**Testing Exploratorio**

# Historial de Cambios

| Versión | Fecha | Referente | Cambios |
| --- | --- | --- | --- |
| 1.0.0 | 21/10/2021 | mterraza | Versión inicial |
|  |  |  |  |

# Objetivo

El objetivo del presente documento registra la documentación del testing para se gane conocimiento de la aplicación y con ese conocimiento luego pueden crear los casos de prueba necesarios

# Alcance

El presente documento contiene pruebas aplicables a la utilización de la app web DB

# Supuestos

Debe estar instalada la versión del depurador que se está homologando.

Debe estar actualizado el alcance del depurador que se está homologando.

# Introducción

En la planilla de casos de prueba se indica la funcionalidad de negocio y el comportamiento esperado de la app web en cada caso probado.

Los casos han sido elaborados por el área homologación, pero no se descarta la necesidad de la generación de casos de prueba adicionales por parte del área de desarrollo.

En esta instancia de exploratorio, se van a ejecutar pruebas a nivel general, se espera que de las mismas se obtengan muchos defectos, teniendo en cuenta que la app aún está en desarrollo.

# Escenarios

Pruebas Front End:

Se van a realizar pruebas sobre la navegación de un usuario por la web, donde el mismo puede iniciar sesión, desloguearse, simular la creación de una cuenta nueva, visualizar categorías de alojamientos, ver recomendaciones y navegar desde distintos dispositivos

1. Usuario que ingresa por primera vez a la app web, no tiene una cuenta creada
2. Usuario que ya tiene una cuenta creada e ingresa nuevamente
3. Usuario que navega por el home
4. Un usuario logueado que cierre sesión

Pruebas Back End:

Se van a realizar pruebas sobre los alojamientos, es decir vamos a tener una base de datos creada desde Mysql, sobre ella vamos a insertar nuevos alojamientos, actualizar los que existan si es necesario, eliminar alguno existente, consultar todos o alguno específico por ID. Esto se va a verificar a través de la herramienta de Postman donde vamos a realizar peticiones y al mismo tiempo Test, que van a ser exportados para ser documentados.

1. Registrar un nuevo alojamiento
2. Registrar un alojamiento para editar
3. Registrar un alojamiento para eliminar
4. Actualizar un alojamiento
5. Eliminar un alojamiento
6. Buscar alojamiento por Id
7. Buscar todos los alojamientos

Es necesario realizar también pruebas negativas para poder comprobar que funciona correctamente y que no se guarden datos erróneos, o realicen operaciones erróneas.

# 

# Datos para las pruebas

Para las pruebas es necesario hacer un git clone de la rama correspondiente para poder testearlo en nuestra computadora.

Front End:

Realizar un npm install y luego npm start

Para probar los test cases diseñados se necesita usuario y contraseña válidos para el inicio de sesión

Con la herramienta inspeccionar del navegador contar con tres dimensiones distintas :

* Desktop: 1366
* Ipad: 768
* Mobile: 414

Las pruebas se van a ejecutar sobre estos 3 dispositivos distintos

Back End:

Luego del git clone es necesario tener la Workbench abierto con la base de datos creada, se corre el script que está definido en la rama, luego desde IntelliJ hacer un update de Maven, un clean y package y luego ejecutar la aplicación.

Luego abrir la herramienta Postman y crear una nueva colección

Para realizar las pruebas en Postman es necesario contar con los textos a incluir en el body de las peticiones Body y los end points donde se van a realizar las peticiones.

Para los test debemos correr scripts para ver si el status code es 200, o 400 y 500 en el caso de que la misma no deba ejecutarse por ser errónea. Al mismo tiempo vamos a validar que incluye el body del box response

# **Caso de Prueba**

# Información complementaria

Documentos relacionados:

| **Nombre del Documento** | **URL** |
| --- | --- |
| Mindmap pruebas | https://miro.com/welcomeonboard/d0NESFZSeWpUWkpLRDhkU1JUeGNBSGJ3dDNTczJLTzhvczZlcnk5R05VRXdEY3FpWHJ0aFB0MUZoUGhlUEpHRXwzMDc0NDU3MzQ3NjM1ODExMDM2?invite\_link\_id=52417568345 |
| Casos de prueba formales | https://drive.google.com/drive/folders/1SF8EC--OmTVsqam1LNORqiypWBsL9-Lq?usp=sharing |

## **Planilla de casos de prueba**

**Prueba Front End:**

· El usuario no se encuentra logueado

| **#** | **Nombre** | **Descripción/Pasos** | **Resultado esperado** | **Resultado obtenido** |
| --- | --- | --- | --- | --- |
| 1 | Visualización página web | Se ingresa a a la app web | Se visualiza la web de acuerdo al diseño de Figma para Sprint I | Se visualizan algunos elementos que se encuentran desbordados sin diseño responsive |
| 2 | Bloque registro | Se ingresa al boton Crear cuenta luego de ingresar a la web | Se visualiza diseño responsive de acuerdo a el prototipo Figma.  Permite ingresar valores en los campos | Permite ingresar valores invalidos en los campos de Nombres y apellido, se pueden dejar campos vaciós. No redirige al crear |
| 3 | Bloque login | Se ingresa al boton Iniciar sesión luego de ingresar a la web | Se visualiza diseño responsive de acuerdo a el prototipo Figma  Permite ingresar valores en los campos | No redirige, se pueden dejar campos vacios |
| 4 | Mobile | Se ingresa con la dimensión mobile | Se visualiza diseño responsive de acuerdo a el prototipo Figma | No se visualiza diseño responsive |
| 5 | Ipad | Se ingresa con la dimensión Ipad | Se visualiza diseño responsive de acuerdo a el prototipo Figma | No se visualiza diseño responsive |
| 6 | Bloque buscador | Se visualiza el bloque buscador y se seleccionan los inputs de ciudad y calendario. | Se despliega lista de ciudades y en las fechas muestra el calendario que permite seleccionar fechas | No se despliega listado y no permite en fechas poder ver el calendario ni seleccionar fechas |
| 7 | Bloque Listado | Se visualiza el apartado de recomendaciones | Se visualizan dos columnas en desktop y ipad, y una en mobile con las categorías | Se visualizan los elementos desbordados de las cards |
| 8 | Bloque categorias | Se visualizan el bloque de categorías de alojamiento | Se visualizan 4 categorías de alojamientos | OK |

**Defectos asociados:**

* Puede ingresar números y caracteres especiales en el campo Nombre y Apellido cuando se quiere registrar
* No se visualiza el contenido de forma correcta en los distintos dispositivos
* No se ven la credenciales luego de iniciada la sesión
* No se ve el menú hamburguesa en la versión mobile
* No se ve el listado de ciudades
* No se ve el calendario de acuerdo a lo establecido en Figma
* Se desbordan los elementos en las cards de Listado

**Prueba Back End:**

| **#** | **Nombre** | **Descripción/Pasos** | **Resultado esperado** | **Resultado obtenido** |
| --- | --- | --- | --- | --- |
| 1 | Nuevo alojamiento | Se ingresa nuevo alojamiento con peticion put en postman | **Se registra el alojamiento con ID #1** | **Se registro con id#2** |
| 2 | Buscar todos | Se buscan todos los alojamientos con peticion GET | **Se muestra el alojamiento creado anteriormente** | **ok** |
| 3 | Eliminar alojamiento | Se eliminan calendarios con peticion DELETE | **Se elimina el alojamiento con id#2** | **ok** |
| 4 | Nuevo alojamiento | Se ingresa nuevo alojamiento con peticion put en postman | **Se registra el alojamiento con ID #3** | **Se registro con id#3** |
| 5 | Actualizar alojamiento | Se actualizan calendarios con peticion PUT | **Se muestra el alojamiento #3 actualizado** | **ok** |
| 6 | Buscar por Id | Se busca alojamiento especifico con peticion GET y el id requerido #3 | **Se muestra el alojamiento #3** | **ok** |

**Defectos asociados:**

* El primer alojamiento lo crea con ID #1