

# ANÁLISIS, DISEÑO Y CONSTRUCCIÓN DEL SUBSISTEMA DE INGRESO Y SELECCIÓN DE CANDIDATOS A LAS ESCUELAS DE FORMACIÓN DEL EJÉRCITO ECUATORIANO UTILIZANDO HERRAMIENTAS OPEN SOURCE

Sr. Edwin Ricardo Meneses Becerra<sup>1</sup>, Ing. Ramiro Delgado<sup>2</sup>, Ing. Santiago Salvador<sup>3</sup>  
1,2,3, Departamento de Ciencias de la Computación, Escuela Politécnica del Ejército,  
Sangolqui, Ecuador, [ermeneses@gmail.com](mailto:ermeneses@gmail.com), [rndelgado@espe.edu.ec](mailto:rndelgado@espe.edu.ec),  
[mssalvador@espe.edu.ec](mailto:mssalvador@espe.edu.ec)

## RESUMEN

En la actualidad el departamento de obtención de la dirección de Recursos Humanos no consta con un sistema informático que le permita realizar el proceso de selección y reclutamiento de una manera optima, en esta tesis se realizo el análisis, diseño y construcción de un sistema que llene las expectativas del departamento de obtención y les ayude con los procesos de selección y reclutamiento. El desarrollo del sistema se realizo con herramientas open source con excepción de la base de datos que es Oracle, debido a que el Ejército Ecuatoriano tiene desarrollados todos sus sistemas en esta herramienta, Para elaborar este sistema se opto por una aplicación web desarrollada con Java, debido a que es necesario trabajar en diferentes repartos militares y la manera de interactuar más segura y eficiente es por medio del internet puesto que estos repartos militares no tiene conexión directa con la base de datos. Como servidor de aplicaciones utilizaremos Glassfish V2.0 debido a la integración con Netbeans y Java, este servidor de aplicaciones se encontrara administrado por la Dirección de Comunicaciones y Sistemas del Ejército Ecuatoriano y se encontrara instalado conjuntamente con la Base de datos Oracle en el cuarto de servidores de la Comandancia del Ejército.

**Palabras Clave:** herramientas open source, aplicación web, Java, Glassfish V2.0, Netbeans, Base de datos Oracle

## ABSTRACT

Currently the department to obtain the direction of Human Resources has not a computer system that allows you to make the selection and recruitment process in an optimum way, this thesis is the analysis, design and construction of a complete system department's expectations and help them obtain the recruitment and selection processes. The system development was done with open source tools with the exception of the database is Oracle, because the Ecuadorian Army has developed all systems on this tool, to develop this system we opted for a web application developed with Java, because it is necessary to work in various divisions military and how to interact more safely and efficiently is through the internet as these military deals has no direct connection to the database. As Glassfish Application Server V2.0 use due to integration with Netbeans and Java, this application server is managed by the find of Communications and Systems Ecuadorian Army and was found installed in conjunction with the Oracle database in the room servants of the Army Command.

**Keywords:** open source tools, Web application, Java, Glassfish v2.0, Netbeans, Oracle Database

## 1. INTRODUCCIÓN

El Ejército Ecuatoriano cada año recluta cerca de 5000 candidatos para las Escuelas de Formación, de estos candidatos selecciona a los aspirantes que cumplan con los requisitos y aprueben las diferentes pruebas, por lo tanto; es indispensable desarrollar un sistema de información que permita administrar esta selección de una manera técnica y de forma eficiente que esté de acuerdo a las necesidades y requerimientos de la Institución.

El proceso de selección requiere de un sistema de información, que permita administrar la información relevante y llevar un control real en cada una de las etapas de selección que se realiza en un periodo determinado.

En el mercado nacional e internacional, se han comercializado sistemas de software, que pretenden satisfacer las necesidades del Ejército Ecuatoriano, pero estos no cubren todas las expectativas y requerimientos particulares que el proceso de selección tiene.

Se necesita elaborar un sistema que tome en cuenta otros factores personalizados que satisfagan las perspectivas del nivel gerencial y apoyen al nivel operativo, por lo que se ha visto indispensable la construcción de una herramienta informática, que permita estructurar un Sistema de Ingreso y Selección eficaz y eficiente.

Esta solución está orientada a proporcionar esa herramienta, utilizando métodos y técnicas de Ingeniería de Software que logren un producto de calidad, en funcionalidad, oportunidad y costo.

## 2. METODOLOGÍA

El trabajo planteado exige inicialmente un estudio detallado de cada uno de los procesos de Ingreso y Selección, así también una Recopilación de información en cada uno de las tres Escuelas de Formación con las que cuenta el Ejército Ecuatoriano en este momento, para desarrollar reglas de negocios certeras y con validas.

Posteriormente se utilizará para el desarrollo del Sistema, las metodologías combinadas RUP.

### 2.1. PROCESO UNIFICADO RACIONAL (*Rational Unified Process, RUP*)

Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos, **y que además es el estándar que utiliza el Ejército Ecuatoriano.**

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

## 3. HERRAMIENTAS

Las Herramientas que se utilizan para el desarrollo del Sistema se detallan según las funciones para las que se utilizará cada una:

### 3.1 NETBEANS

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos*. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

### 3.2 JAVA

Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Fue desarrollado por la compañía Sun Microsystems.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si hacemos un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, entre otros. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa y posibilita que este último se entienda perfectamente.

### 3.3 GLASSFISH.

Es un servidor de aplicaciones de software libre desarrollado por Sun Microsystems, compañía adquirida por Oracle Corporation, que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. GlassFish está basado en el código fuente donado por Sun y Oracle Corporation, éste último proporcionó el módulo de persistencia *TopLink*. GlassFish tiene como base al servidor *Sun Java System Application Server* de Oracle Corporation, un derivado de Apache Tomcat, y que usa un componente adicional llamado Grizzly que usa Java NIO para escalabilidad y velocidad.

### 3.4 ORACLE

Es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), desarrollado por Oracle Corporation.

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando:

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Soporte multiplataforma.

Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo GNU/Linux.

### **3.5 POWERDESIGNER 12.**

Herramienta case intuitiva y estructurada, que permite el modelamiento de datos con UML, arquitectura, metadatos, etc., para el diseño y análisis del software de manera fácil y ágil, para que así las diferentes tecnologías de desarrollo estructuren soluciones para la arquitectura empresarial y de información.

## **4. METODOLOGÍA DE DESARROLLO DE SOFTWARE RUP, PROCESO UNIFICADO RACIONAL, SUS FASES Y CARACTERÍSTICAS.**

El Proceso Unificado Racional, Rational Unified Process en inglés, y sus siglas RUP, es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino que trata de un conjunto de metodologías adaptables al contexto y necesidades de cada organización, donde el software es organizado como una colección de unidades atómicas llamados objetos, constituidos por datos y funciones, que interactúan entre sí.

También se conoce por este nombre al software desarrollado por Rational, hoy propiedad de IBM, el cual incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades. Está incluido en el Rational Method Composer (RMC), que permite la personalización de acuerdo a necesidades.

Originalmente se diseñó un proceso genérico y de dominio público, el Proceso Unificado, y una especificación más detallada, el Rational Unified Process, que se vendiera como producto independiente.

RUP se divide en 4 fases, dentro de las cuales se realizan varias iteraciones según el proyecto y en las que se hace mayor o menos esfuerzo en las distintas actividades. En las iteraciones de cada fase se hacen diferentes esfuerzos en diferentes actividades:

### **4.1. FASE DE INICIO (Inspección y Concepción).**

Se hace un plan de fases, donde se identifican los principales casos de uso y se identifican los riesgos. Se concreta la idea, la visión del producto y como se enmarca en el negocio así como se determina el alcance del proyecto.

### **4.2. FASE DE ELABORACIÓN.**

Se realiza el plan de proyecto, donde se completan los casos de uso y se mitigan los riesgos. Se planifica las actividades necesarias y los recursos requeridos, especificando las características y el diseño de la arquitectura.

### **4.3. FASE DE CONSTRUCCIÓN.**

Se basa en la elaboración de un producto totalmente operativo y en la elaboración del manual de usuario. Construir el producto, la arquitectura y los planes, hasta que este listo para ser enviado a la comunidad de usuarios.

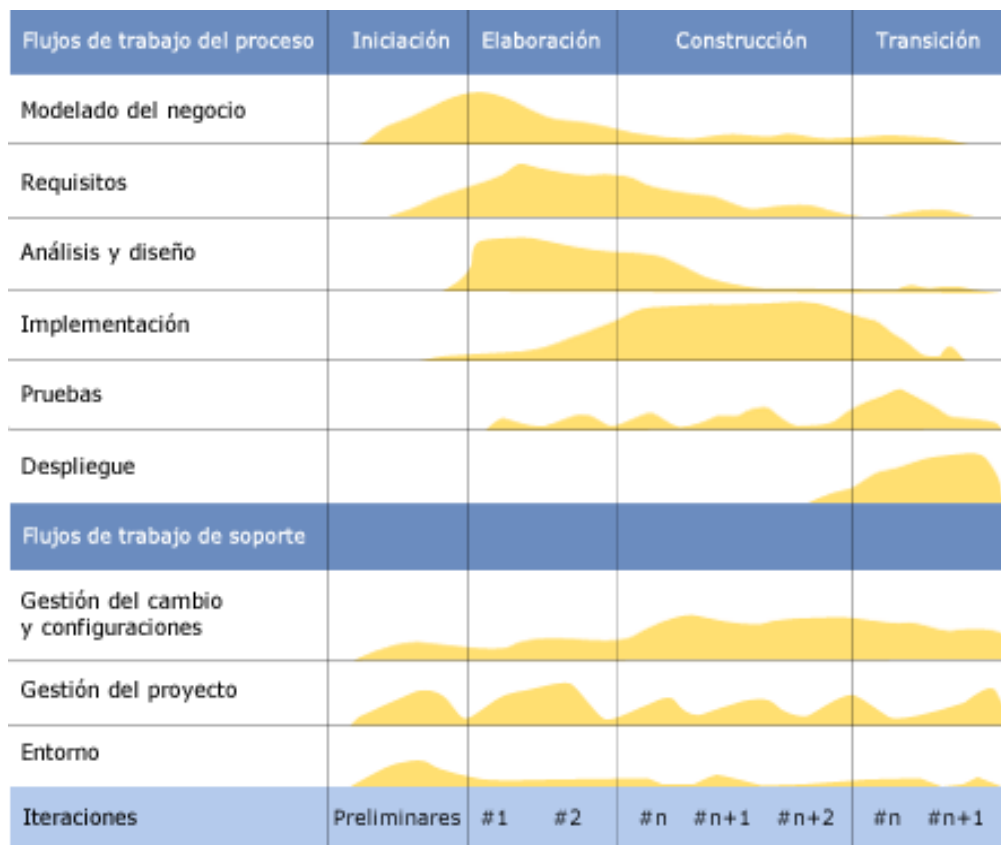
#### 4.4. FASE DE TRANSICIÓN.

Se realiza la instalación del producto en el cliente y se procede al entrenamiento de los usuarios. Se realiza el entrenamiento en el uso del producto a los usuarios, lo cual incluye: construcción, entrenamiento, soporte y mantenimiento del producto, hasta que el cliente quede satisfecho, por tanto en esta fase suelen ocurrir cambios.

Con estas fases se logra ejecutar el conjunto de mejores prácticas que se listan a continuación:

- Desarrollar Software Iterativamente
- Modelar el software visualmente
- Gerenciar los Requerimientos
- Usar arquitecturas basadas en componentes
- Verificación continua de la calidad
- Gerenciar los cambios

La **Figura 4.1** muestra las interacciones en las etapas de RUP.



**Figura 4.1 Iteraciones de las etapas de RUP**

#### 4.5. ESCRIPCIÓN DE LAS FASES

Dependiendo de la iteración el equipo de desarrollo puede realizar diferentes tipos de actividades. Veamos de qué trata cada fase.

## **4.6. FASE DE INICIO**

Durante la fase de inicio las iteraciones ponen mayor énfasis en actividades de modelado del negocio y de requisitos.

En esta fase se elaboran los siguientes entregables.

- Un documento con la visión del proyecto.
- El modelo de Casos de Uso con una lista de todos los Casos de Uso y los actores que puedan ser identificados.
- Un glosario inicial del proyecto.
- Un Caso de Uso inicial de Negocio el cual incluye: contexto del negocio, criterios de éxito y planificación financiera.
- Un estudio inicial de riesgos.
- Un plan del proyecto que muestre las fases y las iteraciones.

El objetivo de esta fase, es definir el modelo de negocio que permita entender las funciones de la organización y del cliente, tanto en estructura como en sus procesos. En esta fase se modela las funciones y roles que realiza la organización para optimizarla mediante una reingeniería de procesos y la implantación del nuevo sistema. También se describe lo que el sistema tendrá que realizar y permitir que los desarrolladores y el cliente estén de acuerdo con esta descripción.

Para ello se realizarán las siguientes subfases:

- Describir los requerimientos funcionales y no funcionales (rendimiento esperado, plataformas soportadas, integración con sistemas externos, etc.).
- Capturar un glosario o vocabulario del sistema o proyecto (mediante documento y clases conceptuales).
- Encontrar actores y casos de uso.
- Describir los casos de uso mediante su flujo principal, variaciones y excepciones.
- Asignar prioridades a los casos de uso encontrados para poder planificar la iteración en forma de análisis, diseño e implementación.
- Modelar la interfaz de usuario (diseño lógico).
- Desarrollar un prototipo de la interfaz de usuario (diseño físico).

## **4.7. FASE DE ELABORACIÓN**

En esta fase las iteraciones se orientan al desarrollo de la arquitectura, que incluye los flujos de trabajo de requerimientos, modelo de negocio (refinamiento), análisis, diseño y una parte de implementación orientado a la arquitectura.

En esta fase se elabora:

- Un modelo de Casos de Uso con todos los actores identificados y la mayor parte de las descripciones de estos.
- Requerimientos adicionales: no funcionales o pseudo requerimientos.
- Descripción de la arquitectura del software.
- Prototipo ejecutable de arquitectura.
- Una lista revisada de riesgos.
- Plan del proyecto, incluyendo iteraciones y criterios de evaluación para cada iteración.
- Manual preliminar de usuario.

En esta fase se especifican los requerimientos y se describe cómo se van a implementar en el sistema, esto es, transformar los requisitos al diseño del sistema, se desarrolla la arquitectura para el sistema, y se adapta el diseño para que sea consistente con el entorno de implementación.

#### **4.8. FASE DE CONSTRUCCIÓN**

Se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y demás. El resultado final es un sistema ejecutable.

Para ello se realiza lo siguiente:

- Se integra el producto de software sobre la plataforma adecuada.
- Se elaboran los manuales de usuario.
- Se dé una descripción de la versión actual.
- Se planifica qué subsistemas deben ser implementados y en qué orden deben ser integrados, formando el Plan de Integración.
- Cada implementador decide en qué orden implementa los elementos del subsistema.
- Si se encuentra errores de diseño, se notifica.
- Se integra el sistema siguiendo el plan.

En la fase de Pruebas se evalúa la calidad del producto, pero no para aceptar o rechazar el mismo al final del proceso de desarrollo, sino que debe ir integrado en todo el ciclo de vida. Se debe encontrar y documentar defectos en la calidad del software. Generalmente proporciona datos sobre la calidad del software, provee la validación de los supuestos realizados en el diseño y especificación de requisitos por medio de demostraciones concretas, verificar las funciones del producto de software según lo diseñado y que los requisitos tengan su apropiada implementación.

En la parte de despliegue se produce distribuciones del producto y se entrega a los usuarios. Las actividades implicadas incluyen:

- Probar el producto en su entorno de ejecución final.
- Empaquetar el software para su distribución.
- Distribuir el software.
- Instalar el software.
- Proveer asistencia y ayuda a los usuarios.
- Formar a los usuarios y al cuerpo de ventas.
- Migrar el software existente o convertir bases de datos.

Durante todo el proyecto se ejecutan las fases de gestión del proyecto, donde se vigila el cumplimiento de los objetivos, gestión de riesgos y restricciones para desarrollar un producto que sea acorde a los requisitos de los clientes y los usuarios.

En esta etapa se realizan las siguientes tareas:

- Proveer un marco de trabajo para la gestión de proyectos de software intensivos.
- Proveer guías prácticas realizar planeación, contratar personal, ejecutar y monitorear el proyecto.
- Proveer un marco de trabajo para gestionar riesgos.

La fase de configuración y control de cambios, permite mantener la integridad de todas las actividades que se crean en el proceso, así como de mantener información del proceso evolutivo que han seguido.

En la fase de Entorno, la finalidad es dar soporte al proyecto con las adecuadas herramientas, procesos y métodos. Brinda una especificación de las herramientas que se van a necesitar en cada momento, así como definir la instancia concreta del proceso que se va a seguir.

En concreto las responsabilidades de este flujo de trabajo incluyen:

- Selección y adquisición de herramientas
- Establecer y configurar las herramientas para que se ajusten a la organización.
- Configuración del proceso.
- Mejora del proceso.
- Servicios técnicos.

## **4.9. ROLES QUE SE CUMPLEN EN EL RUP.**

### **4.9.1. ANALISTAS**

- Analista de procesos de negocio.
- Diseñador del negocio.
- Analista de sistema.
- Especificador de requisitos.

### **4.9.2. DESARROLLADORES**

- Arquitecto de software.
- Diseñador
- Diseñador de interfaz de usuario
- Diseñador de cápsulas.
- Diseñador de base de datos.
- Implementador.
- Integrador.

### **4.9.3. GESTORES**

- Jefe de proyecto
- Jefe de control de cambios.
- Jefe de configuración.
- Jefe de pruebas
- Jefe de despliegue
- Ingeniero de procesos
- Revisor de gestión del proyecto
- Gestor de pruebas.

### **4.9.4. APOYO**

- Documentador técnico
- Administrador de sistema
- Especialista en herramientas



- Desarrollador de cursos
- Artista gráfico

#### **4.9.5. ESPECIALISTA EN PRUEBAS**

- Especialista en Pruebas (tester)
- Analista de pruebas
- Diseñador de pruebas

#### **4.9.6. OTROS ROLES**

- Stakeholders.
- Revisor
- Coordinación de revisiones
- Revisor técnico
- Cualquier rol
- Para grandes organizaciones con un número de equipos de ingenieros y la comunicación entre cada equipo es crítica por lo tanto es necesario que los artefactos sean completos y bastante comprensivos
- En tanto que para pequeños proyectos no es recomendable presentarse tanto rigor en las preparaciones de los artefactos, la eficiencia del proceso depende más de las habilidades de cada trabajador.

#### **4.9.7. BENEFICIOS DE LA METODOLOGÍA ORIENTADA A OBJETOS.**

- Promueve la reusabilidad.
- Reduce la complejidad del mantenimiento (extensibilidad y facilidad de cambios).
- Riqueza semántica.
- Disminuye la brecha semántica entre la visión interna y la visión externa del sistema.
- Facilita la construcción de prototipos.

#### **4.9.8. VENTAJAS DE LA METODOLOGÍA ORIENTADA A OBJETOS.**

- Reutilización
- El diseñador piensa en términos del comportamiento de objetos y no en detalles de bajo nivel
- Confiabilidad, Integridad y Estabilidad.
- Mantenimiento más sencillo. Modificaciones locales.
- Modelado más realista.
- Modelos empresariales inteligentes.
- Independencia del diseño.
- Mejores herramientas CASE.
- Bibliotecas de clases para las empresas.
- Se construyen clases cada vez más complejas.
- Nuevos mercados para el software.
- Diseño de mayor calidad.
- Programación más sencilla.
- Mejor comunicación entre los profesionales de los Sistemas de Información y los empresarios.

- Mayor nivel de automatización de las bases de datos.
- La comprensión del sistema es más fácil porque la semántica entre el sistema y la realidad son similares.

#### **4.10 JUSTIFICACIÓN DE LA METODOLOGÍA RUP.**

En el Desarrollo de un Sistema de Información para llevar el Registro y Control de las Víctimas y Afectados por Accidentes de Tránsito y las Denuncias sobre el estado de la Infraestructura Vial para la Asociación Civil ASOTRANSITO, se utilizara la metodología RUP. Esta metodología se encuentra dividida en 4 fases: fase de inicio, fase de elaboración, fase de construcción y fase de transiciones. En la fase de inicio identificaremos los principales caso de uso, los riesgos y el alcance del proyecto. En la fase de elaboración se realizara el plan de proyecto, competición de los casos de uso y se mitigan riesgos. En la fase construcción se concreta la elaboración del sistema totalmente operativo y eficiente, y la realización del manual de usuarios, en la fase de transición se entregara el sistema al usuario haciendo la respectiva instalación del mismo con el fin que el usuario pueda realizar las pruebas correspondientes.

### **5. Bibliografía**

- Brink, T.; Gergle D.; Wood, S.D.: Usability for the Web: M- Kauf. 2002
- Bruegge B., Dutoit A.H. Ingeniería de Software Orientado a Objetos, Prentice Hall – Pearson educación, México, 2002.
- Pressman R.S. Ingeniería del Software. Un enfoque práctico (5ª ed.) Mc Graw- Hill; New York , 2001.
- Rumbaugh J., Jacobson I., Booch G. El Lenguaje Unificado de Modelado. Manual de Referencia, Addison-Wesley, Madrid, 2000.
- Sommerville I. Ingeniería de software, 6ª edición, Prentice Hall – Pearson educación, México, 2002.
- Stevens P., Pooley R. Utilización de UML en Ingeniería del Software con Objetos y Componentes, Addison Wesley, Madrid, 2002.
- Silva, Andrés; Mercerat, Bárbara; Construyendo aplicaciones web con una metodología de diseño orientada a objetos. Internet.