

Testing Exploratorio FrontEnd y crud de postman

Objetivo

El objetivo del documento es registrar 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 de desarrollo para homologación

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.

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

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: 1366x768
- Tablet: 768x1024
- Mobile: 414x736

Las pruebas se van a ejecutar sobre estos 3 dispositivos distintos

Caso de Prueba

Planilla de casos de prueba

Prueba Front End:

- El usuario no se encuentra logueado

#	Nombre	Descripción	Resultado	Resultado
---	--------	-------------	-----------	-----------

			esperado	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 botón Crear cuenta luego de ingresar a la web	Se visualiza diseño responsive de acuerdo con el prototipo Figma. Permite ingresar valores en los campos	ok
3	Bloque login	Se ingresa al botón Iniciar sesión luego de ingresar a la web	Se visualiza diseño responsive de acuerdo a el prototipo Figma Permite ingresar valores en los campos	Redirigie, y se pueden completar los campos, no permite dejar campos vacios
4	Mobile	Se ingresa con la dimensión mobile	Se visualiza diseño responsive de acuerdo a el prototipo Figma	ok
5	Ipad	Se ingresa con la dimensión Ipad	Se visualiza diseño responsive de acuerdo a el prototipo Figma	ok
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	ok
7	Bloque Listado	Se visualiza el apartado de recomendaciones	Se visualizan dos columnas en desktop y ipad, y una en	ok

			mobile con las categorías	
8	Bloque categorías	Se visualiza el bloque de categorías de alojamiento	Se visualizan 4 categorías de alojamientos	OK

Defectos asociados:

- Ningún defecto encontrado

Pruebas de postman

```
{
  "info": {
    "_postman_id": "233567f9-7b9a-4000-a2b9-a15fd1eb1aae",
    "name": "api_booking",
    "schema":
      "https://schema.getpostman.com/json/collection/v2.1.0/collection.json",
    "_exporter_id": "20325200"
  },
  "item": [
    {
      "name": "categories",
      "item": [
        {

```

```

"name": "findById",

"request": {

    "method": "GET",

    "header": []

},

"response": []

},

{

"name": "createCategory",

"request": {

    "method": "POST",

    "header": [],

    "body": {

        "mode": "raw",

        "raw": "{\r\n  \"title\": \"title two\", \r\n  \"description\": \"description two\", \r\n  \"url\": \"url two\"\r\n}",

        "options": {

            "raw": {

                "language": "json"

            }

        }

    },

    "url": {

```

```

    "raw":
    "http://localhost:8080/api/categories/create",

    "protocol": "http",

    "host": [

        "localhost"

    ],

    "port": "8080",

    "path": [

        "api",

        "categories",

        "create"

    ]

    },

    "response": []

},

{

    "name": "findAll",

    "request": {

        "method": "GET",

        "header": []

    },

    "response": []
}
```

```

    },
    {
        "name": "updateCategory",
        "request": {
            "method": "PUT",
            "header": [],
            "body": {
                "mode": "raw",
                "raw": "{\r\n  \"id\": 1,\r\n  \"title\": \"title one up\", \r\n  \"description\": \"description one\", \r\n  \"url\": \"url one\"\r\n}",
                "options": {
                    "raw": {
                        "language": "json"
                    }
                }
            },
            "url": {
                "raw":
"http://localhost:8080/api/categories/update",
                "protocol": "http",
                "host": [
                    "localhost"
                ],

```

```
        "port": "8080",

        "path": [

            "api",

            "categories",

            "update"

        ]

    }

},

"response": []

},

{

    "name": "deleteCategory",

    "request": {

        "method": "DELETE",

        "header": [],

        "url": {

            "raw":

"http://localhost:8080/api/categories/delete/3",

            "protocol": "http",

            "host": [

                "localhost"

            ],

            "port": "8080",
```



```
        "path": [
            "api",
            "categories",
            "delete",
            "3"
        ]
    },
    "response": []
}

]
```