

Documento de diseño del sistema

Sentido del documento

Este documento describe el diseño del sistema de software para este proyecto. Se propondrán soluciones para los problemas de diseño y se describirán las decisiones de diseño tomadas. El documento de diseño del sistema (SDD) es un documento de referencia para el diseño del sistema de software y la arquitectura del software. Es un documento vinculante para el desarrollo del software y la implementación de los subsistemas.

El contenido de este documento sigue y se ajusta a los requisitos especificados en el IEEE 1016–1998[1]

Índice

1. Introducción	2
1.1. Propósito del documento	2
1.2. Definiciones, acrónimos y abreviaturas	2
Referencias	2
2. Objetivos de diseño	2
3. Composición del sistema, arquitectura y capas	3
3.1. Capas del sistema y sus responsabilidades	3
3.2. Diagrama de capas/arquitectura	3
3.3. Componentes identificados y sus responsabilidades	3
3.4. Dependencias y flujos de control	5
3.5. Rationale de la arquitectura	6
4. Diseño de la Interfaz Gráfica de Usuario (GUI)	6
4.1. Resumen de las Pantallas	6
4.2. Estilo visual y usabilidad	8
4.3. Diagrama de navegación	8
4.4. Bocetos	9

1. Introducción

Este documento describe el diseño del sistema de software para el proyecto BiciFast. El diseño del sistema se basa en los requisitos funcionales y no funcionales especificados en el documento de requisitos del sistema (SRD). [2] Este documento está destinado a ser utilizado por los desarrolladores del sistema, los arquitectos de software y los diseñadores de la interfaz de usuario.

1.1. Propósito del documento

El propósito de este documento es proporcionar una descripción detallada del diseño del sistema de software para el proyecto BiciFast.

1.2. Definiciones, acrónimos y abreviaturas

Referencias

- [1] «IEEE Guide for Software Requirements Specifications», Institute of Electrical and Electronics Engineers, IEEE Std. 1016-1998, 1998. DOI: 10.1109/IEEESTD.2009.5167255.
- [2] M. Fort Muñoz, A. Cruces Uruñuela, F. Duque Sán Milán y H. Giménez Palomo, «BiciFast Software Requirements Document», Universidad de Antonio Nebrija, SRD, 2025.

2. Objetivos de diseño

Para diseñar este sistema, se ha priorizado:

1. **Modularidad:** Hacer que las partes del sistema sean independientes entre ellas, permitiendo la fácil modificación de los módulos.
2. **Simplicidad:** Hacer que el sistema sea fácil de entender y de usar.
3. **Separación de responsabilidades:** Separar las diferentes partes del sistema en módulos independientes, cada uno con su propia responsabilidad.
4. **Escalabilidad:** Hacer que el sistema sea fácil de escalar y de modificar en el futuro.

3. Composición del sistema, arquitectura y capas

La aplicación sigue una arquitectura por capas, añadiendo el patrón «Façade» para la comunicación entre capas. El objetivo de dicha arquitectura es facilitar el desarrollo, mantenimiento y escalabilidad de la aplicación.

3.1. Capas del sistema y sus responsabilidades

- **Capa de presentación:** Gestiona la interacción con el exterior (usuarios) y muestra información, centralizando mediante *FachadaAplicacion*. Las operaciones de esta capa son canalizadas a través de la *FachadaGUI*, que es la responsable de la interfaz de usuario.
- **Capa aplicación:** Es la capa de coordinación entre la capa de presentación y la capa de negocio. A ella la GUI le envía las peticiones de los usuarios y el estado de la aplicación. Delegando en la capa de negocio (Control) la lógica de negocio.
- **Capa de control:** Contiene la lógica de negocio y las reglas de negocio. Centralizada en *ControladorAplicacion*, controla el flujo de la aplicación y delega en los *Controladores* específicos de cada módulo. Esta capa es la responsable de la persistencia de los datos, delegando en el módulo *Capa de datos* la gestión de la base de datos.
- **Capa de datos:** Contiene la lógica de acceso a datos y la persistencia de los mismos. Esta capa es la responsable de la gestión de la base de datos y de la persistencia de los datos. Hace uso de *DAO* (Data Access Object) para la gestión de la base de datos. Está centralizada en *FachadaBaseDatos*, que es la responsable de la gestión de la base de datos.

3.2. Diagrama de capas/arquitectura

La Figura 1 muestra la arquitectura de capas del sistema, mostrando las principales fachadas que conectan cada capa y las relaciones entre ellas. Estas fachadas encapsulan la funcionalidad de cada capa, promoviendo el desacoplamiento y la claridad en el flujo de datos.

3.3. Componentes identificados y sus responsabilidades

Después de hacer un análisis de los requisitos funcionales y no funcionales, hemos definido los siguientes componentes y sus responsabilidades:

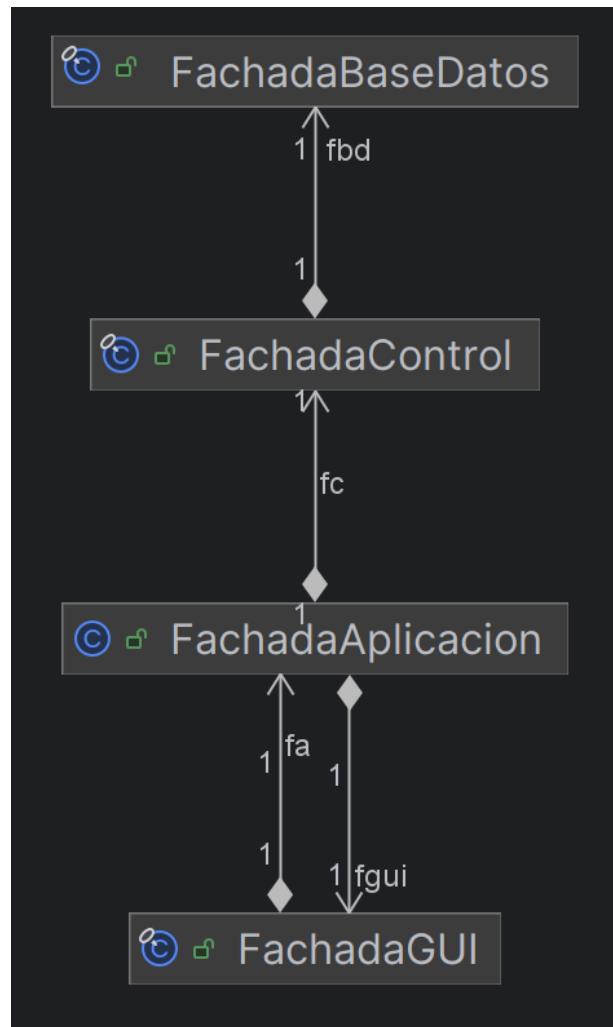


Figura 1: Diagrama de arquitectura por capas

3.3.1. Componentes identificados

- **Formularios de la GUI:** Son los responsables de la interacción con el usuario y de la presentación de la información.
- **Controladores (Servicios y lógica de negocio):** Gestionan la lógica de la aplicación e interceden entre los DAO y la interfaz gráfica.
- **DAO:** Se encargan de «hablar» con la base de datos. Permiten encapsular y abstraer la base de datos y a los datos mismos de la lógica de la aplicación.
- **Base de datos:** Gestiona los datos «permanentes».

3.3.2. Responsabilidades

Para cada componente, hemos identificado las siguientes responsabilidades funcionales:

GUI (Presentación) ■ Mostrar la interfaz gráfica.

- Gestionar las diferentes vistas.

Controladores (Servicios) ■ Gestionar usuarios.

- Autenticar.
- Cambiar datos.
- Gestionar alquileres.
- Gestionar la criptografía.

DAOs (Data Access Object) ■ Hacer consultas a la base de datos.

- Modificar la base de datos.
- Eliminar entradas de la base de datos.

Base de datos (Persistencia) ■ Guardar los datos de:

- Usuarios
- Bicicletas
- Estaciones
- Alquileres

3.4. Dependencias y flujos de control

El flujo habitual de control de la aplicación es el que se muestra en la Figura 2. En este flujo, la *GUI* envía una petición a la *FachadaAplicacion*, que

[GUI] → [FachadaAplicacion] → [FachadaControl] → [Gestores / BD]

Figura 2: Flujo de control

delega en la *FachadaControl* para que gestione la petición y pueda solicitar a la *FachadaBaseDatos* el acceso/almacenamiento de datos en la base de datos.

Ciertamente, existen algunas dependencias cíclicas entre las clases de la capa de presentación, la fachada de presentación y la fachada de aplicación, ya que la *FachadaAplicacion* necesita conocer la *FachadaGUI* para poder enviarle mensajes. Puede parecer alarmante, pero no es un problema, ya que el ciclo ocurre principalmente dentro de la capa de presentación y no afecta a la lógica de negocio. Además, las llamadas desde la *FachadaAplicacion* a la *FachadaGUI* son para situaciones muy específicas, como cambiar el idioma o relanzar la interfaz.

3.5. Rationale de la arquitectura

La elección de la arquitectura por capas se ha tomado por su capacidad de desacoplar y diferenciar las responsabilidades de las distintas capas. Gracias a esta arquitectura, este sistema gana mayor facilidad de mantenimiento, escalabilidad y modularidad.

Por otro lado, la elección del patrón *Facade* se ha tomado por su capacidad de simplificar la interfaz de las capas y facilitar la comunicación entre ellas. Este patrón permite ocultar la complejidad de las capas y proporcionar una interfaz más sencilla y fácil de usar. El otro patrón utilizado es el *DAO*, que centraliza y separa el acceso a datos en módulos específicos. Evitando malas prácticas como el acceso directo a la base de datos desde la capa de presentación o la capa de negocio.

4. Diseño de la Interfaz Gráfica de Usuario (GUI)

La interfaz gráfica de la aplicación está desarrollada empleando Java Swing, orientada a un uso intuitivo por parte de los usuarios finales. La aplicación permitirá el acceso a los servicios de alquiler de bicicletas a través de una serie de pantallas conectadas por un flujo lógico y sencillo. Por falta de tiempo, sólo se ha realizado el diseño de las pantallas relativas al usuario, dejando de lado las pantallas de administración y mantenimiento de estaciones y bicicletas. Esta sección será actualizada en futuras versiones del documento para incluir las pantallas restantes.

4.1. Resumen de las Pantallas

A continuación se detallan las principales pantallas de la aplicación:

4.1.1. Pantalla 1: Iniciar sesión

- **Nombre:** DiaLogin
- **Objetivo:** Permitir al usuario autenticarse en el sistema.
- **Componentes:**
 - Campo de texto para correo electrónico.
 - Campo de texto para contraseña.
 - Botón "Iniciar sesión".
 - Enlace para registrarse si no se tiene cuenta.

- **Flujo:** Al iniciar sesión con éxito, se redirige a la pantalla principal.

4.1.2. Pantalla 2: Principal del usuario

- **Nombre:** VPrincipalUsuario
- **Objetivo:** Servir como punto central de acceso a las funciones principales.
- **Componentes:**
 - Vista de mapa con la localización de las estaciones.
 - Botones para ver estaciones, alquilar, devolver bicicleta.
 - Acceso al perfil del usuario.
- **Flujo:** Permite navegar hacia la selección de estación o a la edición del perfil.

4.1.3. Pantalla 3: Bicicletas en una estación

- **Nombre:** DiaBicis
- **Objetivo:** Mostrar bicicletas disponibles en una estación concreta.
- **Componentes:**
 - Lista de bicicletas con identificador, tipo y estado.
 - Botón de reserva/alquiler.

4.1.4. Pantalla 4: Perfil de usuario

- **Nombre:** DiaUsuario
- **Objetivo:** Permitir la visualización y edición de los datos personales.
- **Componentes:**
 - Nombre, correo electrónico, idioma preferido.
 - Historial de alquileres.
 - Botón "Editar perfil".

4.1.5. Pantalla 5: Selección de idioma

- **Nombre:** VIdioma
- **Objetivo:** Configurar el idioma de la aplicación.
- **Componentes:**
 - Lista desplegable de idiomas.
 - Botón “Guardar”.

4.2. Estilo visual y usabilidad

La interfaz presenta un diseño sencillo, funcional y adaptado a usuarios con diferentes niveles de experiencia tecnológica. Los colores son neutros y consistentes, y los componentes están dispuestos de manera que facilitan la navegación. Los textos son breves y claros, y los botones tienen etiquetas intuitivas.

4.3. Diagrama de navegación

El diagrama de navegación muestra cómo se conectan las diferentes pantallas de la aplicación y cómo los usuarios pueden navegar entre ellas. Cada pantalla está conectada a las demás según el flujo lógico del uso de la aplicación, permitiendo un acceso fácil a todas las funcionalidades.

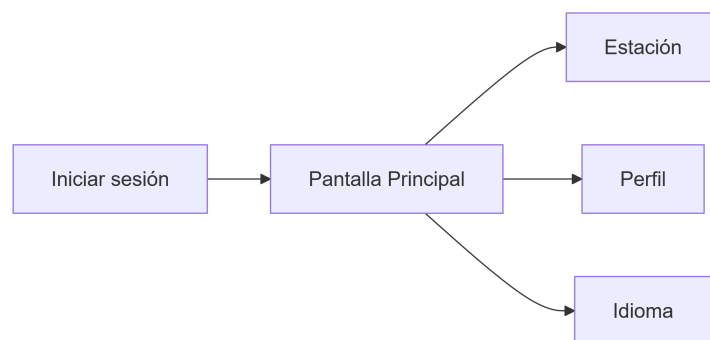


Figura 3: Diagrama de navegación de la aplicación

4.4. Bocetos

A continuación se presentan los mockups de las pantallas principales de la aplicación. Estos mockups ilustran el diseño visual y la disposición de los elementos en cada pantalla. Estas han sido creadas con la herramienta Draw.io y se encuentran en formato PNG para su inclusión en el documento.

The mockup shows a rectangular window titled "Login". Inside the window, on the left side, are the labels "Usuario" and "Contraseña" stacked vertically. To the right of "Usuario" is a single-line text input field. To the right of "Contraseña" is a single-line text input field. To the right of these two input fields is a rectangular button labeled "Aceptar". Below the "Aceptar" button, and shifted towards the right side of the window, is another rectangular button labeled "Salir".

Figura 4: Pantalla 1 - Iniciar sesión

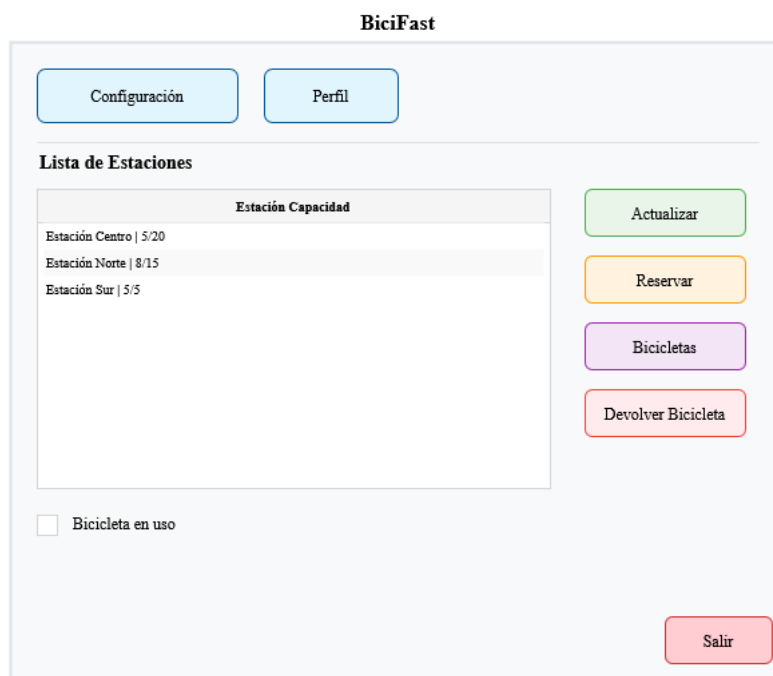


Figura 5: Pantalla 2 - Principal del usuario

Bicicletas de la Estación

Estación: Estación Centro - Calle Principal 123

Bicicletas Disponibles	Actualizar
001	Cancelar

Figura 6: Pantalla 3 - Bicycles in a station

Datos del Usuario

Nombre:	<input type="text" value="Juan Carlos"/>
Primer apellido:	<input type="text" value="García"/>
Segundo apellido:	<input type="text" value="López"/>
DNI:	<input type="text" value="12345678A"/>
Email:	<input type="text" value="juan.garcia@email.com"/>
Dirección:	<input type="text" value="Calle Principal 123, 4º B"/>
Fecha nacimiento:	<input type="text" value="1985-03-15"/>
Teléfono:	<input type="text" value="666 123 456"/>
Método de pago:	<input type="text" value="TARJETA_CREDITO"/>
Inicio suscripción:	<input type="text" value="2024-01-01"/>
Fin suscripción:	<input type="text" value="2024-12-31"/>
Tipo usuario:	<input type="text" value="PREMIUM"/>

Cerrar

Figura 7: Pantalla 4 - Datos del usuario



Figura 8: Pantalla 5 - Selección de idioma