****

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

**Proyecto *Digital PDF Solutions***

Curso: *ARQUITECTURA DE SOFTWARE*

Docente: *Ing. Patrick Cuadros Quiroga*

Integrantes:

***Ayma Choque, Erick Yoel (2021072616)***

***Flores Ramos, Mario Antonio (2018000597)***

***Poma Machicado, Fabiola Estefani (2021070030)***

***Tapia Vargas, Dylan Yariet (2021072630)***

**Tacna – Perú**

***2024***

| CONTROL DE VERSIONES | | | | | |
| --- | --- | --- | --- | --- | --- |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | MPV | ELV | ARV | 10/10/2020 | Versión Original |

Sistema *Digital PDF Solutions*

Documento de Arquitectura de Software

Versión *{1.0}*

| CONTROL DE VERSIONES | | | | | |
| --- | --- | --- | --- | --- | --- |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | MPV | ELV | ARV |  | Versión Original |

ÍNDICE GENERAL

[**1. INTRODUCCIÓN 4**](#_1fob9te)

[1.1. Propósito (Diagrama 4+1) 4](#_3znysh7)

[1.2. Alcance 5](#_2et92p0)

[1.3. Definición, siglas y abreviaturas 5](#_tyjcwt)

[1.4. Organización del documento 6](#_3dy6vkm)

[**2. OBJETIVOS Y RESTRICCIONES ARQUITECTÓNICAS 6**](#_1t3h5sf)

[2.1.1. Requerimientos de prioridad 6](#_yju87xd1xqsi)

[2.1.2. Requerimientos No Funcionales 7](#_7m94d96boc3s)

[2.1.3 Requerimientos No Funcionales – Atributos de Calidad 8](#_17dp8vu)

[2.2. Restricciones 8](#_3fvmfpriyj6w)

[**3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA 9**](#_3rdcrjn)

[3.1. Vista de Caso de uso 9](#_lnxbz9)

[3.1.1. Diagramas de Casos de uso 9](#_7mwh3zy3j6sl)

[3.2. Vista Lógica 10](#_1ksv4uv)

[3.2.1. Diagrama de Subsistemas (paquetes) 10](#_2jxsxqh)

[3.2.2. Diagrama de Secuencia (vista de diseño) 11](#_z337ya)

[3.2.3. Diagrama de Colaboración (vista de diseño) 12](#_3j2qqm3)

[Escenario: Fusión de PDF 12](#_vadi014vi22n)

[3.2.4. Diagrama de Objetos 14](#_1y810tw)

[3.2.5. Diagrama de Clases 15](#_4i7ojhp)

[3.2.6. Diagrama de Base de datos (relacional o no relacional) 15](#_2xcytpi)

[3.3. Vista de Implementación (vista de desarrollo) 16](#_1ci93xb)

[3.3.1. Diagrama de arquitectura software (paquetes) 19](#_3whwml4)

[3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes) 21](#_2bn6wsx)

[**3.4. Vista de procesos 23**](#_qsh70q)

[3.4.1. Diagrama de Procesos del sistema (diagrama de actividad) 23](#_3as4poj)

[3.5. Vista de Despliegue (vista física) 24](#_1pxezwc)

[3.5.1. Diagrama de despliegue 25](#_49x2ik5)

[**4. ATRIBUTOS DE CALIDAD DEL SOFTWARE 26**](#_2p2csry)

[Escenario de Funcionalidad: 26](#_6rskvuiizicn)

[Escenario de Usabilidad: 26](#_jpodf44us7db)

[Escenario de Confiabilidad: 26](#_ks63zxgckear)

[Escenario de Rendimiento: 27](#_a9en8xbcxvj2)

[Escenario de Mantenibilidad: 27](#_tyheze5blq50)

[Otros Escenarios 27](#_qeeysyzh1euv)

# **INTRODUCCIÓN**

## **Propósito (Diagrama 4+1)**

A continuación se presenta una descripción de cada una de las vistas del modelo 4+1 aplicado a nuestro sistema.

Vista Lógica

* **Descripción**: Muestra la estructura del sistema en términos de sus componentes principales y sus interacciones. Utiliza el patrón MVC.
* **Componentes Principales**:
  + **Modelos**: Representan los datos y la lógica de negocio (Usuario, Suscripción, OperaciónPDF).
  + **Vistas**: Páginas JSP que presentan la interfaz de usuario.
  + **Controladores**: Servlets que manejan las solicitudes del usuario y coordinan las interacciones entre el modelo y la vista.

Vista de Desarrollo

* **Descripción**: Muestra la estructura del sistema desde la perspectiva del desarrollo, enfocándose en la organización de los paquetes y módulos de código.
* **Paquetes Principales**:
  + **Modelo**: Contiene las clases de negocio y de acceso a datos.
  + **Vista**: Contiene las páginas JSP.
  + **Controlador**: Contiene los servlets que actúan como controladores.

Vista de Procesos

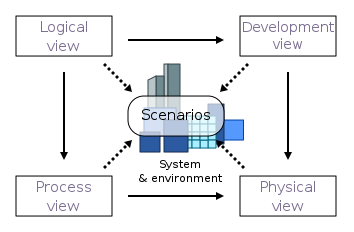
* **Descripción**: Muestra los procesos del sistema y cómo interactúan entre ellos. Incluye los procesos de usuario y los procesos de backend para la gestión de archivos PDF.
* **Procesos Clave**:
  + **Proceso de Autenticación**: Maneja el registro e inicio de sesión de usuarios.
  + **Proceso de Gestión de Suscripciones**: Maneja la compra y verificación de suscripciones.
  + **Proceso de Operaciones PDF**: Maneja la fusión y corte de archivos PDF.

Vista Física

* **Descripción**: Muestra la distribución física del sistema en términos de hardware y redes, detallando cómo se despliegan los componentes del sistema en los diferentes nodos.
* **Componentes Principales**:
  + **Servidor Web**: Aloja las páginas JSP y los servlets.
  + **Servidor de Base de Datos**: Aloja la base de datos MySQL.
  + **Clientes**: Navegadores web que acceden a la aplicación.

Vista de Casos de Uso

* **Descripción**: Muestra cómo los actores interactúan con el sistema para cumplir con los casos de uso clave.
* **Casos de Uso Clave**:
  + **Registrar Usuario**
  + **Iniciar Sesión**
  + **Gestionar Suscripción**
  + **Operaciones con PDF**



## **Alcance**

El alcance del sistema incluye funcionalidades para fusionar múltiples archivos PDF en un solo documento, así como para recortar secciones específicas de documentos PDF según las necesidades del usuario. Además, el sistema proporcionará una interfaz intuitiva y fácil de usar, permitiendo a los usuarios realizar estas operaciones de manera rápida y sin complicaciones. Sin embargo, el sistema no incluirá funciones avanzadas de edición de PDF, como la creación de formularios o la edición de texto, ya que su enfoque principal está en la fusión y el corte de archivos PDF.

## **Definición, siglas y abreviaturas**

Definiciones:

* Sistema de Fusión y Corte de PDFs: Herramienta de software diseñada para fusionar múltiples archivos PDF en uno solo y recortar secciones específicas de documentos PDF según sea necesario.
* Archivo PDF: Formato de archivo diseñado para presentar e intercambiar documentos de manera fiable, independientemente del software, hardware o sistema operativo utilizado.

Siglas:

* PDF: Portable Document Format (Formato de Documento Portátil).

Abreviaturas:

* Fusión PDF: Fusionar varios archivos PDF en uno solo.
* Corte PDF: Recortar secciones específicas de un documento PDF.

## **Organización del documento**

Se intentó estructurar los diagramas conforme a un estándar de programación. Para la organización del documento, se empleó un modelo de diseño que detalla las funciones a utilizar. Este modelo mantiene la relación entre los diferentes diagramas.

# **OBJETIVOS Y RESTRICCIONES ARQUITECTÓNICAS**

### Requerimientos de prioridad

### 

| ID | Requerimiento Funcional | Descripción | Prioridad |
| --- | --- | --- | --- |
| RF01 | Registrar Usuario | El sistema debe permitir a los usuarios registrarse con su nombre, correo y contraseña. | 2 |
| RF02 | Iniciar Sesión | El sistema debe permitir a los usuarios iniciar sesión utilizando su correo y contraseña. | 3 |
| RF03 | Seleccionar Tipo de suscripción | El sistema debe permitir mostrar los tipos de suscripción. | 5 |
| RF04 | Pagar suscripción | El sistema debe permitir que un usuario registrado pague su suscripción. | 4 |
| RF05 | Ver Boleta de pago | El sistema debe permitir que un usuario registrado vea su boleta de pago | 11 |
| RF06 | Seleccionar Operación a realizar | El sistema debe permitir a los usuarios seleccionar operaciones como cortar y fusionar PDFs | 7 |
| RF07 | Subir Archivo PDF | El sistema debe permitir a los usuarios subir archivos PDF para ser procesados. | 6 |
| RF08 | Configurar Parámetros de Operación | El sistema debe permitir a los usuarios configurar parámetros específicos para cada operación. | 8 |
| RF09 | Descargar el pdf generado | El sistema debe permitir a los usuarios descargar el PDF procesado. | 9 |
| RF10 | Mostrar historial de operaciones | El sistema debe permitir a los usuarios ver el historial de operaciones realizadas. | 10 |
| RNF01 | Seguridad | El sistema debe asegurar que todos los datos del usuario están protegidos mediante encriptación y otras medidas de seguridad. | 1 |
| RNF02 | Rendimiento | El sistema debe ser capaz de procesar y realizar operaciones en PDFs de manera rápida y eficiente, soportando múltiples usuarios simultáneamente. | 12 |
| RNF03 | Usabilidad | El sistema debe ser fácil de usar y comprender, proporcionando una interfaz de usuario intuitiva. | 13 |
| RNF04 | Compatibilidad | El sistema debe ser compatible con los principales navegadores web | 14 |

### **Requerimientos No Funcionales**

| ID | Requerimiento Funcional | Descripción | Prioridad | Estado |
| --- | --- | --- | --- | --- |
| RF01 | Registrar Usuario | El sistema debe permitir a los usuarios registrarse con su nombre, correo y contraseña. | Alta | Finalizado |
| RF02 | Iniciar Sesión | El sistema debe permitir a los usuarios iniciar sesión utilizando su correo y contraseña. | Alta | Finalizado |
| RF03 | Seleccionar Tipo de suscripción | El sistema debe permitir mostrar los tipos de suscripción. | Alta | Finalizado |
| RF04 | Pagar suscripción | El sistema debe permitir que un usuario registrado pague su suscripción. | Alta | Finalizado |
| RF05 | Ver Boleta de pago | El sistema debe permitir que un usuario registrado vea su boleta de pago | Alta | Finalizado |
| RF06 | Seleccionar Operación a realizar | El sistema debe permitir a los usuarios seleccionar operaciones como cortar y fusionar PDFs | Alta | Finalizado |
| RF07 | Subir Archivo PDF | El sistema debe permitir a los usuarios subir archivos PDF para ser procesados. | Media | Finalizado |
| RF08 | Configurar Parámetros de Operación | El sistema debe permitir a los usuarios configurar parámetros específicos para cada operación. | Alta | Finalizado |
| RF09 | Descargar el pdf generado | El sistema debe permitir a los usuarios descargar el PDF procesado. | Alta | Finalizado |
| RF10 | Mostrar historial de operaciones | El sistema debe permitir a los usuarios ver el historial de operaciones realizadas. | Alta | Finalizado |

### 

### **2.1.3 Requerimientos No Funcionales – Atributos de Calidad**

| ID | Requerimiento No Funcional | Descripción | Prioridad |
| --- | --- | --- | --- |
| RNF01 | Seguridad | El sistema debe asegurar que todos los datos del usuario están protegidos mediante encriptación y otras medidas de seguridad. | Alta |
| RNF02 | Rendimiento | El sistema debe ser capaz de procesar y realizar operaciones en PDFs de manera rápida y eficiente, soportando múltiples usuarios simultáneamente. | Alta |
| RNF03 | Usabilidad | El sistema debe ser fácil de usar y comprender, proporcionando una interfaz de usuario intuitiva. | Alta |
| RNF04 | Compatibilidad | El sistema debe ser compatible con los principales navegadores web | Media |

## Restricciones

**Compatibilidad del Sistema Operativo:**

* La aplicación debe ser compatible con Windows 7 o superior.

**Requisitos de Hardware:**

* Procesador**:** Procesador Intel Atom® Z2520 1.2 GHz o uno más rápido.
* Memoria RAM: Mínimo de 1 GB, se recomiendan 4 GB.
* Almacenamiento: 2 GB de espacio en disco duro disponible para su instalación; se requiere espacio libre adicional durante la instalación.

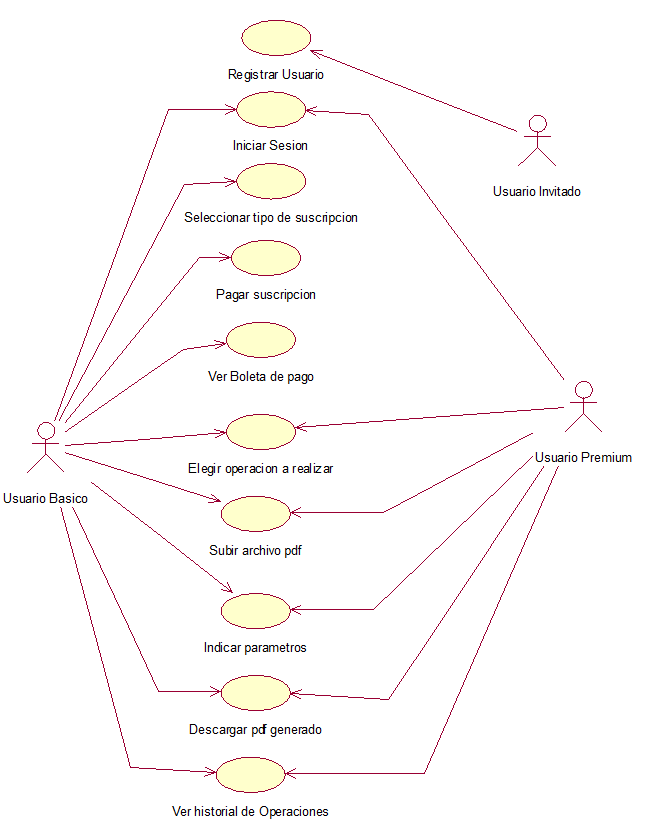
**Acceso a Internet:**

* Esta aplicación ha sido diseñada para trabajar mediante conexión a Internet para funciones de almacenamiento en la nube y servicios en línea.

# **REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA**

## Vista de Caso de uso

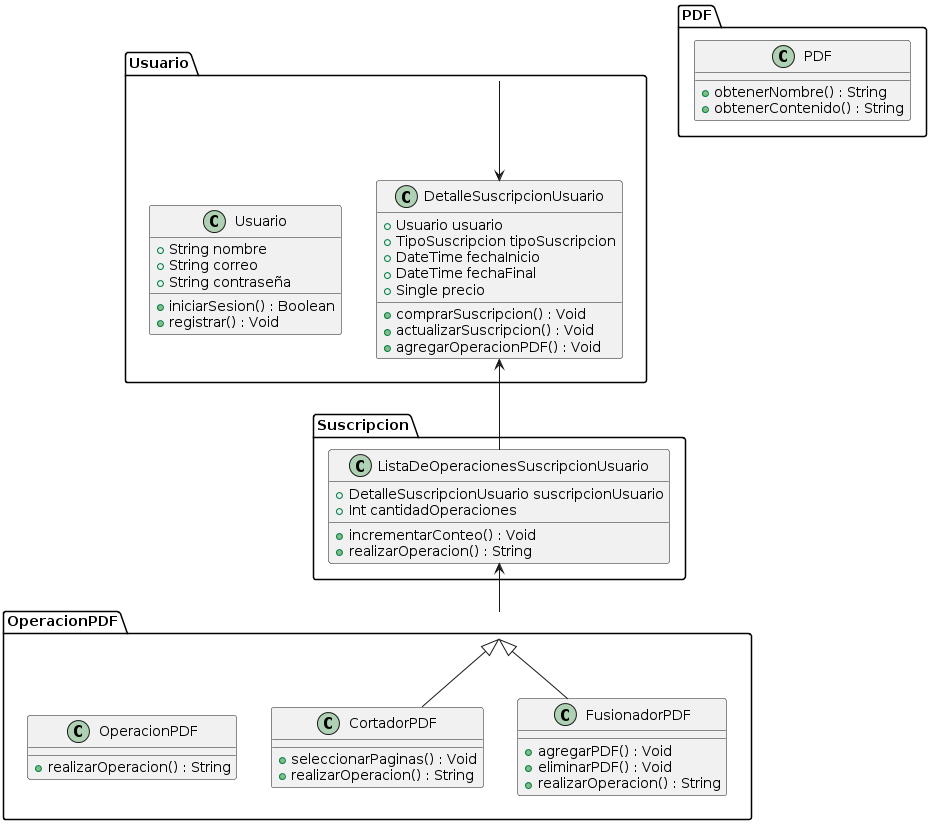
### Diagramas de Casos de uso



## Vista Lógica

### 

### Diagrama de Subsistemas (paquetes)

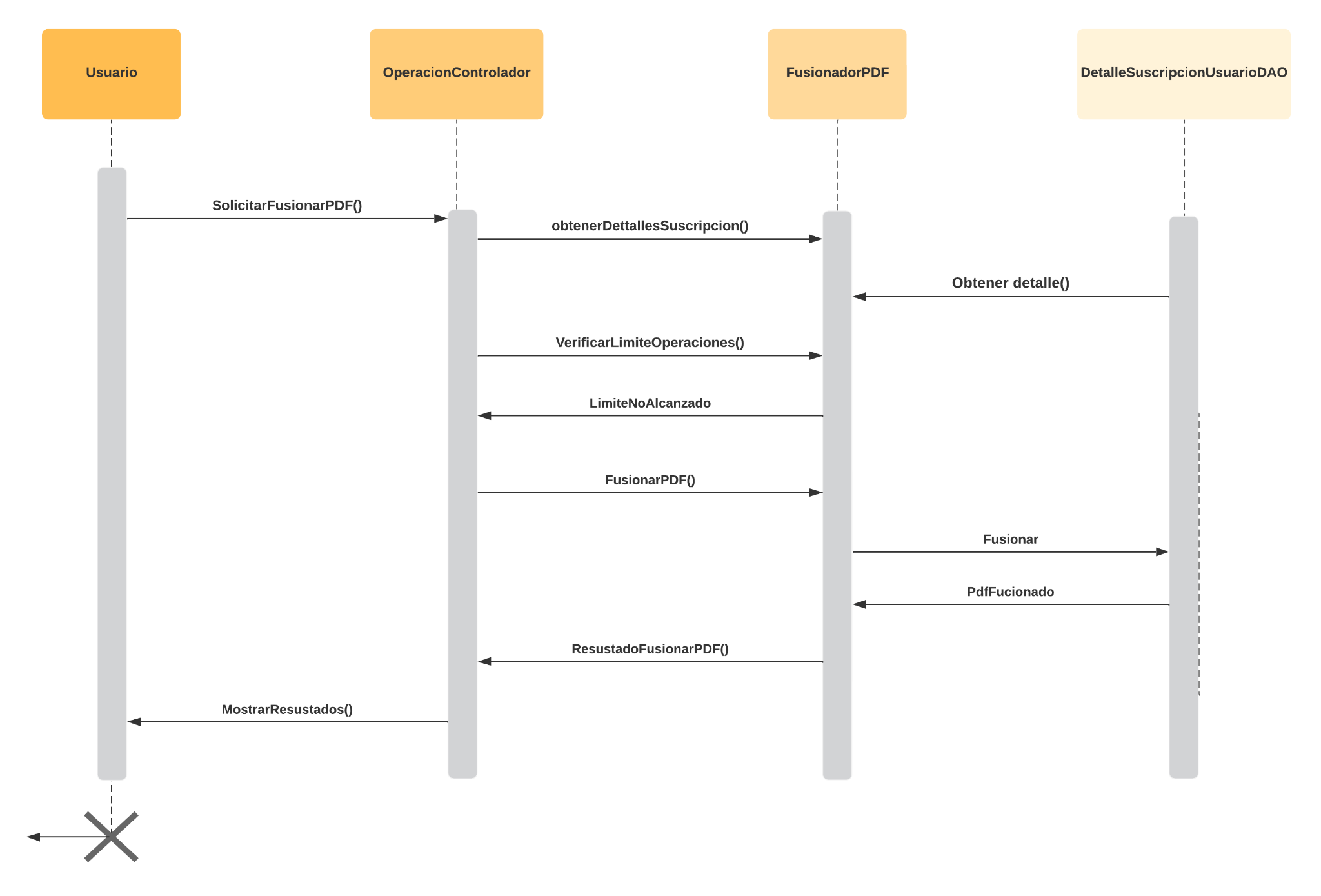


### Diagrama de Secuencia (vista de diseño)

Escenario: Fusión de PDF

1. **Usuario**: Inicia la solicitud de fusión de PDF.
2. **Controlador**: OperacionControlador recibe la solicitud del usuario.
3. **Servicio**: FusionadorPDF maneja la lógica de fusión de PDF.
4. **Modelo**: DetalleSuscripcionUsuario y ListaDeOperacionesSuscripcionUsuario manejan la lógica de suscripciones y el conteo de operaciones.
5. **DAO**: DetalleSuscripcionUsuarioDAO interactúa con la base de datos para actualizar detalles de suscripción.

Diagrama de Secuencia



**Usuario**:

El usuario inicia la solicitud de fusión de PDF.

**Operación Controlador**:

**solicitarFusionarPDF():** Recibe la solicitud del usuario y comienza el proceso de fusión.

**obtenerDetalleSuscripcion():** Consulta a **DetalleSuscripcionUsuarioDAO** para obtener los detalles de la suscripción del usuario.

**verificarLimiteOperaciones():** Verifica si el usuario ha alcanzado el límite de operaciones permitidas.

**fusionarPDFs():** Llama a **FusionadorPDF** para realizar la fusión de los archivos PDF.

**resultadoFusionarPDF():** Procesa el resultado de la operación y devuelve el resultado al usuario.

**Fusionador PDF**:

**fusionar():** Realiza la lógica de fusión de los archivos PDF y devuelve el resultado.

**Detalle Suscripción UsuarioDAO**:

**obtenerDetalle():** Devuelve los detalles de la suscripción del usuario.

### Diagrama de Colaboración (vista de diseño)

El diagrama de colaboración es un tipo de diagrama de interacción que muestra las relaciones entre objetos y su interacción para realizar una funcionalidad específica en el sistema. Este diagrama es útil para entender cómo los objetos colaboran entre sí para cumplir con un requerimiento del sistema.

Vamos a ilustrar un escenario donde un usuario realiza una operación de fusión de PDF. Este proceso involucra varias clases y sus interacciones.

#### Escenario: Fusión de PDF

1. **Usuario**: El usuario inicia la solicitud de fusión de PDF.
2. **Controlador**: OperacionControlador recibe la solicitud del usuario.
3. **Servicio**: FusionadorPDF maneja la lógica de fusión de PDF.
4. **Modelo**: DetalleSuscripcionUsuario y ListaDeOperacionesSuscripcionUsuario manejan la lógica de suscripciones y el conteo de operaciones.
5. **DAO**: DetalleSuscripcionUsuarioDAO interactúa con la base de datos para actualizar detalles de suscripción.

### 

### 

### 

### 

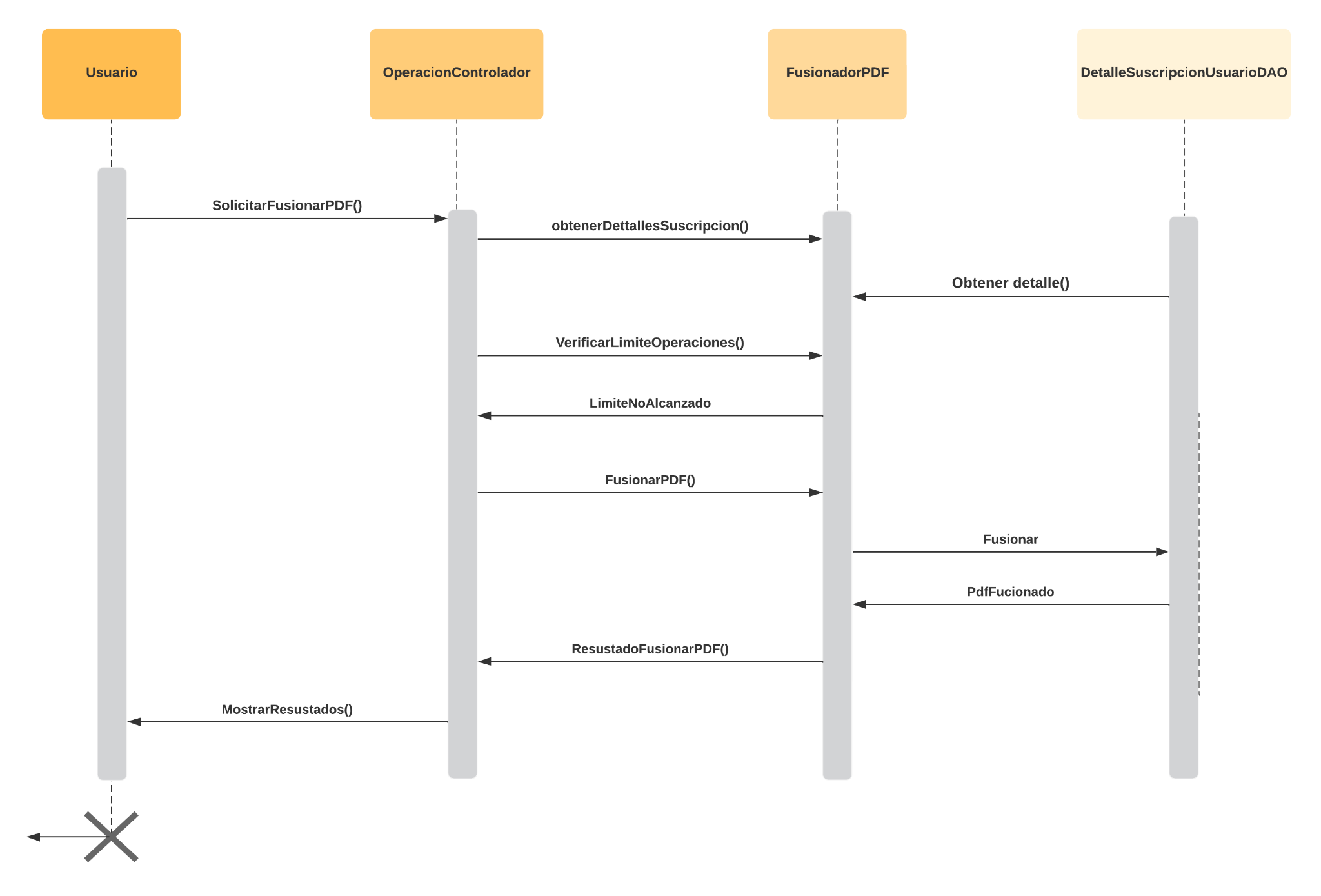
### 

### 

### 

### 

**Diagrama de Colaboración Visual:**



**Descripción del Diagrama de Colaboración:**

**Usuario**: El proceso comienza cuando el usuario inicia la solicitud de fusión de PDF.

**OperacionControlador**: El controlador OperacionControlador recibe la solicitud y maneja la lógica de la operación. Crea un detalle de la operación y delega la lógica específica a FusionadorPDF.

**FusionadorPDF**: El servicio FusionadorPDF realiza la lógica de fusión de los archivos PDF y devuelve el resultado de la operación.

**DetalleSuscripcionUsuarioDAO**: Interactúa con la base de datos para actualizar el número de operaciones realizadas por el usuario y guarda la información de la operación.

**Interacciones**:

El Usuario envía una solicitud de fusión de PDF a OperacionControlador.

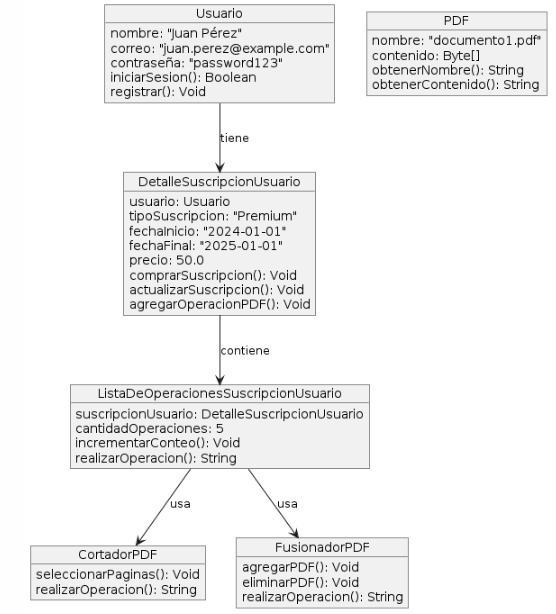
OperacionControlador crea un detalle de operación y llama a FusionadorPDF para realizar la fusión.

FusionadorPDF realiza la operación de fusión y devuelve el resultado al OperacionControlador.

OperacionControlador guarda los detalles de la operación a través de DetalleSuscripcionUsuarioDAO.

OperacionControlador envía el resultado de la operación de vuelta al Usuario.

### Diagrama de Objetos

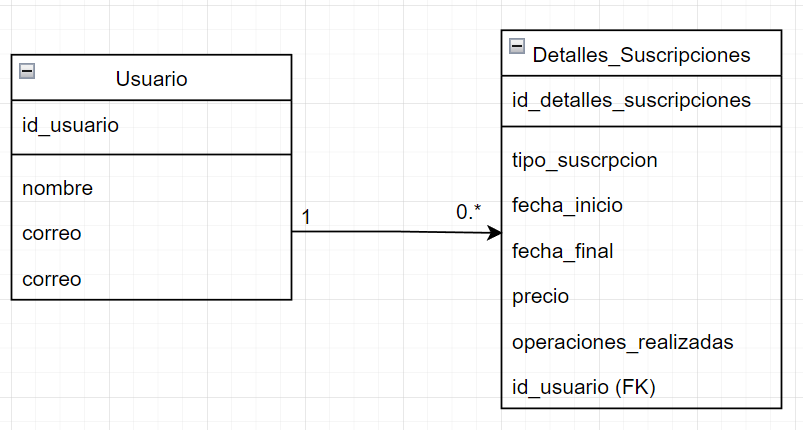


### Diagrama de Clases

Diagrama

Descripción generada automáticamente

### Diagrama de Base de datos (relacional o no relacional)



## Vista de Implementación (vista de desarrollo)

Nuestro proyecto se basa en un patrón de diseño MVC (Modelo-Vista-Controlador) utilizando JSP y Servlets. La estructura general del proyecto se describe de la siguiente manera:

/ProyectoPDF

│

├── /src

│ ├── /Config

│ │ └── Conexion.java

│ ├── /Controlador

│ │ └── UsuarioControlador.java

│ │ └── OperacionControlador.java

│ │ └── DetalleSuscripcionUsuarioControlador.java

│ ├── /Modelo

│ │ └── Usuario.java

│ │ └── DetalleSuscripcionUsuario.java

│ │ └── ListaDeOperacionesSuscripcionUsuario.java

│ ├── /ModeloDAO

│ │ └── UsuarioDAO.java

│ │ └── DetalleSuscripcionUsuarioDAO.java

│ ├── /Servicio

│ │ └── CortadorPDF.java

│ │ └── FusionadorPDF.java

│

├── /web

│ ├── /vista

│ │ └── home.jsp

│ │ └── login.jsp

│ │ └── register.jsp

│ │ └── operaciones.jsp

│ │ └── fusionarPDF.jsp

│ │ └── cortarPDF.jsp

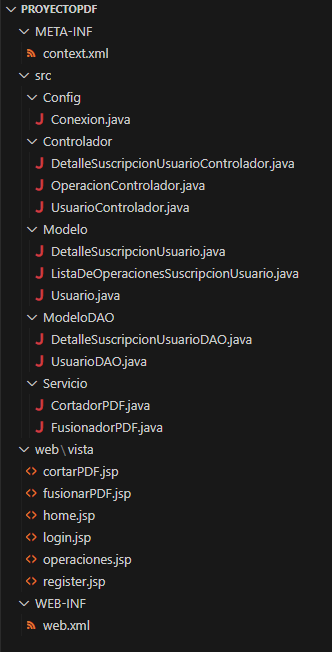
│

├── /META-INF

│ └── context.xml

├── /WEB-INF

│ └── web.xml



**Subsistemas y Paquetes**

**Config**: Este paquete contiene clases de configuración y utilidades necesarias para la conexión con la base de datos.

* + Conexion.java: Clase responsable de manejar la conexión a la base de datos MySQL.

**Controlador**: Este paquete contiene los Servlets que actúan como controladores en el patrón MVC. Manejan las solicitudes del usuario y deciden qué vistas mostrar.

* + UsuarioControlador.java: Controla las acciones relacionadas con el usuario, como registro e inicio de sesión.
  + OperacionControlador.java: Maneja las operaciones relacionadas con PDF, como fusionar y cortar.
  + DetalleSuscripcionUsuarioControlador.java: Maneja las acciones relacionadas con la suscripción del usuario.

**Modelo**: Este paquete contiene las clases de modelo que representan los datos de la aplicación.

* + Usuario.java: Clase que representa a un usuario.
  + DetalleSuscripcionUsuario.java: Clase que representa los detalles de suscripción de un usuario.
  + ListaDeOperacionesSuscripcionUsuario.java: Clase que gestiona las operaciones que un usuario puede realizar según su suscripción.

**ModeloDAO**: Este paquete contiene las clases DAO (Data Access Object) que se encargan de interactuar con la base de datos.

* + UsuarioDAO.java: Clase que maneja las operaciones de base de datos relacionadas con los usuarios.
  + DetalleSuscripcionUsuarioDAO.java: Clase que maneja las operaciones de base de datos relacionadas con los detalles de suscripción de los usuarios.

**Servicio**: Este paquete contiene las clases de servicio que realizan operaciones específicas sobre los datos.

* + CortadorPDF.java: Clase que proporciona la funcionalidad para cortar PDFs.
  + FusionadorPDF.java: Clase que proporciona la funcionalidad para fusionar PDFs.

**Detalles de Mapeo de Clases a Componentes de Implementación**

**Controladores (Servlets)**

* UsuarioControlador.java
  + Maneja solicitudes para registro, inicio de sesión y cierre de sesión.
  + Mapea a las vistas login.jsp, register.jsp, y home.jsp.
* OperacionControlador.java
  + Maneja solicitudes para operaciones de fusión y corte de PDF.
  + Mapea a las vistas fusionarPDF.jsp, cortarPDF.jsp, y operaciones.jsp.
* DetalleSuscripcionUsuarioControlador.java
  + Maneja solicitudes para la compra de suscripción premium.
  + Mapea a la vista comprarPremium.jsp.

**Modelos**

* Usuario.java
  + Representa la entidad Usuario con atributos como id, nombre, correo, contraseña.
* DetalleSuscripcionUsuario.java
  + Representa los detalles de la suscripción de un usuario con atributos como id, usuario, tipoSuscripcion, fechaInicio, fechaFinal, precio, operacionesRealizadas.
* ListaDeOperacionesSuscripcionUsuario.java
  + Gestiona la lógica de conteo de operaciones según la suscripción del usuario.

**DAOs**

* UsuarioDAO.java
  + Contiene métodos para registrar un usuario y verificar sus credenciales de inicio de sesión.
* DetalleSuscripcionUsuarioDAO.java
  + Contiene métodos para agregar, obtener y actualizar los detalles de la suscripción de un usuario.

**Servicios**

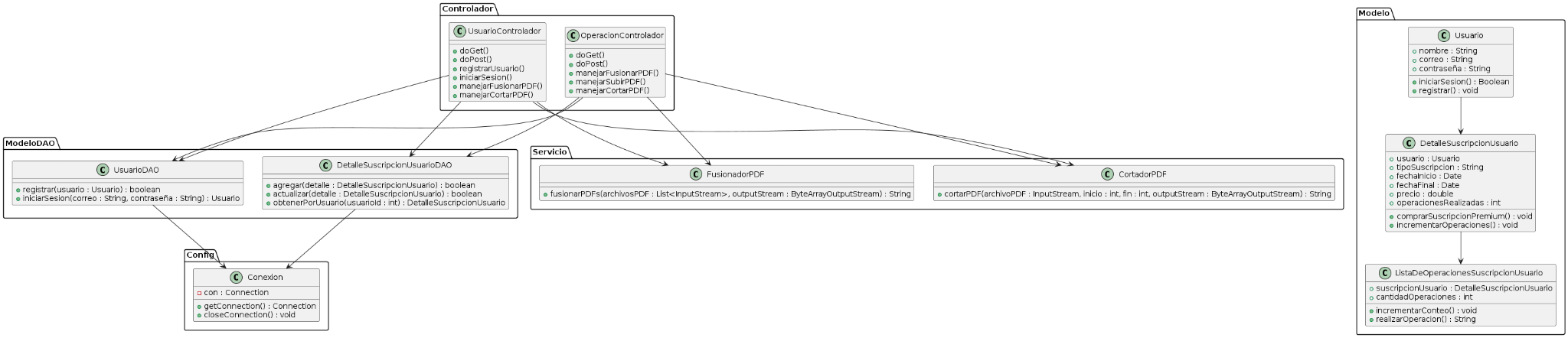
* CortadorPDF.java
  + Proporciona métodos para cortar un archivo PDF en un rango de páginas especificado.
* FusionadorPDF.java
  + Proporciona métodos para fusionar múltiples archivos PDF en uno solo.

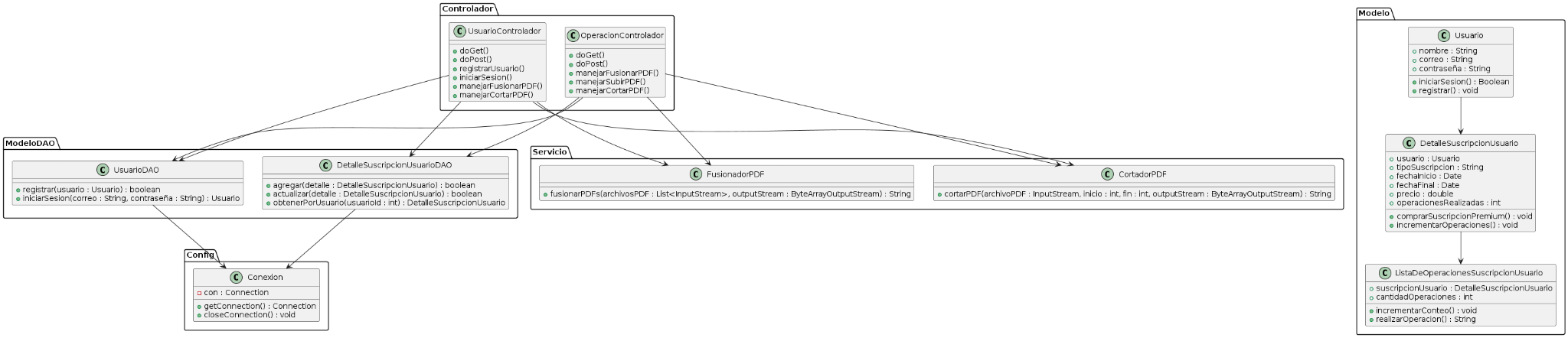
### Diagrama de arquitectura software (paquetes)

Este sistema sigue el patrón de diseño MVC (Modelo-Vista-Controlador), que se organiza en tres capas principales:

1. **Capa de Presentación (Vista)**:
   * Contiene las páginas JSP que los usuarios interactúan con.
   * Se encuentra en el paquete /web/vista.
2. **Capa de Controlador**:
   * Contiene los servlets que manejan las solicitudes HTTP, controlan la lógica de aplicación y deciden qué vistas mostrar.
   * Se encuentra en el paquete /Controlador.
3. **Capa de Modelo**:
   * Contiene las clases que representan los datos de la aplicación.
   * Se encuentra en el paquete /Modelo.
4. **Capa de Acceso a Datos (DAO)**:
   * Contiene las clases que interactúan con la base de datos para realizar operaciones CRUD.
   * Se encuentra en el paquete /ModeloDAO.
5. **Capa de Servicios**:
   * Contiene las clases que proporcionan funcionalidades específicas sobre los datos (por ejemplo, cortar y fusionar PDFs).
   * Se encuentra en el paquete /Servicio.

Se detalla la manera como fue implementado el sistema propuesto, se describe visualmente las capas que tiene el sistema, como están distribuidas y sus principales funciones

****

****

**Explicación de las Capas**

**Capa de Presentación (Vistas JSP)**:

* **Funciones Principales**: Interfaz de usuario, donde los usuarios interactúan con el sistema.
* **Componentes Clave**: Páginas JSP (home.jsp, login.jsp, register.jsp, operaciones.jsp, fusionarPDF.jsp, cortarPDF.jsp).

**Capa de Controlador**:

* **Funciones Principales**: Manejar las solicitudes HTTP, controlar la lógica de la aplicación, y decidir qué vista mostrar.
* **Componentes Clave**: Servlets (UsuarioControlador.java, OperacionControlador.java, DetalleSuscripcionUsuarioControlador.java).

**Capa de Modelo**:

* **Funciones Principales**: Representar los datos de la aplicación.
* **Componentes Clave**: Clases de modelo (Usuario.java, DetalleSuscripcionUsuario.java, ListaDeOperacionesSuscripcionUsuario.java).

**Capa de Acceso a Datos (DAO)**:

* **Funciones Principales**: Interactuar con la base de datos para realizar operaciones CRUD.
* **Componentes Clave**: Clases DAO (UsuarioDAO.java, DetalleSuscripcionUsuarioDAO.java).

**Capa de Servicios**:

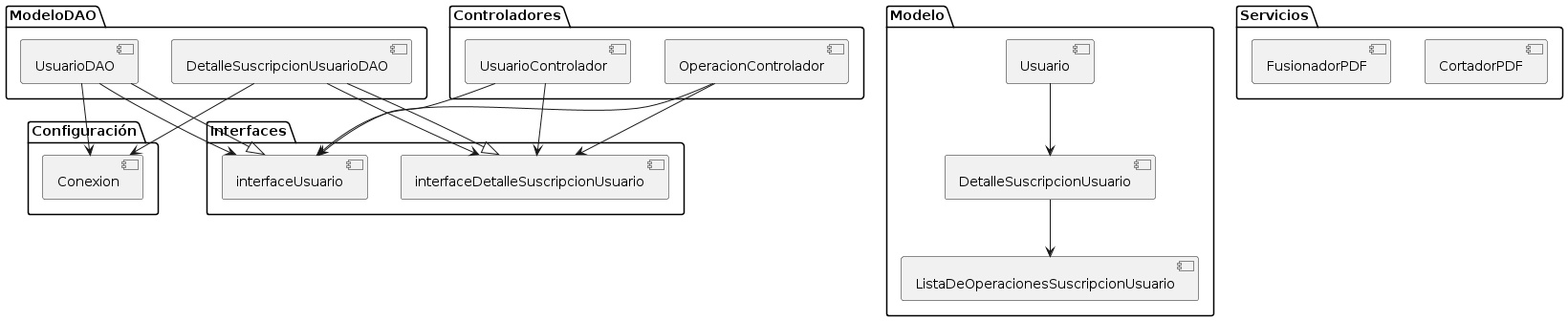
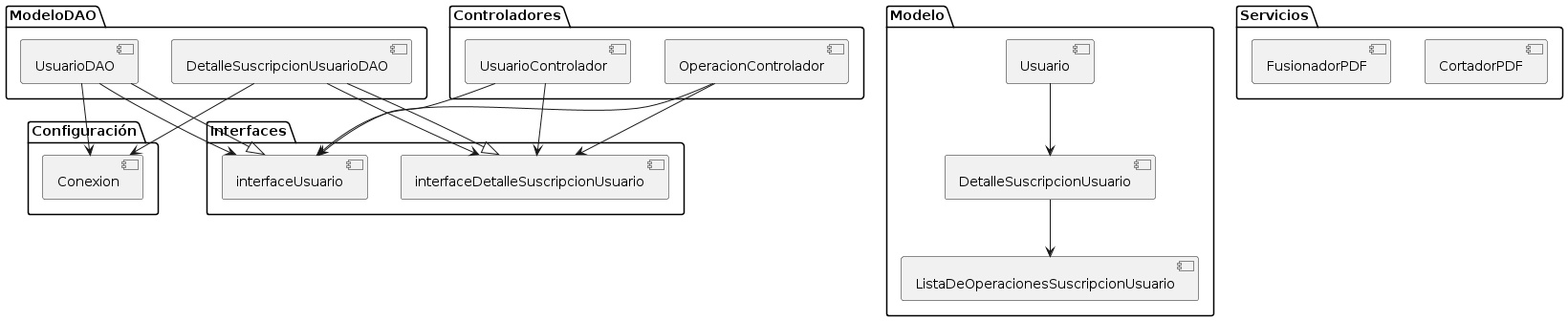
* **Funciones Principales**: Proporcionar funcionalidades específicas sobre los datos.
* **Componentes Clave**: Clases de servicio (CortadorPDF.java, FusionadorPDF.java).

### Diagrama de arquitectura del sistema (Diagrama de componentes)

Componentes Principales:

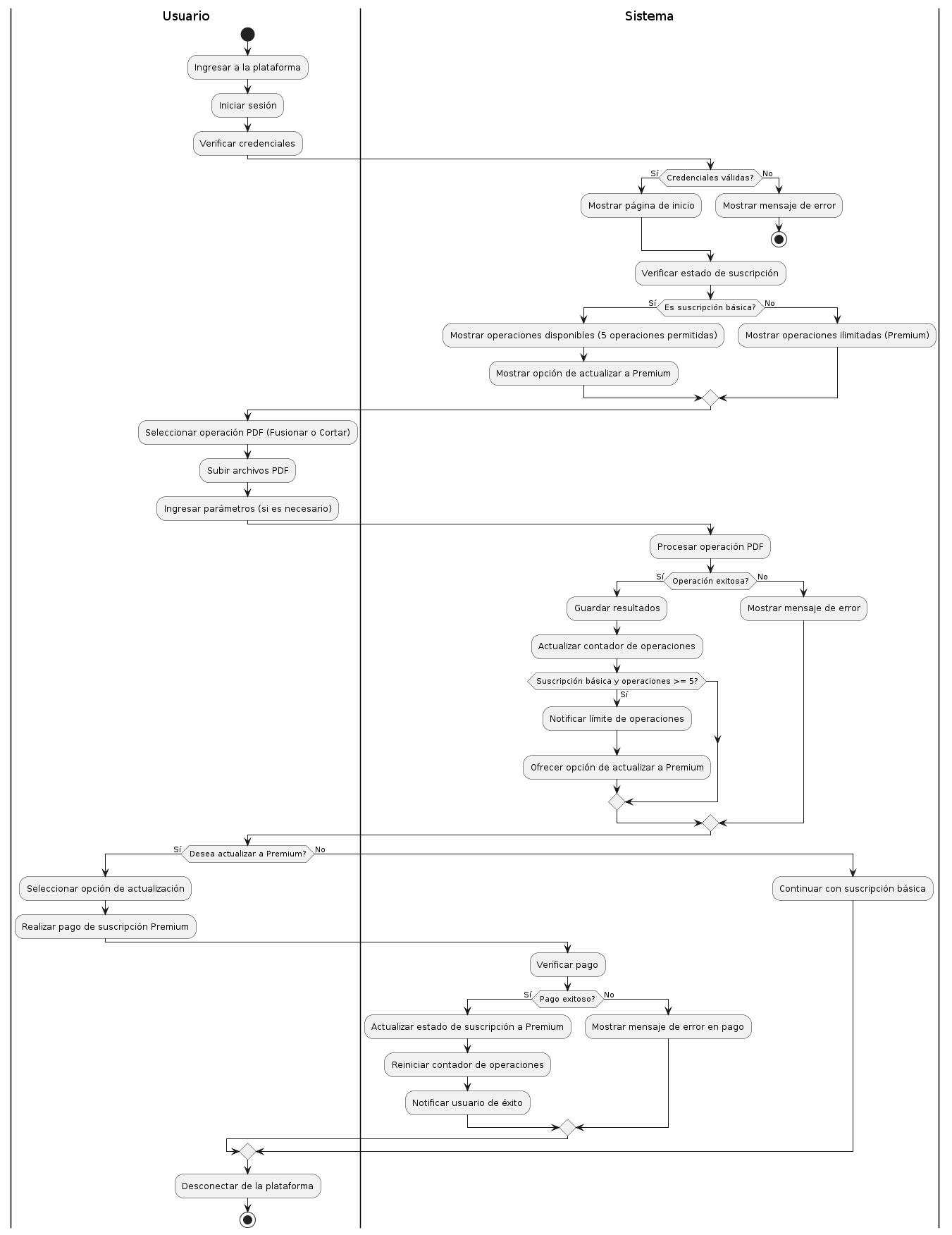
* **Presentación (Vistas JSP)**
  + home.jsp
  + login.jsp
  + register.jsp
  + operaciones.jsp
  + fusionarPDF.jsp
  + cortarPDF.jsp
* **Controladores (Servlets)**
  + UsuarioControlador.java
  + OperacionControlador.java
  + DetalleSuscripcionUsuarioControlador.java
* **Modelos**
  + Usuario.java
  + DetalleSuscripcionUsuario.java
  + ListaDeOperacionesSuscripcionUsuario.java
* **DAOs**
  + UsuarioDAO.java
  + DetalleSuscripcionUsuarioDAO.java
* **Servicios**
  + CortadorPDF.java
  + FusionadorPDF.java

**Diagrama de Componentes:**



# Vista de procesos

### Diagrama de Procesos del sistema (diagrama de actividad)



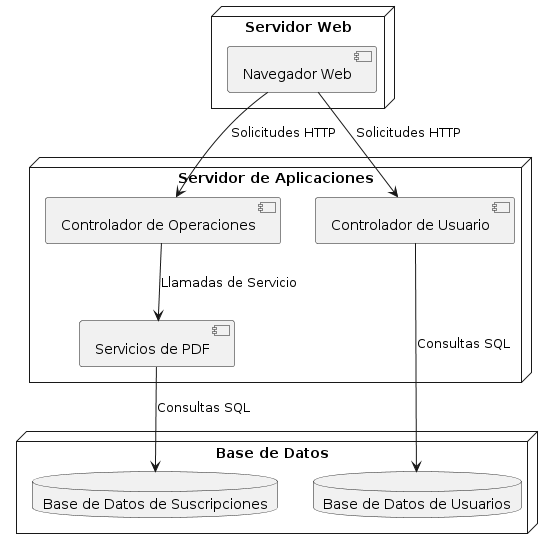
## Vista de Despliegue (vista física)

Escenario de Distribución Física

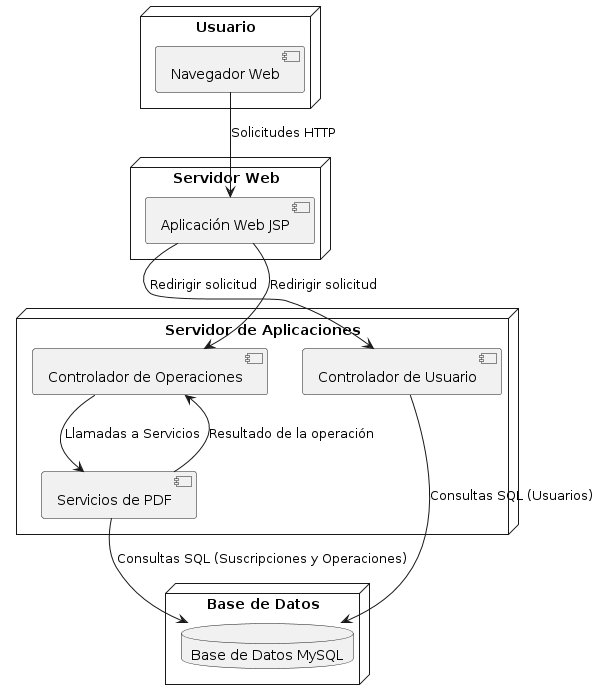
* ***Servidor Web/Aplicaciones****: Donde se ejecutan las aplicaciones Java (Servlets, JSP).*
* ***Servidor de Base de Datos****: Donde se almacena la base de datos MySQL.*
* ***Cliente****: El navegador web del usuario.*

Componentes del Sistema y Nodos

* ***Cliente (Navegador Web)***
* *Interfaz de usuario*
* *Envío y recepción de solicitudes HTTP*
* ***Servidor Web/Aplicaciones***
* *Contiene el servidor Apache Tomcat*
* *Despliegue de Servlets y JSPs*
* *Procesamiento de lógica de negocio*
* ***Servidor de Base de Datos***
* *MySQL*
* *Almacenamiento de datos persistentes*
* Diagrama de Despliegue

**

### Diagrama de despliegue

**

# **ATRIBUTOS DE CALIDAD DEL SOFTWARE**

## Escenario de Funcionalidad:

Cuando un usuario desea realizar operaciones en archivos PDF (como cortar, fusionar, etc.), el sistema debe permitir la ejecución de dichas operaciones y proporcionar notificaciones sobre el estado del proceso.

Validación del escenario

* Origen del estímulo: Interfaz del usuario.
* Estímulo: Realizar operaciones en archivos PDF.
* Entorno: Durante la ejecución de las operaciones en archivos PDF.
* Artefacto: Notificación para el usuario.
* Respuesta: El sistema debe notificar al usuario sobre el estado de la operación (inicio, progreso, finalización) mediante mensajes visuales o sonoros.
* Medida de la respuesta: La respuesta se puede medir por la claridad y efectividad de las notificaciones para informar al usuario sobre el estado de la operación.

## Escenario de Usabilidad:

La usabilidad se refiere a la facilidad con la que un usuario puede aprender a utilizar e interpretar los resultados producidos por un sistema (Barbacci 1995). Los aspectos de usabilidad en tu sistema incluyen:

* **Interfaz intuitiva**: Las páginas JSP deben ser fáciles de navegar y utilizar.
* **Facilidad de aprendizaje**: Los nuevos usuarios deben poder aprender a usar el sistema rápidamente.
* **Minimización de errores**: El sistema debe guiar a los usuarios y minimizar el impacto de los errores.
* **Satisfacción del usuario**: Los usuarios deben sentirse satisfechos con la experiencia de uso del sistema.

## Escenario de Confiabilidad:

La confiabilidad es el equilibrio entre la confidencialidad, la integridad, la irrefutabilidad y la disponibilidad de la información y datos manipulados por el sistema. Abarca los planos de observación físico, lógico y humano y posee enfoques de prevención, precaución y reacción. En tu sistema, la confiabilidad se puede evaluar mediante:

* **Confidencialidad**: Protección de datos sensibles como contraseñas y detalles de suscripción.
* **Integridad**: Garantizar que los datos no se alteren de manera no autorizada.
* **Disponibilidad**: Asegurar que el sistema esté disponible cuando los usuarios lo necesiten.
* **Prevención y reacción ante fallos**: Implementación de mecanismos para prevenir y responder a fallos y ataques.

## Escenario de Rendimiento:

El sistema debe ser capaz de procesar operaciones en archivos PDF (como cortar, fusionar, etc.) eficientemente para un usuario, sin que se degrade el rendimiento ni se ralentice el tiempo de respuesta.

Validación del escenario

* Origen del estímulo: Administrador del sistema.
* Estímulo: Identificar el requisito del sistema.
* Entorno: Momento en el que el usuario desea realizar operaciones en archivos PDF.
* Artefacto: El sistema.
* Respuesta: Soportar exitosamente el incremento de información sin degradar el rendimiento.
* Medida de la respuesta: El tiempo de respuesta del sistema no debe exceder los 2 segundos por operación.

## Escenario de Mantenibilidad:

La mantenibilidad combina la capacidad del programa para ser ampliable, adaptable y servicial (Pressman 2010, pág. 187). La mantenibilidad se puede evaluar mediante:

* **Modularidad del código**: Separación clara de responsabilidades entre componentes del sistema.
* **Documentación**: Disponibilidad de documentación clara y detallada del código y del sistema.
* **Facilidad de pruebas y depuración**: Capacidad de realizar pruebas y depurar el sistema de manera eficiente.

## Otros Escenarios

* **Escenario de Performance:**

El sistema debe ser capaz de procesar operaciones en archivos PDF (como cortar, fusionar.) eficientemente para un usuario, sin que se degrade el rendimiento ni se ralentice el tiempo de respuesta.

Validación del escenario

* Origen del estímulo: Administrador del sistema.
* Estímulo: Identificar el requisito del sistema.
* Entorno: Momento en el que el usuario desea realizar operaciones en archivos PDF.
* Artefacto: El sistema.
* Respuesta: Soportar exitosamente el procesamiento de archivos PDF sin degradar el rendimiento.
* Medida de la respuesta: El tiempo de respuesta del sistema no debe exceder los 2 segundos por operación.