

1. Introducción.....	2
Nombre del Proyecto:.....	2
Descripción del Proyecto:.....	2
2. Requisitos Funcionales.....	3
Usuarios.....	3
Administradores.....	3
3. Arquitectura.....	4
4. Diagrama Entidad-Relación Base de Datos.....	5
5. Estructura de las Clases Entidad.....	6
Acceso:.....	6
Token:.....	7
Usuario:.....	8
Suplemento:.....	9
Carrito:.....	10
Orden:.....	11
RelOrdenCarrito:.....	12
6. Tecnologías Utilizadas.....	13

1.Introducción

Nombre del Proyecto:

Tienda de Suplementos Deportivos

Descripción del Proyecto:

Este proyecto tiene como objetivo desarrollar una aplicación web para la venta de suplementos deportivos. Los usuarios podrán registrarse, gestionar su perfil, agregar productos (suplementos) al carrito, realizar compras y ver sus pedidos. Los administradores tendrán control sobre la gestión de usuarios y productos (suplementos).

2.Requisitos Funcionales

Usuarios

- 2.1 Registro de usuarios
- 2.2 Activación de cuenta mediante enlace de correo electrónico.
- 2.3 Autenticación (inicio de sesión).
- 2.4 Recuperación de contraseña.
- 2.5 Ver todos los suplementos disponibles (poder ver todos y filtrar).
- 2.6 Agregar suplementos al carrito de compras.
- 2.7 Eliminar suplementos del carrito de compras.
- 2.8 Realizar compra del carrito.
- 2.9 Ver todos los pedidos realizados.
- 2.10 Generar PDF de un pedido.
- 2.11 Editar perfil del usuario.
- 2.12 Eliminar cuenta del usuario.

Administradores

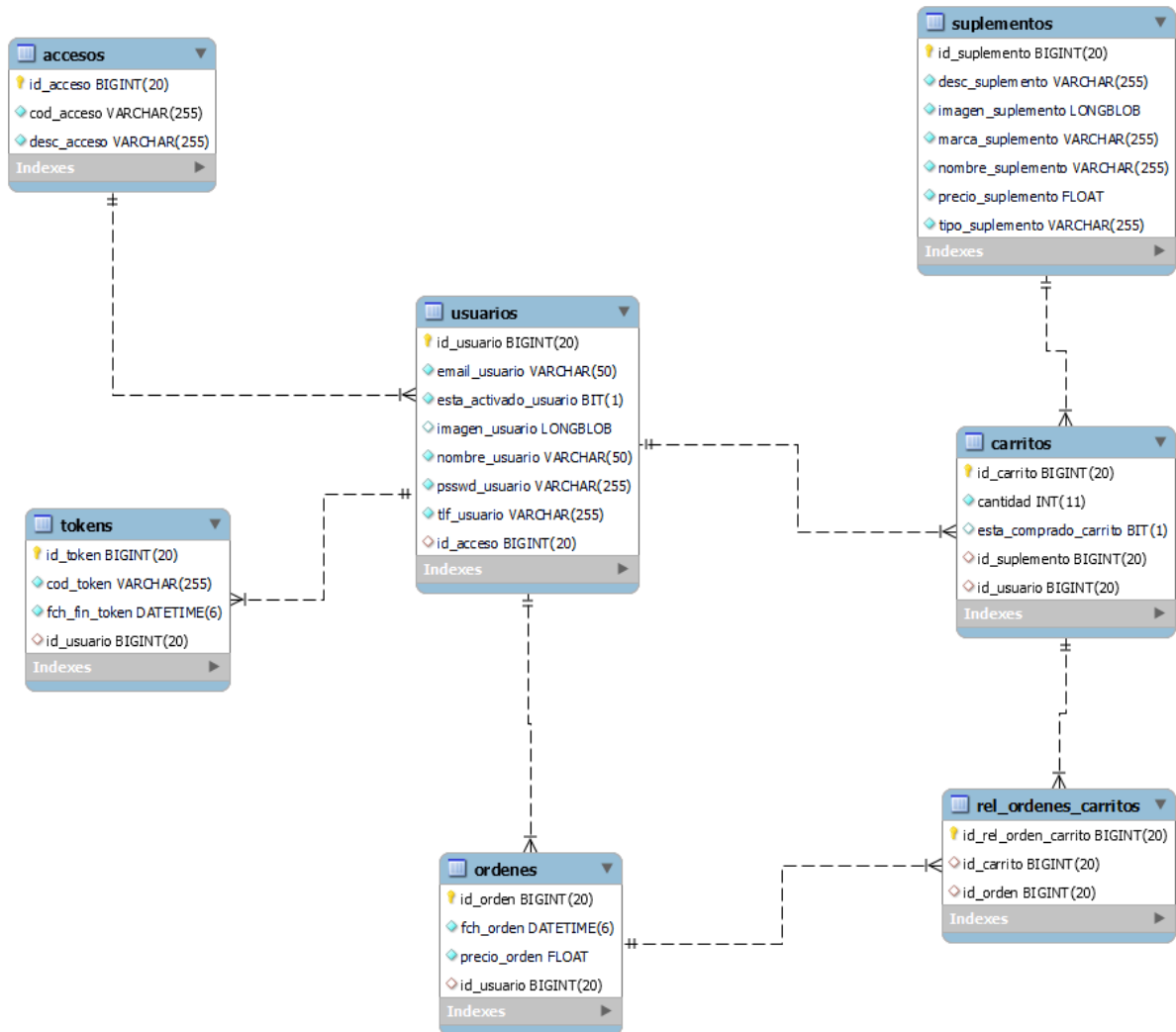
- 2.13 Administrar usuarios (agregar, modificar, eliminar).
- 2.14 Administrar suplementos (agregar, modificar, eliminar).

3.Arquitectura

La aplicación sigue una arquitectura Modelo-Vista-Controlador (MVC):

- Modelo: Maneja datos y lógica de negocio.
- Vista: Se encarga del diseño y presentación.
- Controlador: Se encarga de solicitar datos al modelo y enviarlos a la vista.

4. Diagrama Entidad-Relación Base de Datos.



5. Estructura de las Clases Entidad

Acceso:

```
@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "accesos")
public class Acceso {

    // Atributos

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long idAcceso;
    @Column(name = "cod_acceso", nullable = false)
    private String codAcceso;
    @Column(name = "desc_acceso", nullable = false)
    private String descAcceso;
    @OneToMany(mappedBy = "acceso")
    private List<Usuario> listaUsuarios;

    // Constructores

    // Constructor con todos los parámetros -> Lombok
    // Constructor sin parámetros -> Lombok
    // Constructor sin lista usuario ->
    public Acceso(long idAcceso, String codAcceso, String descAcceso) {
        this.idAcceso = idAcceso;
        this.codAcceso = codAcceso;
        this.descAcceso = descAcceso;
    }
}
```

You, last month • 22/04/2024, login, registro, activar cuenta

Token:

```
@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "tokens")
public class Token {

    // Atributos

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long idToken;
    @Column(name = "cod_token", nullable = false)
    private String codToken;
    @Column(name = "fch_fin_token", nullable = false)
    @Temporal(TemporalType.TIMESTAMP)
    private Calendar fchFinToken;
    @ManyToOne
    @JoinColumn(name = "id_usuario")
    private Usuario usuario;
}
```

Usuario:

```
@Data      You, last month • 22/04/2024, login, registro, activar cuenta, mo...
@Entity
@Table(name = "usuarios")
public class Usuario {

    // Atributos

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long idUsuario;
    @Column(name = "nombre_usuario", nullable = false, length = 50)
    private String nombreUsuario;
    @Column(name = "tlf_usuario", nullable = false)
    private String tlfUsuario;
    @Column(name = "email_usuario", nullable = false, length = 50)
    private String emailUsuario;
    @Column(name = "psswd_usuario", nullable = false)
    private String psswdUsuario;
    @ManyToOne
    @JoinColumn(name = "id_acceso")
    private Acceso acceso;
    @Column(name = "esta_activated_usuario", nullable = false)
    private boolean estaActivadoUsuario;
    @Column(name = "imagen_usuario", nullable = true, columnDefinition = "LONGBLOB")
    private byte[] imagenUsuario;
    @OneToMany(mappedBy = "usuario", cascade = CascadeType.REMOVE)
    private List<Token> listaTokens;
    @OneToMany(mappedBy = "usuario", cascade = CascadeType.REMOVE)
    private List<Carrito> listaCarrito;
    @OneToMany(mappedBy = "usuario")
    private List<Orden> listaOrden;
}
```


Suplemento:

```
@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "suplementos")
public class Suplemento {

    // Atributos

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long idSuplemento;

    @Column(name = "nombre_suplemento", nullable = false)
    private String nombreSuplemento;

    @Column(name = "desc_suplemento", nullable = false)
    private String descSuplemento;

    @Column(name = "precio_suplemento", nullable = false)
    private float precioSuplemento;

    @Column(name = "tipo_suplemento", nullable = false)
    private String tipoSuplemento;

    @Column(name = "marca_suplemento", nullable = false)
    private String marcaSuplemento;

    @Column(name = "imagen_suplemento", nullable = false, columnDefinition = "LONGBLOB")
    private byte[] imagenSuplemento;

    @OneToMany(mappedBy = "suplemento", cascade = CascadeType.REMOVE)
    private List<Carrito> listaCarrito;
}
```

Carrito:

```
@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "carritos")
public class Carrito {

    // Atributos

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long idCarrito;

    @ManyToOne
    @JoinColumn(name = "id_usuario")
    private Usuario usuario;

    @ManyToOne
    @JoinColumn(name = "id_suplemento")
    private Suplemento suplemento;

    @Column(name = "cantidad", nullable = false)
    private int cantidad;

    @Column(name = "esta_comprado_carrito")
    private boolean estaCompradoCarrito;

    @OneToMany(mappedBy = "carrito")
    private List<RelOrdenCarrito> listaRelacion;
}
```

Orden:

```
@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "ordenes")
public class Orden {

    // Atributos

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long idOrden;

    @Column(name = "precio_orden", nullable = false)
    private float precioOrden;

    @Column(name = "fch_orden", nullable = false)
    @Temporal(TemporalType.TIMESTAMP)
    private Calendar fchOrden;

    @ManyToOne
    @JoinColumn(name = "id_usuario")
    private Usuario usuario;

    @OneToMany(mappedBy = "orden")
    private List<RelOrdenCarrito> listaRelacion;
}
```

RelOrdenCarrito:

```
@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "rel_ordenes_carritos")
public class RelOrdenCarrito {

    // Atributos

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long idRelOrdenCarrito;

    @ManyToOne
    @JoinColumn(name = "id_orden")
    private Orden orden;

    @ManyToOne
    @JoinColumn(name = "id_carrito")
    private Carrito carrito;
}
```

6. Tecnologías Utilizadas

- **Backend:**
 - Java 17
 - Spring Boot 3.2.5
 - Spring Data JPA
 - Spring Security
- **Frontend:**
 - HTML
 - CSS
 - JavaScript
 - Thymeleaf
 - AJAX
 - Bootstrap
- **Base de Datos:**
 - MariaDB 11.3.2
- **Servidor:**
 - Nginx
 - Tomcat 10.1.23
 - JDK 17.0.11
- **Herramientas de Desarrollo:**
 - IDE: Eclipse / Visual Studio Code.
 - Control de versiones: Git
 - Build Tool: Maven