

Especificación patrón arquitectura.

Administrador MVC y Servicios WebAPI.

Prueba Técnica TecnoFactory.

Edwin Giovanny Castillo Ruiz

3123095767

Edwin.castillo012@gmail.com

27 de julio de 2024

1. Introducción

1.1. Propósito

Este documento proporciona una descripción detallada del patrón de arquitectura utilizado en el desarrollo de la prueba técnica basada en .NET Web API, DDD y .NET MVC.

1.2. Alcance

Cubre la estructura del sistema, los patrones de diseño utilizados y cómo se implementan en el contexto de .NET Web API y .NET MVC.

1.3. Definiciones, Acrónimos y Abreviaturas

1.3.1.1. **DDD:** Domain-Driven Design

1.3.1.2. **MVC:** Model-View-Controller

1.3.1.3. **Web API:** Application Programming Interface para servicios web

2. Patrones de Arquitectura

2.1. Patrones de Diseño

2.1.1. Patrón Principal: DDD con .NET Web API y .NET MVC

- **Descripción:**
 - **DDD:** Enfoque en el diseño del dominio y la separación de responsabilidades.
 - **.NET Web API:** Exposición de servicios RESTful para interactuar con el sistema.
 - **.NET MVC:** Implementación de la capa de presentación para aplicaciones web.
- **Justificación:** Este patrón proporciona una separación clara entre el dominio, la lógica de negocio y la interfaz de usuario, facilitando la escalabilidad y el mantenimiento del sistema.

3. Estructura del Sistema

Componentes Principales:

- **Capa de Dominio:** Contiene las entidades, agregados, repositorios y servicios del dominio.

- **Capa de Aplicación:** Coordina las operaciones entre el dominio y la interfaz de usuario. Incluye los casos de uso y servicios de aplicación.
- **Capa de Infraestructura:** Implementa la persistencia de datos, el acceso a servicios externos y otras funcionalidades de infraestructura.
- **Capa de Presentación (MVC):** Contiene los controladores y vistas para manejar las solicitudes del usuario y presentar los datos.

3.1. Arquitectura de Capas

- **Capa de Presentación (MVC):**
 - **Controladores:** Manejan las solicitudes HTTP y devuelven las respuestas.
 - **Vistas:** Renderizan la interfaz de usuario.
 - **Modelos:** Representan los datos que se muestran en las vistas.
- **Capa de Aplicación:**
 - **Casos de Uso:** Lógica específica de la aplicación y coordinación de operaciones.
 - **Servicios de Aplicación:** Interfaces para la interacción con la capa de dominio.
- **Capa de Dominio:**
 - **Entidades y Agregados:** Modelos que representan conceptos del dominio.
 - **Repositorios:** Interfaces para acceder y manipular entidades.
 - **Servicios del Dominio:** Lógica de negocio que no encaja en las entidades.
- **Capa de Infraestructura:**
 - **Persistencia de Datos:** Implementación de repositorios y acceso a bases de datos.
 - **Servicios Externos:** Integraciones con la Api de Marvel.

4. Tecnologías y Herramientas

4.1. Tecnologías Usadas

- **.NET Core:** Framework de desarrollo para .NET Web API.

- **Entity Framework Core:** ORM para la persistencia de datos.
- **ASP.NET MVC:** Para la implementación de la capa de presentación.

4.2. Herramientas de Desarrollo

- **Visual Studio:** IDE para el desarrollo en .NET.
- **Postman:** Para probar los endpoints de la Web API.
- **Git:** Para el control de versiones.