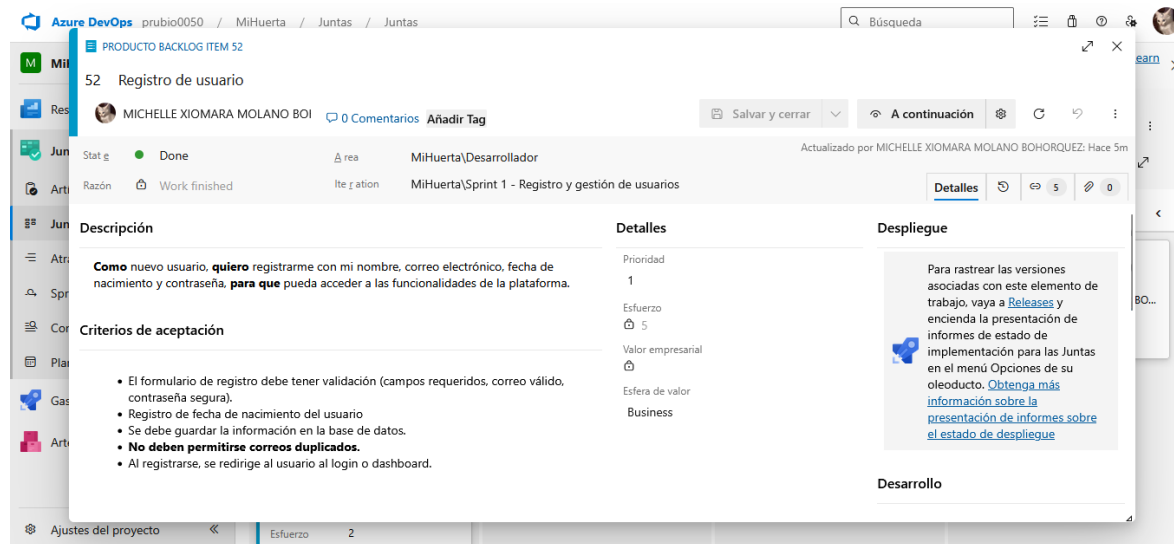


Repositorio del archivo en GitHub

<https://github.com/mxmolano/Evaluacion-DesingSoftware/tree/main>

Historia de usuario



Título: Creación de módulo de gestión de registros de usuario

Como usuario administrador del sistema,

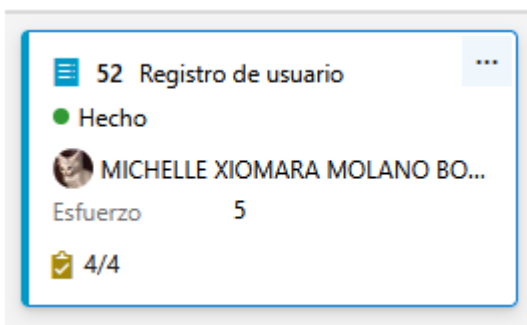
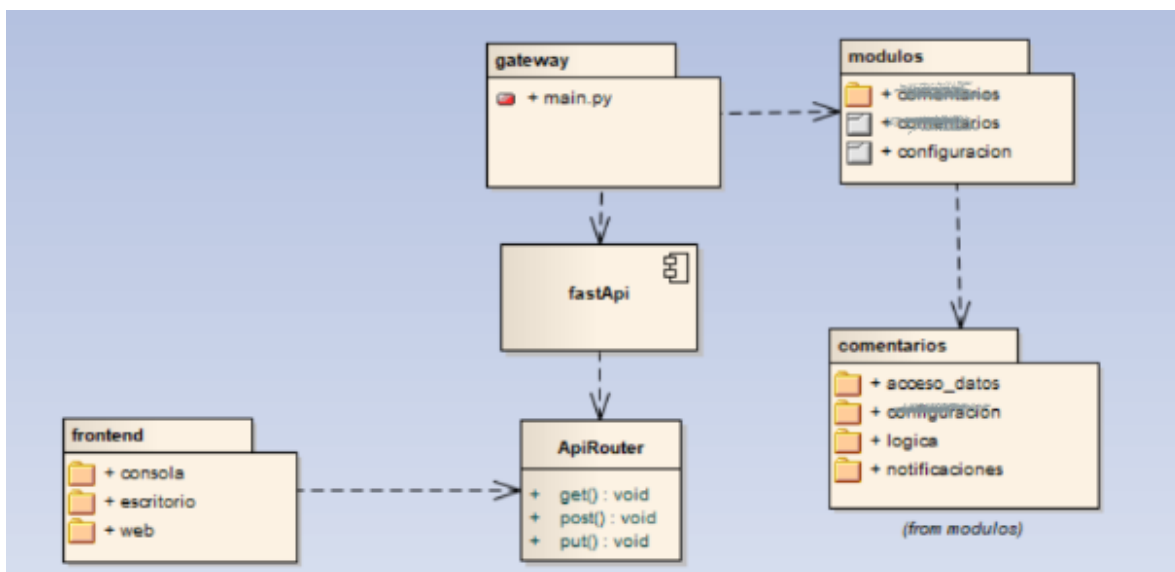
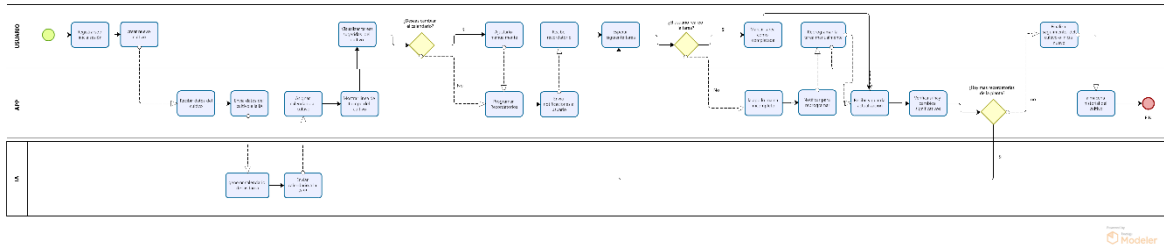
quiero poder crear, leer, actualizar y eliminar registros de usuarios,

para mantener actualizada la información de acceso al sistema.

Criterios de aceptación:

- El módulo debe permitir crear un nuevo usuario con nombre, apellido, fecha de nacimiento, correo electrónico y contraseña.
- Debe ser posible consultar todos los usuarios o un usuario por su ID.
- Se debe permitir la edición de la información de un usuario existente.
- Se debe poder eliminar un usuario.
- La información debe almacenarse y consultarse desde la base de datos configurada en el proyecto.

- Los endpoints deben funcionar desde las tres interfaces: web, consola y escritorio.



Explicación Técnica del Diseño

DAO (Data Access Object)

MICHELLE XIOMARA MOLANO BOHORQUEZ

Archivo: registro_dao.py

Responsable de la conexión y ejecución de consultas SQL sobre la tabla registro.

Funciones implementadas:

- crear_registro(dto): inserta un nuevo usuario.
- obtener_registros(): obtiene todos los registros.
- obtener_registro_por_id(id): busca un registro por ID.
- actualizar_registro(id, dto): actualiza un registro.
- eliminar_registro(id): elimina un registro.

DTO (Data Transfer Object)

Archivo: registro_dto.py

Representa un usuario como objeto con atributos:

- id
- nombre
- apellido
- nacimiento
- email
- contraseña

Sirve para transferir datos entre la capa de acceso a datos y la lógica de negocio.

Service

Archivo: registro_service.py

Valida la información recibida antes de insertarla o actualizarla.

Maneja excepciones y devuelve mensajes claros al router.

Router

Archivo: registro_router.py

Define los endpoints REST:

- POST /registro/
- GET /registro/
- GET /registro/{id}

- PUT /registro/{id}
- DELETE /registro/{id}

Observer (opcional)

En este módulo no se implementó, pero podría usarse para notificar a otros módulos cuando se cree o elimine un registro.

4. Evidencias de Pruebas

Pruebas realizadas desde HTTP Client

- **POST** → Creación de usuario con datos correctos → Respuesta 201 Created.
- **GET** → Listado de usuarios → Devuelve lista JSON con todos los registros.
- **GET por ID** → Devuelve un único registro.
- **PUT** → Actualización de datos → Respuesta 200 OK y datos modificados.
- **DELETE** → Eliminación de usuario → Respuesta 204 No Content.

Pruebas desde Interfaz de Escritorio (Tkinter)

- Creación de usuario desde ventana emergente → registro añadido a la tabla.
- Edición de usuario → datos modificados correctamente.
- Eliminación de usuario → registro quitado de la tabla.