

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



TESIS:

**“DESARROLLO DE UN SISTEMA QUE AUTOMATICE LOS PROCESOS
DE SOLICITUD Y CONTROL DE SERVICIOS TÉCNICOS EN EL
DEPARTAMENTO DE INFORMÁTICA DE LA DIRECCIÓN REGIONAL
DE SALUD ORIENTAL”**

PRESENTADA POR:

KONNY LIZBETH AMAYA BARRERA
EDWARD VLADIMIR HERNÁNDEZ GARCÍA
JOSÉ ALFREDO JIMÉNEZ BENÍTEZ
CARLOS FERMIN PADILLA FERRUFINO

PARA OPTAR AL TÍTULO DE:

INGENIERO DE SISTEMAS INFORMÁTICOS

DOCENTE ASESOR:

ING. HÉCTOR JAVIER PAIZ RAMOS

CIUDAD UNIVERSITARIA ORIENTAL, DICIEMBRE 2020
SAN MIGUEL, EL SALVADOR, CENTROAMÉRICA

UNIVERSIDAD DE EL SALVADOR
AUTORIDADES

MASTER ROGER ARMANDO ARIAS ALVARADO
RECTOR

DOCTOR RAÚL ERNESTO AZCÚNAGA LÓPEZ
VICERECTOR ACADÉMICO

INGENIERO JUAN ROSA QUINTANILLA
VICERECTOR ADMINISTRATIVO

MASTER FRANCISCO ANTONIO ALARCÓN SANDOVAL
SECRETARIO GENERAL

LICENCIADO RAFAEL HUMBERTO PEÑA MARÍN
FISCAL GENERAL

FACULTAD MULTIDISCIPLINARIA ORIENTAL
AUTORIDADES

LICENCIADO CRISTÓBAL HERNÁN RÍOS BENÍTEZ
DECANO

LICENCIADO OSCAR VILLALOBOS
VICEDECANO

LICENCIADO ISRAEL LÓPEZ MIRANDA
SECRETARIO INTERNO

LICENCIADO JORGE PASTOR FUENTES CABRERA
DIRECTOR GENERAL DE PROCESOS DE GRADUACIÓN

INGENIERA MILAGRO DE MARÍA ROMERO BARDALES
COORDINADORA DE PROCESOS DE GRADUACIÓN

TRIBUNAL EVALUADOR

INGENIERO HÉCTOR JAVIER PAIZ RAMOS

JURADO ASESOR

MASTER JULIO CÉSAR ARIAS GUEVARA

DOCENTE JURADO CALIFICADOR

INGENIERA LIGIA ASTRID HERNÁNDEZ BONILLA

DOCENTE JURADO CALIFICADOR

Agradecimientos

Primeramente, agradezco a Dios por haberme dado la oportunidad de estudiar esta carrera y por sobretodo culminarla, después de tantos altos y bajos a lo largo de mis años de estudio.

Agradezco a mi familia, mis padres Nelson Amaya y Zulma Barrera por brindarme sus consejos y apoyo cuando más los necesitaba. A mis hermanos, que siempre, de una manera u otra han estado brindándome todo su apoyo incondicional en todas las metas que me propongo. A mi mascota, por siempre estar acompañándome en las largas noches de trabajo.

Doy gracias a los docentes, tanto al asesor Ing. Héctor Paiz como a los jurados Ing. Astrid Hernández y Master Julio Arias, y a todos los que he conocido a lo largo de la carrera. De ellos he aprendido lo que significa ser profesional.

Y finalmente, a mis compañeros de tesis y a la vez mis amigos; Carlos Padilla, Edward Hernández y Alfredo Benítez, por todas las aventuras que hemos vivido a lo largo de nuestra carrera. Ha sido todo un honor hacer el proyecto de tesis con ustedes.

Konny Lizbeth Amaya Barrera

Agradecimientos

A Dios: por permitirme culminar este proyecto y cumplir mi sueño de ser profesional.

A mi madre Gloria Betty García: por darme día a día su apoyo incondicional, por estar siempre, por todos esos consejos y enseñanzas que me ha dado hasta este día, con el propósito de que sea una persona de bien.

A toda mi Familia: Porque a pesar de las carencias y dificultades siempre me han brindado su apoyo.

A mis amigos y compañeros: Por brindarme su amistad y apoyo cuando lo necesitaba.

Edward Vladimir Hernández García

Agradecimientos

A Dios: por permitirme culminar este proyecto y cumplir mi sueño de ser profesional.

A mis Padres Gladis Benítez y José Jiménez: por darme día a día su apoyo incondicional, por estar siempre, por todos esos consejos y enseñanzas que me ha dado hasta este día, con el propósito de que sea una persona de bien.

A mi hermana, mi pareja, mi mascota Misty y mis amigos por estar ahí cuando lo necesite.

José Alfredo Jiménez Benítez

Agradecimientos

A Dios: por permitirme culminar este proyecto y cumplir mi sueño de ser profesional.

A mi mamá Iris Yolanda Ferrufino: Por darme día a día su apoyo incondicional, por estar siempre, por todos esos consejos y enseñanzas que me ha dado hasta este día, con el propósito de que sea una persona de bien.

A mi Hermana Ana Padilla: por brindarme consejo y apoyo a lo largo de toda la carrera.

A mi Abuela Vitelia Bonilla: Por estar presente en cada momento de mi vida y ser inspiración para mí.

A mis compañeros de tesis: Por su amistad, esfuerzo y dedicación durante el transcurso de nuestra carrera y durante el desarrollo de este proyecto.

A mis amigos y compañeros: Por brindarme su amistad y apoyo cuando lo necesitaba.

Carlos Fermin Padilla Ferrufino

Desarrollo de un sistema que automatice los procesos de solicitud y control de servicios técnicos en el Departamento de Informática de la Dirección Regional de Salud Oriental

Konny Lizbeth Amaya Barrera

Edward Vladimir Hernández García

José Alfredo Jiménez Benítez

Carlos Fermin Padilla Ferrufino

Universidad de El Salvador

Notas de Autor

Konny Lizbeth Amaya Barrera

Edward Vladimir Hernández García

José Alfredo Jiménez Benítez

Carlos Fermin Padilla Ferrufino

Tesis para optar al título de Ingeniero en Sistemas Informáticos

Universidad de El Salvador

Facultad Multidisciplinaria Oriental

Resumen

El objetivo principal de este trabajo de tesis que se detalla en este documento pretende modernizar los servicios que actualmente ofrece el Departamento de Informática de la Dirección Regional de Salud Oriental, específicamente la gestión de solicitudes técnicas mediante un sistema web y aplicación móvil, de fácil uso para los trabajadores del Ministerio de Salud y para ellos mismos. Este sistema permite agilizar los procesos de servicio del área de informática. De la misma manera, se podrá encontrar el contexto de desarrollo, las tecnologías implementadas y los estándares del Ministerio de Salud, los cuales son Symfony v2.8 como framework de desarrollo PHP, PostgreSQL 9.6 como base de datos y Flutter como propuesta de framework para desarrollo en aplicaciones móviles. Así como también el plan de implementación, características de los equipos a utilizar, manuales de desarrollo, de usuarios y de instalación para las aplicaciones web y móvil. De esta manera se espera que el Ministerio de Salud modernice sus instalaciones y que en un futuro próximo puedan extender el alcance del sistema de control de servicios técnicos a todo El Salvador.

Keywords: Software, informática, Ministerio de Salud, Dirección Regional de Salud Oriental, PostgreSQL, Symfony, Flutter, peticiones de servicio.

Abstract

The main objective of this thesis work that is detailed in this document aims to modernize the services currently offered by the Department of Informatics of the Eastern Regional Health Directorate, specifically the management of technical requests through a web system and mobile application, easy to use for the workers of the Ministry of Health and for themselves. This system makes it possible to streamline the service processes of the IT area. In the same way, you can find the development context, the implemented technologies and the standards of the Ministry of Health, which are Symfony v2.8 as a PHP development framework, PostgreSQL 9.6 as a database and Flutter as a proposed framework for development in mobile applications. As well as the implementation plan, characteristics of the equipment to be used, development, user and installation manuals for web and mobile applications. In this way, it is expected that the Ministry of Health will modernize its facilities and that in the near future they will be able to extend the scope of the technical services control system to all of El Salvador.

Keywords: Software, IT, Ministry of Health, Eastern Regional Health Directorate, PostgreSQL, Symfony, Flutter, requests services.

Índice

Notas de Autor	9
Resumen	10
Abstract	11
Introducción	23
Planteamiento del problema	24
Definición del tema	24
Descripción de la institución	24
Dirección Regional de Salud Oriental	24
Dirección de Tecnologías de Información y Comunicaciones	25
Departamento de Informática	26
Situación problemática	27
Gestión de información	27
Optimización de procesos	28
Enunciado del problema	29
Justificación	29
Objetivos	31
Objetivo general	31
Objetivos específicos	31
Alcances y limitaciones	31
Alcances	31
Limitaciones	33
Marco de referencia	34
Marco histórico	34

SISTEMA DE SOLICITUDES TÉCNICAS	13
Dirección Regional de Salud Oriental	34
Dirección de Tecnologías de Información y Comunicaciones	34
Departamento de Informática	36
Marco teórico	37
Internet	37
La web (WWW)	39
Arquitectura Cliente - Servidor	39
Aplicaciones web	40
Tecnologías móviles	41
Metodologías para el desarrollo de software	42
Metodologías para pruebas del sistema	48
Metodologías para documentación	49
Herramientas para el desarrollo de software	49
Marco normativo	74
Ley de propiedad intelectual	74
Manual de procedimientos del Departamento de Informática	76
Metodología de investigación	78
Tipo de investigación	78
Investigación descriptiva	78
Investigación tecnológica	78
Universo	79
Tipo de muestreo	79
Muestreo no probabilístico	79
Método por conveniencia	79
Muestra	80
Técnicas e instrumentos para recolección de información	80
Procedimientos para la validación de instrumentos	81

SISTEMA DE SOLICITUDES TÉCNICAS	14
Procedimientos para la recolección de datos	81
Procedimientos para el tratamiento de los datos	82
Procedimientos para presentar e interpretar los datos	82
Análisis e interpretación de resultados	82
Análisis de entrevista	82
Diagnóstico	88
Hallazgos	88
Hallazgos de observación	88
Hallazgos en entrevistas	89
Problemas generados por el sistema actual	89
Procesos actuales	90
Proceso actual de una petición	90
Proceso de respuesta a una petición	90
Análisis FODA	91
Enfoque del sistema propuesto	95
Desarrollo del software	96
Diagrama causa-efecto	96
Requerimientos de información	98
Aplicación web	98
Aplicación móvil	100
Estudio de factibilidades	101
Factibilidad técnica	101
Factibilidad operativa	106
Factibilidad económica	107
Estándares de desarrollo	110
Tipos de usuarios	110

Roles	111
Perfiles para el desarrollo del sistema	113
Arquitectura de desarrollo: MVC	114
API	115
Seguridad	116
Diseño del sistema informático	120
Diagramas de casos de uso del sistema web	120
Diagramas de casos de uso de la aplicación móvil	130
Diagramas de secuencia de la aplicación web	137
Diagramas de secuencia de la aplicación móvil	145
Diagrama de clases	150
Estandarización de base de datos	151
Diccionario de datos	154
Diagrama entidad-relación	154
Diagrama lógico de base de datos	156
Diagrama conceptual de base de datos	158
Descripción de base de datos	160
Diseño de interfaces	182
Sistema web	182
Aplicación móvil	186
Programación del sistema informático	191
Symfony v2.8	191
Flutter	196
Pruebas	199
Tipos de pruebas	199
Resultados de las pruebas	200

SISTEMA DE SOLICITUDES TÉCNICAS	16
Documentación y plan de implementación	212
Documentación	212
Manuales	212
Plan de implementación	213
Arquitectura	213
Actividades a realizar	216
Plan de capacitación al personal	216
Recursos a utilizar	217
Módulos a capacitar por usuario	218
Conclusiones	222
Recomendaciones	223
Glosario	224
Anexos	230
Carta de compromiso de la Dirección Regional de Salud Oriental	230
Cuestionario de entrevista para los profesionales	231
Formato de Hoja de servicio	235
Formato de Hoja de descarte	236
Carta de aceptación del sistema web y móvil	238

Índice de figuras

1.	Ubicación de la Dirección Regional de Salud Oriental	24
2.	Organigrama de la Dirección Regional de Salud Oriental	25
3.	Organigrama de la DTIC	35
4.	Red internet y servicios básicos.	38
5.	Modelo Cliente Servidor	40
6.	Modelo en cascada	42
7.	Diagrama de metodología SCRUM.	47
8.	Ejemplo de diagrama de secuencia.	53
9.	Componentes importantes en un sistema PostgreSQL	55
10.	Vista de dashboard en pgAdmin 4.	59
11.	Logo de XAMPP.	60
12.	Página de inicio de Mozilla Firefox.	64
13.	Alcance de Flutter.	70
14.	Diagrama causa efecto del Departamento de Informática.	96
15.	Funcionamiento del patrón MVC.	114
16.	Proceso de autenticación <i>anonymous</i> en Symfony.	117
17.	Proceso de autenticación denegada en Symfony.	117
18.	Proceso de autenticación de usuario en Symfony.	118
19.	Caso de uso del sistema web.	120
20.	Caso de uso de la aplicación web.	130
21.	Diagrama de secuencia: inicio de sesión.	137
22.	Diagrama de secuencia: agregar equipos e insumos.	138
23.	Diagrama de secuencia: agregar petición.	139
24.	Diagrama de secuencia: agregar usuario.	140
25.	Diagrama de secuencia: cambiar estado de petición.	141
26.	Diagrama de secuencia: consultar estado de petición.	142

27.	Diagrama de secuencia: generar reporte.	143
28.	Diagrama de secuencia: gestión de inventarios.	144
29.	Diagrama de secuencia: iniciar sesión desde la aplicación móvil.	145
30.	Diagrama de secuencia: agregar petición.	146
31.	Diagrama de secuencia: agregar usuario.	147
32.	Diagrama de secuencia: gestión de peticiones.	148
33.	Diagrama de secuencia: consultar estado de la petición.	149
34.	Diagrama de clases de la aplicación web (Parte 1).	150
35.	Diagrama de clases de la aplicación web (Parte 2).	151
36.	Diagrama entidad-relación de la base de datos db_sistec (Parte 1).	154
37.	Diagrama entidad-relación de la base de datos db_sistec (Parte 2).	155
38.	Diagrama lógico de la base de datos db_sistec (Parte 1).	156
39.	Diagrama lógico de la base de datos db_sistec (Parte 2).	157
40.	Diagrama conceptual de la base de datos db_sistec (Parte 1).	158
41.	Diagrama conceptual de la base de datos db_sistec (Parte 2).	159
42.	Vista de login de la aplicación web.	182
43.	Vista de logout de la aplicación web.	183
44.	Vista de usuarios de la aplicación web.	183
45.	Vista de establecimientos de la aplicación web.	184
46.	Vista de agregar movimiento de la aplicación web.	184
47.	Vista de agregar servicio de la aplicación web.	185
48.	Vista de agregar tipo de insumo de la aplicación web.	185
49.	Vista de inicio de la aplicación móvil.	186
50.	Vista de login de la aplicación móvil.	187
51.	Vista de lista de solicitud de la aplicación móvil.	188
52.	Vista de tipo de solicitud de la aplicación móvil.	189
53.	Vista de agregar solicitud de la aplicación móvil.	190

54.	Carpeta de proyecto Symfony v2.8.	192
55.	Carpeta <i>app</i> del proyecto Symfony v2.8.	193
56.	Carpeta <i>config</i> del proyecto Symfony v2.8.	193
57.	Archivo <i>parameters</i> del proyecto Symfony v2.8.	194
58.	Carpeta <i>resources</i> del proyecto Symfony v2.8.	194
59.	Carpeta <i>base</i> del proyecto Symfony v2.8.	194
60.	Carpeta <i>src</i> del proyecto Symfony v2.8.	195
61.	Carpeta <i>web</i> del proyecto Symfony v2.8.	196
62.	Carpeta del proyecto Flutter.	196
63.	Carpeta <i>assets</i> del proyecto Flutter.	197
64.	Carpeta <i>lib</i> del proyecto Flutter.	197
65.	Carpeta <i>src</i> del proyecto Flutter.	197
66.	Archivo <i>pubspec.yaml</i> del proyecto Flutter.	198
67.	Vista login con datos correctos.	200
68.	Vista login con datos incorrectos.	201
69.	Vista agregar solicitud con datos correctos.	202
70.	Vista agregar solicitud con datos incorrectos.	203
71.	Hoja de servicio con datos correctos.	204
72.	Hoja de servicio con datos incorrectos.	204
73.	Ingreso de visita con datos correctos.	205
74.	Vista de la visita en el calendario.	205
75.	Formulario de Hoja de Descarte.	206
76.	Archivo PDF de la Hoja de Descarte.	206
77.	Agregar solicitud desde aplicación móvil.	207
78.	Ver solicitud en la aplicación web.	208
79.	Lista de solicitud con 200 registros.	209
80.	Lista de solicitud en la aplicación móvil con 200 registros.	209

81.	Lista de solicitud con 2000 registros.	210
82.	Lista de solicitud en la aplicación móvil con 2000 registros.	211
83.	Arquitectura de implementación del sistema.	215

Índice de cuadros

1.	Símbolos para el diagrama de casos de uso.	51
2.	Criterios de elección de la tecnología móvil.	69
3.	Criterios de elección del entorno de programación.	72
4.	Análisis FODA: Fortalezas y debilidades del Departamento de Informática. .	92
5.	Análisis FODA: Oportunidades y amenazas del Departamento de Informática.	94
6.	Análisis de situación-solución.	98
7.	Especificaciones del equipo del MINSAL.	102
8.	Especificaciones del equipo de desarrollo.	103
9.	Especificaciones de software.	104
10.	Software del equipo de desarrollo.	105
11.	Factibilidad económica del hardware y software.	109
12.	Recursos consumibles.	109
13.	Otros gastos.	109
14.	Costo total del proyecto.	110
15.	Caso de uso: Inicio de sesión.	121
16.	Caso de uso: Consultar estado de petición.	122
17.	Caso de uso: Gestión de inventarios.	123
18.	Caso de uso: Cambiar estado de servicio	124
19.	Caso de uso: Registrar petición de servicio.	126
20.	Caso de uso: Agregar equipo o insumo.	127
21.	Caso de uso: Generar reportes.	128
22.	Caso de uso: Registro de usuario.	129
23.	Caso de uso: Iniciar sesión en la aplicación móvil.	132
24.	Caso de uso: Gestionar peticiones en la aplicación móvil.	133
25.	Caso de uso: Agregar petición en la aplicación móvil.	134
26.	Caso de uso: Consultar estado de petición en la aplicación móvil.	135

27.	Caso de uso: Registro de usuario en la aplicación móvil.	137
28.	Descripción de tablas de la base de datos db_sistec	163
29.	Descripción de tabla tbl_bitacora.	163
30.	Descripción de tabla tbl_departamento_pais.	164
31.	Descripción de tabla tbl_descarte.	165
32.	Descripción de tabla tbl_detalle_hoja.	166
33.	Descripción de tabla tbl_equipos.	167
34.	Descripción de tabla tbl_establecimiento.	168
35.	Descripción de tabla tbl_hoja_servicio.	170
36.	Descripción de tabla tbl_insumos.	171
37.	Descripción de tabla tbl_mov_insumos.	173
38.	Descripción de tabla tbl_prioridad.	173
39.	Descripción de tabla tbl_solicitud.	174
40.	Descripción de tabla tbl_tipo_equipos.	175
41.	Descripción de tabla tbl_tipo_insumo.	176
42.	Descripción de tabla tbl_tipo_mov.	178
43.	Descripción de tabla tbl_tipo_servicio.	179
44.	Descripción de tabla tbl_usuarios.	181
45.	Descripción de tabla tbl_visitas.	182
46.	Requerimientos mínimos en hardware para implementación del sistema.	214
47.	Software necesario para la implementación del sistema.	215
48.	Recursos a utilizar para la capacitación del personal.	217
49.	Personal capacitador.	218
50.	Personal a capacitar.	218
51.	Módulos a capacitar al administrador.	219
52.	Módulos a capacitar al técnico.	221
53.	Módulos a capacitar al usuario final.	221

Introducción

Actualmente la forma en la que se obtiene información específica ayuda a aligerar los procesos y tareas que necesitamos realizar, acortando tiempos de espera y optimizando el desempeño de las personas. En este sentido, El Salvador se encuentra en un proceso de transición, poco a poco las diferentes carteras del estado implementan sistemas informáticos que ayudan a mejorar el trabajo del personal. Ministerios como el de Salud cuentan con área específica de desarrollo y tecnología; sin embargo, la demanda es mucha y este tipo de herramientas no son tomadas en cuenta para desarrollar e implementar por el esfuerzo que implican hacerlas.

En el Departamento de Informática de la Dirección Regional de Salud Oriental los procesos de solicitud y control de mantenimiento de equipos se realizan manualmente, así mismo el registro y calendarización de visitas técnicas dentro de la misma institución o a las diferentes unidades de salud que así lo requieran.

En la presente investigación se muestran las herramientas o elementos necesarios para llevar a cabo dicha investigación; proponiendo una alternativa a la necesidad de agilizar los procesos de gestión, solicitud y control de peticiones de servicio técnico para el Departamento de Informática de la Dirección Regional de Salud Oriental. De esta manera se pretende contribuir a la modernización de este departamento, facilitando la comunicación de las diferentes áreas que necesiten sus servicios.

Planteamiento del problema

Definición del tema

Desarrollo de un sistema que automatice los procesos de solicitud y control de servicios técnicos en el Departamento de Informática de la Dirección Regional de Salud Oriental.

Descripción de la institución

Dirección Regional de Salud Oriental

Fundada el 10 de agosto de 2005, La Dirección Regional de Salud Oriental es la institución encargada de los establecimientos de salud, como hospitales, unidades de salud, centros de atención de emergencias, etc. Se encuentra ubicada en la Carretera Panamericana salida a San Salvador, Kilometro 132 1/2 San Miguel (Instituto de Acceso a la Información Pública, 2020), como se muestra en la Figura 1.

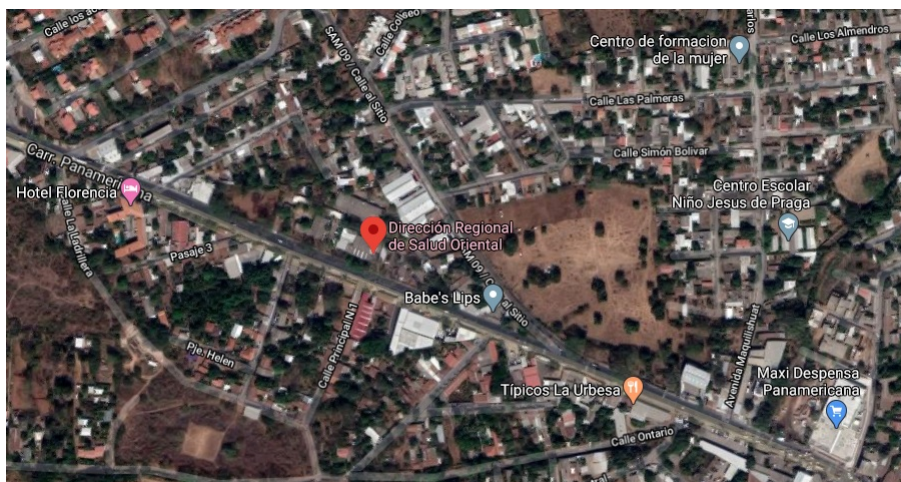


Figura 1
Ubicación de la Dirección Regional de Salud Oriental

Según el Decreto Presidencial N° 77 (2006), la institución se encarga de tareas de administración, vela por cumplimientos de leyes, reglamentos, norma sanitarias, y supervisa

la ejecución de las unidades a su cargo (p. 4); siendo intermediaria entre el Ministerio de Salud y unidades de salud orientales, ilustrándose en el organigrama de la Figura 2.

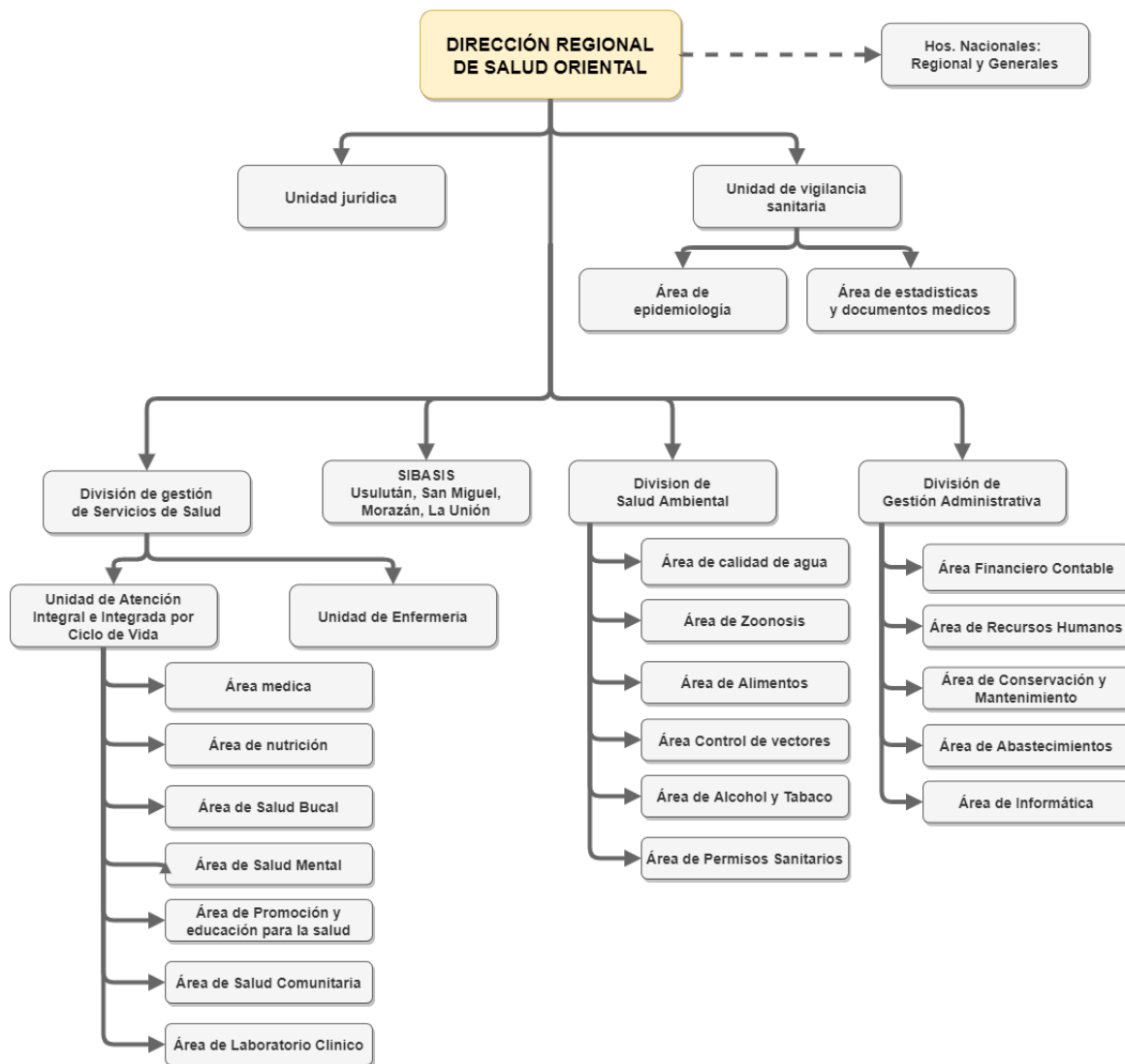


Figura 2
Organigrama de la Dirección Regional de Salud Oriental

Dirección de Tecnologías de Información y Comunicaciones

La Dirección de Tecnologías de Información y Comunicaciones, tiene la responsabilidad de la administración de equipamiento informático y de comunicaciones de datos, así como planificación y desarrollo de sistemas de información (MINSAL, 2010). Es la que está directamente ligada a la administración informática de todo el Ministerio de

Salud, así como máxima autoridad en lo que respecta al Departamento de Informática de la Dirección Regional de Salud Oriental.

Departamento de Informática

El Departamento de Informática de la Dirección Regional de Salud Oriental es una extensión supervisora de la DTIC (MINSAL, 2011b), encargada de realizar actividades de administración de tecnologías de información en la zona oriental del país.

Dicha área es la encargada principalmente de administrar y dar soporte técnico a la infraestructura de equipos de cómputo y comunicaciones que sustenta el Ministerio de Salud, además de realizar actualizaciones en sistemas que lo requieran, actualizaciones de aplicaciones móviles, soporte al sistema operativo, administrar la infraestructura de red, brindar capacitaciones de nuevas tecnologías a implementar y gestionar la red interna de la institución a nivel oriental.

En el Acuerdo Ministerial N° 1208 (2015), la entonces llamada Unidad de Información, se describe como responsable del funcionamiento de los sistemas de información, cumpliendo así su deber de brindar información oportuna, fiable y útil para la toma de decisiones.

Algunas de las competencias y facultades según el MINSAL (2011b) son:

- Implementar y mantener la infraestructura de telecomunicaciones del MINSAL.
- Proveer la seguridad necesaria a los equipos y datos disponibles en el SUIS¹.
- Llevar a cabo tareas de soporte técnico, asesoría y capacitación al personal que trabaja con tecnologías de información y comunicaciones.
- Mejorar la pertinencia, eficacia y eficiencia de las actividades realizadas sobre equipamiento de cómputo, periféricos y aplicaciones informáticas.

¹ Sistema Único de Información en Salud

Según el MINSAL (2014), este departamento brinda sus servicios a 231 Unidades Comunitarias de Salud Familiar, o por sus siglas UCSF, ubicados en los departamentos de Usulután, Morazán, San Miguel y La Unión, además de las diferentes áreas de la misma Dirección Regional de Salud Oriental.

Situación problemática

Actualmente el Ministerio de Salud se encuentra en el proceso de modernización y adaptación de las tecnologías de la información y comunicación (TIC). Sin embargo, en la Dirección Regional de Salud Oriental, específicamente en su Departamento de Informática, se están dando situaciones problemáticas que a grandes rasgos se dividen en dos:

- Gestión de información
- Optimización de procesos

Gestión de información

Esta situación se da con frecuencia y puede llegar a afectar el desempeño completo del Departamento. Una de las principales tareas es la administración de las solicitudes de mantenimiento informático. A priori se puede intuir que es una actividad sencilla, pero al poseer una demanda de más de 256 unidades de salud², adicionales a todos los departamentos dentro de la DRSO³ se genera una sobre demanda de peticiones que desencadenan problemas de planificación de visitas, resolución de solicitudes, organización del personal y duplicidad de información.

Gran parte de las peticiones realizadas al Departamento de Informática son desde unidades fuera de la Regional de Salud Oriental, es decir que los técnicos tienen que trasladarse hasta dicho lugar para solventar el mantenimiento. Este tipo de visita se realizan de forma diaria, pero su organización no es la adecuada, dado que hay ocasiones

² Fuente: Ing. Juan Soto, jefe del Departamento de Informática.

³ Dirección Regional de Salud Oriental.

puntuales en la cual la demanda es tanta que se sobreponen fechas o también físicamente no se pueden movilizar por la magnitud del trabajo a realizar.

Otra situación problemática que se genera por la falta de una correcta gestión de la información es que los responsables o solicitantes de un tipo de mantenimiento no se encuentre en el área al momento de la visita lo que al no poder notificar de la llegada de los técnicos y estos no poseen las autorizaciones respectivas, dicho mantenimiento se queda en pausa y se vuelve a programar con mínimo de una semana.

Optimización de procesos

En síntesis, una petición de servicio informático es la solicitud que se le realiza al Departamento de Informática para solventar un problema relacionado con equipo de computo o soporte técnico. Esto implica, que el personal de las diferentes áreas que ellos atienden realicen una visita solamente para solicitar este servicio. Este mismo proceso se puede hacer también por medio de una llamada telefónica o un correo electrónico.

El problema de estos medios por los que se hacen las peticiones se lleve a cabo de una forma lenta debido a la gran demanda, lo que hace que las tareas se pospongan demasiado o incluso que algunos mantenimientos no sean realizados. A la larga, genera una acumulación de actividades, mala administración del tiempo y desorganización en las tareas. Lo que en simultaneo hace que actividades de relevancia que realizan áreas determinadas se vean afectadas por no tener los equipos a disposición.

Esta forma de trabajar se debe a que no existe un medio por el cual realizar las peticiones de forma más ordenada, que lo presente por orden de prioridad y mostrar el estado de las solicitudes al momento. Un sistema que realice tareas como calendarización de visitas, llenado de formularios de mantenimiento, reportes de tareas prioritarias realizadas durante un determinado tiempo, se vuelven necesarias para mejorar la calidad de trabajo y tiempo de respuesta del personal que labora en dicha área.

La razón principal por la cual aún no existe es debido a las prioridades que tiene la

DTIC⁴, que según el MINSAL (2011b) es el desarrollo del Sistema Único de Información en Salud, teniendo poco que ver con el problema que se presenta en el Área de Informática de la Dirección Regional de Salud Oriental.

Esta problemática se puede solucionar con la contratación de más personal en el área de la DTIC, pero para ello se requiere de recursos económicos, los cuales no se disponen. Otra alternativa es desarrollar un sistema y aplicación para el personal de dicha área y así aportar a la realización de sus tareas más eficientemente. Así surge la pregunta, ¿cómo desarrollar dicha herramienta que le permita al personal del área de Informática agilizar sus procesos administrativos y tareas diarias?

Enunciado del problema

¿De que forma el desarrollo de un sistema que automatice los procesos de solicitud y control de servicios técnicos contribuirá a mejorar la administración del Departamento de Informática de la Dirección Regional de Salud Oriental?

Justificación

La tecnología avanza exponencialmente, y con ella, la manera de administrar y gestionar todo lo relacionado a las actividades de una organización. Las tecnologías de la información han ido evolucionando a tal punto que es imprescindible para realizar una tarea eficientemente. Esta nos permite registrar, editar, y eliminar datos, para así obtener información clara, concisa y detallada, que ayudan a la toma de decisiones más acertadas. En este contexto, muchas de las organizaciones desean ser partícipes de esta forma de gestión, pero son conscientes que es un estudio muy amplio además de una gran inversión que conlleva desarrollar una herramienta que cumpla con sus necesidades.

Según el Acuerdo Ministerial N° 1208 (2015), en la Dirección Regional de Salud Oriental trabajan 146 personas, seis de las cuales pertenecen a la Unidad de Informática

⁴ Dirección de Tecnologías de Información y Comunicaciones

para su año de publicación. Actualmente, laboran solamente cuatro especialistas en el área, que según dicho acuerdo recae la responsabilidad del funcionamiento de los sistemas de información para que brinden información oportuna, fiable y útil para la toma de decisiones en general.

El Departamento de Informática brinda asistencia técnica a 264 UCSF⁵, incluyendo al Almacén Regional, que es de donde abastecen de medicina a todas las UCSF; y a la misma Dirección Regional. Entre ellas, 92 unidades de salud pertenecen al departamento de Usulután, 62 en San Miguel, 58 a Morazán y 48 a La Unión; estando ubicados en lugares estratégicos, como cantones, municipios o lugares de difícil acceso. Estas unidades poseen un aproximado de seis personas laborando cada una; atendiendo a su vez a población aledaña de cada establecimiento.

La mayoría de establecimientos poseen de dos a más equipos de computo. Esto significa, que los técnicos del Departamento de Informática están encargados del buen funcionamiento de 528 máquinas; además de los equipos del Almacén Regional y los de la misma Dirección Oriental. Teniendo en cuenta que muchos de estas unidades, los equipos de cómputo son imprescindibles para la realización de sus tareas diarias⁶.

Gracias a la gran demanda que posee el Departamento de Informática, el personal necesita una herramienta que posea características que les ayude a gestionar toda la documentación propiamente de la Unidad. Es decir, que la solución debe ayudarles a garantizar el funcionamiento de todos los equipos que las unidades posean, dándole resolución a sus solicitudes en el menor tiempo posible y a su vez, mejorando la calidad de atención del personal, optimizando la búsqueda de información y realizando de forma automática los informes oportunos para una buena toma de decisiones.

Con el desarrollo de un sistema para automatizar procesos de solicitud y control de servicios técnicos para el Departamento de Informática, serían beneficiados todos los

⁵ Unidad Comunitaria de Salud Familiar.

⁶ Fuente: Ing. Juan Soto, jefe del Departamento de Informática.

departamentos que posean equipo informático y que se encuentran dentro de la Dirección Regional de Salud Oriental, así como también las unidades de salud que también reciban soporte por parte de ellos, incentivando así al personal del MINSAL a actualizar sus procesos manuales y gradualmente cambiarlos a un entorno digital.

Objetivos

Objetivo general

Desarrollar un sistema para automatizar procesos de solicitud y control de servicios técnicos que brinda el Departamento de Informática de la Dirección Regional de Salud Oriental, para mejorar la atención que se proporciona a los usuarios.

Objetivos específicos

- Analizar las necesidades de información que se debe incluir para dar solución a la problemática.
- Diseñar el funcionamiento del sistema incluyendo estándares que faciliten la experiencia del usuario.
- Programar el sistema web y aplicación móvil que facilite el trabajo y optimice el tiempo del personal del Departamento de Informática.
- Trazar un plan de implementación dirigido a la Dirección de Tecnologías de Información y Comunicaciones del Ministerio de Salud.

Alcances y limitaciones

Alcances

- Este proyecto tomará en cuenta el análisis, diseño y programación que constará de un sistema web y una aplicación móvil para el apoyo de la gestión de tareas del Departamento de Informática de la Dirección Regional de Salud Oriental.

- El sistema web está orientado principalmente a la administración del Departamento, y teniendo como extensión para sus usuarios una aplicación móvil por la que se podrá comunicar directamente al Departamento de Informática de dicha institución.
- El área espacial que abarcará la investigación son los departamentos de la Dirección Regional de Salud Oriental. La aplicación posteriormente se podrá utilizar en todo el oriente del país y en otras zonas si así lo desea la institución responsable.
- Los perfiles a desempeñar en el desarrollo del sistema son: diseñador, analista de sistemas, programador y tester.
- Se incluirá un manual de usuario para el correcto uso del sistema, una guía de instalación para los servidores y explicación necesaria para su respectivo uso.
- El proyecto se realizará bajo la metodología de investigación descriptiva, siendo el objetivo analizar la situación actual de la institución y basado en los resultados diseñar la solución.
- El proyecto iniciará el 3 de febrero de 2020 y finalizará el 9 de noviembre del mismo año.
- Aplicación móvil tendrá un acceso a los soportes técnicos realizados y estados de los mismos para los usuarios.
- Las principales funciones que deberá realizar el sistema son:
 - Agilizar la gestión de peticiones de servicio informático.
 - Agendar rutas de visita técnicas a las unidades que lo han solicitado según su prioridad.
 - Estado actual de los mantenimientos de equipo de cómputo.
 - Optimizar el tiempo de llenado de hojas de trabajo.

- Administración de servicios o mantenimientos informáticos según su estado y prioridad.
- Historial detallado del estado del servicio informático.

Limitaciones

- Recursos económicos limitados de parte del Ministerio de Salud para la implementación del sistema.
- El Ministerio de Salud tiene acceso restringido a su Base de datos interna, por lo que el proyecto se realizará en una base de datos independiente.
- Efectos de la contingencia y riesgos que se presentan ante la pandemia mundial COVID-19 en el año en que se desarrollará el proyecto.

Marco de referencia

Marco histórico

Dirección Regional de Salud Oriental

La Dirección Regional de Salud Oriental fue fundada el 10 de agosto de 2005 según el Reglamento de la ley del Sistema Básico de Salud Integral (Decreto Presidencial N° 77, 2006); siendo publicada en el diario oficial No. 161, Tomo No. 368 del 1 de septiembre de ese mismo año, empezando así su servicio hacia toda la zona oriental. Comenzó a funcionar encargándose de la administración de 60 establecimientos, en los cuales podemos mencionar hospitales, unidades de salud, centros de atención de emergencias, casas de salud, clínicas y centros rurales de salud y nutrición.

Desde entonces, la importancia de esta institución radica en dicha zona al haber aumentado la cantidad de establecimientos, logrando mejorar la calidad de atención a los pacientes, ampliar los servicios médicos, abastecer de medicina a las diferentes farmacias y llevando centros de atención en salud más cerca de la población.

Dirección de Tecnologías de Información y Comunicaciones

En el año 2010, específicamente el 16 de junio, la entonces ministra de salud Dra. María Isabel Rodríguez, emite un decreto para para creación de la Dirección de Tecnologías de la Información y Comunicaciones, que se desempeña hasta la actualidad como máximo responsable en materias de tecnología de información y comunicaciones, en el interior del Ministerio de Salud y Asistencia Pública Social. Delegando así a esta institución la responsabilidad de la administración de equipamiento informático y de comunicaciones de datos, planificación y desarrollo de sistemas de información (MINSAL, 2010).

A continuación, se muestra en la Figura 3 el diagrama organizacional de la institución, actualizado a septiembre de 2011.

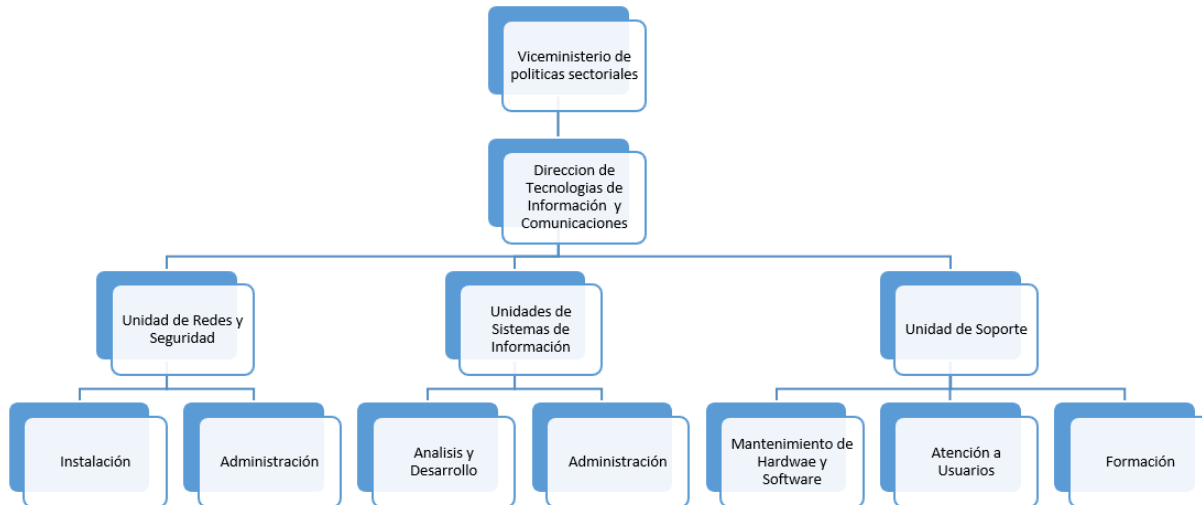


Figura 3
Organigrama de la DTIC

La DTIC, además de la relación jerárquica anteriormente descrita tendrá dos órganos para la toma de decisiones y coordinación interna y externa (MINSAL, 2011b):

- *Dirección:* formada por la persona que ostente el cargo de Director/a. Se elegirá una de las jefaturas para asumir las funciones de la dirección en caso de ausencia de el/la director/a.
- *Comité coordinador:* formada por la Dirección más las jefaturas de las tres unidades funcionales. Celebrarán reuniones de forma ordinaria para la coordinación de las funciones propias de cada Unidad y los aspectos de interdependencia.

Misión. "Somos la máxima autoridad en materia de Tecnologías de Información y Comunicaciones (TICs) del Ministerio de Salud que, de forma responsable y comprometida, impulsa y sostiene el desarrollo de la infraestructura tecnológica del Sistema Único de Información en Salud, a fin de colaborar en el ofrecimiento de una mejor atención a las personas usuarias del Sistema Nacional de Salud así como coadyuvar en la toma de decisiones, el diseño de políticas, la planificación y ejecución de estrategias y la atención a las necesidades administrativas institucionales de forma ágil y oportuna".

Visión. "Ser la instancia del Ministerio de Salud que garantice ante el Sistema Nacional de Salud y la población en general, la plataforma de Tecnologías de Información y Comunicaciones idónea para la construcción y sostenibilidad de un sistema de Información en Salud eficaz, eficiente, transparente, democratizado, seguro y sostenible, contando para ello con la infraestructura, tecnologías y personal más apropiado".

Departamento de Informática

El Departamento de Informática es una extensión supervisora de la DTIC en la Regional de Salud, por lo tanto esta bajo la jefatura de ambas entidades. Fue fundada en conjunto con la Regional de Salud; y según el Acuerdo Ministerial N° 1208 (2015), se definen los objetivos y funciones detallados a continuación.

Objetivos. Los objetivos propios del Departamento de informática son:

General:

Coordinar el mantenimiento preventivo y correctivo de los equipos informáticos asegurando la estabilidad, eficiencia y seguridad de la infraestructura de las redes y equipos de cómputo, a fin de permitir el funcionamiento adecuado de los diferentes sistemas de información.

Específicos:

- Brindar soporte técnico a la infraestructura de tecnologías de información y comunicaciones, inherente al equipamiento físico, sistemas operativos, sistemas gestores de bases de datos y otros.
- Garantizar el adecuado respaldo y seguridad física de los datos contenidos en los sistemas de información institucional.
- Configurar y administrar la infraestructura de red y equipo informático de interconexión necesario para el correcto funcionamiento de los sistemas informáticos y las relaciones con instituciones públicas o privadas.

Funciones. Las funciones propias del Departamento de Informática son:

General:

Administrar y brindar el soporte técnico a los sistemas de comunicación informática en las dependencias de la Región de Salud, SIBASI y UCSF.

Específicas:

- Elaborar el plan operativo anual, monitoreo a los equipos informáticos distribuidos en los diferentes establecimientos de salud de responsabilidad de la Región de Salud.
- Elaborar y ejecutar plan de mantenimiento preventivo y correctivo de los equipos informáticos.
- Realizar el diagnóstico de los equipos informáticos distribuidos en los establecimientos que conforman la Región de Salud.
- Elaborar informes mensuales de supervisiones y solicitudes atendidas de mantenimiento correctivo de los equipos informáticos.
- Coordinar e instalar las redes de comunicación interna y externa, nuevos equipos de cómputo y periféricos en los diferentes establecimientos de salud.
- Brindar capacitación al personal de los establecimientos de salud de los sistemas operativos y ofimática.
- Evaluación y descarga del inventario de equipo de cómputo y periféricos dañados y obsoletos.

Marco teórico

Internet

Las personas y los sistemas de información necesitan comunicarse con recursos (información) que se encuentran alojados en redes externas mediante la herramienta llamada internet.

La internet es la red de computadoras más grande del planeta, cuenta con la base de datos más completa y actualizada, y su principal función es que permite a usuarios de cualquier parte del mundo se conecten a la red para compartir información y recursos (Michelena, 2005).

La internet fue creada en 1969 bajo el nombre de ARPnet que significa (Advanced Research Projects Agency), en el cual se necesitaría una manera de organizar todos estos recursos.

La red global mundial, por sus siglas en inglés World Wide Web [WWW], fue creada en 1990 en el Centro Europeo de Investigaciones Nucleares (CERN) por Tim Berners y Robert Caillau, quienes ocuparon este nombre para un sistema de almacenamiento y recuperación de datos con hipervínculos. El cual, era interpretado por el programa que ahora se conoce como navegador. El esquema general se muestra en la Figura 4.

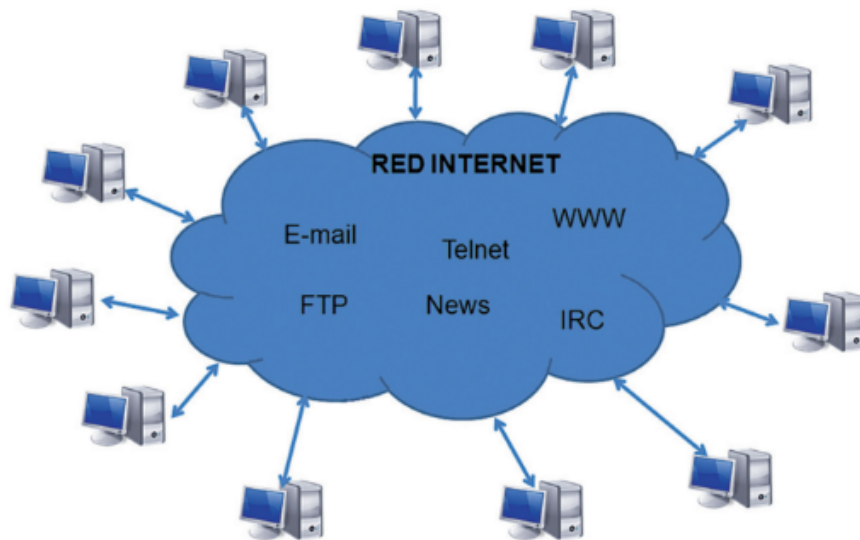


Figura 4
Red internet y servicios básicos.

Y fue a partir de 1993 cuando se produjo la primera versión del navegador mosaico que tuvo un crecimiento como medio de comunicación la internet.

La web (WWW)

La red global mundial, por sus siglas en inglés World Wide Web (WWW), fue creada en 1989 en el Centro Europeo de Investigaciones Nucleares (CERN), por Tim Berners y Robert Caillau, quienes ocuparon este nombre para un sistema de almacenamiento y recuperación de datos con hipervínculos, popularmente llamado “La Web”; es la que nos permite acceder a gran cantidad de información y a todos los tipos de documentos de manera sencilla (Martí & Marín, 2014).

La web utiliza un protocolo HTTP y creó la web, por este motivo las direcciones comienzan con este protocolo `http://www.google.com`.

Arquitectura Cliente - Servidor

Este tipo de arquitectura consiste en un cliente que solicita una petición a un programa (servidor) el cual le da una respuesta, normalmente los servidores son ordenadores potentes que se encargan de la administración de la información (Corrales, 2010). La interacción cliente-servidor es el soporte de la mayor parte de la comunicación por redes.

Los clientes y los servidores pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es la Internet. En la Figura 5 se muestra el diagrama de interacción del modelo cliente servidor.

Dentro de las principales características se encuentran:

- La relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a los recursos compartidos.
- El concepto de escalabilidad tanto horizontal como vertical.
- Combinación de un cliente que interactúa con el usuario, y un servidor que interactúa con los recursos a compartir.

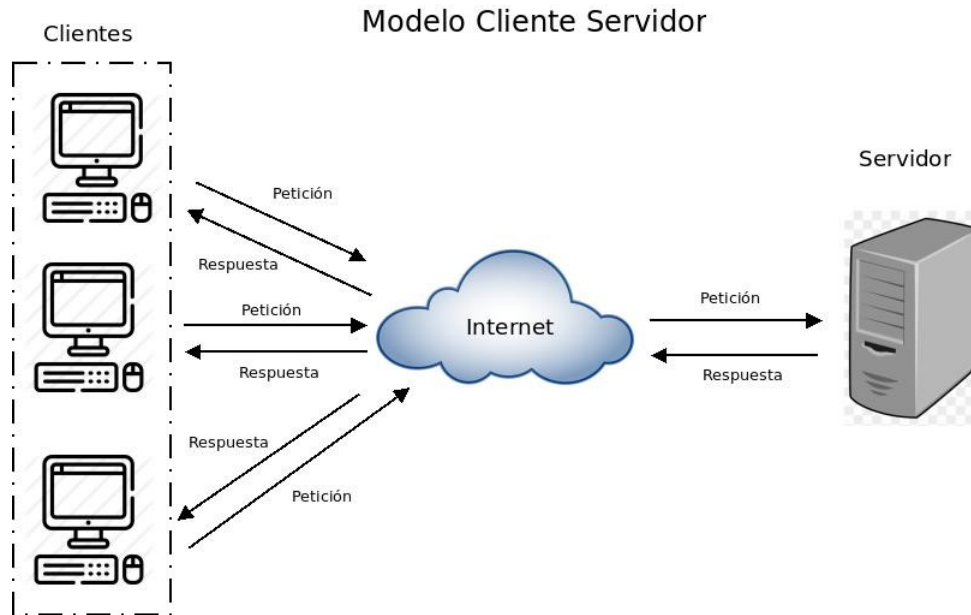


Figura 5
Modelo Cliente Servidor

El tipo de arquitectura más utilizado en esta rama es la de arquitectura de tres capas en la que existe un nivel intermediario (Corrales, 2010), eso significa que la arquitectura generalmente está compartida por un cliente que solicita los recursos equipado con una interfaz de usuario o mediante un navegador web.

La capa del medio es denominada software intermedio cuya tarea es proporcionar los recursos solicitados pero que requiere de otro servidor para hacerlo. La última capa es el servidor de datos que proporciona al servidor de aplicaciones los datos necesarios para poder procesar y generar el servicio que solicitó el cliente en un principio.

Aplicaciones web

Una aplicación web es una herramienta práctica para administrar, recoger y analizar los datos para un usuarios (Martí & Marín, 2014). Estos son utilizados mediante un navegador que accede a un Servidor web a través del internet.

Una aplicación web debe contener elementos que permitan una comunicación activa entre el usuario y la información. Normalmente se utiliza una estructura de tres capas:

- *Una base de datos*, donde se almacena precisamente eso; todos los datos generales, permisos, contenidos, información de interés, etc.
- *El código de la aplicación*, la cual se aloja en un servidor en la nube. En algunos casos se almacena en un servidor local.
- *Acceso a través del navegador*, donde se puede entrar a través de un computador o dispositivo móvil por medio de un navegador.

Tecnologías móviles

Las aplicaciones móviles se basan en ciertas tecnologías y lenguajes, que se utilizan para programar una aplicación en sus respectivas plataforma, así que se denominara “Aplicaciones móviles” a aquellos programas que son desarrollados para ser ejecutada a través de un dispositivo móvil inteligente, tablet u otro para la cual se desee implementar (Vittone & Cuello, 2013).

Los tipos de aplicaciones móviles que se conocen son:

- *Aplicaciones nativas*. Las aplicaciones nativas son aquellas desarrolladas bajo un lenguaje y entorno de desarrollo específico, lo cual permite, que su funcionamiento sea muy fluido y estable para el sistema operativo que fue creada.
- *Aplicaciones web*. Son aquellas desarrolladas usando lenguajes para el desarrollo web como lo son HTML, CSS y JavaScript y un Framework para el desarrollo de aplicaciones web, como por ejemplo JQuery Mobile, Sencha, Kendo UI, entre otros. Se podría decir que este tipo de aplicaciones es muy usada para brindar accesibilidad a la información desde cualquier dispositivo, sin importar el sistema operativo, ya que solo se necesita contar con un navegador para acceder a esta.
- *Aplicaciones híbridas*. Las aplicaciones híbridas se desarrolla utilizando lenguajes de desarrollo web y un framework dedicado para la creación de aplicaciones híbridas. La

facilidad que brinda este tipo de desarrollo es que no hay un entorno específico para su desarrollo y la mayoría de las herramientas son de uso gratuito.

De todos los terminales, el teléfono móvil es uno de los más dinámicos por lo que a su evolución se refiere. Es así como a tenido un gran impacto en el diario vivir de millones de personas y de empresas en general que utilizan esta grandiosa herramienta.

Metodologías para el desarrollo de software

Las metodologías para el desarrollo de software se refiere a un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo de un sistema de información hasta su implementación.

Existen diferentes enfoques en las metodologías de desarrollo de software los cuales se describen a continuación:

Modelo en cascada. Es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma tal que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior (Pressman, 2010, p. 33), tal como se muestra en la Figura 6.

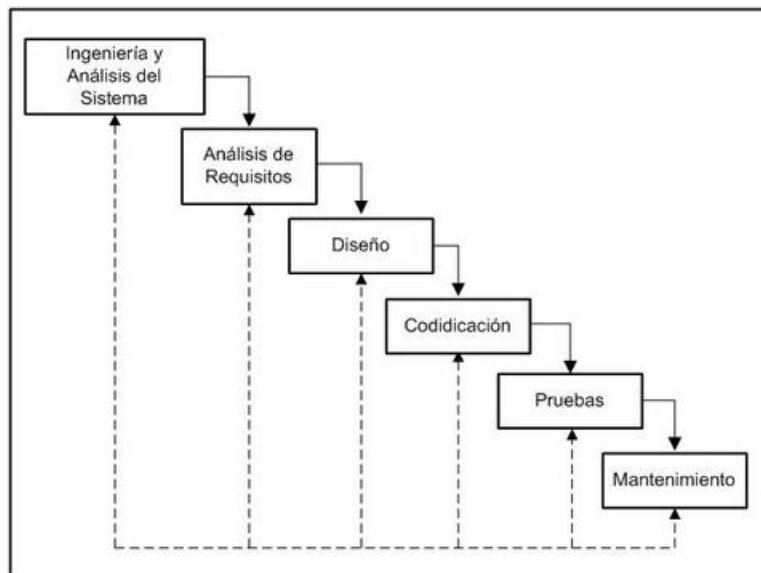


Figura 6
Modelo en cascada

De esta forma, cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costos del desarrollo.

Fases del modelo en cascada. Las fases del modelo en cascada son las siguientes:

1. *Ingeniería y Análisis del Sistema.* Debido a que el software es siempre parte de un sistema mayor, el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.
2. *Análisis de los requisitos del software.* El proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El ingeniero de software debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.
3. *Diseño.* El diseño del software se enfoca en cuatro atributos distintos del programa; la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.
4. *Codificación.* El diseño debe traducirse en una forma legible para la máquina. Si el diseño se realiza de una manera detallada, la codificación puede realizarse mecánicamente.
5. *Prueba.* Una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.

6. *Mantenimiento*. El software sufrirá cambios después de que se entrega al cliente. Los cambios ocurrirán debidos a que se haya encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos) o a que el cliente requiera ampliaciones funcionales o del rendimiento.

Modelo según Kendall y Kendall. Los autores Kendall y Kendall (1997) tienen el siguiente concepto del ciclo de vida:

El ciclo de vida del desarrollo de sistemas es un enfoque por fases para el análisis y el diseño cuya premisa principal consiste en que los sistemas se desarrollan mejor utilizando un ciclo específico de actividades del analista y el usuario. (p. 558)

Según el autor, el ciclo de vida de un sistema consta de siete partes:

1. *Identificación del problema*. En la primera fase el analista es el encargado de identificar los problemas de la organización, detallarlos, examinar, evaluar las oportunidades y objetivos. El analista debe identificar y evaluar los problemas existentes en la organización de manera crítica y precisa. Mayormente los problemas son detectados por alguien más y es cuando el analista es solicitado a fin de precisarlos.

Las actividades de esta fase son las entrevistas a los encargados de coordinar a los usuarios, sintetizar el conocimiento obtenido, estimar el alcance del proyecto y documentar los resultados. El resultado de esta fase es un informe de viabilidad que incluye la definición del problema y un resumen de los objetivos. La administración debe decidir si se sigue adelante o si se cancela el proyecto propuesto.

2. *Identificación de requisitos de información*. En esta fase el analista se esfuerza por comprender la información que necesitan los usuarios para llevar a cabo sus actividades. Entre las herramientas que se utilizan para determinar los requerimientos

de información de un negocio se encuentran métodos interactivos como las entrevistas, los muestreos, la investigación de datos impresos y la aplicación de cuestionarios; métodos que no interfieren con el usuario como la observación del comportamiento de los encargados de tomar las decisiones y sus entornos e oficina, al igual que métodos de amplio alcance como la elaboración de prototipos.

Esta fase es útil para que el analista confirme la idea que tiene de la organización y sus objetivos. Los implicados en esta fase son el analista y los usuarios, por lo general los trabajadores y gerentes del área de operaciones.

3. *Análisis de las necesidades del sistema.* En esta fase el analista evalúa las dos fases anteriores, usa herramientas y técnicas como el uso de diagramas de flujo de datos para ilustrar las entradas, los procesos y las salidas de las funciones del negocio en una forma gráfica estructurada.
4. *Diseño del sistema recomendado.* El analista utiliza los resultados de las primeras fases para realizar el diseño lógico del sistema. Este diseña procedimientos precisos para la captura de información asegurándose que los datos que ingresen a la aplicación sean correctos. Además, esta fase se encarga del diseño de vistas o plantillas, bases de datos y los controles y procedimientos respectivos de respaldo para protección del sistema.
5. *Desarrollo y documentación del sistema.* En la quinta fase del ciclo del desarrollo de sistemas, el analista trabaja de manera conjunta con los programadores para desarrollar cualquier software original necesario. Entre las técnicas estructuradas para diseñar y documentar software se encuentran los diagramas de estructuras, los diagramas de Nassi-Shneiderman y el pseudocódigo. Durante esta fase el analista trabaja con los usuarios para desarrollar documentación efectiva para el software, como manuales de procedimientos, ayuda en línea y sitios web que incluyan respuestas a preguntas frecuentes en archivos “léame” que se integrarán al nuevo software.

6. *Prueba y mantenimiento.* Antes de poner en funcionamiento el sistema es necesario probarlo es mucho menos costoso encontrar los problemas antes que el sistema se entregue a los usuarios. Una parte de las pruebas la realizan los programadores solos, y otra la llevan a cabo de manera conjunta con los analistas de sistemas. Primero se realizan las pruebas con datos de muestra para determinar con precisión cuáles son los problemas y posteriormente se realiza otra con datos reales del sistema actual. El mantenimiento del sistema de información y su documentación empiezan en esta fase y se llevan de manera rutinaria durante toda su vida útil.
7. *Implementación y evaluación.* Esta es la última fase del desarrollo de sistemas, y aquí el analista participa en la implementación del sistema de información. En esta fase se capacita a los usuarios en el manejo del sistema. Parte de la capacitación la imparten los fabricantes, pero la supervisión de ésta es responsabilidad del analista de sistemas. Se menciona la evaluación como la fase final del ciclo de vida del desarrollo de sistemas principalmente en áreas del debate. En realidad, la evaluación se lleva a cabo durante cada una de las fases.

SCRUM. SCRUM es una metodología de desarrollo ágil que permite trabajar en una serie de interacciones en equipo (Subra, 2020). Las fases que definen y en las que se divide un proceso de SCRUM son las siguientes:

- *El quién y el qué:* identifica los roles de cada uno de los miembros del equipo y define su responsabilidad en el proyecto.
- *El dónde y el cuándo:* que representan el Sprint.
- *El por qué y el cómo:* representan las herramientas que utilizan los miembros de Scrum.

Roles. El equipo de Scrum consiste en tres diferentes roles:

- El Product Owner/Dueño del producto es la “voz del cliente” y el responsable de desarrollar, mantener y priorizar las tareas en el backlog.
- El Scrum Master es responsable de asegurarse que el trabajo del equipo vaya bien siguiendo las bases de Scrum. Además, se encarga de remover cualquier obstáculo que pueda encontrar el equipo de desarrollo.
- Los Development Team Members/Miembros del Equipo de desarrollo son los encargados de escribir y probar el código.

El Sprint es la unidad básica de trabajo para un equipo Scrum. Esta es la característica principal que marca la diferencia entre Scrum y otros modelos para el desarrollo ágil. Es una simple iteración llevada a cabo por los miembros del equipo. Un equipo puede completar varios sprints durante el desarrollo del proyecto. Un Sprint inicia con un equipo que se compromete a realizar el trabajo y finaliza con la demostración de un entregable. El esquema se ilustra en la Figura 7.

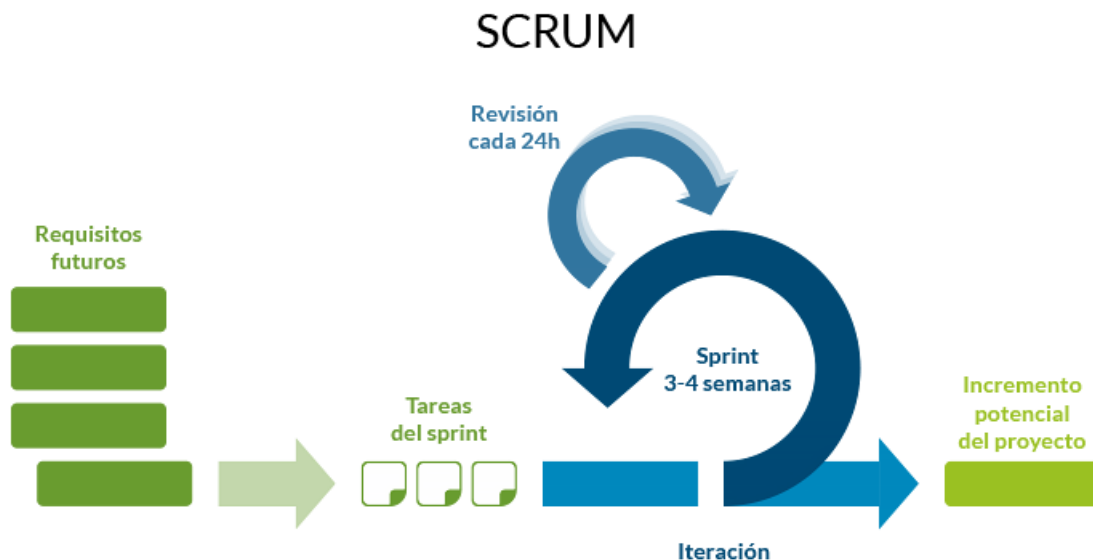


Figura 7
Diagrama de metodología SCRUM.

Metodologías para pruebas del sistema

Las pruebas son un elemento muy importante ya que estas nos permiten desarrollar software de calidad. Estas no pueden asegurar la ausencia de defectos, sólo puede demostrar que existen y están presentes en el software (Lozano, 2015).

Las pruebas que se realizarán al sistema son:

- *Pruebas unitarias:* prueban el diseño y comportamiento de cada componente del sistema una vez construido. Las pruebas de unidad es un proceso para probar los subprogramas, subrutinas, los procedimientos individuales o las funciones, es decir, se deben probar primero los bloques desarrollados, e inicialmente probar el software en su totalidad.
- *Pruebas de integración:* prueban la correcta relación entre los componentes del sistema a través de sus interfaces y si ellas cumplen con la funcionalidad establecida. Consiste en unir el sistema a partir de los distintos componentes y probarlos con todos integralmente. Estas pruebas se realizan de forma progresiva. Entre los tipos de integración se encuentran:
 - Integración incremental: Cada componente se prueba con el conjunto de componentes que ya fueron probados y se va incrementando el número de componentes a probar.
 - Integración no incremental: cada componente se prueba de forma individual y luego se integran todos de una vez, realizando las pruebas respectivas.
 - Pruebas de Sistema: prueban el sistema comprobando su funcionalidad y atributos de calidad. En ésta el sistema es probado en un ambiente lo más parecido posible al ambiente operacional. Estas 44 buscan probar el sistema como un todo. Están basadas en los requerimientos generales y abarca las partes combinadas del sistema.

- *Pruebas de aceptación:* evalúan que el sistema cumple con todos los requisitos indicados y permite que los usuarios del sistema provean su aceptación.
- *Pruebas de regresión:* el objetivo es comprobar que los cambios sobre un componente del sistema no generan errores adicionales en otros componentes no modificados.

Metodologías para documentación

Cuando se desarrolla un Sistema informático, se debe de tener en cuenta la documentación. La documentación de sistema consiste en un conjunto de información relacionada a un sistema determinado que explica las características técnicas, las diferentes funciones del sistema, la lógica de los procesos, los diagramas de flujos, los programas, la naturaleza, capacidades del sistema y cómo usarlo, en otras palabras, la documentación de sistemas es el conjunto de información que nos dice qué hacen los sistemas, cómo lo hacen y para quién lo hacen (Del Sur, 2017). Esta consiste en la elaboración de manuales los cuales son la guía que muestra al usuario, el uso, instalación y configuración de la aplicación.

Entre los manuales que se desarrollarán en este proyecto están:

- *Manual de Usuario:* muestra a los usuarios los pasos para poder manejar de una forma correcta el servidor web.
- *Manual de Instalación:* muestra los pasos a seguir para poder instalar el sistema
- *Manual de Programador:* este manual es destinado para el administrador del sistema, el cual debe tener conocimientos básicos en programación Web como HTML, PHP y administración del Gestor de base de datos MySQL.

Herramientas para el desarrollo de software

UML. UML proporciona un conjunto estandarizado de herramientas para documentar el análisis y diseño de un sistema de software (Kendall & Kendall, 1997). El conjunto de herramientas de UML incluye diagramas que permiten a las personas

visualizar la construcción de un sistema orientado a objetos, similar a la forma en que un conjunto de planos permite a las personas visualizar la construcción de un edificio.

Modelado de casos de uso. El UML está basado fundamentalmente en una técnica de análisis orientada a objetos conocida como modelado de casos de uso, en la cual la palabra uso se pronuncia como sustantivo en lugar de verbo. Un modelo de caso de uso describe lo que hace un sistema sin describir cómo lo hace; es decir, es un modelo lógico del sistema.

El modelo de caso de uso refleja la vista del sistema desde la perspectiva de un usuario fuera del sistema (es decir, los requerimientos del sistema). El UML se puede usar para analizar el modelo de caso de uso y para derivar objetos del sistema y sus interacciones entre sí y con los usuarios del sistema. Usando las técnicas de UML, analiza más a fondo los objetos y sus interacciones para derivar comportamiento del objeto, atributos y relaciones.








Símbolos del caso de uso. Un diagrama de caso de uso contiene el actor y símbolos de caso de uso, junto con líneas de conexión, las cuales se muestran en el Cuadro 1. Los actores son parecidos a las entidades externas; existen fuera del sistema. El término actor se refiere a un papel particular de un usuario del sistema.

Por motivos de ejemplificar, un actor podría ser un empleado, pero también podría ser un cliente en el almacén de la compañía. Aunque quizás es la misma persona en el mundo real, se representa como dos símbolos diferentes en un diagrama de caso de uso, debido a que la persona interactúa con el sistema en diferentes papeles.

El actor existe fuera del sistema e interactúa con éste de una forma específica. Un actor puede ser un humano, otro sistema o un dispositivo tal como un teclado, módem o conexión Web. Los actores pueden iniciar una instancia de un caso de uso. Un actor podría interactuar con uno o más casos de uso y viceversa.

Los actores se podrían dividir en dos grupos. Los actores principales proporcionan datos o reciben información del sistema. Los actores secundarios ayudan a mantener el sistema en ejecución o proporcionan ayuda. Éstas son las personas que operan el centro de

atención telefónica, los analistas, programadores, etcétera.

Símbolo	Relación	Significado
	Comunica	Un actor se conecta a un caso de uso usando una línea sin punta de flecha.
	Incluye	Un caso de uso contiene un comportamiento que es más común que otro caso de uso. La flecha apunta al caso de uso común.
	Extiende	Un caso de uso diferente maneja las excepciones del caso de uso básico. La flecha apunta desde el caso de uso extendido hacia el básico.
	Generaliza	Una cosa de UML es más general que otra cosa. La flecha apunta a la cosa general.
	Actor	Se refiere a las entidades externas que interactúan con el sistema.
	Caso de uso	Son las acciones que realiza el actor con el sistema.
	Limite del sistema	Muestra de una forma gráfica el limite que tendrá el sistema a desarrollar.

Cuadro 1

Símbolos para el diagrama de casos de uso.

Un caso de uso proporciona a los desarrolladores una visión de lo que quieren los

usuarios. No contiene detalles técnicos o de implementación. Podemos pensar en un caso de uso como una secuencia de transacciones en un sistema. El modelo de caso de uso se basa en las interacciones y relaciones de casos de uso individuales.

Un caso de uso siempre describe tres cosas:

1. Un actor que inicia un evento.
2. El evento que activa un caso de uso.
3. El caso de uso que desempeña las acciones activadas por el evento.

En un caso de uso, un actor que usa el sistema comienza un evento que empieza una serie relacionada de interacciones en el sistema. Los casos de uso se utilizan para documentar una sola transacción o evento. Un evento es una entrada al sistema que pasa en un tiempo y lugar específicos y ocasiona que el sistema haga algo.

Diagramas de secuencia. Los diagramas de secuencias pueden ilustrar una sucesión de interacciones entre clases o instancias de objetos en un periodo determinado. Los diagramas de secuencias se utilizan con frecuencia para representar el proceso descrito en los escenarios de caso de uso. En la práctica, los diagramas de secuencias se derivan del análisis de casos de uso y se emplean en el diseño de sistemas para generar las interacciones, relaciones y métodos de los objetos del sistema.

Los diagramas de secuencias se utilizan para mostrar el patrón general de las actividades o interacciones en un caso de uso. Cada escenario de caso de uso podría crear un diagrama de secuencias, aunque no siempre se crean diagramas de este tipo para los escenarios menores.

El objeto del extremo izquierdo es el objeto inicial y podría ser una persona (para la cual se emplea símbolo de actor de caso de uso), una ventana, un cuadro de diálogo u otra interfaz de usuario. Algunas de las interacciones sólo son físicas, como firmar un contrato. Los rectángulos de la parte superior usan indicadores en el nombre para denotar si el rectángulo representa un objeto, una clase, o una clase y un objeto.

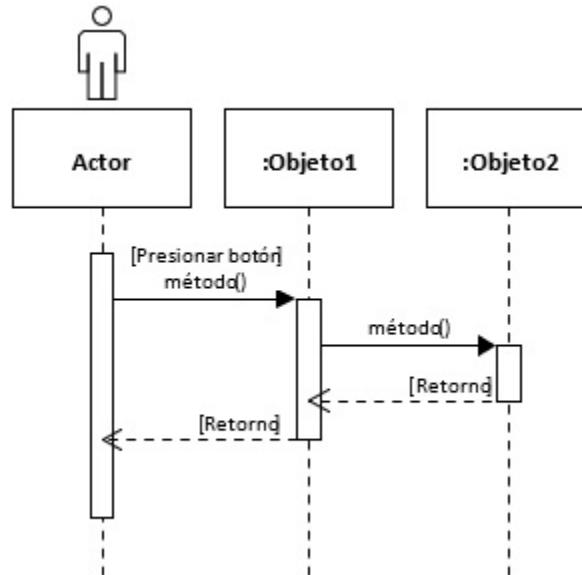


Figura 8
Ejemplo de diagrama de secuencia.

Una línea vertical representa la trayectoria de la vida de la clase o del objeto, que comienza cuando se crea y finaliza cuando se destruye. Una X en el fondo de la trayectoria de la vida indica cuándo se destruye el objeto. Una barra lateral o rectángulo vertical en la trayectoria de la vida muestran el enfoque de control cuando el objeto se encuentra realizando algo.

Las flechas horizontales muestran mensajes o signos que se envían entre las clases. Los mensajes pertenecen a la clase receptora. Hay algunas variaciones en las flechas de mensaje. Las puntas de flecha sólidas representan llamadas síncronas, que son las más comunes. Éstas se usan cuando la clase emisora espera una respuesta de la clase receptora, y el control se devuelve a la clase emisora cuando la clase que recibe el mensaje termina su ejecución.

Las flechas con media punta (o abiertas) representan llamadas asíncronas, es decir, llamadas que se envían sin esperar a que sean devueltas a la clase que las emite. Un ejemplo podría ser el de usar un menú para ejecutar un programa. Un retorno se muestra como una flecha, a veces con una línea punteada.

En el diagrama de secuencias, el tiempo se despliega de arriba hacia abajo. La primera interacción se representa en la parte superior del diagrama, y la última, en la parte inferior.

Las flechas de interacción comienzan en la barra del actor o del objeto que inicia la interacción, y terminan apuntando hacia la barra del actor o el objeto que recibe la solicitud de interacción. El actor, la clase o el objeto iniciales se muestran a la izquierda. Éste podría ser el actor que inicia la actividad o podría ser una clase que represente la interfaz de usuario.

PostgreSQL. PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales (PostgreSQL, 2020).

PostgreSQL utiliza un modelo cliente-servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Esto es ilustrado en la Figura 9.

Características. La última serie de producción es la 9.3. Sus características técnicas la hacen una de las bases de datos más potentes y robustos del mercado. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.

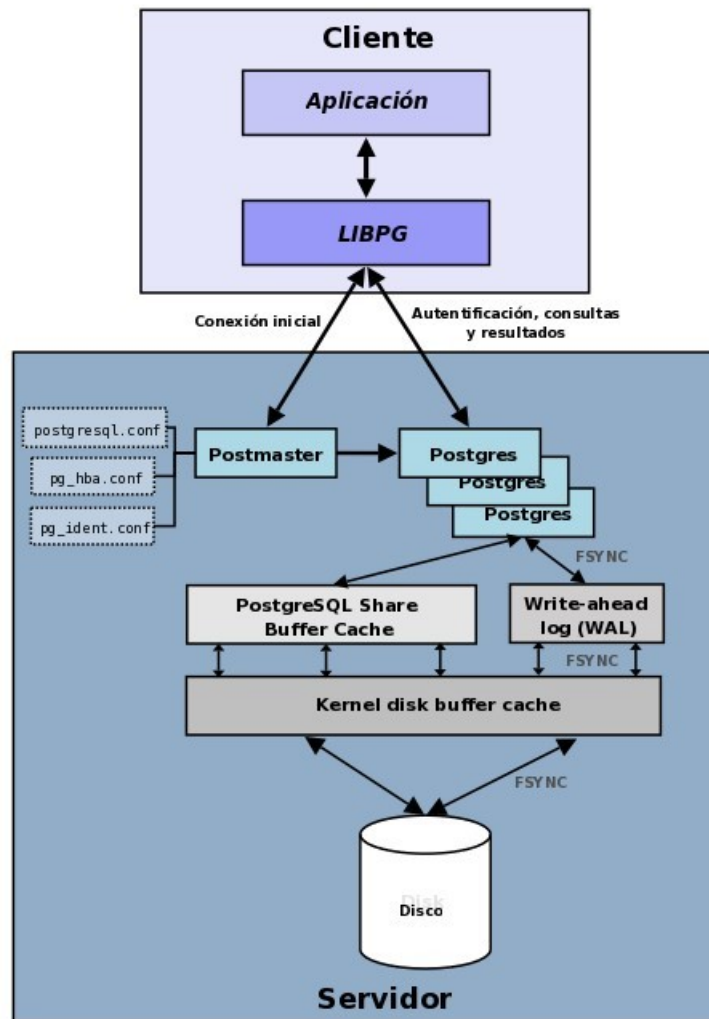


Figura 9
Componentes importantes en un sistema PostgreSQL

Entre las características destacadas se encuentran:

- Es una base de datos 100% ACID⁷.
- Integridad referencial.
- Tablespaces, ayuda a definir ubicaciones del sistema de archivos de la base de datos.
- Transacciones.

⁷ Del acronimo Atomicity, Consistency, Isolation and Durability: Atomicidad, Consistencia, Aislamiento y Durabilidad en español.

- Replicación asincrónica/sincrónica / Streaming replication - Hot Standby.
- Two-phase commit.
- Juegos de caracteres internacionales.
- Copias de seguridad en caliente (Online/hot backups).
- Multi-Version Concurrency Control⁸, permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente.
- Múltiples métodos de autenticación.
- Acceso encriptado vía SSL.
- Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.

Características en programación/desarrollo. PostgreSQL se utiliza ampliamente en el desarrollo de software con diversos lenguajes de programación. Entre sus características en esta área destacan:

- Funciones/procedimientos almacenados (stored procedures) en numerosos lenguajes de programación, entre otros PL/pgSQL (similar al PL/SQL de oracle), PL/Perl, PL/Python y PL/Tcl.
- Bloques anónimos de código de procedimientos (sentencias DO).
- Numerosos tipos de datos y posibilidad de definir nuevos tipos. Además de los tipos estándares en cualquier base de datos, tenemos disponibles, entre otros, tipos geométricos, de direcciones de red, de cadenas binarias, UUID, XML, matrices, etc.

⁸ Acceso Concurrente Multiversión, por sus siglas en inglés MVCC.

- Soporta el almacenamiento de objetos binarios grandes (videos, sonido, etc.).
- APIs para programar en C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, PHP, Lisp, Scheme, Qt y muchos otros.

Características SQL. Las características principales en administración de bases de datos relacionales son:

- SQL92, SQL99, SQL2003, SQL2008.
- Llaves primarias (primary keys) y foráneas (foreign keys).
- Restricciones: Check, Unique y Not null.
- Restricciones de unicidad postergables (deferrable constraints).
- Columnas auto-incrementales.
- Índices compuestos, únicos, parciales y funcionales en cualquiera de los métodos de almacenamiento disponibles, B-tree, R-tree, hash ó GiST.
- Sub-consultas (subqueries).
- Consultas recursivas.
- Funciones 'Windows'.
- Uniones (joins).
- Vistas (views).
- Disparadores (triggers).
- Reglas (rules).
- Herencia de tablas (Inheritance).
- Eventos LISTEN/NOTIFY.

Licencia. PostgreSQL es distribuido bajo licencia BSD (tipo de licencia de software libre permisiva) y con su código fuente disponible libremente.

Justificación. La DTIC maneja ciertos estándares de desarrollo para implementar en sus servidores. Uno de ellos es que utilizan PostgreSQL como gestor de base de datos, por tener todas las características descritas anteriormente. Por lo tanto, se seguirá con estándar de la DTIC para sus proyectos de desarrollo (MINSAL, 2011a).

PgAdmin 4. PgAdmin es una herramienta indispensable para gestionar y administrar PostgreSQL, la base de datos de código abierto más avanzada del mundo. Por lo tanto pgAdmin es la herramienta para gestionar bases de datos espaciales PostGIS (MappingGIS, 2020).

Características. Incluye un conjunto de iconos actualizado y fuentes incrustadas para mostrar una apariencia coherente en todas las plataformas.

Esta aplicación está diseñada para funcionar tanto en escritorio como en un servidor web. Se puede implementar directamente en un servidor web para que uno o más usuarios lo utilicen a través de su navegador web.

Destaca el completo Panel de control, que sirve para monitorizar el estado del servidor y de las bases de datos. La pestaña Panel de control se puede cerrar si se desea, para minimizar el tráfico de consultas resultante de la actualización del gráfico. Además desde el Panel de control se pueden cancelar las consultas, mostrado en la Figura 10.

La interfaz de usuario es más flexible, permite que las pestañas se acoplen y reorganicen en más formas que anteriormente. Ahora es posible disponer de pestañas desmontables, lo que permite que la Herramienta de consulta (Query Tool) y el Depurador se abran en pestañas nuevas y luego se puedan mover a pantallas alternativas. Se minimiza la necesidad de cambiar entre diálogos, usando cuadros combinados que se pueden buscar y más.

Mejoras de velocidad significativas, particularmente en el momento de inicio y en la herramienta de consulta (Query Tool) que permite ver los resultados de consultas de gran

tamaño mucho más rápido. Por ejemplo, se ha realizado un test en I que se ha ejecutado una consulta simple con 96.000 filas y se ha mostrado el resultado en tan solo 1 segundo.

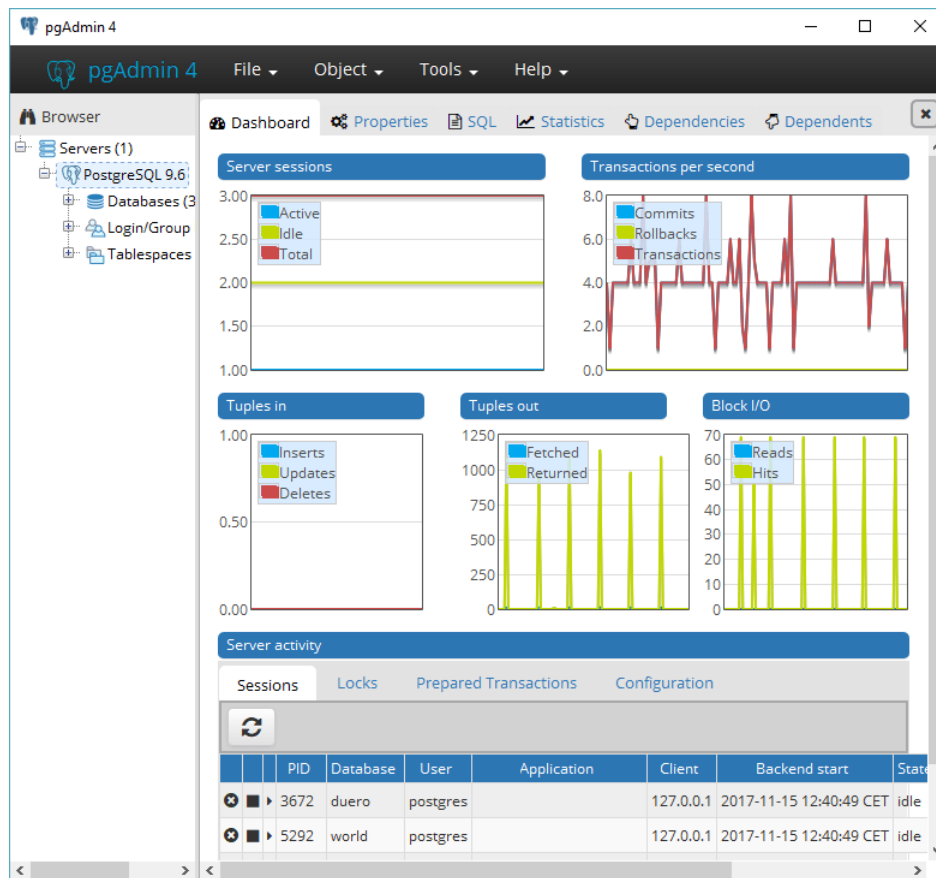


Figura 10
Vista de dashboard en pgAdmin 4.

La Cuadrícula de resultados de la Herramienta de consulta se ha revisado con un aspecto novedoso y elegante, con soporte de selección de filas, columnas o bloques de celdas y se permite la copia completa y retención del tamaño de columna. También incluye soporte para autenticación basada en certificados SSL.

Desventajas. La arquitectura de pgAdmin 4 requiere, como es natural, más CPU y memoria que una aplicación de escritorio tradicional. PgAdmin 4 también incluye una actualización continua del panel de control.

El panel actualiza los gráficos cada segundo de forma predeterminada, incluso si no están visibles (para evitar lagunas en los datos). El uso de la CPU puede reducirse

notablemente ajustando la velocidad de actualización de los gráficos a un período más largo, como por ejemplo diez segundos.

Justificación. PgAdmin es una de las mejores herramientas para manejar bases de datos de PostgreSQL. Es una aplicación open source, sencilla de manejar y ofrece todas las funcionalidades avanzadas de la base de datos.

XAMPP. XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP esta disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X (EcuRed, 2020q).



Figura 11
Logo de XAMPP.

Características y requisitos. XAMPP solamente requiere descargar y ejecutar un archivo .zip, .tar, o .exe, con unas pequeñas configuraciones en alguno de sus componentes que el servidor Web necesitará. XAMPP se actualiza regularmente para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. También incluye otros módulos como OpenSSL y phpMyAdmin. Para instalar XAMPP se requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar los programas por separado.

Aplicaciones. Los diseñadores de XAMPP fueron los "Baiker y Anthony Corporation", quienes solo pretendían su uso como una herramienta de desarrollo, para permitir a los diseñadores de sitios webs y programadores testear su trabajo en sus propios

ordenadores cuando no tienen ningún acceso a Internet.

En la práctica, sin embargo, XAMPP se utiliza actualmente como servidor de sitios web, ya que, con algunas modificaciones, es generalmente lo suficientemente seguro para serlo. Con el paquete se incluye una herramienta especial para proteger fácilmente las partes más importantes en una página.

Justificación. XAMPP es una de las herramientas con las que el equipo de desarrollo se encuentra familiarizado. Se utilizará para testear y desarrollar en modo local la aplicación web del sistema de administración de peticiones para el Departamento de Informática.

PowerDesigner. PowerDesigner DataArchitect es la herramienta lidera de modelización de datos. Permite fortalecer y alinear negocio y IT. PowerDesigner permite a los administradores de bases de datos visualizar, analizar y manipular de manera más fácil los Metadatos para tener una arquitectura de información de organización eficaz (SAP, 2020).

Características. Entre las principales características (EcuRed, 2020l) de la herramienta PowerDesigner se encuentran:

- Permite a las empresas, de manera más fácil, visualizar, analizar y manipular metadatos, logrando un efectiva arquitectura empresarial de información.
- Brinda un enfoque basado en modelos, el cual permite alinear al negocio con la tecnología de información, facilitando la implementación de arquitecturas efectivas de información empresarial.
- Brinda potentes técnicas de análisis, diseño y gestión de metadatos a la empresa.
- Combina varias técnicas estándar de modelamiento con herramientas líder de desarrollo, como .NET, Sybase WorkSpace, Sybase Powerbuilder, Java y Eclipse, para darle a las empresas soluciones de análisis de negocio y de diseño formal de base de datos.

- Trabaja con más de 60 bases de datos relacionales.

Beneficios. Los principales beneficios que aporta PowerDesigner en proyectos informáticos son:

- *Mejora la productividad individual.* El enfoque orientado a modelos de PowerDesigner incorpora una serie de generadores DDL y de código personalizables, y capacidades de ingeniería reversa y sincronización de código, que reduce significativamente los esfuerzos de creación, mantenimiento y reingeniería manual de código.
- *Brinda facilidad de uso gráfica.* La interfaz gráfica es altamente personalizable, hace que las tareas comunes sean muy fáciles y le da el poder a los usuarios avanzados de tener acceso rápido a todas las funciones.
- *Alinea el negocio con el área de tecnología.* Facilita el alineamiento del negocio con el área de tecnología a través de técnicas de colaboración en grupo.
- *Mejora la productividad en grupo.* Brinda a todos los modeladores un ambiente ideal para compartir recursos a través de un repositorio de metadatos único, completo y seguro para todos los tipos de modelos.
- *Documenta los sistemas existentes.* Adopta una mayor colaboración a nivel empresarial a través de generación de reportes flexible y basada en asistentes, o RTF / HTML multi-modelo.
- *Es altamente personalizable.* Puede ser fácilmente programado para asegurar los estándares y prácticas corporativas o legales, a través del soporte a scripts VB, una interfaz COM completamente programable, un meta-modelo personalizable y un API totalmente documentado.
- *Reduce el impacto del cambio.* Reduce significativamente el costo y tiempo al implementar cualquier cambio a través de una vista exacta, bidireccional y

multimodelo para análisis de impacto que integra todos los modelos de requerimientos, análisis, base de datos y aplicación.

PowerDesigner soporta las técnicas de modelización siguientes:

- *Modelización de datos.* Soporta los modelos de datos conceptuales, lógicos y físicos, así como las extensiones de modelado de almacén, basados en Ingeniería de Información o en la notación IDEF 1/x Más información sobre la modelización de datos.
- *Modelización de aplicación.* Ofrece un mapeo relacional/de objeto avanzado para una gestión de implementación persistente. Soporta también las técnicas de modelización específicas XML relacionadas con modelos de datos y UML.
- *Modelización del proceso de negocio.* Ofrece una descripción no técnica e intuitiva de los procesos de negocios y una definición de diagramas así como la modelización de lenguajes de ejecución de procesos.

Justificación. PowerDesigner ofrece una amplia gama de herramientas de modelado para desarrollo. Entre ellas, las que se utilizarán son los diagramas Entidad-Relación y diagrama físico para el diseño de la base de datos.

Mozilla Firefox. Mozilla Firefox es un navegador web libre y de código abierto desarrollado para Linux, Android, IOS OS X y Microsoft Windows coordinado por la Corporación Mozilla y la Fundación Mozilla (Mozilla, 2020). Usa el motor Gecko para renderizar páginas web, el cual implementa actuales y futuros estándares web. Cabe destacar que esta aplicación se utilizará como interprete para compilar y realizar pruebas de funcionamiento del sistema en desarrollo.

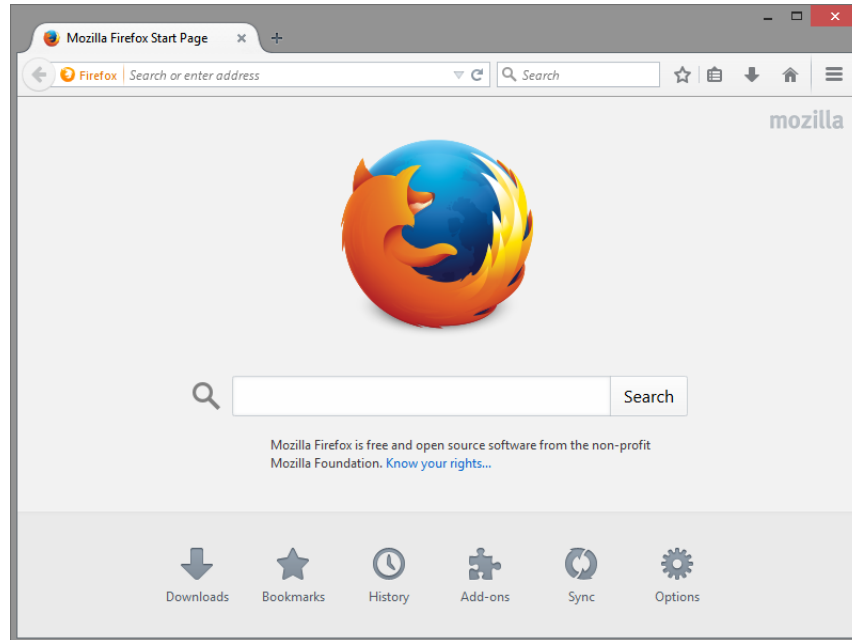


Figura 12

Página de inicio de Mozilla Firefox.

Características. Mozilla Firefox posee características que lo hacen uno de los mejores navegadores para uso personal y desarrollo web. Entre las principales se encuentran:

- **Interfaz.** Firefox está basado en la interfaz Starta, diseñada por el contribuidor Stephen Horlander. El enfoque consiste en un menú unificado (no disponible en OS X) con las pestañas de navegación hacia arriba, y la barra de direcciones con los botones de acción debajo. Con ella, es posible personalizar mediante temas. Cada tema puede decorar de fondo (un papel tapiz de hojas o de paisajes) o cambiar drásticamente al navegador.
- **Estándares web.** Es compatible con varios lenguajes web, incluyendo HTML, XML, XHTML, SVG 1.1 (parcial), CSS 1, 2 y 3, ECMAScript (JavaScript), DOM, MathML, DTD, XSLT, XPath, e imágenes PNG con transparencia alfa. También incorpora las normas propuestas por el WHATWG, y es compatible con el elemento HTML Canvas.

- *Seguridad.* Implementa el sistema SSL/TLS para proteger la comunicación con los servidores web, utilizando fuerte criptografía cuando se utiliza el protocolo https. También soporta tarjetas inteligentes para fines de autenticación. Cuenta con una protección antiphishing, antimalware e integración con el antivirus. También y como medida prudencial que ha causado controversia, no incluye compatibilidad con los sistemas ActiveX, debido a la decisión de la Fundación Mozilla de no incluirlo por tener vulnerabilidades de seguridad.
- *Localizaciones.* Es el navegador web más localizado hasta la fecha, cubriendo el 97 % de la población con conexión a internet. El primer lanzamiento oficial en noviembre de 2011 fue distribuido en 28 diferentes idiomas, incluyendo inglés británico/inglés estadounidense, Español de España/Español rioplatense y chino en caracteres chinos tradicionales/caracteres chinos simplificados. Las versiones actuales con soporte 10.0.7 y 15.0.1 están disponibles para 85 variantes regionales (77 idiomas) y 83 variantes regionales (74 idiomas), respectivamente.
- *Licencia.* El código fuente de Firefox es libre y abierto, y es distribuido bajo triple licencia; Licencia Pública de Mozilla (MPL), Licencia pública general de GNU (GPL), o la Licencia pública general reducida de GNU (LGPL). Estas licencias permiten a cualquiera ver, modificar y/o redistribuir el código fuente, y son varias las aplicaciones que se conocen hacen uso: por ejemplo Flock, Miro, GNU IceCat y Songbird están hechos a partir del código de Firefox. Aproximadamente el 40 % del código de Firefox está escrito por voluntarios.

Justificación. Mozilla Firefox es el navegador que se usa por defecto en los equipos del Ministerio de Salud⁹, además de ser de licencia libre, es liviano y fácil de usar. Se utilizará como interprete para el desarrollo de la aplicación web.

⁹ Fuente: Ing. Juan Soto, jefe del Departamento de Informática.

Symfony. Es un framework diseñado para desarrollar aplicaciones web basado en el patrón Modelo Vista Controlador. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación (Symfony, 2020).

Symfony está desarrollado completamente en PHP 5.3. Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.

Características. Symfony fue diseñado para ajustarse a los siguientes requisitos de desarrollo de proyectos de software:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y Unix-like estándares).
- Independiente del sistema gestor de bases de datos. Su capa de abstracción y el uso de ORM (Doctrine 2, Propel), permiten cambiar con facilidad de SGBD en cualquier fase del proyecto.
- Utiliza programación orientada a objetos y características como los espacios de nombres, de ahí que sea imprescindible PHP 5.3.
- Sencillo de usar en la mayoría de casos, aunque es preferible para el desarrollo de grandes aplicaciones web que para pequeños proyectos.
- Aunque utiliza Modelo Vista Controlador (MVC), tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.

- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador sólo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con las bibliotecas de otros fabricantes.
- Una línea de comandos que facilita la generación de código.

Características para el desarrollo automatizado de proyectos web. Las características más comunes para el desarrollo de proyectos web están automatizadas en symfony, tales como:

- Permite la internacionalización para la traducción del texto de la interfaz, los datos y el contenido de localización.
- La presentación usa templates y layouts que pueden ser construidos por diseñadores de HTML que no posean conocimientos del framework.
- Los formularios soportan la validación automática, lo cual asegura mejor calidad de los datos en las base de datos.
- El manejo de cache reduce el uso de banda ancha y la carga del servidor.
- El soporte de autenticación y credenciales facilita la creación de áreas restringidas y manejo de seguridad de los usuarios.

- El enrutamiento y las URLs inteligentes hacen amigable las direcciones de las páginas de la aplicación.
- Permite la paginación, clasificación y filtraje automáticos.
- Los plugins proveen un alto nivel de extensibilidad.
- Dispone de interacción con AJAX.

Justificación. Dentro de los estándares de desarrollo de la DTIC se encuentra Symfony como framework de programación para sus aplicaciones. Por lo tanto, se seguirá con estándar de la DTIC para sus proyectos de desarrollo (MINSAL, 2011a).

Tecnología para programación móvil. En el Cuadro 2, se muestran diferentes herramientas para el desarrollo de aplicaciones móviles. Estos serán evaluados por diferentes criterios para encontrar la que se adaptará mejor al proyecto. Los puntos se considerarán en una escala del 1 al 10, siendo el 10 el puntaje más alto.

Como se puede apreciar, la herramienta que mejor se adapta al proyecto es Flutter, debido a las bondades que poseen las aplicaciones nativas en rendimiento y estabilidad en el tiempo.

Flutter. Es un SDK de aplicaciones móviles para la creación de aplicaciones de alto rendimiento y alta fidelidad para iOS y Android, a partir de un único código base (Flutter-es, 2020).

El objetivo es permitir a los desarrolladores que lancen aplicaciones de alto rendimiento que se adapten de forma natural a diferentes plataformas. Se abarcan las diferencias en los comportamientos de scrolling, tipografía, iconos, y más.

Las aplicaciones están escritas en Dart, un lenguaje de programación muy parecido a Java o JavaScript. Utilizando el enfoque de programación orientada a objetos, el desarrollo de aplicaciones en Flutter puede ser muy veloz.

Nombre	Descripción	Criterios a evaluar				Puntaje total
		Compatibilidad con diferentes SO	Documentación	Licencia	Experiencia	
Ionic	Framework de programación móvil para aplicaciones híbridas.	10	8	Libre	7	25
Flutter	Framework de programación móvil para aplicaciones nativas.	10	9	Libre	7	26
React Native	Es un framework para desarrollar aplicaciones multiplataformas utilizando JavaScript.	10	8	Libre	4	22

Cuadro 2

Criterios de elección de la tecnología móvil.



Figura 13
Alcance de Flutter.

Principios básicos. Flutter incluye un framework moderno de estilo reactivo, un motor de renderizado 2D, widgets listos y herramientas de desarrollo. Estos componentes trabajan juntos para ayudar a diseñar, construir, probar y depurar aplicaciones. Todo está organizado en torno a unos pocos principios básicos:

- *Todo es un widget:* Los widgets son los elementos básicos de la interfaz de usuario de una aplicación Flutter. Cada widget es una declaración inmutable de parte de la interfaz de usuario. A diferencia de otros frameworks que separan vistas, controladores de vistas, layouts y otras propiedades, Flutter tiene un modelo de objeto unificado y consistente: el widget.
- *Herencia entre widget:* Los widgets se componen a menudo de muchos widgets pequeños y de un solo propósito, que se combinan para producir efectos poderosos. Por ejemplo, Container, un widget de uso común está compuesto por varios widgets responsables del diseño, pintado, posicionamiento y dimensionado.
- *Estructura en capas:* El framework Flutter está organizado en una serie de capas, con cada capa construyéndose sobre la capa anterior. El objetivo de este diseño es ayudar

a hacer más con menos código. Por ejemplo, la capa Material se construye componiendo widgets básicos a partir de la capa de widgets, y la capa de widgets se construye organizando los objetos de nivel inferior a partir de la capa de renderizado.

Características. Las principales características que ofrece Flutter como framework de desarrollo para dispositivos móviles son:

- Comunidad en español en crecimiento.
- Desde una única fuente podremos llegar a las plataformas que soporta este Framework (Android e iOS).
- Se puede hacer más con menos código, gracias al lenguaje moderno y expresivo y al enfoque declarativo del framework.
- Se experimenta cambiando el código y recargando a medida que la aplicación se ejecuta gracias al Hot Reload.
- Amplio conjunto de widgets Material Design y Cupertino (de iOS) construidos usando el propio Framework Flutter.
- Puede realizar diseños personalizados, agradables y de marca sin las limitaciones de los conjuntos de widgets OEM.

Justificación. La DTIC actualmente no posee un estándar para desarrollo de aplicaciones móviles. Por lo tanto, para la aplicación móvil se propone utilizar el framework de programación Flutter, que además de poseer una curva de aprendizaje relativamente pequeña, es de licencia libre; tal como lo recomiendan el MINSAL (2011a).

Entorno de Desarrollo Integrado (IDE). Un entorno de desarrollo integrado o entorno de desarrollo interactivo, en inglés Integrated Development Environment (IDE) (EcuRed, 2020e), es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

Nombre	Descripción	Criterios a evaluar			Puntaje total
		Compatibilidad con diferentes SO	Documentación	Soporte para diferentes lenguajes de programación	
Visual Studio Code	Es un editor de código fuente desarrollado por Microsoft para Windows, Linux y MacOS.	10	8	9	37
Sublime Text	Es un editor de código fuente escrito en C++.	10	9	7	33
Atom	Es un editor de código fuente con soporte para multiples plug-in y control de versiones GIT integrado.	10	8	7	32

Cuadro 3
Criterios de elección del entorno de programación.

En el Cuadro 3, se muestran diferentes herramientas para editar código fuente. Estos serán evaluados por diferentes criterios para encontrar la que se adaptará mejor al proyecto. Los puntos se considerarán en una escala del 1 al 10, siendo el 10 el puntaje más alto.

Como se puede apreciar, la herramienta que mejor se adapta al proyecto es Visual Studio Code, por su versatilidad en diferentes lenguajes de programación. Lo que a su vez, nos permite utilizarlo para desarrollo de aplicaciones móviles y aplicaciones web.

Visual Studio Code. Es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para depuración, control de Git integrado, resaltado de sintaxis, finalización de código inteligente, fragmentos de código y refactorización de código. También es personalizable, de modo que los usuarios pueden cambiar el tema del editor, los métodos abreviados de teclado y las preferencias. Es gratuito y de código abierto (VisualStudioCode, 2020).

El código combina la interfaz de usuario optimizada de un editor moderno con asistencia y navegación de código enriquecido y una experiencia de depuración integrada, sin la necesidad de un IDE completo.

Cuenta con herramientas de Debug hasta opciones para actualización en tiempo real de nuestro código en la vista del navegador y compilación en vivo de los lenguajes que lo requieran. Además de las extensiones, tendremos la posibilidad de optar por otros themes o bien configurarlo a nuestro gusto. Para modificar el esquema de colores y los iconos.

Ventajas. Las principales ventajas que posee Visual Studio Code como editor de código son:

- Se puede utilizar como lenguajes de programación.
- Visual Studio Code es una herramienta que tiene soporte nativo para gran variedad de lenguajes, entre ellos podemos destacar los principales del desarrollo Web: HTML, CSS, y JavaScript, entre otros.
- Posibilidad de configurar la interfaz a nuestro gusto. De esta forma, podremos tener

más de un código visible al mismo tiempo, las carpetas de nuestro proyecto y también acceso a la terminal o un detalle de problemas, entre otras posibilidades.

- Existencia de una amplísima gama de temas o estilos visuales para Visual Studio Code, que hacen el trabajo con el Software más agradable a la vista.
- Goza de un soporte técnico formidable pues debido a su frecuente uso por la comunidad de desarrolladores, se puede encontrar fácilmente documentación y ayuda en foros y sitios relacionados.

Justificación. El editor de código fuente Visual Studio Code es una poderosa herramienta que es de código abierto. Esta es elegida por la familiaridad que posee el equipo de desarrollo, además de contar con plugins para adaptar el entorno de programación, un editor de comandos (consola) directamente integrada, y la familiaridad que cuenta el equipo de desarrollo con este IDE.

Marco normativo

Ley de propiedad intelectual

Es la ley encargada en El Salvador de regular el desarrollo de software y especificar sus derechos y deberes y que consideraciones deben tener los desarrolladores, sobre el licenciamiento que este tendrá, cuando el software es entregado a una institución ya sea por un contrato o como una contribución al desarrollo tecnológico los derechos del mismo pasan a ser propiedad de la institución.

Según el **Art. 1**, dicha ley tiene por objeto asegurar una protección suficiente y efectiva de la propiedad intelectual, estableciendo las bases que la promuevan, fomenten y protejan.

Esta ley comprende el derecho de autor, los derechos conexos y la propiedad industrial en lo relativo a invenciones, modelos de utilidad, diseños industriales y secretos industriales o comerciales y datos de prueba.

Según el **Art. 7**, el derecho económico del autor es el derecho exclusivo de autorizar o prohibir el uso de sus obras, así como la facultad de percibir beneficios económicos provenientes de la utilización de las obras. En este caso podemos mencionar el software libre ya que este se rige bajo su propia licencia.

Una licencia de software libre es un impreso que otorga al receptor de una pieza de software derechos extensivos para modificarlo y redistribuirlo. Una de las más utilizadas es la licencia GPL, La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License (o simplemente sus siglas en inglés GNU GPL).

Es la licencia de derecho de autor más ampliamente usada en el mundo del software libre y código abierto, y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software. Su propósito es doble: declarar que el software cubierto por esta licencia es libre, y protegerlo (mediante una práctica conocida como Copyleft) de intentos de apropiación que restrinjan esas libertades a nuevos usuarios cada vez que la obra es distribuida, modificada o ampliada. Esta licencia fue creada originalmente por Richard Stallman fundador de la Free Software Foundation (FSF) para el proyecto GNU. Cabe mencionar que el software libre no quiere decir que sea gratuito si no que permite al desarrollador en este caso quien posea los derechos sobre el software considerar si percibirá o no una remuneración económica.

La sección E de la ley de propiedad intelectual se refiere al software de la siguiente manera:

Art. 32.- Programa de ordenador, ya sea programa fuente o programa objeto, es la obra literaria constituida por un conjunto de instrucciones expresadas mediante palabras, códigos, planes o en cualquier otra forma que, al ser incorporadas en un dispositivo de lectura automatizada, es capaz de hacer que un ordenador, o sea, un aparato electrónico o similar capaz de elaborar informaciones, ejecute determinada tarea u obtenga determinado resultado. Se presume que es productor del programa de ordenador, la persona que aparezca indicada como tal en la obra de la manera acostumbrada, salvo prueba en contrario.

Art. 33.- El contrato entre los autores del programa de ordenador y el productor, implica la cesión ilimitada y exclusiva a favor de éste de los derechos patrimoniales reconocidos en la presente ley, así como la autorización para decidir sobre su divulgación y la de ejercer los derechos morales sobre la obra, en la medida que ello sea necesario para la explotación de la misma, salvo pacto en contrario.

Reglamento general de procesos de graduación de la Universidad de El Salvador.

Derechos de autor. La Universidad de El Salvador dice en su reglamento con respecto a los trabajos de investigación en los procesos de graduación lo siguiente:

Art. 29.- Los derechos de autor sobre los trabajos de investigación elaborados en los procesos de graduación, serán de propiedad exclusiva de la Universidad de El Salvador, la cual podrá disponer de los mismos de conformidad a su marco jurídico interno y legislación aplicable.

Manual de procedimientos del Departamento de Informática

El Departamento de Informática se encarga el mantenimiento general de los equipos de cómputo y periféricos (MINSAL, 2011a), en las diferentes instalaciones de las unidades de salud en el oriente del país. Esto con el propósito de garantizar el buen funcionamiento los dispositivos en cuestión.

Entre los procesos que realizan este Departamento se encuentran:

- Mantenimiento preventivo.
- Mantenimiento correctivo: diagnóstico de fallas y eventual reparación.
- Instalación de nuevo equipo de cómputo y periféricos.
- Evaluación y descarga del inventario de equipo de cómputo y periféricos dañados u obsoletos.
- Administración de software (instalación, actualización, desinstalación).

- Mantenimiento de las medidas de seguridad contra malware.
- Capacitación.

Mantenimiento preventivo. El mantenimiento preventivo se debe realizar al menos una vez al año con el fin de que los dispositivos se encuentren en óptimas condiciones para su funcionamiento. Deberán ser oportunamente registradas en la plataforma informatizada de soporte, calendarizada y comunicada con al menos dos semanas de antelación.

Mantenimiento correctivo. En caso de una falla en un equipo de cómputo o periférico, deberá ser reportada a la instancia correspondiente. Todas estas actividades de mantenimiento correctivo deberán ser actualizadas en la plataforma informatizada de soporte incluyendo los costos de partes, herramientas y materiales necesarios para realizar la reparación, instalación de nuevos equipos de cómputo y periféricos.

Los técnicos informáticos serán los encargados de instalar, configurar, actualizar o reinstalar el o los equipos informáticos el cual deberá quedar configurado para la realización del inventario automatizado de software y Hardware.

Administración de software. Para la instalación de aplicaciones en los equipos informáticos, deberá ser coordinada por medio de una solicitud al personal técnico respectivo, quienes poseen los privilegios de administración.

Mantenimiento de las medidas de seguridad contra malware. Los equipos desde su instalación deberán quedar configurados para sus actualizaciones periódicas, en el caso de ser necesario, se actualizarán de manera manual el sistema operativo, software de usuario o herramientas de seguridad como antivirus.

Capacitación. Las capacitaciones estarán bajo la supervisión de Formación de la Unidad de Soporte de la DTIC.

Metodología de investigación

Tipo de investigación

Investigación descriptiva

La investigación será de tipo descriptiva. La investigación descriptiva busca definir las características, propiedades y perfiles de grupos de personas, comunidades, organizaciones, procesos, objetos o cualquier fenómeno que se someta al análisis (Sampieri y col., 2014, p. 92).

Los estudios descriptivos se enfocan en detallar las propiedades, características y perfil de seres vivos, grupos, procesos objetos, situaciones, fenómenos entre otros. Es decir, su objetivo es medir o recoger información de manera individual o conjunta sobre los conceptos o variables en que se aplica, sin tomar en cuenta la relación que existe entre estas.

En esta clase de estudios, el investigador es capaz de visualizar qué se medirá, por ejemplo, tamaño de las unidades de salud, cantidad de unidades prioritarias, tiempo de respuesta, calidad de asistencia, etc. Y así mismo, sobre quienes o qué se investigará, para este caso, a los trabajadores de la DRSO y las unidades de la región oriental.

Para Tamayo y Tamayo (2003, p. 46), este tipo de investigación es útil para mostrar con precisión las dimensiones del objeto de estudio en cuestión. Él sostiene que el enfoque es describir como una persona, objeto o situación se conduce o funciona en el presente, trabajando sobre realidades de hecho y exponiendo una interpretación correcta.

Investigación tecnológica

La investigación tecnológica es experimental, y tiene por finalidad la invención de herramientas o procesos con el fin de obtener un beneficio económico (Sánchez, 2004, p. 50). Este tipo de investigación se realiza esencialmente en la empresa, y es desarrollada por técnicos especialistas en el área de interés.

Bajo esta perspectiva, la presente investigación realizará un estudio de la situación actual del Departamento de Informática de la Dirección Regional de Salud Oriental,

detallando sus procesos y procedimientos a seguir, para el desarrollo de la herramienta que cumpla con las características necesarias y dé solución al problema interno de la Unidad.

Universo

El universo de la investigación estará conformado por todo el personal que en sus labores diarias utilicen un equipo informático en la Dirección Regional de Salud Oriental; dado que el sistema está orientado a facilitar la gestión y administración del equipo informático para garantizar su perfecto funcionamiento.

Dicha población tiene un tamaño aproximado de 2,000 trabajadores¹⁰ en función de todas las unidades de salud a nivel oriental y los diferentes departamentos que conforman la DRSO.

Tipo de muestreo

Muestreo no probabilístico

Para Sampieri y col. (2014, p. 386), la muestra no probabilística, es un tipo de selección orientado por las características propuestas por el investigador, más que por un criterio generalizado. La ventaja de esta es que se puede estudiar una cuidadosa cantidad de casos de ciertas características según el problema a estudiar.

Método por conveniencia

Estas muestras están formadas por los casos disponibles a los que tenemos acceso (Sampieri y col., 2014, p. 390). Esto significa que el muestreo por conveniencia es una técnica donde se selecciona una muestra poblacional por el hecho de que sea accesible.

El método por conveniencia se centra en la proximidad y accesibilidad entre los sujetos para el investigador. La muestra se compone de aquellos que sean más convenientes. Se selecciona a los individuos más cercanos para participar (Cohen y col., 2003).

¹⁰ Fuente: Ing. Juan Soto, jefe del Departamento de Informática.

Muestra

Tomando en cuenta las características del muestreo por conveniencia, se tomó a bien realizar la entrevista al personal que labora en el Departamento de Informática de la Dirección Regional de Salud Oriental.

El área está conformada por cuatro personas: el jefe de dicho departamento y tres técnicos informáticos, quienes son los encargados de velar por el buen funcionamiento de toda el área informática en todas las unidades de salud a nivel oriental.

Por lo tanto, para cumplir los objetivos del presente trabajo, ellos son la parte de la población mas conveniente para realizar la entrevista y conseguir datos más acertados.

Técnicas e instrumentos para recolección de información

Una técnica para la recolección de datos es el instrumento dirigido a validar y hacer más eficiente la investigación. Existen diversos valores en la información donde unas son más confiables que otras. Dicha fiabilidad dependerá de la manera en cómo se ha llegado a ella.

La validez de la información está fuertemente ligada a la técnica utilizada para la adquisición de dicho conocimiento, la fuente, los resultados obtenidos y el análisis de estos. La primera etapa de recopilación de información se le conoce como “Trabajo de campo” en el cual se aplican las herramientas brindadas por el instrumento de recolección de datos seleccionado (Sabino, 1992).

Existen diferentes instrumentos para recolectar información. Por lo tanto, para el desarrollo de esta investigación se centrará en la entrevista.

La entrevista, según Torrecilla (2002), es la técnica con la cual el investigador pretende obtener información de una forma oral y personalizada. La información versará entorno a acontecimientos vividos y aspectos subjetivos de la persona tales como creencias, actitudes, opiniones o valores en relación con la situación que se está estudiando. En la entrevista se efectuarán preguntas a un experto en la materia enriqueciendo así la

información obtenida.

Para la presente investigación, la entrevista se aplicará a los expertos de la Dirección Regional de Salud Oriental seleccionados con el método de muestreo por conveniencia.

Procedimientos para la validación de instrumentos

Para la validación de instrumentos a utilizar, se realizarán los siguientes pasos:

- Entre todo el grupo se elaborará una lista de las posibles preguntas que se realizarán en la entrevista.
- Se hará un filtro a las preguntas previamente elaboradas, seleccionando las de mayor interés, que proporcionen un mejor resultado acorde al objetivo de la entrevista.
- Se presentará el formato de entrevista al docente asesor quien verificará si las preguntas están orientadas con del objetivo de la investigación.
- El grupo aplicará cambios a las preguntas seleccionadas sugeridos por el docente asesor si así fuese necesario.
- Una vez la entrevista se encuentre aceptada por el docente asesor se pasará a la ejecución.

Procedimientos para la recolección de datos

Para la recolección de los datos y debido a una de las principales limitantes como es la pandemia por COVID-19, se realizará una entrevista, a través del llenado de un formulario online. Más adelante se va a usar un programa de ofimática para la respectiva tabulación y exhibición de los datos. La entrevista estará dirigida al personal del Departamento de Informática de la DRSO. Para lo mencionado anteriormente se realizarán los siguientes pasos:

- Se solicitará permiso al jefe de departamento para poder realizar la entrevista al personal elegido.

- Se van a dar indicaciones acerca de cómo responder las preguntas, además se les comunicará el objetivo de las mismas.

Después se va a realizar el respectivo examen e interpretación de los datos los cuales proporcionarán información sustancial para las funcionalidades del sistema. De esta forma se obtendrá una aproximación de la aceptación que vaya tener con los usuarios administradores de la misma.

Procedimientos para el tratamiento de los datos

Una vez realizada la entrevista a cada uno de los expertos, se utilizará el análisis por pregunta y posterior, el respectivo diagrama FODA como resultado final.

Este se realizará en un documento de Word, donde se escribirá la pregunta con las respectivas respuestas, y se realizará el análisis respectivo para elaborar el resultado de cada ítem.

Al final, se incluirán los resultados en el diagrama FODA para la elaboración de conclusiones acerca del estado actual del Departamento de Informática de la DRSO.

Procedimientos para presentar e interpretar los datos

En este proceso es donde ordenamos, interpretamos y presentamos los resultados obtenidos de las entrevistas en un formato fácil de leer e interpretar.

La información se presentará de la siguiente manera:

- Se presentará cada pregunta de la entrevista.
- Se añadirá la interpretación de los datos obtenidos.

Análisis e interpretación de resultados

Análisis de entrevista

1. Enumere los siguientes procesos del 1 al 5 donde 5 es el que tiene más importancia dentro de su trabajo en el Departamento de Informática:

- Soporte técnico a usuarios.
- Capacitaciones.
- Mantenimiento de redes informáticas.
- Otro.

Análisis. Esta pregunta tiene como objetivo indagar el nivel de importancia que le dan a cada una de las tareas que les corresponden realizar al personal del Departamento de Informática.

Según los expertos, le brindan mayor importancia cuando reciben peticiones que tienen que ver con mantenimiento de equipos y servidores, teniendo un total de 3 votos para el puesto #5.

Para el puesto #4, los ingenieros mencionaron que las capacitaciones de nuevas tecnologías es muy importante dentro de sus obligaciones. Igualmente, para los puestos #3, #2 y #1 dijeron que eran las peticiones de red, el soporte a los usuarios en general y los enlaces a intranet respectivamente.

2. Enumere los siguientes problemas del 1 al 5 donde 5 es el que representa a los usuarios con más frecuencia:

- Conexión de red.
- Funcionamiento de una PC.
- Funcionamiento de un scanner/impresor.
- Funcionamiento de aplicaciones móvil o web.
- Otros.

Análisis. Esta pregunta tiene como objetivo conocer la frecuencia en que llegan las peticiones al departamento. De esta manera, se sabrá cual tipo de petición es la más solicitada en dicha área.

Según los expertos, reciben con mayor frecuencia peticiones que tienen que ver con mantenimiento de equipos y servidores, teniendo un total de 3 votos para el puesto #5.

Para el puesto #4, los ingenieros mencionaron que reciben peticiones seguido acerca de la aplicación móvil que usan en los ECOS Familiares. Igualmente, para los puestos #3, #2 y #1 coincidieron que se realizan servicios por conexión de red, impresoras o escáneres y mal funcionamiento de PC.

3. ¿Cada cuánto tiempo le realizan el mantenimiento a los equipos informáticos a cargo del Departamento de Informática?

- Cuando el usuario solicita mantenimiento.
- Cada 6 meses.
- Cada año.

Análisis. La pregunta tiene como objetivo saber con que regularidad se le da mantenimiento al equipo del Ministerio de Salud. Todos los ingenieros coincidieron que se le da mantenimiento al equipo cada vez que el usuario realiza una petición.

4. ¿Se lleva registro digital del mantenimiento preventivo y correctivo del equipo tecnológico?

- Si.
- No.

¿Cómo lo realiza actualmente?

Análisis. Con esta pregunta se pretende descubrir si en el Departamento de Informática llevan algún tipo de control en sus servicios de mantenimiento. Los ingenieros concuerdan que si se lleva un registro, el cual se realiza en hojas de calculo de Libre Office y en hojas impresas llamadas por ellos Hojas de Trabajo.

Los expertos comentaron que las Hojas de Trabajo son obligatorias para cada equipo que le realizan mantenimiento. Esta deberá ir firmada y sellada por el encargado del área a la que pertenece la computadora.

5. ¿El Departamento de Informática cuenta con un control de todo el equipo a su cargo?
- Si.
 - No.

Si la respuesta es si, ¿de qué manera lo manejan actualmente?

Análisis. La pregunta pretende dar a conocer si utilizan algún tipo de registro o pequeño sistema para llevar el control de los insumos y equipo que utilizan para realizar el mantenimiento, además tareas de las que están encargados.

Los técnicos encargados mencionan que si llevan un control de los insumos, y lo realizan por medio de hojas de calculo de Libre Office. Además, comentaron que el área de Activos Fijos les llevan un control de las máquinas y dispositivos que suministra el Ministerio de Salud.

6. ¿Qué cargo desempeña? ¿Que reportes realiza con frecuencia o son más importantes para su cargo?

Análisis. Con esta pregunta se pretende conocer los cargos que existen dentro del Departamento de Informática, así como los reportes que tienen a cargo.

Curiosamente, los expertos mencionaron que eran técnicos, a excepción del jefe del departamento. Además, todos estuvieron de acuerdo que los reportes que más les urge a ambos cargos son las Hojas de Trabajo, y al jefe en especial, el reporte del Plan de Trabajo Mensual.

7. ¿Considera que existe la necesidad de utilizar un sistema informático para desempeñar mejor su labor?

- Si.
- No.

¿Por qué?

Análisis. El objetivo de esta pregunta es si los expertos reconocen la importancia de un sistema informático para la administración del departamento.

El personal del área de informática expresó que si es importante un sistema de administración, reconociendo que se pueden controlar mejor los recursos, permitiría sistematizar las solicitudes de los usuarios y llevar un mejor control de las actividades que se realizan.

8. ¿Qué factores o circunstancias le impedirían realizar su trabajo? (Por ejemplo, que el encargado del área no se encuentre).

Análisis. La pregunta pretende explorar las posibilidades de no cumplimiento del trabajo por situaciones ajenas al departamento.

Los expertos expresaron que el mayor impedimento para realizar su trabajo es la toma de decisiones. Los jefes de las áreas mandan a mantenimiento a los equipos muy tarde, por lo que terminan dañándose y no pueden ser reparados.

Otra de las razones es que el encargado de la Unidad o Área no se encuentre en el lugar para firmar y sellar la hoja de trabajo. Y luego, que la Unidad de Salud no posea conexión a energía eléctrica en ese momento, que sucede más frecuentemente en invierno.

9. ¿Qué procesos aparte del control de equipo informático y de solicitudes de soporte le sería útil que incluyera el sistema informático?

Análisis. Con esta pregunta se pretende conocer las necesidades que ellos sienten que poseen en el área administrativa del Departamento de Informática.

Los ingenieros expresaron que les gustaría tener un inventario de materiales informáticos, como por ejemplo, para llevar el control de los tóner y equipo que necesitan para cambios y reparaciones.

Otra de las características que les gustaría que incluyeran es las notificaciones de la realización del servicio al usuario que realiza la petición.

Diagnóstico

Hallazgos

Hallazgos de observación

La observación, según la autora Sanjuán (2010), es una adquisición activa de información por medio del sentido de la vista, en ella se apoya el investigador para obtener un mayor número de datos, gran parte del acervo de conocimientos que constituye la ciencia ha sido lograda mediante la observación por lo tanto es fundamental para el proceso de investigación.

Existen diferentes tipos de observación de las cuales se puso en práctica la observación participante. Esta consiste en que para obtener los datos, el investigador se incluye en el grupo, hecho o fenómeno observado para conseguir la información desde adentro.

Sus características son:

- La observación participante es una de las técnicas privilegiadas por la investigación cualitativa.
- Es la observación del contexto desde la participación del propio investigador no encubierta y no estructurada.
- Suele prolongarse en el tiempo y no se realiza desde la realización de matrices o códigos estructurados previamente, sino más bien desde la inmersión del contexto.
- Este tipo de observación proporciona descripciones de los acontecimientos, las personas y las interacciones que se observan, pero también, la vivienda, la experiencia y la sensación de la propia persona que observa.

En el tiempo que se visitó a los profesionales de la Dirección Regional de Salud Oriental, fuimos partícipes de los procedimientos que se realizan en el Departamento de

Informática, dándonos a conocer los métodos que implementan para la gestión y control del mantenimiento del equipo informático.

En una primera instancia se pudo observar que el departamento maneja un control a base de un inventario de hojas de cálculo de Libre Office en una laptop, en la que se lleva el registro de los mantenimientos, debido a esto no siempre se actualiza la información de los equipos a los cuales se les ha dado el soporte o que se encuentran esperando el mantenimiento.

Esta situación lleva a una segunda observación, manejan un control manual de hojas volantes para la entrada y salida de los diferentes dispositivos lo cual es una práctica con la que no se lleva un registro exacto de todo el trabajo realizado por el departamento. Los integrantes del departamento también realizan rutas de soporte con lo que no siempre se encuentran dentro de las instalaciones.

Hallazgos en entrevistas

En el análisis realizado por medio de las entrevistas al personal del Departamento de Informática, se puede apreciar la necesidad de sistematizar las solicitudes de los usuarios para optimizar tiempo y recursos, así como también tener una herramienta que administre de una manera más centralizada todas las operaciones que realizan en dicho departamento.

Dentro del área cada integrante tiene una función en específico, pero esto no limita a que un integrante no pueda realizar el trabajo del otro; pero si se ve afectado en rendimiento al tener que recopilar los trabajos realizados en un periodo determinado de todo el departamento para la generación de diferentes reportes.

Problemas generados por el sistema actual

Actualmente el departamento divide su procesamiento de datos entre un registro de hoja de cálculo, reportes digitales, petición de soporte por correo o llamada.

Como se puede apreciar el departamento carece de un sistema que centralice todas las operaciones, generando esto una un déficit en la exactitud de los reportes, trabajos

pendientes, trabajos realizados, equipo con prioridad y coordinación de rutas.

Entre los problemas más comunes se encuentra la falta de comunicación entre los usuarios y el equipo técnico, ya que no se posee un medio confiable que sirva como puente para gestionar una revisión o saber si su equipo ya fue procesado.

Algunos usuario utilizan un método incluso de menos rendimiento que es la desplazarse hasta el departamento para la petición de soporte, en algunos casos los usuarios esperan todo el día para hacer la entrega del dispositivo a algún técnico. Por no tener un medio de certeza de cuándo estará reparado, los usuarios llaman regularmente o pasan por el departamento preguntando por su equipo.

En varias ocasiones los integrantes del departamento no se encuentran disponibles, ya que se encuentran en misión fuera de la Dirección Regional de Salud Oriental atendiendo otras peticiones de soporte.

Estas son algunos de los problemas que el sistema actual ha generado, afectando el rendimiento del equipo de informática teniendo en cuenta la cantidad de integrantes y el alta demanda que se poseen.

Procesos actuales

Proceso actual de una petición

Normalmente el Departamento de Informática de la DRSO suele recibir las peticiones de servicios por medios convencionales, correos, peticiones físicas, llamadas telefónicas o solicitarlo físicamente en el departamento, sin embargo no hay un medio definido o sistema informático estándar para hacerlas.

Proceso de respuesta a una petición

Cada petición tiene una forma diferente en la que se opera. Las peticiones por escrito en su mayoría las trae alguien del personal de la unidad de donde proviene junto con el equipo con fallas. Este tipo de petición se recibe y al solventarse el problema se le

llama a el encargado de dicha unidad para que pueda recoger el equipo.

Las peticiones telefónicas pueden ser tanto del personal de las diferentes áreas de la DRSO como también de unidades de salud que necesitan que se les agende una visita; si es del personal dentro de la DRSO se procede a atender a la brevedad posible, de igual manera las que se agendan se establece un día de visita y se elaboran rutas para dar soporte a las unidades cercanas a cargo del departamento, esto se hace así para realizar el trabajo lo más eficiente posible. Los correos electrónicos en su mayoría son de áreas externas a la DRSO pero que siempre atiende el departamento.

Análisis FODA

La matriz de análisis FODA es una herramienta conocida por brindar información actual de una organización (Espinosa, 2013). Este tipo de herramienta identifica tanto las oportunidades como amenazas que posee el departamento.

En el análisis externo, tienen que ver aspectos relacionados con la situación fuera del Departamento de Informática, por ejemplo, tecnologías, tipos de usuarios, economía, etc. Y el análisis interno, se asocia con factores claves del departamento como lo son personal, capacitaciones, recursos disponibles etc.

En el Cuadro 5 se presenta la matriz de análisis FODA correspondiente a los procesos actuales del Departamento de Informática de la Dirección Regional de Salud Oriental.

Análisis FODA del Departamento de Informática	
Fortalezas	Debilidades
<p>- Cuentan con material, herramientas y equipo para realizar su trabajo.</p> <p>- Cuentan con personal especializado en el área de informática.</p> <p>Análisis interno - Todo el personal conoce todas las tareas que se realizan en el Departamento.</p> <p>- Poseen rápida respuesta en casos de emergencia.</p> <p>- Poseen auto propio del Departamento.</p>	<p>- No existe una buena comunicación de tareas realizadas o datos relacionados con ellas.</p> <p>- No poseen todos los registros de materiales y equipo que usan para los mantenimientos o reparaciones.</p> <p>- No poseen una agenda debidamente ordenada para el registro de solicitudes.</p> <p>- Poco tiempo disponible para la administración y papeleo con el sistema de administración actual.</p> <p>- La calendarización de visitas falla por la falta de organización de las mismas.</p>

Cuadro 4
Análisis FODA: Fortalezas y debilidades del Departamento de Informática.

Análisis FODA del Departamento de Informática	
Oportunidades	Amenazas
<p>- El personal se puede desplazar de la DRSO a la unidad donde se requiera atención.</p> <p>- Con el uso de un sistema de administración de solicitudes, el Departamento garantiza el buen rendimiento de los recursos informáticos, además de la pronta respuesta a peticiones y agilidad de realización de tareas administrativas.</p>	<p>- El encargado del equipo no se encuentra en la unidad.</p> <p>- El personal no se encuentra en el Departamento.</p> <p>- Información errónea sobre el equipo para reparación y mantenimiento.</p> <p>- Constantes cambios en el director de la DRSO.</p>
<p>Análisis externo</p>	

<ul style="list-style-type: none"> - El sistema será escalable, para 	<ul style="list-style-type: none"> - Constantes cambios en tecnologías a
<p>posteriormente actualizar los módulos o escalarlos según se necesite.</p> <ul style="list-style-type: none"> - La aplicación móvil permitirá informarles a los usuarios que días se realizará el mantenimiento, además de poder consultar el estado de servicio que se les realiza, disponibilidad del personal en el departamento y realización de peticiones de servicio. 	<p>implementar para los usuarios de las unidades de salud.</p> <ul style="list-style-type: none"> - Usuarios desconocen el estado del servicio que solicitan. - Realización de tareas muy lenta. - Peticiones de servicio se pierden por falta de administración. - Se posponen tareas de administración por mucho tiempo.
<p>Análisis externo</p>	

Cuadro 5

Análisis FODA: Oportunidades y amenazas del Departamento de Informática.

Enfoque del sistema propuesto

Debido a las dificultades con las que cuenta el Departamento de Informática en el área administrativa, tomando como base las debilidades y amenazas planteadas en el análisis FODA se toma a bien lo siguiente:

- El sistema deberá contar con un módulo que permita administrar las tareas realizadas, donde también se visualicen los datos generales de cada una; facilitando así al equipo informático llevar un mejor control en sus solicitudes y evitar las confusiones o choques de tareas en el departamento.
- El sistema deberá contar con un módulo de inventario de materiales, equipo e insumos a utilizar en mantenimientos.
- El sistema deberá contar con un módulo de calendarización para las solicitudes que recibe el Departamento de Informática desde las unidades de salud fuera de la DRSO.
- El sistema deberá contar con el módulo de generación de reportes en general para optimizar el tiempo laboral del Departamento de Informática.
- El sistema deberá contar con notificaciones hacia el Departamento de Informática sobre los mantenimientos, tareas o reparaciones previas a la fecha actual, para evitar pasar por alto dichas actividades en una fecha específica.
- La aplicación móvil deberá contar con notificaciones dirigidas hacia los jefes de unidad o encargados de equipos que soliciten el servicio, para informarles fecha y estado de la solicitud y así garantizar que se encuentren en la unidad de salud correspondiente.
- Se deberá garantizar un manual corto de fácil aprendizaje para la adaptación rápida a la aplicación y capacitación en caso de cambios de personal en la unidad de salud y el Departamento de Informática.

Desarrollo del software

Diagrama causa-efecto

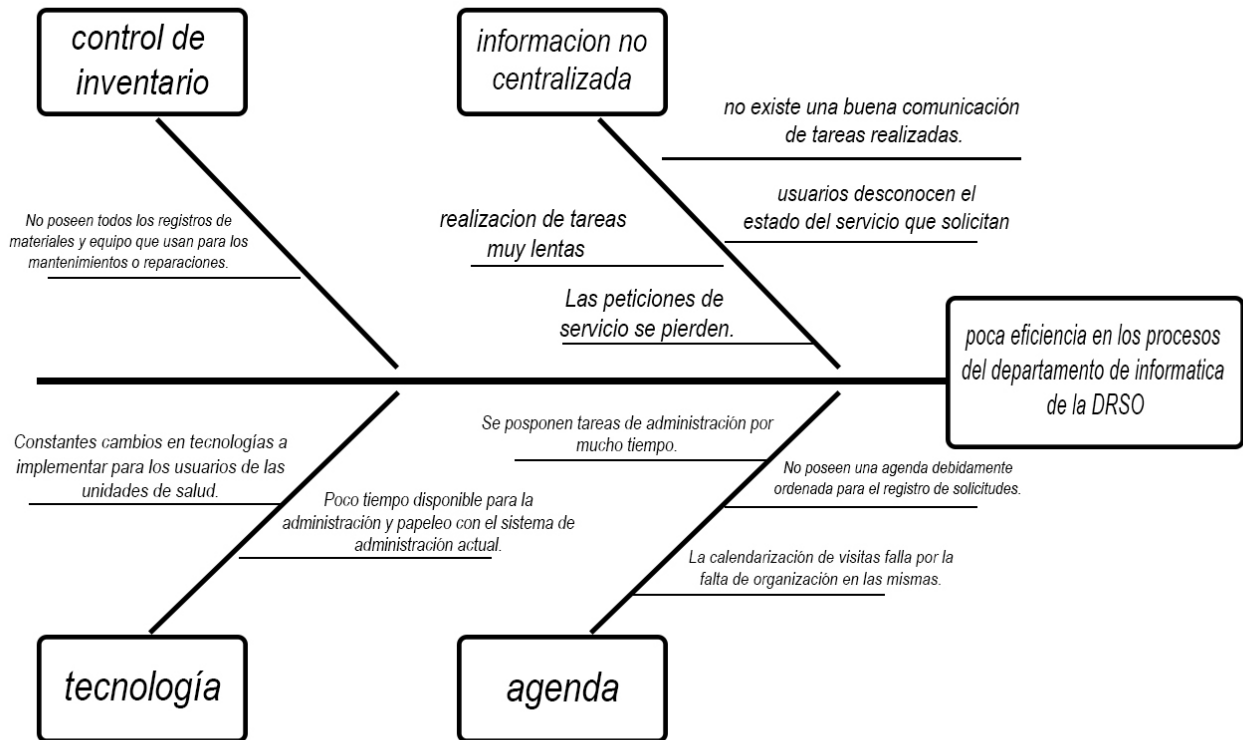


Figura 14

Diagrama causa efecto del Departamento de Informática.

El diagrama Causa-Efecto o diagrama de Espina de Pescado, se puede implementar cuando se realizan análisis en problemas de cualquier índole. Este diagrama ayuda a graficar las causas y problemas para analizarlas posteriormente (EcuRed, 2020j). Para este caso, es utilizado para describir los orígenes y consecuencias de los problemas del Departamento de Informática detallado en la Figura 14 y su descripción en el Cuadro 6.

Situación	Solución
Procesos	
Procesamiento control de inventario:	
<ul style="list-style-type: none"> - El inventario no se actualiza en el momento indicado. - Pérdida de información. - Cantidades erróneas en el inventario. - Retraso en saber la existencia de repuestos para el equipo informático. 	<p>Implementación de un módulo de inventarios que permita llevar el control de insumos de una forma eficiente, mostrando las cantidades disponibles y los movimientos de los mismos.</p>
Procesamiento manual de las solicitudes de servicio:	
<ul style="list-style-type: none"> - Proceso de atención a solicitudes tardío. - Órdenes de servicio no registradas. - Solicitudes de servicio perdidas. - Procesos manuales para el control de solicitudes. 	<p>Para solventar esta situación, se desarrollará un módulo para poder gestionar las solicitudes de soporte técnico. Este módulo llevará un control sobre el estado en las que se encuentren las solicitudes.</p>
Proceso de calendarización de solicitudes de servicio técnico:	
<ul style="list-style-type: none"> - Se posponen tareas administrativas durante mucho tiempo. - Chocan horarios con las visitas técnicas. - Agenda desordenada. 	<p>Para solventar esta problemática, se implementará un módulo para agendar visitas técnicas, por las solicitudes fuera de la DRSO que lo requieran.</p>

Proceso de entrega de equipos con solicitudes resueltas:	Se implementará una aplicación móvil para que los usuarios estén pendiente de su equipo. Podrán ver el estado de su solicitud, realizar peticiones y ver el momento de finalización de las mismas.
- Los usuarios llegan a preguntar por su equipo.	
No se le puede entregar si la ficha de trabajo está llena.	
- No se les puede entregar si no están presentes los técnicos de informática.	

Cuadro 6

Análisis de situación-solución.

Requerimientos de información

Para el desarrollo del sistema de solicitudes para el Departamento de Informática de la DRSO, fue de vital importancia conocer las necesidades que tenían los técnicos y el jefe del área; con el fin de proporcionarles la herramienta adecuada que les sirva de apoyo y les permita agilizar las tareas diarias que realizan.

Por lo tanto, la herramienta tecnológica tanto en su versión web como aplicación móvil contará con una serie de requerimientos que se detallan en los siguientes apartados.

Aplicación web

Requerimientos funcionales. Los requerimientos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de manera en que este debe reaccionar a entradas particulares.

Los requerimientos funcionales para el sistema que automatice los procesos de solicitud y control de servicios técnicos en el Departamento de Informática de la Dirección Regional de Salud Oriental son los siguientes:

- El usuario administrador podrá ver la lista de solicitudes de servicio técnico.
- El usuario administrador podrá generar reportes varios en formato .pdf y .odf.
- Los usuarios podrán crear peticiones de servicio técnico desde su smartphone.
- Los usuarios podrán consultar el estado de su petición.
- El sistema contara con roles de usuarios, para dar permisos específicos de accesos a usuarios específicos.
- El sistema contara con control de acceso por medio de nombre de usuario y contraseña.
- El sistema contara con control de inventario de insumos de trabajo.
- El sistema contara con control de inventario y ubicaciones de equipos informáticos.
- El sistema contara con calendarización de peticiones de servicio.

Además, la DTIC tiene tecnologías definidas para este tipo de aplicaciones, las cuales se seguirán para el desarrollo bajo sus estándares. Estas son:

- Lenguaje de programación: PHP versión 5.6
- Framework de desarrollo: Symfony versión 2.8
- Gestor de base de datos: PostgreSQL versión 9.6

Requerimientos no funcionales. Los requerimientos no funcionales para el sistema que automatice los procesos de solicitud y control de servicios técnicos en el Departamento de Informática de la Dirección Regional de Salud Oriental son los siguientes:

- El sistema podrá ser utilizado en cualquier sistema operativo.
- El sistema funcionara en cualquier navegador web.

- El sistema contara con una interfaz de rápido aprendizaje.
- El sistema tendrá botón de cerrar sesión de usuario.

Aplicación móvil

Requerimientos funcionales. Son declaraciones de los servicios que debe proporcionar la aplicación móvil, de la manera en que este debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares.

Los requerimientos funcionales para la aplicación móvil que brindará apoyo a los usuarios del Departamento de Informática son los siguientes:

- El usuario podrá generar peticiones desde su smartphone.
- El usuario podrá ver el estado de la petición desde la aplicación móvil.
- El usuario recibirá notificaciones en la aplicación móvil cuando su petición sea solventada.
- El usuario podrá ver todas las peticiones realizadas.
- El usuario podrá cambiar su contraseña de acceso mediante la aplicación móvil.

Además, se deja a criterio de los desarrolladores la tecnología a implementar para la producción de la aplicación móvil; Siguiendo ciertos estándares para su selección, los cuales son:

- Debe ser de licencia libre.
- Debe de tener documentación.
- Debe tener soporte.
- Debe ser escalable.

Con lo cual, Flutter, que es el framework de desarrollo que se utilizará, cumple con todos los criterios antes mencionados.

Requerimientos no funcionales. Los requerimientos no funcionales para la aplicación móvil son los siguientes:

- El usuario podrá cerrar sesión en la aplicación por medio de un botón.
- La aplicación móvil contara con una interfaz minimalista y amigable.
- La aplicación móvil será de fácil acceso.

Estudio de factibilidades

Para que un Proyecto tenga éxito, se debe realizar un estudio de factibilidad, el cual es uno de los resultados mas esperados y también importantes en la etapa de un proyecto. Para ello, dicho estudio se dividirá en 3 fases: técnico, económico y operativo.

Factibilidad técnica

Mediante esta factibilidad se establece si el sistema propuesto puede ponerse en marcha con los recursos técnicos que cuenta el equipo; esto se hace en base a la disponibilidad de los recursos existentes en términos de hardware, software y recurso humano, para así garantizar la existencia de la tecnología y el conocimiento necesario para que factible el desarrollo del proyecto.

La factibilidad técnica se puede diferenciar en tres grandes aspectos. El primero, será el hardware con el que cuenta la DTIC y el Ministerio de Salud donde será montado eventualmente el sistema informático. El segundo será revisar el uso del software necesario para que la implementación del sistema sea satisfactoria. Y por último, se evaluará el conocimiento tanto en el personal que labora en el Departamento de Informática, como del equipo desarrollador del proyecto.

El interés principal del Departamento de Informática de la DRSO es con el fin de llevar un mejor control de las peticiones de servicio y desean centralizar toda la información mediante un sistema informático.

Hardware. El equipo de informática con el que cuenta el Departamento de Informática de la DRSO y las distintas Unidades de Salud¹¹ actualmente se detalla en el Cuadro 7. El equipo de informática con el que cuenta el equipo de desarrollo se detalla en el Cuadro 8

Equipo	Especificaciones
Computadoras de escritorio en el Departamento de informática de la DRSO.	<p>Computadora HP de escritorio.</p> <ul style="list-style-type: none"> - Procesador Intel Core i5 4ta generación. - 8 GB de memoria RAM. - 1 TB de disco duro.
Computadoras de escritorio en las Unidades de Salud.	<p>Computadoras DELL escritorio.</p> <ul style="list-style-type: none"> - Procesador Intel Pentium. - 4 GB de memoria RAM. - 216 GB de disco duro.
Servidor	<p>DELL PowerEdge M830 servidor Familia Intel Xeon E5 2.7 GHz 16 GB DDR4 - SDRAM 37 TB hoja.</p> <ul style="list-style-type: none"> - Pantalla de 5". - Cámara trasera de 8 MP, cámara frontal de 5 MP.
Teléfono celular	<ul style="list-style-type: none"> - Batería extraíble de 2,000 mAh o superior. - Android 6.0 o superior. - Procesador de 1.4 GHz o superior. - 1 GB RAM, 8 GB ROM o superior.

Cuadro 7

Especificaciones del equipo del MINSAL.

¹¹ Fuente: Ing. Juan Soto, Jefe del Departamento de Informática.

Equipo	Especificaciones
Computadora de escritorio.	<p>Aeon.</p> <ul style="list-style-type: none"> - Procesador AMD Ryzen 7 tercera generación. - 16 GB de memoria RAM. - 1.5 TB de disco duro.
Computadora de escritorio.	<p>Aeon.</p> <ul style="list-style-type: none"> - Procesador AMD Ryzen 5 tercera generación. - 16 GB de memoria RAM. - 1 TB de disco duro.
Computadora laptop.	<p>Asus.</p> <ul style="list-style-type: none"> - Procesador AMD Ryzen 7 cuarta generación. - 24 GB de memoria RAM. - 512 GB de disco duro.
Computadora laptop.	<p>Acer.</p> <ul style="list-style-type: none"> - Procesador Intel Core i5 octava generación. - 8 GB de memoria RAM. - 1 TB de disco duro.
Teléfono celular.	<p>LG Aristo 2.</p> <ul style="list-style-type: none"> - Cámara trasera de 13 MP, frontal de 5 MP. - Batería extraíble de 2,410 mAh. - Android 7.1.2. - Procesador cuádruple núcleo de 1.4 GHz. - 2 GB RAM, 16 GB ROM.

Cuadro 8

Especificaciones del equipo de desarrollo.

Software. Para la correcta ejecución del sistema informático desarrollado, los equipos antes mencionados necesitaran un software para su funcionamiento.

El software para el correcto funcionamiento con el que deben contar los equipos se detalla en el Cuadro 9.

Máquina	Clasificación	Software
Máquina servidor	Sistema Operativo	Debian 10
	Gestor de bases de datos	PostgreSQL 9.6
	Servidor web	Symfony 2.8 + Apache 2 + PHP 5
Máquina cliente	Sistema Operativo	Windows o GNU Linux
	Paquete de ofimática	Libre Office
	Lector PDF	Adobe Reader, Foxit Reader entre otros.
	Navegador web	Mozilla Firefox o Google chrome

Cuadro 9

Especificaciones de software.

Para el correcto funcionamiento del sistema las maquinas deben de contar con un sistema operativo Windows o GNU/Linux, el sistema está desarrollado para entorno web, por lo cual deberá contar con un navegador web y también con un paquete de ofimática.

Por medio de un estudio de observación directa y consultas hacia el equipo técnico del Departamento de Informática de la DRSO se descubrió que el equipo informático en general cumplía con los requisitos de software necesarios para el perfecto funcionamiento del sistema.

El software con el que deben contar las maquinas del equipo de desarrollo se

especifica en el Cuadro 10.

Máquina	Clasificación	Software
Máquina servidor	Sistema Operativo	GNU/Linux
para desarrollo	Gestor de bases de	PostgreSQL 9.6
de pruebas	datos	
	Servidor web	Symfony 2.8 + Xampp 5.6+ Bootstrap 3
	Sistema Operativo	Windows o GNU Linux
Máquina para el	Paquete de ofimática	Libre Office
desarrollo de	Servidor web	Xampp 5.6 + Symfony 2.8 + Bootstrap 3
software	Diseño gráfico	Gimp
	Editor de código	Visual Studio Code
	Navegador web	Mozilla Firefox o Google chrome

Cuadro 10

Software del equipo de desarrollo.

Recurso humano. El Ministerio de Salud cuenta con personal especializado en el área de informática, por lo que tareas como manejar un sistema informático puede resultar hasta cierto punto sencillo para ellos. En general, el personal del Departamento de Informática deberá contar con conocimientos básicos de informática, por lo que este requisito es cumplido por ellos.

Los conocimientos con los que debe contar el equipo de desarrollo son los siguientes:

- Conocimientos avanzados del lenguaje de programación PHP 5.
- Conocimientos y manejo del framework Symfony 2.8.

- Conocimiento de HTML 5, Javascript, CSS 3 y Bootstrap 3.
- Manejo del gestor de base de datos PostgreSQL 9.6.

El recurso humano para el desarrollo de este proyecto posee los conocimientos antes mencionados y estará conformado por cuatro estudiantes egresados de la carrera de Ingeniería de Sistemas Informáticos de la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador.

Luego de un estudio realizado, se logró determinar que el desarrollo del sistema informático es factible, debido a que el equipo desarrollador posee tanto el software, el hardware y los conocimientos necesarios para el desarrollo del proyecto.

Factibilidad operativa

La factibilidad operativa depende de los recursos humanos disponibles para el proyecto e involucra planear si el sistema operará y si será utilizado una vez instalado.

Volviendo un poco atrás, según las Cuadros 7 y 9, se afirma que el Ministerio de Salud cuenta efectivamente con el equipo donde será alojado el sistema informático eventualmente; contando con espacio para datos y metadatos del sistema por cinco años, según las estimaciones de los autores Kendall y Kendall (1997), que será la vida útil de la misma.

Además, mediante entrevistas por parte del equipo técnico del Departamento de Informática de la DRSO y observación de su trabajo, se determinó que el proyecto será factible operativamente y que este vendrá a ayudar en gran manera al Departamento como también a los usuarios finales en las Unidades de Salud; ya que la Unidad de Informática carece de un sistema de control de solicitudes y no cuentan con un método eficaz para el manejo de la información centralizada por el momento.

También se pudo observar que el personal quien estará a cargo del manejo del sistema cuenta con la capacidad intelectual suficiente para no tener ningún inconveniente

en el momento de su ejecución, ya que el manejo de un ordenador y navegar por su interfaz como el acceso a navegadores web no se les dificulta en lo más mínimo.

En la Dirección Regional de Salud Oriental, el Departamento de Informática es el que se encarga de soporte y mantenimiento a equipo y software en general, por lo tanto la institución ya cuenta con el personal adecuado para darle mantenimiento al sistema informático.

Concluyendo con este apartado tenemos la certeza que el proyecto es factible operativamente; ya que el recurso humano cuenta con los conocimientos informáticos necesarios para el manejo del sistema y que también están totalmente de acuerdo para la realización de este, ya que vendrá a facilitarles el manejo de las solicitudes de servicio técnico y les permitirá poder tener toda la información de manera centralizada.

Factibilidad económica

El termino factibilidad hace referencia a la disponibilidad de los recursos necesarios para llevar a cabo los objetivos o metas señaladas. El estudio de factibilidad es una tarea que normalmente esta organizada y realizada por los analistas de sistemas.

Este estudio consume entre un 5 % y un 10 % del costo estimado total del proyecto, el periodo de la elaboración varía dependiendo del tamaño y tipo de sistema que se desarrollará.

La factibilidad económica estudia la relación costo beneficio que conlleva el desarrollo del proyecto. Con este estudio se desea saber si la alternativa de desarrollo junto con las metodologías que se han elegido son las adecuadas y si con sus beneficios satisfacen los costos que se han tenido.

En este proyecto y ya comentado en las factibilidades anteriores se ha especificado que la institución ya cuenta con el equipo necesario para el funcionamiento del sistema por lo que no se deberá recurrir a gastos para la adquisición de nuevo equipo.

Los costos de desarrollo del sistema se definen en las Cuadros 11, 12 y 13; a manera

de describir el proceso y los insumos que se necesitan para la realización de este, ya que la institución no se hará cargo de ningún costo.

Hardware					
Rubros	Cantidad	Costo Mensual (\$)	Meses	Costo Unitario (\$)	Total (\$)
Depreciación de hardware por uso	-	-	-	-	-
Software					
Rubros	Cantidad	Costo Mensual (\$)	Meses	Costo Unitario (\$)	Total (\$)
Libre Office	4	-	-	-	-
Visual Studio Code	4	-	-	-	-
PostgreSQL 9.6	4	-	-	-	-
Gimp	4	-	-	-	-
Symfony 2.8	4	-	-	-	-
Xampp 5.6	4	-	-	-	-
Flutter	4	-	-	-	-
Amortización del software (16.8 %)	-	-	-	-	0.00
Costos directos					
Rubros	Cantidad	Costo Mensual (\$)	Meses	Costo Unitario (\$)	Total (\$)

Internet por contrato 5 MB	4	25.99	10	-	1,039.60
Mano de obra	4	600.00	10	-	24,000.00
Total					25,039.60

Cuadro 11*Factibilidad económica del hardware y software.*

Recursos consumibles			
Detalle	Cantidad	Costo Unitario (\$)	Total (\$)
Papelería	100	0.05	5.00
Útiles varios	100	1.00	100.00
Impresiones	100	0.50	50.00
Total			155.00

Cuadro 12*Recursos consumibles.*

Otros gastos					
Rubro	Costo por reunión (\$)	Reuniones al mes	Meses	Número de personas	Costo total (\$)
Pasaje	2.00	1	10	4	80.00
Refrigerio	3.00	1	10	4	120.00
Total					200.00

Cuadro 13*Otros gastos.*

Costo total	
Rubro	Valor (\$)
Recursos tecnológicos	25,039.60
Recursos consumibles	155.00
Otros gastos	200.00
Sub-total	25,394.60
Imprevistos (15 %)	3,809.19
Costo total del proyecto	29,203.79

Cuadro 14

Costo total del proyecto.

Como se puede observar según el Cuadro 14, el costo del proyecto asciende a los \$ 29,203.79. El Ministerio de Salud podría invertir en consultores para que realicen este proyecto. En conclusión, esta cantidad representa un ahorro para el MINSAL, ya que el proyecto será realizado como parte del proceso de graduación de los desarrolladores.

Estándares de desarrollo***Tipos de usuarios***

El sistema de automatización de procesos de solicitud y control de servicios técnicos del Departamento de Informática, llamado de hoy en adelante SISTEC, contará con tres tipos de usuarios, los cuales son:

- *Administrador.* Este usuario que tiene acceso a todo el sistema. Será asignado al Jefe del Departamento de Informática. Entre los permisos que posee, destacan:
 - Acceso a todos los módulos del sistema.
 - Realizar los movimientos de insumos desde su propia vista.

- Crear nuevos usuarios de tipo Administrador, Técnico y Usuario Final.
 - Reestablecer la contraseña de los usuarios.
 - Realizar informes.
- *Técnico*. Este usuario es para ser asignado a los técnicos del Departamento de Informática. Entre sus permisos se encuentran:
- Crear usuarios de tipo Usuario Final.
 - Aceptar las solicitudes entrantes.
 - Crear/actualizar hojas de servicio.
 - Ver/agregar/actualizar visitas.
 - Ver/agregar/actualizar equipos.
 - Ver/agregar hojas de descarte.
- *Usuario final*. Este usuario es para las personas que reciben servicios de parte del Departamento de Informática. Ellos pueden:
- Ver el estado de sus peticiones.
 - Crear solicitudes en base a los equipos que se encuentran en su establecimiento.
 - Ver/actualizar/borrar solicitudes.
 - Solicitar una visita técnica.
 - Solicitar un cambio de contraseña.

Roles

Los roles¹² son funciones que alguien desempeña. Por lo tanto, es de gran importancia identificar los roles a desempeñar dentro del presente proyecto.

Los roles identificados para el proyecto SISTEC son:

¹² rol

- *Jefe de proyecto.* Responsable de la provisión del servicio de desarrollo o de mantenimiento de aplicaciones demandado por el cliente y de la calidad de los productos obtenidos.
- *Cliente.* Es la persona que define y acuerda con el jefe de proyecto la provisión del servicio de desarrollo o de mantenimiento de una aplicación. Se refiere al Jefe del Departamento de Informática.
- *Usuario final.* La persona que utiliza y explota el producto o aplicación obtenido como resultado del servicio de desarrollo. Se refiere a los trabajadores que reciben servicios de parte del Departamento de Informática y al jefe y técnicos de la misma.
- *Analista de sistemas.* Es el encargado del desarrollo de aplicaciones en lo que respecta a su diseño y obtención de los algoritmos.
- *Diseñador de bases de datos.* Es responsable de definir el diseño detallado de la base de datos, incluyendo tablas, índices, vistas, restricciones, desencadenantes, procedimientos almacenados y otras construcciones específicas de la base de datos necesarias para almacenar, recuperar y suprimir objetos persistentes.
- *Desarrollador web.* Responsable del desarrollo de un conjunto de artefactos que componen un módulo o subsistema. Se refiere al analista-programador asignado para la construcción de un módulo.
- *Desarrollador móvil.* Responsable del desarrollo de un conjunto de artefactos que componen una aplicación móvil.
- *Tester.* Responsable del diseño y ejecución de las pruebas. Se refiere al equipo de pruebas (responsable de pruebas e ingenieros de pruebas).

Perfiles para el desarrollo del sistema

Según las tecnologías a utilizar por los estándares de la DTIC, los perfiles de cada rol serán los siguientes:

- *Diseñador de bases de datos*
 - Conocimientos de bases de datos relacionales.
 - Amplio conocimiento de PostgreSQL.
 - Dominio del lenguaje de consultas PL/SQL.
 - Experiencia en Doctrine ORM.
 - Manejo de procedimientos almacenados.

- *Desarrollador web*
 - Experiencia en desarrollo web con Symfony v2.8.
 - Manejo del lenguaje PHP y JS.
 - Conocimientos en el área de desarrollo web.
 - Manejo de maquetación web con Twig, HTML y CSS.
 - Conocimientos del patrón de desarrollo MVC.
 - Experiencia desarrollando web API.

- *Desarrollador móvil*
 - Experiencia en desarrollo móvil con Flutter.
 - Conocimientos a cerca de web API.
 - Manejo de lenguaje de programación DART.
 - Conocimientos del patrón de desarrollo MVC.

Arquitectura de desarrollo: MVC

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC (Uniwebsidad, 2020b), que está formado por tres niveles:

- *El modelo*, que representa la información con la que trabaja la aplicación, o en otras palabras, su lógica de negocio.
- *La vista*, que transforma el modelo en una página web y permite al usuario interactuar con ella.
- *El controlador*, que se encarga de procesar las interacciones del usuario y realiza los cambios solicitados en el modelo o la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones.

Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como en un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original.

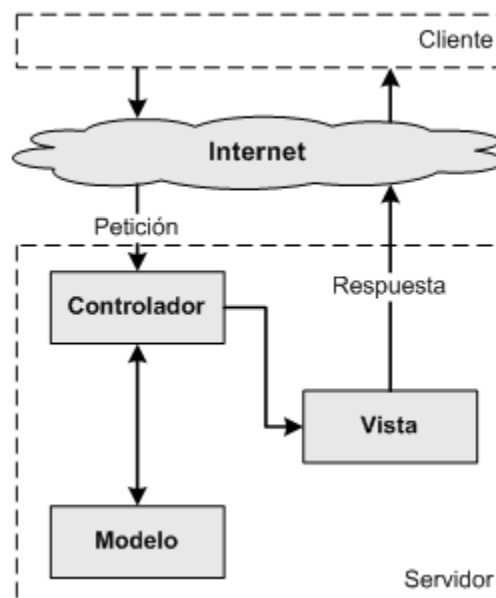


Figura 15

Funcionamiento del patrón MVC.

El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. Toda el funcionamiento se ilustra en la Figura 15.

API

Una interfaz de programación de aplicaciones o API (del inglés application programming interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción (EcuRed, 2020a). Son usados ampliamente para el desarrollo de aplicaciones móviles.

Tipos de web API. La API es un mecanismo para conectar dos software entre sí, para intercambiar mensajes o datos en forma estándar, como XML o JSON. Existen varios tipos de API (ApiService, 2020), entre las cuales están:

API de servicios web. Son las interfaces de desarrollo de aplicaciones que permiten el intercambio de información entre un servicio web (software que da acceso a un servicio concreto a través de una URL) y una aplicación. Normalmente ese intercambio se produce a través de peticiones HTTP. En la petición de la aplicación y respuesta, también en HTTP del servicio web, se contiene información de todo tipo tanto en los metadatos de la cabecera como en los del mensaje, normalmente en dos tipos de formatos muy usados: XML o JSON.

Dentro de las API de servicios web, los más usados son los siguientes:

- *SOAP*, del inglés *Simple Object Access Protocol*. Es un protocolo estándar de intercambio de información y datos en XML entre dos objetos.
- *XML-RPC*, es un protocolo de llamada a procedimiento remoto que usa XML como formato de datos y llamadas HTTP como sistema de comunicación.

- *JSON-RPC*, que es el mismo protocolo anterior pero en formato JSON.
- *REST*, de sus siglas en inglés *Representational State Transfer*. Es una arquitectura de software para sistemas hipermedia en la WWW. Este usa el protocolo HTTP.

El proyecto a desarrollar contará con un servicio de API web, usando el protocolo JSON-RPC, donde por medio de peticiones HTTP, se intercambiarán los datos de la base de datos alojada en el servidor con la aplicación móvil, respondiendo por medio de información en formato JSON.

Seguridad

Autenticación y autorización. Según los libros de Symfony, el sistema de seguridad de este framework se basa en identificar primero al usuario, que es a lo que se refiere con *autenticación*; y luego comprobarlo si tiene acceso al recurso solicitado, lo que se le conoce como *autorización* (Uniwebsidad, 2020a).

Autenticación. . Esta se activa cuando el usuario hace una petición a una URL que está protegida por un firewall. El trabajo del firewall consiste en determinar si el usuario necesita autenticarse, y por ende envía una respuesta al usuario para que lo haga.

El firewall se activa cuando la URL de una petición entrante concuerda con el valor de su opción de configuración rol. En la Figura 16, se ilustra el proceso de autenticación para un usuario anónimo, donde el firewall permite el acceso a los usuarios sin que estos estén autorizados, debido a la configuración *anonymous* del mismo. Esto significa, que para ciertas URL, es posible exigir o no la autenticación del usuario para su acceso.

Autorización. . Para los usuarios que solicitan una URL que necesita permisos específicos, la aplicación se comporta de manera diferente. Esto se puede configurar en la sección *acceso_control*, que indica que cualquier URL que coincida con una expresión regular, en este caso */admin/** y requiere el rol *ROLE_ADMIN*. Entonces, los roles son la clave del sistema de autorización: el usuario puede acceder a */admin/foo* sólo si cuenta con el rol *ROLE_ADMIN*.

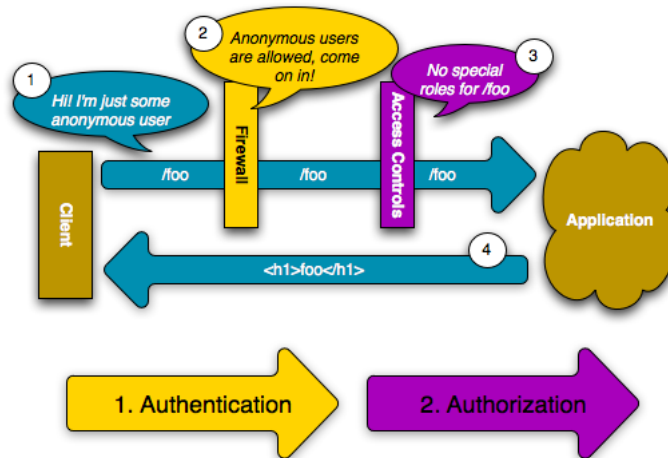


Figura 16
Proceso de autenticación anonymous en Symfony.

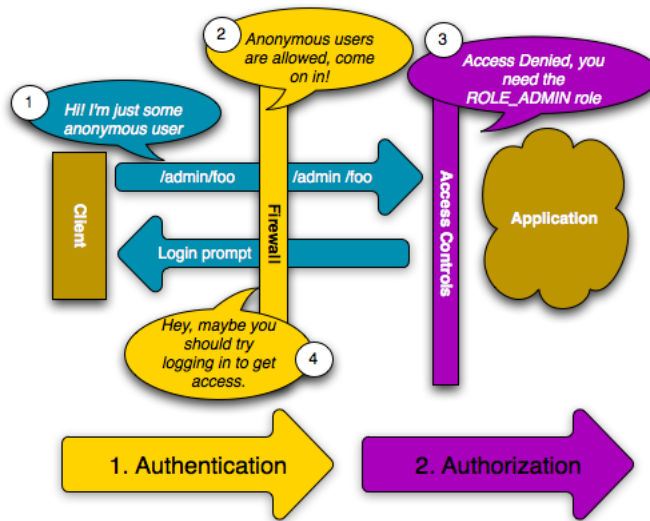


Figura 17
Proceso de autenticación denegada en Symfony.

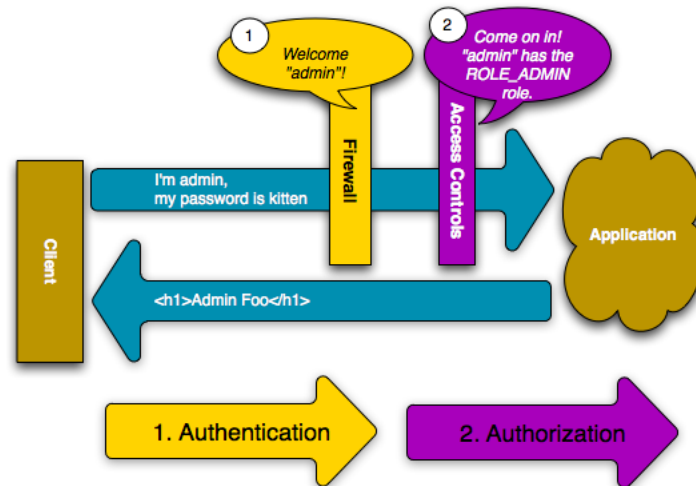


Figura 18

Proceso de autenticación de usuario en Symfony.

En la Figura 17, al igual que sucedía anteriormente, cuando el usuario realiza su petición, el firewall no solicita ningún tipo de identificación. Sin embargo, en cuanto la capa de control de acceso deniega el acceso al usuario (porque los usuarios anónimos no cuentan con el rol `ROLE_ADMIN`), el firewall toma el control de la aplicación e inicia el proceso de autenticación, redirigiéndolo al método o formulario de autenticación que se utilice.

Por último, si el usuario `admin` solicita `/admin/foo`, se lleva a cabo un proceso similar, excepto que ahora, después de haberse autenticado, la capa de control de acceso permitirá pasar a la petición, ilustrada en la Figura 18.

Encriptación. La encriptación es un proceso de codificación para que resulte indescifrable a cualquier persona que no conozca la clave para poder descifrarlo (EcuRed, 2020d). Por ello la encriptación es aquel proceso por el que la información es cifrada para que resulte ilegible a menos que conozcamos los medios para su interpretación.

La encriptación es clave en la seguridad de un sistema informático, para que su información se intercambie de manera segura con la aplicación, base de datos y aplicación móvil.

Algoritmo de encriptación. Un algoritmo de encriptación es básicamente una función matemática que contiene datos de entrada junto con una clave. Si la función se calcula con la clave correcta, se generan datos cifrados como formato de salida. Si el mensaje debe ser decodificado, debe ser devuelto a su forma original con una llave.

El algoritmo utilizado para la aplicación web sera **bcrypt**, que es una función de hash de contraseñas diseñada por Niels Provos y David Mazières, basada en el cifrado Blowfish y presentada en USENIX en 1999 (Codahale, 2020).

Además de incorporar una sal (texto añadido ya sea de forma dinámica o estática) para protegerse contra los ataques de Tablas arco iris, bcrypt es una función adaptativa: con el tiempo el recuento de iteraciones se puede aumentar para hacerlo más lento, por lo que sigue siendo resistente a los ataques de búsqueda de fuerza bruta incluso con un poder de cálculo creciente.

Este proyecto utilizará la encriptación de la siguiente forma:

- *Contraseña.* Las contraseñas se guardarán encriptadas en la base de datos, por lo que el login lo realizará directamente el controlador de seguridad de Symfony.
- *Formularios.* Los formularios propios de Symfony incluyen el cifrado de las variables e información de los formularios que se envían, por lo que es muy difícil la manipulación de ellos en la vista de la aplicación.
- *Token.* El token es una clave única que se utilizará para autenticar al usuario mediante la aplicación móvil. Esta se guardará en la base de datos ya encriptada.

Diseño del sistema informático

Diagramas de casos de uso del sistema web

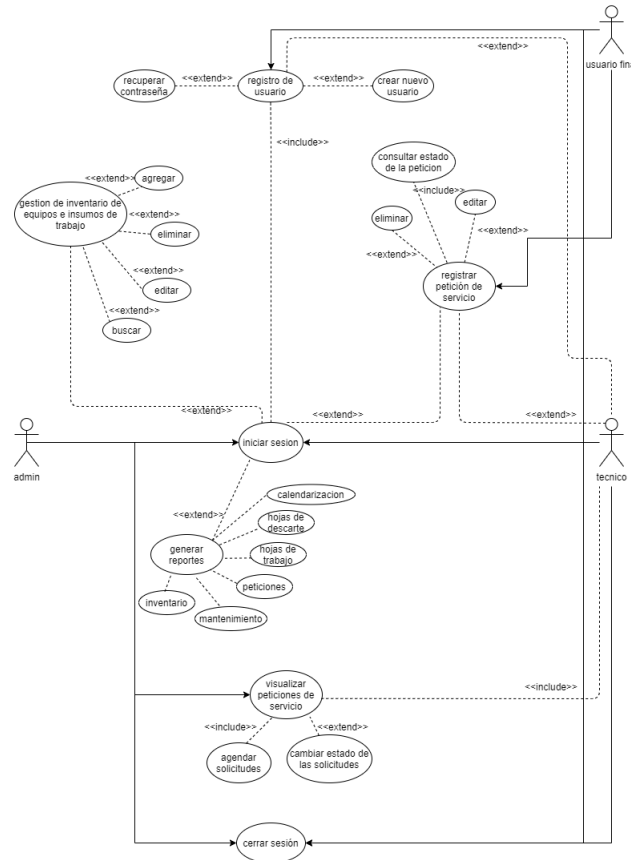


Figura 19
Caso de uso del sistema web.

Caso de uso	Iniciar sesión				
Prioridad	Alta.				
Actores	Administrador, Usuario Técnico, Usuario final				
Precondición	-				
Descripción	El sistema requiere iniciar sesión para acceder a sus funciones. En la pantalla se mostrará un formulario donde se solicitará las credenciales del usuario.				
	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>	Paso	Acción		
Paso	Acción				

Secuencia normal

	1	El usuario inicia el sistema mediante una petición al servidor.	
	2	El sistema solicita las credenciales al usuario.	
	3	El sistema mostrará la pantalla de inicio con las opciones de menú según sea el rol del usuario.	
Postcondición	Se ejecutará el inicio según el rol del usuario en que se mostrarán los procesos que este puede realizar.		
Excepciones	Paso	Acción	
	2	Si el sistema no encuentra las credenciales ingresadas al sistema.	
	2.1	El sistema notificará al usuario que sus credenciales son incorrectas y mostrará el formulario de inicio de sesión nuevamente, con la opción de recuperar contraseña.	

Cuadro 15

Caso de uso: Inicio de sesión.

Caso de uso	Consultar estado de petición	
Prioridad	Alta.	
Actores	Administrador, Usuario Técnico.	
Precondición	El usuario debe iniciar sesión en el sistema.	
Descripción	Le muestra a los usuarios Administrador y Técnico si hay nuevas peticiones dentro del sistema, además de poder verificar el estado en la que se encuentre en ese momento.	
	Paso	Acción

Secuencia normal

	1	El usuario selecciona dentro del menú del sistema la opción de peticiones.	
	2	El sistema muestra un listado de las peticiones con opciones para realizar diferentes procesos.	
	3	Se muestran los datos seleccionados por el usuario.	
Postcondición	Se ejecutarán los casos de uso siguientes según la elección del usuario: agendar, solicitudes, cambiar el estado de las solicitudes.		
Excepciones	Paso	Acción	
	2	Si el usuario no selecciona ninguna opción.	
		2.1	Se mostrarán todas las solicitudes ordenadas por prioridad y fecha.

Cuadro 16

Caso de uso: Consultar estado de petición.

Caso de uso	Gestión de inventarios
Prioridad	Alta.
Actores	Administrador.
Precondición	Debe haber seleccionado la opción de inventario en el menú del sistema.

Descripción	El módulo de inventarios contará con dos apartados, uno para equipo y otro para insumos. Al seleccionar uno de ellos se mostrará una tabla con todos los equipos e insumos que hay actualmente, clasificados por los tipos: herramientas, periféricos de reemplazo y hardware de mantenimiento. Además de poder agregar un nuevo insumo, editarlo o eliminarlo, así como también un equipo informático.	
Secuencia normal	Paso	Acción
	1	El usuario elige la opción inventario del menú donde se le mostrarán los dos apartados, equipos e insumos.
	2	Según la opción elegida, se le mostrará una tabla con todos los datos correspondientes.
3	Se muestran los datos solicitados por el usuario, con opción de agregar, editar o eliminar.	
Postcondición	Se ejecutarán los casos de uso siguientes según la elección del usuario: buscar, agregar, editar y eliminar.	
Excepciones	Paso	Acción
	2	Si el usuario no selecciona ninguna opción.
2.1	Se mantendrá en la pantalla de inicio del módulo.	

Cuadro 17*Caso de uso: Gestión de inventarios.*

Caso de uso	Cambiar estado de la petición de servicio	
Prioridad	Alta.	
Actores	Administrador, Usuario Técnico.	
Precondición	Debe haber iniciado sesión en el sistema y accedido al módulo de peticiones.	
Descripción	Permite ver la lista de peticiones registradas por los usuarios; permitiendo hacer cambios en ellas, como lo es cambiar el estado de la petición.	
Secuencia normal	Paso	Acción
	1	El usuario accede al módulo de peticiones del menú.
	2	El sistema mostrará la lista de peticiones registradas.
	3	Al seleccionar la petición, mostrará la información de ese registro.
	4	Habrán 2 botones, uno para eliminar petición y otro para cambiar estado.
	5	El botón cambiar estado permitirá que la petición cambie de “en proceso” a “finalizado”.
Postcondición	Se ejecutarán los casos de uso siguientes según la elección del usuario: cambiar estado de las solicitudes.	
Excepciones	Paso	Acción
	2	Si el usuario no selecciona ninguna opción.
	2.1	Se mantendrá en la pantalla de inicio del módulo.

Cuadro 18

Caso de uso: Cambiar estado de servicio

Caso de uso	Registrar una petición de servicio	
Prioridad	Alta.	
Actores	Jefe de Unidad de Salud o Departamento.	
Precondición	Debe haber iniciado sesión en el sistema.	
Descripción	El usuario podrá acceder al apartado de peticiones, donde podrá seleccionar previamente desde el menú donde podrá agregar una nueva petición de servicio técnico.	
Secuencia normal	Paso	Acción
	1	El usuario elige la opción peticiones desde el menú.
	2	Se le mostrará un botón donde le permitirá agregar una nueva petición, como también la lista de peticiones agregada previamente.
	3	Se solicitarán los datos para crear una nueva petición, los cuales serán tipo de petición y detalle de petición.
	4	Se guardará el registro de la petición de servicio técnico mediante el botón “guardar petición”.
Postcondición	Se ejecutarán los casos siguientes según la elección del usuario: registrar petición de servicio.	
Excepciones	Paso	Acción
	2	Si el usuario no agrega una petición.
		2.1

Cuadro 19

Caso de uso: Registrar petición de servicio.

, H

Caso de uso	Agregar equipo o insumo	
Prioridad	Alta.	
Actores	Administrador.	
Precondición	Debe haber iniciado sesión y accedido al módulo de inventarios.	
Descripción	El administrador encontrará los apartados de equipo e insumos, donde al acceder a cada uno de ellos se mostrará una tabla con los registros previamente hechos y con un botón para agregar nuevo equipo o insumo dependiendo a que apartado acceda.	
Secuencia normal	Paso	Acción
	1	El usuario accede al módulo de inventarios desde el menú.
	2	El sistema mostrará los apartados de equipos e insumos donde el usuario decidirá a cual acceder.
	3	Al acceder a cualquiera de ellos se le mostrará una tabla con la información que contiene cada apartado y un botón para agregar nuevo equipo o insumo.

	4	Al agregar un nuevo equipo o insumo el sistema solicitará los datos correspondientes para crear el registro.	
	5	Al completar los datos se deberá finalizar el registro por medio del botón guardar.	
Postcondición	Se ejecutarán los casos siguientes según la elección del usuario: gestión de inventarios de equipos e insumos de trabajos, agregar.		
Excepciones	Paso	Acción	
	2	Si el usuario no agrega ningún equipo o insumo.	
	2.1	Se le mostrará la lista con los equipos e insumos registrados previamente.	

Cuadro 20

Caso de uso: Agregar equipo o insumo.

Caso de uso	Generar reportes	
Prioridad	Alta.	
Actores	Administrador.	
Precondición	Debe haber seleccionado la opción “generar reporte” en el menú del sistema.	
Descripción	El sistema le mostrará un menú al usuario donde elegirá el proceso a realizar.	
Secuencia normal	Paso	Acción
	1	El usuario elige una opción del menú.
	2	Según la opción elegida, se le mostrará la vista en la que se desarrolla el proceso.

	3	Se muestran los ratos solicitados por el usuario.	
Postcondición	Se ejecutarán los casos siguientes según la elección del usuario: generar reporte de inventario, generar reporte de mantenimientos, generar reporte de peticiones, generar horas de trabajo, generar hojas de descarte, generar hojas de calendarización.		
Excepciones	Paso	Acción	
	2	Si el usuario no selecciona ninguna opción.	
		2.1	Se mantendrá en la pantalla de inicio del módulo presente.

Cuadro 21

Caso de uso: Generar reportes.

Caso de uso	Registro de usuario	
Prioridad	Alta.	
Actores	Administrador, Usuario técnico	
Precondición	Debe haber accedido a la vista registro e inicio de sesión del sistema.	
Descripción	El sistema mostrará una vista donde el usuario tenga la opción de crear su usuario.	
Secuencia normal	Paso	Acción
	1	El usuario accede a la vista del registro de usuario.
	2	El usuario elige la opción de crear nuevo usuario.

	3	El sistema solicita las credenciales para el nuevo usuario.
	4	El sistema solicitará: - Nombre completo - Ubicación - Correo electrónico - Contraseña
	5	El usuario llenará los campos y guardará el registro.
Postcondición	-	
Excepciones	Paso	Acción
	2	Si el usuario no selecciona la opción de crear nuevo usuario.
	2.1	Se mostrarán las opciones para iniciar sesión o recuperar contraseña al ser un usuario ya registrado.

Cuadro 22

Caso de uso: Registro de usuario.

Diagramas de casos de uso de la aplicación móvil

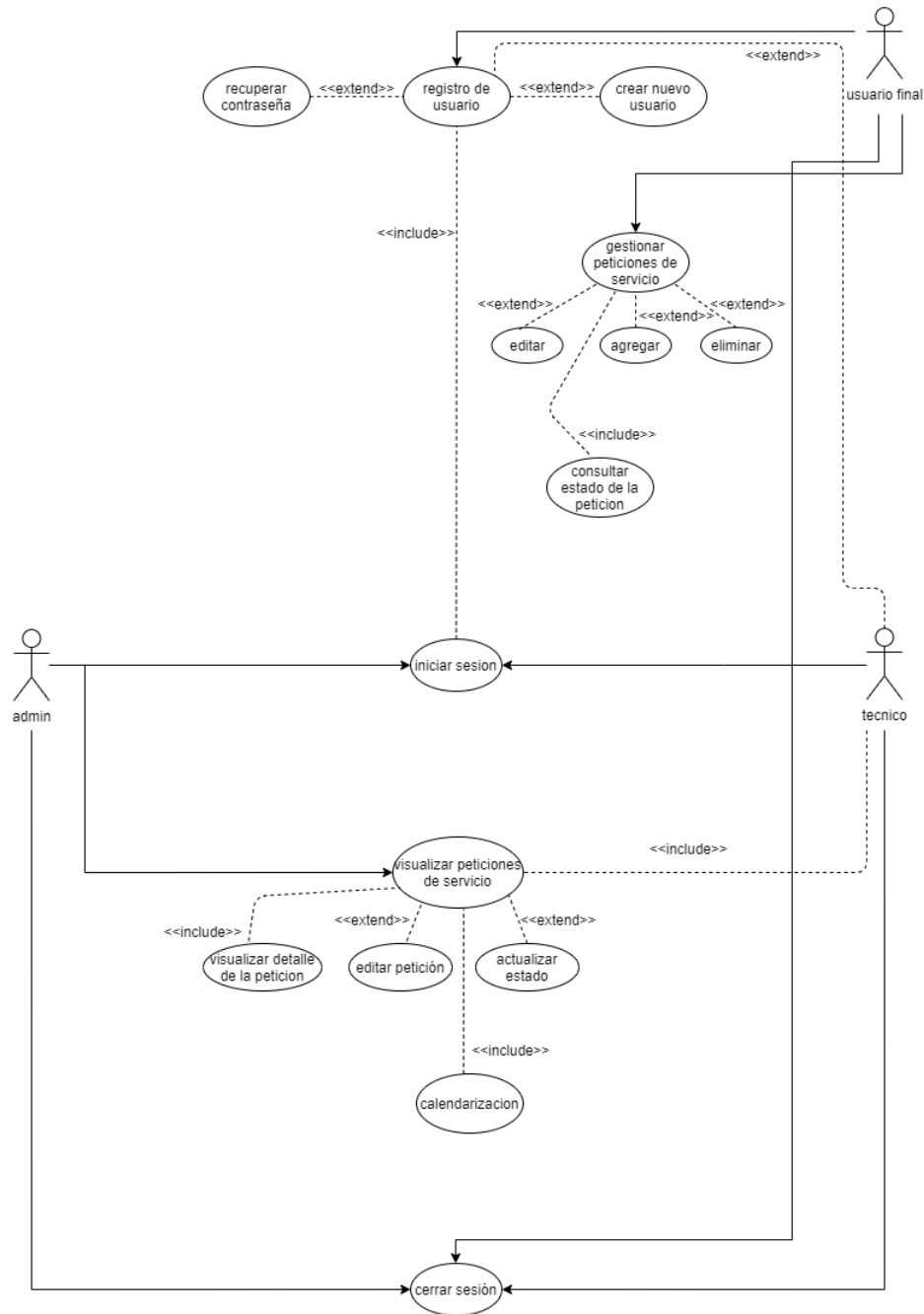


Figura 20
Caso de uso de la aplicación web.

Caso de uso	Iniciar sesión	
Prioridad	Alta.	
Actores	Administrador, Usuario Técnico, Usuario Jefe de Unidad o Departamento.	
Precondición	-	
Descripción	La aplicación móvil requiere iniciar sesión para poder acceder a sus funciones. En la pantalla se mostrará un formulario donde se solicitará las credenciales del usuario.	
Secuencia normal	Paso	Acción
	1	El usuario inicia la aplicación móvil mediante una petición al servidor.
	2	La aplicación móvil solicitará las credenciales de inicio de sesión del usuario.
	3	La aplicación móvil mostrará la pantalla de inicio con las opciones del menú según sea el rol del usuario.
Postcondición	Se ejecutará el inicio según sea el rol del usuario en que se mostrarán los procesos que este pueda realizar.	
Excepciones	Paso	Acción
	2	Si la aplicación móvil no encuentra las credenciales ingresadas.

		2.1	La aplicación móvil notificará al usuario que sus credenciales son incorrectas y mostrará el formulario de inicio de sesión nuevamente, con opción de recuperar contraseña.
--	--	-----	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Cuadro 23

Caso de uso: Iniciar sesión en la aplicación móvil.

Caso de uso	Gestionar peticiones de servicio	
Prioridad	Alta.	
Actores	Administrador, usuario técnico, usuario jefe de unidad o departamento.	
Precondición	El usuario debe haber iniciado sesión en la aplicación móvil.	
Descripción	Este proceso tiene como fin mostrarle a los usuarios el estado de una petición ingresada, en el caso de los usuarios jefe de unidad o departamento para verificar si su petición ya fue solventada o sigue en proceso; así mismo para los usuarios administrativo y técnicos para visualizar la lista de peticiones a atender.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona dentro del menú la aplicación móvil la opción de peticiones.
	2	Se mostrará una lista de las peticiones con opciones para realizar diferentes procesos.

	3	Se muestran los datos correspondientes según la opción seleccionada por el usuario.
Postcondición	<p>Se ejecutarán los casos de uso siguientes según la elección del usuario final: editar, agregar, eliminar, consultar estado de la petición.</p> <p>Se ejecutarán los casos de uso siguientes según la elección del usuario técnico o administrador:</p> <p>visualizar detalle de la petición, editar petición, actualizar estado, calendarización.</p>	
Excepciones	Paso	Acción
	2	Si el usuario no selecciona ninguna opción.
		2.1

Cuadro 24

Caso de uso: Gestionar peticiones en la aplicación móvil.

Caso de uso	Agregar nueva petición de servicio
Prioridad	Alta.
Actores	Usuario jefe de unidad o departamento.
Precondición	El usuario debe haber iniciado sesión en la aplicación móvil.

Descripción	El usuario podrá acceder al apartado de peticiones que podrá seleccionar previamente desde el menú donde podrá agregar una nueva petición de servicio técnico.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la opción peticiones en el menú.
	2	Se le mostrará un botón donde se le permitirá agregar una nueva petición, como también la lista de peticiones agregadas previamente.
	3	Se le solicitarán los datos para crear una nueva petición los cuales serán: tipo de petición y detalles de petición.
4	Se guardará el registro de la petición de servicio técnico mediante el botón "Guardar petición".	
Postcondición	Se ejecutarán los casos de uso siguientes según la elección del usuario: gestionar peticiones de servicio técnico, agregar petición.	
Excepciones	Paso	Acción
	2	Si el usuario no selecciona ninguna opción. 2.1 Se mostrarán todas las solicitudes ordenadas por prioridad y fecha.

Cuadro 25

Caso de uso: Agregar petición en la aplicación móvil.

Caso de uso	Consultar estado de petición	
Prioridad	Alta.	
Actores	Usuario jefe de unidad o departamento.	
Precondición	El usuario debe haber iniciado sesión en la aplicación móvil y accedido al apartado de peticiones.	
Descripción	La aplicación móvil mostrará la lista de peticiones que el usuario ha ingresado.	
Secuencia normal	Paso	Acción
	1	El usuario seleccionará una petición de la que desee ver el estado.
	2	Al seleccionar la petición se le mostrará el estado actual.
Postcondición	Se ejecutarán los casos de uso siguientes según la elección del usuario: consultar estado de la petición.	
Excepciones	Paso	Acción
	2	Si el usuario no selecciona ninguna opción.
		2.1

Cuadro 26

Caso de uso: Consultar estado de petición en la aplicación móvil.

Caso de uso	Registro de usuario
Prioridad	Alta.
Actores	Usuario técnico, usuario jefe de unidad o departamento.

Precondición	Debe haber accedido a la vista de registro e inicio de sesión de la aplicación móvil.	
Descripción	La aplicación móvil en su vista inicial mostrará un formulario para el registro del usuario podrá ingresar sus datos que se volverán sus credenciales para poder acceder a la interfaz de la aplicación.	
Secuencia normal	Paso	Acción
	1	El usuario técnico o usuario final acceden a la vista de registro de usuario.
	2	El usuario elige la opción de crear nuevo usuario.
	3	La aplicación móvil solicita los datos para el nuevo usuario.
	4	La aplicación móvil solicitará: - Nombre completo - Ubicación - Correo electrónico - Contraseña
	5	El usuario llenará los campos y guardará el registro.
	6	Se guarda la solicitud en la aplicación móvil.
Postcondición	Se ejecutarán los casos de uso siguientes según la elección del usuario: consultar estado de la petición.	
	Paso	Acción

Excepciones

	2	Si el usuario no selecciona la opción de crear un nuevo usuario.
	2.1	Se mostrarán las opciones para iniciar sesión o recuperar contraseña al ser un usuario ya registrado.

Cuadro 27

Caso de uso: Registro de usuario en la aplicación móvil.

Diagramas de secuencia de la aplicación web

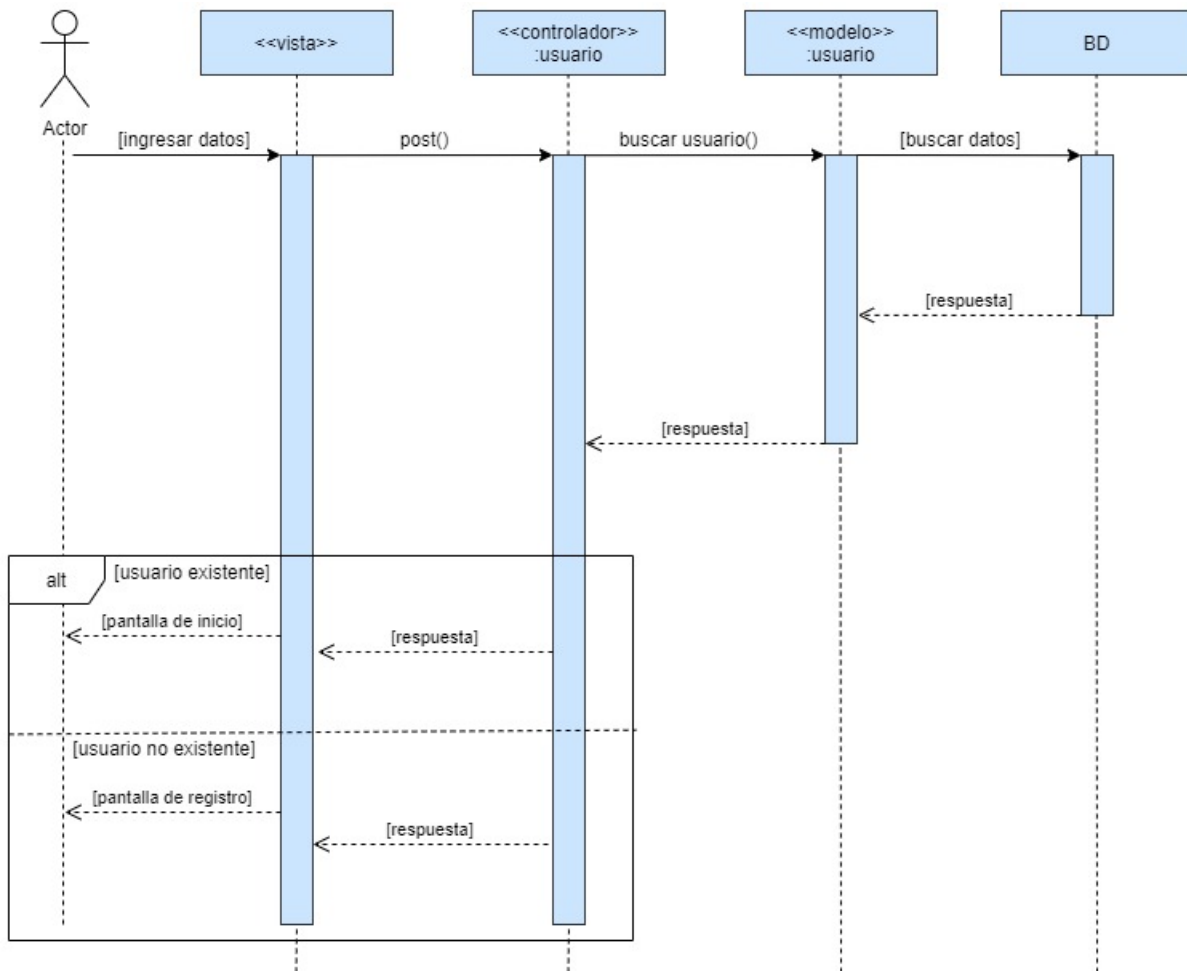


Figura 21

Diagrama de secuencia: inicio de sesión.

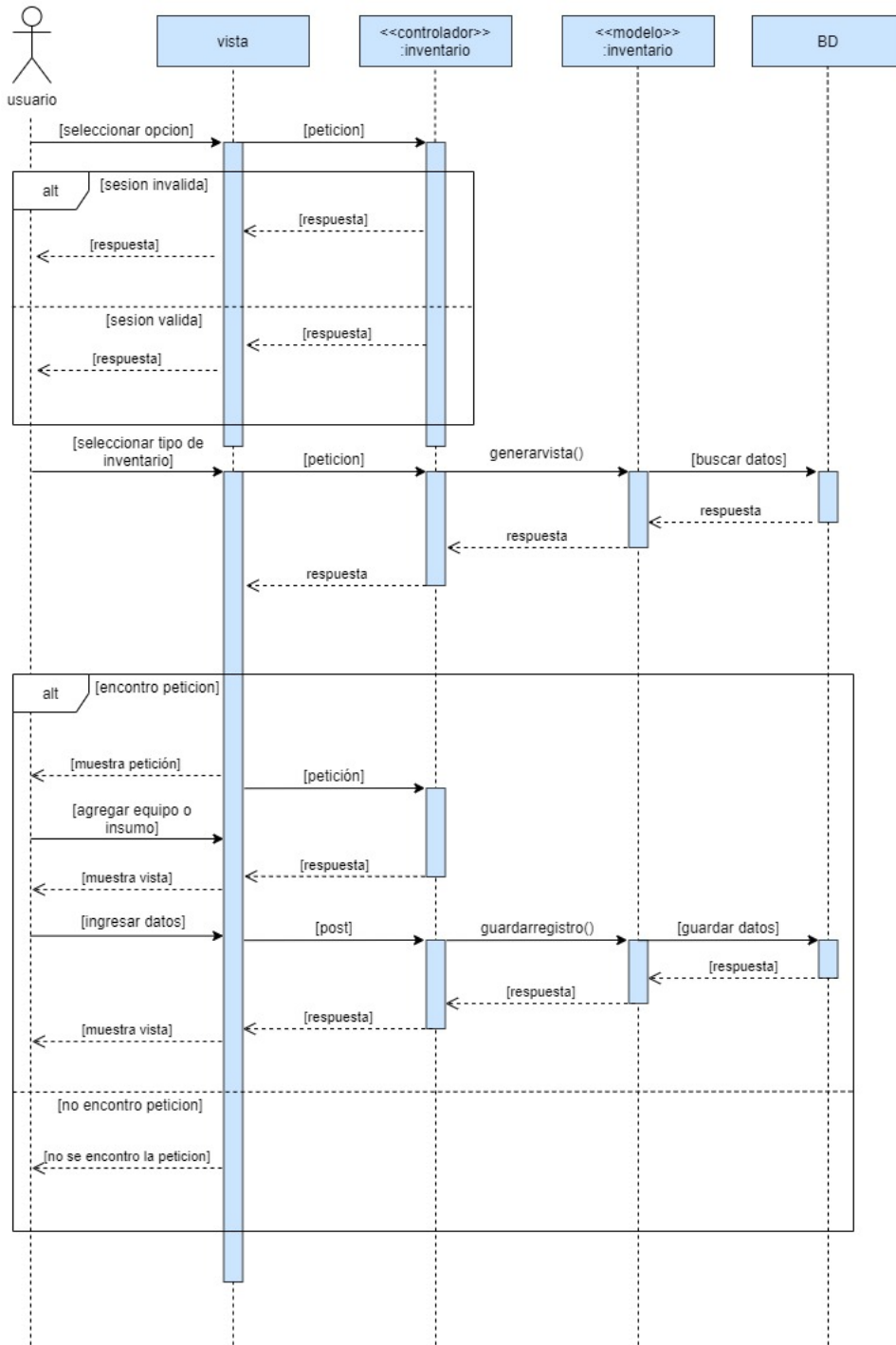


Figura 22
 Diagrama de secuencia: agregar equipos e insumos.

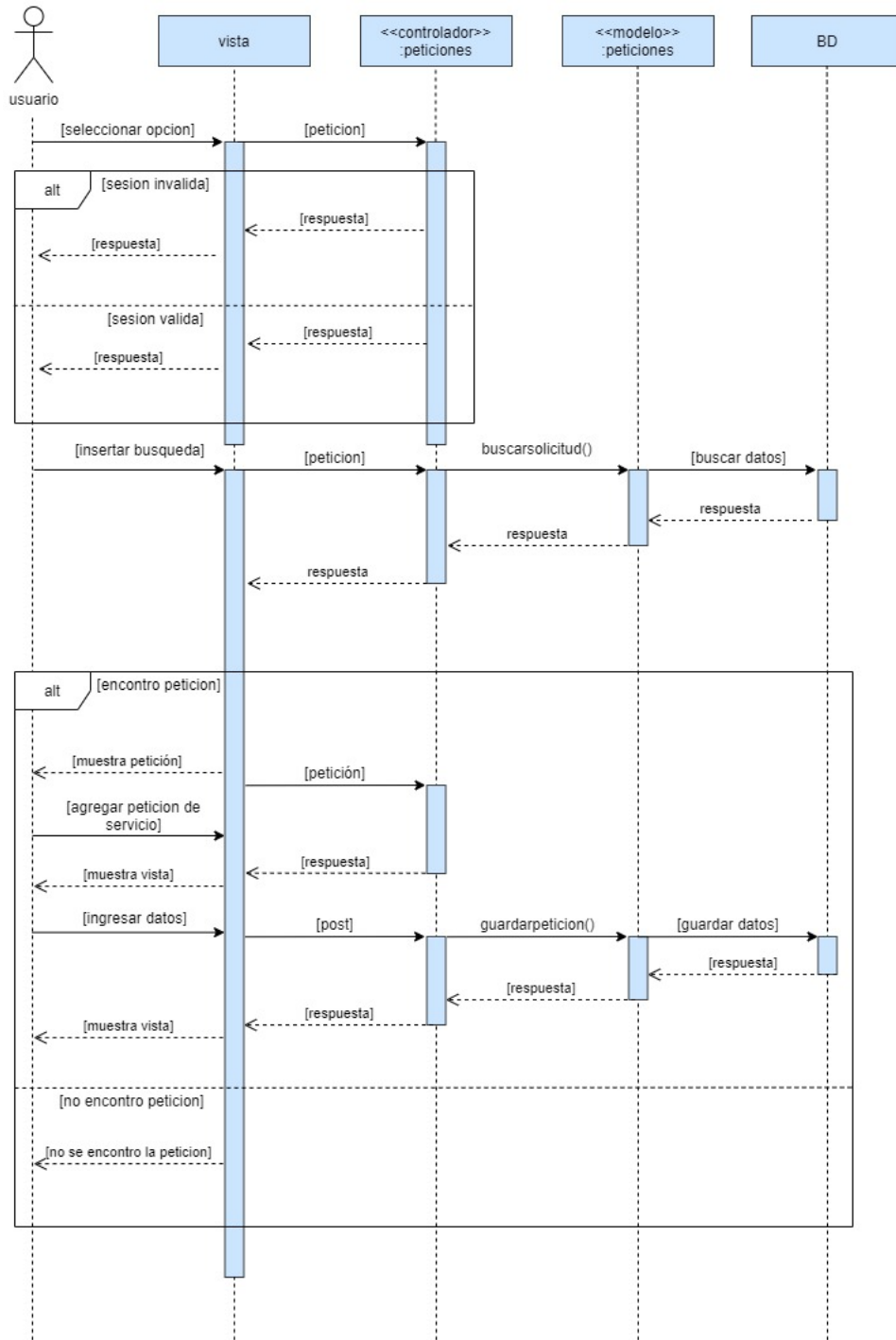


Figura 23
Diagrama de secuencia: agregar petición.

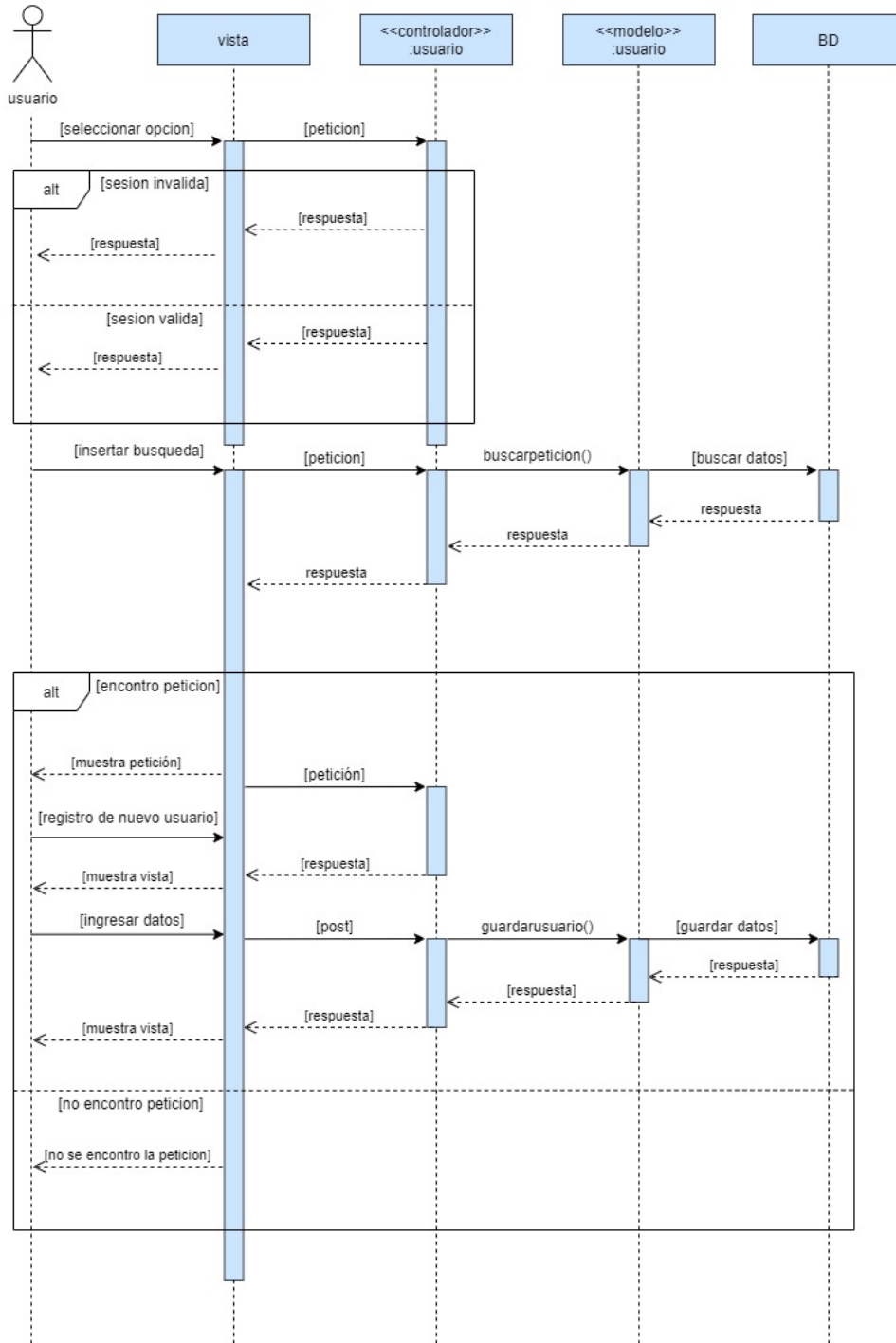


Figura 24
 Diagrama de secuencia: agregar usuario.

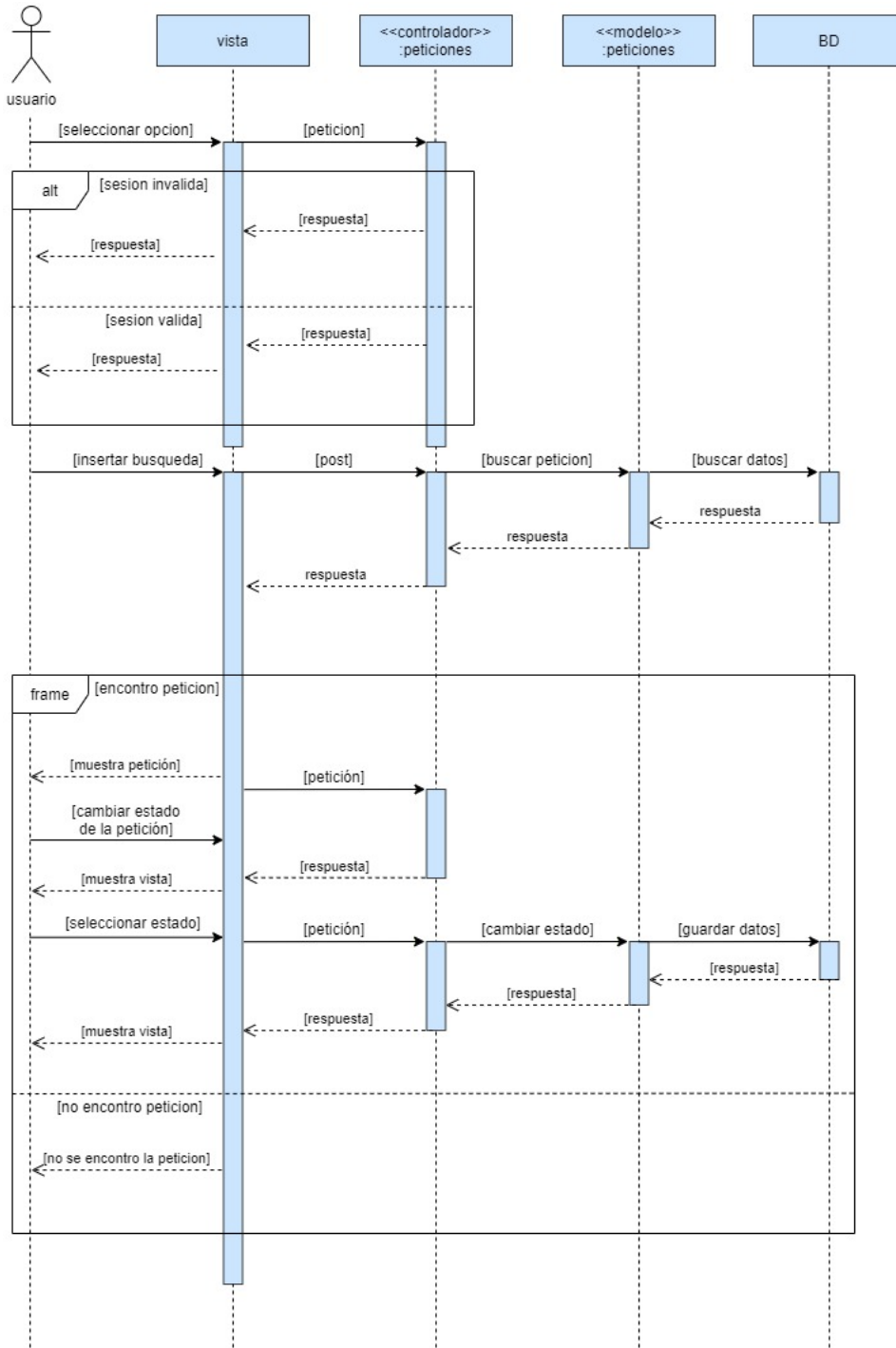


Figura 25
 Diagrama de secuencia: cambiar estado de petición.

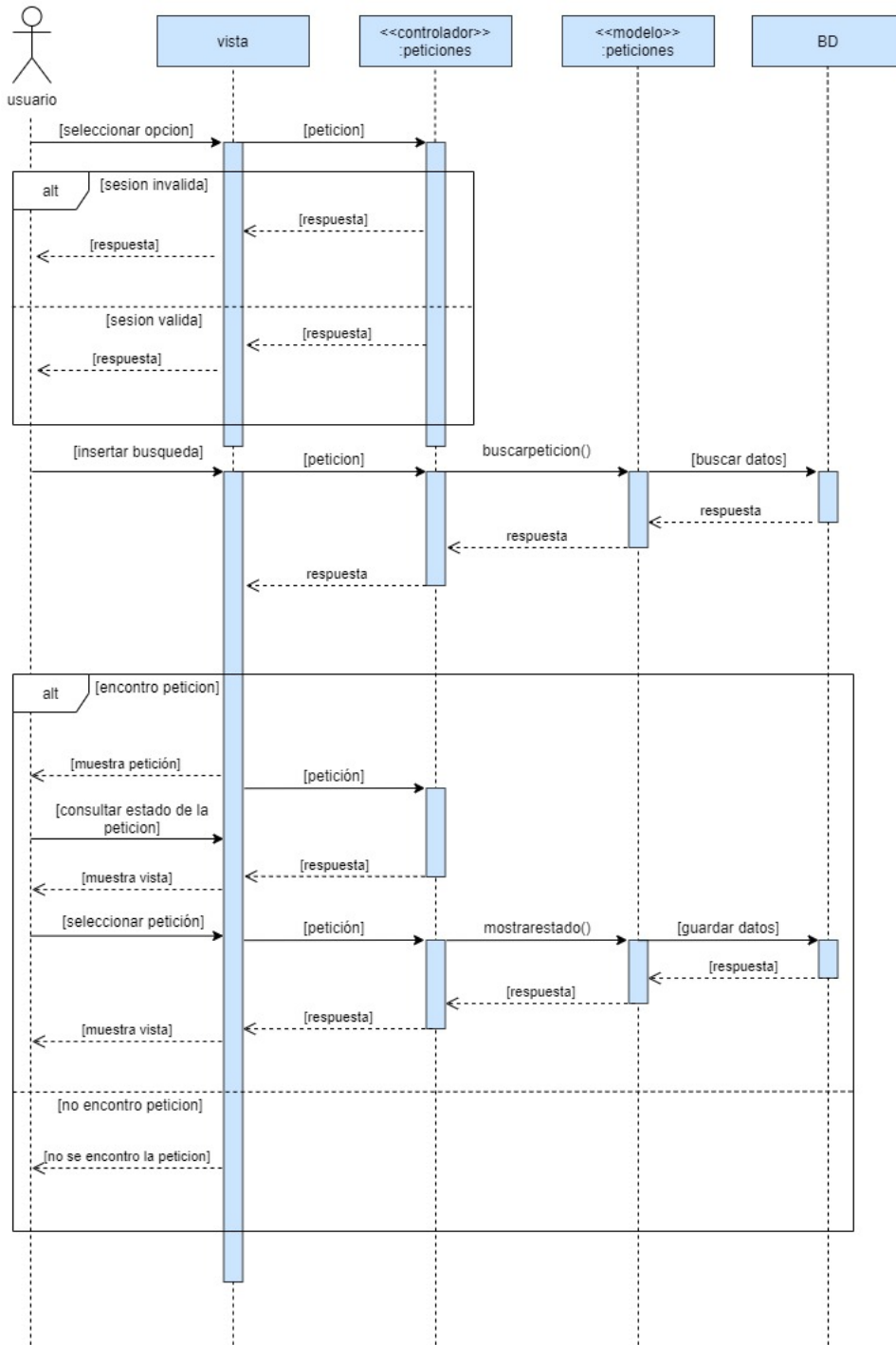


Figura 26
Diagrama de secuencia: consultar estado de petición.

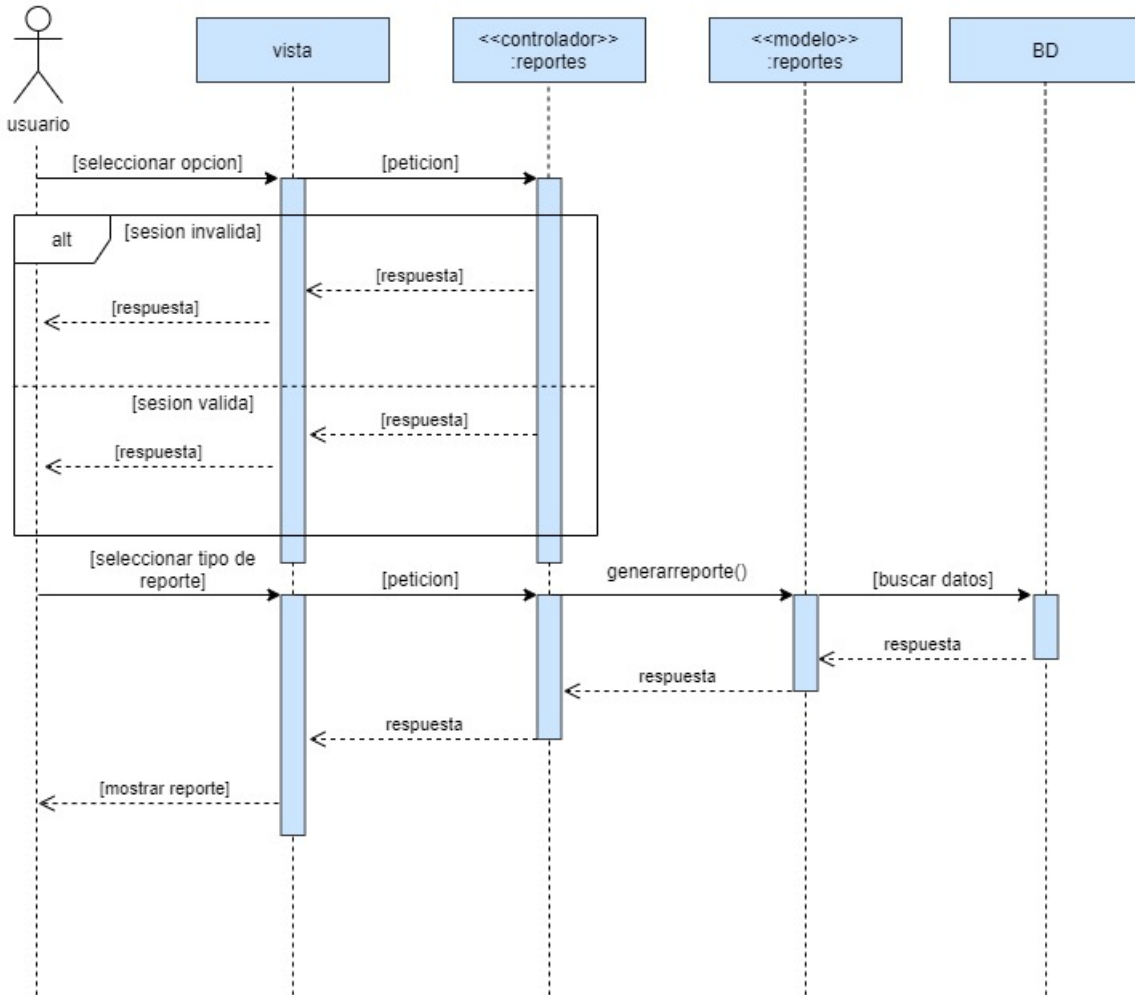


Figura 27
 Diagrama de secuencia: generar reporte.

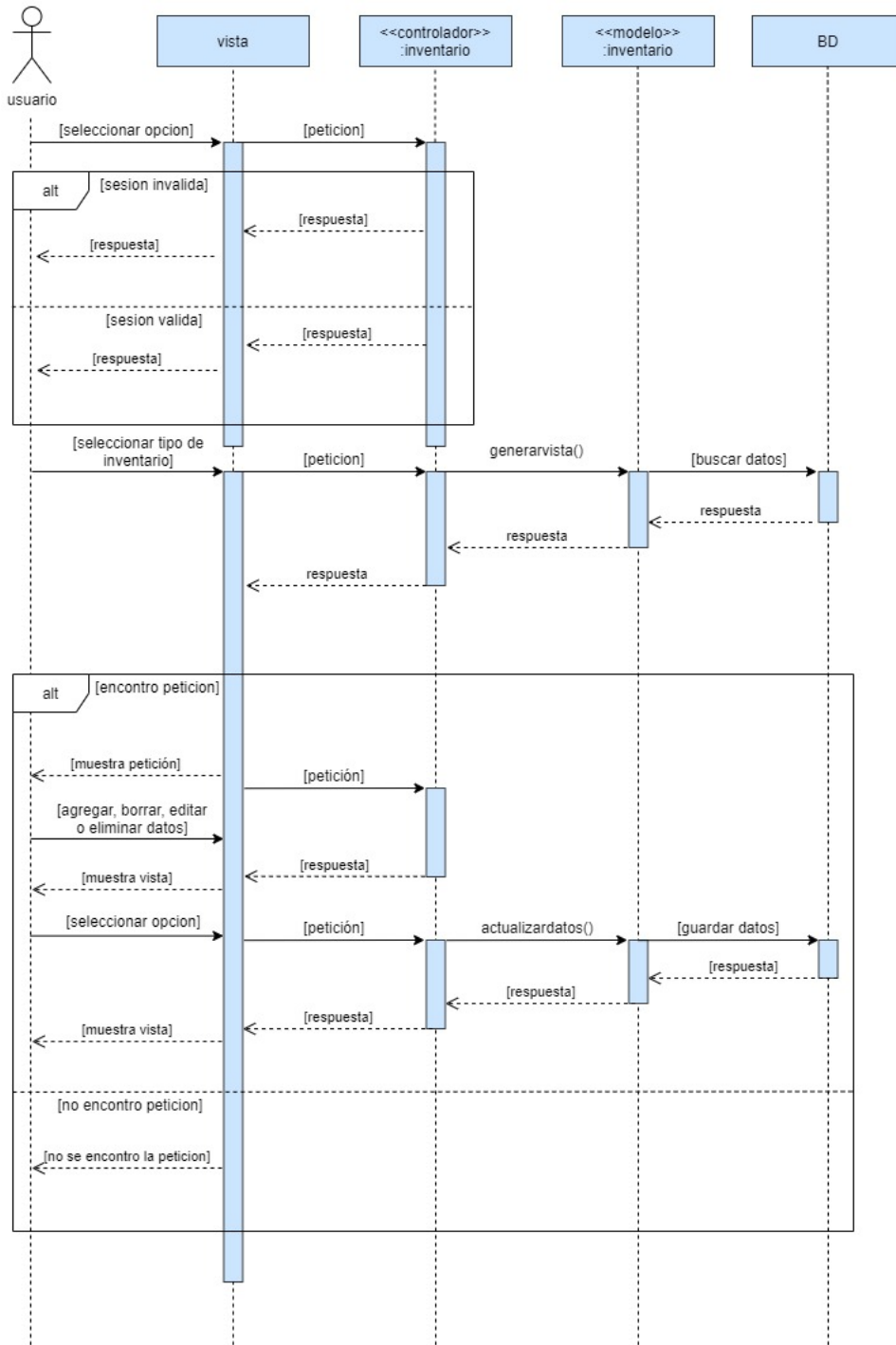


Figura 28
Diagrama de secuencia: gestión de inventarios.

Diagramas de secuencia de la aplicación móvil

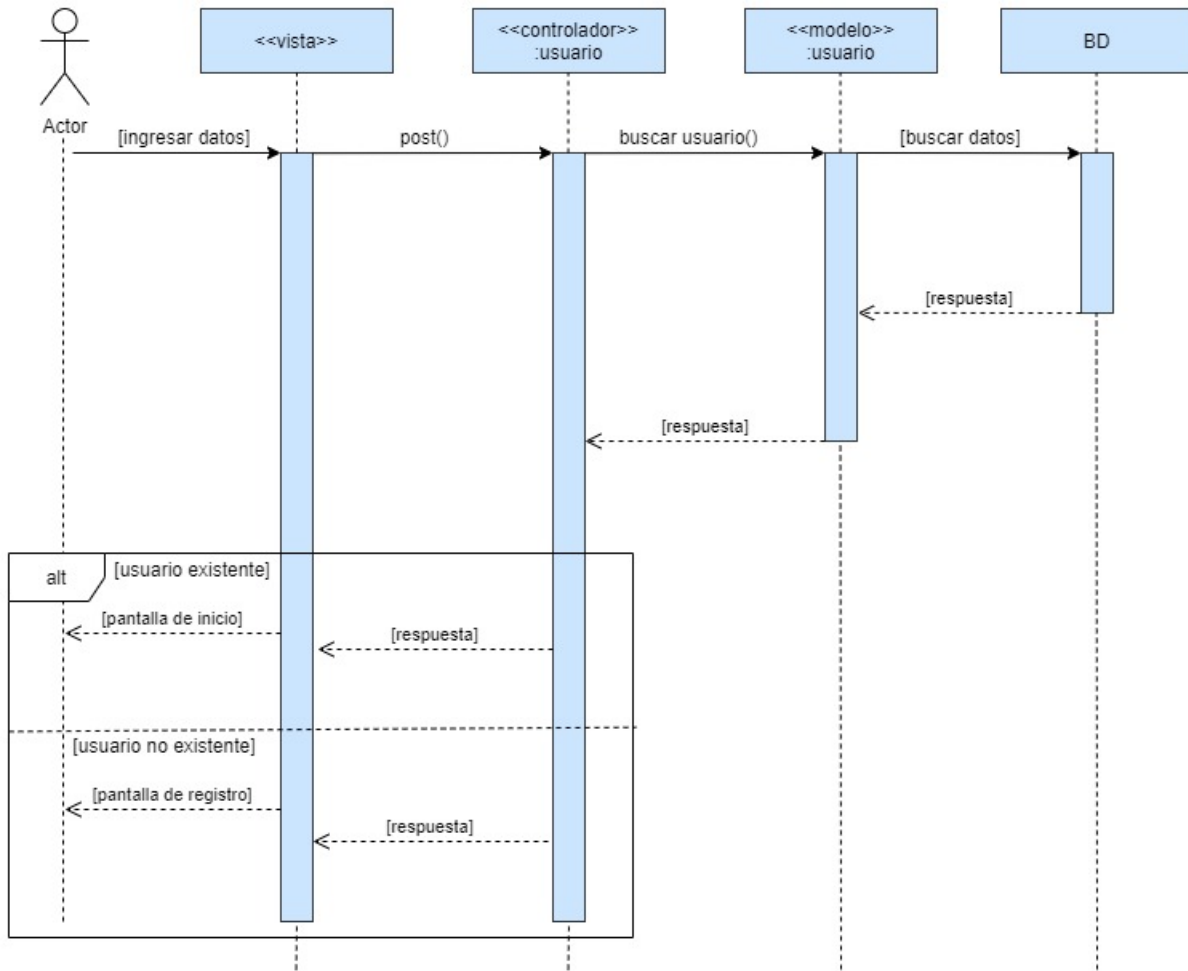


Figura 29

Diagrama de secuencia: iniciar sesión desde la aplicación móvil.

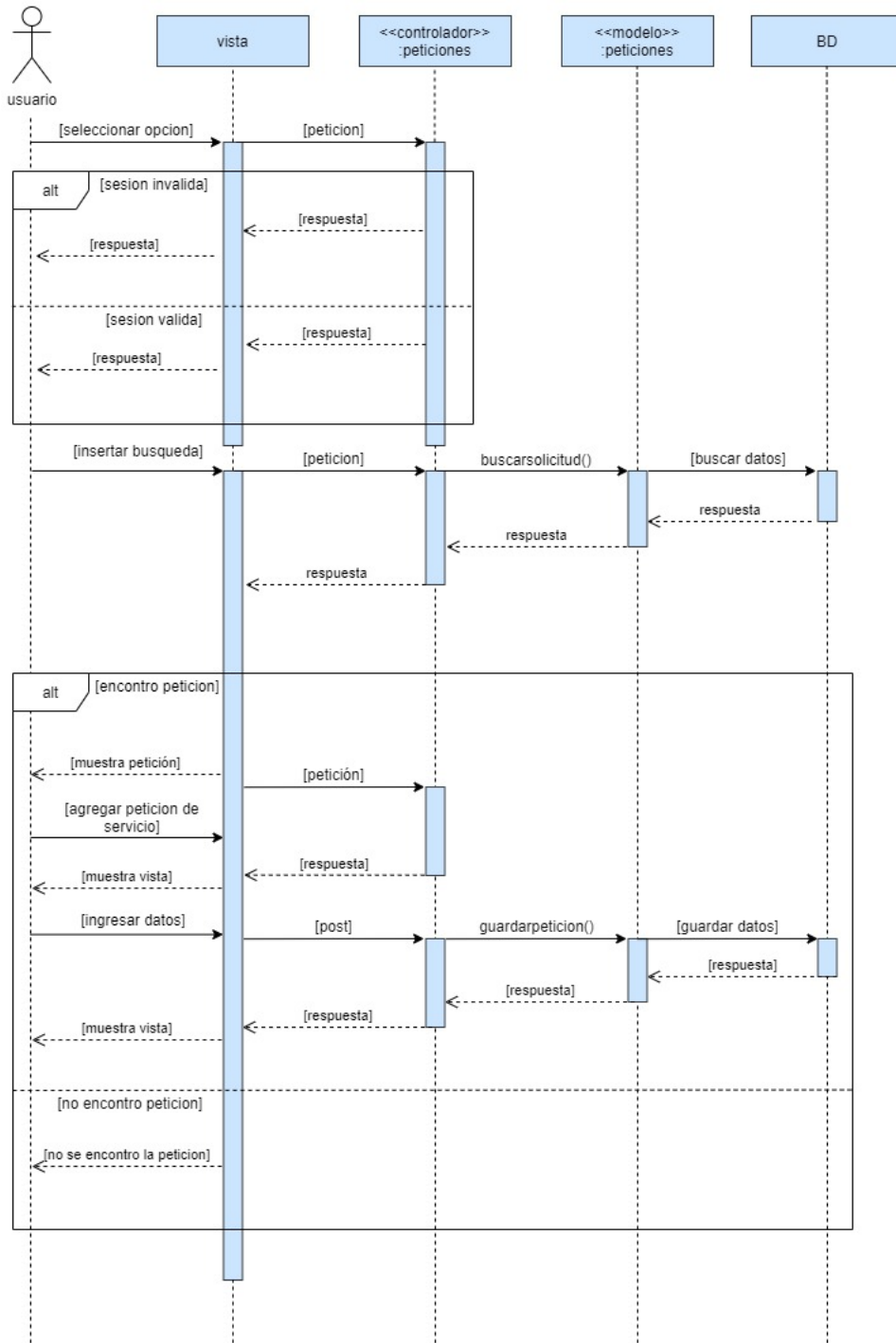


Figura 30
 Diagrama de secuencia: agregar petición.

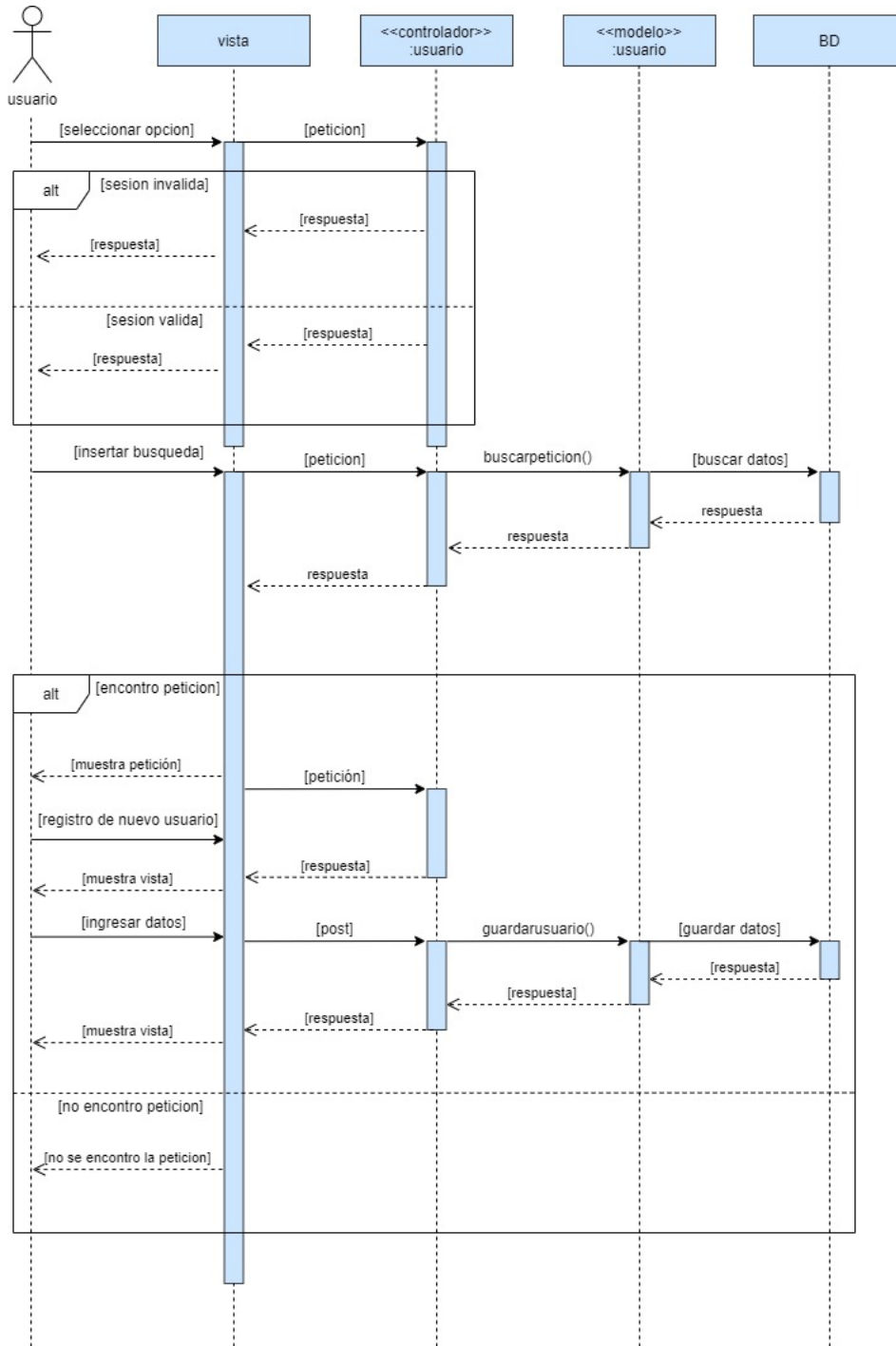


Figura 31
 Diagrama de secuencia: agregar usuario.

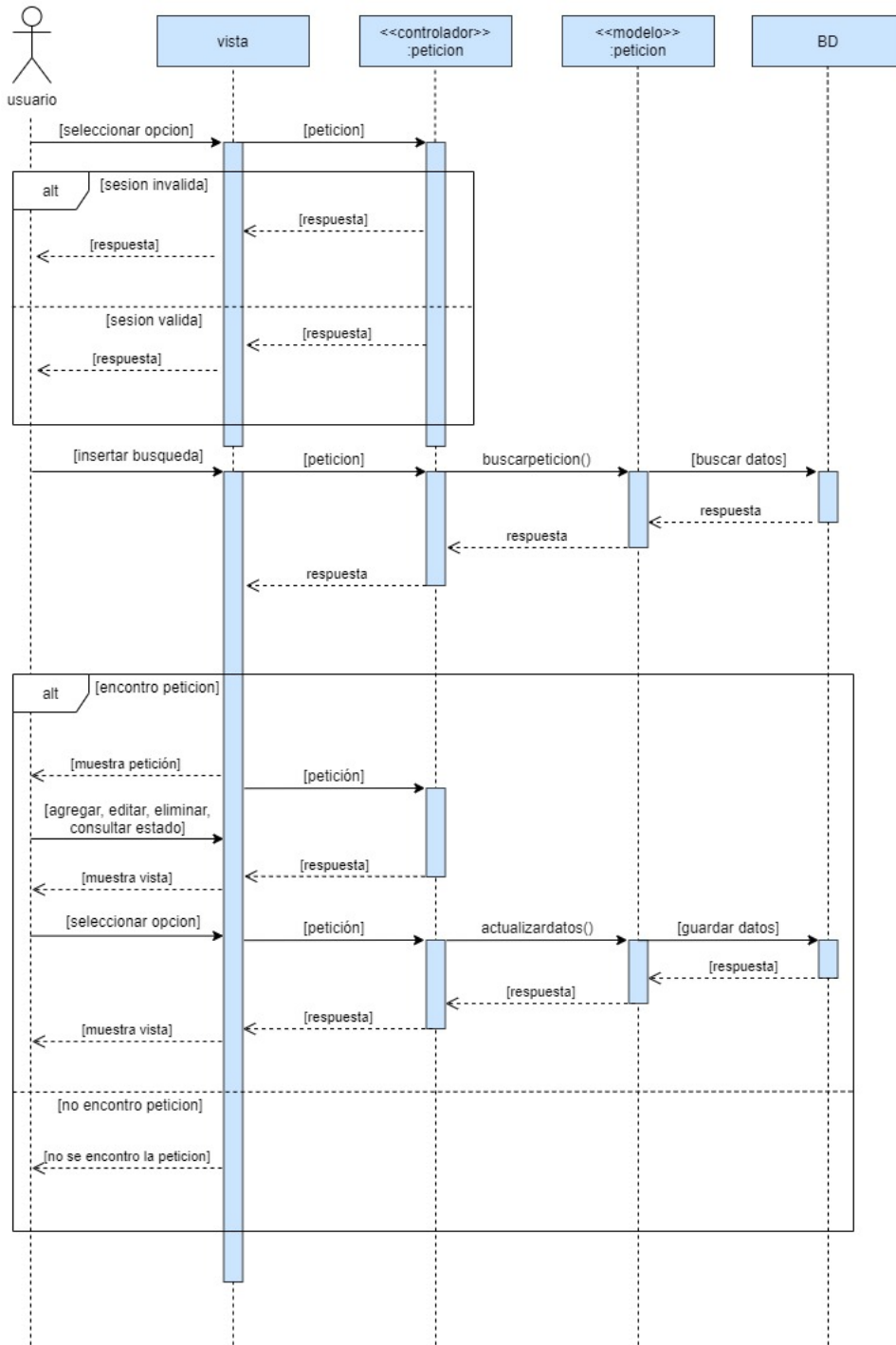


Figura 32
Diagrama de secuencia: gestión de peticiones.

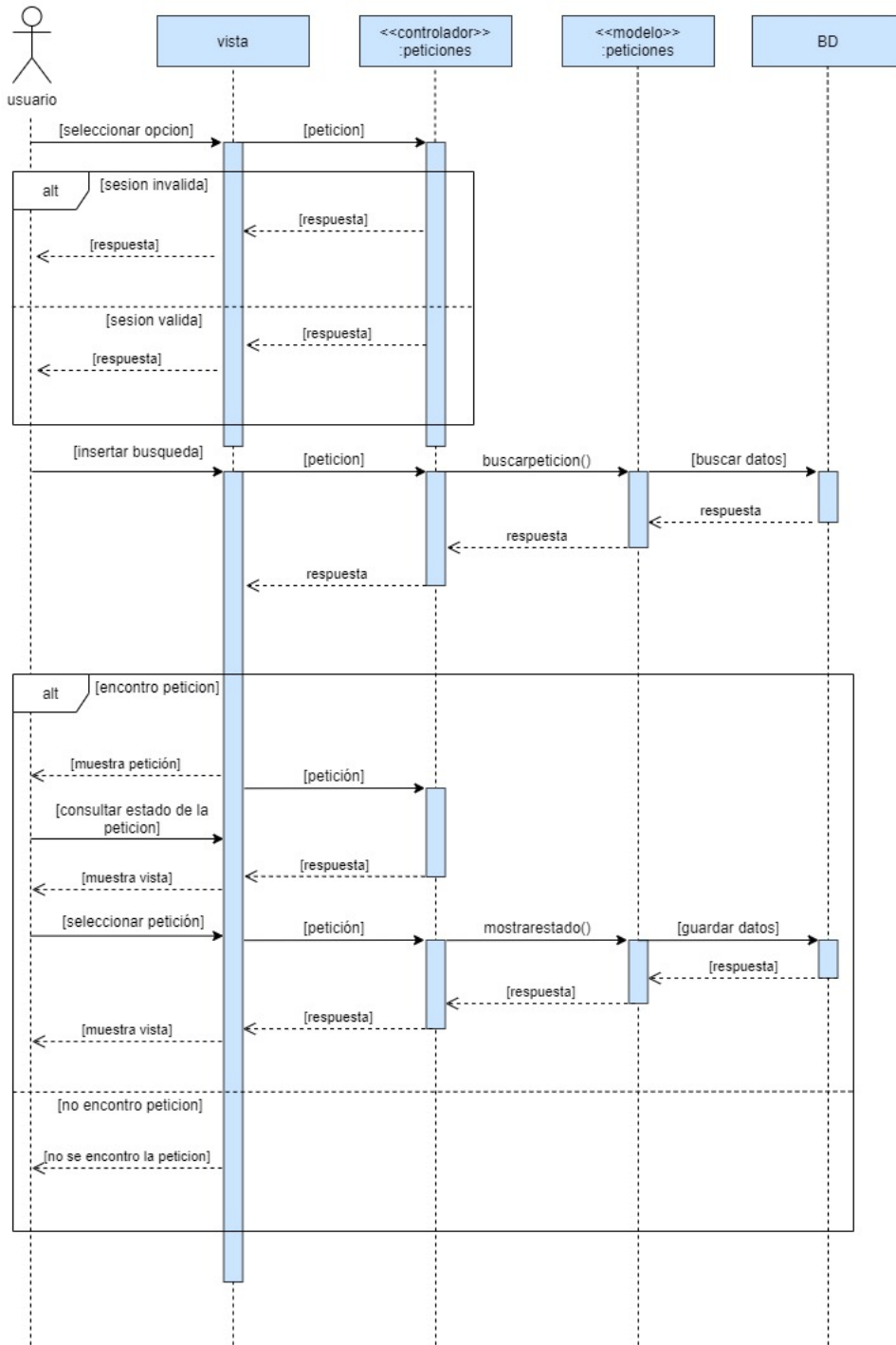


Figura 33
Diagrama de secuencia: consultar estado de la petición.

Diagrama de clases

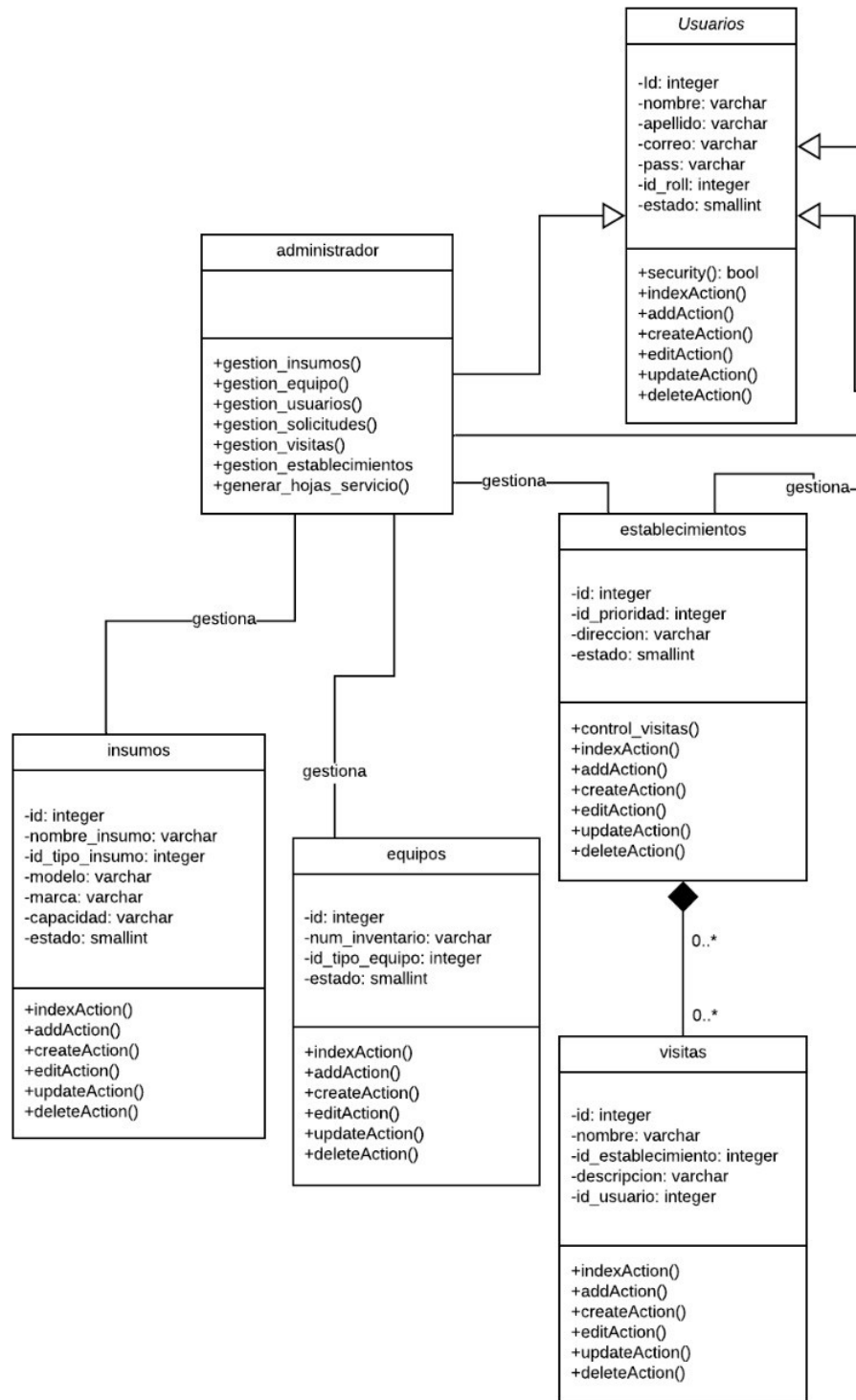


Figura 34
Diagrama de clases de la aplicación web (Parte 1).

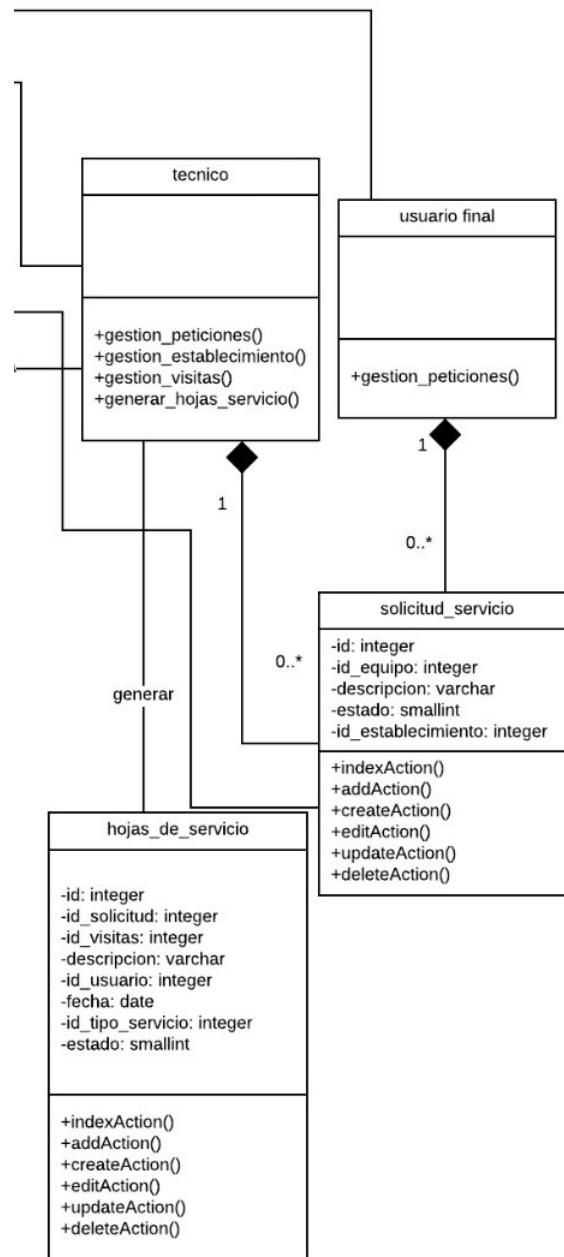


Figura 35
 Diagrama de clases de la aplicación web (Parte 2).

Estandarización de base de datos

En todo proyecto es necesario aplicar diferentes estándares, donde se plantean diferentes reglas a seguir para tener una mejor comprensión del mismo. Por esa razón, se detalla a continuación los estándares a seguir en la base de datos y las consultas que se

realizarán desde el servidor.

Para el desarrollo del sistema de gestión de solicitudes, en adelante llamado SISTEC, se utilizará, como antes mencionado el framework Symfony v2.8. Dicho framework viene integrado con mapeador de objetos relacionales llamado Doctrine ORM; el cual se utilizará para la administración de la base de datos desde Symfony.

Doctrine ORM, es un Mapeador Relacional de Objetos para el lenguaje PHP. Esto significa que trasladará los datos de una base de datos relacional, como puede ser MySQL o PostgreSQL, a un sistema de clases y de objetos, donde las clases serían las tablas y los registros pasarían a ser lo equivalente a objetos (Doctrine Project, 2020).

Para la base de datos, se utilizarán los siguientes estándares:

1. **Nombramiento de tablas.** Las tablas se nombrarán con el prefijo *tbl_nombre_tabla* evitando los espacios. Las tablas que cumplan con la función de puente, se nombrarán *tbl_nombre1_nombre2* para identificar la relación de los datos entre las tablas.
2. **Campos clave.** Todas las tablas deberán tener un campo que hacen referencia a una llave primaria (primary key). Además, se colocará las llaves foráneas (foreign key) en caso de tener relaciones con otras tablas. Doctrine genera automáticamente la llave primaria de la tabla, por lo que siempre será su nombre *id*, será de tipo número entero (integer) con un contador incremental asociado.
3. **Campos de información.** Los demás campos de la base de datos serán nombrados con palabras que hagan referencia a la información que almacenan. Si el nombre del campo se describe mejor en dos palabras, este tendrá por nombre *nombre1_nombre2*, como por ejemplo *numero_inventario*. Además, los campos con llaves foráneas, tendrán por nombre el prefijo *id_* más el nombre de la tabla a la que hace referencia, por ejemplo *id_categorias*.
4. **Campos especiales.** Los campos especiales son los que se le añadirán a cada una de las tablas para casos de auditorías. Los campos a añadir serán *usuario_creacion*, que

guardará el *id* del usuario que ha creado el registro; *fecha_creacion*, que guardará la fecha y hora en que se creó el registro; *usuario_actualizar*, que guardará el usuario que hizo modificación en el registro y *fecha_actualizar*, que guardará la fecha y hora en que el registro fue actualizado.

5. **Consultas para mostrar registros.** Las consultas para mostrar registros completos de una tabla se realizará usando las funciones que proporciona Doctrine. En caso de las consultas específicas (seleccionando ciertos datos de la tabla) o consultas complejas, se utilizará el generador de consultas DQL que proporciona Doctrine (Doctrine Project, 2020).
6. **Consultas para inserción, modificación ó eliminación.** Estas consultas se realizarán por medio de las funciones de persistencia de datos que proporciona Doctrine.
7. **Restricciones de la base de datos.** Se utilizará la restricción CASCADE, en el cual se eliminarán y actualizarán los registros pertenecientes a las tablas con los parámetros especificados mediante el generador de consultas DQL de Doctrine.

Diccionario de datos

Diagrama entidad-relación

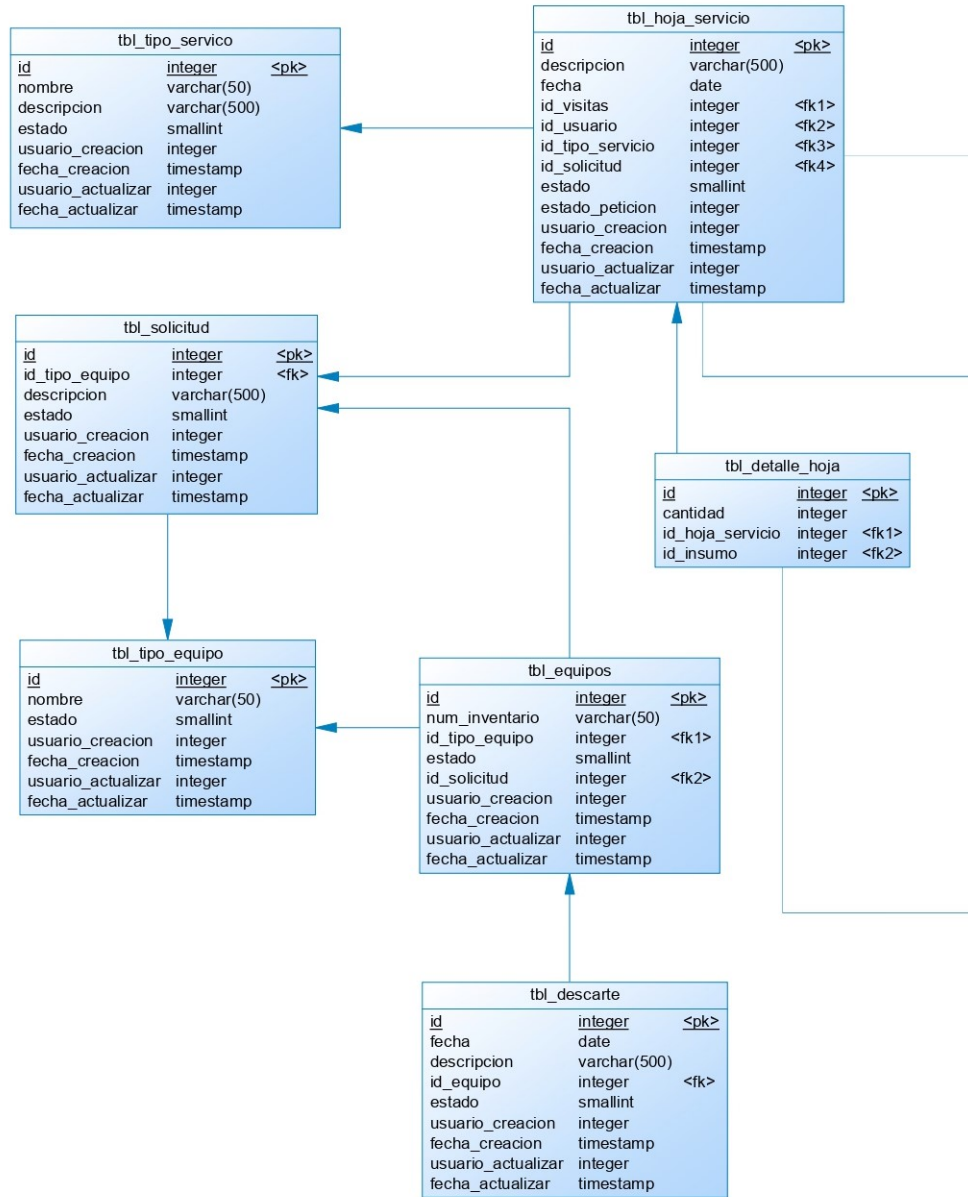


Figura 36

Diagrama entidad-relación de la base de datos db_sistec (Parte 1).

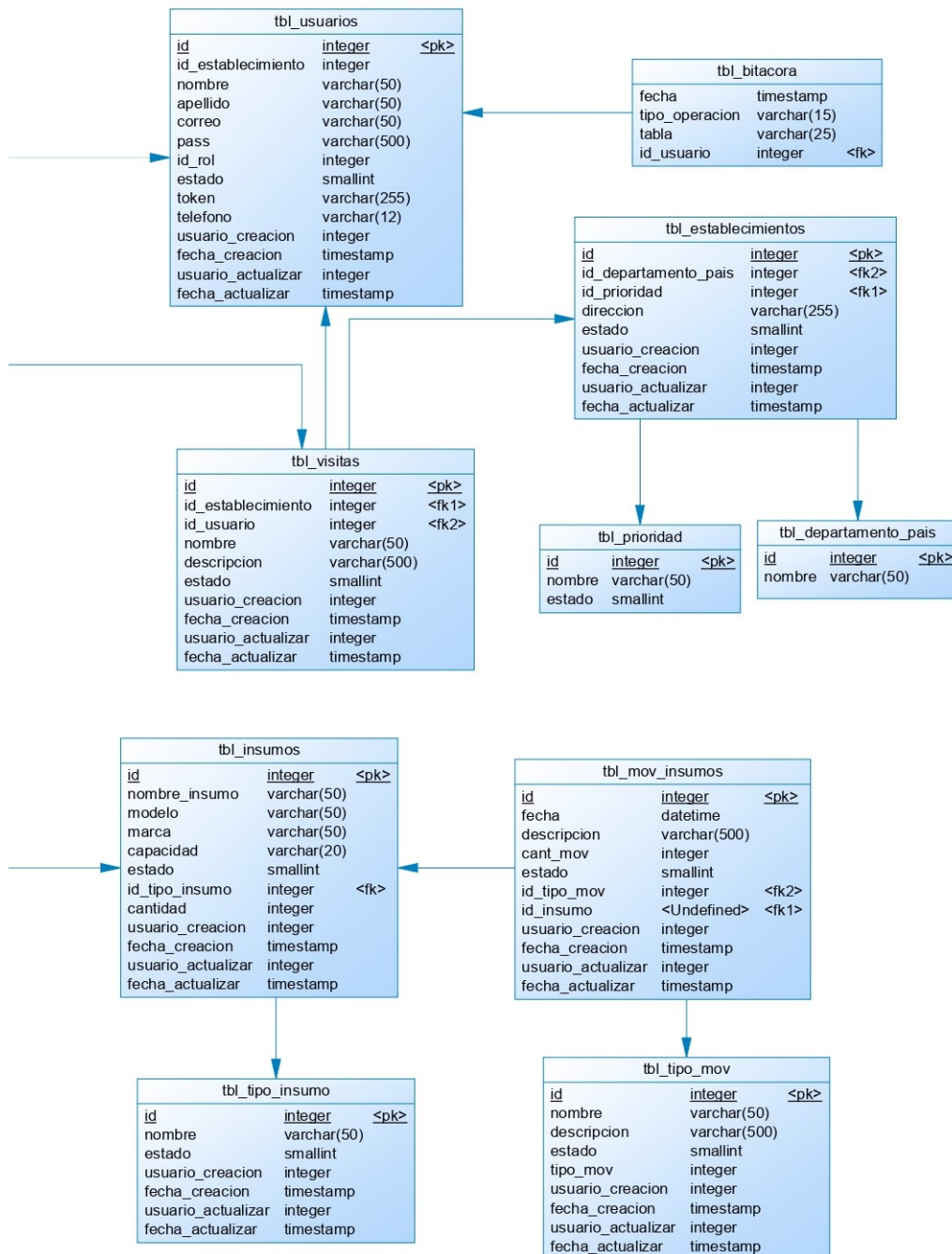


Figura 37

Diagrama entidad-relación de la base de datos db_sistec (Parte 2).

Diagrama lógico de base de datos

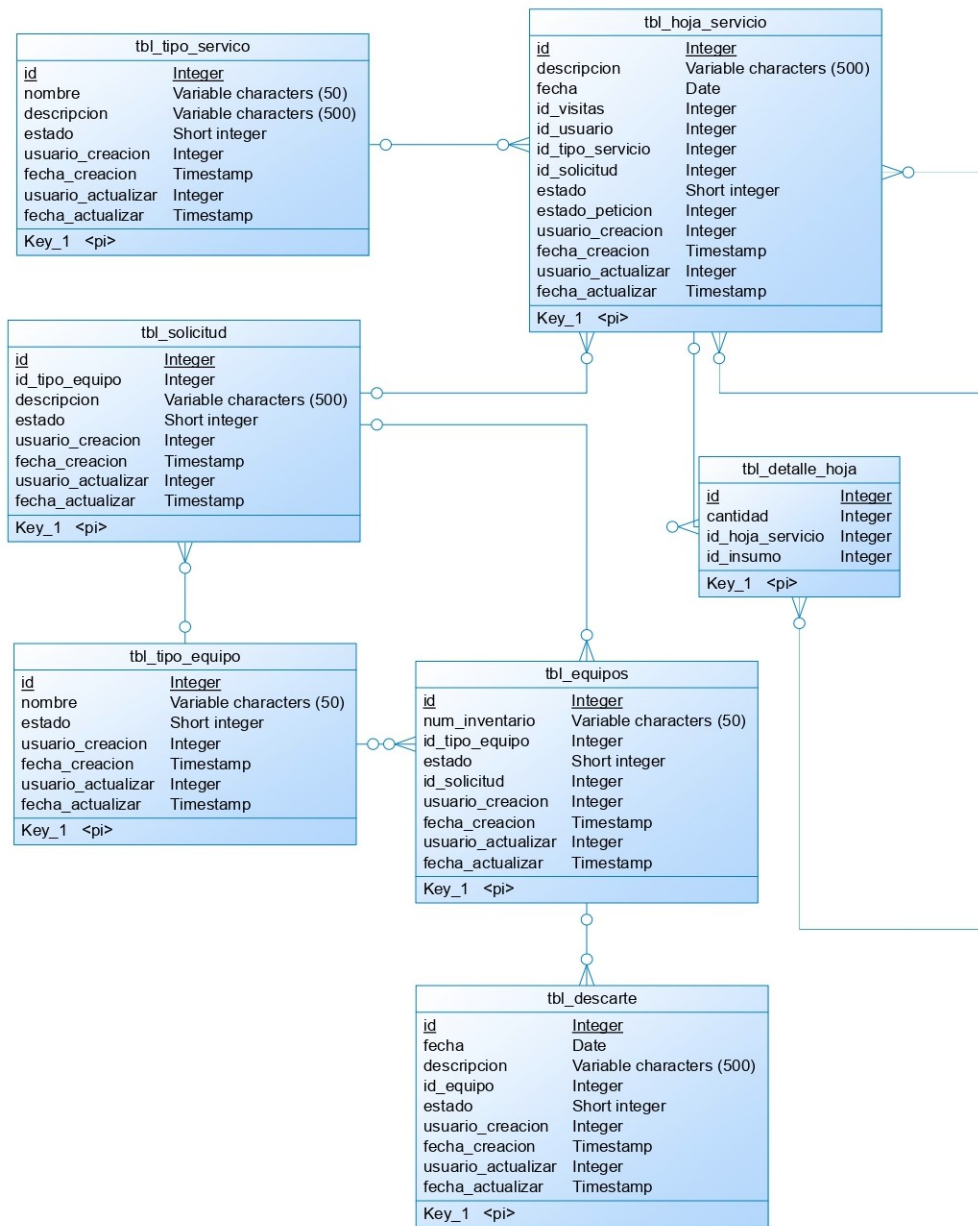


Figura 38

Diagrama lógico de la base de datos db_sistec (Parte 1).

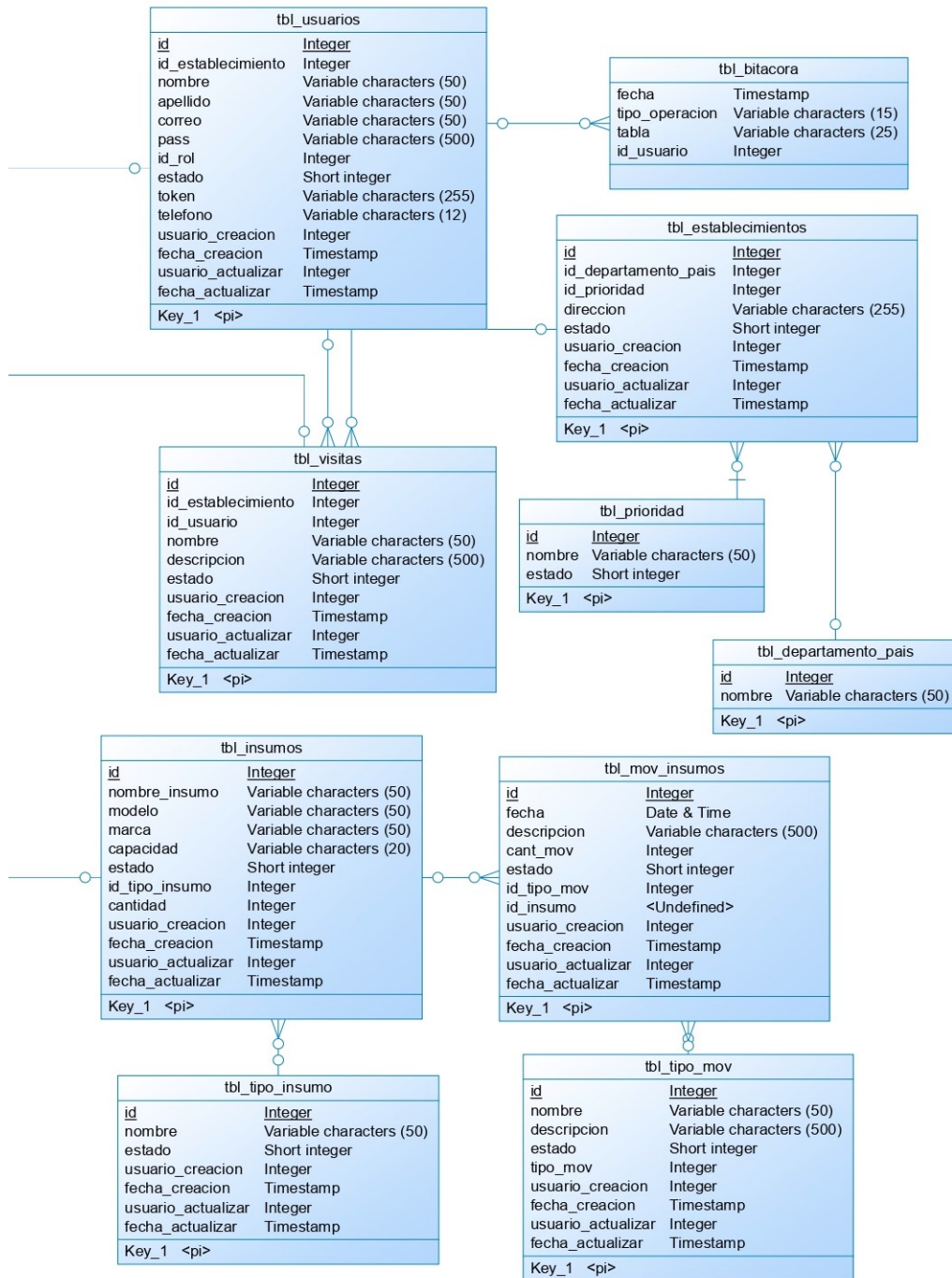


Figura 39
 Diagrama lógico de la base de datos db_sistec (Parte 2).

Diagrama conceptual de base de datos

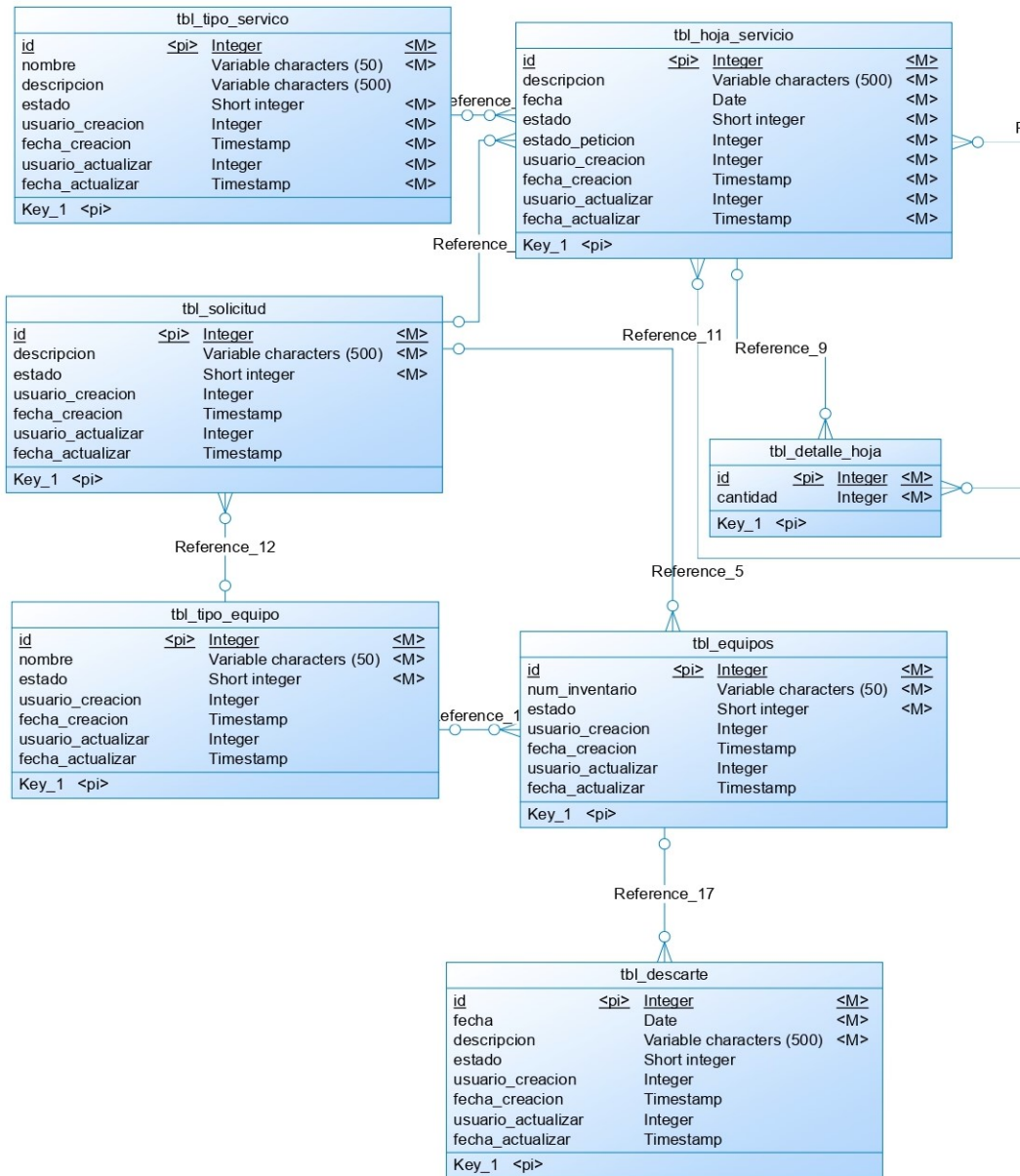


Figura 40

Diagrama conceptual de la base de datos db_sistec (Parte 1).

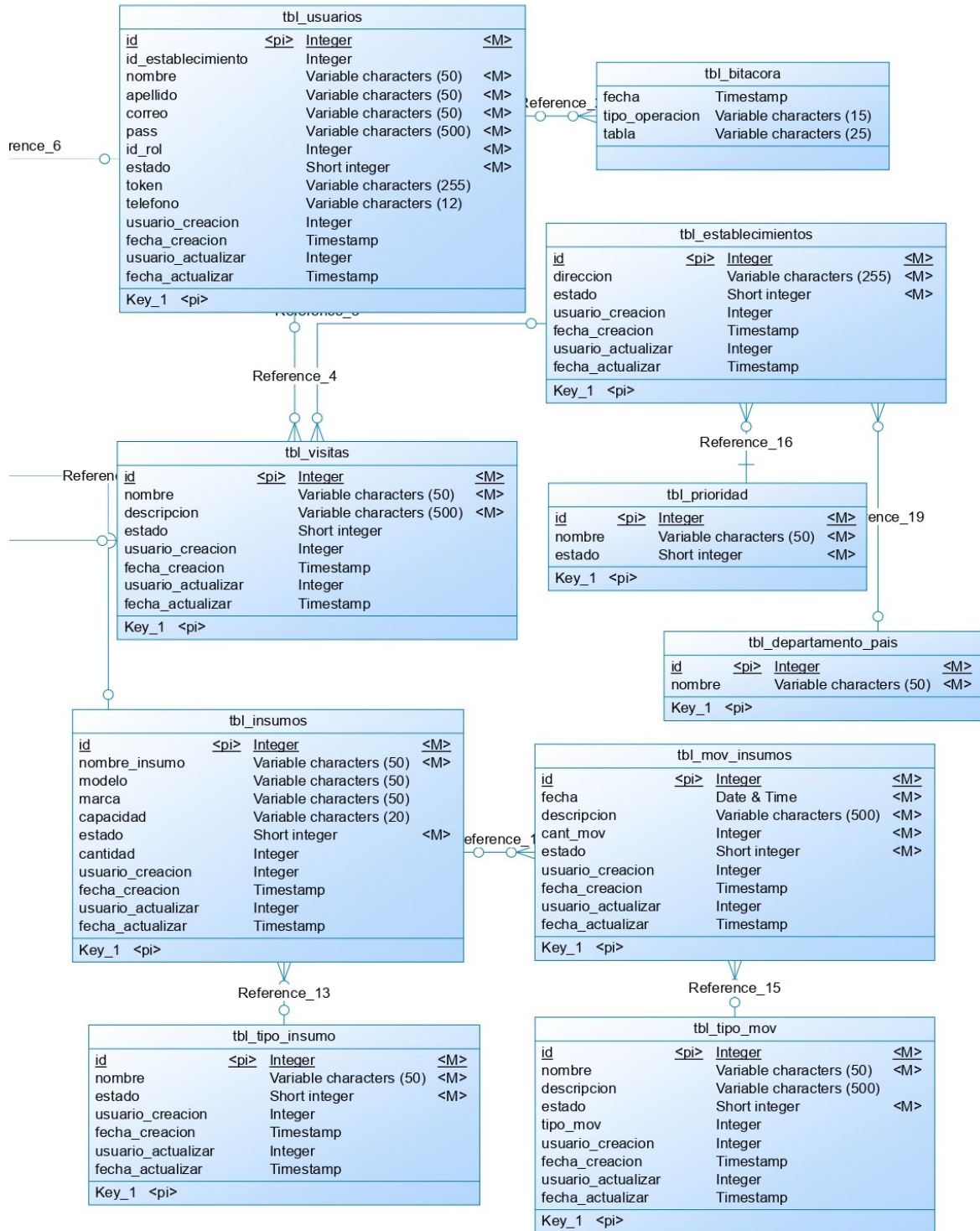


Figura 41
 Diagrama conceptual de la base de datos db_sistec (Parte 2).

Descripción de base de datos

El diccionario de datos es una lista de todos los elementos incluidos en el conjunto de los diagramas de flujo de datos y ayuda en la determinación de los requerimientos que describen un sistema. En ese contexto, el diccionario de datos almacena detalles y descripción de estos elementos.

Definiciones. A continuación, se detallan algunos conceptos importantes que se utilizará en la estructura de la base de datos.

- **Llave primaria o Primary key (PK).** Es una columna (o combinación de columnas) que permite identificar de forma inequívoca cada fila de la tabla, por lo que no puede haber en una tabla dos registros con el mismo valor en la columna definida como clave principal.
- **Llave foránea o Foreign key (FK).** Es uno de los campos de una tabla que hacen referencia al campo o campos de clave principal de otra tabla. Una llave foránea indica como están relacionadas las tablas.
- **Tipos de datos.** Son los diferentes tipos de datos que le podemos asignar a un campo, haciendo referencia a los datos que se guardarán en él posteriormente. Entre ellos tenemos:
 - *Varchar.* Es una cadena de caracteres sin espacios al final, cuya longitud es especificada al momento de su creación.
 - *Boolean.* Significa que contiene un valor booleano. Puede ser *true* o *false*.
 - *Date.* Almacena un valor de fecha ANSI SQL, formato *aaaa-mm-dd*.
 - *Timestamp.* Almacena un valor de fecha y hora, formato *aaaa-mm-dd hh:mm:ss*.
 - *Integer.* Almacenan números enteros.

A continuación se muestra la cantidad, descripción, nombre tipo y otros atributos que conforman la base de datos *bd_sistec*. Cabe destacar que todas las tablas contendrán

los campos especiales de auditoría, a excepción de la tabla *tbl_prioridad*, *tbl_bitacora*, *tbl_departamento_pais* y *tbl_detalle_hoja*.

Tabla	Columnas	Tipo	Descripción
tbl_bitacora	4	PostgreSQL	Almacenará la información de las operaciones realizadas en las tablas.
tbl_departamento_pais	2	PostgreSQL	Almacenará la lista de departamentos del país.
tbl_descarte	9	PostgreSQL	Almacenará las hojas de descarte emitidas por el Departamento.
tbl_detalle_hoja	4	PostgreSQL	Almacenará los detalles de insumos utilizados para los servicios requeridos.
tbl_equipos	9	PostgreSQL	Contendrá la información de los equipos informáticos del MINSAL.
tbl_establecimiento	8	PostgreSQL	Almacenará información acerca de las Unidades de Salud a cargo del Departamento.
tbl_hoja_servicio	13	PostgreSQL	Se utilizará para guardar toda la información referente a los servicios que se realicen.

tbl_insumos	12	PostgreSQL	Almacena los datos de los insumos utilizados por el Departamento.
tbl_mov_insumos	11	PostgreSQL	Contendrá información del inventario de los insumos del Departamento.
tbl_prioridad	3	PostgreSQL	Guarda la información de las prioridades en la solicitud manejadas por el Departamento.
tbl_solicitud	8	PostgreSQL	Almacena la información de la solicitud enviada por el usuario jefe de unidad.
tbl_tipoEquipos	7	PostgreSQL	Contendrá la información de los tipos de equipos que supervisa el Departamento.
tbl_tipo_insumo	7	PostgreSQL	Guarda la información de los tipos de insumos que maneja el Departamento.
tbl_tipo_mov	9	PostgreSQL	Contendrá los tipos de movimientos que se le puede aplicar a los insumos para su gestión de inventarios.
tbl_tipo_servicio	8	PostgreSQL	Almacenará los tipos de servicios que se brindan en el Departamento.

tbl_usuarios	14	PostgreSQL	Guarda la información de los usuarios del sistema.
tbl_visitas	10	PostgreSQL	Almacenará la información referente a las visitas que se realicen fuera de la DRSO.
17 tablas	138	–	–

Cuadro 28

Descripción de tablas de la base de datos db_sistec

Tabla:		tbl_bitacora			
Descripción:		Almacenará la información de las operaciones realizadas en las tablas.			
Columna	Tipo	Nulo	Único	Descripción	Key
fecha	timestamp	No	No	Almacenará la fecha y hora de la operación.	
tipo_operacion	varchar(15)	No	No	Almacena el tipo de operación que se realiza.	
tabla	varchar(25)	No	No	Guarda el nombre de la tabla que se opera.	
id_usuario	integer	No	No	Referencia a la tabla tbl_usuario.	FK

Cuadro 29

Descripción de tabla tbl_bitacora.

Tabla:		tbl_departamento_pais			
Descripción:		Almacenará la información de los departamentos del país.			
Columna	Tipo	Nulo	Único	Descripción	Key
id	integer	No	Si	Almacenará el id del registro.	PK
nombre	varchar(50)	No	Si	Almacena el nombre del departamento.	

Cuadro 30

Descripción de tabla tbl_departamento_pais.

Tabla:		tbl_descarte			
Descripción:		Almacenará las hojas de descarte emitidas por el Departamento.			
Columna	Tipo	Nulo	Único	Descripción	Key
id	integer	No	Si	Almacenará el id del registro.	PK
id_equipo	integer	No	No	Funciona como referencia a la tabla tbl_equipos	FK
fecha	date	No	No	Almacena la fecha del descarte.	
descripción	varchar(500)	No	No	Guardarán una descripción del descarte.	

estado	boolean	No	No	Guardaré si el descarte se encuentra activo o no.	
usuario_creacion	integer	No	No	Guardaré el id del usuario que crea el registro.	
fecha_creacion	timestamp	No	No	Guardaré la fecha y hora de creación del registro.	
usuario_actualizar	integer	No	No	Guardaré el id del usuario que edita el registro.	
fecha_actualizar	timestamp	No	No	Guardaré la fecha y hora de modificación del registro.	

Cuadro 31

Descripción de tabla tbl_descarte.

Tabla:		tbl_detalle_hoja			
Descripción:		Almacenará los detalles de insumos utilizados para los servicios requeridos.			
Columna	Tipo	Nulo	Único	Descripción	Key
id	integer	No	Si	Almacenará el id del registro.	PK

id_hoja_servicio	integer	No	No	Funciona como referencia a la tabla tbl_hoja_servicio.	FK
cantidad	integer	No	No	Almacena la cantidad del insumo requerido.	
id_insumo	integer	No	No	Referencia a la tabla tbl_insumos.	FK

Cuadro 32

Descripción de tabla tbl_detalle_hoja.

Tabla:		tbl_equipos			
Descripción:		Contendrá la información de los equipos informáticos del MINSAL.			
Columna	Tipo	Nulo	Único	Descripción	Key
id	integer	No	Si	Almacenará el id del registro.	PK
num_inventario	varchar(50)	No	Si	Almacena el número de inventario del equipo.	
id_tipo_equipo	integer	Si	No	Referencia a la tabla tbl_tipo_equipo.	FK
id_solicitud	integer	Si	No	Referencia a la tabla tbl_solicitud.	FK
estado	boolean	No	No	Almacena si el registro está activo o no.	

usuario_creacion	integer	No	No	Guardaré el id del usuario que crea el registro.	
fecha_creacion	timestamp	No	No	Guardaré la fecha y hora de creación del registro.	
usuario_actualizar	integer	No	No	Guardaré el id del usuario que edita el registro.	
fecha_actualizar	timestamp	No	No	Guardaré la fecha y hora de modificación del registro.	

Cuadro 33

Descripción de tabla tbl_equipos.

Tabla:		tbl_establecimiento			
Descripción:		Almacenará información acerca de las Unidades de Salud a cargo del Departamento.			
Columna	Tipo	Nulo	Único	Descripción	Key
id	integer	No	Si	Almacenará el id del registro.	PK
id_prioridad	integer	No	No	Referencia a la tabla tbl_prioridad.	FK
direccion	varchar(255)	No	No	Almacena la dirección del establecimiento.	

estado	boolean	No	No	Almacena si el registro está activo o no.	
usuario_creacion	integer	No	No	Guardaré el id del usuario que crea el registro.	
fecha_creacion	timestamp	No	No	Guardaré la fecha y hora de creación del registro.	
usuario_actualizar	integer	No	No	Guardaré el id del usuario que edita el registro.	
fecha_actualizar	timestamp	No	No	Guardaré la fecha y hora de modificación del registro.	

Cuadro 34

Descripción de tabla tbl_establecimiento.

Tabla:		tbl_hoja_servicio			
Descripción:		Almacenará la información de las hojas de trabajos que realicen.			
Columna	Tipo	Nulo	Único	Descripción	Key
id	integer	No	Si	Almacenará el id del registro.	PK
id_solicitud	integer	No	No	Referencia a la tabla tbl_solicitud	FK

id_visitas	integer	No	No	Referencia a la tabla tbl_visitas	FK
descripcion	varchar(500)	No	No	Guarda la información adicional de la hoja de trabajo.	
id_usuario	integer	No	No	Referencia a la tabla tbl_usuario	FK
fecha	date	No	No	Guardará la fecha de la hoja de trabajo.	
id_tipo_servicio	integer	No	No	Referencia a la tabla tbl_tipo_servicio.	FK
estado	boolean	No	No	Guardará si la hoja de servicio se encuentra activa o no.	
estado_peticion	integer	No	No	Tendrá como valores - 1: Recibido - 2: En proceso - 3: Finalizado - 4: Cancelado - 5: Equipo descartado	
usuario_creacion	integer	No	No	Guardará el id del usuario que crea el registro.	
fecha_creacion	timestamp	No	No	Guardará la fecha y hora de creación del registro.	

usuario_actualizar	integer	No	No	Guardaré el id del usuario que edita el registro.	
fecha_actualizar	timestamp	No	No	Guardaré la fecha y hora de modificación del registro.	

Cuadro 35

Descripción de tabla tbl_hoja_servicio.

Tabla:		tbl_insumos			
Descripción:		Almacenará los datos de los insumos utilizados por el Departamento.			
Columna	Tipo	Nulo	Único	Descripción	Key
id	integer	No	Si	Almacenará el id del registro.	PK
nombre_insumo	varchar(50)	No	No	Guardaré el nombre del insumo.	
id_tipo_insumo	integer	No	No	Referencia a la tabla tbl_tipo_insumo.	FK
modelo	varchar(50)	No	No	Guarda la información del modelo del equipo.	
capacidad	varchar(50)	No	No	Guarda la información de la capacidad del equipo.	
marca	varchar(50)	No	No	Guardaré la marca del equipo.	

cantidad	integer	No	No	Almacenará el stock del insumo.	
estado	boolean	No	No	Guardará si el insumo se encuentra activo o no.	
usuario_creacion	integer	No	No	Guardará el id del usuario que crea el registro.	
fecha_creacion	timestamp	No	No	Guardará la fecha y hora de creación del registro.	
usuario_actualizar	integer	No	No	Guardará el id del usuario que edita el registro.	
fecha_actualizar	timestamp	No	No	Guardará la fecha y hora de modificación del registro.	

Cuadro 36

Descripción de tabla tbl_insumos.

Tabla:		tbl_mov_insumos			
Descripción:		Contendrá información del inventario de los insumos del Departamento.			
Columna	Tipo	Nulo	Único	Descripción	Key
id	integer	No	Si	Almacenará el id del registro.	PK

fecha	date	No	No	Guardaré la fecha del movimiento.	
id_tipo_mov	integer	No	No	Referencia a la tabla tbl_tipo_mov	FK
descripción	varchar(500)	Si	No	Guarda la información adicional del movimiento.	
cant_mov	integer	No	No	Guarda la cantidad del movimiento.	
id_insumos	integer	No	No	Referencia a la tabla tbl_insumos	FK
estado	boolean	No	No	Guardaré si el movimiento se encuentra activo o no.	
usuario_creacion	integer	No	No	Guardaré el id del usuario que crea el registro.	
fecha_creacion	timestamp	No	No	Guardaré la fecha y hora de creación del registro.	
usuario_actualizar	integer	No	No	Guardaré el id del usuario que edita el registro.	
fecha_actualizar	timestamp	No	No	Guardaré la fecha y hora de modificación del registro.	

Cuadro 37

Descripción de tabla tbl_mov_insumos.

Tabla:		tbl_prioridad			
Descripción:		Contendrá información de la prioridad en las solicitudes manejadas por el Departamento.			
Columna	Tipo	Nulo	Único	Descripción	Key
id	integer	No	Si	Almacenará el id del registro.	PK
nombre	varchar(50)	No	Si	Guardará el nombre de la prioridad.	
estado	boolean	No	No	Guardará si la prioridad se encuentra activa o no.	

Cuadro 38

Descripción de tabla tbl_prioridad.

Tabla:		tbl_solicitud			
Descripción:		Contendrá información de la solicitud enviada por el usuario jefe de unidad.			
Columna	Tipo	Nulo	Único	Descripción	Key
id	integer	No	Si	Almacenará el id del registro.	PK
id_tipo_equipo	integer	No	No	Referencia a la tabla tbl_tipo_equipo	FK

descripcion	varchar(500)	No	No	Guardará la información adicional de la solicitud.	
estado	boolean	No	No	Guardará si la solicitud se encuentra activa o no.	
usuario_creacion	integer	No	No	Guardará el id del usuario que crea el registro.	
fecha_creacion	timestamp	No	No	Guardará la fecha y hora de creación del registro.	
usuario_actualizar	integer	No	No	Guardará el id del usuario que edita el registro.	
fecha_actualizar	timestamp	No	No	Guardará la fecha y hora de modificación del registro.	

Cuadro 39

Descripción de tabla tbl_solicitud.

Tabla:		tbl_tipo_equipos			
Descripción:		Contendrá información de los tipos de equipos que supervisa el Departamento.			
Columna	Tipo	Nulo	Único	Descripción	Key

id	integer	No	Si	Almacenará el id del registro.	PK
nombre	varchar(50)	No	Si	Almacena el nombre del tipo de equipo.	
estado	boolean	No	No	Guardarás si la solicitud se encuentra activa o no.	
usuario_creacion	integer	No	No	Guardarás el id del usuario que crea el registro.	
fecha_creacion	timestamp	No	No	Guardarás la fecha y hora de creación del registro.	
usuario_actualizar	integer	No	No	Guardarás el id del usuario que edita el registro.	
fecha_actualizar	timestamp	No	No	Guardarás la fecha y hora de modificación del registro.	

Cuadro 40

Descripción de tabla tbl_tipo_equipos.

Tabla:		tbl_tipo_insumo			
Descripción:		Guarda la información de los tipos de insumos que maneja el Departamento.			
Columna	Tipo	Nulo	Único	Descripción	Key

id	integer	No	Si	Almacenará el id del registro.	PK
nombre	varchar(50)	No	Si	Almacena el nombre del tipo de insumo.	
estado	boolean	No	No	Guardará si el tipo se encuentra activo o no.	
usuario_creacion	integer	No	No	Guardará el id del usuario que crea el registro.	
fecha_creacion	timestamp	No	No	Guardará la fecha y hora de creación del registro.	
usuario_actualizar	integer	No	No	Guardará el id del usuario que edita el registro.	
fecha_actualizar	timestamp	No	No	Guardará la fecha y hora de modificación del registro.	

Cuadro 41

Descripción de tabla tbl_tipo_insumo.

Tabla:	tbl_tipo_mov
Descripción:	Contendrá los tipos de movimientos que se le pueden aplicar a los insumos para gestión de inventarios.

Columna	Tipo	Nulo	Único	Descripción	Key
id	integer	No	Si	Almacenará el id del registro.	PK
tipo_mov	integer	No	No	Guardará el tipo de movimiento con los valores: - 1: Entrada - 2: Salida	
nombre	varchar(50)	No	Si	Almacena el nombre del tipo de movimiento.	
descripcion	varchar(500)	Si	No	Almacena información adicional del movimiento.	
estado	boolean	No	No	Guardará si el tipo se encuentra activo o no.	
usuario_creacion	integer	No	No	Guardará el id del usuario que crea el registro.	
fecha_creacion	timestamp	No	No	Guardará la fecha y hora de creación del registro.	
usuario_actualizar	integer	No	No	Guardará el id del usuario que edita el registro.	

fecha_actualizar	timestamp	No	No	Guardaré la fecha y hora de modificación del registro.	
------------------	-----------	----	----	--------------------------------------------------------	--

Cuadro 42

Descripción de tabla tbl_tipo_mov.

Tabla:		tbl_tipo_servicio			
Descripción:		Contendrá los tipos de servicios que se brindan en el Departamento.			
Columna	Tipo	Nulo	Único	Descripción	Key
id	integer	No	Si	Almacenará el id del registro.	PK
nombre	varchar(50)	No	Si	Almacena el nombre del tipo de servicio.	
descripcion	varchar(500)	Si	No	Almacena información adicional del servicio.	
estado	boolean	No	No	Guardaré si el tipo se encuentra activo o no.	
usuario_creacion	integer	No	No	Guardaré el id del usuario que crea el registro.	
fecha_creacion	timestamp	No	No	Guardaré la fecha y hora de creación del registro.	

usuario_actualizar	integer	No	No	Guardaré el id del usuario que edita el registro.	
fecha_actualizar	timestamp	No	No	Guardaré la fecha y hora de modificación del registro.	

Cuadro 43

Descripción de tabla tbl_tipo_servicio.

Tabla:		tbl_usuarios			
Descripción:		Guarda la información de los usuarios del sistema.			
Columna	Tipo	Nulo	Único	Descripción	Key
id	integer	No	Si	Almacenará el id del registro.	PK
nombre	varchar(50)	No	No	Almacena el nombre del usuario.	
apellido	varchar(50)	No	No	Almacena el apellido del usuario.	
telefono	varchar(12)	No	No	Almacena el teléfono del usuario.	
correo	varchar(50)	No	Si	Almacena el correo del usuario.	
pass	varchar(255)	No	No	Guarda la contraseña del usuario.	

id_rol	integer	No	No	Guarda el tipo de usuario con los valores: - 1: Administrador - 2: Técnico - 3: Jefe de unidad	
id_establecimiento	integer	No	No	Referencia a la tabla tbl_establecimiento	
token	varchar(255)	No	Si	Almacena el token del usuario.	
estado	boolean	No	No	Guardará si el tipo se encuentra activo o no.	
usuario_creacion	integer	No	No	Guardará el id del usuario que crea el registro.	
fecha_creacion	timestamp	No	No	Guardará la fecha y hora de creación del registro.	
usuario_actualizar	integer	No	No	Guardará el id del usuario que edita el registro.	
fecha_actualizar	timestamp	No	No	Guardará la fecha y hora de modificación del registro.	

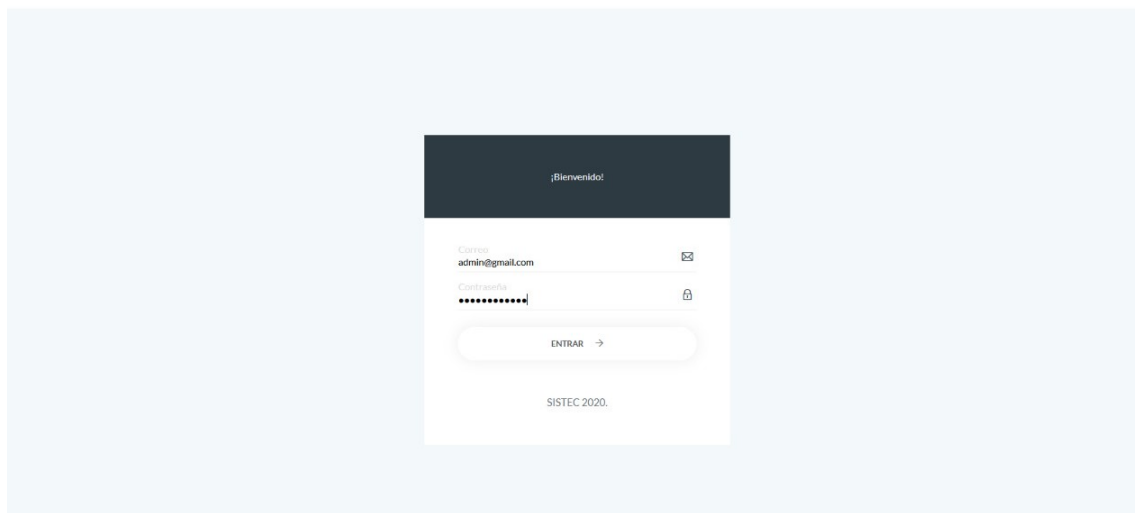
Cuadro 44*Descripción de tabla tbl_usuarios.*

Tabla:		tbl_visitas			
Descripción:		Guarda la información de las visitas que se realicen fuera de la DRSO.			
Columna	Tipo	Nulo	Único	Descripción	Key
id	integer	No	Si	Almacenará el id del registro.	PK
nombre	varchar(50)	No	No	Almacena el nombre de la visita.	
id_establecimiento	integer	No	No	Referencia a la tabla tbl_establecimiento.	FK
descripcion	varchar(500)	Si	No	Almacena información adicional de la visita.	
id_usuario	integer	No	No	Referencia a la tabla tbl_usuarios.	FK
estado	boolean	No	No	Guardará si la visita se encuentra activa o no.	
usuario_creacion	integer	No	No	Guardará el id del usuario que crea el registro.	
fecha_creacion	timestamp	No	No	Guardará la fecha y hora de creación del registro.	

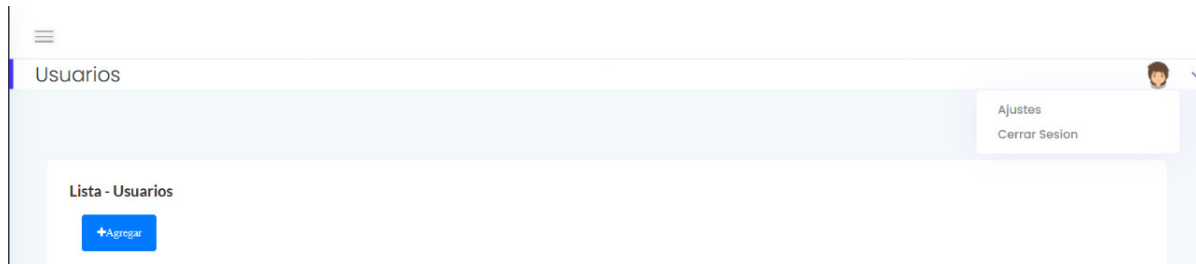
usuario_actualizar	integer	No	No	Guardará el id del usuario que edita el registro.
fecha_actualizar	timestamp	No	No	Guardará la fecha y hora de modificación del registro.

Cuadro 45

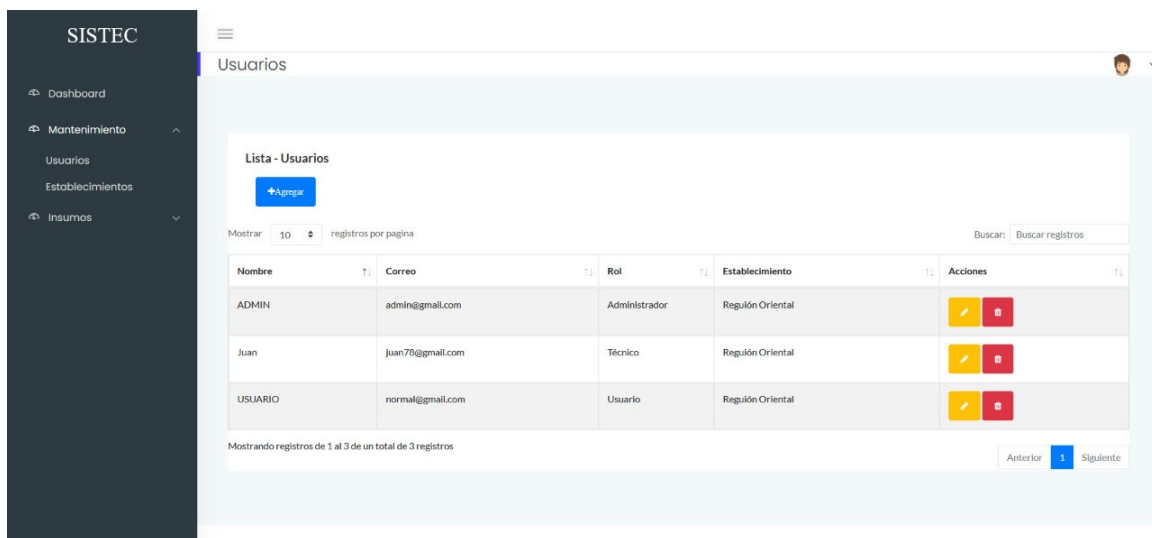
Descripción de tabla tbl_visitas.

Diseño de interfaces***Sistema web*****Figura 42**

Vista de login de la aplicación web.

**Figura 43**

Vista de logout de la aplicación web.

**Figura 44**

Vista de usuarios de la aplicación web.

The screenshot displays the 'Establecimientos' (Establishments) page. On the left is a dark sidebar with the 'SISTEC' logo and navigation links for 'Dashboard', 'Mantenimiento', and 'Insumos'. The main content area has a header 'Establecimientos' and a user profile icon. Below this is a section titled 'Lista - Establecimientos' containing a '+ Agregar' button, a 'Mostrar: 10 registros por página' dropdown, and a search bar labeled 'Buscar: Buscar registros'. A table with columns 'Id', 'Nombre', 'Direccion', 'Prioridad', and 'Acciones' contains one row: '2', 'Región Oriental', 'San Miguel', 'Alta', and edit/delete icons. At the bottom, it says 'Mostrando registros de 1 al 1 de un total de 1 registros' and has 'Anterior' and 'Siguiente' buttons.

Figura 45
Vista de establecimientos de la aplicación web.

The screenshot displays the 'Agregar Tipo de Movimiento' (Add Movement Type) page. It features the same sidebar as Figure 45. The main content area has a header 'Agregar Tipo de Movimiento' and a user profile icon. Below is a form with three input fields: 'Nombre' with the value 'Donación', 'Tipo de Movimiento' with a dropdown menu showing 'Entrada', and 'descripcion' with the value 'Insumo donado por organización sin fines de lucro'. At the bottom right, there are 'Close' and 'Guardar' (Save) buttons.

Figura 46
Vista de agregar movimiento de la aplicación web.

The screenshot shows the 'Agregar Servicio' (Add Service) form in the SISTEC application. The form is located on the right side of the page, with a dark sidebar on the left containing the application logo and navigation menu. The form fields are as follows:

- Nombre de usuario: Javier Alexis
- Apellido: Hernandez Garcia
- Correo: alexG@gmail.com
- Password: [Redacted]
- Rol: ADMINISTRADOR
- Establecimiento: Reusión Oriental

At the bottom right of the form, there are two buttons: 'Close' and 'Guardar' (Save).

Figura 47

Vista de agregar servicio de la aplicación web.

The screenshot shows the 'Tipo de insumos' (Types of Supplies) list in the SISTEC application. The table displays two records with the following columns: Id, Nombre, Usuario, Fecha de creación, Usuario actualización, Fecha de actualización, and Acciones. The data is as follows:

Id	Nombre	Usuario	Fecha de creación	Usuario actualización	Fecha de actualización	Acciones
10	Limpiador de contacto	Admin	2020-10-23	Admin	2020-10-23	[Edit] [Delete]
11	Pasta térmica	Admin	2020-10-23	Admin	2020-10-23	[Edit] [Delete]

At the bottom of the table, it says 'Mostrando registros de 1 al 2 de un total de 2 registros'. There are also navigation buttons: 'Anterior', '1', and 'Siguiente'.

Figura 48

Vista de agregar tipo de insumo de la aplicación web.

Aplicación móvil**Figura 49***Vista de inicio de la aplicación móvil.*



Figura 50
Vista de login de la aplicación móvil.

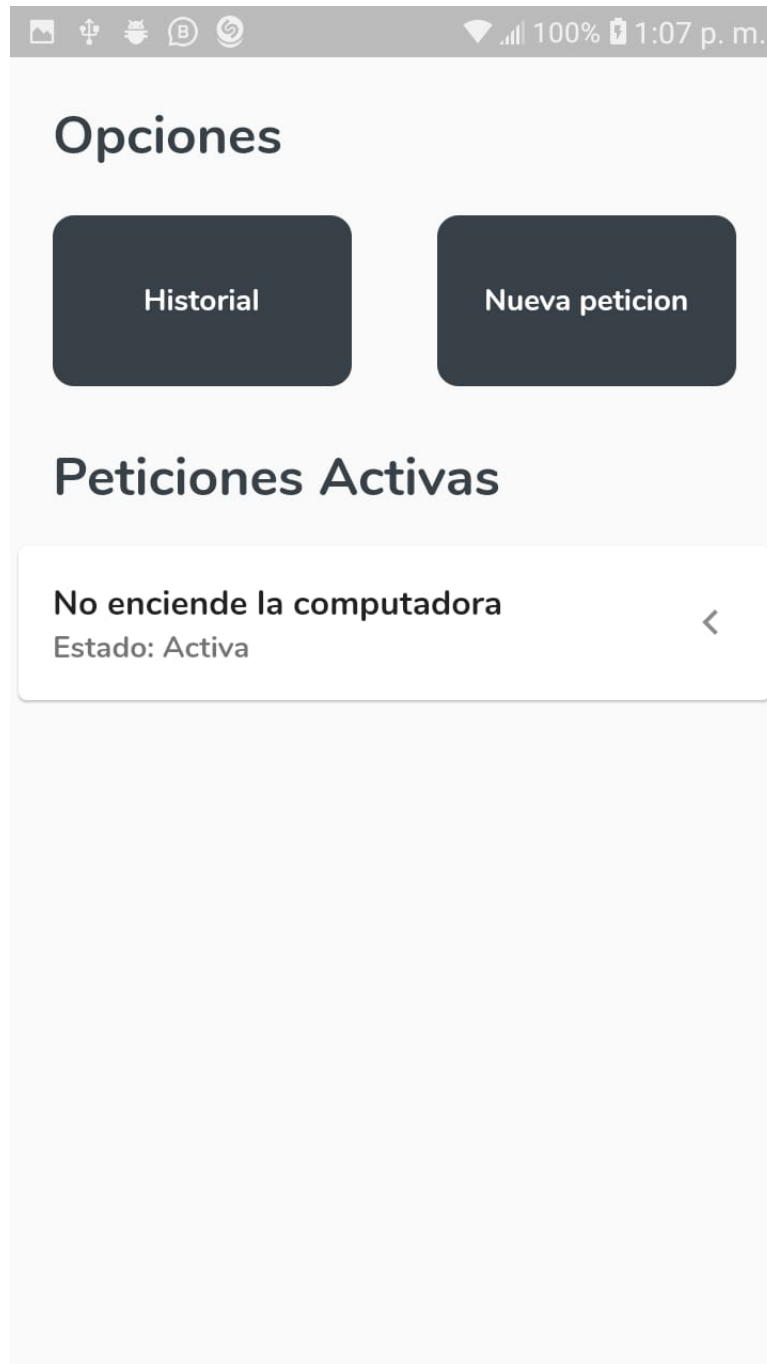


Figura 51

Vista de lista de solicitud de la aplicación móvil.

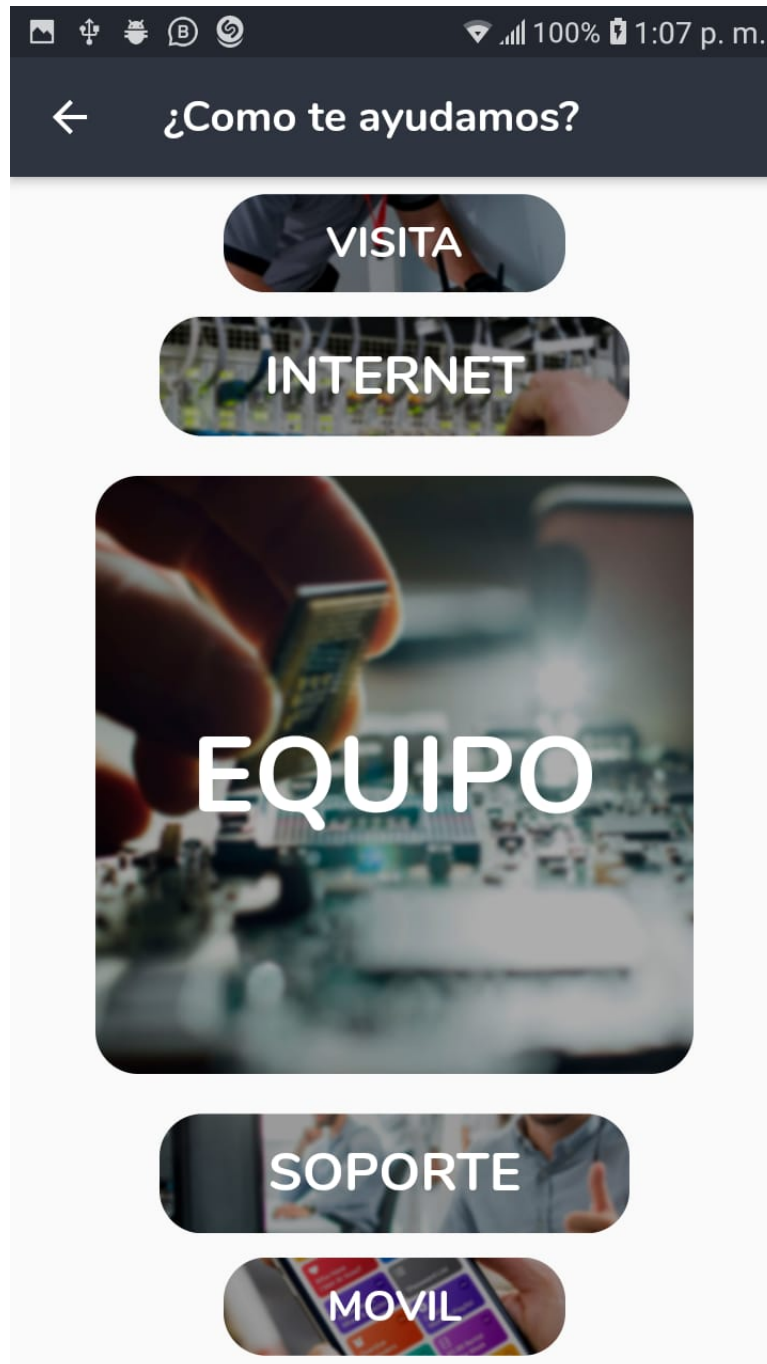



Figura 52
Vista de tipo de solicitud de la aplicación móvil.

Tipo de petición ▼

Detalle su petición 

Ingresar

Figura 53

Vista de agregar solicitud de la aplicación móvil.

Programación del sistema informático

Symfony v2.8

Como mencionamos anteriormente, Symfony es un framework de arquitectura MVC¹³ para construir aplicaciones web con PHP. Este tiene un enorme conjunto de herramientas y utilidades que simplifican el desarrollo de las aplicaciones web (Symfony, 2020).

Estándares de programación. Symfony es uno de los framework más famosos y robustos que existe para el desarrollo de aplicaciones. Por lo tanto, los estándares de programación a utilizar serán los mismos que la herramienta solicita en su documentación (Symfony, 2020):

- **Variables.** Las variables deberán ser de nombre cortos, comenzar en letra minúscula, y deberán ser en formato *camelCase*, es decir, se separa cada palabra con una letra mayúscula. Ejemplo, *nombreVariable*.
- **Constantes.** Éstas deberán de ir en mayúsculas, con guión bajo para separar las palabras. Ejemplo, *EJEMPLO_CONSTANTE*.
- **Funciones.** Symfony dicta que las funciones de acción deberán tener el sufijo *Action*, además de respetar el formato *camelCase*. Ejemplo, *editAction()*.
- **URLs.** En Symfony, se puede modificar con libertad el nombre de las URL a la que apunta el controlador. Para tal caso, se manejarán de la siguiente forma:
 - *index*, que será la vista index o la vista home de cada módulo.
 - *add*, que lleva al formulario de crear un nuevo registro para cualquier módulo.
 - *create*, que recibirá los datos por POST para la creación del registro correspondiente.
 - *edit*, que llevará al formulario para editar el registro.

¹³ Modelo, Vista, Controlador.

- *update*, que recibirá los datos por POST para la edición del registro.
- *delete*, que en algunos casos, se utilizará para eliminar un registro, y en otros, para hacer un update del campo *activo* del registro seleccionado.

Configurando el proyecto. Para configurar el framework Symfony en su versión 2.8, se detalla la estructura del proyecto:

Se extrae el contenido del proyecto y obtendríamos una carpeta con el sistema web que estaría estructurada de la manera en que se muestra en la Figura 54.

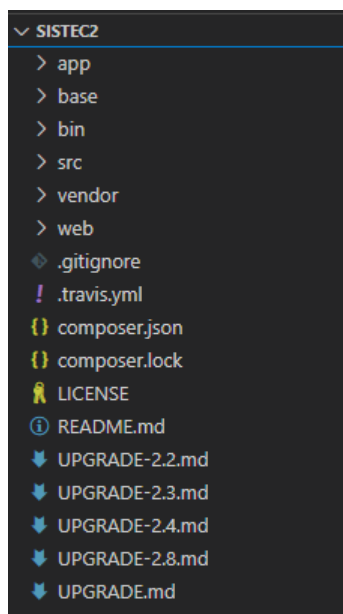
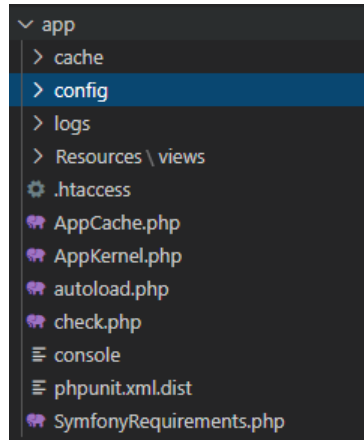


Figura 54
Carpeta de proyecto Symfony v2.8.

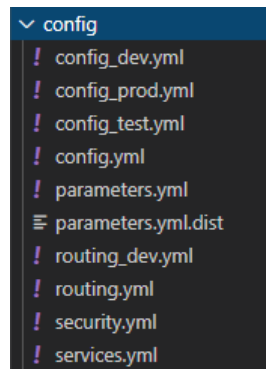
Como se observa, se despliega una serie de carpetas y archivos que conforman todo el proyecto, sin embargo las que vamos a utilizar son las carpetas *app*, *base*, *src* y *web*.

Carpeta *app*. Es donde se guarda la configuración básica del sistema y contiene la estructura que muestra la Figura 55.

**Figura 55**

Carpeta app del proyecto Symfony v2.8.

Dentro de esta carpeta, se encuentran dos más que son importantes, que son *config* y *resources*. La carpeta *config* tiene la estructura de la Figura 56.

**Figura 56**

Carpeta config del proyecto Symfony v2.8.

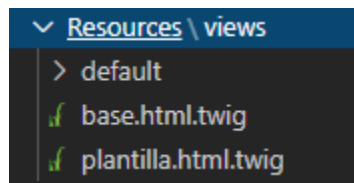
Y dentro de ella, se encuentra la carpeta *parameters.yml*, que es donde se configuran las variables para la conexión a la base de datos, como el host, puerto, nombre de la base de datos, usuario y contraseña de la base de datos. Su contenido se muestra más detalladamente en la Figura 57.

```
# This file is auto-generated during the composer install
parameters:
  database_host:
  database_port: null
  database_name:
  database_user:
  database_password:
  mailer_transport: smtp
  mailer_host: 127.0.0.1
  mailer_user: null
  mailer_password: null
  secret: ThisTokenIsNotSoSecretChangeIt
```

Figura 57

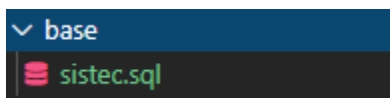
Archivo parameters del proyecto Symfony v2.8.

La carpeta *resources* contiene la estructura de archivos que muestra la Figura 58. Dentro de ella, en la carpeta *views*, se encuentran todas las vistas que se utilizarán en el proyecto con la extensión *.twig*, que es el motor de plantillas HTML que utiliza Symfony por defecto.

**Figura 58**

Carpeta resources del proyecto Symfony v2.8.

Carpeta *base*. Esta carpeta se utiliza para almacenar la base de datos de respaldo que utiliza el sistema para funcionar. Esta carpeta no es nativa de Symfony, sin embargo se creó para gestionar de una manera ordenada los archivos de base de datos. Su estructura se muestra en la Figura 59.

**Figura 59**

Carpeta base del proyecto Symfony v2.8.

Carpeta *src*. Esta carpeta almacena toda la estructura del sistema informático. Contiene las entidades, vistas y controladores específicos del proyecto. También guarda librerías y archivos que contienen información para el funcionamiento del sistema. Su estructura se muestra en la Figura 60.

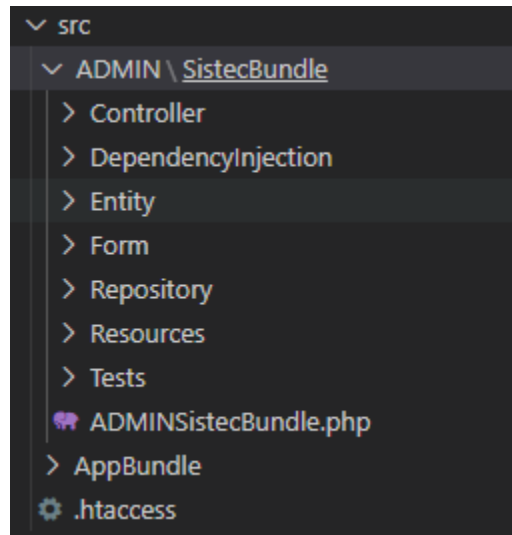


Figura 60

Carpeta src del proyecto Symfony v2.8.

Dentro de ella se encuentran los Bundles del sistema, en los que se encuentran las siguientes carpetas que son utilizadas de forma directa para el desarrollo:

- *Controller*: Es donde se almacenan los controladores del sistema.
- *Entity*: En ella se encuentran las entidades del sistema que se comunican con la base de datos.
- *Form*: Es donde se guardan los formularios generados por Symfony que utiliza el sistema.
- *Repository*: Es donde se almacenan las consultas complejas a la base de datos para controladores que lo necesitan.
- *Resources*: Se encuentran todas las vistas del sistema web. Tienen extensión *.html.twig*.

Carpeta *web*. Se encuentran almacenados la carpeta *public*, que contiene los assets que utiliza el sistema web. Guarda los archivos con las extensiones *.js*, *.css*, imágenes y fuentes. Su estructura se muestra en la Figura 61.

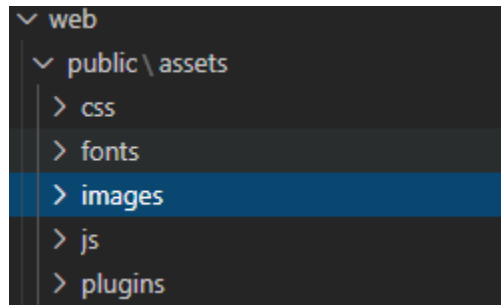


Figura 61

Carpeta web del proyecto Symphony v2.8.

Flutter

Configurando el proyecto. Flutter es el framework que se utilizará para el desarrollo de la aplicación móvil. El proyecto se compone de la carpeta raíz descrita en la Figura 62.

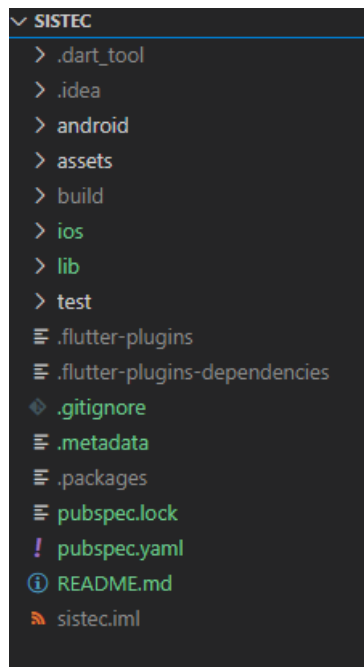


Figura 62

Carpeta del proyecto Flutter.

Dentro de esta carpeta se encuentra una serie de archivos y otras carpetas que conforman la aplicación. Pero, para el proyecto son indispensables dos carpetas y archivo, que son los que se describen a continuación.

Carpeta *assets*. La carpeta *assets* está conformada por los archivos externos que utiliza la aplicación para funcionar, por ejemplo las imágenes y fuentes. Su estructura de archivos se muestra en la Figura 63.



Figura 63

Carpeta assets del proyecto Flutter.

Carpeta *lib*. En esta carpeta se encuentra el archivo raíz de la aplicación *main.dart*; y su estructura de archivos se muestra en la Figura 64. La carpeta *src* que contiene las diferentes interfaces que posee la aplicación para sus operaciones internas.

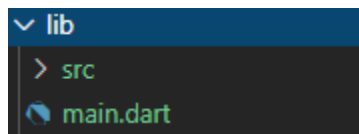


Figura 64

Carpeta lib del proyecto Flutter.

Carpeta *src*. En esta carpeta se encuentra una estructura de archivos mostrada en la Figura 65.

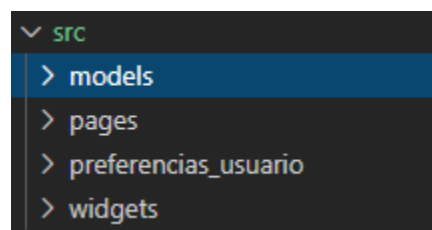


Figura 65

Carpeta src del proyecto Flutter.

- *models*. Contiene las peticiones que se realizan al servidor para hacer las operaciones con la base de datos.
- *pages*. Contiene todas las vistas que posee la aplicación, tanto para el usuario estándar como para el usuario técnico o administrador.
- *preferencias_usuario*. Contiene los procesos que almacenan en la memoria interna del dispositivo, estos son datos de vital importancia, para hacer las operaciones de la aplicación con el servidor.
- *widgets*. Contiene elementos visuales específicos que se utilizan dentro de algunas vistas de aplicación.

Archivo *pubspec.yaml*. En este archivo se realizan importaciones de los *assets*, librerías y cualquier archivo externo a Flutter que necesite la aplicación. Su estructura se presenta en la Figura 66.

```
! pubspec.yaml
1  name: sistec
2  description: A new Flutter project.
3
4
5  version: 1.0.0+1
6
7  environment:
8    sdk: ">=2.1.0 <3.0.0"
9
10 dependencies:
11   flutter:
12     sdk: flutter
13
14   cupertino_icons: ^0.1.2
15   vertical_card_pager: ^1.4.0
16   http:
17   flutter_slidable: ^0.5.7
18   toast: ^0.1.5
19   shared_preferences:
20
21 dev_dependencies:
22   flutter_test:
23     sdk: flutter
24
25
26
27 flutter:
28
29   uses-material-design: true
30   assets:
31     - assets/images/
32
33
34
35   fonts:
36     - family: Nunito
37       fonts:
38         - asset: assets/fonts/Nunito-Bold.ttf
39
```

Figura 66

Archivo pubspec.yaml del proyecto Flutter.

Pruebas

Las pruebas de software comprenden un conjunto de actividades que se realizan para identificar posibles fallas en el funcionamiento, configuración o usabilidad del sistema informático por medio de pruebas sobre el comportamiento de este.

Tipos de pruebas

Las pruebas que se realizarán al sistema son:

- *Pruebas unitarias:* prueban el diseño y comportamiento de cada componente del sistema una vez construido. Las pruebas de unidad es un proceso para probar los subprogramas, subrutinas, los procedimientos individuales o las funciones, es decir, se deben probar primero los bloques desarrollados, e inicialmente probar el software en su totalidad.
- *Pruebas de integración:* prueban la correcta relación entre los componentes del sistema a través de sus interfaces y si ellas cumplen con la funcionalidad establecida. Consiste en unir el sistema a partir de los distintos componentes y probarlos con todos integralmente. Estas pruebas se realizan de forma progresiva.
- *Pruebas de aceptación:* evalúan que el sistema cumple con todos los requisitos indicados y permite que los usuarios del sistema provean su aceptación.
- *Pruebas de regresión:* el objetivo es comprobar que los cambios sobre un componente del sistema no generan errores adicionales en otros componentes no modificados.
- *Pruebas de carga:* tienen como objetivo saber cuánta carga puede soportar un sistema sin empeorar su rendimiento. Estas pruebas sirven para conocer la capacidad máxima de un sistema bajo una carga determinada y poder dar con las causas que puedan condicionar su rendimiento.

Resultados de las pruebas

Pruebas unitarias en la aplicación web. Las pruebas unitarias son las que se realizan para cada ponente en forma individual. Por lo que se procederá a efectuar las pruebas por es tester para los componentes principales de la aplicación web.

Login. El login es la vista principal de la aplicación web, donde el usuario pasa por una autenticación y autorización, procesos descritos en secciones anteriores en este documento. La prueba en este módulo consiste en ingresar datos correctos y datos incorrectos.

En la Figura 67, podemos ver una autenticación de usuario completamente correcta. Esto es así al ingresar un usuario y contraseña válidos, y nos redirigirá a la vista *Dashboard* de la aplicación web.

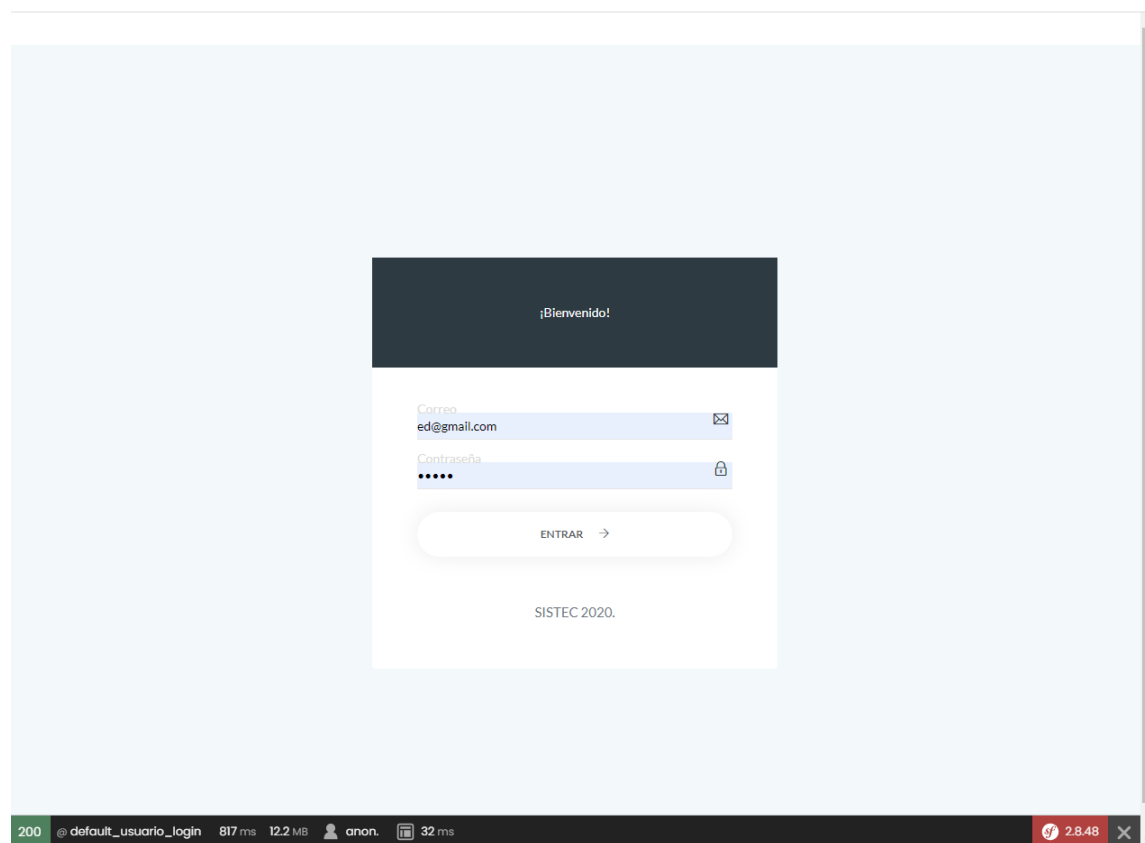


Figura 67

Vista login con datos correctos.

De igual manera, si se intenta ingresar un usuario que no existe en la base de datos, nos mostrará un error diciendo que las credenciales son inválidas y no se ingresará al sistema web, tal como se muestra en la Figura 68.

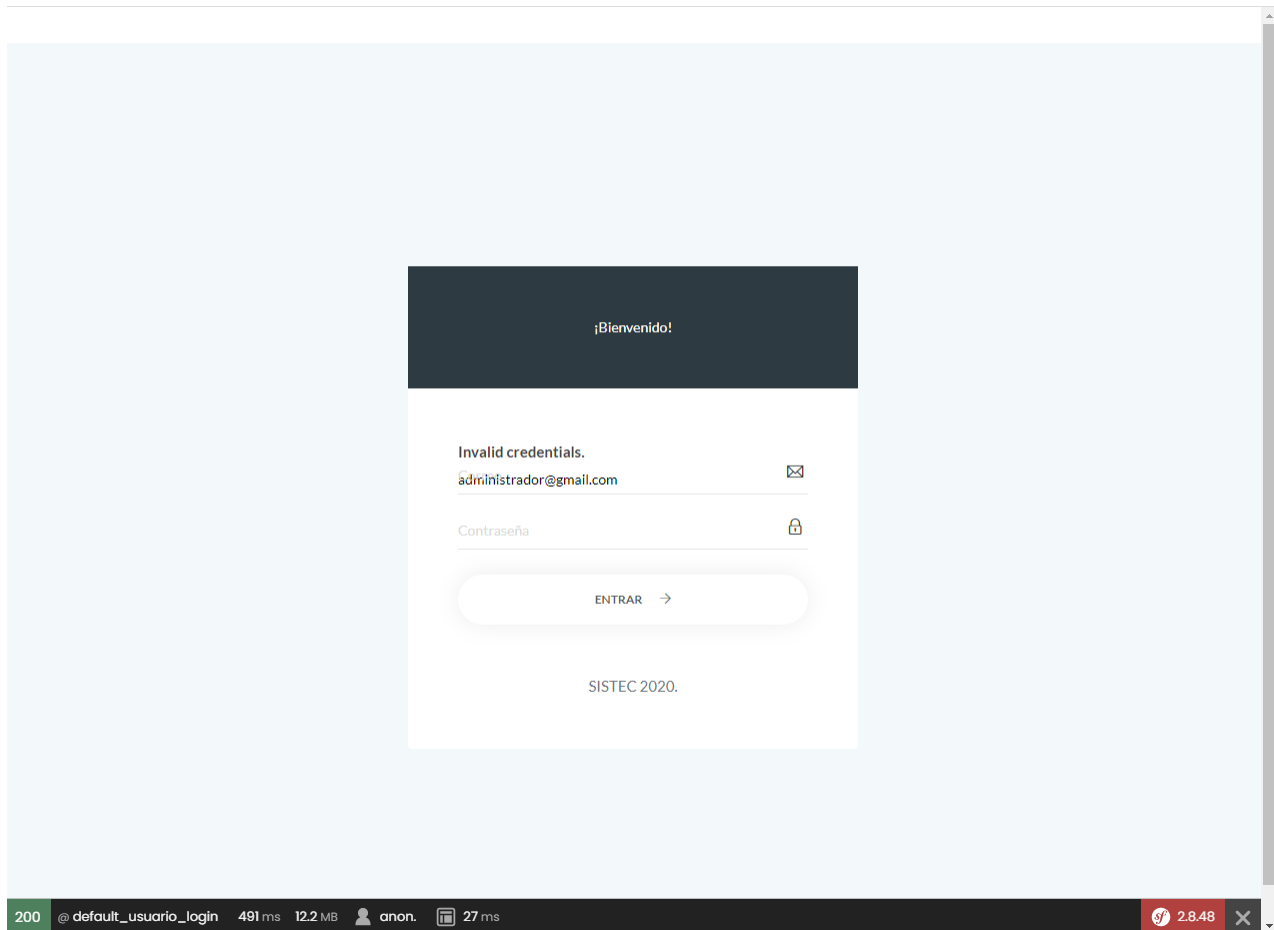


Figura 68

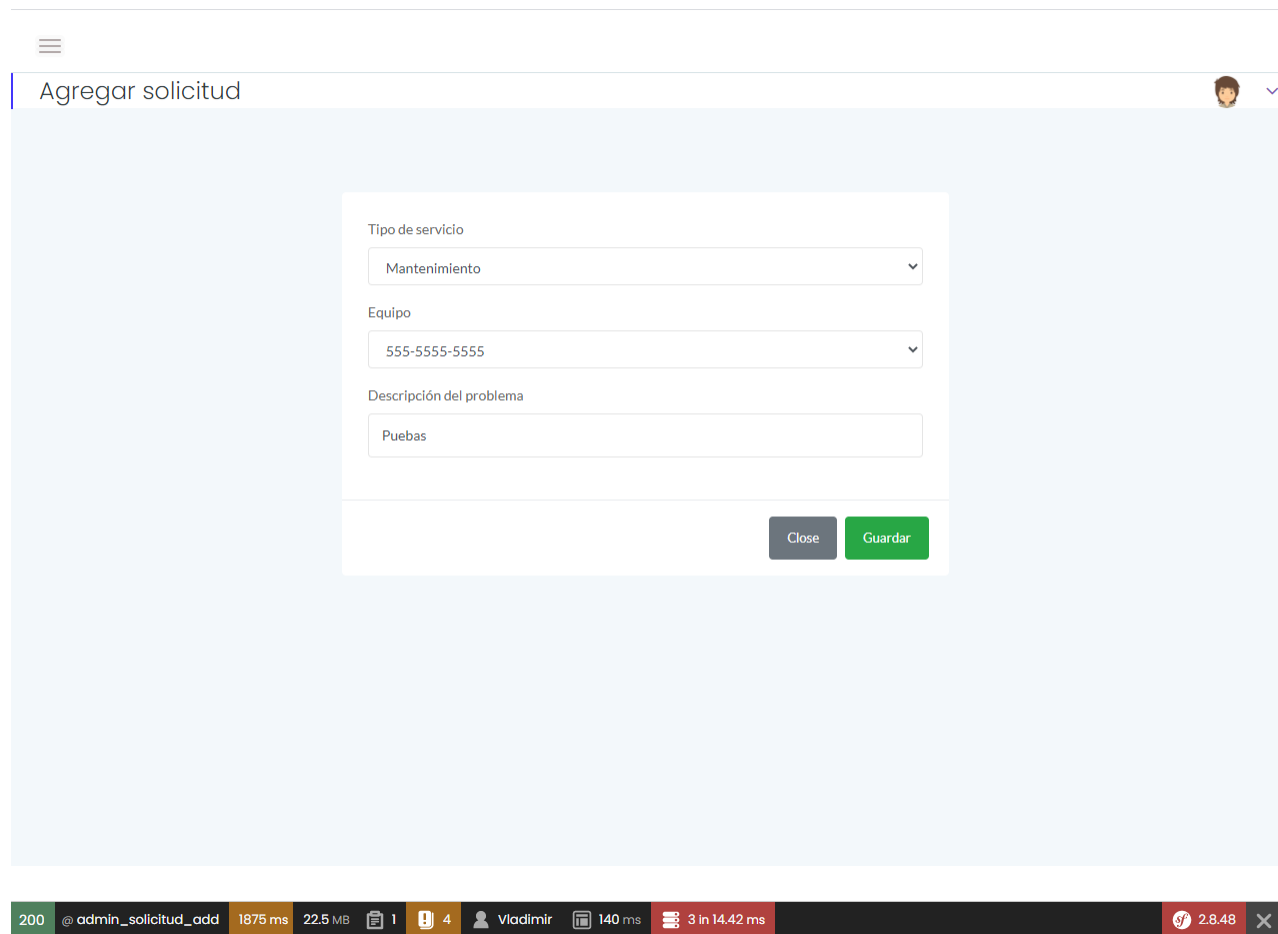
Vista login con datos incorrectos.

Solicitud. La vista Solicitud es donde se administrarán las solicitudes, desde el usuario Administrador y Técnico, que tendrán facultades para realizar cualquier tipo de solicitud, actualizar y eliminar las mismas, y el usuario Final, que podrá realizar solicitudes solamente desde su establecimiento, con los equipos disponibles en el mismo, y teniendo facultades de actualizar y eliminar las que ha creado.

La pruebas consisten en ingresar datos válidos para crear una nueva solicitud desde la vista administrador y desde la vista usuarios, e ingresando datos erróneos desde la

misma.

En la solicitud de la Figura 69, los datos son correctos para la creación de solicitud desde el usuario Administrador y Técnico, donde le cargan todos los equipos de todos los establecimientos. Al ser correctos, redirige a la vista de las solicitudes pendientes. Esto es igual en la vista de *Solicitudes* para usuarios; donde en lugar de traer todos los equipos, solo muestra los equipos por el establecimiento del mismo.



The screenshot shows a web application interface for adding a request. The page title is "Agregar solicitud". The form contains the following fields:

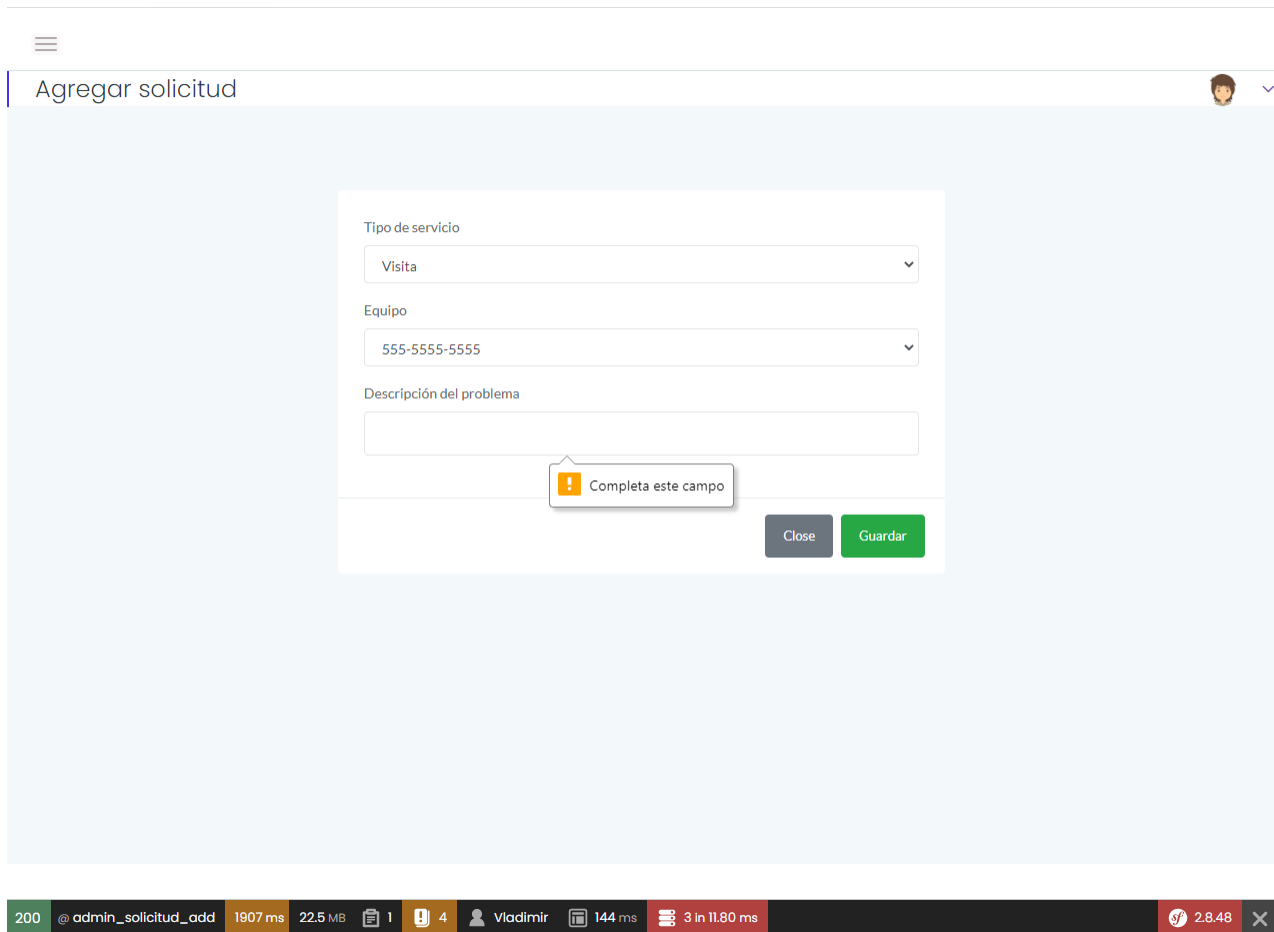
- Tipo de servicio: Mantenimiento
- Equipo: 555-5555-5555
- Descripción del problema: Puebas

At the bottom right of the form, there are two buttons: "Close" and "Guardar".

Figura 69

Vista agregar solicitud con datos correctos.

Tanto para los usuarios Administrador, Técnicos y Finales, el campo *descripción* es obligatorio, provocando que el formulario no se envíe mientras no se ingresen datos, como muestra la Figura 70.



Agregar solicitud

Tipo de servicio
Visita

Equipo
555-5555-5555

Descripción del problema

Completa este campo

Close Guardar

Figura 70

Vista agregar solicitud con datos incorrectos.

Hoja de servicio. Las hojas de servicio son los comprobantes de que se le dio resolución al problema del usuario. Para que una exista, debe de aceptarse una solicitud, y además, desde la administración se podrá agendar una visita si así es requerido.

Al aceptar una solicitud, nos muestra el formulario de la Figura 71, donde se le agregará la resolución y los insumos que se utilizaron para solventar la solicitud.

Caso contrario, nos pedirá que llenemos la hoja de servicio con los datos requeridos, como lo son los insumos utilizados y la resolución final, tal como se muestra en la Figura 72.

The screenshot shows a web form titled "Hoja De Servicio". It contains the following fields and controls:

- Establecimiento:** Departamento financiero
- Usuario:** Vladimir
- Equipo:** 666-666-666
- Descripción de la solicitud:** desde usuario admin
- Soporte Técnico:** A grid of eight toggle switches:
 - Mantenimiento preventivo:
 - Instalación de red:
 - Mantenimiento correctivo:
 - Instalación de S.O:
 - Configuración de red:
 - Im. y con. De impresora:
 - Inventario:
 - Supervisión:
- Buscar Insumo:** A text input field containing "Disco duro" and an "Agregar" button.
- Table:** A table with columns: Codigo, Nombre, Stock, cantidad, Opciones.
- Descripción de la resolución:** A text input field containing "Se le cambio el disco duro".
- Buttons:** "Procesar" at the bottom.

Figura 71
Hoja de servicio con datos correctos.

The screenshot shows the same web form as Figure 71, but with the following differences:

- Descripción de la resolución:** The text input field is empty and contains the placeholder text "Ingrese una descripción".
- Buttons:** The "Procesar" button is present at the bottom.

Figura 72
Hoja de servicio con datos incorrectos.

Visitas. Las visitas serán gestionadas por el usuario Administrador, por lo que se pueden crear desde una solicitud y desde la vista *Visitas*. La prueba consiste en agregar la visita desde las vistas antes mencionada y luego visualizarla en el calendario.

En la Figura 73, se muestra el formulario con los datos correctos, y así mismo en la Figura 74 se muestra respectivamente en el calendario.

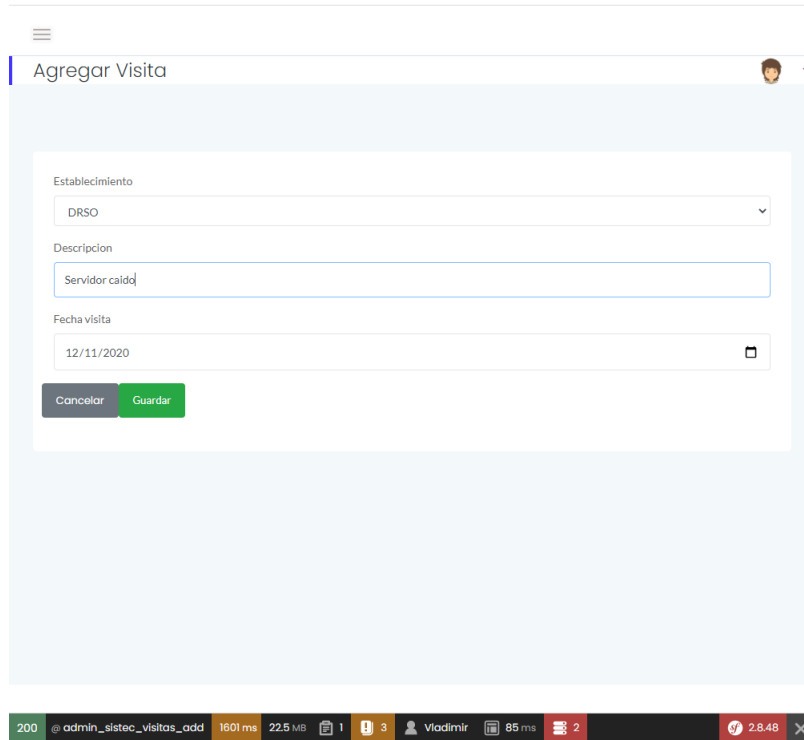


Figura 73
Ingreso de visita con datos correctos.

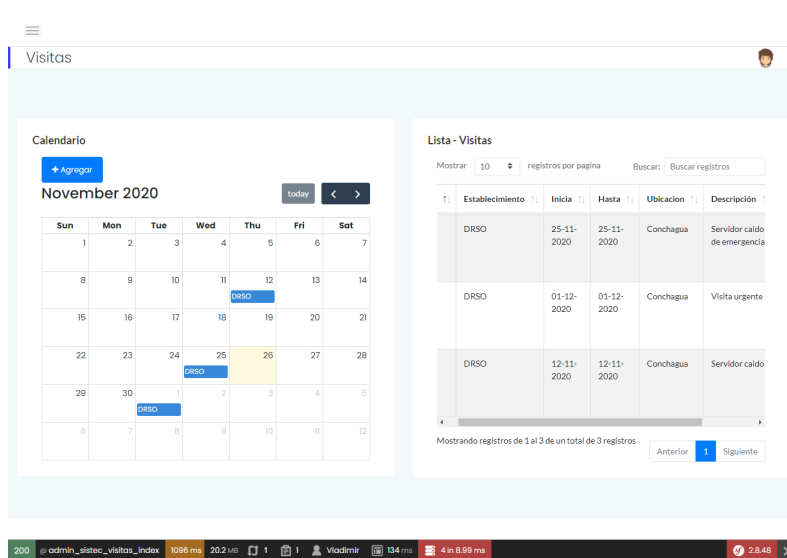


Figura 74
Vista de la visita en el calendario.

Descarte. El módulo *Descarte* hace que desde una solicitud, se pueda descartar un equipo al cual se le solicitó revisión anteriormente. La prueba consistirá en realizar un

descarte desde la vista *Lista de Hojas de Servicio* e imprimir un archivo PDF de la misma.

Primero seleccionamos desde la vista *Lista de Hojas de Servicio* la solicitud al formulario de *Descarte*, tal como se muestra en la Figura 75, y muestra el archivo PDF, en la Figura 76.

Figura 75
Formulario de Hoja de Descarte.

ESTABLECIMIENTO: Departamento financiero		FECHA: 27-11-2020
MOTIVO PARA DESCARTAR: Remplazo por uno nuevo		
EQUIPO: CPU		
MARCA: Asus		N° DE INVENTARIO: 555-5555-5555
MODELO: Frix		
SERIE: 12358874555		
OTRAS CARACTERÍSTICAS: Equipo de 3 años de uso		
OBSERVACIONES: Se le realiza el descarte porque la pieza que necesita reemplazo es la Motherboard.		

F. _____
Coordinador de Informática

1 / 1

Figura 76
Archivo PDF de la Hoja de Descarte.

Pruebas de integración en las aplicaciones web y móvil.

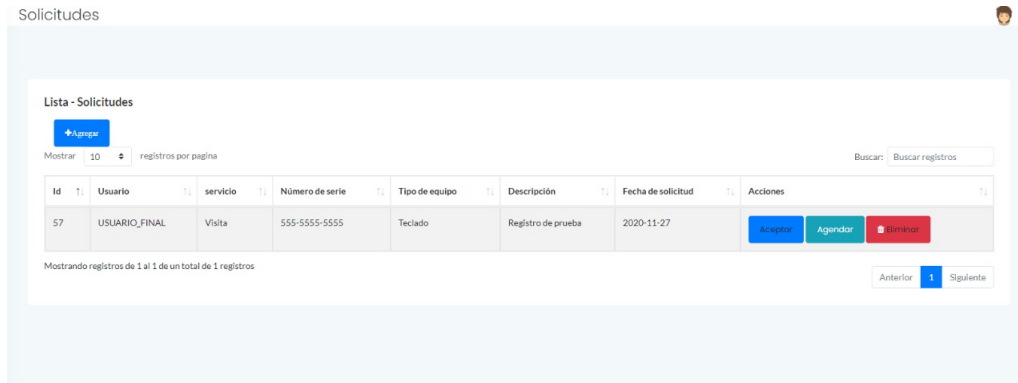
Solicitudes. Las solicitudes se pueden crear desde la aplicación móvil y cargarla en la vista de *Solicitudes* de aplicación web para aceptarla. La prueba consiste en agregar una solicitud y mostrarla en el panel de solicitudes del administrador.

En la Figura 77 se observa como se crea una nueva solicitud desde la aplicación móvil. Luego, si vemos en la Figura 78, ya se puede observar la solicitud en la vista del usuario administrador.



Figura 77

Agregar solicitud desde aplicación móvil.



Solicitudes

Lista - Solicitudes

Mostrar 20 registros por página

Buscar: Buscar registros

Id	Usuario	servicio	Número de serie	Tipo de equipo	Descripción	Fecha de solicitud	Acciones
57	USUARIO_FINAL	Visita	555-5555-5555	Teclado	Registro de prueba	2020-11-27	Agregar Agendar Eliminar

Mostrando registros de 1 al 1 de un total de 1 registros

Anterior 1 Siguiente

Figura 78

Ver solicitud en la aplicación web.

Pruebas de carga. Las pruebas de carga se utilizan para saber cuanta información puede manejar el sistema sin empeorar su rendimiento. Para esta prueba, se realizaron pruebas de 200 registros en cada tabla y de 2000 registros en cada tabla.

En la Figura 79 se puede observar la lista de solicitudes con 200 registros, donde se listan por el *id* del mismo, con su respectivo paginador y búsqueda entre todos los registros mostrados. Con respecto al rendimiento, la información se muestra sin problemas, en *29.65 ms* según datos de desarrollo del propio framework Symfony.

De igual manera, podemos observar en la Figura 80 el listado de las solicitudes correspondientes, agregando navegación entre las paginas y su respectivo buscador. Con respecto al rendimiento, la información se muestra sin problemas, en *22 ms* según datos de desarrollo del propio framework Flutter.

Lista - Solicitudes

Mostrar 10 registros por página

Buscar:

Id	Usuario	servicio	Número de serie	Tipo de equipo	Descripción	Fecha de solicitud	Acciones
399	Yuka	Soporte	1025-231805-55465-30	Tipo de equipo 1013	prueba	2020-12-06	Aceptar
400	Yuka	Soporte	1025-231805-55465-30	Tipo de equipo 1013	prueba	2020-12-06	Aceptar
401	Yuka	Soporte	1025-231805-55465-30	Tipo de equipo 1013	prueba	2020-12-06	Aceptar
402	Yuka	Soporte	1025-231805-55465-30	Tipo de equipo 1013	prueba	2020-12-06	Aceptar
403	Yuka	Soporte	1025-231805-55465-30	Tipo de equipo 1013	prueba	2020-12-06	Aceptar
404	Yuka	Soporte	1025-231805-55465-30	Tipo de equipo 1013	prueba	2020-12-06	Aceptar
405	Yuka	Soporte	1025-231805-55465-30	Tipo de equipo 1013	prueba	2020-12-06	Aceptar
406	Yuka	Soporte	1025-231805-55465-30	Tipo de equipo 1013	prueba	2020-12-06	Aceptar
407	Yuka	Soporte	1025-231805-55465-30	Tipo de equipo 1013	prueba	2020-12-06	Aceptar
408	Yuka	App móvil	1025-231805-55465-31	Tipo de equipo 1014	prueba	2020-12-06	Aceptar

Mostrando registros de 1 al 10 de un total de 79 registros

Anterior 1 2 3 4 5 ... 8 Siguiente

Figura 79
Lista de solicitud con 200 registros.



Figura 80
Lista de solicitud en la aplicación móvil con 200 registros.

Para la siguiente prueba, se agrego un total de 2000 registros en cada tabla en la base de datos. Por lo que, en la Figura 81 se observa la lista de solicitudes activas en el sistema. Y el rendimiento, la información se muestra en un promedio de *2 minutos*, lo que sera un atraso evidente para obtener esta información.

De igual manera, podemos observar en la Figura 82 el listado de las solicitudes correspondientes, agregando navegación entre las paginas y su respectivo buscador. Con respecto al rendimiento, la información se muestra un poco mas lenta de lo usual, en *1.55 minutos* según datos de desarrollo del propio framework Flutter, por lo que representaría un atraso en obtener información mediante la aplicación móvil.

Id	Usuario	servicio	Número de serie	Tipo de equipo	Descripción	Fecha de solicitud	Acciones
399	Yuka	Soporte	1025-231805-55465-30	Tipo de equipo 1013	prueba	2020-12-06	Aceptar
400	Yuka	Soporte	1025-231805-55465-30	Tipo de equipo 1013	prueba	2020-12-06	Aceptar
401	Yuka	Soporte	1025-231805-55465-30	Tipo de equipo 1013	prueba	2020-12-06	Aceptar
402	Yuka	Soporte	1025-231805-55465-30	Tipo de equipo 1013	prueba	2020-12-06	Aceptar
403	Yuka	Soporte	1025-231805-55465-30	Tipo de equipo 1013	prueba	2020-12-06	Aceptar
404	Yuka	Soporte	1025-231805-55465-30	Tipo de equipo 1013	prueba	2020-12-06	Aceptar
405	Yuka	Soporte	1025-231805-55465-30	Tipo de equipo 1013	prueba	2020-12-06	Aceptar
406	Yuka	Soporte	1025-231805-55465-30	Tipo de equipo 1013	prueba	2020-12-06	Aceptar
407	Yuka	Soporte	1025-231805-55465-30	Tipo de equipo 1013	prueba	2020-12-06	Aceptar
408	Yuka	App móvil	1025-231805-55465-31	Tipo de equipo 1014	prueba	2020-12-06	Aceptar

Figura 81

Lista de solicitud con 2000 registros.



Figura 82

Lista de solicitud en la aplicación móvil con 2000 registros.

Pruebas de aceptación de las aplicaciones web y móvil. Las pruebas de aceptación se utilizan para verificar que la aplicación cumple con los requerimientos de desarrollo.

Para esta prueba, se le presentó tanto la aplicación web como la móvil a los técnicos del Departamento de Informática; a lo que comentaron que les parece funcional y que cumple con lo querido.

Documentación y plan de implementación

Documentación

La documentación de sistema consiste en un conjunto de información relacionada a un sistema determinado que explica las características técnicas, la funcionabilidad del sistema, la parte lógica, los diagramas de flujos, los programas, la naturaleza, capacidades del sistema y cómo usarlo, en otras palabras, la documentación de sistemas es el conjunto de información que nos dice qué hacen los sistemas, cómo lo hacen y para quién lo hacen.

Manuales

Los manuales son la guía que muestra al usuario el uso, instalación y configuración de las aplicaciones web y móvil.

Manual de usuario. El manual de usuario se ha elaborado con el propósito de ser una herramienta que guíe al usuario en el uso adecuado del sistema informático. Esta disponible tanto para la aplicación web como para la aplicación móvil.

Manual de instalación y configuración de la aplicación web. En este manual se muestran los pasos a seguir para la instalación y configuración del sistema. El contenido de este manual es básico para instalar todos los componentes necesarios para que la aplicación funcione de forma óptima, se debe aclarar que es importante seguir los pasos tal como indica el manual para realizar una instalación exitosa. En este manual se reúne la información y Documentación necesaria para que el administrador conozca y manipule adecuadamente la aplicación.

Manual de configuración de la aplicación móvil. En este manual se muestran los pasos a seguir para la configuración y compilación de la aplicación móvil. El contenido de este manual es básico para configurar todos los componentes necesarios para que la aplicación funcione de forma óptima, se debe aclarar que es importante seguir los pasos tal como indica el manual para realizar una compilación e instalación exitosa. En este manual se reúne la información y Documentación necesaria para que el administrador

conozca y manipule adecuadamente la aplicación móvil.

Manual del programador. En este manual contendrá toda la información necesaria para la persona que desee mejorar o actualizar las funcionalidades del sistema, así como también implementar nuevas funcionalidades al sistema informático y añadir funciones a la aplicación móvil.

Plan de implementación

En el momento de implementar un sistema informático se debe asegurar la operatividad de este y que permita al usuario recibir beneficios por su operación, que ayude agilizar procesos y a mantener la información de una manera más segura, centralizada y de fácil acceso.

El plan de implementación es una guía que ofrece apoyo al proceso de poner en marcha el sistema informático, en el cual se definen los pasos a seguir y demás aspectos que se deben tener en cuenta a lo largo de dicho proceso.

Dicho plan es un instrumento de programación y control de la ejecución de los proyectos y actividades que se deben llevar a cabo para dar cumplimiento a la puesta en marcha de un proyecto.

El plan contiene la arquitectura de red necesaria para que el sistema funcione, también se describen desde las actividades que se deben realizar, hasta el personal necesario para llevarlas a cabo, así como los controles necesarios para verificar el buen desarrollo de todo el proceso, con el fin de poder obtener los resultados deseados.

Arquitectura

Un elemento principal para el buen desempeño del sistema informático es la arquitectura de los equipos en los que este se ejecutara, por lo tanto, se han hecho sugerencias como las características del hardware y software necesarios para su implementación.

Recurso tecnológico. Como recomendación se detalla a continuación el Hardware y Software mínimos necesarios para que el sistema automatizado pueda ejecutarse sin ningún problema.

Hardware. Los requerimientos mínimos de hardware para implementar el sistema se describen en la Cuadro 46.

Elemento	Servidor	Usuarios
Memoria RAM	12 GB	8 GB
Disco duro	1 TB	500 GB
Procesador	Intel Core i5 3 GHz	Intel Pentium - 2 GB de RAM - 16 GB Almacenamiento
Smartphone		- Pantalla de 5.0" - Procesador 1.5 GHz - Android 5.0

Cuadro 46

Requerimientos mínimos en hardware para implementación del sistema.

Software. El software para implementar el sistema se describen en el Cuadro 47.

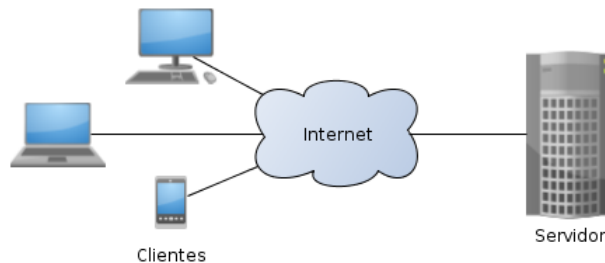
Elemento	Servidor	Usuarios
Sistema operativo	Debian	Windows o GNU/Linux
Base de datos	PostgreSQL 9.6	-
Servidor web	Apache	-
Navegador web	-	Mozilla Firefox o Google Chrome

	Framework	
Otras herramientas	Symfony 2.8, Visual Studio Code.	-

Cuadro 47

Software necesario para la implementación del sistema.

Descripción de la arquitectura del sistema. La arquitectura del sistema se encuentra compuesta por los elementos presentados en la Figura 83, en él se muestran los enlaces de comunicación que tienen entre estos.

**Figura 83**

Arquitectura de implementación del sistema.

En donde intervienen:

- *Cientes.* Los usuarios estarán conectados a través de estaciones de trabajo haciendo peticiones hacia el servidor.
- *Internet.* Será el medio de comunicación utilizado para el intercambio de datos entre el servidor y los usuarios del sistema.
- *Servidor.* El servidor está compuesto por un servidor web apache que tiene soporte para aplicaciones PHP y funciona en cualquier sistema operativo y un servidor de bases de datos PostgreSQL que procesa toda la información necesaria.

Actividades a realizar

Las actividades a realizar, antes y durante la ejecución del plan de implementación son:

- Planificación de recursos informáticos y recursos humanos.
- Programación de actividades.
- Creación de guías prácticas para los usuarios.
- Acondicionamiento de equipo.
- Presentación del proyecto.
- Capacitación al personal.
- Puesta en marcha del sistema informático y aplicación móvil.
- Evaluación de aplicación por parte de los usuarios.

Plan de capacitación al personal

El plan de capacitación a los usuarios de un sistema informático es una herramienta importante para que estos sean hábiles, al utilizarla aplicación. De esta manera, la institución se volverá más fuerte, productiva y rentable. El plan de capacitación; que consta de cinco fases: planeación, organización, ejecución, evaluación y seguimiento.

El plan de capacitación tiene como objetivo que los usuarios puedan utilizar de una forma fácil y eficiente la aplicación web y móvil; por lo que es necesario brindar capacitaciones a los usuarios involucrados.

Las capacitaciones irán en el sentido de orientar al usuario en el uso de los diferentes módulos que integran la aplicación, con la finalidad de mejorar el desempeño del personal en las tareas diarias que realiza en la institución. El uso adecuado del sistema depende en

gran medida de la comprensión y usabilidad de cada una de las partes que conforman el mismo.

Es por lo anterior que se ha elaborado el plan de capacitación para los usuarios, para que puedan orientar en el uso del sistema web según el nivel de usuarios que posean.

Recursos a utilizar

Recurso informático. Se refiere al recurso informático necesario para realizar el proceso de capacitación a los usuarios. Por lo que los recursos necesarios se describen en el Cuadro 48.

Número	Equipo	Descripción
10	Computadoras o laptop	Se utilizarán para las capacitaciones del personal de equipo de informática. Serán los equipos que usará el
2	Laptop	personal encargado del sistema y de brindar la capacitación.
1	Proyector	Para realizar la presentación y el paso a paso durante la capacitación.
1	Impresora	Se utilizará para imprimir el manual que se les proporcionará a los usuarios correspondientes.

Cuadro 48

Recursos a utilizar para la capacitación del personal.

Recurso humano. El recurso humano es un componente fundamental en un proceso de capacitación e implementación del sistemas. Los recursos necesarios para este proceso se describen en el Cuadro 49; mientras que los usuarios que recibirán capacitaciones por rol se describen en el Cuadro 50.

Número	Cargo	Descripción
2	Administradores del sistema	Serán los encargados de brindar la capacitación a los usuarios.

Cuadro 49*Personal capacitador.*

Número	Cargo	Descripción
1	Administrador	Serán los encargados de toda la administración del sistema.
3	Técnico	Usuarios que tendrán acceso a la administración de las solicitudes y las visitas.
2,000 aprox.	Usuario final	Personas que laboran en las diferentes entidades a las que presta servicios la DRSO.

Cuadro 50*Personal a capacitar.****Módulos a capacitar por usuario***

Usuario administrador. Este usuario es el encargado del área de sistemas informáticos, tiene acceso a todos los módulos del sistema informático, su capacitación estará a cargo de los desarrolladores. Los módulos específicos a capacitar se encuentran detallados en el Cuadro 51.

Módulo	Descripción	Duración
Introducción al sistema web	- Acceso al sistema - Navegación - Ingreso de equipos	20 minutos
Equipos	- Búsqueda de equipos - Actualización de equipos - Ingreso de insumos	30 minutos
Insumos	- Búsqueda de insumos - Movimientos de insumos - Ingreso de solicitudes de usuario final	1 hora
Solicitudes	- Búsqueda de solicitudes - Aceptación de solicitud - Actualización de estado de petición	1 hora
Hoja de servicio	- Búsqueda de petición - Imprimir hoja de servicio	1 hora
Descarte	- Agregar descarte - Búsqueda de hojas de descarte	40 minutos
Reportes	- Generar reportes - Agregar usuario	30 minutos
Herramientas	- Cambiar contraseña - Actualizar información de usuarios	30 minutos
Visitas	- Agendar visitas - Actualización de visitas	30 minutos
Duración total de capacitación		6 horas

Cuadro 51

Módulos a capacitar al administrador.

Usuario Técnico. Este usuario es encargado de los mantenimientos y visitas principalmente. Su capacitación estará a cargo del personal que administre el sistema. Estas se muestran en el Cuadro 52.

Módulo	Descripción	Duración
Introducción al sistema web	- Acceso al sistema - Navegación	20 minutos
Equipos	- Ingreso de equipos - Búsqueda de equipos - Actualización de equipos	30 minutos
Solicitudes	- Ingreso de solicitudes de usuario final - Búsqueda de solicitudes - Aceptación de solicitud - Actualización de estado de	1 hora
Hoja de servicio	petición - Búsqueda de petición - Imprimir hoja de servicio	1 hora
Descarte	- Agregar descarte - Búsqueda de hojas de descarte	40 minutos
Reportes	- Generar reportes - Agregar usuario	30 minutos
Herramientas	- Cambiar contraseña - Actualizar información de usuarios	30 minutos
Duración total de capacitación		4 horas 30 minutos

Cuadro 52

Módulos a capacitar al técnico.

Usuario Final. Los usuarios finales son los que reciben servicios de parte del Departamento de Informática. Las capacitaciones por módulo se muestran en el Cuadro 53. Las capacitaciones serán impartidas según el organigrama de la Figura 2 en la cual, se dará inicio tomando como prioridad la división de Gestión administrativa y la división de Gestión de Servicios de Salud; para finalizar con la división de Salud Ambiental y las diferentes Unidades de la DRSO.

Finalizado el proceso de capacitación, que todas las unidades y divisiones que se encuentran dentro de la DRSO, se procederá a la capacitación de los SIBASI de cada una de las cabeceras departamentales de la zona oriental y los ECOS y UCSF de mayor a menor prioridad.

Módulo	Descripción	Duración
Introducción al sistema web	- Acceso al sistema	20 minutos
	- Navegación	
	- Ingreso de solicitudes de usuario	
Solicitudes final		15 minutos
Hoja de servicio	- Búsqueda de solicitudes	15 minutos
	- Ver estado de petición	
	- Búsqueda de petición	
Duración total de capacitación		50 minutos

Cuadro 53

Módulos a capacitar al usuario final.

Conclusiones

En el Ministerio de Salud, en la Dirección Regional de Salud Oriental existe una demora en el procesamiento de las órdenes de soporte técnico, retrasos en visitas fuera de las Unidades de Salud e ineficiencia en la toma de decisiones.

La elaboración e implementación de una aplicación web y móvil permite agilizar los procesos que se desarrollan, mejorar la gestión de órdenes de servicio y ofrece una mejor administración del inventario de insumos informáticos del Departamento ya que permite la creación, almacenamiento, actualización y búsqueda de la información relacionada con los diferentes equipos tecnológicos, proporcionando eficientes procesos de búsqueda, administrando las visitas y creando informes en el momento que sean solicitados.

El Sistema informático SISTEC, que consta de una aplicación web y una móvil, es técnicamente factible puesto que existen los recursos necesarios para que este pueda ser desarrollado, dicho sistema fue elaborado con una interfaz adecuada para los usuarios y a la vez facilitando la obtención de la información agilizando el tiempo de generación de los reportes. Con los manuales elaborados, los usuarios pueden utilizar y mantener el sistema y dar capacitación a nuevos usuarios de la aplicación.

Recomendaciones

- Para la implementación del sistema se recomienda la consideración del Plan de Implementación propuesto en el documento.
- A los usuarios, auxiliarse de la ayuda o manual de usuario del sistema, tanto web como móvil cuando se presenten dudas sobre la operación de este.
- A los administradores del sistema, la realización de una API para consumir los datos del propio MINSAL, para los establecimientos y equipos. Con esto, se garantiza una mayor automatización del mismo.
- A los futuros desarrolladores, se puede implementar la generación de rutas desde la aplicación móvil. Así, se aumenta la eficiencia de los técnicos del Departamento de Informática.

Glosario

- Base de datos** Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. (EcuRed, 2020b). 33
- Copyleft** Es el término que se utiliza en el ámbito informático (y se aplica de manera análoga a la creación literaria, artística, y científica) para designar el tipo de protección jurídica que confieren determinadas licencias que garantizan el derecho de cualquier usuario a utilizar, modificar y redistribuir un programa o sus derivados, siempre que se mantengan estas mismas condiciones de utilización y difusión. (EcuRed, 2020c). 75
- Framework** En el desarrollo de Software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, librerías y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. (EcuRed, 2020f). 41
- Gestor de base de datos** Un Sistema Gestor de Base de Datos (SGBD, en inglés DBMS: DataBase Management System) es un sistema de software que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación. Un SGBD relacional es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos. (EcuRed, 2020g). 49
- Hardware** corresponde a todas las partes físicas y tangibles de una computadora, sus componentes eléctricos, electromecánicos y mecánicos; sus cables, gabinetes o cajas,

periféricos de todo tipo y cualquier otro elemento físico involucrado; contrariamente al soporte lógico e intangible que es llamado software.(EcuRed, 2020h). 77

HTTP De sus siglas en ingles, *Hypertext Transfer Protocol*, es el protocolo usado en cada transacción de la World Wide Web, o WWW. Mediante HTTP los clientes y los servidores determinan de forma dinámica el formato de los documentos, lo que permiten que utilicen formato de datos no estándar para el intercambio de datos. Si el receptor no tiene un modo de ver o acceder a los datos, puede descargar un programa complemento que le permita recibir el contenido. (EcuRed, 2020m).. 115

JSON Acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML. (EcuRed, 2020i).. 115

Metadatos Datos altamente estructurados que describen información. Es "Información sobre información" o "datos sobre los datos", ya que proporciona información mínima necesaria para identificar un recurso e incluye información descriptiva sobre el contexto, calidad y condición o características de un dato u objeto que tiene la finalidad de facilitar la recuperación, autenticación, evaluación, preservación y/o interoperabilidad. (EcuRed, 2020k). 61

Servidor web Es un programa que gestiona cualquier aplicación en el lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando una respuesta en cualquier lenguaje o aplicación en el lado del cliente. (EcuRed, 2020n). 40

Sistema informático Puede ser definido como un sistema de información que basa la parte fundamental de su procesamiento, en el empleo de la computación, como cualquier sistema, es un conjunto de funciones interrelacionadas, hardware, software y de Recurso Humano. Un sistema informático normal emplea un sistema que usa

dispositivos que se usan para programar y almacenar programas y datos. (EcuRed, 2020o). 49

Software Se refiere al equipamiento lógico o soporte lógico de una Computadora digital, y comprende el conjunto de los componentes legales necesarios para hacer posible la realización de tareas específicas; en contraposición a los componentes físicos del sistema, llamados Hardware. (EcuRed, 2020p). 74

Tablas arco iris Son un poderoso instrumento con el que es posible decifrar contraseñas. A esta tarea se dedican un buen número de personas a ambos lados de la legalidad, unos por el beneficio que aspiran a obtener con ellas, los otros porque su trabajo como expertos consiste en comprobar periódicamente la efectividad de los estándares de seguridad vigentes y las tablas arco iris que permiten encontrar claves en unos pocos segundos si se dan ciertas circunstancias. (IONOS, 2020).. 119

XML Por sus siglas en inglés de *eXtensible Markup Language*, es un lenguaje de marcado sencillo similar al HTML. Su objetivo es facilitar la representación, almacenamiento y transmisión de información varia por parte de aplicaciones informáticas, computadoras y medios de comunicación digital en general. (EcuRed, 2020r).. 115

Referencias

Acuerdo Ministerial N° 1208. (2015). Manual de Organización y Funciones de la Región de Salud.

ApiService. (2020). *Qué es una API y que puede hacer por mi negocio.*

<https://apiservice.cl/que-es-una-api-y-que-puede-hacer-por-mi-negocio>

Codahale. (2020). *How to safely store a password.*

<https://codahale.com/how-to-safely-store-a-password>

Cohen, L., Manion, L. & Morrison, K. (2003). *Research Methods in Education*. Routledge.

Corrales, J. D. (2010). *Técnicos de Soporte Informático de la Comunidad de Castilla Y León. Temario Volumen II*. MAD-Eduforma.

Decreto Presidencial N° 77. (2006). Reglamento de la ley del Sistema Básico de Salud Integral.

del Sur, U. N. (2017). *Guía para la documentación de proyectos de software.*

<https://cs.uns.edu.ar/guia-para-la-documentacion-de-proyectos-de-software.pdf>

Doctrine Project. (2020). *Doctrine ORM*.

<https://www.doctrine-project.org/projects/doctrine-orm/en/current/tutorials/getting-started.html%5C#what-is-doctrine>

EcuRed. (2020a). *API*. <https://www.ecured.cu/API>

EcuRed. (2020b). *Bases de datos*. <https://www.ecured.cu/Bases-de-datos>

EcuRed. (2020c). *Copyleft*. <https://www.ecured.cu/copyleft>

EcuRed. (2020d). *Encriptación*. <https://www.ecured.cu/Encriptacion>

EcuRed. (2020e). *Entorno de Desarrollo Integrado*.

<https://www.ecured.cu/IDE-de-Programacion>

EcuRed. (2020f). *Framework*. <https://www.ecured.cu/framework>

EcuRed. (2020g). *Gestor de base de datos*.

<https://www.ecured.cu/sistema-gestor-de-base-de-datos>

EcuRed. (2020h). *Hardware*. <https://www.ecured.cu/hardware>

- EcuRed. (2020i). *JSON*. <https://www.ecured.cu/JSON>
- EcuRed. (2020j). *Kaoru Ishikawa*. <https://www.ecured.cu/Kaoru-Ishikawa>
- EcuRed. (2020k). *Metadatos*. <https://www.ecured.cu/metadatos>
- EcuRed. (2020l). *Power Designer*. <https://www.ecured.cu/PowerDesigner>
- EcuRed. (2020m). *Protocolo de Transferencia de Hipertexto*.
<https://www.ecured.cu/Protocolo-de-Transferencia-de-Hipertexto>
- EcuRed. (2020n). *Servidor web*. <https://www.ecured.cu/servidor-web>
- EcuRed. (2020o). *Sistema informático*. <https://www.ecured.cu/sistema-informatico>
- EcuRed. (2020p). *Software*. <https://www.ecured.cu/software>
- EcuRed. (2020q). *XAMPP*. <https://www.ecured.cu/XAMPP>
- EcuRed. (2020r). *XML*. <https://www.ecured.cu/XML>
- Espinosa, R. (2013). *La matriz de análisis DAFO*.
<https://robertoespinosa.es/2013/07/29/la-matriz-de-analisis-dafo-foda>
- Flutter-es. (2020). *Flutter*. <https://flutter-es.io/>
- Instituto de Acceso a la Información Pública. (2020). *Dirección Regional de Salud Oriental*.
<https://www.transparencia.gob.sv/institutions/drs-oriental>
- IONOS. (2020). *Rainbow tables: qué son y como funcionan las tablas arco iris*.
<https://www.ionos.es/digitalguide/servidores/seguridad/rainbow-tables/>
- Kendall, K. E. & Kendall, J. E. (1997). *Análisis y diseño de sistemas*. Prentice Hall Hispanoamericana S.A.
- Lozano, J. (2015). *Metodologías y tipos de pruebas de software*.
<https://prezi.com/metodologia-y-tipos-de-pruebas-de-software>
- MappingGIS. (2020). *pgAdmin 4*. <https://mappinggis.com/2017/trabaja-con-postgree/>
- Martí, A. & Marín, M. (2014). *Aplicaciones Web*. Paraninfo.
- Michelena, A. V. (2005). *Enredados. El mundo de la Internet*. Estudio Gherzi Editores.
- MINSAL. (2010). Resolución Ministerial N° 91.

- MINSAL. (2011a). Lineamientos técnicos para el mantenimiento, administración y desarrollo de tecnologías de información y comunicaciones.
- MINSAL. (2011b). Manual de organización y funciones de la Dirección de Tecnologías de Información y Comunicaciones.
- MINSAL. (2014). Mapa sanitario del Ministerio de Salud.
- Mozilla. (2020). *Mozilla Support*. <https://support.mozilla.org/es/products/firefox>
- PostgreSQL. (2020). *PostgreSQL: Documentation*. <https://www.postgresql.org/docs/>
- Pressman, R. S. (2010). *Ingeniería de software. Un enfoque practico*. McGraw-Hill Interamericana.
- Sabino, C. (1992). *El proceso de la investigación*. Editorial Panamericana.
- Sampieri, R. H., Lucio, P. B. & Collado, C. F. (2014). *Metodología de la investigación*. McGraw-Hill Interamericana.
- Sánchez, J. C. (2004). *Metodología de la investigación científica y tecnológica*. Editorial Días de Santos, S.A.
- Sanjuán, L. D. (2010). *La observación*. UNAM.
- SAP. (2020). *Power Designer*. <https://www.powerdesigner.biz/>
- Subra, J.-P. (2020). *Scrum. Un metodo agil para sus proyectos*. ENI.
- Symfony. (2020). *Symfony*. <https://symfony.es/documentacion/>
- Tamayo, M. & Tamayo. (2003). *El proceso de la investigación científica*. Limusa.
- Torrecilla, J. M. (2002). La entrevista: metodología de investigación avanzada.
- Uniwebsidad. (2020a). *Cómo funciona la autenticación y autorización*.
<https://uniwebsidad.com/libros/symfony-2-4/capitulo-13/como-funciona-la-seguridad-autenticacion-y-autorizacion>
- Uniwebsidad. (2020b). *Patrón MVC*. <https://dle.rae.es/rol>
- VisualStudioCode. (2020). *Visual Studio Code: Docs*. <https://code.visualstudio.com/docs>
- Vittone, J. & Cuello, J. (2013). *Diseñando apps para móviles*.

Anexos

Carta de compromiso de la Dirección Regional de Salud Oriental

MINISTERIO
DE SALUD

San Miguel, 05 de noviembre de 2019

Señores
Departamento de Ingeniería y Arquitectura
Universidad de El Salvador
Facultad Multidisciplinaria Oriental
Presente

Es para nuestra institución un gusto dirigirnos a ustedes y desearles éxitos en sus actividades cotidianas en pro del desarrollo profesional de nuestro querido El Salvador.

El motivo de la presente es para informarles que estamos en la disposición de apoyar en su proceso de tesis a los estudiantes: **Carlos Fermín Padilla Ferrufino, Konny Lizbeth Amaya Barrera, Edward Vladimir Hernández García, José Alfredo Jiménez Benítez**, para los cuales daremos permiso de que realicen un proyecto de creación de una aplicación web el cual se enfocará en lo siguiente:

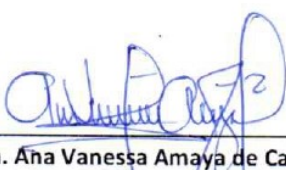
Automatizar procesos de solicitud y control de servicios técnicos para:

- Administrar solicitudes de mantenimiento informático.
- Gestionar el historial de mantenimientos en proceso y los realizados.
- La calendarización de visitas a unidades de salud.
- Facilitar la emisión de reportes.

No omito manifestar que en el **Departamento de informática** cuentan con ciertas herramientas de gestión de estos procesos, las cuales no se adaptan a la necesidad de la institución de poder llevar un mejor control de esta información.

Sin más que agregar y esperando una pronta respuesta favorable me suscribo a ustedes,

Atentamente.


Dra. Ana Vanessa Amaya de Calderón
Directora de Región Oriental de Salud



Cuestionario de entrevista para los profesionales



Universidad de El Salvador
Facultad Multidisciplinaria Oriental
Departamento de Ingeniería y Arquitectura

**Cuestionario dirigido al personal del Departamento de Informática de la
Dirección Regional de Salud Oriental**

Proyecto: Desarrollo de un sistema que automatice los procesos de solicitud y control de servicios técnicos en el Departamento de Informática de la Dirección Regional de Salud Oriental.

Objetivo: Conocer los requerimientos para un sistema informático a desarrollar para el Departamento de Informática de la Dirección Regional de Salud Oriental.

Indicaciones: Responda las siguientes preguntas.

1. Enumere los siguientes procesos del 1 al 5 donde 5 es el que tiene más importancia dentro de su trabajo en el Departamento de Informática:

- a. Mantenimiento preventivo y correctivo de computadoras/servidores _____
- b. Soporte técnico a usuarios _____
- c. Mantenimiento de redes informáticas _____
- d. Capacitaciones _____
- e. Otros: _____

2. Enumere los siguientes problemas del 1 al 5 donde 5 es el que representa a los usuarios con más frecuencia:

- a. Conexión de red _____
- b. Funcionamiento de una PC _____
- c. Funcionamiento de un impresor/scanner _____
- d. Funcionamiento de una app móvil o web _____
- f. Otros: _____

3. ¿Cada cuánto tiempo le realizan el mantenimiento a los equipos informáticos a cargo del Departamento de Informática?

- a. Cuando el usuario solicita mantenimiento _____
- b. Cada 6 meses _____
- c. Cada año _____
- e. Otros: _____

4. ¿Se lleva registro digital del mantenimiento preventivo y correctivo del equipo tecnológico?

Si ____ No ____

Descríbanos cómo lo realiza actualmente.

5. ¿El Departamento de Informática cuenta con un control de todo el equipo a su cargo?

Si ____ No ____

Si la respuesta es sí, ¿de qué manera lo hacen actualmente?

6. ¿Qué cargo desempeña? : _____

¿Que reportes realiza con frecuencia o son más importantes para su cargo?

7. ¿Considera que existe la necesidad de utilizar un sistema informático para desempeñar mejor su labor?



Si ____ No ____

¿Por qué?

8. ¿Qué factores o circunstancias le impedirían realizar su trabajo? (Por ejemplo, que el encargado del área no se encuentre).

9. ¿Qué procesos aparte del control de equipo informático y de solicitudes de soporte le sería útil que incluyera el sistema informático?

Formato de Hoja de servicio

 GOBIERNO DE EL SALVADOR	MINISTERIO DE SALUD	REGIÓN ORIENTAL DE SALUD HOJA DE SERVICIO	 Dirección de Tecnologías de Información y Comunicaciones
Establecimiento: _____		Fecha: / /	
OBJETIVO DE LA VISITA			
TRABAJO REALIZADO			
Soporte técnico:			
Mantenimiento preventivo	<input type="checkbox"/>	Instalación de red	<input type="checkbox"/>
Mantenimiento correctivo	<input type="checkbox"/>	Instalación de S.O.	<input type="checkbox"/>
Configuración de red	<input type="checkbox"/>	Ins. Y conf. De impresora	<input type="checkbox"/>
Inventario	<input type="checkbox"/>	Supervisión	<input type="checkbox"/>
Material y equipo utilizado:			
Fuente de poder:	Cable de red:		
Memoria RAM:	Baterías UPS:		
Disco duro:	Batería electrónica:		
Motherboard:	Switch:		
Tarjetas de red:	Otros:		
Observaciones:			
Técnicos responsables: Juan Soto____ Saul Guevara____ Ernesto Argueta____ Henry Solano____ Willians Herrera____			
Recibo conforme: _____			
Nombre, Firma y Sello			

Formato de Hoja de descarte



MINISTERIO DE SALUD

REGION ORIENTAL DE SALUD
HOJA DE DESCARTE



Establecimiento: _____ Fecha: _____

MOTIVOS PARA DESCARTE:

- Alto costo de repuestos. Reemplazo por uno nuevo.
 Repuesto en desuso. Otro: _____

ESPECIFICACIÓN DE EQUIPO:

EQUIPO:	
MARCA:	
MODELO:	
SERIE:	
No. DE INVENTARIO:	
OTRAS CARACTERISTICAS:	
OBSERVACIONES:	

Coord. Informática

Carta de aceptación del sistema web y móvil



MINISTERIO
DE SALUD

San Miguel, 01 de diciembre de 2020

Señores
Departamento de Ingeniería y Arquitectura
Universidad de El Salvador,
Facultad Multidisciplinaria Oriental.
Presente

Por este medio hago de su conocimiento que los jóvenes bachilleres, **Konny Lizbeth Amaya Barrera, Edward Vladimir Hernández García, José Alfredo Jiménez Benítez y Carlos Fermin Padilla Ferrufino**, hicieron una presentación del sistema multiplataforma para la automatización y control de solicitudes de servicios técnicos que se realizan en el departamento de informática de la Dirección Regional de Salud Oriental, en la cual informo de manera muy atenta que la colaboración del departamento de informática con los bachilleres fue muy eficiente, además de que el sistema y aplicación presentados cumplen con los requerimientos y las diferentes funciones solicitadas en las diferentes reuniones que se realizaron y estas facilitarían el trabajo de los técnicos que laboran en el departamento de informática, posteriormente se enviara para la evaluación y análisis de la propuesta de implementación.

Agradeciendo los lazos establecidos con la Universidad de El Salvador, quedamos a disposición de seguir colaborando en próximos proyectos.

Por su atención, gracias.

Dra. Ana Vanessa Amaya de Calderón
Directora de Región Oriental de Salud