Universidad de las Ciencias Informáticas

Facultad 1

Diseño e implementación del módulo Plan de Distribución de Graduados de Nivel Superior para el Sistema Unificado de Gestión de Fuerza de Trabajo Calificada.

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

Autores: María del Carmen Matías Bordón.

Pedro Salvá Rodriguez.

Tutores: Ing. Aneyty Martín García.

Ing. Enmanuel Azahares Reyes.

Ciudad de La Habana, junio 2010.

"Año 52 de la Revolución"

“Quien tenga una computadora dispone de todos los conocimientos publicados. La privilegiada memoria de la máquina le pertenece también a él”.

“Las ideas nacen de los conocimientos y de los valores éticos. Una parte importante del problema estaría resuelta tecnológicamente, la otra hay que cultivarla sin descanso o de lo contrario se impondrán los instintos más primarios”.

Fidel Castro Ruz

A:

Mi Papá, mi Mamá, mi hermano,

mis tías y mi abuela.

María del C.

A:

Mi Mamá, mi papá, Nápoles, mi hermano,

mis abuelos y a mis tíos.

Pedro.

Agradezco a:

Nuestro Comandante en Jefe Fidel Castro Ruz y a la Revolución Cubana, ya que gracias a ella pude realizar mis sueños de superarme y prepararme para la vida.

Mi mamá y mi papá por servirme de guía y ejemplo, por hacer de mí la persona que soy hoy, por batallar conmigo y con mi hermano todos estos años de universidad.

Mi hermano por ser mi motor impulsor ya que desde niña le sigo los pasos.

Mi tía Mirta que es como mi otra madre, por siempre estar pendiente de que no me falte nada y por su preocupación.

A mi tía Marlen que vive pendiente de mis estudios desde los primeros años, por su amistad ya que ha estado presente en momentos difíciles de mi vida.

Mi abuela Aida por su amor incondicional, su ternura, su preocupación y sus consejos.

A mi compañero de tesis y mi novio Pedro, por su apoyo en todos estos años de estudios, en los momentos malos y en los buenos. Por su comprensión.

A mis tutores Aneyty y Enmanuel por su constante apoyo, y su disposición de ayudar en todo.

A todas las chicas de mi apartamento, Dayana por ayudarme en todo, Yaima, Yailin, Elizabeth, la Chiqui por la buena convivencia a lo largo de estos años.

A Yankiel, Yordan, Alejandro y a todo el grupo 1304 por su amistad incondicional.

Al jefe del proyecto Carlos Tonet, a todos los miembros del equipo de desarrollo Frank, Alexander Norat, Luis Ernesto, Alexander Escalona, Ernesto, Alain, Javier, etc. En general a todas aquellas personas que me han apoyado a lo largo de estos años.

Especialmente a Alberto y Enrique por su ayuda.

Mary

Agradecimientos

Agradezco a:

La Revolución por darme la posibilidad de estudiar y de convertirme en ingeniero.

A mi mamá, a mi papá, a Nápoles, a mis abuelos y a mis tíos que siempre me han servido de ejemplo para convertirme en una persona mejor.

A mi compañera de tesis y novia Mary, por su apoyo en todos estos años de estudio, en los buenos y malos momentos. Por su comprensión y paciencia.

A mis suegros que tanto me han apoyado y que me han acogido en su familia como un hijo más.

A mis tutores Aneyty y Enmanuel por su constante apoyo, y su disposición de ayudar en todo.

A Yankiel, Yordan, Alejandro, Alexander Norat, y a todo el grupo 1304 por su amistad incondicional.

A mis compañeros de proyecto que tanto me han apoyado y ayudado.

Especialmente a Alberto y Enrique por su ayuda, por apoyarme en los buenos y malos momentos.

Pedro

Declaración de autoría

Declaramos que somos los únicos autores del trabajo titulado:

Diseño e implementación del módulo Plan de Distribución de Graduados de Nivel Superior para el Sistema Unificado de Gestión de Fuerza de Trabajo Calificada.

Autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ___ días del mes de _____ del año _____.

_____	_____
María del Carmen Matías Bordón	Pedro Salvá Rodriguez
_____	_____
Ing. Enmanuel Azahares Reyes	Ing. Aneyty Martín García

República de Cuba

MINISTERIO DE ECONOMÍA
Y PLANIFICACIÓN

1960-2010
Aniversario Cincuenta
DE LA PLANIFICACIÓN

Dirección Desarrollo Social

La Habana, 18 de mayo de 2010
"Año 52 de la Revolución"

Opinión del usuario del Trabajo de Diploma

El Sistema Unificado de Gestión de Fuerza de Trabajo Calificada (GeForza), fue desarrollado para el Departamento Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado hasta la fecha, satisfará los procesos de gestión de la información y realización del Plan de Ingreso a la Educación Superior del Curso Regular Diurno, el Plan de Continuidad de Estudio de los estudiantes de noveno grado, el Plan de Ingreso a los Centros Universitarios Municipales y el Plan de Distribución de graduados de la Educación Superior formados en el Curso Regular Diurno.

El producto Sistema Unificado de Gestión de Fuerza de Trabajo Calificada (GeForza) será de gran utilidad para el Departamento Fuerza de Trabajo Calificada, para la organización y realización de sus actividades. Logrará que el trabajo de las personas que laboran de este departamento sea más fácil, así como, conseguirá homogeneizar los nomencladores, establecer y centralizar la seguridad de la información, agilizar el proceso de gestión de la información necesaria para la realización de los planes pertinentes al departamento y la elaboración de los mismos.

Los diplomantes del trabajo han dado muestra de preparación y creatividad, lo que está permitiendo dar solución a los problemas que presentan las aplicaciones actuales.

Y para que así conste, se firma la presente a los 18 días del mes de mayo del año 2010.

Lic. Virginia Martín Martínez
J' Dpto. Fuerza de Trabajo Calificada

MINISTERIO DE ECONOMIA Y PLANIFICACION
DIRECCION DESARROLLO SOCIAL

cc Manuel Puerta La Rosa

Opinión del tutor del trabajo de diploma.

Título: **Diseño e implementación del módulo Plan de Distribución de Graduados de Nivel Superior para el Sistema Unificado de Gestión de Fuerza de Trabajo Calificada.**

Autores: María del C Matías Bordón.

Pedro Salvá Rodriguez.

Resumen

El departamento Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación es el encargado de la gestión de los recursos humanos en el país, para ello desarrolla diversas tareas entre las que se encuentra la elaboración del Plan de Distribución de los Graduados de Nivel Superior, formados en el Curso Regular Diurno. El departamento cuenta con una aplicación que realiza dicho plan, pero la misma ya no cubre las necesidades del cliente, al no contar con la información necesaria para la realización del mismo, por otra parte, este sistema fue programado en Visual FoxPro 5.0 bajo Windows NT, tiene varios errores y poca seguridad, ya que todo el intercambio de información se realiza mediante disquetes de 3½ pulgadas.

Este trabajo tiene como objetivo analizar, diseñar e implementar una aplicación web que realice el Plan de Distribución de Graduados de Nivel Superior de mejor forma, tratando de erradicar las deficiencias de la actual aplicación. Para el desarrollo del trabajo se determinó utilizar por decisiones arquitectónicas de la dirección del proyecto PHP, JavaScript y UML como lenguajes; como sistema gestor de base de datos: PostgreSQL 8.3.5; como metodología de desarrollo: RUP; como servidor web: Apache 2.0; como herramienta de modelado: Visual Paradigm 6.1; como marcos de trabajo: Ext JS 2.2 para la vista, Zend Framework 1.5 para la lógica de negocio, Doctrine 1.011 para el acceso a datos y Sauxe para la gestión de aplicaciones y como entorno de desarrollo integrado NetBeans 6.8.

Palabras claves: Plan de Distribución de los Graduados de Nivel Superior y aplicación.

Índice

Índice de figuras

Índice de tablas

Introducción

Hoy en día la informática, como alternativa obligada, en un mundo tecnológico, se extiende a todos los sectores de la economía y la vida social. Son muchas las instituciones que han apostado a ella para minimizar costos en sus procesos, brindar un mejor servicio o mejorar su desempeño. En Cuba, no es menor esta tendencia a las tecnologías de la informática y las comunicaciones, cuya política es informatizar, en primer lugar, la sociedad cubana y brindarles a sus ciudadanos un mayor nivel de vida.

El departamento Fuerza de Trabajo Calificada (FTC) del Ministerio de Economía y Planificación (MEP), es el encargado de la gestión de los recursos humanos en el país. Para ello realiza diferentes planes, este trabajo se centrará en la elaboración del Plan de Distribución de Graduados de Nivel Superior, formados en el Curso Regular Diurno, el cual constituye una tarea muy importante para el país, pues permite la ubicación laboral de los graduados, de acuerdo con su lugar de residencia, de esta forma, se garantiza que todas las regiones del país cuenten con los recursos laborales necesarios de acuerdo con las necesidades socioeconómicas.

El departamento FTC cuenta con una aplicación para la elaboración del Plan de Distribución de Graduados de Nivel Superior. Este sistema ya no cubre las necesidades del cliente, pues para su desarrollo es necesario tener en cuenta la disponibilidad de graduados de Orden 18, cadetes del Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR) y cadetes del Ministerio del Interior (MININT), por otra parte, este sistema fue programado en Visual FoxPro 5.0 bajo Windows NT, tiene varios errores y su operabilidad se dificulta bastante, tiene poca seguridad, ya que todo el intercambio de información se realiza mediante disquetes de 3½ pulgadas, no se tiene control de las personas que acceden al sistema, además tiene una dificultad de orden crítico y es que no cuenta con ninguna documentación consistente que respalde su desarrollo.

Al realizar el Plan de Distribución de Graduados del Nivel Superior con todas estas dificultades se ve comprometida la calidad del mismo, por lo que se plantea la siguiente **situación problémica**: difícil operabilidad, falta de información necesaria, poca seguridad, poca accesibilidad y errores en el sistema actual.

Teniendo en cuenta lo anteriormente expuesto, el **problema científico** queda formulado de la siguiente manera: ¿Cómo mejorar el Plan de Distribución de Graduados de Nivel Superior utilizado por el departamento Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación?

Por tanto, el **objeto de estudio** lo constituyen los procesos de gestión del departamento de Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación. Todo ello vinculado a las herramientas y metodologías para el diseño de sistemas informáticos, y específicamente, el **campo de acción** de este trabajo será el Plan de Distribución de Graduados de Nivel Superior.

El **objetivo general** que se persigue es desarrollar una aplicación web que permita gestionar el Plan de Distribución de Graduados de Nivel Superior.

Como **objetivos específicos:**

- Realizar el marco teórico de la investigación del proceso de distribución de graduados de nivel superior.
- Analizar y diseñar el módulo Plan de Distribución de Graduados de Nivel Superior.
- Implementar el módulo Plan de Distribución de Graduados de Nivel Superior.
- Realizar pruebas al módulo Plan de Distribución de Graduados del Nivel Superior.

Para lograr el cumplimiento de los objetivos se proponen las siguientes **tareas**:

- Realizar un estudio del estado del arte de los sistemas de distribución.
- Realizar entrevistas a los clientes con el objetivo de caracterizar el proceso de distribución de graduados del nivel superior.
- Realizar un estudio detallado acerca de todo el mecanismo de trabajo del departamento Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación.
- Realizar un estudio de la arquitectura propuesta por el proyecto.
- Definir los procesos vinculados al negocio.
- Definir los requerimientos del módulo Plan de Distribución de Graduados del Nivel Superior.
- Realizar la planificación y estimación del módulo Plan de Distribución de Graduados del Nivel Superior.
- Realizar el análisis del módulo Plan de Distribución de Graduados del Nivel Superior.
- Realizar diseño del módulo Plan de Distribución de Graduados del Nivel Superior, ajustándose a la arquitectura propuesta.

- Definir los estándares de codificación.
- Realizar la implementación del módulo Plan de Distribución de Graduados del Nivel Superior.
- Realizar los casos de pruebas del módulo.
- Realizar las pruebas y evaluar los resultados.

Para guiar la investigación se plantea la siguiente **hipótesis**: con la realización del análisis, diseño e implementación de una aplicación web que realice el Plan de Distribución de Graduados de Nivel Superior, se mejorará el proceso de gestión de información, en cuanto a seguridad, operabilidad y eficiencia.

- **Variable independiente:** módulo Plan de Distribución de Graduados de Nivel Superior.
- **Variable dependiente:** mejorar del proceso de gestión de información.

Para ver la operalización de las variables ir al Anexo #1.

Para llevar a cabo las tareas de la investigación se emplearon métodos **teóricos y empíricos** de la investigación científica.

Los **métodos teóricos** utilizados para cumplir con las tareas a desarrollar son:

Histórico-Lógico: Se realizó un estudio del sistema actual para definir los principales errores y deficiencias, en vista de perfeccionar las vulnerabilidades.

Modelación: El mismo se pone en práctica en el trabajo, al realizar el análisis de la realidad, mediante diversos modelos y diagramas que ayudan a comprender mucho más el objeto en su totalidad.

Analítico-Sintético: Posibilitó el análisis del proceso de distribución para determinar con exactitud cómo funciona. Como resultado, se tomaron todas las características principales para lograr modelar un módulo que logre una integración eficaz y una armonía dentro de los procesos que rigen su comportamiento.

Los **métodos empíricos** utilizados para obtener información sobre el objeto de estudio son:

Observación: este método se usa porque permite investigar los procesos externamente sin tener que llegar a la esencia de los mismos, lo que ayudó al planteamiento del problema científico, además de permitir conocer bien el proceso, delimitando el objeto de estudio. Esto ayudó a tener un conocimiento más detallado de lo que se quiere, lo que hace falta hacer y cómo hacerlo.

Entrevista: Se realizaron entrevistas a cada uno de los integrantes del departamento Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación, con el objetivo de conocer las principales fallas del sistema actual así como las nuevas funcionalidades que requieren.

Estudio documental: Se realizó un estudio a partir de los documentos entregados por el cliente y otros que fueron necesarios para estimular la autosuperación en temas de interés para el desarrollo de la aplicación.

Como **posible resultado** se obtendrá:

- La implementación de una aplicación web que realice el Plan de Distribución de Graduados del Nivel Superior, capaz de gestionar la información de forma rápida y eficiente para el departamento Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación.
- La realización de toda la documentación asociada a dicha aplicación, incluyendo el manual de usuario.

El presente trabajo está estructurado en:

Capítulo 1: Fundamentación teórica.

Este capítulo es el respaldo teórico de los temas tratados en el informe, necesarios para el entendimiento correcto de la solución planteada. Se describen los conceptos fundamentales asociados al dominio del problema y el objeto de estudio, haciéndose un análisis de la situación actual. Se presenta la fundamentación de las tecnologías utilizadas para el diseño del módulo y las propuestas para su implementación y desarrollo. Se abordan los principales problemas que motivan la realización de este trabajo.

Capítulo 2: Características del sistema.

Se presenta el modelo de dominio, los requerimientos funcionales y no funcionales con los que debe cumplir el sistema propuesto, así como sus actores y diagrama de casos de uso, con un resumen de la descripción de cada uno de ellos.

Capítulo 3: Análisis y diseño.

Se modelan, los diagramas de clases del análisis y del diseño. Se hace referencia a los patrones utilizados para la elaboración de la propuesta de diseño. Se muestra el modelo lógico y físico de datos de la base de datos y por último se realizará el diagrama de despliegue.

Capítulo 4: Implementación y pruebas.

Se realiza la implementación del módulo Plan de Distribución de Graduados del Nivel Superior, los diagramas de componentes, se hace referencia a los métodos utilizados para el tratamiento de errores, se tratan los aspectos de seguridad, se citan los estándares utilizados para la implementación en cada una de las capas y finalmente se realizan las pruebas correspondientes.

Capítulo 5: Análisis de la viabilidad del sistema.

Se realiza la estimación del esfuerzo necesario para la confección del módulo a partir del método de estimación por puntos de casos de uso. Se realizará también el análisis de costo y se plantean los beneficios tangibles e intangibles.

Capítulo 1: Fundamentación teórica.

1.1. Introducción

El proceso de selección de las condiciones, ambiente de trabajo y la forma en que se va a elaborar un software es fundamental para el progreso adecuado de la confección del mismo, pues puede influir tanto en el tiempo como en la calidad del trabajo.

En este capítulo se realiza un análisis de los sistemas de distribución en el mundo, en Cuba y en la Universidad de las Ciencias Informáticas (UCI), centrándose en sus principales procesos, de los cuales se tomaron las características que son compatibles con el sistema a desarrollar. También se realiza un análisis de las herramientas y metodologías que se usan para el desarrollo del sistema, mencionando sus características y ventajas.

1.2. Conceptos asociados al dominio del problema.

Vector: se refiere al resultado de aplicar diversas acciones sobre una variable, la cual se incrementa o decrece.

Demanda: se refiere a la cantidad de estudiantes graduados de nivel superior del curso regular diurno, que demanda el país de una determinada carrera, por empresas, provincia y municipio de residencia.

Asignación: se refiere al algoritmo de distribución para cada una de las provincias, cada carrera y cada municipio.

Disponibilidad de graduados del curso regular diurno: se refiere a la cantidad de estudiantes graduados por carreras, atendiendo a la provincia y a su municipio de residencia.

Disponibilidad de graduados de Orden 18: se refiere a la cantidad de estudiantes graduados que pertenecen a la Orden 18 por carreras, atendiendo a la provincia y a su municipio de residencia.

Disponibilidad de graduados cadetes del MINFAR: se refiere a la cantidad de estudiantes graduados que pertenecen al MINFAR y estudiaron en escuelas civiles por carreras, atendiendo a la provincia y a su municipio de residencia.

Disponibilidad de graduados cadetes del MININT: se refiere a la cantidad de estudiantes graduados que pertenecen al MININT y estudiaron en escuelas civiles por carreras, atendiendo a la provincia y a su municipio de residencia.

1.3. Estado del arte

En este epígrafe se realizará un estudio de los sistemas de distribución que existen en el mundo, en Cuba y en la UCI con el objetivo de caracterizar el proceso.

1.3.1. Sistemas de distribución en el mundo.

Gdarim: sistema automatizado para la asignación de aulas y la distribución de espacios.

Es un sistema para la asignación de aulas el cual propone usar algoritmos genéticos para el desarrollo de la asignación, fue desarrollado en Java por su portabilidad y su característica de código abierto.

El sistema consta de tres módulos:

1. Información: contiene y administra la información del problema y sus características.
2. Motor AG: corazón del algoritmo genético y la inteligencia de la solución.
3. Asignación: registra y administra las posibles asignaciones de recursos que genera el algoritmo.

El proceso se realiza de la siguiente forma:
Se carga la información requerida en varios formularios. La misma es almacenada en archivos que luego son accedidos y “entendidos” por la aplicación. La información ingresada es codificada en genomas, que son evaluados, mutados y cruzados de acuerdo con las operaciones de AG definidas para el problema y a funciones de aptitud que serán las encargadas de determinar que alternativa resuelve mejor el problema.

El proceso empieza definiendo el tipo de problema, Gdarim que consta de 21 variables binarias y una cantidad de restricciones que, en principio, dependen de los formularios y otras restricciones implícitas (por ejemplo, que las materias no se superpongan). Con esto, se invoca el motor MOEA. Se crea una población inicial que está formada por un conjunto finito de individuos para un día determinado. Cada uno es una asignación completa para todas las aulas ingresadas y los recursos correspondientes. En la definición del gen se especifica qué parámetros se deben evaluar para la representación del problema. Algunos de los parámetros son: número de aula, materia, profesor, turno, etc. Una vez

definido el gen, se genera la población inicial con n0 individuos. Se eliminan las asignaciones incorrectas y se eligen los mejores individuos, resultando una población de trabajo de n1 individuos. (Pablo Cababie, 2008)

Después de haber realizado la investigación acerca de este sistema, se llegó a la conclusión de que el mismo no se ajusta a la distribución que se necesita hacer para del departamento Fuerza de Trabajo Calificado. Pues este sistema no gestiona, disponibilidad ni demanda ya que realiza una distribución aleatoria donde ninguna de las restrinciones se asemeja a las utilizadas en la aplicación a desarrollar.

OPTIHPER (Asignación Optimizada de Horarios al Personal)

OPTIHPER es un sistema software para la asignación automática de horarios y tareas al personal de una empresa, teniendo en cuenta el conjunto de tareas a realizar, el personal disponible y su calificación, restricciones y preferencias existentes.

OPTIHPER es un sistema altamente flexible, que se adapta a diversas tipologías de tareas, carga de trabajo requerida, restricciones, etc., así como a las diferentes capacidades, turnos, disponibilidad y preferencias del personal y de la organización.

Características del sistema:

- Realiza una asignación optimizada de las tareas, satisfaciendo las restricciones y criterios existentes en su asignación/ejecución, facilitando la ejecución eficiente de las mismas. Puede realizar también la asignación optimizada y equilibrada de turnos y horarios.
- Permite una asignación optimizada del personal a la carga de tareas requeridas, ahorrando consumo de recursos y optimizando las capacidades disponibles.
- Permite satisfacer las preferencias del personal, respecto a horarios, turnos, capacidades, calificación, etc.
- Permite determinar la configuración óptima de la plantilla en cuanto al número de trabajadores de cada tipo y habilidades.
- Permite modificar y validar interactivamente una planificación previa ante incidencias o cambios posteriores (reactividad en línea).
- Permite obtener respuestas muy rápidamente.
- Permite compartir la actividad de los trabajadores entre distintos centros de trabajo.

- Permite aplicar criterios estándar de asignación de tareas/horarios. (Valencia)

Después de ver las características de este sistema se decide no utilizar su método de asignación, ya que las restricciones que posee el mismo no se ajusta a las restricciones necesarias para el sistema a desarrollar. Una de las dificultades que presenta es que hay que introducirle datos específicos, para que a partir de ellos se realice la distribución, como son: número de personas involucradas, número de horarios laborales, etc.

1.3.2. Sistemas de distribución en Cuba.

DRSOFT: Un soporte computacional para el diseño de rutas de distribución.

DRSOFT es un software destinado a diseñar rutas para la distribución utilizando de forma simultánea, varios criterios como son:

1. Se ordenan los clientes no sólo por la distancia salvada sino además por la prioridad.
2. Antes de pasar a formar las rutas, se analiza si existe algún cliente que su demanda total exceda la capacidad (t o m3) de los vehículos disponibles; de existir, será necesario la realización de recorridos unitarios para esos clientes, que se forman siguiendo el orden determinado por la prioridad, asignándole el vehículo disponible de mayor capacidad, es decir, el que menor número de viajes represente y esté disponible más temprano. Una vez asignado el vehículo se le irán incorporando productos solicitados por ese cliente, comprobando las restricciones del problema y sólo se cargará la cantidad que quepa en el mismo, dejando los demás pendientes; se comprobará si se ha satisfecho o no la demanda total del cliente; en caso afirmativo se elimina de la lista de posibles clientes; si no se comprueba si son necesarios más recorridos unitarios; de concluir los recorridos unitarios necesarios para ese cliente se pasará a analizar si existen otros clientes con la misma prioridad que también necesiten esos recorridos.
3. Una vez formados todos los recorridos unitarios y actualizadas las demandas de los clientes, se conforman los núcleos generadores con los clientes que mayor distancia ahorren y mayor prioridad tengan, utilizando el vehículo que esté disponible más rápido; de no cumplirse algunas de las restricciones del problema, chequear si se puede formar la ruta entre un cliente de mayor prioridad y otro que tenga la prioridad siguiente; una vez formado el núcleo generador tratar de añadirle clientes a la ruta en la misma forma del método clásico.
4. Analizados ya todos los recorridos (unitarios o no) para los clientes de mayor prioridad se pasan a la siguiente, repitiéndose el procedimiento.

Este sistema está destinado a almacenes o empresas que distribuyan productos a un número importante de clientes, donde los costos implicados en este proceso tienen un impacto significativo en los beneficios, pero donde el estado de satisfacción del cliente juega un importante papel en el posicionamiento de la empresa y en la fidelización de sus clientes, aspectos de gran importancia en estos momentos en la gestión empresarial.

Los algoritmos implementados para las operaciones de solución del sistema están basados en técnicas heurísticas, para la determinación de las alternativas de ruteo y en las técnicas de análisis de la decisión multiatributo.

El sistema propuesto incluye los siguientes módulos:

Formación de alternativas: elabora un conjunto de alternativas de enrutamiento tomando como base: los pedidos de los clientes, los productos con que cuenta el centro, la disponibilidad de vehículos, la prioridad de los clientes, la fecha de entrega del pedido, la distancia que separa al centro de los clientes y a ellos entre sí y las restricciones del problema, capacidad y tiempo.

Módulo preferencial: es el módulo que intenta capturar la subjetividad del decisor en la comparación de las alternativas. De la precisión de este módulo depende la calidad de la decisión, en el mismo se determinarán los umbrales de preferencia, indiferencia y veto, así como la importancia de cada uno de los criterios que intervendrán en la decisión.

Selección de alternativas: selecciona la mejor alternativa dentro del conjunto de alternativas generadas no dominadas y de acuerdo con las preferencias del decisor. (Sánchez, 2004)

Este sistema posee una apariencia muy compleja, fue implementado en el 2004, por lo que las tecnologías empleadas en él ya están demasiado atrasadas para el desarrollo informático actual. Además, ni la distribución que el sistema realiza, ni las condiciones sobre las cuales se realiza la distribución se ajustan a las que requiere el Plan de Distribución de Graduados de Nivel Superior.

SENTAI utilizado en la Corporación CIMEX (Cuba).

El SENTAI es un sistema integrado de gestión empresarial implementado sobre el sistema de base de datos “Postgre” y soportado en el sistema operativo “UNIX”. Este sistema, por sus características fundamentales, es muy flexible y adaptable a las necesidades específicas de los usuarios que lo explotan, pues es completamente configurable y por medio de la parametrización puede lograrse

delimitación de los accesos en una división oportuna de funciones y tareas, con alto nivel de seguridad en las operaciones y transacciones que se ejecutan por los usuarios. Se distingue, además, por trabajar para múltiples usuarios en forma de multiprocesamiento.

Los módulos de Órdenes de compras, Órdenes de ventas, Inventarios, Administración de la distribución y Administración de almacenes, están orientados y relacionados directamente a las actividades de la gestión comercial de compras y ventas de mercancías, tanto mayoristas como minoristas o detallistas, así como el almacenamiento, manipulación y entrega de las mismas, en aquellas empresas, compañías y otras entidades, que las desarrollan.

El sistema funciona de la siguiente manera, el cliente solicita la mercancía mediante los códigos de los productos en un sistema, una vez realizado el pedido el sistema archiva una lista según los pedidos de los clientes. De acuerdo con el orden del pedido, la cantidad de productos disponibles, y al tipo de zona que pertenece (tipo de zona provincia o municipio), se realiza la distribución de los mismos. (CIMEX, 2010)

Este sistema fue implementado con herramientas de software libre, lo que es una característica que se persigue para la implementación del Sistema Plan de Distribución de Graduados del Nivel Superior, pero la distribución que realiza es según sus características y necesidades, las cuales se corresponden a la compra y venta de productos, por lo que no se ajusta a las necesidades de la aplicación a desarrollar.

1.3.3. Sistemas de distribución en la UCI.

Subsistema de gestión de pregrado que pertenece al proyecto Gestión Universitaria.

Está implementado con herramientas de software libre. Tiene como objetivo fundamental, permitir la gestión automatizada de los elementos que intervienen en la labor académica de un centro de estudio y enfrentar los cambios de forma natural. Está dividido en cinco módulos, los cuales son: Diseño de carrera, Personal y secretaría, Planificación docente, Registro y control docente y por último Tesis y títulos. Estos módulos abarcan todos los procesos involucrados con la labor docente e interactúan entre sí, para llevar a cabo cada una de las tareas que automatiza el sistema. Esta investigación se centrará en el módulo de Personal y secretaría, el cual tiene entre uno de sus objetivos la distribución de los estudiantes de nuevo ingreso por facultades y grupos de estudio. Para desarrollar la distribución el sistema debe contar con los estudiantes que ingresan al centro, la cantidad de facultades en las que

se desean ubicar, las prioridades que tengan, la cantidad de grupos por facultad y por último la cantidad de estudiantes por grupo. Una vez conformada toda esta información se ejecuta el algoritmo que consta de 4 pasos, los cuales son:

- **Obtener los estudiantes ordenados por criterios (id_expediente).**

Se van ordenando los estudiantes por los criterios seleccionados, estos criterios tienen una prioridad definida por el usuario, y además se ordenan alfabéticamente.

- **Calcular cantidad de grupo**

Cantidad de grupos = cantidad total de estudiantes / cantidad de estudiantes por grupo.

Luego de este cálculo se obtiene la cantidad de grupos que van a existir.

- **Repartir los estudiantes por grupo** (esta repartición es uno a uno hasta que todos los estudiantes tengan un grupo).
- **Repartir los grupos con estudiantes por facultad.** (Estilo baraja). Aquí se debe tener en cuenta la cantidad de grupos que se definió para cada facultad, en caso de que no se haya definido la cantidad de grupos en algunas facultades se va realizando una distribución equitativa hasta que se terminen los grupos.

Después de realizado el algoritmo se obtiene una propuesta de ubicación, quedando conformados los grupos administrativos. En caso de no quedar conforme con la propuesta de ubicación se realiza nuevamente la distribución. (UCI L. L., 2010)

Esta aplicación cumple con las políticas de software libre que es uno de los objetivos de la aplicación a desarrollar, pero la distribución que se realiza es completamente distinta y las condiciones son diferentes, además en esta aplicación es necesario escoger los parámetros sobre los cuales se realiza la distribución, por lo cual se decide no utilizarla.

Sistema Integrado de Transportación

El Sistema Integrado de Transportación es una herramienta desarrollada en software libre. Su principal función es gestionar y agilizar las reservaciones masivas de la transportación en la Universidad de las Ciencias Informáticas. Este sistema cuenta con 3 módulos los cuales son: Reportes, Reservaciones y

Gestión de la información. La investigación se centrará en el módulo de Reservaciones el cual se explicará a continuación.

El módulo de Reservaciones está vinculado a la distribución de los estudiantes y profesores de la Universidad de las Ciencias Informáticas en el transporte que se brinda para las salidas masivas que se efectúan en la misma.

Cuando se necesita gestionar un pase masivo la dirección del transporte de la UCI solicita al departamento de recursos humanos los datos de las personas que van a viajar, conformando con este listado la demanda. A partir de este listado la dirección de transporte de la UCI tramita la compra de los pasajes con la Agencia de Viajes que corresponda, dicha agencia envía un listado de asientos que son los que corresponden con los comprados, conformando este listado la disponibilidad. Una vez obtenido todos los datos de la transportación se procede a la distribución es decir:

Primero se verifica para dónde va la persona que realizó la reservación, si el ómnibus llega hasta la provincia cabecera o su destino es hasta los municipios y en dependencia de este dato y la capacidad del ómnibus se comienzan a llenar los siguientes, cuando este se complete, se procede a llenar el siguiente ómnibus con el mismo destino hasta que todas las reservaciones hayan sido cubiertas. (UCI Y. B., 2007)

Este sistema realiza una distribución para la transportación de estudiantes y profesores de la UCI, pero posee características que no se ajustan para realizar la distribución de graduados de nivel superior en el país, por ejemplo la disponibilidad se obtiene según la demanda de estudiantes y profesores que deseen viajar, creando una dependencia entre estos dos valores, no siendo así en la aplicación a desarrollar donde la demanda y la disponibilidad son dos factores separados que no dependen uno del otro.

1.4. Necesidad de implementar una nueva aplicación que realice el Plan de Distribución de Graduados de Nivel Superior.

Los sistemas de distribución han surgido como alternativa para mejorar la eficiencia, rapidez, claridad y calidad en los procesos que gestionan gran cúmulo de información. Tienen como tarea fundamental distribuir la información siguiendo determinadas reglas o restricciones colocándola automáticamente en el lugar adecuado y con las cantidades precisas.

El departamento Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación es el encargado de desarrollar los procesos de gestión de la formación y el empleo de la fuerza de trabajo calificada en el país. Dentro del proceso de empleo entra el de distribución de graduados de nivel superior, este proceso depende de dos factores esenciales:

- La disponibilidad de los graduados de nivel superior, formados en el curso regular diurno, esta información permitirá la elaboración del plan anual de ubicación laboral por parte del departamento Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación.
- La demanda de los organismos que sería la cantidad de graduados de este nivel necesarios para completar la fuerza de trabajo calificada que posee cada empresa, acorde a las perspectivas de desarrollo del organismo en los territorios**.**

Teniendo en cuenta los dos aspectos anteriores se realiza el Plan de Distribución de Graduados de Nivel Superior, formados en el curso regular diurno para cada una de las provincias, cada carrera y cada municipio, teniendo en cuenta los siguientes pasos:

La disponibilidad se compara con la demanda:

- Si la disponibilidad es mayor se realiza la asignación restándole a la disponibilidad la demanda y quedarán disponibles sin asignar.
- Si la disponibilidad es menor que la demanda se realiza la asignación de la forma más equivalente posible, quedando demandas sin cubrir.
- Si la disponibilidad es igual a la demanda se realiza la asignación y no quedan disponibles, ni demandas por cubrir.

Actualmente se cuenta con un sistema que realiza los procesos de gestión de la información del Plan Anual de Distribución de los Graduados de Nivel Superior. Este fue programado en Visual FoxPro 5.0 bajo Windows NT y es una aplicación desktop, además le faltan valores a considerar como la disponibilidad de graduados de Orden 18, cadetes del MINFAR y cadetes del MININT, presenta poca seguridad, poca accesibilidad, errores en el sistema que dificultan el desarrollo de este plan y no cuenta con ninguna documentación que sustente su desarrollo.

Con el desarrollo actual de las tecnologías en el país y con el apoyo de la Universidad de las Ciencias Informáticas se decidió mejorar la realización del Plan de Distribución de Graduados de Nivel Superior, informatizando de forma más eficiente sus procesos, mediante el empleo de sistemas automatizados.

Con el desarrollo del nuevo sistema de distribución se asegurará la realización de forma más eficiente y segura del Plan de Distribución de Graduados de Nivel Superior, formados en el Curso Regular Diurno, además de optimizar los recursos necesarios para el desarrollo del mismo.

Después de haber realizado un estudio sobre los sistemas de distribución en el mundo, en Cuba y en la UCI, analizando de cada uno de ellos sus principales características y centrándose en un análisis detallado del proceso de distribución, se llegó a la conclusión de que no existe una aplicación que realice el Plan de Distribución de Graduados de Nivel Superior de forma eficaz, por lo que se decide implementar un nuevo sistema donde se desarrollen todas las funcionalidades y requerimientos que la aplicación a desarrollar requiere.

1.5. Software libre

El software libre es un movimiento tecnológico que ha revolucionado la sociedad. Presenta características especiales que han permitido la experimentación de nuevas formas de desarrollo y mantenimiento de programas, nuevos modelos económicos y nuevas normas legales. “Es un asunto de libertad, no de precio”. Para entender el concepto, se debe pensar en “libre” como en “libertad de expresión”. “Software libre es aquel que puede ser distribuido, modificado, copiado y usado; por lo tanto debe venir acompañado del código fuente para hacer efectivas las libertades que lo caracterizan. De modo más preciso, se refiere a cuatro clases de libertad para los usuarios del software: la libertad de usar el programa, con cualquier propósito (libertad 0), la libertad de estudiar cómo funciona el programa y adaptarlo a sus necesidades (libertad 1), la libertad de distribuir copias, con lo que puede ayudar a sus amigos (libertad 2) y la libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie (libertad 3)”. (GNU, 2010)

1.5.1. Tendencias actuales del software libre.

“Los expertos afirman que el software libre es hoy en día un recurso potente y maduro, capaz de dar soluciones prácticas y democráticas a las necesidades de la sociedad moderna. Bajo el lema “Innovación y Libertad”, que alude a la aplicación práctica de los sistemas de fuente abierta para conseguir mayores niveles de desarrollo y bienestar. Las principales personalidades en la materia coinciden en señalar que el software libre es clave en la educación digital y su universalización”. (Libre, 2006)

Cuando se habla de software libre, se debe evitar utilizar expresiones como “regalar” o “gratis”, ya que se puede caer en el error de interpretarlo como una mera cuestión de precio y no de libertad. Evidentemente, el software libre, es la alternativa para los países subdesarrollados. Ya que una vez comprado el software libre, se puede desarrollar a partir de él nuevos software, los cuales pueden venderse sin la necesidad de tener que pagar las licencias como es el caso del software propietario. Además, si durante el desarrollo de una aplicación se hace necesaria una funcionalidad que el software no posee, sin ningún problema el mismo puede ser modificado, incluso hasta redistribuido y vendido sin ningún tipo de restricción.

La migración hacia el software libre evita la dependencia de los proveedores, Cuba y en particular la Universidad de las Ciencias Informáticas, han sustentado la posibilidad de migrar hacia software libre, por las ventajas que representa con respecto a los de tipo propietario. Desde el ámbito político, representa la no utilización de productos informáticos que demanden la autorización de sus propietarios (licencias) para su explotación. Es por esta razón que se decidió utilizar para la realización del nuevo Sistema Plan de Distribución de Graduados de Nivel Superior herramientas de software libre.

1.6. Herramientas, lenguajes y metodologías utilizadas en la propuesta de desarrollo.

Este sistema forma parte de los proyectos de la UCI que se rigen por la arquitectura del ERP CEDRUX. Todas las herramientas, lenguajes y metodologías mencionadas a continuación, están fundamentadas en el marco de trabajo del proyecto al cual pertenece el módulo Plan de Distribución.de Graduados de Nivel Superior.

1.6.1. Aplicación web

Una aplicación web es una interfaz diseñada para cubrir las necesidades de un negocio y gestionar su información (la información puede ser de dominio público o restringido a ciertas personas a través de un nombre de usuario y contraseña) con el objetivo de que cualquier persona pueda consultarla e interactuar con ella a través de la red. Se pueden adaptar a muchas situaciones y su objetivo es mejorar la forma de trabajo y la productividad de una empresa o grupo de personas de una manera sencilla. (Netcommerce, 2009)

Se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una Intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web (HTML, JavaScript, Java, ASP, .NET, PHP, etc.) en la que se confía la ejecución al navegador.

Ventajas de una aplicación web:

- **Conectividad:** se puede acceder a la aplicación desde cualquier punto y a cualquier distancia.
- **Ahorra tiempo**: se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- **No hay problemas de compatibilidad**: basta tener un navegador mínimamente actualizado para poder utilizarlas.
- **No ocupan espacio** en el disco duro.
- **Actualizaciones inmediatas**: como el software lo gestiona el propio desarrollador, cuando el cliente se conecta está usando siempre la última versión que haya lanzado.
- **Consumo de recursos bajo**: dado que toda (o gran parte) de la aplicación no se encuentra en el ordenador del cliente, muchas de las tareas que realiza el software no consumen recursos porque se realizan desde otro ordenador.
- **Multiplataforma**: se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- **Portables**: es independiente del ordenador donde se utilice (una computadora de sobremesa, una portátil, un móvil) porque se accede a través de una página web (sólo es necesario disponer de acceso a Internet).
- **Los virus no dañan** los datos porque estos están guardados en el servidor de la aplicación.

1.6.2. Modelo Cliente-Servidor

La utilización y aplicación del modelo Cliente-Servidor está fuertemente ligada al desarrollo de sistemas web. Este modelo permite al usuario en una máquina, llamado “cliente”, requerir algún tipo de servicio de una máquina a la que está unida, llamado “servidor”, mediante una Red de Área Local (LAN) o una Red de Área Amplia (WAN).

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. Para ver imagen del modelo Cliente- Servidor ir al Anexo #2.

En este modelo cuando un cliente desea un servicio, que proporciona cierto servidor, le envía un mensaje solicitando ese servicio; una petición. Los procesos clientes y servidores han de seguir un protocolo de comunicación que defina cómo se codifican las peticiones y cómo se sincronizan entre sí los procesos. Para ver la solicitud de servicios utilizando modelo Cliente-Servidor ver Anexo #3.

Ventajas del modelo Cliente-Servidor:

- **Centralización del control**: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos.
- **Escalabilidad**: se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes o servidores).
- **Fácil mantenimiento**: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación. (Cliente/Servidor)

1.6.3. Servidor web: Apache.

Los servidores web son una herramienta indispensable a la hora de la publicación y explotación de un sistema que esté desarrollado para su uso en la web. Estos servidores contienen la lógica para la interpretación de las peticiones de los usuarios a través de los protocolos especificados, que hace posible la interacción dinámica entre el cliente y sus acciones sobre el sistema en que esté trabajando.

Apache se basó originalmente en codificación e ideas basadas en el servidor HTTP. Actualmente por su flexibilidad, rapidez, eficiencia, su constante actualización y adaptación a los nuevos protocolos (HTTP 1.1) es uno de los mejores servidores web utilizados en Internet, según Netcraft Survey, empresa dedicada a la realización de encuestas a nivel global y estudios sobre el tráfico en Internet. Para ver análisis realizado por Netcraft de la explotación de servidores web ir a Anexo #4.

Fue hecho para proveer un alto grado de calidad y fortaleza para las implementaciones que utilizan el protocolo HTTP, ha sido desde su salida al mercado, uno de los servidores de mayor popularidad, considerado el proyecto punta del movimiento de software libre.

Características más representativas de Apache:

- Multiplataforma, funcionando tanto en Windows como en Linux o Unix.
- Es un servidor web conforme al protocolo HTTP 1.1.
- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Tecnología de código fuente abierto.
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor. (Ciberaula, 2010)

1.6.4. Lenguajes de programación para la web

En la actualidad los lenguajes de programación para la web se clasifican en dos grupos teniendo en cuenta dónde se implementan respecto al modelo Cliente-Servidor, nombrándose lenguajes del lado del cliente y lenguajes del lado del servidor.

Lenguajes utilizados del lado del cliente.

- **JavaScript:** Lenguaje de programación que ha permitido gran desarrollo en la animación de las páginas web. La aparición de JavaScript ha resuelto de manera fácil y elegante la mayoría de los problemas con que se enfrenta el diseñador de páginas web referidos a la programación, debido fundamentalmente, a que sus requerimientos son relativamente sencillos y lo que quizás sea más importante, es un lenguaje cuyos códigos se resuelven en el navegador del cliente, sin tener que ir y venir del cliente al servidor actualizando la información. JavaScript es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros. (Valdés, 2007)

Lenguajes utilizados del lado del servidor

En este grupo se encuentran entre los más utilizados: PERL, ASP, JSP y PHP. Estos lenguajes permiten desarrollar lógica del negocio dentro del servidor, acceso a las bases de datos y el procesamiento de la información.

PHP: Acrónimo recursivo Pre-procesador de Hipertexto de su significado en inglés *PHP Hypertext Pre-Processor*, constituye un lenguaje *script* de alto nivel interpretado del lado del servidor. Es un lenguaje

de programación (originario del nombre *PHP Tools* o *Personal Home Page Tools*) que sirve principalmente para proporcionar características dinámicas a una página web. Al ser ejecutado del lado del servidor, PHP permite acceder a los recursos internos del mismo o a otros externos, como por ejemplo a una base de datos, siendo el resultado normalmente una página HTML con los datos y acciones enviadas al cliente.

Algunas de las características básicas del lenguaje son:

- "Gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, que garantiza que los fallos de funcionamiento se encuentren y reparen rápidamente".
- Puede interactuar con muchos motores de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otros.
- Integración con varias bibliotecas externas que permite, entre otras funcionalidades, manejo de gráficos y generar documentos en PDF (documentos de *Acrobat Reader*).
- Permite técnicas de programación orientada a objetos. (S., 2007)

1.6.5. Metodología de desarrollo de software.

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no se lleva una metodología de por medio, lo que se obtienen son clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Lo más importante antes de elegir la metodología que se usará para la implementación de un software, es determinar el alcance que tendrá y luego de ahí ver cuál es la que más se acomoda a la aplicación. Las metodologías fueron diseñadas bajo un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para lograr la calidad, que es el principal objetivo estratégico en las organizaciones.

El Proceso Unificado de Desarrollo (RUP):

Es una metodología pesada, que cuenta con un marco de desarrollo de software iterativo e incremental, está compuesto por cuatro fases denominadas Inicio la cual tiene como objetivo determinar la visión del proyecto, Elaboración en esta etapa el objetivo es determinar la arquitectura óptima, Construcción en esta etapa el objetivo es obtener la capacidad operacional inicial y Transición la cual tiene como objetivo llegar a obtener el *release* del proyecto. Está basado en componentes e

interfaces bien definidas y utiliza el Lenguaje Unificado de Modelado (UML) para visualizar, especificar, construir y documentar un sistema de software. (Sanchez, 2004)

Características de RUP:

- **Dirigido por los casos de uso:** los casos de uso son una forma abstracta de representar lo que los usuarios finales necesitan, por lo cual ellos son los que deben guiar el proceso de desarrollo. Esto se garantiza con la obtención de los diferentes modelos que son el resultado de los flujos de trabajo que propone RUP.
- **Centrado en la arquitectura:** la arquitectura de un proyecto muestra una visión común del sistema completo, visión en la que deben estar de acuerdo tanto el equipo de desarrollo como los clientes. Es por ello que la arquitectura describe los elementos del modelo que son más importantes para la construcción del sistema, así como los cimientos para comprenderlo y desarrollarlo de forma económica.
- **Iterativo e incremental:** RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y las cuales se definen según el nivel de madurez que alcanzan los productos que se van obteniendo con cada actividad ejecutada. La terminación de cada fase ocurre en el hito correspondiente a cada una, donde se evalúa que se hayan cumplido los objetivos de la fase en cuestión. (software1, 2007-2008)

1.6.6. Lenguaje unificado de modelado

UML es un grupo de especificaciones de notación orientadas a objeto, las cuales están compuestas por distintos diagramas, que representan las diferentes etapas del desarrollo de un proyecto de software. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad y está respaldado por el Grupo de Gestión de Objetos (OMG de su significado en inglés *Object Management Group*). Ofrece un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Este lenguaje indica cómo crear y leer los modelos, pero no dice cómo desarrollarlos, esto último es el objetivo de las metodologías de desarrollo. (Grady Booch)

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- Visualizar: permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.

- Especificar: permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: a partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión. (Ivar Jacobson)

1.6.7. Herramienta de modelado de software

Visual Paradigm.

Visual Paradigm es una herramienta que utiliza el lenguaje de modelado estándar UML, diseñada para desarrollar software que utiliza la programación orientada a objetos. Una de las características más atractivas de Visual Paradigm es que permite generar a partir del modelado código PHP 5, realizar ingeniería tanto directa como inversa y generar la documentación del proyecto automáticamente en varios formatos como web o PDF y facilita a través de la herramienta Team Framework el trabajo colaborativo. Es una herramienta multiplataforma que se puede utilizar tanto en Linux como en Windows, por lo que cumple con las políticas de migración a software libre. Tiene una interfaz muy intuitiva y es de fácil aprendizaje para los desarrolladores. Permite la generación automática de diagramas a partir de las descripciones de los casos de uso, por ejemplo diagramas de secuencia, permitiendo la agilidad en el trabajo del analista. Presenta licencia gratuita y comercial.

1.7. Sistema para la gestión de base de datos (SGBD).

Los sistemas que gestionan datos, así como las transacciones entre estos, constituyen una potente herramienta, la cual está estrechamente vinculada a sistemas de software robustos que necesiten de la persistencia, localización y explotación de un volumen amplio de datos.

PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional basado en el proyecto Ingres, de la universidad de Berkeley, California. Pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, PostgreSQL incluye características de la orientación a objetos como pueden ser: la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, además de otras específicas del gestor, como son: un mejor soporte para *sub-selects*, *triggers*, vistas y procedimientos almacenados. (PostgreSQL)

Entre sus características más específicas se encuentran:

- Implementación del estándar SQL92/SQL99.
- Soporte para distintos tipos de datos, como son: datos de tipo fecha, datos sobre redes (MAC, IP), cadenas de bits, etc., así como la creación de tipos de datos propios.
- Al igual que con los tipos de datos, PostgreSQL permite la declaración de funciones propias, así como la definición de dispensadores.
- Soporte en la mayoría de los sistemas operativos más utilizados incluyendo, Linux, varias versiones de UNIX, BeOS y Windows.

1.8. Marcos de trabajo

Los marcos de trabajo presentan una estructura organizada que obliga a los programadores a seguir estándares y a trabajar de manera organizada. El uso de estas aplicaciones ha demostrado que la organización de la programación influye notablemente en la calidad de las aplicaciones. Son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional.

1.8.1. Zend Framework

Zend Framework es un marco de trabajo para construir sitios web modernos, robustos y seguros utilizando PHP. Posee una arquitectura flexible orientada a componentes que usa el patrón Modelo-Vista-Controlador (MVC) desacoplado y es acoplable a varias tecnologías de desarrollo, pues se le pueden adicionar múltiples complementos externos. Además, es código abierto, lo que ha posibilitado que al código fuente se le hayan hecho numerosas pruebas y que posea una extrema simplicidad y productividad. (Framework)

Entre sus principales características se encuentran:

- Permite migrar aplicaciones de forma sencilla.
- Sus componentes se pueden utilizar de forma independiente sin tener que generar dependencias internas.
- Constituye una solución para el acceso a base de datos que balancea el Mapeo Objeto-Relacional (ORM de su significado en inglés Object Relational Mapping) con eficiencia y simplicidad.

- Posee un marco de trabajo que incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar su base de datos, sin tener que escribir ninguna consulta SQL.

1.8.2. Ext JS

Ext JS es una librería JavaScript que permite crear aplicaciones complejas utilizando tecnologías como AJAX, DHTML y DOM, componentes predefinidos, manejador de *layouts*; gracias a esto provee una experiencia consistente sobre cualquier navegador, evitando el tedioso problema de validar, que el código escrito funcione bien en cada uno (Firefox, Safari, etc.). Además, provee al desarrollador de una ventana flotante que permite dibujar bordes con sólo moverla o redimensionarla, haciendo que el movimiento sea fluido, siendo esta una de las ventajas principales.

Dispone de un conjunto de componentes *widgets* para incluir dentro de una aplicación web como: cuadros, áreas de texto, campos para fechas, campos numéricos, combos, r*adiobuttons*, *checkboxs*, editor HTML, árbol de datos, pestañas, barra de herramientas, menús al estilo de Windows, paneles divisibles en secciones, entre otros. También contiene numerosas funcionalidades que permiten añadir interactividad a las páginas HTML, como son los cuadros de diálogos y *quicktips* para mostrar mensajes de validación e información sobre campos individuales.

- Existe un balance entre Cliente – Servidor. La carga de procesamiento se distribuye, permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo.
- Comunicación asíncrona. En este tipo de aplicación el motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente se dé cuenta.
- Eficiencia de la red. El tráfico de red puede disminuir al permitir que la aplicación elija qué información desea transmitir al servidor y viceversa, sin embargo, la aplicación que haga uso de la pre-carga de datos puede que revierta este beneficio por el incremento del tráfico. (Corzo, 2008)

1.8.3. Doctrine

Doctrine es un mapeador de objetos relacionales (ORM) para PHP, es decir, una interfaz capaz de traducir la lógica de los objetos a la lógica relacional lo que permite acceder de forma efectiva a una base de datos desde un contexto orientado a objetos. Brinda la posibilidad de exportar una base de

datos existente a sus clases correspondientes y también a la inversa, es decir, convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos.

1.8.4. Sauxe

El marco de trabajo Sauxe es una solución de software creado en la Universidad de las Ciencias Informáticas (UCI) para desarrollar grandes aplicaciones web de gestión. Fue desarrollado usando las últimas tecnologías para el desarrollo web: PHP, Zend Framework, Ext JS, Doctrine y PostgreSQL como gestor de base de datos. Es una herramienta que permite fácilmente ser configurada para cualquier empresa.

El marco de trabajo Sauxe se ha estructurado de manera tal que facilite la reutilización de los diferentes componentes y que sea de fácil entendimiento para todos los programadores que desarrollen sobre él. (ERP, 2010)

1.9. Generador de Reportes Dinámico Patdsi.

El Generador de Reportes Dinámico es una aplicación web que tiene como objetivo generar reportes de forma rápida, interactiva y con una amplia gama de alternativas para los usuarios. La extensión en su uso puede estandarizar la generación de reportes en diferentes aplicaciones independientemente del sistema gestor de base de datos que utilicen ya sea MySQL, Oracle o PostgreSQL.

La herramienta permite abstraerse a los conocimientos relacionados con los gestores de bases de datos, agilizar la toma de decisiones y generar reportes en varios formatos y con gran variedad de opciones en su diseño, marcando una diferencia entre los reportes tradicionales y los reportes dinámicos. Está compuesto por varias aplicaciones entre las que se encuentran el Visor de reportes, Diseñador de modelos, Diseñador de reportes, Diseñador de consulta y Administrador de reportes. (UCI-Patdsi, 2010)

Principales funcionalidades de cada aplicación:

- Visor de reportes: permite visualizar, exportar e imprimir los reportes existentes, además da la opción de filtrar dichos reportes por los campos que desee el usuario, para ello brinda la posibilidad de escoger los mismos.
- Diseñador de modelos: permite la creación de modelos a partir de un origen de datos dando la posibilidad de seleccionar las rutinas, tablas y vistas a incluir en el modelo.

- Diseñador de reportes: permite diseñar reportes a partir de componentes, fuentes de datos, modelos, vistas y funciones, para ello brinda la posibilidad de seleccionar los mismos.
- Diseñador de consulta: permite seleccionar el modelo, las tablas o vistas que se utilizarán para realizar la consulta, brinda además la posibilidad de relacionar las tablas, agregar condiciones a las consultas y verificar la consulta una vez terminada.
- Administrador de reportes: permite administrar las plantillas, los reportes y los modelos anteriormente creados en las aplicaciones Diseñador de reportes y Diseñador de modelos.

1.10. Entorno de desarrollo integrado

Un entorno de desarrollo integrado (IDE de su significado en inglés *Integrated Development Environment*), es un conjunto de herramientas que ha sido creado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Un IDE provee un marco de trabajo amigable para la mayoría de los lenguajes de programación y en algunos casos puede funcionar como un sistema en tiempo de ejecución en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto.

NetBeans

NetBeans IDE es un reconocido entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris. Está formado por un IDE de código abierto y una plataforma de aplicación que permite a los desarrolladores crear con rapidez aplicaciones web, empresariales, de escritorio y móviles utilizando la plataforma Java, así como JavaFX, PHP, JavaScript y Ajax, Ruby y Ruby on Rails, Groovy and Grails y C/C++. Cuenta con el apoyo de una comunidad de desarrolladores dinámica, ofrece documentación y recursos de formación exhaustivos. (NetBeans)

Entre las características de la plataforma se encuentran:

- Administración de las interfaces de usuario.
- Administración de las configuraciones del usuario.
- Administración del almacenamiento.
- Marco de trabajo basado en asistentes.
- Soporte para PHP.
- Instalación y actualización más simple.
- Características visuales para el desarrollo web.

1.11. Diseño de prototipos de interfaz.

Axure RP Pro es una aplicación ideal para crear prototipos y especificaciones muy precisas para páginas web. Se trata de una herramienta especializada en la tarea, así que cuenta con todo lo que se puede necesitar para crear los prototipos de forma más eficiente.

Axure RP Pro permite componer la página web visualmente, añadiendo, quitando y modificando los elementos con suma facilidad, demostrando su grado de especialización en las anotaciones. En este punto, permite especificar el estado de cada elemento (Propuesto, Aceptado, Incorporado), el beneficio esperado (Crítico, Importante, Útil), el riesgo, la estabilidad, a quién va dirigido y a quién se le asignará la tarea. Otra característica a destacar del Axure RP Pro es que permite un diseño colaborativo. (Tripix.net, 2006)

1.12. Herramienta de gestión de configuración y control de versiones.

1.12.1. Subversion

Subversion es un software de sistema de control de versiones; se le conoce también como SVN por ser ese el nombre de la herramienta de línea de comandos. Una característica importante de Subversion es que todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo.

Subversion puede acceder al repositorio a través de redes. Tiene la capacidad para que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. (Ben Collins-Sussman)

Entre las ventajas de su uso están:

- Se sigue la historia de los archivos y directorios a través de copias y renombrados.
- Las modificaciones (incluyendo cambios a varios archivos) son atómicas.
- La creación de ramas y etiquetas es una operación más eficiente; tiene costo de complejidad constante (O (1)) y no lineal (O (n)).
- Se envían sólo las diferencias en ambas direcciones.
- Maneja eficientemente archivos binarios.
- Cuando se usa integrado a Apache permite utilizar todas las opciones que este servidor provee a la hora de autentificar archivos.

1.12.2. TortoiseSVN

TortoiseSVN es un cliente Subversion, implementado como una extensión al Shell de Windows. Es software libre, liberado bajo la licencia GNU GPL. (TortoiseSVN)

Características:

- Integración con el Shell de Windows.
- Puede ser usado sin un entorno de desarrollo.
- Pequeñas imágenes decoran los íconos de los archivos mostrando qué archivos o directorios necesitan ser enviados al repositorio.
- Disponible en 28 idiomas diferentes.

1.13. Conclusiones

En este capítulo se detallaron las condiciones que rodean el objeto de estudio, mediante los diferentes conceptos y definiciones planteadas. Se realizó un estudio detallado de varios sistemas de distribución en el ámbito nacional e internacional, el cual aportó un conjunto de conocimientos básicos sobre el estado del arte, así como posibles ideas y mejoras a desarrollar en la nueva solución.

Como metodología de desarrollo se utilizará RUP y como herramienta de modelado: Visual Paradigm 6.1, como lenguajes se usará PHP y JavaScript, como SGBD PostgreSQL 8.3.5, como servidor web Apache 2.0, como generador de reportes dinámico Patdsi, como marcos de trabajo Ext JS 2.2, Zend Framework 1.5, Doctrine 1.011, Sauxe; como IDE NetBeans 6.8; para el diseño de los prototipos no funcionales se utilizará Axure. Como herramienta de gestión de configuración y control de versiones del proyecto se utilizará Subversion y como cliente de esta se utilizará TortoiseSVN.

Capítulo 2: Características del sistema

2.1. Introducción

En el presente capítulo se realiza un estudio del negocio en el que se enmarca el módulo Plan de Distribución de Graduados de Nivel Superior, para ello se realiza una modelación del dominio, identificando los conceptos principales que se manejan y las relaciones entre ellos. Se analizan las necesidades del sistema, especificándolas mediante los requerimientos funcionales y no funcionales. A partir de ellos se realiza la definición de los casos de uso del sistema y sus descripciones.

2.2. Descripción de los procesos vinculados al negocio.

Debido a las nuevas exigencias y responsabilidades del departamento Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación se da la necesidad de realizar un nuevo sistema para la gestión de información en cuanto a la formación y el empleo de estudiantes de la Educación Superior y Técnico Profesional. Esta nueva aplicación nombrada Sistema Unificado de Gestión de Fuerza de Trabajo Calificada (GeForza) contará con varios módulos los cuales son: Módulo de Administración el cual es el encargado de la seguridad del sistema y de estandarizar los clasificadores y nomencladores utilizados en el mismo. Sistema Matriz el cual es el encargado de gestionar toda la información necesaria, dicha información está compuesta por la disponibilidad de los graduados y la demanda de los organismos, la cual es utilizada en el desarrollo de otros módulos, para la realización de los planes anuales, módulo Plan de Ingreso, módulo Plan de Ingreso a los Centros Universitarios Municipales (CUM) y el módulo Plan de Distribución de Graduados de Nivel Superior, siendo este último el objetivo del trabajo.

El módulo Plan de Distribución de Graduados de Nivel Superior es el encargado de distribuir los graduados universitarios teniendo en cuenta las necesidades socioeconómicas del país, para ello atenderá la demanda de cada organismo por carreras, provincias y municipios contando con la disponibilidad de los graduados por centro de estudio, carreras, provincias y municipios y diferenciando los tipos de graduados en graduados de Orden 18, graduados civiles, graduados del MINFAR, graduados del MININT; quedando de esta forma distribuidos todos los graduados por sus lugares de residencia e indicando a los organismos la cantidad de personal con que cuenta para cubrir sus demandas y diferenciándolos por el tipo de graduado. Este sistema está estrechamente relacionado

con otros módulos del sistema GeForza como es el caso del Módulo de Administración y el Sistema Matriz.

2.3. Modelo del dominio

Un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno donde estará el sistema. (Ivar Jacobson)

Después de haber realizado un estudio de los procesos que intervienen en el sistema de distribución y de haber realizado entrevistas a los clientes expertos en el proceso, se decidió realizar un modelo del dominio, debido a que no era necesario modelar el negocio pues existe una aplicación que define el mismo. El modelo del dominio representa cosas del mundo real y para poder identificar los conceptos, se hace necesario investigar el dominio del problema. Se especifican las relaciones que existen entre los conceptos, permitiendo de esta forma comprender mejor el modelo de dominio.

2.3.1. Modelo del dominio.

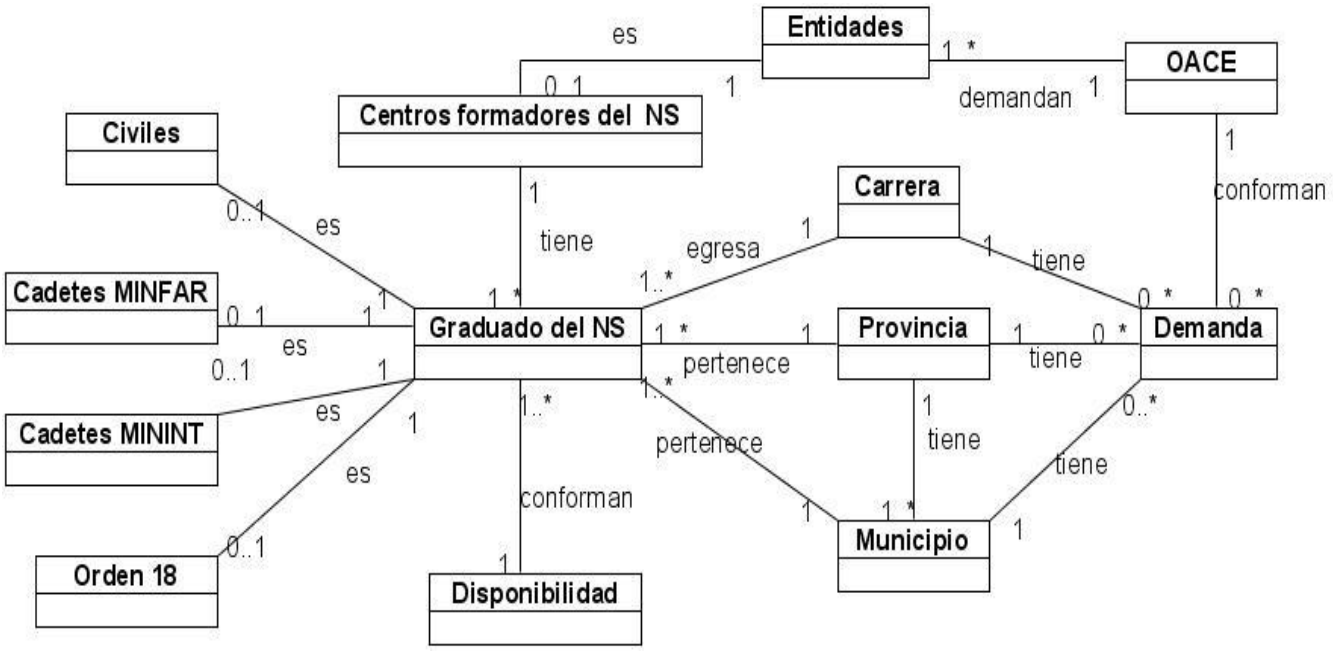

Figura #1 Diagrama de dominio.

2.3.2. Definición de las clases del modelo del dominio.

OACE: se refiere a los Organismos de la Administración Central del Estado.

Entidades: se refiere a todas las empresas que forman parte de un organismo.

Centros formadores del NS: se refiere a los centros encargados de la formación de estudiantes universitarios que pertenecen al curso regular diurno

Graduado del NS: se refiere a aquellos estudiantes del nivel superior que se gradúan.

Civiles: es una definición del graduado que se les da a los graduados que no pertenecen ni al MINFAR, ni al MININT, ni a la Orden 18.

Cadetes MINFAR: es una definición que se les da a los graduados que pertenecen al MINFAR.

Cadetes MININT: es una definición que se les da a los graduados que pertenecen al MININT.

Orden 18: es una definición que se les da a los graduados que forman parte de la Orden 18.

Carrera: es un conjunto de asignaturas, que se estudian en un centro de enseñanza superior que corresponden a una materia específica.

Provincia: se refiere a la división política administrativa de Cuba.

Municipio: se refiere a la división política administrativa de cada una de las provincias de Cuba.

Disponibilidad: se refiere a la cantidad de estudiantes graduados del curso regular diurno, por carreras, atendiendo a la provincia y a su municipio de residencia.

Demanda: se refiere a la cantidad de estudiantes graduados de nivel superior, formados en curso regular diurno, que demanda el país de una determinada carrera, por empresas, provincia y municipio de residencia.

2.4. Especificación de los requerimientos del software.

El propósito fundamental del levantamiento de requerimientos es guiar el desarrollo hacia el sistema correcto. Esto sólo se puede conseguir mediante una descripción de los requerimientos del sistema, es decir, las condiciones o capacidades que el sistema debe cumplir. Las cuales tienen que ser

suficientemente buenas como para llegar a un acuerdo entre los clientes incluyendo dentro de estos a los usuarios y los desarrolladores, sobre qué debe y qué no debe hacer el sistema en cuestión.

2.4.1. Requerimientos funcionales

Los requerimientos funcionales permiten una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el mismo.

RF 1. Tramitar la disponibilidad.

1.1. Importar disponibilidad.

1.1.1. Importar la disponibilidad desde el Sistema Matriz.

1.1.2. Realizar la importación.

1.2. Mostrar listado de la disponibilidad.

- Provincia de residencia del graduado.
- Municipio de residencia del graduado.
- Centro de educación superior del graduado.
- Carrera del graduado.
- Total de graduados que pertenecen a la Orden 18.
- Total de graduados cadetes MINFAR.
- Total de graduados cadetes MININT.
- Total de graduados civiles.
- Total general que incluye a todos los graduados ya sean Orden 18, cadetes MINFAR, cadetes MININT o civiles.

1.3. Modificar la disponibilidad.

1.3.1. Permitir modificar una disponibilidad sólo por sus campos de total.

1.4. Buscar la disponibilidad.

1.4.1. Permitir buscar una disponibilidad por cualquiera de sus componentes.

- Provincia de residencia del graduado.
- Provincia y municipio de residencia del graduado.
- Nombre o código del centro de educación superior del graduado.
- Nombre o código de la carrera del graduado.

1.5. Eliminar disponibilidad.

1.5.1. Permitir eliminar una disponibilidad de la base de datos con todos sus componentes.

1.6. Agregar disponibilidad.

1.6.1. Permitir agregar una disponibilidad con todos sus componentes.

- Provincia de residencia del graduado.
- Municipio de residencia del graduado.
- Centro de educación superior del graduado.
- Carrera del graduado.
- Total de graduados que pertenecen a la Orden 18.
- Total de graduados cadetes MINFAR.
- Total de graduados cadetes MININT.
- Total de graduados normales.
- Total general que incluye a todos los graduados ya sean Orden 18, cadetes MINFAR, cadetes MININT o civiles.

1.7. Cargar disponibilidad de años posteriores y anteriores.

1.8. Al modificar, adicionar, eliminar, el sistema realiza una salva en la base de datos.

RF 2. Tramitar la demanda.

2.1. Importar demanda.

2.1.1. Importar la demanda desde el Sistema Matriz.

2.1.2. Realizar la importación.

2.2. Mostrar listado de la demanda por:

- Provincia donde está enclavada la entidad.
- Municipio donde está enclavada la entidad.
- Organismo (incluye los CAP).
- Entidad que pertenece a ese organismo.
- Carrera que se está demandando.
- Total de demanda/provincia/municipio/organismo/empresa/carrera.

2.3. Buscar la demanda.

2.3.1. Permitir buscar una demanda por cualquiera de sus componentes.

- Provincia.
- Provincia y municipio.
- Nombre y código del organismo.
- Nombre y código de la entidad que pertenece al organismo.
- Nombre y código de carrera que se está demandando.

2.4. Agregar la demanda.

2.4.1. Permitir agregar una demanda con todos sus componentes.

- Provincia donde está enclavada la entidad.
- Municipio donde está enclavada la entidad.
- Organismo (incluye los CAP).
- Entidad que pertenece a ese organismo.
- Carrera que se está demandando.
- Total de demanda/ provincia/ municipio/ organismo/ empresa/ carrera.

2.5. Modificar la demanda.

2.5.1. Permitir modificar una demanda sólo por el campo total.

2.6. Eliminar la demanda.

2.6.1. Permitir eliminar una demanda de la base de datos con todos sus componentes.

2.7. Cargar demanda de años posteriores y anteriores.

2.8. Al modificar, adicionar, eliminar, el sistema realiza una salva en la base de datos.

RF 3. Tramitar la asignación para organismos.

3.1. Generar asignación

3.1.1. Ejecutar (Para la ejecución del algoritmo es necesario tener demanda y disponibilidad).

3.1.2. Realizar el algoritmo matemático.

3.2. Mostrar asignación.

- Provincia
- Carrera.
- Centro de estudio.
- Organismo.

3.2.1. Mostrar todos los vectores que genera la asignación por cada uno de los municipios de esa provincia.

- Demanda.
- Faltante (demandas que no se satisfacen).
- Disponibilidad.
- Sobrante (disponibles que no se han asignado).
- Cantidad de la disponibilidad que son del MINFAR.
- Cantidad de la disponibilidad que son del MININT.

- Cantidad de la disponibilidad que son de Orden 18.
- Cantidad de la disponibilidad que son civiles.
- Asignación al organismo.
- Asignación de Orden 18 (cantidad que fueron asignado de la Orden 18).
- Asignación de civiles.
- Asignación de MINFAR.
- Asignación de MININT.

3.3. Eliminar asignación.

3.3.1. Permitir eliminar una asignación de la base de datos con toda su información.

3.4. Modificar asignación.

3.4.1. Permitir modificar una asignación por los vectores.

- Disponibilidad de Orden 18.
- Disponibilidad de civiles.
- Disponibilidad de MINFAR.
- Disponibilidad de MININT.

3.4.2. Actualizar los vectores:

- Asignación a organismo.
- Asignación de Orden 18.
- Asignación de civiles.
- Asignación de MINFAR.
- Asignación de MININT.
- Sobrante
- Faltante

3.5. Buscar una asignación

3.5.1. Permitir buscar una asignación por:

- Provincia.
- Carrera.
- Centro de estudio.
- Organismo.

3.6. Agregar asignación.

3.6.1. Permitir adicionar una asignación por los vectores.

- Disponibilidad de Orden 18.

- Disponibilidad de civiles.
- Disponibilidad de MINFAR.
- Disponibilidad de MININT.

3.6.2. Actualizar los vectores:

- Provincia.
- Carrera.
- Centro de estudio.
- Organismo.
- Municipio.
- Asignación a organismo.
- Asignación de Orden 18.
- Asignación de civiles.
- Asignación de MINFAR.
- Asignación de MININT.
- Sobrante
- Faltante

3.7. Cargar asignación de años anteriores y posteriores.

3.8. Al modificar, adicionar, eliminar, el sistema realiza una salva en la base de datos.

RF 4. Tramitar la asignación para entidades.

4.1. Mostrar asignación.

- Provincia
- Carrera.
- Centro de estudio.
- Organismo.
- Entidad.

4.1.1. Mostrar todos los vectores por cada uno de los municipios de esa provincia.

- Demanda de la entidad.
- Faltante (demandas que no se satisfacen).
- Disponibilidad del organismo (sale como resultado de la asignación al organismo).
- Sobrante (disponibles que no se han asignado).
- Cantidad de la disponibilidad de Orden 18.

- Cantidad de la disponibilidad que son civiles.
- Cantidad de la disponibilidad que son cadetes del MINFAR.
- Cantidad de la disponibilidad que son cadetes del MININT.
- Asignación a la entidad
- Asignación de Orden 18 (cantidad que fueron asignado de la Orden 18).
- Asignación de civiles.
- Asignación de MINFAR.
- Asignación de MININT.

4.2. Eliminar asignación.

4.2.1. Permitir eliminar una asignación de la base de datos con toda su información.

4.3. Modificar asignación.

4.3.1. Permitir modificar una asignación por los vectores.

- Disponibilidad de Orden 18.
- Disponibilidad de civiles.
- Disponibilidad de cadetes MINFAR.
- Disponibilidad de cadetes MININT.

4.3.2. Actualizar los vectores:

- Municipio.
- Asignación a entidad.
- Asignación de Orden 18.
- Asignación de civiles.
- Asignación de MINFAR.
- Asignación de MININT.
- Sobrante.
- Faltante.

4.4. Buscar una asignación

4.4.1. Permitir buscar una asignación por:

- Provincia.
- Carrera.
- Centro de estudio.
- Organismo.
- Entidad.

4.5. Agregar asignación.

4.5.1. Permitir adicionar una asignación por los vectores.

- Disponibilidad de Orden 18.
- Disponibilidad de civiles.
- Disponibilidad de cadetes MINFAR.
- Disponibilidad de cadetes MININT.

4.5.2. Actualizar los vectores:

- Provincia.
- Carrera.
- Centro de estudio.
- Organismo.
- Entidad
- Municipio.
- Asignación a entidad.
- Asignación de Orden 18.
- Asignación de civiles.
- Asignación de MINFAR.
- Asignación de MININT.
- Sobrante
- Faltante

4.6. Cargar asignación de años posteriores y anteriores.

4.7. Al modificar, adicionar, eliminar, el sistema realiza una salva en la base de datos.

RF 5. Obtener información de los nomencladores.

5.1. Permitir obtener listados de cada uno de estos nomencladores. Esta información se carga del Módulo de Administración

- Carreras.
- Centros.
- Centros priorizados.
- Correlacionador.
- Empresas.
- Rama de las ciencias.
- Familia de especialidades.

- Municipios.
- Organismos.
- Provincias.
- Subordinación local.

RF 6. Obtener reportes de disponibilidad.

6.1. Mostrar un listado de los siguientes reportes con los datos específicos de cada uno:

- Disponibilidad plan por provincia/ municipio/centro/ carrera.
- Disponibilidad plan por provincia/ municipio/ carrera.
- Disponibilidad plan por provincia/ carrera.
- Disponibilidad plan por provincia/ carrera/ municipio.
- Disponibilidad sobrante por centro/ carrera/ provincia/ municipio.
- Disponibilidad sobrante por centro/ carrera.
- Disponibilidad sobrante por provincia y carrera.
- Disponibilidad plan por carrera y provincia.
- Disponibilidad sobrante por carrera y provincia.

RF 7. Obtener reportes de demanda.

7.1. Mostrar un listado de los siguientes reportes con los datos específicos de cada uno:

- Demanda por provincia/ carrera/ organismo/ empresa.
- Demanda por provincia/ organismo/ carrera/ empresa.
- Resumen de organismo abierto por provincia.
- Resumen organismo/ carrera abierta por provincia.
- Resumen carrera/ organismo abierto por provincia.
- Resumen organismo/ empresa/ carrera abierto por provincia.
- Balance resumen por carrera/ organismo/ empresa abierto por provincia.
- Resumen subordinación local por actividad/ provincia/ empresa/ carrera.
- Resumen subordinación nacional por organismos/ empresas/ carrera.
- Demanda por provincia/ carrera/ organismo/ empresa abierto por municipio.
- Demanda por organismo/ carrera abierta por provincia.
- Demanda por provincia/ municipio/ carrera/ organismo/ empresa.
- Demanda por carrera abierto por provincias.
- Listado de organismos que demandan, cargados en el sistema.

RF 8. Obtener reportes de asignación.

8.1. Mostrar un listado de los siguientes reportes con los datos específicos de cada uno:

- Plan de ubicación de graduados por centro/ carrera/ organismo/ empresa/ provincia/ municipio.
- Plan de ubicación de graduados por organismo/ centro/ carrera/ empresa/ provincia/ municipio.
- Balance disponibilidad – asignación de carreras.
- Asignación por provincia/ organismo/empresa/ carrera/ centro abierto por municipio.
- Asignación por provincia/ carrera/ centro/ organismo/ empresa abierto por municipio.
- Balance de demanda y asignación por provincia/ carrera/ organismo.

RF 9. Exportar a PDF.

9.1. Crear un documento PDF con toda la información.

2.4.1. Requerimientos no funcionales:

Los requerimientos no funcionales especifican cualidades, propiedades del sistema; como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, etc.

Usabilidad.

- A los administradores finales de la aplicación se les debe dar un adiestramiento básico en el uso de la aplicación. Estas personas deben tener un nivel de acceso amplio en la aplicación para poder darle respuesta a cada incidente ocurrido.

Seguridad.

- Políticas de seguridad por usuarios y roles: el sistema debe contar con un grupo de políticas de accesibilidad a las diferentes funcionalidades del mismo en dependencia del nivel de autorización que presente un usuario determinado.
- El sistema garantizará la autenticación como primera acción para los casos en que sea necesario. Esta consistirá en suministrar un nombre de usuario único y una contraseña que debe ser de conocimiento exclusivo de la persona que se autentica.
- El sistema dejará registradas todas las operaciones realizadas por el personal que opera el mismo y los supervisores.
  - Código y nombre de la persona.
  - Fecha y hora de inicio y de terminación de la operación.
  - Operación realizada y oficina desde donde se realizó la operación.

- Estación de trabajo desde la que operó.

Eficiencia.

- El sistema debe funcionar con un máximo rendimiento pero ajustado a bajas prestaciones de las computadoras debido que no todos los organismos poseen tecnología de punta.
- El sistema requiere de un buen rendimiento que se apoya en el mínimo acceso a base de datos y realización de consultas no redundantes.
- La aplicación debe garantizar un acceso fácil y rápido, contando con un menú que satisfaga las necesidades de los usuarios.

Soporte

- El sistema deberá presentar un manual de usuario, permitiendo con ello un correcto uso de sus funcionalidades y brindarle al usuario una mayor experiencia del trabajo con el mismo.
- La documentación del sistema debe estar actualizada en todos los aspectos, fases de trabajo y ciclos de desarrollo del mismo, permitiendo con ello un respaldo tanto ingenieril como legal del desarrollo de dicho sistema.
- El sistema contará con un grupo de soporte y asesoría al cliente del producto destinado a brindar asesoría y soporte técnico al mismo.

Restricciones de diseño.

- Lenguaje de programación: PHP 5.
- Como gestor de base de datos se utilizará PostgreSQL 8.3 o superior.
- Los marcos de trabajo que se utilizarán son: Zend Framework para la capa de negocio, Ext JS para la capa de presentación, Doctrine para la capa de acceso a datos y Sauxe.
- Como Entorno de Desarrollo Integrado se empleará NetBeans 6.8.
- El servidor de aplicaciones será Apache-2.0 o superior
- El modelado UML se hará con Visual Paradigm 6.1.
- Se utilizará como metodología de desarrollo de software RUP.
- Las interfaces destinadas al usuario, se programarán en JavaScript.

Hardware.

- Se necesitará una impresora, para la impresión de los reportes del sistema.
- Las PCs clientes deben tener las siguientes características una RAM 256 MB o superior, 20 GB o más de disco duro, 1.2 GHz o más de procesador.
- Todas las computadoras clientes deben estar conectadas a la red para poder acceder a la aplicación.

- Se requiere de un servidor para bases de datos con las siguientes características.
  - Servidor Xeon a 3.0 GHz.
  - 1 GB de memoria RAM.
  - Dos discos duros de 36 y 250 GB, este último con dos particiones.

Software.

- Para el cliente: se usará un sistema operativo Windows 95, cualquier versión superior o GNU/Linux.
- Las computadoras cliente del sistema y ubicadas en el dominio de la organización, deben tener el navegador Mozilla Firefox.
- Para el servidor: sistema operativo Windows Server 2000 o superior o Linux; Ubuntu Server 7.10 o superior.
- Servidor Web: Apache 2.0 o superior.
- Gestor de base de datos: PostgreSQL.

Apariencia o interfaz externa.

- Interfaz web: la interfaz deberá ser sencilla, amigable con colores suaves a la vista y sin cúmulo de imágenes u objetos que distraigan al cliente del objetivo de su empleo, brindando facilidades que permitan interactuar con el sistema de forma fácil y rápida, permitiendo que personas con pocas habilidades informáticas manejen el sistema.

Requerimientos de licencia.

- El proyecto utiliza la política de software libre donde todas las herramientas que utilizan son software libre. Para la herramienta Visual Paradigm se utiliza la licencia que la universidad compró.

Requerimientos legales, de derecho de autor y otros.

- La Universidad de las Ciencias Informáticas tiene el derecho de autor sobre el Sistema Unificado de Gestión de Fuerza de Trabajo Calificada.

2.4. Modelo de casos de uso del sistema.

El modelo de casos de uso del sistema permitió llegar a un acuerdo entre el cliente y los desarrolladores sobre las condiciones que debe cumplir el sistema. A continuación se podrá apreciar el diagrama de casos de uso, donde se representa gráficamente a los procesos que se llevarán a cabo en el sistema y su interacción con el actor.

2.4.1. Actor del sistema.

El actor es una entidad externa del sistema que de alguna manera participa en la historia del caso de uso. Por lo regular estimula el sistema con eventos de entrada o recibe algo de él. (Ivar Jacobson)

Actor	Descripción
Funcionario	Es un actor común para: el funcionario del MEP, el funcionario de los OACE y los CAP.

Tabla #1 Descripción de los actores del sistema.

2.4.2. Diagrama de casos de uso del sistema.

Los casos de uso son "fragmentos" de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. De manera más precisa, un caso de uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con su actor, incluyendo alternativas dentro de la secuencia. (Ivar Jacobson)

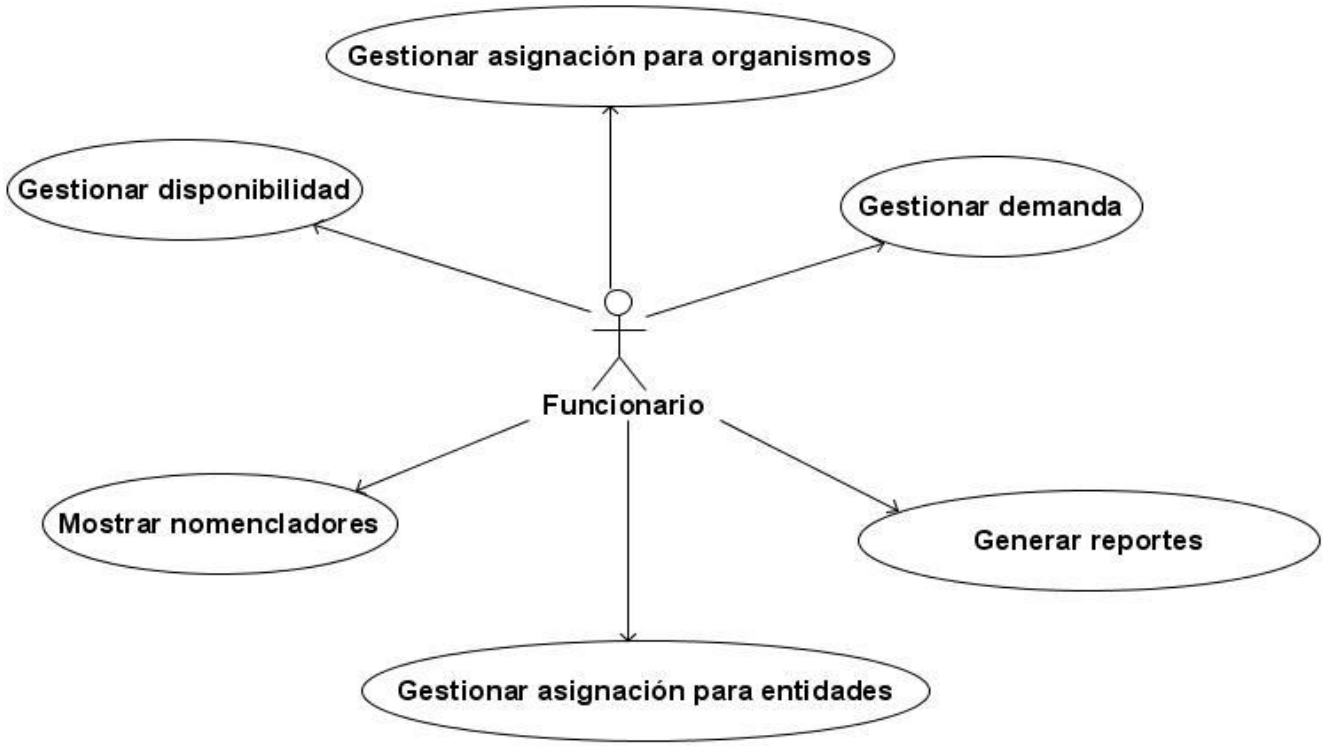

Figura #2 Diagrama de casos de uso del sistema.

2.5. Descripciones abreviadas de los casos de uso del sistema

La descripción de los casos de uso del sistema permite comprender el comportamiento del sistema desde el punto de vista del usuario. A continuación se muestran las descripciones abreviadas de los casos de uso, las descripciones detalladas pueden ser encontradas en el Anexo #5.

Caso de uso:	Gestionar demanda.
Actores:	Funcionario.
Resumen:	El caso de uso comienza cuando el funcionario selecciona la opción Gestionar demanda. A partir de ahí el funcionario tiene la opción de Adicionar, Modificar, Eliminar, Buscar, Exportar a PDF, Cargar datos desde Matriz y Distribución por años. Posteriormente introduce los datos necesarios para realizar la operación deseada y el sistema adiciona, modifica, elimina, busca, exporta, carga los datos desde Matriz o permite escoger por años, según la opción deseada. Finaliza el caso de uso cuando se realizan todas las operaciones deseadas y el funcionario selecciona la opción salir.
Precondiciones:	El actor debe estar autenticado en el sistema.
Referencias	RF 2, RF 9.
Prioridad	Crítico.

Tabla #2 Resumen del caso de uso "Gestionar demanda".

Caso de uso:	Gestionar disponibilidad.
Actores:	Funcionario.
Resumen:	El caso de uso comienza cuando el funcionario selecciona la opción Gestionar disponibilidad. A partir de ahí el funcionario tiene la opción de Adicionar, Modificar, Eliminar, Buscar los registros creados de la disponibilidad hasta el momento, Exportar a PDF, Cargar datos desde Matriz y Distribución por años. Posteriormente introduce los datos necesarios para realizar la operación deseada y el sistema adiciona, modifica, elimina, busca, exporta, carga los datos del Sistema Matriz o cambia el año según la opción deseada, finaliza el caso de uso cuando se realizan todas las operaciones deseadas y el funcionario selecciona la opción salir.
Precondiciones:	El actor debe estar autenticado en el sistema.
Referencias	RF 1, RF 9.
Prioridad	Crítico.

Tabla #3 Resumen del caso de uso "Gestionar disponibilidad".

Caso de uso:	Gestionar asignación para organismos.
Actores:	Funcionario.
Resumen:	El caso de uso comienza cuando el funcionario selecciona la opción Gestionar asignación para organismos. A partir de ahí el funcionario tiene la opción de Generar Asignación, Adicionar-Modificar, Eliminar, Buscar y Distribución por años. Posteriormente introduce los datos necesarios para realizar la operación deseada y el sistema genera la asignación, adiciona o modifica, elimina, busca o cambia el año según la opción deseada. Finaliza el caso de uso cuando se realizan todas las operaciones deseadas y el funcionario selecciona la opción salir.
Precondiciones:	El actor debe estar autenticado en el sistema.
Referencias	RF 3.
Prioridad	Crítico.

Tabla #4 Resumen del caso de uso "Gestionar asignación para organismos".

Caso de uso:	Gestionar asignación para entidades.
Actores:	Funcionario.
Resumen:	El caso de uso comienza cuando el funcionario selecciona la opción Gestionar asignación para entidades. A partir de ahí el funcionario tiene la opción de Adicionar, Modificar, Eliminar, Buscar y Distribución por años. Posteriormente introduce los datos necesarios para realizar la operación deseada y el sistema adiciona, modifica, elimina, busca o cambia el año según la opción deseada. Finaliza el caso de uso cuando se realizan todas las operaciones deseadas y el funcionario selecciona la opción salir.
Precondiciones:	El actor debe estar autenticado en el sistema.
Referencias	RF 4.
Prioridad	Crítico.

Tabla #5: Resumen del caso de uso "Gestionar asignación para entidades".

Caso de uso:	Mostrar nomencladores.
Actores:	Funcionario.
Resumen:	El caso de uso se inicia cuando el funcionario escoge la opción Mostrar nomencladores, a partir de ahí puede seleccionar el informe que desea ver y puede imprimir. El caso de uso finaliza cuando el funcionario solicita salir.
Precondiciones:	El actor debe estar autenticado en el sistema.
Referencias	RF 5, RF 9.
Prioridad	Auxiliar.

Tabla #6 Resumen del caso de uso "Mostrar nomencladores".

Caso de uso:	Generar reportes.
Actores:	Funcionario.
Resumen:	El caso de uso se inicia cuando el funcionario escoge la opción Generar reportes, a partir de ahí el funcionario selecciona el reporte que desea ver, puede imprimir e importar el reporte en diferentes formatos. El caso de uso finaliza cuando el funcionario realiza todas las acciones deseadas y solicita salir.
Precondiciones:	El actor debe estar autenticado en el sistema.
Referencias	RF 6, RF 7, RF 8.
Prioridad	Auxiliar.

Tabla #7 Resumen del caso de uso "Generar reportes".

2.6. Conclusiones

En este capítulo se definió que sería mejor hacer una modelación del dominio para el sistema propuesto, describiéndose de esta forma los conceptos fundamentales y las relaciones entre ellos. Además, se describen los requerimientos funcionales y no funcionales del sistema. Se describe el actor del sistema que se automatizará. Se realizó el diagrama de casos de uso del sistema y las descripciones de los mismos. Quedando planteadas de esta forma, las condiciones y características del sistema propuesto.

Capítulo 3: Análisis y Diseño

3.1. Introducción.

En el presente capítulo se describe cómo debe ser el sistema, para ello se desarrollan los modelos de análisis y diseño de todos los casos de uso del sistema, los cuales son de suma importancia para el desarrollo del software. Además, se realizan los diagramas de colaboración del análisis y de secuencia del diseño de dichos casos de uso, así como los modelos lógico y físico de la base de datos y el diagrama de despliegue.

3.2. Análisis.

Durante el análisis, se analizan los requisitos que se describen en la captura de requerimientos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requerimientos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema entero, incluyendo su arquitectura. (Ivar Jacobson)

El análisis ofrece una especificación más detallada de los requerimientos, lo cual permite comprender mejor las funcionalidades del sistema, este se describe utilizando el lenguaje de los desarrolladores para describir los resultados. El modelo de análisis se considera como una primera aproximación al modelo de diseño.

3.2.1. Diagramas de clases del análisis.

En el modelo de análisis los casos de uso son descritos mediante clases del análisis y sus objetos. Las clases que se modelan refinan los requerimientos funcionales obtenidos anteriormente, son identificadas con sus relaciones y descritas en un diagrama de clases utilizando los siguientes estereotipos:

- Clases interfaz: modelan la interacción entre el sistema y sus actores.
- Clases controladoras: coordinan la realización de uno o unos pocos casos de uso, relacionando las actividades de los objetos que implementan sus funcionalidades.
- Clases entidad: modelan información que posee larga vida y que es a menudo persistente. (Ivar Jacobson)

A continuación se representa el diagrama de clases del análisis del caso de uso “Gestionar demanda”, ya que el mismo se toma como referencia en todo el trabajo.

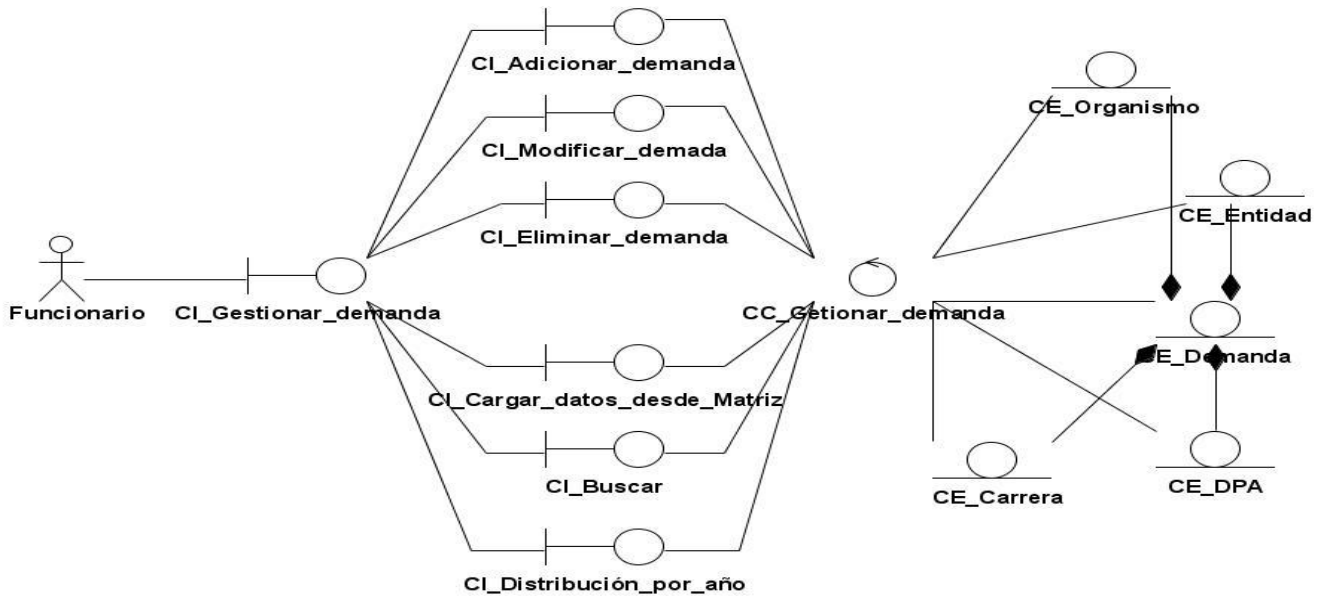

Figura #3 Diagrama de clases del análisis CUS "Gestionar demanda".

Para ver el resto de los diagramas de clases del análisis ir al Anexo #6.

En el análisis se usan los diagramas de colaboración, ya que el objetivo principal es identificar las funcionalidades de cada objeto y las responsabilidades sobre ellos. En el Anexo #7 se pueden consultar los diagramas de colaboración del análisis para cada uno de los escenarios de los casos de uso.

3.3. Diseño

Una entrada esencial del diseño es el resultado del modelo de análisis. El diseño es el encargado de modelar el sistema, contribuyendo a lograr una arquitectura sólida y estable, para que soporte todos los requerimientos, incluyendo los no funcionales. (Ivar Jacobson)

El diseño será utilizado para visualizar la implementación y para soportar las técnicas de programación gráfica, ya que se ha decidido realizar ingeniería inversa en esta fase partiendo de la de implementación.

3.3.1. Patrones de diseño

Los patrones son soluciones a problemas recurrentes en el entorno de desarrollo. Se pueden considerar como recetas para solucionar varias veces un problema del mismo tipo. En el diseño de la

propuesta de solución se tiene en cuenta un patrón clásico del diseño conocido como arquitectura MVC que implementa Zend Framework.

Modelo Vista Controlador (MVC) es un patrón que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos, aparta esencialmente la lógica del negocio de la lógica de la presentación, factor que posibilita la simplificación del trabajo y el mantenimiento de los sistemas. Para ver imagen del Modelo Vista Controlador ir Anexo #8.

La correcta implementación de este patrón dispone de tres entidades básicas: el Modelo, la Vista y el Controlador.

Modelo: representa la información con la que trabaja la aplicación, es decir, su lógica de negocio. Se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.

Vista: muestra la información al usuario. Pueden existir múltiples vistas del modelo. Transforma el modelo en una página web que permite al usuario interactuar con ella.

Controlador: se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. Mantiene aislado al modelo y a la vista de los detalles del protocolo utilizado para las peticiones.

Este modelo de arquitectura presenta varias ventajas:

- Hay una clara separación entre los componentes de un programa, lo cual permite implementarlos por separado.
- Hay una API muy bien definido; cualquiera que lo use, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- La conexión entre el Modelo y sus vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación. (Software, 2007-2008)

3.3.2. Patrones GRASP (patrones generales de software para asignación de responsabilidades) utilizados.

En la realización del diseño se utilizaron además patrones GRASP, los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos y patrones GOF.

- Experto: este patrón permite asignar una responsabilidad al más experto en información, es decir, la clase que cuenta con la información necesaria para cumplir la responsabilidad. Esto se evidencia en el diseño cuando el controlador frontal le asigna responsabilidades a las clases controladoras para realizar acciones de acuerdo con las peticiones que recibe. El uso de este patrón permitirá a los objetos valerse de su propia información para hacer lo que se les pide, favorece la existencia de mínimas relaciones entre las clases, lo que permite contar con un sistema robusto, fácil de mantener y fáciles de ampliar. Ejemplo: para el caso de uso Gestionar demanda, este patrón se pone de manifiesto cuando la *server page* instancia a la Zend_Controller_Front la cual es la encargada de determinar a qué clase instanciar para que se ejecute el método que va a hacer la petición a la base de datos.
- Creador: este patrón es el responsable de guiar la asignación de responsabilidades relacionadas con la creación de objetos, de forma tal que una instancia de un objeto sólo pueda ser creada por el objeto que contiene la información necesaria para ello. Existe un único *script* PHP que crea una única instancia del controlador frontal, dicho controlador frontal sólo crea instancias de las clases controladoras y estas, a su vez, sólo crean objetos de la clase *Zend_View*. El uso de este patrón permite crear las dependencias mínimas necesarias entre las clases, lo que favorece al mantenimiento del sistema y ofrece mejores oportunidades de reutilización. Este patrón permitirá tener clases fáciles de mantener, entender y reutilizar. Ejemplo: para el caso de uso Gestionar demanda, este patrón se pone de manifiesto en la clase *server page* la cual es la única clase que instancia Zend_Controller_Front la cual crea una instancia de la clase controladora GestdemandaController y esta a su vez crea un objeto de la clase Zend_View.
- Alta cohesión: este patrón se tiene en cuenta para asignar responsabilidades de modo que la cohesión siga siendo alta. De modo que las clases con responsabilidades estrechamente relacionadas no realicen un trabajo enorme. Con el uso de este patrón se evita tener clases con un alto grado de abstracción, que asuman responsabilidades que deberían haber delegado a otros objetos. Se tienen las clases controladoras que se encargan de ejecutar acciones de acuerdo con las peticiones que le llegan y las clases de acceso a datos que interactúan con el modelo, de forma tal que se elimina la sobrecarga de funcionalidades en las clases controladoras. Este patrón permitirá tener clases fáciles de mantener, entender y reutilizar. Ejemplo: para el caso de uso Gestionar demanda, este patrón se pone de manifiesto en la clase GestdemandaController la cual es una clase sencilla sin ningún grado de abstracción, donde sólo se implementan los métodos correspondientes a la misma.

- Controlador: este patrón se tiene en cuenta para realizar las asignaciones en cuanto al manejo de los eventos del sistema y definir sus operaciones. Zend Framework contribuye a la utilización de este patrón ya que define un controlador frontal que implica que todas las solicitudes son dirigidas a un único *script* PHP que se encarga de instanciarlo y redirigir las llamadas. Ejemplo: para el caso de uso Gestionar demanda, este patrón se pone de manifiesto en la clase Zend_Controller_Front la cual es la encargada de llamar a las clases necesarias para responder a las peticiones realizadas.
- Bajo Acoplamiento: el acoplamiento mide la fuerza con que una clase está conectada a otra, de esta forma, una clase con bajo acoplamiento debe tener un número mínimo de dependencia con otras clases. Una clase de acceso a datos sólo depende de la clase controladora que la usa, así se pueden realizar cambios en cada clase de forma independiente. Este patrón se tuvo en cuenta por la importancia que significa realizar un diseño de clases independientes que puedan soportar los cambios de una manera fácil y permitan la reutilización. Este patrón permitirá tener clases fáciles de mantener, entender y reutilizar. Ejemplo: para el caso de uso Gestionar demanda, este patrón se pone de manifiesto en la clase GestdemandaController la cual es una clase sencilla sin dependencia de ningún tipo con otra clase.

3.3.3. Patrones GOF utilizados.

- Decorator (Decorador): este patrón permite añadir funcionalidad a una clase dinámicamente. Zend Framework implementa este patrón en la clase Zend_View, encargada de asignarle responsabilidades a objetos de manera dinámica y configurarlos con nuevos atributos. Ejemplo: para el caso de uso Gestionar demanda, este patrón se pone de manifiesto con una plantilla global, en la cual se cargan todas las acciones correspondientes al caso de uso.
- Singleton (Instancia única): garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Zend Framework posee una instancia única del controlador frontal disponible mediante este patrón para lograr una vía de entrada única a las solicitudes. Ejemplo: para el caso de uso Gestionar demanda, este patrón se pone de manifiesto en la clase Zend_Controller_Front la cual es la clase encargada de realizar las llamadas a las otras clases.
- Facade (Fachada): este patrón permite utilizar una interfaz común para un conjunto de interfaces del módulo, haciendo que el mismo sea más fácil de usar. Ejemplo: para el caso de uso Gestionar demanda, este patrón se pone de manifiesto en la clase ADO_GestasignacionempresaController,

clase que sirve como interfaz, encargada de realizar las llamadas desde la capa de negocio a las clases que se encuentran en la capa de acceso a datos.

- Abstract Factory (Fábrica abstracta): proporciona una interfaz para la creación de objetos interdependientes o interrelacionados, sin especificar sus clases concretas. Ejemplo: este patrón se pone de manifiesto mediante el uso de Zend Framework con la clase Zend_Cache la cual se utiliza como fábrica para construir objetos de tipo Zend_Controller, Zend_Controller_Front y Zend_Controller_Backend.

3.3.4. Diagrama de clases del diseño

La forma tradicional de modelar las clases del diseño, no es factible a la hora de diseñar una aplicación web. Por ese motivo, se utiliza una extensión de UML para web, que se adapta a la arquitectura de este tipo de sistemas.
Para obtener un nivel correcto de abstracción y detalle se modelan los artefactos del sistema como las páginas, los enlaces entre las mismas, todo el código que irán creando, así como el contenido dinámico de estas una vez que estén en el navegador del cliente.

A continuación se representa el diagrama de clases del diseño del caso de uso “Gestionar demanda”; para ver el resto de los diagramas de clases del diseño ir al Anexo #9.

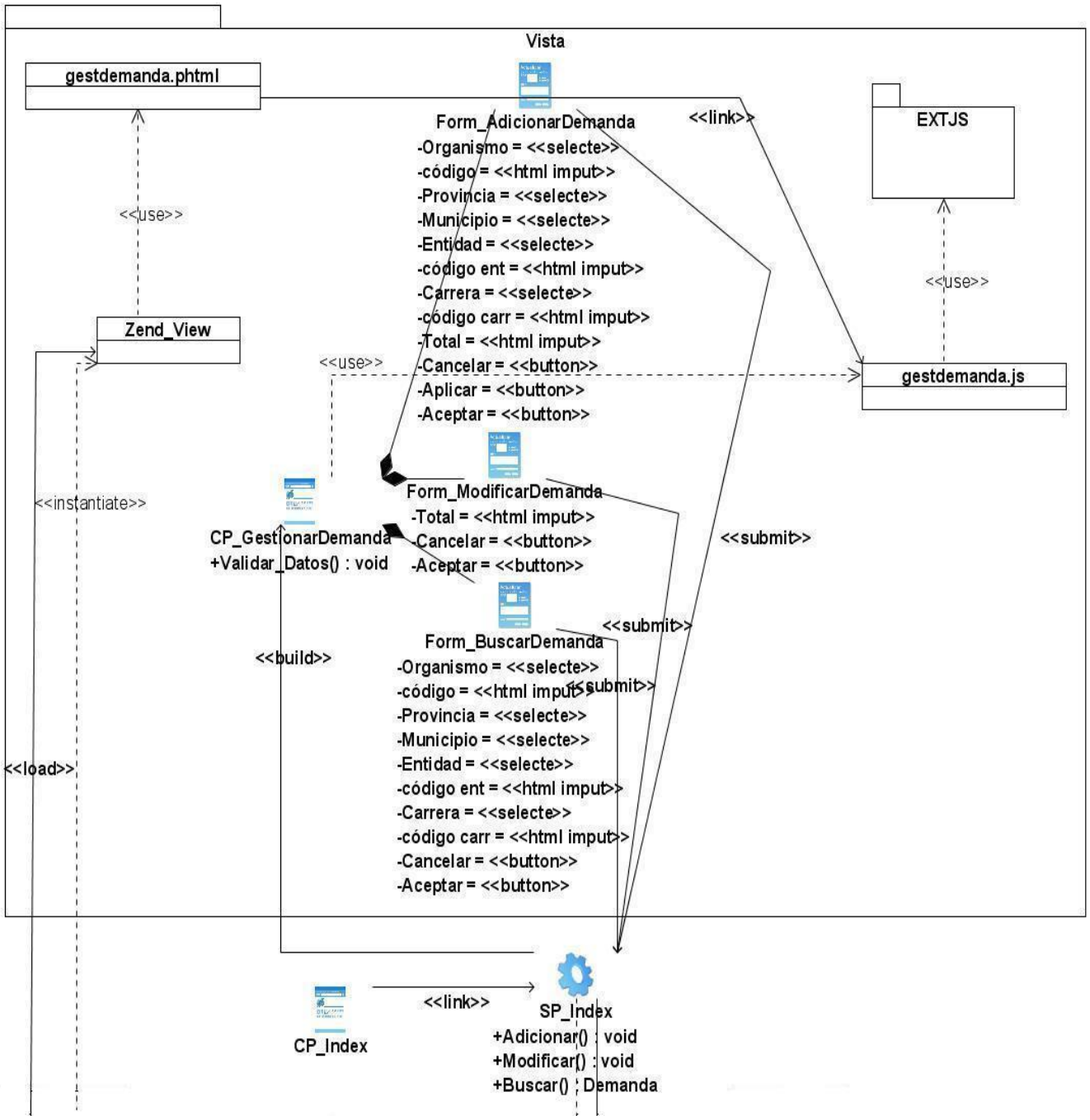

Figura #4 Diagrama de clases del diseño CUS "Gestionar demanda", paquete "Vista".

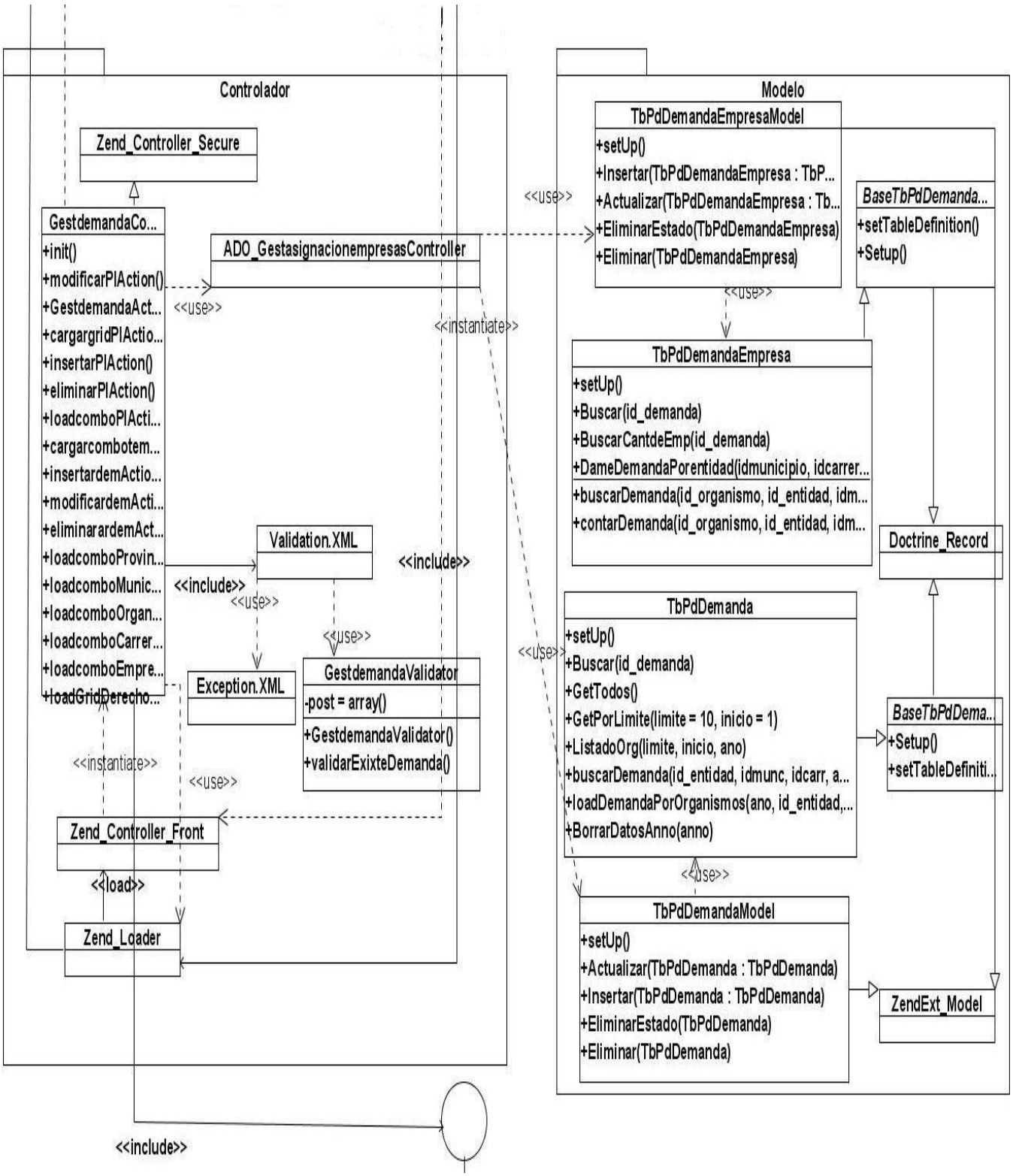

Figura #5 Diagrama de clases del diseño CUS "Gestionar demanda", paquetes "Controlador y Modelo".

Figura #6 Diagrama de clases del diseño CUS "Gestionar demanda", paquete "Servicios".

Para ver las descripciones de las clases de los diagramas del diseño ir al Anexo #10.

Los diagramas de secuencia forman parte del modelado dinámico del sistema y proporcionan una vista detallada de los casos de uso. Son diagramas que muestran la interacción organizada de objetos, mediante mensajes que se envían entre sí, en una secuencia de tiempo. Son útiles para observar la vida de los objetos en un sistema, identificar llamadas a realizar o posibles errores del modelado estático que imposibiliten el flujo de información. En el Anexo #11 se puede consultar una representación de los diagramas de secuencia del diseño.

3.4. Diseño de la base de datos.

El módulo Plan de Distribución de Graduados de Nivel Superior forma parte del proyecto GeForza, el cual decidió realizar una base de datos conjunta diferenciándola por esquemas. Para el diseño de la base de datos se utilizaron los modelos físicos y lógicos de datos, a través de los cuales se diseñaron las clases persistentes con sus relaciones y se modeló la distribución de las tablas en la base de datos. El diseño propuesto satisface las necesidades de persistencia de los datos que el módulo requiere en el cumplimiento de sus requerimientos funcionales y permite la integración con el resto del sistema. En el Anexo #12 se pueden encontrar las descripciones de cada una de las tablas de la base de datos.

3.4.1. Modelo lógico.

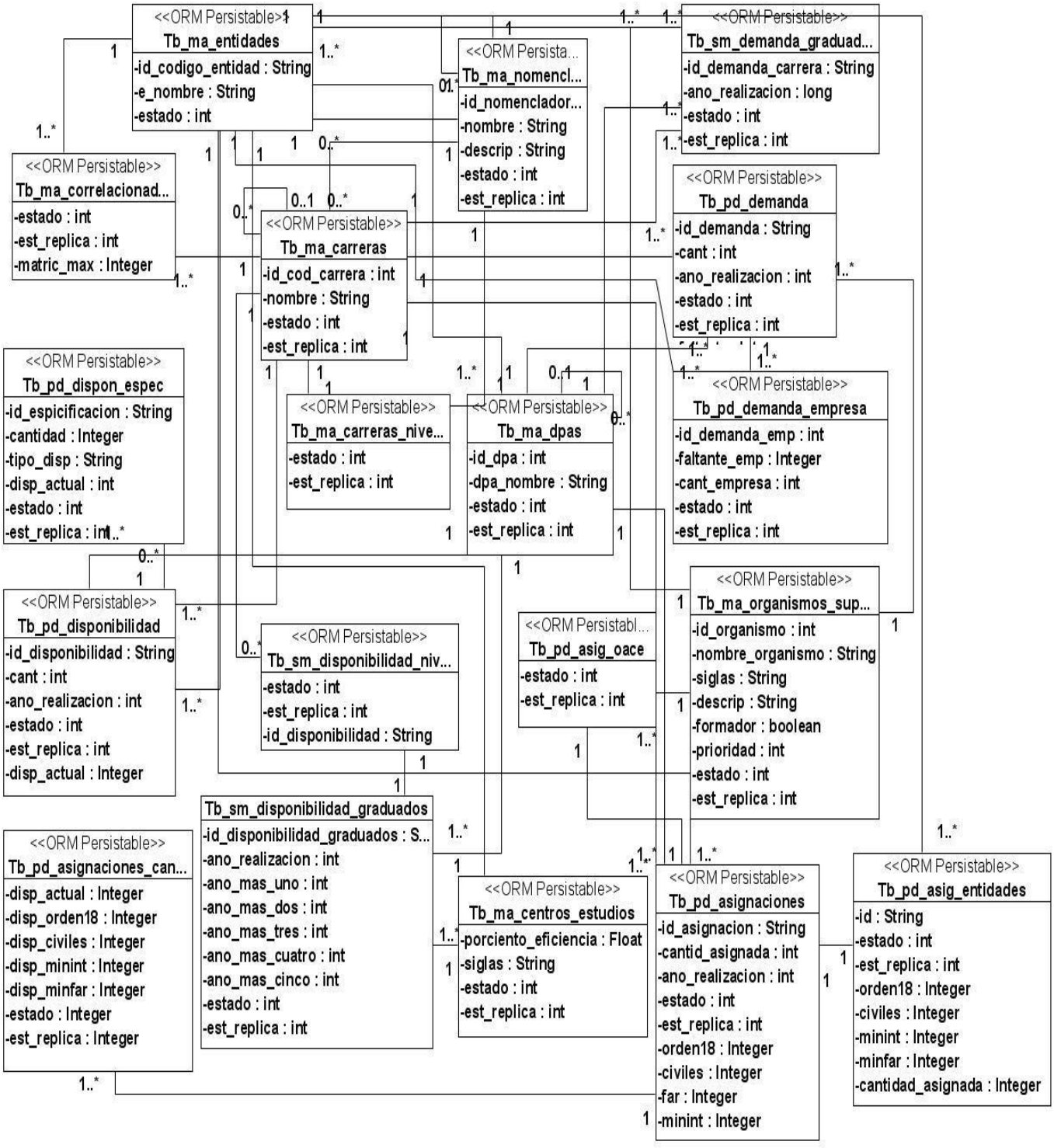

Figura #7 Modelo lógico de la base de datos.

3.4.2. Modelo físico.

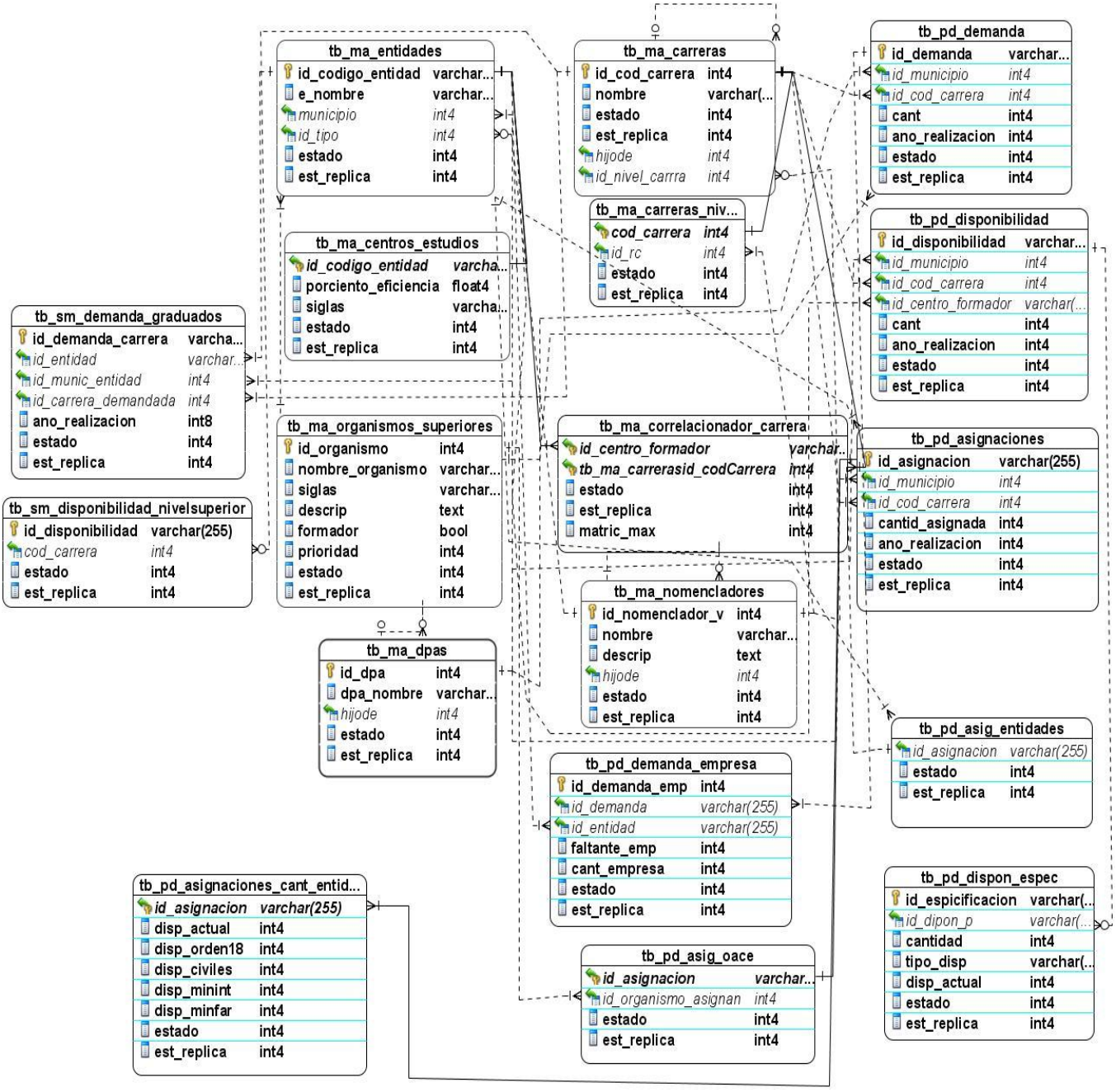

Figura #8 Modelo físico de la base de datos.

Donde las tablas nombradas tb_sm son del Sistema Matriz, las nombradas tb_ma son las del Módulo de Administración y las nombradas tb_pd son las del módulo Plan de Distribución de Graduados de Nivel Superior. Todas estas tablas son utilizadas por el sistema desarrollado.

3.5. Diagrama de despliegue

El modelo de despliegue representa los nodos que conforman la topología de hardware sobre la que se ejecuta el sistema y sus relaciones. Típicamente, los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP, IP, microondas, etc. A continuación se muestra la distribución física del sistema teniendo en cuenta la arquitectura del software y de hardware.

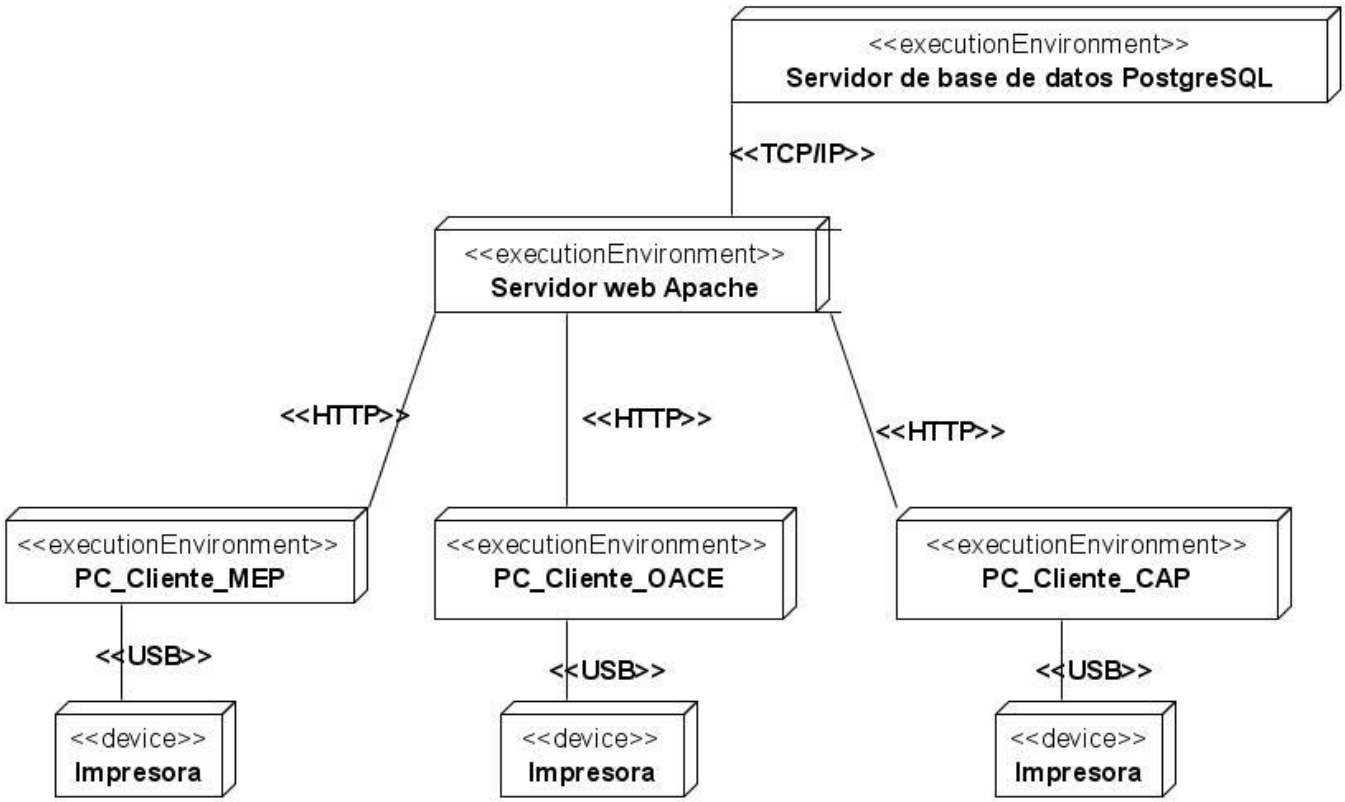

Figura #9 Diagrama de despliegue.

3.6. Conclusiones

Se finaliza la etapa de análisis y diseño del sistema obteniendo un modelo más detallado de la solución propuesta. Se realizaron los diagramas de clases del análisis y diseño de los casos de uso del sistema, así como los diagramas de colaboración y secuencia para cada una de las funcionalidades descritas, lo que permite una idea más específica del sistema que se propone. Además, se describe la arquitectura y los patrones que se tuvieron en cuenta para el diseño de los casos de uso, se realiza el modelo lógico y físico de la base de datos y por último se representan los nodos que conforman la topología de hardware sobre la que se ejecutará la aplicación.

Capítulo 4: Implementación y pruebas

4.1. Introducción

El presente capítulo abarca la fase de implementación y pruebas que comienza con los resultados obtenidos del modelo de análisis y continúa con la implementación del sistema en términos de componentes para terminar en la realización del modelo de diseño y la aplicación de las pruebas de caja negra en pos de garantizar la calidad del sistema. Se realizan también los diagramas de componentes.

4.2. Implementación.

Este flujo de trabajo enmarca el comienzo de la fase de Construcción. Esta fase tiene como propósito dejar listo un producto de software en su versión operativa inicial (versión beta). A esta versión le incumbe tener la calidad requerida para su uso y cumplir con los requerimientos de software determinados en el segundo capítulo.

4.2.1. Diagramas de componentes.

Un diagrama de componentes se representa como un grafo de componentes unidos por medio de relaciones de dependencia (compilación, ejecución), pudiendo mostrarse las interfaces que estos soporten. Cada componente representa una parte modular del sistema, desplegable y reemplazable que encapsula implementación y un conjunto de interfaces y proporciona la realización de los mismos.

Estos diagramas son usados para estructurar el modelo de implementación. El uso más importante de estos diagramas es mostrar la estructura de alto nivel del modelo de implementación, especificando:

- Los subsistemas de implementación y sus dependencias a la hora de importar código.
- Organizar los subsistemas de implementación en capas.

También se utilizan para mostrar las dependencias de compilación de los ficheros de código, relaciones de derivación entre ficheros de código fuente y ficheros que son resultados de la compilación, dependencias entre elementos de implementación y los correspondientes elementos de diseños que son implementados. (SoftwareII, 2007-2008)

A continuación se muestra el diagrama de componentes del caso de uso “Gestionar demanda”; el resto de los diagramas de componentes se encuentran en el Anexo #13.

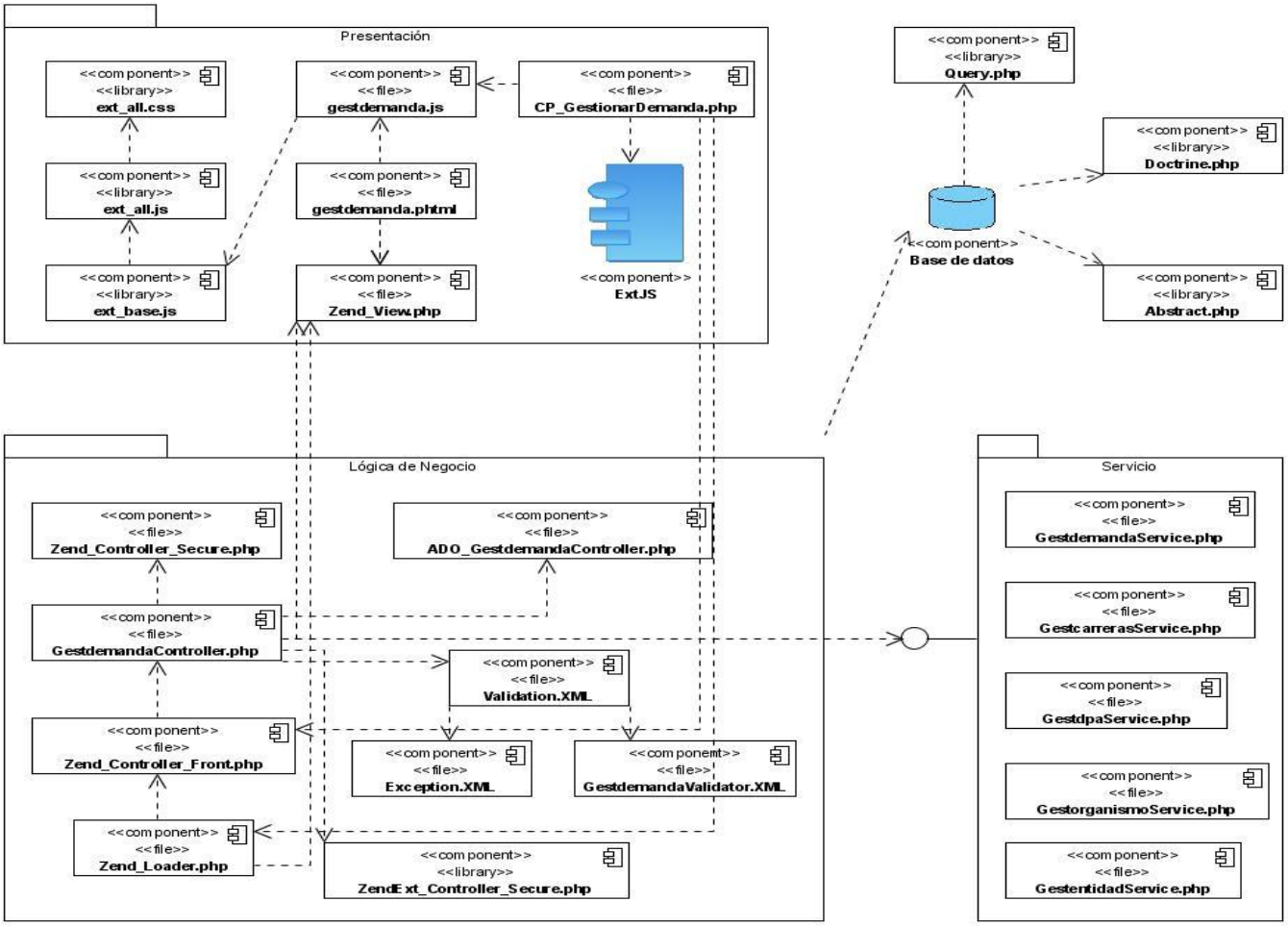

Figura #10 Diagrama de componente CU “Gestionar demanda”.

4.2.2. Tratamiento de errores

Para el tratamiento de errores se validan los datos que son introducidos por los usuarios al sistema. Se valida que el formato de los datos sea el esperado y que no se omita información de importancia para el procesamiento de la misma. En estos casos se informa al usuario mediante un mensaje que ha realizado alguna operación incorrecta.

4.2.3. Seguridad

Para garantizar la seguridad de la información, el proyecto GeForza cuenta con un Módulo de Administración que se encarga de la seguridad del sistema. El módulo Plan de Distribución de Graduados de Nivel Superior se integra a este módulo el cual le permite restringir los permisos a los usuarios según su rol, permitiendo a estos usuarios sólo ver las funcionalidades para las cuales obtengan permisos. En este caso se encuentra el funcionario del MEP el cual tendrá todos los permisos de la aplicación y los funcionarios de los OACE y los CAP los cuales sólo tendrán acceso a las funcionalidades de cada uno de sus organismos y consejos de administración provincial por separados. El sistema pedirá como primera acción para realizar sus operaciones un nombre de usuario y su contraseña garantizando que sólo las personas autorizadas puedan acceder al mismo. Estas contraseñas serán codificadas mediante una herramienta de codificación convirtiendo los caracteres en 0 y 1 para que nadie aunque tenga acceso a la base de datos pueda obtener información. Se realizarán trazas que registrarán información sobre las operaciones que se realizan, permitiendo conocer las acciones realizadas sobre la aplicación en un momento dado. Aparte de la seguridad que el módulo Plan de Distribución de Graduados de Nivel Superior obtiene del Módulo de Administración; en él se validan la entrada de datos incorrectos y las inyecciones SQL.

4.2.4. Estándares de codificación.

En la vista.

Para esta capa se definen como estándares los siguientes:

- El nombre de las clases debe comenzar con gest en el caso de los casos de uso gestionar, seguido por el nombre del caso de uso. Ejemplo: gestdemanda, gestdisponibilidad.
- El nombre de la clase debe comenzar con gen en el caso de los casos de uso generar, seguido por el nombre del caso de uso. Ejemplo: gennomencladores.
- Teniendo en cuenta el componente que se utilizará el nombre del mismo estará definido de la siguiente forma:

  Los botones se nombrarán: btn+Nombre. Ejemplo: btnAddDemanda.

  Los stores se nombrarán: st+Nombre. Ejemplo: stProv stOrg.

  Los combobox se nombrarán: cb+Nombre. Ejemplo: cbProvincia.

  Los texfields se nombrarán: txf+Nombre. Ejemplo: txfOrganismo.

  Los gridpanels se nombrarán: gp+Nombre. Ejemplo: gpNomenclador.

  Las ventanas se nombrarán: win+Nombre. Ejemplo: winAddDemanda.

En el negocio.

Para esta capa se definen:

- El nombre de la clase comenzará con Gest seguida por el nombre si el caso de uso es un gestionar y la palabra *Controller*. Ejemplo: GestdemandaController.
- El nombre de la clase comenzará con Gen seguida por el nombre si el caso de uso es un generar y la palabra *Controller*. Ejemplo: GennomencladoresController.
- El nombre de los atributos será en letra minúscula y en caso de ser más de una palabra se escribirán sin separación. Ejemplo: datosfinal.
- El nombre de las funciones será nombre seguido por la palabra *Action*. Ejemplo: cargarcomboprovinciaAction.
- Los servicios se nombrarán Gest seguido del nombre y la palabra *Service*. Ejemplo: GestdisponibilidadService.
- Las validaciones se nombrarán Gest seguido del nombre y la palabra *Validator*. Ejemplo: GestdisponibilidadValidator.

En el modelo.

Para esta capa se utilizan:

- El nombre de las entidades se define por tb seguido de las siglas del módulo y el nombre separando cada una de las palabras por un "_". Ejemplo: tb_sm_demanda_graduados.
- El nombre de los atributos será en letra minúscula y en caso de ser más de una palabra serán separadas por un "_". Ejemplo: id_entidad.

4.3. Pruebas del sistema.

La prueba de **caja negra** se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

La prueba de caja negra se centra principalmente en los requerimientos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requerimientos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requerimientos funcionales del sistema y ejercitándolos. (Software, 2005-2006)

Estas pruebas permiten encontrar:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Para preparar los casos de pruebas hace falta un número de datos que ayuden a la ejecución de estos casos y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos para el programa según si lo que se desea es hallar un error o probar una funcionalidad. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien.

En este trabajo se realizarán las pruebas de caja negra utilizando la técnica de la Partición de equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software. Además, permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubriendo de forma inmediata una clase de errores.

A continuación se muestra el caso de prueba del caso de uso “gestionar demanda”; para ver el resto ir al Anexo #13**.**

- CP #1 Nombre del caso de uso: Gestionar demanda. Fuente: elaboración propia.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC1: Adicionar demanda.	EC 1.1: Adicionar una demanda correctamente.	Se adiciona una nueva demanda al sistema.	-Se selecciona la opción “Adicionar”. -Llenar todos los campos. -Seleccionar “Aplicar”. -Seleccionar “Aceptar”. -Muestra un mensaje de confirmación “Creado satisfactoriamente” informando que la demanda se ha insertado con éxito.

	EC 1.2: Adicionar una demanda dejando campos en blanco.	No se adiciona una nueva demanda al sistema.	-Se selecciona la opción “Adicionar”. -Llenar los campos dejando al menos uno vacío. -Seleccionar “Aplicar”. -Seleccionar “Aceptar”. -Muestra el mensaje de error “Datos incorrectos”, notificando al usuario que los datos son incorrectos.
	EC 1.3: Adicionar una demanda pasando datos existentes.	No se adiciona una nueva demanda al sistema.	-Se selecciona la opción “Adicionar”. -Llenar todos los campos. -Seleccionar “Aplicar”. -Seleccionar “Aceptar”. -Muestra el mensaje “La demanda ya existe” indicando que la demanda ya existe.
	EC 1.4: Cancelar la operación.	No se adiciona una nueva demanda al sistema.	-Se selecciona la opción “Adicionar”. -Llenar los campos. -Seleccionar “Cancelar”.
SC2: Modificar demanda.	EC 2.1: Modificar una demanda correctamente.	Se modifica una demanda al sistema.	-Se selecciona una demanda. -Se selecciona la opción “Modificar”. -Se cambia el campo total. -Seleccionar “Aceptar”. -Muestra el mensaje “Modificado satisfactoriamente”.

	EC 2.2: Modificar demanda pasando datos existentes.	No se modifica la demanda al sistema.	-Se selecciona una demanda. -Se selecciona la opción “Modificar” -Se cambia el campo total por el mismo valor o no se cambia. -Seleccionar “Aceptar”. -Se muestra el mensaje “La demanda no se ha modificado”, indicando que no se ha modificado.
	EC 2.3: Modificar demanda sin poner valor.	No se modifica la demanda al sistema.	-Se selecciona una demanda. -Se selecciona la opción “Modificar” -Se cambia el campo total y no se le pone valor. -Seleccionar “Aceptar”. -Se muestra el mensaje “Datos incorrectos”, indicando que los datos son incorrectos.
	EC 2.4: Cancelar operación.	No se modifica la demanda al sistema.	-Se escoge una demanda. -Se selecciona la opción “Modificar”. - Se cambia el campo total. - Seleccionar “Cancelar”.
SC3: Eliminar demanda	EC 3.1: Eliminar demanda correctamente.	Se elimina una demanda al sistema.	-Se selecciona una demanda. -Se selecciona la opción “Eliminar”. -Muestra el mensaje de confirmación “¿Seguro que desea eliminar la demanda?” para eliminar. -Seleccionar la opción “Aceptar” -Muestra el mensaje “Se ha eliminado satisfactoriamente la demanda”.
	EC 3.2: Cancelar la operación.	No se elimina una demanda al sistema.	-Se selecciona una demanda. -Se selecciona la opción “Eliminar”. -Muestra el mensaje de confirmación “¿Seguro

			que desea eliminar la demanda?" para eliminar. -Se selecciona la opción "Cancelar".
SC4: Buscar demanda.	EC 4.1: Buscar demanda correctamente.	Se busca una demanda en el sistema.	-Selecciona "Buscar". -Se introducen los criterios por lo que desea buscar. -Selecciona "Aceptar". -Se muestran los datos de la demanda.
	EC 4.2: Limpiar el resultado de la búsqueda.	Regresa a la ventana principal.	-Selecciona "Buscar". -Se introducen los criterios por lo que desea buscar. -Selecciona "Aceptar". -Se muestran los datos de la demanda. -Selecciona Limpiar búsqueda.
	EC 4.3: Buscar demanda inexistente.	No se encuentra resultados de búsqueda de una demanda en el sistema.	-Selecciona "Buscar". -Se introducen los criterios por lo que desea buscar. -Se selecciona "Aceptar". -Muestra el mensaje de error "La búsqueda no obtuvo resultados" indicando que no se encontró ninguna demanda con esos parámetros.
	EC4.3: Buscar demanda sin pasar el criterio de búsqueda.	No se busca una demanda en el sistema.	-Selecciona "Buscar". -Se selecciona "Aceptar". -Muestra el mensaje de error "Debe seleccionar al menos un campo" indicando que se debe seleccionar al menos un campo.
	EC 4.4: Cancelar la operación.	No se busca una demanda en el sistema.	-Selecciona "Buscar". -Se introducen los criterios por lo que desea buscar. -Selecciona "Cancelar".
SC7: Cargar datos desde el Sistema Matriz.	EC 7.1 Cargar datos del Sistema Matriz	Se carga la demanda del Sistema Matriz	-Selecciona "Cargar datos desde Matriz". -Muestra un cartel "¿Seguro que desea cargar los datos desde el Sistema Matriz? Una vez realizada

	correctamente		esta acción, se perderán todos los datos actuales". -Selecciona "Aceptar". -Obtiene toda la información de la demanda que se encuentra en el Sistema Matriz para ese año.
	EC 7.2 Cargar datos del Sistema Matriz sin éxito.	No se carga la información deseada.	-Selecciona "Cargar datos desde Matriz". -Muestra un cartel ¿Seguro que desea cargar los datos desde el Sistema Matriz? Una vez realizada esta acción, se perderán todos los datos actuales. -Selecciona "Aceptar". -Muestra un mensaje de error "No hay información disponible"
SC8: Distribución por años.	EC 8.1: Distribución por años	Se muestra una lista con cinco años anteriores y cinco posteriores.	-Selecciona Distribución por años. -Selecciona un año. -Carga automáticamente la demanda del año seleccionado.

Tabla #8 Caso de prueba "Gestionar demanda".

4.4. Resultado de las pruebas.

Durante el desarrollo de las pruebas se realizó una prueba exploratoria y dos iteraciones, las cuales permitieron detectar un total de 28 errores, de los cuales 20 fueron de errores de funcionalidad y 8 de interfaz. Todos estos errores fueron corregidos por lo que se puede concluir que se obtuvo un buen resultado del flujo de trabajo de pruebas, logrando el buen funcionamiento de todas las funcionalidades del módulo, que era el objetivo principal.

4.5. Conclusiones.

Con la conclusión de este capítulo finaliza la implementación del módulo Plan de Distribución de Graduados de Nivel Superior, formados en el Curso Regular Diurno, para el Sistema Unificado de Gestión de Fuerza de Trabajo Calificada, junto con su fase de pruebas asegurando de esta forma la calidad del software.

Capítulo 5: Análisis de la factibilidad del sistema.

5.1. Introducción

En el presente capítulo se analiza la factibilidad del sistema, para ello se realiza la estimación del esfuerzo y el cálculo del tiempo de desarrollo del sistema en horas-hombre. Todo ello se obtiene como resultado de la aplicación del método de estimación por puntos de casos de uso. Se realiza un análisis del costo del producto y se exponen sus principales beneficios.

5.2. Estimación del esfuerzo

La estimación mediante el análisis de puntos de casos de uso es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. (Software, 2007-2008)

Existe una posibilidad de predecir el tamaño de un sistema a partir de las características de sus requerimientos, expresados en los casos de uso, por lo que es decisión de la universidad por las características de los proyectos la utilización de esta técnica. A continuación, se realiza la estimación sobre las descripciones detalladas de los casos de uso lo que permitirá estimar con exactitud el tamaño del sistema y el esfuerzo empleado para su desarrollo.

Pasos a seguir para la aplicación de este método:

El primer paso para la estimación consiste en el cálculo de los puntos de casos de uso sin ajustar.

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.

Factor de peso de los actores sin ajustar (UAW) “Este valor se calcula mediante un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema y en segundo lugar la forma en la que el actor interactúa con el sistema”.

Actores	Descripción	Complejidad	Factor de Peso
Funcionario	Una persona que interactúa con el sistema mediante una interfaz gráfica.	Complejo	3

Tabla #9 Factor de peso de los actores sin ajustar.

UAW = Sumatoria de la multiplicación de la cantidad de actores de un tipo por su factor de peso.
Cantidad de actores de tipo complejo: 1
UAW = 1 * 3
UAW = 3

Factor de peso de los casos de uso sin ajustar (UUCW): "Este valor se calcula mediante un análisis de la cantidad de casos de uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los casos de uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia y está representada por uno o más pasos del flujo de eventos principal del caso de uso, pudiendo existir más de una transacción dentro del mismo caso de uso".

Caso de Uso	Descripción	Complejidad	Factor de Peso
CUS Gestionar disponibilidad.	7 transacciones	medio	10
CUS Gestionar demanda.	7 transacciones	medio	10
CUS Gestionar asignación para organismos.	5 transacciones	medio	10
CUS Generar asignación para entidades.	4 transacciones	medio	10
GUS Mostrar nomencladores.	3 transacciones	simple	5
CUS Generar reportes	3 transacciones	simple	5

Tabla #10 Factor de peso de los casos de uso sin ajustar.

UUCW = ∑ CU*Peso

UUCW = Sumatoria de los casos de uso por su factor de peso.
UUCW = 4* 10 + 2 * 5.
UUCW = 50.

Finalmente, los puntos de casos de uso sin ajustar resultan:

UUCP = UAW + UUCW.

UUCP = 3 + 50.
UUCP = 53.

El segundo paso para la estimación consiste en el cálculo de los puntos de casos de uso ajustados. Para ello se utiliza la siguiente ecuación:

UCP = UUCP * TCF * EF donde,
UCP: Puntos de casos de uso ajustados.
UUCP: Puntos de casos de uso sin ajustar.
TCF: Factor de complejidad técnica.
EF: Factor de ambiente.

Factor de complejidad técnica (TCF): Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la siguiente tabla se muestra el significado y el peso de cada uno de estos factores:

Factor	Descripción	Peso	Valor asignado	Peso * Valor
T1	Sistema distribuido	2	4	8
T2	Objetivos de performance o tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	3	3
T4	Procesamiento interno complejo	1	3	3
T5	El código debe ser reutilizable	1	5	5
T6	Facilidad de instalación	0.5	4	2
T7	Facilidad de uso	0.5	5	2.5
T8	Portabilidad	2	4	8
T9	Facilidad de cambio	1	4	4
T10	Concurrencia	1	3	3
T11	Incluye objetivos especiales de seguridad	1	5	5
T12	Provee acceso directo a terceras partes	1	0	0
T13	Se requieren facilidades especiales de entrenamiento a usuarios	1	5	5

Tabla #11 Factor de complejidad técnica.

El Factor de complejidad técnica resulta:

TCF = 0.6 + 0.01 * Σ (Peso * Valor asignado)
TCF = 0.6 + 0.01 * (8+4+3+3+5+2+2.5+8+4+3+5+0+5)
TCF = 0.6 + 0.01 * 52.5.
TCF = 0.6 + 0.525.
TCF = 1.125.

Factor de ambiente (EF): Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente. El cálculo del mismo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

Factor	**Descripción**	Peso	Valor asignado	Peso * Valor
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	0	0
E2	Experiencia en la aplicación.	0.5	3	1.5
E3	Experiencia en orientación a objetos.	1	3	3
E4	Capacidad del analista líder.	0.5	5	2.5
E5	Motivación.	1	4	4
E6	Estabilidad de los requerimientos.	2	3	6
E7	*Personal part-time*.	-1	0	0
E8	Dificultad del lenguaje de programación.	-1	3	-3

Tabla #12 Factor de ambiente.

El Factor de ambiente resulta:

EF = 1.4 – 0.03 * Σ (Peso * Valor asignado)
EF = 1.4 – 0.03 * Σ (0+1.5+3+2.5+4+6+0-3)
EF = 1.4 – 0.03 * 14.
EF = 1.4 - 0.42.
EF = 0.98.

Finalmente, los puntos de casos de uso ajustados resultan:

UCP = UUCP * TCF * EF

UCP = 53 * 1.125 * 0.98.
UCP = 58.4325.

Paso número 3. El esfuerzo en horas-hombre viene dado por:

E = UCP * CF donde,
E: esfuerzo.
UCP: Puntos de casos de uso ajustados.
CF: Factor de conversión.

Para calcular CF Total EF = Cantidad EF < 3 (entre E1 – E6) + Cantidad EF > 3 (entre E7, E8)

Total EF = 1 + 0 Total
EF = 1
CF = 20 horas-hombre (si el Total EF ≤ 2)
CF = 28 horas-hombre (si el Total EF = 3 ó el Total EF = 4)
CF = abandonar o cambiar el proyecto (si el Total EF ≥ 5)

Para este proyecto 20 horas-hombre/punto de casos de uso, es decir, un punto de caso de uso toma 20 horas-hombre.

E = UCP * CF
E = 58.4325 * 20
E = 1168.65 horas-hombre

Este método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso, es decir, este método da la cantidad de horas-hombre que se necesitan para la implementación. Para una estimación más completa de la duración total del proyecto, hay que agregar a la estimación del esfuerzo obtenida por los puntos de casos de uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software.

Actividad	Porcentaje	Horas-hombre
Análisis	10.00 %	292.1625
Diseño	20.00 %	548.325
Programación	40.00 %	1168.65
Pruebas	15.00 %	438.24375
Sobrecarga (otras actividades)	15.00 %	438.24375
Total	100.00 %	2921.625

Tabla #13 Esfuerzo total (horas-hombres).

El esfuerzo total en horas hombre es de 2921.625. Para la etapa de análisis y diseño se requiere de un esfuerzo de 840.4875 horas-hombre, para la etapa de implementación se requiere de un esfuerzo de 1168.65 horas-hombre, si se considera que trabajan 2 personas, 60 horas como promedio en la semana, este sistema debe terminarse en 24.346875 semanas, lo que representa un aproximado de 25 semanas, que serán unos seis meses y medio.

5.3. Costo del proyecto

Ahora se van a analizar los costos asociados al desarrollo del producto.

C = **CHM** * **E**$_T$ donde,

C: Costo.

CHM: Cantidad de hombre-mes.

E$_T$: Esfuerzo total del proyecto.

Cálculo de la cantidad de hombres-mes (**CHM**)

CHM =**CH** * **SBM** donde:

CH: Cantidad de hombres trabajando en el proyecto.

SBM: Salario básico por hombres-mes.

SBM = $100

CH = 2. Por lo que:

CHM = 2 * 100

CHM = 200 pesos/mes.

Entonces:

C = 200 * 6.5

C = **$ 1300.**

Por tanto, para el desarrollo del producto se obtiene un costo de 1300 pesos.

5.4. Beneficios tangibles e intangibles

Tangibles

- El desarrollo de una aplicación web que realiza el Plan de Distribución de Graduados del Nivel Superior, formados en el Curso Regular Diurno, hacia sus centros laborales.

Intangibles

- Un aumento de la calidad en el proceso de distribución.
- Gestión de la información de forma rápida y segura, puesto que todo el flujo de información debe realizarse mediante la red, con usuarios autenticados.
- Una interfaz amigable que le permitirá al usuario comprender fácilmente su contenido.
- Gestión de información que es necesaria para el desarrollo del Plan de Distribución de Graduados de Nivel Superior como es el caso de la disponibilidad de graduados de Orden 18, cadetes del MINFAR, cadetes del MININT.
- Generación de gran cantidad de reportes de los que se podrá seleccionar información más específica.
- Impresión de informes desde la aplicación.
- Búsquedas rápidas y eficientes.
- Contará con una documentación que sustentará su desarrollo para posteriores versiones.

5.5. Conclusiones

En este capítulo se midió la factibilidad del sistema, así como la cantidad de horas-hombres necesarias para desarrollarlo entre 2 personas. Después de realizar la estimación del esfuerzo con el método de estimación por puntos de casos de uso, se llegó a la conclusión de que es factible desarrollar el módulo Plan de Distribución de Graduados de Nivel Superior. El cual trabajando 60 horas como promedio en la semana concluirá en 6 meses y medio. Se realizó además un análisis del costo del producto y de sus beneficios tangibles e intangibles.

Conclusiones generales

Luego de dar cumplimiento a los objetivos y tareas trazadas para lograr una investigación exitosa se arribó a las siguientes conclusiones:

- El profundo estudio de los sistemas de distribución en el mundo, en Cuba y en la Universidad de las Ciencias Informáticas, permitió definir que no existía ningún sistema que realizara el Plan de Distribución de Graduados de Nivel Superior como requería el cliente.
- Se definieron las tecnologías, metodologías, lenguajes y herramientas necesarias para el posterior desarrollo de la aplicación.
- El desarrollo del trabajo con la metodología RUP permitió documentar el mismo desde sus inicios, lo que servirá como base de estudio para futuras versiones, permitiendo de esta forma una comprensión más rápida y fácil de todas las etapas de concepción del módulo.
- La implementación de una aplicación web que permite realizar el Plan de Distribución de Graduados de Nivel Superior de forma más segura, contando con toda la información necesaria para la realización de dicho plan, mejorando su acceso a la información y su operabilidad.

Por todo lo antes expuesto se concluye que se le ha dado cumplimiento al objetivo general de la investigación: desarrollar una aplicación web que permita gestionar el Plan de Distribución de Graduados de Nivel Superior, formados en el Curso Regular Diurno.

Recomendaciones

Una vez cumplido con los objetivos de la investigación, teniendo en cuenta las experiencias obtenidas en la misma, se recomienda:

- Valorar la posibilidad de implementar una nueva funcionalidad en la que se automatice la distribución para las entidades.
- Valorar la posibilidad de mejorar la interfaz visual de la herramienta utilizada para la generación de reportes en la aplicación.
- Utilizar el presente trabajo como futuro material de consulta para la preparación de nuevas generaciones, sugiriendo también la bibliografía documentada.

Bibliografía consultada

Ben Collins-Sussman, Brian W. Fitzpatrick y Michael Pilato. Control de versiones Open Source de siguiente generación- Subversion. [En línea] [Citado el: 23 de 2 de 2010.] http://svnbook.red-bean.com/index.es.html.

Ciberaula. 2010. Ciberaula. Una Introducción a APACHE. [En línea] 2010. [Citado el: 30 de 2 de 2010.] http://linux.ciberaula.com/articulo/linux_apache_intro.

CIMEX., Manual de usuario. Corporación. 2010. Manual de usuario-Corporación CIMEX. Camaguey : GERENCIA CIMEX, 2010.

Cliente/Servidor, El Modelo. EL MODELO CLIENTE/SERVIDOR. [En línea] [Citado el: 21 de 2 de 2010.] http://agamenon.uniandes.edu.co/~revista/articulos/cliser.html.

Corzo, Giancarlo. 2008. Desarrollo en Web- ExtJS lo bueno, lo malo y lo feo. [En línea] 2008. [Citado el: 24 de 2 de 2010.] http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/.

ERP, UCI-Proyecto. 2010. Manual de usuarios del instalador de SAUXE. Ciudad de La Habana : s.n., 2010.

Framework, Zend. Zend Framework. [En línea] [Citado el: 23 de 2 de 2010.] http://zendframework.com/manual/en/learning.quickstart.intro.html.

gestorweb.com. 2003. GestorWeb- La aplicación web empresarial. [En línea] 16 de 6 de 2003. [Citado el: 20 de 2 de 2010.] http://www.gestorweb.com/docu/webapps_arti.html.

GNU. 2010. GNU Operating System. La Definición de Software Libre. [En línea] 27 de 05 de 2010. http://www.gnu.org/philosophy/free-sw.es.html.

Grady Booch, Jim. Rumbaugh y Ivar Jacobson. El Lenguaje Unificado de Modelado. [En línea] [Citado el: 20 de 2 de 2010.] http://elvex.ugr.es/decsai/java/pdf/3E-UML.pdf.

Ivar Jacobson, Grady Booch, James Rumbaugh. El Proceso Unificado de Desarrollo de Software. [En línea] [Citado el: 10 de 3 de 2010.] http://bibliodoc.uci.cu/pdf/reg00060.pdf.

Libre, Conferencia Internacional de Software. 2006. Software Libre II Conferencia Internacional. [En línea] 21 de 1 de 2006. [Citado el: 11 de 2 de 2010.] http://malaga06.opensourceworldconference.com/es/modules/news/article.php?storyid=354.

MEP. 2009. Indicaciones Metodológicas VIII Fuerza de Trabajo Calificada. Ciudad de La Habana : s.n., 2009.

NetBeans. Información de la versión de NetBeans IDE 6.8. [En línea] [Citado el: 23 de 2 de 2010.] http://netbeans.org/community/releases/68/index_es.html.

Netcommerce. 2009. Netcommerce. Aplicación web. [En línea] 2009. [Citado el: 22 de 2 de 2010.] http://www.netcommerce.com.mx/desarrollo.asp.

Pablo Cababie, Facundo Cancelo, Daniela López De Luise. 2008. Sistema automático para Asignación de aulas y distribución de espacios. [En línea] 2008. [Citado el: 20 de 2 de 2010.] http://www.palermo.edu/ingenieria/downloads/2008_wicc_gdarim_v3.pdf.

PostgreSQL, El equipo de desarrollo de. Manual del usuario de PostgreSQL. [En línea] [Citado el: 23 de 2 de 2010.] http://palomo.usach.cl/Docs/postgres/Postgres-User.pdf.

Rojas, Juan Carlos Olivares. 2007. Patrones de Diseño. [En línea] 2007. [Citado el: 6 de 4 de 2010.] http://antares.itmorelia.edu.mx/~jcolivar/courses/dp07a/patrones.pdf.

S., Christian Van Der Henst. 2007. Maestros de la web. ¿Qué es el PHP? [En línea] 2007. [Citado el: 23 de 2 de 2010.] http://www.maestrosdelweb.com/editorial/phpintro/.

Sanchez, María A. Mendoza. 2004. informatizate- Metodologías de Desarrollo de Software. [En línea] 7 de 6 de 2004. [Citado el: 22 de 2 de 2010.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

Sánchez, Rosario Garza Río y Caridad González. 2004. Revista investigación operacional. DRSOFT: un soporte computacional para el diseño de rutas de distribución. [En línea] 2004. [Citado el: 21 de 2 de 2010.] http://rev-inv-ope.univ-paris1.fr/files/25304/IO-25304-7.pdf. 3.

Software, UCI- Ingeniería de. 2005-2006. Entormo virtual de Aprendizaje- Ingeniería de Software II, Conferencia 5, Flujo de trabajo de pruebas, pruebas de caja negra. [En línea] 2005-2006. [Citado el: 13 de 5 de 2010.] http://eva.uci.cu/mod/resource/view.php?id=2241&subdir=/Conferencias_IS2_05-06.

Software, UCI-Ingeniería de. 2007-2008. Entorno Virtual de Aprendizaje- Ingeniería de software, CTP Estimación. [En línea] 2007-2008. [Citado el: 20 de 3 de 2010.] http://eva.uci.cu/mod/resource/view.php?id=22433.

Software, UCI-Ingeniería de. 2007-2008. Entorno Virtual de Aprendizaje-Ingeniería de Software. Conferencia 8, Arquitectura y patrones de diseño. [En línea] 2007-2008. [Citado el: 5 de 4 de 2010.] http://eva.uci.cu/mod/resource/view.php?id=2241&subdir=/Conferencias_IS1_2007-2008.

Software I, UCI-Ingeniería de. 2007-2008. Entorno Virtual de Aprendizaje. Asignatura: Ingeniería de Software I. Introducción a la Ingeniería de Software. [En línea] 2007-2008. http://eva.uci.cu/mod/resource/view.php?id=2241&subdir=/Conferencias_IS1_2007-2008.

Software II, UCI-Ingeniería de. 2007-2008. Entorno Virtual de Aprendizaje- Ingeniería de Software II, Conferencia 3, Fase de Elaboración, Flujo de trabajo implementación. [En línea] 2007-2008. [Citado el: 10 de 5 de 2010.] http://eva.uci.cu/mod/resource/view.php?id=2241&subdir=/Conferencias_IS2_05-06.

TortoiseSVN. TortoiseSVN the cooles Interface to (Sub) Version Control. [En línea] [Citado el: 23 de 2 de 2010.] http://tortoisesvn.net/.

Tripix.net. 2006. Tripix.net- Axure. Software para Wireframes. [En línea] 2006. [Citado el: 23 de 2 de 2010.] http://www.tripix.net/2006/06/05/axure-software-para-wireframes/.

UCI, Lianet Liben Martínez. 2010. Subsistema de Gestión de pregrado que pertenece al proyecto Gestión Universitaria. Descripción del algoritmo de balanceo. 2010.

UCI, Yohana Baró Montenegro y Reynaldo Hernández Rodríguz. 2007. Sistema de Reservación del Pase Macivo. Ciudad de La Habana : s.n., 2007.

UCI-Patdsi. 2010. Manual de usuario Patdsi. Ciudad de La Habana : s.n., 2010.

Valdés, Damián Pérez. 2007. Maestros del web. ¿Qué es Javascript? [En línea] 2007. [Citado el: 23 de 2 de 2010.] http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/.

Valencia, Universidad Politécnica de. OPTIHPER Asignación Optimizada de Horarios al Personal. [En línea] Universidad Politécnica de Valencia. [Citado el: 20 de 2 de 2010.] http://users.dsic.upv.es/grupos/gps/optihper/index.html.

vives?, El mundo informático y tú en que mundo. 2006. El mundo informático y tú en que mundo vives?- Patrones Grasp (Craig Larman) Parte I. [En línea] 17 de 8 de 2006. [Citado el: 3 de 4 de 2010.] http://jorgesaavedra.wordpress.com/2006/08/17/patrones-grasp-craig-larman/.

Glosario de términos

API: Interfaz de Programación de Aplicaciones de su significado en inglés *Application Programming Interface,* es un conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Distribución: es la acción y efecto de distribuir. Aplicado a diferentes campos.

DOM: Modelo en Objetos para la Representación de Documentos de su significado en inglés *Document Object Model*, es una interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos y una interfaz estándar para acceder a ellos y manipularlos.

DHTML: HTML Dinámico de su significado en inglés *Dynamic HTML*, es un conjunto de técnicas que permiten crear sitios web interactivos utilizando una combinación de lenguaje HTML estático, un lenguaje interpretado en el lado del cliente (como JavaScript), el lenguaje de hojas de estilo en cascada (CSS) y la jerarquía de objetos DOM.

EC: Escenarios.

HTTP: Protocolo de Transferencia de Hipertexto de su significado en inglés *HyperText Transfer Protocol*. Es el protocolo usado para las transferencias de la web.

***Layouts*:** es la ordenación y colocación de todos los elementos que componen una página web, es decir, textos, imágenes, tablas, gráficos, colores, tipo de letra, etc.

MEP: Ministerio de Economía y Planificación.

SC: Secciones.

Web: es el término más usado para referirse al *World Wide Web* que es la red mundial de páginas o documentos de texto entrelazados. Un documento entrelazado no es más que un documento que contiene enlaces a otros documentos o páginas de texto.

Widgets: pequeña aplicación o módulo que realiza una función concreta, generalmente de tipo visual, dentro de otras aplicaciones o sistemas operativos.