

Universidad de San Carlos de Guatemala

Centro Universitario de Occidente

División de Ciencias de la Ingeniería

Área Profesional

Manejo e Implementación de Archivos

Sección "A"

Ing. Bryan Misael Monzón Fuentes

Aux. Luis Cifuentes



“MANUAL TECNICO”

Melvin Eduardo Ordoñez Sapón RA | 202230552

Quetzaltenango, octubre del 2025

“Id y enseñad a todos”

ACERCA DEL PROGRAMA:

IDE Utilizado: IntelliJ, Visual Studio Code

Programado en Sistema Operativo: Ubuntu 24.04 LTS

Terminal Utilizada: Terminal de Linux

Compatibilidad: Windows, Linux, macOS

Lenguaje de Programacion: Java

Base de datos: Postgresql 16.10

Versión de Spring: 3.5.6

Versión de Angular: 20.1.0

JDK: 21.0.8

Versión del Programa: 1.0

CONFIGURACIONES IMPORTANTES

METODO DE CONFIGURACION

App

spring.application.name=ecommerce_api

server.port=8080

PostgreSQL

spring.datasource.url=jdbc:postgresql://localhost:5432/ecommerce_gt

spring.datasource.username=postgres

spring.datasource.password=123

spring.datasource.driver-class-name=org.postgresql.Driver

JPA / Hibernate

spring.jpa.hibernate.ddl-auto=update

spring.jpa.open-in-view=false

spring.jpa.properties.hibernate.format_sql=true

spring.jpa.show-sql=false

jwt.secret=ebTPQz90e9f0oW20KpuHDyMDVn6bm0YFBZ78r3By7NSSt3hkIG3yoQgT8AFXDSOm

jwt.expiration-ms=3600000

jwt.refresh-expiration-ms=1209600000

Límite de archivos

spring.servlet.multipart.max-file-size=5MB

spring.servlet.multipart.max-request-size=5MB

Carpeta donde se guardarán (relativa al jar)

app.upload-dir=uploads

server.forward-headers-strategy=framework

CLASE DE CONFIGURACION PARA CORS Y DESPLIEGUE

```
package ecommerce_api.ecommerce_api.config;

import ecommerce_api.ecommerce_api.security.JwtAuthFilter;
import lombok.RequiredArgsConstructor;
import org.springframework.context.annotation.*;
import org.springframework.http.HttpMethod;
import org.springframework.security.config.Customizer;
import org.springframework.security.config.annotation.method.configuration.EnableMethodSecurity;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.http.SessionCreationPolicy;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.security.web.*;
import org.springframework.security.web.authentication.UsernamePasswordAuthenticationFilter;
import org.springframework.web.cors.*;
import org.springframework.web.filter.CorsFilter;

import java.util.List;

@Configuration@EnableMethodSecurity@RequiredArgsConstructorpublic class
SecurityConfig {
    private final JwtAuthFilter jwtAuthFilter;

    @Bean    public SecurityFilterChain filterChain(HttpSecurity http) throws
Exception {
    http
```

```

        .csrf(csrf -> csrf.disable())
        .sessionManagement(sm ->
sm.sessionCreationPolicy(SessionCreationPolicy.STATELESS))
        .cors(Customizer.withDefaults()) // habilita CORS con la configuración
del bean corsConfigurationSource        .authorizeHttpRequests(auth -> auth
        // preflight siempre
permitido        .requestMatchers(HttpMethod.OPTIONS, "**").permitAll()

        // público        .requestMatchers(HttpMethod.POST,
"/api/auth/register", "/api/auth/login").permitAll()
        .requestMatchers(HttpMethod.GET, "/api/public/**").permitAll()
        .requestMatchers(HttpMethod.GET, "/files/**").permitAll()

        // protegido por
rol        .requestMatchers("/api/comun/**").hasRole("COMUN")
        .requestMatchers("/api/moderador/**").hasAnyRole("MODERADOR
", "ADMIN")
        .requestMatchers("/api/logistica/**").hasAnyRole("LOGISTICA", "AD
MIN")
        .requestMatchers("/api/admin/**").hasRole("ADMIN")

        .anyRequest().authenticated()
    )
    .addFilterBefore(jwtAuthFilter,
UsernamePasswordAuthenticationFilter.class);

    return http.build();
}

@Bean public CorsConfigurationSource corsConfigurationSource() {
    CorsConfiguration cfg = new CorsConfiguration();
    // Para despliegue en Netlify, ngrok, localhost:4200 (Angular dev server)
cfg.setAllowedOrigins(List.of(
        "https://ecommercemv.netlify.app",
        "http://localhost:4200",
        "https://hypercatalectic-donnetta-tubulously.ngrok-free.dev"    ));

cfg.setAllowedMethods(List.of("GET", "POST", "PUT", "PATCH", "DELETE", "OPTION
S"));
    cfg.setAllowedHeaders(List.of(
        "Authorization", "Content-Type", "Accept", "Origin",
        "X-Requested-With", "ngrok-skip-browser-warning"    ));
    cfg.setExposedHeaders(List.of("Authorization", "Location"));
    cfg.setAllowCredentials(true);
    cfg.setMaxAge(3600L);

    UrlBasedCorsConfigurationSource source = new

```

```

UrlBasedCorsConfigurationSource();
    source.registerCorsConfiguration("/**", cfg);
    return source;
}

// @Bean// public CorsFilter corsFilter(CorsConfigurationSource source) {
return new CorsFilter((UrlBasedCorsConfigurationSource) source); // } @Bean
public PasswordEncoder passwordEncoder() { return new
BCryptPasswordEncoder(); }
}

```

ORGANIZACION DEL PROYECTO PARA IDENTIFICAR UN ARCHIVO FACIL

