

Diseño de un sistema para el préstamo de bicicletas con herramientas IA

Integrantes: Alejandro García Sánchez

Kevin Andrés Jaramillo Hernández

Requerimientos e historias de usuario

Requisitos funcionales

1. Gestión de bicicletas

- 1.1. El sistema debe permitir **registrar** nuevas bicicletas con sus características (tipo, marca, color, estado, tarifa por hora).
 - 1.2. El sistema debe **actualizar** la información de una bicicleta.
 - 1.3. El sistema debe **mostrar la disponibilidad** de las bicicletas (disponible, en préstamo, en mantenimiento).
 - 1.4. El sistema debe **eliminar o desactivar** bicicletas fuera de servicio.
-

2. Gestión de usuarios

- 2.1. El sistema debe permitir **registrar usuarios** con datos personales (nombre, identificación, correo, teléfono).
 - 2.2. El sistema debe permitir **autenticar** a los usuarios mediante usuario y contraseña.
 - 2.3. El sistema debe **almacenar el historial de préstamos** de cada usuario.
-

3. Préstamo de bicicletas

- 3.1. El sistema debe permitir al usuario **seleccionar una bicicleta disponible** para préstamo.
 - 3.2. El sistema debe **registrar la hora de inicio** del préstamo.
 - 3.3. El sistema debe **registrar la hora de devolución** de la bicicleta.
 - 3.4. El sistema debe **calcular el tiempo total del préstamo en minutos**.
 - 3.5. El sistema debe **calcular el costo total** del préstamo con base en la tarifa por hora y los minutos usados:
-

4. Consulta de información

- 4.1. El sistema debe permitir consultar el **precio por hora** de cada bicicleta.
- 4.2. El sistema debe permitir consultar las **características técnicas** de las bicicletas (tipo, tamaño, estado, etc.).

Requisitos no funcionales

Categoría	Requisito
Rendimiento	El sistema debe calcular y mostrar el costo del préstamo en menos de 2 segundos.
Disponibilidad	El sistema debe estar disponible el 95% del tiempo.
Seguridad	Las credenciales deben almacenarse cifradas. Solo usuarios autenticados pueden registrar préstamos.
Escalabilidad	Debe permitir incorporar más bicicletas y usuarios sin afectar el rendimiento.
Mantenibilidad	El código debe seguir una arquitectura modular que facilite actualizaciones.

Historias de usuario

ID	Historia de Usuario	Criterios de Aceptación
HU-01	Como usuario , quiero ver las bicicletas disponibles para saber cuáles puedo alquilar.	Se muestra una lista con bicicletas en estado "disponible".
HU-02	Como usuario , quiero ver el precio por hora de una bicicleta antes de alquilarla.	El sistema muestra la tarifa por hora y el tipo de bicicleta.
HU-03	Como usuario , quiero ver las características de una bicicleta (tipo, color, marca, tamaño) para elegir la más adecuada.	El sistema despliega la ficha técnica de la bicicleta.
HU-04	Como usuario registrado , quiero realizar un préstamo seleccionando una bicicleta disponible.	El sistema registra la hora de inicio y cambia el estado a "en préstamo".

ID Historia de Usuario

Criterios de Aceptación

HU- Como **usuario**, quiero **devolver una**
05 **bicicleta** para finalizar mi préstamo.

El sistema registra la hora de devolución y vuelve el estado a "disponible".

HU- Como **usuario**, quiero **ver el costo total**
06 **del préstamo** al devolver la bicicleta.

El sistema calcula el tiempo total y muestra el valor con base en la tarifa.

HU- Como **administrador**, quiero **registrar**
07 **nuevas bicicletas** en el sistema.

Se pueden ingresar bicicletas con todos sus datos y tarifa.

HU- Como **administrador**, quiero **actualizar la**
08 **información** de una bicicleta.

La actualización de datos se refleja inmediatamente en nuevas operaciones.

Diagramas UML

El diseño del sistema contempla el diagrama de clases y el diagrama de casos uso como herramientas visuales esenciales para la comprensión del sistema.

Diagrama de Clases

Sistema de Préstamo de Bicicletas

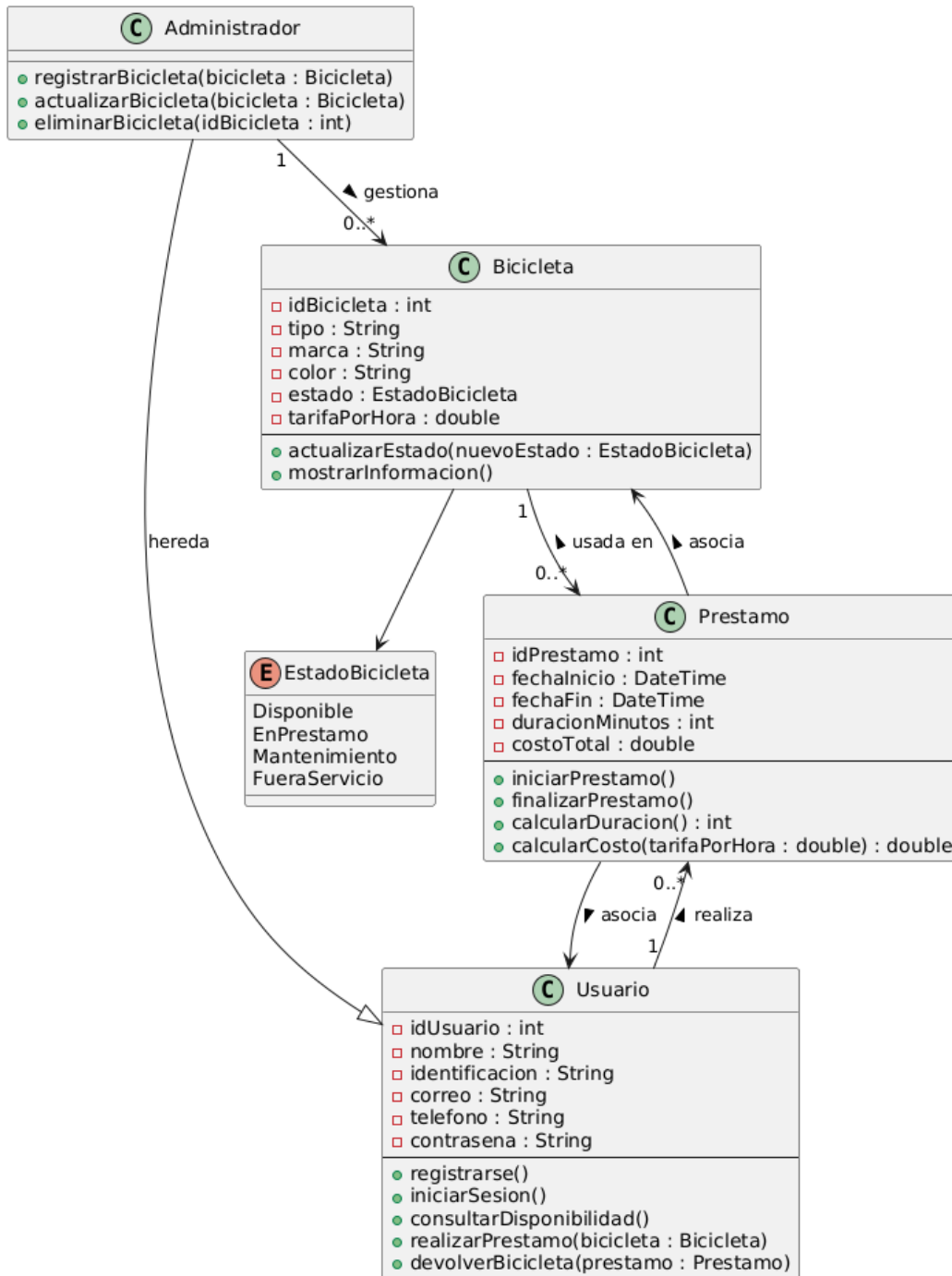


Diagrama de casos de uso

