**Reto No. 2: Autenticación - BD Usuarios (CRUD) - Inventario de Productos (CRUD)**  
  
Para el segundo reto el equipo construirá tanto en Front-end como en Back-end las funcionalidades de Autenticación - BD Usuarios (CRUD) - Inventario de Productos (CRUD), así como la subida al sistema del logo de la empresa, creado por ellos mismos.  
  
Lo anterior se llevará a cabo en una BD NOSQL con MongoDB y se ajustarán los servicios web utilizando las dependencias adecuadas.  
  
**CREACIÓN DEL FRONTEND.**  
  
En lo que al Front-end se refiere, el equipo de trabajo debe crear lo siguiente:  
  
1. El logo de la empresa (Este se utilizará en el header del Sitio).  
  
2. Un formulario para que el usuario pueda iniciar sesión.  
  
3. Un formulario para que el Administrador pueda ingresar los datos de los Usuarios (Coordinadores de Zona y Asesores Comerciales) a la BD y una tabla de Administrador de Usuarios (CRUD).  
  
4. Un formulario para que el administrador pueda ingresar los datos de los productos a la BD y una tabla de Inventario de Productos (CRUD).  
  
Así mismo, harán uso del lenguaje JavaScript para validaciones del lado del cliente como son controlar la obligatoriedad de campos, e-mail no válido, y contraseña/confirmación no coincidente, al igual que para los formularios creados para los usuarios del sistema y los productos ofrecidos por la empresa.  
  
De otro lado, mediante el uso de HTML y CSS aplicarán estilo a los formularios, e incluirán características responsive web design gracias a la utilización del framework CSS Bootstrap.  
  
Para finalizar, deben integrar el Front-end con el Back-end mediante la invocación de los servicios web definidos a través de peticiones ajax con JQuery.  
  
**a. Sobre el proceso de autenticación...**  
  
Al iniciar el Front-end desde el navegador web o browser, el sistema mostrará una página que incluirá un formulario en donde el Asesor Comercial o Coordinador de Zona podrán ingresar su e-mail/clave para autenticarse en el sistema, seleccionarán el botón Ingresar y la aplicación validará que exista una cuenta asociada con la combinación e-mail/clave, donde presentará el mensaje de bienvenida "Bienvenido " en caso de superar la validación, o el mensaje "No existe un usuario" en caso contrario.  
  
Si el Asesor Comercial o Coordinador de Zona no tienen una cuenta para autenticarse en el sistema, el Administrador del sistema deberá crearlos por Back-end.  
  
Validaciones:

* Obligatoriedad de campos, e-mail no válido, y clave/confirmaciones no coincidentes.
* Cuenta asociada a la combinación e-mail/clave no existe.
* Obligatoriedad de campos, e-mail no válido.
* Cuenta asociada a la combinación e-mail/clave ya existe.
* E-mail ya existe.
* Validación datos modificados.
* Borrado de la BD de usuarios creados con anterioridad.
* Obligatoriedad de campos.
* Validación datos modificados.
* Borrado de la BD de usuarios creados con anterioridad.
* Email existe: true.
* Email no existe: false.

**b. Sobre el proceso de Creación de usuarios (CRUD)...**  
  
Desde el navegador web o browser, el sistema mostrará una página que incluirá un formulario en donde el Administrador del sistema podrá Agregar los datos de los Usuarios (Coordinadores de Zona y Asesores Comerciales) a la BD, es decir, registrar su número de Identificación, Nombres, Dirección, Celular, E-mail, Clave y Zona asignada de la ciudad. En el momento de hacer clic en el botón Guardar para validar la creación de la cuenta, la aplicación validará que no exista el e-mail, que no exista la combinación email/contraseña, presentará el mensaje "Cuenta creada de forma correcta" en caso de superar la validación, o el mensaje "No fue posible crear la cuenta" en caso contrario.  
  
Validaciones:  
  
**c. Sobre el proceso de Administración de usuarios (CRUD)...**  
  
Desde el navegador web o browser, el sistema mostrará una página que incluirá una tabla en donde el Administrador del sistema podrá Editar los datos de los Usuarios (Coordinadores de Zona y Asesores Comerciales) registrados previamente en el sistema. En el momento de hacer clic en el botón Guardar el sistema validará los datos modificados, presentará el mensaje "Usuario editado" en caso de superar la validación, o el mensaje "Usuario no editado" en caso contrario.  
  
Si por el contrario el Administrador quiere borrar el registro del Asesor Comercial o Coordinador de Zona creado previamente, el Administrador podrá seleccionar la opción Eliminar, el sistema presentará el mensaje "Usuario borrado de la BD".  
  
Validaciones:  
  
**d. Sobre el proceso de Creación de productos (CRUD)...**  
  
Desde el navegador web o browser, el sistema mostrará una página que incluirá un formulario en donde el Administrador del sistema podrá Agregar los datos de los productos calzado a la BD, es decir, registrar su referencia, género, categoría, número, materiales, descripción, disponibilidad (SI/NO), precio, cantidad en stock y fotografía (URL). En el momento de hacer clic en el botón Guardar el sistema creará la ficha de producto.  
  
Validaciones:  
  
**e. Sobre el proceso de Inventario de Productos (CRUD)...**  
  
Desde el navegador web o browser, el sistema mostrará una página que incluirá una tabla en donde el Administrador del sistema podrá Editar productos calzado registrados previamente en el sistema. En el momento de hacer clic en el botón Guardar el sistema validará los datos modificados, presentará el mensaje "Producto editado" en caso de superar la validación, o el mensaje " Producto no editado" en caso contrario.  
  
Si por el contrario el Administrador quiere borrar el registro de un producto calzado creado previamente, el Administrador podrá seleccionar la opción Eliminar, el sistema presentará el mensaje "Usuario borrado de la BD".  
  
Validaciones:  
  
**CREACIÓN DEL BACKEND.**  
  
El equipo de trabajo continuará haciendo uso de Java, Spring Boot e iniciarán con el motor de la base de datos NO relacional MongoDB en el Back-end.  
  
1. Administración de usuarios (CRUD).  
  
Modelarán la colección para administrar la información de los usuarios del sistema, crearán servicios REST para gestionar operaciones de listar usuarios, crear usuarios, autenticar usuarios y validar si un e-mail ya se encuentra registrado en la base de datos.  
  
Ejemplo 1:  
  
@Document(collection = "usuarios").  
  
@Data.  
  
@NoArgsConstructor.  
  
@AllArgsConstructor.  
  
public class User {.  
  
@Id.  
  
private Integer id;.  
  
private String identification;.  
  
private String name;.  
  
private Date birthtDay;.  
  
private String monthBirthtDay;.  
  
private String address;.  
  
private String cellPhone;.  
  
private String email;.  
  
private String password;.  
  
private String zone;.  
  
private String type;.  
  
}.  
  
2. Inventario de Productos (CRUD).  
  
Modelarán la colección para administrar la información de los productos del sistema, crearán servicios REST para gestionar operaciones de listar productos, crear productos, eliminar productos.  
  
Ejemplo:  
  
@Document(collection = "footwears").  
  
@Data.  
  
@NoArgsConstructor.  
  
@AllArgsConstructor.  
  
public class Footwear {.  
  
@Id.  
  
private String reference;.  
  
private String brand;.  
  
private String category;.  
  
private String material;.  
  
private String gender;.  
  
private String size;.  
  
private String description;.  
  
private boolean availability = true;.  
  
private double price;.  
  
private int quantity;.  
  
private String photography;.  
  
}.  
  
**SERVICIOS.**  
  
1. Listar usuarios: Este servicio procesará una petición de tipo GET, retornará un listado con todos los usuarios registrados (endpoint: /api/user/all).  
  
Ejemplo: http://BASE\_URL/api/user/all.  
  
2. Registrar usuarios: Este servicio procesará una petición de tipo POST, y registrará un usuario en la base de datos, este puede ser de tipo: COORD para el Coordinador de Zona, ASE para el Asesor Comercial y ADM para el Administrador y se guardará internamente en el sistema o retornará la información enviada en la petición en caso de fallo.  
  
Ejemplo:  
  
http://BASE\_URL/api/user/new.  
  
Nota: El tipo de usuario será COORD para el Coordinador de Zona, ASE para el Asesor Comercial y ADM para el Administrador y se guardará internamente en el sistema.  
  
3. Validar si existe email: Este servicio procesará una petición de tipo GET, retornará true si el email ya existe o false en caso contrario. para esto, se agregará a la URL una barra "/" y seguido a esto, el correo electrónico (endpoint: /api/user/"correo electrónico").  
  
Ejemplo: http://BASE\_URL/api/user/emailexist/{email}.  
  
http://BASE\_URL/api/user/emailexist/cangalotty@gmail.edu.com.  
  
4. Autenticar usuario: Este servicio procesará una petición de tipo GET, validará si existe un usuario que corresponda con la combinación email/clave.  
  
Si existe un usuario que corresponda con la combinación email/clave ingresada, retornará la información del usuario en base de datos, en caso contrario retornará un usuario con el email y contraseña ingresados y la cadena de texto "NO DEFINIDO" en el nombre: (endpoint: /api/user/).  
  
Ejemplo: http://BASE\_URL/api/user/cangalotty@gmail.edu.com/Demo123.  
  
Usuario existe:  
  
{.  
  
"id":1,.  
  
"identification":"52369563",.  
  
"name":"CARLO ANGELOTY",.  
  
"address":"CR 19 A 63 C 37",.  
  
"cellPhone":"3256323212",.  
  
"email":"cangalotty@gmail.edu.com",.  
  
"password":"Demo123.",.  
  
"zone":"ZONA 1",.  
  
"type":"ADM".  
  
}.  
  
Usuario no existe:  
  
{.  
  
"id":null,.  
  
"identification":null,.  
  
"name":null,.  
  
"address":null,.  
  
"cellPhone":null,.  
  
"email":null,.  
  
"password":null,.  
  
"zone":null,.  
  
"type":null.  
  
}.  
  
5. Editar usuarios: Este servicio procesará una petición de tipo PUT, y registrará la nueva información editada del usuario en la base de datos, o retornará la información enviada en la petición en caso de fallo.  
  
Ejemplo:  
  
http://BASE\_URL/api/user/update.  
  
6. Eliminar usuarios: Este servicio procesará una petición de tipo DELETE, y borrará la información ingresada previamente del usuario en la base de datos, o retornará la información enviada en la petición en caso de fallo.  
  
Ejemplo:  
  
http://BASE\_URL/api/user/{id}.  
  
7. Listar productos: Este servicio procesará una petición de tipo GET, retornará un listado con todos los usuarios registrados (endpoint: /api/accessory/all).  
  
Ejemplo: http://BASE\_URL/api/accessory/all.  
  
8. Registrar productos: Este servicio procesará una petición de tipo POST, y registrará un producto en la base de datos, o retornará la información enviada en la petición en caso de fallo.  
  
Ejemplo:  
  
http://BASE\_URL/api/accessory/new.  
  
9. Editar productos: Este servicio procesará una petición de tipo PUT, y registrará la nueva información editada del producto en la base de datos, o retornará la información enviada en la petición en caso de fallo.  
  
Ejemplo:  
  
http://BASE\_URL/api/accessory/update.  
  
10. Eliminar productos: Este servicio procesará una petición de tipo DELETE, y borrará la información ingresada previamente del producto en la base de datos, o retornará la información enviada en la petición en caso de fallo.  
  
Ejemplo:  
  
http://BASE\_URL/api/accessory/{id}.  
  
  
**Historias de usuario:**

|  |  |
| --- | --- |
| **Hisotoria de usuario Nro.** | 1 |
| **Título:** | Listar usuarios |
| **Descripción** | |  |  | | --- | --- | | **COMO:** | Usuario | | **QUIERO:** | Ingresar a la lista de usuarios del Sistema | | **PARA:** | Ver los usuarios que hay en el sistema | |
| **Criterios de aceptación** | * La lista de usuarios debe presentar los campos: identificación, nombres, email, tipo de usuario (perfil), zona. |

|  |  |
| --- | --- |
| **Hisotoria de usuario Nro.** | 2 |
| **Título:** | Registrar usuarios |
| **Descripción** | |  |  | | --- | --- | | **COMO:** | Usuario | | **QUIERO:** | Ingresar los valores de cuenta de usuario: Identificación, Nombres, Dirección, Celular, E-mail, Clave y Zona. | | **PARA:** | Poder registrar un usuario en el sistema. | |
| **Criterios de aceptación** | * El id se genera automáticamente y no se captura o envía. * La longitud del nombre del usuario es de obligatorio diligenciamiento y su longitud no podrá superar los 80 caracteres. * El email del usuario es obligatorio, la longitud no podrá superar los 50 caracteres. * La clave del usuario es obligatoria, la longitud no podrá superar los 50 caracteres. |

|  |  |
| --- | --- |
| **Hisotoria de usuario Nro.** | 3 |
| **Título:** | Validar email |
| **Descripción** | |  |  | | --- | --- | | **COMO:** | Usuario | | **QUIERO:** | Validar si un correo electrónico ya existe en el sistema. | | **PARA:** | Poder registrar un usuario en el sistema. | |
| **Criterios de aceptación** | * El valor del correo es obligatorio, y debe cumplir con las condiciones de un correo electrónico. * Los valores de retorno serán true si el usuario existe, o false en caso contrario. |

|  |  |
| --- | --- |
| **Hisotoria de usuario Nro.** | 4 |
| **Título:** | Autenticar usuario |
| **Descripción** | |  |  | | --- | --- | | **COMO:** | Usuario | | **QUIERO:** | Validar si un correo electrónico ya existe en el sistema. | | **PARA:** | Poder registrar un usuario en el sistema. | |
| **Criterios de aceptación** | * Si existe un usuario que corresponda con la combinación email/clave ingresada, retornará la información del usuario en base de datos, en caso contrario en caso contrario retornará un usuario con el email y contraseña ingresados y la cadena de texto "NO DEFINIDO" en el nombre. |

|  |  |
| --- | --- |
| **Hisotoria de usuario Nro.** | 5 |
| **Título:** | Editar usuarios |
| **Descripción** | |  |  | | --- | --- | | **COMO:** | Usuario | | **QUIERO:** | Editar los valores de cuenta de usuario: Identificación, Nombres, Dirección, Celular, E-mail, Clave y Zona. | | **PARA:** | Poder editar un usuario en el sistema. | |
| **Criterios de aceptación** | * Validación datos modificados. * Validar nuevamente el E-mail, si este fue modificado por el Administrador. * Si el E-mail no fue modificado el sistema no debe realizar la validación de dicho campo. |

|  |  |
| --- | --- |
| **Hisotoria de usuario Nro.** | 6 |
| **Título:** | Eliminar usuarios |
| **Descripción** | |  |  | | --- | --- | | **COMO:** | Usuario | | **QUIERO:** | Eliminar los usuarios creados previamente en el sistema. | | **PARA:** | Limpiar la base de datos. | |
| **Criterios de aceptación** | Validar el borrado de la BD de usuarios creados con anterioridad. |

|  |  |
| --- | --- |
| **Hisotoria de usuario Nro.** | 7 |
| **Título:** | Listar productos |
| **Descripción** | |  |  | | --- | --- | | **COMO:** | Usuario | | **QUIERO:** | Ingresar a la lista de productos del Sistema | | **PARA:** | Ver los productos que hay en el sistema | |
| **Criterios de aceptación** | * La lista de productos calzado a la BD, es decir, registrar su referencia, género, categoría, número, materiales, descripción, disponibilidad (SI/NO), precio, cantidad en stock y fotografía (URL). |

|  |  |
| --- | --- |
| **Hisotoria de usuario Nro.** | 8 |
| **Título:** | Registrar productos |
| **Descripción** | |  |  | | --- | --- | | **COMO:** | Usuario | | **QUIERO:** | Ingresar los valores de la ficha de producto: referencia, género, categoría, número, materiales, descripción, disponibilidad (SI/NO), precio, cantidad en stock y fotografía (URL). | | **PARA:** | Poder registrar un producto en el sistema. | |
| **Criterios de aceptación** | * El id se genera automáticamente y no se captura o envía. * Todos los campos son obligatorios. * La categoría debe estar asociada al tipo productos (calzado). * La descripción no podrá superar los 80 caracteres. * El precio debe ser numérico. * La disponibilidad deberá mostrar las opciones (SI/NO). * La cantidad de stock debe ser numérico. * La fotografía deberá subirse al sistema por medio de una URL. |

|  |  |
| --- | --- |
| **Hisotoria de usuario Nro.** | 9 |
| **Título:** | Editar productos |
| **Descripción** | |  |  | | --- | --- | | **COMO:** | Usuario | | **QUIERO:** | Editar los valores de la ficha de producto: referencia, género, categoría, número, materiales, descripción, disponibilidad (SI/NO), precio, cantidad en stock y fotografía (URL). | | **PARA:** | Poder editar un producto en el sistema. | |
| **Criterios de aceptación** | * Validación datos modificados. |

|  |  |
| --- | --- |
| **Hisotoria de usuario Nro.** | 10 |
| **Título:** | Eliminar productos |
| **Descripción** | |  |  | | --- | --- | | **COMO:** | Usuario | | **QUIERO:** | Eliminar los productos creados previamente en el sistema. | | **PARA:** | Limpiar la base de datos. | |
| **Criterios de aceptación** | * Validar el borrado de la BD los productos creados con anterioridad. |

Dificultad

Instrucciones

**Instrucciones para la calificación automática**  
  
En este caso, no se ejecutará el código directamente sino que se evaluarán los resultados obtenidos al realizar peticiones HTTP al API construida y que actualmente debe estar desplegada en Oracle Cloud. Las peticiones que se realizarán serán de tipo GET y POST Para realizar las pruebas, las tablas de la base datos deben estar vacías y el valor

|  |  |
| --- | --- |
| **Entrada** | Cada grupo debe proveer URL (o la ip) donde se encuentra desplegada el API. Esta entrada se denominará URL Base. |
| **Salida** | Corresponde a los mensajes JSON que retorna cada petición. MasterTech hará la comparación de forma automática. |

A continuación se presenta un ejemplo del proceso de evaluación. La prueba estará formada por 6 casos de prueba y tendrá los siguientes parámetros.  
**Tipo:** describe el tipo de petición HTTP  
**​Endpoint:** BASE\_URL/enlace de acceso al servicio a evaluar  
**Body:** JSON de entrada para la petición HTTP  
**Salida:** JSON de salida de la petición HTTP  
En el primer paso, MasterTech hará una petición de tipo GET para conocer el estado de la base de datos. Luego, se hará una petición de tipo POST en la cual se agregará un registro a la tabla User. Finalmente, la prueba terminará con una serie de peticiones GET para validar si el registro ingreso de forma exitosa, y otras dos más para validar existencia de correo y autenticación de usuario en el sistema. Todo este proceso se realiza de manera automática por MasterTech.  
  
**Recomendaciones generales**  
Antes de enviar la solución del reto, por favor tenga en cuenta los siguientes aspectos:  
Verifique que la aplicación esté desplegada correctamente y sea accesible a través de la URL.  
Verifique que todos los endpoint funcionen. Recuerde que puede probarlos a través de PostMan.  
Las tablas de la base de datos deben estar vacías y recién creadas para que el identificador generado automáticamente en cada tabla esté en 0.  
Debe proveer a MasterTech para las pruebas es la URL de su API (Dirección IP). Esa URL será entendida como BASE\_URL. Tener presente incluir http:// antes de BASE\_URL.  
Debe copiar el contenido del archivo UserController.java en el editor de código para validar las buenas prácticas de programación.  
Debe subir un archivo .zip con la carpeta src de su proyecto. Se debe asegurar que el contenido sea inferior a 5mb. No debe incluir librerías, ni la carpeta target (en caso que las tenga o el .jar del proyecto) para disminuir el tamaño del archivo.  
Se debe utilizar una URL distinta por estudiante. Se validará que las urls utilizadas no sean utilizadas por estudiantes diferentes para evitar acciones fraudulentas.  
Todos los integrantes del equipo, harán integración y despliegue para que sus pruebas sean exitosas.  
En el momento de ejecutar la evaluación automática, la aplicación debe estar en ejecución.

|  |
| --- |
| **Casos de prueba** |
| |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Caso 1** | |  |  | | --- | --- | | **Tipo** | POST | | **Endpoint** | BASE\_URL/api/user/new | | **Body** | { "id": 1, "identification": "123456", "name": "alan brito", "address": "CR 34-45", "cellPhone": "311222222", "email": "alanbrito@gmail.com", "password": "Demo123.", "zone": "ZONA 1", "type": "COORD"} | | **Salida** |  | | | **Caso 2** | |  |  | | --- | --- | | **Tipo** | GET | | **Endpoint** | BASE\_URL/api/user/all | | **Body** |  | | **Salida** | [{"id":1,"identification":"123456","name":"alan brito","address":"CR 34-45","cellPhone":"311222222","email":"alanbrito@gmail.com","password":"Demo123.","zone":"ZONA 1","type":"COORD"}] | | | **Caso 3** | |  |  | | --- | --- | | **Tipo** | GET | | **Endpoint** | BASE\_URL/api/user/emailexist/alanbrito@gmail.com | | **Body** |  | | **Salida** | true | | | **Caso 4** | |  |  | | --- | --- | | **Tipo** | GET | | **Endpoint** | BASE\_URL/api/user/emailexist/alanbrito@hotmail.com | | **Body** |  | | **Salida** | false | | | **Caso 5** | |  |  | | --- | --- | | **Tipo** | GET | | **Endpoint** | BASE\_URL/api/user/alanbrito@hotmail.com/Demo123. | | **Body** |  | | **Salida** | {"id":1,"identification":"123456","name":"alan brito","address":"CR 34-45","cellPhone":"311222222","email":"alanbrito@gmail.com","password":"Demo123.","zone":"ZONA 1","type":"COORD"} | | | **Caso 6** | |  |  | | --- | --- | | **Tipo** | GET | | **Endpoint** | BASE\_URL/api/user/alanbrito@gmail.com/Demo12 | | **Body** |  | | **Salida** | {"id":null,"identification":null,"name":null,"address":null,"cellPhone":null,"email":null,"password":null,"zone":null,"type":null} | | | **Caso 7** | |  |  | | --- | --- | | **Tipo** | PUT | | **Endpoint** | BASE\_URL/api/user/update | | **Body** | { "id": 1, "identification": "1111112", "name": "Alan Brito", "address": "CR 19-45C 22", "cellPhone": "31111111", "email": "alanbrito@hotmail.com", "password": "alanb", "zone": "ZONA 2", "type": "COORD"} | | **Salida** |  | | | **Caso 8** | |  |  | | --- | --- | | **Tipo** | GET | | **Endpoint** | BASE\_URL/api/user/all | | **Body** |  | | **Salida** | [{"id":1,"identification":"1111112","name":"Alan Brito","address":"CR 19-45C 22","cellPhone":"31111111","email":"alanbrito@hotmail.com","password":"alanb","zone":"ZONA 2","type":"COORD"}] | | | **Caso 9** | |  |  | | --- | --- | | **Tipo** | DELETE | | **Endpoint** | BASE\_URL/api/user/1 | | **Body** |  | | **Salida** |  | | | **Caso 10** | |  |  | | --- | --- | | **Tipo** | GET | | **Endpoint** | BASE\_URL/api/user/all | | **Body** |  | | **Salida** | [] | | | **Caso 11** | |  |  | | --- | --- | | **Tipo** | POST | | **Endpoint** | BASE\_URL/api/accessory/new | | **Body** | {"reference": "ABX-202","brand": "ITLO","category": "CASUAL","material": "LEATHER","gender": "MAN","size": "40","description": "Zapatos italianos de alta calidad","availability": true,"price": 450000,"quantity": 23,"photography": "domo.jpg"} | | **Salida** |  | | | **Caso 12** | |  |  | | --- | --- | | **Tipo** | GET | | **Endpoint** | BASE\_URL/api/accessory/all | | **Body** |  | | **Salida** | [{"reference":"ABX-202","brand":"ITLO","category":"CASUAL","material":"LEATHER","gender":"MAN","size":"40","description":"Zapatos italianos de alta calidad","availability":true,"price":450000.0,"quantity":23,"photography":"domo.jpg"}] | | | **Caso 13** | |  |  | | --- | --- | | **Tipo** | PUT | | **Endpoint** | BASE\_URL/api/accessory/update | | **Body** | {"reference": "ABX-202","brand": "ITLO","category": "FORMAL","material": "LEATHER 100%","gender": "UNISEX","size": "39 en adelante","description": "100% importados desde la vieja italia. Zapatos italianos de alta calidad","availability": true,"price": 500000,"quantity": 50,"photography": "domos.jpg"} | | **Salida** |  | | | **Caso 14** | |  |  | | --- | --- | | **Tipo** | GET | | **Endpoint** | BASE\_URL/api/accessory/all | | **Body** |  | | **Salida** | [{"reference":"ABX-202","brand":"ITLO","category":"FORMAL","material":"LEATHER 100%","gender":"UNISEX","size":"39 en adelante","description":"100% importados desde la vieja italia. Zapatos italianos de alta calidad","availability":true,"price":500000.0,"quantity":50,"photography":"domos.jpg"}] | | | **Caso 15** | |  |  | | --- | --- | | **Tipo** | DELETE | | **Endpoint** | BASE\_URL/api/accessory/ABX-202 | | **Body** |  | | **Salida** |  | | | **Caso 16** | |  |  | | --- | --- | | **Tipo** | GET | | **Endpoint** | BASE\_URL/api/accessory/all | | **Body** |  | | **Salida** | [] | | |