

Actividad 3 - Funcionamiento del Carrito


Desarrollo de Sistemas Web II

Ingeniería en Desarrollo de Software

Tutor: Aaron Iván Salazar Macias

Alumno: Homero Ramirez Hurtado

Fecha: 13 de Febrero del 2024



Índice.

. Introducción.

. Descripción.

. Justificación.

. Desarrollo:

- Codificación.
- Pruebas del Sitio Web.

. Conclusión.

. Referencias.



Introducción.

Las API REST (Interfaz de Programación de Aplicaciones basada en el estilo Representational State Transfer) desempeñan un papel crucial en el mundo digital actual. Permiten la comunicación eficiente y segura entre aplicaciones, mejorando la experiencia del usuario y aumentando la eficiencia empresarial.

En esencia, una API REST es un conjunto de buenas prácticas para las solicitudes HTTP realizadas por una API en una aplicación web. Algunos puntos clave sobre su importancia son:

- **Comunicación entre sistemas:** Las API REST permiten que diferentes sistemas se comuniquen sin problemas. Esto es fundamental para integrar aplicaciones web y móviles, así como para conectar servicios de terceros.
- **Estándares definidos:** Las API REST siguen restricciones específicas, como el principio cliente-servidor, la independencia de estado y el uso de caché. Estos estándares garantizan una comunicación coherente y confiable.
- **Escalabilidad y flexibilidad:** Las API REST son escalables y se adaptan a las necesidades cambiantes de una aplicación. Pueden manejar grandes volúmenes de solicitudes sin problemas.
- **Seguridad:** Las API REST pueden implementar medidas de seguridad, como autenticación y autorización, utilizando estándares como OAuth. Esto las hace aptas para aplicaciones sensibles.

En resumen, las API REST son herramientas fundamentales para el desarrollo de aplicaciones modernas. Facilitan el intercambio de datos y contribuyen al éxito de las empresas en un mundo cada vez más conectado.

Descripción.

Las API REST (Interfaz de Programación de Aplicaciones Representational State Transfer) son fundamentales en el desarrollo de sistemas web y desempeñan un papel crucial en la comunicación entre diferentes componentes de una aplicación. Aquí te explico su relevancia:

- **Comunicación eficiente:** Las API REST permiten que aplicaciones y sistemas intercambien datos de manera rápida y segura. Esto es esencial para integrar servicios y compartir información entre diferentes partes de una aplicación.

- Independencia tecnológica: Puedes utilizar una API REST sin importar el lenguaje o tecnología que estés utilizando para desarrollar tu proyecto. Esto facilita la colaboración y la interoperabilidad entre sistemas heterogéneos.
- Flexibilidad: Los desarrolladores se benefician de la versatilidad de las API REST. Pueden adaptar las solicitudes y respuestas según las necesidades específicas de su aplicación.
- Escalabilidad: Al separar el cliente (front-end) del servidor (back-end), las API REST permiten que los sistemas crezcan y se adapten sin afectar otros componentes.

En resumen, las API REST son esenciales para mejorar la experiencia del usuario, aumentar la eficiencia del negocio y garantizar una comunicación fluida entre los diferentes componentes de una aplicación web o móvil.

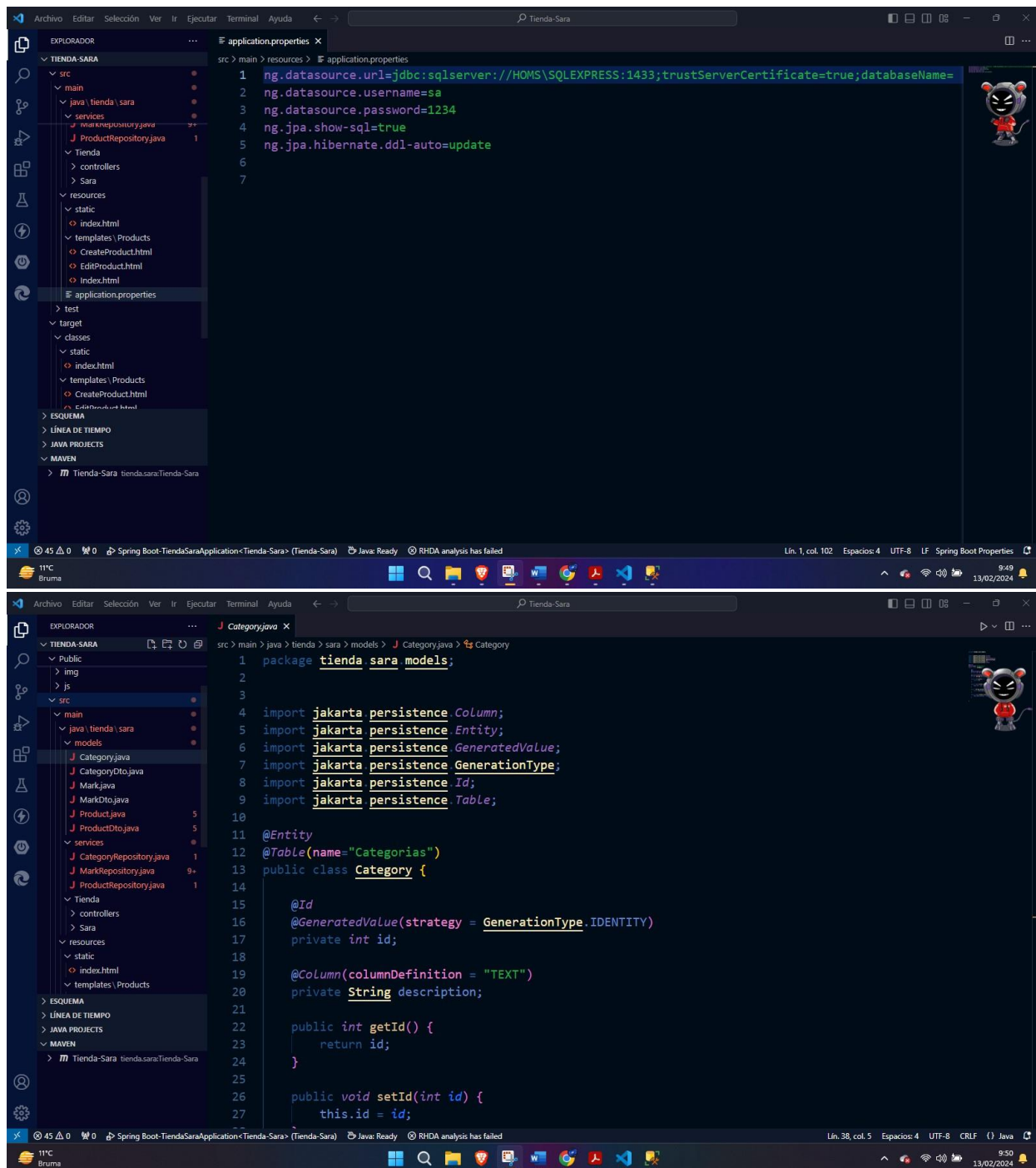
Justificación.

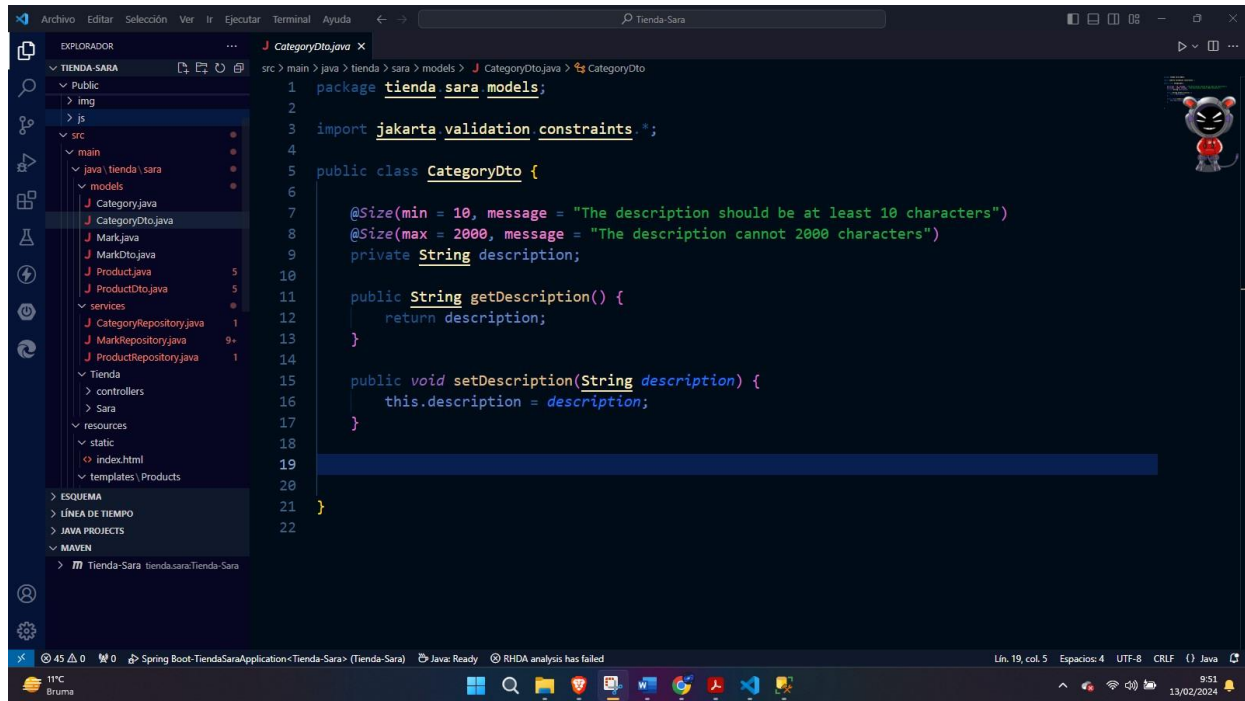
Las API REST son una forma de diseñar e implementar servicios web que se basan en los principios de la arquitectura REST (Representational State Transfer). Algunas de las ventajas de usar las API REST en el desarrollo de sistemas web son las siguientes:

- Facilitan la interoperabilidad entre diferentes plataformas y lenguajes de programación, ya que utilizan el protocolo HTTP y el formato JSON o XML para el intercambio de datos.
- Permiten el escalado y la distribución de los servicios web, ya que se pueden desplegar en múltiples servidores y balancear la carga de las peticiones.
- Simplifican el mantenimiento y la evolución de los servicios web, ya que se basan en un conjunto de recursos identificados por URI y operaciones estandarizadas (GET, POST, PUT, DELETE, etc.).
- Mejoran el rendimiento y la eficiencia de los servicios web, ya que reducen la sobrecarga de datos y aprovechan las características del protocolo HTTP, como la caché y la compresión.
- Fomentan las buenas prácticas de diseño e implementación de los servicios web, ya que siguen un estilo arquitectónico que promueve la modularidad, la reusabilidad, la seguridad y la documentación.

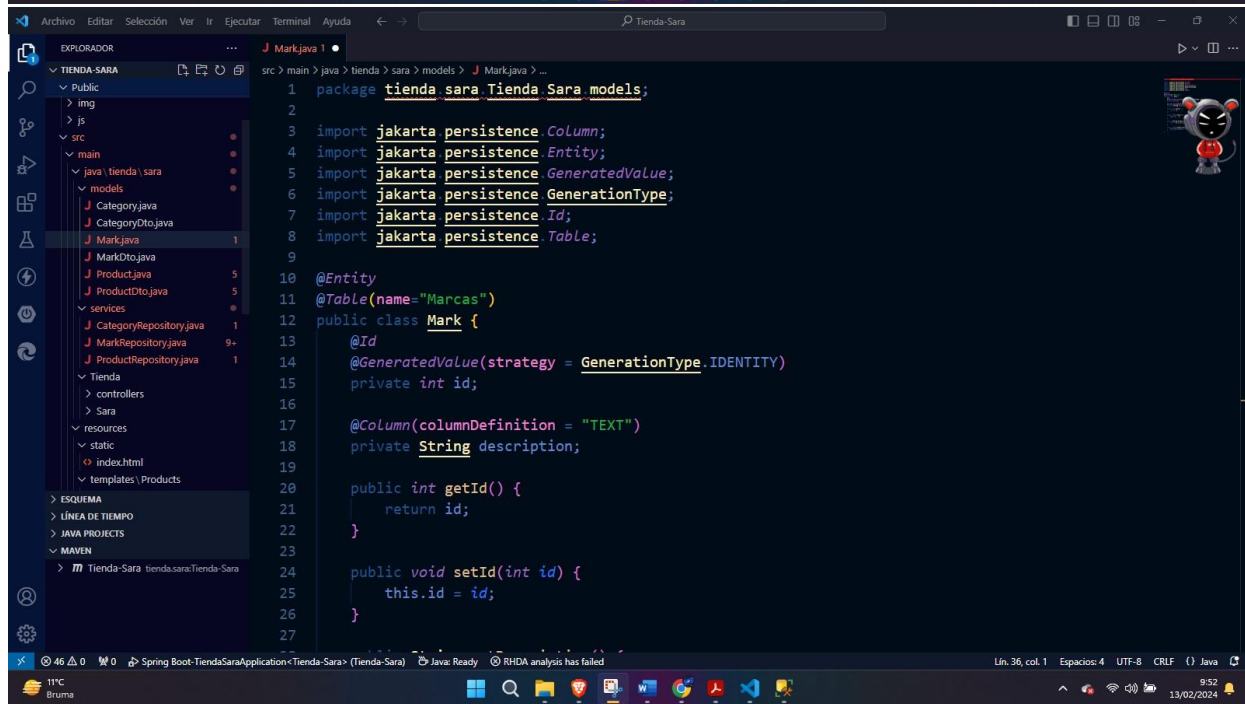
En conclusión, las API REST son una opción recomendable para el desarrollo de sistemas web, ya que ofrecen múltiples beneficios tanto para los desarrolladores como para los usuarios finales.

Desarrollo.

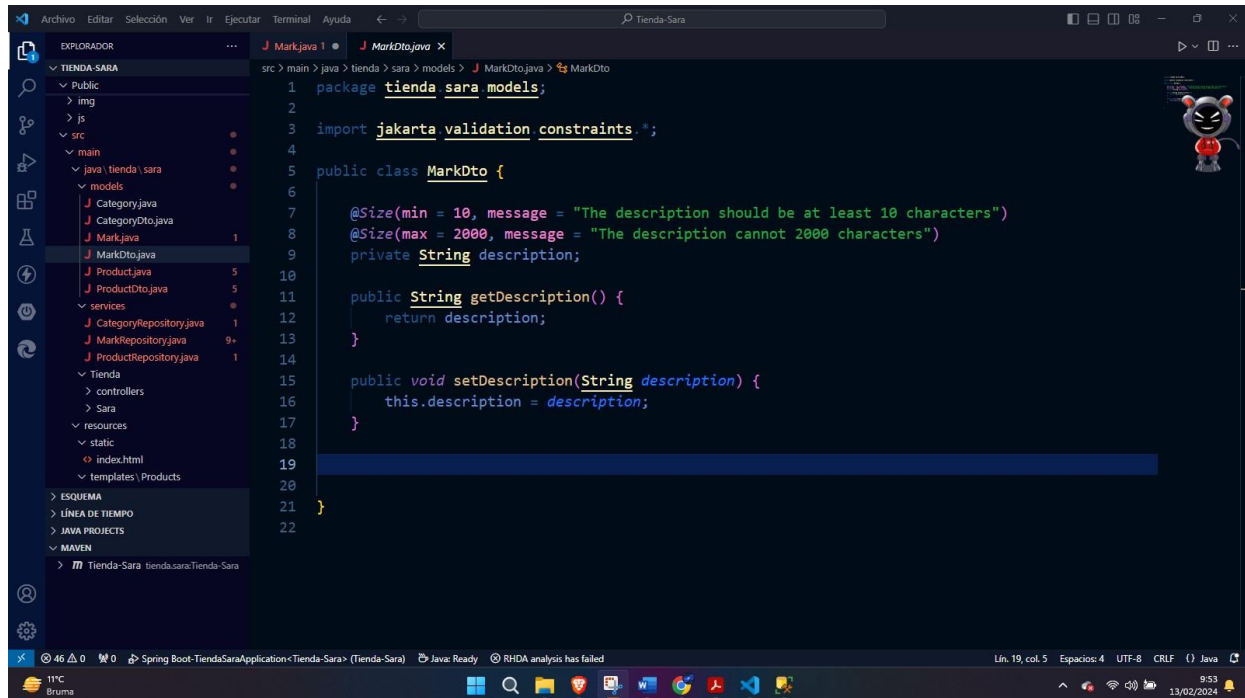




```
1 package tienda.sara.models;
2
3 import jakarta.validation.constraints.*;
4
5 public class CategoryDto {
6
7     @Size(min = 10, message = "The description should be at least 10 characters")
8     @Size(max = 2000, message = "The description cannot 2000 characters")
9     private String description;
10
11     public String getDescription() {
12         return description;
13     }
14
15     public void setDescription(String description) {
16         this.description = description;
17     }
18
19
20
21 }
22
```

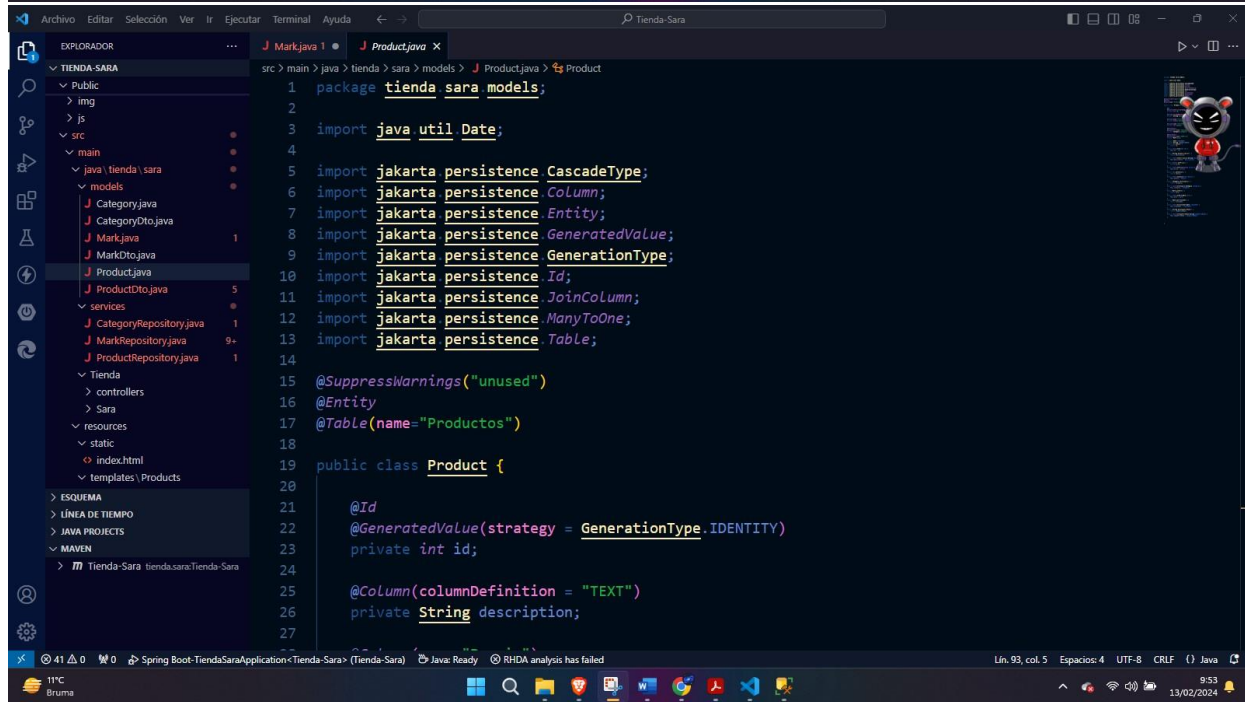


```
1 package tienda.sara.Tienda.Sara.models;
2
3 import jakarta.persistence.Column;
4 import jakarta.persistence.Entity;
5 import jakarta.persistence.GeneratedValue;
6 import jakarta.persistence.GenerationType;
7 import jakarta.persistence.Id;
8 import jakarta.persistence.Table;
9
10 @Entity
11 @Table(name="Marcas")
12 public class Mark {
13     @Id
14     @GeneratedValue(strategy = GenerationType.IDENTITY)
15     private int id;
16
17     @Column(columnDefinition = "TEXT")
18     private String description;
19
20     public int getId() {
21         return id;
22     }
23
24     public void setId(int id) {
25         this.id = id;
26     }
27
28 }
29
```



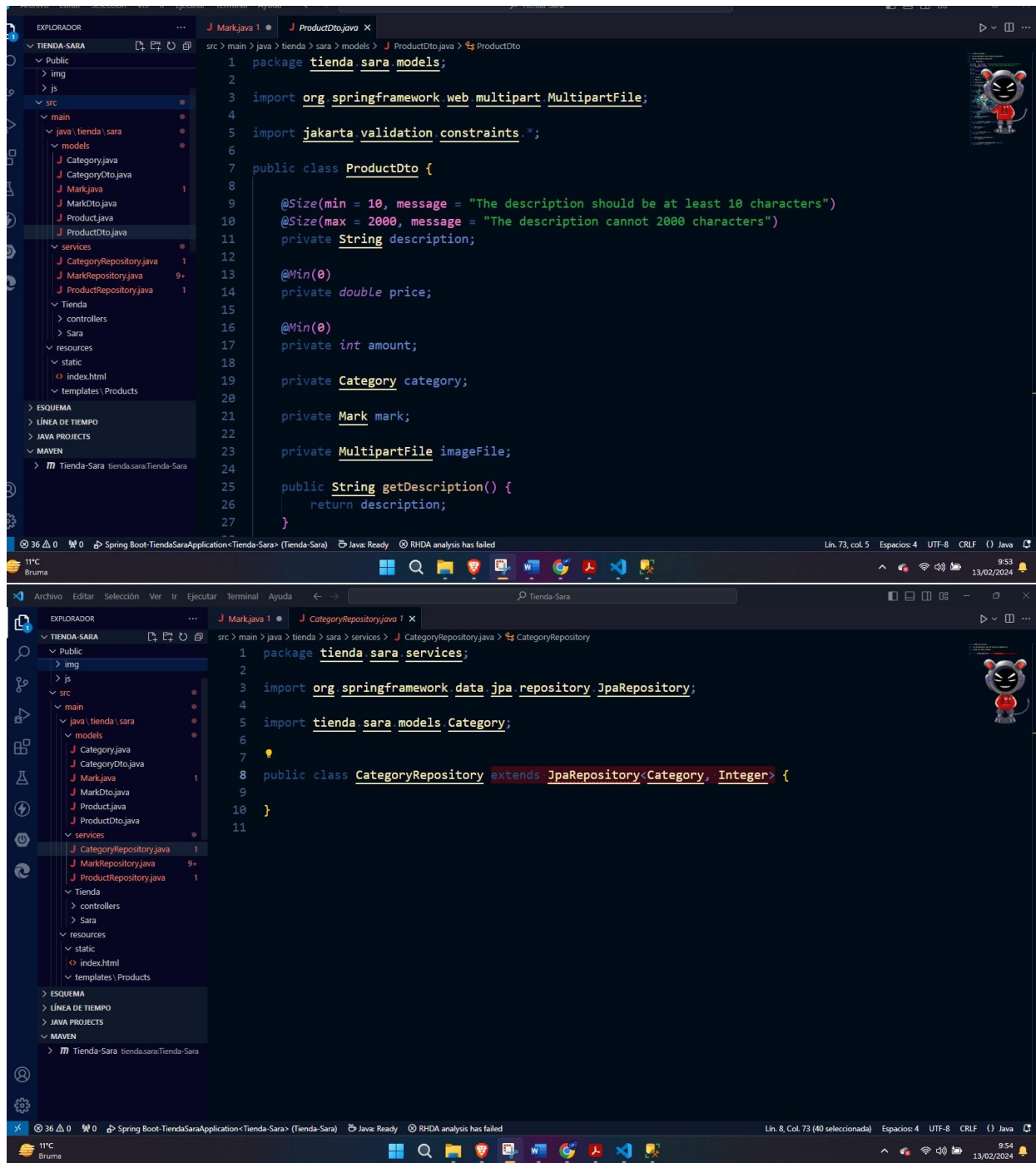
The screenshot shows an IDE with the Tienda-Sara project open. The Explorer on the left shows the project structure, including the models directory. The MarkDto.java file is selected, and its content is displayed in the editor. The code defines a MarkDto class with a description field and methods to get and set the description. The status bar at the bottom indicates the file is at line 19, column 5.

```
1 package tienda.sara.models;
2
3 import jakarta.validation.constraints.*;
4
5 public class MarkDto {
6
7     @Size(min = 10, message = "The description should be at least 10 characters")
8     @Size(max = 2000, message = "The description cannot 2000 characters")
9     private String description;
10
11     public String getDescription() {
12         return description;
13     }
14
15     public void setDescription(String description) {
16         this.description = description;
17     }
18
19 }
20
21
22
```



The screenshot shows the same IDE with the Product.java file selected. The code defines a Product class with an id field and a description field. It uses Jakarta Persistence annotations for entity management. The status bar at the bottom indicates the file is at line 93, column 5.

```
1 package tienda.sara.models;
2
3 import java.util.Date;
4
5 import jakarta.persistence.CascadeType;
6 import jakarta.persistence.Column;
7 import jakarta.persistence.Entity;
8 import jakarta.persistence.GeneratedValue;
9 import jakarta.persistence.GenerationType;
10 import jakarta.persistence.Id;
11 import jakarta.persistence.JoinColumn;
12 import jakarta.persistence.ManyToOne;
13 import jakarta.persistence.Table;
14
15 @SuppressWarnings("unused")
16 @Entity
17 @Table(name="Productos")
18
19 public class Product {
20
21     @Id
22     @GeneratedValue(strategy = GenerationType.IDENTITY)
23     private int id;
24
25     @Column(columnDefinition = "TEXT")
26     private String description;
27
28 }
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
```

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

TIENDA-SARA

EXPLORADOR

- Public
 - img
 - js
 - src
 - main
 - java \ tienda \ sara
 - models
 - Category.java
 - CategoryDto.java
 - Mark.java
 - MarkDto.java
 - Product.java
 - ProductDto.java
 - services
 - CategoryRepository.java
 - MarkRepository.java
 - ProductRepository.java
 - Tienda
 - controllers
 - Sara
 - resources
 - static
 - index.html
 - templates \ Products

ESQUEMA

LÍNEA DE TIEMPO

JAVA PROJECTS

MAVEN

TIENDA-SARA

src > main > java > tienda > sara > services > MarkRepository.java > Language Support for Java(TM) by Red Hat > MarkRepository

```
1 package tienda.sara.services;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5 import tienda.sara.models.Mark;
6
7
8 public class MarkRepository extends JpaRepository<Mark, Integer> {
9
10 }
11
```

Lin. 8, Col. 55 (4 seleccionada) Espacios: 4 UTF-8 CRLF Java

11°C Bruma

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

TIENDA-SARA

EXPLORADOR

- Public
 - img
 - js
 - src
 - main
 - java \ tienda \ sara
 - models
 - Category.java
 - CategoryDto.java
 - Mark.java
 - MarkDto.java
 - Product.java
 - ProductDto.java
 - services
 - CategoryRepository.java
 - MarkRepository.java
 - ProductRepository.java
 - Tienda
 - controllers
 - Sara
 - resources
 - static
 - index.html
 - templates \ Products

ESQUEMA

LÍNEA DE TIEMPO

JAVA PROJECTS

MAVEN

TIENDA-SARA

src > main > java > tienda > sara > services > ProductRepository.java > ...

```
1 package tienda.sara.services;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5 import tienda.sara.models.Product;
6
7
8 public class ProductRepository extends JpaRepository<Product, Integer> {
9
10 }
11
```

Lin. 7, col. 1 Espacios: 4 UTF-8 CRLF Java

11°C Bruma

```
Archivo  Editor  Selección  Ver  Ir  Ejecutar  Terminal  Ayuda  Tienda-Sara
EXPLORADOR
TIENDA-SARA
  Public
  img
  js
  src
  main
    java
      tienda.sara
        models
          Category.java
          CategoryDto.java
          Mark.java
          MarkDto.java
          Product.java
          ProductDto.java
        services
          CategoryRepository.java
          MarkRepository.java
          ProductRepository.java
        tienda
          controllers
            ProductController.java 9+
          Sara
          resources
          static
ESQUEMA
LÍNEA DE TIEMPO
JAVA PROJECTS
MAVEN
  Tienda-Sara tienda.sara:Tienda-Sara

J Mark.java 1 J ProductController.java 9+
src > main > java > tienda > sara > tienda > controllers > J ProductController.java > ...
1 package tienda.sara.Tienda.Sara.controllers;
2
3 import java.io.InputStream;
4 import java.nio.file.Files;
5 import java.nio.file.Path;
6 import java.nio.file.Paths;
7 import java.nio.file.StandardCopyOption;
8 import java.util.Date;
9 import java.util.List;
10
11 import org.springframework.beans.factory.annotation.Autowired;
12 import org.springframework.data.domain.Sort;
13 import org.springframework.stereotype.Controller;
14 import org.springframework.ui.Model;
15 import org.springframework.validation.BindingResult;
16 import org.springframework.validation.FieldError;
17 import org.springframework.web.bind.annotation.GetMapping;
18 import org.springframework.web.bind.annotation.ModelAttribute;
19 import org.springframework.web.bind.annotation.PostMapping;
20 import org.springframework.web.bind.annotation.RequestMapping;
21 import org.springframework.web.bind.annotation.RequestParam;
22 import org.springframework.web.multipart.MultipartFile;
23
24 import tienda.sara.Tienda.Sara.models.Category;
25 import tienda.sara.Tienda.Sara.models.Mark;
26 import tienda.sara.Tienda.Sara.models.Product;
27 import tienda.sara.Tienda.Sara.models.ProductDto;
```

```
Archivo  Editor  Selección  Ver  Ir  Ejecutar  Terminal  Ayuda  Tienda-Sara
EXPLORADOR
TIENDA-SARA
  mvnw
  maven-wrapper.jar
  maven-wrapper.properties
  .vscode
  Public
  css
  img
  js
  src
  target
  classes
  static
  templates
    Products
      CreateProduct.html
      EditProduct.html
      Index.html
  tienda
    application.properties
  test-classes
  .gitignore
  HELP.md
  mvnw
  mvnw.cmd
ESQUEMA
LÍNEA DE TIEMPO
JAVA PROJECTS
MAVEN
  Tienda-Sara tienda.sara:Tienda-Sara

J Mark.java 1 J ProductController.java 9+ J Index.html x
target > classes > templates > Products > J Index.html > ...
1 <!doctype html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <title>Best Store</title>
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
9 </head>
10 <body>
11   <div class="container">
12     <h1 class="text-center my-4">Products</h1>
13     <a class="btn btn-primary" href="/products/create">Create Product</a>
14
15     <table class="table">
16       <thead>
17         <tr>
18           <th>ID</th>
19           <th>Description</th>
20           <th>Price</th>
21           <th>Amount</th>
22           <th>Category</th>
23           <th>Mark</th>
24           <th>Image</th>
25           <th>Created At</th>
26           <th>Action</th>
27         </tr>
```

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

EXPLORADOR

- TIENDA-SARA
 - mvnw.wrapper
 - maven-wrapper.jar
 - maven-wrapper.properties
 - .vscode
 - Public
 - css
 - img
 - js
 - src
 - target
 - classes
 - static
 - index.html
 - templates > Products
 - CreateProduct.html
 - EditProduct.html
 - Index.html
 - tienda
 - application.properties
 - test-classes
 - .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - ESQUEMA
 - LÍNEA DE TIEMPO
 - JAVA PROJECTS
 - MAVEN
 - Tienda-Sara tienda.sara.Tienda-Sara

target > classes > templates > Products > CreateProduct.html

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title>Best Store</title>
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="style
8 </head>
9 <body>
10  <div class="container">
11    <div class="row">
12      <div class="col-md-8 mx-auto rounded border p-4 m-4">
13        <h2 class="text-center mb-5">New Product</h2>
14
15        <form method="post" enctype="multipart/form-data" th:object="${productDto}">
16          <div class="row mb-3">
17            <label class="col-sm-4 col-form-label">Description</label>
18            <div class="col-sm-8">
19              <textarea class="form-control" th:field="${productDto.description}"></textarea>
20              <p th:if="${#fields.hasErrors('description')}" th:errorclass="text-danger"
21                th:errors="${productDto.description}"></p>
22            </div>
23          </div>
24
25          <div class="row mb-3">
26            <label class="col-sm-4 col-form-label">Price</label>
27            <div class="col-sm-8">
```

Lin. 107, col. 1 Espacios 2 UTF-8 CRLF HTML

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

EXPLORADOR

- TIENDA-SARA
 - mvnw.wrapper
 - maven-wrapper.jar
 - maven-wrapper.properties
 - .vscode
 - Public
 - css
 - img
 - js
 - src
 - target
 - classes
 - static
 - index.html
 - templates > Products
 - CreateProduct.html
 - EditProduct.html
 - Index.html
 - tienda
 - application.properties
 - test-classes
 - .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - ESQUEMA
 - LÍNEA DE TIEMPO
 - JAVA PROJECTS
 - MAVEN
 - Tienda-Sara tienda.sara.Tienda-Sara

target > classes > templates > Products > EditProduct.html

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title>Best Store</title>
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="style
8 </head>
9 <body>
10  <div class="container">
11    <div class="row">
12      <div class="col-md-8 mx-auto rounded border p-4 m-4">
13        <h2 class="text-center mb-5">Edit Product</h2>
14
15        <form method="post" enctype="multipart/form-data" th:object="${productDto}">
16          <div class="row mb-3">
17            <label class="col-sm-4 col-form-label">ID</label>
18            <div class="col-sm-8">
19              <input readonly class="form-control-plaintext" th:value="${product.id}" >
20            </div>
21          </div>
22
23          <div class="row mb-3">
24            <label class="col-sm-4 col-form-label">Description</label>
25            <div class="col-sm-8">
26              <textarea class="form-control" th:field="${productDto.description}" th:value="${
27                <p th:if="${#fields.hasErrors('description')}" th:errorclass="text-danger"
```

Lin. 125, col. 1 Espacios 2 UTF-8 CRLF HTML

Conclusión.

Los sistemas web son aplicaciones que se ejecutan en un servidor y se comunican con los clientes a través de Internet. Para desarrollar sistemas web de calidad, se requiere de una buena combinación de tecnologías y herramientas que permitan gestionar la información, la presentación y la funcionalidad de la aplicación.

Una de las tecnologías más importantes para los sistemas web es la base de datos, que es el componente encargado de almacenar, organizar y recuperar los datos que se utilizan en la aplicación. Una base de datos bien diseñada permite optimizar el rendimiento, la seguridad y la integridad de los datos, así como facilitar su acceso y manipulación.

Otra de las tecnologías clave para los sistemas web es el diseño web, que es el proceso de crear la interfaz gráfica y la experiencia de usuario de la aplicación. Un buen diseño web permite mejorar la usabilidad, la accesibilidad y la estética de la aplicación, así como adaptarse a diferentes dispositivos y navegadores.

Finalmente, una de las tecnologías más recomendadas para los sistemas web es la API REST, que es una forma de implementar servicios web que se basan en los principios de la arquitectura REST. Una API REST permite simplificar la comunicación entre el servidor y el cliente, utilizando el protocolo HTTP y el formato JSON o XML para el intercambio de datos. Además, una API REST facilita la interoperabilidad, el escalado, el mantenimiento y la evolución de los servicios web.

En conclusión, utilizar una base de datos con un diseño web y una API REST en el desarrollo de sistemas web es una buena práctica, ya que ofrece múltiples ventajas tanto para los desarrolladores como para los usuarios finales.

Referencias.

Video Tutoría 3.

Bing.

YouTube.