**Documento de Arquitectura de Software**

**Proyecto Biblio F-easy**

**Versión 1.2**

**Control de versiones**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 19/05/2024 | 1.0 | Creación del documento. | Sebastian Ayala y Leslie Diaz |
| 23/06/2024 | 1.1 | Verificación del documento. | Sebastian Ayala |
| 30/06/2024 | 1.2 | Verificación y finalización del documento. | Abraham Carbajal |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**ÍNDICE**

[1. Introducción 4](#_heading=h.f9lx5fz8p2zk)

[1.1. Propósito 4](#_heading=h.acj1o92okewf)

[1.2. Alcance 4](#_heading=h.jhczf1mjb7kq)

[1.3. Definición, siglas y abreviaturas 4](#_heading=h.78lqgges3d8j)

[1.4. Vista global 5](#_heading=h.cfzjj3a5c434)

[2. Macro Arquitectura 5](#_heading=h.jhgu102miua1)

[2.1. Modelo multicapa 5](#_heading=h.f9lx5fz8p2zk)

[2.2. Metas y restricciones arquitectónicas 5](#_heading=h.f9lx5fz8p2zk)

[3. Vista Física 7](#_heading=h.rsx7tau9q47u)

[4. Vista Funcional o Lógica 8](#_heading=h.f9lx5fz8p2zk)

[5. Vista de despliegue 9](#_heading=h.f9lx5fz8p2zk)

[6. Vista de procesos 11](#_heading=h.fworok1v4vvb)

# Introducción

## Propósito

El propósito de este documento es definir y dar a conocer la Arquitectura de Software para el desarrollo del proyecto "Biblio F-easy. Se utilizarán las vistas de proceso, lógica y física, para describir la estructura y funcionalidad del sistema, así como su distribución física, interacción y descripción general.

Este documento está dirigido al equipo de desarrollo, con el fin de proporcionar las bases arquitecturales y una visión global del diseño para llevar a cabo el proyecto.

## Alcance

Este documento describe la arquitectura del sistema "Biblio F-easy", el cual permitirá a los usuarios buscar, reservar y acceder a recursos bibliográficos de manera eficiente, así como gestionar reservas de pupitres y emisión de carnets de forma transparente. El sistema se enfocará en mejorar los procesos internos y externos relacionados con la gestión bibliotecaria. La aplicación será web y funcionará en dispositivos con navegación web estándar. Además, se especifican las restricciones, atributos de calidad, vistas arquitecturales y patrones de diseño a implementar por el equipo de desarrollo.

## Definición, siglas y abreviaturas

* DAS: Documento que describe la arquitectura de software.
* API: Interfaz que permite a diferentes aplicaciones comunicarse entre sí.
* HTTP: Protocolo utilizado para transferir datos en la web.
* IIS (Internet Information Services): Servidor web utilizado para alojar aplicaciones y sitios web.
* DBMS: Sistema que administra bases de datos.
* SQL (Structured Query Language): Lenguaje utilizado para interactuar con bases de datos.
* GUI (Graphical User Interface): Interfaz visual que permite interactuar con un programa.
* AngularJS: Framework de JavaScript para crear aplicaciones web dinámicas.
* Spring Boot: Framework para desarrollar aplicaciones Java fácilmente.
* Responsive Design: Diseño que se adapta a diferentes dispositivos y tamaños de pantalla.
* Bootstrap: Framework de diseño web para crear interfaces modernas.
* Singleton: Patrón de diseño que garantiza una única instancia de una clase.
* Modelo Multicapa (Modelo n-tier): Patrón de arquitectura de software que separa la lógica de negocio de la interfaz del usuario en capas.

## Vista global

Se presentará una macro arquitectura del sistema "Biblio F-easy", con especificaciones de las vistas utilizadas en el proyecto, metas y restricciones arquitectónicas, así como vistas con diagramas para una comprensión completa del sistema.

# Macro Arquitectura

## Modelo multicapa

El modelo multicapa, también conocido como 3-tier, se utiliza para separar la lógica de negocio de la interfaz del usuario en distintos componentes, ofreciendo:

* Capa de Presentación/UI: Interfaz de usuario, componentes Front-End.
* Capa de Lógica de Negocio: Gestión de procesos, reglas de negocio, controladores y ruteador.
* Capa de Acceso a Datos/Backend: Gestión de datos, Interacción con la base de datos, microservicios y APIs.

Este modelo mejora la mantenibilidad, escalabilidad y flexibilidad del sistema, ya que cada capa cumple una función específica, facilitando la incorporación de nuevas funcionalidades y adaptaciones futuras.

## Metas y restricciones arquitectónicas

| **Atributos de calidad “Observables”** | | | |
| --- | --- | --- | --- |
| Atributo | Descripción | Tácticas /Patrón de  Arquitectura | Aplicación |
| Seguridad | Garantizar que el acceso a los recursos y funcionalidades esté restringido según los roles asignados a los usuarios autorizados. | Control de Acceso Basado en Roles: Asignación de permisos según los roles de usuario (estudiante, bibliotecaria). | Aplicación al momento de autenticación de usuarios y accesos a funcionalidades específicas. |
| Disponibilidad | Capacidad del sistema para estar operativo y accesible para los usuarios en todo momento. | Balanceo de Carga: Distribución equitativa de la carga de trabajo entre varios servidores para evitar sobrecargas. | Implementación en la infraestructura de servidores de Biblio F-easy. |
| Usabilidad | Facilidad de uso del sistema para los usuarios finales. | Diseño Centrado en el Usuario (UCD): Integración de procesos de diseño y pruebas centrados en la experiencia del usuario. | Desarrollo de interfaces intuitivas, búsqueda rápida de recursos bibliográficos y demás procesos |

# 

| **Atributos de calidad “No observables”** | | | |
| --- | --- | --- | --- |
| Atributo | Descripción | Tácticas /Patrón de  Arquitectura | Aplicación |
| Modificabilidad | Facilidad para realizar cambios en el sistema, como agregar nuevas funcionalidades o adaptarse a requisitos cambiantes. | Arquitectura de Microservicios: Descomposición del sistema en servicios independientes y autocontenidos. | División de la lógica de negocios, gestión de recursos y acceso a datos en microservicios. |
| Mantenibilidad | Capacidad de realizar mantenimiento y mejoras al sistema de manera eficiente y económica. | Separación de capas: División clara de las capas de presentación, lógica de negocios y acceso a datos para facilitar las actualizaciones. | Implementación de patrones de diseño y estándares de codificación que favorezcan la legibilidad, cohesión y reutilización del código. |
| Portabilidad | Adaptabilidad del sistema para funcionar en diferentes entornos. | Diseño Responsivo: Desarrollo de interfaces de usuario que se ajustan automáticamente. | Implementación de técnicas de diseño web responsivo para una experiencia de usuario consistente y accesible. |

# Vista Física

Diagrama informal de la vista física de Biblio F-easy:

El Diagrama informal de la vista física contempla el uso del aplicativo por parte del usuario, quien ingresa al navegador desde diversos dispositivos, como computadoras de escritorio, portátiles y dispositivos móviles. El usuario accede a la página web de Biblio F-easy, la cual está alojada en un servidor de aplicaciones.

Elementos físicos:

* Usuarios: Personas que acceden al sistema desde diferentes dispositivos.
* Dispositivos electrónicos: Computadoras de escritorio, portátiles y dispositivos móviles con conexión a internet.
* Internet: La red global que permite la comunicación entre todos los dispositivos y servidores.
* Base de datos: Almacena la información del sistema, como datos de libros, usuarios y reservas. Utiliza MySql como gestor de bases de datos.

Atributos de calidad:

**Escalabilidad**: El sistema está diseñado para crecer con la carga de trabajo, utilizando un enfoque de diseño multicapa que separa la lógica de negocio y los datos de la interfaz de usuario. Esto permite que nuevas funcionalidades se implementen fácilmente manipulando las capas del sistema.

**Portabilidad:** Biblio F-easy implementa el patrón de diseño Responsive Design, lo que le permite adaptarse a diferentes dispositivos con diferentes tamaños de pantalla, como computadoras de escritorio, portátiles y dispositivos móviles.

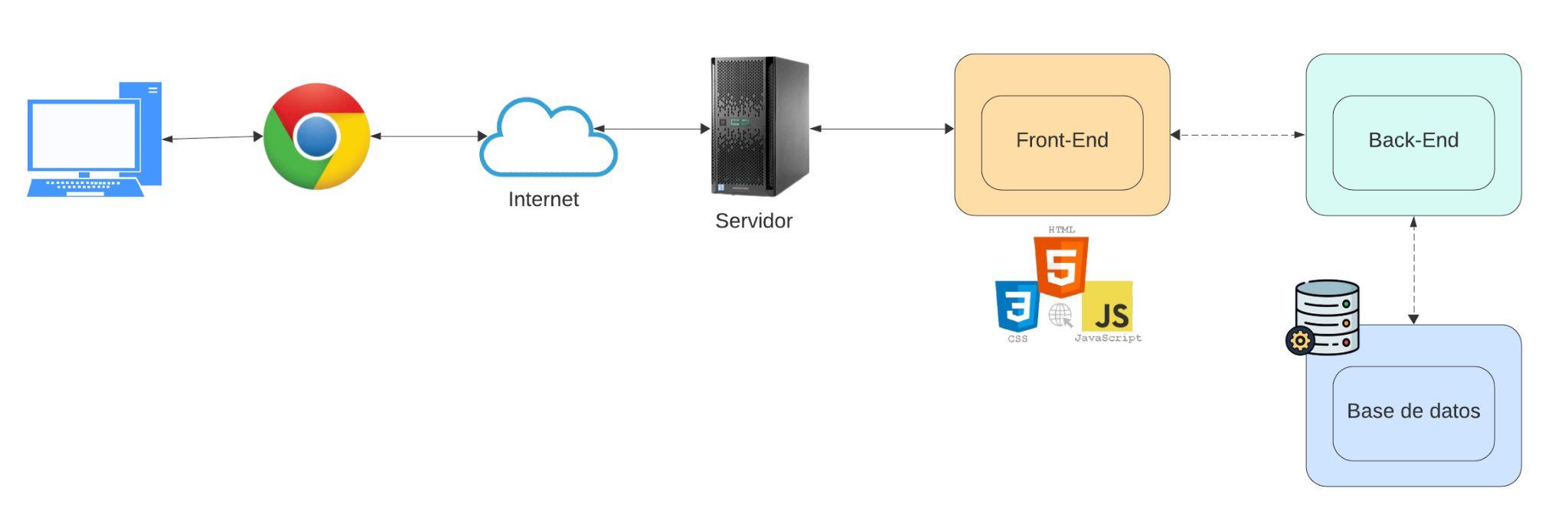


Figura xx. Diagrama informal físico - Biblio F-easy

# Vista Funcional o Lógica

Diagrama de Componentes de Biblio F-easy:

El siguiente diagrama representa el marco de trabajo de la relación entre los componentes que se encuentran dentro de esta arquitectura para la funcionalidad del aplicativo web. En este diagrama se ve reflejado que el sistema es multicapa representado por la capa de presentación, capa de lógica del negocio y capa de acceso a datos, lo que permite una óptima y eficiente mantenibilidad, modificabilidad y mayor funcionalidad del sistema.

Componentes:

* **Capa de Presentación:**

Descripción:

Responsable de la interfaz de usuario del aplicativo web.

* **Capa de Lógica de Negocio:**

Descripción: Gestiona la lógica de procesos y funcionalidades del sistema.

Ruteador y Controladores: Utiliza un ruteador para redireccionar a controladores, los cuales ejecutan funciones específicas de acuerdo con la ruta invocada. Incluye funciones propias del marco de trabajo y lógica de procesos del sistema.

* **Capa de Acceso a Datos:**

Descripción: Encargada del acceso directo a la base de datos y la gestión de microservicios.

Atributos de Calidad:

**Mantenibilidad**: Se garantiza mediante la alta cohesión de componentes, la separación de tareas y la aplicación del patrón multicapa. Esto permite que el sistema sea dinámico y adaptable a las necesidades de expansión, facilitando la modificación y el mantenimiento del aplicativo web.

**Modificabilidad**: Se logra a través de la separación de tareas y la estandarización de componentes, lo que permite realizar cambios en una parte de la aplicación sin afectar los demás componentes. Esto se logra gracias al diseño multicapa y la modularización del sistema.

**Funcionalidad**: Se garantiza mediante la estandarización de componentes y la interoperabilidad del aplicativo. El uso del patrón multicapa y la separación de responsabilidades facilitan la integración de nuevos componentes y funcionalidades, mejorando así la funcionalidad global del sistema.

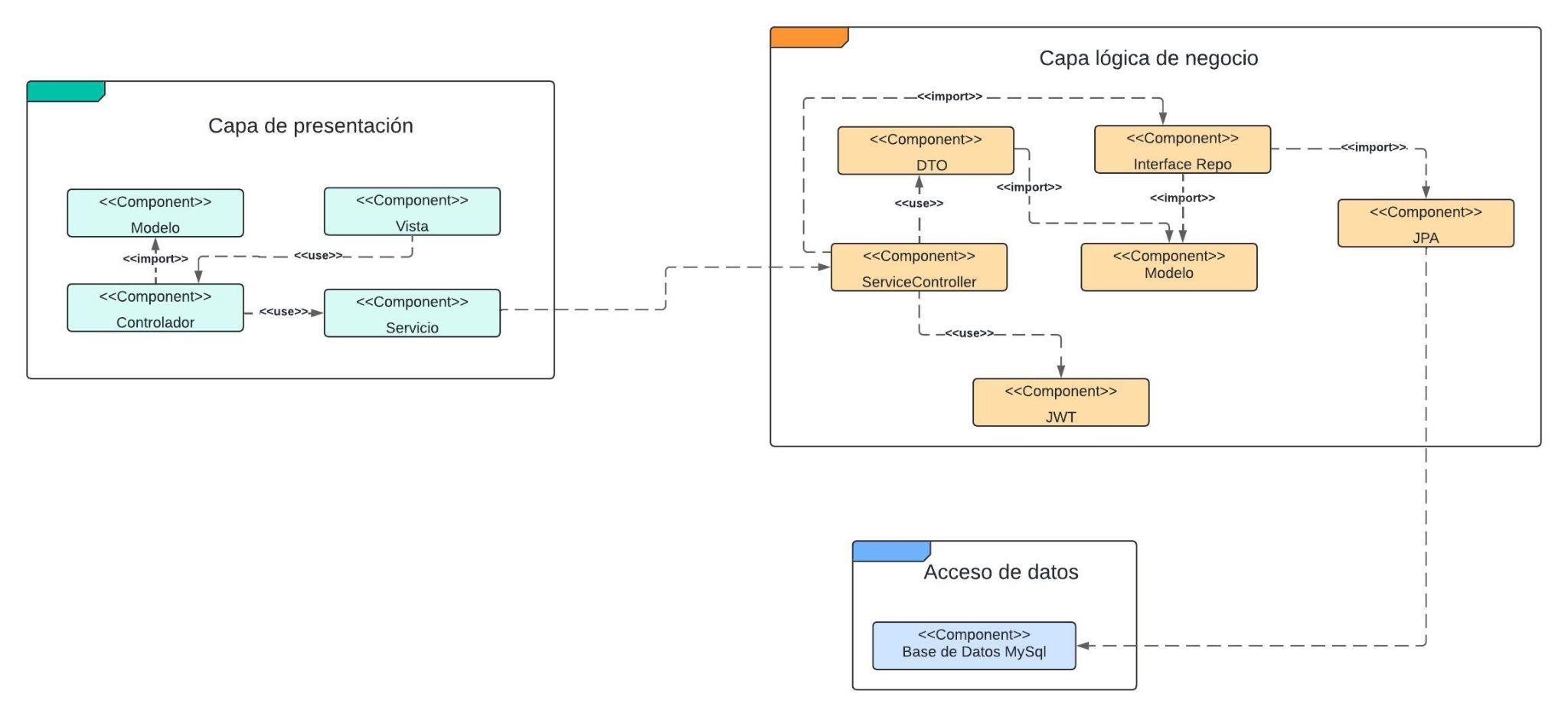


Figura xx. Diagrama de componentes - Biblio F-easy

# Vista de despliegue

Diagrama de Despliegue de Biblio F-easy:

Este diagrama describe las configuraciones de redes físicas sobre las cuales el software será desarrollado. Se indican los nodos físicos que interactúan con la aplicación web, tales como un nodo para el dispositivo donde se accede a la aplicación web en un PC con acceso a Internet y un navegador. Este envía las peticiones por HTTP para mayor seguridad de la información accediendo al nodo del Web Server donde aloja el Internet Information Service (IIS) HTTP Server y El DBMS con el componente de base de datos MySQL.

Componentes:

* **Nodo de Acceso a la Aplicación Web:**

Descripción: Dispositivo desde donde se accede a la aplicación web, como una PC con acceso a Internet y un navegador.

Protocolo de Comunicación: HTTP

Interacción: Envía peticiones HTTP al servidor web para interactuar con la aplicación.

* **Nodo del Servidor Web:**

Descripción: Aloja el Internet Information Service (IIS) HTTP Server y el DBMS con el componente de base de datos SQL Server Standard 2019.

Funcionalidad de la Aplicación: Divide la funcionalidad de la aplicación en capas de presentación, lógica del negocio y acceso a datos, lo que permite una mejor mantenibilidad, modificabilidad y funcionalidad del sistema.

* **Lógica de Negocios y Backend:**

Descripción: Integra las interfaces de APIs, que son los backend de la lógica del negocio creada por microservicios.

Interacción: Gestiona la lógica de negocio y la comunicación con la capa de acceso a datos para realizar operaciones en la base de datos.

Atributos de Calidad:

**Modificabilidad**: Se aplica mediante la separación de tareas y la modularización del sistema, lo que permite realizar cambios en una parte de la aplicación sin afectar los demás componentes. El diseño multicapa facilita la modificación y mantenimiento del sistema, lo que mejora su adaptabilidad y escalabilidad a medida que evoluciona el proyecto

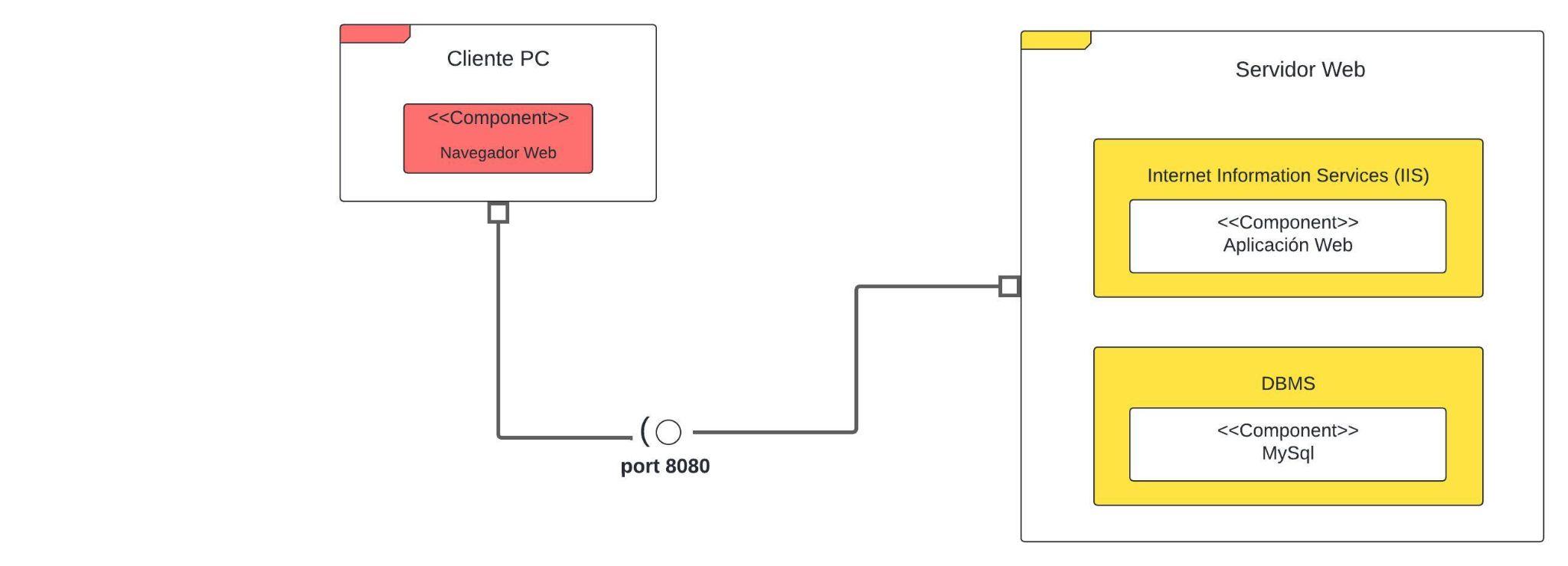
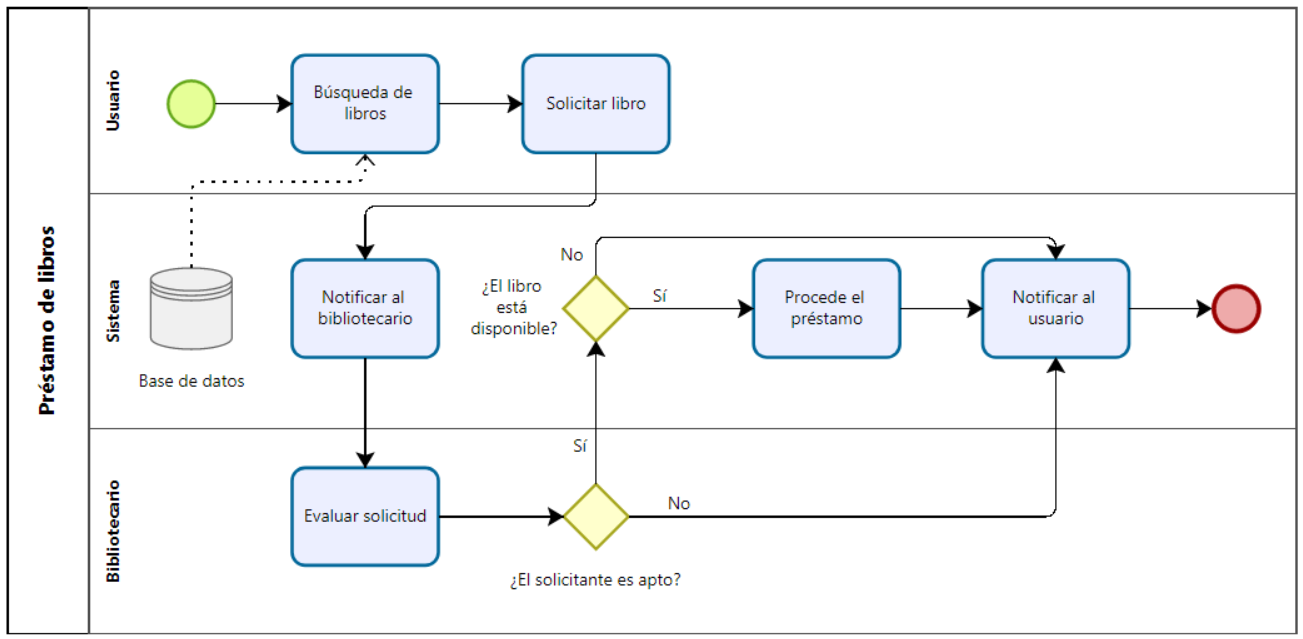


Figura xx. Diagrama de despliegue - Biblio F-easy

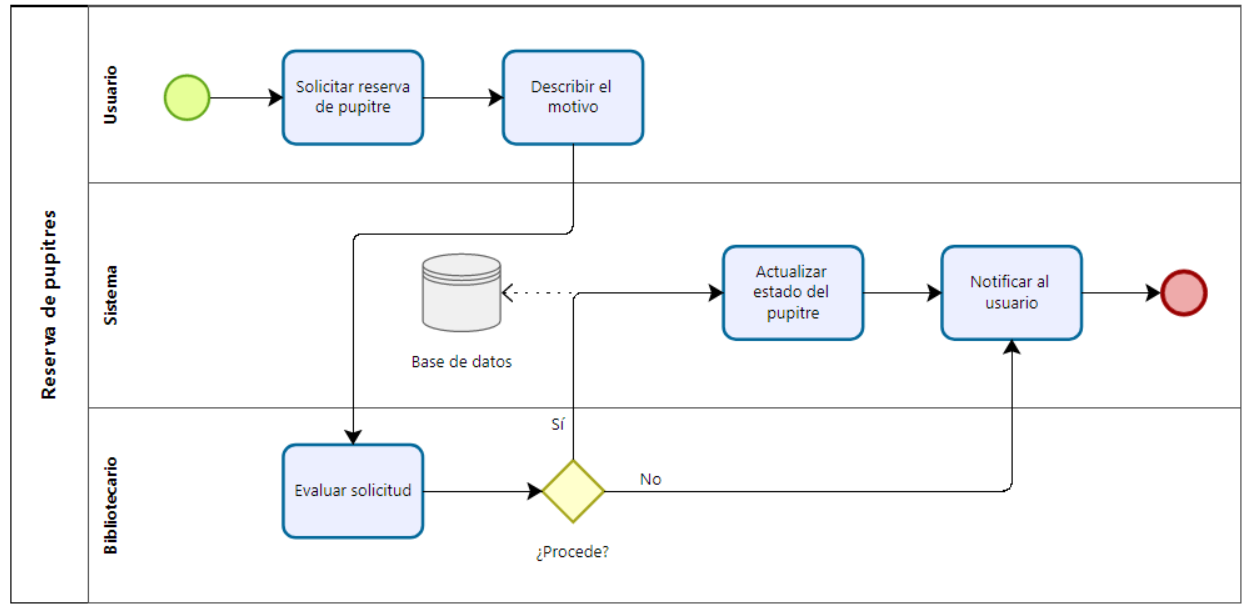
# Vista de procesos

Diagrama de actividades: Préstamo de libros



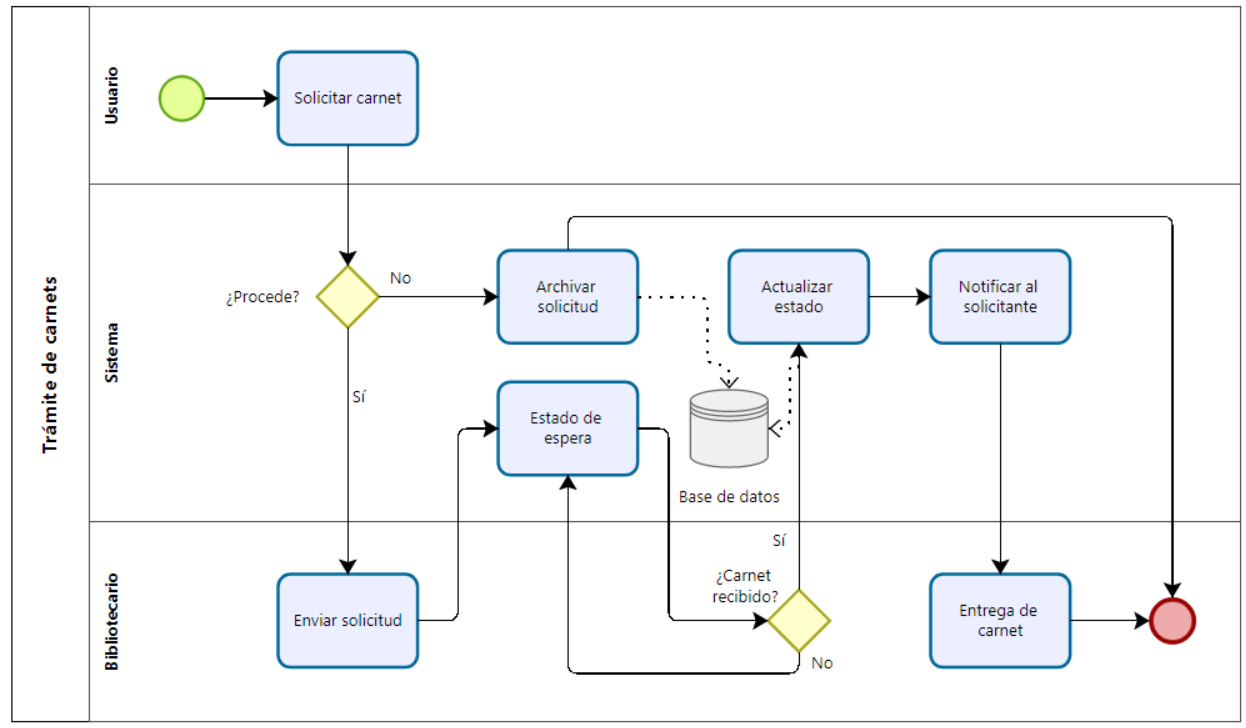
El usuario busca un libro de su gusto, él tiene la opción a solicitarlo, si en caso manda solicitud, el sistema le notifica al bibliotecario, y éste aprueba la solicitud, para que luego el sistema de encargue de la gestión y avisarle al usuario que su solicitud ha sido aceptada.

Diagrama de actividades: Reserva de pupitres



El usuario solicita la reserva de algún pupitre de la biblioteca, el bibliotecario evalúa esta solicitud, y el sistema se encarga de actualizar el estado del pupitre.

Diagrama de actividades: Trámite de carnets



En esta funcionalidad, el usuario puede solicitar su carnet, este mensaje le llega al bibliotecario, y esté se encarga de aprobar la solicitud, para que el sistema solo se encargue de la gestión del estado del carnet.