificarla cuando sea necesario.
addresstypes	En esta tabla se almacenarán los diferentes tipos de direcciones que el usuario manejará en su perfil, como ejemplo la dirección de trabajo, domicilio, etc.
attaches	En esta tabla se almacenara información de los elementos que los usuarios pueden adjuntar en los artículos que postee.
attachtype	En esta tabla se almacenara los tipos de elementos que serán adjuntados cuando el usuario publique un post.
comments	En esta tabla se almacenarán los comentarios que se realicen en los post, información de los adjuntos, usuarios y estado del comentario (activo o inactivo).
commentsgroup	En esta tabla como en la anterior se almacenará información de los comentarios que se realizan en los post realizados dentro de los grupos, además de la información de adjuntos y estados de los comentarios.
departments	En esta tabla se almacenará información con respecto a la dirección que el usuario especifique, información del departamento del país en el que usuario vive y labora (si es el caso).
education	En esta tabla se almacenará información académica, como por ejemplo la o las carreras universitarias, maestrías, cursos, etc. que el usuario

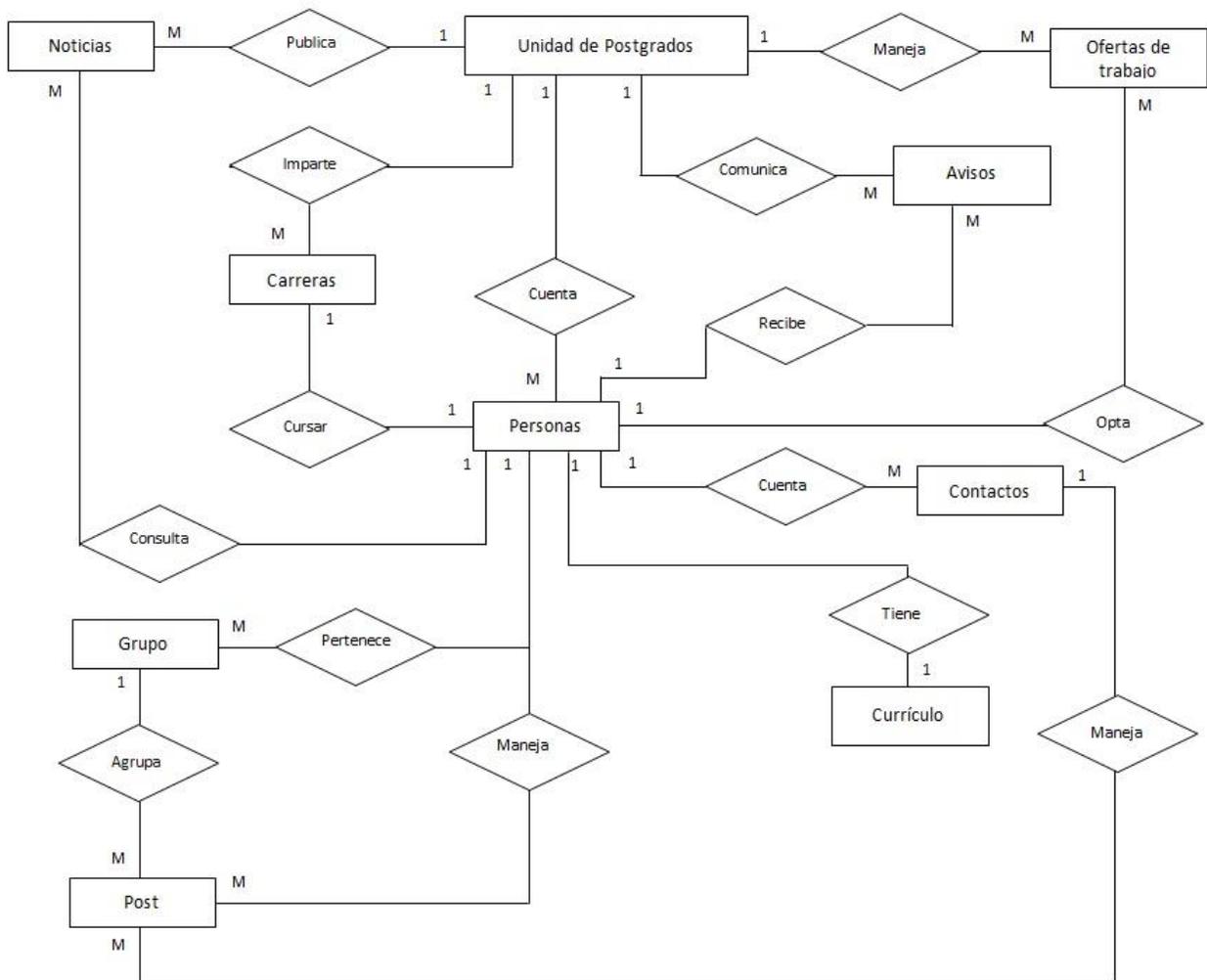
	haya realizado y que formarán parte de su currículum.
experiences	En esta tabla se almacenará información sobre la experiencia laboral actual o anterior que el usuario, el lugar donde trabaja, cargo, fechas de inicio y finalización (si es el caso).
friendship	En esta tabla permitirá establecer la relación de “amistad” entre contactos; que facilitará conocer el contacto que envió y recibió comentarios en un determinado post.
groups	En esta tabla se almacenará información sobre los grupos que los usuarios pueden crear o del que pueden formar parte.
joboffers	En esta tabla se almacenará información sobre las ofertas de trabajo que las empresas comuniquen a la Unidad de Postgrados.
jobs	En esta tabla se almacenarán información sobre los cargos en los empleos de los usuarios; y que serán especificados en su currículum, información que posteriormente será utilizada en las búsquedas de candidatos para ofertas de trabajo.
logs	Esta tabla servirá para llevar el control de la sesión del usuario; información como día, hora y un indicador único que el servidor brindará para cada sesión establecida.
memberships	Esta tabla servirá para llevar el control de la relación entre los usuarios que pertenecen a determinados grupos.
messages	Esta tabla servirá para llevar el control de las personas que recibirán y publicaran mensajes, el mensaje en si y el estado de vigencia de estos.
municipio	En esta tabla se almacenarán información sobre el municipio del departamento que el usuario ha establecido en su(s) domicilio(s).
names	En esta tabla se almacenará información sobre el nombre que manejará el usuario en el sistema y si este esta vigente o ha sido cambiado.
news	En esta tabla se almacenará información sobre noticias, promociones, etc. que la unidad de postgrados desee hacer públicos mediante el sitio web y al que estudiantes, graduado(as) y público en general podrán tener acceso.

people	En esta tabla se almacenará la mayor parte de la información del usuario final como correo electrónico, genero, password, etc.
peopleeducation	Esta tabla servirá para llevar el control entre el usuario, las carreras que el usuario almacenó en education, además de almacenar la fecha de inicio de estudios y la fecha de finalización.
peopletypes	En esta tabla se almacenará la información de los diferentes tipos de usuarios que manejará el sistema como por ejemplo los usuarios finales y los usuarios intermedios que en este caso será el personal de la Unidad de Postgrados.
petitions	En esta tabla se almacenará la información de los usuarios que reciben o hagan peticiones de amistad a otros usuarios y si estas peticiones están aun activas o no.
phones	En esta tabla se almacenará la información de los diferentes números telefónicos que el usuario ingrese en su currículum.
phonetypes	En esta tabla se almacenará el tipo de numero telefónico por ejemplo teléfono de domicilio, móvil, trabajo etc.
postgroups	En esta tabla se almacenará información sobre los post que se realizarán dentro de un grupo, su vigencia, usuario origen, etc.
post	En esta tabla se almacenará información sobre los artículos que los usuarios publican, adjuntos y estado de vigencia.
sectortype	Esta tabla almacenará información sobre los sectores de trabajo de los usuarios.
statuspetitions	En esta tabla se almacenará información sobre el estado de vigencia de las peticiones realizadas por los usuarios.
titletypes	En esta tabla se almacenará información sobre el tipo de titulo académico obtenido en los estudios del usuario.
validitystatus	Esta tabla servirá para llevar el control de la validación del estado de la información almacenada en varias tablas de la base de datos y que será activo o inactivo.

3.5.2 MODELO ENTIDAD-RELACIÓN.

Un diagrama o modelo entidad-relación es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones.²⁹

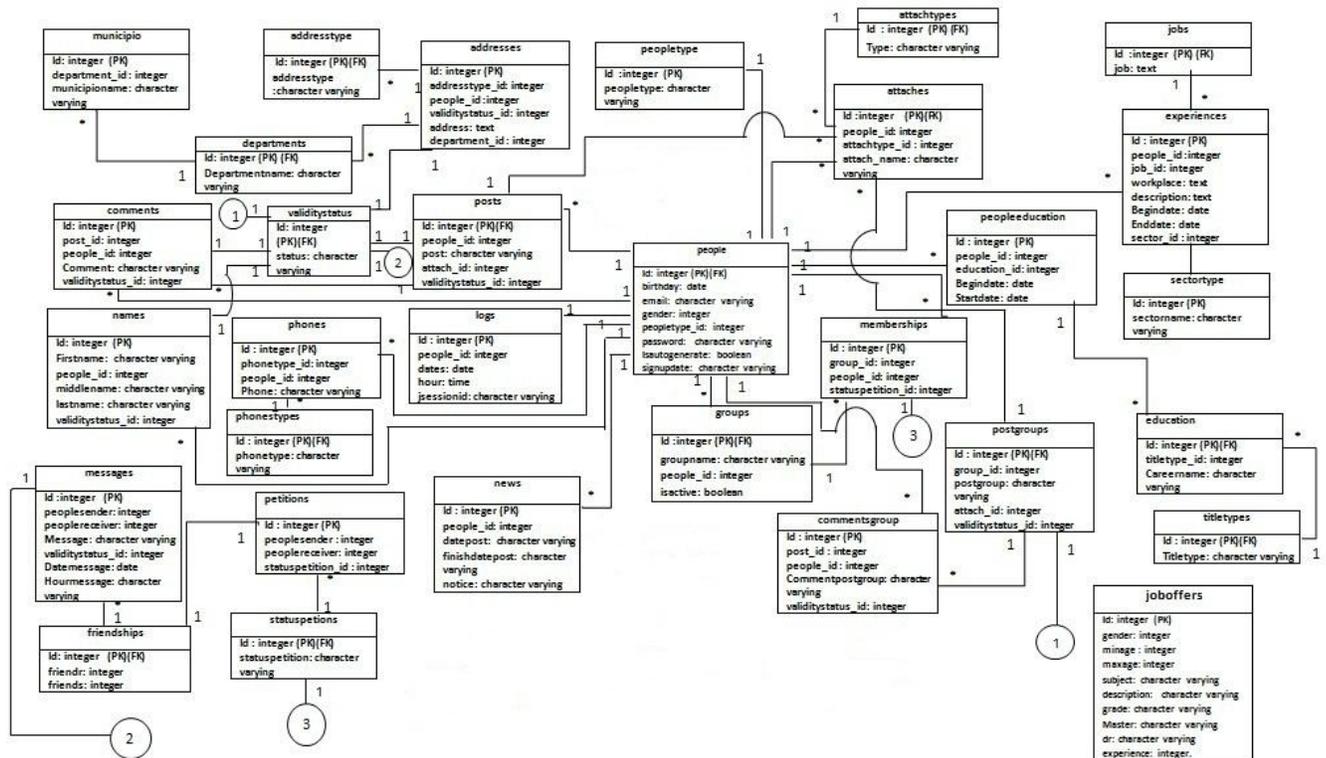
SISTEMA DE CONTROL DE SEGUIMIENTO INFORMÁTICO PARA LOS GRADUADOS(AS) DE POSTGRADO DE LA UNIVERSIDAD DE EL SALVADOR FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE (SICSGUP).



29 Modelo de Entidad – Relación <http://www.slideshare.net/oswchavez/clase-3-modelo-entidad-relacion> 01 de Julio de 2012.

3.5.3 MODELO FÍSICO DE BASE DE DATOS.

SISTEMA DE CONTROL DE SEGUIMIENTO INFORMÁTICO PARA LOS GRADUADOS(AS) DE POSTGRADO DE LA UNIVERSIDAD DE EL SALVADOR FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE (MODELO FÍSICO DE BASE DE DATOS).



3.5.4 DICCIONARIO DE DATOS.

Tabla: Addresses.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
FK	addresstype_id	integer	Identificador único de el tipo de dirección.
FK	people_id	integer	Identificador único del usuario.
FK	validitystatus_id	integer	Identificador único de la validación del estado de la dirección.
	address	text	Dirección.
FK	department_id	integer	Identificador único de departamento.

Tabla: addresstypes.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
	addresstype	character varying (15)	Se detallan los diferentes tipos de direcciones.

Tabla: attaches.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
FK	people_id	integer	Identificador único del usuario.
FK	attachtype_id	integer	Identificador único para el tipo de adjunto.
	attach_name	character varying(100)	Nombre de elemento a adjuntar.

Tabla: attachtypes.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
	type	character varying(15)	Tipo de elemento a adjuntar.

Tabla: comments.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
FK	post_id	integer	Identificador único del post.
FK	people_id	integer	Identificador único del usuario.
	comment	character varying(250)	Comentario.
FK	validitystatus_id	integer	Identificador único de validación sobre el estado del comentario.

Tabla: commentsgroups.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
FK	post_id	integer	Identificador único del post.
FK	people_id	integer	Identificador único del usuario.
	commentpostgroup	character varying(250)	Comentario que se hace en el grupo.
FK	validitystatus_id	integer	Identificador único para validar el estado del comentario.

Tabla: departments.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
	departmentname	character varying(15)	Contendrá el nombre del departamento que forma parte de la dirección que el usuario almacene.

Tabla: education.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
FK	titletype_id	integer	Identificación para tipo de títulos académicos obtenidos.
	careername	character varying(100)	Nombre de la carrera.

Tabla: experiences.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
FK	people_id	integer	Identificador único para el usuario.
FK	job_id	integer	Identificador único para trabajo del usuario.
	workplace	text	Lugar de trabajo.
	description	text	Descripción.
	begindate	date	Fecha de inicio de trabajo.
	enddate	date	Fecha de finalización de trabajo.
FK	sector_id	integer	Identificador único para el sector de trabajo.

Tabla: friendships.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	Id	integer	Identificador único.
	friendr	integer	Establecer relación de recepciones entre contactos.
	friends	integer	Establecer relación de publicación para otros contactos.

Tabla: groups.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
	groupname	character varying(100)	Nombre de grupo.
FK	people_id	integer	Identificador único para usuario.
	isactive	boolean	Verifica si el grupo esta activo.

Tabla: joboffers.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
	gender	integer	Sexo del usuario.
	minage	integer	La edad mínima del usuario evaluada en oferta de trabajo.
	maxage	integer	La edad máxima del usuario evaluada en oferta de trabajo.
	subject	character varying	Encabezado.
	description	character varying	Descripción de oferta de trabajo.
	grade	character varying	Titulo académico requerido.

	master	character varying	Maestría requerida.
	dr	character varying	Doctorado requerido.
	experience	integer	Experiencia laboral.

Tabla: Jobs.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	Integer	Identificador único.
	job	text	Empleos ingresados por el usuario.

Tabla: Logs.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
FK	people_id	Integer	Identificador único de usuario.
	dates	date	Fecha de logeo de sesión.
	hour	time	Hora de logeo de sesión.
	jsessionid	character varying(32)	Identificador único de inicio de sesión por parte del servidor.

Tabla: memberships .

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
FK	group_id	integer	Identificador único del grupo.
FK	people_id	integer	Identificador único del usuario.
FK	statuspetition_id	integer	Identificador único de estado de petición.

Tabla: messages.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
	peoplesender	integer	Usuario que envía mensaje.
	peoplereceiver	integer	Usuario que recibe mensaje.
	message	character varying(400)	Cuerpo del mensaje.
FK	validitystatus_id	integer	Identificador único para validar el estado del mensaje.
	datemessage	date	
	hourmessage	character varying	

Tabla: municipio.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
FK	department_id	integer	Identificador único para el departamento.
	municipioname	character varying(50)	Nombre de municipio del departamento establecido en la dirección.

Tabla: names.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
	firstname	character varying (25)	Primer nombre.
FK	people_id	integer	Identificador único para usuario.
	middlename	character varying (25)	Segundo nombre.
	lastname	character varying (25)	Apellidos.
FK	validitystatus_id	integer	Identificador de validador del estado del nombre, activo o inactivo.

Tabla: news.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
FK	people_id	integer	Identificador único del usuario.
	datepost	character varying	Fecha inicio de post.
	finishdatepost	character varying	Fecha final de post.
	notice	character varying	Noticia.

Tabla: people.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único para el usuario.
	birthday	date	Fecha de nacimiento.
	email	character varying(25)	Correo electrónico.
	gender	integer	Genero.
FK	peopletype_id	integer	Identificador único para tipo usuario.
	password	character varying(100)	Password.
	isautogenerate	boolean	Verifica si la contraseña fue establecida por el usuario o es la predeterminada.
	signupdate	character varying	Registra el año de ingreso del usuario en su maestría.

Tabla: peopleeducation.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador única.
FK	people_id	integer	Identificador único para el usuario.
FK	education_id	integer	Identificador único educación de usuario.
	begindate	date	Fecha de inicio.
	startdate	date	Fecha de final.

Tabla: peopletypes.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
	peopletype	character varying(15)	Tipo de usuario.

Tabla: petitions.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
	peoplesender	integer	Persona que envía petición.
	peoplereceiver	integer	Persona que recibe petición.
FK	statuspetition_id	integer	Identificador único de el estado de petición.

Tabla: phones.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
FK	phonetype_id	integer	Identificador único de tipo de usuario.
FK	people_id	integer	Identificador único de usuario.
	phone	character varying(15)	Numero telefónico.

Tabla: phonestypes.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
	phonetype	character varying(15)	Tipo de número telefónico.

Tabla: postgroups.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
FK	group_id	integer	Identificador único de grupo.
	postgroup	character varying(400)	Post realizado dentro del grupo.

FK	attach_id	integer	Identificador único de elemento adjunto.
FK	validitystatus_id	integer	Identificador único de validador de estado de post del grupo.

Tabla: posts.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
FK	people_id	integer	Identificador único de usuario.
	post	character varying(400)	Post.
FK	attach_id	integer	Identificador único de elemento adjunto.
FK	validitystatus_id	integer	Identificador único de validador de estado de post del usuario.

Tabla: sectortype.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
	sectorname	character varying	Sector de trabajo.

Tabla: statuspetions.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
	statuspetition	character varying	Estado de petición.

Tabla: titletypes.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
	titletype	character varying(25)	Tipo del titulo académico.

Tabla: validitystatus.

KEYTYPE	Nombre	Tipos de Datos	Descripción
PK	id	integer	Identificador único.
	status	character varying(10)	Estado.

3.6 REQUERIMIENTOS DE SEGURIDAD DEL SISTEMA.

La seguridad informática en un sistema se enfoca en la protección de la infraestructura computacional y todo lo relacionado con esta (incluyendo la información contenida). La seguridad informática comprende software, bases de datos, archivos y todo lo que la organización valore (activos) y signifique un riesgo si ésta llega a manos de otras personas.³⁰

3.6.1 ESPECIFICACIONES DE SEGURIDAD DEL SISTEMA.

Durante el diseño de SICSGUP se consideraron las siguientes medidas en cuanto a la seguridad:

3.6.1.1 ENCRIPCIÓN MD5.

La encriptación md5 es un algoritmo de encriptación muy utilizado para la encriptación de las contraseñas utilizadas en aplicaciones web, como es muy sabido estas aplicaciones utilizan bases de datos para poder manejar portales dinámicos y atractivos; por lo que la protección de los datos que se manejan en estas es muy importante, principalmente si se manejan nombres de usuarios y sus respectivas contraseñas o password, es esta última la que debe de encriptarse y luego guardarse así en la bases de datos.

La encriptación de contraseñas es una medida eficiente para evitar la vulnerabilidad en el servidor; de modo que una persona sin autorización no pueda acceder a la información de un usuario determinado.

Java implementa un paquete de encriptación de palabras que utiliza el algoritmo MD5 que ya se encuentran por defecto en java.

30 Seguridad Informática <http://www.slideshare.net/Financieros2008/seguridad-y-control-de-los-sistemas-de-informacin> 18 de Julio de 2012.

3.6.1.2 GENERACIÓN DE CONTRASEÑAS CON GUID.³¹

Un globally unique identifier («identificador globalmente único») o GUID es un número pseudoaleatorio empleado en aplicaciones de software parece generado al azar pero en realidad es un producto de un algoritmo completamente determinista. Aunque no se puede garantizar que cada GUID generado sea único, el número total de claves únicas (2^{128} o 3.4028×10^{38}) es tan grande que la posibilidad de que se genere un mismo número dos veces puede considerarse nula en la práctica.

El GUID es una implementación de Microsoft de un estándar llamado universally unique identifier o UUID, especificado por la Open Software Foundation (OSF). Es esencialmente un número de 16 bytes. Los GUID se escriben empleando una palabra de cuatro bytes, tres palabras de dos bytes y un bloque de seis bytes, aunque los dos primeros aparecen separados. Por ejemplo: {3F2504E0-4F89-11D3-9A0C-0305E82C3301}.

3.6.1.3 MANEJO DE SESIONES.³²

Una sesión es una serie de comunicaciones entre un cliente y un servidor en la que se realiza un intercambio de información. Por medio de una sesión se puede hacer un seguimiento de un usuario a través de la aplicación.

El tiempo de vida de una sesión comienza cuando un usuario se conecta por primera vez a un sitio web pero su finalización puede estar relacionada con tres circunstancias:

- ✓ Cuando se abandona el sitio web.
- ✓ Cuando se alcanza un tiempo de inactividad que es previamente establecido, en este caso la sesión es automáticamente eliminada. Si el usuario siguiera navegando se crearía una nueva sesión.
- ✓ Se ha cerrado o reiniciado el servidor.

31 GUID <http://technet.microsoft.com/es-es/library/cc739558%28v=ws.10%29.aspx> 01 AGOSTO 2012.

32 Manejo de sesiones <http://www.slideshare.net/jallendeastigarraga/seguridad-en-web> 01 AGOSTO 2012.

También se utilizan para la identificación de usuarios, en la que se deben de introducir un usuario y una contraseña. Después de haber hecho esto el usuario tendrá una serie de permisos sobre las páginas que va a visitar, de tal forma que si un usuario intenta pasar a una página si haberse identificado, el sistema comprobará que no se ha identificado y sería redireccionado a la página de identificación. Para poder realizarse estas operaciones es necesario almacenar en una sesión la información necesaria para saber que el usuario se ha identificado correctamente.

De hecho cuando a través de una página JSP se comienza una sesión, se crea un cookie llamada JSSESSIONID. La diferencia es que esta cookie es temporal y durará el tiempo que permanezca el navegador ejecutándose, siendo borrada cuando el usuario cierre el navegador.

Una vez que un usuario se ha autenticado, esa autenticación será válida para peticiones de cualquier aplicación Web que use el mismo nombre de cookie. No se le pedirá al usuario que vuelva a autenticarse.

Un Jsessionid cuenta con un número de 36 dígitos hexadecimales. Cada dígito hexadecimal es de cuatro bits de longitud. Es decir $36 \times 4 = 144$ bits.

3.6.1.4 MANEJO DE ROLES POR GLASSFISH.

Se utiliza para mantener independencia entre los roles en la aplicación, un rol define permisos para un conjunto de recursos dentro de una aplicación, ya que no todos los usuarios tienen los mismos privilegios, y no tendrán acceso a las mismas opciones en la aplicación mediante el manejo de roles el servidor podrá controlarse el acceso de los usuarios.

3.6.1.5 ACCESO CONTROLADO POR LA UNIDAD DE POSTGRADOS.

El Acceso de usuarios al Sistema de Control de Seguimiento de Graduados será controlado por la Unidad de Postgrados; quien será el que proporcione una contraseña al usuario final, después de que el este le indique que dirección de email utilizará para acceder al sistema. Con esto la unidad de Postgrados se asegura que personas que no pertenezcan a su comunidad de estudiantes y graduados(as) tenga acceso libre al sistema.

3.6.2 SEGURIDAD DEL SERVIDOR.

Ya que la implementación se llevará a cabo en el servidor del Departamento de Ingeniería y Arquitectura que ya cuenta con el servidor GlassFish; también se debe tomar en cuenta que debe estar implementado en el servidor:

3.6.2.1 EL PROTOCOLO HTTPS.

Hyper Text Transfer Protocol Secure (en español: Protocolo seguro de transferencia de hipertexto), más conocido por sus siglas HTTPS, es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de Hiper Texto, es decir, es la versión segura de HTTP. Es utilizado principalmente para el envío de datos personales o contraseñas.

El sistema HTTPS utiliza un cifrado basado en SSL/TLS para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP. De este modo se consigue que la información sensible (usuario y claves de paso normalmente) no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará imposible de descifrar.

El puerto estándar para este protocolo es el 443. ³³

33 Protocolo HTTPS <http://www.slideshare.net/babp/protocolo-https-3617275> 18 Julio de 2012.

IMPLEMENTACIÓN.

CAPÍTULO 4.

4.1 OBJETIVOS DEL PLAN DE IMPLEMENTACIÓN.

- ✓ Determinar las actividades o tareas que se realizarán durante la implementación del sistema SICSGUP.
- ✓ Instalación del sistema SICSGUP en el servidor del Departamento de Ingeniería y Arquitectura.
- ✓ Verificar que opere correctamente.

4.2 PLAN DE IMPLEMENTACIÓN.

Como la implementación se llevará a cabo en el servidor del Departamento de Ingeniería y Arquitectura y este ya cuenta con un Servidor http GlassFish, soporte para lenguaje Java EE6, Servidor de Base de Datos PostgreSQL, entre otros servicios necesarios para el funcionamiento del sistema; solo se determinarán las actividades de la instalación en sí.

4.2.1 ACTIVIDADES DEL PLAN DE IMPLEMENTACIÓN.

Actividad	Descripción.
A	Iniciar implementación.
B	Solicitar permiso para acceder al Servidor.
C	Instalar sistema SICSGUP.
D	Revisión de Hardware (servidor).
E	Revisión de Software en servidor.
F	Creación de base de datos en el Servidor de PostgreSQL.
G	Instalación de SICSGUP.
H	Revisar el Funcionamiento del Sistema.
I	Ingreso de información al sistema.
J	Realización de pruebas.
K	Corrección de resultados.
L	Capacitación del personal de Unidad de Postgrados.

4.2.2 DESCRIPCIÓN DE ACTIVIDADES.

A- Iniciar implementación.

Se establecerá una fecha para el inicio de la implementación del sistema.

B- Instalar Sistema SICSGUP.

Dentro de la Instalación de SICSGUP encontramos las siguientes subtareas:

C- Revisión de hardware en Servidor.

Evaluar que el hardware del Servidor cumpla con los requisitos necesarios para la instalación de SICSGUP.

D- Revisión de software en servidor.

Evaluar que el software del Servidor cumpla con los requisitos necesarios para la instalación de SICSGUP.

E- Creación de la Base de Datos en el Servidor PostgreSQL.

Se procederá a la creación de la base de datos llamada “uesoccfmopostgradodb” en el Servidor PostgreSQL a partir del Script sql.

F- Instalación de SICSGUP.

Se procederá a instalar el ejecutable de la aplicación que ya cuenta con las librerías, paquetes, y demás componentes que el sistema necesitará para ser funcional.

G- Revisar el Funcionamiento del Sistema.

Evaluar que las librerías, paqueterías y otros componentes hayan sido instalados en la forma correcta lo que permitirá que el sistema funcione normalmente.

H- Ingreso de información.

Ingresa todos los datos o información como carreras, personal administrativo, etc. que la Unidad de Postgrados considere necesario para iniciar las operaciones del sistema.

I- Realización de pruebas.

Prueba del funcionamiento del sistema con las configuraciones que se han manejado anteriormente.

J- Corrección de resultados.

En el caso de encontrar fallas durante la realización de pruebas, realizar las correcciones necesarias para que SICSGUP trabaje óptimamente.

K- Capacitación del personal de Unidad de Postgrados.

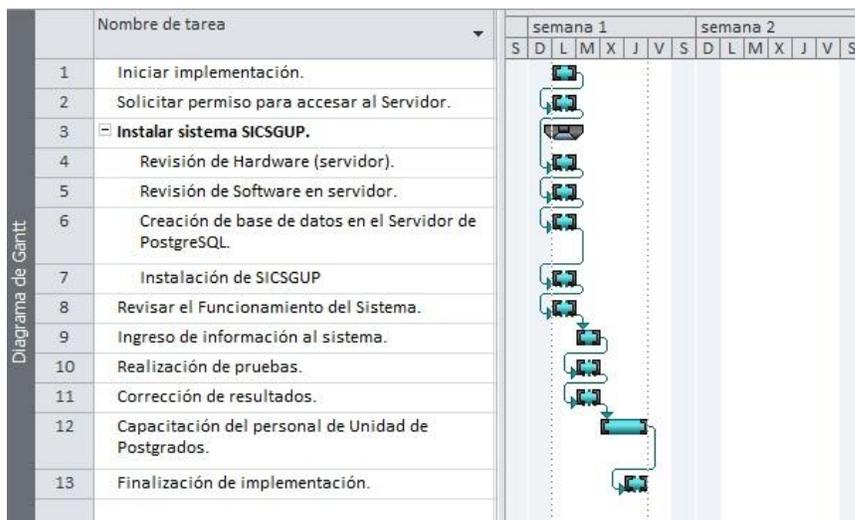
Se capacitará a todo el personal involucrado de la Unidad de Postgrados sobre la funcionalidad del Sistema de Control de Seguimiento de Graduados(as).

L- Finalización de implementación.

Fecha de finalización de la implementación.

4.2.3 CRONOGRAMA DEL PLAN DE IMPLEMENTACIÓN.

Seguendo el Cronograma general del trabajo de graduación (Anexo F); el proyecto finalizaría en la semana 29 continuando inmediatamente con la implementación del sistema en la semana 30.



4.3 COSTO DE LA IMPLEMENTACIÓN.

Con el fin de que la implementación no tenga retrasos se aprovecharán los recursos que la universidad ya posee, ya sean humanos o tecnológicos, para que la Unidad de Postgrados haga uso del sistema SICSGUP lo antes posible.

Como se ha mencionado anteriormente la implementación se llevará a cabo en el servidor del Departamento de Ingeniería y Arquitectura; lo que por ahora no origina gastos para el hardware necesario; si en el futuro la Unidad de Postgrados desea trasladar el sistema SICSGUP a un servidor propio e independiente se propone el equipo necesario en el Apartado 6.2 Recomendaciones.

Los costos de la implementación se detallan a continuación:

Recurso	Costo Unitario	Cantidad	Costo total
Capacitación	\$ 3.50/hora	4 horas por 2 días.	\$ 28.00
Manual del Usuario	\$ 0.25 c/impresión.	100 impresiones.	\$ 25.00
TOTAL			\$ 53.00

DOCUMENTACIÓN DEL SISTEMA.

CAPÍTULO 5.

5.1 MANUAL DEL USUARIO DEL SISTEMA DE CONTROL DE SEGUIMIENTO INFORMÁTICO, PARA LOS GRADUADOS(AS) DE POSTGRADO (SICSGUP).

Este documento es una ayuda para guiarle sobre el funcionamiento del Sistema (SICSGUP), dando instrucciones sobre los procedimientos que realiza el sistema.

5.1.1 GENERALIDADES.

Para ingresar al sistema desde una computadora no importa el sistema operativo que usted maneje, solo debe contar con un navegador web como Firefox Mozilla o Google Chrome y estar conectado a la red interna de la Facultad.

Usted debe contar con un nombre de usuario y una clave; como nombre de usuario utilizará el correo que haya entregado a la persona encargada de la administración del sistema en la Unidad de Postgrado quien a su vez le entregará la clave otorgando así la autorización del acceso al sistema.

5.2 FUNCIONALIDAD DEL SISTEMA.

5.2.1 MODULO PARA EL USUARIO FINAL.

Este modulo será la sección a la que podrán acceder todos los usuarios que sean autorizados por la Unidad de Postgrados.

Entre los usuarios que utilizarán este modulo se encuentra la comunidad estudiantil y los graduados(as) de la Unidad de postgrados de la Facultad Multidisciplinaria de Occidente, a continuación se detallan las opciones que el sistema ofrece para estos usuarios.

5.2.1.2 ACCESO AL SISTEMA.



Esta es la primera página que ud. observará al digitar la URL del sitio en su explorador; es aquí donde ingresará su correo electrónico y la clave que la persona encargada del sistema en la Unidad de Postgrado le entrego.

Usted puede consultar las noticias que la Unidad de Postgrados emita para el público en general sin tener que acceder la sitio; además podrá ingresar a su Expediente estudiantil de la Universidad dando clic en el botón “Expediente UES”, a la página de Administración Académica dando clic en el botón Académica entre otros.

5.2.1.3 CREACIÓN DE CURRÍCULO.

Si Usted accesa por primera vez al sistema, deberá crear su currículum, el sistema solicitará el ingreso de los siguientes datos:

- ✓ Datos personales.
- ✓ Estudios realizados.
- ✓ Experiencia labora.

5.2.1.3.1 AGREGAR DATOS PERSONALES.

Numeros Telefonicos Tipo			
1234444			
77089999			
5452334			
112331			
2449-1306			
24924234			

Esta opción le permite ingresar sus datos personales, además de que modifique sus datos en un determinado momento para lo que deberá eliminar la información que desea modificar y agregar la información correcta.

Los datos personales que se solicitarán en esta interfaz son los siguientes:

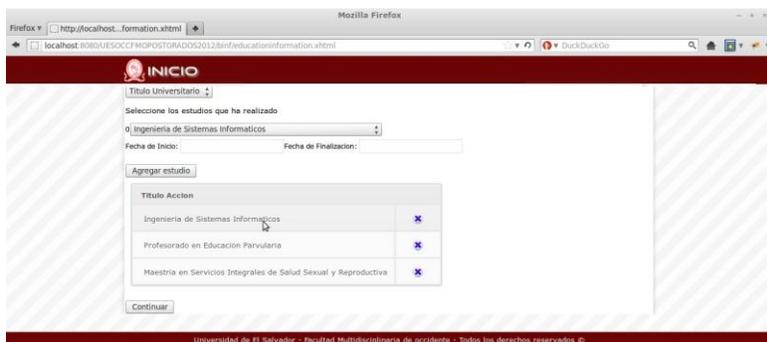
- ✓ Primer Nombre.
- ✓ Segundo Nombre.
- ✓ Apellidos.
- ✓ Genero: seleccionar entre las opciones presentadas.
- ✓ Dirección.
- ✓ Teléfono: seleccione el tipo de teléfono que ingresará (móvil, domicilio, oficina, etc.), ingrese el número en la caja de texto y de clic en el botón “Agregar número”. Usted podrá observar el listado de teléfonos que ingrese y eliminar el número que desee dando clic en



Cuando haya terminado de ingresar los datos personales deberá dar clic en el botón “continuar”.

Si desea regresar a su perfil de clic en Inicio en la esquina superior izquierda.

5.2.1.3.2 AGREGAR LOS ESTUDIOS REALIZADOS.



Esta opción le permite ingresar la información sobre los estudios que ha realizado, además de permitir que modifique sus datos en un determinado momento.

Los datos que se solicitarán en esta interfaz son los siguientes:

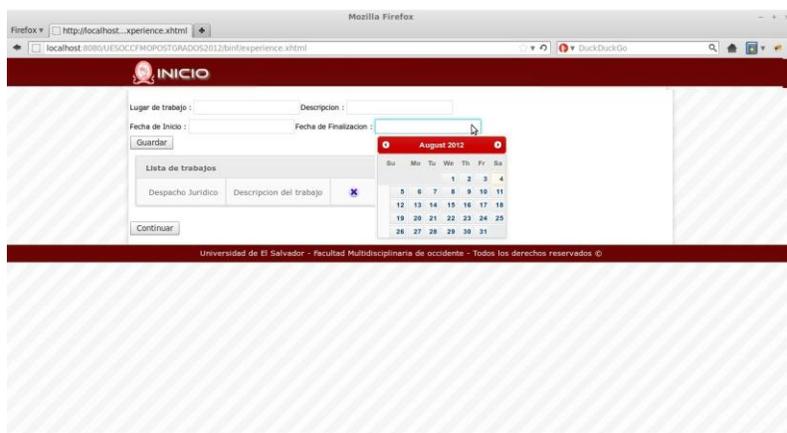
- ✓ Estudios que ha realizado: seleccione el tipo de estudio realizado ya sea una carrera universitaria, una maestría, etc.
- ✓ Seleccionar el título obtenido: seleccione el título que ha obtenido en sus estudios académicos o al que está optando si aún está en curso.
- ✓ Fecha de inicio: de clic en la caja de texto, se desplegará un calendario en el que el usuario seleccionará el año, mes y día de inicio de la carrera establecida anteriormente.
- ✓ Fecha de finalización: de clic en la caja de texto, se desplegará un calendario en el que el usuario seleccionará el año, mes y día de finalización de la carrera establecida anteriormente, dejarla en blanco si aún está cursando la carrera.

Dará clic en el botón “Agregar estudio” cuando los datos estén completos. El usuario podrá ingresar más de una carrera y estas serán mostradas en un listado; podrán ser eliminadas y modificadas cuando el usuario lo desee.

Si ha terminado de ingresar los datos de sus estudios realizados deberá dar clic en el botón “continuar”.

Si desea regresar a su perfil de clic en Inicio en la esquina superior izquierda.

5.2.1.3.3 AGREGAR SU EXPERIENCIA LABORAL.



Esta opción le permite ingresar la información sobre la experiencia laboral que tenga, además de permitir que modifique sus datos en un determinado momento.

Los datos que se le solicitarán son los siguientes:

- ✓ Lugar de trabajo: especifique el lugar donde ha trabajado o trabaja actualmente.
- ✓ Descripción: especifique el cargo o una pequeña descripción del cargo que desarrollo o desarrolla.
- ✓ Fecha de inicio: de clic en la caja de texto, se desplegará un calendario en el que el usuario seleccionará el año, mes y día de inicio del trabajo especificado anteriormente.
- ✓ Fecha de finalización: de clic en la caja de texto, se desplegará un calendario en el que el usuario seleccionará el año, mes y día de finalización del trabajo especificado anteriormente, dejarla en blanco si a la fecha es el trabajo actual.

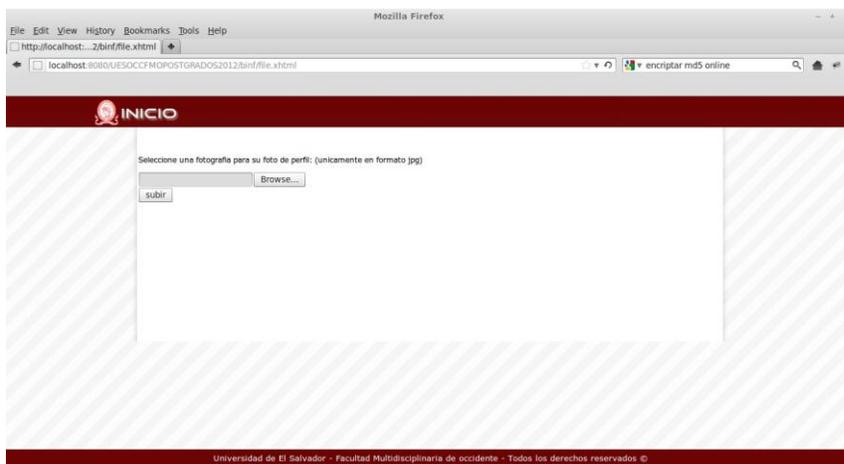
Dará clic en el botón “Guardar” cuando los datos estén completos. Podrá ingresar más de una experiencia laboral y estas serán mostradas en un listado; podrán ser eliminadas y modificadas cuando el usuario lo desee.

Si ha terminado de ingresar los datos de sus experiencias laborales deberá dar clic en el botón “continuar”.

Si desea regresar a su perfil de clic en “Inicio” en la esquina superior izquierda.

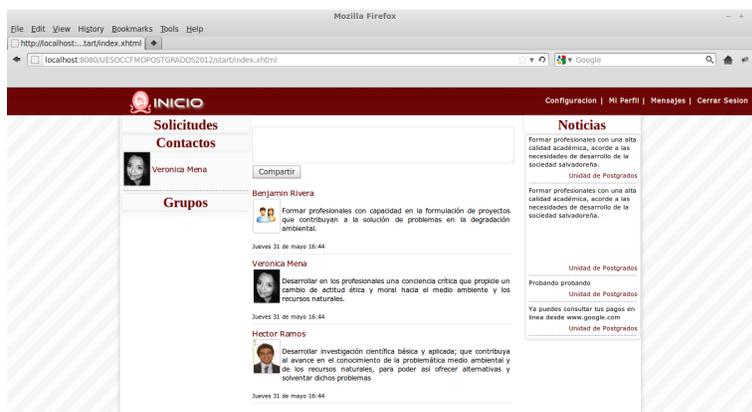
Cuando usted haya finalizado con la creación de su currículum se le presentará la página principal de su perfil.

5.2.1.4 AGREGAR FOTOGRAFÍA DE PERFIL.



Esta opción le permite ingresar su foto de perfil, deberá buscar la imagen y cuando la haya encontrado de clic en subir.

5.2.1.5 PAGINA PRINCIPAL DE PERFIL.



En la página principal de su perfil; a la izquierda se le presentan:

- ✓ Solicitudes: Aparecerán las solicitudes que otros usuarios le han enviado para formar parte de sus contactos; usted podrá aceptarla o rechazarla.
- ✓ Contactos: muestra una lista de todos sus contactos.
- ✓ Grupos: muestra todos los grupos de los cuales usted forma parte.

En la parte central encontrará:

- ✓ Los post que sus contactos realicen y sus propios post con su respectiva fecha y hora de emisión.
- ✓ Si el usuario desea agregar un comentario deberá ingresarlo en la caja de texto que se le presenta en la parte superior; para publicarlo haga clic en el botón “compartir”.

En el lado derecho:

- ✓ Noticias: Se muestran las noticias que la Unidad de Postgrados emita para toda la comunidad de estudiantes y graduados(as).

Barra de inicio: presentara hipervínculos como Mensajes, Configuración, Inicio, Mi perfil y Cerrar Sesión.

- ✓ Inicio: al dar clic en Inicio usted regresará a su página principal.
- ✓ Mi Perfil: al dar clic en Mi Perfil podrá observar su currículum.
- ✓ Mensajes: al dar clic en Mensajes ingresará a sus mensajes personales.
- ✓ Configuración: al dar clic en Configuración usted será enviado de nuevo a las páginas para la creación de su currículum.
- ✓ Cerrar Sesión: al dar clic saldrá de sistema.

5.2.1.6 VISTA DE UN PERFIL.



Si usted desea conocer los datos profesionales de uno de sus contactos puede acceder a su perfil, dando clic directamente en la lista de contactos de donde este formará parte o en el nombre que aparece en los comentarios que este realice en su perfil.

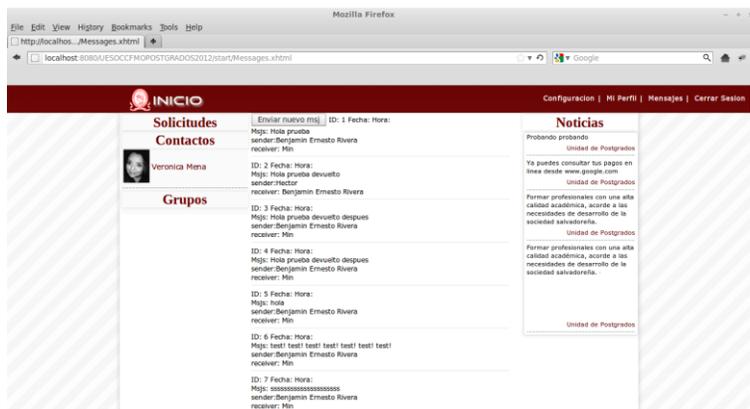
Como se muestra en la figura anterior, en esta opción las noticias emitidas por la Unidad de Postgrados siempre serán mostradas a la derecha; Las solicitudes de otros usuarios que usted haya recibido, sus contactos y sus grupos serán mostradas a la izquierda y en la parte central se mostrará el currículum de el contacto que usted desea consultar.

5.2.1.7 VER TODOS LOS CONTACTOS.



Esta opción le permitirá ver todos los contactos de los que usted ha aceptado solicitud.

5.2.1.8 MENSAJES PRIVADOS.



Esta opción le permite consultar los mensajes personales entre sus contactos y usted, además de permitirle enviar nuevos mensajes a otros contactos.

Dando clic en el botón “Enviar nuevo msj” aparecerá el siguiente cuadro de dialogo:



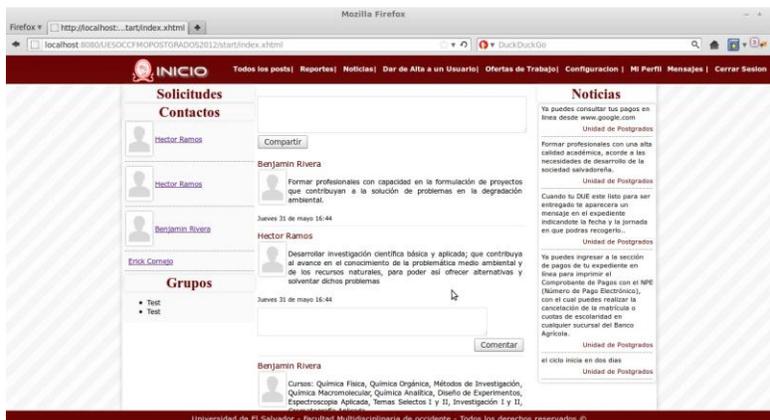
Usted podrá seleccionar el contacto al que desea enviar el mensaje, en la caja de texto escriba el cuerpo del mensaje y de clic en el botón “enviar” para completar la operación, si desea cancelar la operación solo de clic en el botón “Cancelar”.

El Usuario también podrá realizar búsquedas de ofertas de trabajo a las que según su criterio pueda optar y que la Unidad de postgrados de a conocer.

5.2.2 MODULO PARA EL ADMINISTRADOR.

Además de contar con las opciones que un usuario normal como configurar su información personal, mensajería privada y ver su perfil, el administrador tendrá opciones extras que se detallan a continuación.

5.2.2.1 PANTALLA PRINCIPAL DEL ADMINISTRADOR.



Esta es la pantalla principal del administrador, además de contar con las opciones de un usuario normal cuenta con opciones extras como se mencionó anteriormente, estas opciones se detallan en el menú superior y son las siguientes:

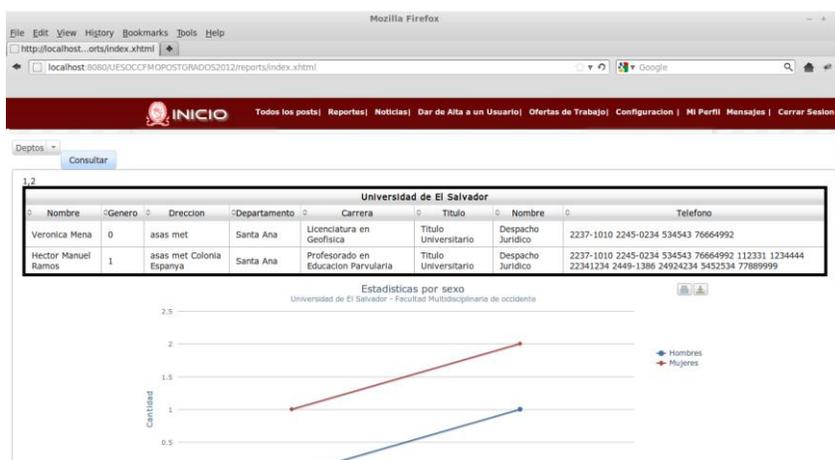
- ✓ Todos los post.
- ✓ Reportes.
- ✓ Noticias.
- ✓ Dar de alta a un usuario.
- ✓ Ofertas de trabajo.

5.2.2.2 VER TODOS LOS POST HECHOS POR LOS USUARIO.



Esta opción es más que todo para monitoreo de lo que los usuarios postean en el sitio, y saber si hay post que no estén de acorde con la finalidad del sitio, comentarios ofensivos o fuera de lugar; de esta manera La Unidad de Postgrados podrá llamar la atención mediante mensajes privador a los usuarios que estén involucrados en este tipo de acciones.

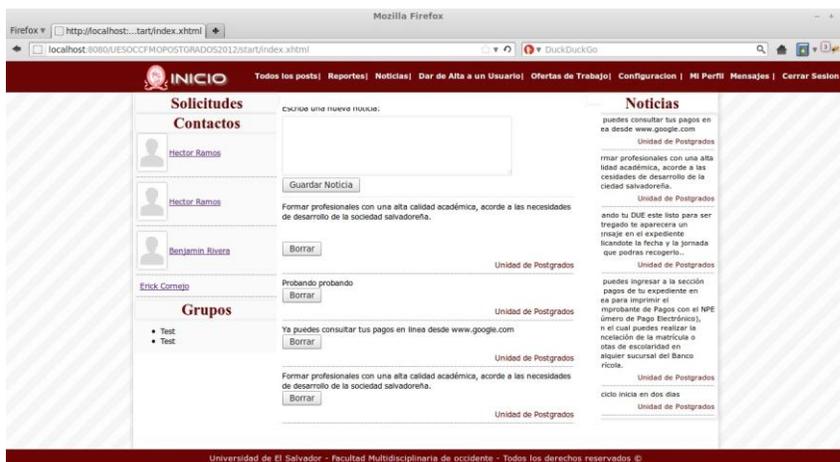
5.2.2.3 REPORTE Y GRÁFICOS.



Esta opción permitirá que el administrador del sistema pueda elaborar reportes con sus respectivos gráficos basándose en diferentes criterios de búsqueda como sexo, sector de trabajo, residencia, carrera, edad.

Dando clic en la caja de opciones que a su vez permitirá seleccionar entre diferentes opciones de búsqueda; cuando ya haya seleccionado los criterios de clic en el botón “Consultar” y se genera así el reporte visual y la grafica que pertenece a este reporte.

5.2.2.4 NOTICIAS.



En esta opción el administrador podrá agregar noticias nuevas, revisar las noticias existentes y eliminar las noticias que ya no sean necesarias, estas serán las noticias que haya emitido la Unidad de Postgrados y que serán observadas por todos los usuarios del sistema, mostrándose en la parte derecha del sitio.



En esta opción al administrador se le dará la opción de subir imágenes para mostrar como noticias en la página de inicio de sesión.

5.2.2.5 OFERTAS DE TRABAJO.

5.2.2.5.2 BÚSQUEDA DE CANDIDATOS.



En esta sección tendrá diferentes opciones para realizar búsquedas de candidatos para las ofertas de trabajo, por carrera universitaria, maestría, lugar de residencia, genero y experiencia laboral; Una vez haya seleccionado los criterios requeridos de clic en Consultar y se desplegará una lista de posibles candidatos.

5.3 MANUAL DEL ADMINISTRADOR.

- ✓ Si se desea consultar el Script de la base de datos vea el Anexo I.

=====

GroupsBean (Vease GroupsDao)

Notaciones:

@ManagedBean

@ViewScoped

Este método se ejecuta luego de inicializado el constructor. Evalúa si un usuario pertenece a un grupo.

@PostConstruct

```
public void verificacion() {
    try {
        HttpServletRequest request = (HttpServletRequest)
FacesContext.getCurrentInstance().getExternalContext().getRequest();
        String p = request.getParameter("id");
        if (postdao.verifiedMemberships(Integer.parseInt(p))) {
            setSuscripcion(false);
        } else {
            setSuscripcion(true);
        }
    } catch (Exception e) {
    }
}
```

Este método retorna un modelo de tipo `postViewModel`, el cual es un contenedor para el objeto devuelto por la consulta realizada en `GroupsDao`. Este método selecciona las primeros 100 publicaciones de las personas con las que el usuario tiene vínculos dentro del sistema.

```
public List<postViewModel> getFirstOneHundredPosts() {  
    HttpServletRequest request = (HttpServletRequest)  
FacesContext.getCurrentInstance().getExternalContext().getRequest();  
  
    String p = request.getParameter("id");  
  
    List<postViewModel> posts = new LinkedList<postViewModel>();  
  
    if (p != null) {  
        if (Integer.parseInt(p) > 0) {  
            posts = postdao.getFirstOneHundredPosts(Integer.parseInt(p));  
        } else {  
            posts = postdao.getFirstOneHundredPosts();  
        }  
    } else {  
        posts = postdao.getFirstOneHundredPosts();  
    }  
  
    return posts;  
}
```

Este método de tipo `void` se encarga de almacenar un post en la base de datos, gestiona su almacenamiento `savePost` de la clase `PostDao`.

```
public void savePost() {  
  
    HttpServletRequest request = (HttpServletRequest)  
FacesContext.getCurrentInstance().getExternalContext().getRequest();  
  
    String p = request.getParameter("id");  
  
    if (Postgroups != "") {  
        postdao.savePost((Integer.parseInt(p)), getPost());  
  
        setPost(""); } }
```

Este método de tipo void se encarga de realizar la petición al dao para registrar un miembro.

```
public void aceptmember(int membershipid) {  
    postdao.AceptarPetition(membershipid);  
}
```

GroupsDao

Notaciones:

@Stateless

@LocalBean

Almacena una publicación en la base de datos.

```
public void savePost(int groupID, String post) {  
    try {  
        People people = new People();  
        people = getCurrentPeople();  
        Groups group = new Groups();  
        group = dao.findById(Groups.class, groupID);  
        Validitystatus vstatus = new Validitystatus();  
        vstatus = dao.findById(Validitystatus.class, 1);  
        Postgroups pg = new Postgroups();  
        Posts posts = new Posts();  
        pg.setId(getNextPostId());  
        pg.setGroupId(group);  
        pg.setPostgroup(post);  
        pg.setValiditystatusId(vstatus);  
        dao.save(pg);  
    } catch (Exception ex) {  
        ex.printStackTrace();    } }  
}
```

Obtiene la lista de miembros dentro de un grupo.

```
public List<Groups> getmemberships() {  
    List<Groups> ldevuelta = new LinkedList<Groups>();  
    try {  
        People p = dao.findById(People.class, getCurrentPeople().getId());  
        List<Memberships> lista = (List<Memberships>) p.getMembershipsCollection();  
        for (Memberships item : lista) {  
            Groups Gtemp = new Groups();  
            Gtemp = dao.findById(Groups.class, item.getGroupid().getId());  
            ldevuelta.add(Gtemp);  
        }  
    } catch (Exception ex) {  
        ex.printStackTrace();  
        Logger.getLogger(GroupsDao.class.getName()).log(Level.SEVERE, null, ex);  
    }  
    return ldevuelta;  
}
```

isMember verifica si un usuario pertenece a un grupo para poder darle acceso a este. Retorna un valor booleano.

```
public boolean isMember(int groupid) {  
    boolean devuelto = false;  
    try {  
        People p = getCurrentPeople();  
        List<Memberships> IMembers = (List<Memberships>)  
entityManager.createNamedQuery("Memberships.findByGroupid").setParameter("group_id",  
groupid);  
        for (Memberships item : IMembers) {  
            if (p.getId() == item.getPeopleId().getId()) {
```

```

        devuelto = true;
    } else {
        devuelto = false;
    }
} catch (Exception ex) {
}
return devuelto;
}

```

La lista de peticiones se realiza mediante una modelo de tipo `PeticionesViewModel` el cual es un contenedor para el resultado de la consulta hecha a la base de datos.

```

public List<PeticionesViewModel> getPetitions() {
    List<Memberships> devuelta = new LinkedList<Memberships>();
    List<PeticionesViewModel> devueltaVm = new LinkedList<PeticionesViewModel>();
    List<Integer> Lids = (List<Integer>) entityManager.createNativeQuery("Select id from
memberships where statuspetition_id=1 and group_id in(Select id from groups where people_id="
+ getCurrentPeople().getId() + ")").getResultList();
    try {
        for (Integer item : Lids) {
            Memberships temp = dao.findById(Memberships.class, item);
            PeticionesViewModel tempvm = new PeticionesViewModel();
            tempvm.setGroupId(String.valueOf(temp.getGroupId().getId()));
            tempvm.setGroupName(dao.findById(Groups.class,
temp.getGroupId().getId()).getGroupname());
            tempvm.setMembershipId(temp.getId());
            tempvm.setPetitionerName("temporal");
            devueltaVm.add(tempvm);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return devueltaVm;
}

```

AceptarPetition Gestiona las peticiones para incorporar a un usuario a un grupo. Cambia el valor de la base de datos a 1 en la tabla StatusPetition para avalar el acceso al usuario.

```
public void AceptarPetition(int membershipid) {  
    try {  
        Memberships member = dao.findById(Memberships.class, membershipid);  
        Statuspetitions st = dao.findById(Statuspetitions.class, 2);  
        member.setStatuspetitionId(st);  
        dao.update(member);  
    } catch (Exception ex) {  
        ex.printStackTrace();    } } }
```

verifiedMemberships verifica si un usuario pertenece a un grupo para poder darle acceso a este. Retorna un valor booleano.

```
public boolean verifiedMemberships(int groupId) {  
    People actual = getCurrentPeople();  
    List<Memberships> Ldevuelta = (List<Memberships>) actual.getMembershipsCollection();  
    int contadorV = 0;  
    for (Memberships item : Ldevuelta) {  
        if (groupId == item.getGroupId().getId()) {  
            contadorV++;  
        }  
    }  
    if (contadorV > 0) {  
        return true;  
    } else {  
        return false;    } } }
```

AddGroup gestiona la creación de un nuevo grupo asignado a la persona que creo el grupo.

```
public void AddGroup(String Name) {  
    try {  
        Groups grupoNuevo = new Groups();  
        grupoNuevo.setGroupname(Name);  
        grupoNuevo.setIsActive(true);  
        grupoNuevo.setPeopleId(getCurrentPeople());  
        grupoNuevo.setId(getNextGroupId());  
    } catch (Exception ex) {  
    } }  
}
```

InformationBean (Vease tambien InformationDao)

implements Serializable

@ManagedBean

@ViewScoped

Esta función de tipo void es la encargada de enviar la información a InformationDao para almacenar el nombre de un nuevo usuario y redirecciona a la página de educación al finalizar el guardado.

```
public void insertInfo() {  
    try {  
        informationDao.InsertInfo(firstname, middlename, lastname);  
        informationDao.InsertAddress(address);  
        informationDao.updateGender(gender);  
    }  
}
```

```
FacesContext.getCurrentInstance().getExternalContext().redirect("../binf/educationinformation.xhtml");
```

```
    } catch (Exception ex) {  
        ex.printStackTrace();    } }  
}
```

getloadExperiences retorna una lista de experienciaViewModel, se comunica con InformationDao extrayendo la lista de las experiencias para el usuario que inicio sesión.

```
public List<experienceViewModel> getloadExperiences(){  
    List<experienceViewModel> listOfExperience = new LinkedList<experienceViewModel>();  
    try{  
        listOfExperience = informationDao.getListOfExperiences();  
    }  
    catch(Exception ex){  
        ex.printStackTrace();  
    }    return listOfExperience; } }
```

saveExperience esta función de tipo void se encarga de asignar una experiencia laboral a la persona que inicio sesión.

```
public void saveExperience(){  
    informationDao.saveExperience(workplace, description);  
    workplace = "";  
    description = "";  
    beginDate = "";  
    endDate = "";  
}
```

getphonesNumbersForPeople retorna una lista de números telefónicos junto a su tipo telefónico respectivo.

```
public List<basicinformationViewModel> getphonesNumbersForPeople() {  
    List<basicinformationViewModel> listOfPhones = new  
    LinkedList<basicinformationViewModel>();  
    try {  
        listOfPhones = informationDao.getphonesNumbersForPeople();
```

```
    } catch (Exception ex) {  
        ex.printStackTrace();  
    }    return listOfPhones; } }
```

savePhoneNumber almacena un numero telefónico para la persona que inicio sesión, al igual que el tipo de teléfono (teléfono fijo, móvil, trabajo, hogar, extranjero).

```
public void savePhoneNumber() {  
    if (getPhoneNumber() != "") {  
        informationDao.savePhoneNumber(getPhoneNumber(), getPhoneType());  
        setPhoneNumber("");  
    } else {    } }
```

fillCareers retorna una lista de elementos para realizar un llenado de datos dentro de un combobox con su respectivo identificador y descripción.

```
public List<SelectItem> fillCareers() {  
    return informationDao.getCareers(test);  
}
```

saveCarrer asigna un código de carrera al usuario que inicio sesión. En el caso de que se desee asignar una carrera que anteriormente había sido asignado, mostrara un mensaje de error.

```
public void saveCarrer() {  
    Boolean result;  
    result = informationDao.savePeopleEducation(careerSelected);  
    duplicateCareer = (result) ? "" : "El estudio que desea agregar ya fue asignado."  
}
```

DeleteCarrer desasigna la carrera seleccionada al usuario que inicio sesion

```
public void deleteCareer(int peopleeducation_id) {  
    informationDao.deletePeopleEducation(peopleeducation_id);  
}
```

deletePhone Desasigna un numero telefonico. Este borrado tambien lo realiza en la base de datos y no solo de forma logica.

```
public void deletePhone(int phone_id){  
    informationDao.deletePhone(phone_id);  
}
```

=====

InformationDao

@Stateless

@LocalBean

@Named

getAddress envia al Bean retorna la dirección actual de un usuario especifico. Una vez encontrada una dirección valida para dicho usuario, se envía al Bean.

```
public String getAddress() {  
    String address = "";  
    People people = new People();  
    people = getCurrentPeople();  
    List<Addresses> ListOfAddress = new LinkedList<Addresses>();  
    ListOfAddress = (List<Addresses>) people.getAddressesCollection();  
    for (Addresses n : ListOfAddress) {  
        if (n.getValiditystatusId().getId() == 1) {  
            address = n.getAddress();  
        }  
    }  
}
```

```
        break;    }    }  
return address; }
```

InsertInfo esta función de tipo void guarda en la base de datos el nombre de una persona o lo actualiza en el caso ya posea uno asignado. Los nombres no serán borrados bajo ninguna circunstancia, solo serán deshabilitados.

```
public void InsertInfo(String firstname, String middlename, String lastname) {  
    try {  
        People people = new People();  
        people = getCurrentPeople();  
        List<Names> listOfNames = new LinkedList<Names>();  
        listOfNames = (List<Names>) people.getNamesCollection();  
        Validitystatus vstatus = new Validitystatus();  
  
        Names name = new Names();  
        name.setFirstname(firstname);  
        name.setMiddlename(middlename);  
        name.setLastname(lastname);  
        name.setPeopleId(people);  
        name.setValiditystatusId(dao.findById(Validitystatus.class, 1));  
        name.setId(getNextNameId());  
    };  
  
    //if (!nameaux.getFirstname().equals(name.getFirstname()) ||  
    !nameaux.getMiddlename().equals(name.getMiddlename()) ||  
    !nameaux.getLastname().equals(name.getLastname()))  
        dao.save(name);
```

```
} catch (Exception ex) {  
    ex.printStackTrace();    } }
```

InsertAddress almacena en la base de datos una dirección para el usuario que ha iniciado sesión.

```
public void InsertAddress(String Address) {  
    try {  
        People people = new People();  
        people = getCurrentPeople();  
        List<Addresses> listOfAddress = new LinkedList<Addresses>();  
        listOfAddress = (List<Addresses>) people.getAddressesCollection();  
        Validitystatus vstatus = new Validitystatus();  
  
        for (Addresses addr : listOfAddress) {  
            if (addr.getValiditystatusId().getId() == 1) {  
                vstatus = dao.findByld(Validitystatus.class, 2);  
                addr.setValiditystatusId(vstatus);  
                dao.update(addr);  
                break;  
            }  
        }  
  
        Addresses address = new Addresses();  
        Addresstypes addrtype = new Addresstypes();  
        address.setId(getNextAddressId());  
        address.setAddress(Address);  
        address.setPeopleId(people);
```

```

        address.setValiditystatusId(dao.findById(Validitystatus.class, 1));
        address.setAdresstypeld(dao.findById(Adresstypes.class, 1));
        dao.save(address);
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

```

savePhoneNumber almacena en la base de datos un numero telefónico y un tipo de numero telefónico.

```

public void savePhoneNumber(String phoneNumer, int phonetype_id) {
    try {
        Phones phone = new Phones();
        People people = new People();
        people = getCurrentPeople();
        Phonestypes phonestypes = new Phonestypes();
        phonestypes = dao.findById(Phonestypes.class, phonetype_id);
        phone.setPhone(phoneNumer);
        phone.setPeopleId(people);
        phone.setPhonetypeId(phonestypes);
        phone.setId(getNextPhoneId());
        dao.save(phone);
    } catch (Exception ex) {
        ex.printStackTrace();
    } }

```

getAllCarrers esta funcion retorna una lista de tipo Education no formateada, para ser procesada por el Bean y posteriormente a la vista.

```
public List<Education> getAllCareers(int titleSelected) {  
  
    List<Education> listOfCarrers = new LinkedList<Education>();  
  
    try {  
  
        List<Integer> listOfIds = (List<Integer>) entityManager.createNativeQuery("SELECT e.id  
FROM Education e WHERE e.titletype_id = " + titleSelected ).getResultList();  
  
        Titletypes titleTypes = new Titletypes();  
  
        titleTypes = dao.findByld(Titletypes.class, titleSelected);  
  
        listOfCarrers = (List<Education>) titleTypes.getEducationCollection();  
  
    } catch (Exception ex) {  
  
        ex.printStackTrace();  
  
    }  
  
    return listOfCarrers;  
  
}
```

savePeopleEducation almacena en la base de datos un estudio de pregrado y/o postgrados al usuario que ha iniciado sesión.

```
public Boolean savePeopleEducation(int education_id) {  
  
    Boolean result = false;  
  
    try {  
  
        Peopleeducation peducation = new Peopleeducation();  
  
        People people = new People();  
  
        people = getCurrentPeople();  
  
        Education education = new Education();  
  
        List<Peopleeducation> listOfcareer = new LinkedList<Peopleeducation>();  
  
        listOfcareer = (List<Peopleeducation>) getCurrentPeople().getPeopleeducationCollection();  
  
        for (Peopleeducation pe : listOfcareer) {
```

```

        if (pe.getEducationId().getId() == education_id) {
            return false;
        }
    }

    education = dao.findById(Education.class, education_id);
    peducation.setId(getNextPeopleEducationId());
    peducation.setEducationId(education);
    peducation.setPeopleId(people);
    dao.save(peducation);
    result = true;
} catch (Exception ex) {
    ex.printStackTrace();
    result = false;
}
return result;
}

```

getAllPeopleCareer es una función que retorna una lista de peopleEducation con el listado de estudios asignados a una persona específica.

```

public List<Peopleeducation> getAllPeopleCareer() {
    List<Peopleeducation> listOfcareer = new LinkedList<Peopleeducation>();
    People people = new People();
    try {
        listOfcareer = (List<Peopleeducation>) getCurrentPeople().getPeopleeducationCollection();
    } catch (Exception ex) {
        ex.printStackTrace();
    }
    return listOfcareer; }

```

deletePhone elimina de la base de datos un numero telefónico asignado al usuario que inicio sesión.

```
public void deletePhone(int phone_id){  
    try{  
        dao.delete(dao.findById(Phones.class, phone_id));  
    } catch (Exception ex) {  
        ex.printStackTrace();  
    }  
}
```

getListOfExperiences obtiene la lista de experiencias laborales del usuario que inicio sesión.

```
public List<experienceViewModel> getListOfExperiences(){  
    List<Experiences> listOfExperiences = new LinkedList<Experiences>();  
    List<experienceViewModel> listOfResult = new LinkedList<experienceViewModel>();  
    experienceViewModel viewModel;  
    try{  
        People people = new People();  
        people = getCurrentPeople();  
        listOfExperiences = (List<Experiences>)people.getExperiencesCollection();  
        for(Experiences exp : listOfExperiences ){  
            viewModel = new experienceViewModel();  
            viewModel.setId(exp.getId());  
            viewModel.setWorkPlace(exp.getWorkplace());  
            viewModel.setDescription(exp.getDescription());  
            listOfResult.add(viewModel);  
        }  
    }  
}
```

```
catch(Exception ex){
    ex.printStackTrace();
} return listOfResult; }
```

saveExperience recibe un lugar de trabajo y una descripción del puesto de trabajo para asignarlo a un usuario.

```
public void saveExperience(String workplace, String Description){
    try {
        Experiences experience = new Experiences();
        experience.setWorkplace(workplace);
        experience.setDescription(Description);
        Jobs jobs = new Jobs();
        jobs = dao.findById(Jobs.class, 1);
        experience.setJobId(jobs);
        experience.setPeopleId(getCurrentPeople());
        experience.setId(getNextExperienceId());
        dao.save(experience);
    } catch (Exception ex) {
        ex.printStackTrace();
    } }
```

Esta función genera un hash md5 a partir de una contraseña ingresada por un usuario y la almacena encriptada en la base de datos.

```
public String md5(String input) {
    String md5 = null;
    if(null == input) return null;
    try {
```

```

MessageDigest digest = MessageDigest.getInstance("MD5");
digest.update(input.getBytes(), 0, input.length());
md5 = new BigInteger(1, digest.digest()).toString(32);
} catch (NoSuchAlgorithmException e) {
    e.printStackTrace();
}public void updatePassword(People people, String newPassword){
try{
    people.setIsautogenerate(false);
    people.setPassword(md5(newPassword));
    dao.update(people);
}catch(Exception ex){
    ex.printStackTrace();
}
}

```

updatePassword Actualiza una contraseña para el usuario que ha iniciado sesión, realizando una conversión a md5.

```

public void updatePassword(String newPassword){
try{
    People people = new People();
    people = getCurrentPeople();
    people.setIsautogenerate(false);
    people.setPassword(md5(newPassword));
    dao.update(people);
}catch(Exception ex){
    ex.printStackTrace();    } }

```

resetPassword reinicia una contraseña basándose en un valor autogenerado GUID.

```
public String resetPassword(int people_id){  
    String newPass = "";  
  
    try{  
        People people = new People();  
  
        people = dao.findById(People.class, people_id);  
  
        UUID pass = UUID.randomUUID();  
  
        System.out.println("La nueva contraseña es: " + pass.toString());  
  
        newPass = pass.toString();  
  
        newPass=newPass.replace("-", "");  
  
        updatePassword(people, newPass);  
  
    }catch(Exception ex){  
        ex.printStackTrace();  
    }    return newPass; }  
  
return md5; }
```

=====

JobOffersBean

@EJB

@ManagedBean

@ViewScoped

public void save(){

String rescarreras="";

String rescarrerasM="";

String rescarrerasD="";

```

if(selectedcarreras.size())>0){
    for(Integer i=0; i<selectedcarreras.size(); i++){
        rescarreras+= selectedcarreras.get(i);
        rescarreras+=",";    }    }
if(selectedcarrerasM.size())>0){
    for(Integer i=0; i<selectedcarrerasM.size(); i++){
        rescarrerasM+= selectedcarrerasM.get(i);
        rescarrerasM+=",";    }    }
if(selectedcarrerasD.size())>0){
    for(Integer i=0; i<selectedcarrerasD.size(); i++){
        rescarrerasD+= selectedcarrerasD.get(i);
        // if(i+1 < selectedcarrerasD.size() ){
        rescarrerasD+=",";
        //}    }    }

    jobsdao.add(gender, Subject, rescarreras, rescarrerasM, rescarrerasD, Description, maxage,
minage); }

```

=====

JobOffersDao

@EJB

@PersistenceContext

@Stateless

@LocalBean

```

public List<Joboffers> ListOfapplications(int Minage,int Maxage,List<String> Carreras,List<String>
CarrerasM,List<String> CarrerasD,int exp){

```

```

    System.out.println("Aplicaciones");

```

```

List<Joboffers> Ldevuelta=new LinkedList<Joboffers>();

People actual=getCurrentPeople();

Date anho=new Date();

int edad=anho.getYear()-actual.getBirthday().getYear();

try{

    String sql = "SELECT * FROM joboffers WHERE minage <= " + String.valueOf(Minage) + "
AND maxage >= " + String.valueOf(Maxage) + " AND experience=" + exp;

    if(Carreras.size())>0){

        sql+=" AND ";

        for (int z = 0; z < Carreras.size(); z++) {

            String temp = "," + Carreras.get(z) + ",";

            if (z == Carreras.size() - 1) {

                sql += "grade LIKE '%" + temp + "%'";

            } else {

                sql += "grade LIKE '%" + temp + "%' OR ";           }       }}

    if(CarrerasM.size())>0){

        sql+=" AND ";

        for (int z = 0; z < CarrerasM.size(); z++) {

            String temp = "," + CarrerasM.get(z) + ",";

            if (z == CarrerasM.size() - 1) {

                sql += "master LIKE '%" + temp + "%'";

            } else {

                sql += "master LIKE '%" + temp + "%' OR ";           }       }       }

    if(CarrerasD.size())>0){

        sql+=" AND ";

```

```

for (int z = 0; z < CarrerasD.size(); z++) {
    String temp = "," + CarrerasD.get(z) + ",";
    if (z == CarrerasD.size() - 1) {
        sql += "dr LIKE '%" + temp + "%'";
    } else {
        sql += "dr LIKE '%" + temp + "%' OR ";
    }
}

System.out.println(sql);

Ldevuelta=(List<Joboffers>) entityManager.createNativeQuery(sql,
Joboffers.class).getResultList();

}catch(Exception e){
}

return Ldevuelta;
}

```

=====

NewsBean

implements Serializable

@ManagedBean

@ViewScoped

getListOfNews obtiene una lista de noticias.

```

public List<newViewModel> getListOfNews() {
    listOfNews = newsdao.getListOfNews();
    return listOfNews;
}

```

add agrega noticias a la base de datos.

```

public void add(){
    newsdao.saveNew(newtoadd);
}

```

delete se relaciona con NewsDao y realiza un borrado el articulo/noticia tomando como criterio la llave primaria de la tabla.

```
public void delete(int idnews){  
    newsdao.deleteNews(idnews);  
}
```

=====

newsDao

implements Serializable

@Stateless

@LocalBean

saveNew recibe una noticia y la almacena en la base de datos con la fecha y hora en la que se ingreso con respeto a la hora del servidor y asignándosela al administrador que ingreso la noticia.

```
public void saveNew(String notice){  
    try {  
        News news = new News();  
        People people = new People();  
        news.setId(getNextNewsId());  
        news.setFinishdatepost(new Date.now());  
        news.setNotice(notice);  
        // people = dao.findById(People.class, getCurrentPeople().getId());  
        news.setPeopleId(getCurrentPeople());  
        dao.save(news);  
    } catch (Exception ex) {  
        ex.printStackTrace();    } } }
```

deleteNews Borra una noticia de la base de datos basándose en su llave primaria.

```
public void deleteNews(int idnews){  
    try{  
        News news = dao.findById(News.class , idnews);  
        dao.delete(news);  
    }catch(Exception ex){  
        ex.printStackTrace();    } }
```

getListOfNews retorna una lista de noticias actuales para ser mostradas en el slide derecho con su respectiva fecha y autor.

```
public List<newViewModel> getListOfNews(){  
    List<newViewModel> listOfNews = new LinkedList<newViewModel>();  
    List<News> allNews = new LinkedList<News>();  
    newViewModel viewModel;  
    try{  
        allNews = dao.getAll(News.class);  
        for(int i=allNews.size()-1;i>=0;i--){  
            viewModel=new newViewModel();  
            viewModel.setId(allNews.get(i).getId());  
            viewModel.setMessage(allNews.get(i).getNotice());  
            viewModel.setPeopleId(allNews.get(i).getPeopleId().getId());  
            viewModel.setMessageDate(allNews.get(i).getDate());  
            listOfNews.add(viewModel);  
        }  
    }catch(Exception ex){  
        ex.printStackTrace();    }  
    return listOfNews; }
```

Verifica si la persona loggeada es un administrador. A partir de esta función se otorgan permisos y muestran vínculos (Validados únicamente para los administradores) para realizar operaciones administrativas

```
public boolean isAdmin(){  
    People persona=getCurrentPeople();  
    if(persona.getPeopletypeld().getId()==1){  
        return true;  
    }else{  
        return false;  
    }  
}
```

CONCLUSIONES Y RECOMENDACIONES.

CAPÍTULO 6.

6.1 CONCLUSIONES.

- ✓ El sistema contribuye al establecimiento del control, seguimiento y comunicación por parte de la Unidad de Postgrados con la comunidad de estudiantes y graduados(as).
- ✓ Con este sistema se ha logrado que la elaboración de informes sea más rápida y brinde diferentes opciones sobre la información que se necesite presentar en estos.
- ✓ Se ha logrado centralizar la información de la comunidad estudiantil y graduados(as) de la Unidad de Postgrados; lo que permitirá reducir errores en datos, eliminando duplicidad o pérdida de estos al elaborar reportes.
- ✓ El uso de este sistema permitirá a los usuarios como estudiantes y graduados(as), la oportunidad de optar a ofertas de trabajo que la Unidad de Postgrados reciba de diferentes empresas.

6.2 RECOMENDACIONES.

- ✓ Si a futuro la Unidad de Postgrado desea implementar su propio servidor, para alojar en este el Sistema SICSGUP, se evaluó el equipo que puede ser tomado en cuenta para esto; se le indica al encargado de evaluación de hardware que los precios de estos pueden estar sujetos a cambio desde el mismo día en que se realizaron las consultas del equipo (Anexo H).
- ✓ Ante cualquier inquietud con el uso del sistema consultar el manual del usuario.

REFERENCIAS BIBLIOGRÁFICAS.

Trabajos de Grado:

- ✓ Flores Lucero, Cecybeth María; 2011, Diseño, Desarrollo e Implementación de un Sistema Informático para la Automatización de Trámites de Servicio Social realizados en la Facultad Multidisciplinaria de Occidente de la Universidad de El Salvador.
- ✓ Cornejo Moran, Erick Alexis; 2011, Desarrollo de un Sistema para la Realización de Tramites Académicos – Administrativos para la Unidad de Administración Académica de la Universidad de El Salvador de la Facultad Multidisciplinaria de Occidente.
- ✓ Barrera Flores, Luis Alonso; 2008, Diseño, Desarrollo e Implementación de un Sistema para el Registro y Control de Activo Fijo de la Facultad Multidisciplinaria de Occidente.

Libros y/o Documentos:

- ✓ E. Kendall, Kenneth y E. Kendall, Julie; 2005 Análisis y Diseño de Sistema; Sexta edición, Pearson Education. México.
- ✓ Oracle and/or its affiliates, Julio 2012 Tutorial Java EE 6; Web del autor: <http://docs.oracle.com/javaee/6/tutorial/doc/>.

ANEXOS

ANEXO A – DESARROLLO CAPITULAR.

CAPITULO 1 GENERALIDADES

- 1.1 Introducción
- 1.2 Objetivos
- 1.3 Antecedentes
- 1.4 Planteamiento del problema
- 1.5 Justificación
- 1.6 Alcances
- 1.7 Limitantes

CAPITULO 2 METODOLOGÍA DE INVESTIGACIÓN.

- 2.1 Marco teórico.
- 2.2 Metodología de trabajo.
 - 2.2.1 Fase de planificación y recopilación de información.
 - 2.2.2 Fase de diseño.
 - 2.2.3 Fase de desarrollo.
 - 2.2.4 Fase de implementación.
- 2.3 Técnicas de investigación Utilizadas.
 - 2.3.1 Encuesta
 - 2.3.2 Interpretación de Resultados.
- 2.4 Situación Actual.
 - 2.4.1 Resultado de entrevista al Personal y Jefatura de Unidad de Postgrado.
- 2.5 Planteamiento de la Solución.

CAPITULO 3 DISEÑO Y DESARROLLO

- 3.1 Definición de requerimientos del Sistema
- 3.2 Análisis de factibilidad.
- 3.3 Diseño Lógico de Sistema
- 3.4 Diseño de Interfaces.
- 3.5 Diseño de Base de datos.
 - 3.5.1 Modelo físico de la Base de Datos.
 - 3.5.2 Modelado Entidad-Relación.
 - 3.5.3 Diccionario de Datos.
- 3.6 Requerimientos de seguridad del sistema.

CAPITULO 4 IMPLEMENTACIÓN.

- 4.1 Plan de implementación.
- 4.2 Costo de implementación.

CAPITULO 5 DOCUMENTACIÓN DEL SISTEMA.

- 5.1 Manual del Administrador.
- 5.2 Manual de usuario intermedio y usuario Final.

CAPITULO 6 CONCLUSIONES Y RECOMENDACIONES.

- 6.1 Conclusiones.
- 6.2 Recomendaciones.

REFERENCIAS BIBLIOGRÁFICAS.

ANEXOS.

ANEXO B – ENTREVISTA A PERSONAL ADMINISTRATIVO DE LA UNIDAD DE POSTGRADO DE LA UNIVERSIDAD DE EL SALVADOR FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE.



UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA

Entrevista a Personal Administrativo de la Unidad de Postgrados de la Universidad de El Salvador Facultad Multidisciplinaria de Occidente.

OBJETIVOS: Obtener información primordial para establecimiento de la situación actual de manejo de información sobre estudiantes y graduados(as) de postgrados de la Universidad de El Salvador FMO para establecer Límites y Alcances físicos en el desarrollo del Sistema de control de seguimiento informático, para los estudiantes de postgrado de la Universidad de El Salvador Facultad Multidisciplinaria de Occidente.

Entrevistadores:

- Mena Recinos, Verónica Lize.t
- Ramos Salazar, Héctor Manuel.
- Rivera Díaz, Benjamín Ernesto.

1-¿Cuál es su nombre? _____

2-¿Cuál es su cargo? _____

3- ¿Cuántas personas laboran en este Departamento? _____

4- ¿Con cuántos estudiantes cuenta en la actualidad la Unidad de Postgrado? _____

5-¿Qué tan a menudo se reciben ofertas de trabajo para profesionales egresados de postgrado?
Un aproximado mensual. _____

6-¿Se cuenta con un método para obtener el perfil solicitado en dichas ofertas de trabajo?

Si

No

Si la respuesta es No pase a la pregunta 8.

7-Si la respuesta a la anterior es SI ¿Cuánto tiempo aproximadamente se tardan en obtener la información?

Horas

Días

Semanas

Número aproximado: _____

8-¿De que forma la Unidad de Postgrado comunica sus actividades y avisos a sus actuales alumnos?

Por Teléfono

Por e-Mail

Directamente o en Persona

Otro

Especificar: _____

9-¿Mantiene comunicación con los estudiantes Graduados(as) para actualizar sus datos?

Si

No

10-¿Cuántos graduados(as) han logrado optar por ofertas de trabajo que las empresas dirigen a la Unidad de Postgrado? _____

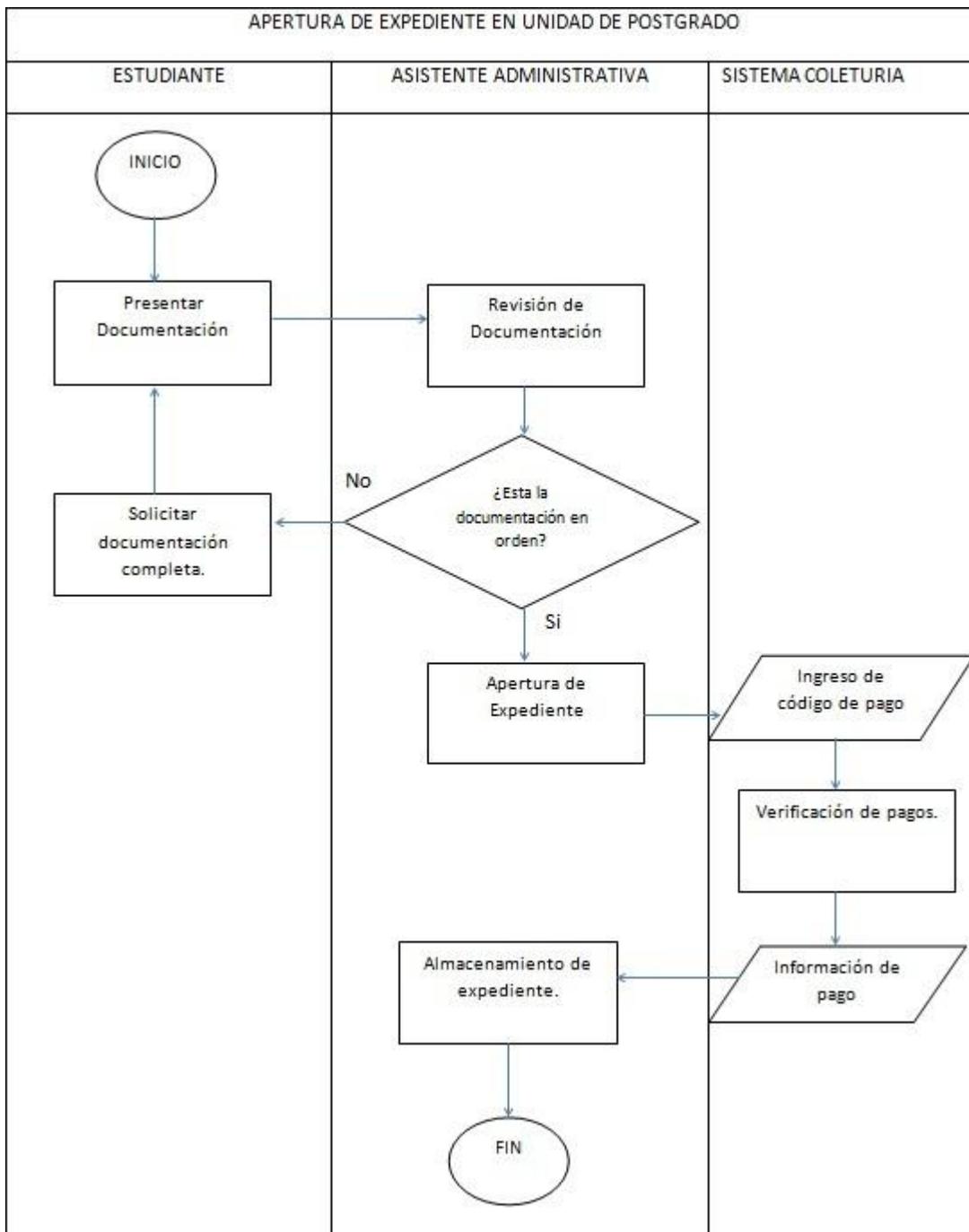
11-¿Cree Usted que la difusión de este tipo de información seria mas ágil si es publicada en un sitio Web?

Si

No

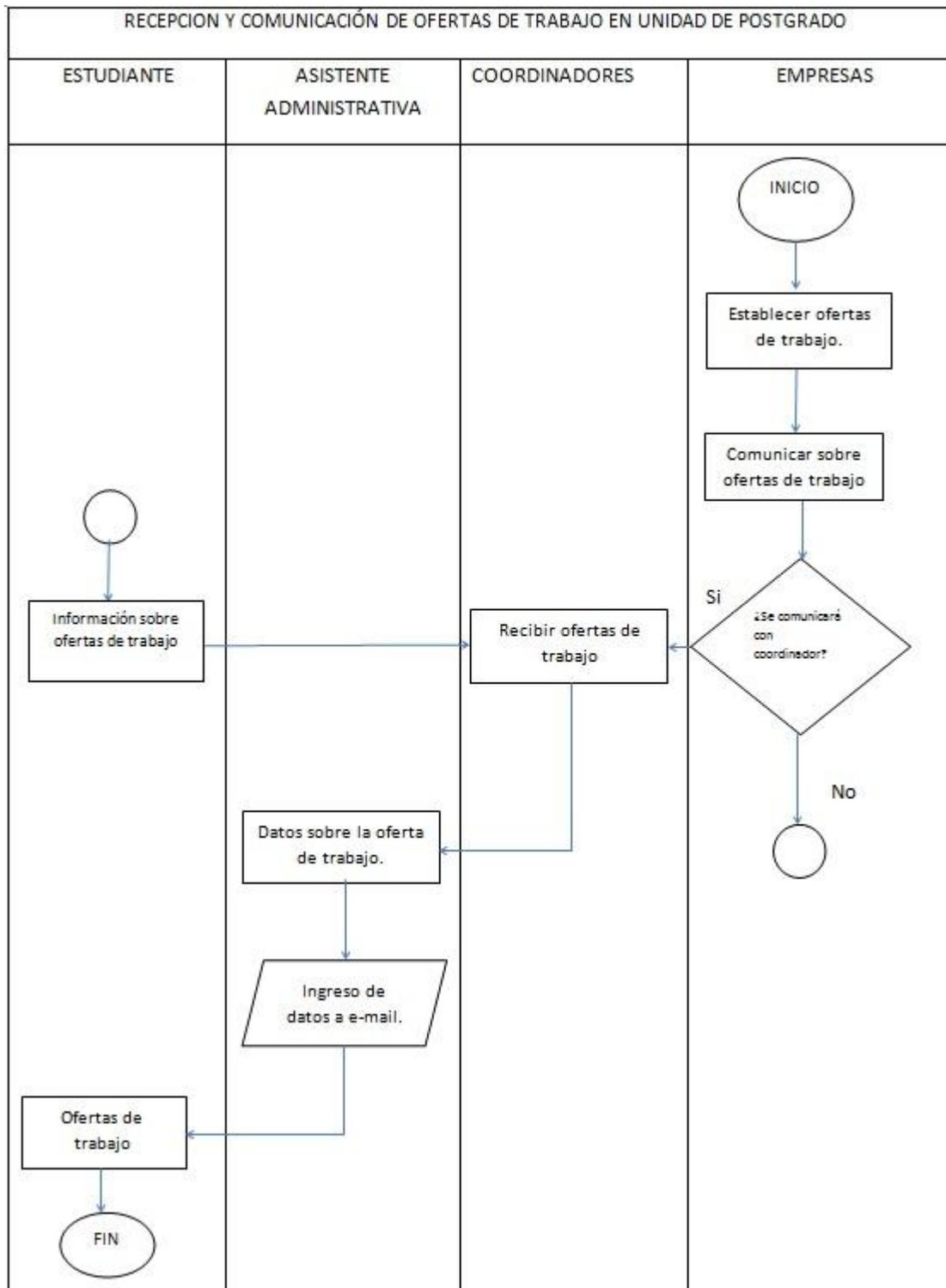
12- ¿En qué podría ayudar a la Unidad de Postgrado poder mantener información actualizada sobre sus egresados y actuales alumnos? Mencione algunos puntos:

ANEXO C - DIAGRAMA DE FLUJO DE DATOS (ALMACENAMIENTO DE EXPEDIENTE).



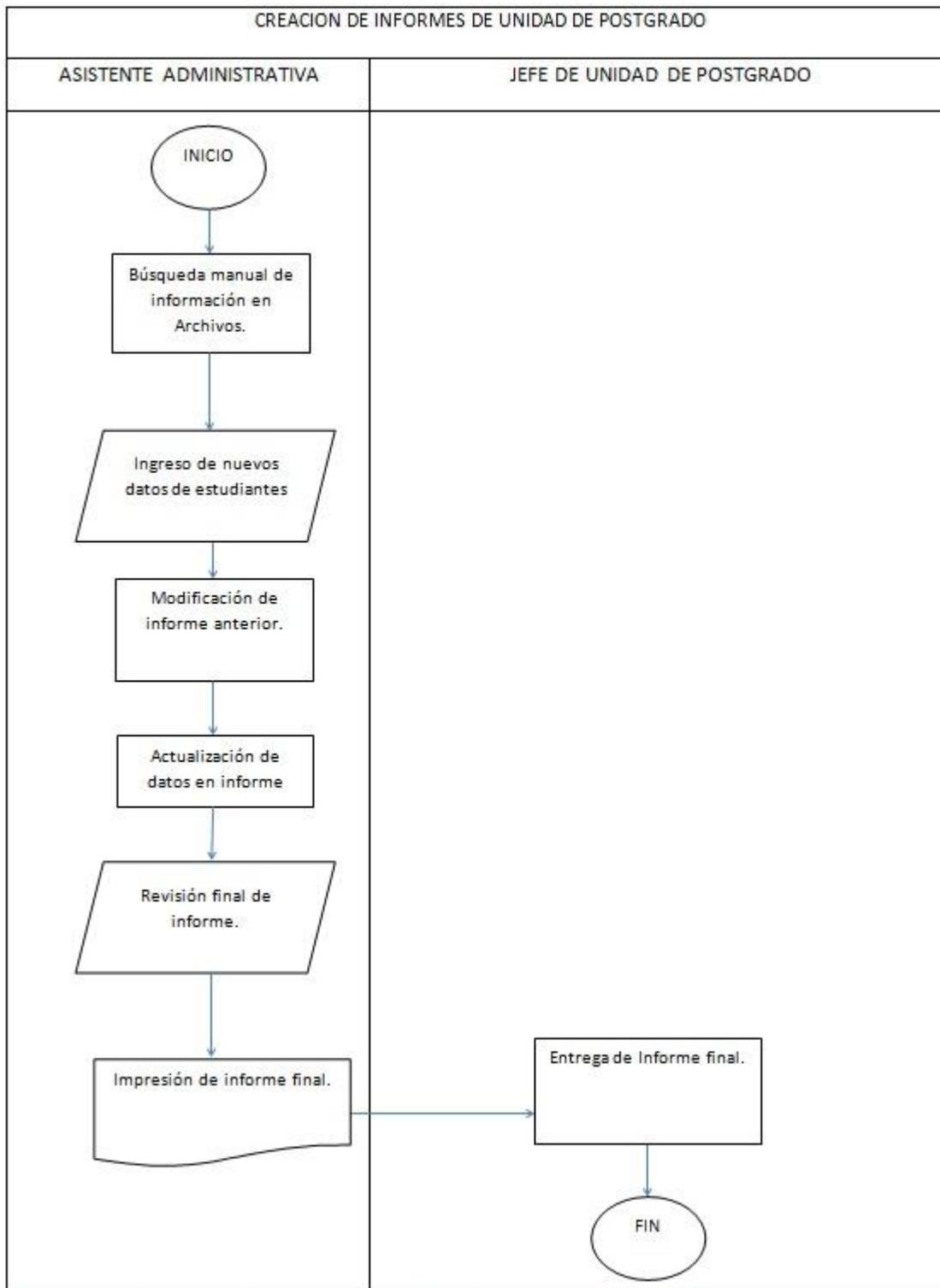
(Diagrama de Flujo de Funciones Enlazadas – APERTURA DE EXPEDIENTE - Diseño Propio)

ANEXO D - DIAGRAMA DE FLUJO DE DATOS (MANEJO DE OFERTAS DE TRABAJO).



(Diagrama de Flujo de Funciones Enlazadas – MANEJO DE OFERTAS DE TRABAJO - Diseño Propio)

ANEXO E- DIAGRAMA DE FLUJO DE DATOS (ELABORACIÓN DE INFORMES).



(Diagramas de Flujo de Funciones Enlazadas – CREACIÓN DE INFORMES - Diseño Propio)

ANEXO F- UNIVERSIDADES SONDEADAS-SITIOS WEB OFICIALES.

Universidad Albert Einstein. www.uae.edu.sv/

Universidad Alberto Masferrer. www.usam.edu.sv/

Universidad Centroamericana José Simeón Cañas. www.uca.edu.sv/

Universidad Capitán General Gerardo Barrios. www.ugb.edu.sv/

Universidad Evangélica de El Salvador. www.uees.edu.sv/

Universidad Tecnológica de El Salvador. www.utec.edu.sv/

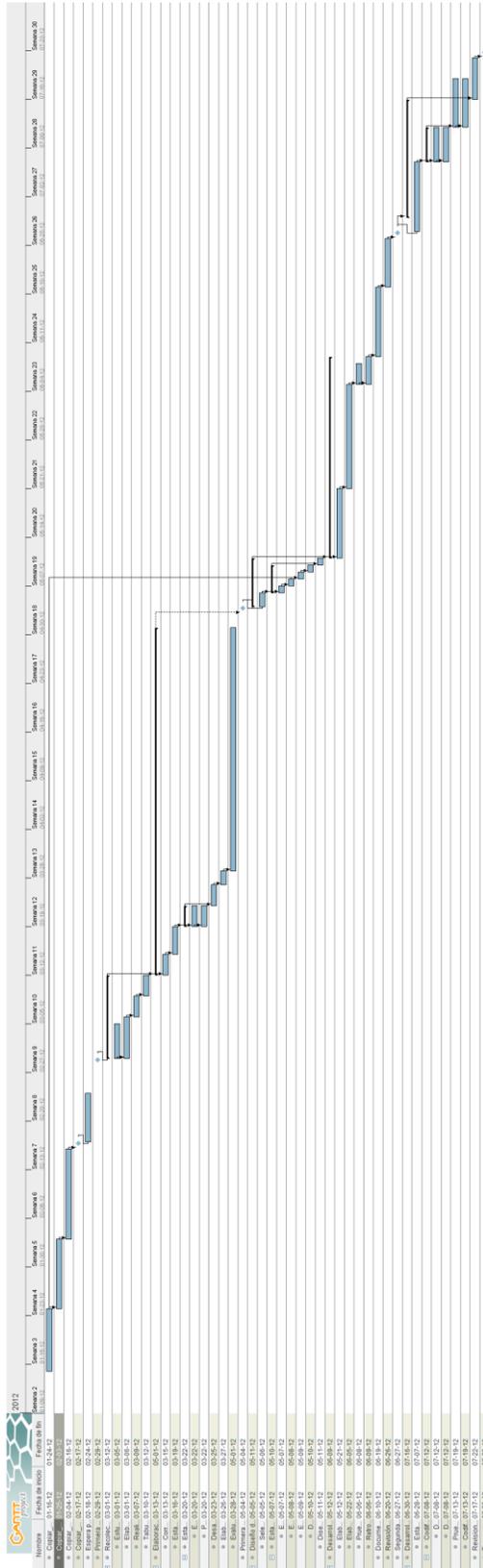
Universidad Francisco Gavidia. www.utec.edu.sv/

Universidad Autónoma de Santa Ana. www.unasa.edu.sv/

Universidad Católica de El Salvador. www.catolica.edu.sv/

Universidad Don Bosco. www.udb.edu.sv/

ANEXO G- CRONOGRAMA.



ANEXO H- ESPECIFICACIONES PARA EL SERVIDOR DE LA UNIDAD DE POSTGRADOS.

✓ HP PROLIANT ML150 G6 E5504 2.0GHZ QUAD CORE HOT PLUG SAS/SATA TOWER SERVER.



El HP ProLiant ML150 G6 es ampliable y asequible para los negocios en crecimiento. Las fuentes de alimentación redundantes y el ventilador redundante ofrecen más protección de la corriente y de la temperatura para que su negocio funcione sin problemas. La seguridad de los datos, una de las mayores preocupaciones, queda asegurada por TPM, que ofrece cifrado de datos. Las empresas con ubicaciones remotas pueden aprovechar

las ventajas de la función de licencias avanzadas de gestión remota Lights-Out 100i del ML150 G6 para reducir el tiempo de inactividad de la empresa o el servidor y aumentar la productividad total. Ahora se entrega con implementación y control mejorados con CD de configuración sencilla e imagen ISO para descarga, además de Insight Management Agents para gestión SIM. En total, el HP ProLiant ML150 G6 respeta su presupuesto de informática con espacio para hacer crecer su negocio.

PRECIO: \$ 1,951.25 USD. (Más costo de envío). El precio puede estar sujeto a cambios sin previo aviso.

Procesador	
Descripción de procesador	Intel® Xeon® E5504 (2.00 GHz, 4MB L3, 80W, DDR3-800)
Velocidad del procesador	2.00 GHz
Tipo de procesador	Intel® Xeon® E5504 (2.00 GHz, 4MB L3, 80W, DDR3-800)
Nombre del procesador	Intel® Xeon® E5504 (2.00 GHz, 4MB L3, 80W, DDR3-800)
Memoria	
Memoria máxima	48GB
Características de protección de memoria	Advanced ECC
Unidades	
Unidades de disco rígido incluidas	None ship standard

Conexión de red	
Controlador de red	Embedded HP NC107i PCI Express Gigabit Server Adapter
Almacenamiento	
Capacidad de almacenamiento	SAS: 4.0TB SATA: 4.0TB
Conexión de almacenamiento estándar	Hot plug 3.5-inch SAS; Hot plug 3.5-inch SATA; Non-hot plug 3.5-inch SAS; Non-hot plug 3.5-inch SATA
Características de sistema	
Detalle de administración remota	Lights-Out 100i Remote Management IPMI 2.0 compliance
Software	
Software de administración remota	Lights-Out 100i Remote Management IPMI 2.0 compliance
Chasis	
Chasis del "Form factor"	Tower
Configuración total del "Form factor"	5U
Dimensiones y peso	
Dimensiones	20 x 42.4 x 61.68 cm

HP ProLiant ML150 G6, Sitio **hp** América Central http://www.hp.com/latam/catalogo/ac/ml100_smb/sp/466132-001.html; Compras http://www8.hp.com/lamerica_nsc_cnt_amer/es/contact-hp/how-to-buy.html. 20 Julio de 2012.

✓ **SERVICIO WEB HOSTING.**

Puede utilizarse el mismo que utiliza la Facultad Multidisciplinaria de Occidente o puede visitar estos sitios:

- <http://www.mochahost.com/>
- <http://www.hostdime.com/>

✓ MAQUINA DE ESCRITORIO VOSTRO 470 DE DELL.



Precio: \$ 875.00 (Más costo de envío). El precio puede estar sujeto a cambios sin previo aviso.

Características:	
Procesador	Intel® Core™ i5-3450 de tercera generación (6MB Caché, hasta 3.50 GHz).
Sistema operativo	Windows® 7 Professional, 64-bit, con modo XP Instalado, Español.
Monitor	Monitor Dell Serie E E2011H de 20" Full HD, VIS/VGA.
Memoria³	4 GB ⁴ Dos canales SDRAM DDR3 a 1600 MHz.
Disco duro	Disco Duro SATA de 500GB 7200 RPM (3.0 Gb/s).
Tarjeta de video	NVIDIA® GeForce® GT 620 1GB DDR3.
Software de seguridad	Trend Micro™ Worry-Free™ Business Security Services, 30 días.
Garantía	1 Año de ProSupport para IT, con respuesta al siguiente día laborable.
Unidad óptica	Unidad de 16x (DVD +/- RW), lectura y escritura de CD/DVD.

http://www.dell.com/sv/empresas/p/vostro-470/pd?oc=vd47btpes&model_id=vostro-470 20 JULIO 2012.

ANEXO I- SCRIPT DE LA BASE DE DATOS.

```
--
-- PostgreSQL database dump
--

-- Dumped from database version 9.1.3
-- Dumped by pg_dump version 9.1.3
-- Started on 2012-08-03 23:52:11 CST

SET statement_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SET check_function_bodies = false;
SET client_min_messages = warning;

CREATE EXTENSION IF NOT EXISTS plpgsql WITH SCHEMA pg_catalog;

COMMENT ON EXTENSION plpgsql IS 'PL/pgSQL procedural language';

SET search_path = public, pg_catalog;

CREATE SEQUENCE address_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE public.address_id_seq OWNER TO admin;

SELECT pg_catalog.setval('address_id_seq', 1, false);

SET default_tablespace = '';

SET default_with_oids = false;

CREATE TABLE addresses (
    id integer NOT NULL,
    addresstype_id integer NOT NULL,
    people_id integer NOT NULL,
    validitystatus_id integer NOT NULL,
    address text NOT NULL,
    department_id integer
);

ALTER TABLE public.addresses OWNER TO admin;
```

```

CREATE SEQUENCE adresstype_id_seq
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER TABLE public.adresstype_id_seq OWNER TO admin;

SELECT pg_catalog.setval('adresstype_id_seq', 1, false);

CREATE TABLE adresstypes (
  id integer NOT NULL,
  adresstype character varying(15) NOT NULL
);

ALTER TABLE public.adresstypes OWNER TO admin;

CREATE TABLE attaches (
  id integer NOT NULL,
  people_id integer NOT NULL,
  attachtype_id integer NOT NULL,
  attach_name character varying(100) NOT NULL
);

ALTER TABLE public.attaches OWNER TO admin;

CREATE TABLE attachtypes (
  id integer NOT NULL,
  type character varying(15) NOT NULL
);

ALTER TABLE public.attachtypes OWNER TO admin;

CREATE SEQUENCE carrers_id_seq
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER TABLE public.carrers_id_seq OWNER TO admin;

SELECT pg_catalog.setval('carrers_id_seq', 1, false);

CREATE TABLE comments (
  id integer NOT NULL,
  post_id integer NOT NULL,

```

```

        people_id integer NOT NULL,
        comment character varying(250) NOT NULL,
        validitystatus_id integer NOT NULL
    );

ALTER TABLE public.comments OWNER TO admin;

CREATE TABLE commentsgroups (
    id integer NOT NULL,
    post_id integer NOT NULL,
    people_id integer NOT NULL,
    commentpostgroup character varying(250) NOT NULL,
    validitystatus_id integer NOT NULL
);

ALTER TABLE public.commentsgroups OWNER TO admin;

CREATE TABLE departments (
    id integer NOT NULL,
    departmentname character varying(15)
);

ALTER TABLE public.departments OWNER TO admin;

CREATE TABLE education (
    id integer NOT NULL,
    titletype_id integer NOT NULL,
    careername character varying(100) NOT NULL
);

ALTER TABLE public.education OWNER TO admin;

CREATE TABLE experiences (
    id integer NOT NULL,
    people_id integer NOT NULL,
    job_id integer NOT NULL,
    workplace text NOT NULL,
    description text NOT NULL,
    begindate date,
    enddate date,
    sector_id integer
);

ALTER TABLE public.experiences OWNER TO admin;

CREATE TABLE friendships (
    id integer NOT NULL,
    friendr integer NOT NULL,
    friends integer NOT NULL
);

ALTER TABLE public.friendships OWNER TO admin;

CREATE TABLE groups (

```

```
        id integer NOT NULL,  
        groupname character varying(100) NOT NULL,  
        people_id integer NOT NULL,  
        isactive boolean NOT NULL  
    );
```

```
ALTER TABLE public.groups OWNER TO admin;
```

```
CREATE TABLE joboffers (  
    id integer NOT NULL,  
    gender integer,  
    minage integer,  
    maxage integer,  
    subject character varying,  
    description character varying,  
    grade character varying,  
    master character varying,  
    dr character varying,  
    experience integer  
);
```

```
ALTER TABLE public.joboffers OWNER TO admin;
```

```
CREATE TABLE jobs (  
    id integer NOT NULL,  
    job text NOT NULL  
);
```

```
ALTER TABLE public.jobs OWNER TO admin;
```

```
ALTER TABLE public.knowledge OWNER TO admin;
```

```
CREATE TABLE logs (  
    id integer NOT NULL,  
    people_id integer NOT NULL,  
    dates date,  
    hour time without time zone,  
    jsessionid character varying(32) NOT NULL  
);
```

```
ALTER TABLE public.logs OWNER TO admin;
```

```
CREATE TABLE memberships (  
    id integer NOT NULL,  
    group_id integer NOT NULL,  
    people_id integer NOT NULL,  
    statuspetition_id integer  
);
```

```
ALTER TABLE public.memberships OWNER TO admin;
```

```
CREATE TABLE messages (  
    id integer NOT NULL,  
    peoplesender integer NOT NULL,
```

```

        peoplereceiver integer NOT NULL,
        message character varying(400),
        validitystatus_id integer NOT NULL,
        datemessage date,
        hourmessage character varying
    );

ALTER TABLE public.messages OWNER TO admin;

CREATE TABLE municipio (
    id integer NOT NULL,
    department_id integer,
    municipioname character varying(50)
);

ALTER TABLE public.municipio OWNER TO admin;

CREATE TABLE names (
    id integer NOT NULL,
    firstname character varying(25) NOT NULL,
    people_id integer NOT NULL,
    middlename character varying(25),
    lastname character varying(25) NOT NULL,
    validitystatus_id integer NOT NULL
);

ALTER TABLE public.names OWNER TO admin;

CREATE SEQUENCE names_seq_id
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE public.names_seq_id OWNER TO admin;

SELECT pg_catalog.setval('names_seq_id', 1, false);

CREATE TABLE news (
    id integer NOT NULL,
    people_id integer,
    datepost character varying,
    finishdatepost character varying,
    notice character varying
);

ALTER TABLE public.news OWNER TO admin;

CREATE TABLE people (
    id integer NOT NULL,
    birthday date,
    email character varying(25) NOT NULL,
    gender integer,
    peopletype_id integer NOT NULL,
    password character varying(100) NOT NULL,
    isautogenerate boolean NOT NULL,
    signupdate character varying NOT NULL

```

```

);

ALTER TABLE public.people OWNER TO admin;

CREATE SEQUENCE people_id_seq
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER TABLE public.people_id_seq OWNER TO admin;

SELECT pg_catalog.setval('people_id_seq', 1, false);

CREATE TABLE peopleeducation (
  id integer NOT NULL,
  people_id integer NOT NULL,
  education_id integer NOT NULL,
  begindate date,
  startdate date
);

ALTER TABLE public.peopleeducation OWNER TO admin;

CREATE TABLE peopletypes (
  id integer NOT NULL,
  peopletype character varying(15) NOT NULL
);

ALTER TABLE public.peopletypes OWNER TO admin;

CREATE TABLE petitions (
  id integer NOT NULL,
  peoplesender integer NOT NULL,
  peoplereceiver integer NOT NULL,
  statuspetition_id integer NOT NULL
);

ALTER TABLE public.petitions OWNER TO admin;

CREATE SEQUENCE phone_id_seq
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER TABLE public.phone_id_seq OWNER TO admin;

```

```

SELECT pg_catalog.setval('phone_id_seq', 1, false);

CREATE TABLE phones (
    id integer NOT NULL,
    phonetype_id integer NOT NULL,
    people_id integer NOT NULL,
    phone character varying(15) NOT NULL
);

ALTER TABLE public.phones OWNER TO admin;

CREATE TABLE phonestypes (
    id integer NOT NULL,
    phonetype character varying(15) NOT NULL
);

ALTER TABLE public.phonestypes OWNER TO admin;

CREATE SEQUENCE phonetypes_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE public.phonetypes_id_seq OWNER TO admin;

SELECT pg_catalog.setval('phonetypes_id_seq', 1, false);

CREATE TABLE postgroups (
    id integer NOT NULL,
    group_id integer NOT NULL,
    postgroup character varying(400) NOT NULL,
    attach_id integer,
    validitystatus_id integer NOT NULL
);

ALTER TABLE public.postgroups OWNER TO admin;

CREATE TABLE posts (
    id integer NOT NULL,
    people_id integer NOT NULL,
    post character varying(400),
    attach_id integer,
    validitystatus_id integer NOT NULL
);

```

```

ALTER TABLE public.posts OWNER TO admin;

CREATE TABLE sectortype (
    id integer NOT NULL,
    sectorname character varying
);

ALTER TABLE public.sectortype OWNER TO admin;

CREATE TABLE statuspetitions (
    id integer NOT NULL,
    statuspetition character varying(12) NOT NULL
);

ALTER TABLE public.statuspetitions OWNER TO admin;

CREATE TABLE titletypes (
    id integer NOT NULL,
    titletype character varying(25) NOT NULL
);

ALTER TABLE public.titletypes OWNER TO admin;

CREATE SEQUENCE titletypes_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE public.titletypes_id_seq OWNER TO admin;

SELECT pg_catalog.setval('titletypes_id_seq', 1, false);

CREATE SEQUENCE user_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE public.user_id_seq OWNER TO admin;

SELECT pg_catalog.setval('user_id_seq', 1, false);

CREATE SEQUENCE usertypes_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

```

```

ALTER TABLE public.usertypes_id_seq OWNER TO admin;

SELECT pg_catalog.setval('usertypes_id_seq', 1, false);

CREATE TABLE validitystatus (
    id integer NOT NULL,
    status character varying(10) NOT NULL
);

ALTER TABLE public.validitystatus OWNER TO admin;

ALTER TABLE ONLY addresses
    ADD CONSTRAINT addresses_pkey PRIMARY KEY (id);

ALTER TABLE ONLY adresstypes
    ADD CONSTRAINT adresstypes_pkey PRIMARY KEY (id);

ALTER TABLE ONLY attachtypes
    ADD CONSTRAINT attaches_pkey PRIMARY KEY (id);

ALTER TABLE ONLY attaches
    ADD CONSTRAINT attaches_pkey1 PRIMARY KEY (id);

ALTER TABLE ONLY comments
    ADD CONSTRAINT comments_pkey PRIMARY KEY (id);

ALTER TABLE ONLY commentsgroups
    ADD CONSTRAINT commentsgroups_pkey PRIMARY KEY (id);

ALTER TABLE ONLY departments
    ADD CONSTRAINT departments_pkey PRIMARY KEY (id);

ALTER TABLE ONLY education
    ADD CONSTRAINT education_pkey PRIMARY KEY (id);

ALTER TABLE ONLY experiences
    ADD CONSTRAINT experiences_pkey PRIMARY KEY (id);

ALTER TABLE ONLY friendships
    ADD CONSTRAINT friendships_pkey PRIMARY KEY (id);

```

```
ALTER TABLE ONLY groups
  ADD CONSTRAINT groups_pkey PRIMARY KEY (id);

ALTER TABLE ONLY joboffers
  ADD CONSTRAINT joboffers_pkey PRIMARY KEY (id);

ALTER TABLE ONLY jobs
  ADD CONSTRAINT jobs_pkey PRIMARY KEY (id);

ALTER TABLE ONLY logs
  ADD CONSTRAINT logs_pkey PRIMARY KEY (id);

ALTER TABLE ONLY memberships
  ADD CONSTRAINT memberships_pkey PRIMARY KEY (id);

ALTER TABLE ONLY messages
  ADD CONSTRAINT messages_pkey PRIMARY KEY (id);

ALTER TABLE ONLY municipio
  ADD CONSTRAINT municipio_pkey PRIMARY KEY (id);

ALTER TABLE ONLY names
  ADD CONSTRAINT names_pkey PRIMARY KEY (id);

ALTER TABLE ONLY news
  ADD CONSTRAINT news_pkey PRIMARY KEY (id);

ALTER TABLE ONLY people
  ADD CONSTRAINT people_pkey PRIMARY KEY (id);

ALTER TABLE ONLY peopleeducation
  ADD CONSTRAINT peopleeducation_pkey PRIMARY KEY (id);

ALTER TABLE ONLY petitions
  ADD CONSTRAINT petitions_pkey PRIMARY KEY (id);

ALTER TABLE ONLY phones
  ADD CONSTRAINT phones_pkey PRIMARY KEY (id);

ALTER TABLE ONLY phonestypes
  ADD CONSTRAINT phonestypes_pkey PRIMARY KEY (id);
```

```

ALTER TABLE ONLY postgroups
    ADD CONSTRAINT postgroups_pkey PRIMARY KEY (id);

ALTER TABLE ONLY posts
    ADD CONSTRAINT posts_pkey PRIMARY KEY (id);

ALTER TABLE ONLY sectortype
    ADD CONSTRAINT sectortype_pkey PRIMARY KEY (id);

ALTER TABLE ONLY statuspetitions
    ADD CONSTRAINT statuspetitions_pkey PRIMARY KEY (id);

ALTER TABLE ONLY titletypes
    ADD CONSTRAINT titletypes_pkey PRIMARY KEY (id);

ALTER TABLE ONLY peopletypes
    ADD CONSTRAINT usertypes_pkey PRIMARY KEY (id);

ALTER TABLE ONLY validitystatus
    ADD CONSTRAINT validitystatus_pkey PRIMARY KEY (id);

ALTER TABLE ONLY addresses
    ADD CONSTRAINT addresses_adresstype_id_fkey FOREIGN KEY (adresstype_id)
REFERENCES adresstypes(id);

ALTER TABLE ONLY addresses
    ADD CONSTRAINT addresses_department_id_fkey FOREIGN KEY (department_id)
REFERENCES departments(id);

ALTER TABLE ONLY addresses
    ADD CONSTRAINT addresses_people_id_fkey FOREIGN KEY (people_id) REFERENCES
people(id);

ALTER TABLE ONLY addresses
    ADD CONSTRAINT addresses_validitystatus_id_fkey FOREIGN KEY
(validitystatus_id) REFERENCES validitystatus(id);

ALTER TABLE ONLY attaches
    ADD CONSTRAINT attaches_attachtype_id_fkey FOREIGN KEY (attachtype_id)
REFERENCES attachtypes(id);

ALTER TABLE ONLY attaches
    ADD CONSTRAINT attaches_people_id_fkey FOREIGN KEY (people_id) REFERENCES
people(id);

```

```
ALTER TABLE ONLY comments
  ADD CONSTRAINT comments_people_id_fkey FOREIGN KEY (people_id) REFERENCES
  people(id);
```

```
ALTER TABLE ONLY comments
  ADD CONSTRAINT comments_post_id_fkey FOREIGN KEY (post_id) REFERENCES
  posts(id);
```

```
ALTER TABLE ONLY comments
  ADD CONSTRAINT comments_validitystatus_id_fkey FOREIGN KEY
  (validitystatus_id) REFERENCES validitystatus(id);
```

```
ALTER TABLE ONLY commentsgroups
  ADD CONSTRAINT commentsgroups_people_id_fkey FOREIGN KEY (people_id)
  REFERENCES people(id);
```

```
ALTER TABLE ONLY commentsgroups
  ADD CONSTRAINT commentsgroups_post_id_fkey FOREIGN KEY (post_id) REFERENCES
  people(id);
```

```
ALTER TABLE ONLY commentsgroups
  ADD CONSTRAINT commentsgroups_validitystatus_id_fkey FOREIGN KEY
  (validitystatus_id) REFERENCES validitystatus(id);
```

```
ALTER TABLE ONLY education
  ADD CONSTRAINT education_titletype_id_fkey FOREIGN KEY (titletype_id)
  REFERENCES titletypes(id);
```

```
ALTER TABLE ONLY experiences
  ADD CONSTRAINT experiences_job_id_fkey FOREIGN KEY (job_id) REFERENCES
  jobs(id);
```

```
ALTER TABLE ONLY experiences
  ADD CONSTRAINT experiences_people_id_fkey FOREIGN KEY (people_id) REFERENCES
  people(id);
```

```
ALTER TABLE ONLY experiences
  ADD CONSTRAINT experiences_sector_id_fkey FOREIGN KEY (sector_id) REFERENCES
  sectortype(id);
```

```
ALTER TABLE ONLY friendships
  ADD CONSTRAINT friendships_friendr_fkey FOREIGN KEY (friendr) REFERENCES
  people(id);
```

```
ALTER TABLE ONLY friendships
  ADD CONSTRAINT friendships_friends_fkey FOREIGN KEY (friends) REFERENCES
  people(id);
```

```
ALTER TABLE ONLY groups
```

```

    ADD CONSTRAINT groups_people_id_fkey FOREIGN KEY (people_id) REFERENCES
people(id);

ALTER TABLE ONLY logs
    ADD CONSTRAINT logs_people_id_fkey FOREIGN KEY (people_id) REFERENCES
people(id);

ALTER TABLE ONLY memberships
    ADD CONSTRAINT memberships_group_id_fkey FOREIGN KEY (group_id) REFERENCES
groups(id);

ALTER TABLE ONLY memberships
    ADD CONSTRAINT memberships_people_id_fkey FOREIGN KEY (people_id) REFERENCES
people(id);

ALTER TABLE ONLY memberships
    ADD CONSTRAINT memberships_statuspetition_id_fkey FOREIGN KEY
(statuspetition_id) REFERENCES statuspetitions(id);

ALTER TABLE ONLY messages
    ADD CONSTRAINT messages_peoplereceiver_fkey FOREIGN KEY (peoplereceiver)
REFERENCES people(id);

ALTER TABLE ONLY messages
    ADD CONSTRAINT messages_peoplesender_fkey FOREIGN KEY (peoplesender)
REFERENCES people(id);

ALTER TABLE ONLY messages
    ADD CONSTRAINT messages_validitystatus_id_fkey FOREIGN KEY
(validitystatus_id) REFERENCES validitystatus(id);

ALTER TABLE ONLY municipio
    ADD CONSTRAINT municipio_department_id_fkey FOREIGN KEY (department_id)
REFERENCES departments(id);

ALTER TABLE ONLY names
    ADD CONSTRAINT names_people_id_fkey FOREIGN KEY (people_id) REFERENCES
people(id);
ALTER TABLE ONLY names
    ADD CONSTRAINT names_validitystatus_id_fkey FOREIGN KEY (validitystatus_id)
REFERENCES validitystatus(id);

ALTER TABLE ONLY news
    ADD CONSTRAINT news_people_id_fkey FOREIGN KEY (people_id) REFERENCES
people(id);

ALTER TABLE ONLY people
    ADD CONSTRAINT people_peopletype_id_fkey FOREIGN KEY (peopletype_id)
REFERENCES peopletypes(id);

ALTER TABLE ONLY peopleeducation
    ADD CONSTRAINT peopleeducation_education_id_fkey FOREIGN KEY (education_id)
REFERENCES education(id);

```

```

ALTER TABLE ONLY peopleeducation
    ADD CONSTRAINT peopleeducation_people_id_fkey FOREIGN KEY (people_id)
    REFERENCES people(id);

ALTER TABLE ONLY petitions
    ADD CONSTRAINT petitions_peoplereceiver_fkey FOREIGN KEY (peoplereceiver)
    REFERENCES people(id);

ALTER TABLE ONLY petitions
    ADD CONSTRAINT petitions_peoplesender_fkey FOREIGN KEY (peoplesender)
    REFERENCES people(id);

ALTER TABLE ONLY phones
    ADD CONSTRAINT phones_people_id_fkey FOREIGN KEY (people_id) REFERENCES
    people(id);

ALTER TABLE ONLY phones
    ADD CONSTRAINT phones_phonetype_id_fkey FOREIGN KEY (phonetype_id) REFERENCES
    phonestypes(id);

ALTER TABLE ONLY postgroups
    ADD CONSTRAINT postgroups_attach_id_fkey FOREIGN KEY (attach_id) REFERENCES
    attaches(id);

ALTER TABLE ONLY postgroups
    ADD CONSTRAINT postgroups_group_id_fkey FOREIGN KEY (group_id) REFERENCES
    groups(id);

ALTER TABLE ONLY postgroups
    ADD CONSTRAINT postgroups_validitystatus_id_fkey FOREIGN KEY
    (validitystatus_id) REFERENCES validitystatus(id);

ALTER TABLE ONLY posts
    ADD CONSTRAINT posts_attach_id_fkey FOREIGN KEY (attach_id) REFERENCES
    attaches(id);
ALTER TABLE ONLY posts
    ADD CONSTRAINT posts_people_id_fkey FOREIGN KEY (people_id) REFERENCES
    people(id);

ALTER TABLE ONLY posts
    ADD CONSTRAINT posts_validitystatus_id_fkey FOREIGN KEY (validitystatus_id)
    REFERENCES validitystatus(id);

REVOKE ALL ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON SCHEMA public FROM postgres;
GRANT ALL ON SCHEMA public TO postgres;
GRANT ALL ON SCHEMA public TO PUBLIC;

```