

Universidad de las Ciencias Informáticas

Facultad 1



“Diseño e Implementación del Sistema Plan de Ingreso 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:

Elizabeth Parra Vázquez

Luis Ernesto Acosta Ortíz


Tutores:

Ing. Anabel Parra Vázquez

Ing. Enmanuel Azaharez Reyes

Ciudad de la Habana, junio 2010

“Año 52 de la Revolución”



"Lo que caracteriza al hombre de ciencia no es la posesión del conocimiento o de verdades irrefutables, sino la búsqueda desinteresada e incesante de la verdad."

Karl Popper (filósofo austríaco).

Dedicatoria:

Elizabeth

A:

Mi papá, mi mamá y mi hermana.

Luis Ernesto

A:

Mi familia, mi nenita linda Day, mis amigos y al realizador de todos nuestros sueños: Fidel.

Agradecimientos

Elizabeth

A mis padres y mi hermana, este trabajo es la recompensa de sus años de esfuerzo y dedicación, gracias por creer en mí, los amo.

A la Revolución y en especial a Fidel, por hacer tantos sueños realidad.

A mis tíos: Máximo, Víctor, Cucú, Valdín, Ana, Adela y Magalis, por estar al tanto siempre que en mis estudios todo marchara bien.

A mis primos en especial a: Anamaris, Raúl Ernesto, Victorito, Yasmina, Alexander y Marisol, que siempre me han acompañado.

A mis amistades de la UCI en especial a: Karina, Mama, Liudmila, Lisbeth y Ruber, que siempre me ayudaron y me guiaron a lo largo de estos cinco años.

A mi grupo 1104, por ser los mejores del mundo y soportarme todos estos años.

A mis amistades de Bayamo en especial a: Sandrita, Noraidis, Mirlena, Ileana, Merlin, Yisel, Nancy, Ana María, José Carlos y Frank, por estar siempre al pendiente de todo lo que me pasaba aquí.

A las enfermeras locas del policlínico: Diana, Alina y Mailín, por estar siempre ahí cuando las necesitaba.

A mis vecinos Pedro Castro y su familia, Carmen Kjala, Ena y Orlando.

A Toty que fue como otra madre para mí.

A mis compañeros de proyecto.

A todos los profesores que he tenido en el transcurso de toda mi vida, que han confiado en mí.

A la gente que siempre estuvo muy cerca de mí, María, Pedro, Dayana, Yordan, Yaima, Yankiel, Yailín, Carlos y a la chiqui, por estar hasta cuando no los necesité ahí.

A todos los que de una forma u otra contribuyeron a mi formación como ingeniero informático. A todos muchas gracias.

Agradecimientos

Luis Ernesto

A mi mamá y mi papá, por todo su amor, su sacrificio y sobre todo por su confianza en mí, gracias a ustedes he llegado hasta aquí y ustedes son los artífices de este triunfo.

A mi novia linda Day, eres la luz de mis ojos, gracias por compartir cada segundo conmigo.

A mis hermanos Kiki y Yuri por su incondicionalidad, cariño y complicidad.

A mi abuela Puchún, por ser siempre el nieto de su corazón, por su fe inquebrantable en mí, por malcriarme, y a mi diablita Rocío.

A la Revolución cubana y a nuestro Comandante Fidel por darme la posibilidad de cumplir todos mis sueños.

A toda mi familia, mis tíos y tías, Mercedes y Salva, Juan, Víctor, Geño, Peco, Rafael, Mirta, Victoria, Sala, Toti, Nery, Cecilio, Cacha y, aunque ya no están conmigo, Margot, Fela, Rosa y Pirimpollo, por estar siempre pendientes de mí, por darme todo el calor de una familia. A todos mis primos también.

A toda la familia que he tenido desde que nací en ese edificio 13, Torres y Misnelis, Pipo y Galia, Andino y la china, Maceo y Maricel, Roberto e Idelaida, Idalmis y Mauris y a mis hermanos de la vida Yasmani, Yani, Raidel, Corojo, Mailén, los Maceitos, Dailé y Danielito.

A mis amigos de toda la vida, Adrián, Francisco, Abdel, David, Andy, Pilo, Luis Carlos, Rubén, Raciél, Marien, Magui, Daneidis por todos los momentos que hemos vivido juntos.

A todos los amigos que conocí aquí en la UCI y que perdurarán toda la vida, a mi grupo 4 por educarme y ayudarme a convertirme en una mejor persona.

A Andrés Ochoa, por ser incondicional, por ser mi familia aquí en la escuela, por ayudarme siempre sin poner nunca un pero.

A Geidis, por ser antes de profesora una amiga incondicional, por toda su ayuda, sus consejos, sus regaños.

A la familia de mi novia, que me ha acogido como un miembro más de los suyos, a mis suegros Belkis y Gollito, a Tito, Yuli, Marlon, a mi viejita Olga.

A todos mis maestros y profesores, por educarme e impulsarme siempre a buscar más, por hacerme la persona que soy hoy en día.

A esa gran familia que tuve la oportunidad de conocer en Venezuela, que me acogieron y cuidaron como su niño chiquito, a Mayker, Lisbeth (Chini), Ailén, Yuno, Iván, Yucel, Alberto, Arista, Mireya, Lina, estarán conmigo siempre.

A todas esas personas que han puesto un granito de arena en cada una de las etapas de mi vida y que han contribuido a que me gradúe hoy como ingeniero informático.

A todos, de corazón, gracias.

Declaración de autoría

Declaramos que somos los únicos autores del trabajo titulado:

Diseño e Implementación del Módulo Administración para el Sistema Unificado de Gestión de Fuerza de Trabajo Calificada.

Y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

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

Elizabeth Parra Vázquez

Luis Ernesto Acosta Ortiz

Ing. Anabel Parra Vázquez

Ing. Enmanuel Azaharez Reyes



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:

A partir de la estrategia de actualizar el Estudio Integral para la Formación y Empleo de la Fuerza de Trabajo Calificada, trazada en el año 2008 por el Ministerio de Economía y Planificación, en conjunto con los Organismos de la Administración Central del Estado y demás organismos implicados en estos procesos, surge en nuestra universidad el proyecto Sistema Unificado de Gestión de Fuerza de Trabajo Calificada (GeForza) y como parte de este, el Sistema Plan de Ingreso a la Educación Media y Superior. El desarrollo de este sistema es el tema de la presente investigación.

El departamento de Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación es el encargado de realizar los planes de ingreso a la Educación Media y Superior en el país. Para ello cuenta con una aplicación, la cual no supe sus necesidades y presenta un sinnúmero de problemas, por lo que se plantea el desarrollo de este nuevo sistema. El mismo persigue automatizar los procesos de gestión de los planes de ingreso logrando una mayor eficiencia en el desarrollo de estos procesos.

Para el desarrollo del sistema se utiliza RUP como metodología de desarrollo, UML como lenguaje de modelado, Visual Paradigm como herramienta CASE para el modelado, PHP como lenguaje de programación, PostgreSQL como gestor de base de datos, Apache como servidor web y se utilizan los marcos de trabajo ExtJS , Zend Framework y Doctrine.

La investigación fue presentada en la Jornada Científica Estudiantil a nivel de facultad obteniendo la categoría de Relevante.

Palabras clave: *educación media y superior, Geforza, plan de ingreso.*

Índice

Índice de Tablas	XII
Índice de Figuras	XIII
Introducción	1
Capítulo 1: Fundamentación teórica	6
Introducción.....	6
Necesidad del Sistema Plan de ingreso a la Educación Media y Superior	6
Los procesos de ingreso a la educación media y superior en Cuba	7
Estado del arte de sistemas informáticos relacionados con el campo de acción	8
Tecnologías, metodología y herramientas actuales.....	13
Metodología de desarrollo de software	14
Herramienta de Ingeniería de Software Asistido por Ordenadores (CASE) para el modelado.....	16
Lenguaje de modelado visual.....	17
Lenguaje de programación para el desarrollo web.....	17
Sistema gestor de base de datos.....	19
Axure	20
Marcos de trabajo: Zend Framework, ExtJS y Doctrine	21
Entorno de Desarrollo Integrado (IDE): NetBeans	23
Servidor web: Apache	23
Aplicación web	24
Modelo cliente-servidor	25
Patrón de arquitectura Modelo Vista Controlador (MVC)	26
Fundamentación de las herramientas, metodología y tecnologías a utilizar	27
Conclusiones.....	28
Capítulo 2: Características del sistema	29
Introducción.....	29
Sistema Unificado de Gestión de Fuerza de Trabajo Calificada (GeForza).....	29
Modelo de dominio	29
Definición de las clases del modelo de dominio	30
Especificación de los requerimientos de software	30
Requerimientos funcionales	31

Requerimientos no funcionales	39
Propuesta de sistema	41
Modelo del sistema	42
Actores y casos de uso del sistema.....	42
Diagrama de casos de uso del sistema.....	42
Descripción abreviada de los casos de uso del sistema.....	44
Conclusiones.....	48
Capítulo 3: Análisis y diseño del sistema.....	49
Introducción.....	49
Estudio de factibilidad	49
Beneficios tangibles e intangibles.....	54
Análisis de costos y beneficios	54
Análisis	55
Diagrama de clases del análisis.....	55
Diagramas de colaboración del análisis.....	56
Diseño	57
Propósitos del diseño	58
Arquitectura y patrones de diseño utilizados.....	58
Modelo de diseño	61
Realización de los diagramas de clases del diseño	62
Modelo de datos	64
Diseño de la base de datos	65
Tratamiento de errores.....	66
Seguridad	66
Conclusiones.....	66
Capítulo 4: Implementación y prueba	67
Introducción.....	67
Implementación	67
Estándares de implementación.....	67
Diagrama de componentes.....	68
Diagrama de despliegue	69

Prueba.....	70
Métodos de prueba.....	70
Pruebas del sistema.....	71
Resultados de las pruebas	78
Conclusiones.....	79
Conclusiones.....	80
Recomendaciones.....	81
Bibliografía referenciada.....	82
Bibliografía consultada.....	83
Glosario de términos.....	85

Índice de Tablas

Tabla 1. Operacionalización de las variables	4
Tabla 2. Diferencias entre metodologías ágiles y tradicionales	14
Tabla 3 Descripción textual de los actores del Sistema	42
Tabla 4 Descripción breve del CU Gestionar disponibilidad de graduados de noveno grado	45
Tabla 5 Descripción breve del CU Gestionar disponibilidad de graduados de enseñanza media	45
Tabla 6 Descripción breve del CU Gestionar capacidad docente de preuniversitarios provinciales	45
Tabla 7 Descripción breve del CU Gestionar capacidad docente disponible a centros universitarios	46
Tabla 8 Descripción breve del CU Gestionar demanda de técnicos medio y obreros calificados	46
Tabla 9 Descripción breve del CU Gestionar demanda de graduados de la enseñanza superior	46
Tabla 10 Descripción breve del CU Gestionar plan de ingreso de técnicos medio y obreros calificados	47
Tabla 11 Descripción breve del CU Gestionar plan de ingreso de la enseñanza superior	47
Tabla 12 Descripción breve del CU Mostrar fuentes de ingreso sobrantes	47
Tabla 13 Descripción breve del CU Mostrar carreras por provincia	48
Tabla 14 Factor de peso de los actores sin ajustar	50
Tabla 15 Factor de peso de los casos de uso sin ajustar	50
Tabla 16 Factor de complejidad técnica	51
Tabla 17 Factor de ambiente	52
Tabla 18 Esfuerzo del proyecto	53
Tabla 19 Caso de prueba del CU Gestionar Disponibilidad de Graduados de la Enseñanza Media	78

Índice de Figuras

Figura 1 Modelo Cliente-Servidor	25
Figura 2 Patrón de Arquitectura MVC	26
Figura 3 Diagrama de Clases del Modelo de Dominio	30
Figura 4 Diagrama de Paquetes.....	42
Figura 5 Diagrama de casos de uso del sistema Paquete Entrada de datos	43
Figura 6 Diagrama de casos de uso del sistema Paquete Salida de Datos	44
Figura 7 DCA: Gestionar Plan de Ingreso de la Enseñanza Superior.	56
Figura 8 DC: Gestionar Plan de Ingreso de la Enseñanza Superior (Adicionar)	56
Figura 9 DC: Gestionar Plan de Ingreso de la Enseñanza Superior (Modificar).....	57
Figura 10 DC: Gestionar Plan de Ingreso de la Enseñanza Superior (Eliminar)	57
Figura 11 DCD: Gestionar Plan de Ingreso de la Enseñanza Superior	63
Figura 12 Modelo Físico de la Base de Datos	64
Figura 13 Modelo Lógico de Datos.....	65
Figura 14 Diagrama de Componentes CU Gestionar Plan de Ingreso a la Educación Superior	69
Figura 15 Diagrama de despliegue del sistema.....	70

Introducción

En la actualidad, el uso de las Tecnologías de la Informática y las Comunicaciones (TIC) en todos los procesos económicos y sociales es cada vez más imprescindible. El desarrollo de las mismas permite que los procesos de la humanidad se realicen con mayor calidad, control y eficiencia. Estas tecnologías influyen de manera directa en los procesos económicos y empresariales, contribuyendo a mejorar la calidad de los productos y servicios, tributando al perfeccionamiento empresarial y aumentando la competitividad de las empresas. El mundo ha arribado a la era de la gestión de la información, de la automatización de los procesos, de la informatización de la sociedad.

La Universidad de las Ciencias Informáticas (UCI), se ha convertido en el principal pilar de la industria cubana del software y es protagonista de todo el proceso de informatización social y empresarial que se desarrolla en Cuba. Su misión es convertirse en una importante fuente de ingresos a la economía cubana.

Como parte de la Batalla de Ideas, la educación superior y la formación técnica y profesional en Cuba se ven inmersas en un proceso vertiginoso de cambio radical y de expansión, surgiendo no solo un sinnúmero de carreras novedosas, sino abriéndose también sedes en las provincias para la mayoría de las carreras, que anteriormente se veían centralizadas en una provincia determinada para cada región del país, lográndose así un incremento acelerado de nuevas fuentes de ingresos a la educación superior.

El departamento de Fuerza de Trabajo Calificada (FTC) del Ministerio de Economía y Planificación (MEP) es el encargado de realizar la gestión de los planes de ingreso a la educación media y superior, como parte de todo un proceso de formación y empleo de la FTC, del cual depende, en gran medida, el plan económico del país, por la estrecha relación que existe entre el plan de generación de puestos de trabajo para un año y la cantidad de graduados de esas especialidades que se deben incorporar como fuerza laboral ese año.

Actualmente, el MEP cuenta con una aplicación, con la que desde hace varios años se gestiona la información referente al plan de ingreso a la educación superior, careciendo de una para el plan de ingreso a la educación media. Ante todos los cambios sucedidos en la educación, esta aplicación en uso se muestra ineficiente y prácticamente obsoleta. Por otra parte, la gestión de la información necesaria para confeccionar el plan de ingreso es llevada en sistemas separados, cuya comunicación e intercambio de datos se hace a través de ficheros, los cuales presentan un nivel de seguridad muy bajo que no garantiza la integridad de la información, aspecto de vital importancia para un sistema de esta magnitud. Otro de los inconvenientes de la aplicación es la interfaz rígida que posee y que dificulta la facilidad de uso de la misma, el sistema de reportes que presenta no es lo suficientemente

genérico y no es configurable por el cliente, el tratamiento de excepciones y la recuperación ante los errores son bastante primarios y rígidos en la misma.

Para la elaboración de la propuesta de plan de ingreso se diseñó un algoritmo que contempla la demanda de graduados de cada carrera de los organismos para dentro de cinco y seis años en dependencia de la duración de los estudios, a partir de esta demanda y en dependencia de la disponibilidad de graduados de nivel medio del territorio se le va dando satisfacción a la misma. En este algoritmo no se tiene en cuenta la capacidad de las instituciones donde se estudiarán estas carreras, por lo que no llega hasta el nivel de asignar una matrícula de una carrera para cada provincia en su respectivo centro de estudio, si el mismo tiene capacidad, dejando todo este proceso de asignación de centros de estudios al funcionario del departamento encargado del plan de ingreso.

Como consecuencia de esta situación problemática, la información del plan de ingreso demora mucho en ser gestionada, por lo que las planillas y reportes a entregar se atrasan y dificultan la toma de decisiones en el momento preciso, el algoritmo que genera la propuesta de plan de ingreso no lo hace con la eficiencia necesaria, ralentizando aún más el proceso de elaboración y aprobación del plan y, debido a la descentralización de la información, la persistencia e integridad de la misma no es garantizada.

Teniendo en cuenta lo anteriormente expuesto el **problema científico** queda formulado de la siguiente forma ¿Cómo mejorar los procesos de gestión de los planes de ingreso a la Educación Media y Superior en el departamento de Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación?

Como **objetivo general** de la investigación se define automatizar la gestión del plan de ingreso a la educación media y superior, permitiendo agilizar la gestión de la información del ingreso por el Ministerio de Economía y Planificación del cual se derivan como **objetivos específicos** los siguientes:

- ✓ Analizar los procesos relacionados con la gestión del plan de ingreso a la educación media y superior.
- ✓ Desarrollar el análisis y diseño de los procesos de gestión de plan de ingreso a la educación media y superior.
- ✓ Implementar el Sistema Plan de Ingreso a la Educación Media y Superior.

El **objeto de estudio** de la investigación es conformado por los procesos desarrollados en el departamento de la Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación del cual se extraería como **campo de acción** los procesos de gestión del plan de ingreso a la educación media y superior del departamento de la Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación.

Para dar cumplimiento a los **objetivos** se definen las siguientes **tareas investigativas**:

- ✓ Estudio del estado del arte relacionado con la gestión de plan de ingreso.
- ✓ Análisis de los procesos de negocio que se desarrollan en la gestión del plan de ingreso a la educación media y superior.
- ✓ Caracterización de las tecnologías y tendencias actuales a aplicar en el desarrollo de la aplicación.
- ✓ Implementación del Sistema Plan de Ingreso a la Educación Media y Superior haciendo uso de herramientas libres.

Para guiar la investigación se plantea la siguiente **hipótesis**: Con la implementación del Sistema Plan de Ingreso a la Educación Media y Superior ajustado a los nuevos clasificadores y nomencladores, se mejorará el proceso de gestión de información, en cuanto a factibilidad, eficiencia y satisfacción del cliente.

Variable independiente: sistema plan de ingreso a la educación media y superior.

Variable dependiente: procesos de gestión de información.

VARIABLES	DIMENSIONES	INDICADORES	UNIDAD DE MEDIDA
Sistema plan de ingreso a la educación media y superior	Factibilidad	Tiempo de desarrollo	Extenso Moderado Breve
		Costo	Costoso Moderado Barato
		Esfuerzo	Alto Moderado Despreciable
Proceso de gestión de la información	Eficiencia	Rapidez	Alta Media Baja

		Seguridad	Alta Media Baja
		Organización del trabajo	Buena Mala
	Satisfacción del cliente	Satisfacción con el sistema	Alta Media Baja
		Agilidad en el proceso	Rápido Medio Lento

Tabla 1. Operacionalización de las variables

Para llevar a cabo las tareas investigativas se emplearon diversos métodos científicos tanto teóricos como empíricos. Los **métodos teóricos** que se utilizan son:

- ✓ **Histórico Lógico:** Se aplica al realizar el estudio a nivel nacional e internacional del desarrollo de sistemas similares al que se quiere implementar, así como de la evolución de los procesos de gestión para el plan de ingreso.
- ✓ **Analítico-Sintético:** Este método se emplea para organizar y sintetizar toda la información obtenida del estudio del estado del arte, las tecnologías, metodología, lenguajes y herramientas propuestas para la solución e identificar los principales problemas del sistema actual. La utilización de este método permite comprender mejor toda la información anterior, la cual será de gran utilidad para lograr un adecuado análisis y diseño del sistema.
- ✓ **Modelación:** Es usado para realizar los diagramas necesarios en el desarrollo del sistema que mejorarán el entendimiento de los procesos de gestión de plan de ingreso a la educación media y superior.

Y como **métodos empíricos** se utilizan los siguientes:

- ✓ **Entrevista:** Se entrevista al líder del proyecto así como algunos de los clientes del sistema que actualmente se explota, donde se obtuvo la mayor cantidad de información sobre los problemas que presenta dicho sistema, las posibles soluciones a estos problemas, los procesos que realizará el sistema plan de ingreso a la educación media y superior y todo lo nuevo que se le

pretende incorporar. Este método tiene la ventaja de aportar conocimientos del tema que no se poseen.

- ✓ **Método de observación:** Este método sirve de apoyo para obtener información sobre el sistema actual, que se usa para gestionar el plan de ingreso y ayuda a dar solución a prácticamente todas las tareas. Haciendo uso de él, se puede ver cómo funciona actualmente la gestión del plan de ingreso, así como el sistema que está en explotación, las ventajas y desventajas de las tecnologías, metodología, lenguajes y herramientas a utilizar en el desarrollo del nuevo sistema. Además permite apreciar qué es lo que realmente necesitan los clientes del sistema de acuerdo a las nuevas concepciones.

El contenido del presente trabajo está distribuido en 4 capítulos:

En el **Capítulo 1**, fundamentación teórica, se realiza un estudio del estado del arte relacionado con los planes de ingreso de la educación media y superior, además se exponen los conceptos fundamentales relacionados con el tema de investigación y se describen los lenguajes, las herramientas y metodología a utilizar para el desarrollo de la aplicación.

En el **Capítulo 2**, características del sistema, se realiza la descripción del sistema a implementar, para esto se describe el problema y la situación problemática a resolver, el objeto de automatización, se representan los procesos del negocio, se especifican los requerimientos funcionales y no funcionales, y se define el modelo de casos de uso del sistema.

En el **Capítulo 3**, análisis y diseño del sistema, se realiza todo lo que comprende dicho flujo de trabajo para el sistema a desarrollar, dentro de este se exponen algunos de los diagramas del diseño.

En el **Capítulo 4**, implementación y prueba, se desarrolla el flujo de trabajo Implementación, empleando para ello la metodología RUP, registrándose los diagramas relacionados con el mismo y además se le realizan las pruebas al producto.

Se espera como **posible resultado** de la investigación, una aplicación web para el Ingreso a la Educación Media y Superior del Ministerio de Economía y Planificación.

Capítulo 1: Fundamentación teórica

Introducción

En el presente capítulo se realiza un estudio de los procesos desarrollados en el departamento de Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación (MEP) asociados a la gestión de los planes de ingreso a la educación media y superior, además de un estado del arte de los sistemas informáticos similares al que se desarrollará y se caracterizan las tecnologías y metodologías que se utilizan durante el desarrollo del sistema.

Necesidad del Sistema Plan de ingreso a la Educación Media y Superior

El departamento de Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación (MEP) es el encargado de desarrollar los procesos de formación y empleo de la fuerza de trabajo calificada en el país. Dentro de estos procesos se encuentran la elaboración de los planes de ingreso a la educación media y superior, así como la distribución de los graduados en los organismos. Debido a la crisis económica mundial, de la cual Cuba no escapa, y a la importancia vital de realizar una planificación eficiente de los recursos con los que se cuenta, se hace necesaria la elaboración de estos planes de ingreso a partir de la demanda de graduados por territorio, la disponibilidad de graduados del nivel inmediato inferior con los que cuenta el territorio y la capacidad de los centros educacionales en los que matricularán los estudiantes. Otro de los factores a los que deben responder estos planes es a la línea de desarrollo económico trazada por el país. El departamento cuenta actualmente con un software para la gestión del plan de ingreso a la educación superior pero carece de uno para la educación media o continuidad de estudios de los estudiantes de noveno grado hacia la Educación Técnica y Profesional y Preuniversitaria. Además de esto, con los cambios que se han sucedido en los últimos años en la educación, como resultado de la Batalla de Ideas, viéndose incrementado vertiginosamente el número de carreras universitarias y de los centros de estudios superiores, el software con el que se cuenta ha quedado prácticamente obsoleto. Otro de los problemas actuales que presenta el departamento es la descentralización de la información que gestiona, pues existen varios sistemas que gestionan la información de forma separada y cuya comunicación con los demás sistemas es precaria e insegura. Por todo lo antes planteado, se hace necesario el desarrollo de un sistema que gestione de forma unificada toda la información con la que trabaja el departamento, y que sea capaz de gestionar eficientemente los planes de ingreso a la educación media y superior. Ante esta problemática se plantea el estudio de los procesos actuales ligados a la gestión de los planes de ingreso, con el fin de desarrollar una aplicación que automatice estos procesos de manera eficiente,

que sea además lo suficientemente genérica y robusta para adaptarse a posibles cambios venideros y tener así una larga vida de explotación.

Los procesos de ingreso a la educación media y superior en Cuba

En Cuba al culminar el noveno grado el estudiante tiene la opción de continuar estudios en la educación preuniversitaria o en la educación técnica y profesional. La educación es absolutamente gratuita y generalizada. Las carreras o especialidades son ofertadas y para acceder a ellas se realizan exámenes de ingreso o pruebas de aptitud, en algunos casos el ingreso solo tiene en cuenta la promoción del estudiante. El sistema universitario es completamente planificado desde el ingreso del estudiante a las instituciones de educación superior. En primer lugar, por la necesidad de que la fuerza de trabajo calificada responda a las necesidades socioeconómicas del país y a la situación real del empleo y, en segundo lugar, de particular importancia para un país en vías de desarrollo como es Cuba, los recursos materiales y financieros son factores limitantes del desarrollo educacional y deben utilizarse con la mayor eficiencia posible. Esto influye sobre los proyectos de desarrollo de cada institución de estudios en cuanto a las carreras que ofrece y también, en la creación de nuevos centros. Estos dos elementos determinan un carácter centralizado sobre:

- a) Matrícula total y por carreras.
- b) Estructura de los centros de educación superior.
- c) Estructura de las carreras en cada centro de estudios.

La centralización de estos aspectos contienen dos ideas que resultan importantes para su total comprensión: un sistema de acceso a la universidad basado en el mérito académico, y un plan de becas que facilita el ingreso a cualquier carrera, en cualquier institución del país, a partir de la estimación de las necesidades territoriales de fuerza de trabajo calificada.

Además de las necesidades a las que debe responder la elaboración del plan de ingreso, la misma cuenta con procesos diferentes atendiendo la fuente de ingreso que está siendo tratada:

- ✓ La fuente de ingreso a la enseñanza técnica y profesional, que es brindada en los centros politécnicos del país a los estudiantes que al terminar estudios de noveno grado, deciden optar por este tipo de carreras. Para el acceso a estos centros se toma en consideración la promoción del estudiante, y en algunos casos, pruebas de aptitud. Para el plan de ingreso a esta enseñanza se debe tener en cuenta la disponibilidad de estudiantes graduados de noveno grado, la demanda de obreros calificados por territorio para dentro de dos años y de técnicos medios para dentro de cuatro años, y la capacidad de los centros en los que se estudian estas carreras.
- ✓ La fuente de ingreso al curso regular diurno, que brindan las instituciones universitarias del país a los estudiantes que, como requisito, deben tener el bachiller vencido y aprobar los exámenes de

ingreso a la institución requerida o, en algunos casos, pruebas de aptitud. Para el plan de ingreso a este curso regular diurno se debe tener en cuenta la disponibilidad de estudiantes graduados de preuniversitario que continuará estudios, la demanda de empleos por territorios para dentro de 5 años y en el caso de las carreras de Ciencias Médicas y Lenguas Extranjeras para dentro de 6 años, y la capacidad de los centros universitarios para satisfacer la demanda de matrícula.

- ✓ La fuente de ingreso a los cursos para trabajadores que, con dedicación a tiempo parcial, se ofertan a trabajadores que mantienen su vínculo laboral mientras estudian. Pueden ser vespertino-nocturnos, por encuentros o también en cursos a distancia, donde el estudiante recibe apoyo mediante guías de estudio, textos y páginas web, además de consultas presenciales con profesores. No existen restricciones para el acceso. En la actualidad esta modalidad se amplía con la oferta de nuevos productos multimedia y en los cursos de continuación de estudios que se desarrollan en las sedes universitarias municipales. La modalidad de estudio es semipresencial y el modelo pedagógico se caracteriza por su flexibilidad para adaptarse a diversas situaciones laborales, a particularidades territoriales y al ritmo individual de aprovechamiento académico del estudiante.

Cada fuente de ingreso aquí detallada tiene sus particularidades por lo que los procesos de ingreso para cada una son diferentes y la información para cada una se gestiona de manera diferente. Estos procesos son administrados y desarrollados por el Ministerio de Economía y Planificación, en específico por el departamento de Fuerza de Trabajo Calificada.

Estado del arte de sistemas informáticos relacionados con el campo de acción

Luego de consultar varias bibliografías electrónicas y de efectuar varias búsquedas en Google se concluye que, por las características del sistema que se desarrollará, este puede ser clasificado como un sistema de planificación de recursos empresariales, pues el mismo será utilizado no directamente para el control de los recursos materiales del Ministerio de Economía y Planificación, pero si gestionará la información de los recursos humanos del país que se maneja en este ministerio, así como realizará la planificación del ingreso de estos recursos humanos a la educación media y superior.

Los sistemas de Planificación de Recursos Empresariales (*Enterprise Resource Planning*, ERP) son sistemas de gestión de información gerenciales que integran, automatizan y manejan muchas de las acciones asociadas con las operaciones de producción y distribución de una compañía dedicada a la producción de bienes o servicios (1).

Los ERP pueden intervenir en el control de muchas actividades de negocios, tales como ventas, compras, entregas, pagos, producción, contabilidad, administración de inventarios y de recursos

humanos. Funcionan en todo tipo de empresas. Para ello, integran todos los departamentos funcionales de la empresa, herramientas de mercadotecnia y administración estratégica en un solo sistema.

Un ERP se distingue de otro software empresarial porque es un sistema integral, con modularidad y adaptable.

Características de los sistemas ERP:

- ✓ Integrales: Todos los departamentos o áreas de una empresa se relacionan entre sí, permitiendo de esta forma controlar los distintos procesos de la entidad, donde la salida de un proceso puede ser la entrada de otro.
- ✓ Modulares: Se dividen en módulos, que son las áreas que se integran como un todo para el manejo óptimo de la información. Estos módulos se instalan según las necesidades del cliente.
- ✓ Adaptables: Son diseñados para ser configurables según lo que cada empresa necesite.
- ✓ Base de datos centralizada.
- ✓ Sus componentes interactúan entre sí consolidando todas las operaciones.
- ✓ En un sistema ERP los datos se ingresan sólo una vez y deben ser consistentes, completos y comunes.
- ✓ Las empresas que lo implantan suelen tener que modificar alguno de sus procesos para alinearlos con los del sistema ERP. Este proceso se conoce como Reingeniería de Procesos, aunque no siempre es necesario (1).

Sistemas estudiados en el mundo:

➤ **Sistema Integrado de Planificación Estadística (SINPLE)**

El Sistema Integrado de Planificación Estadística es una aplicación creada en el estado autónomo de Andalucía para el desarrollo de los planes estadísticos, así como la aprobación y seguimiento de los programas estadísticos. Su objetivo es mejorar la coordinación del sistema estadístico de Andalucía, aumentando su eficiencia mediante el avance en la mejora de la coordinación interna entre el Instituto de estadística de Andalucía como órgano coordinador del sistema y los distintos órganos del mismo.

Es un sistema basado en Flujos de Trabajo (*Workflow*), que permite agilizar los trámites de aprobación y seguimiento de los programas estadísticos; de hecho, la aplicación contempla el proceso completo de declaración de una actividad estadística a incluir en un programa estadístico, con todos los actores implicados, y permite añadir detalles del seguimiento de su ejecución. Además la utilización del SINPLE no queda acotada al período de vigencia de un plan estadístico concreto, sino que permite la

incorporación de un nuevo plan con sus diferentes estructuras y poder así implantar este sistema durante un período largo de tiempo.

Es un software orientado a procesos, usa Oracle como gestor de base de datos y está dividido en cuatro subsistemas que gestionan todo el flujo de procesos de los planes y programas estadísticos:

- ✓ Subsistema de estructura del plan estadístico.
- ✓ Subsistema de elaboración del programa/plan estadístico.
- ✓ Subsistema de seguimiento del programa/plan estadístico.
- ✓ Subsistema de evaluación de planes y programas.

Entre las utilidades del sistema se pueden resaltar las siguientes:

- ✓ La interconexión y coordinación entre el Instituto de Estadística de Andalucía y las Unidades Estadísticas de las Consejerías u Organismos Autónomos pertenecientes al Sistema Estadístico de Andalucía.
- ✓ La definición de nuevas actividades estadísticas y la elaboración de los planes y programas estadísticos anuales.
- ✓ El seguimiento de la ejecución de toda actividad estadística programada o planificada por parte de todo el Sistema Estadístico.
- ✓ La normalización de la información contenida en el Inventario de actividades del Sistema Estadístico de Andalucía (2).

ADAM5, Sistema integral de Nómina y Recursos Humanos

ADAM5 es un sistema integral para la administración de los recursos humanos dentro de las empresas e instituciones. Es un sistema modular cuya cobertura y alcance engloba las diferentes disciplinas de la administración de los recursos humanos en las organizaciones, sin importar el tipo de industria ni legislación, ya que su alto grado de parametrización le permite adaptarse a diferentes prácticas en la administración del capital humano, así como para el control y administración del pago de las nóminas y cumplimiento de todas las obligaciones fiscales y legales con las diferentes dependencias gubernamentales(3).

Fue desarrollado en Visual Basic, Visual .NET y Java y puede ser implantado tanto en Oracle como SQL Server como gestores de bases de datos, es multiplataforma y el tipo de licencia del producto varía en dependencia del número de empleados de la empresa.

ADAM es la solución vanguardista con la base instalada más importante en México, Centro América y Sudamérica, con más de 1000 clientes, que entre ellos se encuentran algunas de las empresas más importantes del mundo. Está conformado por cinco familias de aplicaciones totalmente modulares, con el objetivo de que las empresas puedan configurar la solución ideal a sus necesidades (3).

Sistemas estudiados en Cuba:

✓ **FTCINGRE**

Es el sistema de gestión de información del plan de ingreso a la educación superior que se encuentra en explotación actualmente, por el departamento de Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación. Fue desarrollado en el año 1998, usando Visual FoxPro 5.0. No se cuenta con el código fuente del sistema por lo que se hace imposible la reutilización del código en función de realizarle mejoras. Se estudió detalladamente en busca de definir las funcionalidades principales con las que debe contar el sistema a desarrollar, así como definir las funcionalidades que deben ser agregadas y las que deben ser mejoradas. FTCINGRE cuenta con los módulos de Reportes y de gestión de ingreso al curso regular diurno y al curso para trabajadores. Se encuentra prácticamente obsoleto ante toda esta problemática surgida con las novedades en la educación superior y en consecuencia en el plan de ingreso a la misma. La seguridad del sistema es muy baja pues no tiene un sistema de autenticación de usuario robusto, y cualquier usuario, con acceso al computador donde se encuentra el software, puede acceder al mismo y realizar cualquier operación. El intercambio de datos entre este y los demás sistemas que gestionan la información necesaria para elaborar el plan de ingreso se hace mediante ficheros, de forma muy insegura, por lo que no se garantiza la integridad de la información, aspecto de vital importancia por la información gestionada. La interfaz del sistema es rígida y poco amigable lo que dificulta y ralentiza la utilización del mismo. El sistema de reportes que ofrece no es lo suficientemente genérico y no supe las necesidades del cliente. El tratamiento de excepciones y la recuperación ante los errores son bastante primarios y rígidos en el software.

Sistemas estudiados en la UCI

✓ **CEDRUX: Sistema Integral de Gestión de Entidades**

Es una solución de software creada en la Universidad de las Ciencias Informáticas (UCI) para la completa gestión empresarial en cualquier organismo o entidad del país. La solución en sí es un potente marco de trabajo que permite el desarrollo de cualquier tipo de sistema de gestión de entidades. Se estudió con la finalidad de encontrar una arquitectura candidata para el sistema a desarrollar. CEDRUX ya trae integrado un sistema muy robusto y eficiente de gestión de la seguridad basado en roles, a cada usuario del sistema le es asignado un rol, y a cada rol se le asignan permisos

sobre las funcionalidades del sistema así como se le definen las acciones que puede desarrollar en cada funcionalidad. Además, permite el desarrollo de varios sistemas dentro del mismo marco de trabajo y la interacción y comunicación entre los mismos de forma segura, mediante servicios web. Cuenta con varios módulos integrados, entre ellos el de Traza, que permite al administrador del sistema visualizar informes precisos del comportamiento y utilización de la aplicación en un tiempo determinado. Propone una arquitectura Modelo Vista Controlador. Se utilizó en su desarrollo como lenguaje de programación PHP5, y se utilizaron los marcos de trabajo ExtJS 2.2 para la capa presentación, Zend Framework 1.5 para la capa el negocio y Doctrine 1.1 para la capa de acceso a datos. Es una aplicación multiplataforma.

Luego de realizar diferentes búsquedas, en los idiomas español e inglés, atendiendo diferentes criterios de búsqueda y haciendo diversas combinaciones de las palabras claves: ingreso, planificación, asignación, algoritmo y distribución, la búsqueda no proporcionó la existencia de un software, fuera de Cuba, que sea usado específicamente para la planificación del ingreso de todos los estudiantes de un país a la educación media y superior. Teniendo en cuenta que esta búsqueda no arrojó resultados positivos, y que el modelo social existente en Cuba es auténtico y único, pues no existe otro país en el mundo, capitalista o socialista, en el que la educación sea totalmente gratuita, en la que el gobierno de ese país base todo su desarrollo económico en función del bienestar de su pueblo y, específicamente, en el que se vincule estratégicamente el ingreso de los estudiantes a los centros de enseñanza media y superior, en función de las necesidades económicas del país y las necesidades de empleo en cada territorio, debido a que todos los recursos en un país bloqueado como Cuba deben ser planificados con la mayor eficiencia posible para lograr un desarrollo económico favorable, se realizó el estudio de otros sistemas que, aunque no fueran específicamente lo que se buscaba, se asemejaran al sistema a desarrollar, con el fin de encontrar características en las cuales apoyarse para el desarrollo del mismo.

Del estudio realizado se concluye que: Los sistemas de planificación de recursos empresariales son la elección más acertada a la hora de automatizar la gestión de información y el control de los recursos de una empresa, las funcionalidades estudiadas de FTCINGRE servirán para ser optimizadas en el nuevo sistema y se tomarán como punto de partida a la hora de desarrollar las funcionalidades para la gestión del plan de ingreso a la educación media. Se concluye también que el uso de la aplicación Cedrux como base del sistema, por todas las ventajas y facilidades estudiadas, por ser un software desarrollado en la UCI, por su desarrollo en herramientas libres, además conocidas por los desarrolladores, y por la arquitectura robusta que ofrece, no solo mejorará la eficiencia y seguridad del

producto final sino que hará de este un producto multiplataforma, de fácil configuración, extensible y con una interfaz amigable al usuario.

Tecnologías, metodología y herramientas actuales

El éxito en el desarrollo de un software depende de un sinnúmero de actividades y etapas, donde el impacto de elegir la mejor metodología para un equipo de un determinado proyecto, es trascendental para el éxito del producto. El papel decisivo de las metodologías es, sin duda, esencial en un proyecto y el paso inicial que debe encajar en el equipo, guiar y organizar actividades que conlleven a las metas trazadas en el grupo.

De acuerdo con el análisis de algunos conceptos se pudo concluir que las metodologías de desarrollo son el conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a elaborar un nuevo producto. Imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo crean ejecutando un proceso desarrollado con un fuerte énfasis en planificar, inspirado por otras disciplinas de la Ingeniería de software. Tienen el propósito de garantizar eficacia a la hora de cumplir los requisitos y de disminuir la pérdida de tiempo empleada durante el desarrollo del mismo.

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a cambios.
Impuestas internamente (por el equipo).	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.

Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles.
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.

Tabla 2. Diferencias entre metodologías ágiles y tradicionales

Metodología de desarrollo de software

El Proceso Unificado de Desarrollo de Software (RUP) es uno de los procesos más generales existentes en la actualidad. RUP es una metodología para la Ingeniería de software que da como resultado procesos basados en componentes, incluye artefactos (que son los productos tangibles del proceso como por ejemplo el modelo de casos de uso y el código fuente.) y roles (papel que desempeña una persona en un determinado momento). Los proyectos realizados utilizando RUP cuentan con cuatro fases: inicio, elaboración, construcción y transición.

Inicio: Se describe el negocio y se delimita el proyecto, describiendo sus alcances con la identificación de los casos de uso del sistema.

Elaboración: Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.

Construcción: Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene una o varias versiones del producto que han pasado las pruebas. Se ponen estas versiones a consideración de un subconjunto de usuarios.

Transición: La versión ya está lista para su instalación en las condiciones reales. Puede implicar reparación de errores.

El ciclo de vida de RUP es desarrollado por nueve flujos de trabajo: modelo de negocio, requerimientos, análisis y diseño, implementación, prueba, despliegue, administración de configuración y cambios, administración de proyecto y ambiente.

Modelo de negocio: Describe los procesos del negocio, identificando quiénes participan y las actividades que requieren automatización.

Requerimientos: Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

Análisis y diseño: Describe cómo el sistema será realizado a partir de las funcionalidades previstas y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.

Implementación: Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.

Prueba: Busca los defectos a lo largo del ciclo de vida.

Despliegue: Produce las versiones del producto y realiza actividades (empaquete, instalación, asistencia de usuario, entre otras.) para entregar el software a los usuarios finales.

Administración de configuración y cambios: Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, entre otras.

Administración de proyecto: Involucra actividades con las que busca elaborar un producto que satisfaga las necesidades del cliente.

Ambiente: Dirige sus actividades a la configuración necesaria para el proceso de desarrollo de software. El propósito de esta disciplina es proveer las herramientas necesarias y un ambiente técnico adecuado para el proceso de desarrollo.

Las características fundamentales de RUP son:

Dirigido por casos de uso: En RUP los casos de uso modelan los requisitos funcionales del sistema. También guían su diseño, implementación y prueba. Los casos de uso constituyen un elemento integrador y una guía del trabajo, no solo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

Centrado en la arquitectura: La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además de la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo.

Iterativo e incremental: El equilibrio correcto entre los casos de uso y la arquitectura es algo muy parecido al equilibrio de la forma y la función en el desarrollo del producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental donde el trabajo se divide en partes más pequeñas o mini proyectos, permitiendo que el equilibrio entre los casos de uso y la arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos

completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto. Al finalizar se realiza una integración de los resultados con lo obtenido de las iteraciones anteriores (4).

Herramienta de Ingeniería de Software Asistido por Ordenadores (CASE) para el modelado

Visual Paradigm es una herramienta de modelado que utiliza el Lenguaje Unificado de Modelado (UML), permite la generación de código e ingeniería inversa. Con una clase de diseño bien especificada, Visual Paradigm puede generar código hasta en 15 lenguajes de programación, entre ellos PHP, que es el que se utilizará en el desarrollo del sistema.

Visual Paradigm actualmente cumple con las políticas de migración a Software Libre en Cuba, ya que es una herramienta multiplataforma que se puede utilizar tanto en Linux como en Windows. Tiene una interfaz muy intuitiva y es de fácil aprendizaje para los desarrolladores. Permite la generación automática de diagramas a partir de descripciones de casos de uso, por ejemplo, diagramas de secuencia, permitiendo la agilidad en el trabajo del analista.

Permite además hacer descripción de los casos de uso dando una gran variedad de plantillas predeterminadas permitiendo personalizarlas. Con Visual Paradigm los analistas pueden generar la documentación necesaria de los artefactos obtenidos hasta el momento en el proyecto.

Visual Paradigm ofrece:

- ✓ Entorno de creación de diagramas para UML 2.0.
- ✓ Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- ✓ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✓ Capacidades de ingeniería directa (versión profesional) e inversa.
- ✓ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- ✓ Disponibilidad de múltiples versiones, para cada necesidad.
- ✓ Disponibilidad de integrarse en los principales entornos integrados de desarrollo (IDE).
- ✓ Disponibilidad de múltiples plataformas.

Visual Paradigm para UML Enterprise: representa la herramienta de modelado más poderosa y de mejor valor en el mercado actual. Combina las funcionalidades de todas las ediciones en una amplia plataforma de modelado visual. Está diseñado para brindar apoyo a arquitectos, desarrolladores, diseñadores UML, analistas de casos de uso del negocio, y modeladores de datos con el fin de agilizar todo el proceso de desarrollo de código del modelo para aplicaciones empresariales complejas. Visual Paradigm para UML soporta un conjunto de lenguas como son: Java, C++, CORBA IDL, PHP y

Python. Emplea una respuesta rápida y bajos requisitos de memoria, lo que permite manejar grandes y complicadas estructuras de proyecto de forma eficiente. Facilita a los desarrolladores de software una herramienta vanguardia para crear aplicaciones de elevada calidad, rápidas y más baratas. Brinda una mejor interfaz gráfica de usuario.

Lenguaje de modelado visual

El lenguaje unificado de modelado (UML, del inglés *Unified Modeling Language*) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software (5).

Sus creadores pretendieron con este lenguaje unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas en un acercamiento estándar. UML establece la base para un estándar en el dominio del análisis y el diseño orientados a objetos, fundado en una amplia base de experiencia de los usuarios.

Permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional más fácil de comprender, para comunicarlas a otras personas que estén involucradas en el proceso de desarrollo de los sistemas, esto se lleva a cabo mediante conjunto de símbolos y diagramas.

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas y proporcionar un estándar, que permite al analista de sistema generar un anteproyecto de varias facetas que sean comprensibles para los clientes, desarrolladores y todos aquellos que estén involucrados en el proceso de desarrollo. Un modelo UML indica qué es lo que supuestamente hará el sistema pero no cómo lo hará.

De forma general las principales características de UML son:

- ✓ Es un lenguaje de representación visual.
- ✓ Permite cambiar diversos elementos gráficos y crear diagramas.
- ✓ Tecnología orientada a objetos.
- ✓ Describe lo que hará el sistema pero no dice cómo implementarlo.
- ✓ Existen varias herramientas CASE que le dan asistencia.

Lenguaje de programación para el desarrollo web

PHP (acrónimo de Hypertext Preprocessor) posee una comunidad de desarrollo muy grande, la cual implementa constantemente mejoras en su código, en la Universidad de las Ciencias Informáticas este lenguaje es utilizado por muchos programadores, pues cuenta con diversas ventajas que le son de gran beneficio a los mismos. Es un lenguaje “del lado del servidor” (esto significa que PHP funciona en un servicio remoto que procesa la página web antes de que sea abierta por el navegador del usuario)

especialmente creado para el desarrollo de páginas web dinámicas. Puede ser incluido con facilidad dentro del código HTML, y permite una serie de funcionalidades tan extraordinarias que se han convertido en el favorito de millones de programadores en todo el mundo.

Características:

- ✓ Gratuito.
- ✓ De gran popularidad.
- ✓ Posee una enorme eficiencia.
- ✓ Se integra de forma sencilla con múltiple bases de datos.
- ✓ Es versátil.
- ✓ Tiene un gran número de funciones predefinidas.

Ventajas:

- ✓ Muy fácil de aprender.
- ✓ Se caracteriza por ser un lenguaje muy rápido.
- ✓ Soporta en cierta medida la orientación a objetos, clases y herencia.
- ✓ Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- ✓ Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- ✓ Capacidad de expandir su potencial utilizando módulos.
- ✓ Posee documentación en una 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 del bajo nivel.

PHP 5

El 13 de julio de 2004, fue lanzado PHP 5, utilizando el motor Zend Engine II (o Zend Engine 2). La versión más reciente de PHP es la 5.3.2, del 4 de marzo de 2010, que incluye todas las ventajas que provee el nuevo Zend Engine 2 como:

- ✓ Mejor soporte para la Programación Orientada a Objetos, que en versiones anteriores era extremadamente rudimentario, con PHP Data Objects.
- ✓ Mejoras de rendimiento.
- ✓ Soporte de espacios de nombres (*namespaces*).
- ✓ Herencia de métodos estáticos.
- ✓ Driver nativo para MySQL (*mysqlnd*) en remplazo de *libmysql*.

- ✓ Mejor soporte a XML (XPath, DOM, etc.).
- ✓ Soporte nativo para SQLite.
- ✓ Soporte integrado para SOAP.
- ✓ Iteraciones de datos.

Sistema gestor de base de datos

PostgreSQL es un Sistema de Gestión de Base de Datos de Objetos Relacional (ORDBMS) que ha sido desarrollado de varias formas desde la década de 1980.

El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto.

PostgreSQL es ampliamente considerado como una de las alternativas de sistemas de base de datos de código abierto.

Características fundamentales:

- ✓ **Transacciones:** Permiten el paso entre dos estados consistentes manteniendo la integridad de los datos.
- ✓ **Integridad referencial:** PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- ✓ **Bloqueos de tablas y filas:** PostgreSQL ofrece varios modos de bloqueo para controlar el acceso concurrente de los datos en tablas. Algunos de estos modos de bloqueo los adquiere PostgreSQL automáticamente antes de la ejecución de una declaración, mientras que otros son proporcionados para ser usados por las aplicaciones.
- ✓ **Limitaciones (*constraints*) y disparadores (*triggers*):** Tienen la función de mantener la integridad y consistencia de la base de datos. Ejecución de acciones antes o después de un evento de base de datos.
- ✓ **Múltiples tipos de datos predefinidos:** Como todos los manejadores de base de datos, PostgreSQL implementa los tipos de datos definidos para el estándar SQL3 y aumenta algunos datos.
- ✓ **Soporte de tipos y funciones de usuarios:** PostgreSQL soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario. Incorpora una estructura de datos Array.
- ✓ **Conectividad:** TCP/IP, JDBC y ODBC.
- ✓ **Interfaz con diversos lenguajes:** C, C++, Java, Delphi, Python, Perl, PHP, Bash.

Ventajas de PostgreSQL:

- ✓ **Instalación ilimitada:** Con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software.
- ✓ **Soporte:** PostgreSQL cuenta con una importante comunidad de desarrollo que constantemente realizan mejoras al código y trabajan en la implementación de nuevas versiones.
- ✓ **Ahorros considerados en costos de operación:** PostgreSQL ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que otros productos, conservando todas las características, estabilidad y rendimiento.
- ✓ **Estabilidad y confiabilidad:** Es un sistema gestor de base de datos estable, que ha dado garantías ofreciendo un gran rendimiento en aplicaciones complejas y de grandes niveles de acceso a la base de datos.
- ✓ **Extensible:** el código fuente está disponible para todos sin costo. Esto es complementado por la comunidad de desarrolladores de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días.
- ✓ **Multiplataforma:** PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y ahora en versión nativa para Windows.
- ✓ **Diseñado para ambientes de alto volumen:** PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes. Los principales proveedores de sistemas de base de datos comerciales usan también esta tecnología, por las mismas razones.
- ✓ **Herramientas gráficas de diseño y administración de base de datos:** existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin, pgAccess) y para hacer diseño de bases de datos (Tora, Data Architect).

Desventajas de PostgreSQL:

- ✓ Tiene un límite de 8K por línea, aunque se puede aumentar a 32K recopilando, pero con una disminución considerable del rendimiento.
- ✓ Es algo lento.

Axure

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

Axure RP permite componer la página web visualmente, añadiendo, quitando y modificando los elementos con suma facilidad, demostrando su grado de especialización en las anotaciones. En este

punto, permite especificar el estado de cada elemento (Propuesto, Aceptado, Incorporado), el beneficio esperado (Crítico, Importante, Útil), el riesgo, la estabilidad, a quién va dirigido y a quién se le asignará la tarea. Otra característica a destacar del Axure RP es que permite un diseño colaborativo.

Marcos de trabajo: Zend Framework, ExtJS y Doctrine

Un marco de trabajo (*framework*) no es más que una estructura de soporte definida mediante la cual otro proyecto de software puede ser desarrollado y organizado. Puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. Son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los detalles de bajo nivel de proveer un sistema funcional. Es el esqueleto sobre el cual varios objetos son integrados para una solución dada. No es más que una base de programación que atiende a sus descendientes (manejado de una forma estructural y/o en cascada) posibilitando cualquier respuesta ante las necesidades de sus miembros, o secciones de una página web (6).

Zend Framework

Conocido también como Zf, Zend Framework es un marco de trabajo de código abierto para desarrollar tanto aplicaciones como servicios web en PHP5. Usa una implementación completa orientada a objetos. Sus componentes fueron desarrollados con una baja dependencia unos de otros, lo que posibilita al programador el uso de estos por separado. Ofrece un gran rendimiento y una robusta implementación del patrón Modelo Vista Controlador (MVC). Incluye una muy completa gama de componentes, que sirven al programador para realizar toda la abstracción de la base de datos, la autenticación de usuarios, el acceso a los servicios web, entre otras. Es un marco de trabajo extensible, reutilizable y que puede ser integrado fácilmente con otros marcos de trabajos y librerías de clases. Tiene una buena documentación y una amplia comunidad de desarrolladores. Presenta soporte para la internacionalización de las aplicaciones e implementa clases que permiten la realización de pruebas a la aplicación en desarrollo [8].

ExtJS

ExtJS es un nuevo framework JavaScript para la programación del lado del cliente. Está basado originalmente en la interfaz de usuarios de Yahoo (YUI, del inglés *Yahoo User Interfaces*). Es una

potente herramienta para crear las interfaces de usuarios de una aplicación web y que ofrece muchas facilidades a la hora de vincularse con un lenguaje del lado del servidor (Java, PHP, entre otros).

Ofrece gran cantidad de controles (*widgets*) para crear interfaces de usuario complejas. PHP-Ext es una librería de código abierto que permite potenciar la capa de interfaz de usuario (UI) de JavaScript en nuestras aplicaciones. Para ello ofrece una serie de librerías (compatibles con PHP 4 y 5) para integrar ExtJS en el sistema a desarrollar. Funciona como un mapeo en clases de la librería JS. Entre las posibilidades que ofrece se encuentran la creación de formularios, combos, paneles de tablas o menús. Además ayuda a la comunicación entre el cliente y el servidor mediante JSON y XML.

Tiene un sistema dual de licencia: Comercial y Código Abierto (*Open Source*). Este framework 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).

Ventajas:

- ✓ Código reutilizable.
- ✓ Independiente o adaptable a frameworks diferentes.
- ✓ Orientada a la programación de interfaces de tipo escritorio en la web.
- ✓ El API es homogeneizado independientemente del adaptador usado. Los controles siempre se verán igual.
- ✓ Soporte comercial.
- ✓ Una extensa comunidad de usuarios.
- ✓ La orientación a objetos intensa te hará modular todos los scripts.
- ✓ El diseño está completamente separado de la funcionalidad.
- ✓ Funciones comunes como validación, combos editables, ventanas deslizables (con minimizar y maximizar) y tablas editables, son muy fáciles de implementar.
- ✓ Utilizando AJAX JSON como mecanismos de comunicación con el servidor.
- ✓ Implementación basada en patrones de diseño.
- ✓ Amplia librería de componentes gráficos fácilmente extensibles.
- ✓ Buena y amplia documentación, así como también su comunidad.

Doctrine

Doctrine es un potente y completo sistema ORM para PHP 5.2 o mayor, con una capa de abstracción de la base de datos (DBAL del inglés *Database Abstraction Layer*) incorporada.

Entre muchas otras cosas ofrece 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 de 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 frameworks de desarrollo utilizados actualmente (7).

Entorno de Desarrollo Integrado (IDE): NetBeans

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 de programación (*Bugs*) que pudiesen existir.

Ventajas:

- ✓ La plataforma NetBeans permite que las aplicaciones se desarrollen a partir de un conjunto de módulos o componentes de software.
- ✓ Un módulo contiene clases de Java escritas para interactuar con las APIs de NetBeans y un archivo especial que lo identifica como módulo.
- ✓ NetBeans IDE es fácil de instalar y de uso instantáneo y se ejecuta en varias plataformas incluyendo Windows, Linux y Mac OS X y Solaris.
- ✓ Además del soporte completo para todas las plataformas Java (Java SE, Java EE, Java ME y Java FX), NetBeans IDE 6.8 es la herramienta ideal para el desarrollo de software con PHP, AJAX y JavaScript y C/C++.

Servidor web: Apache

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

De acuerdo al estudio realizado en internet sobre los servidores web, Apache es sin lugar a dudas uno de los mejores servidores web.

Apache ha sido, desde su salida al mercado, uno de los servidores de mayor notoriedad, considerado por muchos el proyecto punta del movimiento del software libre. Apache se caracteriza por ser un servidor ligero, altamente configurable y de amplia explotación, según Netcraft, empresa dedicada a la realización de encuestas a nivel global y estudios sobre el tráfico en internet, el mayor por ciento de los servidores web actuales son servidores Apache.

Características fundamentales de Apache:

- ✓ Tecnología gratuita de código fuente abierta.
- ✓ Multiplataforma, funcionando tanto en Windows como en Linux o Unix.
- ✓ Soporte para interfaz de entrada común (CGI del inglés *Common Gateway Interface*).
- ✓ Soporte para varios lenguajes: PHP, Java, Perl y librerías ASP.
- ✓ Soporte para el protocolo HTTP.
- ✓ Soporte de host virtuales.
- ✓ Servidor proxy integrado.
- ✓ Estable.

Aplicación web

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

Ventajas de una aplicación web:

- ✓ No es necesario instalar la aplicación en su ordenador.
- ✓ Puede trabajar desde cualquier lugar del mundo con conexión a Internet.
- ✓ Puede utilizar dispositivos móviles.
- ✓ Ahorro de costos en hardware puesto que cualquier ordenador puede utilizar un navegador.
- ✓ Ahorro de costos en formación puesto que actualmente todos los usuarios conocen el funcionamiento de los navegadores y el correo electrónico.
- ✓ Las actualizaciones se incorporan en el servidor, por lo que todos los usuarios tienen las nuevas versiones a la vez e inmediatamente.
- ✓ Aumenta la seguridad frente a un virus y desastres puesto que los datos están alojados en un servidor con las más altas cotas de seguridad.

- ✓ Seguridad frente a accesos indeseados mediante la protección con contraseñas y servidores seguros.
- ✓ Posibilidad de compartir información con delegaciones y otras empresas.
- ✓ Posibilidad de tener un contacto más cercano con el usuario a través de servicios de mensajería y chats.

Modelo cliente-servidor

Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa –el servidor- que le da la respuesta. El uso y aplicación del modelo Cliente-Servidor está potentemente ligado al desarrollo de sistemas Web.

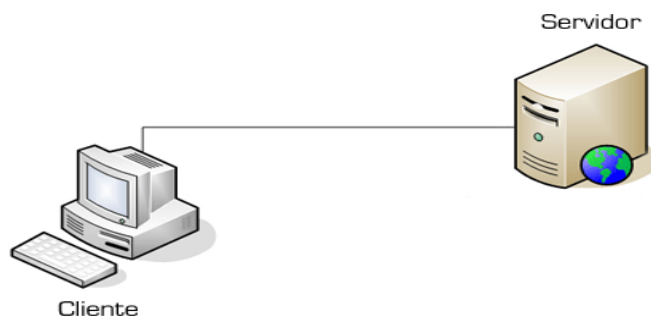


Figura 1 Modelo Cliente-Servidor

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 puede 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 presión, 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.

Patrón de arquitectura Modelo Vista Controlador (MVC)

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

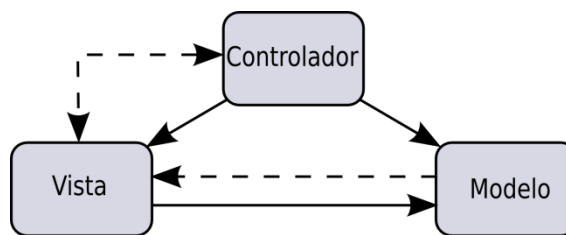


Figura 2 Patrón de Arquitectura MVC

La correcta implementación de este patrón dispone, como se observa en la figura anterior, de tres entidades básicas: el Modelo, la Vista y el Controlador.

Modelo: Encapsula los datos y las funcionalidades. Es independiente de cualquier representación de salida y/o comportamiento de entrada.

Vista: Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.

Controlador: Recibe las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas u otros. Los eventos son traducidos a solicitudes de servicio para el modelo o la vista.

Este modelo 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 y 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.

Fundamentación de las herramientas, metodología y tecnologías a utilizar

Se considera a RUP el proceso de desarrollo más general de los existentes actualmente, posee grandes ventajas para el análisis, implementación y documentación de sistemas orientados a objetos. Esta metodología está preparada para desarrollar proyectos grandes y complejos, es muy organizada y genera desde sus inicios una documentación robusta y especificada de todo el proceso de manera general. Se caracteriza por ser iterativo e incremental, centrado en la arquitectura y dirigido por casos de uso, la arquitectura provee la estructura sobre la cual guiar el trabajo en iteraciones, mientras que los casos de uso definen las metas y dirigen el trabajo en cada iteración.

La herramienta utilizada para la modelación del sistema es Visual Paradigm, aunque es un software propietario, la universidad posee su licencia por tanto puede utilizarlo. Posee cualidades como la generación automática de diagramas a partir de 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.

El lenguaje a utilizar durante el desarrollo del sistema es PHP pues está diseñado principalmente para el desarrollo web, controla el manejo de excepciones, posee una biblioteca nativa de funciones sumamente amplia e incluida y es uno de los lenguajes más utilizados actualmente lo que trae consigo que exista una numerosa comunidad de desarrolladores. Es libre, por lo que es de fácil acceso para todos.

Se usarán varios marcos de trabajo, entre ellos: Doctrine para la capa de abstracción de base de datos pues brinda un gran rendimiento en ejecución y admite escribir de forma segura consultas muy complejas, ExtJS para programar la capa de presentación, ya que permite crear interfaces de usuarios muy amigables y eficaces además de robustas, y Zend Framework para programar la capa de negocio, pues posibilita diseñar su estructura de componentes individualmente los cuales forman una potente y extensible aplicación web al ser mezclados.

Como entorno integrado de desarrollo se usará NetBeans porque es un producto de código abierto 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 que las aplicaciones se desarrollen a partir de un conjunto de módulos o componentes de software, además es la herramienta ideal para el desarrollo con PHP. Se utiliza Axure para el diseño de los Prototipos no Funcionales.

Se usará, como servidor web Apache, que desde su salida al mercado ha sido uno de los servidores de mayor notoriedad, considerado por muchos el proyecto punta del movimiento de software libre, se caracteriza por ser un servidor ligero, altamente configurable, de amplia explotación y un valioso grado

de calidad y fortaleza para las implementaciones que utilizan el protocolo HTTP, soporta el lenguaje PHP que es el que será usado para la implementación del sistema.

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.

Conclusiones

En este capítulo se realizó un estudio de sistemas que realizan procesos similares a los que se desarrollan en la investigación, dando una panorámica general de cómo estos se implementan hoy en día, se exponen algunos ejemplos de los mismos, lo que contribuye con conocimientos e ideas para la realización del sistema.

Se ha realizado un análisis de las tecnologías informáticas a emplear a lo largo del desarrollo de la solución del problema, haciendo énfasis y decidiendo utilizar las propuestas por la Dirección de Informatización de la UCI, teniendo en cuenta las tendencias de migración a Software Libre.

Capítulo 2: Características del sistema

Introducción

En este capítulo se define por qué se utiliza el modelo de dominio para describir los procesos del negocio, se realiza una descripción de los requisitos funcionales y no funcionales, los actores y los casos de uso del sistema, realizando además las descripciones de cada una de ellas, con el objetivo de guiar el desarrollo hacia el sistema deseado, con objetivos generales concretos y cuáles beneficios reporta para los actores del mismo. En este capítulo quedarán identificadas las necesidades de los usuarios como requisitos.

Sistema Unificado de Gestión de Fuerza de Trabajo Calificada (GeForza)

El Sistema Unificado de Gestión de Fuerza de Trabajo Calificada, es el sistema para la gestión de toda la información llevada por el departamento de Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación. Es una aplicación integral que se encuentra dividida en varios módulos, que se encargan de llevar los diferentes procesos del departamento: Administración, Matriz, Plan de Ingreso a los Centros Universitarios Municipales (CUM), Plan de Distribución, Réplica y Plan de Ingreso. El módulo Administración se encarga de garantizar la seguridad del sistema, además de brindarle todos los clasificadores y nomencladores a los demás módulos. El módulo Matriz se encarga de la recopilación en todo el país de la información necesaria para el departamento, gestiona la disponibilidad de graduados tanto de noveno grado como del nivel medio, la demanda de graduados que realizan los organismos y la existencia de empleados; este módulo brinda esta información a los demás módulos para que realicen sus planes. El módulo Plan de Distribución se encarga de asignarles un empleo a todos los graduados del país, para ello carga los datos necesarios del módulo Matriz. El módulo Plan de Ingreso a las CUM se encarga de realizar el plan de ingreso en los centros universitarios municipales. El módulo Réplica es el encargado de mantener las bases de datos del sistema actualizadas y de mantener una concurrencia en la información de las mismas. El módulo Plan de Ingreso, objeto de la presente investigación, se encarga de realizar los planes de ingreso tanto a la Educación Superior como a los Técnicos Medio y Obreros Calificados.

Modelo de dominio

Después de efectuar un profundo análisis del problema, se llega a la conclusión con la presente investigación, que es necesario diseñar un modelo de dominio, debido a que estos ofrecen un negocio definido y para que el equipo de desarrollo pueda entender lo que se quiere. Para ello se realiza este modelado, el cual contribuye posteriormente a identificar las clases que se utilizarán en el sistema.

Un modelo de dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema. A continuación se muestra el diagrama de clases del modelo de dominio:

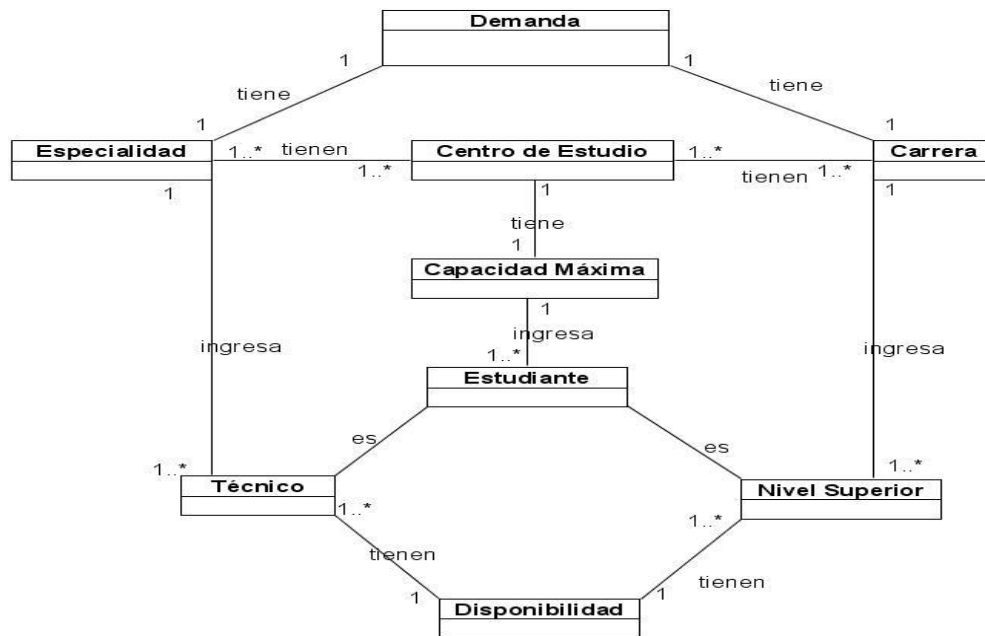


Figura 3 Diagrama de Clases del modelo de dominio

Definición de las clases del modelo de dominio

Demanda: Es la cantidad de especialistas que necesita el país para cada una de las carreras.

Especialidad: Estudios de la enseñanza técnica y profesional repartidos en una serie de años con los que se obtiene un título profesional.

Carrera: Estudios universitarios repartidos en una serie de años con los que se obtiene un título profesional.

Centro de estudio: Lugar donde se estudia una carrera o especialidad determinada.

Capacidad máxima: Es la capacidad que tienen un centro de estudio disponible para una carrera.

Estudiante: Persona que cursa estudios en un centro docente.

Técnico Medio: Estudios del nivel medio en una especialidad.

Nivel Superior: Estudios del nivel superior en una carrera.

Disponibilidad: Cantidad de personas disponibles para cursar estudios.

Especificación de los requerimientos de software

El análisis de requerimientos es la tarea que plantea la asignación de software a nivel de sistemas y el diseño del programa. El análisis de requerimientos facilita al ingeniero de sistemas especificar la

función y comportamiento de los programas, indicar la interfaz con otros elementos del sistema y establecer las ligaduras de diseño que debe cumplir el programa. Además permite al ingeniero refinar la asignación de software y representa el dominio de la información que será tratada por el programa, ofrece al diseñador la representación de la información y las funciones que pueden ser traducidas en datos, arquitectura y diseño procedimental. Finalmente, la especificación de requerimientos suministra al desarrollador y al cliente, los medios para valorar la calidad de los programas, una vez que se haya construido (5).

Requerimientos funcionales

RF 1. Realizar la disponibilidad de graduados de noveno grado.

1.1. Adicionar la disponibilidad de noveno grado con los siguientes datos:

- Provincia.
- Municipio.
- Matrícula en último año.
- % de Eficiencia.
- Estimado de graduados.

1.2. Modificar los siguientes datos de la disponibilidad de noveno grado:

- Matrícula en último año.
- % de Eficiencia.

1.3. Eliminar la disponibilidad de noveno grado:

- Eliminar la disponibilidad o las disponibilidades que se encuentran seleccionadas en ese momento.

1.4. Búsqueda avanzada por los siguientes criterios de búsqueda:

- Provincia.
- Municipio.

1.5. Limpiar Búsqueda:

- Elimina de la ventana lo que se había buscado y muestra todas las disponibilidades que se encuentran.

1.6. Cargar datos desde el Sistema Matriz:

- Carga todas las disponibilidades que el sistema matriz ha recogido en el transcurso del año y lo guarda en la base de datos del sistema.

1.7. Exportar:

- Muestra en un archivo PDF o Excel toda la información que se encuentra en la ventana, además imprime y guarda.

1.8. Seleccionar año:

- Año en que se desea trabajar.

RF 2. Realizar la disponibilidad de graduados de la enseñanza media.

2.1. Adicionar la disponibilidad de graduados de la enseñanza media con los siguientes datos:

- Organismo.
- Provincia.
- Fuente de ingreso.
- Matrícula en último año.
- % de Eficiencia.
- Estimado de graduados.

2.2. Modificar los siguientes datos de la disponibilidad de graduados de la enseñanza media:

- Matrícula en último año.
- % de Eficiencia.

2.3. Eliminar la disponibilidad de graduados de la enseñanza media:

- Elimina la disponibilidad o las disponibilidades que se encuentran seleccionadas en ese momento.

2.4. Búsqueda avanzada por los siguientes criterios de búsqueda:

- Organismo.
- Provincia.
- Fuente de ingreso.

2.5. Limpiar búsqueda:

- Elimina de la ventana lo que se había buscado y muestra todas las disponibilidades que se encuentran.

2.6. Cargar datos desde el Sistema Matriz:

- Carga todas las disponibilidades que el sistema matriz ha recogido en el transcurso del año y lo guarda en la base de datos del sistema.

2.7. Exportar:

- Muestra en un archivo PDF o Excel toda la información que se encuentra en la ventana, además imprime y guarda.

2.8. Seleccionar año:

- Año en que se desea trabajar.

RF 3. Realizar la capacidad docente de los preuniversitarios provinciales.

3.1. Adicionar la capacidad docente de los preuniversitarios provinciales con los siguientes datos:

- Provincia.
 - Fuente de ingreso.
 - Matrícula.
- 3.2. Modificar el siguiente dato de la capacidad docente de los preuniversitarios provinciales:
- Matrícula.
- 3.3. Eliminar la capacidad docente de los preuniversitarios provinciales:
- Elimina la matrícula o las matrículas que se encuentran seleccionadas en ese momento.
- 3.4. Búsqueda avanzada por los siguientes criterios de búsqueda:
- Provincia.
 - Fuente de ingreso.
- 3.5. Limpiar búsqueda:
- Elimina de la ventana lo que se había buscado y muestra todas las matrículas que se encuentran.
- 3.6. Exportar:
- Muestra en un archivo PDF o Excel toda la información que se encuentra en la ventana, además imprime y guarda.
- 3.7. Seleccionar año:
- Año en que se desea trabajar.

RF 4. Realizar la capacidad docente disponible para los centros universitarios.

- 4.1. Adicionar la capacidad docente disponible para los centros universitarios con los siguientes datos:
- Organismo.
 - Centro de estudio.
 - Carrera.
 - Matrícula máxima.
 - Matrícula para ese centro, en esa carrera, por provincia.
- 4.2. Modificar el siguiente dato de la capacidad docente disponible para los centros universitarios:
- Matrícula.
- 4.3. Eliminar la capacidad docente disponible para los centro universitarios:
- Eliminar la matrícula o las matrículas que se encuentran seleccionadas en ese momento.
- 4.4. Búsqueda avanzada por los siguientes criterios de búsqueda:
- Organismo.
 - Centro de estudio.

- Carrera.
 - Provincia.
- 4.5. Limpiar búsqueda:
- Elimina de la ventana lo que se había buscado y muestra todas las matrículas que se encuentran.
- 4.6. Exportar:
- Muestra en un archivo PDF o Excel toda la información que se encuentra en la ventana, además imprime y guarda.
- 4.7. Seleccionar año:
- Año en que se desea trabajar.

RF 5. Realizar la demanda de los Técnicos Medio y Obreros Calificados.

- 5.1. Adicionar la demanda de los Técnicos Medio y Obreros Calificados con los siguientes datos:
- Organismo.
 - Carrera.
 - Provincia.
 - Municipio.
 - Demanda.
- 5.2. Modificar el siguiente dato de la demanda de los Técnicos Medio y Obreros Calificados:
- Demanda.
- 5.3. Eliminar la demanda de los Técnicos Medio y Obreros Calificados:
- Eliminar la demanda o las demandas que se encuentran seleccionadas en ese momento.
- 5.4. Búsqueda avanzada por los siguientes criterios de búsqueda:
- Organismo.
 - Carrera.
 - Provincia.
 - Municipio.
- 5.5. Limpiar búsqueda:
- Elimina de la ventana lo que se había buscado y muestra todas las demandas que se encuentran.
- 5.6. Cargar datos desde el Sistema Matriz:
- Carga todas las demandas que el sistema matriz ha recogido en el transcurso del año y lo guarda en la base de datos del sistema.
- 5.7. Exportar:

- Muestra en un archivo PDF o Excel toda la información que se encuentra en la ventana, además imprime y guarda.

5.8. Seleccionar año:

- Año en que se desea trabajar.

RF 6. Realizar la demanda de graduados de la educación superior.

6.1. Adicionar la demanda de graduados de la enseñanza superior con los siguientes datos:

- Organismo.
- Carrera.
- Provincia.
- Demanda.

6.2. Modificar el siguiente dato de la demanda de graduados de la enseñanza superior:

- Demanda.

6.3. Eliminar la demanda de graduados de la enseñanza superior:

- Elimina la demanda o las demandas que se encuentran seleccionadas en ese momento.

6.4. Búsqueda avanzada por los siguientes criterios de búsqueda:

- Organismo.
- Provincia.
- Carrera.

6.5. Limpiar búsqueda:

- Elimina de la ventana lo que se había buscado y muestra todas las demandas que se encuentran.

6.6. Cargar datos desde el Sistema Matriz:

- Carga todas las demandas que el sistema matriz ha recogido en el transcurso del año y lo guarda en la base de datos del sistema.

6.7. Exportar:

- Muestra en un archivo PDF o Excel toda la información que se encuentra en la ventana, además imprime y guarda.

6.8. Seleccionar año:

- Año en que se desea trabajar.

RF 7. Realizar plan de ingreso a la enseñanza superior.

7.1. Adicionar el plan de ingreso a la enseñanza superior con los siguientes datos:

- Provincia.
- Carrera.

- Centro de estudio.
- Capacidad.
- Demanda por cubrir.
- Fuente de ingreso.
- Nuevo ingreso del centro.
- Sobrante provincia.
- Nuevo ingreso provincial.

7.2. Modificar los siguientes datos del plan de ingreso a la enseñanza superior:

- Nuevo ingreso del centro.
- Sobrante provincial.
- Nuevo ingreso provincial.

7.3. Eliminar el plan de ingreso a la enseñanza superior:

- Elimina la asignación de ingreso o las asignaciones de ingreso que se encuentran seleccionadas en ese momento.

7.4. Búsqueda avanzada por los siguientes criterios de búsqueda:

- Centro de estudio.
- Carrera.
- Provincia.

7.5. Limpiar búsqueda:

- Elimina de la ventana lo que se había buscado y muestra todas las asignaciones de ingreso que se encuentran.

7.6. Ejecutar algoritmo:

- Ejecuta el algoritmo matemático y realiza la asignación de ingreso.

7.7. Exportar:

- Muestra en un archivo PDF o Excel toda la información que se encuentra en la ventana, además imprime y guarda.

7.8. Seleccionar año:

- Año en que se desea trabajar.

RF 8. Realizar plan de ingreso a los Técnicos Medios (TM) y Obreros Calificados (OC).

8.1. Adicionar el plan de ingreso a los TM y OC con los siguientes datos:

- Especialidad.
- Demanda.
- % de Eficiencia.

- Propuesta de plan de ingreso para el año en cuestión.

8.2. Modificar el siguiente dato del plan de ingreso a los TM y OC:

- Propuesta de plan de ingreso para el año en cuestión.

8.3. Eliminar el plan de ingreso para los TM y OC:

- Elimina la asignación de ingreso o las asignaciones de ingreso que se encuentran seleccionadas en ese momento.

8.4. Actualizar:

- Carga todas las asignaciones de ingreso que se encuentren en la base de datos.

8.5. Ejecutar algoritmo:

- Ejecuta el algoritmo matemático y realiza la asignación de ingreso.

8.6. Exportar:

- Muestra en un archivo PDF o Excel toda la información que se encuentra en la ventana, además imprime y guarda.

8.7. Selecciona año:

- Año en que se desea trabajar.

RF 9. Mostrar carreras por fuentes de ingreso (provincial)

9.1. Mostrar para una provincia dada una lista de todas las carreras y para cada una de ellas la cantidad por fuente de ingreso.

RF 10. Mostrar carreras por fuentes de ingreso (nacional)

10.1. Mostrar una lista de todas las carreras y para cada una de ellas la cantidad por fuentes de ingreso.

RF 11. Mostrar pronóstico de graduados por fuentes de ingreso y provincia

11.1. Mostrar para cada provincia una lista de todas las fuentes de ingreso y para cada una de ellas la matrícula en último año, el porcentaje de eficiencia y el estimado de graduados.

RF 12. Mostrar pronóstico de graduados por fuentes de ingreso nacional

12.1. Mostrar una lista de todas las fuentes de ingreso y para cada una de ellas la matrícula en último año, el porcentaje de eficiencia y el estimado de graduados.

RF 13. Mostrar carreras por fuentes de ingreso y organismos

13.1. Mostrar de un organismo una lista de todas las carreras que provienen de una fuente de ingreso dada y dar la cantidad para cada provincia.

RF 14. Mostrar carreras por provincia

14.1. Mostrar una lista de todas las carreras y la cantidad para cada provincia.

RF 15. Mostrar demanda por carreras

15.1. Mostrar una lista de todas las carreras y la demanda de cada una de ellas para cada provincia.

RF 16. Mostrar demanda por organismo/carreras

16.1. Mostrar para cada organismo una lista de sus carreras y la demanda de cada una de ellas para cada provincia.

RF 17. Mostrar demanda por carrera/organismos

17.1. Mostrar una lista de todas las carreras y dentro de ellas una lista de todos los organismos a los que esa carrera pertenece y la demanda de cada organismo para cada provincia.

RF 18. Mostrar pronóstico de graduados por fuentes de ingreso

18.1. Mostrar una lista de todas las fuentes de ingreso y el pronóstico de graduados de cada una de ellas para cada provincia.

RF 19. Mostrar proyección del nuevo ingreso por carrera

19.1. Mostrar una lista de todas las carreras y para cada una de estas la proyección de nuevo ingreso para cada provincia.

RF 20. Mostrar proyección del nuevo ingreso por centro de estudio

20.1. Mostrar para un centro de estudio dado una lista de todas las carreras que se estudian en él y para cada una de ellas la proyección de nuevo ingreso para cada provincia.

RF 21. Mostrar fuentes de ingreso para el ingreso por organismo formador

21.1. Mostrar para un organismo dado y en cada provincia, una lista de fuentes de ingreso y para cada una de ellas la matrícula en último año, el porcentaje de eficiencia y el estimado de graduados.

RF 22. Mostrar capacidad docente disponible para el nuevo ingreso a la educación superior

22.1. Mostrar una lista de todas las carreras y para cada una de ellas la capacidad docente disponible para el nuevo ingreso por provincia.

RF 23. Mostrar fuentes de ingreso sobrantes

23.1. Mostrar una lista de todas las fuentes de ingreso y para cada una de ellas la cantidad de ingreso sobrante por provincia.

RF 24. Mostrar emisión del plan para TM y OC tanto a nivel nacional como provincial

24.1. Mostrar una lista de los indicadores a nivel nacional con los siguientes datos:

- Nombre de los indicadores.
- Total para cada indicador.

24.2. Mostrar una lista de los indicadores provincial con los siguientes datos:

- Nombre de la provincia.
- Nombre de los indicadores.
- Total para cada indicador.

RF 25. Mostrar continuidad de estudios de los graduados de noveno grado hacia la educación técnica y profesional y el preuniversitario

25.1. Mostrar una lista de los indicadores a nivel nacional con los siguientes datos:

- Código.
- Nombre de los indicadores.
- Demanda TM (año) y OC (año).
- Índice de eficiencia.
- Ingreso necesario.
- Propuesta de plan (año).
- Diferencia entre el ingreso necesario y la propuesta de plan de ingreso.

25.2. Mostrar una lista de los indicadores provincial con los siguientes datos:

- Familia de especialidades.
- Demanda TM (año) y OC (año).
- Índice de eficiencia.
- Ingreso necesario (año).
- Propuesta de plan de ingreso (año).

25.3. Diferencia entre el ingreso necesario y la propuesta de plan de ingreso.

Requerimientos no funcionales

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.

RNF 26.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.

Eficiencia

El sistema debe funcionar con un máximo rendimiento pero ajustado a bajas prestaciones de las computadoras debido que no todos los organismos poseen tecnologías de punta.

La aplicación debe garantizar un acceso fácil y rápido, contando con un menú que satisfaga las necesidades de los usuarios.

Soporte

El sistema deberá presentar un manual de usuario, permitiendo con ello un correcto uso de sus funcionalidades y brindarle al usuario una mayor experiencia del trabajo con el mismo.

La documentación del sistema debe estar actualizada en todos los aspectos, fases de trabajo y ciclos de desarrollo del mismo, permitiendo con ello un respaldo tanto ingenieril como legal del desarrollo de dicho sistema.

El sistema contará con un grupo de soporte y asesoría al cliente del producto destinado a brindar asesoría y soporte técnico al mismo.

Restricciones de diseño

Lenguaje de programación: PHP 5.

Como gestor de base de datos se utilizará PostgreSQL 8.3 o superior.

Los marcos de trabajo que se utilizarán son: Zend Framework para la capa de negocio, ExtJS para la capa de presentación y Doctrine para la capa de acceso a datos.

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.

Hardware

Se necesitará una impresora, para la impresión de los reportes del sistema.

Las PC clientes deben tener las siguientes características: memoria RAM de 256 MB o superior, 20 GB o más de disco duro, 1.2 GHz o más de procesador.

Todas las PC clientes deben estar conectadas a la red para poder acceder a la aplicación.

Se requiere de un servidor para Bases de Datos con las siguientes características:

- Servidor Xeon a 3.0 GHz.
- 1 GB de memoria RAM.
- Dos discos duros de 36 y 250 GB, este último con dos particiones.

Software

Para el cliente: se usará un sistema operativo Windows 95, cualquier versión superior o GNU/Linux.

Las computadoras cliente del sistema y ubicadas en el dominio de la organización, deben tener el navegador Mozilla Firefox.

Para el servidor: sistema operativo Windows Server 2000 o superior o Linux; Ubuntu Server 7.10 o superior.

Servidor Web: Apache 2.0 o superior.

Gestor de base de datos: PostgreSQL.

Apariencia o interfaz externa

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

Requisitos de licencia

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

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.

Propuesta de sistema

Con el objetivo de mejorar las dificultades que existen con el proceso de plan de ingreso a la educación media y superior se decidió realizar una aplicación web, que se encargue de realizar los planes de ingreso tanto a la educación superior como a los Técnicos Medio y Obreros Calificados, la misma tendrá una interfaz amigable, simulando una aplicación de escritorio que permite una fácil interacción con el cliente.

Esta aplicación debe ser capaz de gestionar la disponibilidad de graduados tanto de los estudiantes de noveno grado como de los estudiantes del nivel medio, de gestionar la demanda de graduados tanto para las carreras de nivel superior como para las especialidades de técnicos medio y obreros calificados, de gestionar la capacidad docente disponible para los centro universitarios y la capacidad docente de los preuniversitarios por provincia. Luego de gestionada toda esta información que puede ser cargada del módulo Matriz o puede ser entrada manualmente, entonces la aplicación estará lista para realizar los planes de ingreso tanto para el nivel superior como para los técnicos medio y obreros calificados.

Modelo del sistema

Actores y casos de uso del sistema

Considerando que la definición de actor del sistema relaciona todo aquello que intercambie información con la aplicación y que puede representar el rol que juega una o varias personas, un equipo o un sistema automatizado, además de intercambiar información con él o ser recipientes pasivos de información, se definen los siguientes actores.

Actor	Descripción
Funcionario	Representa a todos los usuarios que tienen acceso a la aplicación, o sea el Funcionario CAP y él mismo. El cual es el encargado de realizar todas las funcionalidades del sistema, brindándole así los diferentes servicios para realizar la gestión de los planes de ingreso a la Educación Superior y a los Técnicos Medio y Obreros Calificados.
Funcionario CAP	Es la persona encargada de realizar algunas de las funcionalidades del sistema, brindándole así los diferentes servicios para realizar la gestión de los Planes de Ingreso a los Técnicos Medio y Obreros Calificados.

Tabla 3 Descripción textual de los actores del Sistema

Diagrama de casos de uso del sistema

A continuación se muestra el diagrama de casos de uso del sistema.

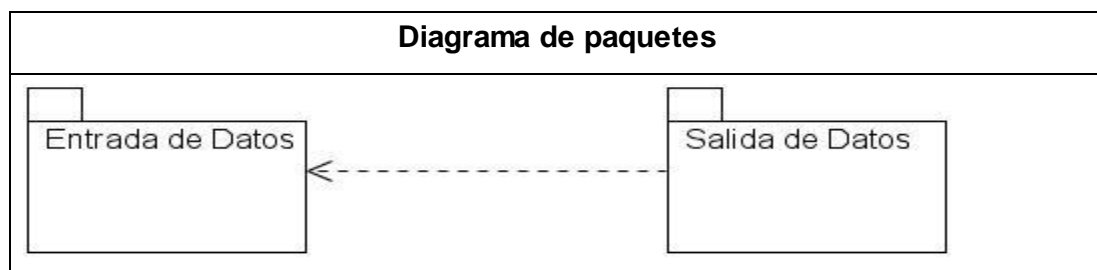


Figura 4 Diagrama de Paquetes

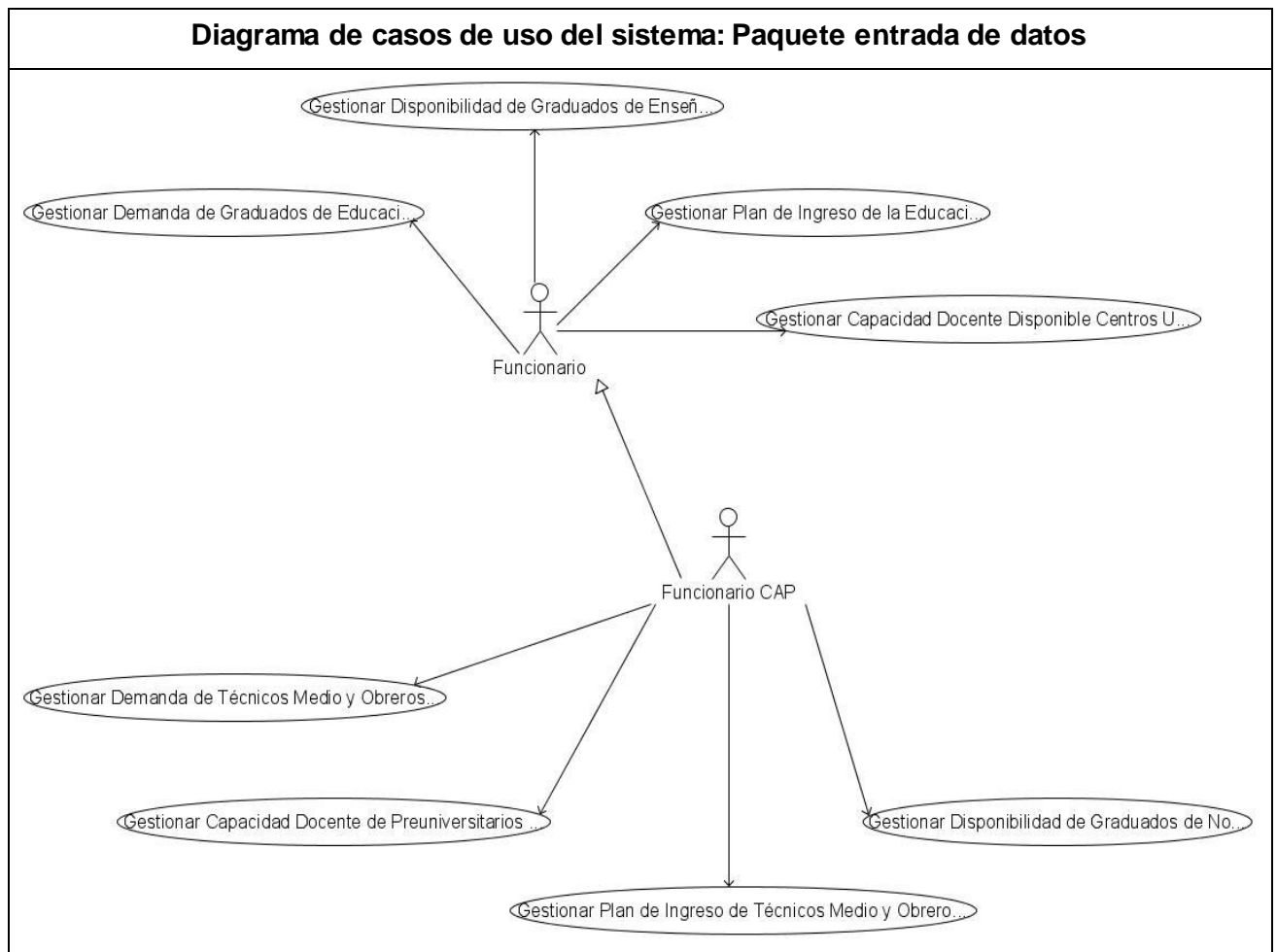


Figura 5 Diagrama de casos de uso del sistema Paquete Entrada de datos

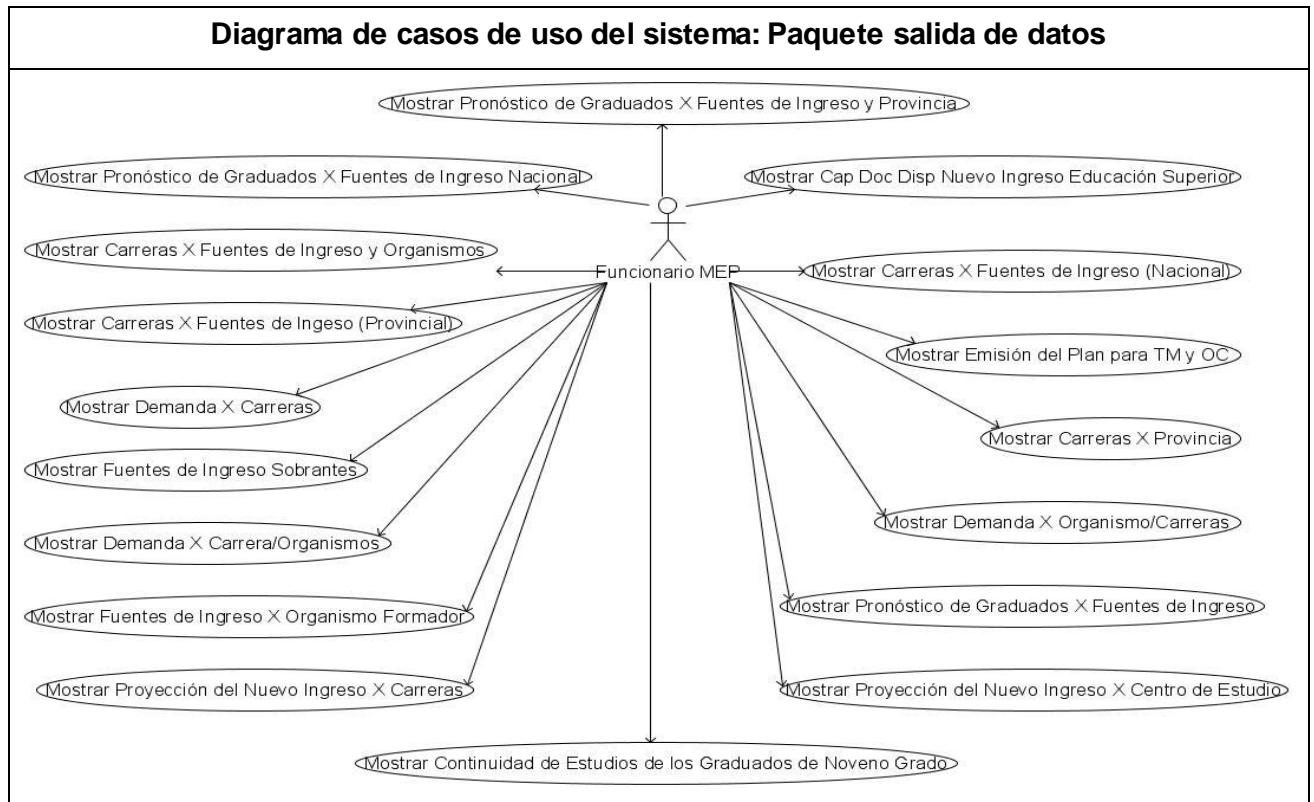


Figura 6 Diagrama de casos de uso del sistema Paquete Salida de Datos

Descripción abreviada de los casos de uso del sistema

Mediante la descripción de los casos de uso del sistema se detalla la secuencia de eventos que los actores llevan a cabo para completar un proceso a través de la aplicación.

A continuación se muestran las descripciones abreviadas de cada caso de uso, las descripciones extendidas están disponibles en los anexos.

Caso de Uso:	Gestionar disponibilidad de graduados de noveno grado
Actores:	Funcionario (inicia)
Resumen:	El caso de uso comienza cuando el Funcionario selecciona la opción Gestionar disponibilidad de graduados de noveno grado. Selecciona la opción que desea realizar como: Adicionar, Modificar, Eliminar, Búsqueda Avanzada, Limpiar Búsqueda, Cargar Datos, Exportar o Año en Plan. Introduce los datos necesarios y finaliza cuando el Funcionario selecciona la opción cerrar.
Referencias	RF 1
Prioridad	Crítico

Tabla 4 Descripción breve del CU Gestionar disponibilidad de graduados de noveno grado

Caso de Uso:	Gestionar disponibilidad de graduados de enseñanza media
Actores:	Funcionario (inicia)
Resumen:	El caso de uso comienza cuando el Funcionario selecciona la opción Gestionar disponibilidad de graduados de enseñanza media. Selecciona la opción que desea realizar como: Adicionar, Modificar, Eliminar, Búsqueda Avanzada, Limpiar Búsqueda, Cargar Datos, Exportar o Distribución por año. Introduce los datos necesarios y finaliza cuando el Funcionario selecciona la opción cerrar.
Referencias	RF 2
Prioridad	Crítico

Tabla 5 Descripción breve del CU Gestionar disponibilidad de graduados de enseñanza media

Caso de Uso:	Gestionar capacidad docente de pre-universitarios provinciales
Actores:	Funcionario (inicia)
Resumen:	El caso de uso comienza cuando el Funcionario selecciona la opción Gestionar capacidad docente de pre-universitarios provinciales. Selecciona la opción que desea realizar como: Adicionar, Modificar, Eliminar, Búsqueda Avanzada, Limpiar Búsqueda, Exportar o Año en Plan. Introduce los datos necesarios y finaliza cuando el Funcionario selecciona la opción cerrar.
Referencias	RF 3
Prioridad	Crítico

Tabla 6 Descripción breve del CU Gestionar capacidad docente de preuniversitarios provinciales

Caso de Uso:	Gestionar capacidad docente disponible a centros universitarios
Actores:	Funcionario (inicia)
Resumen:	El caso de uso comienza cuando el Funcionario selecciona la opción Gestionar capacidad docente disponible a centros universitarios. Selecciona la opción que desea realizar como: Adicionar, Modificar, Eliminar, Búsqueda Avanzada, Limpiar Búsqueda, Exportar o Año en Plan. Introduce los datos necesarios y finaliza cuando el Funcionario selecciona la opción cerrar.

Referencias	RF 4
Prioridad	Crítico

Tabla 7 Descripción breve del CU Gestionar capacidad docente disponible a centros universitarios

Caso de Uso:	Gestionar demanda de técnicos medio y obreros calificados
Actores:	Funcionario (inicia)
Resumen:	El caso de uso comienza cuando el Funcionario selecciona la opción Gestionar demanda de técnicos medio y obreros calificados. Selecciona la opción que desea realizar como: Adicionar, Modificar, Eliminar, Búsqueda Avanzada, Limpiar Búsqueda, Cargar Datos, Exportar o Año en Plan. Introduce los datos necesarios y finaliza cuando el Funcionario selecciona la opción cerrar.
Referencias	RF 5
Prioridad	Crítico

Tabla 8 Descripción breve del CU Gestionar demanda de técnicos medio y obreros calificados

Caso de Uso:	Gestionar demanda de graduados de la enseñanza superior
Actores:	Funcionario (inicia)
Resumen:	El caso de uso comienza cuando el Funcionario selecciona la opción Gestionar demanda de graduados de la enseñanza superior. Selecciona la opción que desea realizar como: Adicionar, Modificar, Eliminar, Búsqueda Avanzada, Limpiar Búsqueda, Cargar Datos, Exportar o Año en Plan. Introduce los datos necesarios y finaliza cuando el Funcionario selecciona la opción cerrar.
Referencias	RF 6
Prioridad	Crítico

Tabla 9 Descripción breve del CU Gestionar demanda de graduados de la enseñanza superior

Caso de Uso:	Gestionar plan de ingreso de técnicos medio y obreros calificados
Actores:	Funcionario (inicia)
Resumen:	El caso de uso comienza cuando el Funcionario selecciona la opción Gestionar plan de ingreso de técnicos medio y obreros calificados.

	Selecciona la opción que desea realizar como: Adicionar, Modificar, Eliminar, Actualizar, Ejecutar Algoritmo o Exportar. Introduce los datos necesarios y finaliza cuando el Funcionario selecciona la opción cerrar.
Referencias	RF 8
Prioridad	Crítico

Tabla 10 Descripción breve del CU Gestionar plan de ingreso de técnicos medio y obreros calificados

Caso de Uso:	Gestionar plan de ingreso de la enseñanza superior
Actores:	Funcionario (inicia)
Resumen:	El caso de uso comienza cuando el Funcionario selecciona la opción Gestionar plan de ingreso de la enseñanza superior. Selecciona la opción que desea realizar como: Adicionar, Modificar, Eliminar, Búsqueda Avanzada, Limpiar Búsqueda, Ejecutar Algoritmo, Exportar o Distribución por año. Introduce los datos necesarios y finaliza cuando el Funcionario selecciona la opción cerrar.
Referencias	RF 7
Prioridad	Crítico

Tabla 11 Descripción breve del CU Gestionar plan de ingreso de la enseñanza superior

Caso de Uso:	Mostrar fuentes de ingreso sobrantes
Actores:	Funcionario (inicia)
Resumen:	El caso de uso comienza cuando el Funcionario selecciona la opción Visor de Reportes. Selecciona el reporte Mostrar fuentes de ingreso sobrantes para ver o imprimir. Introduce los datos necesarios y finaliza cuando el Funcionario selecciona la opción cerrar.
Referencias	RF 23
Prioridad	Secundario

Tabla 12 Descripción breve del CU Mostrar fuentes de ingreso sobrantes

Caso de Uso:	Mostrar carreras por provincia
Actores:	Funcionario (inicia)
Resumen:	El caso de uso comienza cuando el Funcionario selecciona la opción Visor

	de Reportes. Selecciona el reporte Mostrar carreras por provincia para ver o imprimir. Introduce los datos necesarios y finaliza cuando el Funcionario selecciona la opción cerrar.
Referencias	RF 14
Prioridad	Secundario

Tabla 13 Descripción breve del CU Mostrar carreras por provincia

En el Anexo No.1 se encuentran las descripciones de cada uno de los casos de uso representados en los diagramas anteriores.

Conclusiones

En este capítulo se abordó lo referente al modelo de dominio, sus objetivos y alcance, se entendió de una mejor forma el sistema a desarrollar con el apoyo de los diagramas, la definición del actor y los casos de uso que intervendrán.

Capítulo 3: Análisis y diseño del sistema

Introducción

En el presente capítulo se describe cómo debe ser el sistema, realizando su análisis y diseño. Se muestran los diagramas de clases del análisis, con las interfaces, controladoras, entidades y las relaciones entre las mismas. Del diseño se presentan los diagramas de clases y de interacción correspondientes a cada caso de uso del sistema, mientras que de la Base de Datos se expone el diagrama de entidad relación y la descripción de cada tabla.

Estudio de factibilidad

La estimación mediante el análisis de puntos de casos de uso es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de pesos a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. Mediante esta técnica existe una probabilidad de estimar el esfuerzo que el equipo de desarrollo empleará para elaborar el sistema, a partir de las características de sus requisitos, expresados en los casos de uso por lo que es decisión de la universidad por las características de los proyectos la utilización de esta técnica. A continuación, se realiza la estimación para la primera iteración del desarrollo donde se detallan los pasos a seguir para la aplicación de este método.

Paso 1. Cálculo de los Puntos de casos de uso Desajustados.

$$UUCP = UAW + UUCW$$

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)

Tipo de Actor	Descripción	Factor de peso	Actores	Total
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API Application Programming Interface)	1	2	2

Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto	2	0	0
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica	3	1	1

Tabla 14 Factor de peso de los actores sin ajustar

$$UAW = \sum \text{cantidad de actores} * \text{peso}$$

$$UAW = 2+3$$

$$UAW=5$$

Factor de Peso de los Casos de Uso sin ajustar (UUCW)

Tipo de Actor	Descripción	Factor de peso	Cantidad de CU	Total
Simple	El Caso de Uso contiene de 1 a 3 transacciones	5	7	35
Medio	El Caso de Uso contiene de 4 a 7 transacciones	10	0	0
Complejo	El Caso de Uso contiene más de 8 transacciones	15	5	75

Tabla 15 Factor de peso de los casos de uso sin ajustar

$$UUCW = \sum \text{cantidad de CU} * \text{Peso}$$

$$UUCW = 35 + 75 = 110$$

$$UUCP = 115$$

Paso 2. Cálculo de los Puntos de casos de uso ajustados

$$UCP = UUCP * TCF * EF$$

Donde:

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

Factor de complejidad técnica

El factor de complejidad técnica (TCF) se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada factor se cuantifica en un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido	2	4	8
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	4	4
T4	Funcionamiento interno complejo	1	2	2
T5	El código debe ser reutilizable	1	5	4
T6	Facilidad de instalación	0.5	3	1.5
T7	Facilidad de uso	0.5	3	1.5
T8	Portabilidad	2	3	6
T9	Facilidad de cambio	1	3	3
T10	Concurrencia	1	2	2
T11	Incluye objetivos especiales de seguridad	1	4	4
T12	Provee acceso directo a terceras partes	1	5	5
T13	Se requieren facilidades especiales de entrenamiento de usuario	1	2	2

Tabla 16 Factor de complejidad técnica

$$TCF = 0.6 + 0.01 * \sum (\text{peso} * \text{valor asignado})$$

$$TCF = 0.6 + 0.01 * (47)$$

$$TCF = 0.6 + 0.47$$

$$TCF = 1.07$$

Factor de ambiente (EF)

El factor de ambiente (EF) está relacionado con las habilidades y entrenamiento del grupo de desarrollo que realiza el sistema. Cada factor se cuantifica con un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1.5	4	6
E2	Experiencia en la aplicación	0.5	4	2
E3	Experiencia en la orientación a objetos	1	5	5
E4	Capacidad del analista líder	0.5	5	2.5
E5	Motivación	1	5	5
E6	Estabilidad de requerimientos	2	4	8
E7	Personal Part-Time	-1	1	-1
E8	Dificultad del lenguaje de programación	-1	3	-3

Tabla 17 Factor de ambiente

$$EF = 1.4 - 0.03 * \sum (\text{peso} * \text{valor asignado})$$

$$EF = 1.4 - 0.03 * (6 + 2 + 5 + 2.5 + 5 + 8 + -1 + -3)$$

$$EF = 1.4 - 0.735$$

$$EF = 0.665$$

$$UCP = UUCP * TCF * EF$$

$$UCP = 115 * 1.07 * 0.665$$

$$UCP = 81.82825$$

Paso 3. Estimación de esfuerzo a través de los puntos de caso de uso

$$E = UCP * CF$$

Donde:

E: Esfuerzo estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: Factor de conversión

Para obtener el factor de conversión (CF) se cuentan cuantos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por encima de la media para los restantes (E7, E8). Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre/Puntos de casos de uso. Si el total es 3 o 4 se utiliza el factor de conversión 28 Horas-Hombre/Puntos de casos de uso. Si el total es mayor o igual a 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

En este caso se puede decir que:

CF = 20 Horas-Hombre/ Puntos de casos de uso

$E = 81.82825 * 20$

$E = 1636.565$ Horas-Hombre

Paso 4. Calcular el esfuerzo de todo el proyecto

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	409.14125
Diseño	20	818.2825
Implementación	40	1636.565
Prueba	15	613.711875
Sobrecarga (otras actividades)	15	613.711875
Total	100	4091.4125

Tabla 18 Esfuerzo del proyecto

Si $E_T = 4091.4125$ Horas-Hombre y cada mes tiene como promedio 270 horas, eso daría un $E_T = 15.15$ mes-hombre.

Esto quiere decir que 2 personas pueden realizar el problema analizado en 7.575 meses.

Costo del proyecto

Se asume como salario promedio mensual \$50.00

$CHM = 1 * \text{salario promedio}$

CHM = 50.00 \$/mes

Costo = CHM * E_T

Costo = 50.00 * 15.15

Costo = 757.5

Beneficios tangibles e intangibles

El sistema automatizado para el Plan de Ingreso a la Educación Media y Superior del Sistema Unificado de Gestión de la Fuerza de Trabajo Calificada no es un producto con fines comerciales, su principal objetivo es resolver los problemas que existen durante el desarrollo de esta tarea en el departamento de Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación.

El beneficio fundamental del sistema es contar con una aplicación web flexible, dinámica y de interfaz agradable que le permita registrar, actualizar y conocer de una forma más precisa y en el menor tiempo posible datos de interés de los participantes en esta actividad.

Por lo tanto, los beneficios inmediatos son generalmente intangibles:

- ✓ Disminución del tiempo y esfuerzo que se invierte en esta tarea que se realiza, hasta ahora, prácticamente de forma manual.
- ✓ Fácil detección de problemas.
- ✓ Fácil y rápido acceso y publicación de la información actualizada.
- ✓ Fácil procesamiento de la información y obtención, dinámica, de reportes del plan de ingreso en cualquier momento.

Análisis de costos y beneficios

Desarrollar un producto informático cuesta. Justificar entonces su desarrollo depende de los beneficios que reportaría su implantación y utilización. Los beneficios pueden ser económicos o de orden social, estos últimos son de tanta importancia como los primeros. El sistema que se propone está dirigido fundamentalmente al departamento de Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación, por tanto su mayor beneficio es de orden social.

Una vez implantado el sistema, éste contribuirá a aumentar la eficiencia y a disminuir el tiempo de respuesta del sistema y obtener resultados con mayor rapidez y certeza.

La tecnología utilizada para el desarrollo del sistema es totalmente libre, por tanto no hay que incurrir en gastos en el pago de licencias de uso. El sistema es portable por lo que un cambio de plataforma

para la implantación del mismo es viable y factible, y no hay que incurrir en muchos cambios; debido a la estructuración en capas de los procesos del negocio que se diseñaron.




Analizando el costo del proyecto, los numerosos beneficios que reporta, detallados con anterioridad, se puede concluir que su implementación es realmente factible

Análisis

El flujo de trabajo análisis y diseño juega un papel fundamental en la fase de elaboración. El principal objetivo del análisis es comprender los requisitos funcionales con que cuenta el software para así estructurar el proyecto de forma clara y precisa, se profundiza en el dominio de la aplicación lo que permite una mayor comprensión del problema para modelar la solución (5).

Diagrama de clases del análisis

El diagrama de clases es un diagrama estático que describe la estructura de un sistema mostrando sus, clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y el diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargan del funcionamiento y la relación entre uno y otro (5).

Nombre	Característica	Figura
Entidad	Modelan información que posee larga vida y que es a menudo persistente.	 Entidad
Interfaz	Modelan la interacción entre el sistema y sus actores.	 Interfaz
Control	Coordinan la realización de uno o unos pocos casos de uso, coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.	 Control

A continuación se expone uno de los diagramas de clases del análisis de los caso de uso más críticos del sistema.

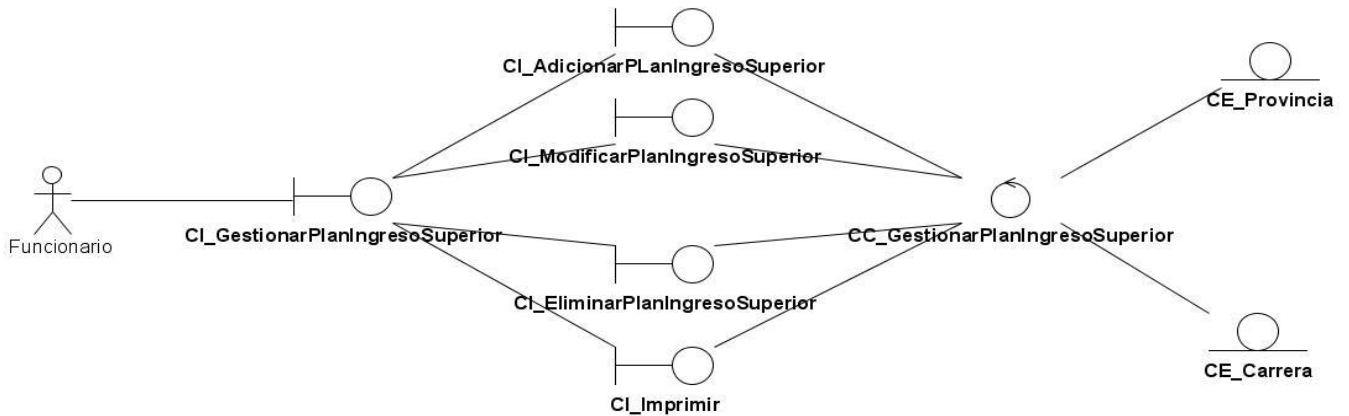


Figura 7 DCA: Gestionar Plan de Ingreso de la Enseñanza Superior.

Diagramas de colaboración del análisis

Los diagramas de interacción representan una vista dinámica del sistema y se pueden clasificar en dos tipos, diagramas de colaboración y diagramas de secuencia. Un diagrama de interacción representa la secuencia de acciones que ocurren desde que el actor comienza el caso de uso, así como los mensajes que se envían desde cada una de las clases. En el análisis se usan los diagramas de colaboración, ya que el objetivo principal es identificar las funcionalidades de cada objeto y las responsabilidades sobre ellos.

A continuación se exponen los diagramas de colaboración del análisis de uno de los diagramas más críticos del sistema.

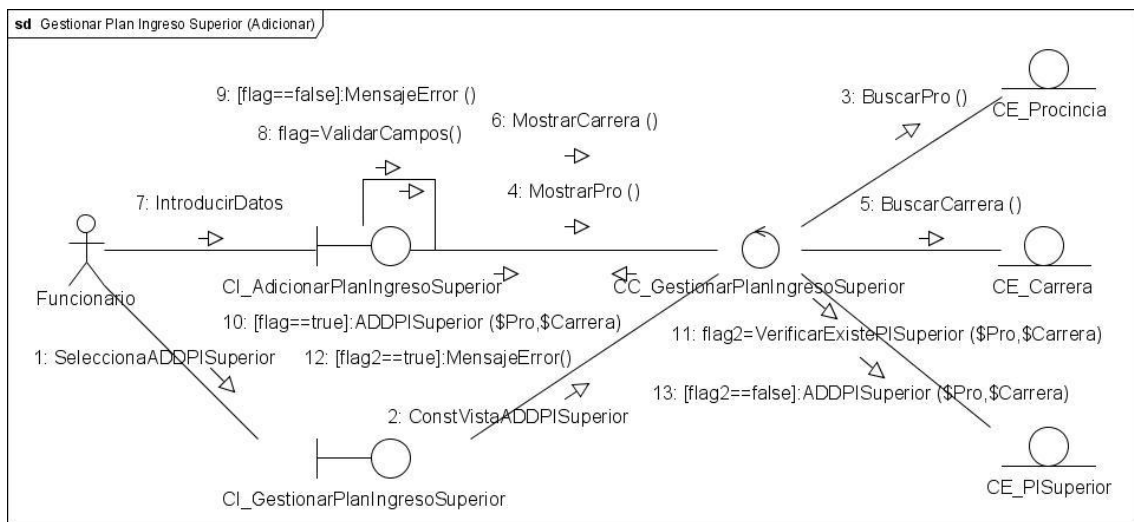


Figura 8 DC: Gestionar plan de ingreso de la enseñanza superior (Adicionar)

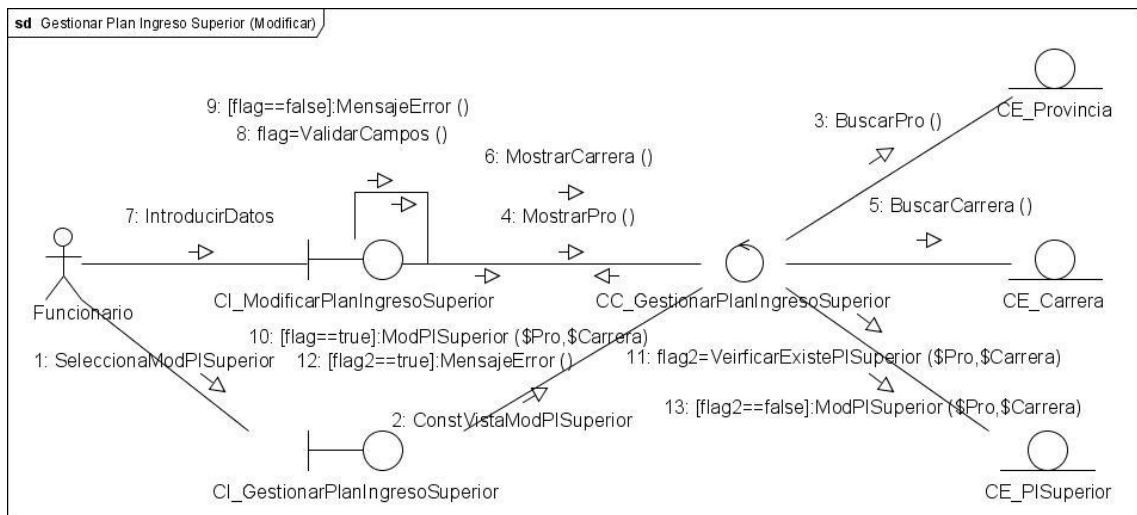


Figura 9 DC: Gestionar plan de ingreso de la enseñanza superior (Modificar)

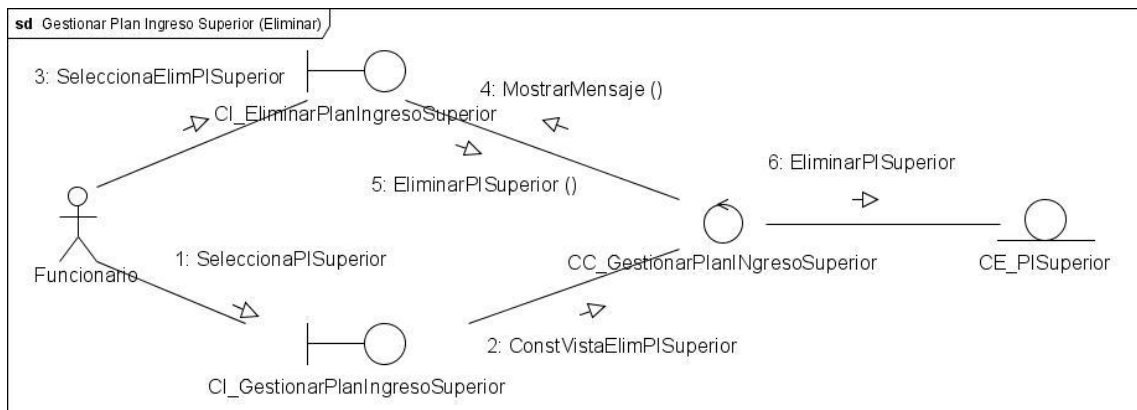


Figura 10 DC: Gestionar plan de ingreso de la enseñanza superior (Eliminar)

Los demás diagramas de colaboración del análisis se pueden encontrar en el Anexo 3.

Diseño

En la fase de diseño se modela el sistema de manera que soporte todos los requisitos, tanto funcionales como no funcionales, creándose así una entrada apropiada para las actividades de implementación.

El propósito del diseño es especificar una solución que trabaje y pueda ser fácilmente convertida en código fuente y construir una arquitectura simple y fácilmente extensible (5).

Propósitos del diseño

- ✓ Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario.
- ✓ Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces o clases.
- ✓ Descomponer los trabajos de implementación en partes más manejables que pueden ser llevadas a cabo por diferentes equipos de desarrollo.
- ✓ Capturar las interfaces entre los subsistemas antes en el ciclo de vida del software, lo cual es muy útil cuando se utiliza interfaces como elementos de sincronización entre diferentes equipos de desarrollo.

Arquitectura y patrones de diseño utilizados

La arquitectura de software es quien le da forma al software para que soporte todos los requisitos. Establece todos los fundamentos para que analistas, diseñadores, programadores, etc. trabajen en una línea común que permita alcanzar los objetivos y necesidades del sistema.

Se necesita una arquitectura robusta, que guíe el proceso de desarrollo y que defina de manera abstracta los componentes que lleven a cabo alguna tarea, sus interfaces y la comunicación entre ellos.

La arquitectura seleccionada fue Modelo Vista Controlador. Esta se ha vuelto muy común a la hora de construir software de gestión debido a las facilidades que brinda, pues permite la reutilización y la independencia entre las capas, se pueden realizar cambios en capas sin tener que modificar las otras, facilita la estandarización, la utilización de los recursos y la administración.

Las capas definidas son:

- ✓ **Vista o Capa de Presentación:** Tiene que ver con las interfaces y la interacción con los usuarios. Es la encargada de contener los controles de usuarios que son usados para la realización de los distintos casos de uso y de mantener todas las ventanas y menús que permiten el intercambio de información con la aplicación.
- ✓ **Controlador o Capa de Negocio:** Ubica los subsistemas y clases encargadas de realizar la mayoría de las funcionalidades que pide el cliente, así como los aspectos del software que automatizan o apoyan los procesos de negocio que llevan a cabo los usuarios, como tareas, reglas y restricciones.

- ✓ **Modelo o Capa de Acceso a Datos:** Contiene las clases que interactúan con la base de datos y que permiten realizar las distintas operaciones sobre ella, como son: procedimientos almacenados, consultas.

Los patrones de diseño son soluciones a problemas ya conocidos que ayudan a un mejor rendimiento, desarrollo y mantenimiento del software donde se apliquen, contribuyen a la realización de una arquitectura más factible, simple y entendible. Ayudan a diseñar correctamente en menos tiempo, a construir problemas reutilizables y facilitan la documentación.

Patrones GRASP utilizados

Experto: Este patrón propone como solución asignar la responsabilidad a la clase que cuenta con la información necesaria para cumplir la responsabilidad. 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 el Doctrine serán las encargadas del acceso a datos pues contienen y representan los datos que manejará el sistema, permitiendo que se conserve el encapsulamiento, soportando un bajo acoplamiento y una alta cohesión.

Bajo Acoplamiento: El acoplamiento mide la fuerza con que una clase está conectada a otra, de esta forma una clase con bajo acoplamiento debe tener un número mínimo de dependencia con otras clases. Las diferentes clases controladoras sólo dependen de un único controlador frontal para realizar sus funcionalidades. Además, una clase de acceso a datos solo depende de la clase controladora que la usa, así que pueden realizar cambios en cada clase de forma independiente. Ofrece como beneficio que 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 se tiene en cuenta para realizar un diseño que evite contener clases con un alto grado de abstracción, que asuman responsabilidades que podían haber delegado a otros objetos o que tengan responsabilidades muy complejas. Se diseñaron las clases de forma tal que contengan las mínimas responsabilidades necesarias y colaboren con otras para llevar a cabo una tarea. Se tienen las clases controladoras que se encargan de ejecutar acciones de acuerdo a las peticiones que le llegan y las clases de acceso a datos que interactúan con el modelo, de forma tal que se elimina la sobrecarga de funcionalidades en las clases controladoras. El utilizar Zend Framework permite crear y trabajar con clases con una alta cohesión, ya que posibilita organizar el trabajo de acuerdo a la estructura del proyecto. Este patrón permitirá tener las clases fáciles de mantener, entender y reutilizar.

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.

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 solo 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, 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.

Patrones GOF utilizados

Decorator: Este patrón permite añadir funcionalidad a una clase dinámicamente. Zend Framework implementa dicho patrón a la clase Zend_View, encargada de asignarle responsabilidades a objetos de manera dinámica y configurarlos con nuevos atributos. Se evidencia este patrón en la aplicación con el uso de una plantilla global de la aplicación dentro de la cual se van insertando las vistas de cada una de las funcionalidades.

Singleton: 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 disponible mediante este patrón para lograr una vía de entrada única a las solicitudes.

Facade: 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, solo 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.

Factory: 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. El uso

de Zend Framework permitió el uso de este patrón, ya que las clases de este marco de trabajo fueron programadas haciendo uso de este patrón, y se evidencia en la creación de los controladores (frontends y backends) a través de una clase Zend_Cache, la creación de las vistas, por mencionar algunas.

Patrones del modelo utilizados

Active Record: Este patrón representa de forma Orientada a Objetos los datos de una Base de Datos Relacional, definiendo interfaces sencillas para acceder y manipular esos datos. Es un enfoque al problema de acceder a los datos de una base de datos. Una fila en la tabla de la base de datos (o vista) se envuelve en una clase, de manera que se asocian filas únicas de la base de datos con objetos del lenguaje de programación usado. Cuando se crea uno de estos objetos, se añade una fila a la tabla de la base de datos. Cuando se modifican los atributos del objeto, se actualiza la fila de la base de datos. Doctrine usa este patrón para manejar la base de datos.

Row Data Gateway: Este patrón asume el comportamiento de un objeto que actúa como puerta de enlace a una fila de una tabla de la base de datos. Tiene propiedades que reflejan las columnas de la tabla, y métodos de actualización en la base. Es implementado en la aplicación con el uso de Doctrine, que al realizar el mapeo de la base de datos, crea objetos para cada una de las tablas que simulan el comportamiento de una fila en la tabla (Ejemplo: la clase TbPiDemandaCrd crea objetos que simulan el comportamiento de una fila en la tabla tb_pi_demanda_crd).

Table Data Gateway: Similar a Row Data Gateway, pero a diferencia de este define la estructura de acceso por registros en las entidades de la base de datos, específica su acceso a nivel de tabla, proponiendo un objeto que se comporte como puerta de enlace a cada tabla de la base de datos. Contiene una interfaz que permite actualizar, buscar, borrar e insertar en la tabla y puede retornar un registro, un grupo de registro y hasta un objeto del dominio. Es implementado en la aplicación con la utilización de Doctrine, que al realizar el mapeo de la base de datos, crea objetos que se encargan de realizar las operaciones sobre una tabla o vista determinada (Ejemplo: la clase TbPiDemandaCrdModel es la que se encarga de realizar las operaciones sobre la tabla tb_pi_demanda_crd del modelo de datos).

Modelo de diseño

En el flujo de trabajo de Análisis y Diseño, definido por RUP, se describe cómo se debe implementar el sistema, esto como resultado de un análisis realizado a los requerimientos no funcionales del mismo. El diseño no debe tener ambigüedades para que el modelo final que se obtenga sea suficiente para la

implementación. El modelo de diseño es el resultado más importante de este flujo de trabajo, el cual consiste en colaboraciones de clases, las que pueden ser agrupadas en paquetes y subsistemas, además describe la realización de los casos de uso.

Realización de los diagramas de clases del diseño

Una realización de casos de uso del diseño es una colaboración en el modelo de diseño que describe cómo se realiza un caso de uso específico, y como se ejecuta, términos de las clases de diseño y sus objetivos.

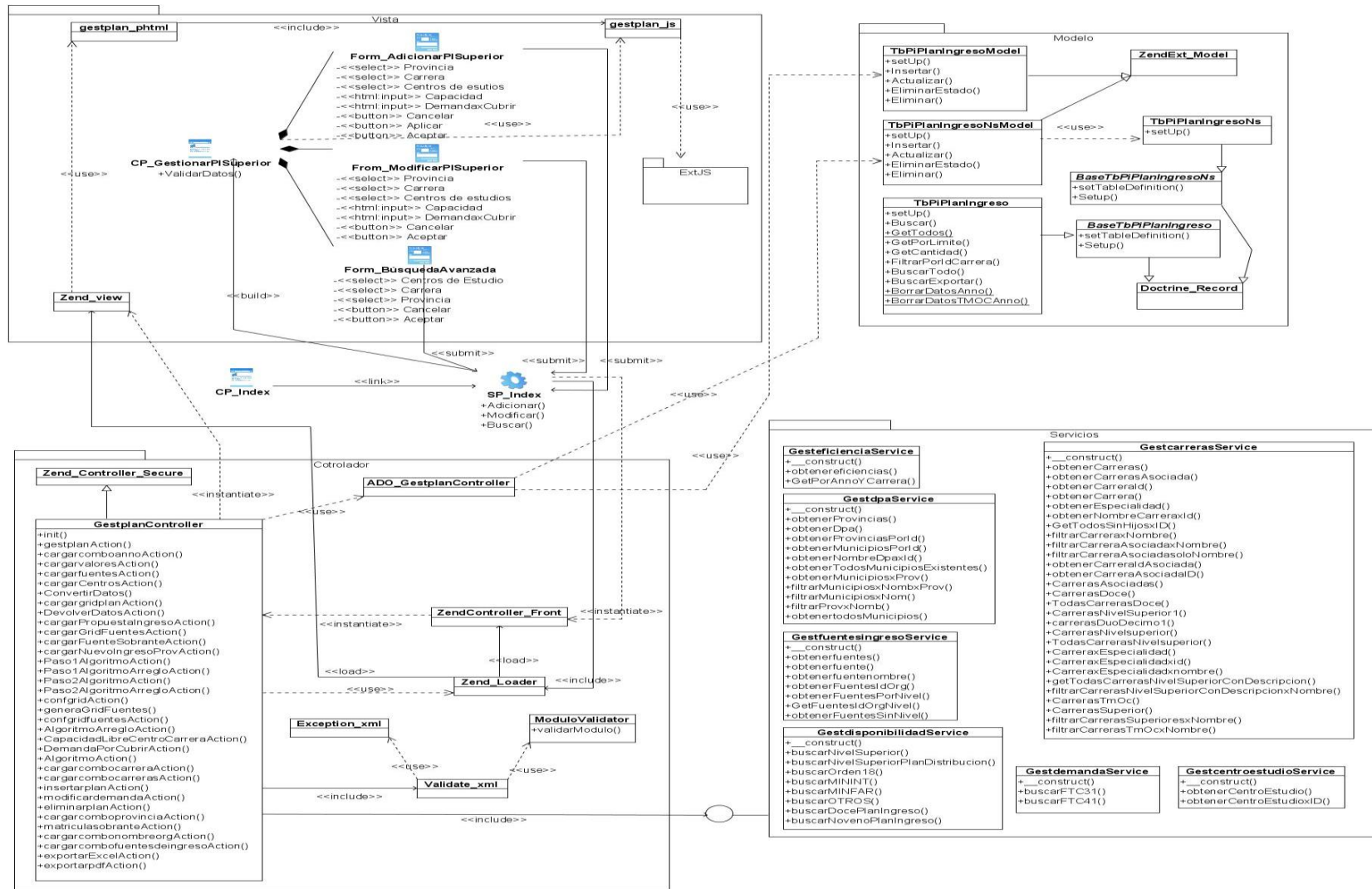


Figura 11 DCD: Gestionar Plan de Ingreso de la Enseñanza Superior

Modelo de datos

El modelo de datos describe la representación lógica y física de los datos persistentes usados por la aplicación. El modelo lógico se puede apreciar en el Anexo # 6.

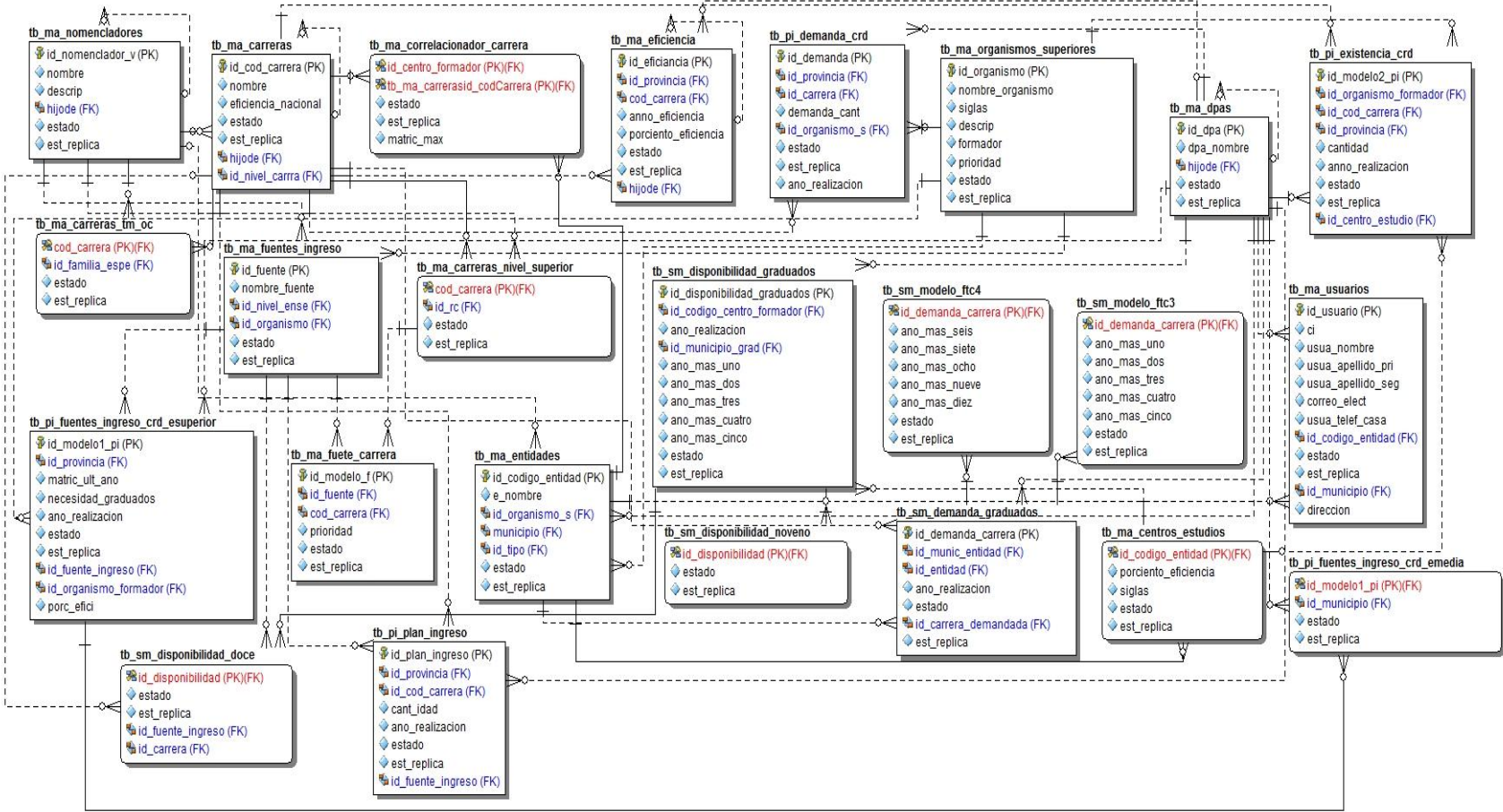


Figura 12 Modelo Físico de la Base de Datos

Diseño de la base de datos

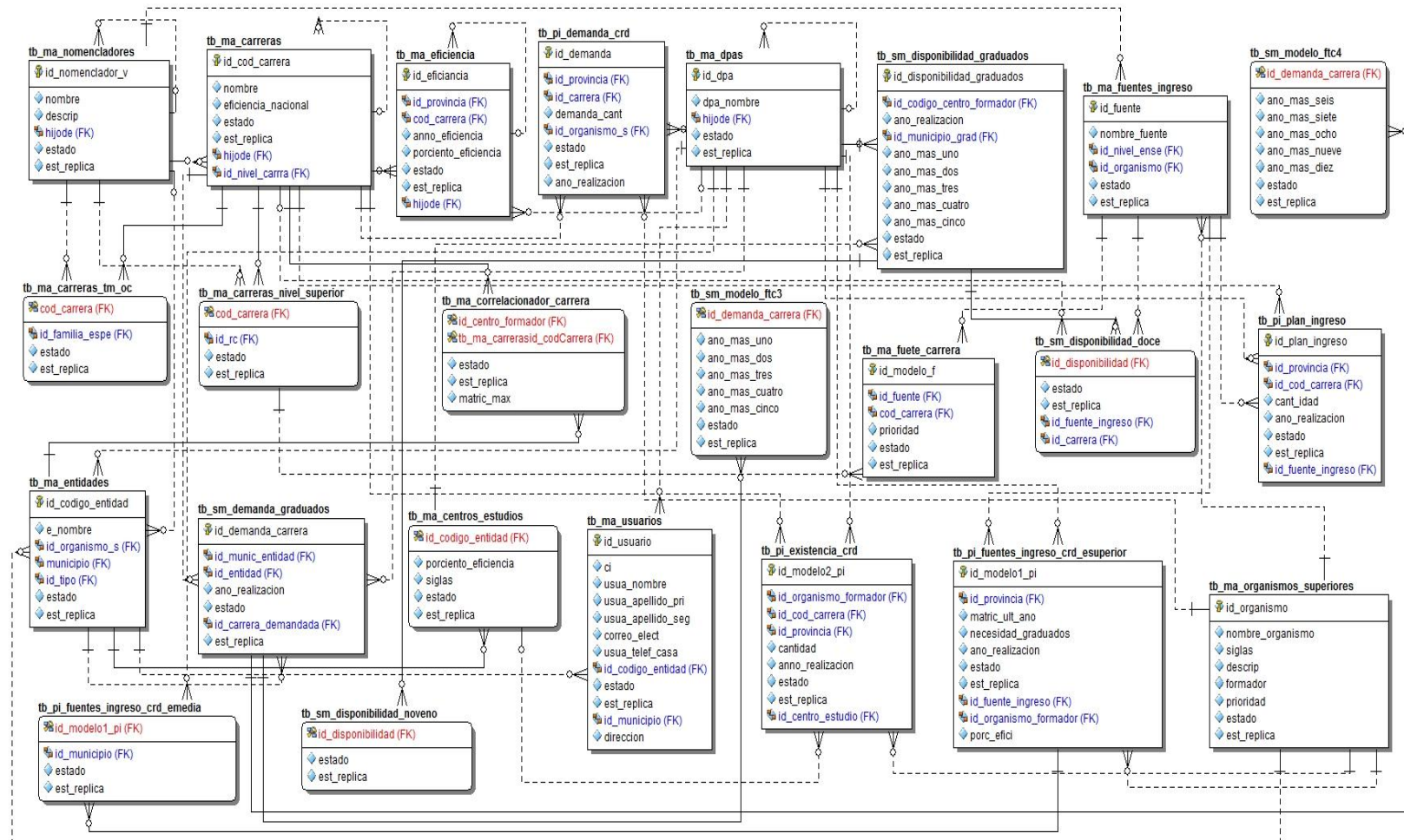


Figura 13 Modelo Lógico de Datos

Tratamiento de errores

En el sistema propuesto se evitan, minimizan y tratan los posibles errores, con el fin de garantizar la integridad y confiabilidad de la información. Para el tratamiento de errores se validan los datos que son introducidos por el usuario al sistema. Se valida que el formato de los datos sea el esperado y que no se omita información de importancia para el procesamiento del plan de ingreso a la educación media y superior. En estos casos se informa al usuario mediante un mensaje que ha realizado alguna operación incorrecta. Se muestran mensajes de advertencia para confirmar las acciones irreversibles, previendo la pérdida no deseada de datos. Cada usuario tendrá acceso restringido según su rol y se le permitirá solo ver las funcionalidades para las cuales obtenga permisos.

Seguridad

Para garantizar la seguridad de la información se creó un módulo de administración el cual se encarga de definir roles de usuarios, que pueden ser: funcionario MEP, administrador. El administrador es el encargado de los cambios y del buen funcionamiento del sistema, por tanto poseerá control total del mismo. El resto de los usuarios solo tendrá acceso a la información determinada por el administrador del sistema según el rol que posea.

Conclusiones

En este capítulo se abordó todo lo referente al flujo de Análisis y Diseño, la arquitectura y los patrones que serán utilizados en el desarrollo del sistema, se realizó el modelo de datos y se entendió de una mejor forma el sistema que será realizado.

Se describió el estudio de factibilidad realizado correspondiente al sistema propuesto, utilizando la técnica de estimación por Puntos de Caso de Uso, la cual resulta muy efectiva para estimar el esfuerzo requerido en el desarrollo de los primeros Casos de Uso de un sistema, si se sigue una aproximación iterativa como el Proceso Unificado de Rational. En este tipo de aproximación, los primeros Casos de Uso a desarrollar son los que ejercitan la mayor parte de la arquitectura de software y los que a su vez ayudan a mitigar los riesgos más significativos.

Teniendo en cuenta el costo estimado y los beneficios que reportará al ser implantado dicha proposición, al contribuir a mejorar el proceso de gestión del plan de ingreso a la educación media y superior se llega a la conclusión de que es factible implementar la herramienta propuesta.

Capítulo 4: Implementación y prueba

Introducción

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

Implementación

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

Propósitos del flujo de implementación:

- ✓ Definir una organización del sistema en términos de Subsistemas de Implementación organizados en capas.
- ✓ Implementar los elementos de diseño en términos de “Elementos de Implementación” (ficheros fuente, binarios, ejecutables y otros).
- ✓ Probar los componentes desarrollados independientemente como unidades.
- ✓ Integrar los resultados producidos por desarrolladores independientes o equipos en un sistema ejecutable (5).

Estándares de implementación

En la vista

Para esta capa se definen como estándares los siguientes:

- ✓ El nombre de las clases debe comenzar con gest en los casos de uso gestionar, seguido por el nombre del caso de uso. Ejemplo: gestplan, gestdemanda.
- ✓ El nombre de la clase debe comenzar con gen en los casos de uso generar, seguido por el nombre del caso de uso. Ejemplo: gennomencladores.
- ✓ Teniendo en cuenta el componente que se utilizará, el nombre del mismo estará definido de la siguiente forma:
 - Los botones se nombrarán: btn+Nombre. Ejemplo: btnAdicionar.
 - Los almacenes de datos (*stores*) se nombrarán: st+Nombre. Ejemplo: stPlan.
 - Los combos se nombrarán: cmb+Nombre. Ejemplo: cmbProvincia.

- Los campos de texto (*textfield*) se nombrarán: txf+Nombre. Ejemplo: txfbuscarProvincia
- Los paneles de tablas (*gridpanels*) se nombrarán: gp+Nombre. Ejemplo: gpPlan.
- Las ventanas se nombrarán: win+Nombre. Ejemplo: winAdd.

En el negocio

Para esta capa se definen:

- ✓ El nombre de las clases comenzará con Gest seguida por el nombre en los casos de uso de gestionar, finalizando con la palabra Controller. Ejemplo: GestplanController.
- ✓ El nombre de las clases comenzará con Gen seguida por el nombre en los casos de uso de generar, finalizando con la palabra Controller. Ejemplo: GennomencladoresController.
- ✓ El nombre de los atributos será en letra minúscula y en caso de ser más de una palabra se escribirá sin separación. Ejemplo: demandacant.
- ✓ El nombre de las funciones se formará por el nombre de la función seguido de la palabra Action. Ejemplo: insertarDemandaAction.
- ✓ Los servicios se nombrarán Gest seguido del nombre, finalizando con la palabra Service. Ejemplo: GestcarrerasService.
- ✓ Las validaciones se nombrarán Gest seguido del nombre, finalizando con la palabra Validator. Ejemplo: GestnovenoValidator.

En el modelo

Para esta capa se utilizan:

- ✓ El nombre de las entidades se define por tb seguido de las siglas del módulo y el nombre, separando cada una de las palabras por un “_”. Ejemplo: tb_pi_demanda_crd.
- ✓ El nombre de los atributos será en letra minúscula y en caso de ser más de una palabra serán separadas por un “_”. Ejemplo: id_modelo1_pi.

Diagrama 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, despegable y reemplazable que encapsula implementación y un conjunto de interfaces y proporciona la realización de los mismos. Estos diagramas son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos. Dicho diagrama contiene los nodos que forman la topología hardware sobre la que se ejecuta el sistema y la distribución de las partes del sistema en ellos (8).

A continuación se expone uno de los diagramas de componentes de los caso de uso más significativos del sistema. Los demás se pueden ver en el Anexo # 7.

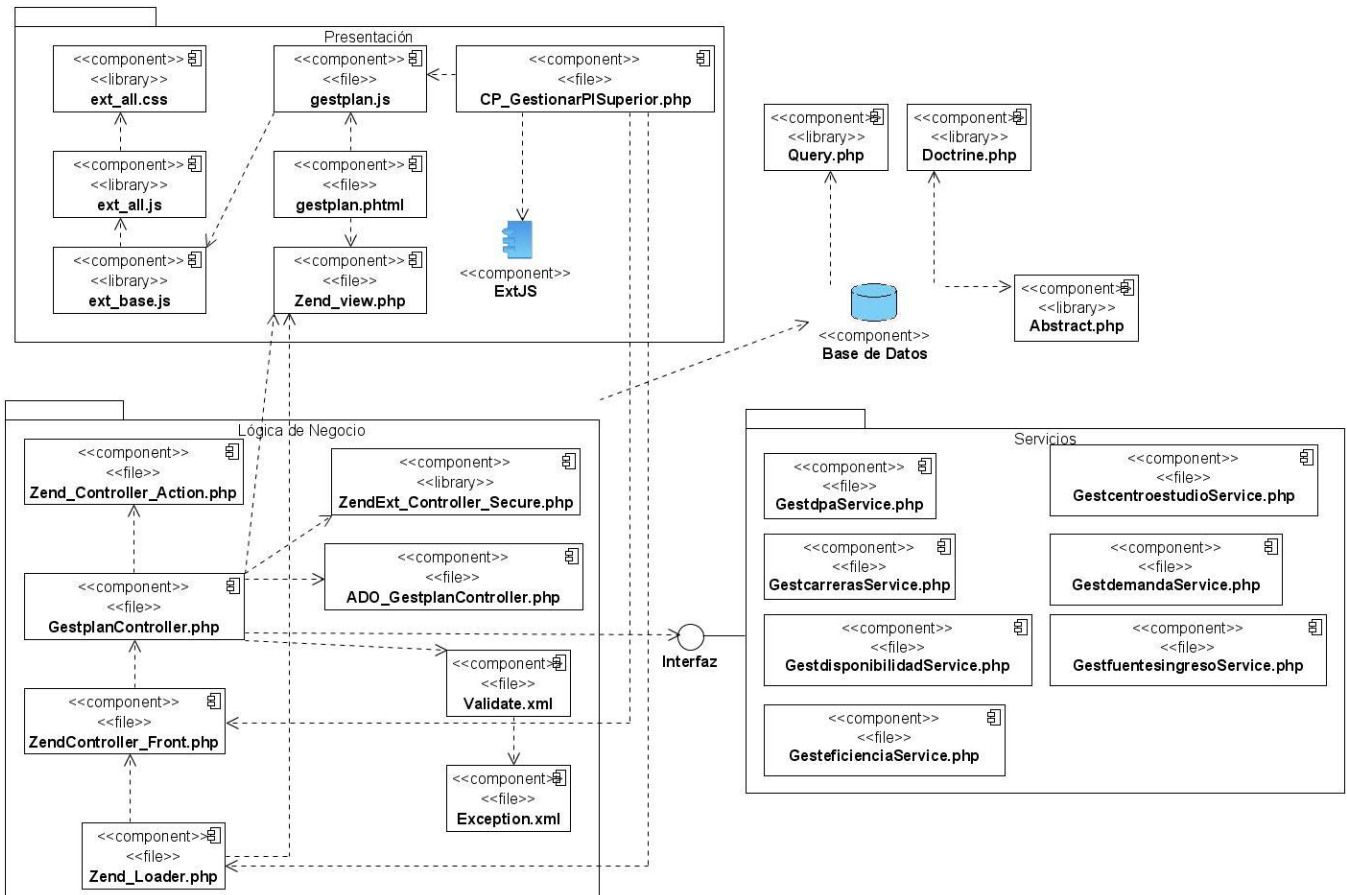


Figura 14 Diagrama de Componentes CU Gestionar plan de ingreso a la educación superior

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 (8).

En el caso del sistema a desarrollar, necesitará para su despliegue de una computadora cliente, un servidor web al que se conectará mediante el protocolo http y el cual estará conectado por TCP/IP con el servidor de base de datos de la aplicación, y una impresora conectada a la computadora cliente por el puerto USB.

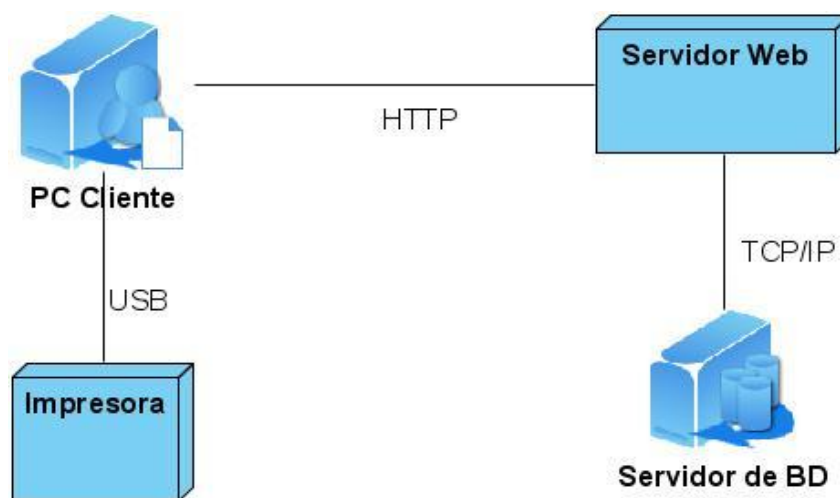


Figura 15 Diagrama de despliegue del sistema

Prueba

En el desarrollo de un software específico, el proceso de prueba es clave a la hora de detectar errores o fallas. Conceptos como estabilidad, escalabilidad, eficiencia y seguridad se relacionan con la calidad de un producto bien desarrollado. 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 un software más confiable. Los objetivos principales de la realización de una prueba son tener un buen caso de prueba que sirva para descubrir un error no descubierto antes.

Métodos de prueba

Los métodos de Prueba de Software tienen como objetivo diseñar pruebas que descubran diferentes tipos de errores en el menor tiempo y esfuerzo posible.

Existen dos métodos de prueba:

Pruebas de caja negra: El sistema de pruebas de caja negra no considera la codificación dentro de sus parámetros a evaluar, es decir, no están basadas en el conocimiento del diseño interno del programa. Estas pruebas se enfocan en los requerimientos establecidos y en la funcionalidad del sistema.

Según (Pressman, 2002) las pruebas de Caja Negra intentan encontrar los siguientes errores:

- ✓ Funciones incorrectas o ausentes.
- ✓ Errores de interfaz.
- ✓ Errores de estructura de datos o accesos a base de datos externas.

- ✓ Errores de rendimiento.
- ✓ Errores de inicialización y de terminación.

Algunos de los métodos empleados en las pruebas de Caja Negra son los siguientes:

- ✓ Métodos de prueba basados en grafos.
- ✓ Partición equivalente.
- ✓ Análisis de valores límite.
- ✓ Prueba de la tabla ortogonal.
- ✓ Adivinando el error.

Pruebas de caja blanca: Al contrario de las de caja negra, estas se basan en el conocimiento de la lógica interna del código del sistema. Las pruebas completan los distintos caminos que se pueden generar gracias a las estructuras condicionales, a los distintos estados del mismo, entre otros factores.

Para la realización de estas pruebas existen diferentes métodos los cuales son:

- ✓ Prueba del camino básico.
- ✓ Prueba de condición.
- ✓ Prueba del flujo de datos.
- ✓ Prueba de bucles.

Pruebas del sistema

Las pruebas se escogerán atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa se ejecute correctamente. Criterios mínimos que guiarán el resultado de las pruebas como son:

Valores fáciles: El programa se depurará con datos fáciles de comprobar.

Valores típicos realistas: Se ensayará un programa con datos seleccionados para que representen cómo se aplicará. Estos datos han de ser suficientemente sencillos, de modo que los resultados sean verificables de forma manual, valores futuros candidatos a ser usados en la aplicación.

Valores extremos: Muchos programas cometen errores en los límites de sus rangos de aplicaciones.

Valores ilegales: Se entrarán datos cuyos valores no corresponden con los requerimientos por el sistema, su reacción inmediata habrá de ser por lo menos un mensaje de error adecuado para el usuario.

El tipo de prueba seleccionada fue pruebas de caja negra y dentro del mismo el método de particiones equivalentes. Estas pruebas son realizadas una vez concluido el software y luego de que todos los

componentes de software y hardware han sido integrados. El método de particiones equivalentes principalmente se centra en la definición de casos de prueba que descubre las clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

El diseño de caso de prueba para la partición equivalente se basa en la evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada.

Regularmente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica (9).

A continuación se muestra un caso de prueba de los casos de uso significativos:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC1: Adicionar	EC 1.1: Adicionar una disponibilidad correctamente pulsando el botón Aplicar.	Se adiciona una nueva disponibilidad al sistema.	<ul style="list-style-type: none"> -Se selecciona la opción "Adicionar". - Llenar todos los campos. - Se selecciona "Aplicar". - Muestra un mensaje informando que se ha insertado satisfactoriamente, mantiene el organismo y la provincia, limpia el resto de los campos dando la opción de adicionar nuevamente. - Se selecciona "Aceptar".
	EC 1.2: Adicionar una disponibilidad correctamente pulsando el botón Aceptar.	Se adiciona una nueva disponibilidad al sistema.	<ul style="list-style-type: none"> -Se selecciona la opción "Adicionar". - Llenar todos los campos. - Se selecciona "Aceptar". - Muestra un mensaje informando que se ha insertado satisfactoriamente.

			- Se selecciona "Aceptar".
	<i>EC 1.3: Adicionar una disponibilidad dejando campos en blanco.</i>	<i>No se adiciona una nueva disponibilidad al sistema.</i>	-Se selecciona la opción "Adicionar". - Llenar los campos. - Se selecciona "Aplicar" o "Aceptar". - Muestra un mensaje de error, notificando al usuario que los datos son incorrectos. -Se selecciona "Aceptar".
	<i>EC 1.4: Adicionar una disponibilidad pasando datos existentes.</i>	<i>No se adiciona una nueva disponibilidad al sistema.</i>	-Se selecciona la opción "Adicionar". - Llenar todos los campos. - Se selecciona "Aplicar" o "Aceptar". -Muestra un mensaje indicando que la disponibilidad ya existe. -Se selecciona "Aceptar".
	<i>EC 1.5: Cancelar la operación.</i>	<i>No se adiciona una nueva disponibilidad al sistema.</i>	-Se selecciona la opción "Adicionar". - Llenar los campos. - Se selecciona "Cancelar". - Se cierra la ventana para adicionar.
<i>SC2: Modificar</i>	<i>EC 2.1: Modificar una disponibilidad</i>	<i>Se modifica una disponibilidad al</i>	-Se selecciona una disponibilidad.

	<i>correctamente.</i>	<i>sistema.</i>	<ul style="list-style-type: none"> -Se selecciona la opción "Modificar". - Se cambian los datos deseados. - Se selecciona "Aceptar". - Muestra el mensaje "La disponibilidad fue modificada satisfactoriamente". - Se selecciona "Aceptar".
	<i>EC 2.2: Modificar disponibilidad pasando datos existentes.</i>	<i>No se modifica la disponibilidad al sistema.</i>	<ul style="list-style-type: none"> -Se selecciona una de las disponibilidades existentes. -Se selecciona la opción "Modificar" -Se selecciona "Aceptar". -Se muestra un mensaje, indicando que no se ha modificado. -Se selecciona "Aceptar".
	<i>EC 2.3: Cancelar operación.</i>	<i>No se modifica la disponibilidad al sistema.</i>	<ul style="list-style-type: none"> -Se escoge una disponibilidad. -Se selecciona la opción "Modificar". - Se cambian los datos deseados. - Se selecciona "Cancelar". - Se cierra la ventana para modificar.
<i>SC3: Eliminar</i>	<i>EC 3.1: Eliminar disponibilidad</i>	<i>Se elimina una disponibilidad al</i>	<ul style="list-style-type: none"> -Se selecciona una disponibilidad.

	<i>correctamente.</i>	<i>sistema.</i>	<p>-Se selecciona la opción “Eliminar”.</p> <p>-Muestra el mensaje “Seguro que desea eliminar la disponibilidad seleccionada”.</p> <p>-Se selecciona la opción “Aceptar”</p> <p>-Muestra el mensaje “La disponibilidad seleccionada fue eliminada satisfactoriamente”.</p> <p>-Se selecciona “Aceptar”.</p>
	<i>EC 3.2: Cancelar la operación.</i>	<i>No se elimina una disponibilidad al sistema.</i>	<p>-Se selecciona una disponibilidad.</p> <p>-Se selecciona la opción “Eliminar”.</p> <p>-Se selecciona la opción “Cancelar”.</p> <p>- Se cierra la ventana para eliminar.</p>
<i>SC4: Buscar</i>	<i>EC 4.1: Buscar disponibilidad correctamente.</i>	<i>Se busca una disponibilidad en el sistema.</i>	<p>-Se selecciona la opción “Buscar”.</p> <p>-Se selecciona el(los) criterio(s) de búsqueda deseado(s).</p> <p>- Se selecciona “Aceptar”.</p> <p>- Muestra todas las disponibilidades existentes del año seleccionado, correspondiente a los datos</p>

			seleccionados.
	<i>EC 4.2: Buscar disponibilidad sin pasar el criterio de búsqueda.</i>	<i>No se busca una disponibilidad en el sistema.</i>	<ul style="list-style-type: none"> - Se selecciona la opción "Buscar". - Se selecciona "Aceptar". - Muestra un mensaje indicando que debe seleccionar al menos un campo. - Se selecciona "Aceptar".
	<i>EC 4.3: Buscar sin existir disponibilidades correspondientes.</i>	<i>No se busca una disponibilidad en el sistema.</i>	<ul style="list-style-type: none"> - Se selecciona la opción "Buscar". - Se selecciona el(los) criterio(s) de búsqueda deseado(s). - Muestra un mensaje notificando que no hay resultados para mostrar. - Se selecciona "Aceptar".
	<i>EC 4.4: Cancelar la operación.</i>	<i>No se busca una disponibilidad en el sistema.</i>	<ul style="list-style-type: none"> - Se selecciona la opción "Buscar". - Se seleccionan los campos por lo que desea buscar. - Se selecciona "Cancelar". - Se cierra la ventana para buscar.
	<i>EC 4.5: Limpiar Búsqueda.</i>	<i>Se limpian los datos introducidos para la búsqueda</i>	<ul style="list-style-type: none"> - Se selecciona la opción "Buscar". - Se selecciona el(los)

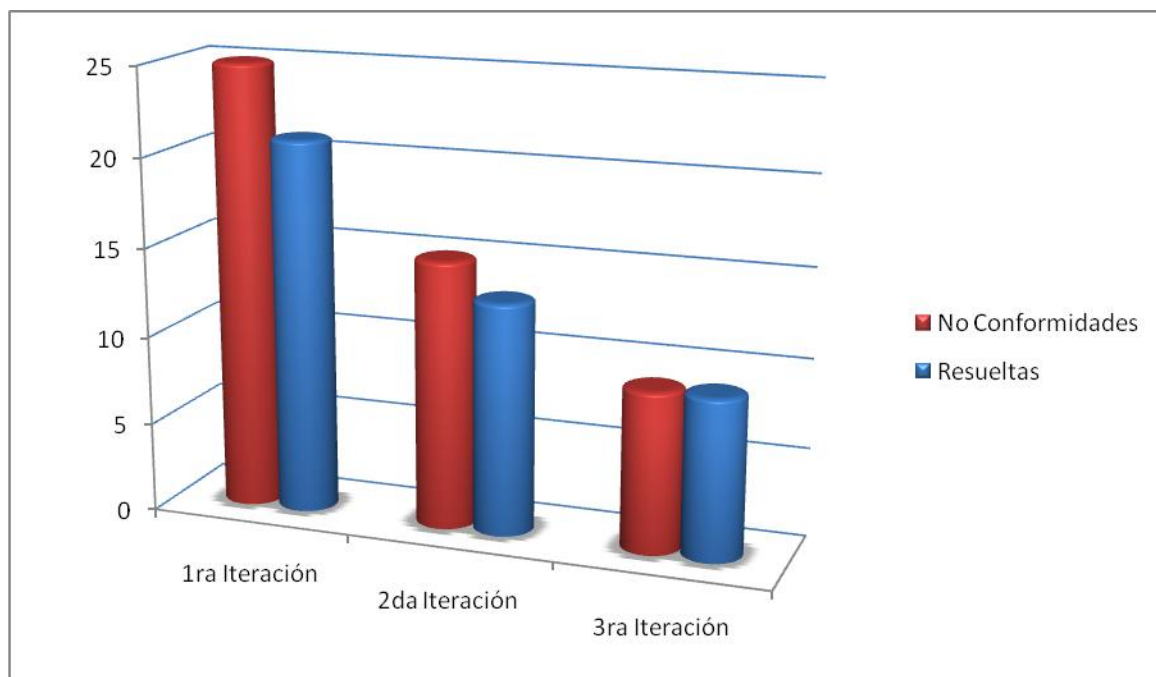
		<i>actual.</i>	<p><i>criterio(s) de búsqueda deseado(s).</i></p> <ul style="list-style-type: none"> - <i>Se selecciona “Aceptar”.</i> - <i>Muestra todas las disponibilidades existentes del año seleccionado, correspondiente a los datos seleccionados.</i> - <i>Se selecciona la opción “Limpiar búsqueda”.</i> - <i>Vuelve a la ventana principal para Gestionar Disponibilidad de Graduados de Nivel Medio mostrando todas las disponibilidades existentes.</i>
<i>SC 5: Exportar</i>	<i>EC 5.1: Exportar a PDF.</i>	<i>Se crea un documento PDF.</i>	<ul style="list-style-type: none"> - <i>Se selecciona la opción “Exportar”.</i> - <i>Se selecciona el tipo de documento PDF.</i> - <i>Se selecciona “Aceptar”.</i> - <i>Crea un documento PDF con los datos seleccionados por el funcionario, teniendo habilitadas todas las opciones que brinda el PDF.</i>
	<i>EC 5.2: Exportar a Excel.</i>	<i>Se crea un documento Excel.</i>	<ul style="list-style-type: none"> - <i>Se selecciona la opción “Exportar”.</i> - <i>Se selecciona el tipo de documento Excel.</i>

			<ul style="list-style-type: none"> -Se selecciona "Aceptar". -Crea un documento Excel con los datos seleccionados por el funcionario, teniendo habilitadas todas las opciones que brinda el Excel.
	EC 5.3: Cancelar la operación.	No se exporta la información.	<ul style="list-style-type: none"> -Se selecciona la opción "Exportar". -Se selecciona el tipo de documento a exportar. -Se selecciona la opción "Cancelar". -Se cierra la ventana para exportar.
SC 6: Año	EC 6.1: Mostrar disponibilidades por año.	Se muestra una lista de disponibilidades.	<ul style="list-style-type: none"> -Se selecciona un año. -Muestra una lista con todas las disponibilidades del año seleccionado.
	EC 6.2: Mostrar disponibilidades por año.	No se muestran disponibilidades.	<ul style="list-style-type: none"> -Se selecciona un año. - Muestra un mensaje indicando que no hay datos para mostrar.

Tabla 19 Caso de prueba del CU Gestionar disponibilidad de graduados de la enseñanza media

Resultados de las pruebas

Luego de realizadas tres iteraciones de pruebas, en las cuales se detectó un total de 49 no conformidades, las cuales fueron resueltas en su totalidad al término de la tercera iteración, se puede concluir que las pruebas fueron exitosas, el sistema está apto para su utilización y cumple con todas las funcionalidades para las que fue concebido.



Conclusiones

Con la conclusión de este capítulo finaliza la implementación del Sistema Plan de Ingreso a la Educación Media y Superior para el Sistema Unificado de Gestión de Fuerza de Trabajo Calificada junto con su fase de pruebas de aseguramiento de la calidad del software. Fueron implementadas las clases obtenidas en el diseño. Se obtuvieron los diagramas de componentes que describían el módulo a construir y el de despliegue que muestra su organización. Fueron realizadas las pruebas al sistema con resultados satisfactorios que demuestran que la aplicación cuenta con las características y funcionalidades para las que fue concebido.

Conclusiones

Luego de realizarse la investigación, completándose un estudio profundo y minucioso de los procesos de gestión de los planes de ingreso a la Educación Media y Superior y de las tecnologías y tendencias actuales en el desarrollo de software, la modelación de los procesos a automatizar, así como su análisis, diseño y la posterior implementación del sistema, de acuerdo a los requerimientos especificados, se concluye que:

- ✓ Del estudio de los procesos realizados en el departamento de Fuerza de Trabajo Calificada del Ministerio de Economía y Planificación para la gestión de los planes de ingreso a la Educación Media y Superior, se obtuvo como resultado todos los problemas que afronta el departamento en la actualidad, lo que valida la necesidad de la investigación y la actualidad del problema.
- ✓ No se encontró un sistema en el mundo que le diera solución al problema a resolver, aunque los sistemas internacionales estudiados aportaron ideas y conocimientos para el desarrollo del sistema. El estudio de la aplicación que se encuentra actualmente en explotación en el departamento sirvió para definir las funcionalidades del nuevo sistema, y por las características estudiadas de Cedrux, se decidió el desarrollo del sistema a partir de esta aplicación.
- ✓ De los flujos de análisis y diseño se obtuvieron todos los artefactos necesarios para la posterior implementación del sistema, logrando así un mayor entendimiento de los procesos a la hora de ser implementados.
- ✓ Del flujo de implementación se obtuvo como resultado una versión entregable del producto, la cual luego de ser sometida a un conjunto de pruebas por los desarrolladores, el grupo de calidad del proyecto y el cliente, y de corregidas las no conformidades encontradas, se encuentra listo para su liberación a Calidad UCI y su posterior entrega al cliente.

El objetivo general de la investigación se cumplió, pues se lograron automatizar los procesos de gestión de los planes de ingreso a la Educación Media y Superior, logrando agilizar así la gestión de la información del ingreso en el Ministerio de Economía y Planificación. El sistema desarrollado es genérico, maleable ante posibles cambios futuros, posee el nivel de seguridad requerido, una interfaz agradable al usuario y fácil de manipular. La comunicación con los demás subsistemas de Geforza se establece mediante servicios web garantizando así seguridad e integridad en los datos. Se creó una base de datos unificada para todo el sistema evitando así la redundancia de la información. Por todo lo antes planteado se concluye que la investigación fue completada con éxito y la hipótesis planteada queda demostrada.

Recomendaciones

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

- ✓ Actualizar las librerías y marcos de trabajo de la aplicación Geforza a las últimas versiones disponibles (ExtJS 3.2, Zend Framework, 1.10 y Doctrine 1.4), con el fin de lograr mejoras sustanciales en el rendimiento del sistema, así como la corrección de algunos errores que presentan las versiones en uso.
- ✓ Llevar el algoritmo de asignación del Plan de Ingreso a Técnicos Medio y Obreros Calificados hasta el nivel de municipio, puesto que en las últimas reuniones del departamento se decidió la realización de estos planes a nivel municipal.
- ✓ Agregar en los reportes de salida para el Plan de Ingreso a Técnicos Medio y Obreros Calificados tres columnas donde se muestren el plan del año anterior, la cantidad real de estudiantes que matricularon y el porcentaje que esa cantidad matriculada representa con respecto al plan, con el fin de brindar una herramienta más para la toma de decisiones a la hora de aprobar estos planes.

Bibliografía referenciada

1. **DRAUTA**. Sitio oficial de DRAUTA. [En línea] 2010. [Citado el: 15 de 01 de 2010.]
<http://www.drauta.com/erp/>.
2. **Juan Antonio Gonzales García, Gabriela Granados Núñez y Jorge Alonso Aroza Ruano**. Sistema integrado de planificación estadística. [En línea] [Citado el: 17 de 01 de 2010.]
http://www.ican.es/opencms/opencms/ponencias/ANDALUCIA/SISTEMA_INTEGRADO_DE_PLANIFICACION_ESTADISTICA._SINPLE.doc.
3. **ALCOFIN**. ADAM 5 Sistema integral de nómina y recursos humanos. [En línea] [Citado el: 05 de 03 de 2010.] <http://www.alcofin.com.mx/adam.html>.
4. **Pressman, Roger S**. *Ingeniería del software. Un enfoque práctico*. s.l. : McGraw-Hill/Interamericana de España., 2002.
5. **Ivar Jacobson, Grady Booch y James Rumbaugh**. *El Proceso Unificado de Desarrollo de Software*. 2000.
6. ¿Qué es un Framework? [En línea] [Citado el: 18 de 01 de 2010.] <http://buenmaster.com/?a=554>.
7. Doctrine: ORM Open Source para PHP 5.2+. [En línea] [Citado el: 20 de 01 de 2010.]
<http://www.joangarnet.com/blog/?p=415>.
8. *Fase de Elaboración. Flujo de Trabajo de Análisis y Diseño*. Ciudad de la Habana : s.n., 2009.
9. Entorno virtual de aprendizaje, ISW 2 . [En línea] <http://eva.uci.cu/mod/resource/view.php?id=14105>.

Bibliografía consultada

1. ¿Qué es Doctrine ORM? <http://www.tecnoretas.com/programacion/que-es-doctrine-orm/>
2. Aplicación de recursos humanos y nómina universal versión 5
<http://www.guiagrh.com/producto.aspx?pid=932>
3. *Aprendiendo UML en 24 horas*. México PEARSON EDUCACIÓN 2000 ISBN: 968-444-463-X
4. *Avaliação: Revista da Avaliação da Educação Superior. Revolución cubana y educación superior. 2*
Campinas ISSN 1414-4077
5. Doctrine Vs Propel. <http://www.hasheado.com/doctrine-vs-propel.html>
6. Documentación-Manual de PHP http://www.hospedajeydominios.com/mambo/documentacion-manual_php.html
7. *El lenguaje unificado de modelado. Manual de Referencia*
8. Entorno Virtual de Aprendizaje. Clase Teórica-Práctica 6. Técnicas de Estimación. Ingeniería de Software 1 <http://eva.uci.cu/mod/resource/view.php?id=22433>
9. *Estudio Integral para la Formación y Empleo de la Fuerza de Trabajo Calificada. Indicaciones Metodológicas para su Actualización* Ciudad de la Habana 2008
10. ExtJS en Español. Comunidad de desarrollo <http://extjs.es/>
11. Ingenieros de Software. Patrones de diseño.
<http://www.ingenierossoftware.com/analisisydiseno/patrones-diseno.php>
12. *La biblia Servidor Apache 2*.
13. *La calidad de software y su medida* Madrid Centro de Estudios Ramón Areces S.A ISBN: 84-8004-61 1-2
14. NetBeans 6.5 el único IDE que necesitas. http://www.techblog.com/talks/netbeans65es_cl.pdf
15. NetBeans IDE. Sitio Oficial. <http://netbeans.org/>
16. *PHP 5. Power Programming*. Hagerstown, Maryland Prentice Hall 2004 ISBN 0-131-47149-X

17. PHP. Hypertext Preprocessor <http://php.net/>
18. PostgreSQL La Base datos Open Source más avanzada. <http://www.postgresql.cl/>
19. PostgreSQL.com. Kit de prensa de PostgreSQL 8.3
<http://www.postgresql.org/about/press/presskit83>
20. PostgreSQL-Documentación en español. <http://palomo.usach.cl/docshtml/node4.html>
21. *Practical PostgreSQL* O´Reilly Media Inc. 2002 ISBN 1-56592-846-6
22. Sistema de planificación de recursos empresariales. <http://www.infor.es/erp-software/>
23. Tutorial de Apache .Servidor Web <http://www.ayuda-internet.net/tutoriales/desarrollo/apache/index.html>
24. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México PRENTICE HALL
1999 ISBN: 970-17-0261-1
25. UML: Diagramas. UML ¿Qué es UML? <http://www.ingenierossoftware.com/analisisydiseno/uml.php>
26. Visual Paradigm para UML. Sitio Oficial.<http://www.visual-paradigm.com/product/vpum/>
27. Zend Framework documentación en español <http://zfdes.com/index.php/Portada>

Glosario de términos

BD: Base de Datos.

CAP: Consejo de Administración Provincial.

DC: Diagrama de colaboración.

DCA: Diagrama de clases del análisis.

DCD: Diagrama de clases del diseño.

DPA: División Político Administrativa.

FTC: Fuerza de Trabajo Calificada.

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

IDE: Entorno Integrado de Desarrollo.

MEP: Ministerio de Economía y Planificación.

MVC: Modelo Vista Controlador.

NS: Nivel Superior.

OACE: Organismos de Administración Central del Estado.

OC: Obrero Calificado.

ORM: Mapeador Relacional de Objetos.

PI: Plan de Ingreso.

POO: Programación Orientada a Objetos.

RF: Requerimiento funcional.

RNF: Requerimiento no funcional.

RUP: Proceso Unificado de Desarrollo.

RUP: Proceso Unificado de Rational.

SGBD: Sistema Gestor de Base de Datos.

TIC: Tecnologías de la Información y las Comunicaciones.

TM: Técnico Medio.

UCI: Universidad de las Ciencias Informáticas.

UML: Lenguaje Unificado de Modelado.