



PRUEBAS DEL SOFTWARE

Sistema de gestión y visualización de rutas de transporte urbano en tiempo real



Formato pruebas del software

Brayan Estiven Carvajal Padilla

Diego Fernando Cuellar Hernandez

Aprendices

Carlos Julio Cadena

Instructor

ANÁLISIS Y DESARROLLO DE SOFTWARE

FICHA 2900177

SERVICIO NACIONAL DE APRENDIZAJE SENA

CENTRO DE LA INDUSTRIA DE LA EMPRESA Y LOS SERVICIOS

REGIONAL HUILA

2025

Denominación del Programa de Formación: Tecnólogo en Análisis y Desarrollo de Software.

Código del Programa de Formación: 2900177

Nombre del Proyecto: UrbanTracker - Sistema de gestión y visualización de rutas de transporte urbano en tiempo real

Fase del Proyecto: Ejecución

Actividad de Proyecto: Implementar y validar la capa de datos, lógica de negocio y presentación del sistema de información, asegurando su calidad mediante pruebas funcionales y de integración.

Competencia: Construir y validar el sistema que cumpla con los requisitos funcionales y no funcionales de la solución informática.

Resultado de aprendizaje: Ejecutar y documentar las pruebas de software, siguiendo los planes y procedimientos establecidos, garantizando la correcta operación de los servicios y módulos del sistema.

INTRODUCCIÓN

El presente documento tiene como objetivo consolidar las pruebas realizadas al sistema **UrbanTracker**, abarcando sus diferentes componentes y servicios. Estas pruebas permiten validar que el sistema cumpla con los requerimientos funcionales y no funcionales definidos, garantizando su correcto desempeño en escenarios reales.

Las pruebas se han diseñado para evaluar aspectos como la funcionalidad de los módulos, la integración entre componentes, el rendimiento y la seguridad de la aplicación. De esta manera, se busca identificar errores tempranos y asegurar la fiabilidad del sistema antes de su puesta en producción.

Dentro del plan de validación se incluyen los siguientes tipos de pruebas:

- **Pruebas unitarias:** Validan el correcto funcionamiento de métodos y procesos individuales.
- **Pruebas de integración:** Evalúan la comunicación entre los distintos módulos y servicios.
- **Pruebas de sistema:** Comprueban el comportamiento global como una solución completa.

- **Pruebas de aceptación:** Verifican que el sistema responde a los requisitos establecidos por los usuarios finales y cumple con las expectativas planteadas en el proyecto.

DOCUMENTOS DE REFERENCIA

- Especificación OpenAPI de UrbanTracker (JSON)
- Formato base de prueba de software (Word)

Resumen de Endpoints

Método	Ruta	Operación
GET	/api/v1/vehicle-type/{id}	Consultar por ID — Vehicle Type
PUT	/api/v1/vehicle-type/{id}	Actualizar — Vehicle Type
DELETE	/api/v1/vehicle-type/{id}	Eliminar — Vehicle Type
GET	/api/v1/vehicle-assignment/{id}	Consultar por ID — Vehicle Assignment
PUT	/api/v1/vehicle-assignment/{id}	Actualizar — Vehicle Assignment
DELETE	/api/v1/vehicle-assignment/{id}	Eliminar — Vehicle Assignment
GET	/api/v1/user-profile/{id}	Consultar por ID — User Profile
PUT	/api/v1/user-profile/{id}	Actualizar — User Profile
DELETE	/api/v1/user-profile/{id}	Eliminar — User Profile
GET	/api/v1/security/user/{id}	Consultar por ID — User
PUT	/api/v1/security/user/{id}	Actualizar — User
DELETE	/api/v1/security/user/{id}	Eliminar — User
GET	/api/v1/security/role/{id}	Consultar por ID — Role
PUT	/api/v1/security/role/{id}	Actualizar — Role
DELETE	/api/v1/security/role/{id}	Eliminar — Role
GET	/api/v1/security/recovery-	Consultar por ID — Recovery

	request/{id}	Request
PUT	/api/v1/security/recovery-request/{id}	Actualizar — Recovery Request
DELETE	/api/v1/security/recovery-request/{id}	Eliminar — Recovery Request
GET	/api/v1/route-schedule/{id}	Consultar por ID — Route Schedule
PUT	/api/v1/route-schedule/{id}	Actualizar — Route Schedule
DELETE	/api/v1/route-schedule/{id}	Eliminar — Route Schedule
PUT	/api/v1/route-schedule/bulk/{id}	Actualizar — Route Schedule
GET	/api/v1/public/vehicle/{id}	Consultar por ID — Vehicle
PUT	/api/v1/public/vehicle/{id}	Actualizar — Vehicle
DELETE	/api/v1/public/vehicle/{id}	Eliminar — Vehicle
GET	/api/v1/public/user-identification/{id}	Consultar por ID — User Identification
PUT	/api/v1/public/user-identification/{id}	Actualizar — User Identification
DELETE	/api/v1/public/user-identification/{id}	Eliminar — User Identification
GET	/api/v1/public/route-waypoint/{id}	Consultar por ID — Route Waypoint
PUT	/api/v1/public/route-waypoint/{id}	Actualizar — Route Waypoint
DELETE	/api/v1/public/route-waypoint/{id}	Eliminar — Route Waypoint
GET	/api/v1/public/route-trajectory/{id}	Consultar por ID — Route Trajectory
PUT	/api/v1/public/route-trajectory/{id}	Actualizar — Route Trajectory
DELETE	/api/v1/public/route-	Eliminar — Route Trajectory

	trajectorie/{id}	
GET	/api/v1/public/identification-type/{id}	Consultar por ID — Identification Type
PUT	/api/v1/public/identification-type/{id}	Actualizar — Identification Type
DELETE	/api/v1/public/identification-type/{id}	Eliminar — Identification Type
GET	/api/v1/public/driver/{id}	Consultar por ID — Driver
PUT	/api/v1/public/driver/{id}	Actualizar — Driver
DELETE	/api/v1/public/driver/{id}	Eliminar — Driver
GET	/api/v1/public/company/{id}	Consultar por ID — Company
PUT	/api/v1/public/company/{id}	Actualizar — Company
DELETE	/api/v1/public/company/{id}	Eliminar — Company
GET	/api/v1/vehicle-type	Listar — Vehicle Type
POST	/api/v1/vehicle-type	Crear — Vehicle Type
GET	/api/v1/vehicle-assigment	Listar — Vehicle Assigment
POST	/api/v1/vehicle-assigment	Crear — Vehicle Assigment
GET	/api/v1/user-profile	Listar — User Profile
POST	/api/v1/user-profile	Crear — User Profile
GET	/api/v1/security/user	Listar — User
POST	/api/v1/security/user	Crear — User
GET	/api/v1/security/role	Listar — Role
POST	/api/v1/security/role	Crear — Role
GET	/api/v1/security/recovery-request	Listar — Recovery Request
POST	/api/v1/security/recovery-request	Crear — Recovery Request

GET	/api/v1/route/{id}	Consultar por ID — Route
POST	/api/v1/route/{id}	Actualizar — Route
DELETE	/api/v1/route/{id}	Eliminar — Route
POST	/api/v1/route/with-images	Crear — Route
GET	/api/v1/route-schedule	Listar — Route Schedule
POST	/api/v1/route-schedule	Crear — Route Schedule
POST	/api/v1/route-schedule/bulk	Crear — Route Schedule
GET	/api/v1/public/vehicle	Listar — Vehicle
POST	/api/v1/public/vehicle	Crear — Vehicle
GET	/api/v1/public/user-identification	Listar — User Identification
POST	/api/v1/public/user-identification	Crear — User Identification
GET	/api/v1/public/route-waypoint	Listar — Route Waypoint
POST	/api/v1/public/route-waypoint	Crear — Route Waypoint
GET	/api/v1/public/route-trajectory	Listar — Route Trajectory
POST	/api/v1/public/route-trajectory	Crear — Route Trajectory
GET	/api/v1/public/identification-type	Listar — Identification Type
POST	/api/v1/public/identification-type	Crear — Identification Type
GET	/api/v1/public/driver	Listar — Driver
POST	/api/v1/public/driver	Crear — Driver
GET	/api/v1/public/company	Listar — Company
POST	/api/v1/public/company	Crear — Company
POST	/api/v1/public/auth/validate-token	Crear — Auth
POST	/api/v1/public/auth/validate-code	Crear — Auth
POST	/api/v1/public/auth/login	Crear — Auth

POST	/api/v1/public/auth/forgot-password	Crear — Auth
POST	/api/v1/public/auth/change-password	Crear — Auth
GET	/docs	Listar — Scalar
GET	/api/v1/route	Listar — Route
GET	/api/v1/route/{id}/{type}	Consultar por ID — Route
GET	/api/v1/public/route-waypoint/route/{routeId}	Listar — Route Waypoint

Prueba Funcional No 1.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Consultar por ID — Vehicle Type
-----------------	--

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Consultar por ID — Vehicle Type' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a <code>http://localhost:8080/api/v1/vehicle-type/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoOptionalVehicleTypeResDto</code> .



Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()
---------------------------	-----------------------------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 2.



DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Actualizar — Vehicle Type
-----------------	--------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Actualizar — Vehicle Type' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1 Cuerpo (JSON) de ejemplo: { "name": "string-ejemplo" }
Procedimiento de Prueba	1. Preparar la solicitud PUT a http://localhost:8080/api/v1/vehicle-type/{id} 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado	HTTP 200 OK. Estructura conforme al

Esperado	esquema CrudResponseDtoVehicleTypeResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 3.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Eliminar — Vehicle Type
-----------------	------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Eliminar — Vehicle Type' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud DELETE a <code>http://localhost:8080/api/v1/vehicle-type/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoVehicleTypeResDto.
Resultado Obtenido Prueba	¿Prueba exitosa?



	Sí (X) No ()
--	---------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 4.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Consultar por ID — Vehicle Assignment
-----------------	--

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Consultar por ID — Vehicle Assignment' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a <code>http://localhost:8080/api/v1/vehicle-assignment/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoOptionalVehicleAssignmentResDto</code> .
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 5.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Actualizar — Vehicle Assignment
-----------------	--

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Actualizar — Vehicle Assignment' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1 Cuerpo (JSON) de ejemplo: <pre>{ "assignmentStatus": "ACTIVE", "driverId": 1, "vehicleId": 1 }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud PUT a http://localhost:8080/api/v1/vehicle-assignment/{id} 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos,

	campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoVehicleAssignmentResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 6.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Eliminar — Vehicle Assigment
-----------------	---

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Eliminar — Vehicle Assigment' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud DELETE a http://localhost:8080/api/v1/vehicle-assigment/{id} 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoVehicleAssigmentResDto.

Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()
---------------------------	-----------------------------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 7.


DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Consultar por ID — User Profile
-----------------	--

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Consultar por ID — User Profile' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a <code>http://localhost:8080/api/v1/user-profile/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoOptionalUserProfileResDto.

Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()
---------------------------	-----------------------------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 8.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Actualizar — User Profile
-----------------	--------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Actualizar — User Profile' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1 Cuerpo (JSON) de ejemplo: <pre>{ "firstName": "string-ejemplo", "lastName": "string-ejemplo", "email": "string-ejemplo", "phone": "string-ejemplo", "userId": 1 }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud PUT a <code>http://localhost:8080/api/v1/user-profile/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de

	respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoUserProfileResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 9.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Eliminar — User Profile
-----------------	------------------------------------

Detalle de la prueba



Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Eliminar — User Profile' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud DELETE a <code>http://localhost:8080/api/v1/user-profile/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoUserProfileResDto.
Resultado Obtenido Prueba	¿Prueba exitosa?



CENTRO DE LA INDUSTRIA, LA EMPRESA Y LOS SERVICIO
SENA – REGIONAL HUILA



	Sí (X) No ()
--	---------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 10.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Consultar por ID — User
-----------------	------------------------------------



Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Consultar por ID — User' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a <code>http://localhost:8080/api/v1/security/user/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoOptionalUserResDto</code> .
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()



CENTRO DE LA INDUSTRIA, LA EMPRESA Y LOS SERVICIO
SENA – REGIONAL HUILA



Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 11.


DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Actualizar — User
-----------------	------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Actualizar — User' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1 Cuerpo (JSON) de ejemplo: <pre>{ "password": "string-ejemplo", "roleId": 1, "userName": "string-ejemplo" }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud PUT a <code>http://localhost:8080/api/v1/security/user/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).

Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoUserResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 12.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Eliminar — User
-----------------	----------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Eliminar — User' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud DELETE a <code>http://localhost:8080/api/v1/security/user/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoUserResDto</code> .
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()



CENTRO DE LA INDUSTRIA, LA EMPRESA Y LOS SERVICIO
SENA – REGIONAL HUILA



Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 13.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Consultar por ID — Role
-----------------	------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Consultar por ID — Role' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a <code>http://localhost:8080/api/v1/security/role/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoOptionalRoleResDto</code> .
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()



CENTRO DE LA INDUSTRIA, LA EMPRESA Y LOS SERVICIO
SENA – REGIONAL HUILA



Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 14.



DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Actualizar — Role
-----------------	------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Actualizar — Role' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1 Cuerpo (JSON) de ejemplo: { "name": "string-ejemplo" }
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud PUT a <code>http://localhost:8080/api/v1/security/role/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoRoleResDto.

Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()
---------------------------	-----------------------------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025



Prueba Funcional No 15.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Eliminar — Role
-----------------	----------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Eliminar — Role' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud DELETE a <code>http://localhost:8080/api/v1/security/role/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoRoleResDto</code> .
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025



Prueba Funcional No 16.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Consultar por ID — Recovery Request
-----------------	--

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Consultar por ID — Recovery Request' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a <code>http://localhost:8080/api/v1/security/recovery-request/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoOptionalRecoveryRequestResDto</code> .
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 17.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Actualizar — Recovery Request
-----------------	--

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Actualizar — Recovery Request' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1 Cuerpo (JSON) de ejemplo: <pre>{ "code": "string-ejemplo", "expirationTime": "2025-09-29T08:00:25Z", "userId": 1 }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud PUT a <code>http://localhost:8080/api/v1/security/recovery-request/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos)

	obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoRecoveryRequestResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 18.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Eliminar — Recovery Request
-----------------	--

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Eliminar — Recovery Request' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud DELETE a <code>http://localhost:8080/api/v1/security/recovery-request/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoRecoveryRequestResDto</code> .
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 19.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Consultar por ID — Route Schedule
-----------------	--

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Consultar por ID — Route Schedule' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a <code>http://localhost:8080/api/v1/route-schedule/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoOptionalRouteScheduleResDto</code> .
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 20.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Actualizar — Route Schedule
-----------------	--

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Actualizar — Route Schedule' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1 Cuerpo (JSON) de ejemplo: <pre>{ "routeId": 1, "dayOfWeek": "SUNDAY", "startTime": "2025-09-29T08:00:25Z", "endTime": "2025-09-29T08:00:25Z" }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud PUT a <code>http://localhost:8080/api/v1/route-schedule/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de

	respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoRouteScheduleResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 21.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Eliminar — Route Schedule
-----------------	--------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Eliminar — Route Schedule' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud DELETE a <code>http://localhost:8080/api/v1/route-schedule/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoRouteScheduleResDto</code> .

Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()
---------------------------	-----------------------------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 22.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Actualizar — Route Schedule
-----------------	--

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Actualizar — Route Schedule' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1 Cuerpo (JSON) de ejemplo: <pre>[{ "routeId": 1, "dayOfWeek": "SUNDAY", "startTime": "2025-09-29T08:00:25Z", "endTime": "2025-09-29T08:00:25Z" }]</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud PUT a <code>http://localhost:8080/api/v1/route-schedule/bulk/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio).

	7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoListRouteScheduleResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 23.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Consultar por ID — Vehicle
-----------------	---------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Consultar por ID — Vehicle' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a <code>http://localhost:8080/api/v1/public/vehicle/{id}</code> 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: <code>application/json</code> (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoOptionalVehicleResDto</code> .
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 24.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Actualizar — Vehicle
-----------------	---------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Actualizar — Vehicle' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1 Cuerpo (JSON) de ejemplo: <pre>{ "brand": "string-ejemplo", "companyId": 1, "licencePlate": "string-ejemplo", "model": "string-ejemplo", "passengerCapacity": 1, "vehicleTypeId": 1, "year": 1 }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud PUT a <code>http://localhost:8080/api/v1/public/vehicle/{id}</code> 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: <code>application/json</code> (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en

	caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoVehicleResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025



Prueba Funcional No 25.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Eliminar — Vehicle
-----------------	-------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Eliminar — Vehicle' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud DELETE a <code>http://localhost:8080/api/v1/public/vehicle/{id}</code> 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: <code>application/json</code> (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoVehicleResDto</code> .
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 26.

DOCUMENTOS DE REFERENCIA


Versión: v01	Título: Consultar por ID — User Identification
-----------------	---

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Consultar por ID — User Identification' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a <code>http://localhost:8080/api/v1/public/user-identification/{id}</code> 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: <code>application/json</code> (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoOptionalUserIdentificationResDto</code> .
Resultado Obtenido Prueba	¿Prueba exitosa?



	Sí (X) No ()
--	---------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 27.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Actualizar — User Identification
-----------------	---

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Actualizar — User Identification' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1 Cuerpo (JSON) de ejemplo: { "active": true }
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud PUT a http://localhost:8080/api/v1/public/user-identification/{id} 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos,

	campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoUserIdentificationResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 28.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Eliminar — User Identification
-----------------	---

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Eliminar — User Identification' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud DELETE a <code>http://localhost:8080/api/v1/public/user-identification/{id}</code> 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoUserIdentificationResDto</code> .

Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()
---------------------------	-----------------------------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 29.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Consultar por ID — Route Waypoint
-----------------	--

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Consultar por ID — Route Waypoint' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a <code>http://localhost:8080/api/v1/public/route-waypoint/{id}</code> 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: <code>application/json</code> (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoOptionalRouteWaypointResDto</code> .
Resultado Obtenido Prueba	¿Prueba exitosa?



	Sí (X) No ()
--	---------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 30.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Actualizar — Route Waypoint
-----------------	--

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Actualizar — Route Waypoint' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1 Cuerpo (JSON) de ejemplo: <pre>{ "destine": "string-ejemplo", "latitude": 1.0, "longitude": 1.0, "sequence": 1, "type": "string-ejemplo" }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud PUT a <code>http://localhost:8080/api/v1/public/route-waypoint/{id}</code> 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio).

	7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoRouteWaypointResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 31.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Eliminar — Route Waypoint
-----------------	--------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Eliminar — Route Waypoint' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud DELETE a <code>http://localhost:8080/api/v1/public/route-waypoint/{id}</code> 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoRouteWaypointResDto</code> .



Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()
---------------------------	-----------------------------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 32.

DOCUMENTOS DE REFERENCIA


Versión: v01	Título: Consultar por ID — Route Trajectory
-----------------	--

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Consultar por ID — Route Trajectory' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a <code>http://localhost:8080/api/v1/public/route-trajectory/{id}</code> 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: <code>application/json</code> (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoOptionalRouteTrajectoryResDto</code> .
Resultado Obtenido Prueba	¿Prueba exitosa?



	Sí (X) No ()
--	---------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 33.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Actualizar — Route Trajectory
-----------------	--

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Actualizar — Route Trajectory' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1 Cuerpo (JSON) de ejemplo: <pre>{ "routeId": 1, "vehicleId": 1, "startTime": "2025-09-29T08:00:25Z", "endTime": "2025-09-29T08:00:25Z", "trajectoryStatus": "ACTIVE" }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud PUT a <code>http://localhost:8080/api/v1/public/route-trajectory/{id}</code> 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado

	(200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoRouteTrajectoryResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 34.


DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Eliminar — Route Trajectory
-----------------	--

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Eliminar — Route Trajectory' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud DELETE a http://localhost:8080/api/v1/public/route-trajectory/{id} 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoRouteTrajectoryResDto.

Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()
---------------------------	-----------------------------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 35.



DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Consultar por ID — Identification Type
-----------------	---

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Consultar por ID — Identification Type' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a <code>http://localhost:8080/api/v1/public/identification-type/{id}</code> 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: <code>application/json</code> (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoOptionalIdentificationTypeResDto</code> .

Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()
---------------------------	-----------------------------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 36.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Actualizar — Identification Type
-----------------	---

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Actualizar — Identification Type' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1 Cuerpo (JSON) de ejemplo: <pre>{ "active": true }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud PUT a http://localhost:8080/api/v1/public/identification-type/{id} 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).

Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoIdentificationTypeResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 37.

DOCUMENTOS DE REFERENCIA



Versión: v01	Título: Eliminar — Identification Type
-----------------	---

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Eliminar — Identification Type' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud DELETE a http://localhost:8080/api/v1/public/identification-type/{id} 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoIdentificationTypeResDto.
Resultado Obtenido Prueba	¿Prueba exitosa?



	Sí (X) No ()
--	---------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025



Prueba Funcional No 38.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Consultar por ID — Driver
-----------------	--------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Consultar por ID — Driver' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a <code>http://localhost:8080/api/v1/public/driver/{id}</code> 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: <code>application/json</code> (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoOptionalDriverResDto</code> .
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 39.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Actualizar — Driver
-----------------	--------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Actualizar — Driver' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	<p>Parámetros:</p> <ul style="list-style-type: none"> - id (path, obligatorio): integer. Ejemplo: 1 <p>Cuerpo (JSON) de ejemplo:</p> <pre>{ "email": "string-ejemplo", "firstName": "string-ejemplo", "idNumber": "string-ejemplo", "lastName": "string-ejemplo", "password": "string-ejemplo", "phone": "string-ejemplo", "roleId": 1 }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud PUT a <code>http://localhost:8080/api/v1/public/driver/{id}</code> 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: <code>application/json</code> (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en

	caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoDriverResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 40.



DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Eliminar — Driver
-----------------	------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Eliminar — Driver' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud DELETE a <code>http://localhost:8080/api/v1/public/driver/{id}</code> 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: <code>application/json</code> (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoDriverResDto</code> .
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()



Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025


Prueba Funcional No 41.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Consultar por ID — Company
-----------------	---------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Consultar por ID — Company' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a <code>http://localhost:8080/api/v1/public/company/{id}</code> 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: <code>application/json</code> (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoOptionalCompanyResDto</code> .
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 42.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Actualizar — Company
-----------------	---------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Actualizar — Company' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1 Cuerpo (JSON) de ejemplo: <pre>{ "name": "string-ejemplo", "nit": "string-ejemplo", "phone": "string-ejemplo", "email": "string-ejemplo", "country": "string-ejemplo" }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud PUT a <code>http://localhost:8080/api/v1/public/company/{id}</code> 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: <code>application/json</code> (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta

	según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoCompanyResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 43.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Eliminar — Company
-----------------	-------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Eliminar — Company' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud DELETE a <code>http://localhost:8080/api/v1/public/company/{id}</code> 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: <code>application/json</code> (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoCompanyResDto</code> .
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 44.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Listar — Vehicle Type
-----------------	----------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Listar — Vehicle Type' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a http://localhost:8080/api/v1/vehicle-type 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoListVehicleTypeResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()



CENTRO DE LA INDUSTRIA, LA EMPRESA Y LOS SERVICIO
SENA – REGIONAL HUILA



Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 45.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — Vehicle Type
-----------------	---------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — Vehicle Type' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: { "name": "string-ejemplo" }
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/vehicle-type 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoVehicleTypeResDto.

Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()
---------------------------	-----------------------------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 46.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Listar — Vehicle Assignment
-----------------	--

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Listar — Vehicle Assignment' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a http://localhost:8080/api/v1/vehicle-assignment 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoListVehicleAssignmentResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()



Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 47.


DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — Vehicle Assignment
-----------------	---------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — Vehicle Assignment' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: <pre>{ "assignmentStatus": "ACTIVE", "driverId": 1, "vehicleId": 1 }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/vehicle-assignment 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).

Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoVehicleAssignmentResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 48.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Listar — User Profile
-----------------	----------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Listar — User Profile' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a http://localhost:8080/api/v1/user-profile 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoListUserProfileResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()



CENTRO DE LA INDUSTRIA, LA EMPRESA Y LOS SERVICIO
SENA – REGIONAL HUILA



Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 49.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — User Profile
-----------------	---------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — User Profile' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: <pre>{ "firstName": "string-ejemplo", "lastName": "string-ejemplo", "email": "string-ejemplo", "phone": "string-ejemplo", "userId": 1 }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/user-profile 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).

Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoUserProfileResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 50.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Listar — User
-----------------	--------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Listar — User' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a http://localhost:8080/api/v1/security/user 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoListUserResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()



CENTRO DE LA INDUSTRIA, LA EMPRESA Y LOS SERVICIO
SENA – REGIONAL HUILA



Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 51.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — User
-----------------	-------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — User' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: <pre>{ "password": "string-ejemplo", "roleId": 1, "userName": "string-ejemplo" }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/security/user 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoUserResDto.

Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()
---------------------------	-----------------------------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 52.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Listar — Role
-----------------	--------------------------



Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Listar — Role' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a http://localhost:8080/api/v1/security/role 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoListRoleResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()



CENTRO DE LA INDUSTRIA, LA EMPRESA Y LOS SERVICIO
SENA – REGIONAL HUILA



Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 53.

DOCUMENTOS DE REFERENCIA



Versión: v01	Título: Crear — Role
-----------------	-------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — Role' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: { "name": "string-ejemplo" }
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/security/role 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoRoleResDto.
Resultado Obtenido Prueba	¿Prueba exitosa?



	Sí (X) No ()
--	---------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 54.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Listar — Recovery Request
-----------------	--------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Listar — Recovery Request' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a http://localhost:8080/api/v1/security/recovery-request 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoListRecoveryRequestResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()



CENTRO DE LA INDUSTRIA, LA EMPRESA Y LOS SERVICIO
SENA – REGIONAL HUILA



Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 55.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — Recovery Request
-----------------	-------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — Recovery Request' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: { "code": "string-ejemplo", "expirationTime": "2025-09-29T08:00:26Z", "userId": 1 }
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/security/recovery-request 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado	HTTP 200 OK. Estructura conforme al esquema

Esperado	CrudResponseDtoRecoveryRequestResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025



Prueba Funcional No 56.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Consultar por ID — Route
-----------------	-------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Consultar por ID — Route' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a <code>http://localhost:8080/api/v1/route/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoOptionalRouteResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 57.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Actualizar — Route
-----------------	-------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Actualizar — Route' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1 Cuerpo (JSON) de ejemplo: <pre>{ "description": "string-ejemplo", "numberRoute": "string-ejemplo", "outboundImage": "string-ejemplo", "returnImage": "string-ejemplo", "totalDistance": "string-ejemplo", "waypoints": "string-ejemplo" }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a <code>http://localhost:8080/api/v1/route/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de

	respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoRouteResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 58.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Eliminar — Route
-----------------	-----------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Eliminar — Route' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud DELETE a <code>http://localhost:8080/api/v1/route/{id}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoRouteResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()



CENTRO DE LA INDUSTRIA, LA EMPRESA Y LOS SERVICIO
SENA – REGIONAL HUILA



Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 59.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — Route
-----------------	--------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — Route' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: <pre>{ "description": "string-ejemplo", "numberRoute": "string-ejemplo", "outboundImage": "string-ejemplo", "returnImage": "string-ejemplo", "totalDistance": "string-ejemplo", "waypoints": "string-ejemplo" }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/route/with-images 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos,

	campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoRouteResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 60.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Listar — Route Schedule
-----------------	------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Listar — Route Schedule' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a http://localhost:8080/api/v1/route-schedule 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoListRouteScheduleResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 61.



DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — Route Schedule
-----------------	-----------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — Route Schedule' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: <pre>{ "routeId": 1, "dayOfWeek": "SUNDAY", "startTime": "2025-09-29T08:00:26Z", "endTime": "2025-09-29T08:00:26Z" }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/route-schedule 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos,

	campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoRouteScheduleResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 62.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — Route Schedule
-----------------	-----------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — Route Schedule' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - dto (query, obligatorio): array. Ejemplo: [{'routeId': 1, 'dayOfWeek': 'SUNDAY', 'startTime': '2025-09-29T08:00:26Z', 'endTime': '2025-09-29T08:00:26Z'}]
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/route-schedule/bulk 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema

	CrudResponseDtoListRouteScheduleResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 63.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Listar — Vehicle
-----------------	-----------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Listar — Vehicle' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a http://localhost:8080/api/v1/public/vehicle 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoListVehicleResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 64.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — Vehicle
-----------------	----------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — Vehicle' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: <pre>{ "brand": "string-ejemplo", "companyId": 1, "licencePlate": "string-ejemplo", "model": "string-ejemplo", "passengerCapacity": 1, "vehicleTypeId": 1, "year": 1 }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/public/vehicle 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de

	respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoVehicleResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 65.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Listar — User Identification
-----------------	---

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Listar — User Identification' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a http://localhost:8080/api/v1/public/user-identification 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoListUserIdentificationResDto.
Resultado Obtenido Prueba	¿Prueba exitosa?



	Sí (X) No ()
--	---------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 66.


DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — User Identification
-----------------	--

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — User Identification' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: { "active": true }
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/public/user-identification 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado	HTTP 200 OK. Estructura conforme al

Esperado	esquema CrudResponseDtoUserIdentificationResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 67.

DOCUMENTOS DE REFERENCIA



Versión: v01	Título: Listar — Route Waypoint
-----------------	------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Listar — Route Waypoint' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a http://localhost:8080/api/v1/public/route-waypoint 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoListRouteWaypointResDto.
Resultado Obtenido Prueba	¿Prueba exitosa?



	Sí (X) No ()
--	---------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 68.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — Route Waypoint
-----------------	-----------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — Route Waypoint' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: <pre>{ "destine": "string-ejemplo", "latitude": 1.0, "longitude": 1.0, "sequence": 1, "type": "string-ejemplo" }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/public/route-waypoint 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos,

	campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoRouteWaypointResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 69.

DOCUMENTOS DE REFERENCIA



Versión: v01	Título: Listar — Route Trajectory
-----------------	--------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Listar — Route Trajectory' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a http://localhost:8080/api/v1/public/route-trajectory 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoListRouteTrajectoryResDto.
Resultado Obtenido Prueba	¿Prueba exitosa?



	Sí (X) No ()
--	---------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 70.


DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — Route Trajectory
-----------------	-------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — Route Trajectory' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: <pre>{ "routeId": 1, "vehicleId": 1, "startTime": "2025-09-29T08:00:26Z", "endTime": "2025-09-29T08:00:26Z", "trajectoryStatus": "ACTIVE" }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/public/route-trajectory 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de

	respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoRouteTrajectoryResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 71.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Listar — Identification Type
-----------------	---

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Listar — Identification Type' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a http://localhost:8080/api/v1/public/identification-type 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoListIdentificationTypeResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 72.



DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — Identification Type
-----------------	--

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — Identification Type' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: { "active": true }
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/public/identification-type 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoIdentificationTypeResDto.

Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()
---------------------------	-----------------------------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025


Prueba Funcional No 73.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Listar — Driver
-----------------	----------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Listar — Driver' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a http://localhost:8080/api/v1/public/driver 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoListDriverResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 74.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — Driver
-----------------	---------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — Driver' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: <pre>{ "email": "string-ejemplo", "firstName": "string-ejemplo", "idNumber": "string-ejemplo", "lastName": "string-ejemplo", "password": "string-ejemplo", "phone": "string-ejemplo", "roleId": 1 }</pre>
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/public/driver 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de

	respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoDriverResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 75.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Listar — Company
-----------------	-----------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Listar — Company' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a http://localhost:8080/api/v1/public/company 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoListCompanyResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()



Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 76.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — Company
-----------------	----------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — Company' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: { "name": "string-ejemplo", "nit": "string-ejemplo", "phone": "string-ejemplo", "email": "string-ejemplo", "country": "string-ejemplo" }
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/public/company 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos)

	obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoCompanyResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 77.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — Auth
-----------------	-------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — Auth' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - Authorization (header, opcional): string. Ejemplo: string-ejemplo
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/public/auth/validate-token 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al contrato de respuesta.

Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()
---------------------------	-----------------------------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 78.


DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — Auth
-----------------	-------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — Auth' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: { "email": "string-ejemplo", "code": "string-ejemplo" }
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/public/auth/validate-code 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado	HTTP 200 OK. Estructura conforme al contrato de

Esperado	respuesta.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 79.


DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — Auth
-----------------	-------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — Auth' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: { "userName": "string-ejemplo", "password": "string-ejemplo" }
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/public/auth/login 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al contrato de respuesta.

Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()
---------------------------	-----------------------------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 80.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — Auth
-----------------	-------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — Auth' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: { "email": "string-ejemplo" }
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/public/auth/forgot-password 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema ForgotPasswordResponseDTO.

Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()
---------------------------	-----------------------------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 81.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Crear — Auth
-----------------	-------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Crear — Auth' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Cuerpo (JSON) de ejemplo: { "email": "string-ejemplo", "newPassword": "string-ejemplo" }
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud POST a http://localhost:8080/api/v1/public/auth/change-password 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado	HTTP 200 OK. Estructura conforme al contrato de

Esperado	respuesta.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 82.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Listar — Scalar
-----------------	----------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Listar — Scalar' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a http://localhost:8080/docs 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al contrato de respuesta.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()



CENTRO DE LA INDUSTRIA, LA EMPRESA Y LOS SERVICIO
SENA – REGIONAL HUILA



Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 83.

DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Listar — Route
-----------------	---------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Listar — Route' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a http://localhost:8080/api/v1/route 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema CrudResponseDtoListRouteResDto.
Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()



CENTRO DE LA INDUSTRIA, LA EMPRESA Y LOS SERVICIO
SENA – REGIONAL HUILA



Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 84.


DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Consultar por ID — Route
-----------------	-------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Consultar por ID — Route' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - id (path, obligatorio): integer. Ejemplo: 1 - type (path, obligatorio): string. Ejemplo: string-ejemplo
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a <code>http://localhost:8080/api/v1/route/{id}/{type}</code> 2. Añadir cabecera Authorization: Bearer <JWT> (según configuración de seguridad). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoRouteDetailsResDto</code> .
Resultado Obtenido Prueba	¿Prueba exitosa?

	Sí (X) No ()
--	---------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025

Prueba Funcional No 85.


DOCUMENTOS DE REFERENCIA

Versión: v01	Título: Listar — Route Waypoint
-----------------	------------------------------------

Detalle de la prueba

Fecha de realización	29/09/2025
Duración de la prueba	
Requerimientos Funcional de la prueba	Validar la operación 'Listar — Route Waypoint' sobre el recurso correspondiente.
Objetivo	Comprobar que el servicio responde con HTTP 200 y que la estructura de los datos coincide con el contrato.
Tipo de Prueba	Funcional (API) / Caja negra.
Datos de entrada de la prueba	Parámetros: - routeId (path, obligatorio): integer. Ejemplo: 1
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Preparar la solicitud GET a <code>http://localhost:8080/api/v1/public/route-waypoint/route/{routeId}</code> 2. Este endpoint es público; no requiere autenticación (a menos que el despliegue lo especifique). 3. Establecer encabezados: Content-Type: application/json (cuando aplique). 4. Incluir el cuerpo de la solicitud con los datos de ejemplo (cuando aplique). 5. Enviar la solicitud. 6. Verificar el código de estado esperado (200 en caso satisfactorio). 7. Validar la estructura del cuerpo de respuesta según el contrato (tipos de datos, campos obligatorios y valores permitidos).
Datos de salida - Resultado Esperado	HTTP 200 OK. Estructura conforme al esquema <code>CrudResponseDtoListRouteWaypointReqDto</code> .

Resultado Obtenido Prueba	¿Prueba exitosa? Sí (X) No ()
---------------------------	-----------------------------------

Realizado por:	Firma	Fecha
Diego Fernando Cuellar Hernandez		16/09/2025
Aprobado por:	Firma	Fecha
Brayan Estiven Carvajal Padilla		17/09/2025