

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE CARTERA PARA LA ASOCIACIÓN DE DOCENTES DE LA ESCUELA POLITÉCNICA DEL EJÉRCITO

Gabriela Molina Vintimilla¹, Ricardo David Barrera², Mario Ron³, Carlos Procel⁴

¹ Escuela Politécnica del Ejército, Ecuador, anaga_31@hotmail.com

² Escuela Politécnica del Ejército, Ecuador, odracir6891@hotmail.com

³ Escuela Politécnica del Ejército, Ecuador, mbron@espe.edu.ec

⁴ Escuela Politécnica del Ejército, Ecuador, ctprocel@espe.edu.ec

RESUMEN

La Asociación de Docentes de la ESPE, realiza todos sus movimientos de cartera de forma manual, este proceso comienza con la revisión de pagos y descuentos mensuales que tiene cada docente, dependiendo del servicio que tenga, ya sea para operadoras celulares (Claro, Movistar) y/o Seguro de Vida.

La ADESPE registra esta información en una hoja electrónica de Excel, y la recolección de la misma no es óptima, ya que ésta proviene de varios archivos enviados por los proveedores, detallando el consumo mensual de cada socio; finalmente se consolida toda esta información en un solo documento, donde se inserta manualmente la cédula, el nombre del docente y el valor que envía cada proveedor, para luego realizar los cálculos respectivos y obtener el total que será descontado de su salario mensual.

Llevar de ésta manera el proceso de cartera, dificulta el manejo y el acceso a la información concerniente para la toma de decisiones, situando así a la ADESPE en desventaja ya que no cuenta con un proceso automatizado.

Esta realidad hizo que sea indispensable el análisis, diseño e implementación de un Sistema de Información que brinde todas las facilidades en la entrega oportuna de la información que se requiera en cuanto a descuentos se refiere, el mismo que debe considerar la metodologías Object Modeling Technique (OMT¹) y el lenguaje de modelamiento UML², herramientas de desarrollo como Eclipse IDE, StarUML, Java, My SQL, a fin de conseguir un producto que permita satisfacer las expectativas y requerimientos de la ADESPE.

Como resultado final, del desarrollo realizado se obtuvo una aplicación web, que fue probada e instalada dentro de la ADESPE, la misma facilita la gestión de los descuentos de los diferentes servicios, para cada socio.

Palabras Clave: OMT, ADESPE, aplicación WEB, servicios de telefonía, servicios médicos.

ABSTRACT

The Army Polytechnic School Teacher's Association, performs all its portfolio movements manually, this process starts with the review of monthly payments and discounts from each teacher, depending of the service, either cellular operators (Claro, Movistar) or life insurance.

The ADESPE registers this information in an Excel's electronic sheet, and this is not an optimum process, as this comes from some files sent by the providers, detailing the monthly consumption of each member, finally all the information is consolidated in a single file, where the ID, the teacher's name, and the values sent for each

¹ Object Modeling Technique

² Unified Modeling Language

provider is inserted manually, to subsequently made the respective calculation, and get the total to be deducted in the month.

Have this way the portfolio process, difficult the control and access to the information used to make decisions, creating a disadvantage to the ADESPE as this does not have an automatic process.

This reality made indispensable the analysis, design and implementation of an Information System, that provides the facilities in the opportune deliver of the information about the discounts, the same that have to consider Object Modeling Technique (OMT) methodologies and the modeling language UML, development tools as Eclipse IDE, StartUML, Java, MySQL, to get a product that allows to satisfy the expectations and requirements of the ADESPE.

As a final result of the development, we got a web application that was proved and installed in the ADESPE, application that facilitates the management of discounts about the different services for each member.

Keywords: OMT, ADESPE, web application, telephony services, medical services.

1. INTRODUCCIÓN

Hoy en día el manejo de la información es de vital importancia en el desempeño de cualquier actividad empresarial y es por ello que la implementación de aplicaciones informáticas, es necesario dentro de una organización como base para la automatización de procesos, toma de decisiones, reducción de tiempos y costos.

Automatizando un sistema, podremos encontrar varias ventajas para cumplir los objetivos de una organización, permitiendo así incrementar la productividad, la optimización del equipo de trabajo, mejorar la calidad y el control del trabajo en una oficina.

El proyecto tiene como objetivo principal, automatizar el proceso de administración de descuentos que permita a los docentes conocer la información detallada, llevando un control de sus descuentos, ya sea de servicios telefónicos, servicios médicos o seguro de vida; utilizando para el desarrollo software libre para la base de datos y open source para el sistema; el hardware responde a las necesidades de la ADESPE y a un costo razonable. Uno de los elementos más importantes del sistema es la administración de hojas de cálculo que provienen mensualmente de cada proveedor que brinda servicio a la ADESPE, ya que depende de estos para poder realizar los cálculos pertinentes para que cada socio pueda acceder a la información.

Para el desarrollo de este proyecto de tesis, se empezó por definir los requerimientos del usuario y las especificaciones del sistema, para así poder llevar una definición y un control del desarrollo entre el cliente y los desarrolladores, este documento se desarrolló basándose en la normativa IEEE-830³; utilizado esta norma pudimos obtener un mayor entendimiento de las necesidades del cliente, los requerimientos de los usuarios de la ADESPE fueron diagramados con la herramienta StarUML que facilitó la construcción del sistema para los desarrolladores.

En primer lugar se describirá la metodología que se utilizó, a continuación se detallará cada una de las herramientas que se usaron para el desarrollo del sistema y por último se describirán los resultados obtenidos.

2. METODOLOGÍA

En el desarrollo de aplicaciones web es importante el análisis, diseño, modelamiento y planificación, permitiendo así, obtener un producto de calidad que satisfaga todos los requerimientos del usuario.

La metodología OMT fue escogida para el desarrollo de este proyecto, ya que da prioridad al modelo y el uso del mismo para lograr una abstracción, el análisis se realiza en base a la realidad. [3]

³ Institute of Electrical and Electronics Engineers

La Metodología OMT se divide en cuatro etapas que son:

1. **Análisis de Objetos:** Se enfoca en comprender, modelar el problema mostrando propiedades importantes. Se realiza una abstracción precisa y resumida de lo que se desea que haga el sistema. [3]
2. **Diseño del Sistema:** Definir la arquitectura del sistema, organizándolo en subsistemas basados en la arquitectura propuesta.[4]
3. **Diseño de Objetos:** Se depura el análisis de objetos, incorporando detalles de implementación, centrándose en datos esenciales para la implementación de las clases.[4]
4. **Implementación:** Se codifica las clases y relaciones desarrolladas durante el análisis de objetos y se los prueba.[3]

2.1 Análisis de Objetos

Se describe el problema, obteniendo requisitos y cubriendo todos los aspectos relevantes sin dejar dudas; se realizan los diagramas de objetos, definiendo las estructuras de los objetos y las relaciones entre ellos, así como también el diagrama de clases con un diccionario de datos que de una descripción de las mismas.[4]

Después se crea un modelo dinámico, describiendo aspectos de control y evolución del sistema, incluyendo un diagrama flujo del sistema y diagramas de estado de las clases que tengan un comportamiento dinámico.

Posteriormente se realiza el modelo funcional, describiendo las funciones, los valores de entrada y salida. [4]

Finalmente se revisan los modelos realizados para corregir los mismos. En resumen:

1. Definir el problema
2. Realizar el modelo de objetos.
3. Realizar el modelo dinámico.
4. Realizar el modelo funcional.
5. Verificar los tres modelos.

2.2 Diseño

- **Diseño del Sistema**

Es la estrategia para resolver el problema y construir una solución, incluyendo decisiones para la organización del sistema en subsistemas.

El modelo de diseño debe ser eficiente y práctico a la hora de transformarlo en código, incluyendo detalles de bajo nivel que se omiten en el análisis.[3]

Pasos a seguir son:

1. Organización del sistema en subsistemas.
2. Identificar la concurrencia del problema.
3. Asignación de los subsistemas a procesadores y tareas.
4. Elegir como se van a implementar los almacenes de datos, en términos de estructura de datos.
5. Determinar los datos globales y como controlar el acceso a los mismos.
6. Selección de un esquema para el control de software.

- **Diseño de Objetos**

Se toman decisiones necesarias para realizar el sistema sin tener que llegar a los detalles particulares de un lenguaje o base de datos.

Se recomiendan las siguientes etapas:

1. Obtener las operaciones para el modelo en base a los demás modelos.
2. Diseñar algoritmos para implementar las operaciones.
3. Optimizar las vías de accesos a los datos.
4. Implementar el control de software de acuerdo al esquema seleccionado en el diseño del sistema.
5. Ajustar la estructura de clases para introducir la herencia.
6. Diseñar la implementación de las asociaciones.
7. Determinar la expresión de los atributos que son objetos.
8. Empaquetar clases y asociaciones en módulos.

2.3 Implementación del sistema

Se codifican las estructuras en el dominio de la aplicación como en el dominio de la solución, basándose en el diseño de objetos.[4]

3. ANALISIS DE REQUERIMIENTOS

El objetivo que tiene la especificación de requerimientos, es cumplir con todas las expectativas que tiene la ADESPE, llegando así a un acuerdo entre los desarrolladores y el usuario, para el correcto desarrollo e implementación del sistema de gestión de cartera para la asociación de docentes.

3.1 IEEE-830

IEEE-830 es una estándar que nos ayudó a especificar los requerimientos que tiene el usuario, donde se describió lo que quieren obtener, para mayor comprensión de los desarrolladores, el producto final es un acuerdo documentado entre el usuario y el grupo de desarrollo.[1]

3.2 Perspectiva del producto

El Sistema SIADCA, fue estructurado y diseñado para trabajar en un entorno WEB que trabajará de manera independiente, es decir no interactuará con otro sistema.

El desarrollo del Sistema pretende optimizar recursos, tiempo y sobre todo automatizar los procesos de descuento y pagos de cada docente.[7]

Las funciones se clasifican en módulos de la siguiente manera:

- **Módulo de Seguridad:** Se encarga de la creación, eliminación, búsqueda y edición de usuarios. Este módulo será gestionado únicamente por el administrador, que será el responsable de realizar el registro cada que sea necesario.
- **Módulo de Mantenimiento:** Se encarga de la creación, eliminación, búsqueda y edición de socios y tipos de socios. En el caso de la búsqueda se lo realiza por socio. Este módulo será administrado de acuerdo al perfil.
- **Módulo de Proceso:** Este módulo se encarga de la carga de archivos, los mismos que son enviados de las empresas que brindan los diferentes servicios que necesita la ADESPE, se podrá visualizar de manera global y detallada los descuentos que tiene los socios.
- **Módulo de Consultas:** En este módulo se obtiene el valor a pagar de cada socio por mes, valor que el del departamento financiero de la ESPE se encarga de descontar en el rol de pagos de cada uno.

3.4 Herramientas de desarrollo

3.4.1 IDE Eclipse Índigo

Eclipse es una plataforma de desarrollo open source basada en Java. En sí mismo Eclipse es un marco y un conjunto de servicios para construir un entorno de desarrollo a partir de componentes conectados (plug-in).

Eclipse fue originado inicialmente por IBM y actualmente se desarrolla por la fundación Eclipse sin fines de lucro. En cuanto se refiere a las aplicaciones clientes, Eclipse provee al desarrollador framework muy ricos para aplicaciones gráficas, definición y manipulación de modelos de software, aplicaciones web, etc.

Es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar aplicaciones de cliente. [9]

3.4.2 Java

Java es un lenguaje de programación orientado a objetos portable a cualquier plataforma, permitiendo así realizar aplicaciones distribuidas que tienen un alto control de seguridad en las aplicaciones. Esta tecnología permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios.

Java puede ofrecer un gran número de herramientas como compiladores (javac), intérprete (java), generador de documentos (javadoc), etc. [10]

Es un lenguaje orientado a objetos debido a que Java trabaja con sus datos como objetos y con interfaces a esos objetos. Abarca tres características de propias del paradigma de orientación a objetos:

1. **Encapsulamiento:** Es reunir a todos los elementos que pertenecen a una misma entidad, al mismo nivel de abstracción, en la combinación de los datos y las operaciones que pueden producirse sobre esos datos en un objeto.[10]
2. **Herencia:** Permite la creación de nuevas clases que son descendientes que se construyen sobre otras donde los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. [10]
3. **Polimorfismo:** Permite que un mismo mensaje pueda actuar sobre diferentes tipos de objetos y comportarse de modo distinto. [10]

3.4.3 Motor de base de datos MYSQL

MySQL es un sistema gestor de bases de datos muy conocido y ampliamente usado por su simplicidad y notable rendimiento, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta.

Está disponible para múltiples plataformas, sin embargo, las diferencias con cualquier otra plataforma son prácticamente nulas, ya que la herramienta utilizada en este caso es el cliente mysql-client, que permite interactuar con un servidor MySQL (local o remoto) en modo texto. De este modo es posible realizar todos los ejercicios sobre un servidor instalado localmente o, a través de Internet, sobre un servidor remoto.

MySQL es reconocido por que su rapidez en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede existir algunos problemas de integridad. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. [11]

3.4.4 Herramienta case Star UML

Es una herramienta que permite modelar distintas vistas como: Vista de Casos de Uso, Vista lógica, Vista de Componentes, Vista de Despliegue, en donde cada vista define trabajo en procesos y diagramas UML. Por medio de cada vista permite al usuario desarrollar diferentes tipos de diagramas que serán descritos a continuación.

- **Vistas de Casos de Usos:** Diagramas de Casos de Uso, Diagramas de Iteración y diagramas de Actividad.
- **Vista Lógica:** Diagramas de Clase y Diagramas de Estado.
- **Vista de Componentes:** Diagrama de Componentes.
- **Vista de despliegue:** Diagrama de Despliegue.[2]

3.5 Funciones de Control

Es Sistema controla los permisos de acceso que tienen los usuarios de manera correcta, de tal forma que puedan acceder a la información que tiene el sistema de acuerdo al rol que cada uno desempeña.[1]

3.6 Protocolos Señalados

Para lograr la conexión del sistema y de una manera ágil, se utilizará el protocolo TCP/IP y HTTP, es un protocolo conocido por los usuarios, de la misma manera facilita su desarrollo y si es el caso el mantenimiento en un futuro.[1]

3.7 Credibilidad de la Aplicación

El sistema tuvo una serie de pruebas, donde se pudo establecer que cumple con los requerimientos que solicitó la ADESPE, tanto en consistencia de información como en rendimiento de la aplicación y sus tiempos de respuesta.

4. ANÁLISIS Y DISEÑO

Una vez que ya se cumplió todos los requisitos señalados por la norma IEEE-830, podemos dar paso a la creación de una abstracción de la realidad, para el mayor entendimiento del sistema que fue desarrollado.

La parte dinámica del sistema estará representada por los siguientes diagramas:

- Diagramas de casos de Usos
- Diagramas de secuencias
- Diagramas de estado

La parte estática del sistema estará representada por los siguientes diagramas:

- Diagramas de clases
- Diagramas de objetos
- Diagramas de componentes
- Diagramas de despliegue

4.1 Identificación de Actores

Se definió los actores que intervienen en el sistema, de acuerdo a los requerimientos del se identificó los siguientes actores:

- **Administrador:** Es la persona encargada de administrar todo el funcionamiento del sistema, debe controlar el acceso de usuarios y los privilegios que tiene cada uno.
- **Secretaria:** Es la persona encargada de llevar el control administrativo de descuentos dentro del sistema.
- **Socio:** Es la persona que solo puede consultar los descuentos que ha tenido durante el mes.

4.2 Diagrama de Caso de Uso

Una vez levantado el proceso de análisis de requerimientos, se desarrolló el diagrama de casos de uso. La Ilustración 1 muestra el modelo conceptual del sistema.



Ilustración 1 Diagrama de Casos de uso del Sistema

Cada caso de uso es parte fundamental del Sistema, a partir de ellos se crearon los distintos módulos que dan la funcionalidad necesaria e interacción con él usuario.

4.3 Diagrama de despliegue

Para lograr la implementación del sistema se usó la arquitectura definida en la Ilustración 2.

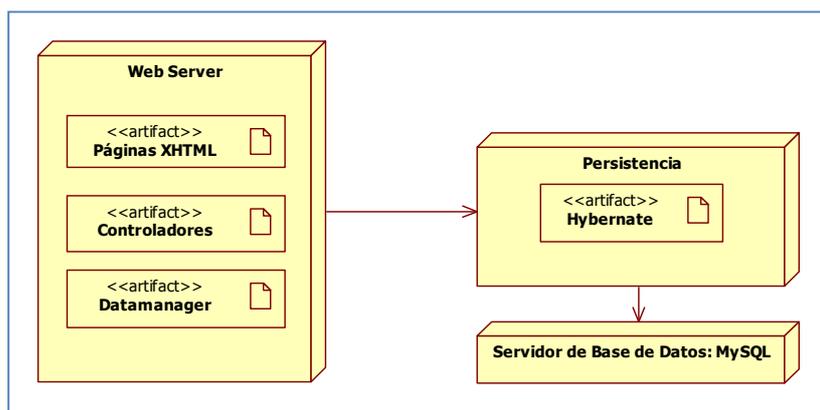


Ilustración.2 Diagrama de Despliegue

Para el desarrollo se utilizó la Arquitectura MVC (Model View Controller), en la cual el modelo se encarga de almacenar la información en una base de datos con reglas de negocio, los controladores son los encargados

de responder a acciones, comúnmente a eventos del usuario y la vista es una representación del modelo en un formato fácil para interactuar con los usuarios (páginas web). [8]

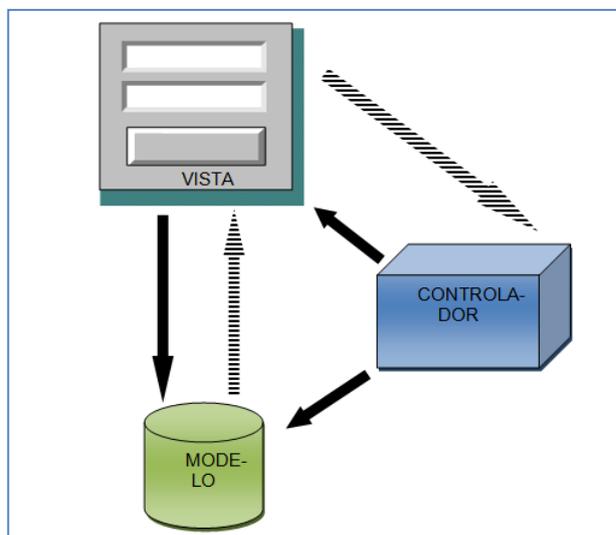


Ilustración 3 Arquitectura MVC

Se organizó el proyecto en 3 capas:

- **Presentación:**
Esta capa se encarga de recibir todas las peticiones del usuario.
- **Negocio**
Esta capa maneja todos los procesos de negocio de la ADESPE.
- **Persistencia**
Esta capa es la encargada almacenar toda la información de la ADESPE en las tablas de la base de datos.

5 PRUEBAS Y RESULTADOS

Una vez desarrollado el Sistema de Gestión de Cartera, se instaló en los computadores de la ADESPE y se realizaron las pruebas correspondientes con los usuarios, obteniendo los siguientes resultados.

1. Las pruebas del sistema se realizaron localmente y se espera que en el futuro el sistema esté disponible en un micro-sitio de la ESPE, de esta manera, el socio podrá consultar a través de la web, por mes y año todos los descuentos que tiene y el total a ser descontado en el rol de pagos.
2. La secretaria puede ingresar información de los socios y así poder llevar un registro de los socios que están activos o inactivos.
3. La secretaria obtendrá una mejor administración de los descuentos, debido a que el sistema le proporciona mensualmente una hoja de cálculo Excel, con el número de cédula, el nombre y el valor total a descontar y el valor que fue descontado. Así, no tendrá que realizar manualmente este proceso.

6 CONCLUSIONES Y TRABAJOS FUTUROS

Se desarrolló una aplicación web, que permite realizar la administración de cartera de la ADESPE optimizando el proceso manual que se llevaba, permitiendo así al personal del departamento en especial a la secretaria llevar un control y una mejor gestión de la información de cada socio.

Utilizar la metodología OMT, hizo que los cambios que se realizaban periódicamente en la fase de análisis de requerimientos no sean de gran impacto, ya que permitió centrarnos en los aspectos relevantes del

problema para plasmar los procesos en papel, así como el desarrollo de los diagramas requeridos por la metodología.

OMT es una metodología iterativa en el análisis de requerimientos, esto hizo que exista un retraso significativo en el inicio de la fase de implementación, ya que necesitamos que esté correctamente definido los requerimientos y así poder construir una aplicación acorde a las necesidades que tienen los usuarios.

En el futuro se podría mejorar el sistema desarrollado reportes que faciliten la visualización de la información para los socios y secretaria.

7 AGRADECIMIENTOS

A los Ingenieros Mario Ron y Carlos Procel por toda la ayuda brindada y por compartir sus conocimientos a lo largo de la elaboración de nuestro proyecto.

A las personas de la ADESPE por brindarnos toda la información y material necesario para el desarrollo de la aplicación.

8 REFERENCIAS BIBLIOGRÁFICAS

- [1] Gmendez. Especificación de Requisitos según el estándar de IEEE 830. Recuperado el 2013, de <http://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>
- [2] Minkyu Lee, Hyunsoo Kim, Jeongil Kim, Jangwoo. StarUML 5.0 Developer Guide. Recuperado el 2013, de http://staruml.sourceforge.net/docs/StarUML_5.0_Developer_Guide.pdf
- [3] Pierre Rodríguez Farías. Metodologías de Desarrollo de Aplicaciones Web. Recuperado el 2013, de <http://es.scribd.com/doc/57552560/Metodologias-de-Desarrollo-Para-Aplicaciones-Web>.
- [4] **Pressman, Roger. S(2010) Ingeniería del Software.** Un Enfoque Práctico (7maEd.). McGraw - Hill: España.
- [5] **Andreas Rueping. Agile Documentation: A Pattern Guide to Producing Lightweight Documents for Software Projects.** Wiley Software Patterns Series.
- [6] **Scott Wampler, Ron Jeffries.** Agile Modeling: Effective Practices for Extreme Programming and the Unified Process.
- [7] **Ricardo Barrera, Gabriela Molina.** Proyecto de titulación. ESPE, 2013.
- [8] By RichFaces. RichFaces 5.0.0.Alpha1 Released. Recuperado el 2013 de Junio de 17, de <http://www.jboss.org/richfaces>
- [9] David Axmark y Michael "Monty" Widenius. MySQL 5.0 Reference Manual.Mysql. Recuperado el 2013 de junio 26 de <http://www.mysql.com/services/>