

**Plan de pruebas**

**Consulta de un inmueble y gestión de usuarios mediante servicios rest**

**V1.0**

**Historias de Usuario**

**HU: TEA-0001 - Consultar inmueble**

Como usuario de la página web <https://sobreplanos-staging.herokuapp.com/>, quiero consultar un inmueble para su precio.

Criterios de aceptación

1. Crear una modal donde el usuario puede elegir el país ya sea México o Colombia.
2. Crear una pantalla donde el usuario pueda ingresar la ubicación del inmueble, el tipo de propiedad y el número de habitaciones.
   1. El campo ubicación debe ser una lista que permita seleccionar la ciudad, zona o barrio.
   2. El campo tipo de propiedad debe ser una y debe contener proyectos de vivienda, apartamentos y casas.
   3. El campo número de habitaciones debe ser una lista donde permita ingresar el número de habitaciones.
3. Crear una pantalla donde el usuario pueda filtrar de acuerdo a los siguientes criterios.
   1. Tipo de propiedad, Nuevo o Usado si el inmueble es apartamento o casa, si el tipo de propiedad es Proyectos de vivienda debe decir En construcción o Terminado, este filtro debe ser un checkbox.
   2. Un filtro que debe ser Precio desde hasta, donde se ingrese un rango con el rango inicial y final para el precio del inmueble.
   3. Un filtro donde se pueda ingresar el número de habitaciones.
   4. Un filtro donde se puede ingresa el número de parqueaderos.
   5. Un filtro donde se pueda ingresar el número de baños.
   6. Un filtro donde se pueda ingresar el área mínima y máxima.
4. Crear un botón que me permita aplicar los filtros y se llame Aplicar precio.
5. En la pantalla de filtros se deben mostrar todas las propiedades con el filtro que el usuario selecciono.
6. Crear una pantalla donde se muestre la información de la propiedad, en esta pantalla debe aparecer lo siguiente:
   1. Precio.
   2. Fotos.
   3. Número de baños.
   4. Número de habitaciones.
   5. Número de parqueaderos.
   6. Descripción del inmueble.

**HU: TEA-0002 – Consultar usuario**

Como usuario de la aplicación https://reqres.in/ quiero consultar un usuario para ver si este esta creado en la base de datos.

Criterios de aceptación

1. El servicio debe permitir ingresar el código del usuario
2. El servicio debe utilizar un método GET.
3. El servicio debe retornar en la respuesta el JSON con la siguiente firma.
4. Si la consulta es exitosa el servicio debe retornar el código de respuesta 200 de lo contrario el código de respuesta 400 si es un bussines exception o 500 si es un error técnico.

{

"data": {

"id":

"email": "",

"first\_name": "",

"last\_name": "",

"avatar": ""

},

"support": {

"url": "",

"text": ""

}

}

**HU: TEA-0002 – Crear usuario**

Como usuario de la aplicación https://reqres.in/ quiero crear un usuario para que este pueda ingresar a la aplicación.

Criterios de aceptación

1. El servicio debe pedir en el request un json con los datos name y job.
2. El servicio debe utilizar un método POST.
3. El servicio debe retornar en la respuesta un json con el id de creación del usuario y el campo createdAt con la fecha de creación del usuario.
4. Si la creación es exitosa el servicio debe retornar el código de respuesta 201 de lo contrario el código de respuesta 400 si es un bussines exception o 500 si es un error técnico.

**HU: TEA-0003 – Eliminar usuario**

Como usuario de la aplicación https://reqres.in/ quiero eliminar un usuario para comprobar que funciona el borrado en la base de datos.

Criterios de aceptación

1. El servicio debe permitir ingresar el código del usuario
2. El servicio debe utilizar un método Delete.
3. Si la eliminación es exitosa el servicio debe retornar el código de respuesta 204 de lo contrario el código de respuesta 400 si es un bussines exception o 500 si es un error técnico.

**HU: TEA-0004 – Actualizar usuario**

Como usuario de la aplicación https://reqres.in/ quiero actualizar un usuario para comprobar el funcionamiento de la actualización en la base de datos.

Criterios de aceptación

1. El servicio debe pedir en el request un json con los datos name y job.
2. El servicio debe utilizar un método PUT.
3. El servicio debe retornar en la respuesta un json con el name, job del usuario que se modificó y el campo updatedAt con la fecha de actualización del usuario.
4. Si la modificación es exitosa el servicio debe retornar el código de respuesta 200 de lo contrario el código de respuesta 400 si es un bussines exception o 500 si es un error técnico.

**HU: TEA-0005 – Actualizar usuario con método patch**

Como usuario de la aplicación https://reqres.in/ quiero actualizar un usuario para comprobar el funcionamiento de la actualización en la base de datos.

Criterios de aceptación

1. El servicio debe pedir en el request un json con los datos name y job.
2. El servicio debe utilizar un método PATCH.
3. El servicio debe retornar en la respuesta un json con el name, job del usuario que se modificó y el campo updatedAt con la fecha de actualización del usuario.
4. Si la modificación es exitosa el servicio debe retornar el código de respuesta 200 de lo contrario el código de respuesta 400 si es un bussines exception o 500 si es un error técnico.

**Alcance**

Dentro del alcance de este plan de pruebas se encuentra los siguientes servicios y funcionalidades:

* Servicio de consultar usuario
* Servicio de crear usuario
* Servicio de eliminar usuario
* Servicio de modificar usuario con método put y path.
* Funcionalidad de consultar inmueble.

Se verificará el funcionamiento de los servicios mencionados anteriormente que los parámetros de entrada, request, response cumplan con los criterios mencionados en la historia de usuario, así como los códigos de respuesta.

Se verificará el funcionamiento de la funcionalidad de consultar propiedad, para la consulta de un inmueble de acuerdo a las características establecidas en los campos de ubicación, tipo de propiedad, número de habitaciones y a los filtros seleccionados, estado, precio, parqueaderos, baños, área, que, al hacer la consulta, se vea reflejado el inmueble con las características seleccionadas.

Se realizarán pruebas en los siguientes navegadores para la aplicación web:

Windows

* Google Chrome
* Mozilla Firefox

Android

* Google Chrome App

Se utilizará el navegador Chrome en modo responsivo para verificar el response de la aplicación.

**Fuera del Alcance**

* No se probarán otras funcionalidades y servicios que no fueron especificados en el alcance.
* No se realizarán pruebas a bases de datos
* No se realizarán pruebas en otros dispositivos o navegadores no especificados dentro del alcance.

**Estrategia**

Se requiere certificar el desarrollo de la funcionalidad consultar inmueble y de 5 servicios descritos anteriormente que el alcance de dicho proyecto cumpla con lo solicitado por el cliente.

* Se realizarán pruebas funcionales manuales y automatizadas.
* Se utilizará la herramienta Insomnia, Google Chrome, Mozilla, serenity bdd con el patrón screenplay para realizar la automatización de los servicios y de dicha funcionalidad.
* Se realizará un smoke test para validar el flujo básico de la aplicación.
* Realizar diseños de negocio desde los diferentes frentes de la transacción e identificar los flujos de los escenarios que se tendrán en cuenta para la automatización con enfoque E2E.
* Identificar los escenarios E2E exitosos y alternos que serán automatizados para la transacción.
* Realizar adecuaciones de ambiente y datos para el funcionamiento de la transacción.
* Realizar codificación de la transacción E2E bajo los lineamientos y estándares proporcionados por sura con el modelo ScreenPlay.

**Escenarios de prueba**

**Actualizar usuario**

Esquema del escenario: Actualizar usuario  
 Cuando se ingresen los datos del usuario a actualizar  
 | **name** | **job** |  
 | **<name>** | **<job>** |  
 Entonces se debe ver el usuario **'<name>'** y el job **'<job>'** con **'<respuesta>'** Ejemplos: Datos de entrada  
 | **name** | **job** | **respuesta** |  
 | **Carlos** | **Abogado** | **OK\_CONSULTA** |

**Borrar usuario**

Esquema del escenario: Borrar usuario  
 Cuando se ingrese el código del usuario a borrar **'<codigo>'** Entonces no se debe visualizar el usuario **'<respuesta>'** Ejemplos: Datos de entrada  
 | **codigo** | **respuesta** |  
 | **2** | **OK\_BORRADO** |

**Buscar usuario**

Esquema del escenario: Buscar usuario  
 Cuando se ingrese el código del usuario a consultar **'<codigo>'** Entonces se debe mostrar el usuario **<codigo>** y la **'<respuesta>'** Ejemplos: Datos de entrada  
 | **codigo** | **respuesta** |  
 | **2** | **OK\_CONSULTA** |

**Crear usuario**

Esquema del escenario: Crear usuario  
 Cuando se ingresen los datos del usuario a crear  
 | **name** | **job** |  
 | **<name>** | **<job>** |  
 Entonces se debe ver el usuario **'<name>'** y el job **'<job>'** con **'<respuesta>'** Ejemplos: Datos de entrada  
 | **name** | **job** | **respuesta** |  
 | **Carlos** | **Abogado** | **OK\_CREACION** |

**Consultar inmueble**

Esquema del escenario: Consultar inmueble  
 Dado que el usuario se encuentre en sobreplanos-staging  
 Cuando ingrese los datos de la propiedad que desee consultar  
 | **pais** | **barrio** | **tipoPropiedad** | **habitaciones** |  
 | **<pais>** | **<barrio>** | **<tipoPropiedad>** | **<habitaciones>** |  
 Cuando filtre por las cualidades de la propiedad  
 | **pais** | **estado** | **parqueaderos** | **baños** | **areaMinima** | **areaMaxima** | **apartamento** |  
 | **<pais>** | **<estado>** | **<parqueaderos>** | **<baños>** | **<areaMinima>** | **<areaMaxima>** | **<apartamento>** |  
 Entonces el usuario vera los datos del inmueble buscado  
 | **habitaciones** | **baños** | **parqueaderos** | **apartamento** |  
 | **<habitaciones>** | **<baños>** | **<parqueaderos>** | **<apartamento>** |  
 Ejemplos: Datos de entrada  
 | **pais** | **barrio** | **tipoPropiedad** | **habitaciones** | **estado** | **parqueaderos** | **baños** | **areaMinima** | **areaMaxima** | **apartamento** |  
 | **Colombia** | **Medellín** | **apartamentos** | **3** | **Nuevo** | **1** | **2** | **30** | **110** | **Riviera De Bulerías** |  
 | **Colombia** | **Bogotá** | **casas** | **3** | **Nuevo** | **3** | **4** | **80** | **320** | **Camino Del Bosque Casas** |  
 | **México** | **Ciudad de México** | **departamentos** | **3** | **Nuevo** | **1** | **2** | **80** | **120** | **Residencial Progreso 88** |  
 | **Colombia** | **Bogotá** | **proyectos-de-vivienda** | **3** | **En construcción** | **1** | **2** | **30** | **120** | **Roble** |

**Prerrequisitos**

* Ambientes de Prueba estables y adecuados para realizar la transacción.
* Requisitos funcionales y no funcionales.
* Historias de usuario.
* Criterios de aceptación
* Endpoint de los servicios

**Herramientas de prueba requeridas**

Para las pruebas funcionales manuales y automatizadas se utilizarán los navegadores Google Chrome y Mozilla.

Para crear los Scripts de Automatización se utilizará IntelliJ IDEA, Screenplay pattern, metodología BDD, serenity BDD, cucumber, gherkin.

**Supuestos y limitaciones**

* El ambiente de pruebas refleja en la mayor forma posible el ambiente de producción, de lo incluido en el alcance de las pruebas.
* El analista de desarrollo suministrará las parametrizaciones requeridas para la realización de las pruebas.
* Se contará con las definiciones de la solución y la estructura definida de cada uno de los archivos nuevos a verificar.
* El analista de certificación cuenta con los permisos necesarios a los archivos, rutas y programas requeridos para la ejecución de la prueba.
* Se contará con las pruebas unitarias y pruebas de aceptación desde desarrollo.
* En el ambiente de pruebas se encuentran todos los objetos que harán parte de la certificación para la prueba E2E.
* En caso de encontrar algún defecto, el equipo de desarrollo se encuentra disponible para dar solución al mismo.
* Se cuenta con los criterios de aceptación definidos por el negocio.

**Infraestructura**

* Ambiente de laboratorio
* Red de internet con velocidad superior a 10mb.
* Computador con sistema operativo Windows 10, memoria RAM de 4gb en adelante.
* Celular Android con versión mayor a 7.0.

**Tipo de prueba**

* Pruebas funcionales automatizadas y manuales.