

**Programa educativo**

INGENIERÍA EN SISTEMAS

COMPUTACIONALES

**Grupo**

8 “B”

**Nombre de la materia**

Arquitectura de Servicios

**Nombre del Profesor**

Roberto Suarez Zinzun

**Nombre del alumno**

Roberto Carlos Mejía Aguilera

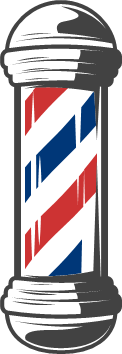
Sergio De Jesús Valdés Peredo

**Nombre del trabajo**

4to Avance del proyecto

Una caricatura de una persona

El contenido generado por IA puede ser incorrecto.



Servicios Por Desarrollar

Estos serán los servicios que haremos en esta entrega de proyecto de la U4

|  |  |  |
| --- | --- | --- |
| Servicio | Operación | Responsable |
| Gestión de Citas | Confirmar Cita | Roberto Carlos Mejía Aguilera |
| Gestión de Citas | Consultar Historial por Usuario | Roberto Carlos Mejía Aguilera |
| Gestión de Barberos | Consultar Barberos por Barbería | Sergio de Jesús Valdés Peredo |
| Gestión de Barberos | Consultar Barbero por ID | Sergio de Jesús Valdés Peredo |
| Gestión de Servicios | Consultar Servicio por ID | Sergio de Jesús Valdés Peredo |
| Gestión de Usuarios | Autenticar Usuario | Sergio de Jesús Valdés Peredo |

Documentación Modelo de Dominio

Diagrama

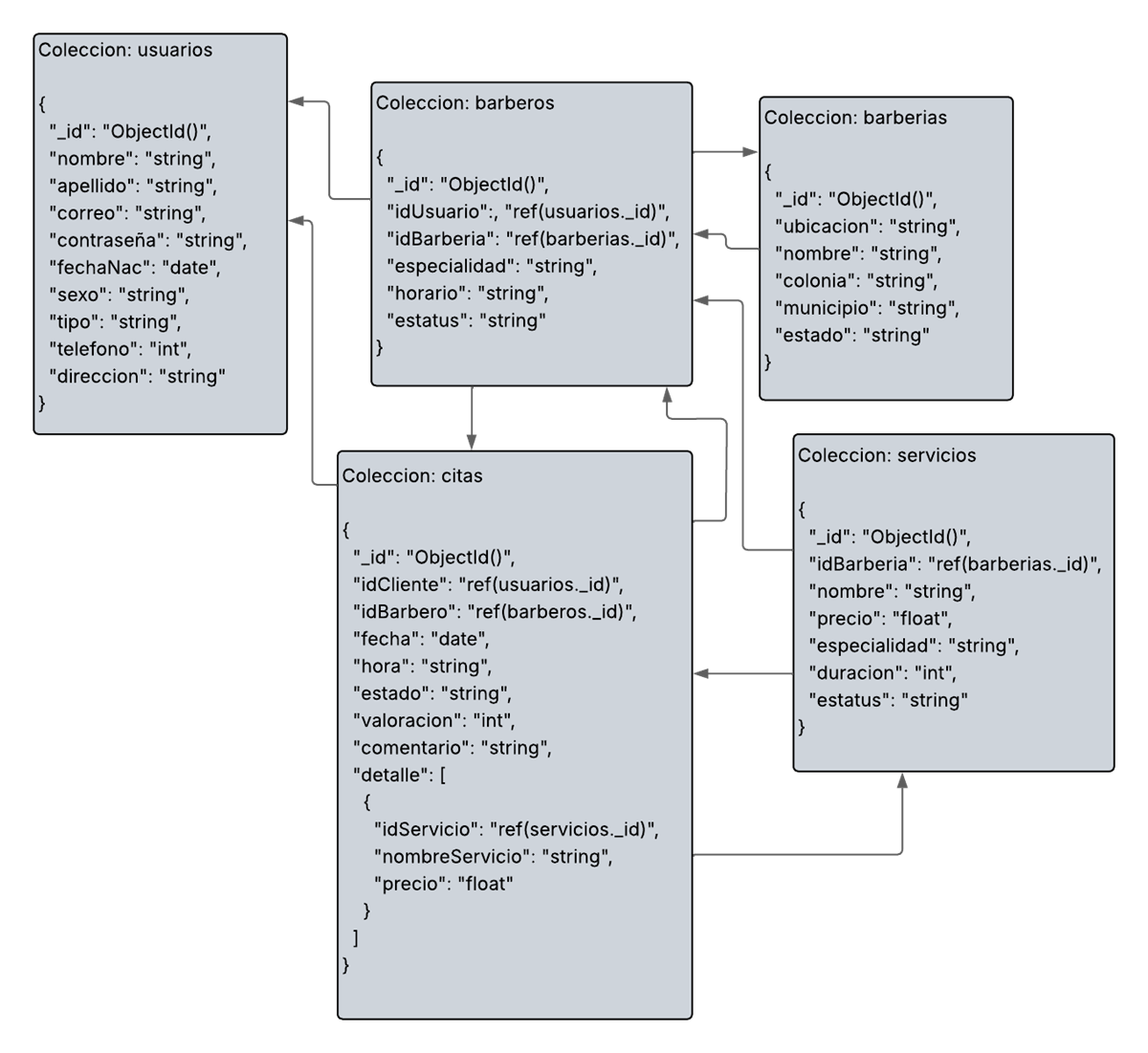
El contenido generado por IA puede ser incorrecto.

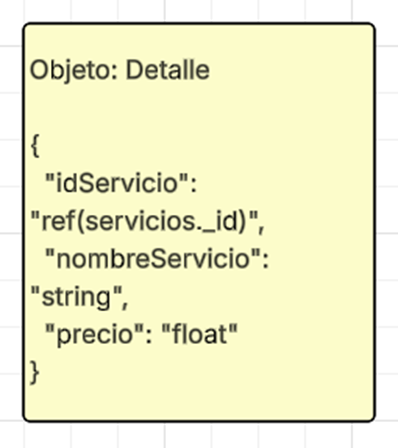
Diagrama de clases

Diagrama

El contenido generado por IA puede ser incorrecto.

Modelo Documental





Servicios Rest “StyleCut”

|  |  |  |
| --- | --- | --- |
| Servicio | Operación | Responsable |
| Gestión de Citas | Agendar Cita | Roberto Carlos Mejía Aguilera |
| Cancelar Cita | Roberto Carlos Mejía Aguilera |
| Consultar Cita por ID | Roberto Carlos Mejía Aguilera |
| Confirmar Cita | Roberto Carlos Mejía Aguilera |
| Consultar Historial por Usuario | Roberto Carlos Mejía Aguilera |
| Gestión de Usuarios | Consultar Usuario por ID | Roberto Carlos Mejía Aguilera/Sergio de Jesús Valdés Peredo |
| Autenticar Usuario | Roberto Carlos Mejía Aguilera/Sergio de Jesús Valdés Peredo |
| Gestión de Servicios | Consultar Servicios por Barbería | Sergio de Jesús Valdés Peredo |
| Consultar Servicio por ID | Sergio de Jesús Valdés Peredo |
| Gestión de Barberías | Consultar Barbería por ID | Sergio de Jesús Valdés Peredo |
| Gestión de Barberos | Consultar Barberos por Barbería | Sergio de Jesús Valdés Peredo |
| Consultar Barbero por ID | Sergio de Jesús Valdés Peredo |

Gestión de Citas

Operación: Gestión de citas

|  |  |
| --- | --- |
| Elemento | Valor |
| Servicio | Gestión de Citas |
| Operación | Agendar Cita |
| Método de Acceso | POST |
| Ruta | /citas |
| Protocolo | HTTP |
| Actor(es) | Cliente |
| Entrada | { "idCliente": int, "idBarbero": int, "fecha": "date", "hora": "string", "servicios": [ "string" ]} |
| Proceso | Registrar una nueva cita en la base de datos. El sistema generará automáticamente: – La fecha de registro actual (fechaRegistro)– El estado inicial de la cita como "Pendiente"Validaciones:1. Que el cliente y barbero existan y estén activos2. Que la cita no se cruce con otra ya existente para el mismo barbero a esa hora |
| Salida | { "estatus": "OK", "mensaje": "Cita agendada con éxito con id: ..."``}` |

Operación: Cancelar Citas

|  |  |
| --- | --- |
| Elemento | Valor |
| Servicio | Gestión de Citas |
| Operación | Cancelar Cita |
| Método de Acceso | PUT |
| Ruta | |  | | --- | | /citas/{id}/cancelar |  |  | | --- | |  | |
| Protocolo | HTTP |
| Actor(es) | Cliente |
| Entrada | |  | | --- | | json "motivo": "string"} |  |  | | --- | |  | |
| Proceso | |  | | --- | | Se actualiza el estado de la cita a "Cancelada" y se registra el motivo enviado por el cliente. Previamente, se valida la existencia del ID de la cita (puede validarse como ObjectId si se usa MongoDB). También se comprueba que el estado actual de la cita no sea "Cancelada" ni "Finalizada". En caso contrario, se retorna un mensaje de error. |  |  | | --- | |  | |
| Salida | json{ "estatus": "ok", "mensaje": "Cita cancelada correctamente."} |

Operación: Consultar Cita por ID

|  |  |
| --- | --- |
| Elemento | Valor |
| Servicio | Gestión de Citas |
| Operación | |  | | --- | | Consultar Cita por ID |  |  | | --- | |  | |
| Método de Acceso | GET |
| Ruta | |  | | --- | | /citas/{id} |  |  | | --- | |  | |
| Protocolo | HTTP |
| Actor(es) | |  | | --- | | Cliente, Barbero, Admin |  |  | | --- | |  | |
| Entrada | |  |  |  | | --- | --- | --- | | |  | | --- | | *ID de la cita (string o int, según implementación), enviado en la URL* |  |  | | --- | |  | |  |  | | --- | |  | |
| Proceso | |  |  |  | | --- | --- | --- | | |  | | --- | | El sistema busca en la base de datos una cita con el ID proporcionado. Se valida que el ID sea correcto (por ejemplo, tipo ObjectId en MongoDB) y que la cita exista. Si no se encuentra, se devuelve un mensaje de error. Si se encuentra, se retorna el objeto completo con la información de la cita. |  |  | | --- | |  | |  |  | | --- | |  | |
| Salida | json{ "estatus": "ok", "mensaje": "Consulta exitosa", "cita": {"id": "string", "fecha": "YYYY-MM-DD", "hora": "HH:MM", "servicio": "string", "barbero": "string", "cliente": "string", "estado": "Agendada" }} |

Operación: Confirmar Cita

|  |  |
| --- | --- |
| Elemento | Valor |
| Servicio | Gestión de Citas |
| Operación | Confirmar Cita |
| Método | PUT |
| Ruta | /citas/{idCita}/confirmar |
| Protocolo | HTTP |
| Actor(es) | Barbero, Administrador |
| Entrada | { "idCita": int } |
| Proceso | Cambia el estado de una cita existente a "Confirmada" si está "Pendiente" |
| Salida | { "estatus": "OK", "mensaje": "Cita confirmada correctamente" } |

Operación: Consultar Historial por Usuario

|  |  |
| --- | --- |
| Elemento | Valor |
| Servicio | Gestión de Citas |
| Operación | Consultar Historial por Usuario |
| Método | GET |
| Ruta | /citas/usuario/{idUsuario} |
| Protocolo | HTTP |
| Actor(es) | Cliente |
| Entrada | { "idUsuario": int } |
| Proceso | Consulta todas las citas de un usuario desde la vista historialUsuarioView. |
| Salida | { "estatus": "OK", "mensaje": "Historial encontrado", "historial": [ ... ] } |

Gestión de Usuarios

Operación: Consultar Usuario por ID

|  |  |
| --- | --- |
| Elemento | Valor |
| Servicio | |  | | --- | | Gestión de Usuarios |  |  | | --- | |  | |
| Operación | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | Consultar Usuario por ID |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |
| Método de Acceso | |  | | --- | | GET |  |  | | --- | |  | |
| Ruta | |  |  |  | | --- | --- | --- | | |  | | --- | | /usuarios/{idUsuario} |  |  | | --- | |  | |  |  | | --- | |  | |
| Protocolo | HTTP |
| Actor(es) | |  |  | | --- | --- | | |  | | --- | |  | |  |  |  |  | | --- | --- | --- | | |  | | --- | | Cliente, Barbero, Admin |  |  | | --- | |  | | |
| Entrada | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | *ID del usuario (int o string) incluido en la URL* |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |
| Proceso | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | |  | | | |  |  | | --- | | El sistema valida el formato del ID y consulta la base de datos para obtener la información del usuario. Si el usuario no existe, se devuelve un mensaje de error. Si se encuentra, se retorna su información personal y de perfil (sin mostrar campos sensibles como contraseña). |  |  | | --- | |  | |  |  | | --- | |  | |
| Salida | json{ "estatus": "ok", "mensaje": "Usuario encontrado", "usuario": { "id": "string", "nombre": "string", "email": "string", "telefono": "string", "tipo": "Cliente/Barbero/Admin"}} |

Operación: Autenticar Usuario

|  |  |
| --- | --- |
| Elemento | Valor |
| Servicio | Gestión de Usuarios |
| Operación | Autenticar Usuario |
| Método | POST |
| Ruta | /usuarios/autenticar |
| Protocolo | HTTP |
| Actor(es) | Cliente / Barbero / Admin |
| Entrada | { "correo": "string", "password": "string" } |
| Proceso | Verifica credenciales en la colección usuarios. |
| Salida | { "estatus": "OK", "usuario": { "nombre": "...", "tipo": "Cliente" } } |

Gestión de Servicios

Operación: Consultar Servicios por Barberia

|  |  |
| --- | --- |
| Elemento | Valor |
| Servicio | |  |  |  | | --- | --- | --- | | |  | | --- | | Gestión de Servicios |  |  | | --- | |  | |  |  | | --- | |  | |
| Operación | |  | | --- | |  | | |  | | --- | | Consultar Servicios por Barbería |  |  | | --- | |  | | | |
| Método de Acceso | |  |  |  | | --- | --- | --- | | |  | | --- | | GET |  |  | | --- | |  | |  |  | | --- | |  | |
| Ruta | |  | | --- | | /servicios/barberia/{idBarberia} |  |  | | --- | |  | |
| Protocolo | HTTP |
| Actor(es) | |  |  | | --- | --- | | |  | | --- | |  | |  |  |  |  | | --- | --- | --- | | |  | | --- | | Cliente, Barbero, Admin |  |  | | --- | |  | | |
| Entrada | |  | | --- | | *ID de la barbería (string o int), enviado en la URL* |  |  | | --- | |  | |
| Proceso | |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | | El sistema consulta la base de datos para obtener todos los servicios asociados a la barbería con el ID indicado. Se valida la existencia de la barbería antes de ejecutar la consulta. En caso de no encontrar servicios o barbería, se responde con un arreglo vacío y un mensaje informativo. | | | | |
| Salida | json{"estatus": "ok", "mensaje": "Servicios encontrados","servicios": [ { "id": "string", "nombre": "string", "descripcion": "string", "duracion": "int (minutos)", "precio": "float"},{...} ]} |

Operación: Consultar Servicio por ID

|  |  |
| --- | --- |
| Elemento | Valor |
| Servicio | Gestión de Servicios |
| Operación | Consultar Servicio por ID |
| Método | GET |
| Ruta | /servicios/{idServicio} |
| Protocolo | HTTP |
| Actor(es) | Cliente |
| Entrada | { "idServicio": int } |
| Proceso | Busca un servicio específico por ID en la colección servicios. |
| Salida | { "estatus": "OK", "servicio": { ... } } |

Gestión de Barberías

Operación: Consultar Barbería por ID

|  |  |
| --- | --- |
| Elemento | Valor |
| Servicio | Gestión de Barberías |
| Operación | Consultar Barbería por ID |
| Método de Acceso | GET |
| Ruta | /barberias/{idBarberia} |
| Protocolo | HTTP |
| Actor(es) | Cliente, Barbero, Admin |
| Entrada | { "idBarberia": int } |
| Proceso | Consulta una barbería específica por ID desde la colección barberias. Devuelve sus datos básicos como nombre, dirección, etc. |
| Salida | { "estatus": "OK", "barberia": { "nombre": "Barbería El Estilo", "direccion": "Av. Siempre Viva 742" } } |

Gestión de Barberos

Operación: Consultar Barberos por Barberia

|  |  |
| --- | --- |
| Elemento | Valor |
| Servicio | Gestión de Barberos |
| Operación | Consultar Barberos por Barbería |
| Método | GET |
| Ruta | /barberos/por-barberia/{idBarberia} |
| Protocolo | HTTP |
| Actor(es) | Cliente, Administrador |
| Entrada | { "idBarberia": int } |
| Proceso | Devuelve todos los barberos asignados a la barbería desde la vista barberosView. |
| Salida | { "estatus": "OK", "barberos": [ ... ] } |

Operación: Consultar Barberos por su ID

|  |  |
| --- | --- |
| Elemento | Valor |
| Servicio | |  | | --- | |  | | Gestión de Barberos | |  |  | | --- | |  | |
| Operación | |  |  |  | | --- | --- | --- | | |  | | --- | | Consultar Barbero por ID |  |  | | --- | |  | |  |  | | --- | |  | |
| Método de Acceso | |  |  |  | | --- | --- | --- | | |  | | --- | | GET |  |  | | --- | |  | |  |  | | --- | |  | |
| Ruta | |  |  |  | | --- | --- | --- | | |  | | --- | | /barberos/{idBarbero} |  |  | | --- | |  | |  |  | | --- | |  | |
| Protocolo | HTTP |
| Actor(es) | |  |  | | --- | --- | | |  | | --- | |  | |  |  |  | | --- | --- | | |  | | --- | | Cliente, Admin | | |
| Entrada | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | *ID del barbero (string o int), enviado en la URL* |  |  | | --- | |  | |  |  | | --- | |  | | | |  |  | | --- | |  | |
| Proceso | |  | | --- | | El sistema valida el ID del barbero y realiza la búsqueda correspondiente en la base de datos. Si se encuentra, retorna su información personal y profesional. En caso de error o inexistencia, se devuelve un mensaje adecuado. |  |  | | --- | |  | |
| Salida | json{ "estatus": "ok", "mensaje": "Barbero encontrado", "barbero": { "id": "string", "nombre": "string", "especialidad": "string","telefono": "string", "barberia": "string" }} |

Actualización del Repositorio

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

ESTRUCTURA DOCUMENTADA – UNIDAD 4: StyleCut API

**1. documentation/**

**Contenido:**

* Proyecto 4 - StyleCut.docx → Documento principal con:
  + Operaciones 7 a 12
  + Tablas REST completas
  + Ejemplos de entrada/salida reales
  + Responsables y rutas
  + Capturas de Swagger
* Documentacion\_Operaciones\_7\_a\_12\_StyleCut.docx → (ya generado)
* StyleCut\_Operaciones\_U4\_con\_Ejemplos.docx → (con ejemplos prácticos de cada ruta)
* README.md → Descripción técnica del proyecto y pasos para ejecutar
* Capturas de Swagger (.png): evidencia visual de pruebas

**Función:**

* Presenta toda la funcionalidad añadida en U4
* Documenta cómo probar la API REST real con ejemplos
* Incluye credenciales, respuestas reales y rutas en uso

**2. BD/**

**Subcarpetas y archivos:**

* conexion.py: Clase Conexion para conectar automáticamente con MongoDB Atlas/local
* colecciones/: Archivos. json actualizados
  + usuarios.json ahora con campo password
  + barberos.json
  + barberias.json
  + servicios.json
  + citas.json
  + detalles.json

**Función:**

* Permite importar fácilmente los datos en Compass para pruebas
* Simula un backend real para pruebas funcionales
* Hace posible validar login y entidades relacionadas sin backend externo

**3. App/ (estructura funcional de la API REST)**

**Subcarpetas:**

**dao/**

Contiene la lógica de cada operación agrupada por entidad, bajo patrón DAO:

| **Archivo** | **Descripción breve** |
| --- | --- |
| CitaDAO.py | Agendar, cancelar, confirmar, consultar historial |
| UsuarioDAO.py | Validar credenciales, consultar usuario por ID |
| ServicioDAO.py | Consultar servicios por barbería o ID |
| BarberiaDAO.py | Obtener barbería por ID |
| BarberoDAO.py | Obtener barberos por barbería o ID |

**models/**

Modelos de entrada y salida para FastAPI, validan estructura JSON:

| **Archivo** | **Contenido** |
| --- | --- |
| CitaModel.py | Entrada para agendar, cancelar y confirmar cita |
| UsuarioModel.py | Entrada para login, salida de datos del usuario |
| ServicioModel.py | Salida de servicios, estructura de detalles |
| BarberiaModel.py | Datos de barberías |
| RespuestaModel.py | Modelo genérico Salida (estatus, mensaje, data) |
| BarberoModel.py | Datos personales, especialidad, horario |

**routers/**

Define los endpoints públicos de la API REST (por entidad):

| **Ruta** | **Archivo correspondiente** |
| --- | --- |
| /citas | citas\_router.py |
| /usuarios | usuarios\_router.py |
| /barberias | barberias\_router.py |
| /servicios | servicios\_router.py |
| /barberos | barberos\_router.py |

Incluyen los métodos GET, POST, PUT necesarios  
Validados con response\_model, retornan salida uniforme

**templates/ + static/ *(nuevo en U4)***

| **Carpeta** | **Contenido** |
| --- | --- |
| templates/index.html | Página de login visual con fondo e input |
| static/images/ | Fondo visual: barber.jpeg |

Login estético que simula acceso visual antes de entrar al /docs

**main.py**

Punto de entrada del sistema.

Incluye:

* Montaje de rutas (con include\_router)
* Montaje de login visual (@app.get("/"))
* Redirección tras login (@app.post("/entrar"))
* Conexión a MongoDB (startup y shutdown)
* Montaje de archivos estáticos (/static) para fondo visual

Valida login por correo y contraseña  
 Redirige a Swagger UI solo si el login es correcto

**MEJORAS AÑADIDAS EN U4**

* 🔐 Autenticación visual + funcional
* 🧾 Documentación extendida con ejemplos reales
* 🔍 Rutas GET y POST 100% probadas con datos reales
* 🧪 Swagger limpio, segmentado por entidad
* 📂 Código modular, mantenible y validado por FastAPI

MATRIZ DE ACCESO – STYLECUT API

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Servicio | Operación | Ruta | Método | Cliente | Barbero | Admin |
| Gestión de Citas | Agendar Cita | /citas | POST | ✔️ | ❌ | ❌ |
| Gestión de Citas | Cancelar Cita | /citas/{idCita}/cancelar | PUT | ✔️ | ❌ | ❌ |
| Gestión de Citas | Confirmar Cita | /citas/{idCita}/confirmar | PUT | ❌ | ✔️ | ✔️ |
| Gestión de Citas | Consultar Cita por ID | /citas/{idCita} | GET | ✔️ | ✔️ | ✔️ |
| Gestión de Citas | Consultar Historial por Usuario | /citas/usuario/{idUsuario} | GET | ✔️ (solo el suyo) | ❌ | ✔️ |
| Gestión de Usuarios | Consultar Usuario por ID | /usuarios/{idUsuario} | GET | ✔️ | ✔️ | ✔️ |
| Gestión de Usuarios | Autenticar Usuario | /usuarios/autenticar | POST | ✔️ | ✔️ | ✔️ |
| Gestión de Barberos | Consultar Barberos por Barbería | /barberos/por-barberia/{idBarberia} | GET | ✔️ | ✔️ | ✔️ |
| Gestión de Barberos | Consultar Barbero por ID | /barberos/{idBarbero} | GET | ❌ | ✔️ | ✔️ |
| Gestión de Servicios | Consultar Servicios por Barbería | /servicios/por-barberia/{idBarberia} | GET | ✔️ | ✔️ | ✔️ |
| Gestión de Servicios | Consultar Servicio por ID | /servicios/{idServicio} | GET | ✔️ | ✔️ | ✔️ |
| Gestión de Barberías | Consultar Barbería por ID | /barberias/{idBarberia} | GET | ✔️ | ✔️ | ✔️ |

**Política aplicada**

* Todos los usuarios deben **autenticarse** (por /usuarios/autenticar) para acceder al sistema.
* Cada usuario tiene un **tipo** (Cliente, Barbero, Admin) que define su acceso.
* El sistema debe validar el tipo antes de permitir acciones sensibles (como confirmar citas).
* En Swagger, si se extiende, esto puede integrarse con un token o sistema de sesión.
* Ya se aplicó seguridad visual con login en /.

Estructura general del proyecto

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

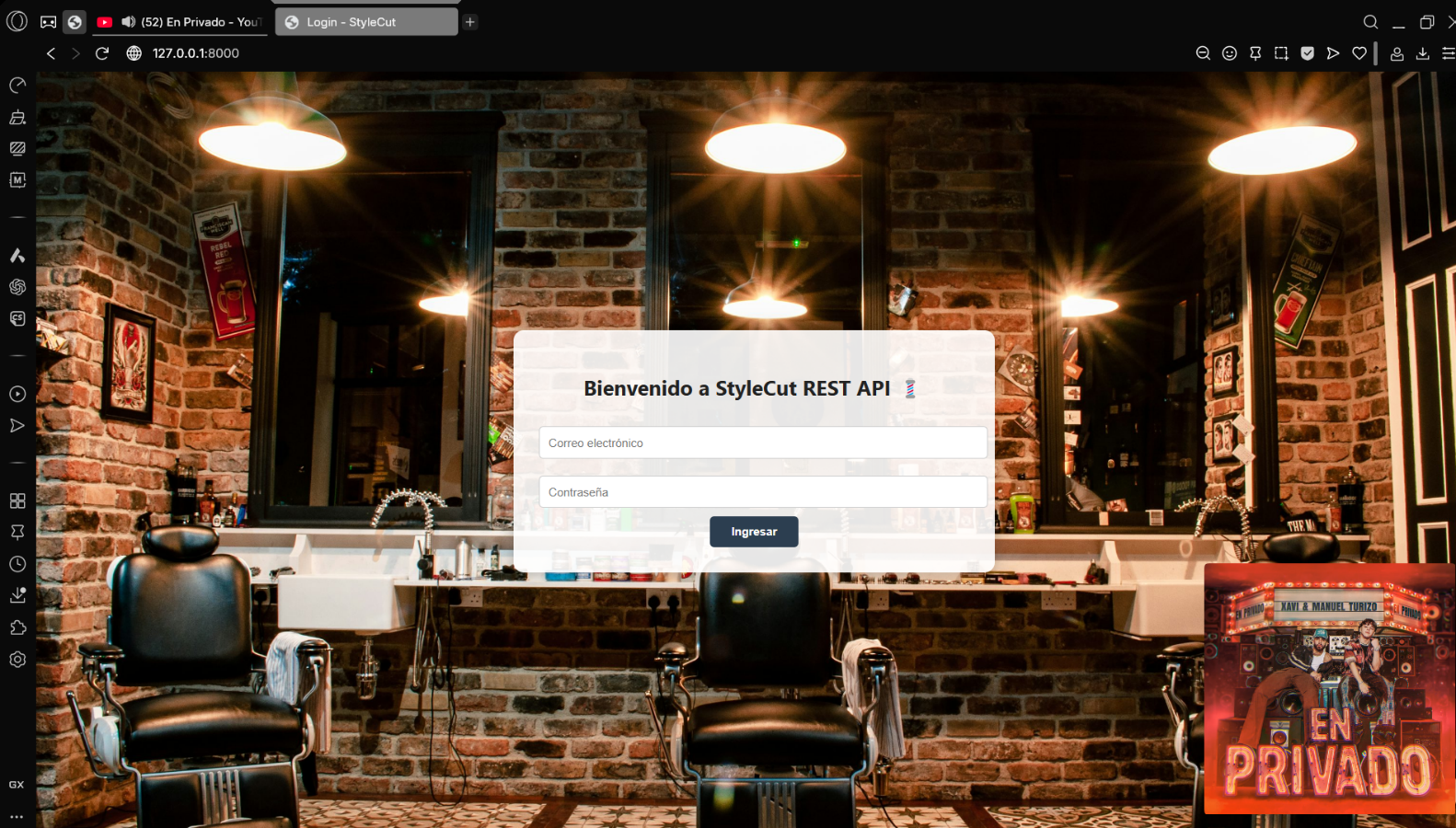
Ejecución del proyecto

Texto

El contenido generado por IA puede ser incorrecto.

Ejecución del proyecto

Primero que nada, decidimos implementar un login, para hacer un mejor diseño



Ahora bien, pasando a Usuarios, ponemos su correo para el método de autenticación

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Forma, Rectángulo

El contenido generado por IA puede ser incorrecto.

Ahora pasando a la consulta de Barberos, ponemos como ID:2 por que ese usuario tiene como rol Barbero, y nos retorna los datos del Barbero

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Ahora bien, consultamos Servicios, por Id de Barberia, en este caso ponemos 1 es el ID de la Barberia

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Ahora ponemos un ID de un servicio para encontrar servicios

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Aquí solo hacemos una cita, como vemos el estatus es pendiente

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

En el servicio de Citas, en la operación de confimar cita agregamos una cita, como vemos el estatus ya es de confirmado

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Ahora con el mismo ID que se nos dio, lo ponemos para cancelar la cita

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Ahora bien, también podemos checar un historial de cita

Texto, Patrón de fondo

El contenido generado por IA puede ser incorrecto.