**DOCUMENTO DE CASOS DE PRUEBA PARA API CONTACTO**

**Índice**

1. Introducción
2. Objetivos
3. Pre-requisitos
4. Casos de prueba
   1. Caso de prueba 1: Crear un contacto
   2. Caso de prueba 2: Obtener todos los contactos
   3. Caso de prueba 3: Obtener un contacto por ID
   4. Caso de prueba 4: Actualizar un contacto
   5. Caso de prueba 5: Eliminar un contacto
5. Criterios de Aceptación
6. Conclusión

**INTRODUCCIÓN**

Este documento detalla los casos de prueba de la aplicación Contacto API, la cual permite realizar operaciones CRUD (Crear, Leer, Actualizar y Eliminar).

**OBJETIVOS**

* Validar que las operaciones CRUD sobre los contactos funcionan correctamente.
* Asegurar que la API responde adecuadamente a las solicitudes
* Verificar que se manejan correctamente los errores y las excepciones

**PRE-REQUISITOS**

* La aplicación Contactos API debe estar en funcionamiento en <http://localhost:8081>
* Una herramienta para realizar peticiones http, como Postman o cURL (se recomienda el uso de Postman)
* Conocimientos básicos de HTTP y JSON

**CASOS DE PRUEBA**

**Caso de prueba 1: Crear un contacto**

**Id del caso de prueba**: CP01

**Descripción**: validar que se puede crear un contacto correctamente

**Pasos:**

1. Realizar una solicitud ‘**POST**’ a la URL: ‘**/contactos**’ con el siguiente cuerpo:

{

"nombre": "EjemploNombre",

"apellidos": "EjemploApellidos",

"email": "ejemplo@ejemplo.com"

}

1. Verificar que la respuesta tiene un código de estado: ‘**201 created**’
2. Verificar que el cuerpo de la respuesta contiene los detalles del contacto creado, incluyendo un ‘**id**’ generado.

**Resultado esperado**

* Código de estado ‘**201 created**’
* El cuerpo de la respuesta contiene los detalles del contacto creado

**Caso de prueba 2: Obtener todos los contactos**

**Id del caso de prueba:** CP02

**Descripción:** validar que se pueden obtener todos los contactos

**Pasos:**

1. Realizar una solicitud ‘**GET**’ a la URL: ‘**/contactos**’
2. Verificar que la respuesta tiene un código de estado: ‘**200 ok**’
3. Verificar que el cuerpo de la respuesta contiene una lista de contactos

**Resultado esperado:**

* Código de estado ‘**200 ok**’
* El cuerpo de la respuesta contiene una lista de contactos

**Caso de prueba 3: Obtener un contacto por ID**

**Id del caso de prueba:** CP03

**Descripción:** validar que se puede obtener un contacto por su ID

**Pasos:**

1. Realizar una solicitud ‘**GET**’ a la URL: ‘**/contactos/{id}**’, donde ‘**{id}**’ es el ID de un contacto existente.
2. Verificar que la respuesta tiene un código de estado ‘**200 ok**’
3. Verificar que el cuerpo de la respuesta contiene los detalles del contacto solicitado

**Resultado esperado:**

* Código de estado ‘**200 ok**’
* El cuerpo de la respuesta contiene los detalles del contacto solicitado.

**Caso de prueba 4: Actualizar un contacto**

**Id del caso de prueba:** CP04

**Descripción:** validar que se puede actualizar un contacto existente

**Pasos:**

1. Realizar una solicitud ‘**PUT**’ a la URL: ‘**/contactos**/**{id}**’, donde ‘**{id}**’ es el ID de un contacto existente.

{

"nombre": "EjemploNombre",

"apellidos": "EjemploApellidos",

"email": "ejemplo@ejemplo.com"

}

1. Verificar que la respuesta tiene un código de estado ‘**200 ok**’
2. Verificar que el cuerpo de la respuesta contiene los detalles del contacto actualizado.

**Resultado esperado:**

* Código de estado: ‘**200 ok**’
* El cuerpo de la respuesta contiene los detalles del contacto actualizado.

**Caso de prueba 5: Eliminar un contacto**

**Id del caso de prueba:** CP05

**Descripción:** Validar que se puede eliminar un contacto existente

**Pasos:**

1. Realizar una solicitud ‘**DELETE**’ a la URL: ‘**/contactos/{id}**’, donde ‘**{id}**’ es el ID de un contacto existente.
2. Verificar que la respuesta tiene un código de estado ‘**204 No Content**’

**Resultado esperado**

* Código de estado: ‘**204 No Content**’

**Criterios de Aplicación**

* Todos los casos de prueba deben pasar sin errores
* Las respuestas deben contener los códigos de estado y los cuerpos esperados
* Los errores deben manejarse adecuadamente, proporcionando mensajes de error claros y precisos.

**Conclusión**

Este documento de casos de prueba proporciona una guía para validar el correcto funcionamiento de la Contacto API. Al seguir estos pasos se asegura que la API cumple con los requisitos funcionales y maneja correctamente las operaciones CRUD sobre los contactos.