

Universidad de las Ciencias Informáticas
Facultad 1



Diseño e implementación del módulo Sistema Matriz 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: Dayana César Fernández.
Alexander Norat Fernández.

Tutores: Ing. Yanet Casado Maceo.
Ing. Yisell Ávila Portales.

Ciudad de La Habana, Junio 2010.



“El aspecto fundamental en el cual la juventud debe señalar el camino es precisamente en el aspecto de ser vanguardia en cada uno de los trabajos que le compete.”

Che

A:
Mi familia, en especial a mis padres.

Dayana.

A:
*A mi familia, mis padres y en especial
a mi esposa.*

Alexander.

Agradezco:

A mi mamá, ante todo, por ser mi amiga, mi confidente, por sus consejos, por todo su apoyo, en fin, porque tener una madre mejor, sería imposible.

A mi padre del alma, Cansino, por su cariño, su apoyo incondicional, por estar siempre ahí para lo que necesite, sé que puedo confiar en ti, te quiero.

A mi novio, por todo su cariño, por comprenderme y ayudarme en los momentos difíciles.

A mis suegros, porque han sido como unos padres para mí, por todo su cariño y sus atenciones.

A Pedro, la China, Dayán, Mercy, Milagros y Eufemia por todo su cariño y su ayuda.

A mis tutoras por todo el apoyo y la preocupación.

A mi compañero de tesis, por todo su trabajo durante este período de desarrollo de la investigación.

A Javier por ayudarnos en el desarrollo de la tesis, sin ti no hubiera sido posible.

A Alberto por estar disponible siempre que lo necesito, por toda su ayuda.

A Yaima, María, Yailín, Grethell y Elizabeth por la convivencia de estos cinco años.

A todos mis compañeros del proyecto en especial a Aneyty, Alain, Evelio, Pedro, Enmanuel, Yuri y Carlos por aclararme tantas dudas.

A toda mi familia por confiar en mí.

En fin, a todos los que de una forma u otra me han ayudado en el transcurso de estos 5 años y han contribuido al desarrollo de este trabajo.

Dayana.

Agradezco a:

Mi compañera de tesis Dayana: por su dedicación, comprensión y por haber sido junto a mí la persona responsable de la creación de esta investigación que tanto esfuerzo nos ha costado.

A mis tutoras Yanet y Yisell: por su disposición y ayuda en todo momento.

A mis compañeros del proyecto: por haber sido un excelente grupo de trabajo y en especial a Javier porque contribuyó a la realización de esta investigación.

A mis compañeros del antiguo grupo 1104: porque junto a ellos viví los mejores momentos como grupo de la universidad.

A mis amigos Yasier, Sailin, Pedro, María y Yelina: por estar presentes en los momentos en los que necesité apoyo y porque son los que verdaderamente significan amistad.

A mis hermanos Alain y Anaisy: pues aunque nos fajamos siempre, nos queremos como la familia que somos.

A mis abuelos Rosa, Sara y Jorge: por sus consejos y preocupación a lo largo de mi vida.

A mi tío Pepe: por ser mi segundo padre y estar presente en mis buenos y malos momentos.

A la familia de mi esposa, mis suegros y Ulises (Toto): porque desde que soy parte de ellos me han hecho sentir la persona más especial.

A mi esposa Dayana: porque desde que la conocí mi vida ha tomado sentido, me ha enseñado a madurar y gracias a ella aprendí que el amor de película existe.

A mi padre: por su ejemplo, sabiduría y por su preocupación hacia mi trabajo de diploma.

A mi madre: porque ha sido mi faro, mi guía, la responsable de esta persona que hoy soy, porque me ha apoyado en mis momentos difíciles, y hace posible que día a día mi vida siga adelante.

Alexander.

Declaración de autoría

Declaramos que somos los únicos autores del trabajo titulado:

Diseño e implementación del módulo Sistema Matriz para el Sistema Unificado de Gestión de Fuerza de Trabajo Calificada.

Autorizamos a la Universidad de las Ciencias Informáticas para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año 2010.

Dayana César Fernández.

Alexander Norat Fernández.

Ing. Yanet Casado Maceo.

Ing. Yisell Ávila Portales.

Opinión del tutor del trabajo de diploma

Título: Diseño e implementación del módulo Sistema Matriz para el Sistema Unificado de Gestión de Fuerza de Trabajo Calificada.

Autores: Dayana César Fernández.

Alexander Norat Fernández.



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

cc Manuel Puerta La Rosa



Resumen

El departamento Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación (MEP) cuenta con una aplicación para la gestión de la información necesaria para la realización del Plan de Ingreso a la Educación Superior y el Plan de Distribución de graduados de la Educación Superior. Dicha aplicación presenta varias deficiencias en cuanto a seguridad y operabilidad, lo cual compromete la calidad e inmediatez de la realización de estos planes, que son de gran importancia a nivel nacional. El objetivo principal de este trabajo es implementar un sistema que permita gestionar la información de los graduados, de forma rápida, segura y organizada, para la elaboración de los planes en el departamento Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación. Para cumplir los objetivos planteados se realizó el análisis, diseño e implementación de una aplicación web que permite gestionar la información de forma organizada, rápida y segura, controlando el intercambio de información mediante usuarios autenticados a través de una interfaz amigable y fácil de operar. Para su desarrollo se estudiaron los procesos llevados a cabo en el departamento Fuerza de Trabajo Calificada del MEP relacionados con la gestión de la información necesaria para la realización de los planes de ingreso y distribución, además de estudiar a profundidad las herramientas, tecnologías y lenguajes que fueron seleccionados por la dirección del proyecto Fuerza de Trabajo Calificada.

Palabras claves: información, planes, aplicación, seguridad, operabilidad.

Introducción	1
Capítulo I: Fundamentación teórica.....	5
1.1. Introducción.....	5
1.2. Sistemas de gestión de información.....	5
1.2.1. Sistemas de gestión de información vinculada a la fuerza de trabajo calificada en el mundo.	5
1.2.2. Sistemas de gestión de información vinculada a la fuerza de trabajo calificada en Cuba.	7
1.2.3. Sistemas de gestión de información vinculada a la fuerza de trabajo calificada en la UCI.	9
1.3. ¿Por qué se decide implementar un nuevo sistema?	9
1.4. Tendencias actuales al software libre.....	10
1.5. Metodología, herramientas y tecnologías utilizadas.....	11
1.5.1. Aplicación web.....	11
1.5.2. Metodología de desarrollo de software: RUP.....	12
1.5.3. Arquitectura: Cliente/Servidor, Modelo Vista Controlador.....	13
1.5.4. Lenguaje de modelado visual: UML 2.0.....	15
1.5.5. Herramienta para el modelado: Visual Paradigm for UML 6.1.....	16
1.5.6. Lenguajes de programación: JavaScript 2.0 y PHP 5.0.....	16
1.5.7. Plataforma de desarrollo: NetBeans 6.8.....	18
1.5.8. Marcos de trabajo: ExtJS, Zend Framework, Doctrine.....	18
1.5.9. Servidor web: Apache 2.0.....	21
1.5.10. Sistema gestor de base de datos: PostgreSQL 8.3.....	22
1.5.11. Generador de reportes dinámico: PATDSI.....	22
1.6. Fundamentación de la metodología, herramientas y tecnologías a utilizar.....	23
1.7. Conclusiones.....	25
Capítulo II: Características del sistema.....	26
2.1. Introducción.....	26
2.2. Modelo de dominio.....	26
2.3. Diagrama de clases del modelo de dominio.....	27
2.4. Definición de las clases del modelo de dominio.....	27
2.5. Especificación de los requerimientos de software.....	28
2.5.1. Requerimientos funcionales.....	28
2.5.2. Requerimientos no funcionales.....	34
2.6. Modelo del sistema.....	37
2.7. Actor del sistema.....	37
2.8. Diagrama de casos de uso del sistema.....	38
2.9. Especificación de los casos de uso del sistema.....	38
2.10. Análisis de la factibilidad del sistema.....	40
2.10.1. Estimación del esfuerzo.....	40
2.10.2. Estimación del tiempo de desarrollo.....	45
2.10.3. Análisis de costos.....	46
2.10.4. Beneficios tangibles e intangibles.....	47
2.11. Conclusiones.....	48
Capítulo III: Análisis y diseño.....	49

3.1. Introducción	49
3.2. Análisis	49
3.2.1. Diagrama de clases del análisis.....	49
3.3. Diseño.	50
3.3.1. Patrones de diseño.....	50
3.3.2. Diagrama de clases del diseño.....	53
3.4. Modelo de datos.	59
3.4.1. Modelo físico.....	60
3.5. Modelo de despliegue.....	61
3.5.1. Diagrama de despliegue.....	61
3.6. Conclusiones.....	61
Capítulo IV: Implementación y pruebas.....	62
4.1. Introducción.....	62
4.2. Implementación.....	62
4.2.1. Diagramas de componentes.....	62
4.2.2. Tratamiento de errores.....	64
4.2.3. Seguridad.....	64
4.2.4. Estándares de codificación.....	64
4.3. Pruebas.....	65
4.3.1. Método de prueba.....	66
4.3.2. Casos de prueba.....	66
4.4. Resultados de las pruebas.....	70
4.5. Conclusiones.....	71
Conclusiones generales.....	72
Recomendaciones.....	73
Bibliografía.....	74
Glosario de términos.....	76

Tabla 2.1. Descripción de los actores del sistema.	38
Tabla 2.2. Descripción abreviada del CU Gestionar disponibilidad de noveno grado.	38
Tabla 2.3. Descripción abreviada del CU Gestionar disponibilidad de nivel medio.	39
Tabla 2.4. Descripción abreviada del CU Gestionar disponibilidad de nivel superior.	39
Tabla 2.5. Descripción abreviada del CU Gestionar existencia de empleados.	39
Tabla 2.6. Descripción abreviada del CU Gestionar demanda de graduados.	39
Tabla 2.7. Descripción abreviada del CU Gestionar proyección de la demanda.	39
Tabla 2.8. Descripción abreviada del CU Generar reportes.	40
Tabla 2.9. Descripción abreviada del CU Importar XML.	40
Tabla 2.10. Descripción abreviada del CU Exportar XML.	40
Tabla 2.11. Factor de peso de actores sin ajustar.	41
Tabla 2.12. Factor de peso de los casos de uso sin ajustar.	41
Tabla 2.13. Factor de complejidad técnica.	43
Tabla 2.14. Factor de ambiente.	44
Tabla 2.15. Distribución del esfuerzo.	45
Tabla 4.1. Caso de prueba CU Gestionar demanda de graduados.	69

Figura 1. Modelo de dominio.....	27
Figura 2. Diagrama de casos de uso del sistema.....	38
Figura 3. Diagrama de clases del análisis CU Gestionar demanda de graduados.	50
Figura 4. Diagrama de clases del diseño CU Gestionar demanda de graduados.....	54
Figura 5. Diagrama de clases del diseño CU Gestionar demanda de graduados (vista).	55
Figura 6. Diagrama de clases del diseño CU Gestionar demanda de graduados (controlador).....	56
Figura 7. Diagrama de clases del diseño CU Gestionar demanda de graduados (modelo).	57
Figura 8. Diagrama de clases del diseño CU Gestionar demanda de graduados (servicios).....	58
Figura 9. Modelo físico de la base de datos.	60
Figura 10. Diagrama de despliegue.	61
Figura 11. Diagrama de componentes CU Gestionar demanda de graduados.	63
Figura 12. Resultados de las pruebas.....	70

Introducción

El departamento Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación (MEP) tiene como misión lograr el máximo equilibrio entre la formación y el empleo de los estudiantes de la Educación Técnica y Profesional, así como los de la Educación Superior acorde a las necesidades reales de los territorios del país y atendiendo a las indicaciones del nivel superior en relación a esta temática. Sus funciones son de carácter global, que trascienden e involucran a muchos factores externos, y cuyos resultados tienen implicaciones económicas, políticas y sociales. (CALIFICADA 2008)

Dentro de las tareas que desarrolla el departamento Fuerza de Trabajo Calificada se encuentran: la elaboración del Plan de Ingreso a la Educación Superior del Curso Regular Diurno (CRD) y el Plan de Distribución de graduados de nivel superior formados en el Curso Regular Diurno.

La elaboración del Plan de Ingreso a la Educación Superior del Curso Regular Diurno permite formar nuevos profesionales que al graduarse cubrirán las plazas de los trabajadores que culminen su vida laboral, evitando así, la falta de profesionales en las empresas. De la misma forma, la elaboración del Plan de Distribución de graduados de nivel superior formados en el Curso Regular Diurno constituye una tarea muy importante para el país pues permite la ubicación laboral de los graduados, de acuerdo a su lugar de residencia, y de esta forma se garantiza que todas las regiones del país cuenten con los recursos laborales necesarios de acuerdo a las necesidades socioeconómicas.

El departamento Fuerza de Trabajo Calificada (FTC), cuenta con una aplicación para la gestión de la información necesaria para la realización del Plan de Ingreso a la Educación Superior del Curso Regular Diurno y el Plan de Distribución de graduados de nivel superior formados en el Curso Regular Diurno, pero, a raíz de los cambios en la dirección del país, se le asignaron nuevas responsabilidades al MEP y a su vez, el departamento Fuerza de Trabajo Calificada asumió nuevas tareas, por ejemplo, la realización del Plan de Continuidad de Estudios de los estudiantes de noveno grado. Debido a que la aplicación no estaba prevista para estos cambios, ya no cubre las necesidades del cliente pues carece de los clasificadores y nomencladores necesarios para cumplir con la realización de esta tarea. Por otra parte, este sistema fue programado en Visual FoxPro 5.0 bajo WINDOW NT, tiene varios errores y su operabilidad se dificulta bastante, además, tiene poca seguridad ya que todo el intercambio de información se realiza mediante disquetes de 3½ pulgadas y no se tiene control de las personas que acceden a la

misma. Por ejemplo, el uso de estos disquetes implica que el proceso no se realice con la suficiente inmediatez y rapidez.

Al realizar el trabajo con todas estas dificultades se ve comprometida la calidad del Plan de Ingreso a la Educación Superior del Curso Regular Diurno, así como el Plan de Distribución de graduados de nivel superior formados en el Curso Regular Diurno y el Plan de Continuidad de Estudios de los estudiantes de noveno grado, por lo que se plantea la siguiente **situación problemática**: difícil operabilidad, falta de nomencladores y clasificadores necesarios para la realización de tareas, poca seguridad y errores en el sistema actual.

Teniendo en cuenta lo anteriormente expuesto, el **problema científico** queda formulado de la siguiente manera: ¿Cómo garantizar que el departamento Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación desarrolle con rapidez, organización y seguridad la gestión de información de graduados?

El **objeto de estudio** lo constituyen los procesos de gestión de la información.

Para dar solución a la problemática planteada anteriormente se ha definido como **objetivo general** de la investigación: implementar un sistema que permita gestionar la información de los graduados, de forma rápida, segura y organizada, para la elaboración de los planes en el departamento Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación.

Derivándose los siguientes **objetivos específicos**:

- Gestionar la información que se requiere para la elaboración del Plan de Ingreso a la Educación Superior del Curso Regular Diurno.
- Gestionar la información necesaria para la confección del Plan de Distribución de graduados de la Educación Superior.
- Gestionar la información que se precisa para la creación del Plan de Continuidad de Estudios de los estudiantes de noveno grado.

Se define como **campo de acción** el módulo Sistema Matriz para la gestión de información de graduados.

Para cumplir los objetivos trazados se realizarán las siguientes tareas:

- Realizar un estudio del estado del arte de los sistemas de gestión de información.

- Estudiar la arquitectura propuesta por el proyecto.
- Definir los requerimientos del módulo Sistema Matriz.
- Realizar el análisis y diseño del módulo Sistema Matriz.
- Realizar la implementación del módulo Sistema Matriz.
- Evaluar los resultados obtenidos.

Se plantea como **hipótesis**: con la implementación del módulo Sistema Matriz ajustado a los nuevos clasificadores y nomencladores, se mejorará el proceso de gestión de información, en cuanto a seguridad y operabilidad.

Variable independiente: Sistema Matriz.

Variable dependiente: proceso de gestión de información vinculada a la fuerza de trabajo calificada en el MEP.

Para consultar la operalización de las variables ir al Anexo 1.

En la investigación se usó un sistema de métodos conformado por métodos teóricos y métodos empíricos.

En relación con los **métodos teóricos** se citan:

- Analítico-Sintético: Se realizó un estudio al actual sistema y documentos relacionados permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio.
- Análisis Histórico-Lógico: Se realizó un estudio del sistema actual para definir los principales errores y deficiencias con el fin de perfeccionar las vulnerabilidades.
- Modelación: Permitted representar la realidad mediante modelos que facilitan el estudio de los fenómenos.

Por otra parte, se utilizaron los métodos empíricos siguientes:

- Encuestas y entrevistas: Se realizaron entrevistas a cada uno de los integrantes del departamento Fuerza de Trabajo Calificada del MEP con el objetivo de conocer las principales fallas del sistema actual así como las nuevas funcionalidades que requieren.
- Observación: Se percibió la realidad directamente; de acuerdo a los resultados obtenidos, se

eliminaron las vulnerabilidades.

- Estudio documental: Se hizo 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 del sistema.

Como **posible resultado** se obtendrá un sistema capaz de gestionar la información para el departamento Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación de forma rápida, garantizando los niveles necesarios de seguridad.

Se propone la implementación de un sistema capaz de garantizar mayor seguridad, pues el intercambio de información se realizará mediante la red y con usuarios autenticados. Se hará la propuesta de una interfaz más sencilla, fácil de usar para que el trabajo con dicho sistema sea más rápido y seguro; así como se solucionarán los errores encontrados en el sistema actual.

La investigación está estructurada en cuatro capítulos:

- Capítulo I: Fundamentación teórica, incluye el estado del arte de los sistemas de gestión de información vinculados al objeto de estudio a nivel internacional, nacional y en la Universidad de Ciencias Informáticas (UCI), así como una caracterización de las herramientas, tecnologías y metodologías a utilizar.
- Capítulo II: Características del sistema, muestra el modelo de dominio y la especificación de los requisitos funcionales y no funcionales, definiéndose los casos de uso del sistema, a partir de los mismos, se realiza el análisis de la factibilidad del sistema.
- Capítulo III: Análisis y diseño, aborda los detalles relacionados con el análisis y diseño del módulo Sistema Matriz, representándose un diagrama de clases, tanto para el análisis como para el diseño de cada caso de uso, los diagramas de colaboración para el análisis y los diagramas de secuencia para el diseño de cada una de las secciones de los casos de uso, además de los modelos lógico y físico de la base de datos y el diagrama de despliegue.
- Capítulo IV: Implementación y pruebas, se realiza el diagrama de componentes para cada caso de uso identificado y las pruebas al sistema.

El análisis económico de la investigación está disponible en el Anexo 2.

Capítulo I: Fundamentación teórica.

1.1. Introducción.

En el presente capítulo se realizará un estudio de la situación actual de los sistemas de gestión de información relacionada con los recursos humanos a nivel internacional, nacional y en la universidad, así como un estudio de las principales herramientas, tecnologías y metodologías de desarrollo de software que permitirán la implementación del sistema.

1.2. Sistemas de gestión de información.

Se define como sistema de información: "la combinación de recursos humanos y materiales que resultan de las operaciones de almacenar, recuperar y usar datos con el propósito de una gestión eficiente en las operaciones de las organizaciones". (CAÑAVATE)

Los sistemas de información se caracterizan por la disponibilidad de información cuando es necesario y por los medios adecuados, el suministro de información de manera selectiva, variedad en la forma de presentación de la información, exactitud, flexibilidad, capacidad de adaptación, fiabilidad, seguridad (protección contra pérdidas) y amigabilidad para el usuario.

1.2.1. Sistemas de gestión de información vinculada a la fuerza de trabajo calificada en el mundo.

➤ SAP ERP Human Capital Management.

SAP ERP Human Capital Management (Administración del capital humano) permite a los empleados actualizar información personal y controlar procesos administrativos. Sus funcionalidades principales permiten optimizar los procesos de administración de capital humano, proporcionar acceso en tiempo real a la información que acelera la toma de decisiones por parte del personal, asignar las personas correctas a los proyectos adecuados y en el momento oportuno, dar soporte tanto a los empleados como a los directivos a lo largo de todo el ciclo de vida del empleado y capacitar a los empleados para gestionar procesos en un entorno colaborativo. Además, ofrece funciones de cálculo de nóminas.

Esta aplicación se centra más bien en el control de los procesos de administración del capital humano en una empresa, además es software propietario, por lo que no constituye una solución al problema planteado.

➤ **Infor Human Capital Management.**

La solución para administrar el capital humano de Infor ayuda a las compañías a optimizar todo el proceso desde el reclutamiento hasta el retiro de los empleados. Proporciona capacidades de autoservicio a empleados y gerentes, hace posible una toma de decisiones efectiva concerniente a la plantilla de empleados. Además, ayuda a mejorar la competitividad mediante la contratación, conservación y la asignación de facultades a las mejores personas. Incluye varios módulos como son:

- Administración del personal que abarca la gama de procesos de gestión del personal, incluyendo a los recursos humanos, administración de prestaciones, cuentas flexibles de gastos, indemnizaciones y nóminas.
- Gestión de talento, el cual presenta funcionalidades relacionadas con la gestión del aprendizaje, gestión del desempeño de los empleados, reclutamiento, gestión de la competencia y la planeación de la sucesión.

A pesar de tener funcionalidades similares a las requeridas por el cliente, no se selecciona pues sólo gestiona la información referente a los empleados, en cuanto al reclutamiento, el desempeño y el aprendizaje, pero no gestiona disponibilidad y demanda, factores claves para la elaboración de los planes anuales en el departamento Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación, además, es software propietario.

➤ **Success Factors**

Success Factors es un sistema de gestión de recursos humanos exclusivo para planes de sucesión, gestionar el desempeño, la formación, las compensaciones variables y tener una ficha con el historial de cada empleado y su foto. Los módulos de este sistema permiten realizar: evaluaciones de desempeño, planes de carrera, sistema de compensaciones, cuadro de mando integral, organización de la capacitación, reclutamiento y selección. Consta con un perfil del empleado, que es una ficha técnica o legajo online donde aparecen todos los datos personales y profesionales, carrera y perspectivas de crecimiento. Tiene una función denominada 360/Multi-rater que facilita la gestión del desempeño y las habilidades de los empleados de múltiples perspectivas. Además, permite realizar encuestas a los empleados y análisis de informes de negocios, entre otras funciones.

En cuanto a la herramienta de gestión del desempeño, permite:

- Identificación de los empleados con mejor desempeño. Le brinda a los ejecutivos y a los gerentes información procesable del desempeño de la organización en tiempo real.
- Desarrollo del talento. Las evaluaciones de desempeño generan reportes para mandos medios.

El sistema gestiona los datos profesionales de los empleados, permite realizar planes de carrera y generar varios reportes, sin embargo, no se ajusta a las necesidades existentes en el departamento Fuerza de Trabajo Calificada pues la información que gestiona no puede utilizarse para realizar los planes de ingreso y distribución de graduados. Además, tiene como principal desventaja que es software propietario.

1.2.2. Sistemas de gestión de información vinculada a la fuerza de trabajo calificada en Cuba.

➤ Sistema de Recursos Humanos (Fastos).

El Sistema de Recursos Humanos (Fastos), está formado por los módulos Configuración, Personal, Capacitación y Cuadros, permite controlar las informaciones fundamentales de los empleados de una entidad, también realizar varios procesos y operaciones que son inherentes al área de recursos humanos, tales como: guardar los datos de los empleados en el registro de empleados, establecer la estructura organizativa de las plazas de la entidad, controlar la asistencia y puntualidad de los trabajadores, actualizando esta información de forma automática, obtener varios informes y modelos, controlar la capacitación de los empleados respecto a: acciones de capacitación, estudios realizados, cursos, eventos, experiencia docente, publicaciones, conocimientos e idiomas extranjeros y controlar la información de los cuadros registrando todo lo referente a evaluaciones, inspecciones, sanciones, necesidades de capacitación, entre otros.

El sistema es una aplicación Cliente/Servidor, con base de datos de SQL 2000 Server. Posee ayuda incorporada, protección contra copias ilegales, control y registro de acceso al sistema y facilidades para la exportación de información. Permite tener acceso a la información con mayor rapidez y seguridad. Corre para el sistema operativo Windows en sus versiones Windows 98 o superiores, requiere un mínimo de 256 MB de memoria RAM.

Este sistema automatiza los procesos y gestiona información relacionada con la superación del capital humano en una empresa, pero la información que se obtiene no es suficiente para elaborar los planes, por lo que no constituye una solución a las necesidades del cliente.

➤ **Sistema Matriz.**

Es un sistema para la gestión de información relacionada con la fuerza de trabajo calificada del Ministerio de Economía y Planificación que permite la retroalimentación de los datos necesarios para la elaboración de los planes anuales de ingreso y distribución de graduados de las enseñanzas Superior y Técnica-Profesional; la definición de las especialidades técnicas de nivel medio y obreros calificados que deben estudiarse en la enseñanza Técnica y Profesional; la definición de las carreras de nivel superior que deben estudiarse en los Centros de Enseñanza Superior (CES) y en los Centros Universitarios Municipales (CUM) u otros tipos de cursos que se determinen; el conocimiento de la existencia actual de graduados por carreras, residencia, edad y entidad donde laboran, así como la cuantía de jóvenes calificados que se encuentran en la reserva científica, disponibles o cumpliendo el servicio social en período de adiestramiento laboral sin posibilidad de plaza fija en los organismos. Para ello se gestiona la existencia de graduados universitarios, técnicos medio y obreros calificados empleados, la proyección de la demanda complementaria de graduados universitarios y técnicos así como la de obreros calificados, los posibles graduados de la Enseñanza Superior, técnicos de nivel medio y obreros calificados del Curso Regular Diurno, que concluirán sus estudios en el período actual. Fue programado en Visual FoxPro 5.0 bajo Windows NT.

Esta aplicación es la que se utiliza actualmente en el departamento Fuerza de Trabajo Calificada, pero presenta varios errores e insuficiencia de clasificadores y nomencladores, por lo que ya no soluciona las necesidades del cliente.

➤ **GEPROP**

GEPROP tiene la misión de gerenciar, evaluar y financiar las acciones vinculadas con los procesos de generación de conocimientos científicos y tecnológicos e innovaciones tecnológicas que den respuesta a prioridades del desarrollo económico y social del país en lo que concierne a productos, servicios claves y a necesidades sociales, educacionales, culturales y medio ambientales de importancia nacional. Planifica, organiza y dirige, toda la actividad de recursos humanos en el centro, concibiendo y elaborando los planes anuales, trimestrales y mensuales de actividad para dicha área. Propone, orienta y controla, la

organización del desarrollo de recursos humanos (preparación de doctores y máster en diferentes especialidades, superación de postgrado, diplomado, cursos y entrenamientos). Organiza, coordina y supervisa el proceso de elaboración de los planes anuales de preparación y superación profesional; el trabajo de planificación, reclutamiento y selección de la fuerza calificada (técnico medio y nivel superior), así como el adiestramiento laboral de esta fuerza. Realiza estudios sobre la efectiva utilización de la fuerza de trabajo, sus potencialidades, fluctuación, etc., elaborando las recomendaciones pertinentes; confecciona, supervisa y evalúa el reglamento disciplinario perfeccionándolo sistemáticamente.

Esta aplicación no constituye una solución al problema planteado, aunque se asemeja bastante a la posible solución, porque la información que gestiona es muy general y no puede utilizarse en la realización de los planes anuales de ingreso y distribución de graduados.

1.2.3. Sistemas de gestión de información vinculada a la fuerza de trabajo calificada en la UCI.

➤ Cedrux.

El módulo de Capital Humano de Cedrux tiene entre sus principales funcionalidades gestionar todos los datos referentes al personal de la empresa (nombre y apellidos, dirección particular, edad), así como información de los dirigentes (nombre y apellidos, área a la que pertenece y dirección de trabajo donde radica). Posibilita un proceso de evaluación del personal y un proceso organizativo que incluye la selección del personal adecuado para los puestos de trabajo disponibles. Además, brinda funcionalidades administrativas como la realización de nóminas, cálculo de submayores y cierres de mes. Está implementado con herramientas de software libre y utiliza la arquitectura Modelo Vista Controlador.

A pesar de ser software libre y gestionar información relacionada con los recursos humanos, no se ajusta a las necesidades del cliente, por lo que no se selecciona.

1.3. ¿Por qué se decide implementar un nuevo sistema?

Al concluir el estudio realizado de las tendencias actuales de los sistemas de gestión de información vinculada a la fuerza de trabajo calificada en el ámbito internacional, nacional y en la universidad, se obtuvo como resultado que todos estos sistemas presentaron funcionalidades similares a las requeridas por el cliente, pero ninguno de ellos cumple, de forma general, con todos los requerimientos que se necesitan para gestionar la información necesaria para elaborar los planes pertinentes del departamento

Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación y solamente aportan ideas para la realización del nuevo sistema.

Lo más parecido a una posible solución sería el propio Sistema Matriz pero no se selecciona por todas las deficiencias que presenta en cuanto a: falta de nuevos clasificadores y nomencladores, difícil operabilidad del sistema y poca seguridad de la información principalmente.

1.4. Tendencias actuales al software libre.

Software libre es la denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, cambiado y redistribuido libremente. (VANESSA PÉREZ CORREIA)

Entre sus principales ventajas se encuentran:

- **Escrutinio público:** Al ser muchas las personas que tienen acceso al código fuente, eso lleva a un proceso de corrección de errores muy dinámico, no hace falta esperar a que el proveedor del software saque una nueva versión.
- **Independencia del proveedor:** Al disponer del código fuente, cualquier persona puede continuar ofreciendo soporte, desarrollo u otro tipo de servicios para el software.
- **Manejo de la lengua:** cualquier persona capacitada puede traducir y adaptar un software libre a cualquier lengua.
- **Mayor seguridad y privacidad:** Los sistemas de almacenamiento y recuperación de la información son públicos. Cualquier persona puede ver y entender cómo se almacenan los datos en un determinado formato o sistema. Existe una mayor dificultad para introducir código malicioso como son: espía (p/ej. capturador de teclas), de control remoto (p/ej. Troyano), de entrada al sistema (p/ej. puerta trasera), etc.
- **Garantía de continuidad:** el software libre puede seguir siendo usado aún después de que haya desaparecido la persona que lo elaboró, dado que cualquier técnico informático puede continuar desarrollándolo, mejorándolo o adaptándolo.
- **Ahorro en costos:** disminuye el costo de adquisición ya que al otorgar la libertad de distribuir copias se puede ejercer con la compra de una sola licencia y no con tantas como computadoras posea (como sucede en la mayoría de los casos de software propietario). Cabe aclarar que también hay

una disminución significativa en el costo de soporte, no ocurriendo lo mismo con los costos de implantación y de interoperabilidad. (VANESSA PÉREZ CORREIA)

Atendiendo a dichas ventajas, Cuba y la Universidad de las Ciencias Informáticas (UCI), en particular, han abogado por la posibilidad de migrar hacia software libre, teniendo en cuenta además, que el mismo permite la utilización de productos informáticos que no requieren el uso de licencias para su explotación; lo cual representa una gran alternativa ante la subordinación de Cuba a Microsoft que atenta contra el programa de informatización del país que se está llevando a cabo como parte de la batalla de ideas. Su utilización no implica gastos adicionales por concepto de cambio de plataforma de software, por cuanto es operable en el mismo soporte de hardware con que cuenta el país. La adquisición de cualquiera de sus distribuciones, puede hacerse de forma gratuita, descargándolas directamente de Internet o en algunos casos a muy bajo costo. Permite su adaptación a los contextos de aplicación, al contar con su código fuente, lo cual garantiza un mayor por ciento de efectividad, además la corrección de sus errores de programación, obtención de las actualizaciones y nuevas versiones.

1.5. Metodología, herramientas y tecnologías utilizadas.

1.5.1. Aplicación web.

Una aplicación web es una aplicación que se codifica en un lenguaje soportado por los navegadores web (HTML, JavaScript, Java, ASP, .NET, PHP, etc.) y en la que se confía la ejecución al navegador. Las aplicaciones web son muy utilizadas debido a lo práctico del navegador web como cliente ligero, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Una página web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, por ejemplo, rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo. (*Aplicación web* 2003)

La aplicación web tiene muchas ventajas, por ejemplo:

- Ahorra tiempo pues 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.
- Las actualizaciones son inmediatas, al conectarse se usa siempre la última versión que se haya lanzado.
- Bajo consumo de recursos, muchas de las tareas que realiza el software no consumen recursos porque se realizan desde otro ordenador.
- Son multiplataforma, se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- Son portables, independientemente del ordenador que se utilice (una computadora de escritorio, una portátil, un móvil...) se tendrá acceso porque sólo es necesario disponer de acceso a Internet.
- La disponibilidad suele ser alta porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
- Los virus no dañan los datos porque estos están guardados en el servidor de la aplicación.

1.5.2. Metodología de desarrollo de software: RUP.

El Proceso Unificado Racional (RUP) es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de actitud y diferentes tamaños de proyectos. (I. JACOBSON 2000)

Según Jacobson, Boch y Rumbaugh, RUP tiene las características siguientes:

- Está dirigido por casos de uso: que orientan el proyecto a la importancia para el usuario y lo que este quiere.
- Está centrado en la arquitectura: que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden.
- Es iterativo e incremental: donde divide el proyecto en miniproyectos para los cuales los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada.

Divide su ciclo de vida en cuatro fases: Inicio, Elaboración, Construcción, Transición y nueve flujos de trabajo de los cuales seis son ingenieriles y tres de soporte.

Como metodología para desarrollar sistemas, RUP es muy potente y presenta elementos significativos como los presentados a continuación:

- Unifica los mejores elementos de las restantes metodologías.
- Preparado para desarrollar grandes y complejos proyectos.
- Admite UML como lenguaje de representación visual.
- Muy organizado y con documentación.
- Define las actividades, roles y artefactos para cada flujo de trabajo.
- Iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura.
- Cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software, las cuales se muestran a continuación:
 - Desarrollo de software en forma iterativa.
 - Manejo de requisitos.
 - Utiliza arquitectura basada en componentes.
 - Modela el software visualmente.
 - Verifica la calidad del software.
 - Controla los cambios.

1.5.3. Arquitectura: Cliente/Servidor, Modelo Vista Controlador.

➤ Cliente/Servidor.

La arquitectura Cliente/Servidor es aquella en la que confluyen una serie de aplicaciones basadas en dos categorías que cumplen funciones diferentes (una requiere servicios y la otra los brinda) pero que a la vez, pueden realizar tanto actividades en forma conjunta como independientemente. Esas dos categorías son justamente, cliente y servidor.

En el caso del cliente, es aquel que requiere un servicio del servidor. En esta categoría se realizan funciones de software basándose en el hardware pero en caso de no tener la capacidad de procesar los datos necesarios, recurre al servidor y espera a que este le brinde los servicios solicitados. El cliente es una estación de trabajo o computadora que está conectada a una red a través de la cual puede acceder al servidor.

Por el contrario, el servidor es la máquina desde la que se suministran servicios y que está a la espera del requerimiento del cliente. Una vez hecho, busca la información solicitada y le envía la respuesta al cliente; incluso puede enviar varios servicios a la vez, lo que es posible porque entre ellos están conectados mediante una red, ya sea una Red de Área Local (LAN) o una Red de Área Amplia (WAN). (Ver figura en el Anexo 3). (LANZILLOTTA).

Entre las características fundamentales de esta arquitectura se encuentran:

- Tanto el cliente como el servidor pueden realizar tareas en forma conjunta como separada ya que el cliente también tiene sus propias aplicaciones, archivos y bases de datos.
- El cliente y el servidor pueden estar en la misma plataforma o en plataformas diferentes.
- Por otra parte, el servidor puede brindar varios servicios a la vez, tanto al mismo cliente como a clientes múltiples.

Ventajas del modelo Cliente-Servidor:

- Se reducen los costos de producción de software y se disminuyen los tiempos requeridos, ya que para la construcción de una nueva aplicación pueden usarse los servidores que estén disponibles, reduciéndose el desarrollo a la elaboración de los procesos del cliente, según los requerimientos deseados.
- Facilita el suministro de información a los usuarios, proporcionando una mayor consistencia a la información de la organización, al contar con un control centralizado de los elementos compartidos.
- Permite llevar más fácilmente la información a donde se necesita y contribuye a aumentar su precisión, pues se puede obtener de la fuente (el servidor) y no de una copia en papel o en medio magnético.
- La habilidad de integrar sistemas heterogéneos es inherente al modelo Cliente/Servidor, pues los clientes y los servidores pueden existir en múltiples plataformas y tener acceso a datos de cualquier sitio de la red.
- Favorece la adaptación a cambios en la tecnología, pues facilita la migración de las aplicaciones a otras plataformas y, al aislar claramente las diferentes funciones de una aplicación, hace más fácil incorporar nuevas tecnologías en esta.

➤ **Modelo Vista Controlador.**

El patrón de arquitectura Modelo Vista Controlador o MVC describe una forma, muy utilizada en la web, de organizar el código de una aplicación separando los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. (Ver figura en el Anexo 3). (*ProgramaciónWeb.net* 2010).

El Modelo (*Model*): Encapsula los datos y las funcionalidades. Es independiente de cualquier representación de salida y/o comportamiento de entrada. Debe preservar la integridad de los datos.

El Controlador (*Controller*): Recibe las entradas, usualmente como eventos, e interpreta las operaciones del usuario; codificando los movimientos, pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio para el modelo o la vista.

La Vista (*View*): Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador que interactúa con la interfaz de usuario.

Este patrón de arquitectura presenta varias ventajas:

- Hay una clara separación entre los componentes de un programa; lo cual permite implementarlos por separado.
- Hay un API muy bien definido; cualquiera que use el API, 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.

1.5.4. Lenguaje de modelado visual: UML 2.0.

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Tiene como objetivos: visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y

construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas. (*Diseño UML*. 2009)

1.5.5. Herramienta para el modelado: Visual Paradigm for UML 6.1.

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a la rápida construcción de aplicaciones de calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación. Esta herramienta también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Permite la generación de bases de datos, ingeniería inversa de bases de datos, desde sistemas gestores de bases de datos existentes a diagramas de entidad relación. (*Free Download Manager*)

1.5.6. Lenguajes de programación: JavaScript 2.0 y PHP 5.0.

➤ Del lado del cliente: JavaScript 2.0.

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. (*librosweb.es*)

Entre las ventajas de JavaScript se encuentran:

- JavaScript no requiere un tiempo de compilación, los *scripts* se pueden desarrollar en un período de tiempo relativamente corto.

- Características de interfaz, por ejemplo, cuadro de diálogo, formularios y otros elementos GUI (Interfaz Gráfico de Usuario), son gestionados por el navegador y por el código HTML. Por lo tanto los programadores que utilizan JavaScript no se deben preocupar en crear o controlar dichos elementos en sus aplicaciones.
- Aunque JavaScript tiene muchas similitudes con Java, no incluye la sintaxis y reglas complejas de Java.
- Como WWW es independiente de la plataforma hardware o sistema operativo, los programas escritos en JavaScript también lo son, siempre y cuando exista un navegador con soporte JavaScript para la plataforma en cuestión.
- Los programas JavaScript tienden a ser pequeños y compactos (en comparación con los *applets* de Java), no requieren mucha memoria ni tiempo adicional de transmisión. Además, al incluirse dentro de las mismas páginas HTML se reduce el número de accesos independientes a la red.

➤ Del lado del servidor: PHP 5.0.

PHP (acrónimo de "PHP: *Hypertext Preprocessor*") es un lenguaje de código abierto interpretado de alto nivel y ejecutado en el servidor. Completamente orientado al desarrollo de aplicaciones webs dinámicas con acceso a información almacenada en una base de datos y puede ser embebido en HTML. Permite aplicar técnicas de programación orientada a objetos. Presenta una biblioteca nativa de funciones muy amplia y un sólido manejo de excepciones. La mayor fuerza de PHP es que está preparado para soportar accesos a muchos tipos de bases de datos y soporta la utilización de diferentes protocolos como IMAP, SNMP, NNTP, POP3 o HTTP. Está disponible para los principales sistemas operativos. (*PHP: Hypertext Processor*)

Las ventajas que presenta son:

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto, clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los gestores de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.

- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado de bajo nivel.

1.5.7. Plataforma de desarrollo: NetBeans 6.8.

NetBeans es un proyecto de código abierto fundado por Sun Microsystems especialmente diseñado para el desarrollo de aplicaciones en Java, pero que acepta otros lenguajes de programación. Consta de una gran base de usuarios y una comunidad en constante crecimiento, lo que le ha permitido, al igual que muchos otros sistemas libres, el progreso paulatino de sus prestaciones y la eliminación de errores que pudiesen existir. (*NetBeans* 2010)

Sus ventajas fundamentales:

- Permite que las aplicaciones se desarrollen a partir de un conjunto de módulos o componentes de software.
- Es fácil de instalar y de uso instantáneo, se ejecuta en varias plataformas incluyendo Windows, Linux y otros.
- Además del soporte completo para todas las plataformas Java (Java SE, Java EE, Java ME, y JavaFX), NetBeans IDE 6.5 es la herramienta ideal para el desarrollo de software con PHP, Ajax, JavaScript y C/C++.

1.5.8. Marcos de trabajo: ExtJS, Zend Framework, Doctrine.

Un *framework* es una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un *framework* se puede considerar como una aplicación genérica incompleta y configurable a la que se pueden añadir las últimas piezas para construir una aplicación concreta. (GUTIÉRREZ)

➤ ExtJS 2.2.

Ext es un nuevo marco de trabajo de JavaScript del lado del cliente para el desarrollo de aplicaciones web, basado originalmente en YUI pero que actualmente es independiente del marco de trabajo que se

utilice (incluso puede usarse sin marco de trabajo). Ofrece una gran cantidad de reproductores para crear interfaces de usuario complejas. PHP-Ext es una librería de código abierto que permite potenciar la capa interfaz de usuario de JavaScript en las aplicaciones. Ofrece una serie de librerías (compatibles con PHP 4 y 5) para integrar ExtJS en el sistema. Entre las posibilidades que ofrece se encuentran la creación de formularios, combos, *grids* o menús. Además, ayuda a la comunicación entre el cliente y el servidor mediante JSON y XML. (CELIS 2005)

Tiene un sistema dual de licencia: Comercial y Open Source. Este marco de trabajo puede correr en cualquier plataforma que pueda procesar POST y devolver datos estructurados (PHP, Java, .NET y algunas otras). Basa toda su funcionalidad en JavaScript a través de librerías ya muy conocidas. En tiempo de ejecución, carga y crea todos los objetos HTML a través del uso intenso de DOM (Modelo de Objetos del Documento).

Tiene como ventajas:

- Código reutilizable.
- Independiente o adaptable a marcos de trabajo diferentes (*prototype, jquery, YUI*).
- Orientada a la programación de interfaces tipo escritorio (*desktop*) en la web.
- El API es homogeneizado independientemente del adaptador utilizado. Los controles siempre se verán igual.
- Soporte comercial.
- Extensa variedad de usuarios.

➤ **Zend Framework 1.5.**

Es utilizado en el desarrollo de aplicaciones web y servicios web con PHP. Brinda soluciones para construir sitios web modernos, robustos y seguros. Es de código abierto, utiliza licencia BSD. Es extremadamente simple consultar su base de datos, sin tener que escribir ninguna consulta SQL. Utiliza arquitectura Modelo Vista Controlador. Este marco de trabajo cuenta con robustas clases para autenticación y filtrado de entrada, así como clientes para servidores web. (*techtásctico* 2007)

Tiene como ventajas:

- Estandariza los procesos más frecuentes, dotándolos de gran robustez.
- Facilita el mantenimiento de las aplicaciones.

- Ofrece muchas facilidades para el acceso a recursos avanzados (servicios web asegurados, por ejemplo) que de otro modo resultan más costosos de desarrollar.
- A diferencia de otros marcos de trabajo, es posible utilizarlo en modo "desacoplado", es decir, aquellas clases o componentes que sean necesarios en cada proyecto, sin arrastrar todo el marco de trabajo detrás para cualquier pequeña necesidad.
- Tiene el respaldo de la propia ZEND, creadora de PHP, lo que asegura su continuidad futura tanto como la del propio lenguaje PHP.

➤ **Doctrine 1.0.11.**

Doctrine es un potente y completo sistema ORM (Mapeador relacional de objetos) para PHP 5.2 con una DBAL (Capa de abstracción de base de datos) incorporada.

Entre muchas otras capacidades 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. Por otro lado, como la librería es bastante grande, esta tiene un método para ser "compilada" al pasar a producción. Su principal ventaja radica en poder acceder a la base de datos utilizando la programación orientada a objetos (POO) debido a que Doctrine utiliza el patrón *Active Record* para manejar la base de datos, tiene su propio lenguaje de consultas y trabaja de manera rápida y eficiente. Es fácilmente integrado a los principales marcos de trabajo utilizados actualmente. (*Joan Ganrmet 2007*)

Tiene como ventajas:

- Cien por ciento desacoplado.
- Las clases entidades no necesitan heredar de ninguna otra clase, entidades mucho más desacopladas y reutilizables.
- No impone otros métodos ni propiedades en el modelo de objetos de dominio o clases entidades (En Java se les conocen como POJO, DTO o Entity).
- Desarrollado completamente en PHP 5.3, el futuro de PHP y desarrollo de aplicaciones empresariales.
- Toma lo mejor de los ORM de Java, está basado en Hibernate, EntityManager y JPA (Java Persistence API).

- Mucho más elegante, con estándares y buenas prácticas de programación orientada a objetos e implementación de patrones de diseño.

1.5.9. Servidor web: Apache 2.0.

Posee facilidad de configuración, robustez, estabilidad y transparencia. Apache está desarrollado por la Fundación de Software Apache (Apache Software Foundation), es una tecnología gratuita de código fuente abierto con licencia descendiente de las licencias BSD, por lo que se puede hacer uso de su código fuente. Lo que hace a este servidor web universal, es que corre en una multitud de sistemas operativos. Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este. Brinda soporte para varios lenguajes: PHP, JAVA, Perl y librerías ASP. Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado *script* cuando ocurra un error en concreto. Un gran número de aplicaciones web están montadas en servidores Apache. (*ciberaula.com*)

Entre sus ventajas se encuentran:

- El paquete del servidor es más flexible en tiempo de ejecución porque el proceso actual del servidor puede ser ensamblado en tiempo de ejecución por medio de LoadModule en httpd.conf en lugar de hacerlo por medio de la configuración en tiempo de compilación. De este modo se pueden arrancar diferentes instancias del servidor (estándar, versión SSL, mínima, versión potenciada [PHP, etc.], etc.) con una única instalación de Apache.
- El paquete del servidor puede ser fácilmente ampliado con módulos de terceros incluso después de la instalación. Esto representa un gran beneficio para los que mantienen paquetes, ya que les permite crear el paquete del núcleo de Apache y adicionalmente paquetes que contengan extensiones como PHP, mod_perl, mod_fastcgi.
- Mayor facilidad en los prototipos de módulos Apache porque con DSO y apxs se puede trabajar fuera del árbol fuente de Apache y necesitar un único comando apxs -i seguido de un apachectl restart para cargar una nueva versión del módulo desarrollado en el servidor Apache que esté funcionando actualmente.

1.5.10. Sistema gestor de base de datos: PostgreSQL 8.3.

Es un potente sistema gestor de base de datos relacional orientado a objetos de software libre, publicado bajo la licencia BSD, desarrollado por PostgreSQL Global Development Group traducido al español como Grupo Global de Desarrollo de PostgreSQL. PostgreSQL tiene un activo desarrollo y una probada arquitectura que se ha ganado una sólida reputación de fiabilidad, es una excelente solución para la administración de la base de datos. Funciona en todos los principales sistemas operativos, como Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. La nueva versión 8.3 incluye una cantidad récord de características nuevas y mejoradas respecto a versiones anteriores. La velocidad de respuesta que ofrece este gestor con bases de datos relativamente pequeñas puede parecer un poco deficiente, aunque esta misma velocidad la mantiene al gestionar bases de datos realmente grandes. A su vez, posee una gran escalabilidad y es capaz de ajustarse al número de CPUs (unidades de proceso central) y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta, acoplándose, fundamentalmente esta característica a las necesidades del sistema a desarrollar. (PECOS 2010)

Presenta como ventajas:

- Estable.
- Alto rendimiento.
- Flexibilidad.
- Se puede extender su funcionalidad.
- Gran compatibilidad.
- Permite crear o migrar aplicaciones desde Access, Visual Basic, Visual Fox Pro, Visual C/C++, Delphi para usar PostgreSQL como servidor de base de datos.
- Varias interfaces de programación: ODBC, JDBC, C/C++, SQL Embebido, Tcl/Tk, Perl, Python, PHP.

1.5.11. 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.

El sistema 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.

Sus principales ventajas son:

- Permite a los usuarios, entre otras opciones, abstraerse a los conocimientos relacionados con los sistemas 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.
- El 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.
- El 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.
- El Administrador de reportes permite administrar las plantillas, los reportes y los modelos anteriormente creados en las aplicaciones Diseñador de reportes y Diseñador de modelos.
- El 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.
- El Diseñador de consultas 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.

1.6. Fundamentación de la metodología, herramientas y tecnologías a utilizar.

La metodología, las herramientas y las tecnologías a utilizar fueron definidas por la dirección del proyecto Fuerza de Trabajo Calificada en su marco de trabajo basándose en las tendencias actuales de migración a software libre.

Se realizará la implementación de una aplicación web como solución al problema planteado por todas las ventajas que esto representa, principalmente, porque no es necesario instalarla en cada una de las máquinas que utilizarán el Sistema Matriz y en caso de haber errores sólo se realizan los cambios en el servidor sin necesidad de ir a arreglarlos en cada una de las máquinas clientes.

La metodología de desarrollo a utilizar es RUP que permite desarrollar aplicaciones sacando el máximo provecho de las nuevas tecnologías, mejorando la calidad, el rendimiento, la reutilización, la seguridad y el mantenimiento del software mediante una gestión sistemática de los riesgos. Genera muchos artefactos que pueden ser usados según las necesidades del sistema y que guían detalladamente el proceso de desarrollo. Se obtiene amplia documentación que facilita la comprensión del sistema. Existen disímiles materiales que facilitan su estudio y puesta en práctica.

Como herramienta para la modelación del sistema se utilizará Visual Paradigm, aunque es un software propietario la universidad posee su licencia por tanto puede utilizarlo. Permite realizar ingeniería inversa y posee cualidades como la generación automática de diagramas a partir de las descripciones de casos de uso, por ejemplo, diagramas de secuencia, permitiendo la agilidad en el trabajo del analista. Utiliza un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación y por sus características, cubre todo el ciclo de vida de un proyecto.

Como lenguaje de programación del lado del cliente se utilizará JavaScript pues es un lenguaje que no requiere compilación, es independiente de la plataforma hardware o sistema operativo, no requiere mucha memoria ni tiempo adicional de transmisión, reduciendo el número de accesos independientes a la red.

Se utilizará PHP como lenguaje de programación del lado del servidor pues, es un lenguaje multiplataforma que presenta una alternativa de fácil acceso para todos, la programación en PHP es segura y confiable debido a que el código fuente escrito en este lenguaje es invisible al navegador y al cliente, debido a que el servidor es el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.

Como plataforma de desarrollo se usará NetBeans porque es un producto de software libre que le permite a los equipos de desarrollo utilizar las mejores prácticas y estándares de la industria del software para la productividad general del grupo, permite desarrollar aplicaciones a partir de un conjunto de módulos o componentes de software, además es una herramienta ideal para el desarrollo de aplicaciones web.

Se empleará ExtJS para programar la capa de presentación ya que permite crear interfaces de usuarios muy amigables y eficaces. Zend Framework para la capa de negocio ya que permite la realización de aplicaciones robustas y seguras; además, facilita el mantenimiento de las mismas. También se usará Doctrine para la capa de abstracción de datos pues brinda un gran rendimiento en ejecución y admite

escribir, de forma segura, consultas muy complejas.

Se usará Apache como servidor web pues desde su salida al mercado es el más utilizado de los servidores web, es un servidor libre y multiplataforma, su instalación y configuración son muy sencillas, brinda soporte para varios lenguajes de programación, entre ellos PHP que es el lenguaje que se utilizará para la implementación del módulo Sistema Matriz, tiene un alto grado de calidad y fortaleza para las aplicaciones que utilizan el protocolo HTTP.

Como sistema gestor de base de datos se utilizará PostgreSQL, pues es uno de los gestores de bases de datos más utilizados actualmente, es capaz de manejar una enorme cantidad de datos, permitiendo gran conjunto de accesos simultáneos de los usuarios, brindando seguridad y estabilidad de los mismos, facilita el trabajo con procedimientos almacenados y consultas.

Se utilizará además el generador de reportes dinámico PATDSI por todas las funcionalidades que brinda en cuanto a la administración de modelos y consultas para agilizar el proceso de generación de reportes.

1.7. Conclusiones.

En este capítulo se realizó un estudio de los sistemas de gestión de información relacionada con la fuerza de trabajo calificada a nivel internacional, nacional y en la universidad. Se demostró la necesidad de implementar un nuevo sistema pues ninguno de los estudiados cumple, en su totalidad, las funcionalidades requeridas. Se fundamentó el uso de software libre por sus ventajas y la utilización de JavaScript como lenguaje de programación del lado del cliente, PHP como lenguaje de programación del lado del servidor, ExtJS, Zend Framework y Doctrine como marcos de trabajo, NetBeans como IDE, PostgreSQL como sistema gestor de base de datos, como servidor web Apache y como herramienta CASE Visual Paradigm. La metodología a utilizar es RUP por las amplias posibilidades que brinda en cuanto a documentación y artefactos generados. Se utilizará además, el generador de reportes dinámico PATDSI por todas las posibilidades que brinda, lo cual facilita y agiliza mucho el trabajo del desarrollador.

Capítulo II: Características del sistema.

2.1. Introducción

En este capítulo se realizará un estudio de los procesos del negocio, identificando los conceptos principales que se manejan y sus relaciones para realizar el modelo de dominio, serán identificados los requisitos funcionales y no funcionales, así como los actores y casos de uso del sistema, que posteriormente permitirán desarrollar el modelo de casos de uso del sistema.

2.2. Modelo de dominio

Un modelo de 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 en el que trabaja el sistema. (I. JACOBSON 2000)

Al finalizar el estudio de los procesos del negocio, se llegó a la conclusión de que no es necesario hacer un modelado completo del mismo, pues se contaba con el sistema que el cliente usa actualmente, y que se asemeja a lo que debería ser la solución actual, por lo que se decide realizar un modelo de dominio representando los conceptos más importantes identificados dentro de los procesos del negocio y las relaciones entre ellos para mejor entendimiento de todo este proceso por parte del equipo de desarrollo.

2.3. Diagrama de clases del modelo de dominio

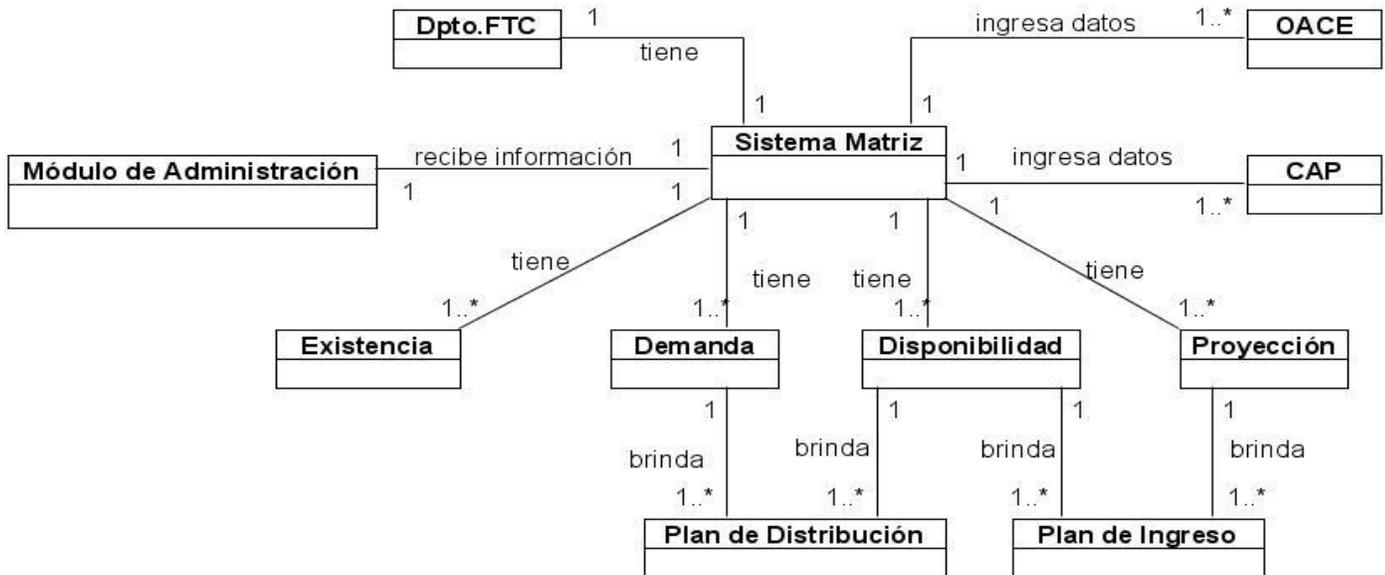


Figura 1. Modelo de dominio

2.4. Definición de las clases del modelo de dominio

- **Departamento FTC:** Departamento Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación (MEP).
- **Sistema Matriz:** Subsistema de GeForza que utiliza el departamento FTC para gestionar toda la información necesaria para realizar el Plan de Ingreso a la Educación Superior del Curso Regular Diurno, así como el Plan de Distribución de Graduados de nivel superior formados en el Curso Regular Diurno y el Plan de Continuidad de Estudios de los estudiantes de noveno grado.
- **OACE:** Organismos de la Administración Central del Estado, son los encargados de nutrir el Sistema Matriz con toda la información referente a su rama, gestionando a su vez dicha información en niveles inferiores.
- **CAP:** Centro de Administración Provincial, son los encargados de nutrir el Sistema Matriz con toda la información referente a su división político administrativa, gestionando a su vez dicha información en niveles inferiores.

- **Módulo de Administración:** Subsistema de GeForza encargado centralizar la seguridad de la información y homogeneizar los clasificadores y nomencladores utilizados por el resto de los subsistemas.
- **Existencia:** Se refiere a los graduados universitarios, técnicos medio y obreros calificados que ocupan plazas laborales.
- **Disponibilidad:** Se refiere a los graduados que culminaron sus estudios y están pendientes a ubicación laboral. También se refiere a los graduados que continuarán estudios en niveles superiores.
- **Demanda:** Se refiere a la necesidad de graduados universitarios, técnicos medio y obreros calificados en los organismos y entidades del país.
- **Proyección:** Se refiere a la necesidad de graduados universitarios, técnicos medio y obreros calificados en los organismos y entidades del país para el quinquenio próximo.
- **Plan de Ingreso:** Subsistema de GeForza encargado de realizar el Plan de Ingreso a la Educación Superior del Curso Regular Diurno y el Plan de Continuidad de Estudios de los estudiantes de noveno grado, a partir de la información recibida del Sistema Matriz.
- **Plan de Distribución:** Subsistema de GeForza encargado de realizar Plan de Distribución de graduados de nivel superior formados en el Curso Regular Diurno a partir de la información recibida del Sistema Matriz.

2.5. Especificación de los requerimientos de software.

Un requerimiento es una condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar u otro documento impuesto. (I. JACOBSON 2000).

2.5.1. Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. (I. JACOBSON 2000).

A continuación se listan los requerimientos funcionales de forma resumida, la lista ampliada está disponible en el Anexo 4.

RF 1 Administrar la demanda de graduados por carreras:

- 1.1 Crear la demanda de graduados por carreras.
- 1.2 Modificar la demanda de graduados por carreras.
- 1.3 Eliminar la demanda de graduados por carreras.
- 1.4 Buscar la demanda de graduados por carreras.
- 1.5 Seleccionar año.
- 1.6 Mostrar demanda de graduados por carreras.

RF 2 Administrar existencia de empleados:

- 2.1 Crear nueva existencia de empleados.
- 2.2 Modificar existencia de empleados.
- 2.3 Eliminar existencia de empleados.
- 2.4 Buscar existencia de empleados.
- 2.5 Seleccionar año.
- 2.6 Mostrar existencia de empleados.

RF 3 Administrar la proyección de la demanda:

- 3.1 Crear una nueva proyección de la demanda de graduados.
- 3.2 Modificar una proyección de demanda de graduados.
- 3.3 Eliminar proyección de la demanda de graduados.
- 3.4 Buscar proyección de la demanda de graduados.
- 3.5 Seleccionar año.
- 3.6 Mostrar proyección de la demanda de graduados.

RF 4 Administrar la disponibilidad de graduados de noveno grado:

- 4.1 Crear la disponibilidad de graduados de noveno grado.
- 4.2 Modificar la disponibilidad de graduados de noveno grado.
- 4.3 Eliminar la disponibilidad de graduados de noveno grado.
- 4.4 Buscar la disponibilidad de graduados de noveno grado.
- 4.5 Seleccionar año.
- 4.6 Mostrar disponibilidad de graduados de noveno grado.

RF 5 Administrar la disponibilidad de graduados de nivel medio (12mo. Grado, técnicos medio y obreros calificados):

- 5.1 Crear la disponibilidad de graduados de nivel medio.
- 5.2 Modificar la disponibilidad de graduados de nivel medio.
- 5.3 Eliminar la disponibilidad de graduados de nivel medio.
- 5.4 Buscar la disponibilidad de graduados de nivel medio.
- 5.5 Seleccionar año.
- 5.6 Mostrar disponibilidad de graduados de nivel medio.

RF 6 Administrar la disponibilidad de graduados de nivel superior (universitarios).

- 6.1 Crear la disponibilidad de graduados de nivel superior.
- 6.2 Modificar la disponibilidad de graduados de nivel superior.
- 6.3 Eliminar la disponibilidad de graduados de nivel superior.
- 6.4 Buscar la disponibilidad de graduados de nivel medio.
- 6.5 Seleccionar año.
- 6.6 Mostrar disponibilidad de graduados de nivel superior.

RF 7 Mostrar organismos:

7.1 Mostrar una lista de todos los organismos.

RF 8 Mostrar DPA:

8.1 Mostrar una lista de todas las provincias y cada uno de sus municipios.

RF 9 Mostrar entidades:

9.1 Mostrar una lista de todas las entidades por cada uno de los organismos.

RF 10 Mostrar tipos de plazas laborales:

10.1 Mostrar una lista con todos los tipos de plazas existentes.

RF 11 Mostrar carreras por ramas de la ciencia:

11.1 Mostrar una lista de todas las ramas de la ciencia y cada una de sus carreras.

RF 12 Mostrar especialidades por familia de especialidades:

12.1 Mostrar una lista de todas las familias de especialidades y cada una de sus especialidades.

RF 13 Mostrar rangos de edades:

13.1 Mostrar una lista de los rangos de edades.

RF 14 Mostrar resumen de disponibilidad:

14.1 Mostrar el total de toda la disponibilidad de graduados universitarios.

14.2 Mostrar el total de toda la disponibilidad de técnicos medio y obreros calificados.

14.3 Mostrar el total de toda la disponibilidad de noveno grado.

14.4 Mostrar el total de toda la disponibilidad de duodécimo grado.

RF 15 Mostrar resumen de existencia de empleados:

15.1 Mostrar un resumen de la existencia de empleados universitarios por Organismo/DPA/Entidad.

15.2 Mostrar un resumen de la existencia de empleados técnicos medio y obreros calificados por Organismo/DPA/Entidad.

- 15.3 Mostrar un resumen de la existencia de empleados universitarios por DPA/Organismo/Entidad.
- 15.4 Mostrar un resumen de la existencia de empleados técnicos medio y obreros calificados por DPA/Organismo/Entidad.
- 15.5 Mostrar un resumen de la existencia de empleados universitarios por Organismo.
- 15.6 Mostrar un resumen de la existencia de empleados técnicos medio y obreros calificados por Organismo.
- 15.7 Mostrar un resumen de la existencia de empleados universitarios por DPA.
- 15.8 Mostrar un resumen de la existencia de empleados técnicos medio y obreros calificados por DPA.

RF 16 Mostrar un resumen de la demanda de graduados:

- 16.1 Mostrar un resumen de toda la necesidad de graduados universitarios por Organismo/DPA/Entidad.
- 16.2 Mostrar un resumen de toda la necesidad de técnicos medio y obreros calificados por Organismo/DPA/Entidad.
- 16.3 Mostrar un resumen de toda la necesidad de graduados universitarios por DPA/Organismo/Entidad.
- 16.4 Mostrar un resumen de toda la necesidad de técnicos medio y obreros calificados por DPA/Organismo/Entidad.
- 16.5 Mostrar un resumen de toda la necesidad de graduados universitarios por Organismo.
- 16.6 Mostrar un resumen de toda la necesidad de técnicos medio y obreros calificados por Organismo.
- 16.7 Mostrar un resumen de toda la necesidad de graduados universitarios por DPA.
- 16.8 Mostrar un resumen de toda la necesidad de técnicos medio y obreros calificados por DPA.

RF 17 Mostrar un resumen de la proyección de la demanda:

- 17.1 Mostrar un resumen de toda la proyección de la demanda de graduados universitarios por Organismo/DPA/Entidad.
- 17.2 Mostrar un resumen de toda la proyección de la demanda de técnicos medio y obreros calificados por Organismo/DPA/Entidad.

- 17.3 Mostrar un resumen de toda la proyección de la demanda de graduados universitarios por DPA/Organismo/Entidad.
- 17.4 Mostrar un resumen de toda la proyección de la demanda de técnicos medio y obreros calificados por DPA/Organismo/Entidad.
- 17.5 Mostrar un resumen de toda la proyección de la demanda de graduados universitarios por Organismo.
- 17.6 Mostrar un resumen de toda la proyección de la demanda de técnicos medio y obreros calificados por Organismo.
- 17.7 Mostrar un resumen de toda la proyección de la demanda de graduados universitarios por DPA.
- 17.8 Mostrar un resumen de toda la proyección de la demanda de técnicos medio y obreros calificados por DPA.

RF 18 Mostrar listado de organismos que han reportado información:

- 22.1 Mostrar una lista de todos los organismos que han reportado información.

RF 19 Exportar a PDF:

- 23.1 Crear un documento pdf con la información que se esté visualizando en pantalla, dejando habilitadas todas las funciones que implementa el Adobe Reader.

RF 20 Exportar XML.

- 24.1 Seleccionar el modelo o reporte.
- 24.2 Exportar modelos o reporte.

RF 21 Importar XML.

- 25.1 Seleccionar el archivo XML.
- 25.2 Restaurar en la base de datos todos los datos obtenidos a partir de un archivo XML.

2.5.2. Requerimientos no funcionales.

Los requerimientos no funcionales especifican propiedades del sistema (I. JACOBSON 2000).

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.

Fiabilidad.

- Crear una copia de la base de datos que abarque toda la información que se necesite según el organismo que lo use (versión MEP o organismo).
- Permitir la transferencia de información desde las distintas versiones (MEP u organismo) hacia las otras.
- Permitir la transferencia de información de manera fragmentada y que después se pueda unir en la base de datos del MEP.
- El sistema deberá realizar el borrado de toda la información que se encuentre en la base de datos y que tenga más de 5 años.

Eficiencia.

- El sistema debe funcionar con un máximo rendimiento pero ajustado a bajas prestaciones de las computadoras, debido a 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.
- Para un funcionamiento óptimo de la aplicación se deben seguir las diferentes técnicas de elaboración en la web, que faciliten el rápido acceso a sus páginas.
- La herramienta propuesta debe ser rápida y el tiempo de respuesta debe ser el mínimo posible, adecuado a la rapidez con que el cliente requiere la respuesta a su petición.

Soporte

- Realizar pruebas, mantenimiento e instalaciones necesarias para lograr el mejoramiento y evolución en el tiempo.
- Para garantizar el soporte de esta herramienta, se debe documentar la aplicación con un manual de ayuda para los usuarios, así como la posibilidad de emitir sus quejas y sugerencias a los desarrolladores de la herramienta, por correo o por teléfono, realizar mantenimiento al sistema y darle solución a cualquier problema que surja con la aplicación.

Restricciones de diseño.

- El sistema debe desarrollarse utilizando la arquitectura Modelo Vista Controlador.
- Lenguaje de programación del lado del servidor: PHP 5.
- Como gestor de base de datos se utilizará PostgreSQL 8.3 o superior.
- Los marcos de trabajo que se utilizarán son: ExtJS 2.2 para la vista, Zend Framework 1.5 para el controlador y Doctrine 1.0.11 para el modelo.
- Como Entorno Integrado de Desarrollo (IDE) 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 2.0.

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 el acceso a la aplicación; esta consistirá en suministrar un nombre de usuario único y una contraseña que debe ser de conocimiento exclusivo de la persona que se autentica.

- Si el usuario no puede autenticarse o no se encuentra registrado no debe tener acceso a la aplicación.
- Si se intenta acceder al sistema desde una estación no autorizada, el acceso será denegado, y la operación será registrada en el sistema.
- El sistema no permitirá que el mismo usuario esté operando desde dos estaciones de trabajo diferentes.
- 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, hora de inicio y fin de la operación.
 - Operación realizada.
 - Oficina desde donde se realizó la operación.
 - Estación de trabajo desde la que operó.
- La versión MEP tendrá todos los privilegios y la de los organismos sólo aquellos que se relacionen con estos (se debe restringir los permisos para esta versión).

Software.

- Se debe disponer de sistema operativo Windows (2000, NT, XP), Windows Server 2000 o superior, GNU/Linux (cualquiera de sus distribuciones).
- Las computadoras cliente del sistema y ubicadas en el dominio de la organización, deben tener instalado el navegador de internet Mozilla Firefox y Adobe Reader 9.0 o superior.
- Servidor web: Apache 2.0 o superior.
- Gestor de base de datos: PostgreSQL 8.3 o superior.

Hardware.

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

- Se requiere de un servidor para bases de datos Xeon a 3.0 GHz, con 1 GB de memoria RAM y dos discos duros, uno de 36 GB y el otro de 250 GB con dos particiones.
- Se requiere de un servidor web Xeon a 3.0 GHz, con 1 GB de memoria RAM y tres discos duros de 36 GB cada uno con RAID 5.

Requisitos para la documentación de usuarios en línea y ayuda del sistema.

- El usuario debe tener acceso al Manual de usuario del producto final, en este debe documentarse la forma de utilizar el sistema.

Interfaz

- El sistema debe poseer una interfaz amigable al usuario, basada en web, brindando facilidades que permitan interactuar con el sistema de forma fácil y rápida.
- El sistema deberá poseer una interfaz que permita la creación y validación de consultas, permitiendo la creación de reportes sobre las operaciones.

Requisitos 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.

Requisitos legales, de derecho de autor y otros.

- La Universidad de las Ciencias de Informática tiene el derecho de autor sobre el Sistema Unificado de Gestión de Fuerza de Trabajo Calificada (GeForza).

2.6. Modelo del sistema

A continuación se realizará el modelado del sistema a través de un diagrama de casos de uso donde se representan gráficamente los procesos que se llevarán a cabo en el sistema y su interacción con los actores.

2.7. Actor del sistema

Cada trabajador del negocio (inclusive si fuera un sistema ya existente) que tiene actividades a automatizar es un candidato a actor del sistema. Si algún actor del negocio va a interactuar con el

sistema, entonces también será un actor del sistema. Los actores del sistema no son parte de él, pueden intercambiar información con él, ser un recipiente pasivo de información y representar el rol que juega una o varias personas, un equipo o un sistema automatizado. (I. JACOBSON 2000)

Actor	Descripción
Funcionario	Es una generalización de actores donde los mismos pueden ser: el funcionario del MEP, el funcionario del OACE o el funcionario del CAP.

Tabla 2.1. Descripción de los actores del sistema.

2.8. Diagrama de casos de uso del sistema



Figura 2. Diagrama de casos de uso del sistema.

2.9. Especificación de los casos de uso del sistema

A continuación se muestra la descripción abreviada de los casos de uso del sistema, las descripciones extendidas están disponibles en el Anexo 5.

Caso de uso:	Gestionar disponibilidad de noveno grado.
Actores:	Funcionario (inicia)
Resumen:	El caso de uso se inicia cuando el funcionario selecciona la opción FTC 1.1 Disponibilidad de noveno grado, selecciona la operación que desea realizar: Adicionar, Modificar, Eliminar o Buscar; introduce los datos necesarios y finaliza cuando el funcionario termina la operación.
Referencias	RF 4, RF 19.
Prioridad	Crítico

Tabla 2.2. Descripción abreviada del CU Gestionar disponibilidad de noveno grado.

Caso de uso:	Gestionar la disponibilidad de nivel medio
Actores:	Funcionario (inicia)
Resumen:	El caso de uso se inicia cuando el funcionario selecciona la opción FTC 1.2 Disponibilidad de nivel medio, selecciona la operación que desea realizar: Adicionar, Modificar, Eliminar o Buscar; introduce los datos necesarios y finaliza cuando el funcionario termina la operación.
Referencias	RF 5, RF 19.
Prioridad	Crítico

Tabla 2.3. Descripción abreviada del CU Gestionar disponibilidad de nivel medio.

Caso de uso:	Gestionar la disponibilidad de nivel superior
Actores:	Funcionario (inicia)
Resumen:	El caso de uso se inicia cuando el funcionario selecciona la opción FTC 1.3 Disponibilidad de nivel superior, selecciona la operación que desea realizar: Adicionar, Modificar, Eliminar o Buscar; introduce los datos necesarios y finaliza cuando el funcionario termina la operación.
Referencias	RF 6, RF 19.
Prioridad	Crítico

Tabla 2.4. Descripción abreviada del CU Gestionar disponibilidad de nivel superior.

Caso de uso:	Gestionar existencia de empleados
Actores:	Funcionario (inicia)
Resumen:	El caso de uso se inicia cuando el funcionario selecciona la opción FTC 2 Existencia de empleados, selecciona la operación que desea realizar: Adicionar, Modificar, Eliminar o Buscar; introduce los datos necesarios y finaliza cuando el funcionario termina la operación.
Referencias	RF 2, RF 19.
Prioridad	Crítico

Tabla 2.5. Descripción abreviada del CU Gestionar existencia de empleados.

Caso de uso:	Gestionar demanda de graduados
Actores:	Funcionario (inicia)
Resumen:	El caso de uso se inicia cuando el funcionario selecciona la opción FTC 3 Demanda de graduados, selecciona la operación que desea realizar: Adicionar, Modificar, Eliminar o Buscar; introduce los datos necesarios y finaliza cuando el funcionario termina la operación.
Referencias	RF 1, RF 19.
Prioridad	Crítico

Tabla 2.6. Descripción abreviada del CU Gestionar demanda de graduados.

Caso de uso:	Gestionar proyección de la demanda
Actores:	Funcionario (inicia)
Resumen:	El caso de uso se inicia cuando el funcionario selecciona la opción FTC 4 Proyección de la demanda, selecciona la operación que desea realizar: Adicionar, Modificar, Eliminar o Buscar; introduce los datos necesarios y finaliza cuando el funcionario termina la operación.
Referencias	RF 3, RF 31.
Prioridad	Crítico

Tabla 2.7. Descripción abreviada del CU Gestionar proyección de la demanda.

Caso de uso:	Generar reportes
Actores:	Funcionario (inicia)
Resumen:	El caso de uso se inicia cuando el funcionario escoge la opción Visor de reportes, selecciona el reporte que desea ver, finaliza con la visualización del mismo.

Referencias	RF 7, RF 8, RF 9, RF 10, RF 11, RF 12, RF 13, RF 14, RF 15, RF 16, RF 17, RF 18.
Prioridad	Auxiliar

Tabla 2.8. Descripción abreviada del CU Generar reportes.

Caso de uso:	Importar XML
Actores:	Funcionario (inicia)
Resumen:	El caso de uso se inicia cuando el funcionario escoge la opción Importar, selecciona la dirección donde está el informe que desea importar, finaliza con la restauración del mismo.
Referencias	RF 21.
Prioridad	Auxiliar

Tabla 2.9. Descripción abreviada del CU Importar XML.

Caso de uso:	Exportar XML
Actores:	Funcionario (inicia)
Resumen:	El caso de uso se inicia cuando el funcionario escoge la opción Exportar, selecciona la información que desea exportar y la dirección donde quiere guardar el archivo, finaliza al guardar la información.
Referencias	RF 20.
Prioridad	Auxiliar

Tabla 2.10. Descripción abreviada del CU Exportar XML.

2.10. Análisis de la factibilidad del sistema.

Para realizar el análisis de la factibilidad del sistema se utilizará el método Estimación por puntos de casos de uso. El mismo permite predecir el tamaño de un sistema a partir de las características de los requerimientos expresados en los casos de uso.

2.10.1. Estimación del esfuerzo.

Cálculo de puntos de casos de uso sin ajustar (UUCP):

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.

Actor	Descripción	Complejidad	Factor de peso
Funcionario	Una persona que interactúa con el sistema mediante una interfaz gráfica.	complejo	3

Tabla 2.11. Factor de peso de actores sin ajustar.

Cantidad de actores complejos: 1

$$UAW = \sum (\text{actores} \times \text{peso})$$

$$UAW = 1 \times 3$$

$$UAW = 3$$

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 del 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. Los criterios se muestran en la siguiente tabla:

Nombre del caso de uso	Descripción	Complejidad	Factor de peso
Gestionar disponibilidad de graduados de noveno grado.	6 transacciones	medio	10
Gestionar disponibilidad de graduados de nivel medio.	6 transacciones	medio	10
Gestionar disponibilidad de graduados de nivel superior.	6 transacciones	medio	10
Gestionar existencia de empleados.	6 transacciones	medio	10
Gestionar demanda de graduados por carreras.	6 transacciones	medio	10
Gestionar proyección de la demanda.	6 transacciones	medio	10
Generar reportes	23 transacciones	complejo	15
Importar XML	2 transacciones	simple	5
Exportar XML	2 transacciones	simple	5

Tabla 2.12. Factor de peso de los casos de uso sin ajustar.

Cantidad de casos de uso complejos: 1.

Cantidad de casos de uso medios: 6.

Cantidad de casos de uso simples: 2.

$$UUCW = \sum CU \text{ complejos} \times \text{peso} + \sum CU \text{ medios} \times \text{peso} + \sum CU \text{ simples} \times \text{peso}$$

$$UUCW = 1 \times 15 + 6 \times 10 + 2 \times 5$$

$$UUCW = 85$$

Por tanto:

$$UUCP = UAW + UUCW$$

$$UUCP = 3 + 85$$

$$UUCP = 88$$

Cálculo de puntos de casos de uso ajustados (UCP):

Una vez que se tienen los puntos de casos de uso sin ajustar, se debe ajustar este valor mediante la siguiente ecuación:

$$UCP = UUCP \times TCF \times EF$$

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	0	0
T2	Objetivos de <i>performance</i> o tiempo de respuesta	1	1	1
T3	Eficiencia del usuario final	1	1	1
T4	Procesamiento interno complejo	1	0	0
T5	El código debe ser reutilizable	1	4	4
T6	Facilidad de instalación	0.5	2	1
T7	Facilidad de uso	0.5	5	2.5
T8	Portabilidad	2	2	4
T9	Facilidad de cambio	1	1	1
T10	Concurrencia	1	1	1
T11	Incluye objetivos especiales de seguridad	1	2	2
T12	Provee acceso directo a terceras partes	1	0	0
T13	Se requieren facilidades especiales de entrenamiento a usuarios	1	0	0

Tabla 2.13. Factor de complejidad técnica.

El factor de complejidad técnica se calcula mediante la siguiente ecuación:

$$TCF = 0.6 + 0.01 \times \sum (Peso_i \times Valor\ asignado_i)$$

$$TCF = 0.6 + 0.01 \times (0 + 1 + 1 + 0 + 4 + 1 + 2.5 + 4 + 1 + 1 + 2 + 0 + 0)$$

$$TCF = 0.6 + (0.01 \times 17.5)$$

$$TCF = 0.6 + 0.175$$

$$TCF = 0.775$$

Factor de ambiente (EF):

Para calcular este coeficiente se analizan una serie de factores que pueden determinar el tiempo requerido para el desarrollo del sistema, teniendo en cuenta aspectos como habilidades, conocimientos, entre otros, de los involucrados en la realización del sistema. A estos factores se les asocia un valor de 0 a 5 que determina el vínculo del mismo con las características deseadas para del sistema.

Factor	Descripción	Peso	Valor asignado	Peso*Valor
E1	Familiaridad con el modelo de proyecto utilizado	1.5	5	7.5
E2	Experiencia en la aplicación	0.5	2	1
E3	Experiencia en orientación a objetos	1	3	3
E4	Capacidad del analista líder	0.5	5	2.5
E5	Motivación	1	5	5
E6	Estabilidad de los requerimientos	2	4	8
E7	<i>Personal part-time</i>	-1	0	0

E8	Dificultad del lenguaje de programación	-1	0	0
----	---	----	---	---

Tabla 2.14. Factor de ambiente.

El factor de ambiente resulta:

$$EF = 1.4 - 0.03 \times \sum (\text{Peso} \times \text{Valor asignado})$$

$$EF = 1.4 - 0.03 \times (7.5 + 1 + 3 + 2.5 + 5 + 8 + 0 + 0)$$

$$EF = 1.4 - (0.03 \times 27)$$

$$EF = 1.4 - 0.81$$

$$EF = 0.59$$

Finalmente, los puntos de casos de uso ajustados resultan:

$$UCP = UUCP \times TCF \times EF$$

$$UCP = 88 \times 0.775 \times 0.59$$

$$UCP = 40.238$$

El esfuerzo en horas-hombre está dado por:

$$E = UCP \times CF$$

Donde:

E: Esfuerzo.

UCP: Puntos de casos de uso ajustados.

CF: Factor de conversión.

Para calcular el factor de conversión (CF):

$$\text{Total } EF = \text{Cantidad } EF < 3 \text{ (entre } E_1 - E_6) + \text{Cantidad } EF > 3 \text{ (entre } E_7, E_8)$$

$$\text{Total } EF = 0 + 2$$

$$\text{Total } EF = 2$$

CF = 20 horas/hombre (si Total EF ≤ 2)

CF = 20 horas/hombre (si Total EF = 3 o Total EF = 4)

CF = abandonar o cambiar el proyecto (si Total EF ≥ 5)

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

$$E = UCP \times CF$$

$$E = 40.238 \times 20$$

$$E = 804.76 \text{ horas/hombre}$$

Se considera que este esfuerzo representa un porcentaje del esfuerzo total del proyecto. Para una estimación más completa de la duración total del proyecto se debe 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. Teniendo en cuenta los siguientes valores porcentuales para la distribución del esfuerzo entre las diferentes actividades de un proyecto, que estadísticamente se considera aceptable, se obtiene:

Actividad	Porcentaje	Horas-hombre
Análisis	10.00 %	201.19
Diseño	20.00 %	402.38
Programación	40.00 %	804.76
Pruebas	15.00 %	301.785
Sobrecarga (otras actividades)	15.00 %	301.785
Total	100 %	2011.9

Tabla 2.15. Distribución del esfuerzo.

Por tanto, el esfuerzo total para la realización del módulo Sistema Matriz es de 2011.9 horas/hombre.

2.10.2. Estimación del tiempo de desarrollo.

Para la realización del módulo Sistema Matriz se requiere de un esfuerzo total de 2011.9 horas/hombre, si se considera que trabajan dos hombres, se debe dividir este valor entre dos, lo que resulta **1005.95 horas/hombres**.

Este valor resultante se divide entre 8 horas que son las horas que se trabaja diariamente y se obtiene la cantidad de horas trabajadas entre 2 hombres: **125.74375 horas/2 hombres**.

Este valor obtenido se divide entre los 5 días laborables que tiene la semana, y se obtiene las semanas trabajadas entre dos hombres: **25.14875 semanas/2 hombres.**

Luego, se divide este valor entre la cantidad de semanas (4) que tiene un mes para saber cuántos meses se demorarían dos hombres en realizar dicho trabajo: **6.2871875 meses/2 hombres.**

Considerando que trabajan 2 personas, 40 horas como promedio en la semana, este sistema debe terminarse en 25.14875 semanas, lo que representa un aproximado de 26 semanas, que serán unos seis meses y medios aproximadamente.

2.10.3. Análisis de costos.

Para realizar el análisis de costos se tuvo en cuenta que el trabajo fue desarrollado por dos estudiantes de la Universidad de Ciencias Informáticas, donde el salario promedio sería el estipendio mensual para un estudiante de quinto año, por lo tanto, el costo del proyecto se calcula:

$$\mathbf{Costo = CM \times T}$$

Donde:

CM: Costo mensual.

T: tiempo estimado.

T = 6.5 meses.

Para calcular el costo mensual:

$$\mathbf{CM = CH \times S}$$

Donde:

CH: Cantidad de hombres.

CH = 2.

S: Salario promedio.

S = \$100.00.

Por tanto:

$$CM = CH \times S$$

$$CM = 2 \times 100$$

$$CM = 200 \text{ \$/mes}$$

Entonces el costo del proyecto sería:

$$\text{Costo} = CM \times T$$

$$\text{Costo} = 200 \times 6.5$$

$$\text{Costo} = \$1300$$

Considerando el análisis anterior se puede decir que para el desarrollo del módulo Sistema Matriz no se necesita disponer de grandes gastos de recursos, ni tampoco de mucho tiempo pues se realizaría en aproximadamente seis meses y medio a un costo de mil trescientos pesos.

2.10.4. Beneficios tangibles e intangibles.

➤ Tangibles.

Como el aporte más valioso se tiene la obtención de una aplicación con tecnología web, que gestiona toda la información necesaria para la realización del Plan de Distribución de graduados de nivel superior, el Plan de Ingreso a la Educación Superior del Curso Regular Diurno y el Plan de Continuidad de Estudios de los estudiantes de noveno grado, desarrollados anualmente en el departamento Fuerza de Trabajo Calificada del MEP ajustada a los nuevos clasificadores y nomencladores, de forma organizada, rápida y segura, con una interfaz amigable y de fácil navegabilidad.

➤ Intangibles.

- Agilidad en el proceso de realización de los planes asegurando mayor calidad en el mismo.
- Incremento de la seguridad del sistema, sustituyendo los disquetes y controlando el intercambio de información por la red con usuarios autenticados.
- Corrección de los errores encontrados en la aplicación anterior.

2.11. Conclusiones

Como resultado de la realización de este capítulo se precisó hacer un modelo de dominio, representando los principales conceptos involucrados en el negocio, se describieron los requisitos funcionales y no funcionales del sistema, se realizó el modelado del sistema describiendo los actores y casos de uso implicados en el mismo. Además, se realizó el análisis de la factibilidad del sistema demostrando que el desarrollo del módulo Sistema Matriz es factible. Quedando planteadas de esta forma, las condiciones y características del sistema propuesto y marcando la línea a seguir para la realización del análisis y el diseño de la aplicación.

Capítulo III: Análisis y diseño.

3.1. Introducción

En este capítulo se realizará el análisis y diseño del sistema que constituyen la vista lógica de la arquitectura. Se mostrarán los diagramas de clases y los diagramas de interacción tanto para el análisis como para el diseño, así como el modelo de datos y el modelo de despliegue. Se presentarán además los patrones de diseño usados para el trabajo con el Zend Framework.

3.2. Análisis.

Durante el análisis, se analizan los requisitos que se describieron en la captura de requisitos, refinándolos y estructurándolos con el objetivo de conseguir una comprensión más precisa y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema entero, incluyendo su arquitectura. (I. JACOBSON 2000)

Para ello se definen las clases del análisis que representan una abstracción de una o varias clases y/o subsistemas del diseño del sistema y que pueden clasificarse en:

- Clases de interfaz: Modelan la interacción entre el sistema y sus actores.
- Clases de entidad: Modelan información que posee larga vida y que es, a menudo persistente.
- Clases de control: Representan coordinación, secuencia, transacciones, y control de otros objetos. Se usan para encapsular el control de un caso de uso en concreto.

3.2.1. Diagrama de clases del análisis.

Los diagramas de clases del análisis muestran sus clases participantes y sus relaciones. Se utilizan para coordinar todos los requisitos sobre una clase y sus objetos que pueden tener diferentes casos de uso. (I. JACOBSON 2000)

Durante la realización del análisis se utilizan los diagramas de colaboración, pues los mismos muestran las interacciones entre objetos creando enlaces entre ellos y añadiendo mensajes a estos enlaces.

Como objeto de estudio para representar el ciclo de vida de los casos de uso se selecciona Gestionar demanda de graduados. A continuación se representa el diagrama de clases del análisis de dicho caso de uso. En el Anexo 6 se pueden consultar los diagramas de clases para el resto de los casos de uso y los

diagramas de colaboración para cada uno de los escenarios de los casos de uso están disponibles en el Anexo 7.

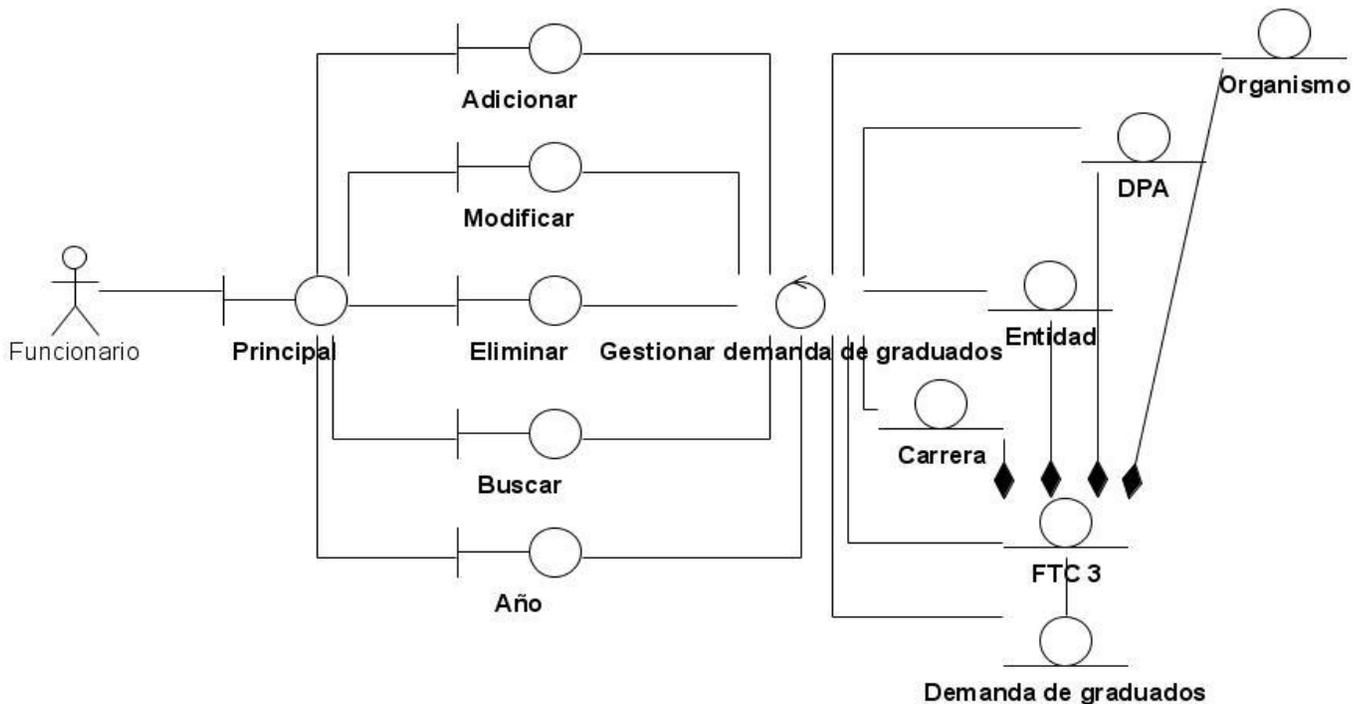


Figura 3. Diagrama de clases del análisis CU Gestionar demanda de graduados.

3.3. Diseño.

En el diseño se modela el sistema y encuentra la forma (incluida la arquitectura) para que soporte todos los requisitos incluyendo los requisitos no funcionales y otras restricciones que se le suponen. (I. JACOBSON 2000)

Se realizará el diseño del módulo Sistema Matriz utilizando ingeniería inversa a partir de la implementación del mismo.

3.3.1. Patrones de diseño.

Un patrón de diseño es una solución estándar para un problema común de programación. Propone una forma de reutilizar la experiencia de los desarrolladores, para ello clasifica y describe formas de solucionar problemas que ocurren de forma frecuente en el desarrollo. Por tanto, están basados en la recopilación del conocimiento de los expertos en desarrollo de software.

Para el diseño de la propuesta de solución del módulo Sistema Matriz, se tienen en cuenta los patrones de arquitectura Modelo Vista Controlador, que implementa Zend Framework y Cliente/Servidor que se utiliza en aplicaciones web.

En la realización del diseño se aplicaron además patrones GRASP que describen los principios fundamentales de asignación de responsabilidades a objetos y patrones de diseño (GOF).

Patrones GRASP:

Experto: este patrón se tiene en cuenta para la asignación de responsabilidades a las clases de forma tal que las mismas contengan la información necesaria para poder ejecutar una acción específica. Las clases que brinda el *framework* ExtJS se encargarán de visualizar las interfaces ya que cuentan con la información para crear los diferentes componentes visuales, las clases controladoras del Zend Framework manejarán las peticiones del cliente, y las clases que genera y utiliza Doctrine serán las encargadas del acceso a datos pues contienen y representan los datos que manejará el sistema. 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 y fácil de mantener.

Creador: este patrón se tiene en cuenta para la asignación de responsabilidades a las clases 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 se encarga de instanciar al controlador frontal (`Zend_Controller_Front`), este último es el encargado de instanciar las clases controladoras y estas, a su vez, instancian objetos de la clase `Zend_View`. El uso de este patrón permite crear las dependencias mínimas necesarias entre las clases, favoreciendo al mantenimiento del sistema.

Bajo acoplamiento: este patrón brinda como solución asignar responsabilidades de manera que las clases no dependan fuertemente unas de otras. De esta forma, las clases son fáciles de entender por separadas, fáciles de reutilizar y no se afectan por cambios de otros componentes. Dicho patrón se tiene en cuenta debido a la importancia de realizar un diseño de clases independientes que soporten los cambios.

Alta cohesión: este patrón propone asignar la responsabilidad de manera que la complejidad se mantenga dentro de límites manejables asumiendo solamente las responsabilidades que deben manejar, evadiendo un trabajo excesivo. Su utilización mejora la claridad y facilidad con que se entiende el diseño, simplifica el

mantenimiento y las mejoras de funcionalidad, generan un bajo acoplamiento, soporta mayor capacidad de reutilización.

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 (`Zend_Controller_Front`) que implica que todas las solicitudes son dirigidas a un único *script* PHP que se encarga de instanciar al controlador frontal y redirigir las llamadas.

Patrones GOF:

Decorator (Decorador): este patrón permite añadir funcionalidad a una clase dinámicamente. Zend Framework implementa dicho patrón en la clase `Zend_View`, encargada de asignarle responsabilidades a objetos de manera dinámica y configurarlos con nuevos atributos.

Singleton (Instancia única): este patrón garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Zend Framework tiene una instancia única del controlador frontal (`Zend_Controller_Front`) disponible, mediante este patrón, para lograr una vía de entrada única a las solicitudes.

Facade (Fachada): este patrón proporciona una interfaz para crear familias de objetos relacionados o dependientes sin especificar su clase concreta. Permite configurar en tiempo de ejecución un sistema con una familia u otra de objetos. Además, garantiza que un conjunto de clases se usen a la vez. Zend Framework provee una API para el acceso a datos conformada por un conjunto de clases que implementa dicho patrón.

Abstract Factory (Fabricación abstracta): este patrón simplifica los accesos a las clases de la capa de acceso a datos proporcionando un objeto que todas las clases de capas superiores utilizarán para acceder a las clases contenidas en la capa del modelo. Define una interfaz de más alto nivel que permite usar el sistema más fácil. El objetivo de la aplicación de este patrón es reducir la dependencia entre clases. Se utilizará una clase intermediaria entre las clases controladoras de Zend Framework y las de acceso a datos de Doctrine, la que brindará, de las operaciones de acceso a datos, sólo las que necesiten los controladores para su funcionamiento, lo que reduce la dependencia de estos entre las múltiples clases de acceso a datos existentes en el sistema.

3.3.2. Diagrama de clases del diseño.

Los diagramas de clases del diseño muestran sus clases participantes, subsistemas y sus relaciones. Se utilizan para coordinar todos los requisitos que diferentes realizaciones de caso de uso imponen a una clase, sus objetos y los subsistemas que contiene. (I. JACOBSON 2000)

En el diseño, los diagramas de interacción se representan mediante diagramas de secuencia, pues el objetivo fundamental es encontrar secuencias de interacciones detalladas y ordenadas en el tiempo.

A continuación se representa el diagrama de clases del diseño del caso de uso Gestionar demanda de graduados, el resto de los diagramas de clases están disponibles en el Anexo 8, en el Anexo 9 se encuentra la descripción de cada una de las clases de los diagramas de clases del diseño de cada caso de uso y los diagramas de secuencia para cada uno de los escenarios de los casos de uso están disponibles en el Anexo 10.

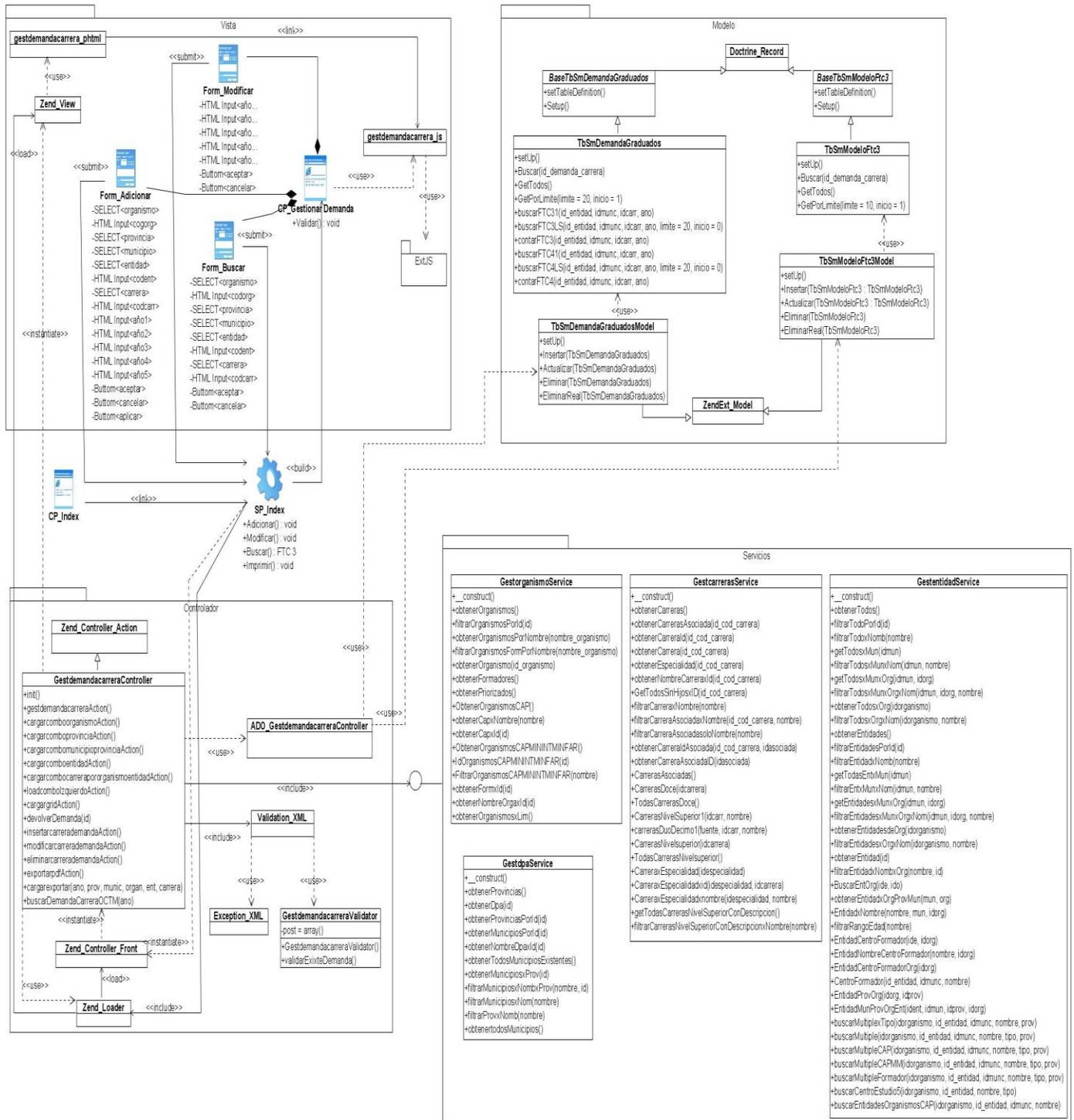


Figura 4. Diagrama de clases del diseño CU Gestionar demanda de graduados.

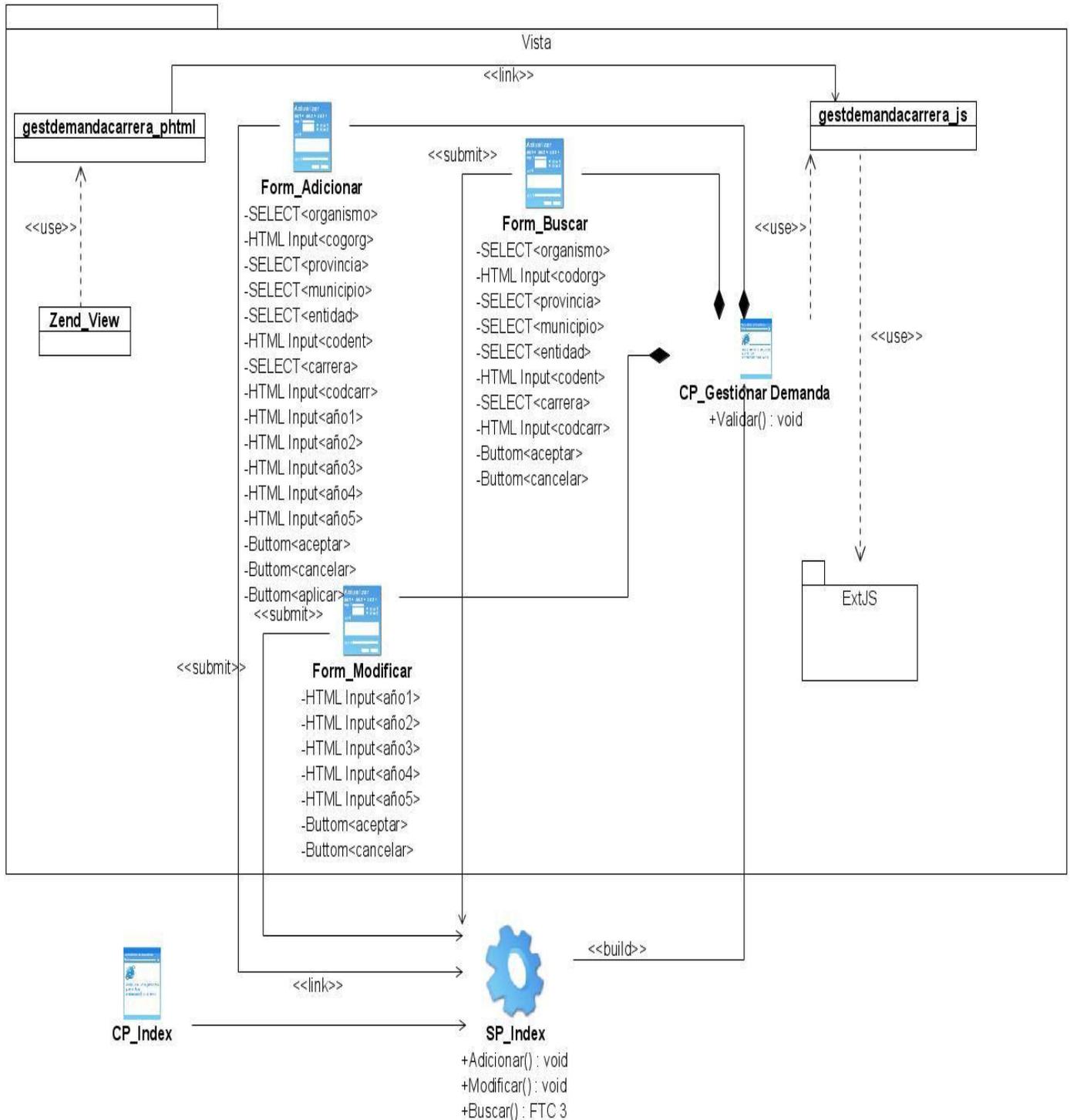


Figura 5. Diagrama de clases del diseño CU Gestionar demanda de graduados (vista).

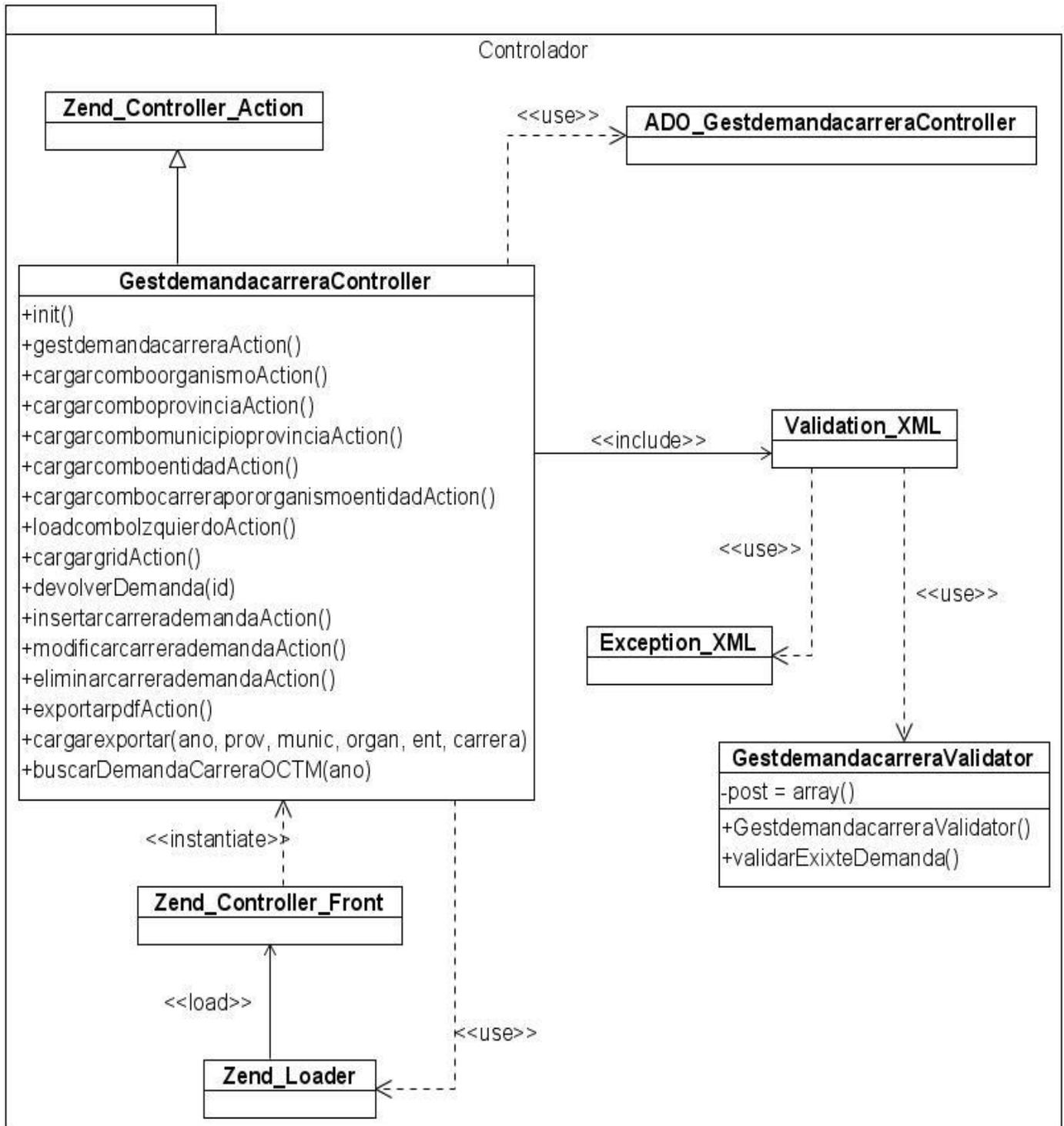


Figura 6. Diagrama de clases del diseño CU Gestionar demanda de graduados (controlador).

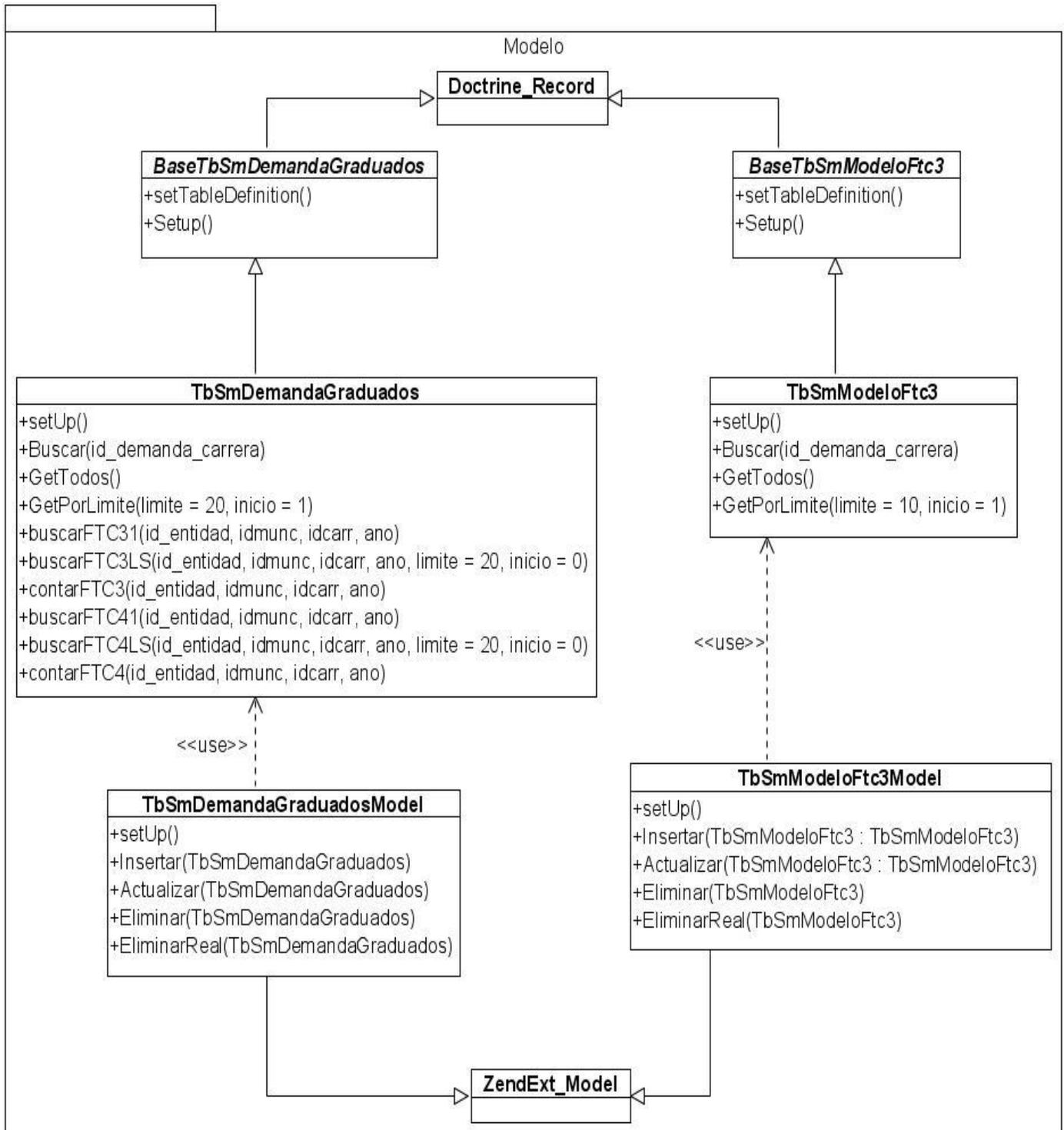


Figura 7. Diagrama de clases del diseño CU Gestionar demanda de graduados (modelo).



Figura 8. Diagrama de clases del diseño CU Gestionar demanda de graduados (servicios).

3.4. Modelo de datos.

El modelo de datos describe la representación lógica y física de los datos persistentes usados por la aplicación. (I. JACOBSON 2000)

A continuación se representa el modelo físico de la base de datos, el modelo lógico está disponible en el Anexo 11 y en el Anexo 12 se encuentra la descripción de cada una de las tablas que conforman la base de datos.

3.4.1. Modelo físico.

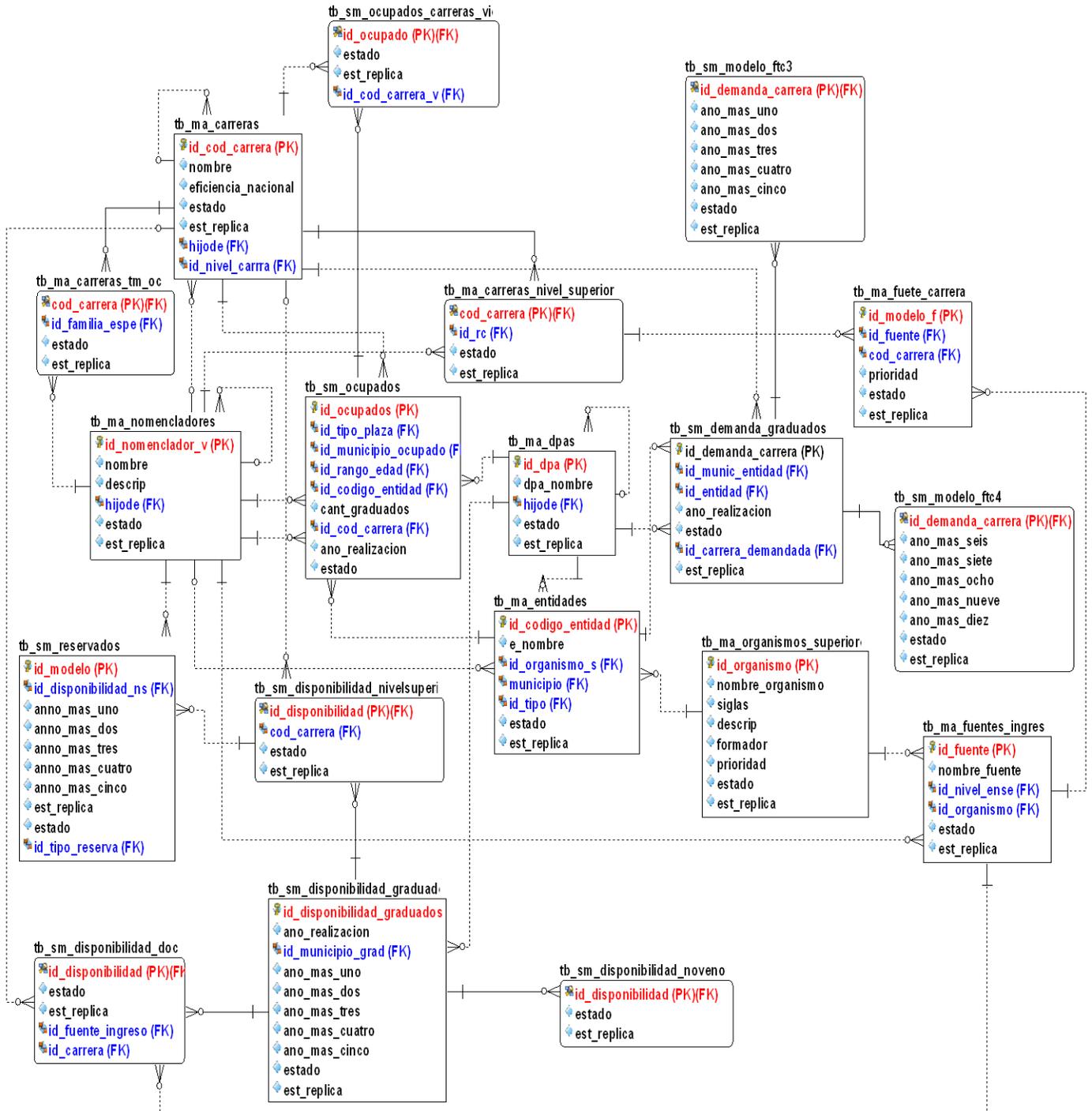


Figura 9. Modelo físico de la base de datos.

3.5. Modelo de despliegue.

El modelo de despliegue describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. (I. JACOBSON 2000)

3.5.1. Diagrama de despliegue.

Es un diagrama que muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos. Gráficamente, un diagrama de despliegue es una colección de nodos y arcos. (I. JACOBSON 2000)

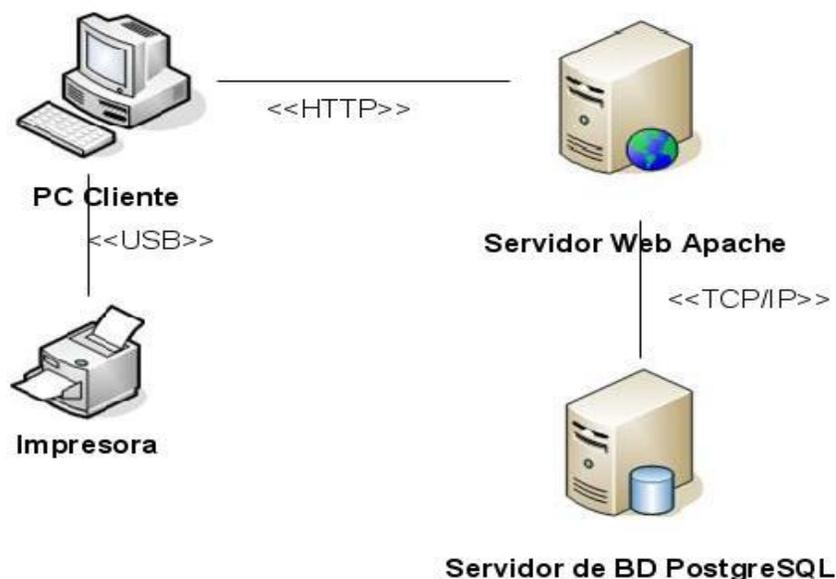


Figura 10. Diagrama de despliegue.

3.6. Conclusiones.

Como resultado de la elaboración de este capítulo se logró la terminación de la etapa de análisis y diseño del subsistema, se realizaron los diagramas de clases, tanto para el análisis como para el diseño de cada caso de uso identificado. Se realizaron además, los diagramas de interacción (diagramas de secuencia y diagrama de colaboración) para cada escenario de los casos de uso. Además, se hizo la descripción de la arquitectura y los patrones de diseño utilizados, se muestran los modelos físico y lógico de datos propuestos para satisfacer las necesidades del subsistema y el diagrama de despliegue.

Capítulo IV: 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 diseño y continúa con la implementación del sistema en términos de componentes para terminar en la aplicación de las pruebas de caja negra en pos de garantizar la calidad del sistema. Se obtendrán también los diagramas de componentes.

4.2. Implementación.

El flujo de trabajo Implementación enmarca el comienzo de la fase de Construcción. El propósito general de la implementación es desarrollar la arquitectura y el sistema como un todo. La aplicación que se desarrolle debe tener la calidad requerida para su uso y cumplir con los requisitos 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. (I. JACOBSON 2000)

A continuación se representa el diagrama de componentes del caso de uso Gestionar demanda de graduados, el resto está disponible en el Anexo 13.

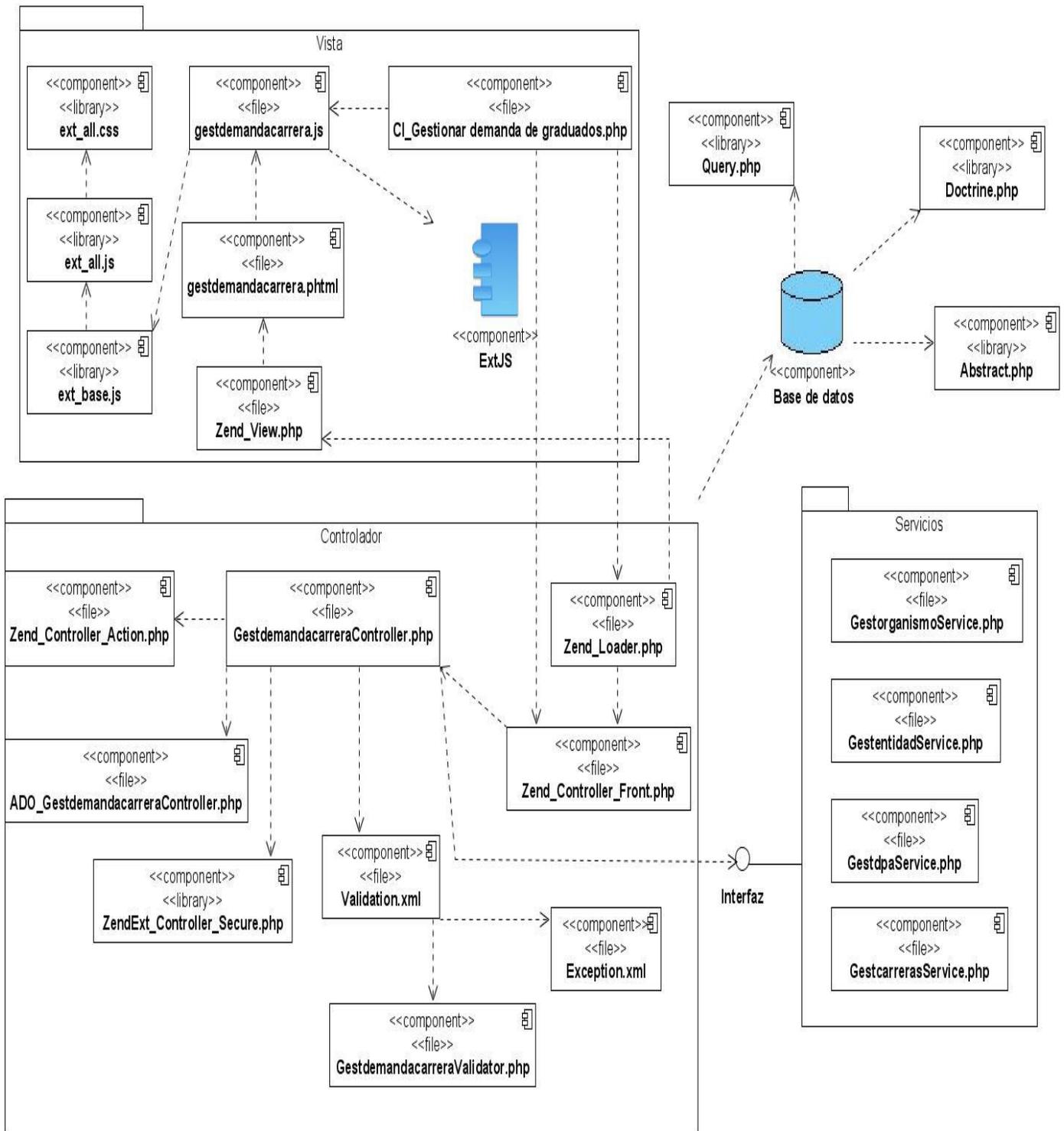


Figura 11. Diagrama de componentes CU Gestionar demanda de graduados.

4.2.2. Tratamiento de errores.

Con el objetivo de reducir las posibilidades de que se introduzca información errónea, por parte del usuario a la base de datos asociada al sistema, se dispone de una serie de validaciones que dan tratamiento a dichos errores. Eventualmente se valida que no queden campos vacíos, que no se introduzcan letras en los campos de información numérica, que los datos tengan el formato esperado y que no se inserten datos irrelevantes en la base de datos. En estos casos se informa al usuario mediante un mensaje que ha realizado alguna operación incorrecta, especificando claramente el error cometido.

4.2.3. Seguridad.

El módulo Sistema Matriz está integrado al Módulo de Administración de GeForza, el cual se encarga de centralizar la seguridad del sistema. Para garantizar la misma, este módulo define la asignación de roles a los usuarios, de forma tal, que estos sólo tengan acceso a las funcionalidades que le sean permitidas. Los usuarios se autenticarán antes de acceder a cualquier funcionalidad posibilitando el control de acceso a las mismas, de esta forma el funcionario del MEP tiene acceso pleno, mientras que los funcionarios de los OACE y los CAP sólo accederán a las funcionalidades que les sean permitidas. Los usuarios y contraseñas se guardarán en la base de datos, estas últimas codificadas, evitando que alguien pueda tener acceso a esa información. También el Módulo de Administración define trazas, que permiten conocer detalladamente las acciones realizadas por un usuario y sobre qué funcionalidades, en un momento determinado.

4.2.4. Estándares de codificación.

➤ Vista.

Para esta capa se definen como estándares los siguientes:

- El nombre de las clases debe comenzar con *gest* seguido por el *nombre*. Ejemplo: *gestocupados*.
- 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*.
 - Los *combobox* se nombrarán: *cb+Nombre*. Ejemplo: *cbProvincia*.

- Los *textfield*s se nombrarán: *txf+Nombre*. Ejemplo: *txfOrganismo*.
- Los *gridpanels* se nombrarán: *gp+Nombre*. Ejemplo: *gpInformacion*.
- Las ventanas se nombrarán: *win+Nombre*. Ejemplo: *winAddDemanda*.

➤ **Controlador.**

Para esta capa se definen:

- El nombre de la clase comenzará con *Gest* seguida por el *nombre* y la palabra *Controller*. Ejemplo: *GestocupadosController*.
- 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 el *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*.

➤ **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.

En el desarrollo de un software, el proceso de prueba es clave a la hora de detectar errores o fallas. Las pruebas son de gran importancia en la garantía del software, una selección cuidadosa de los datos de prueba puede ofrecer mucha confianza en cuanto al desempeño que posee el programa. Esto, asociado a un determinado mecanismo de comprobación de errores, puede producir software más confiable. El objetivo principal de la realización de pruebas es descubrir la mayor cantidad posible de defectos del

software.

4.3.1. Método de prueba.

Pruebas de caja negra (pruebas de comportamiento): son diseñadas para validar los requisitos funcionales sin fijarse en el funcionamiento externo del programa. Se centran en el ámbito de información del programa, de forma que se proporcione una cobertura completa de prueba. No son una alternativa a las técnicas de pruebas de caja blanca. (PRESSMAN 2002)

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están:

- Los métodos de pruebas basados en grafos: exploran las relaciones entre los objetos del programa y su comportamiento.
- La partición equivalente: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- El análisis de valores límite: prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- La prueba de la tabla ortogonal: suministra un método sistemático y eficiente para probar sistemas con un número reducido de parámetros de entrada.

Se utilizará el método de caja negra, aplicando la técnica de particiones equivalentes que está definido en la estrategia de pruebas trazada por el proyecto Fuerza de Trabajo Calificada.

4.3.2. Casos de prueba.

Un caso de prueba describe una forma de probar el sistema, incluyendo la entrada o resultado con la que se ha de probar y las condiciones bajo las que ha de probarse. (I. JACOBSON 2000)

A continuación se expone el caso de prueba para el caso de uso Gestionar demanda de graduados, el resto está disponible en el Anexo 14.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC1: Adicionar	EC 1.1: Adicionar una demanda correctamente pulsando el botón aplicar.	Se adiciona una nueva demanda al sistema y se muestra un mensaje de confirmación.	<ul style="list-style-type: none"> • Selecciona la opción “Adicionar”. • Llena los campos. • Selecciona “Aplicar”. • Muestra el mensaje “Insertado satisfactoriamente”, mantiene el organismo, provincia, municipio y entidad, limpia el resto de los campos dando la opción de adicionar nuevamente. • Selecciona “Aceptar”.
	EC 1.2: Adicionar una demanda correctamente pulsando el botón aceptar.	Se adiciona una nueva demanda al sistema.	<ul style="list-style-type: none"> • Selecciona la opción “Adicionar”. • Llena los campos. • Selecciona “Aceptar”. • Muestra el mensaje “Insertado satisfactoriamente”. • Selecciona “Aceptar”.
	EC 1.3: Adicionar una demanda dejando campos en blanco.	No se adiciona la demanda al sistema.	<ul style="list-style-type: none"> • Selecciona la opción “Adicionar”. • Llena los datos dejando campos vacíos. • Selecciona “Aplicar” o “Aceptar”. • Muestra un mensaje de error notificando al usuario que los datos son incorrectos. • Selecciona “Aceptar”.
	EC 1.4: Adicionar una demanda pasando datos existentes.	No se adiciona la demanda al sistema.	<ul style="list-style-type: none"> • Selecciona la opción “Adicionar”. • Llena los campos. • Selecciona “Aplicar” o “Aceptar”. • Muestra un mensaje indicando que ya existe. • Selecciona “Aceptar”.
	EC 1.5: Adicionar una demanda poniendo todas las cantidades en cero.	No se adiciona la demanda al sistema.	<ul style="list-style-type: none"> • Selecciona la opción “Adicionar”. • Llena los campos. • Selecciona “Aplicar” o “Aceptar”. • Muestra un mensaje indicando que al menos una cantidad debe ser distinta de cero. • Selecciona “Aceptar”.
	EC 1.6: Cancelar la operación.	No se adiciona la demanda al sistema.	<ul style="list-style-type: none"> • Selecciona la opción “Adicionar”. • Llena los campos. • Selecciona “Cancelar”. • Se cierra la ventana para adicionar.

SC2: Modificar	EC 2.1: Modificar demanda correctamente.	Se modifica una demanda al sistema y se muestra un mensaje de confirmación.	<ul style="list-style-type: none"> • Selecciona una de las demandas existentes. • Selecciona la opción "Modificar". • Llena los campos a modificar. • Selecciona "Aceptar". • Muestra un mensaje "Se ha modificado satisfactoriamente". • Selecciona "Aceptar".
	EC 2.2: Modificar demanda pasando datos existentes.	No se modifica demanda al sistema.	<ul style="list-style-type: none"> • Selecciona una de las demandas existentes. • Selecciona la opción "Modificar". • Llena los campos a modificar. • Selecciona "Aceptar". • Muestra un mensaje indicando que no se ha modificado. • Selecciona "Aceptar".
	EC 2.3: Modificar demanda pasando todas las cantidades en cero.	No se modifica demanda al sistema.	<ul style="list-style-type: none"> • Selecciona una de las demandas existentes. • Selecciona la opción "Modificar". • Llena los campos a modificar. • Selecciona "Aceptar". • Muestra un mensaje indicando que al menos una de las cantidades debe ser distinta de cero. • Selecciona "Aceptar".
	EC 2.4: Cancelar operación.	No se modifica demanda al sistema.	<ul style="list-style-type: none"> • Selecciona una de las demandas existentes. • Selecciona la opción "Modificar". • Selecciona "Cancelar". • Se cierra la ventana para modificar.
SC3: Eliminar	EC 3.1: Eliminar demanda correctamente.	Se elimina la demanda del sistema y se muestra un mensaje de confirmación.	<ul style="list-style-type: none"> • Selecciona una o varias de las demandas existentes. • Selecciona la opción "Eliminar". • Muestra el mensaje "Seguro que desea eliminar la demanda". • Selecciona "Aceptar". • Muestra el mensaje "Eliminada satisfactoriamente". • Selecciona "Aceptar".
	EC 3.2: Cancelar la operación.	No se elimina la demanda del sistema.	<ul style="list-style-type: none"> • Selecciona una o varias de las demandas existentes. • Selecciona la opción "Eliminar". • Muestra el mensaje "Seguro que desea eliminar la demanda". • Selecciona "Cancelar". • Se cierra la ventana para eliminar.

SC4: Buscar	EC 4.1: Buscar correctamente.	Se busca una demanda en el sistema.	<ul style="list-style-type: none"> • Selecciona el (los) criterio (s) de búsqueda deseado (s). • Selecciona "Aceptar". • Muestra todas las demandas existentes en el año con que se está trabajando y correspondiente a la selección hecha.
	EC 4.2: Buscar sin pasar criterio de búsqueda.	No se busca demanda en el sistema.	<ul style="list-style-type: none"> • Selecciona la opción "Aceptar". • Muestra un mensaje indicando que debe seleccionar al menos un campo. • Selecciona "Aceptar".
	EC 4.3: Buscar sin existir disponibilidades correspondientes.	No se busca una demanda en el sistema.	<ul style="list-style-type: none"> • Selecciona la opción "Buscar". • Selecciona el (los) criterio (s) de búsqueda deseado (s). • Muestra un mensaje notificando que no hay resultados para mostrar. • Seleccionar "Aceptar".
	EC 4.4: Cancelar la operación.	No se busca demanda en el sistema.	<ul style="list-style-type: none"> • Selecciona el (los) criterio (s) de búsqueda deseado (s). • Selecciona la opción "Cancelar". • Se cierra la ventana para buscar.
	EC 4.5: Limpiar búsqueda.	Se limpian los datos introducidos para la búsqueda actual.	<ul style="list-style-type: none"> • Selecciona la opción "Buscar". • Selecciona el (los) criterio (s) de búsqueda deseado (s). • Selecciona "Aceptar". • Muestra todas las demandas existentes del año seleccionado, correspondiente a los datos seleccionados. • Selecciona la opción "Limpiar búsqueda". • Vuelve a la ventana principal para Gestionar demanda de graduados por carreras mostrando todas las demandas existentes.
SC 5: Exportar	EC 5.1: Exportar.	Se crea un documento pdf.	<ul style="list-style-type: none"> • Selecciona la opción "Exportar". • Crea un documento pdf con los datos seleccionados por el funcionario, teniendo habilitadas todas las opciones que brinda el pdf.
SC 6: Año	EC 6.1: Mostrar demandas por año.	Se muestra una lista de demandas.	<ul style="list-style-type: none"> • Selecciona un año. • Muestra una lista con todas las demandas del año seleccionado.
	EC 6.2: No mostrar demandas por año.	No se muestran demandas.	<ul style="list-style-type: none"> • Selecciona un año. • Muestra un mensaje notificando que no hay resultados para mostrar.

Tabla 4.1. Caso de prueba CU Gestionar demanda de graduados.

4.4. Resultados de las pruebas.

A partir de los casos de prueba se realizó una prueba exploratoria y tres iteraciones resultando 42 no conformidades, en total, donde el flujo de errores estuvo encaminado, en su mayoría, a errores de validación y errores funcionales.

En la prueba exploratoria se encontraron 23 no conformidades, de las cuales 10 eran funcionales y el resto de validación; de ellas se solucionaron 15. Durante la primera iteración se encontraron 3 no conformidades de tipo funcional y 7 de validación para un total de 10 no conformidades, de las cuales se solucionaron 6. En la segunda iteración, se encontraron 6 no conformidades, 1 de tipo funcional y el resto de validación, se resolvieron 5. En la tercera iteración, se encontraron solamente 3 no conformidades de validación quedando resueltas todas ellas, como muestra la figura.

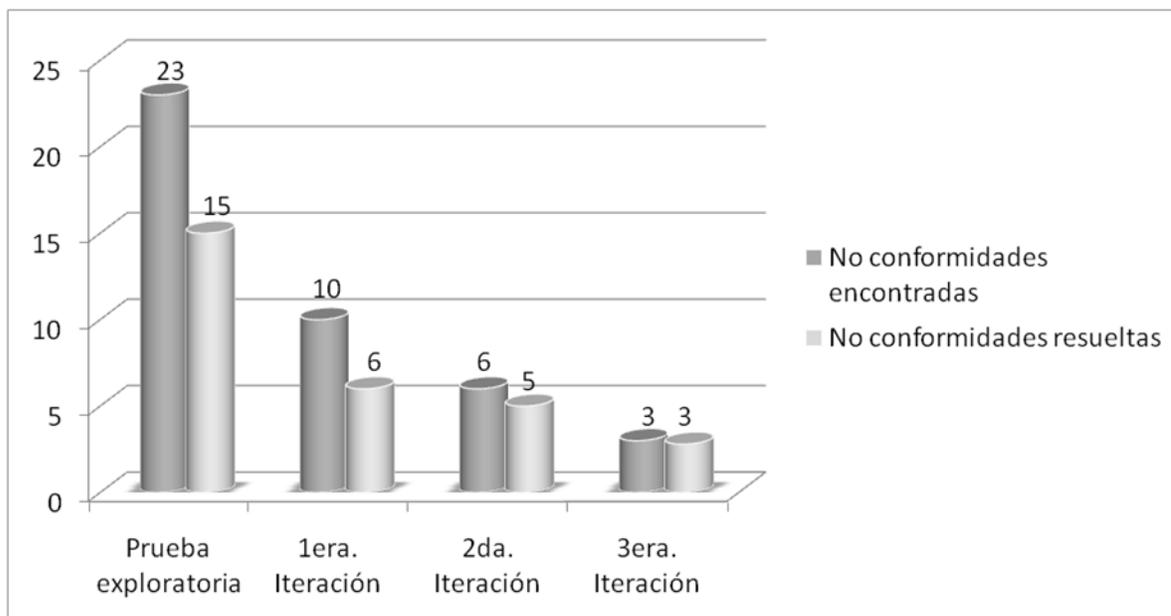


Figura 12. Resultados de las pruebas.

Puede decirse entonces, que el módulo Sistema Matriz queda apto para su utilización y satisface todas las funcionalidades para las cuales fue concebido.

4.5. Conclusiones.

Al concluir este capítulo se obtiene la aplicación realizada atendiendo a los requisitos funcionales y no funcionales. Se realizaron los diagramas de componentes para cada caso de uso y se realizaron las pruebas al sistema obteniendo resultados satisfactorios, lo que demuestra que la aplicación cuenta con las características y funcionalidades para las que fue concebida.

Conclusiones generales

Al finalizar la investigación se obtuvo una serie de resultados que dan cumplimiento al objetivo propuesto, como son:

- Una aplicación web (Sistema Matriz) que favorece el proceso de gestión de información en el departamento Fuerza de Trabajo Calificada del MEP, logrando mayor agilidad en el proceso de elaboración de los planes de ingreso y distribución de graduados, asegurando mayor calidad en el mismo y el incremento de la seguridad del sistema.
- El objetivo trazado en el diseño teórico – metodológico para el desarrollo de la investigación fue cumplido, comprobándose la hipótesis como respuesta del problema a resolver que originó esta investigación, contribuyendo directamente a la gestión de información en el departamento Fuerza de Trabajo Calificada.
- Las tareas de la investigación facilitaron una adecuada organización de la misma contribuyendo al cumplimiento de los objetivos propuestos de acuerdo al cronograma planificado.
- La investigación quedó documentada así como todos los artefactos generados por la metodología de desarrollo de software utilizada.
- La aplicación web fue avalada por el cliente.

Recomendaciones

Una vez concluida la implementación del módulo Sistema Matriz de GeForza, se recomienda:

- Continuar el desarrollo del sistema de manera que pueda ser un sistema generalizado que integre a todas las aplicaciones, utilizadas por los diferentes departamentos, del Ministerio de Economía y Planificación.
- El Ministerio de Economía y Planificación debe solicitar al Ministerio del Interior el permiso para encriptar los archivos XML que se exportan, para asegurar así, un mayor nivel de seguridad de la información.
- La utilización del presente trabajo como futuro material de consulta para la preparación de nuevas generaciones de estudiantes, sugiriendo también la bibliografía utilizada.

Bibliografía

- Aplicación web. 2010]. Disponible en: http://www.albasoft.com/docu/prd/app_intro.html
- CALIFICADA, D. F. D. T. Instrucciones Metodológicas Actualización 12 Febrero 2008. , 2008.
- CAÑAVATE, A. M. Sistemas de información en las empresas: 251.
- CELIS, I. ESTADOBETA desarrollo web con estándares, 2005. [2010]. Disponible en: <http://www.estadobeta.com/2007/10/28/ext-20>
- ciberaula.com. 2010]. Disponible en: http://linux.ciberaula.com/articulo/linux_apache_intro/
- Desoft. Disponible en: <http://www.desoft.cu/Productos1/FastosPagus/tabid/442/Default.aspx>
- Diseño UML., 2009. [2010]. Disponible en: <http://antwan03.blogspot.com/2009/06/metodologia-rup-y-metodologia-uml.html>
- Free Download Manager. Disponible en: http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/
- GEPROP. 2010]. Disponible en: <http://www.geprop.cu/>
- GSInnova. 2010]. Disponible en: <http://www.rational.com.ar/herramientas/rup.html>
- GUTIÉRREZ, J. J. ¿Qué es un framework web?, 2010]. Disponible en: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf
- I. JACOBSON, G. B., J. RUMBAUGH. El proceso unificado de desarrollo de software. Madrid, Pearson Educación. SA, 2000. p. 84-7829-036-2
- Infor. Disponible en: <http://latinamerica.infor.com/soluciones/hcm/>
- Joan Garnnet. 2007. [2010]. Disponible en: <http://www.joangarnet.com/blog/?p=415>
- LANZILLOTTA, A. Definición y significado de Cliente/Servidor, 2010]. Disponible en: <http://www.mastermagazine.info/termino/4294.php>
- LARMAN, C. UML y Patrones. Introducción al análisis y diseño orientado a objetos. México, Prenntice Hall, 1999. p.
- librosweb.es. 2010]. Disponible en: <http://librosweb.es/javascript/capitulo1.html>
- LosRecursosHumanos.com. 2010]. Disponible en: <http://www.losrecursoshumanos.com/contenidos/2968-novedosa-solucion-para-la-gestion-del-capital-humano.html>

- NetBeans. 2010. [2010]. Disponible en: http://netbeans.org/index_es.html
- PECOS, D. danielpecos.com, 2010. [2010]. Disponible en: http://danielpecos.com/docs/mysql_postgres/x15.html
- PHP: Hypertext Processor. 2010]. Disponible en: <http://php.net/index.php>
- PRESSMAN, R. S. Ingeniería de software. Un enfoque práctico., 2002. p.
- ProgramaciónWeb.net. 2010. [2010]. Disponible en: <http://www.programacionweb.net/articulos/articulo/?num=505>
- ROLANDO ALFREDO HERNÁNDEZ LEÓN, S. C. G. El paradigma cuantitativo de la investigación científica., 2002.
- SAP México. Disponible en: <http://www.sap.com/mexico/solutions/business-suite/erp/hcm/index.epx>
- techtástico. 2007. [2010]. Disponible en: <http://techtastico.com/post/zend-framework-una-introduccion/>
- VANESSA PÉREZ CORREIA, A. Q. Ventajas y desventajas del software libre., 2010]. Disponible en: <http://www.slideshare.net/ariquibri/ventajas-y-desventajas-del-software-libre>
- Zend Framework. 2010]. Disponible en: <http://framework.zend.com/>

Glosario de términos

CAP: Consejos de Administración Provincial.

DBAL: *Database Abstraction Layer* (Capa de abstracción de base de datos).

ERP: (*Enterprise Resource Planning*) es un término generalizado en el mundo del software bajo el que se engloban una gran variedad de paquetes software, generalmente multi-modulares, que ofrecen soluciones integradas diseñadas para dar soporte a múltiples procesos de negocio.

FTC: Fuerza de Trabajo Calificada.

GeForza: Sistema Unificado de Gestión de Fuerza de Trabajo Calificada.

GOF: *Gang of Four* es el nombre con el que se conoce comúnmente a los autores del libro *Design Patterns* (ISBN 0-201-63361-2), referencia en el campo del diseño orientado a objetos. La “Banda de los cuatro”, se compone de los siguientes autores: Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides.

GRASP: Acrónimo de *General Responsibility Assignment Software Patterns* (Patrones de Software para la asignación General de Responsabilidad).

HCM: *Human Capital Managment* (Administración de capital humano).

HTTP: *Hypertext Transfer Protocol* (Protocolo de Transferencia de Hipertexto) o HTTP es el protocolo usado en cada transacción de la *World Wide Web*.

IMAP: *Internet Message Access Protocol*, o su acrónimo IMAP, es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor.

Licencia BSD: es la licencia de software otorgada principalmente para los sistemas BSD (Distribución de Software Berkeley). Es una licencia de software libre permisiva como la licencia de OpenSSL o la MIT License. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

MEP: Ministerio de Economía y Planificación.

NNTP: *Network News Transfer Protocol* (NNTP) especifica un protocolo para la distribución, la

investigación, recuperación, y la publicación de artículos periodísticos con un flujo fiable basado en la transmisión de noticias en la comunidad ARPA-Internet.

OACE: Organismos de la Administración Central del Estado.

ORM: *Objet-Relational Mapping* (Mapeador relacional de objetos).

POP3: *Post Office Protocol* (POP3, *Protocolo de la oficina de correo*) es un protocolo de nivel de aplicación en el Modelo OSI. Se utiliza en clientes locales de correo para obtener los mensajes de correo electrónico almacenados en un servidor remoto.

SNMP: El Protocolo Simple de Administración de Red o SNMP es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red.