MANUAL TÉCNICO BACKEND

GESCOMPH

Brayan Santiago Guerrero Mendez – Jesús David Fierro Rivera

Contenido

1 Introducción

El presente Manual Técnico documenta los aspectos técnicos, arquitectónicos y de implementación del componente servidor (Backend) de la plataforma web "GESCOMPH". Su finalidad es optimizar la gestión de contratos, plazas, establecimientos y obligaciones municipales, proporcionando los servicios (APIs) necesarios para la aplicación web y móvil.

El backend es el núcleo central que garantiza la trazabilidad, el análisis inteligente y la seguridad de los datos, proporcionando los servicios necesarios para la gestión municipal eficiente.

El backend de GESCOMPH ha sido desarrollado en C#, bajo el framework .NET 8.0, siguiendo una arquitectura en N capas que promueve la separación de responsabilidades y escalabilidad del sistema. Dicha arquitectura está conformada por los siguientes capas o proyectos dentro de la solución:

- **Entity**: Define las entidades de dominio, modelos de configuración y DTOs compartidos.
- **Data**: Implementa el acceso a datos mediante los patrones Repository y Unit of Work, además de las configuraciones de persistencia con Entity Framework y el soporte para múltiples motores de base de datos (SQL Server, PostgreSQL o MySQL).
- **Business**: Contiene la lógica de negocio principal, aplicando principios SOLID y P.O.O., junto con patrones de diseño como Repository para el manejo modular de los servicios.
- **Utilities**: Agrupa funcionalidades transversales como utilidades JWT, manejo de excepciones, servicios de almacenamiento en Cloudinary, integración con MercadoPago para pagos, y servicios de notificaciones.
- **WebGESCOMPH**: Corresponde al proyecto ASP.NET Core API, responsable de exponer los servicios que interactúan con los clientes web y móviles.

La aplicación se encuentra contenerizada para su despliegue mediante Docker Compose, donde se orquesta el servicio principal de la API. Adicionalmente, el sistema incluye soporte para servicios de pagos (MercadoPago), almacenamiento de imágenes (Cloudinary), notificaciones en tiempo real (SignalR) y trabajos en segundo plano (Hangfire).

2 Objetivo del documento

El propósito de este documento es describir de manera detallada el funcionamiento técnico, la arquitectura, las configuraciones, dependencias, despliegue y mantenimiento del backend del sistema GESCOMPH. Este manual está dirigido a desarrolladores, ingenieros de soporte, personal de infraestructura y cualquier integrante técnico que participe en la implementación o mantenimiento del sistema.

Público Objetivo: Desarrolladores, analistas, diseñadores y personal técnico encargado de la implementación y mantenimiento del sistema.

Finalidad: Garantizar la comprensión y correcta implementación del entorno servidor, asegurando la adecuada comunicación entre los diferentes componentes del sistema (backend, frontend Web, frontend móvil y bases de datos). Asimismo, busca facilitar la replicación, despliegue y personalización de la solución GESCOMPH en diferentes entornos municipales, manteniendo su integridad funcional y compatibilidad con los requerimientos definidos.

Permite:

- Proporcionar las instrucciones para la instalación, configuración y ejecución del backend.
- Documentar la estructura interna del sistema, sus dependencias y servicios principales.
- Servir de guía de soporte para futuras actualizaciones, correcciones y despliegues.

3 Alcance del sistema

El sistema GESCOMPH tiene como alcance el desarrollo y funcionamiento de una plataforma digital de gestión municipal para el manejo de contratos, plazas comerciales, establecimientos y obligaciones mensuales. El backend de la aplicación construye el núcleo funcional encargado de procesar, administrar y centralizar la información, garantizando la comunicación eficiente entre los distintos módulos y usuarios del sistema.

Dentro de su alcance, el sistema permite:

- **Módulo de Seguridad y Autenticación**: Manejo de roles, permisos y control de acceso a los distintos módulos del sistema, incluyendo autenticación JWT y códigos de dos factores.
- **Gestión de Negocios**: Registro y administración de contratos, cláusulas, plazas, establecimientos, citas y obligaciones mensuales.
- **Gestión de Personas**: Manejo de información de personas físicas involucradas en los procesos municipales.
- **Gestión de Ubicaciones**: Administración de ciudades y departamentos para

localización geográfica.

- **Sistema de Administración**: Configuración de formularios, módulos, parámetros del sistema y notificaciones.
- **Utilidades**: Servicios de imágenes (Cloudinary), pagos (MercadoPago), PDFs, notificaciones en tiempo real y trabajos en segundo plano.

El backend está diseñado para soportar estos procesos mediante una arquitectura N-Capas, con una base tecnológica en .NET 8.0, Entity Framework Core, SQL Server/PostgreSQL/MySQL.

4 Descripción de la arquitectura

La arquitectura general de GESCOMPH se basa en un enfoque modular y multicapa (N-capas), diseñada para garantizar la escalabilidad, mantenibilidad y separación de responsabilidades entre los distintos componentes del sistema. El sistema está conformado por tres pilares principales: Backend, Frontend y Base de datos, los cuales se comunican mediante servicios web RESTful.

4.1 Arquitectura general

El sistema se estructura bajo una arquitectura distribuida conformada por los siguientes contenedores:

- **API Backend (ASP.NET Core)**: Contiene la lógica de negocio, controladores y servicios de comunicación con la base de datos.
- **Servidor de Base de Datos (SQL Server/PostgreSQL/MySQL)**: Gestiona la persistencia y consulta de datos.

4.2 Estructura de la solución

Nombre de la solución: GESCOMPH

Capa/Proyecto	Descripción
Entity	Contiene las entidades de dominio, modelos de configuración, enumeraciones y DTOs. Define la estructura de los datos compartidos entre capas. Incluye carpetas como Domain/Models, DTOs, etc.
Data	Implementa el acceso a datos mediante los patrones Repository y Unit of Work. Contiene los contratos, las implementaciones y la carpeta Infrastructure con la configuración de Entity Framework.
Business	Encapsula la lógica de negocio. Integra los módulos funcionales como SecurityAuthentication, Business, Location, Persons, AdministrationSystem, Utilities. Aplica el patrón Repository y principios SOLID.
Utilities	Proporciona herramientas transversales al sistema: gestión de

Capa/Proyecto	Descripción
	autenticación JWT, manejo de imágenes (Cloudinary), integración con MercadoPago, generación de PDFs, mapeadores (Mapster) y validaciones (FluentValidation).
WebGESCOMPH	Proyecto ASP.NET Core API que expone los endpoints RESTful. Contiene los controladores, Extensions para registrar servicios, configuraciones (Program.cs, appsettings.json) y el Dockerfile para la construcción del contenedor de la API.

4.3 Patrones y principios aplicados

Patrones de diseño: Repository, Unit of Work, Factory.

Patrones Creacionales: Dependency Injection.

Principios de desarrollo: SOLID, POO, separación de responsabilidades, inyección de dependencias.

Servicios integrados: Envío de correos electrónicos, almacenamiento de imágenes en Cloudinary, pagos con MercadoPago, autenticación basada en tokens JWT, notificaciones en tiempo real con SignalR, trabajos en segundo plano con Hangfire.

5 Público objetivo del manual

El presente manual técnico está dirigido principalmente al personal técnico encargado de la instalación, configuración, mantenimiento y evolución del backend de la plataforma GESCOMPH. Su contenido está orientado a profesionales con conocimientos en desarrollo de software, administración de sistemas y despliegue de aplicaciones en entornos de servidor o contenedores. De manera específica, este documento está destinado a los siguientes perfiles:

- **Desarrolladores Backend**: Responsables de implementar nuevas funcionalidades, mantener la lógica de negocio y garantizar la correcta interacción entre las capas del sistema.
- **Ingenieros de DevOps o Administradores de Sistemas**: Encargados del despliegue, configuración y supervisión del entorno de ejecución del backend, incluyendo la orquestación con Docker.
- **Integradores o Desarrolladores Full Stack**: Encargados de conectar el backend con los clientes web y móviles, consumiendo los endpoints expuestos por la API RESTful.
- **Técnicos de soporte y mantenimiento**: Responsables de monitorear la operación del sistema, aplicar actualizaciones y resolver incidencias relacionadas con la infraestructura o los servicios del backend.
- **Estudiantes o equipos académicos**: Que deseen comprender la estructura técnica

y arquitectónica del proyecto con fines de estudio, documentación o mejora continua.

El nivel de conocimiento esperado para la correcta comprensión y aplicación de este manual incluye:

- Fundamentos de Programación Orientada a Objetos (P.O.O.) y principios SOLID.
- Experiencia básica o intermedia en C# y .NET Core/8.0.
- Conocimientos de Entity Framework Core, RESTful APIs, y bases de datos relacionales.
- Familiaridad con Docker y conceptos básicos de orquestación de servicios.
- Conocimientos generales sobre configuración de entornos, manejo de variables de entorno y servicios de red.

6 Convenciones y nomenclaturas utilizadas

Para mantener la coherencia, legibilidad y estandarización del código dentro del proyecto GESCOMPH, se han establecido una serie de convenciones y reglas de nomenclatura aplicadas de manera uniforme en todas las capas del backend. Estas convenciones siguen las buenas prácticas recomendadas por la comunidad de desarrollo de .NET y los lineamientos de Microsoft C# Coding Standards.

Estilo general de codificación

- El código fuente se desarrolla íntegramente en C#.
- Se aplican los principios de Programación Orientada a Objetos (P.O.O.) y SOLID.
- Todo el código y los nombres de identificadores (clases, métodos, variables, propiedades, etc.) se escriben en inglés, garantizando consistencia y comprensión internacional del proyecto.
- Los nombres deben ser descriptivos y auto explicativos, evitando abreviaturas ambiguas.
- Se recomienda mantener una estructura clara con sangrías de 4 espacios y comentarios XML o de bloque cuando sea necesario documentar el comportamiento de un método o clase.

Convenciones de nomenclaturas

Elemento	Convención	Ejemplo
Clases	PascalCase	UserManager, ContractService, PlazaController
Interfaces	Prefijo I + PascalCase	IUserRepository, IContractService
Métodos	PascalCase	GetUserById(), CreateContract()
Propiedades	PascalCase	userList, contractItem, dbContext
Variables locales	camelCase	userId, contractRequest, emailAddress
Parámetros de	camelCase	userId, contractRequest,

Elemento	Convención	Ejemplo
métodos		emailAddress
Constantes	UPPER_CASE con guiones bajos	MAX_RETRY_COUNT, DEFAULT_PAGE_SIZE
Enumeraciones (Enums)	PascalCase para el tipo y valores	UserRole.Admin, ContractStatus.Active
Nombres de archivos	Igual que la clase principal contenida	UserService.cs, ContractController.cs
Namespaces	PascalCase según estructura de carpetas	GESCOMPH.Entity.Domain
Migraciones EF	Prefijo con fecha y descripción corta	20240101_InitialCreate

7 Requisitos del sistema

El correcto funcionamiento del backend de GESCOMPH depende de una configuración adecuada del entorno de ejecución, tanto a nivel de hardware, software como de conectividad de red. Esta sección define los requisitos mínimos y recomendados necesarios para garantizar un desempeño estable, seguro y eficiente del sistema en entornos de desarrollo, pruebas o producción.

8 Requisitos de hardware

Recurso	Mínimo requerido	Recomendado (producción)
Procesador (CPU)	Intel Core i3 / AMD Ryzen 3 (2 núcleos, 2.4 GHz)	Intel Core i5 / Ryzen 5 o superior (4 núcleos, 3.0 GHz)
Memoria RAM	4 GB	8 GB o más
Almacenamiento	10 GB libres en disco HDD	20 GB en SSD
Tipo de almacenamiento	HDD	SSD (preferible para rendimiento)
Resolución de pantalla (entornos de desarrollo)	1366x768	1920x1080
Conectividad	Internet estable > 5 Mbps	Banda ancha ≥ 10 Mbps

9 Requisitos de software

Componente	Versión mínima	Recomendada / soporte	Descripción
Sistema Operativo	Windows 10 / Ubuntu 20.04 LTS	Windows 11 / Ubuntu 22.04 LTS	Entornos de desarrollo y despliegue soportados
.NET SDK / Runtime	.NET 8.0	.NET 8.0 (LTS)	Framework principal del backend
Motor de base de datos	SQL Server 2019 / PostgreSQL 12 / MySQL 8	SQL Server 2022 / PostgreSQL 15	Base de datos relacional compatible con Entity Framework Core
Docker	v20.0	v24.0 o superior	Ejecución de contenedores
Visual Studio / VS Code	2022 / Última versión estable	2022 con extensión C# y Docker	IDE de desarrollo y depuración
Git	2.30	Última versión estable	Control de versiones

9.1 Requisitos de red y conectividad

- **Puertos expuestos (API)**: 5100-5103 (dependiendo del entorno), 8080 interno
- **Firewall**: Permitir conexiones salientes para servicios externos (Cloudinary, MercadoPago)
- **Protocolo de comunicación**: HTTP/HTTPS con formato JSON
- **Requisitos externos**: Acceso a servicios de MercadoPago, Cloudinary, y posibles servicios de correo

10 Versiones y dependencias

El backend de GESCOMPH utiliza diversas dependencias y paquetes de NuGet para su correcto funcionamiento.

Dependencia / Paquete	Versión recomendada	Uso principal
Microsoft.EntityFrameworkCore	9.0.x	ORM para acceso y manipulación de datos
Microsoft.EntityFrameworkCore.SqlServer	9.0.x	Soporte para SQL Server
Npgsql.EntityFrameworkCore.PostgreSQL	9.0.x	Soporte para PostgreSQL

Dependencia / Paquete	Versión recomendada	Uso principal
Pomelo.EntityFrameworkCore.MySql	9.0.x	Soporte para MySQL
Microsoft.AspNetCore.Authentication.JwtBearer	8.0.x	Autenticación basada en JWT
Mapster	7.4.x	Mapeo entre entidades y DTOs
MediatR	13.0.x	Implementación de CQRS
FluentValidation	12.0.x	Validación de modelos
Swashbuckle.AspNetCore	6.6.x	Generación de documentación Swagger/OpenAPI
CloudinaryDotNet	1.27.x	Almacenamiento de imágenes
Microsoft.AspNetCore.SignalR	Incluido en .NET 8	Notificaciones en tiempo real
Hangfire.AspNetCore	1.8.x	Trabajos en segundo plano
Syncfusion.DocIO.Net.Core	30.1.x	Generación de documentos Word/PDF

Estas dependencias garantizan la modularidad, seguridad y extensibilidad del backend, facilitando su mantenimiento y futuras actualizaciones.

11 Verificar la estructura del proyecto

Dentro del directorio raíz del backend deben encontrarse los siguientes archivos y carpetas principales:

- GESCOMPH.sln (archivo de solución)
- Entity/ (proyecto de entidades)
- Data/ (proyecto de acceso a datos)
- Business/ (proyecto de lógica de negocio)
- Utilities/ (proyecto de utilidades)
- WebGESCOMPH/ (proyecto API)
- DevOps/ (configuraciones de despliegue por entorno)
- .env (variables de entorno base)

12 Configuración de variables de entorno

El sistema utiliza variables de entorno para configuración sensible. Archivo .env de ejemplo:

```
# Database

  "ConnectionStrings": {
    "SqlServer": "Server=localhost,1438;Database=gescomph_develop;User
Id=sa;Password=Admin123!;Encrypt=False;TrustServerCertificate=True;MultipleActive
ResultSets=True;Connection Timeout=30;"
  }

# JWT

  "Jwt": {
    "key": "E131C348-50CA-4B5B-A481-029185EB9DD7",
    "Issuer": "WebGESCOMPH",
    "Audience": "GESCOMPH",
    "AccessTokenExpirationMinutes": 60,
    "RefreshTokenExpirationDays": 7
  }

# Cloudinary

  "Cloudinary": {
    "URL":
"CLOUDINARY_URL=cloudinary://774511516845582:TE_y4UF_QOZYr4nQ2TonvlroWvU@dmbndpjlh",
    "CloudName": "dmbndpjlh",
    "ApiKey": "774511516845582",
    "ApiSecret": "TE_y4UF_QOZYr4nQ2TonvlroWvU"
  }

# MercadoPago

"AccessToken": "APP_USR-446464791921270-111915-
fef53e759ed3af3116cf3477e91b1cbe-3003444818",


# Email (si aplica)

 "CONFIGURACIONES_EMAIL": {
    "PROVIDER": "smtp",
    "EMAIL": "Brayansantiagoguerreromendez@gmail.com",
    "PASSWORD": "eavp pyja xkzz lucs",
    "HOST": "smtp.gmail.com",
    "PUERTO": 587
  }
```

13 Anexos

- **Repositorio del proyecto**: [gescomph-api]
- **Documentación API**: Disponible en /swagger una vez ejecutado el proyecto
- **Diagramas de arquitectura**: