

**Benemerita Universidad Autonoma de
Puebla**

**Facultad de Ciencias de la
Computación**

Nombre: Provisión de una API REST

Materia: Servicios Web

Profesor: Juan Carlos Conde Ramirez

Integrantes:

María Azucena Flores Sánchez

José Eduardo Guevara Molina

1. Descripción del proyecto

Se implementó un API REST con el micro-framework Slim, donde se expone la lógica de negocio de las siguientes operaciones:

- 1.1 Obtener lista de Productos de determinada categoría.
- 1.2 Obtener lista de Detalles de un producto en particular usando su clave (isbn).
- 1.3 Insertar un nuevo Producto junto con sus Detalles
- 1.4 Actualizar los Detalles de un Producto
- 1.5 Eliminar Producto junto con sus Detalles

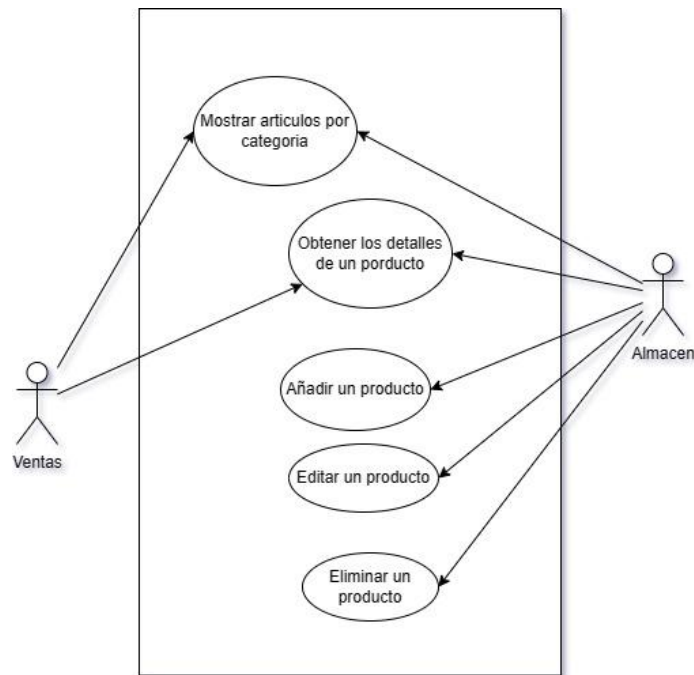
Estas operaciones usan una base de datos almacenada en Firebase, la cual tiene 4 tablas (productos, detalles, respuestas y usuarios.)

 <https://pr06-3e340-default-rtdb.firebaseio.com>

`https://pr06-3e340-default-rtdb.firebaseio.com/`



2. Diagrama UML



Especificación del caso de uso “Mostrar artículos por categoría”

Caso de uso	Mostrar artículos por categoría
Actores	Ventas, almacen
Tipo	Primario
Precondición	Los campos de contraseña y usuario deben estar completos, así como el apartado de categoría, dichos campos deben tener datos válidos al momento de ejecutar este caso de uso
Postcondición	Se mostrarán los datos acorde a los especificado en el apartado categoría, dichos datos son de solo lectura
Escenario principal	Se presiona el boton de busqueda y se realiza la búsqueda del apartado correspondiente, una vez finalizado, se llena la tabla de resultado de búsqueda
Flujo alternativo	Si los datos introducidos son incorrectos en cualquiera de sus apartados, se genera el error correspondiente, que va desde avisos en interfaz gráfica hasta respuestas del servidor con estado de error manejados por messagebox

Especificación del caso de uso Obtener los detalles de un producto

Caso de uso	Obtener los detalles de un producto
Actores	Ventas, almacen

Tipo	Primario
Precondición	Los campos de contraseña y usuario deben estar completos, así como el apartado de ISBN o clave, dichos campos deben tener datos válidos al momento de ejecutar este caso de uso
Postcondición	Se mostrarán los datos acorde a los especificado en el apartado ISBN, dichos datos son de solo lectura
Escenario principal	Se presiona el boton de busqueda y se realiza la búsqueda del apartado correspondiente, una vez finalizado, se llena la tabla de resultado de búsqueda
Flujo alterno	Si los datos introducidos son incorrectos en cualquiera de sus apartados, se genera el error correspondiente, que va desde avisos en interfaz gráfica hasta respuestas del servidor con estado de error manejados por messagebox

Especificación del caso de uso “Añadir un producto”

Caso de uso	Añadir un producto
Actores	almacén
Tipo	Primario
Precondición	Los campos de contraseña y usuario no deben estar vacíos, así como los datos del nuevo producto, dichos campos deben tener datos válidos al momento de ejecutar este caso de uso, por ejemplo ciertos campos solo aceptan números enteros
Postcondición	Se mostrará un MessageBox confirmando la adición del producto, así como su hora
Escenario principal	Se presiona el botón de añadir y se realiza la adición del producto, una vez finalizado, se muestra un messagebox con el resultado exitoso
Flujo alterno	Si los datos introducidos son incorrectos en cualquiera de sus apartados, se genera el error correspondiente, que va desde avisos en interfaz gráfica hasta respuestas del servidor con estado de error manejados por messagebox

Especificación del caso de uso “Editar un producto”

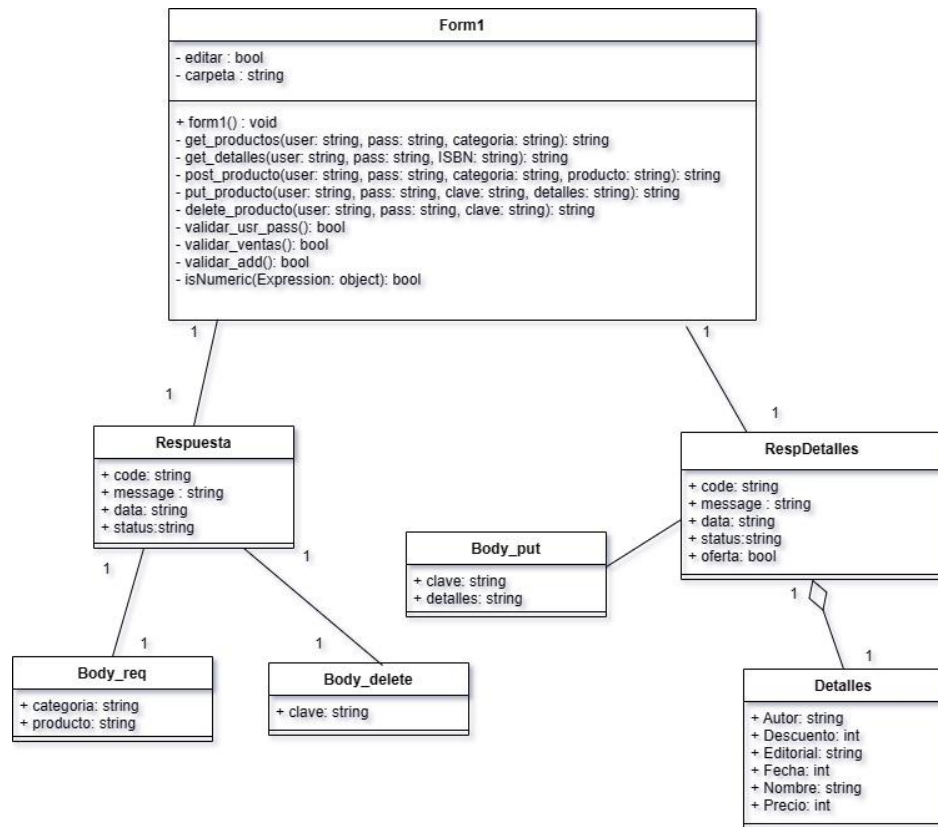
Caso de uso	Editar un producto
-------------	--------------------

Actores	Almacén
Tipo	Primario
Precondición	Los campos de contraseña y usuario no deben estar vacíos, así como los datos del nuevo producto, dichos campos deben tener datos válidos al momento de ejecutar este caso de uso, por ejemplo ciertos campos solo aceptan números enteros
Postcondición	Se mostrará un MessageBox confirmando la adición del producto, así como su hora
Escenario principal	Se presiona el botón editar y se realiza la edición del producto, una vez finalizado, se muestra un messagebox con el resultado exitoso
Flujo alternativo	Si los datos introducidos son incorrectos en cualquiera de sus apartados, se genera el error correspondiente, que va desde avisos en interfaz gráfica hasta respuestas del servidor con estado de error manejados por messagebox

Especificación del caso de uso “Eliminar un producto”

Caso de uso	Eliminar un producto
Actores	Almacén
Tipo	Primario
Precondición	
Postcondición	
Escenario principal	Se presiona el botón eliminar y se realiza la operación de eliminación del producto, una vez finalizado, se muestra un messagebox con el resultado exitoso
Flujo alternativo	Si no se selecciona un producto, el botón eliminar no realizará una acción, si se intenta eliminar un producto inexistente se mostrará un error de producto no encontrado

Diagrama de clases



3. Pruebas con POSTMAN

3.1 Operación 1

Lista de productos

GET ⌵ http://localhost:8080/2ndtestslim/productos/comics

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Headers ⌵ 6 hidden

	Key	Value
<input checked="" type="checkbox"/>	user	pruebas1
<input checked="" type="checkbox"/>	pass	12345678a
	Key	Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON ⌵ ≡

```

1  {
2    "code": "200",
3    "message": "Categoria encontrada exitosamente.",
4    "data": {
5      "COM001": "Comic 1",
6      "COM002": "Comic 2",
7      "COM003": "Comic 3",
8      "COM004": "tetronimos"
9    },
10   "status": "success"
11 }

```

3.2 Operación 2

Lista de detalles de un producto

GET ⌵ http://localhost:8080/2ndtestslim/detalles/COM001

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Headers ⌵ 6 hidden

	Key	Value
<input checked="" type="checkbox"/>	user	pruebas1
<input checked="" type="checkbox"/>	pass	12345678a
	Key	Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON ⌵ ≡

```

1  {
2    "code": "201",
3    "message": "ISBN encontrado exitosamente.",
4    "data": {
5      "Autor": "Autor X",
6      "Descuento": 2,
7      "Editorial": "Editorial X",
8      "Fecha": "2020",
9      "Nombre": "Comic 1",
10     "Precio": 100
11   },
12   "status": "success",
13   "oferta": true
14 }

```

3.3 Operación 3

Insertar producto con detalles


POST ▼ <http://localhost:8080/WS/p08/producto ...>

Params Headers (2) **Body** ●

● none ● form-data ● **x-www-form-urlencoded** ● raw ● binary ● GraphQL

Key	Value
<input checked="" type="checkbox"/> producto	{ "Autor": "Fedor Dostoievski", "De:

Body Headers (7)

Pretty Raw Preview JSON ▼ 

```

1 {
2   "code": "202",
3   "message": "Producto registrado correctamente.",
4   "data": "2023-03-24 06:49:35 - CET",
5   "status": "success"
6 }
```

3.4 Operación 4

Actualizar detalles de un producto


PUT ▼ <http://localhost:8080/WS/p08/producto/detalles...>

Params Headers (2) **Body** ●

● none ● form-data ● **x-www-form-urlencoded** ● raw ● binary ● GraphQL

Key	Value
<input checked="" type="checkbox"/> detalles	{ "Autor": "Eiichirō Oda", "Descuento": 0, "Editor...
<input checked="" type="checkbox"/> clave	COM001

Body Headers (7)

Pretty Raw Preview JSON ▼ 

```

1 {
2   "code": "203",
3   "message": "Producto actualizado correctamente.",
4   "data": "2023-03-24 15:46:45 - CET",
5   "status": "success",
6   "descuento": false
7 }
```

3.5 Operación 5

Eliminar producto con sus detalles

DELETE ⌵ http://localhost:8080/WS/p08/producto ...

Params Headers (2) **Body** ●

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary

<input checked="" type="checkbox"/>	clave	MAN002
<input type="checkbox"/>	detalles	{ "Autor": "yo", "I
	Key	Value

Body Headers (7)

Pretty Raw Preview JSON ⌵ ↺


```
1  {
2    "code": "204",
3    "message": "Producto borrado correctamente.",
4    "data": "2023-03-24 05:58:04 - CET",
5    "status": "success"
6  }
```

4. Evidencias del consumo del servicio desde Aplicacion .NET

4.1 Operación 1

Lista de productos

Gestor de contenidos [cliente] — □ ×

Usuario Contraseña 

Ventas **Almacen**

Tipo de operación: Mostrar productos ⌵

Categoría:

Resultado de la búsqueda ↺

Código de resultado: 200

Respuesta: Categoría encontrada exitosamente.

Estado: success

Datos

ISBN	Nombre
COM001	Comic 1
COM002	Comic 2
COM003	Comic 3
COM004	tetronimos

4.2 Operación 2

Lista de detalles de un producto

Gestor de contenidos [cliente]

Usuario: pruebas2 Contraseña:

Ventas **Almacen**

Tipo de operación: **Mostrar detalles**

ISBN: COM002

Buscar

Resultado de la búsqueda

Código de resultado: 201

Respuesta: ISBN encontrado exitosamente.

Estado: success

Propiedad	Valor
Nombre	Comic 2
Autor	Autor X
Descuento	0
Editorial	Editorial Y
Fecha	2020
Precio	1

4.3 Operación 3

Insertar un producto

Area de edición / adición

ISBN: --Generado automaticamente

Autor: Fedor Dostoievky

Editorial: Porrúa

Precio: 500

Categoría: Libros

Nombre: Crimen y Castigo

Fecha: 2005

Descuento: 100

Añadir Cancelar

Operación exitosa [202]

Operacion realizada con codigo: 202 y con mensaje:
Producto registrado correctamente. con hora de accion
2023-03-27 21:39:41 - CEST

Aceptar

4.4 Operación 4

Actualizar detalles de un producto

Gestor de contenidos [cliente]

Usuario Contraseña

Ventas **Almacen**

Categoría:

ISBN	Nombre
MAN001	Manga 1
MAN003	Manga 3
MAN004	mandarina
MAN005	mango
MAN006	autobiografía

Operación exitosa [203]

Operación realizada con código: 203 y con mensaje:
Producto actualizado correctamente. con hora de acción
2023-03-27 21:35:42 - CEST

Area de edición / adición

ISBN Categoría

Autor Nombre

Editorial Fecha

Precio Descuento

4.5 Operación 5

Eliminar producto con sus detalles

Ventas **Almacen**

Categoría:

ISBN	Nombre
LIB001	no.
LIB002	se actualiza :C
LIB003	Libro 3
LIB004	chistes matemati...
LIB005	pain, suffering ev...
LIB006	Crimen y Castigo

Operación exitosa [204]

Operación realizada con código: 204 y con mensaje:
Producto borrado correctamente. con hora de acción
2023-03-27 21:41:08 - CEST