



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 2

Sistema Integral de Gestión de los Laboratorios



Módulo “Control de Personal”

Trabajo de diploma para optar por el título de Ingeniero Ciencias Informáticas.

Autores: Yunier Matos Lobaina
Yaily Rodriguez Rodriguez

Tutor: Ing. Salvador González Gómez

Co-tutor: Ing. Guillermo Baez Ramos

Ciudad de la Habana
Julio del 2008

DECLARACIÓN DE AUTORÍA

Declaramos ser autores del presente Trabajo de Diploma, de esta forma reconocemos y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo como considere necesario.

Para que así conste, firmamos la presente a los ____ días del mes _____ de 2008.

Firma de los autores
Yaily Rodríguez Rodríguez

Firma de los autores
Yunier Matos Lobaina

Firma del tutor
Ing. Salvador González Gómez

OPINIÓN DE TUTOR DEL TRABAJO DE DIPLOMA

Título: Módulo para el Control del Personal de los Laboratorios

Autores: Yaily Rodríguez Rodríguez

Yunier Matos Lobaina

El tutor del presente Trabajo de Diploma considera que durante su ejecución los estudiantes mostraron las cualidades que a continuación se detallan:

Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingenieros en Ciencias Informáticas; y propongo que se le otorgue al Trabajo de Diploma la calificación de _____.

Ing. Salvador González Gómez

_____ de julio del 2008.

"Lo que convierte la vida en una bendición no es hacer lo que nos gusta, sino que nos guste lo que hacemos"

Goethe

Agradecimientos

De Yaily:

A toda mi familia, especialmente a mi nana, la mejor madre de este mundo.

A mami y papi, por darme todo el amor del mundo y ser los mejores abuelos.

Al hombre que ha sido mi padre toda mi vida y me ha visto crecer, tío Silvito.

A mis hermanitos Yandy y Yailin, de los cuales espero muchas cosas.

A mi tía Martha, tío Ramoncito, mi primita Lumey, mis primos jimaguas Lester y Leiser y la reciente miembro de la familia Angeline, los quiero mucho a todos y son una parte importante de mí.

A mi papá, a Yamilé, a abuela Milagro y todos mis tíos, por ser tan buenos conmigo.

A mi amigo, mi compañero, mi novio Yunier (tati) que es una parte importantísima en mi vida, durante muchos años a sido mi apoyo y agradezco la dicha de poder compartir junto a él esta tesis.

A mis amistades que siempre están cuando más los necesito, especialmente: Lismary, Castillo, Marita, Noli y Ronny.

A mi tutor Salvador y co-tutor Guillermo que me han apoyado incondicionalmente y en todo momento.

A todos mis compañeros de aula.

A los profes que me han ayudado a lo largo de estos 5 años y que han contribuido a mi formación profesional y personal.

A los vecinos que siempre esperan mi llegada: Gilda, Elsita, Nené, Yuni, Ubelkis, Carmita, Victoria y Nena.

A Miriam, Yanara, Francel, Ana, Mercedes y las personas de contabilidad que siempre me auxiliaron para que yo pudiera realizar mis trabajos.

En fin a todas aquellas personas que ayudaron a formar en mí una persona de bien, a todos muchísimas gracias.

Agradecimientos

De Yunier:

Agradecer a toda mi familia por la confianza que han depositado en mí y por toda la ayuda que de una u otra forma me han brindado, a mis hermanos y en especial a mi madre, porque de no ser por ella no hubiera llegado a ser lo que soy, porque siempre que la necesité estuvo ahí para darme su apoyo y por sobre todas las cosas, por todo su amor.

A mi novia Yaily, porque desde que la conocí no ha hecho otra cosa que darme amor, cariño y toda la fuerza necesaria para poder seguir adelante, por estar a mi lado cada minuto regañándome, enseñándome y principalmente ayudándome a hacer las cosas cada día mejor.

A mi suegrita Belkis porque por ella es que tengo la novia que tengo, por la gran ayuda y confianza que ha depositado en mi.

A mis amigos: Noli, Reinier, Ismary, losmara, Ronny en fin a todos por toda la ayuda que me han brindado y a todos mis compañeros de aula.

A mi tutor y co-tutor por apoyarme y ayudarme en el desarrollo de este trabajo de diploma.

A todos los profesores que me han enseñado a superarme tanto personal como profesionalmente.

A mis vecinos que no por ser los últimos dejan de ser los menos importantes.

Dedicatoria

A toda mi familia, con mucho amor y cariño, especialmente a mamá por quererme y apoyarme tanto.

A ti te agradezco infinitamente por haberme cuidado y haber formado en mí lo que soy hoy.

A mami, mi viejita y abuelito Moro.

A tío y a mi hermanito.

En fin, no tengo palabras para escribir todo lo que ustedes inspiran en mí.

Yaily

Dedico este trabajo a toda mi familia, en especial a la mujer que está siempre pendiente de mí, y me ayuda cuando la necesito (mi madre).

Dedico también este trabajo de diploma a la memoria de mi padre por haberme enseñado muchas cosas buenas en la vida y aunque ya no esté conmigo, se que está muy orgulloso de mí.

Yunier

RESUMEN

El presente trabajo tiene como propósito la descripción de una propuesta de sistema informático que automatice los procesos para el Control de Personal en los laboratorios de la Universidad de las Ciencias Informáticas (UCI). Actualmente, la gestión de la información de los trabajadores en esta área resulta deficiente, ya que se realiza de forma manual y se repiten gran cantidad de datos. Teniendo en cuenta esta necesidad, el objetivo de la investigación es modelar e implementar una aplicación Web para el "Control de Personal" vinculado al Sistema Integral de Gestión de los Laboratorios, que sea capaz de: gestionar los datos de los trabajadores, así como ubicarlos en un área, un turno y en un laboratorio.

Resulta importante que se realice la automatización de los procesos, utilizando para ello los servicios Web, de manera tal que se garantice mayor eficiencia, control e integración con otros sistemas disponibles en la universidad. Además, con la puesta en marcha de la aplicación se aseguran mejores condiciones de trabajo para los usuarios, evitándoles el agotamiento que produce el trabajo manual con gran volumen de información.

INTRODUCCIÓN	1
CAPÍTULO 1 “FUNDAMENTACIÓN TEÓRICA”	4
1.1. INTRODUCCIÓN	4
1.2. TÉRMINOS UTILIZADOS	4
1.3. SISTEMAS PARA EL CONTROL DE PERSONAL	4
1.4. TECNOLOGÍAS, LENGUAJES DE PROGRAMACIÓN Y LIBRERÍA	6
1.4.1. <i>Lenguaje de programación</i>	7
1.4.1.1. <i>PHP</i>	7
1.4.2. <i>Librería ExtJS 2.0</i>	8
1.4.2.1. <i>AJAX</i>	8
1.4.3. <i>Servicios Web</i>	12
1.4.4. <i>Servidor Web</i>	13
1.4.5. <i>Sistema Gestor de Base de Datos</i>	13
1.5. ARQUITECTURA	14
1.5.1. <i>Patrón Modelo-Vista-Controlador (MVC)</i>	14
1.5.2. <i>Arquitectura Cliente/Servidor</i>	15
1.6. MODELACIÓN DE FUNCIONES MEDIANTE IDEF0	16
1.7. LENGUAJE DE MODELADO Y METODOLOGÍA DE DESARROLLO	16
1.7.1. <i>Lenguaje de modelado UML</i>	16
1.7.2. <i>Metodología de desarrollo RUP</i>	17
1.8. HERRAMIENTAS UTILIZADAS	19
1.8.1. <i>Visual Paradigm</i>	19
1.8.2. <i>Aptana</i>	19
1.8.3. <i>GIMP</i>	20
1.8.4. <i>PGAdmin III</i>	20
1.9. CONCLUSIONES DEL CAPÍTULO	21
CAPÍTULO 2 “CARACTERÍSTICAS DEL SISTEMA”	22
2.1. INTRODUCCIÓN	22
2.2. OBJETO DE ESTUDIO	22
2.2.1. <i>Definición del problema</i>	22
2.2.2. <i>Objeto de automatización</i>	23

2.2.3. Información que se maneja	24
2.3. PROPUESTA DEL SISTEMA	24
2.4. MODELO DEL NEGOCIO	25
2.4.1. Representación de los procesos con IDEF0.....	26
2.4.2. Descripción detallada de los procesos del negocio.....	26
2.5. ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE	27
2.5.1. Requerimientos funcionales.....	27
2.5.2. Requerimientos no funcionales.....	28
2.6. MODELO DEL SISTEMA.....	31
2.6.1. Actores del sistema.....	31
2.6.2. Diagrama de caso de uso del sistema.....	32
2.6.3. Descripción de los casos de uso del sistema.....	32
2.7. CONCLUSIONES	33
CAPITULO 3 “ANÁLISIS Y DISEÑO”	34
3.1. INTRODUCCIÓN	34
3.2. MODELO DE ANÁLISIS	34
3.2.1. Diagrama de clases del análisis	34
3.3. MODELO DE DISEÑO	38
3.3.1. Diagrama de clases del diseño.....	40
3.4. DIAGRAMAS DE INTERACCIÓN	50
3.5. DISEÑO DE LA BASE DE DATOS.....	50
3.5.1. Modelo lógico de datos	50
3.5.2. Modelo físico de datos	51
3.6. CONCLUSIONES	52
CAPITULO 4 “IMPLEMENTACIÓN”	53
4.1. INTRODUCCIÓN	53
4.2. DIAGRAMA DE DESPLIEGUE	53
4.3. DIAGRAMA DE COMPONENTES	54
4.4. CONCLUSIONES	60
CAPITULO 5 “ESTUDIO DE LA FACTIBILIDAD”	61
5.1. INTRODUCCIÓN	61

5.2.	PLANIFICACIÓN	61
5.2.1.	<i>Puntos de Casos de Uso</i>	61
5.2.2.	<i>Estimación del esfuerzo</i>	65
5.3.	BENEFICIOS TANGIBLES E INTANGIBLES	67
5.4.	ANÁLISIS DE COSTOS Y BENEFICIOS	68
5.5.	CONCLUSIONES	68
CONCLUSIONES		69
RECOMENDACIONES		70
BIBLIOGRAFÍA		71
ANEXOS		75
	<i>ANEXO 1. DESCRIPCIÓN TEXTUAL DE LOS CASOS DE USO DEL SISTEMA</i>	75
	<i>ANEXO 2. PROTOTIPOS DE INTERFAZ</i>	85
	<i>ANEXO 3. DIAGRAMAS DE SECUENCIA</i>	90
GLOSARIO DE TÉRMINOS		97

FIGURA 1 ARQUITECTURA CLIENTE/SERVIDOR	15
FIGURA 2 INTERCAMBIO DE DATOS ENTRE LA PC CLIENTE Y LOS SERVIDORES.....	25
FIGURA 3 PROCESO UBICAR TRABAJADOR EN PUESTO DE TRABAJO	26
FIGURA 4 DIAGRAMA DE CASOS DE USO DEL SISTEMA	32
FIGURA 5 DIAGRAMA DE CLASES DEL ANÁLISIS: GESTIONAR TRABAJADOR.....	34
FIGURA 6 DIAGRAMA DE CLASES DEL ANÁLISIS: MOSTRAR TRABAJADOR POR ÁREA	35
FIGURA 7 DIAGRAMA DE CLASES DEL ANÁLISIS: MOSTRAR TRABAJADOR POR TURNO	35
FIGURA 8 DIAGRAMA DE CLASES DEL ANÁLISIS: UBICAR TRABAJADOR POR ÁREA	36
FIGURA 9 DIAGRAMA DE CLASES DEL ANÁLISIS: UBICAR TRABAJADOR POR TURNO	36
FIGURA 10 DIAGRAMA DE CLASES DEL ANÁLISIS: UBICAR TRABAJADOR POR LABORATORIO.....	37
FIGURA 11 DIAGRAMA DE CLASES DEL ANÁLISIS: GENERAR REPORTE	37
FIGURA 12 DIAGRAMA DE CLASES DEL ANÁLISIS: SELECCIONAR JEFE DE TURNO.....	38
FIGURA 13 DIAGRAMA DE CLASES DEL DISEÑO: GESTIONAR TRABAJADOR: SECCIÓN INSERTAR	40
FIGURA 14 DIAGRAMA DE CLASES DEL DISEÑO: GESTIONAR TRABAJADOR: SECCIÓN ELIMINAR	41
FIGURA 15 DIAGRAMA DE CLASES DEL DISEÑO: GESTIONAR TRABAJADOR: SECCIÓN MOSTRAR	42
FIGURA 16 DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR TRABAJADOR POR ÁREA	43
FIGURA 17 DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR TRABAJADOR POR TURNO	44
FIGURA 18 DIAGRAMA DE CLASES DEL DISEÑO: UBICAR TRABAJADOR POR ÁREA	45
FIGURA 19 DIAGRAMA DE CLASES DEL DISEÑO: UBICAR TRABAJADOR POR TURNO	46
FIGURA 20 DIAGRAMA DE CLASES DEL DISEÑO: UBICAR TRABAJADOR POR LABORATORIO.....	47
FIGURA 21 DIAGRAMA DE CLASES DEL DISEÑO: GENERAR REPORTE	48
FIGURA 22 DIAGRAMA DE CLASES DEL DISEÑO: SELECCIONAR JEFE DE TURNO.....	49
FIGURA 23 DIAGRAMA DE CLASES PERSISTENTES	50
FIGURA 24 DIAGRAMA ENTIDAD RELACIÓN.....	51
FIGURA 25 DIAGRAMA DE DESPLIEGUE	54
FIGURA 26 DIAGRAMA DE COMPONENTE GENERAL.....	55
FIGURA 27 DIAGRAMA DE COMPONENTE: GESTIONAR TRABAJADOR	56
FIGURA 28 DIAGRAMA DE COMPONENTE: MOSTRAR TRABAJADOR POR ÁREA	56
FIGURA 29 DIAGRAMA DE COMPONENTE: MOSTRAR TRABAJADOR POR TURNO	57
FIGURA 30 DIAGRAMA DE COMPONENTE: UBICAR TRABAJADOR POR ÁREA.....	57
FIGURA 31 DIAGRAMA DE COMPONENTE: UBICAR TRABAJADOR POR TURNO	58
FIGURA 32 DIAGRAMA DE COMPONENTE: UBICAR TRABAJADOR POR LABORATORIO.....	58
FIGURA 33 DIAGRAMA DE COMPONENTE: SELECCIONAR JEFE DE TURNO.....	59

FIGURA 34 DIAGRAMA DE COMPONENTE: GENERAR REPORTE	59
FIGURA 35 PROTOTIPO: GESTIONAR TRABAJADOR_SECCIÓN: INSERTAR	85
FIGURA 36 PROTOTIPO: GESTIONAR TRABAJADOR_SECCIÓN: ELIMINAR	85
FIGURA 37 PROTOTIPO: GESTIONAR TRABAJADOR_SECCIÓN: MOSTRAR	86
FIGURA 38 PROTOTIPO: UBICAR TRABAJADOR POR ÁREA	86
FIGURA 39 PROTOTIPO: MOSTRAR TRABAJADOR POR ÁREA	87
FIGURA 40 PROTOTIPO: MOSTRAR TRABAJADOR POR TURNO	87
FIGURA 41 PROTOTIPO: UBICAR TRABAJADOR POR TURNO	88
FIGURA 42 PROTOTIPO: SELECCIONAR JEFE DE TURNO	88
FIGURA 43 PROTOTIPO: UBICAR TRABAJADOR POR LABORATORIO.....	89
FIGURA 44 PROTOTIPO: GENERAR REPORTE	89
FIGURA 45 DIAGRAMA DE SECUENCIA: GESTIONAR TRABAJADOR_ SECCIÓN: MOSTRAR	90
FIGURA 46 DIAGRAMA DE SECUENCIA: GESTIONAR TRABAJADOR_ SECCIÓN: INSERTAR	91
FIGURA 47 DIAGRAMA DE SECUENCIA: GESTIONAR TRABAJADOR_ SECCIÓN: ELIMINAR	92
FIGURA 48 DIAGRAMA DE SECUENCIA: UBICAR TRABAJADOR POR ÁREA	93
FIGURA 49 DIAGRAMA DE SECUENCIA: UBICAR TRABAJADOR POR TURNO	94
FIGURA 50 DIAGRAMA DE SECUENCIA: UBICAR TRABAJADOR POR LABORATORIO.....	95
FIGURA 51 DIAGRAMA DE SECUENCIA: SELECCIONAR JEFE DE TURNO.....	96

TABLA 1 PROCESOS QUE SERÁN AUTOMATIZADOS	24
TABLA 2 DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA	31
TABLA 3 RELACIONES ENTRE CLASES MODELOS Y ENTIDADES.....	39
TABLA 4 CÁLCULO DE PUNTOS DE CASOS DE USO SIN AJUSTAR	61
TABLA 5 FACTOR DE PESO DE LOS ACTORES SIN AJUSTAR.....	62
TABLA 6 FACTOR DE PESO DE LOS CASOS DE USO SIN AJUSTAR	62
TABLA 7 CÁLCULO DE PUNTOS DE CASOS DE USO AJUSTADOS.....	63
TABLA 8 VALORES DEL FACTOR DE COMPLEJIDAD TÉCNICA.....	64
TABLA 9 VALORES DEL FACTOR DE AMBIENTE	65
TABLA 10 CÁLCULO DEL ESFUERZO.	65
TABLA 11 ESFUERZO DEL PROYECTO.....	66
TABLA 12 DESCRIPCIÓN TEXTUAL DEL CASO DE USO: GESTIONAR TRABAJADOR	77
TABLA 13 DESCRIPCIÓN TEXTUAL DEL CASO DE USO: UBICAR TRABAJADOR POR ÁREA	79
TABLA 14 DESCRIPCIÓN TEXTUAL DEL CASO DE USO: MOSTRAR TRABAJADOR POR ÁREA.....	79
TABLA 15 DESCRIPCIÓN TEXTUAL DEL CASO DE USO: MOSTRAR TRABAJADOR POR TURNO	80
TABLA 16 DESCRIPCIÓN TEXTUAL DEL CASO DE USO: UBICAR TRABAJADOR POR TURNO	81
TABLA 17 DESCRIPCIÓN TEXTUAL DEL CASO DE USO: SELECCIONAR JEFE DE TURNO	82
TABLA 18 DESCRIPCIÓN TEXTUAL DEL CASO DE USO: UBICAR TRABAJADOR POR LABORATORIO	84
TABLA 19 DESCRIPCIÓN TEXTUAL DEL CASO DE USO: GENERAR REPORTE	84

INTRODUCCIÓN

La Universidad de las Ciencias Informáticas (UCI) ha tomado un lugar en la sociedad no solo como una institución educativa, sino también productiva, contando con más de doscientos proyectos productivos destinados a la comercialización de software, la informatización de la sociedad cubana y a mejorar los servicios que se brindan en sus instalaciones.

Con el propósito de optimizar el trabajo que se realiza en los laboratorios de la UCI, surge el proyecto: **Sistema Integral de Gestión de los Laboratorios (SIGLA)**, cuyo objetivo fundamental es la elaboración de un sistema de software que permita resolver los principales problemas de esta área, producidos por la considerable matrícula que hoy tiene la universidad conjuntamente con el incremento de las instalaciones, el personal y medios destinados a los laboratorios. Como consecuencia a este incremento, las computadoras no satisfacen la cantidad de personas que existen en el centro. Además, no es posible brindar a los estudiantes y profesores el estado de los medios en los laboratorios, así como, los programas instalados en cada PC.

Las diferentes funcionalidades que debe cumplir el software para solucionar las dificultades mencionadas anteriormente dieron origen a los módulos de: Administración/Configuración, Control de Medios Informáticos, Reservación de tiempo de máquina, Control de Hardware, Control de Solicitud de Software, Gestión de Informe y Reporte, Gestión de Base de Datos y Control de Personal.

Actualmente la universidad cuenta con más de 500 personas que trabajan en el área de los laboratorios. De ellos, se registra gran cantidad de información relacionada con los datos personales y ubicación en un área, un turno y un laboratorio, en dependencia de las responsabilidades del trabajador. El proceso para efectuar estas tareas es de forma manual, resultando complejo registrar y mantener actualizados todos los datos. En caso de que se realice algún cambio, los documentos se envían por el correo electrónico y se almacenan en una o dos computadoras sin cuidados especiales, por lo que no es posible garantizar la seguridad de los datos del personal. Finalmente, los responsables de manejar la información, realizan varios listados por cada reporte que necesitan, generándose un gran volumen de datos, que en algunas ocasiones se encuentran repetidos.

Con el fin de solucionar las dificultades antes planteadas, el problema consiste en: ¿Cómo automatizar los procesos de Control del Personal en los laboratorios de la UCI?

Se ha identificado como objeto de estudio: los procesos de gestión de los laboratorios en la Universidad de las Ciencias Informáticas, y el campo de acción queda enmarcado en los principales procesos de automatización de la gestión del personal en los laboratorios de la UCI.

El Objetivo General del presente trabajo es: Elaborar el módulo “Control del Personal” del proyecto “Sistema Integral de Gestión de los Laboratorios” basándose en tecnologías Web.

Para dar cumplimiento al objetivo mencionado se trazaron las siguientes tareas de la investigación:

- Realizar un estudio de otros sistemas que brinden una solución similar a la que se quiere desarrollar, para obtener las buenas prácticas e identificar sus deficiencias.
- Analizar las tecnologías, lenguajes de programación y herramientas propuestas por el cliente para la realización del proyecto, caracterizando brevemente cada una de ellas.
- Representar los procesos involucrados en el negocio con el uso de IDEF0.
- Desarrollar el módulo “Control de Personal” de manera que cumpla las expectativas esperadas por el cliente.

Con la elaboración del módulo será posible mejorar el trabajo en los laboratorios de la universidad, por su utilidad para manejar la información del personal, así como, su ubicación en un puesto de trabajo. Además, le evitará al usuario el procesamiento manual de un gran volumen de información.

El contenido del presente Trabajo de Diploma está estructurado de la siguiente manera:

Capítulo 1 Fundamentación Teórica: Analiza las tecnologías, lenguajes y herramientas que serán empleadas en el desarrollo del sistema. Además, se efectúa un estudio de otros sistemas con el objetivo de obtener sus buenas ideas para el módulo “Control de Personal”.

Capítulo 2 Características del sistema: Describe detalladamente el problema existente y las características más importante del sistema a desarrollar. Además, se identifican los principales procesos del negocio y las personas que intervienen en el mismo.

Capítulo 3 Análisis y Diseño: Profundiza con el desarrollo del análisis y el diseño, los elementos y funcionalidades que tendrá el sistema, siguiendo la metodología RUP reconocida ampliamente en todo el mundo para el desarrollo de proyectos de calidad.

Capítulo 4 Implementación: Muestra los elementos fundamentales del Modelo de Implementación que son: el diagrama de despliegue y los diagramas de componentes correspondientes a cada caso de uso.

Capítulo 5 Estudio de la factibilidad: Muestra la estimación y cálculo de tamaño del software, así como el esfuerzo, los costos y beneficios que reportará el sistema.

CAPÍTULO 1 “Fundamentación teórica”

1.1. Introducción

Actualmente, el desarrollo de las tecnologías Web ha originado importantes cambios en las aplicaciones informáticas, por su capacidad para mantener actualizadas las mismas sin necesidad de distribuir el software preciso para su funcionamiento.

En el presente capítulo se realiza una breve caracterización de las tecnologías, lenguajes y herramientas propuestas por el cliente para el desarrollo del proyecto. Además, se estudian otros sistemas que gestionan la información referente al personal, con el objetivo de obtener sus aspectos positivos para la elaboración del módulo “Control de Personal”.

1.2. Términos utilizados

Área de trabajo: Se define como área de trabajo a cada una de las instalaciones docentes y productivas que contienen laboratorios en la UCI, estas áreas son: Infraestructura Productiva, Producción, Informatización y los docentes que existen hasta el momento.

Turno de trabajo: Es el horario que se le asigna a un trabajador para desempeñar sus labores, actualmente la universidad tiene reglamentado 4 turnos de trabajo.

Laboratorio: Local equipado con computadoras y los medios que la conforman.

Entidad: Término que identifica a una organización o institución.

Correo electrónico: Servicio de red que permite a los usuarios enviar y recibir mensajes mediante sistemas de comunicación electrónicos.

1.3. Sistemas para el control de personal

El objetivo esencial de esta sección es analizar algunos sistemas empleados en Cuba y el mundo para el control del personal. De esta forma, se identifican las dificultades que presentan y a la vez se toman en consideración sus buenas prácticas para la elaboración del módulo “Control de Personal”.

El primer sistema estudiado es **Gábilos**: diseñado especialmente para el Control de obras y personal de constructoras, empresas de carpintería, etc. Existen dos versiones del programa: una permite gestionar un máximo de dos empresas mientras que la otra versión posibilita gestionar innumerables empresas.

El software Gábilos presenta:

- Altas prestaciones (ágil navegación interna, facilidad de manejo, pantallas intuitivas).
- Ayuda permanente facilitada por los profesionales de Gábilos.
- Rapidez en el aprendizaje, ya que dispone de las herramientas necesarias (videos y demostraciones) para utilizar el software sin ninguna dificultad.

Además la aplicación funciona en el Sistema Operativo Windows en las versiones 95/98/NT/2000/XP.

Otra de las aplicaciones analizadas es: **ASSETS NS** la cual constituye un Sistema de Gestión Integral estándar y parametrizado que consta de varios módulos. Específicamente, el módulo de Recursos Humanos dispone de interfaces amigables con la posibilidad de controlar íntegramente los recursos laborales: empleados y estructura organizativa de la entidad. Asimismo, proporciona opciones de seguridad que le permiten limitar el acceso a los diferentes procesos del sistema de acuerdo con el perfil de cada usuario. ASSETS es una aplicación programada en Visual Basic 6.0 y Microsoft SQL Server 2000. Este sistema es muy empleado en Cuba por su capacidad para ser personalizado fácilmente, permitiendo su uso en entidades de diversa naturaleza.

Requerimientos Técnicos:

- Requerimientos mínimos: PC Pentium a 100 Mhz, 32 MB RAM, 100 MB de espacio disponible de disco duro, Windows 95 o superior.
- Requerimientos recomendados: PC Pentium a 166 Mhz, 64 MB RAM, 100 MB de espacio disponible de disco duro, Windows 95 o superior.

Posteriormente al estudio realizado se determinó que los sistemas citados presentan importantes ventajas para el desarrollo del módulo "Control de Personal", ya que se pretende elaborar un sistema con:

- Facilidad de manejo y una interfaz agradable e intuitiva para el usuario.
- Que brinde la información de los trabajadores con fotos e imágenes.

- Que sea capaz de controlar eficientemente al personal de los laboratorios.
- Que restrinja el acceso a los diferentes procesos del sistema de acuerdo con las responsabilidades de cada usuario.

Sin embargo, Gábilos y ASSETS necesitan de requerimientos específicos en las computadoras donde son instalados para que su funcionamiento sea correcto, mientras que para trabajar con el módulo Control de personal, solamente se requiere de un navegador, ya sea: Internet Explorer en versiones superiores a la 5.0 o Mozilla Firefox en versiones superiores a la 1.5, ambos con las opciones de JavaScript y cookies habilitadas. Gábilos y ASSETS están disponibles específicamente para la plataforma Windows, mientras que con el desarrollo del módulo se espera extender sus funcionalidades al sistema operativo Linux.

Es importante destacar que aunque los sistemas estudiados son muy eficientes para controlar el personal de una entidad, no solucionan los problemas que existen en el área de los laboratorios, como puede ser: la ubicación de los trabajadores en un puesto específico.

Finalmente, ASSETS es una aplicación programada en Visual Basic y en Microsoft SQL Server 2000, mientras que para el desarrollo del Sistema Integral de Gestión de los laboratorios (SIGLA) se emplearán tecnologías y herramientas Web muy novedosas que ayudarán a realizar satisfactoriamente un producto de calidad.

1.4. Tecnologías, lenguajes de programación y Librería

En la actualidad, gracias al avance de Internet, han surgido diversidad de tecnologías y lenguajes para la creación de aplicaciones Web, las cuales ofrecen grandes facilidades para establecer comunicación con el usuario mediante las páginas Web, por su capacidad para ser visualizadas desde cualquier parte del mundo haciendo uso de un navegador.

1.4.1. Lenguaje de programación

1.4.1.1. PHP

PHP es un lenguaje sencillo, de sintaxis cómoda y dispone de muchas librerías que facilitan en gran medida el desarrollo de las aplicaciones; convirtiéndolo en el favorito de millones de programadores en todo el mundo.

Características de PHP:

- Dispone de una conexión propia a varios sistemas de base de datos como: MySQL, PostgreSQL y Oracle.
- Incorpora bibliotecas que contienen funciones integradas para realizar útiles tareas relacionadas con la Web. Puede generar imágenes GIF al instante, establecer conexiones a otros servicios de red, enviar correos electrónicos, trabajar con cookies y generar documentos PDF, todo con unas pocas líneas de código.
- Es un producto de código abierto, soportado por una gran comunidad de desarrolladores que se encargan de encontrar y reparar los fallos de funcionamiento.
- Es un lenguaje multiplataforma.
- Permite las técnicas de Programación Orientada a Objetos.
- No requiere definición de tipos de variables.
- Posee tratamiento de errores.

PHP 5.2 es una versión de PHP que además incluye:

- Soporte sólido para Programación Orientada a Objetos (OOP) con PHP Data Objects.
- Mejoras de rendimiento.
- Mejor soporte a XML.

1.4.2. Librería ExtJS 2.0

ExtJS es una librería de componentes que facilita las herramientas necesarias para la creación de aplicaciones Web con excelentes gráficos; ya que posee una considerable colección de elementos para el diseño de interfaces, ventanas, pestañas, menús, tablas, etc.

Brinda soporte para:

- Construir interfaces gráficas complejas y dinámicas.
- Comunicar datos de forma asíncrona con el servidor.
- Diversos navegadores como: Internet Explorer, Firefox, Safari y Opera.

Actualmente ExtJS es considerado un FrameWork independiente; ya que a principios del 2007 se creó una compañía para comercializar y dar soporte al mismo, dicha compañía proporciona los servicios de consultoría necesarios para ayudar a los clientes en el aprovechamiento máximo de las ventajas de ExtJS. Es importante señalar que la ExtJS 2.0 tiene dos tipos de licencias, LGPL (Open Source) y la comercial, esta última es obligatoria si se desea obtener soporte.

1.4.2.1. AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML) facilita la creación de aplicaciones interactivas en la Web que se ejecutan en el navegador de los usuarios y mantienen comunicación asíncrona con el servidor. De esta forma, es posible realizar cambios sobre una página sin necesidad de recargarla, aumentando la interactividad, velocidad y usabilidad de la misma.

“AJAX no es una tecnología en sí mismo. En realidad, se trata de la unión de varias tecnologías que se desarrollan de forma autónoma y que se unen de formas nuevas y sorprendentes.”¹

¹ Eguíluz Pérez, Javier. 2007. Libros Web. *Introducción a AJAX*. [Online] 2007. [Cited: Enero 8, 2008.] <http://www.librosweb.es/ajax>.

² Eguíluz Pérez, Javier. 2007. Libros Web. *Introducción a XHTML*. [Online] 2007. [Cited: Enero 15, 2008.]

AJAX está conformado por:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

Además:

- Provee un mecanismo para mezclar y hacer coincidir XML con XHTML.
- Las aplicaciones son más rápidas e interactivas, al estilo de aplicaciones de escritorio.
- Reduce de manera significativa tener que cargar información continuamente del servidor, actualizando solamente porciones de la página.
- Cuando se utiliza AJAX adecuadamente en el desarrollo de una aplicación, se reduce de manera significativa los tiempos de carga inicial.

A continuación se explican las características más importantes de algunas de las tecnologías que componen AJAX:

JSON: Notación de Objetos de JavaScript (JavaScript Object Notation), constituye un formato ligero para el intercambio de datos abierto y basado en texto. Está pensado principalmente para usarse en aplicaciones Web con el objetivo de transmitir información estructurada de forma asíncrona entre el servidor y los clientes.

JSON es un subconjunto de la notación literal de objetos de Javascript que se usa generalmente con ese lenguaje. Sin embargo, los tipos básicos y las estructuras de datos de la mayoría de los lenguajes de programación también pueden ser representados en JSON, por lo que es posible usar el formato para intercambiar datos estructurados entre programas escritos en diferentes lenguajes como: ActionScript, C, C#, Java, JavaScript, Perl, PHP, Python, Ruby y Smalltalk.

Características de JSON:

- La sintaxis es muy concisa por lo que requiere menos codificación y procesamiento.

- No se necesita código de aplicación adicional para analizar texto.
- Es legible e independiente de la plataforma.
- Soporta diversos lenguajes de programación.

XHTML: “XHTML es el acrónimo en inglés de Extensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto). Es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de remplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos.”²

XHTML reúne la capacidad de formato de HTML y se consolida con la formalidad del XML (y sus reglas) a la hora de estructurar documentos para la portación de datos. Está encaminado al uso de un etiquetado correcto, por lo que exige una serie de requisitos básicos a cumplir en cuanto al código.

Algunos de estos requisitos son:

- Elementos correctamente anidados.
- Etiquetas en minúsculas.
- Elementos cerrados correctamente.
- Atributos de valores entrecomillados.

CSS: Es un lenguaje de hojas de estilos (Cascading Style Sheets) creado para controlar la presentación de documentos estructurados y escritos en XHTML. Algunos de los aspectos manejados son: el color, el tamaño, el tipo de letra, la separación entre párrafos y la tabulación con la que se muestran los elementos de una lista. El propósito del desarrollo de CSS es separar la estructura y el

² Eguíluz Pérez, Javier.2007. Libros Web. *Introducción a XHTML*. [Online] 2007. [Cited: Enero 15, 2008.] <http://www.librosweb.es/xhtml>.

contenido de la presentación estética en un documento, esto permite un control mayor del documento y sus atributos, convirtiendo al XHTML en un documento muy versátil y liviano.

“Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.”³

Entre los beneficios concretos de CSS se encuentran⁴:

- Control de la presentación de muchos documentos desde una única hoja de estilo.
- Control más preciso de la presentación.
- Aplicación de diferentes presentaciones a diferentes tipos de medios (pantalla, impresión entre otros).

JavaScript: Es un lenguaje basado en objetos y guiado por eventos, diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet. Los programas escritos con este lenguaje se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios, convirtiéndolo en un lenguaje interpretado.

Ventajas de JavaScript:

- Los programas escritos en este lenguaje no requieren de mucha memoria ni tiempo adicional de transmisión, por ser pequeños y compactos.

³ **Eguíluz Pérez, Javier. 2007.** Libros Web. *Introducción a CSS*. [Online] 2007. [Cited: Enero 20, 2008.] <http://www.librosweb.es/css>.

⁴ **HTML.net.** Tutoriales sobre HTML y CSS - Construye tu propio sitio web. [Online] [Cited: Marzo 15, 2008.] <http://es.html.net/tutorials/css/lesson1.asp>.

- JavaScript no requiere un tiempo de compilación; ya que los scripts se pueden desarrollar en un período de tiempo relativamente corto.
- Es independiente de la plataforma hardware o sistema operativo, y funciona correctamente siempre y cuando exista un navegador con soporte JavaScript.

1.4.3. Servicios Web

Los servicios Web pueden definirse como un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones con el objetivo de brindar servicios. Las aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios Web para intercambiar datos en redes de ordenadores como Internet, lo que contribuye a sistemas más flexibles y estables.

Algunos de los estándares utilizados en los servicios Web son:

- XML: Es el formato estándar para los datos que se vayan a intercambiar.
- SOAP: Protocolo sobre el que se establece el intercambio.
- WSDL: Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web.
- UDDI: Protocolo para publicar la información de los servicios Web.
- Otros protocolos: los datos en XML también pueden enviarse de una aplicación a otra mediante protocolos normales como HTTP o FTP.

Los servicios Web aportan las siguientes ventajas:

- Aportan interoperabilidad entre diferentes aplicaciones de software, independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Fomentan los estándares y protocolos basados en texto.
- Al apoyarse en HTTP, permiten acceder a cualquier sistema conectado a la red.

- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar.

1.4.4. Servidor Web

Apache es un servidor Web potente, flexible y disponible para distintas plataformas y entornos. Es altamente configurable de diseño modular, posibilitando que los administradores de sitios Web puedan elegir los módulos que serán incluidos y ejecutados en el servidor.

Características de Apache:

- Es una tecnología gratuita y de código abierto, lo que proporciona transparencia en todo el proceso de instalación.
- Es prácticamente universal, por su disponibilidad en multitud de sistemas operativos.
- Posee una alta configurabilidad en la creación y gestión de logs, de este modo es posible tener un mayor control sobre lo que sucede en el servidor.

Este servidor Web tiene una fácil integración con varios lenguajes de programación como: Java, Perl y especialmente PHP. Dicha relación ha dado lugar al desarrollo de aplicaciones como el APPSERV y XAMPP, los cuales instalan el Apache y el PHP configurados para su uso.

1.4.5. Sistema Gestor de Base de Datos

PostgreSQL es un gestor de bases de datos relacional orientado a objetos, libre y gratuito. Posee una gran escalabilidad, ya que es capaz de soportar una gran cantidad de peticiones simultáneas, ajustándose al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima.

Presenta las siguientes propiedades:

- Atomicidad: Asegura la realización de una operación, por lo que ante un fallo del sistema esta no queda a medias.
- Consistencia: Posibilita la ejecución de aquellas operaciones que no van a romper las reglas y directrices de integridad de la base de datos.
- Durabilidad: Asegura la permanencia de una operación realizada, y aunque falle el sistema esta no podrá deshacerse.

- Aislamiento: Mediante un sistema denominado MVCC (Acceso concurrente multiversión) asegura que una operación no pueda afectar a otras, de esta manera dos transacciones sobre la misma información no genera error.

PostgreSQL presenta las siguientes características:

- Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos.
- Posee confiabilidad, seguridad y flexibilidad.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Soporta el uso de índices, reglas y vistas.

1.5. Arquitectura

1.5.1. Patrón Modelo-Vista-Controlador (MVC)

El Modelo-Vista-Controlador (MVC) se creó para Smalltalk a finales de los setenta; desde entonces su uso se ha ido extendiendo cada día más para la construcción de sistemas con interfaz gráfica. MVC es un patrón de arquitectura utilizado en sistemas Web para separar los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos, permitiendo flexibilidad y facilidad a la hora de hacer futuros cambios.

El **Modelo** representa las estructuras de datos. Típicamente el modelo de clases contendrá funciones para consultar, insertar y actualizar información de la base de datos.

La **Vista** es la encargada de presentar la información al usuario. Una vista puede ser una página Web o parte de una página.

El **Controlador** actúa como intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para generar una página, es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo.

1.5.2. Arquitectura Cliente/Servidor

La arquitectura cliente-servidor es una nueva tendencia en el desarrollo de redes locales, que tiene como objetivo optimizar el uso tanto del hardware como del software a través de la separación de funciones: el cliente, que maneja la porción de la aplicación y el servidor que administra los procesos de almacenamiento y recuperación de los datos.

Puede presentarse como uno a varios clientes y uno o más servidores, junto con un sistema operativo y una plataforma de comunicación para formar un sistema cooperativo que permita la computación distribuida, el análisis y la presentación de datos. Un único servidor típicamente sirve a una multitud de clientes, ahorrando a cada uno de ellos el problema de tener la información instalada y almacenada localmente.

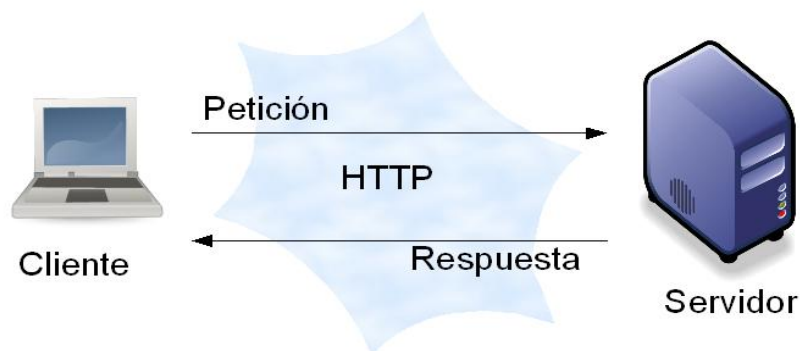


Figura 1 Arquitectura Cliente/Servidor

Características de la arquitectura Cliente/Servidor:

- El servidor presenta una interfaz única y bien definida a todos sus clientes.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor no afectan al cliente.

1.6. Modelación de Funciones mediante IDEF0

La traducción literal de las siglas IDEF es Integration Definition for Function Modeling (Definición de la integración para la modelización de las funciones). IDEF consiste en una serie de normas que definen la metodología para la representación de funciones. Representa de manera estructurada y jerárquica las actividades que conforman un sistema o empresa y los objetos o datos que soportan la interacción de esas actividades.

IDEF0 posee las siguientes características:

- Es capaz de representar gráficamente los principales procesos del negocio de una empresa a cualquier nivel de detalle.
- Permite analizar, documentar y mejorar los procesos del negocio.
- Facilita la comunicación y captura de información.
- Es muy coherente y simple, pero a la vez rigurosa y exacta.
- Puede ser generado fácilmente por una gran variedad de herramientas gráficas en computadores.
- Posee flexibilidad ante cambios.

La notación IDEF puede ser representada de dos formas:

La forma **AS-IS** (tal como es) es la empleada por el negocio para representar los procesos a la manera actual, antes de la automatización.

La forma **TO-BE** (como debe ser) es utilizada para transformar el proceso a la manera deseada para los objetivos de la organización.

1.7. Lenguaje de modelado y metodología de desarrollo

1.7.1. Lenguaje de modelado UML

UML es un lenguaje utilizado para el modelado de un sistema, permitiendo en mayor o menor medida representar todas las disciplinas de un proyecto informático: desde el análisis con los casos de uso, hasta la implementación y configuración mediante los diagramas de despliegue.

UML permite:

- Visualizar gráficamente un sistema de manera que otros puedan entenderlo.
- Especificar cuáles son las características de un sistema antes de su construcción.
- Construir sistemas diseñados a partir de modelos especificados.
- Documentar los elementos gráficos del sistema desarrollado para futuras revisiones.

Aporta las siguientes ventajas:

- Permite realizar una verificación y validación del modelo realizado.
- El modelado con UML es independiente del lenguaje de implementación, de tal forma que los diseños realizados se pueden implementar en cualquier lenguaje.
- Permite generar código a partir de los modelos y a la inversa, lo que posibilita la constante actualización.
- Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es suficientemente expresivo como para modelar sistemas que no son informáticos, como: flujos de trabajo en una empresa y diseño de la estructura de una organización.

1.7.2. Metodología de desarrollo RUP

“Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipo de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.”⁵

⁵ **Rumbaugh, James, Jacobson, Ivar and Booch, Grady.** 2000. El Lenguaje Unificado de Modelado. [Online] 2000. [Cited: Enero 10, 2008.] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.

RUP (Rational Unified Process) es un proceso de desarrollo de software que junto al Lenguaje Unificado de Modelado (UML), constituye una metodología estándar muy utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

- Se utiliza en proyectos que se desarrollan a largo plazo.
- Permite una mejor comunicación entre ingenieros de software y de negocio, manejando un lenguaje común para facilitar la comprensión de procesos del negocio y su traducción a las funcionalidades que brindará el sistema.
- Genera un volumen considerable de documentación, posibilitando que los cambios realizados en los miembros del equipo no resulte un factor negativo para el avance del proyecto.
- Propone el desarrollo en ciclos e iteraciones con los artefactos que se generan, siendo esto un elemento importante para alcanzar una categoría de certificación en el desarrollo del software.
- Asegura la producción de software de calidad dentro de plazos y presupuestos predecibles.

El ciclo de vida de RUP se caracteriza por:

- Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de este momento los casos de uso guían el proceso de desarrollo, pues los modelos que se obtienen en los diferentes flujos de trabajo, representan la realización de los casos de uso.
- Centrado en la arquitectura: La arquitectura muestra la visión común del sistema en la que el equipo del proyecto y los usuarios deben estar de acuerdo. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.
- Iterativo e incremental: Propone que cada fase se desarrolle en iteraciones que involucran actividades de todos los flujos de trabajo. Es práctico dividir el trabajo en partes más pequeñas o miniproyectos. Cada miniproyecto es una iteración que resulta un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

1.8. Herramientas utilizadas

1.8.1. Visual Paradigm

Visual Paradigm para UML (VP-UML) es una poderosa herramienta CASE multiplataforma (Windows/Linux/Mac OS X) que soporta el ciclo de vida completo del desarrollo de software. Está diseñado para un amplio rango de usuarios, incluyendo ingenieros de software, analistas de sistema, analistas de negocio, arquitectos de sistema y quienes estén interesados en la construcción de sistemas de software confiables mediante el uso de la Orientación a Objetos.

Este software facilita una rápida construcción de aplicaciones de calidad y a un menor coste. Visual Paradigm para UML soporta un conjunto de lenguajes (Java, C + +, PHP, Ada y Python), tanto en generación de código como ingeniería inversa.

Entre sus características principales se pueden citar:

- Soporta UML en su versión 2.1.
- Facilita la comunicación de todo el equipo de desarrollo mediante el uso de un lenguaje estándar común.
- Posibilita el desarrollo de la ingeniería directa e inversa.
- Durante todo el ciclo de desarrollo el modelo y el código permanecen sincronizados, permitiendo la generación de código a partir de diagramas y viceversa.
- Se encuentra disponible en múltiples versiones y plataformas.
- Permite la generación de bases de datos a partir de la transformación de diagramas de Entidad-Relación en tablas de base de datos y viceversa.

1.8.2. Aptana

Aptana Studio es un IDE de desarrollo para aplicaciones de la Web 2.0, gratuito y de código fuente abierto. Está basado en el conocido entorno de desarrollo Eclipse, también de código abierto; pero mientras que Eclipse está focalizado en el desarrollo para Java, Aptana Studio es una distribución focalizada en el desarrollo Web, con soporte a HTML, CSS y JavaScript.

Soporta varias librerías como: ExtJS, Prototype, Yahoo UI y JQuery, pudiendo combinarlas fácilmente en una aplicación. Aptana Studio está disponible como una aplicación independiente o como plug-in para Eclipse y se puede encontrar para tres plataformas fundamentales: Windows, Mac OS y GNU/Linux.

La gestión de proyectos, vista previa, autocompletado de código y gestión de documentación, son algunas de las características similares que presenta Aptana con otros entornos de desarrollo integrado (Eclipse, C++ Builder, Visual Studio. NET), pero además:

- Visualiza los errores de sintaxis a medida que se escribe.
- Presenta una previa visualización de estilos CSS con el editor CSS.

1.8.3. GIMP

GIMP es el acrónimo para GNU Image Manipulation Program. Es un programa libre distribuido bajo la licencia GPL, apropiado para tareas como: arreglo fotográfico, composición y edición de imagen. Es especialmente útil para la creación de logotipos y otros gráficos para páginas Web. Inicialmente se desarrolló para sistemas Unix; sin embargo, actualmente existen versiones totalmente funcionales para Windows y para Mac OS.

Características fundamentales de GIMP:

- Lee y escribe la mayoría de los formatos de ficheros gráficos, entre ellos: jpg, gif, png, pcx y tiff.
- Es capaz de importar ficheros en pdf y también imágenes vectoriales en formato svg.
- Permite la automatización de muchos procesos mediante macros o secuencias de comandos.
- Dispone de varias herramientas o filtros para la manipulación de los colores y el aspecto de las imágenes.

1.8.4. PGAdmin III

Es una aplicación gráfica usada para la gestión de PostgreSQL, siendo la más completa y popular con licencia Open Source. PGAdmin está escrito en C++ y utiliza la librería gráfica multiplataforma wxWidgets, permitiendo que se pueda usar en sistemas operativos como: GNU/Linux, FreeBSD, Solaris, Mac OS y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3, ejecutándose en cualquier plataforma.

PGAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples, hasta desarrollar bases de datos complejas.

El producto también incluye:

- Editor SQL con resaltado de sintaxis.
- Editor de código de la parte del servidor.
- Amplia documentación para ayudar a los usuarios menos experimentados.

1.9. Conclusiones del capítulo

En este primer capítulo se ofreció una visión general de algunos sistemas que se emplean en el mundo para controlar al personal de instituciones y empresas; mediante el estudio realizado a estas aplicaciones se concluyó que poseen características relevantes a tener en cuenta, pero de manera general no resuelven las dificultades existentes en los laboratorios de la UCI.

Teniendo en cuenta que el desarrollo de las tecnologías y los sistemas que las emplean se perfeccionan cada día, se decidió implementar la aplicación con el uso de: PHP como lenguaje de programación y la librería de componentes ExtJS 2.0 para crear la interfaz. Además, se utilizarán: RUP como metodología de desarrollo, IDEF0 para la representación de los procesos del negocio y UML como lenguaje de modelado.

CAPÍTULO 2 “Características del sistema”

2.1. Introducción

En el presente capítulo se describen los procesos del negocio correspondientes a la gestión del personal de los laboratorios, con el objetivo de comprender la estructura y dinámica de la entidad (UCI) donde se implantará el sistema; para ello resulta elemental conocer la situación problemática, ya que satisfacer las necesidades del cliente es la meta de todo producto de software. Además, se exponen los requerimientos y las funcionalidades que debe cumplir la aplicación para tener la confiabilidad y calidad esperada por los usuarios.

2.2. Objeto de estudio

2.2.1. Definición del problema

Actualmente, la universidad cuenta con más de 500 personas que trabajan en el área de los laboratorios. De ellos, se registra gran cantidad de datos, resultando complejo llevar el control requerido de tanta información.

Después de analizar y estudiar la situación se detectaron los siguientes inconvenientes:

- La información del personal es controlada de forma manual, lo que trae como consecuencia que se produzcan errores en el momento de interactuar (introducir, eliminar) con los datos.
- Cuando se realiza algún cambio en la información, los documentos son enviados por correo electrónico y almacenados en una o dos computadoras sin cuidados especiales, por lo que no es posible garantizar la seguridad e integridad de los datos del personal.
- Existen gran cantidad de datos repetidos, debido a que los responsables de controlar la información realizan varios listados por cada reporte que necesitan, aún cuando eso implique tener los mismos datos en distintos lugares.

De esta manera surge el problema referido a: ¿Cómo automatizar los procesos de Control del Personal en los laboratorios de la UCI?, ya que los métodos empleados no son los más eficientes para realizar

este trabajo, que por su importancia requiere de la mayor responsabilidad por parte de todos los implicados.

2.2.2. Objeto de automatización

Con la realización del sistema se pretende automatizar los siguientes procesos:

Procesos	Descripción
Registrar datos del trabajador	Debe permitir buscar a un trabajador dado el solapín o el listado de los trabajadores que no han sido registrados en el área de los laboratorios con el uso de los servicios Web. A partir de los resultados obtenidos se seleccionan el o los trabajadores que serán aceptados para trabajar en dicha área.
Mostrar todos los trabajadores	Debe permitir mostrar los siguientes datos de cada trabajador: Nombre, Apellidos, Usuario y Solapín, sin tener en cuenta el área y el turno al que pertenecen.
Eliminar datos del trabajador	Debe permitir buscar al trabajador que se desea eliminar dado el solapín.
Mostrar trabajadores por área Mostrar trabajadores por turno	Ambos procesos deben permitir mostrar los siguientes datos de cada trabajador: Nombre, Apellidos, Usuario, Solapín y Categoría. Además si se desea visualizar otros datos como: Foto, Dirección, Carnet y Cargo Ocupacional, solamente es necesario seleccionar la persona.
Ubicar trabajador por área Ubicar trabajador por turno Ubicar trabajador por laboratorio	Los tres procesos deben permitir buscar a un trabajador dado el solapín o el listado de los trabajadores que no han sido ubicados en un área, un turno o un laboratorio. A partir de la opción seleccionada y los resultados obtenidos debe ser posible ubicar al trabajador, dependiendo del proceso que se esté efectuando.
Seleccionar Jefe de Turno	Debe permitir mostrar un listado de trabajadores dado un turno seleccionado y a partir del resultado obtenido es posible escoger

	al responsable de controlar y ubicar al personal en un laboratorio.
Generar Reporte	Debe permitir crear reportes a partir de cualquier criterio de búsqueda seleccionado. Los criterios son: área, fecha de alta, nivel de escolaridad, militancia y cargo ocupacional.

Tabla 1 Procesos que serán automatizados

2.2.3. Información que se maneja

La información que se maneja en el sistema está relacionada fundamentalmente con los datos personales y laborales de las personas que trabajan en el área de los laboratorios.

Algunos de estos datos son:

- Nombres, Apellidos, Solapín, Usuario, Cargo Ocupacional, Nivel Escolar, Dirección, Fecha de alta, Teléfono y Militancia.

Después que el personal es ubicado en un área, en un turno y en un laboratorio, adquieren nuevos datos referentes a su ubicación.

2.3. Propuesta del sistema

Se ha decidido darle solución al problema existente con una aplicación Web, que se encargue de controlar el personal de los laboratorios y facilite el trabajo de los usuarios. Dicho sistema debe permitir introducir los datos más importantes de los trabajadores a partir de los servicios Web que están disponibles en la universidad y que brindan funcionalidades específicas para ofrecer la información necesitada. Una vez que los datos estén registrados, la aplicación debe permitir ubicar a los trabajadores en un área de trabajo, en un turno y en un laboratorio. De igual forma, el sistema debe ser capaz de: Seleccionar los jefes de turno correspondientes a cada turno de trabajo, Generar reportes a partir de criterios de búsqueda y Mostrar la principal información del personal como: Nombre, Apellidos, Solapín, Carnet de Identidad y Dirección Particular.

De manera general se pretende que la aplicación muestre una interfaz agradable e intuitiva; de modo que personas con poco conocimiento de informática puedan interactuar fácilmente con el sistema. Es

importante señalar que para la realización de la aplicación se utilizarán los Servicios Web con el objetivo de mitigar riesgos y mejorar los procesos de negocio.

En la Figura 2 se muestra la interacción que se realiza en el sistema para el intercambio de información entre la PC Cliente, los servidores y los servicios Web.

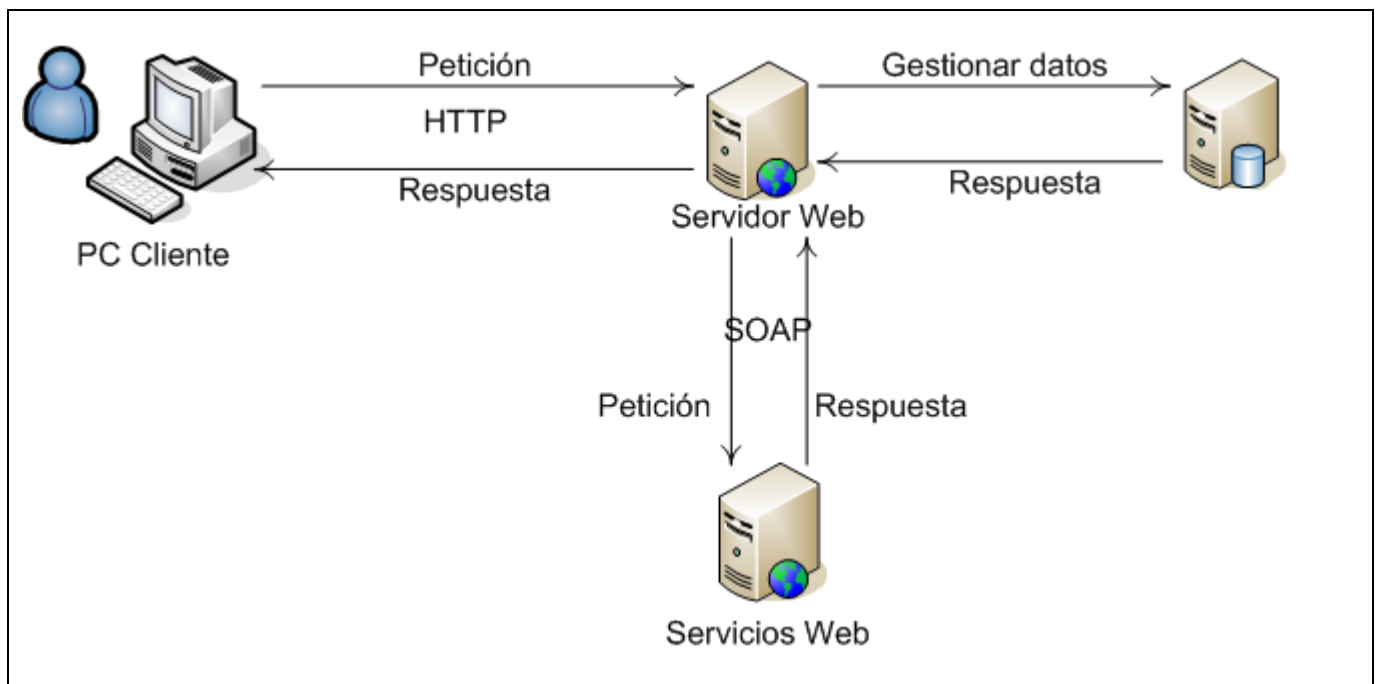


Figura 2 Intercambio de datos entre la PC Cliente y los servidores

El usuario desde su computadora realiza una petición al servidor Web donde se encuentra instalado el Apache, este a su vez consulta al servidor de Base de datos, obteniendo los datos necesarios para realizar las búsquedas en los Servicios Web. Después de obtener los resultados el servidor Web los envía a la PC Cliente del usuario que inició la petición.

2.4. Modelo del Negocio

Para el modelado del negocio se hará uso de IDEF0, por su facilidad para lograr un entendimiento común entre desarrolladores y clientes en la representación de los procesos fundamentales. Como resultado del estudio realizado en el negocio, se definió un proceso nombrado: Proceso Ubicar trabajador en puesto de trabajo, internamente conformado por 5 procesos más pequeños que serán explicados posteriormente.

2.4.1. Representación de los procesos con IDEF0

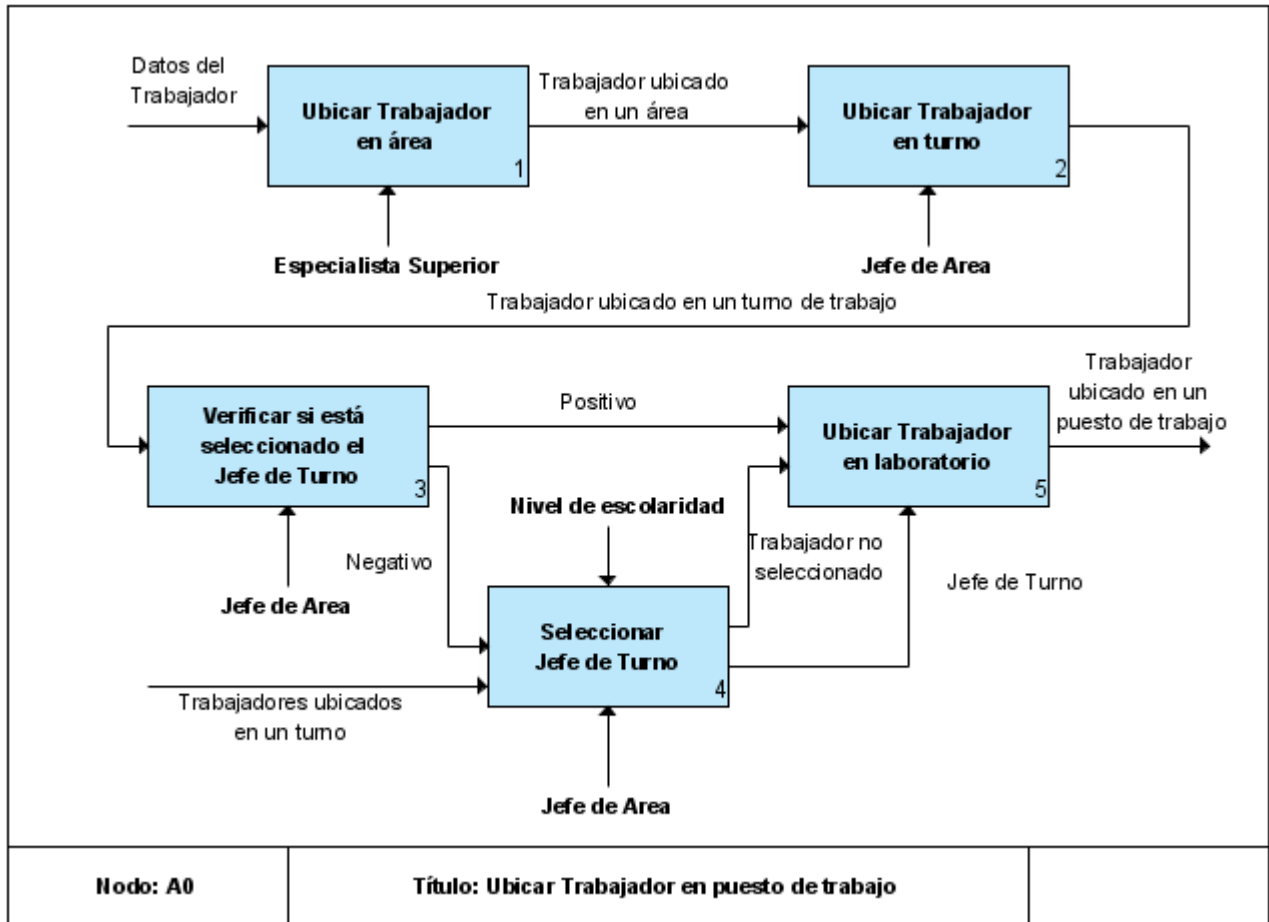


Figura 3 Proceso Ubicar Trabajador en puesto de trabajo

2.4.2. Descripción detallada de los procesos del negocio

- Ubicar Trabajador en área: Este proceso recibe los datos del trabajador que va a ser ubicado en un área de trabajo. Los datos solicitados son los siguientes: Solapín, Adiestrado, Fecha de Alta, Nombres y apellidos, No de CI, Dirección Particular, Teléfono, Militante y Nivel de escolaridad. El resultado final de este proceso es un trabajador ubicado en un área.
- Ubicar Trabajador en turno: Este proceso recibe como entrada al trabajador ubicado en las distintas áreas de la universidad, donde el jefe de área debe ubicarlo un turno de trabajo. De esta manera, el trabajador queda asignado a un turno de trabajo.
- Verificar si está seleccionado el Jefe de Turno: Este proceso recibe al trabajador ubicado en un turno de trabajo, el jefe de área debe verificar si cada turno tiene un responsable (jefe de turno). En

caso positivo (estén seleccionados) se procede a realizar el proceso de Ubicar Trabajador en laboratorio con los trabajadores que tienen asignado un turno, y en caso contrario se realiza el proceso Seleccionar Jefe de Turno.

- Seleccionar Jefe de Turno: Este proceso recibe como entrada a los Trabajadores ubicados en un turno de trabajo junto a la respuesta negativa de que falta seleccionar al responsable, el jefe de área selecciona al jefe de turno teniendo en cuenta el nivel educacional. En caso, de que algunos trabajadores no sean elegidos, estos constituyen una entrada para el proceso de Ubicar Trabajador en Laboratorio. De esta forma quedan elegidos los responsables de cada turno.
- Ubicar Trabajador en laboratorio: Para la realización de este proceso es necesario que el trabajador esté ubicado en un turno de trabajo y estén seleccionados los jefes de turno, ambos resultados se obtienen de las salidas anteriores. El jefe de turno se encarga de ubicar al trabajador en un laboratorio y de esta forma todos quedan ubicados en un puesto de trabajo.

2.5. Especificación de los requisitos de software

2.5.1. Requerimientos funcionales

R1 - Administrar información.

R1.1. Registrar Trabajador.

R1.2. Eliminar datos del Trabajador.

R2 - Mostrar información de Trabajadores.

R2.1. Mostrar listado general.

2.1.1. Nombres y Apellidos.

2.1.2. Usuario.

2.1.3. Solapín.

2.1.4. Área.

R2.2. Mostrar listado por área.

2.2.1. Nombres y Apellidos.

2.2.2. Usuario.

2.2.3. Categoría.

2.2.3. Solapín.

2.2.4. Turno de Trabajo.

R2.3. Mostrar listado por turno de trabajo.

2.2.1. Nombres y Apellidos.

2.2.2. Usuario.

2.2.3. Solapín.

2.2.4. Categoría.

En los mostrar listado por área y por turno se brinda además los detalles de cada trabajador, estos datos son: Foto, Carnet de identidad, Dirección, Teléfono y Cargo Ocupacional.

R3 - Ubicar el personal en un área de trabajo.

R4 - Ubicar el personal en un turno de trabajo.

R5 - Ubicar el personal en los laboratorios.

R6 - Reportes de trabajadores por criterio de búsqueda.

Algunos de los criterios de búsqueda son: área, fecha de alta, cargo ocupacional, graduado y militancia.

R7 - Seleccionar Jefes de Turno por área.

2.5.2. Requerimientos no funcionales

Los requerimientos no funcionales que se muestran a continuación, especifican de manera general las propiedades que tendrá el Sistema Integral de Gestión de los laboratorios.

Requerimientos de Seguridad:

- La información será almacenada en una base de datos, dejando registradas todas las operaciones mediante copias de seguridad.
- El uso y manejo del sistema estará controlado; ya que la información podrá ser consultada y modificada solamente por el personal autorizado; estableciendo para ello una serie de roles con funcionalidades específicas.
- La aplicación estará funcionando a tiempo completo; de esta forma es posible que los usuarios tengan acceso (según sus permisos) en todo momento a la información solicitada.
- Es muy importante que los servidores estén sincronizados correctamente con el día y hora, ya que si ocurre un fallo el sistema no funcionaría correctamente.

Requerimientos de Confiabilidad:

- Todas las salidas del sistema tendrán 100% de veracidad y precisión.
- El sistema tendrá la capacidad de recuperarse rápidamente ante cualquier fallo mediante las copias de seguridad que serán realizadas.
- Se verificará constantemente la conexión con el servidor y en caso de fallo se le notificará al usuario.
- En caso de alguna dificultad con el funcionamiento del sistema, el tiempo medio de reparación deberá ser menor de 1 día.

Requerimientos de Rendimiento:

- El sistema debe tener un tiempo de respuesta rápido y eficiente, inferior a 10 segundos.
- La aplicación debe ser capaz de soportar gran cantidad de usuarios conectados simultáneamente.

Requerimientos de apariencia o interfaz externa:

- La interfaz será agradable, sencilla y sugerente; ya que su diseño simulará una aplicación de escritorio, presentará por defecto un color azul claro y textos que muestran la acción a realizar con un tipo de letra Verdana de tamaño 10pt.
- Se le brindará al usuario la posibilidad de personalizar la apariencia de interfaz externa.
- Debe contener pocas imágenes y gráficos para acelerar la velocidad de respuesta del sistema; por lo que se emplearán íconos sencillos que representarán la operación a realizar.
- El sistema será diseñado para una resolución de pantalla de 1024 X 768 píxeles.

Requerimientos de Hardware:

El servidor Web y de base de datos deben tener como mínimo las siguientes características:

- Computador PENTIUM IV.
- 1 GB de memoria RAM.
- 10 GB de espacio libre en disco duro.

Las computadoras situadas en los puestos de trabajo de los usuarios requieren como mínimo:

- Computador PENTIUM II.

- 128 MB de memoria RAM.

Además es necesario contar con una impresora para poder imprimir los reportes y tarjetas de red de 10/100 MB/s para la conexión.

Requerimientos de software:

- Las estaciones de trabajo clientes utilizarán como sistema operativo GNU/Linux en sus diversas distribuciones o la familia de Windows superior a Windows 98.
- Las máquinas clientes contarán con un navegador, ya sea el Internet Explorer en versiones superiores a la 5.0 o Mozilla Firefox en versiones superiores a la 1.5, ambos con las opciones de JavaScript y cookies habilitadas.
- El servidor Web funcionará sobre el sistema operativo GNU/Linux y tendrá instalado el Apache en su versión 2.0 y PHP 5.2 con las librerías PDO, pgsq y json habilitadas.
- El servidor de base de datos funcionará sobre el sistema operativo GNU/Linux y tendrá instalado el gestor de bases de datos PostgreSQL 8.2.

Requerimientos de Soporte:

- El sistema contará con la documentación apropiada para agilizar su mantenimiento y configuración.
- Se ofrecerán servicios de adiestramiento al personal que trabajará con el software.

Requerimientos de Usabilidad:

- La interfaz será fácil de usar para los diversos usuarios que interactúen con ella.
- El sistema estará bien documentado, con el fin de lograr un mejor uso de los servicios que este ofrecerá, para ello se realizará una ayuda que explique paso a paso cada una de las funcionalidades del software.

Requerimientos de Ayuda y documentación en línea:

- El sistema proporcionará en todo momento la documentación necesaria para que los usuarios puedan acceder a la misma en caso de que necesiten realizar alguna consulta.

Requerimientos legales:

- Los módulos de desarrollo del proyecto SIGLA así como la documentación del mismo, son propios de la UCI y específicamente del área de los laboratorios.

2.6. Modelo del sistema

2.6.1. Actores del sistema

Actor	Descripción
Especialista Superior de Formación	Es encargado de ubicar en un área de trabajo, registrar y eliminar a un trabajador. Además se comporta como Visor de información por área.
Jefe de Turno	Es el responsable de ubicar a cada trabajador en un laboratorio. Además se comporta como Visor de información por turno.
Jefe de Área	Es el responsable de ubicar al trabajador en un turno de trabajo, seleccionar los jefes de turnos. Además, se comporta como Visor de información por área y Visor de información por turno.
Visor de información por área (VIA)	Es el encargado de visualizar los datos fundamentales de los trabajadores, dada un área de trabajo.
Visor de información por turno (VIT)	Es el encargado de visualizar los datos fundamentales de los trabajadores, dado un turno de trabajo.

Tabla 2 Descripción de los actores del sistema

2.6.2. Diagrama de caso de uso del sistema

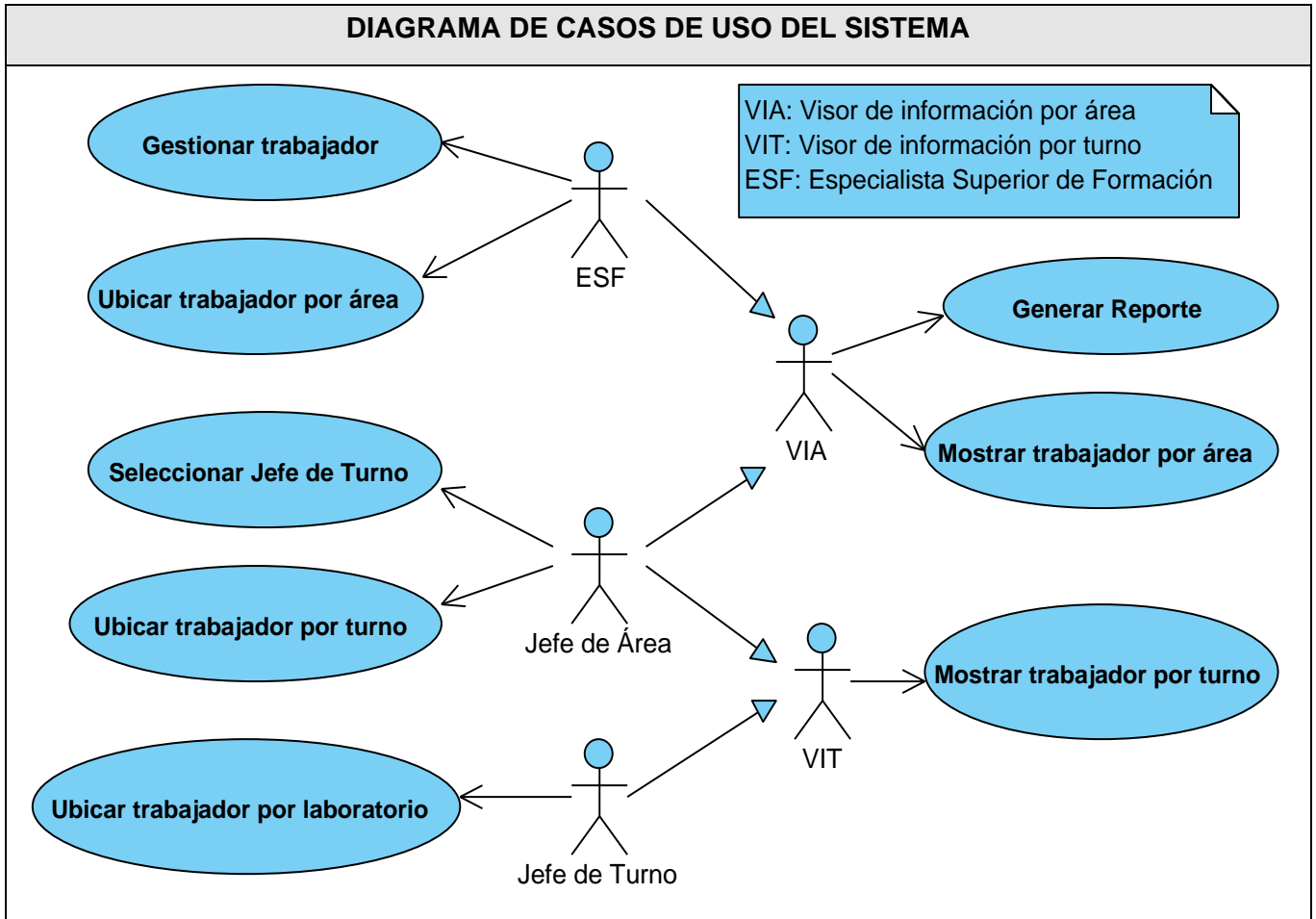


Figura 4 Diagrama de casos de uso del sistema

2.6.3. Descripción de los casos de uso del sistema

La descripción de los casos de uso del sistema constituye una parte muy útil, ya que se explica de forma detallada la interacción entre el sistema y el usuario. El Anexo_1 muestra la descripción de cada uno de los casos de uso del módulo Control de Personal.

2.7. Conclusiones

En este capítulo se realizó un estudio detallado de los procesos fundamentales del negocio que sirvieron como guía para la identificación de las funcionalidades y requerimientos que debe cumplir la aplicación. La definición del problema, el objeto de automatización y la propuesta del sistema ofrecieron una visión general del problema y de lo que será el producto, constituyendo la base fundamental para la realización del análisis y el diseño.

CAPITULO 3 “Análisis y Diseño”

3.1. Introducción

Con la definición y descripción de las principales características de la aplicación expuestas en el capítulo anterior, el propósito de esta sección es transformar los requerimientos en un diseño de lo que será el sistema. El presente capítulo tiene como objetivo la realización del modelo de análisis con las clases que lo componen, el diagrama de clases del diseño de cada caso de uso, los diagramas de secuencia del diseño, así como, el modelo lógico y físico de datos.

3.2. Modelo de análisis

3.2.1. Diagrama de clases del análisis

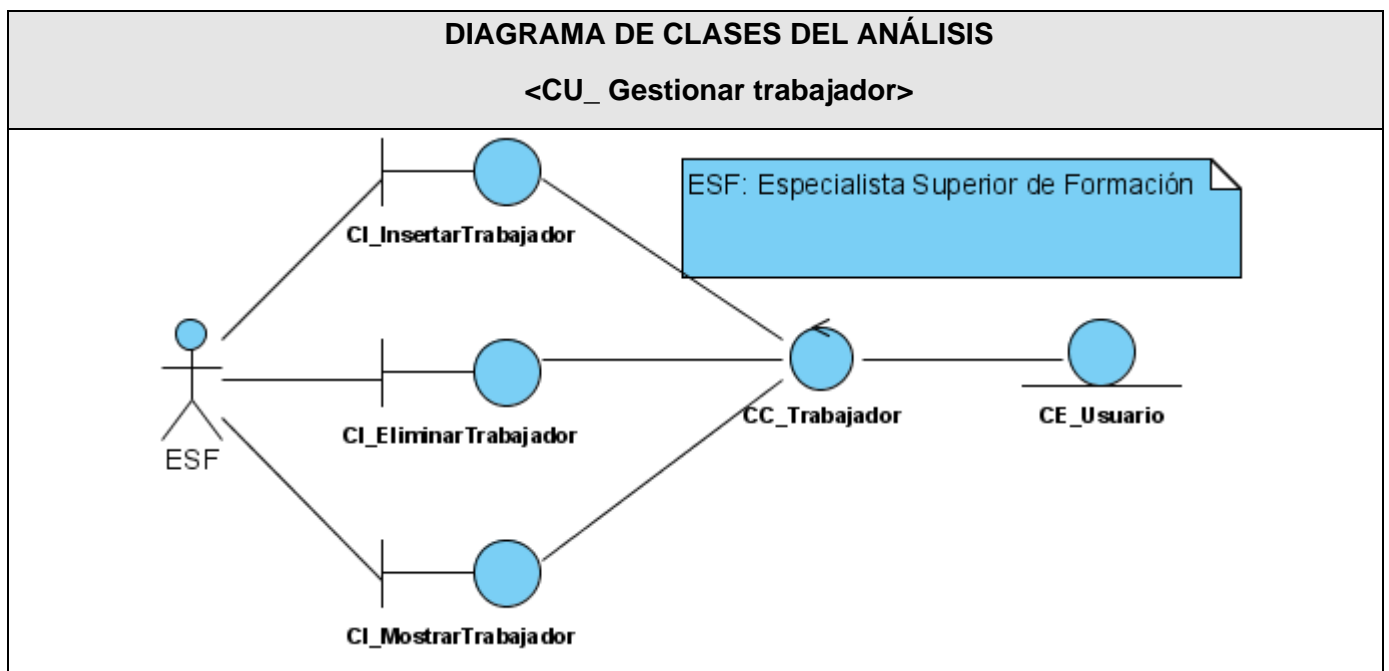


Figura 5 Diagrama de clases del análisis: Gestionar trabajador

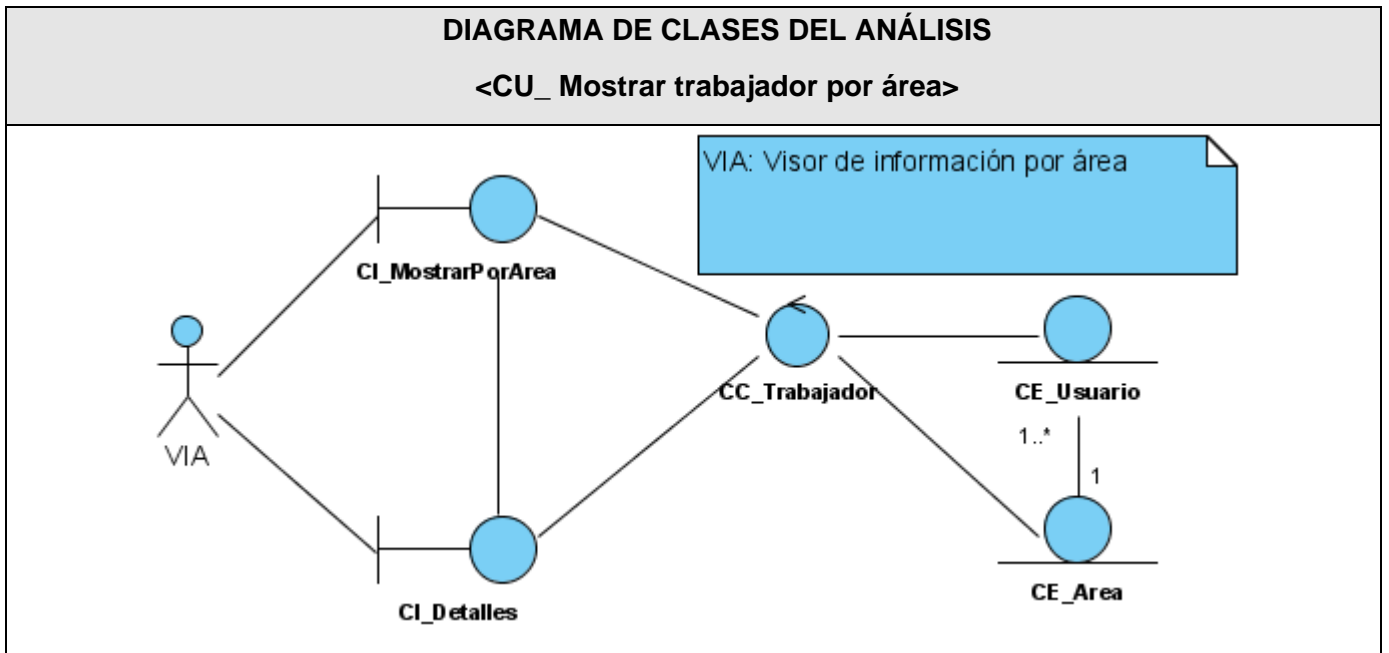


Figura 6 Diagrama de clases del análisis: Mostrar trabajador por área

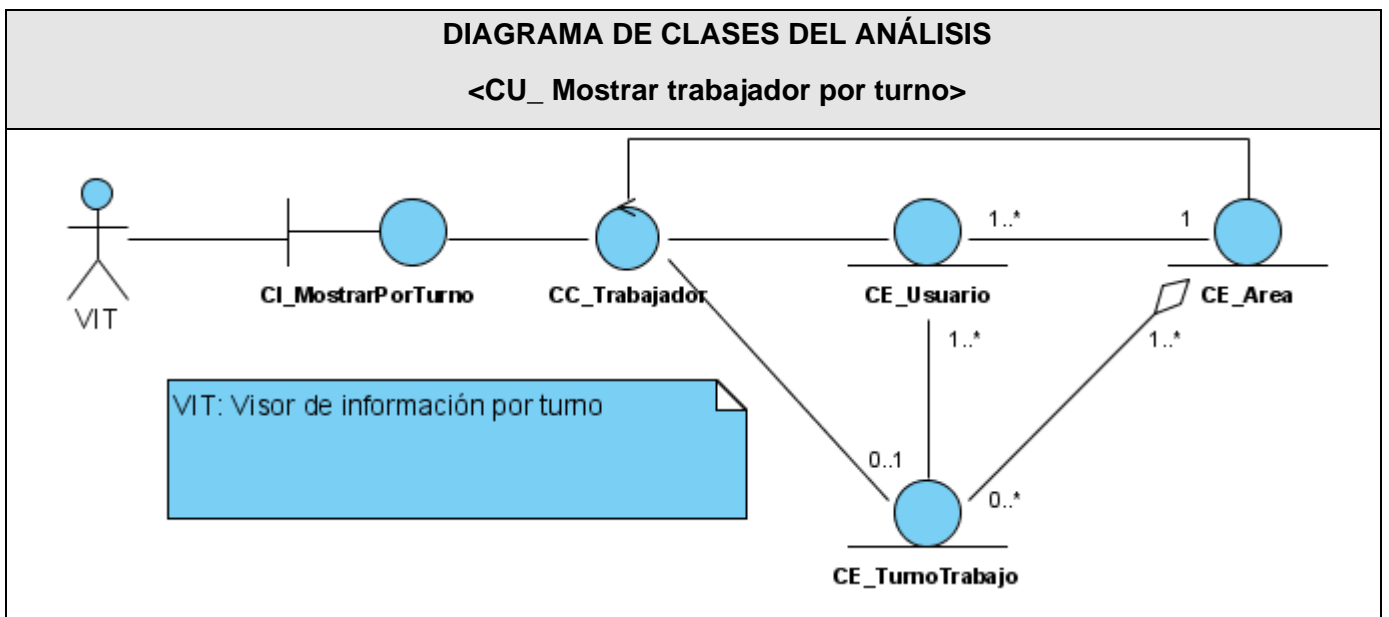


Figura 7 Diagrama de clases del análisis: Mostrar trabajador por turno

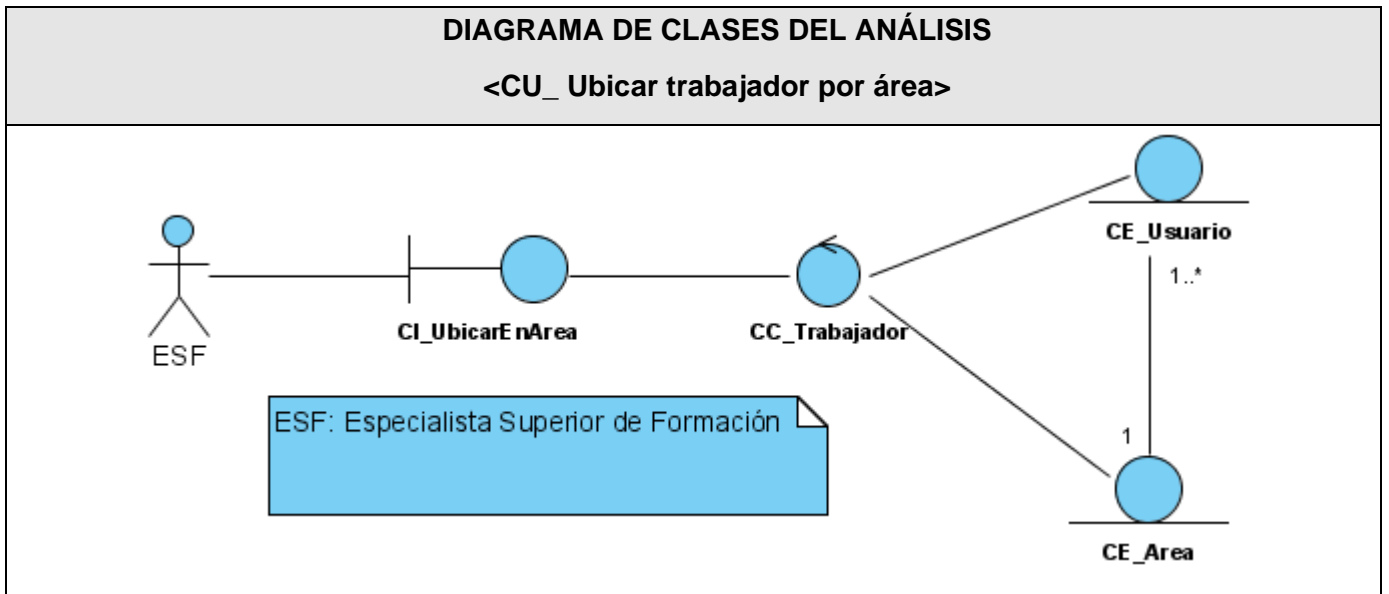


Figura 8 Diagrama de clases del análisis: Ubicar trabajador por área

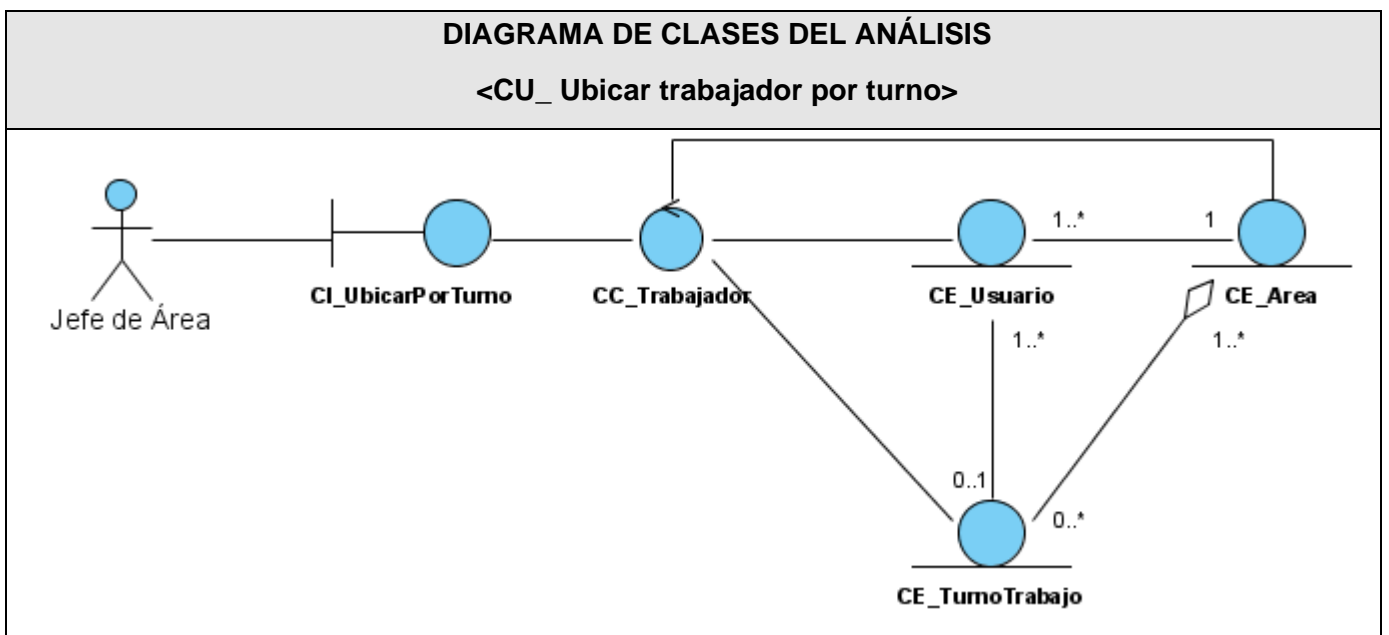


Figura 9 Diagrama de clases del análisis: Ubicar trabajador por turno

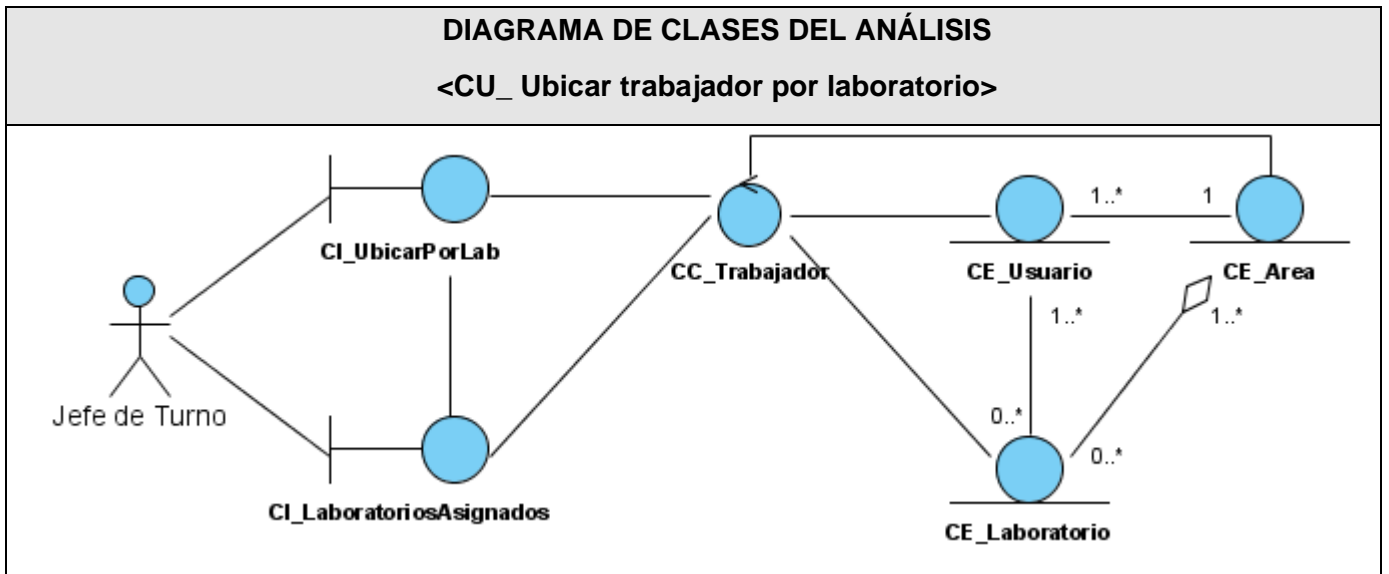


Figura 10 Diagrama de clases del análisis: Ubicar trabajador por laboratorio

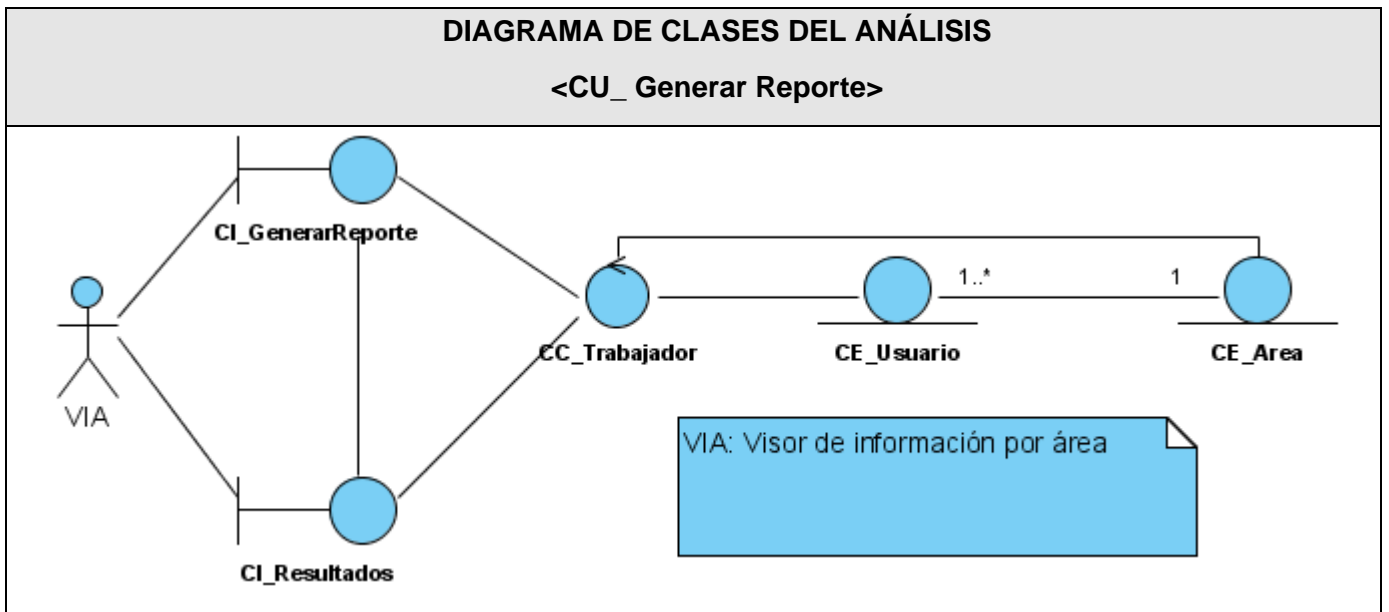


Figura 11 Diagrama de clases del análisis: Generar Reporte

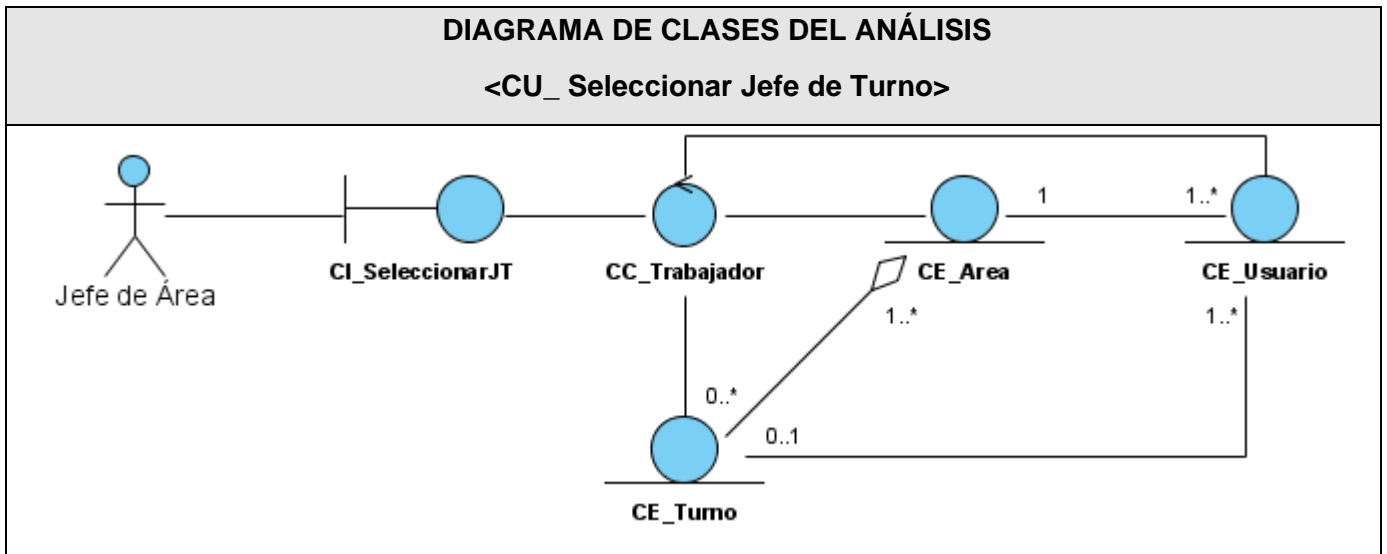


Figura 12 Diagrama de clases del análisis: Seleccionar Jefe de Turno

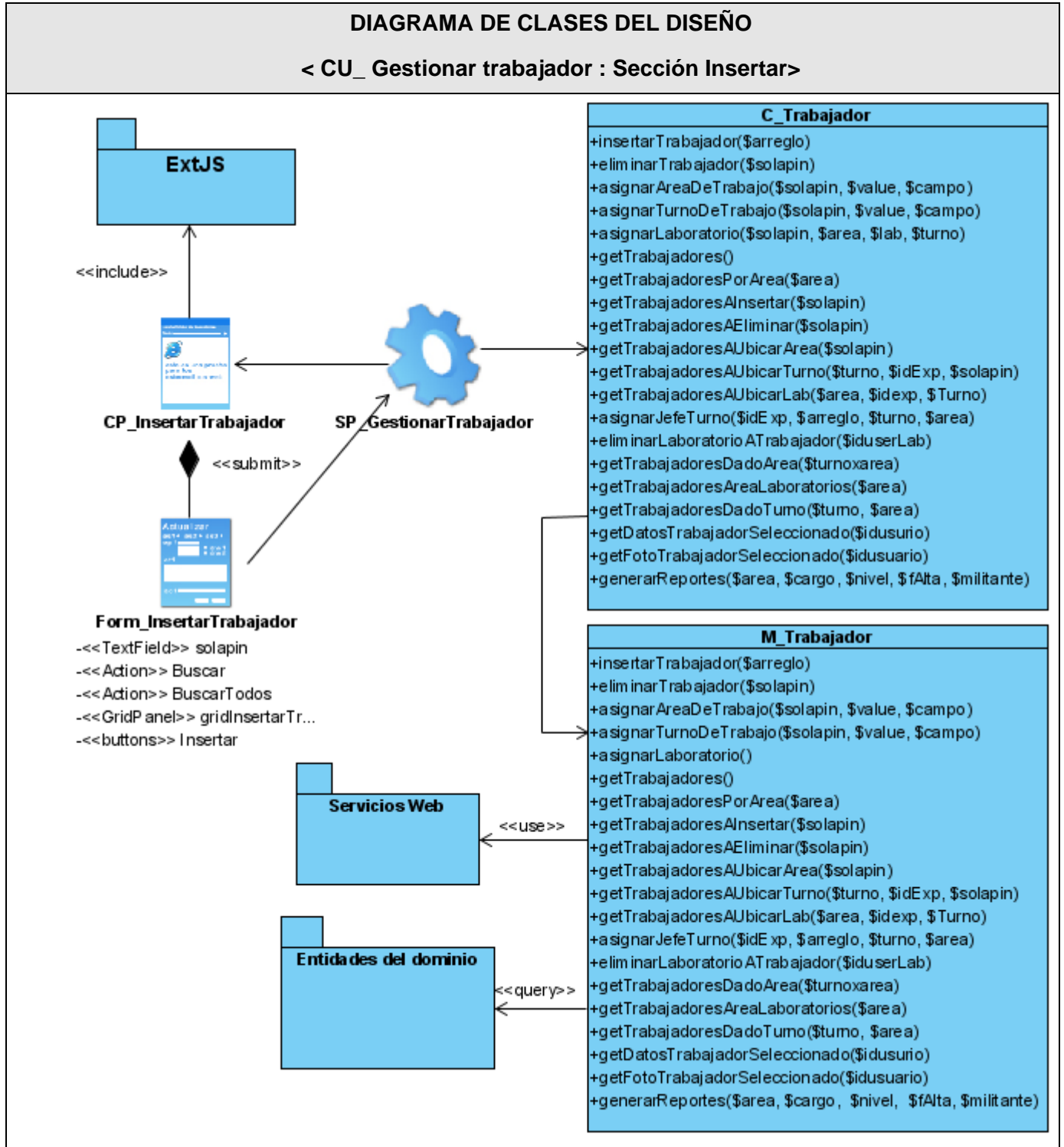
3.3. Modelo de diseño

En los últimos años las tecnologías Web han evolucionado considerablemente, sin embargo los estereotipos existentes para el modelado resultan insuficientes para representar estos grandes cambios que han surgido; por esta razón, se consideró explicar de manera general el funcionamiento con el FrameWork utilizado (ExtJS) y de esta forma eliminar cualquier duda que pueda surgir al respecto. A continuación se representan los aspectos más importantes a destacar:

- El FrameWork (ExtJS) se representa con un paquete que contiene los componentes JavaScript encargados de construir a los formularios.
- Los formularios se encargan de enviar la información de forma asíncrona al servidor, las clases controladoras capturan el evento solicitado por el usuario y utilizando las restantes clases obtiene los datos deseados de los servicios Web y/o la Base de Datos de SIGLA.
- El paquete nombrado *Entidades del dominio* contiene todas las entidades que se muestran en la Figura 23 en el diagrama de clases persistentes. En la Tabla 2 se muestran las relaciones específicas existentes entre las clases modelo y las entidades por cada caso de uso.

Nombre del caso de uso	Figura	Clase Modelo	Entidades que se relacionan
Gestionar Trabajador_Insertar	13	M_Trabajador	E_Usuario
Gestionar Trabajador_Eliminar	14	M_Trabajador	E_Usuario
Gestionar Trabajador_Mostrar	15	M_Trabajador	E_Usuario, E_Area
Mostrar trabajador por área	16	M_Trabajador	E_Usuario, E_Area
Mostrar trabajador por turno	17	M_Trabajador	E_Usuario, E_Area, E_Turno
Ubicar trabajador por área	18	M_Trabajador	E_Usuario, E_Area
		M_Area	E_Area
Ubicar trabajador por turno	19	M_Trabajador	E_Usuario, E_Area, E_Turno
		M_Turno	E_Turno
Ubicar trabajador por laboratorio	20	M_Trabajador	E_Usuario, E_Area, E_Turno, E_Laboratorio
		M_Laboratorio	E_Laboratorio
Generar Reporte	21	M_Trabajador	E_Usuario, E_Area
		M_Area	E_Area
Seleccionar Jefe de Turno	22	M_Trabajador	E_Usuario, E_Area, E_Turno, E_Rol
		M_Turno	E_Turno

Tabla 3 Relaciones entre clases modelos y entidades

3.3.1. Diagrama de clases del diseño

Figura 13 Diagrama de clases del diseño: Gestionar trabajador: Sección Insertar

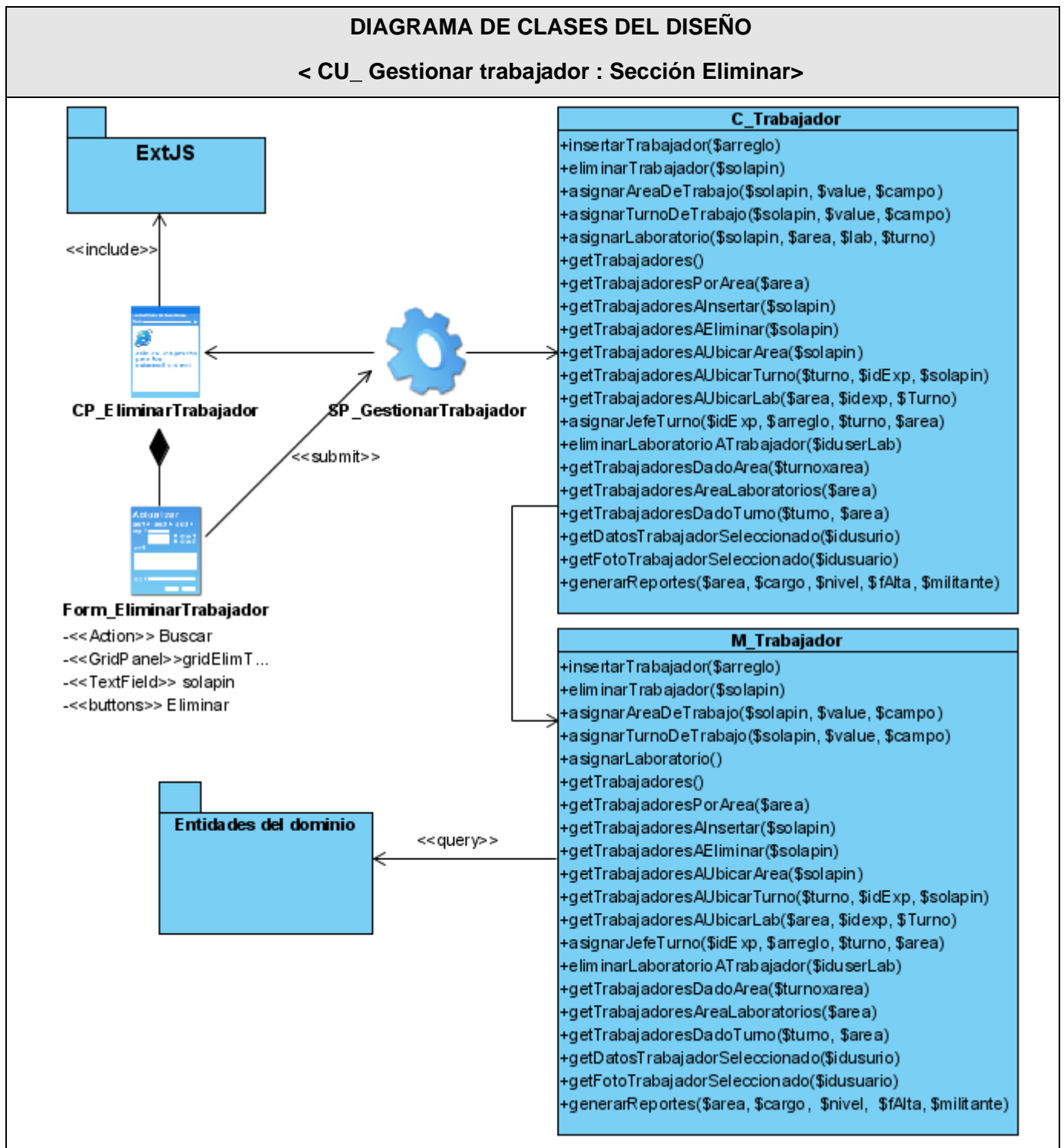


Figura 14 Diagrama de clases del diseño: Gestionar trabajador: Sección Eliminar

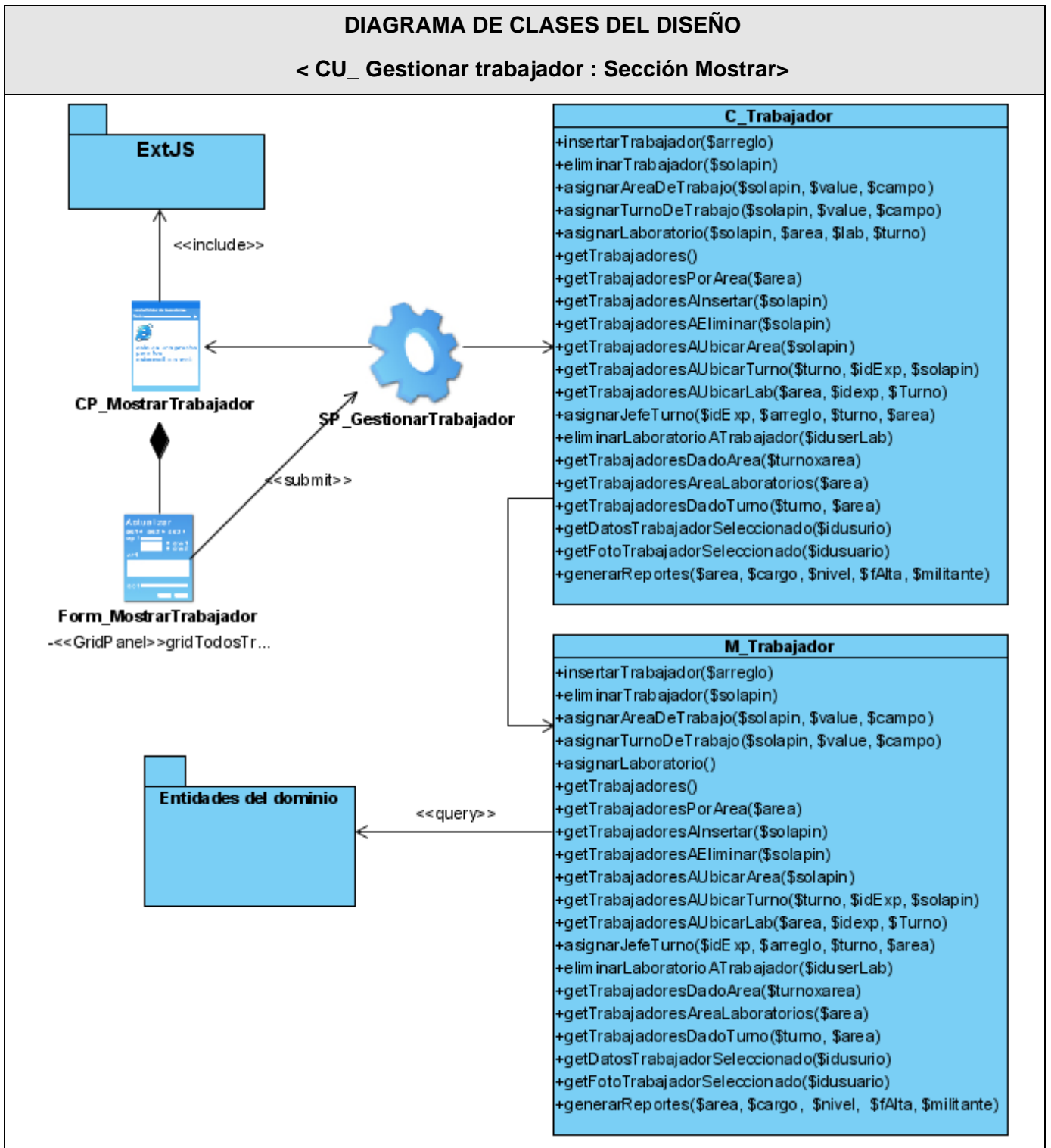


Figura 15 Diagrama de clases del diseño: Gestionar trabajador: Sección Mostrar

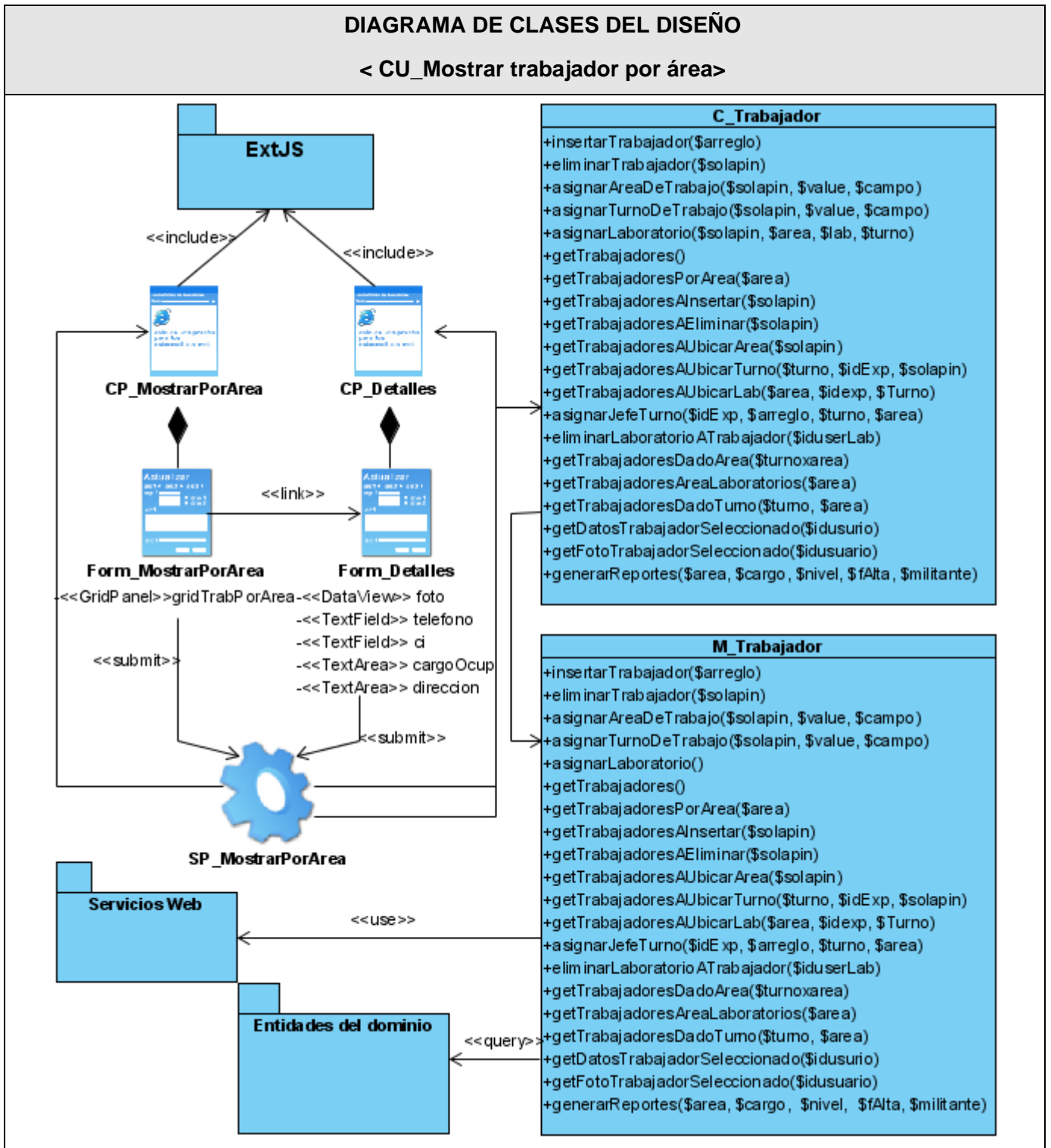


Figura 16 Diagrama de clases del diseño: Mostrar trabajador por área

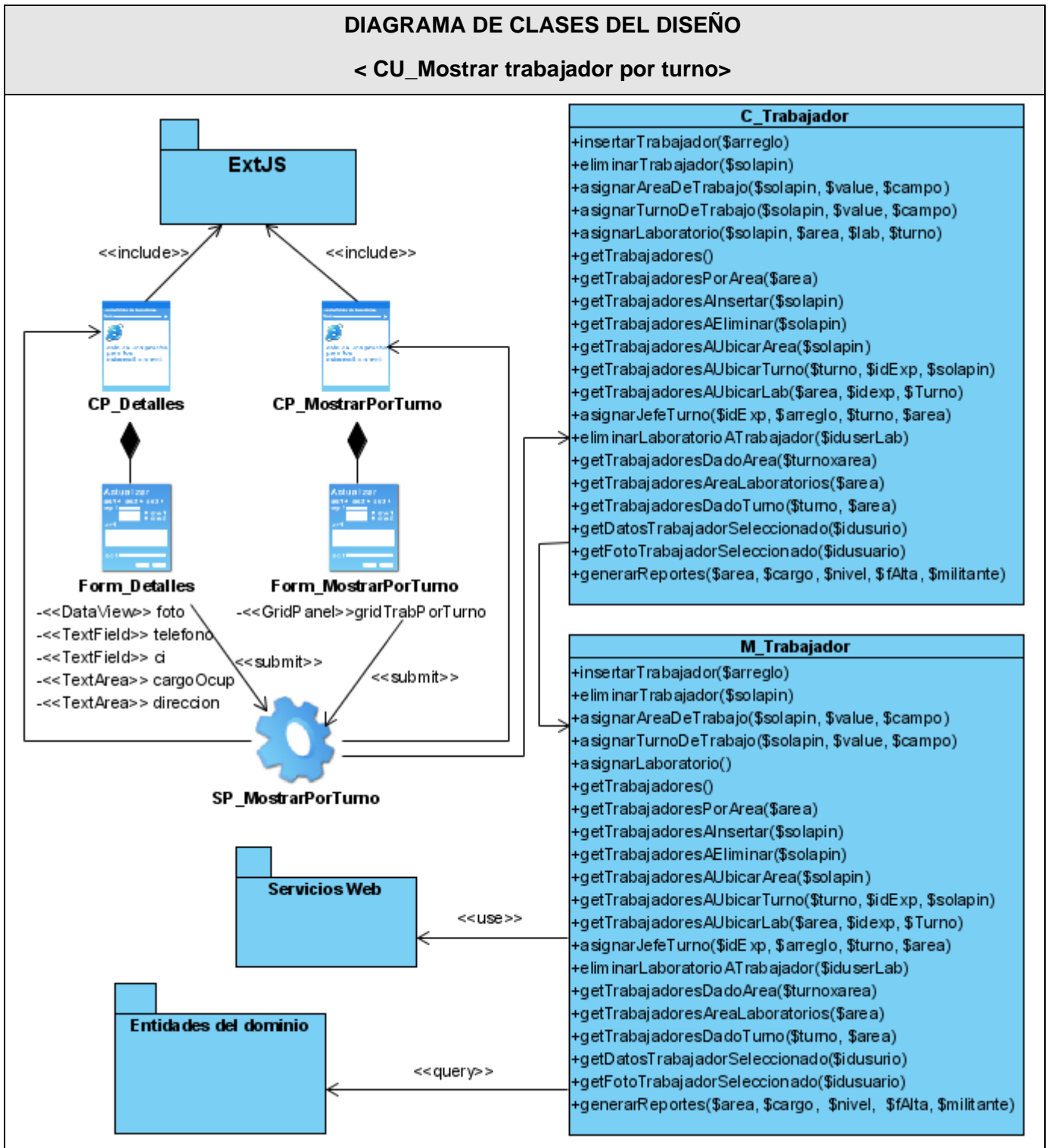


Figura 17 Diagrama de clases del diseño: Mostrar trabajador por turno

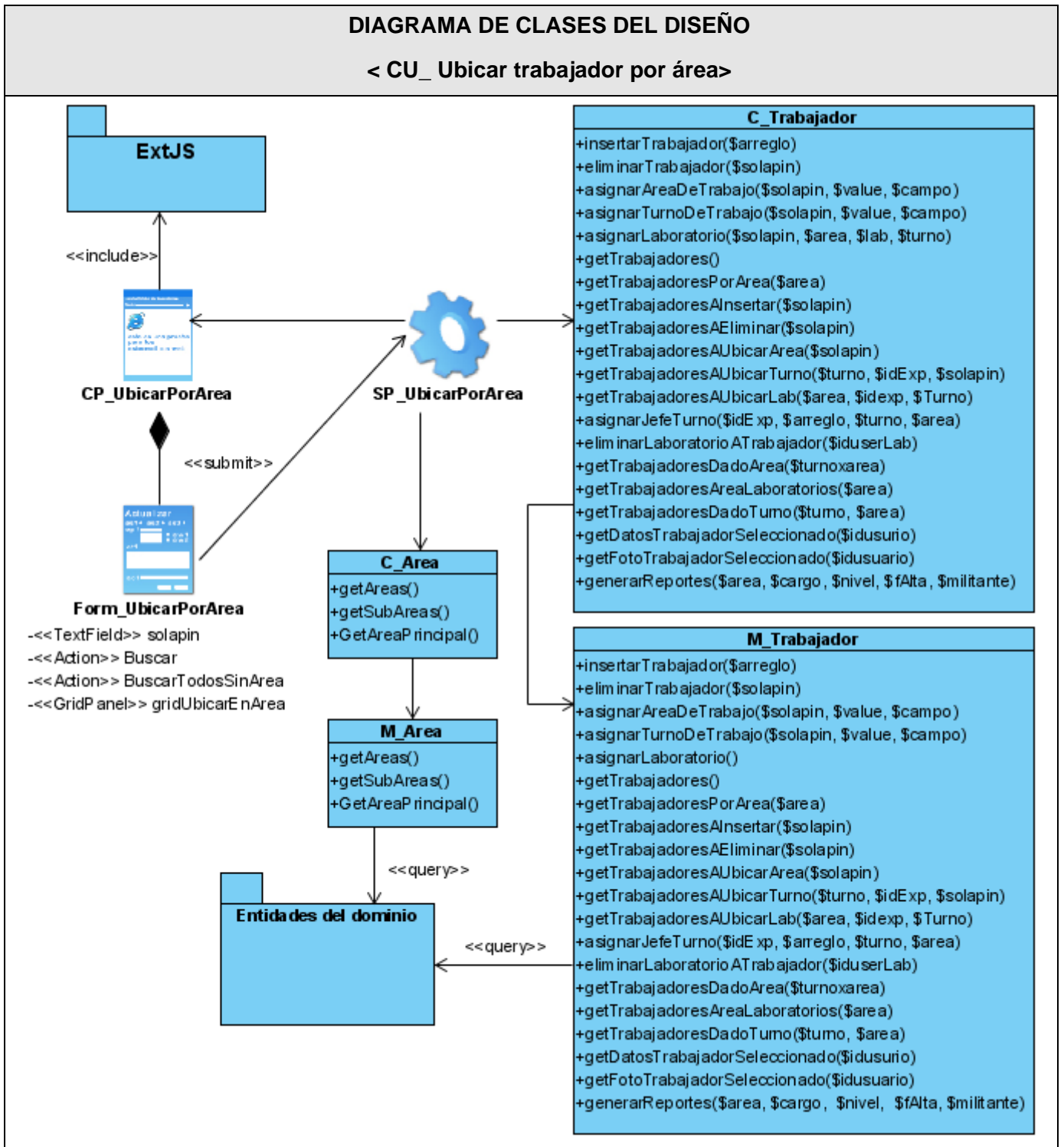


Figura 18 Diagrama de clases del diseño: Ubicar trabajador por área

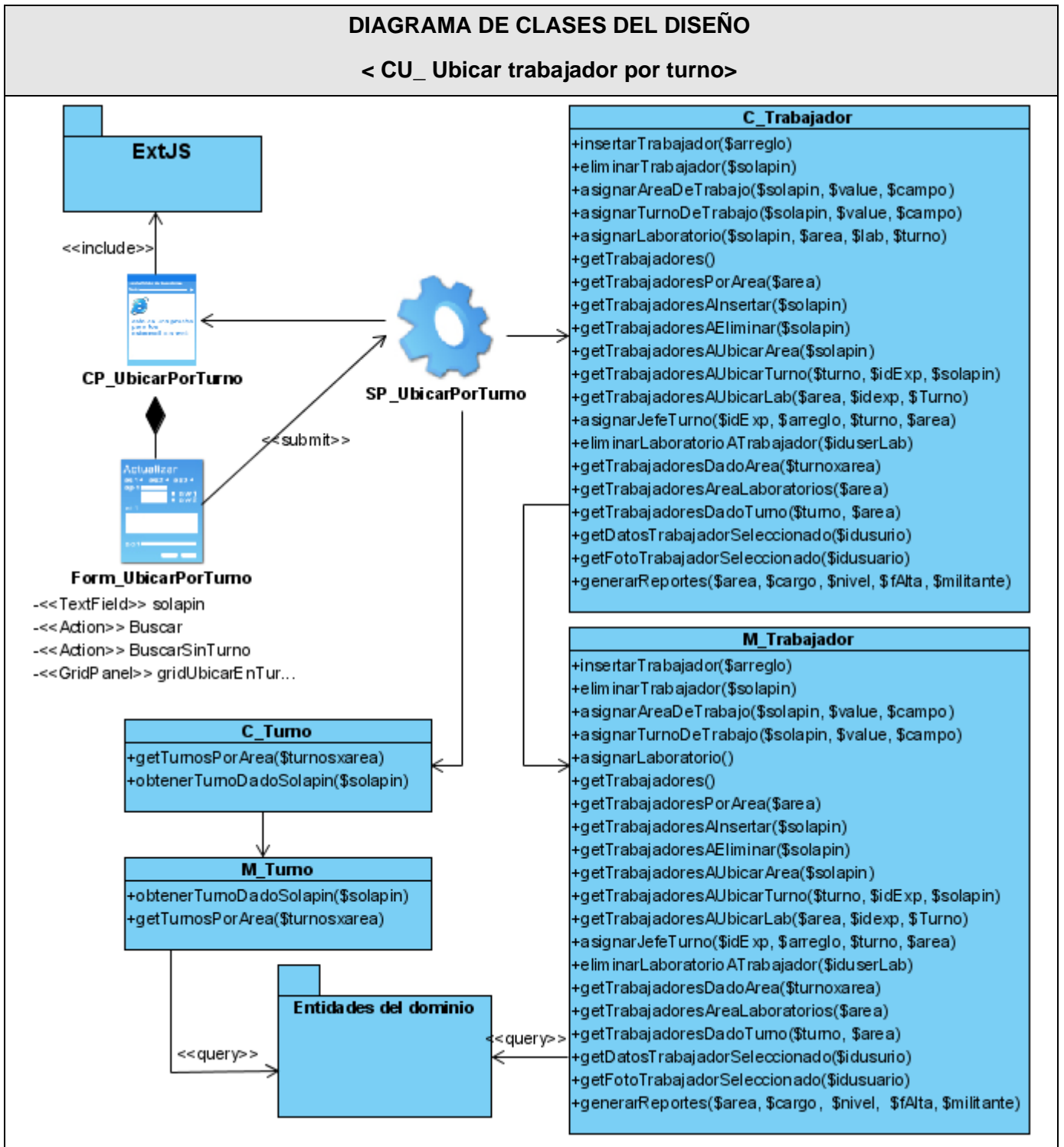


Figura 19 Diagrama de clases del diseño: Ubicar trabajador por turno

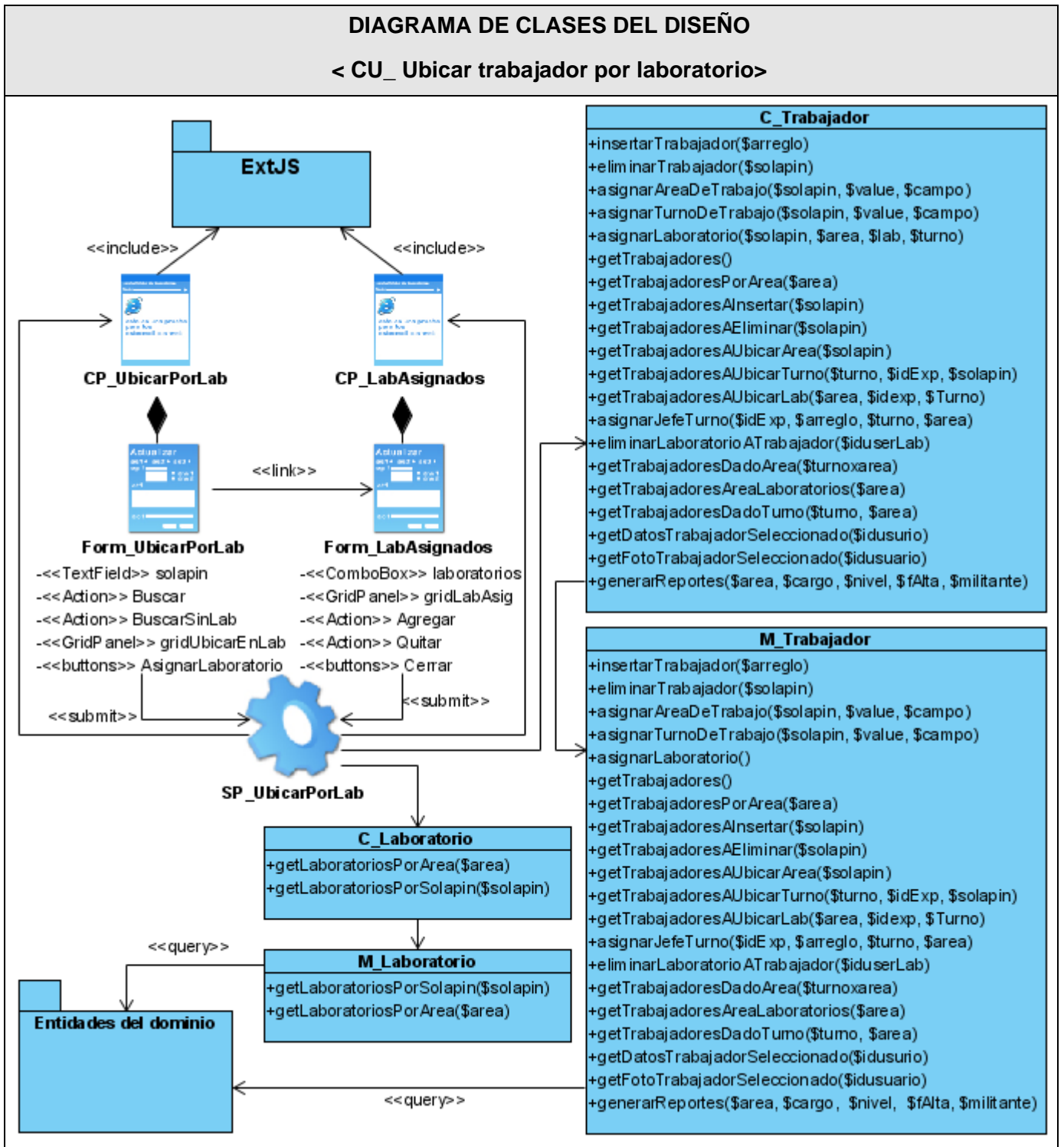


Figura 20 Diagrama de clases del diseño: Ubicar trabajador por laboratorio

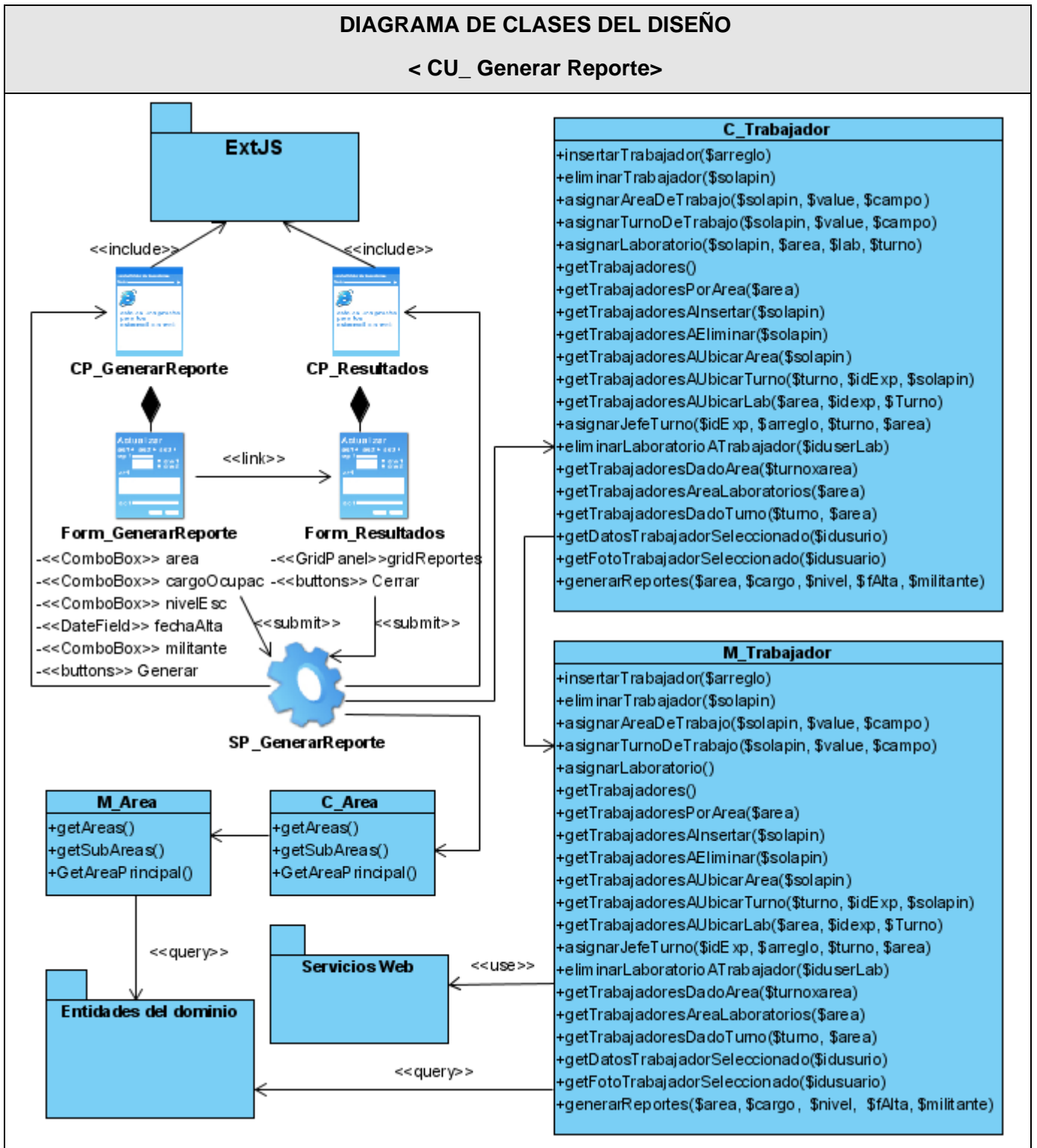


Figura 21 Diagrama de clases del diseño: Generar Reporte

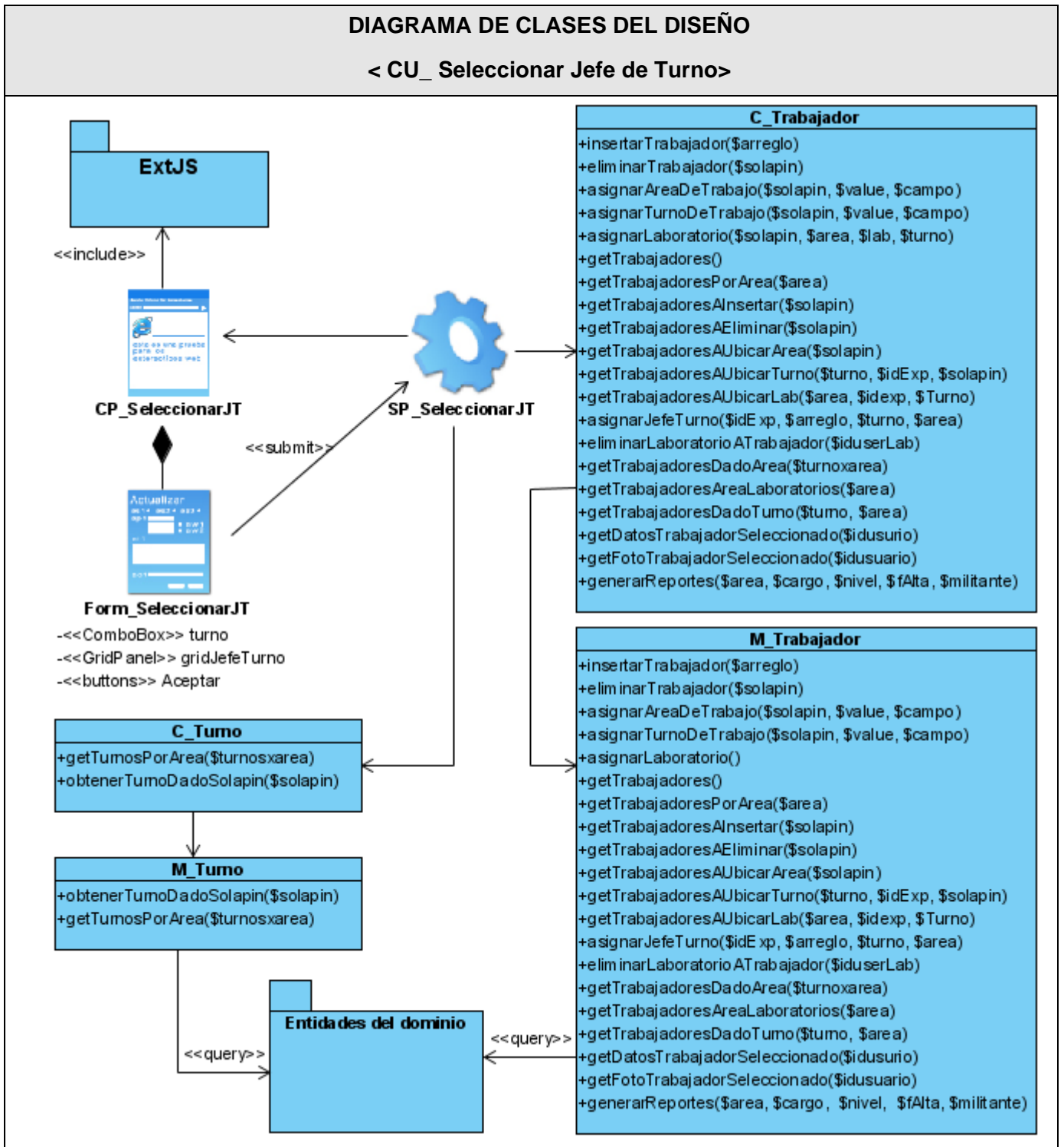


Figura 22 Diagrama de clases del diseño: Seleccionar Jefe de Turno

3.4. Diagramas de interacción

En el Anexo_3 se muestran los diagramas de secuencia del diseño para los casos de uso críticos que son: Gestionar trabajador, Ubicar trabajador por área, Ubicar trabajador por turno, Ubicar trabajador por laboratorio y Seleccionar Jefe de Turno.

3.5. Diseño de la Base de Datos

3.5.1. Modelo lógico de datos

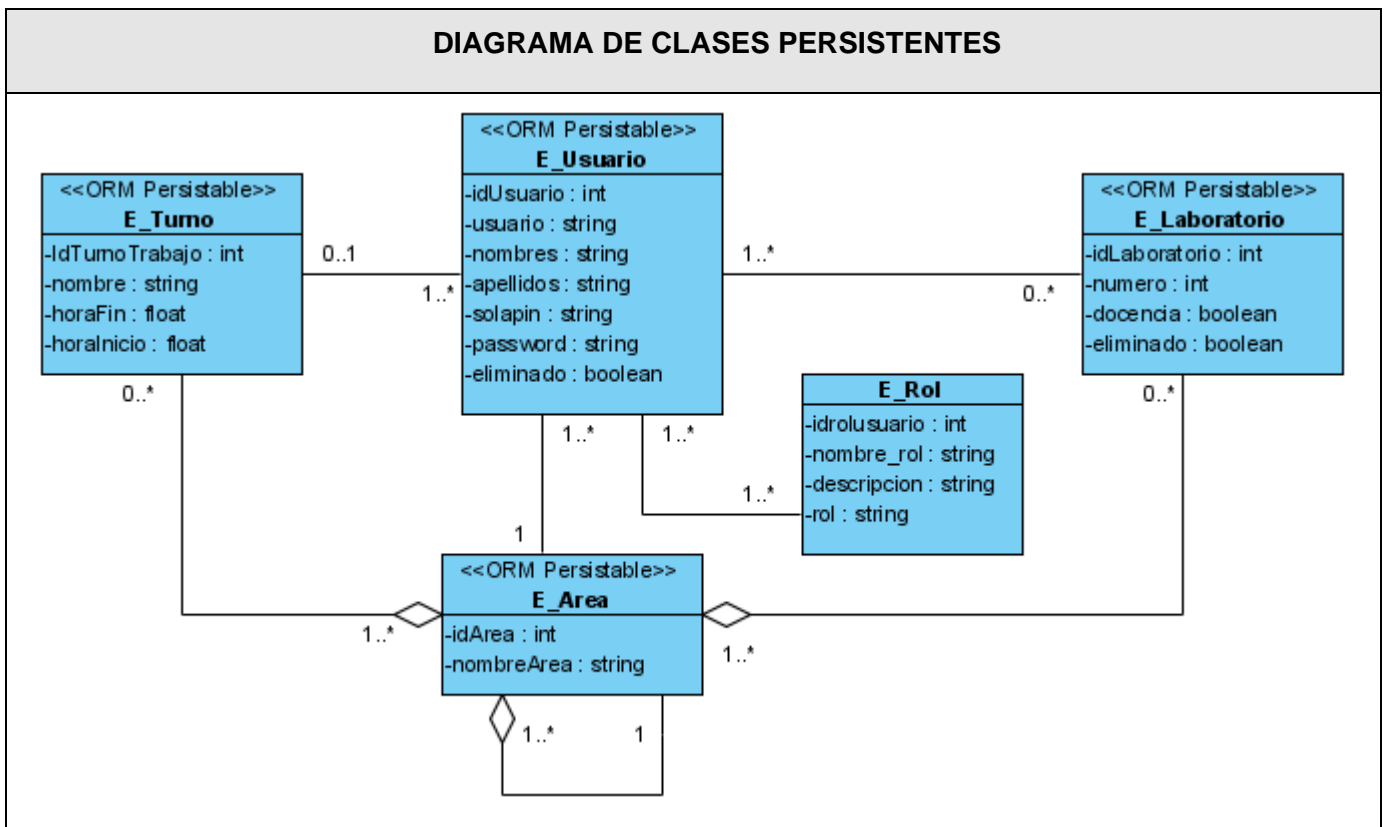
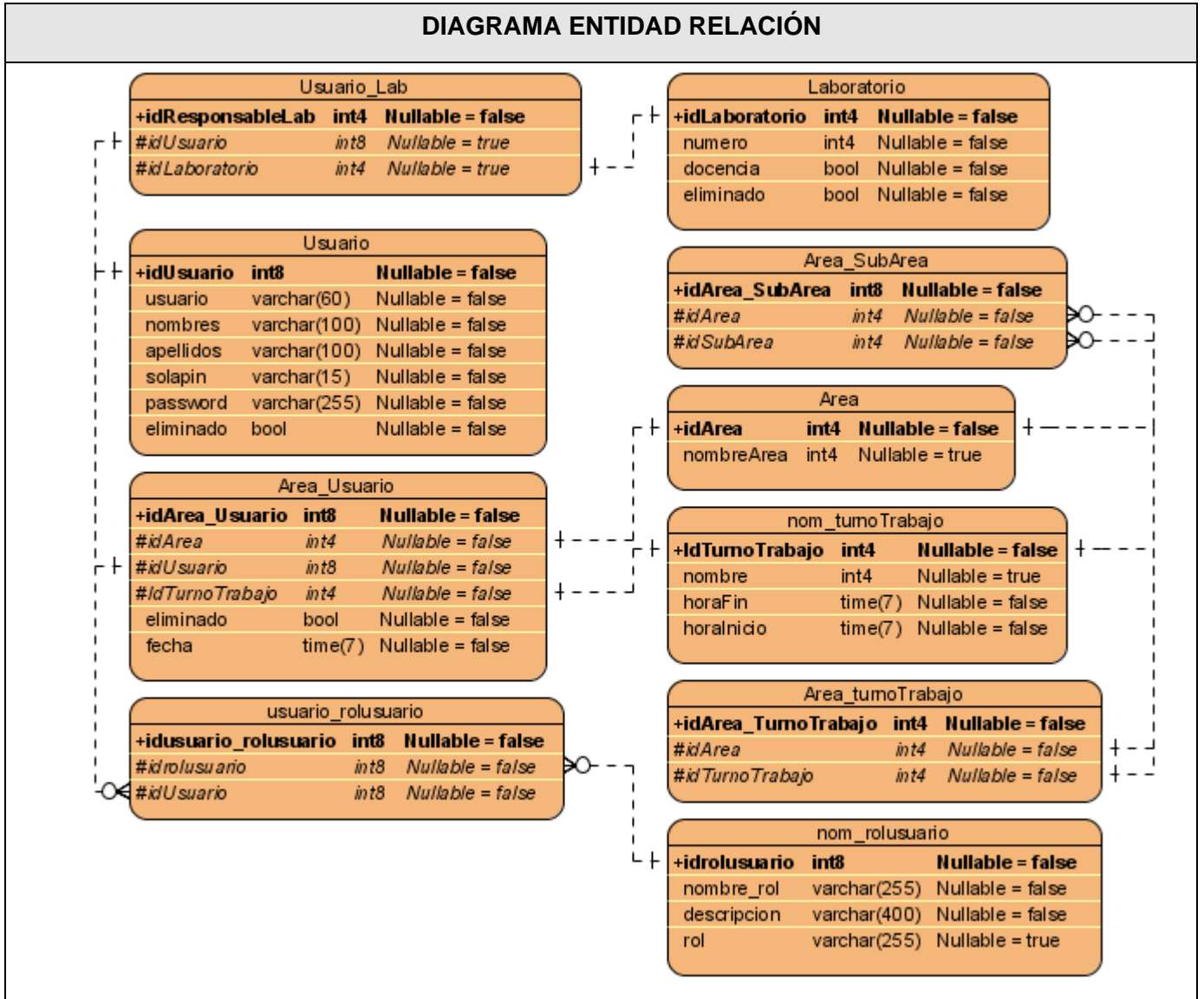


Figura 23 Diagrama de clases persistentes

3.5.2. Modelo físico de datos

Figura 24 Diagrama Entidad Relación

3.6. Conclusiones

En este capítulo se presentaron los artefactos fundamentales generados en el flujo de trabajo de Análisis y Diseño, que son: el modelo de análisis, el modelo de diseño, el modelo lógico y físico de datos. Con el desarrollo del Análisis se obtuvo una visión del sistema y se utilizó como base para la elaboración del resultado final más importante de este flujo de trabajo que es el modelo de diseño, ya que constituye la entrada principal para la implementación.

CAPITULO 4 “Implementación”

4.1. Introducción

El presente capítulo tiene como objetivo desarrollar los artefactos correspondientes a la implementación del sistema, comenzando con el resultado más importante del capítulo anterior: el modelo de diseño. A partir de este artefacto se realizan los diagramas de componentes y despliegue que conforman lo que se conoce como: Modelo de implementación; de esta forma se describen cómo los elementos del modelo del diseño se implementan en términos de componentes y se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

4.2. Diagrama de despliegue

Los diagramas de despliegue describen la arquitectura física del sistema durante la ejecución, en términos de: procesadores, dispositivos y componentes de software. Básicamente este tipo de diagrama se utiliza para modelar el hardware utilizado en la implementación del sistema y las relaciones entre sus componentes.

En la Figura 25 se representan los aspectos más importantes a destacar que son:

- La PC Cliente constituye el puesto donde el usuario estará interactuando con la aplicación, este nodo tendrá una conexión con una impresora mediante <<USB>> para imprimir los reportes que se necesiten y mediante <<HTTP>> con servidor Web (Apache) encargado de responder a todas las peticiones hechas por el Cliente.
- El servidor Web interactúa con un servidor de Base de datos (PostgreSQL) y con los servicios Web mediante <<PDO>> y <<SOAP>> respectivamente, para obtener y brindar los datos solicitados por el usuario.

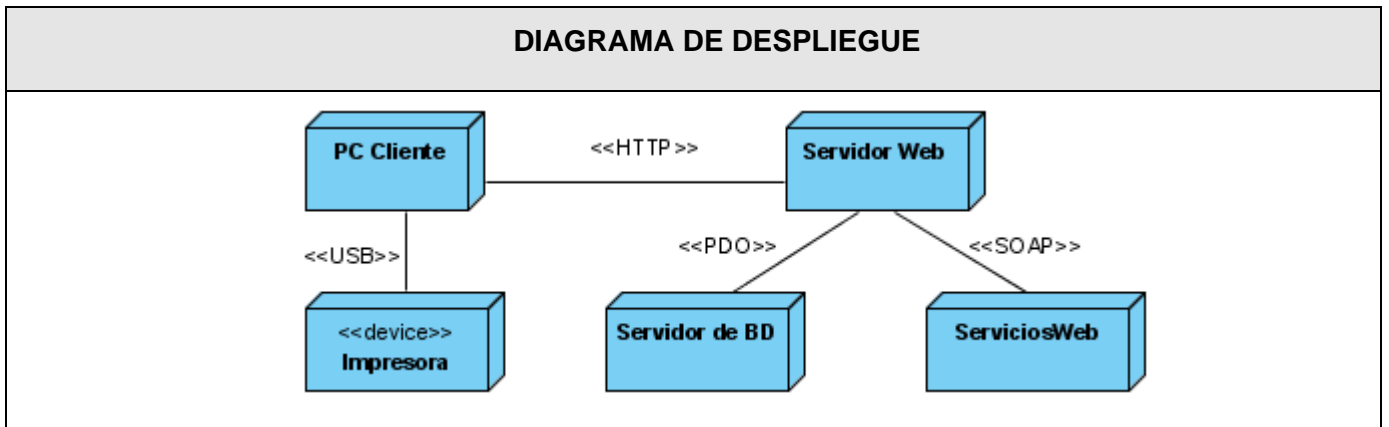


Figura 25 Diagrama de despliegue

4.3. Diagrama de componentes

Los diagramas de componentes muestran la separación de un sistema de software en componentes físicos y las dependencias entre ellos.

- El paquete de los Servicios Web contiene dos servicios: `assetsService` e `Identificación`; el primero brinda los datos esenciales de los trabajadores y el segundo devuelve un conjunto de datos que identifican a una persona en la UCI.
- Del servicio **assetsService** se emplean las funcionalidades: `ObtenerPersonaDadoldExpediente (idExp)` y `ObtenerPersonasDadoldArea (idArea)`, de los cuales es posible obtener los siguientes datos: nombre, apellidos, área, sexo, carnet de identidad, solapín y cargo ocupacional.
- Del servicio **Identificación** se emplean las funcionalidades: `ObtenerPersonaDadoSolapin (solapin)` y `ObtenerUrlRepositorioFoto ()`, de los cuales es posible obtener la foto y el usuario.
- El paquete de la ExtJS contiene tres ficheros: `ext_base.js`, `ext_all.js` y `ext_all.css`, los dos primeros constituyen los ficheros mediante los cuales se pueden crear los componentes visuales y el último proporciona un estilo común a todos los componentes utilizados.

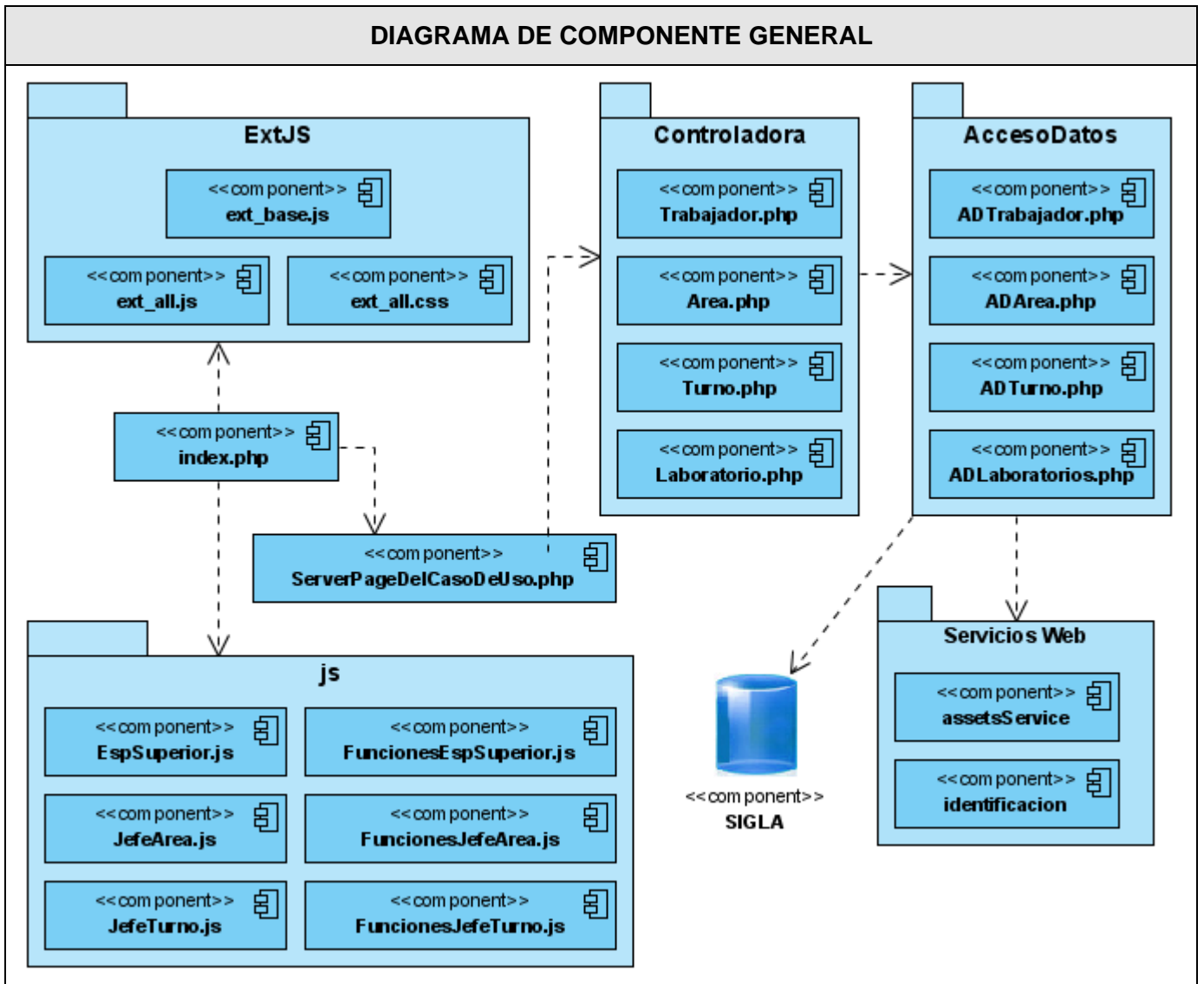


Figura 26 Diagrama de componente general

A continuación se muestran los diagramas de componentes correspondientes a cada uno de los casos de uso del Módulo “Control de Personal”.

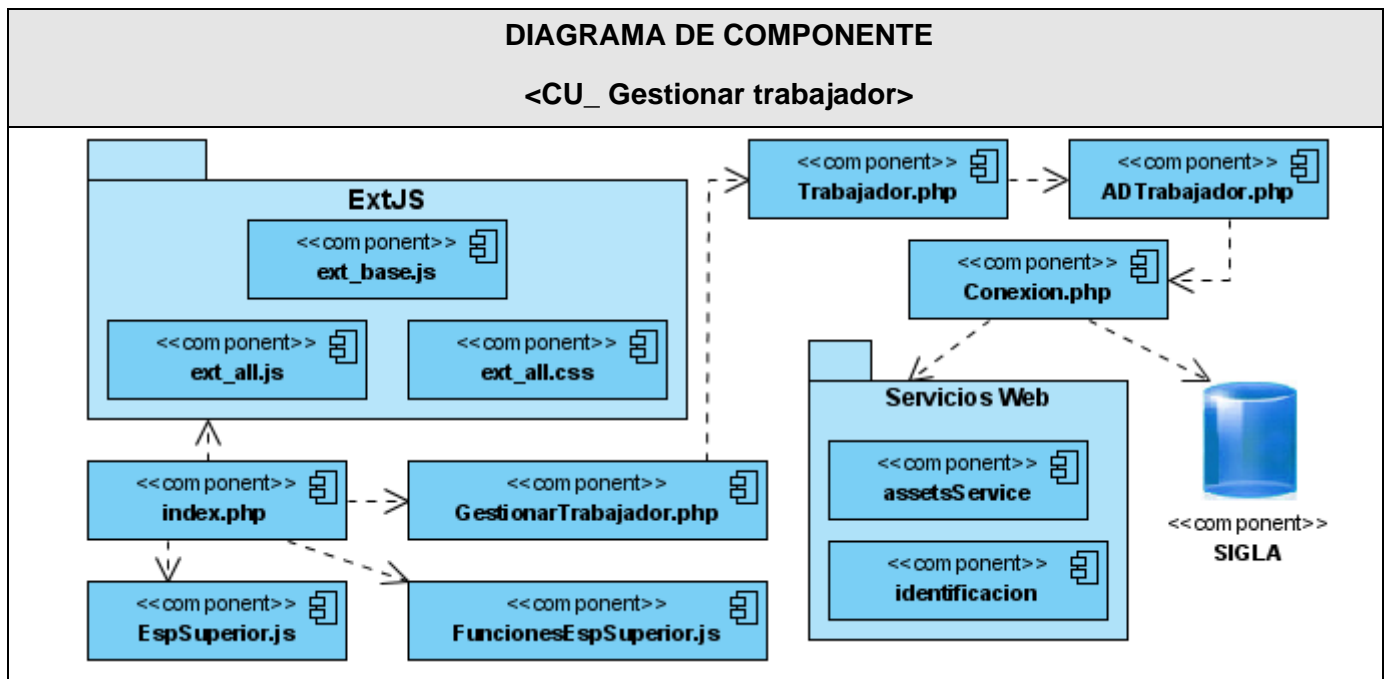


Figura 27 Diagrama de componente: Gestionar trabajador

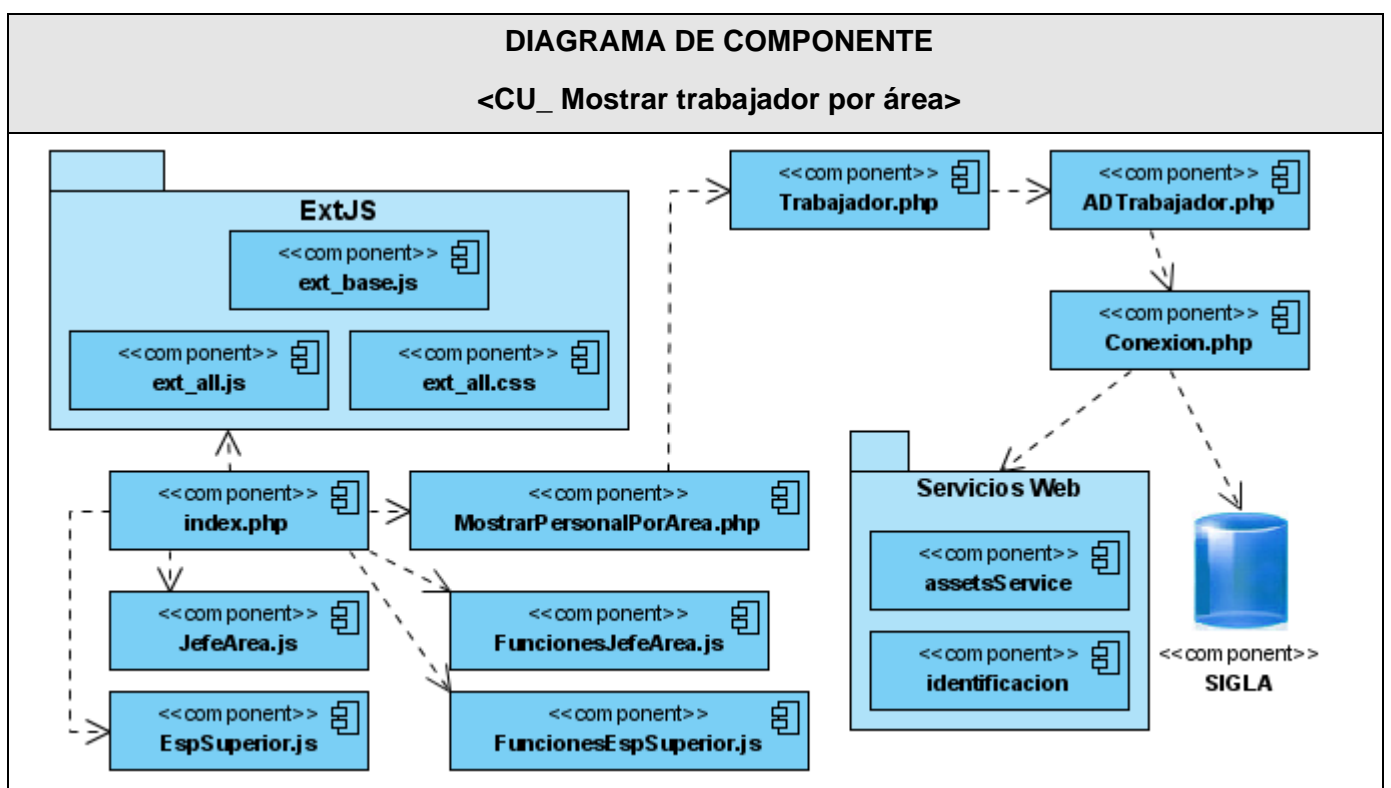


Figura 28 Diagrama de componente: Mostrar trabajador por área

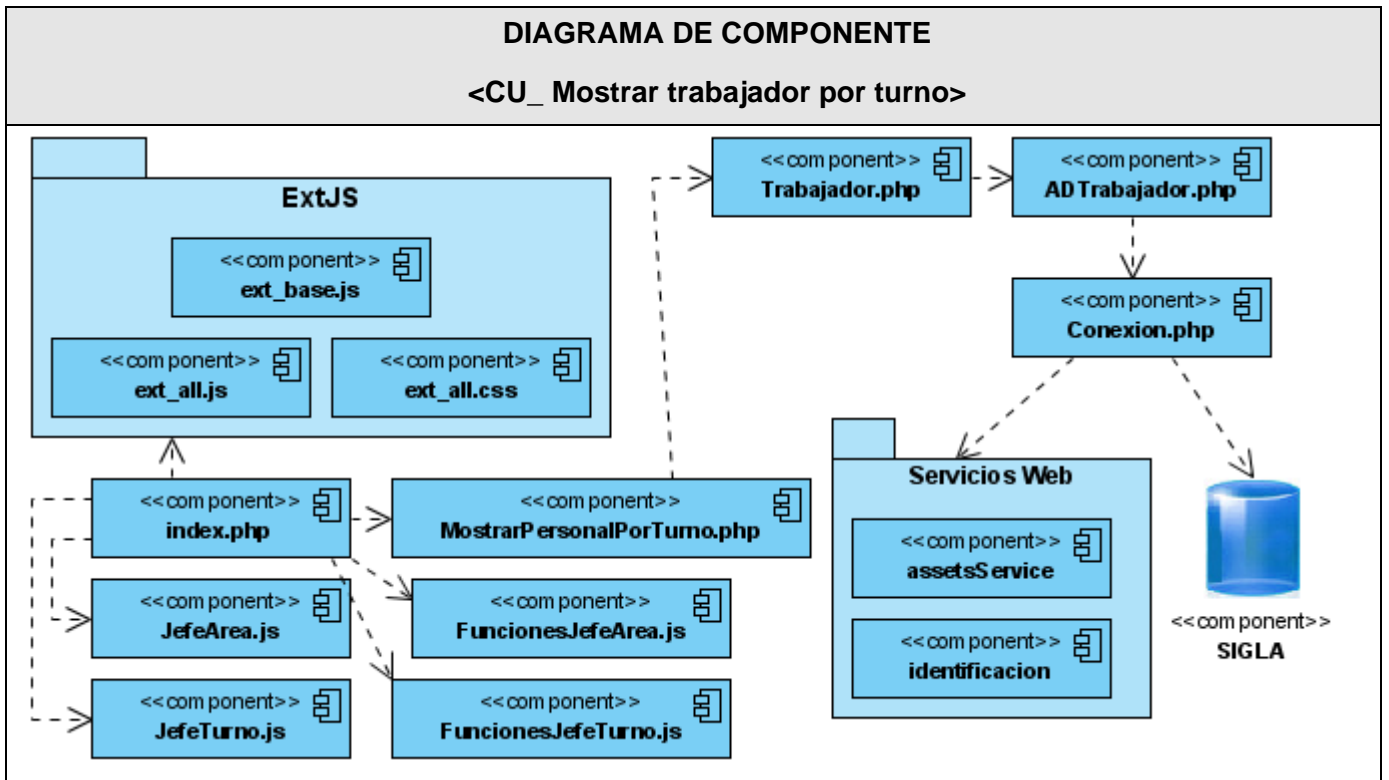


Figura 29 Diagrama de componente: Mostrar trabajador por turno

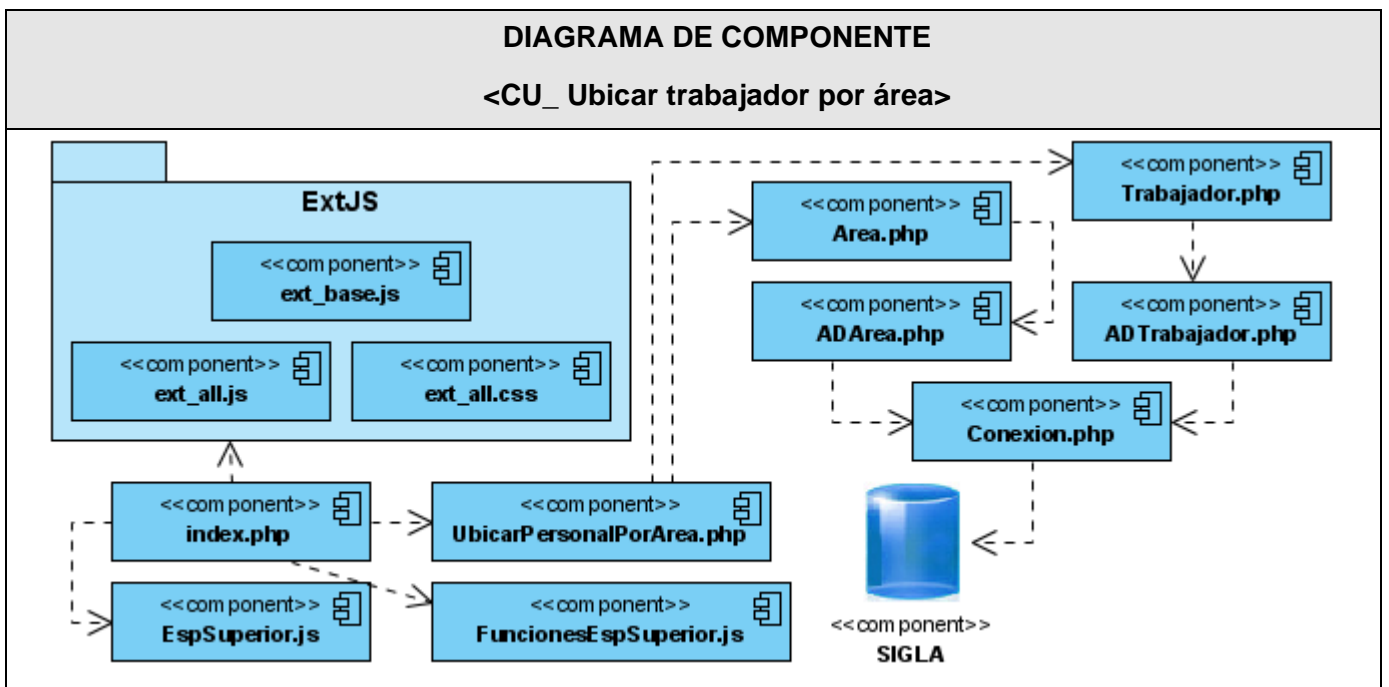


Figura 30 Diagrama de componente: Ubicar trabajador por área

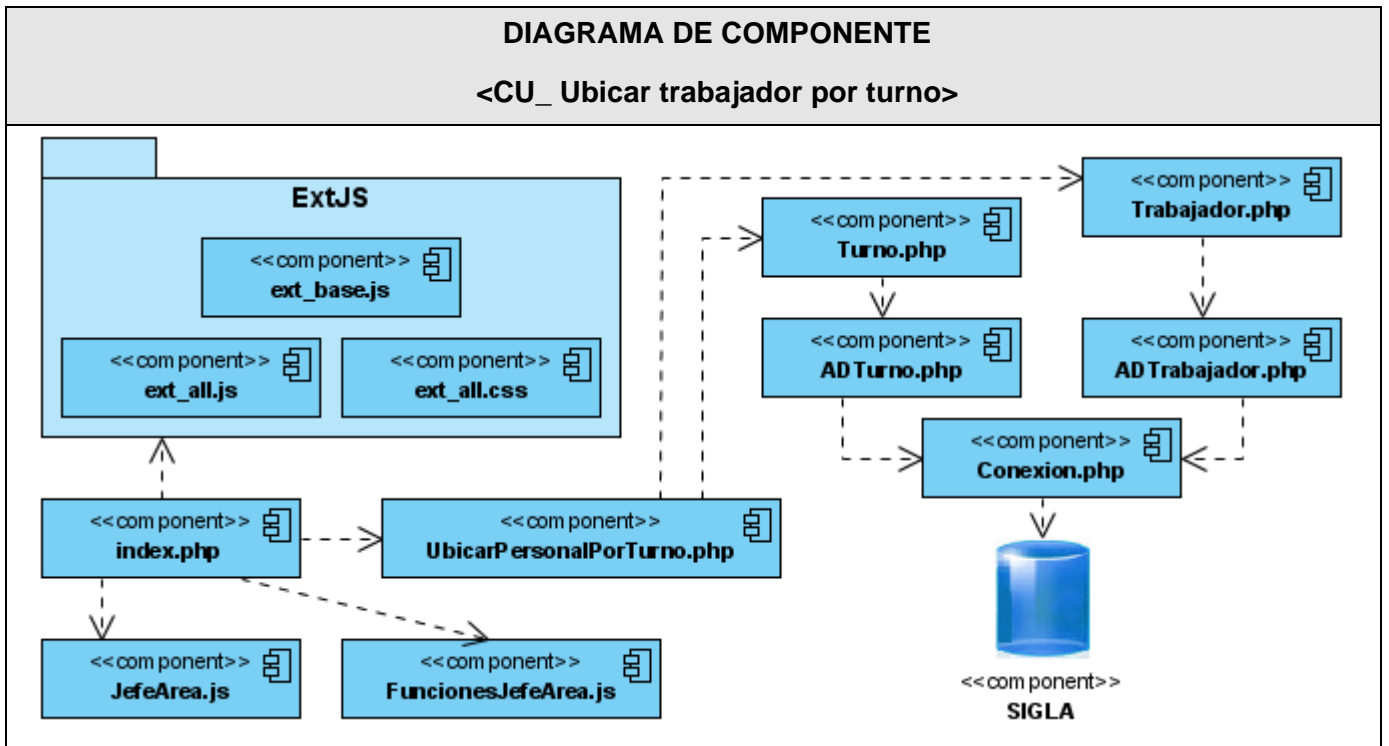


Figura 31 Diagrama de componente: Ubicar trabajador por turno

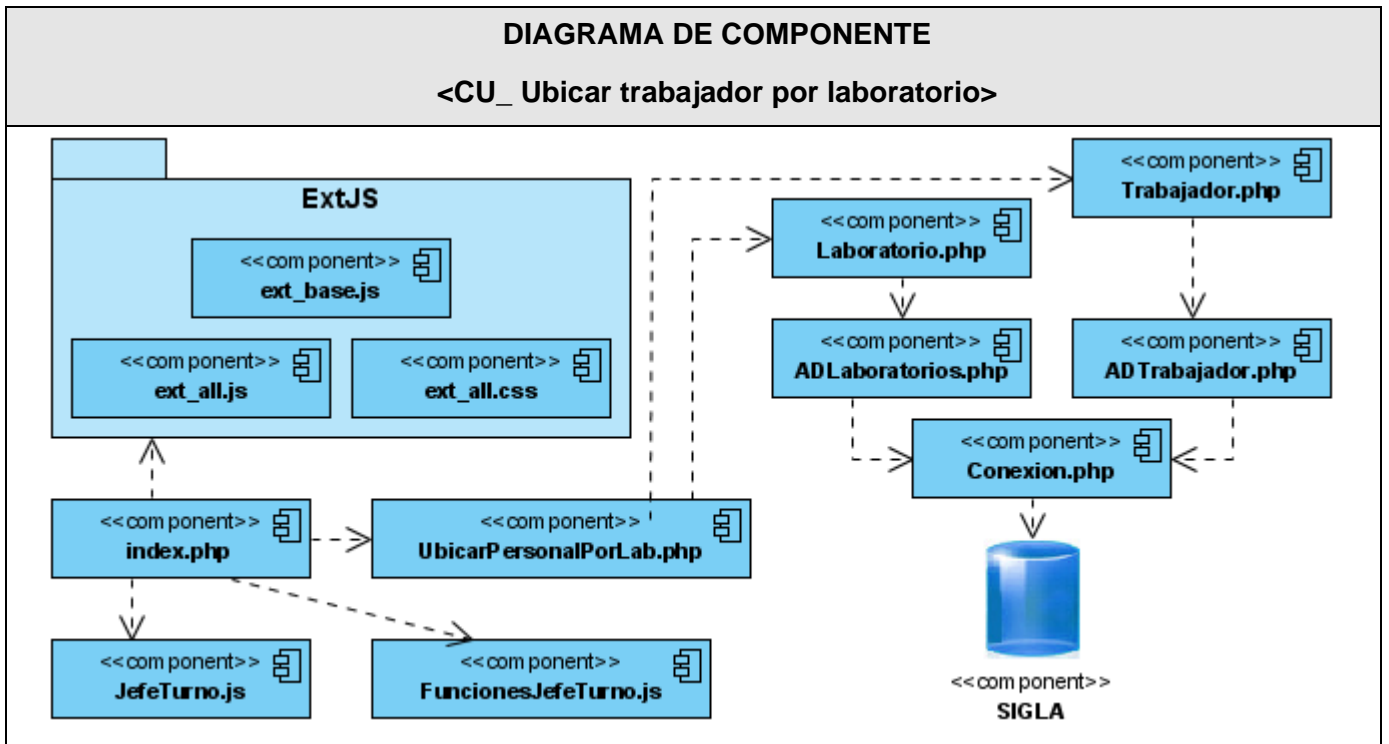


Figura 32 Diagrama de componente: Ubicar trabajador por laboratorio

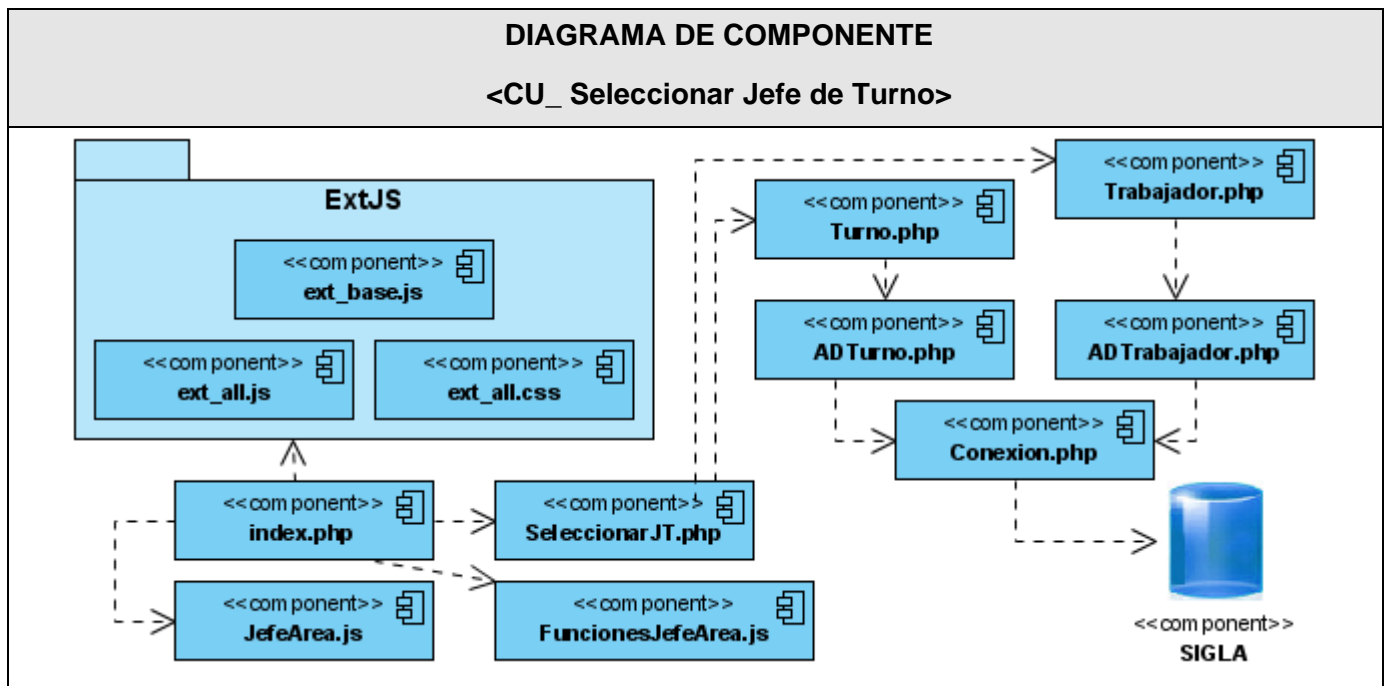


Figura 33 Diagrama de componente: Seleccionar Jefe de Turno

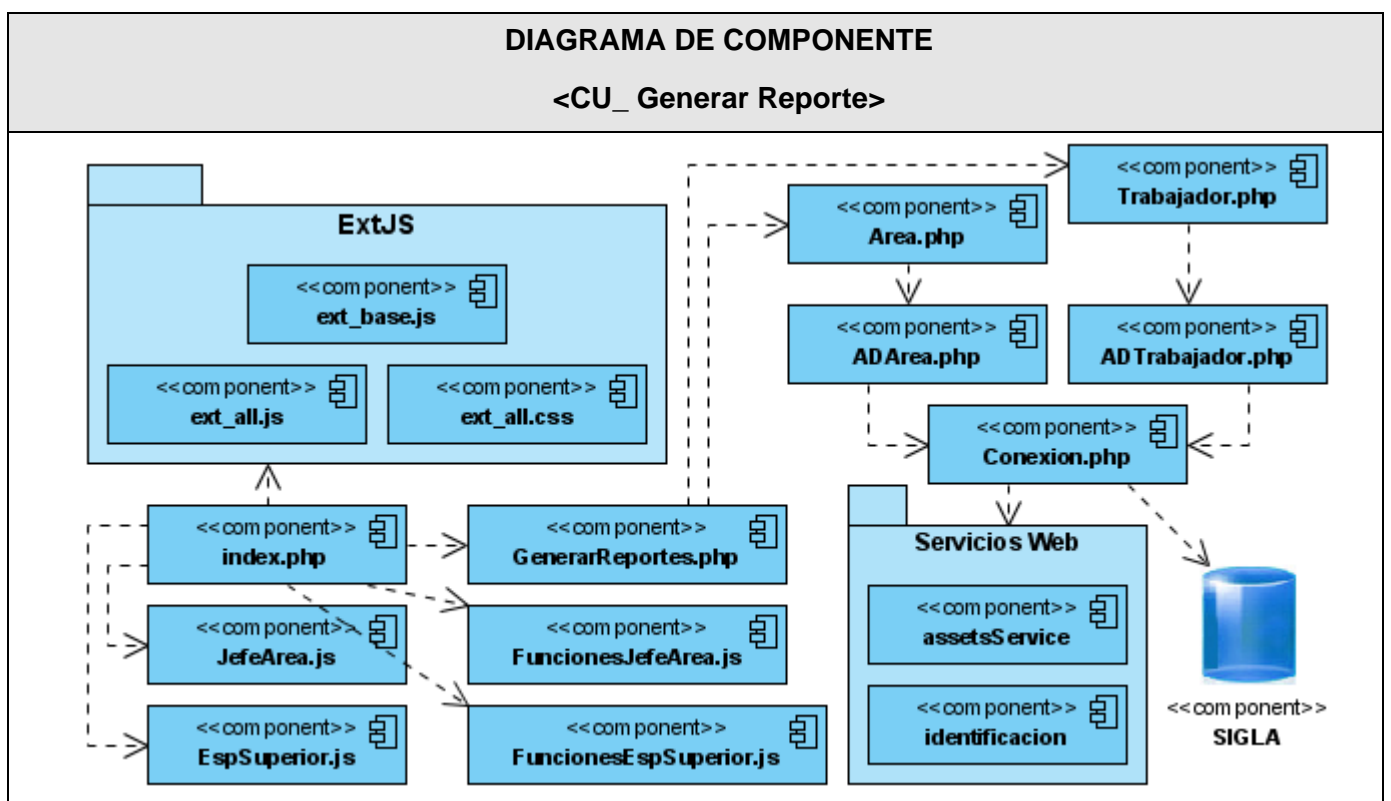


Figura 34 Diagrama de componente: Generar Reporte

4.4. Conclusiones

En este capítulo se desarrollaron los elementos correspondientes al modelo de implementación: el diagrama de despliegue y los de componentes por cada caso de uso. Al concluir se completaron los artefactos fundamentales que constituyen la base de la aplicación del módulo “Control de Personal”, dando cumplimiento de esta forma, a los requerimientos y necesidades del cliente.

CAPITULO 5 “Estudio de la Factibilidad”

5.1. Introducción

Los estudios de factibilidad son investigaciones altamente enfocadas en un proyecto, diseñados para producir información crítica sobre la viabilidad de implementar el sistema de software, y con base en ello decidir si se va a llevar a cabo o no. En este capítulo se determina la estimación y cálculo de tamaño del software, así como el esfuerzo, los costos y beneficios que reportará el sistema realizado para gestionar al personal de los laboratorios.

5.2. Planificación

5.2.1. Puntos de Casos de Uso

Es un método de estimación y cálculo de tamaño del software basado en cuentas hechas sobre los casos de uso para un sistema de software. El método requiere de casos de uso en modo textual y gráfico, por lo que es necesario tener dominio del problema. A continuación se describen los pasos a seguir.

5.2.1.1. Cálculo de Puntos de Casos de Uso sin ajustar

El cálculo de los Puntos de Caso de Uso sin ajustar constituye el primer paso y se calcula a partir de la siguiente ecuación:

UUCP = UAW + UUCW	Donde, UUCP: Puntos de Casos de Uso sin ajustar. UAW: Factor de Peso de los Actores sin ajustar. UUCW: Factor de Peso de los Casos de Uso sin ajustar.
--------------------------	---

Tabla 4 Cálculo de Puntos de Casos de Uso sin ajustar

Para obtener el Factor de Peso de los Actores sin ajustar se debe tener en cuenta la cantidad de actores presentes en el sistema y la complejidad de los mismos. Los criterios se muestran en la siguiente tabla:

Tipo de Actor	Descripción	Peso	Cantidad * Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface).	1	0 * 1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	1 * 2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	3 * 3
Total			11

Tabla 5 Factor de peso de los actores sin ajustar

Para obtener el valor del Factor de Peso de los Casos de Uso sin ajustar se debe tener en cuenta la cantidad de casos de uso y la complejidad de cada uno de ellos. Los criterios se muestran en la siguiente tabla:

Tipo de Caso de Uso	Descripción	Peso	Cantidad * Peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones.	5	3 * 5
Medio	El Caso de Uso contiene de 4 a 7 transacciones.	10	4 * 10
Complejo	El Caso de Uso contiene más de 8 transacciones.	15	1 * 15
Total			70

Tabla 6 Factor de peso de los Casos de Uso sin ajustar

Finalmente los Puntos de Caso de Uso sin ajustar resultan:

$$\mathbf{UUCP} = \mathbf{UAW} + \mathbf{UUCW} = 11 + 70 = 81$$

5.2.1.2. Cálculo de Puntos de Casos de Uso ajustados

Una vez obtenidos los Puntos de Caso de Uso sin ajustar se procede al cálculo de los Puntos de Casos de Uso ajustados mediante la siguiente ecuación:

UCP = UUCP * TCF * EF	<p>Donde,</p> <p>UCP: Puntos de Casos de Uso ajustados.</p> <p>UUCP: Puntos de Casos de Uso sin ajustar.</p> <p>TCF: Factor de complejidad técnica.</p> <p>EF: Factor de ambiente.</p>
------------------------------	--

Tabla 7 Cálculo de Puntos de Casos de Uso ajustados

Factor de complejidad técnica: Este coeficiente se calcula mediante un conjunto de Factores que determinan la complejidad del sistema, cada factor se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. La ecuación para su cálculo es:

$$\mathbf{TCF} = 0.6 + 0.01 * \Sigma (\mathbf{Peso\ } i * \mathbf{Valor\ Asignado\ } i)$$

Factor	Descripción	Peso	Peso * Valor
T1	Sistema distribuido.	2	2 * 3
T2	Objetivos de performance o tiempo de respuesta.	1	1 * 5
T3	Eficiencia del usuario final.	1	1 * 4
T4	Procesamiento interno complejo.	1	1 * 1
T5	El código debe ser reutilizable.	1	1 * 5
T6	Facilidad de instalación.	0.5	0.5 * 5
T7	Facilidad de uso.	0.5	0.5 * 3

T8	Portabilidad.	2	2 * 4
T9	Facilidad de cambio.	1	1 * 3
T10	Concurrencia.	1	1 * 3
T11	Incluye objetivos especiales de seguridad.	1	1 * 3
T12	Provee acceso directo a terceras partes.	1	1 * 4
T13	Se requieren facilidades especiales de entrenamiento a los usuarios.	1	1 * 2
Total			48

Tabla 8 Valores del Factor de complejidad técnica.

Entonces, **TCF** = $0.6 + 0.01 * 48 = 1.08$

Factor de ambiente: Contempla las habilidades y el entrenamiento del grupo de desarrollo por su importancia en las estimaciones de tiempo. Al igual que el factor de complejidad técnica se cuantifican con valores de 0 a 5. La ecuación para su cálculo es:

$$EF = 1.4 - 0.03 * \Sigma (\text{Peso } i * \text{Valor Asignado } i)$$

En la siguiente tabla se muestra el significado y el peso de cada uno de estos factores.

Factor	Descripción	Peso	Peso * Valor
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	1.5 * 1
E2	Experiencia en la aplicación.	0.5	0.5 * 1
E3	Experiencia en orientación a objetos.	1	1 * 4
E4	Capacidad del analista líder.	0.5	0.5 * 3
E5	Motivación.	1	1 * 5
E6	Estabilidad de los requerimientos.	2	2 * 3
E7	Personal part - time.	-1	-1 * 3

E8	Dificultad del lenguaje de programación.	-1	-1 * 3
Total			12.5

Tabla 9 Valores del Factor de ambiente

Entonces, **EF** = $1.4 - 0.03 * 12.5 = 1.025$

Finalmente, **UCP (Puntos de Caso de Uso ajustados)** = $UUCP * TCF * EF = 81 * 1.08 * 1.025 = 89.667$

5.2.2. Estimación del esfuerzo

El esfuerzo en horas - hombre viene dado por:

E = UCP * CF	Donde, UCP: Puntos de Casos de Uso ajustados. CF: Factor de conversión.
---------------------	---

Tabla 10 Cálculo del esfuerzo.

Para calcular el Factor de conversión se contabilizan cuántos factores de los que afectan el factor ambiente están por debajo del valor medio, para los factores E1 a E6.

También se contabilizan cuántos factores de los que afectan el factor ambiente están por encima del valor medio, para los factores E7 y E8.

Entonces, si:

(Total EF \leq 2): CF = 20 horas-hombre

(Total EF = 3 ó Total EF = 4): CF = 28 horas-hombre

(Total EF \geq 5) CF = Hacer cambios en proyecto ya que el riesgo de fracaso es alto.

Para el caso del módulo hay un total de 2 que cumplen con las condiciones planteadas por lo que el Factor de conversión es 20 horas – hombre.

Entonces, **E** = $UCP * CF = 89.667 * 20 = 1793.34 \approx 1793$ horas – hombre.

Por lo que el esfuerzo en el desarrollo de las funcionalidades de los casos de uso es 1793 horas - hombre.

Finalmente, para llevar a cabo una estimación más completa de la duración total del módulo, se agrega el esfuerzo de las demás actividades. Para ello se plantea la distribución del esfuerzo entre las diferentes actividades del proyecto, según la siguiente aproximación:

Actividad	Porcentaje	Horas-Hombre
Análisis	10%	448.25
Diseño	20%	896.5
Programación	40%	1793
Pruebas	15%	672.375
Otras actividades	15%	672.375
Total	100%	4482.5

Tabla 11 Esfuerzo del proyecto

Conociendo además, que un mes tiene como promedio 30 días; que diario se trabaja en el proyecto 8 horas aproximadamente y que la cantidad de horas que trabaja una persona en 1 mes es 240 horas entonces:

Si el Esfuerzo total (ET) es de 4482.5 horas- hombre. Se estima a partir de los datos conocidos que una persona puede realizar el trabajo en aproximadamente en **18 meses**.

Pero teniendo en cuenta que:

CH (Cantidad de hombres): 2 ----- Salario promedio mensual: \$100

CHM (Costo por hombre/mes) = CH * Salario Promedio = 2 * 100 = \$200.00 / mes

Costo = CHM * ET (meses) / CH = 200 * 18/2 = \$1800

Tiempo = ET (meses)/ CH = 18 / 2 = 9 meses.

A partir de los resultados obtenidos se puede demostrar que con dos hombres trabajando en el módulo, este se puede desarrollar en aproximadamente nueve meses para un costo total asociado de \$1800.

5.3. Beneficios tangibles e intangibles

El módulo “Control de Personal” tiene como principal objetivo solucionar los problemas que existen en el área de los laboratorios con relación al manejo de la información de los trabajadores y brindar una interfaz agradable a los usuarios que interactúen con la aplicación. Por tanto, el beneficio esencial que brinda el módulo es el hecho de poder contar con una aplicación web flexible y dinámica que permita gestionar los datos de los trabajadores de forma rápida y eficiente.

Entre los beneficios intangibles del sistema propuesto se pueden incluir:

- Mejoras en el servicio brindado a los clientes, ya que mediante el sistema es posible tener mejor administrada la información personal de los trabajadores.
- Reducción de los errores al interactuar con los datos de las personas, favoreciendo una mayor precisión en los procesos.
- Reducción del tiempo de ejecución de tareas, lo que posibilita que el trabajo se realice rápido y eficiente.
- Interoperabilidad con otros sistemas a partir de la utilización de los servicios Web.
- Flexibilidad al manejar gran volumen y diversidad de información con rapidez y precisión.
- Mejor aprovechamiento de los recursos tecnológicos que dispone la universidad.
- Mejor capacidad de búsqueda y actualización de información, reduciendo la fuerza de trabajo en el proceso y control de recursos.

5.4. Análisis de costos y beneficios

El módulo “Control de Personal” es una aplicación que va dirigida a los encargados de controlar a los trabajadores de los laboratorios, por lo que su mayor beneficio se refleja en el mejoramiento de los servicios que brindará a usuarios y clientes. Además, teniendo en cuenta que las tecnologías y herramientas utilizadas en el desarrollo del proyecto son libres, no es necesario pagar licencias para su uso. Finalmente, después de analizar el costo del proyecto y los beneficios que se obtendrán, es posible concluir que su elaboración es totalmente factible.

5.5. Conclusiones

La estimación, el cálculo de tamaño del software, el esfuerzo y los costos fueron algunos de los elementos presentados en este capítulo correspondientes al estudio de la factibilidad del sistema para el control del personal de los laboratorios y con ello, se destacaron los beneficios que reportará al proceso de informatización en la UCI, al contribuir a optimizar el servicio en el área de los laboratorios.

CONCLUSIONES

Al concluir este trabajo, con el desarrollo del módulo “Control de Personal” perteneciente al proyecto “Sistema Integral de Gestión de los Laboratorios”, se solucionaron satisfactoriamente los problemas existentes en cuanto a la gestión de la información de los trabajadores y su ubicación en un puesto de trabajo. La puesta en marcha de este sistema permitirá ofrecer mejoras en el área de los laboratorios, garantizando de forma eficiente el desempeño, control y calidad de los servicios.

Durante el progreso de la investigación se obtuvieron los siguientes resultados:

- Se realizó un estudio de varios sistemas utilizados en Cuba y el mundo para gestionar al personal de una entidad, donde se identificaron sus principales ventajas, pero además se detectaron los aspectos no deseados para el desarrollo del módulo.
- Se realizó una breve caracterización de las tecnologías, lenguajes de programación y herramientas propuestas para la realización del proyecto.
- Se representaron los procesos involucrados en el negocio con el uso de IDEF0.
- Se desarrolló el módulo Control de Personal de acuerdo a las necesidades planteadas por el cliente.

RECOMENDACIONES

Con el objetivo de brindar una aplicación que incluya todos los procesos que se relacionan con el Control de personal de los laboratorios, se recomienda lo siguiente:

- Realizar pruebas al sistema, para comprobar que cumpla con las funcionalidades esperadas.
- Perfeccionar la ayuda del usuario.
- Investigar continuamente acerca de nuevas tecnologías informáticas, para garantizar mejoras en futuras versiones del sistema.
- Ampliar para los estudiantes de práctica y alumnos ayudantes de laboratorios las funcionalidades implementadas para los trabajadores.
- Capacitar al personal para el trabajo con la aplicación.

BIBLIOGRAFÍA

Álvarez Romero, Eduardo and Pueyo, Daniel. 2004-2005. *Integration Definition For Funcion Modeling.* [Online] 2004-2005. [Cited: Diciembre 12, 2007.] <http://dmi.uib.es/~burguera/download/IDEF0trabajo.doc>.

Alvarez, Miguel Angel. *Aptana Studio.* [Online] [Cited: Diciembre 7, 2007.] <http://www.desarrolloweb.com/articulos/aptana-studio.html>.

Apache. 1995-2006. *Versión 2.2 de la documentación del Servidor de HTTP Apache.* [Online] 1995-2006. [Cited: Enero 20, 2008.] <http://www.apache2.org/>.

Artiles Visual, Leticia, et al. *PERFECCIONAMIENTO DE LA GESTION UNIVERSITARIA EN EL ISCMH: Proyecto Tocaroro.* [Online] [Cited: Enero 8, 2008.] http://www.ucmh.sld.cu/rhab/vol5_num1/rhcm10106.htm.

ASSETS. 2004. *Recursos Humanos.* [Online] 2004. [Cited: Diciembre 5, 2007.] <http://www.assets.co.cu/humanos.asp>.

—. *2004. Qué es ASSETS?* [Online] 2004. [Cited: Diciembre 5, 2007.] <http://www.assets.co.cu/assets.asp>.

Departamento de Control de Calidad y Auditoría Informática. 2001. *Sistemas en Arquitectura Cliente/Servidor.* [Subdirección de Sistemas] s.l.: Dirección General de Servicios de Cómputo Académico, 2001.

DesarrolloWeb. *Qué es PHP.* [Online] [Cited: Noviembre 17, 2007.] <http://www.desarrolloweb.com/articulos/392.php>.

Edinet. 2002. *Qué es XML?* [Online] 2002. [Cited: Diciembre 12, 2007.] <http://www.edinet.com/sabia2.asp>.

Eguíluz Pérez, Javier. 2007. *Libros Web. Introducción a JavaScript.* [Online] 2007. [Cited: Diciembre 5, 2007.] <http://www.librosweb.es/javascript>.

- . **2007.** Libros Web. *Introducción a CSS.* [Online] 2007. [Cited: Enero 20, 2008.] <http://www.librosweb.es/css>.
- . **2007.** Libros Web. *Introducción a XHTML.* [Online] 2007. [Cited: Enero 15, 2008.] <http://www.librosweb.es/xhtml>.
- . **2007.** Libros Web. *Introducción a AJAX.* [Online] 2007. [Cited: Enero 8, 2008.] <http://www.librosweb.es/ajax>.
- Ext JS. 2006.** *Ext JS - JavaScript Library.* [Online] 2006. [Cited: Diciembre 10, 2007.] [http://extjs.com/..](http://extjs.com/)
- Gábilos Software. 2001-2007.** *Gábilos Software.* [Online] 2001-2007. [Cited: Diciembre 2, 2007.] [http://www.gabilos.com/programasdegestion/prestacionesaraper.htm..](http://www.gabilos.com/programasdegestion/prestacionesaraper.htm)
- GNU Image Manipulation Program. 2001.** *GIMP.* [Online] 2001. [Cited: Diciembre 10, 2007.]
- HTML.net.** Tutoriales sobre HTML y CSS - Construye tu propio sitio web. [Online] [Cited: Marzo 15, 2008.] <http://es.html.net/tutorials/css/lesson1.asp>.
- HTMLPOINT.com. 1997-2006.** *¿Qué es Apache?* [Online] 1997-2006. [Cited: Diciembre 10, 2007.] <http://www.htmlpoint.com/faq/apache/01.htm>.
- Lago, Ramiro. 2007.** *Patrón "Modelo-Vista-Controlador".* [Online] 2007. [Cited: Diciembre 10, 2007.] <http://www.proactiva-calidad.com/java/patrones/mvc.html>.
- López Guzmán, Clara. 2000.** *Modelo para el Desarrollo de Bibliotecas Digitales Especializadas.* [Online] Marzo 2000. [Cited: Enero 20, 2008.] http://www.bibliodgsca.unam.mx/tesis/tes7c1lg/sec_26.htm.
- Maestros del Web. 1997-2007.** *Conociendo los principales editores de páginas web del mercado.* [Online] 1997-2007. [Cited: Enero 10, 2008.] <http://www.maestrosdelweb.com/editorial/conociendo-los-principales-editores-de-paginas-web-del-mercado/>.
- Microsoft. 2006.** *La Arquitectura Orientada a Servicios (SOA) de Microsoft.* [Online] Diciembre 2006. [Cited: Marzo 28, 2008.] http://download.microsoft.com/download/c/2/c/2ce8a3a-b4df-4a12-ba18-7e050aef3364/070717-Real_World_SOA.pdf.

- pgAdmin PostgreSQL Tools** . *pgAdmin*. [Online] [Cited: Diciembre 10, 2008.] <http://www.pgadmin.org/>.
- Pitschke, Jurguen. 2002-2008.** *Visual Paradigm Suite*. [Online] 2002-2008. [Cited: Enero 20, 2008.] <http://www.visual-paradigm.eu/content/view/12/1/lang,en/>.
- ProgramacionWeb.net . 2003-2008.** *MVC - Modelo Vista Controlador*. [Online] 2003-2008. [Cited: Enero 9, 2008.] <http://www.programacionweb.net/articulos/articulo/?num=505>.
- Rumbaugh, James, Jacobson, Ivar and Booch, Grady. 2000.** *El Lenguaje Unificado de Modelado*. [Online] 2000. [Cited: Enero 10, 2008.] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.
- San Félix, Alvaro del Castillo. 2000.** *El servidor de web Apache: Introducción práctica*. [Online] 2000. [Cited: Enero 8, 2008.] <http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/index.html>.
- Sánchez González, Carlos. 2004.** ONess: un proyecto open source para el negocio textil mayorista desarrollado con tecnologías open source innovadoras. [Online] Septiembre 28, 2004. [Cited: Enero 20, 2008.] <http://oness.sourceforge.net/proyecto/html/ch03s02.html>.
- Visual Paradigm.** *Visual Paradigm for UML*. [Online] [Cited: Diciembre 10, 2007.] <http://www.visual-paradigm.com/product/vpuml/>.
- Vizcaíno, Aurora, García, Felix Óscar and Caballero, Ismael.** *Una Herramienta CASE para ADOO: Visual Paradigm*. [Online] [Cited: Noviembre 27, 2007.] http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf.
- Welling, Luke and Thomson, Laura.** *Desarrollo Web con PHP y MySQL*. [Online] [Cited: Enero 8, 2008.] <http://bibliodoc.uci.cu/pdf/reg02819.pdf>.
- World Wide Web Consortium. 1994-2005.** *Guía Breve de Tecnologías XML* . [Online] 1994-2005. [Cited: Noviembre 25, 2007.] <http://www.w3c.es/divulgacion/guiasbreves/TecnologiasXML>.
- . **1994-2005** . *Guía Breve de Servicios Web*. [Online] 1994-2005 . [Cited: Abril 30, 2008.] <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.

Worsley, John and Drake, Joshua. 2001. *Características de PostgreSQL*. [Online] 2001. [Cited: Noviembre 29, 2007.] <http://www.sobl.org/traduccion/practical-postgres/node19.html>.

—. **2001.** *¿Qué es PostgreSQL?* [Online] 2001. [Cited: Noviembre 29, 2007.] <http://www.sobl.org/traduccion/practical-postgres/node12.html>.

XperimentoS. *Los mejores 12 Frameworks JavaScript*. [Online] [Cited: Noviembre 25, 2007.] <http://www.xperimentos.com/2007/09/04/los-mejores-12-frameworks-javascript/>.

ANEXOS

Anexo 1. Descripción textual de los casos de uso del sistema

Caso de Uso:	Gestionar trabajador	
Actores:	Especialista Superior de Formación	
Resumen:	El caso de uso se inicia cuando llega un nuevo trabajador para el área de los laboratorios. El Especialista Superior busca los datos y los inserta en el sistema. También se brinda la posibilidad de eliminar datos de un trabajador en caso de presentar la baja de la universidad, así como mostrar sus principales datos.	
Precondiciones:	El Especialista Superior de Formación debe estar autenticado en el sistema.	
Referencias	R1.1, R1.2, R2.1	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Selecciona la opción de Insertar Trabajador.	2. Muestra una ventana para realizar una búsqueda. Y permite: <ul style="list-style-type: none"> • Buscar por el solapín. • Buscar Todos. • Cancelar la operación en cualquier instante. 	
3. Selecciona Buscar Todos.	4. Busca los datos correspondientes en los servicios que brinda la universidad. 5. Comprueba que los datos encontrados no hayan sido insertados anteriormente. 6. Muestra el resultado de la búsqueda, a partir de la comprobación realizada y con los siguientes datos: <ul style="list-style-type: none"> • Nombre, apellidos, CI y Cargo Ocupacional. 	
7. Selecciona cada uno de los trabajadores que desea insertar. 8. Presiona el botón Insertar.	9. Muestra un aviso de que fue insertado satisfactoriamente. 10. Termina el caso de uso.	
Flujos Alternos		
a. El actor selecciona la opción "cancelar".		
Acción del Actor	Respuesta del Sistema	
	a.1. Se cierra la ventana de búsqueda.	

	a.2. Se termina el caso de uso.
b. El actor realiza la búsqueda por solapín.	
Acción del Actor	Respuesta del Sistema
	b.1. Regresa al paso 4 del flujo de trabajo.
6. No muestran resultados en la búsqueda.	
Acción del Actor	Respuesta del Sistema
	6.1. Muestra un aviso de que no existe ninguna persona que cumpla con el criterio seleccionado.
7. El actor no selecciona ningún trabajador a insertar y presiona el botón de esta acción.	
Acción del Actor	Respuesta del Sistema
	7.1. Muestra un mensaje de que debe seleccionar un trabajador. 7.2. Regresa al paso 7 del flujo de trabajo.
Sección "Eliminar trabajador"	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción de Eliminar Trabajador.	2. Muestra una ventana para realizar una búsqueda. Y permite: <ul style="list-style-type: none"> • Buscar por solapín. • Cancelar la operación en cualquier momento.
3. Inserta el Solapín para realizar la búsqueda. 4. Presiona el botón Buscar.	5. Muestra el resultado de la búsqueda con los siguientes datos: <ul style="list-style-type: none"> • Nombre, apellidos, Solapín y Usuario.
6. Selecciona la fila con los datos del Trabajador que desea Eliminar. 7. Presiona el botón Eliminar.	8. Muestra una ventana de confirmación.
9. Confirma que desea eliminar.	10. Elimina los datos seleccionados. 11. Muestra un aviso de que fue eliminado satisfactoriamente. 12. Se termina el caso de uso.
Flujos Alternos	
a. El actor selecciona Cancelar la operación.	
Acción del Actor	Respuesta del Sistema
	a.1. Se cierra la ventana. a.2. Termina el caso de uso.
3. El actor no inserta el solapín y selecciona Buscar.	
Acción del Actor	Respuesta del Sistema

	3.1. Muestra un aviso de que debe llenar el campo obligatorio. 3.2. Regresa al paso 3 del flujo de trabajo.
5. No muestra ningún resultado según el criterio especificado.	
Acción del Actor	Respuesta del Sistema
	5.1. Muestra un aviso de que no existe ninguna persona que cumpla con el criterio seleccionado.
6. No marca ninguna fila para eliminar.	
Acción del Actor	Respuesta del Sistema
	6.1. Muestra un aviso de que debe seleccionar una persona. 6.2. Regresa al paso 6 del flujo de trabajo.
9. No confirma que desea eliminar.	
Acción del Actor	Respuesta del Sistema
	9.1. Se mantiene el listado inicial en caso que el actor retome la acción de eliminar o realizar nuevamente la búsqueda.
Sección "Mostrar Trabajadores"	
Acción del Actor	Respuesta del Sistema
1. Selecciona Mostrar Trabajador.	2. Muestra el listado de los principales datos de los Trabajadores que son: <ul style="list-style-type: none"> • Nombre y Apellidos, Usuario, Solapín y área. 3. Termina el caso de uso.
2. No se encontraron resultados en la búsqueda.	
Acción del Actor	Respuesta del Sistema
	2.1. Muestra un Aviso "No se encontraron resultados en la búsqueda".
Poscondiciones	<ul style="list-style-type: none"> • Se registraron los datos de los Trabajadores. • Se eliminaron los datos de un Trabajador. • Se mostraron los datos de los Trabajadores.

Tabla 12 Descripción textual del caso de uso: Gestionar trabajador

Caso de Uso:	Ubicar trabajador por área	
Actores:	Especialista Superior de Formación	
Resumen:	El caso de uso se inicia cuando el Especialista Superior de Formación verifica la llegada del nuevo personal, para ello puede buscar a todos los que no están en un área o buscar uno específicamente. A partir del resultado mostrado se le posibilita ubicar al personal en un área de trabajo, dicha ubicación estaría a su consideración, en dependencia del área que más personal necesite.	
Precondiciones:	El Especialista Superior de Formación debe estar autenticado en el sistema. Los datos del personal tienen que estar registrados en el sistema.	
Referencias	R3	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Selecciona Ubicar Trabajador en área de trabajo.	2. Muestra una ventana para realizar una búsqueda. Y permite: <ul style="list-style-type: none"> • Buscar por el solapín. • Buscar Todos sin área. • Cancelar la operación en cualquier instante. 	
3. Selecciona buscar todos sin área.	4. Muestra los trabajadores que se corresponden con el criterio seleccionado y de ellos los siguientes datos. <ul style="list-style-type: none"> • Nombre y Apellidos, Usuario, Solapín y área. 5. Lista las áreas por cada trabajador encontrado. 6. Habilita el campo del área para realizar los cambios en la misma.	
7. Especifica el área en que desea ubicar al personal.	8. Muestra un aviso de que fue actualizado correctamente. 9. Termina el caso de uso.	
Flujos Alternos		
a. El actor selecciona la opción "cancelar".		
Acción del Actor	Respuesta del Sistema	
	a.1. Se cierra la ventana de búsqueda. a.2. Se termina el caso de uso.	
3. El actor selecciona la opción Buscar por solapín.		
Acción del Actor	Respuesta del Sistema	
	3.1. Regresa al paso 4 del flujo de trabajo.	

4. No se muestra ningún resultado de la búsqueda.	
Acción del Actor	Respuesta del Sistema
	4.1. Muestra un aviso de que no existe persona con ese criterio.
Poscondiciones	<ul style="list-style-type: none"> Se ubicó el personal en un área de trabajo.

Tabla 13 Descripción textual del caso de uso: Ubicar trabajador por área

Caso de Uso:	<i>Mostrar trabajador por área</i>
Actores:	Visor de información por área
Resumen:	El caso de uso se inicia cuando el Visor de Información por área solicita ver los datos del personal de cada una de las áreas y el sistema proporciona dicha posibilidad, además de mostrar los detalles de un trabajador seleccionado.
Precondiciones:	El Visor de información por área debe estar autenticado en el sistema.
Referencias	R2.2
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona del área deseada la opción ver trabajadores.	2. Muestra los datos fundamentales del trabajador, que son: <ul style="list-style-type: none"> Nombre y Apellidos, Usuario, Solapín, Turno y Cargo que ocupa. Además el sistema permite: <ul style="list-style-type: none"> Mostrar los detalles de un trabajador seleccionado.
3. Selecciona un trabajador.	4. Muestra los siguientes datos del trabajador: Foto, Carnet de Identidad, Teléfono, Dirección y Cargo Ocupacional. 5. El caso de uso termina.
Poscondiciones	<ul style="list-style-type: none"> Se mostraron los datos por área de los trabajadores.

Tabla 14 Descripción textual del caso de uso: Mostrar trabajador por área

Caso de Uso:	<i>Mostrar trabajador por turno</i>
Actores:	Visor de información por turno
Resumen:	El caso de uso se inicia cuando el Visor de Información por turno necesita ver los datos del personal de cada uno de los turnos y el sistema proporciona dicha posibilidad.

Precondiciones:	El Visor de información por turno debe estar autenticado en el sistema.
Referencias	R2.3
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona del turno deseado la opción ver trabajadores.	2. Muestra los datos fundamentales del trabajador, que son: <ul style="list-style-type: none"> • Nombre y Apellidos, Usuario, Solapín y Cargo que ocupa. Además el sistema permite: <ul style="list-style-type: none"> • Mostrar los detalles de un trabajador seleccionado.
3. Selecciona un trabajador.	4. Muestra los siguientes datos del trabajador: Foto, Carnet de Identidad, Teléfono, Dirección y Cargo Ocupacional. 5. El caso de uso termina.
Poscondiciones	<ul style="list-style-type: none"> • Se mostraron los datos por turno de los Trabajadores.

Tabla 15 Descripción textual del caso de uso: Mostrar trabajador por turno

Caso de Uso:	<i>Ubicar trabajador por turno</i>
Actores:	Jefe de área
Resumen:	El caso de uso se inicia cuando el Jefe de área verifica mediante una búsqueda al personal de su área que desea ubicar y después de mostrados los resultados especifica un turno de trabajado, esta ubicación estaría a su consideración, en dependencia del turno que más personal necesite.
Precondiciones:	El Jefe de área debe estar autenticado en el sistema.
Referencias	R4
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona Ubicar Trabajador.	2. Muestra una ventana para realizar una búsqueda. Y permite: <ul style="list-style-type: none"> • Buscar por el solapín. • Buscar todos sin turno. • Cancelar la operación en cualquier instante.
3. Selecciona buscar todos sin Turno.	4. Muestra los trabajadores que se corresponden con el criterio seleccionado y de ellos los siguientes datos.

	<ul style="list-style-type: none"> Nombre y Apellidos, Usuario, Solapín y Cargo que ocupa.
7. Especifica el turno en que desea ubicar al personal.	5. Lista todos los turnos del área a la que pertenece el actor para los trabajadores encontrados. 6. Habilita el campo del turno para realizar los cambios en el mismo. 8. Muestra un aviso de que fue actualizado correctamente. 9. Termina el caso de uso.
Flujos Alternos	
a. El actor selecciona la opción "cancelar".	
Acción del Actor	Respuesta del Sistema
	a.1. Se cierra la ventana de búsqueda. a.2. Se termina el caso de uso.
3. El actor selecciona la opción Buscar por solapín.	
Acción del Actor	Respuesta del Sistema
	1.1. Regresa al paso 4 del flujo de trabajo.
4. No se muestra ningún resultado de la búsqueda.	
Acción del Actor	Respuesta del Sistema
	4.1. Muestra un aviso de que no existe persona con ese criterio.
Poscondiciones	<ul style="list-style-type: none"> Se ubicó el personal en un turno de trabajo dentro de un área.

Tabla 16 Descripción textual del caso de uso: Ubicar trabajador por turno

Caso de Uso:	<i>Seleccionar Jefe de Turno</i>
Actores:	Jefe de área
Resumen:	El caso de uso se inicia cuando el Jefe de área desea seleccionar los responsables de cada turno de su área. El sistema busca los integrantes por turno y permite seleccionar el jefe de turno que asumirá las tareas correspondientes a su cargo.
Precondiciones:	El Jefe de área debe estar autenticado en el sistema.
Referencias	R7
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona Elegir Jefes de turno.	2. Muestra los turnos del área a la que corresponde el actor.

	Y permite: <ul style="list-style-type: none"> • Realizar una búsqueda. • Cancelar la operación.
3. Selecciona la opción de realizar la búsqueda.	5. Muestra los trabajadores del turno seleccionado y de ellos los siguientes datos: <ul style="list-style-type: none"> • Nombres, apellidos, Usuario, Solapín, Cargo que ocupa.
4. Selecciona uno de los turnos.	
6. Selecciona el jefe de turno.	8. Marca con una imagen el nuevo jefe de turno.
7. Elige la opción de Aceptar.	9. El caso de uso termina.
Flujos Alternos	
a. El actor selecciona la opción "cancelar".	
Acción del Actor	Respuesta del Sistema
	a.1. Se cierra la ventana. a.2. Se termina el caso de uso.
5. No se muestran resultados dado el turno seleccionado.	
Acción del Actor	Respuesta del Sistema
	5.1. Muestra un Aviso "No se encontraron resultados en la búsqueda".
Poscondiciones	<ul style="list-style-type: none"> • Se seleccionaron los jefes de turno.

Tabla 17 Descripción textual del caso de uso: Seleccionar Jefe de Turno

Caso de Uso:	<i>Ubicar trabajador por laboratorio</i>
Actores:	Jefe de turno
Resumen:	El caso de uso se inicia cuando el Jefe de turno de un área determinada verifica mediante una búsqueda al personal que desea ubicar y después de mostrados los resultados especifica un laboratorio, esta ubicación estaría a su consideración e incluso puede existir personal responsable de más de un laboratorio. El sistema también brinda la posibilidad de quitar laboratorios a trabajadores que lo tengan asignado.
Precondiciones:	El Jefe de turno debe estar autenticado en el sistema.
Referencias	R5
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona asignar laboratorio a	2. Muestra una ventana para realizar una búsqueda.

trabajador.	Y permite: <ul style="list-style-type: none"> • Buscar por el solapín. • Buscar todos los que no tienen laboratorio. • Cancelar la operación en cualquier instante.
3. Selecciona buscar todos sin laboratorio.	4. Muestra los trabajadores que se corresponden con el criterio seleccionado y de ellos los siguientes datos: <ul style="list-style-type: none"> • Nombre, Apellidos, Usuario y Solapín.
5. Selecciona al trabajador que desea ubicar. 6. Presiona el botón Asignar laboratorio.	7. Muestra una ventana la asignación de laboratorios que tiene la persona seleccionada. 8. Muestra un listado de los laboratorios del área a los que puede ser asignado. Y brinda la posibilidad de: <ul style="list-style-type: none"> • Agregar laboratorio. • Quitar laboratorio asignado. • Cancelar la operación.
9. Selecciona del listado de laboratorios el que le desea asignar al trabajador. 10. Selecciona Agregar laboratorio.	11. Actualiza la ventana con los laboratorios asignados. 12. Termina el caso de uso.
Flujos Alternos	
a. El actor selecciona la opción "cancelar".	
Acción del Actor	Respuesta del Sistema
	a.1. Se cierra la ventana. a.2. Se termina el caso de uso.
3. El actor selecciona la opción Buscar por solapín.	
Acción del Actor	Respuesta del Sistema
	3.1. Regresa al paso 4 del flujo de trabajo.
4. No se encuentra ninguna persona que cumpla con los criterios seleccionados.	
Acción del Actor	Respuesta del Sistema
	4.1. Muestra un aviso de que no existe la persona con ese criterio.
5. No selecciona la persona que desea ubicar pero presiona el botón de la acción.	
Acción del Actor	Respuesta del Sistema
	5.1. Muestra un aviso de que debe seleccionar una persona.
b. Marca en la ventana de asignación un laboratorio a eliminar y presiona el botón de la acción.	
Acción del Actor	Respuesta del Sistema

	b.1. Elimina la asignación de laboratorio. b.2. Actualiza la ventana de asignación.
Poscondiciones	<ul style="list-style-type: none"> Se ubicó el personal en un laboratorio.

Tabla 18 Descripción textual del caso de uso: Ubicar trabajador por laboratorio

Caso de Uso:	Generar Reporte	
Actores:	Visor de información por área	
Resumen:	El caso de uso se inicia cuando el Visor de Información por área desea imprimir los datos de los trabajadores dado un criterio de búsqueda. El sistema muestra el listado y permite imprimir el mismo.	
Precondiciones:	El Visor de información por área debe estar autenticado en el sistema.	
Referencias	R6	
Prioridad	Secundario	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. Selecciona Generar Informes.	2. Muestra una ventana con los siguientes parámetros. <ul style="list-style-type: none"> Áreas, Fecha de alta, Graduado de, Cargo ocupacional y militante. El sistema permite: <ul style="list-style-type: none"> Realizar una búsqueda. Cancelar la operación.
	3. Selecciona la opción de realizar la búsqueda especificando los criterios. 4. Selecciona la opción de generar documento pdf.	5. Muestra en una ventana los resultados obtenidos con la opción de Generar Reporte para imprimir, y de los trabajadores los siguientes datos: <ul style="list-style-type: none"> Nombre, apellidos, usuario y solapin. 6. El caso de uso termina.
Flujos Alternos		
3. El actor no especifica ningún criterio de búsqueda y selecciona la opción de generar documento.		
	Acción del Actor	Respuesta del Sistema
		a.1. Muestra un mensaje especificando que debe llenar al menos un campo.
Poscondiciones	<ul style="list-style-type: none"> Se generaron los documentos para imprimir la información de los trabajadores. 	

Tabla 19 Descripción textual del caso de uso: Generar Reporte

Anexo 2. Prototipos de Interfaz

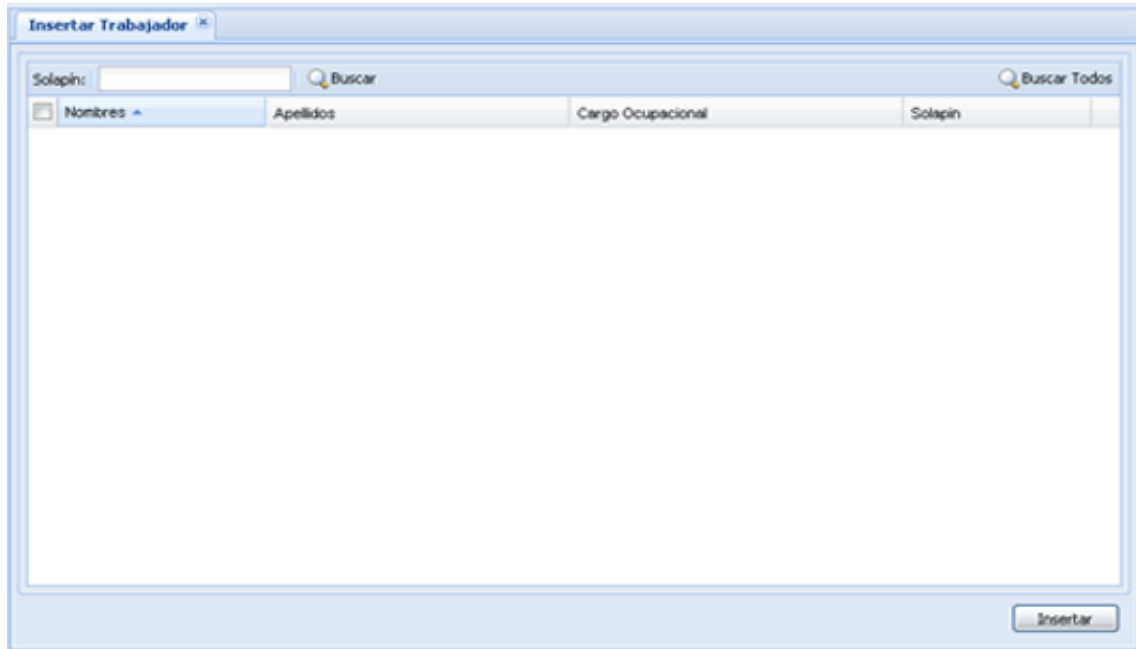


Figura 35 Prototipo: Gestionar trabajador_Sección: Insertar

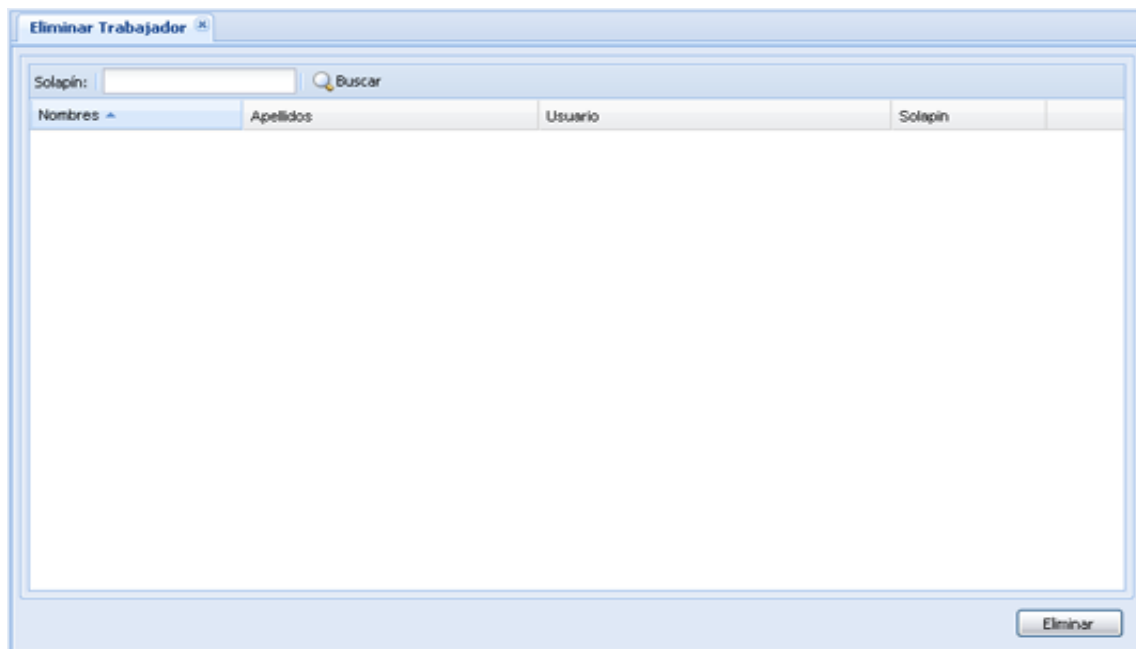


Figura 36 Prototipo: Gestionar trabajador_Sección: Eliminar

Listado de todos los Trabajadores				
Nombres	Apellidos	Usuario	Solapin	Area
[-] Area: Direccion de los Laboratorios(10 Trabajadores)				
[-] Area: Docente 1(16 Trabajadores)				
[-] Area: Docente 2(6 Trabajadores)				
[-] Area: Docente 3(20 Trabajadores)				
[-] Area: Docente 4(5 Trabajadores)				
AILEN	FERNANDEZ GONZALEZ	allego	T14976	Docente 4
AILEN	CAPOTE FALCON	alinc	T14226	Docente 4
AMAYDA	DOMINGUEZ MARTINEZ	amayda	12616	Docente 4
AMEDH	RODRIGUEZ VERDECIA	amedh	12400	Docente 4
ARACELYS IVANA	VALERIO CONCEPCION	aracelysvc	10571	Docente 4
[-] Area: Docente 5(4 Trabajadores)				
[-] Area: Informatizacion(5 Trabajadores)				
[-] Area: Infraestructura Productiva(3 Trabajadores)				
[-] Area: Produccion(2 Trabajadores)				
[-] Area: Sin Area(6 Trabajadores)				

Figura 37 Prototipo: Gestionar trabajador_Sección: Mostrar

Ubicar Trabajador				
Nombres	Apellidos	Usuario	Solapin	Area
ALBERTO	FERNANDEZ LABRADOR	albertof	08809	Sin Area
ALCIDES	LAURENCO RICHARD	alcidest	13172	Direccion de los Laboratorios
ANDY	BAÑADOS VILAN	andybv	12594	Docente 1
DENIS	ALONSO PEREZ	denisa	10909	Docente 2
DINORAH NEYDA	FERNANDEZ GOMEZ	dinorah	08905	Docente 3
DIONISIO	GONZALEZ BORROSO	dioni	14053	Docente 4
EDUARDO	GUTIERREZ AMARAN	eduardog	12615	Docente 5
KENEL	GONZALEZ SANTIAGO	kenel	11225	Informatizacion
				Infraestructura Productiva
				Produccion

Page 1 of 1 | Mostrando del 1 al 8 de 8

Figura 38 Prototipo: Ubicar trabajador por área


Listado de los Trabajadores del Docente 1
Detalles

Categ.	Nombres	Apellidos	Usuario	Solapin	Turno
37	ADIARY	LLANES PEREZ	adiary	13175	1er Turno
	ALEJANDRO CARLOS	LOPEZ BUTTARI	alejo	06267	
	ALEXANDER	ROSALES HERNANDEZ	alexrh	T15203	
	AQUILES	PEREZ MIRANDA	aquiles	10595	4to Turno
	ARTEMA REGLA	MEDINA DE LAS CUEVAS	artemia	09576	
37	ASHLEY	SERRANO CUBERO	ashley	09585	
	AYLEMS	GARCIA RIVERA	aylems	12396	
37	AYLEN RAMONA	GRAVERAN LOPEZ	aylen	09094	2do Turno
	AYME	BAEZ BOFILL	abofill	T14884	
	BARBARA YETNEY	MORALES LAVOY	barbaraml	06304	4to Turno
	BORIS LUIS	RODRIGUEZ PEREZ	borisluis	11795	4to Turno
	BORIS RAMON	RIVERO FERNANDEZ	borisrrf	07814	
37	CARLOS	HERRERA FUMERO	hfcarlos	T14224	
37	CARLOS ERNESTO	CABRERA HERNANDEZ		T14937	4to Turno
	DAIMARIS	PROL ALMAGUER	dprol	T14899	
	ENRIQUE SANTIAGO	MORENO MOURIZ	mouriz	T14943	

Page 1 of 1
Mostrando del 1 al 16 de 16

Detalles
37

Foto



Telefono:
2718407

Carnet de Identidad:
88081109582

Direccion:
Calle 256 # 3737 E/37 Y
51 San Agustin

Cargo Ocupacional:
Recien Graduado En
Adiestramiento

Figura 39 Prototipo: Mostrar trabajador por área


Listado de las Personas del 4to Turno
Detalles

Categ.	Nombres	Apellidos	Usuario	Solapin
37	AQUILES	PEREZ MIRANDA	aquiles	10595
	BARBARA YETNEY	MORALES LAVOY	barbaraml	06304
	BORIS LUIS	RODRIGUEZ PEREZ	borisluis	11795
	CARLOS ERNESTO	CABRERA HERNANDEZ		T14937

Page 1 of 1
Mostrando del 1 al 4 de 4

Detalles
37

Foto



Telefono:
2718407

Carnet de Identidad:
88081109582

Direccion:
Calle 256 # 3737 E/37 Y
51 San Agustin

Cargo Ocupacional:
Recien Graduado En
Adiestramiento

Figura 40 Prototipo: Mostrar trabajador por turno

Asignar Turno a un Trabajador

Solapín: Buscar Todos sin Turno

Nombres	Apellidos	Usuario	Solapín	Turno de Trabajo
ALEJANDRO CARLOS	LOPEZ BUTTARI	alejo	06267	Sin Turno
ALEXANDER	ROSALES HERNANDEZ	alexrh	T15203	Sin Turno
ARTEMA REGLA	MEDINA DE LAS CUEVAS	artemia	09576	Sin Turno
AYLEMIS	GARCIA RIVERA	aylemis	12396	Sin Turno
AYME	BAEZ BOFILL	abofill	T14884	Sin Turno
BORIS RAMON	RIVERO FERNANDEZ	borisrf	07814	1er Turno
DAIMARIS	PROL ALMAQUER	dprol	T14899	2do Turno
ENRIQUE SANTIAGO	MORENO MOURIZ	mouriz	T14943	3er Turno
				4to Turno

Figura 41 Prototipo: Ubicar trabajador por turno

Seleccionar Jefe de Turno

Turnos:

Categ.	Nombres	Apellidos	Usuario	Solapín	<input type="checkbox"/>
	AQUILES	PEREZ MIRANDA	aquiles	10595	<input type="checkbox"/>
	BARBARA YETNEY	MORALES LAVOY	barbaraml	06304	<input type="checkbox"/>
	BORIS LUIS	RODRIGUEZ PEREZ	borisluis	11795	<input type="checkbox"/>
<input checked="" type="checkbox"/>	CARLOS ERNESTO	CABRERA HERNANDEZ		T14937	<input checked="" type="checkbox"/>

Figura 42 Prototipo: Seleccionar Jefe de turno

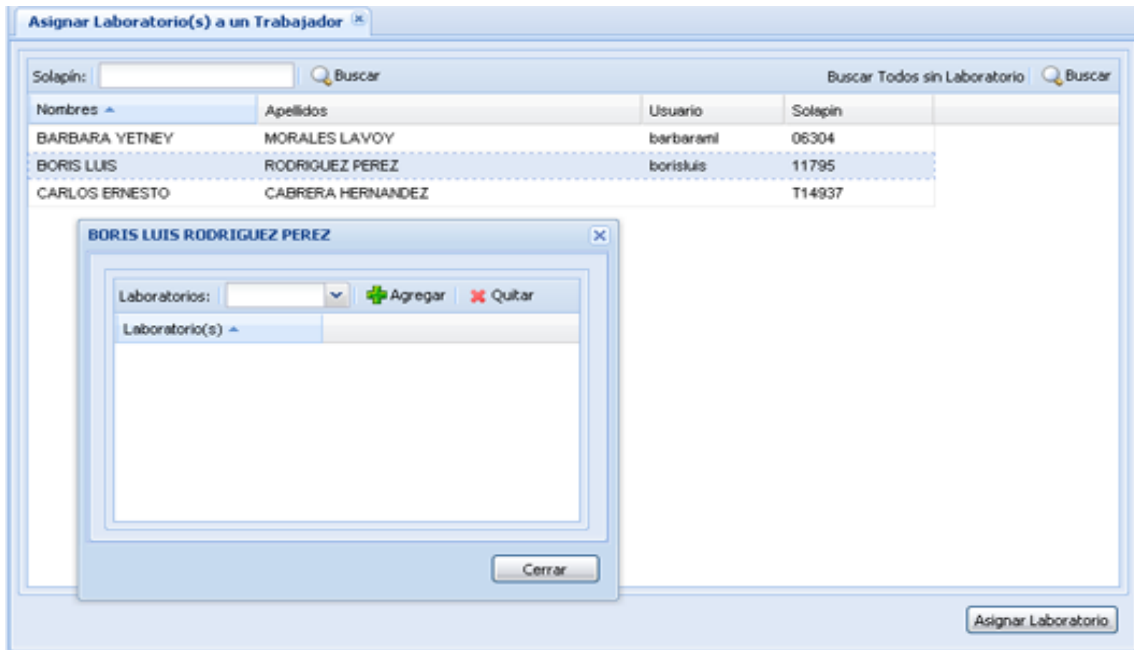


Figura 43 Prototipo: Ubicar trabajador por laboratorio

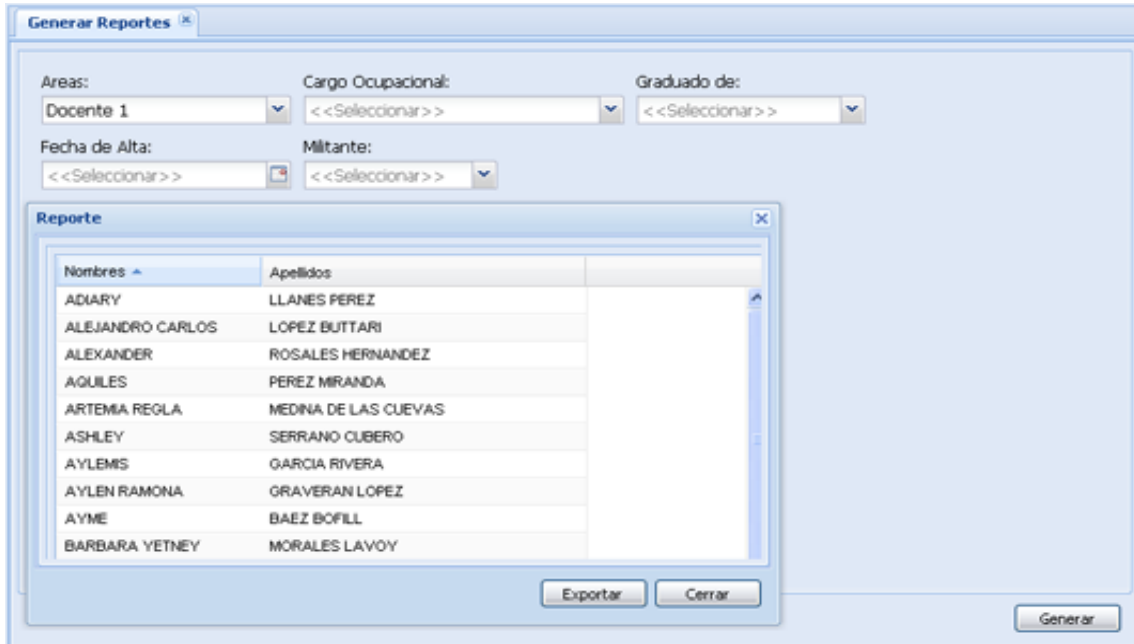
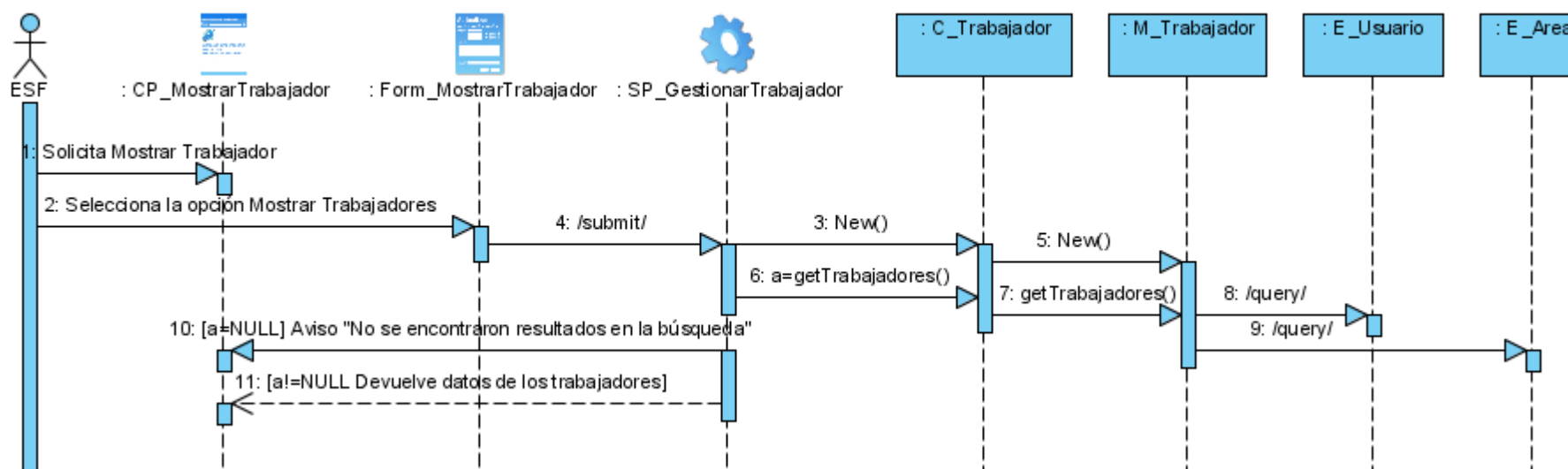


Figura 44 Prototipo: Generar Reporte

Anexo 3. Diagramas de secuencia

Figura 45 Diagrama de secuencia: Gestionar trabajador_ Sección: Mostrar

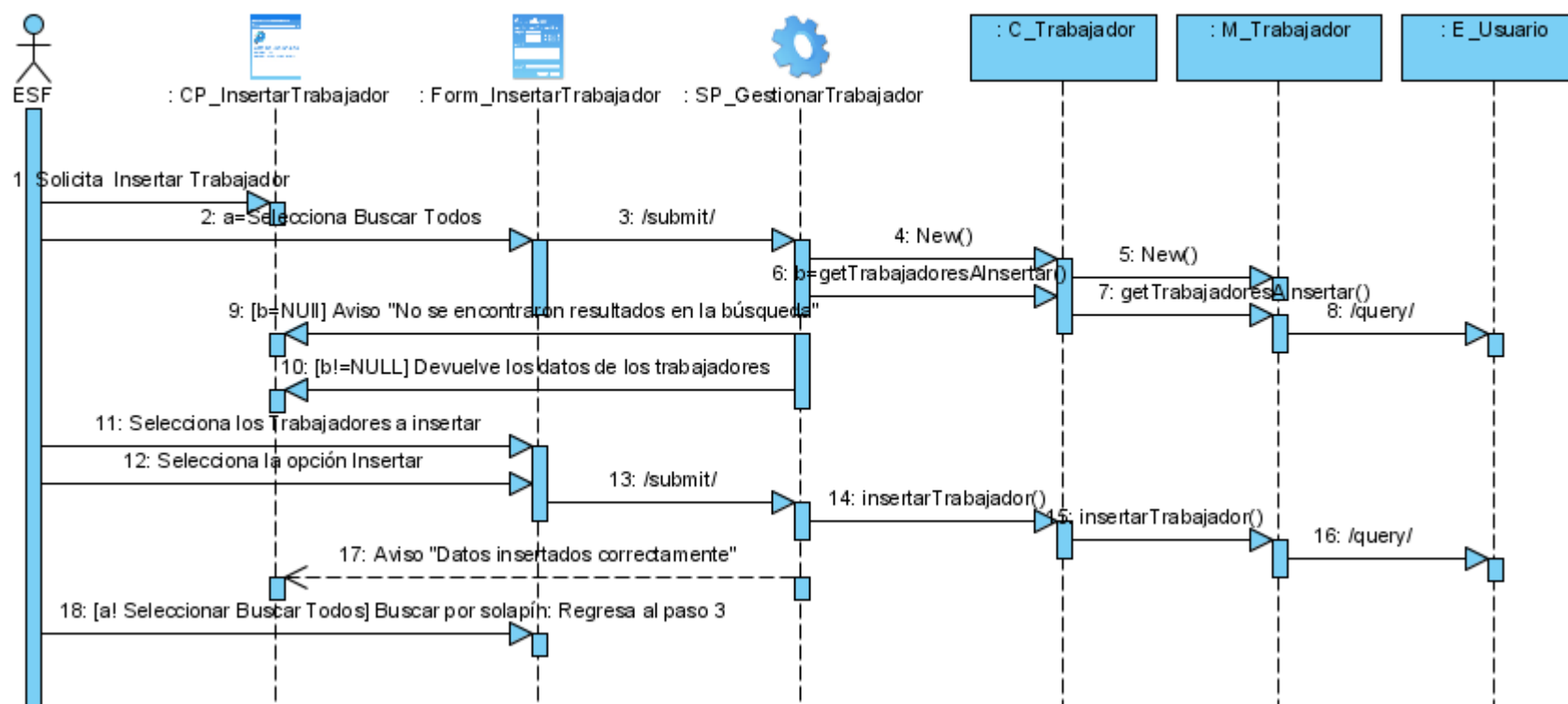


Figura 46 Diagrama de secuencia: Gestionar trabajador_ Sección: Insertar

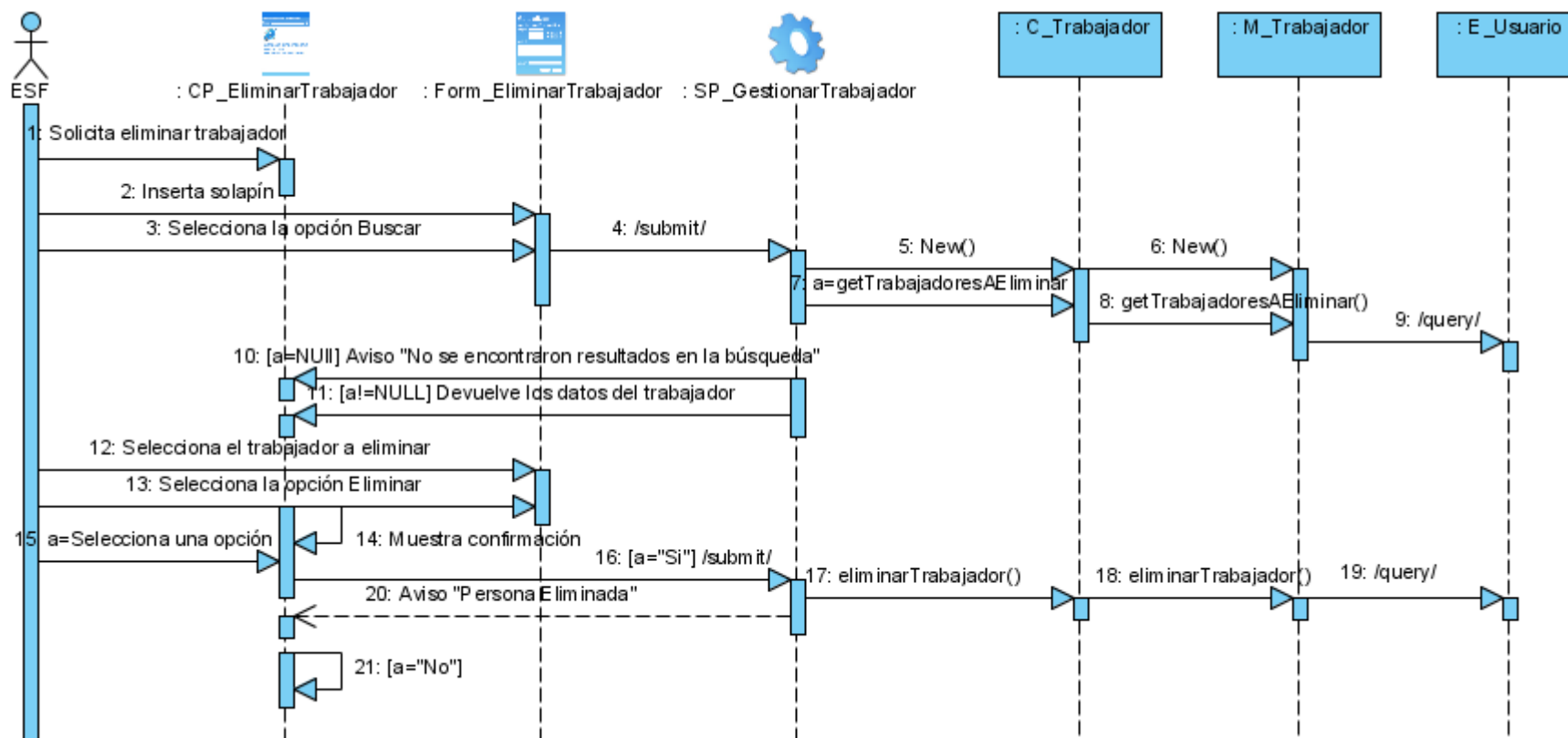


Figura 47 Diagrama de secuencia: Gestionar trabajador_ Sección: Eliminar

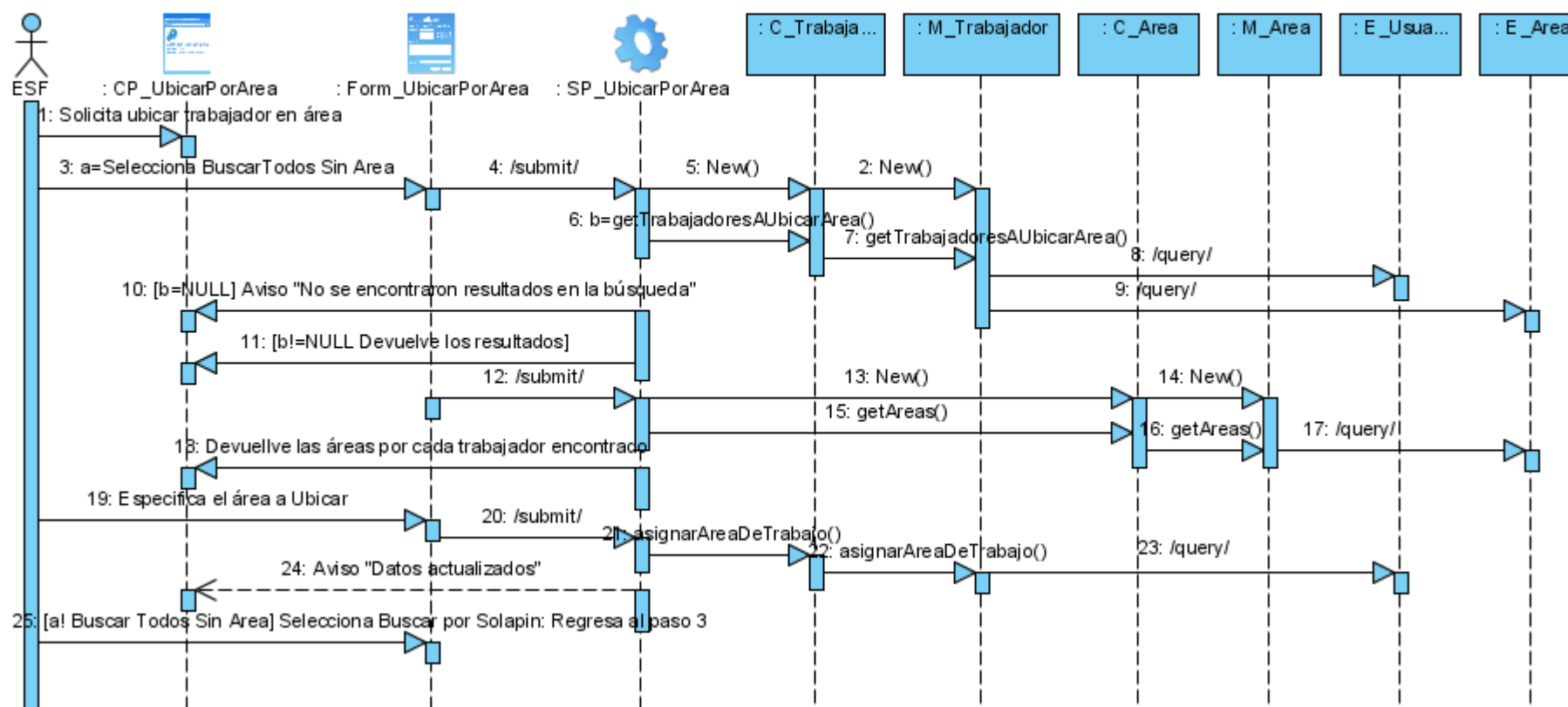


Figura 48 Diagrama de secuencia: Ubicar trabajador por área

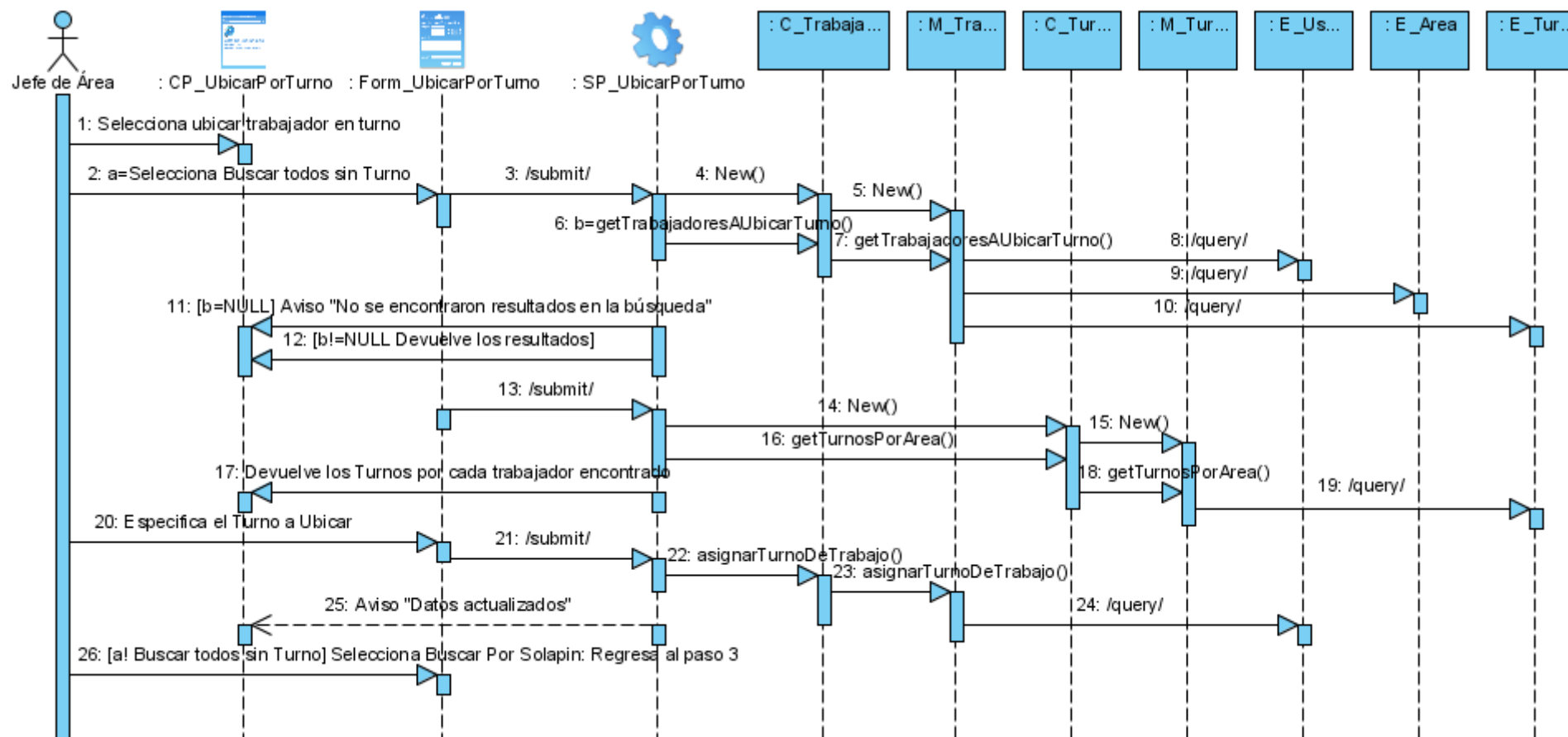


Figura 49 Diagrama de secuencia: Ubicar trabajador por turno

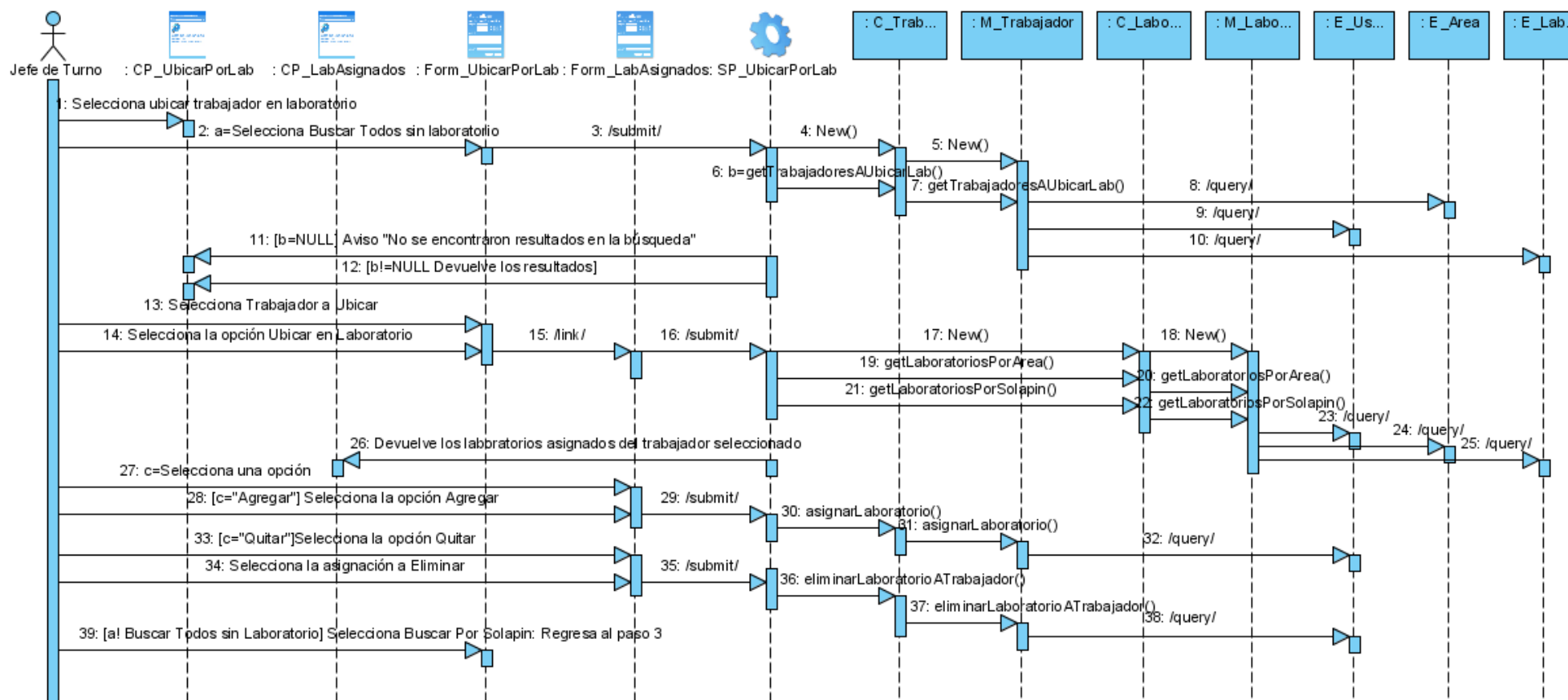


Figura 50 Diagrama de secuencia: Ubicar trabajador por laboratorio

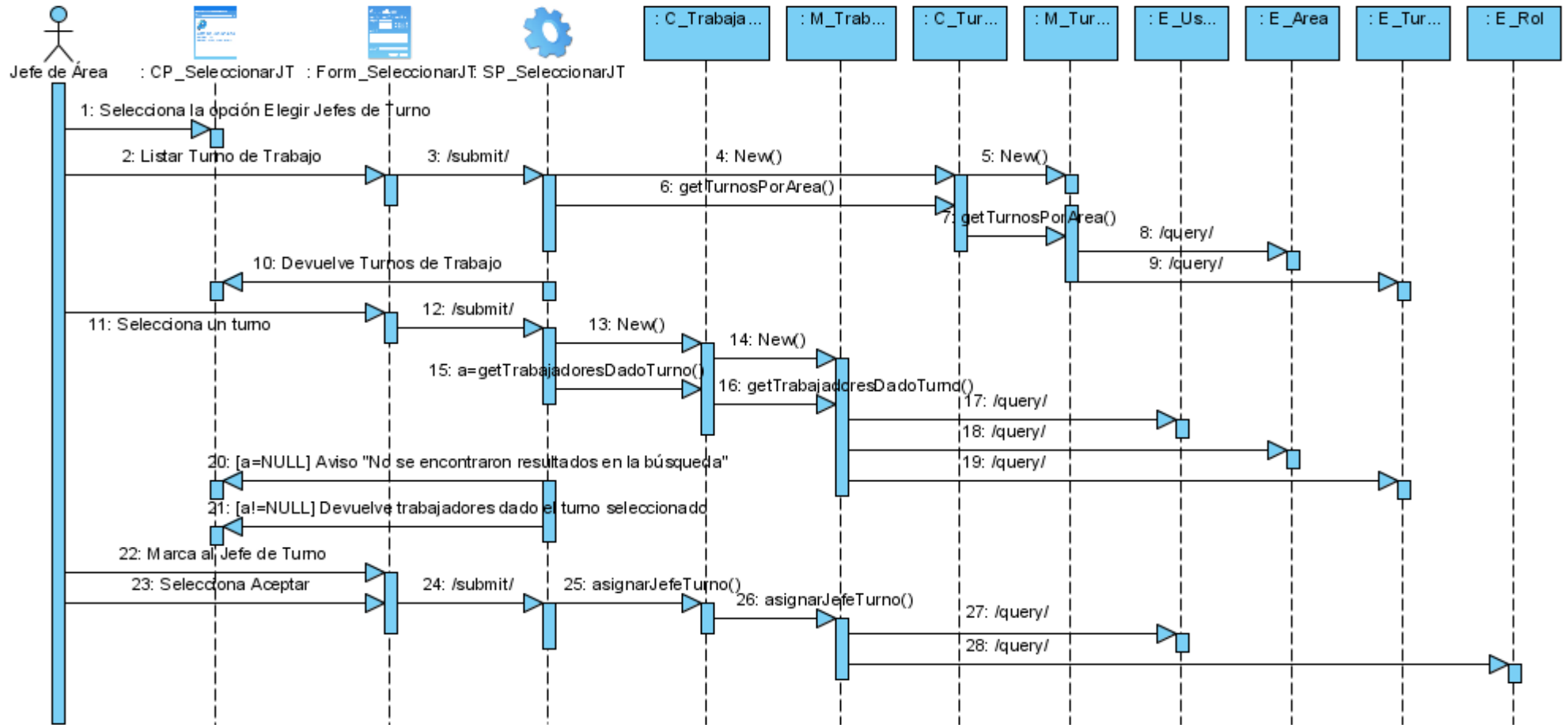


Figura 51 Diagrama de secuencia: Seleccionar Jefe de Turno

GLOSARIO DE TÉRMINOS

A

Aplicaciones Web: Es un sistema informático que los usuarios utilizan accediendo a un servidor Web a través de Internet.

APPSERV: Es una herramienta código abierto para Windows que facilita la instalación de Apache, MySQL y PHP en una sola herramienta.

B

Base de datos: Un conjunto de información almacenada en memoria auxiliar que permite acceso directo y un conjunto de programas que manipulan esos datos.

C

Cookies: Es un fragmento de información que se almacena en el disco duro del visitante de una página Web a través de su navegador.

Código fuente: Un conjunto de líneas que conforman un bloque de texto, escrito según las reglas sintácticas de algún lenguaje de programación destinado a ser legible por humanos.

Código abierto: Es el término con el que se conoce al software distribuido y desarrollado libremente.

D

DOM: Modelo en Objetos para la representación de Documentos. (Document Object Model)

Diagramas: Facilitan el entendimiento de grandes cantidades de datos y la relación entre diferentes partes de los datos.

F

FrameWork: Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

FTP: (File Transfer Protocol) es un protocolo de transferencia de archivos entre sistemas conectados a una red TCP.

G

GPL: La licencia pública general de GNU o más conocida por su nombre en inglés GNU General Public License, es una licencia creada por la Free Software Foundation orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

H

Herramienta Case: Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

HTTPS: es un protocolo de red basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto, es decir, es la versión segura de HTTP.

I

Interfaz: Es la forma en que los usuarios pueden comunicarse con una computadora.

IDE: Integrated Development Environment. Un entorno de desarrollo integrado es un programa compuesto por un conjunto de herramientas para un programador.

J

Java: Es un lenguaje de programación de alto nivel, orientado a objetos.

L

Logs: Es un registro oficial de eventos durante un período de tiempo en particular.

LGPL: GNU Lesser General Public License es una licencia de software creada por la Free Software Foundation, utilizada fundamentalmente en el trabajo con las librerías, que puede ser software libre o software no libre.

M

Mac OS: Macintosh Operating System (Sistema Operativo de Macintosh).

N

Navegador: Software que permite al usuario recuperar y visualizar documentos de hipertexto desde servidores Web a través de Internet.

O

Oracle: Es un sistema de gestión de base de datos relacional.

P

Plataforma: Se refiere al sistema operativo o a sistemas complejos que a su vez sirven para crear programas, como las plataformas de desarrollo.

PDO: Es una extensión que provee una capa de abstracción de acceso a datos para PHP 5.

Perl: Es un lenguaje de programación.

Plug- in: Es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica.

R

Rol: Papel que desempeña cada integrante dentro del ciclo de vida de un proyecto de software.

S

Software: Se refiere al equipamiento lógico o soporte lógico de un computador digital.

Sistema Operativo: Es un conjunto de programas de computadora destinado a permitir una administración eficaz de sus recursos.

Smalltalk: es un sistema informático que permite realizar tareas de computación mediante la interacción con un entorno de objetos virtuales. Metafóricamente, se puede considerar que un Smalltalk es un mundo virtual donde viven objetos que se comunican mediante el envío de mensajes.

T

Tecnologías: Conjunto de teorías y de técnicas que permiten el aprovechamiento práctico del conocimiento científico.

W

WxWidgets: Son unas bibliotecas multiplataforma y libres, para el desarrollo de interfaces gráficas programadas en lenguaje C++.

W3C: es un consorcio internacional donde el personal a tiempo completo y el público en general, trabajan conjuntamente para desarrollar estándares Web.

WSDL: Web Services Description Language, un formato XML que se utiliza para describir servicios Web.

X

XSLT: Transformaciones XSL es un estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML.

XMLHttpRequest: Es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores WEB.