

Cheatsheet – Proyectoavance (Sistema web de zapatillas)

Estructura del proyecto

Carpeta	Contenido
src/main/java/com/example/proyectoavance	ProyectoavanceApplication.java (Main), ControlPaginas.java (Controladores)
src/main/resources/static	css/Login.css, img/ (imágenes)

Rutas – Usuario normal

Ruta	Vista	Descripción
/principal	principal.html	Página principal
/Login	login.html	Formulario de acceso
/registro	registro.html	Registro de usuario
/paginaproductos	paginaproductos.html	Catálogo de productos
/paginaofertas	paginaofertas.html	Sección de ofertas
/paginacontacto	paginacontacto.html	Página de contacto
/carritocompras	carritocompras.html	Carrito de compras

Rutas – Administrador

Ruta	Vista	Descripción
/dashboard	dashboard.html	Panel principal del administrador
/productos	productos.html	Gestión de productos
/usuario	usuario.html	Gestión de usuarios
/pedido	pedido.html	Gestión de pedidos
/reportes	reportes.html	Reportes y estadísticas
/login	login.html	Registro de usuario

Java / Spring Boot

Clase Principal

```
@SpringBootApplication
public class ProyectoavanceApplication {
    public static void main(String[] args) {

        SpringApplication.run(ProyectoavanceApplication.class,
            args);
    }
}
```

Nota:

@SpringBootApplication → Clase principal que arranca Spring Boot

Controlador de Rutas

```
@Controller
public class ControlPaginas {

    @GetMapping("/login")
    public String login() {
        return "login";
    }

    @GetMapping("/dashboard")
    public String dashboard(Model model) {
        model.addAttribute("mensaje", "Bienvenido al
Panel de Administración");
        return "dashboard";
    }

    @PostMapping("/procesarLogin")
    public String procesarLogin(@RequestParam("email")
String email,
    @RequestParam("password") String password,
                                Model model) {
        if("admin@gmail.com".equals(email) &&
"123".equals(password)) {
            model.addAttribute("usuario", email);
            return "redirect:/dashboard";
        } else {
            model.addAttribute("usuario", email);
            return "principal";
        }
    }
}
```

Nota:

@Controller → Maneja vistas HTML

@GetMapping("/ruta") → Ruta GET

@PostMapping("/ruta") → Ruta POST

Model → Para enviar datos a la vista

redirect:/ruta → Redirección a otra ruta

Html / Bootstrap

```
<!DOCTYPE html>
<html lang="es">
```

```

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width,
initial-scale=1.0">

  <title>Tienda Zapatillas</title>

  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/
css/bootstrap.min.css" rel="stylesheet">

</head>

<body>


  <!-- Navbar -->

  <nav class="navbar navbar-expand-lg">

    <a class="navbar-brand" href="/">Jzapatillas</a>

    <ul class="navbar-nav">

      <li><a class="nav-link" href="/">Inicio</a></li>

      <li><a class="nav-link"
href="paginaproductos">Productos</a></li>

      <li><a class="nav-link"
href="paginaofertas">Ofertas</a></li>

      <li><a class="nav-link"
href="paginacontacto">Contacto</a></li>

    </ul>

  </nav>


  <!-- Contenedores para tarjetas dinámicas -->

  <div class="row" id="productos-container"></div>

  <div class="row" id="ofertas-container"></div>


</body>
</html>

```

Nota:

class="navbar-nav" → Menú de navegación con Bootstrap

class="row" → Contenedor de tarjetas o secciones

id="productos-container" → Donde se insertan tarjetas dinámicas con JS

Java Script

```

const productos = [
  { nombre: "Adidas forum", precio: 350, img:
"/img/adidas.png", imgHover: "/img/adidasrosas.png" }
];

function generarTarjetas(lista, contenedorId) {

```

```

    const contenedor =
document.getElementById(contenedorId);

    contenedor.innerHTML = lista.map(p => `

        <div class="col-6 col-md-3 mb-4">

            <div class="card">

                

                <h5>${p.nombre}</h5>

                <p>S/${p.precio}</p>

            </div>

        </div>

    `).join("");
}

generarTarjetas(productos, "productos-container");

```

Nota:

map() → Genera HTML dinámico desde listas

onmouseenter/onmouseleave → Cambiar imagen al pasar el mouse

document.getElementById("id") → Selecciona contenedor donde insertar tarjetas

Cascading Style Sheets (CSS)

Navbar y enlaces

```

.navbar { background-color: #f9f9f9 !important; }

.nav-link {
    position: relative;
    color: #333 !important;
    transition: color 0.3s;
}

.nav-link::after {
    content: "";
    position: absolute;
    bottom: 0;
    left: 50%;
    transform: translateX(-50%);
    width: 0;
    height: 2px;
    background: #000;
    transition: width 0.3s;
}

.nav-link:hover::after { width: 100%; }

```

```
.navbar-brand { font-weight: bold; font-size: 1.3rem; }
```

Cartas de productos (cards)

```
.card { overflow: hidden; }  
.card-img-top { transition: transform 0.5s ease-in-out; }  
.card:hover .card-img-top { transform: scale(1.1); }
```

Secciones y responsividad

```
section { padding: 60px 20px; }  
#productos { background: #ffffff; }  
#ofertas { background: #f8f9fa; }  
#contacto { background: #e9ecef; }  
  
@media (max-width: 991px) {  
  .navbar-collapse { text-align: center; }  
  .nav-link::after { width: 20%; }  
}
```

Notas rápidas de CSS:

.card:hover .card-img-top → Zoom suave al pasar mouse sobre la imagen

@media (max-width: 991px) → Estilos responsive para pantallas pequeñas

.nav-link::after → Subrayado animado al pasar el mouse

section { padding: ... } → Separación uniforme entre secciones

Diagramas de Flujo (texto)

Flujo Usuario

Usuario → Login → Principal → [Ver Productos | Ver Ofertas | Ver Contacto] → Carrito

Flujo Administrador

Admin → Login → Dashboard → [Gestionar Productos | Gestionar Usuarios | Gestionar Pedidos | Ver Reportes]