

Sprint 1: Configuración inicial y CRUD de productos

El sistema está dirigido a usuarios registrados o visitantes que deseen consultar, reservar o administrar vehículos, así como a administradores que gestionan el catálogo de productos.

Funcionalidades desarrolladas:

- Registro de productos desde el panel de administrador
- Listado de productos públicos y administrativos
- Eliminación y edición de productos

2. Identidad de marca

- Logo: logo-rentaya-amarillo.png (ubicado en carpeta assets del frontend)



- Paleta de colores:
 - Amarillo: #ffc107
 - Gris oscuro: #212121
 - Blanco: #ffffff
 - Gris claro:

#f5f5f5 3. Planificación y ejecución de

tests Casos de prueba diseñados:

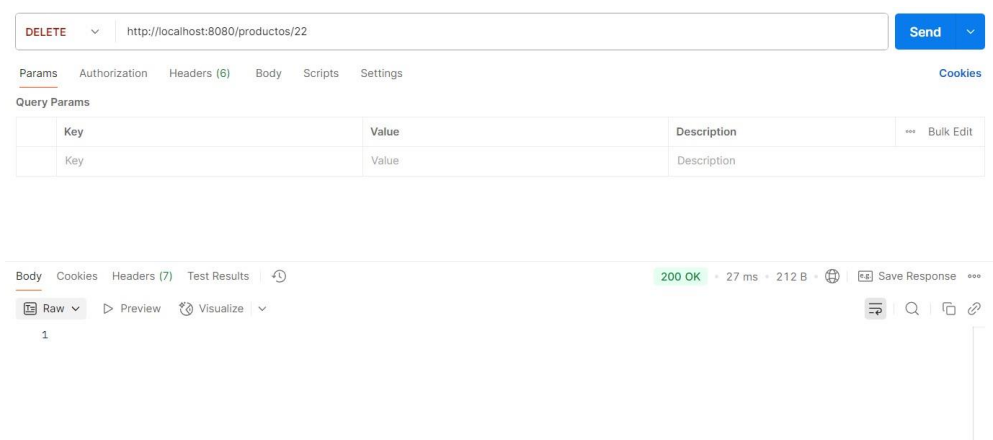
- Registro de productos con campos válidos y no válidos

```

1 POST http://localhost:8080/productos
2 Params Authorization Headers (9) Body Scripts Settings
3 none form-data x-www-form-urlencoded raw binary GraphQL JSON
4
5 {
6   "nombre": "Auto grandote",
7   "descripcion": "un vehiculo familiar",
8   "imagenUrl": "https://ejemplo.com/auto_grande.jpg",
9   "categoria": null
10 }
11
12 Body Cookies Headers (7) Test Results
13 500 Internal Server Error - 77 ms - 15.1
14 {} JSON Preview Visualize
15
16 {
17   "timestamp": "2025-06-24T09:01:54.823+00:00",
18   "status": 500,
19   "error": "Internal Server Error",
20   "trace": "org.springframework.dao.DataIntegrityViolationException: could not execute statement [cannot add foreign key constraint fails ('rentaya', 'producto', CONSTRAINT 'FKodqr796b0k9rwqj1utiamt0m' FOREIGN K REFERENCES 'categoria' ('id'))] (insert into producto (categoria_id,descripcion,imagen_url,nombre) val into producto (categoria_id,descripcion,imagen_url,nombre) values (?,?,?,?)); constraint [null]\n\nat o vendor.HibernateJpaDialect.convertHibernateAccessException(HibernateJpaDialect.java:296)\n\nat o vendor.HibernateJpaDialect.translateExceptionIfPossible(HibernateJpaDialect.java:241)\n\nat org.spr AbstractEntityManagerFactoryBean.translateExceptionIfPossible(AbstractEntityManagerFactoryBean.java:55 springframework.dao.support.ChainedPersistenceExceptionTranslator.translateExceptionIfPossible (ChainedPersistenceExceptionTranslator.java:61)\n\nat org.springframework.dao.support.DataAccessUtil

```

- Eliminación de producto existente



- Edición de producto con datos modificados

```
{
  "id": 21,
  "nombre": "Auto grandote",
  "descripcion": "Un vehículo familiar",
  "imagenUrl": "https://ejemplo.com/auto_grande.jpg",
  "categoria": {
    "id": 7,
    "titulo": "Clásico",
    "descripcion": "De 25 años o más que ha resistido la prueba del",
    "imagenUrl": "https://blog.autochilango.com/wp-content/uploads/",
    "productos": [
      {
        "id": 21,
        "nombre": "Auto grandote",
        "descripcion": "Un vehículo familiar",
        "imagenUrl": "https://ejemplo.com/auto_grande.jpg",
        "categoria": {
          "id": 1,
          "titulo": null,
          "descripcion": null,
          "imagenUrl": null,
          "productos": null
        }
      }
    ]
  }
}
```

ID	Nombre de HU	Objetivo de prueba	Resultado esperado
TC1	HU1: Colocar encabezado	Verificar que el encabezado aparece fijo en todas las páginas	El header es visible, ocupa el 100%, y es fijo al hacer scroll
TC2	HU1: Colocar encabezado	Hacer clic en el logo del encabezado	El usuario es redirigido a la página principal
TC3	HU2: Definir el cuerpo del sitio	Verificar el color de fondo y la disposición del main	El color es armónico con la marca, y se ven las secciones: buscador, categorías, etc.
TC4	HU3: Registrar producto	Agregar un nuevo producto desde el panel	El producto se guarda en la base de datos y aparece en el listado
TC5	HU3: Registrar producto	Intentar registrar un producto con nombre duplicado	Muestra mensaje de error: "El nombre ya está en uso"
TC6	HU4: Visualizar productos en el home	Ingresar al home y observar los productos	Se muestran máximo 10 productos aleatorios sin repetirse

TC7	HU5: Visualizar detalle de producto	Ingresar al detalle de un producto desde el home	Se muestra el título, descripción, imágenes, y flecha para volver
TC8	HU6: Visualizar galería de imágenes	Observar la disposición de imágenes del producto	1 imagen grande y 4 pequeñas con botón "Ver más"
TC9	HU7: Colocar pie de página	Verificar la existencia del footer en todas las páginas	El footer tiene logo, año y copyright visibles y adaptables a pantalla
TC10	HU8: Pagar productos	Ingresar al home y navegar entre páginas	Se muestran 10 productos por página con navegación funcional
TC11	HU9: Panel de administración	Acceder a /administracion desde desktop	Se muestra el menú del panel
TC12	HU9: Panel de administración	Acceder a /administracion desde celular	Se muestra un mensaje que dice "no disponible en móvil"
TC13	HU10: Listar productos	Hacer clic en "Lista de productos" desde el panel admin	Se listan todos los productos con columnas ID, Nombre y Acciones
TC14	HU11: Eliminar producto	Hacer clic en eliminar un producto	Muestra mensaje de confirmación y elimina el producto si se acepta
TC15	HU11: Eliminar producto	Cancelar la acción de eliminar producto	El producto no se elimina

Resultados: Los tests fueron exitosos y se ejecutaron tanto en backend con Postman como en frontend.

4. Repositorio

Código disponible en el repositorio público: <https://github.com/frishlin/rentaya>

Sprint 2: Registro/Login de usuarios y gestión por categorías

En este sprint se añadió la funcionalidad de registro e inicio de sesión de usuarios, con validación por correo y almacenamiento de sesión en localStorage. Se introdujo el CRUD de categorías y la vinculación de productos con una categoría específica.

Funcionalidades desarrolladas:

- Registro e inicio de sesión
- Validación de email duplicado
- Cierre de sesión
- Registro de productos con categoría
- Listado de categorías

2. Planificación y ejecución de tests

ID	Nombre de HU	Objetivo de prueba	Resultado esperado
TC16	HU12: Categorizar productos	Verificar que se puede asignar categoría al registrar o editar producto	El producto queda guardado con la categoría asignada
TC17	HU13: Registrar usuario	Verificar el registro con datos válidos	Se registra correctamente y se redirige o muestra confirmación
TC18	HU14: Identificar usuario	Verificar inicio de sesión con datos válidos	Se accede al sistema y se muestra nombre del usuario
TC19	HU15: Cerrar sesión	Verificar cierre de sesión	Se cierra sesión y se vuelve a modo anónimo
TC20	HU16: Identificar administrador	Verificar que se puede otorgar rol admin	El usuario con el correo electrónico asignado como ADMIN, puede entrar al panel de administración

TC21	HU17: Administrar característica de producto	Verificar añadir, editar y eliminar características	La característica aparece en la lista y se puede asociar a productos
TC22	HU18: Visualizar características al producto	Verificar visualización del bloque de características	Se ve el bloque con íconos y nombres de características
TC22	HU19: Notificación: Confirmación de registro de usuario	Verificar que se notifique que el registro del usuario fue exitoso	El sistema muestra un mensaje de registro exitoso.
TC24	HU20: Crear sección de categorías	Verificar que se pueda crear nueva categoría	La categoría se agrega y aparece en el sistema
TC25	HU21: Agregar categoría	Como administrador, quiero crear nuevas categorías de productos	

Resultados: Los tests fueron exitosos y se ejecutaron tanto en backend con Postman como en frontend.

Sprint 3: Reservas, favoritos, filtros

En este sprint se implementó el sistema de reservas por usuario, permitiendo seleccionar un rango de fechas y consultar disponibilidad. También se añadieron productos favoritos asociados por sesión y la visualización de filtros combinados por nombre, categoría o fechas.

Funcionalidades desarrolladas:

- Reservar producto y ver reservas por usuario
- Cancelar reserva
- Agregar/quitar productos favoritos
- Filtro por categoría y nombre
- Filtro por fechas disponibles

ID	Nombre de HU	Objetivo de prueba	Resultado esperado
TC26	HU22: Realizar búsqueda	Verificar que se puede realizar una búsqueda por nombre de vehículo	Si el producto existe, será mostrado al usuario.
TC27	HU23: Visualizar disponibilidad	Verificar la disponibilidad del producto.	Si el producto existe, en los detalles se visualizarán las fechas en que el auto está reservado.
TC28	HU24: Marcar como favorito	Verificar que el usuario pueda marcar autos como favoritos.	Al hacer clic sobre el botón Agregar a favoritos, el producto se agregará a la lista de favoritos del usuario.
TC29	HU25: Listar productos favoritos	Verificar que el usuario pueda visualizar su lista de favoritos.	Al hacer clic sobre el menú Favoritos, el usuario podrá visualizar los productos marcados como favoritos.

TC30	HU26: Ver bloque de políticas del producto	Verificar que el usuario pueda ingresar a las políticas del producto.	Funcionalidad no realizada.
TC31	HU27: Redes: compartir productos	Verificar que los usuarios puedan compartir recomendar por redes sociales, los productos que les gusten.	Funcionalidad no realizada.
TC32	HU28: Puntuar producto	Verificar que los usuarios que hayan finalizado una reserva, puedan puntuar el producto.	Funcionalidad no realizada.
TC33	HU29: Eliminar categoría.	Verificar que el admin puede eliminar categorías sin productos asociados.	Al oprimir el botón Eliminar, se elimina la categoría.
TC34	HU29 Eliminar productos	Verificar que el admin puede eliminar categorías aun con productos asociados.	Al oprimir el botón Eliminar, el sistema advierte que tiene productos asociados y si se confirma la eliminación, se elimina la categoría y los productos asociados.

2. Planificación y ejecución de tests

Resultados: Todos los casos fueron ejecutados con éxito en entorno local.

SPRINT 4

En este sprint se implementan las funcionalidades relacionadas con el proceso de reserva por parte del usuario. El sistema debe permitir crear nuevas reservas, consultar reservas previas, cancelar una reserva y filtrar productos por disponibilidad en base a fechas.

Funcionalidades desarrolladas:

- Creación del endpoint POST /reservas para registrar una nueva reserva.
- Validación de disponibilidad del producto antes de reservar.
- Implementación del endpoint GET /reservas/usuario/{email} para que el usuario consulte sus reservas.
- Implementación del endpoint DELETE /reservas/{id} para cancelar una reserva.
- Filtro de productos disponibles por fechas usando GET /productos/disponibles?inicio=...&fin=...
- Pruebas con Postman y frontend para cada caso.

ID	Nombre de HU	Objetivo de prueba	Resultado esperado
TC35	HU30: Reservas: seleccionar fecha.	Verifica que el usuario puede realizar búsquedas por fecha.	Al ingresar un periodo de fechas, se mostrarán los productos que estén disponibles en ese periodo.
TC36	HU31: Reservas: visualizar detalles.	Verificar que el usuario pueda acceder a los detalles del producto donde pueda realizar reserva.	El usuario al ingresar a los detalles del producto puede realizar una reserva seleccionando las fechas disponibles.
TC37	HU32: Realizar reserva.	Verificar que el usuario autenticado pueda realizar reservas.	Al hacer clic en el botón reserva, si el usuario está logueado, se realizará la reserva, de lo contrario, lo llevará a la vista de iniciar sesión.

TC38	HU33: Acceder al historial.	Verificar que el usuario pueda visualizar sus reservas.	El usuario autenticado, al hacer clic sobre el menú Mis reservas, podrá visualizar su historial de reservas.
TC39	HU34: Whatsapp: iniciar chat.	Verificar que el usuario se pueda contactar con el proveedor del producto a través de Whatsapp.	Funcionalidad no realizada.
TC40	HU35: Notificación: Confirmar reserva por correo.	Verificar que el usuario registrado pueda recibir una confirmación de su reserva por correo electrónico.	Funcionalidad no realizada.