**Sistema Antiplagium**

**Documento de Arquitectura de SW**

**Versión 3.0**

**Historia de Revisión**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 30/03/10 | 1.0 | Versión inicial | Kim Alvarado / Patricia Natividad |
| 08/04/10 | 2.0 | Versión corregida | Kim Alvarado |
|  |  |  |  |
|  |  |  |  |

Tabla de Contenido

1. Introducción 5

1.1. Propósito 5

1.2. Alcance 5

1.3. Definiciones, acrónimos y abreviaturas 5

1.3.1. Definiciones 5

1.3.2. Acrónimos 5

1.4. Referencias 6

1.5. Visión General del documento 6

2. Representación de la arquitectura 7

3. Metas y Restricciones de la arquitectura 7

3.1. Metas 7

3.2. Restricciones del Sistema 7

4. Vista y Realizaciones de Casos de Uso 8

5. Vista Lógica 13

5.1 Descripción 13

5.2 Diseño en Capas 15

5.2.1 Capa Interfaz de Usuario 16

5.2.2 Capa Lógica del Negocio 16

5.2.3 Capa Persistencia 16

6. Vista de Proceso 16

6.1 Diagrama de clases 16

7. Vista de Despliegue 18

7.1 Servidor 19

7.2 PC Cliente 19

8. Vista de Implementación 20

8.1 Descripción 20

8.2 Niveles o Capas 20

8.2.1 Capa GUI 20

8.2.2 Capa Lógica del Negocio 20

8.2.3 Capa Base de Datos 20

9. Diagramas de Secuencia 21

9.1 Paquete de Administración 21

9.1.1 Registrar Usuario 21

9.2 Buscar Usuario 21

9.2.1 Eliminar Usuario 22

9.2.2 Reporte de Acciones 22

9.2.3 Registrar Cuenta 23

9.3 Paquete de Gestión de Pagos 24

9.3.1 Registrar Documento de Pago 24

9.3.2 Anular Documento de Pago 24

9.3.3 Emitir Documentos de Pago en Lote 25

9.3.4 Buscar Documento de Pago 25

9.3.5 Registrar Pago 26

9.3.6 Registrar Descuento 26

9.3.7 Generar Reporte de Historial de Pago de Cliente 27

9.3.8 Generar Reporte de Historial de Pago por Periodo 27

9.5 Paquete de Cliente 31

9.5.1 Registrar Cliente 31

9.5.2 Modificar Cliente 31

9.5.3 Buscar Cliente 32

9.5.4 Registrar Contrato 32

9.5.5 Modificar Contrato 33

9.5.6 Buscar Contrato 33

10. Tamaño y performance 36

11. Calidad 37

11.1 Usabilidad 37

11.2 Eficiencia 37

11.3 Seguridad 37

11.4 Confiabilidad 37

11.5 Mantenimiento 37

11.6 Estándares: 37

Documento de Arquitectura de Software

# Introducción

Uno de los desarrollos más importantes dentro de la construcción del software es el desarrollo de la arquitectura de software, que permite representar la estructura del sistema, sirviendo de comunicación entre las personas involucradas en el desarrollo, además de ayudar a realizar el análisis para la toma de decisiones.

Este documento provee al usuario especializado una vista de la arquitectura del Sistema Antiplagium.

La plantilla de este documento se basó en las especificaciones de RUP (Rational Unified Process) para el documento de arquitectura de software.

## Propósito

Este documento proporciona una descripción de la arquitectura del sistema, haciendo uso de diversas visiones arquitectónicas para representar diversos aspectos del sistema. Se realiza con el fin de documentar las decisiones de arquitectura significativas que se han tomado en el sistema.

## Alcance

Este documento presenta la arquitectura en la cual estará basado el Sistema para detección de plago Antiplagium, definiendo de manera detallada la distribución de los paquetes del sistema en las diversas capas que éste presenta, así como una descripción de las capas a utilizar.

## Definiciones, acrónimos y abreviaturas

Se brindan definiciones y acrónimos de términos usados en el presente documento que necesiten de alguna explicación para su correcta interpretación.

### Definiciones

|  |  |
| --- | --- |
| Paquetes | Agrupaciones de casos de uso y actores por funcionalidad que proveen. |
| Actor | Alguien o algo externo al sistema que interactúa con él. |
| Caso de Uso | Secuencia de acciones que el sistema realiza, la cual proporciona un resultado de valor observable. |
| Star UML | Se refiere a las herramientas que permiten realizar el modelado de los diagramas presentados en este documento. |

### Acrónimos

|  |  |
| --- | --- |
| ERS | Especificación de Requisitos de Software |
| RUP | Rational Unified Process |
| UML | Unified Modeling Language |

## Referencias

Las referencias aplicables son:

1. Documento de Especificación de Requisitos del Sistema Antiplagium.
2. Documento de Visión del Sistema Antiplagium.
3. Plan de Proyecto del Sistema Antiplagium.

## 

## Visión General del documento

En las diez secciones de este documento se detalla la arquitectura del software a desarrollar. Para ello se presenta de manera clara el caso de uso que mas representa la arquitectura del sistema, empleando un lenguaje sencillo y directo, así como gráficos y vistas de acuerdo a la metodología utilizada.

# Representación de la arquitectura

La Arquitectura a utilizar será Cliente-Servidor. Se desarrollará una sola aplicación integrada, en la que solo se permitirá el acceso a los usuarios registrados en el sistema y a las tareas a las cuales tengan acceso autorizado. Se empleará un solo servidor centralizado.

La arquitectura se muestra como una serie de vistas, utilizando la aplicación Star UML.

La arquitectura se basará en el modelo ‘4+1’, que contendrá las vistas de Lógica, Implementación, Procesos e Implantación, con la vista central de caso de uso.

# Metas y Restricciones de la arquitectura

La meta principal de la arquitectura del sistema es mostrar los aspectos principales que influirán en la etapa de desarrollo.

Se tomarán en cuenta las siguientes metas y restricciones para el diseño de la arquitectura del sistema:

## Metas

1. El Sistema Antiplagium permitirá a los usuarios acceder al sistema desde cualquier Terminal.
2. Para poder acceder al Sistema Antiplagium, se requiere de un código de usuario válido así como de una contraseña. Además, dependiendo del perfil del usuario se deshabilitarán opciones de manejo del Sistema para proteger información confidencial.
3. Los requerimientos estipulados en el Documento de Visión, deben de ser considerados como parte de la arquitectura del sistema a implementar.
4. El uso del sistema para los trabajadores de la Institución Educativa se limita a dentro de la organización. Fuera de ella no se podrá acceder al sistema.

## Restricciones del Sistema

1. El sistema usará PostgreSql como motor de Base de Datos necesariamente. Debe considerarse obligatorio la elaboración de una copia de rescate de las tablas para no afectar la información almacenada antes de poner en total funcionamiento al sistema.
2. Las características técnicas de las computadoras que serán utilizadas no deberán presentar potencias menores a las brindadas por una Pentium 4, con al menos 1 GB de RAM y 500 MB de espacio libre en el disco, con Sistema Operativo Linux.
3. Las computadoras que utilicen el sistema deberán de tener instalado el JRE 1.6 o superior ya que el mismo será desarrollado en java.

# Vista y Realizaciones de Casos de Uso

El diagrama de casos de uso representa la forma de cómo el usuario interactúa con el sistema a desarrollar, así podremos obtener el conjunto de escenarios que contendrá cada iteración.

Cada caso de uso tiene una descripción que especifica la funcionalidad que se incorporará al sistema propuesto.

En este apartado se mostraran los casos de uso del Sistema.

#### Paquete de Usuarios

**Diagrama 1. Paquete de Usuarios**

**CU\_PU001***.*

#### Paquete de Documentos



**Diagrama 2. Paquete de Documentos**

##### CU\_PDOC001Buscar Documentos

Este caso de uso sirve para consultar por los documentos ingresados en la base de datos.

##### CU\_PDOC002Gestionar Documentos

Este caso de uso permite subir y eliminar documentos de la base de datos.

##### CU\_PDOC003Buscar categoría de Documentos

Este caso de uso permite consultar las categorías existentes para clasificar los documentos.

##### CU\_PDOC004Gestionar categorías de documentos

Este caso de uso permite registrar, modificar y eliminar categorías de clasificación de documentos.

#### Paquete de Detección



**Diagrama 3. Paquete de Detección**

**CU\_PDET001Comparar Documentos**

Este caso de uso permite realizar la comparación para detección de plagio entre documentos (esto implica la implementación de un algoritmo adecuado).

**CU\_PDET002Mostrar resultados de comparación**

Este caso de uso permite mostrar los resultados de comparaciones pasadas que se hayan realizado entre archivos y en un determinado rango de fechas.

# Vista Lógica

## Descripción

La vista lógica del sistema Antiplagium está compuesta de cuatro paquetes principales: Interfaz de usuario (GUI), servicios del Lógica (Logic), objetos de procesamiento (Entity) y los objetos de persistencia (Data Access Objects).

El primer paquete, Interfaz de usuario, contiene las clases visuales de la aplicación, las cuales crean las ventanas que utiliza el usuario para comunicarse con el sistema para una adecuada y fácil utilización de los paquetes que se encuentran en las siguientes capas.



El paquete Lógica contiene las clases que permiten llevar a cabo los servicios de procesamiento, como son: Controlar el mantenimiento de usuarios, Mostrar resultados de comparaciones, etc.

****

El paquete Entidades contiene las clases que utilizará el sistema Antiplagium, entre las principales clases se tiene: Usuario, Documento, Detector, Provilegio, etc.

****

El paquete de Objetos de Acceso de Datos contiene las clases que el sistema utiliza para manejar la persistencia de los objetos con la Base de Datos.



## Diseño en Capas

En la siguiente figura podemos observar la distribución de los paquetes en cada una de las capas del sistema.



Figura .5.1: Diagrama de Distribución de Capas del Sistema

### Capa Interfaz de Usuario

Esta capa contiene el paquete Interfaz de Usuario, el cual almacena todas las clases con las cuales el usuario puede interactuar como lo son las ventanas.

### Capa Lógica del Negocio

Esta capa contiene los paquetes de Servicios de Negocio y Entidades de Negocio. Contiene la lógica para el manejo de las operaciones del negocio.

### Capa Persistencia

Esta capa contiene el paquete de Objetos de Acceso de Datos, que brinda una interfaz transparente para la interacción con la Base de Datos.

# Vista de Proceso

## Diagrama de clases

Es el diagrama principal para el análisis y diseño. En este diagrama se presentan las clases del sistema con sus relaciones estructurales y de herencia. Para establecer las clases, objetos atributos y operaciones entre clases se tomó como base a los diagramas de casos de uso.

Figura 6.1: Vista del Diagrama de Clases



# Vista de Despliegue

El presente Diagrama de Despliegue muestra de manera gráfica los nodos que conforman el sistema, además los describe indicando la localización de las tareas en los nodos físicos. Cada nodo representa un recurso de ejecución como una computadora o servidor y se encuentra conectado con otros mediante un enlace de comunicación. En este caso, la red será de tipo LAN.



Figura 7.1: Diagrama de Despliegue del Sistema. Servidor (Base de Datos) y Cliente (Computadora usuario).



Figura 7.2: Diagrama Componentes del Sistema.

## Servidor

El servidor principal donde se encuentra la Base de Datos. Los diferentes usuarios tendrán acceso al servidor vía LAN.

## PC Cliente

Son las computadoras que son utilizadas por cada uno de los usuarios de la empresa para acceder al sistema. Están conectadas directamente al servidor principal vía LAN.

# Vista de Implementación

## Descripción

La vista de implementación seguirá el patrón de diseño MVC (Model View Controller). Este patrón plantea la separación del problema en tres capas: la capa model, la capa controller y la capa view.

## Niveles o Capas

### Capa GUI

El componente GUI maneja las clases que permiten la interacción del usuario con el sistema.

### Capa Lógica

Este componente permite manejar la lógica del negocio de todas las áreas involucradas.

* Administración del Sistema
* Usuarios
* Privilegios
* Comparaciones
* Registro de documentos

### Capa Base de Datos

Este componente contiene toda la información que maneja el Sistema.



Figura 8.1: Diagrama de Implementación del Sistema.

# Diagramas de Secuencia

## Paquete de Usuarios

**Gestionar Categoría**

****

**Gestionar Documentos**

****

## Paquete de Detección

**Comparar Documentos**



**Mostrar resultados comparación**



# Tamaño y performance

La arquitectura de software elegida apoya los requisitos dominantes de capacidad en disco y latencia, según lo estipulado en el Documento de Especificación de Requisitos de Software de Antiplagium y el Documento de Visión del Proyecto del Sistema Antiplagium.

1. **Tiempo de respuesta en el acceso a la Base de Datos**

El sistema proveerá accesos a la base de datos con un tiempo de respuesta no mayor a los 5 segundos.

1. **Tiempo de respuesta de transacciones**

El programa no demorará más de 15 minutos en generar reporte de comparación de documentos haciendo uso del algoritmo elegido y guardándolo en base de datos.

1. **Espacio en disco para el cliente**

El espacio en disco necesario para la parte del cliente deberá tener como mínimo 500 MB de espacio libre para su correcto funcionamiento, esto incluye tanto el tamaño del software como el JRE 1.6.

1. **Espacio en disco para el servidor de Base de datos**

El espacio en disco necesario para la parte del Servidor de Base de datos deberá tener como mínimo 1 GB libres en disco para su correcto funcionamiento.

La arquitectura seleccionada apoya los requisitos de latencia y capacidad en disco en la puesta en práctica de una arquitectura cliente servidor. La porción del cliente solo se pone en ejecución en las PC locales de los distintos ambientes dentro de la empresa. Los componentes se han diseñado para asegurarse de obtener requisitos mínimos de disco y memoria en el lado de las PC del cliente.

# Calidad

Para un mejor aprovechamiento de la arquitectura de software se dan los siguientes requerimientos de calidad:

## Usabilidad

El sistema permitirá un manejo intuitivo por parte de los usuarios.

## Eficiencia

El programa no demorará más de 15 minutos en generar un reporte de comparación de documentos.

## Seguridad

El sistema permitirá el acceso a funcionalidades dependiendo del perfil del usuario que ingresa al sistema, validando su ingreso a través de una clave.

## Confiabilidad

El sistema tendrá en cuenta que la información ingresada en él sea válida, para lo cual mostrará mensajes que expliquen al usuario acerca de los errores que éste pudiera cometer y de aquellos que pueda cometer el mismo sistema.

## Mantenimiento

El sistema será flexible, facilitando su mantenimiento futuro.

## Estándares:

Se usará un estándar para todas las ventanas e interfaces con el usuario que tenga el Sistema Antiplagium, así como los nombres, códigos, etc.