



## SPRINT 3: Desarrollo del Backend

### Identificación Proyecto

Nombre Proyecto:	Vaper Shop App
Número Equipo:	
<b>Integrantes del equipo</b>	
Rol (Líder-Desarrollador – Cliente)	Nombre
(Líder-Desarrollador – Cliente)	Weimar Moisés Álvarez Cardona
(Líder-Desarrollador – Cliente)	Diego Alejandro Vera Jiménez

### Evidencia construcción del Backend

Construcción de Backend con Spring-boot y Visual Studio

The screenshot shows the Visual Studio Code interface with the 'wmsence' project. The Explorer on the left shows the project structure, including 'src/main/java/com/wmsence/wmsence' with sub-packages 'Controllers', 'Models', 'Repositories', and 'Services'. The main editor displays the code for 'ProductosController.java'.

```

1 package com.wmsence.wmsence.controllers;
2
3 import com.wmsence.wmsence.Models.ProductosModel;
4 import com.wmsence.wmsence.Services.ProductosServices;
5 import com.wmsence.wmsence.Repositories.ProductosRepository;
6
7 import java.util.List;
8
9 import org.springframework.web.bind.annotation.*;
10
11 import org.springframework.http.HttpStatus;
12 import org.springframework.http.ResponseEntity;
13
14 import org.springframework.beans.factory.annotation.Autowired;
15
16 @RestController
17 @CrossOrigin("*")
18 @RequestMapping("/productos")
19 public class ProductosController {
20     @Autowired
21     private ProductosRepository productosRepository;
22
23     @Autowired
24     private ProductosServices productosServices;
25
26     @PostMapping(value = "/")
27     @ResponseBody
28     public ResponseEntity<ProductosModel> agregar (@RequestBody ProductosModel productosModel){
29         System.out.println(productosModel);
30         ProductosModel obj = productosServices.save(productosModel);
31         return new ResponseEntity<>(obj, HttpStatus.OK);
32     }
33
34     @DeleteMapping(value =("/{id}")
35     public ResponseEntity<ProductosModel> eliminar (@PathVariable Integer id){
36         ProductosModel obj = productosServices.findById(id);
37         if (obj != null){
38             productosServices.delete(id);
39         } else {
40             return new ResponseEntity<>(obj, HttpStatus.INTERNAL_SERVER_ERROR);
41         }
42     }
43 }
  
```



Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

wmessence

EXPLORADOR

- WMESSENCE
  - .mvn
  - .vscode
  - src
    - main
      - java
        - com
          - wmessence
            - Controllers
              - CategoriasController.java
              - ProductosController.java
              - UsuariosController.java
            - Models
              - CategoriasModel.java
              - ProductosModel.java
              - UsuariosModel.java
            - Repositories
              - CategoriasRepository.java
              - ProductosRepository.java
              - UsuariosRepository.java
            - Services
              - CategoriasServices.java
              - ProductosServices.java
              - UsuariosServices.java
              - WmessenceApplication.java
            - resources
              - application.properties

src > main > resources > application.properties

```

1  You, la semana pasada | 1 author (You)
2  # =====
3  # = DATA_SOURCE
4  # =====
5  # Set here configurations for the database connection
6  spring.datasource.url=jdbc:mariadb://localhost:3306/cn03_wmessence_01
7  spring.datasource.username=root
8  spring.datasource.password=wm.alucard.03
9  spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
10 # Keep the connection alive if idle for a long time (needed in production)
11 # spring.datasource.testWhileIdle=true
12 # spring.datasource.validationQuery=SELECT 1
13 # =====
14 # = JPA / HIBERNATE
15 # =====
16 # Show or not log for each sql query
17 spring.jpa.show-sql=true
18 # Hibernate ddl auto (create, create-drop, update): with "create-drop" the database
19 # schema will be automatically created afresh for every start of application
20 spring.jpa.hibernate.ddl-auto=update
21 # Naming strategy
22 # spring.jpa.hibernate.naming.implicit-strategy=org.hibernate.boot.model.naming.ImplicitNamingStrategyLegacyHbmImpl
23 # spring.jpa.hibernate.naming.physical-strategy=org.springframework.boot.orm.jpa.hibernate.SpringPhysicalNamingStrategy
24 # Allows Hibernate to generate SQL optimized for a particular DBMS
25 # spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
  
```

src > main > java > com > wmessence > wmessence > Models > ProductosModel.java > ...

```

1  You, hace 8 minutos | 1 author (You)
2  package com.wmessence.wmessence.odels;
3
4  import java.io.Serializable;
5
6  import javax.persistence.*;
7
8  import lombok.Getter;
9  import lombok.Setter;
10
11 You, hace 8 minutos | 1 author (You)
12 @Getter
13 @Setter
14 @Entity
15 @Table(name = "cn03_productos")
16 public class ProductosModel implements Serializable{
17     @Id
18     @GeneratedValue(strategy = GenerationType.IDENTITY)
19     @Column(name="_id")
20     private int _id;
21     @Column(name="product_name")
22     private String product_name;
23     @Column(name="product_category")
24     private int product_category;
25     @Column(name="product_description")
26     private String product_description;
27     @Column(name="product_cost")
28     private String product_cost;
29
30     @Override
31     public String toString(){
32         return "Categorías [_id:" + _id + ", product_name:" + product_name + ", product_category:" + product_category + ",
33     }
  
```



```
src > main > java > com > wmessence > wmessence > Models > UsuariosModel.java > {} com.wmessence.wmessence.Models
You, la semana pasada | 1 author (You)
1 package com.wmessence.wmessence.odels;
2
3 import java.io.Serializable;
4
5 import javax.persistence.*;
6
7 import lombok.Getter;
8 import lombok.Setter;
9
10 @Getter
11 @Setter
12 @Entity
13 @Table(name = "cn03_usuarios")
14 public class UsuariosModel implements Serializable{
15     @Id
16     @GeneratedValue(strategy = GenerationType.IDENTITY)
17     @Column(name="_id")
18     private int _id;
19     @Column(name="usser_name")
20     private String usser_name;
21     @Column(name="usser_passwd")
22     private String usser_passwd;
23     @Column(name="usser_email")
24     private String usser_email;
25
26     @Override
27     public String toString(){
28         return "Categorías [_id:" + _id + ", usser_name:" + usser_name + ", usser_email:" + usser_email + "];";
29     }
30 }
31
```

```
Terminal Ayuda < -> wmessence
UsuariosController.java 3, M CategoríasModel.java X TC New Request TC localhost:8080/productos/...
src > main > java > com > wmessence > wmessence > Models > CategoríasModel.java > ...
You, la semana pasada | 1 author (You)
1 package com.wmessence.wmessence.odels;
2
3 import java.io.Serializable;
4
5 import javax.persistence.*;
6
7 import lombok.Getter;
8 import lombok.Setter;
9
10 @Getter
11 @Setter
12 @Entity
13 @Table(name = "cn03_categorias")
14 public class CategoríasModel implements Serializable{
15     @Id
16     @GeneratedValue(strategy = GenerationType.IDENTITY)
17     @Column(name="_id")
18     private int _id;
19     @Column(name="category_name")
20     private String category_name;
21
22     @Override
23     public String toString(){
24         return "Categorías [_id:" + _id + ", category_name:" + category_name + "];";
25     }
26 }
27
```



## Código JAVA Categorías Controller

```
package com.wmessenger.wmessenger.Controllers;

import com.wmessenger.wmessenger.Models.CategoriasModel;
import com.wmessenger.wmessenger.Repositories.CategoriasRepository;
import com.wmessenger.wmessenger.Services.CategoriasServices;

import java.util.List;

import org.springframework.web.bind.annotation.*;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;

import org.springframework.beans.factory.annotation.Autowired;

@RestController
@CrossOrigin ("*")
@RequestMapping ("/categorias")
public class CategoriasController {

    @Autowired
    private CategoriasRepository categoriasRepository;

    @Autowired
    private CategoriasServices categoriasServices;

    @PostMapping (value = "/")
    @ResponseBody
    public ResponseEntity <CategoriasModel> agregar (@RequestBody CategoriasModel categoriasModel){
        System.out.println(categoriasModel);
        CategoriasModel obj = categoriasServices.save(categoriasModel);
        return new ResponseEntity<>(obj, HttpStatus.OK);
    }

    @DeleteMapping (value =("/{id}")
    public ResponseEntity <CategoriasModel> eliminar (@PathVariable Integer id){
        CategoriasModel obj = categoriasServices.findById(id);
        if (obj != null){
            categoriasServices.delete(id);
        } else {
            return new ResponseEntity<>(obj, HttpStatus.INTERNAL_SERVER_ERROR);
        }
    }
}
```



```
    }  
    return new ResponseEntity<>(obj, HttpStatus.OK);  
}  
@PutMapping (value = "/")  
public ResponseEntity <CategoriasModel> editar (@RequestBody CategoriasModel categoriasModel){  
    CategoriasModel obj = categoriasServices.findById(categoriasModel.get_id());  
    if(obj != null){  
        obj.setCategory_name(categoriasModel.getCategory_name());  
        categoriasServices.save(obj);  
    } else {  
        return new ResponseEntity<>(obj, HttpStatus.INTERNAL_SERVER_ERROR);  
    }  
    return new ResponseEntity<>(obj, HttpStatus.OK);  
}  
@GetMapping ("/list")  
public List <CategoriasModel> consultarTodo(){  
    return categoriasServices.findAll();  
}  
@GetMapping ("/list/{id}")  
public CategoriasModel consultarPorId(@PathVariable Integer id){  
    return categoriasServices.findById(id);  
}  
}
```

## Código JAVA Productos Controller

```
package com.wmessenger.wmessenger.Controllers;  
  
import com.wmessenger.wmessenger.Models.ProductosModel;  
import com.wmessenger.wmessenger.Services.ProductosServices;  
import com.wmessenger.wmessenger.Repositories.ProductosRepository;  
  
import java.util.List;  
  
import org.springframework.web.bind.annotation.*;  
  
import org.springframework.http.HttpStatus;  
import org.springframework.http.ResponseEntity;  
  
import org.springframework.beans.factory.annotation.Autowired;
```



```
@RestController
@CrossOrigin ("*")
@RequestMapping ("/productos")
public class ProductosController {
    @Autowired
    private ProductosRepository productosRepository;

    @Autowired
    private ProductosServices productosServices;

    @PostMapping (value = "/")
    @ResponseBody
    public ResponseEntity<ProductosModel> agregar (@RequestBody ProductosModel productosModel){
        System.out.println(productosModel);
        ProductosModel obj = productosServices.save(productosModel);
        return new ResponseEntity<>(obj, HttpStatus.OK);
    }
    @DeleteMapping (value =("/{id}")
    public ResponseEntity<ProductosModel> eliminar (@PathVariable Integer id){
        ProductosModel obj = productosServices.findById(id);
        if (obj != null){
            productosServices.delete(id);
        } else {
            return new ResponseEntity<>(obj, HttpStatus.INTERNAL_SERVER_ERROR);
        }
        return new ResponseEntity<>(obj, HttpStatus.OK);
    }
    @PutMapping (value = "/")
    public ResponseEntity<ProductosModel> editar (@RequestBody ProductosModel productosModel){
        ProductosModel obj = productosServices.findById(productosModel.get_id());
        if (obj != null){
            obj.setProduct_name(productosModel.getProduct_name());
            obj.setProduct_description(productosModel.getProduct_description());
            obj.setProduct_category(productosModel.getProduct_category());
            obj.setProduct_cost(productosModel.getProduct_cost());
            productosServices.save(obj);
        } else {
            return new ResponseEntity<>(obj, HttpStatus.INTERNAL_SERVER_ERROR);
        }
        return new ResponseEntity<>(obj, HttpStatus.OK);
    }
}
```



```
}  
@GetMapping ("/list")  
public List<ProductosModel> consultarTodo(){  
    return productosServices.findAll();  
}  
@GetMapping ("/list/{id}")  
public ProductosModel consultarPorId (@PathVariable Integer id){  
    return productosServices.findById(id);  
}  
@GetMapping ("/consulta")  
@ResponseBody  
public List <ProductosModel> qry_product_category(@RequestParam ("qpc") Integer qpc){  
    return productosServices.qry_product_category(qpc);  
}  
}
```

## Evidencias de los "endpoint" con el consumo de recursos del API REST

Método GET en la función LIST de la tabla Productos.

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://localhost:8080/productos/list
- Status:** 200 OK
- Size:** 1.54 KB
- Time:** 19 ms
- Response Body:**

```
[  
  {  
    "product_name": "Barcelona FC Titular",  
    "product_category": "2",  
    "product_description": "Camisa Azul/Rojo 2022",  
    "product_cost": "270000"  
  },  
  {  
    "_id": 10,  
    "product_name": "Juventus FC Titular",  
    "product_category": "2",  
    "product_description": "Camisa Blanco/Negro 2022",  
    "product_cost": "230000"  
  },  
  {  
    "_id": 11,  
    "product_name": "Paris Saint German Titular",  
    "product_category": "2",  
    "product_description": "Camisa Azul 2022",  
    "product_cost": "230000"  
  }  
]
```





Método POST para AGREGAR ítems a la tabla Productos.

POST <http://localhost:8080/productos/> Send

Status: 200 OK Size: 142 Bytes Time: 115 ms

Response

```
1 {
2   "_id": 11,
3   "product_name": "Paris Saint German Titular",
4   "product_category": "2",
5   "product_description": "Camisa Azul 2022",
6   "product_cost": "230000"
7 }
```

Json Content

```
1 {
2   "product_name": "Paris Saint German Titular",
3   "product_category": "2",
4   "product_description": "Camisa Azul 2022",
5   "product_cost": "230000"
6 }
```

Método PUT para EDITAR ítems de la tabla Productos.

PUT <http://localhost:8080/productos/> Send

Status: 200 OK Size: 130 Bytes Time: 20 ms

Response

```
1 {
2   "_id": 11,
3   "product_name": "PSG FC Titular",
4   "product_category": "2",
5   "product_description": "Camisa Azul 2022",
6   "product_cost": "220000"
7 }
```

Json Content

```
1 {
2   "_id": 11,
3   "product_name": "PSG FC Titular",
4   "product_category": 2,
5   "product_description": "Camisa Azul 2022",
6   "product_cost": "220000"
7 }
```

Método GET en la función LIST de la tabla Categorías.

GET <http://localhost:8080/categorias/list> Send

Status: 200 OK Size: 152 Bytes Time: 33 ms

Response

```
1 [
2   {
3     "_id": 1,
4     "category_name": "Camisa Seleccionado de Futbol"
5   },
6   {
7     "_id": 2,
8     "category_name": "Camisa Club de Futbol"
9   },
10  {
11    "_id": 3,
12    "category_name": "Ropa deportiva"
13  }
14 ]
```

Query Parameters

parameter	value
-----------	-------





Método GET en la función LIST de la tabla Usuarios.

Query Parameters

parameter	value

Status: 200 OK Size: 275 Bytes Time: 29 ms

Response

```

1  [
2    {
3      "_id": 1,
4      "usser_name": "Weimar Alvarez",
5      "usser_passwd": "123",
6      "usser_email": "wm@gmail.com"
7    },
8    {
9      "_id": 2,
10     "usser_name": "Leanny Sanchez",
11     "usser_passwd": "456",
12     "usser_email": "ls@hotmail.com"
13   },
14   {
15     "_id": 3,
16     "usser_name": "Carmen Cardona",
17     "usser_passwd": "987",
18     "usser_email": "cc@outlook.com"
19   }
20 ]
  
```

## Evidencia GitLab o GitHub

WMRioga / CN03\_WMEssence Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file > Code

File	Commit	Time
Backend/wmessence	CN03_G02_08	2 minutes ago
Documentacion	CN03_G02_08	2 minutes ago
Front	Carga Inicial de Información	24 days ago
Leeme.txt	Prueba de carga	24 days ago
README.md	Initial commit	25 days ago

README.md

### CN03\_WMEssence

Ciclo Nivelatorio 03 -- CN03\_WMEssence

About

Ciclo Nivelatorio 03 -- CN03\_WMEssence

Readme

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package



File Edit View Repository Branch Help

Current repository: CN03\_WMEssence | Current branch: main | Push origin: Last fetched just now

Changes 1 | History

No branches to compare

CN03\_G02\_08: WMRIoga • Just now

CN03\_G02\_07: WMRIoga • yesterday

CN03\_G02\_07: WMRIoga • 2 days ago

CN03\_G02\_07: WMRIoga • Nov 22, 2022

CN03\_G02\_06: Weimar M Alvarez C • Nov 21, 2022

CN03\_G02\_05: WMRIoga • Nov 21, 2022

CN03\_G02\_04: WMRIoga • Nov 19, 2022

Merge branch 'main' of https://github...: Weimar M Alvarez C • Nov 18, 2022

CN03\_G02\_03: Weimar M Alvarez C • Nov 18, 2022

CN03\_G02\_02: Weimar M Alvarez C • Nov 17, 2022

CN03\_G02\_01: Weimar M Alvarez C • Nov 7, 2022

Prueba de carga: Weimar M Alvarez C • Nov 5, 2022

Carga Inicial de Información

CN03\_G02\_07: Se generan el Modelo Error y Validación de campos

File	Changes	Diff
Backend\wmesence\pom.xml	53 53	@@ -53,8 +53,11 @@
Backend\...\CategoriasController.java	54 54	<artifactId>spring-boot-starter-test</artifactId>
Backend\wmesence\...\CategoriasModel.java	56 +	<scope>test</scope>
Backend\wmesence\...\ErrorModel.java	57 +	</dependency>
Backend\wmesence\...\ProductosModel.java	58 +	<dependency>
Backend\wmesence\...\UsuariosModel.java	59 +	<groupId>org.springframework.boot</groupId>
Backend\wmesence\...\CategoriasRepository.java	56 60	<artifactId>spring-boot-starter-validation</artifactId>
Backend\wmesence\...\ProductosRepository.java	57 -	</dependency>
Backend\wmesence\...\CategoriasServices.java	58 61	</dependencies>
Backend\wmesence\...\ProductosServices.java	59 62	<build>
Backend\wmesence\...\ValidationHandler.java	60 63	<plugins>
Documentaci...\Formato_Sprint_3.docx		</plugin>

## Evidencia JIRA (Seguimiento del proyecto)

Jira Software | Tu trabajo | Proyectos | Filtros | Paneles | Personas | Aplicaciones | Crear

Buscar

CN03\_WMEssence | Proyecto de software

PLANIFICACIÓN

Hoja de ruta

Backlog

Tablero

DESARROLLO

Código

Páginas de proy...

Añadir acceso rápido

Configuración del pr...

Proyectos / CN03\_WMEssence

### Backlog

WA Epic

Insights

Tablero Sprint 2 7 nov – 22 nov (4 incidencias)

Modelado de Base de datos y Backend con Spring-boot + VS Code

0 0 0 Completar sprint

✓ Ew-7 Esquema Base de datos	CREACIÓN Y MODELADO BACKEND	FINALIZADA	WA
✓ Ew-8 Diseño de la base de datos	CREACIÓN Y MODELADO BACKEND	FINALIZADA	WA
✓ Ew-9 Modelado Spring-boot JAVA	CREACIÓN Y MODELADO BACKEND	FINALIZADA	WA
✓ Ew-10 Pruebas de funcionamiento CRUD	CREACIÓN Y MODELADO BACKEND	FINALIZADA	WA

+ Crear incidencia



		NOV	
Sprints		<input type="checkbox"/> Tablero Sprint 2	
▼ <b>CW-1 Definición del proyecto</b> <input checked="" type="checkbox"/> CW-4 Crear el Repositorio. FINALIZADA <input checked="" type="checkbox"/> CW-2 Describir los entregables y tiempos del proyecto. FINALIZADA <input checked="" type="checkbox"/> CW-3 Objetivos generales y Específicos. FINALIZADA <input checked="" type="checkbox"/> CW-5 Describir los requerimientos funcionales y no funcionales. FINALIZADA			
▼ <b>CW-6 Creación y Modelado Backend</b> <input checked="" type="checkbox"/> CW-7 Esquema Base de datos FINALIZADA WEIMAR A... <input checked="" type="checkbox"/> CW-8 Diseño de la base de datos FINALIZADA WEIMAR A... <input checked="" type="checkbox"/> CW-9 Modelado Spring-boot JAVA FINALIZADA WEIMAR A... <input checked="" type="checkbox"/> CW-10 Pruebas de funcionamiento CRUD FINALIZADA WEIMAR A...			
▼ <b>CW-11 Creación Frontend</b> <input checked="" type="checkbox"/> CW-12 Creación modelado HTML TAREAS PO... WEIMAR A... <input checked="" type="checkbox"/> CW-13 Conexión Front & Back TAREAS PO... WEIMAR A... <input checked="" type="checkbox"/> CW-14 Pruebas del Funcionamiento TAREAS PO... WEIMAR A...			

Tu trabajo ▼
Proyectos ▼
Filtros ▼
Paneles ▼
Personas ▼
Aplicaciones ▼
Crear

**CN03\_WMEssence**  
 Proyecto de software

PLANIFICACIÓN  
 Hoja de ruta  
 Backlog  
**Tablero**  
 DESARROLLO  
 Código  
 Páginas de proy...  
 Añadir acceso rápido  
 Configuración del pr...

Estás en un proyecto gestionado por el equipo  
[Más información](#)

Proyectos / CN03\_WMEssence  
**Tablero Sprint 3**

Restantes: 3 días
 Terminar sprint
 ...

Epic ▼

AGRUPAR POR Nada ▼
 Insights

POR HACER 3 INCIDENCIAS
 Creación modelado HTML  
 CREACIÓN FRONTEND  
☒ CW-12

Conexión Front & Back  
 CREACIÓN FRONTEND  
☒ CW-13

Pruebas del Funcionamiento  
 CREACIÓN FRONTEND  
☒ CW-14

EN CURSO

LISTO ✓