

Informe técnico: Playground

Autores:

Laura Martir Beltrán C311

Adrián Hernández Castellanos C312

Yesenia Valdés Rodríguez C311

Breve introducción:

El presente constituye un informe técnico orientado a desarrolladores encargados del mantenimiento del software al que este informe representa. A continuación se muestra un índice de contenidos aquí presentes.

Diccionario de Datos:

A continuación se muestra la composición del diccionario de datos utilizados:

Activity:

#	Nombre	Tipo de datos	Permitir NULL	Predeterminado
1	Id	UUID	<input type="checkbox"/>	Sin valor predete...
2	Name	TEXT	<input type="checkbox"/>	Sin valor predeter...
3	Description	TEXT	<input type="checkbox"/>	Sin valor predeter...
4	EducatorId	TEXT	<input checked="" type="checkbox"/>	NULL
5	Type	TEXT	<input type="checkbox"/>	Sin valor predeter...
6	Recommende...	INTEGER	<input type="checkbox"/>	Sin valor predeter...
7	ItsPrivate	BOOLEAN	<input type="checkbox"/>	Sin valor predeter...
8	FacilityId	UUID	<input type="checkbox"/>	Sin valor predeter...
9	CreatedAt	TIMESTAMPZ	<input type="checkbox"/>	Sin valor predeter...
10	UpdateAt	TIMESTAMPZ	<input type="checkbox"/>	Sin valor predeter...
11	DeletedAt	TIMESTAMPZ	<input checked="" type="checkbox"/>	NULL

Figura 1: Actividades

Almacena la información de las actividades programadas (nombre, descripción, edad recomendada, etc.). Es fundamental para conocer el catálogo de actividades del parque.

ActivityDate:

#	Nombre	Tipo de datos	Permitir NULL	Predeterminado
1	Id	UUID	<input type="checkbox"/>	Sin valor predete...
2	ActivityId	UUID	<input type="checkbox"/>	Sin valor predeter...
3	DateTime	TIMESTAMPZ	<input type="checkbox"/>	Sin valor predeter...
4	ReservedPlaces	INTEGER	<input type="checkbox"/>	Sin valor predeter...
5	CreatedAt	TIMESTAMPZ	<input type="checkbox"/>	Sin valor predeter...
6	UpdateAt	TIMESTAMPZ	<input type="checkbox"/>	Sin valor predeter...
7	DeletedAt	TIMESTAMPZ	<input checked="" type="checkbox"/>	NULL
8	Pending	BOOLEAN	<input type="checkbox"/>	false

Figura 2: Actividades por Fecha

Registra las fechas y horarios en que se realizan las actividades. Es esencial para gestionar la programación de eventos.

Facility:

#	Nombre	Tipo de datos	Permitir NULL	Predeterminado
1	Id	UUID	<input type="checkbox"/>	Sin valor predete...
2	Name	TEXT	<input type="checkbox"/>	Sin valor predeter...
3	Location	TEXT	<input type="checkbox"/>	Sin valor predeter...
4	Type	TEXT	<input type="checkbox"/>	Sin valor predeter...
5	MaximumCapa...	INTEGER	<input type="checkbox"/>	Sin valor predeter...
6	UsagePolicy	TEXT	<input type="checkbox"/>	Sin valor predeter...
7	CreatedAt	TIMESTAMPZ	<input type="checkbox"/>	Sin valor predeter...
8	UpdateAt	TIMESTAMPZ	<input type="checkbox"/>	Sin valor predeter...
9	DeletedAt	TIMESTAMPZ	<input checked="" type="checkbox"/>	NULL

Figura 3: Instalaciones

Contiene información sobre las instalaciones del parque (nombre, ubicación, tipo, capacidad, etc.). Permite conocer y gestionar los espacios físicos disponibles.

Reservation:

#	Nombre	Tipo de datos	Permitir NULL	Predeterminado
1	Id	UUID	<input type="checkbox"/>	Sin valor predete...
2	ParentId	TEXT	<input checked="" type="checkbox"/>	NULL
3	AdditionalCo...	TEXT	<input type="checkbox"/>	Sin valor predeter...
4	AmmountOfCh...	INTEGER	<input type="checkbox"/>	Sin valor predeter...
5	ReservationSta...	TEXT	<input type="checkbox"/>	Sin valor predeter...
6	CreatedAt	TIMESTAMPZ	<input type="checkbox"/>	Sin valor predeter...
7	UpdateAt	TIMESTAMPZ	<input type="checkbox"/>	Sin valor predeter...
8	DeletedAt	TIMESTAMPZ	<input checked="" type="checkbox"/>	NULL
9	ActivityDateId	UUID	<input type="checkbox"/>	'00000000-0000-0...

Figura 4: Reservaciones

Almacena las reservas realizadas por los padres, incluyendo el código único, actividad asociada, fecha de reserva, estado, número de niños inscritos y comentarios adicionales. Es crucial para el control de la participación en actividades y eventos especiales.

Resource:

#	Nombre	Tipo de datos	Permitir NULL	Predeterminado
1	Id	UUID	<input type="checkbox"/>	Sin valor predete...
2	Name	TEXT	<input type="checkbox"/>	Sin valor predeter...
3	Type	TEXT	<input type="checkbox"/>	Sin valor predeter...
4	ResourceCondi...	TEXT	<input type="checkbox"/>	Sin valor predeter...
5	FacilityId	UUID	<input type="checkbox"/>	Sin valor predeter...
6	CreatedAt	TIMESTAMPZ	<input type="checkbox"/>	Sin valor predeter...
7	UpdateAt	TIMESTAMPZ	<input type="checkbox"/>	Sin valor predeter...
8	DeletedAt	TIMESTAMPZ	<input checked="" type="checkbox"/>	NULL
9	UseFrequency	INTEGER	<input type="checkbox"/>	0

Figura 5: Recursos

Registra los recursos del parque (juguetes, equipos deportivos, materiales educativos), junto con sus características (estado, frecuencia de uso, ubicación, etc.). Permite el seguimiento y gestión de los recursos disponibles.

ResourceDate:

#	Nombre	Tipo de datos	Permitir NULL	Predeterminado
1	Id	UUID	<input type="checkbox"/>	Sin valor predete...
2	ResourceId	UUID	<input type="checkbox"/>	Sin valor predeter...
3	Date	DATE	<input type="checkbox"/>	Sin valor predeter...
4	UseFrequency	INTEGER	<input type="checkbox"/>	Sin valor predeter...
5	CreatedAt	TIMESTAMPZ	<input type="checkbox"/>	Sin valor predeter...
6	UpdatedAt	TIMESTAMPZ	<input type="checkbox"/>	Sin valor predeter...
7	DeletedAt	TIMESTAMPZ	<input checked="" type="checkbox"/>	NULL

Figura 6: Recursos con Fecha y Frecuencia de Uso

Puede servir para llevar el control o historial del uso y estado de los recursos en determinadas fechas. Ayuda a generar reportes sobre la frecuencia de uso y mantenimiento de los recursos.

AspNetUsers:

#	Nombre	Tipo de datos	Longitud/Conjunto	Permitir NULL	Predeterminado
1	Id	TEXT		<input type="checkbox"/>	Sin valor predete...
2	FirstName	TEXT		<input type="checkbox"/>	Sin valor predeter...
3	LastName	TEXT		<input type="checkbox"/>	Sin valor predeter...
4	FullCode	TEXT		<input type="checkbox"/>	Sin valor predeter...
5	CreatedAt	TIMESTAMPZ		<input type="checkbox"/>	Sin valor predeter...
6	UpdatedAt	TIMESTAMPZ		<input type="checkbox"/>	Sin valor predeter...
7	DeletedAt	TIMESTAMPZ		<input checked="" type="checkbox"/>	NULL
8	UserName	VARCHAR	256	<input checked="" type="checkbox"/>	NULL
9	NormalizedUs...	VARCHAR	256	<input checked="" type="checkbox"/>	NULL
10	Email	VARCHAR	256	<input checked="" type="checkbox"/>	NULL
11	NormalizedEm...	VARCHAR	256	<input checked="" type="checkbox"/>	NULL
12	EmailConfirmed	BOOLEAN		<input type="checkbox"/>	Sin valor predeter...
13	PasswordHash	TEXT		<input checked="" type="checkbox"/>	NULL
14	SecurityStamp	TEXT		<input checked="" type="checkbox"/>	NULL
15	ConcurrencySt...	TEXT		<input checked="" type="checkbox"/>	NULL
16	PhoneNumber	TEXT		<input checked="" type="checkbox"/>	NULL
17	PhoneNumber...	BOOLEAN		<input type="checkbox"/>	Sin valor predeter...
18	TwoFactorEna...	BOOLEAN		<input type="checkbox"/>	Sin valor predeter...
19	LockoutEnd	TIMESTAMPZ		<input checked="" type="checkbox"/>	NULL
20	LockoutEnabled	BOOLEAN		<input type="checkbox"/>	Sin valor predeter...
21	AccessFailedC...	INTEGER		<input type="checkbox"/>	Sin valor predeter...
22	RoleId	UUID		<input type="checkbox"/>	'00000000-0000-0...

Figura 7: AspNetUsers

Es la tabla que gestiona la información de autenticación y perfiles de usuario provista por el framework.

Review:

Almacena las calificaciones y comentarios de los padres sobre las actividades, incluyendo la escala de 1 a 5 y la fecha de evaluación. Es vital para la retroalimentación y evaluación de la calidad de las actividades.

#	Nombre	Tipo de datos	Permitir NULL	Predeterminado
1	Id	UUID	<input type="checkbox"/>	Sin valor predete...
2	ParentId	TEXT	<input checked="" type="checkbox"/>	NULL
3	ActivityDateld	UUID	<input type="checkbox"/>	Sin valor predeter...
4	Comments	TEXT	<input type="checkbox"/>	Sin valor predeter...
5	Score	INTEGER	<input type="checkbox"/>	Sin valor predeter...
6	CreatedAt	TIMESTAMP TZ	<input type="checkbox"/>	Sin valor predeter...
7	UpdatedAt	TIMESTAMP TZ	<input type="checkbox"/>	Sin valor predeter...
8	DeletedAt	TIMESTAMP TZ	<input checked="" type="checkbox"/>	NULL

Figura 8: Reseñas

Rol:

#	Nombre	Tipo de datos	Permitir NULL	Predeterminado
1	Id	UUID	<input type="checkbox"/>	Sin valor predete...
2	Name	TEXT	<input type="checkbox"/>	Sin valor predeter...

Figura 9: Roles

Define los roles o perfiles de usuario (padre, educador, administrador) que interactúan con el sistema. Facilita la gestión de permisos y accesos dentro de la aplicación.

A continuación se muestran otros datos que son parte de la infraestructura de autenticación y seguridad (generadas y gestionadas por el framework) y no representan datos centrales del dominio de negocio.

AspNetUserLogins:

#	Nombre	Tipo de datos	Permitir NULL	Predeterminado
1	LoginProvider	TEXT	<input type="checkbox"/>	Sin valor predete...
2	ProviderKey	TEXT	<input type="checkbox"/>	Sin valor predete...
3	ProviderDispla...	TEXT	<input checked="" type="checkbox"/>	NULL
4	UserId	TEXT	<input type="checkbox"/>	Sin valor predeter...

Figura 10: AspNetUserLogins

Se encarga de almacenar los datos relacionados con los inicios de sesión externos y la gestión de identidades, lo que es parte del mecanismo de seguridad y autenticación. No contiene información de negocio propiamente dicha.

AspNetUserTokens:

#	Nombre	Tipo de datos	Permitir NULL	Predeterminado
1	UserId	TEXT	<input type="checkbox"/>	Sin valor predete...
2	LoginProvider	TEXT	<input type="checkbox"/>	Sin valor predete...
3	Name	TEXT	<input type="checkbox"/>	Sin valor predete...
4	Value	TEXT	<input checked="" type="checkbox"/>	NULL

Figura 11: AspNetUserTokens

Maneja tokens para la autenticación y recuperación de sesiones, también parte de la infraestructura de seguridad del sistema. Tampoco forma parte del dominio específico del parque infantil.

UserProfileImages:

#	Nombre	Tipo de datos	Permitir NULL	Predeterminado
1	Id	UUID	<input type="checkbox"/>	Sin valor predete...
2	ProfileImage	TEXT	<input type="checkbox"/>	Sin valor predeter...
3	OtherImages	TEXT	<input type="checkbox"/>	Sin valor predeter...
4	UserId	TEXT	<input checked="" type="checkbox"/>	NULL
5	CreatedAt	TIMESTAMP TZ	<input type="checkbox"/>	Sin valor predeter...
6	UpdatedAt	TIMESTAMP TZ	<input type="checkbox"/>	Sin valor predeter...
7	DeletedAt	TIMESTAMP TZ	<input checked="" type="checkbox"/>	NULL

Figura 12: UserProfileImages

Usualmente almacena o referencia las imágenes de perfil de los usuarios. Aunque está relacionada con la personalización del usuario, es un dato complementario y no central al negocio que gestiona actividades, recursos o reservas en el parque.

Esquema con el diseño de la aplicación:

A continuación se muestra un esquema que representa el diseño de la aplicación: El esquema

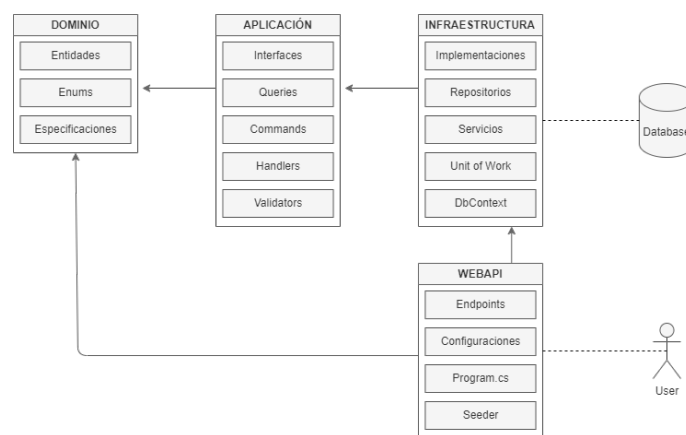


Figura 13: Esquema de diseño de la aplicación

expuesto representa las diferentes capas que componen el diseño de nuestra aplicación:

- La capa de Dominio es el núcleo de la aplicación y contiene la lógica de negocio fundamental. Es independiente de las preocupaciones técnicas y de infraestructura.
- La capa de Aplicación coordina las operaciones de la aplicación, orquestando la lógica de negocio y las interacciones con la infraestructura.
- La capa de Infraestructura es responsable de proporcionar las implementaciones concretas necesarias para interactuar con sistemas externos, como bases de datos, servicios de mensajería, y otros recursos de infraestructura.
- La capa de WebApi actúa como el punto de entrada para las solicitudes HTTP, exponiendo la funcionalidad de la aplicación a través de endpoints que los clientes pueden consumir.

Las funcionalidades del producto se representan a través del siguiente diagrama, que representa los casos de uso y flujo de navegación de los usuarios a través de la aplicación.

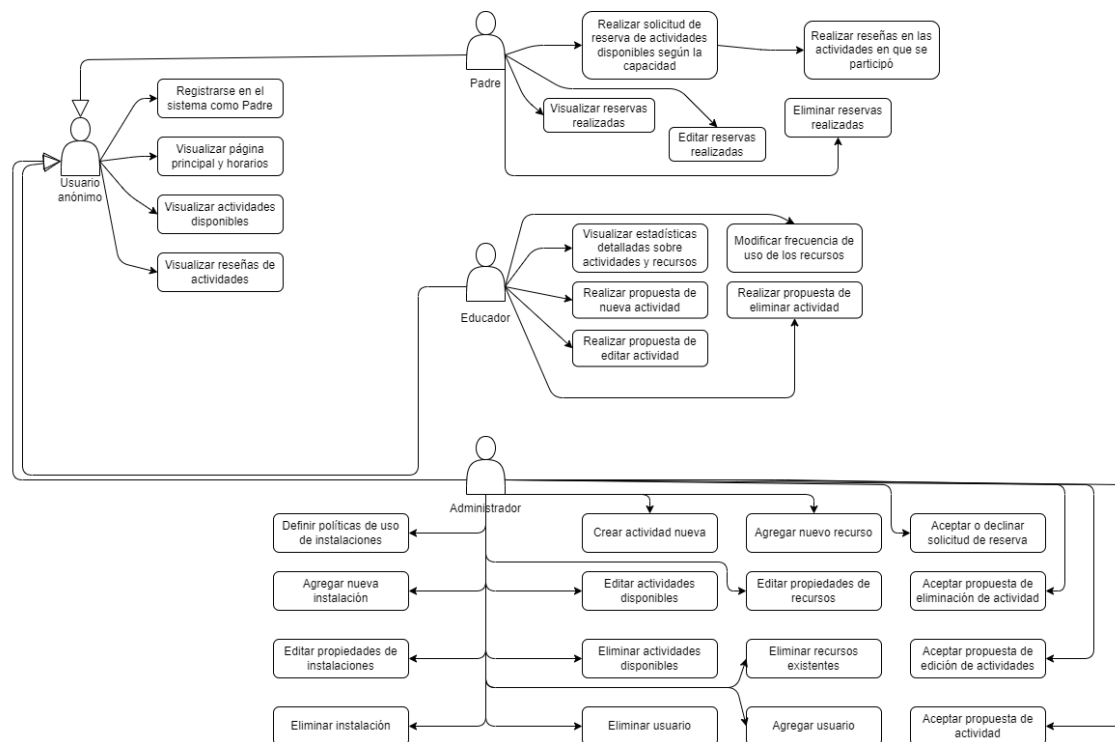


Figura 14: Diagrama de Casos de Uso

Los usuarios padres pueden consultar un resumen de actividades programadas y reservar para una cantidad específica de participantes en aquellas que estén disponibles. Luego, también podrá calificar y comentar en aquellas actividades en las que haya reservado una vez finalicen.

Los educadores son los encargados de gestionar y proponer nuevas actividades que posteriormente deben ser aprobadas o declinadas por un administrador. También serán los encargados de gestionar estas actividades, por lo cual poseen un apartado de estadísticas detalladas que les permitirán manejar los recursos disponibles para cada actividad y los horarios de las mismas.

El administrador es el encargado de controlar y gestionar toda la información respectiva a cada entidad, por lo cual cuenta con estadísticas completas con múltiples funcionalidades y es el encargado de manejar la publicación de cualquier oferta en la página.

Esquema de clases definidas:

El siguiente esquema muestra las interacciones, dependencias y relaciones entre las clases definidas en el sistema.

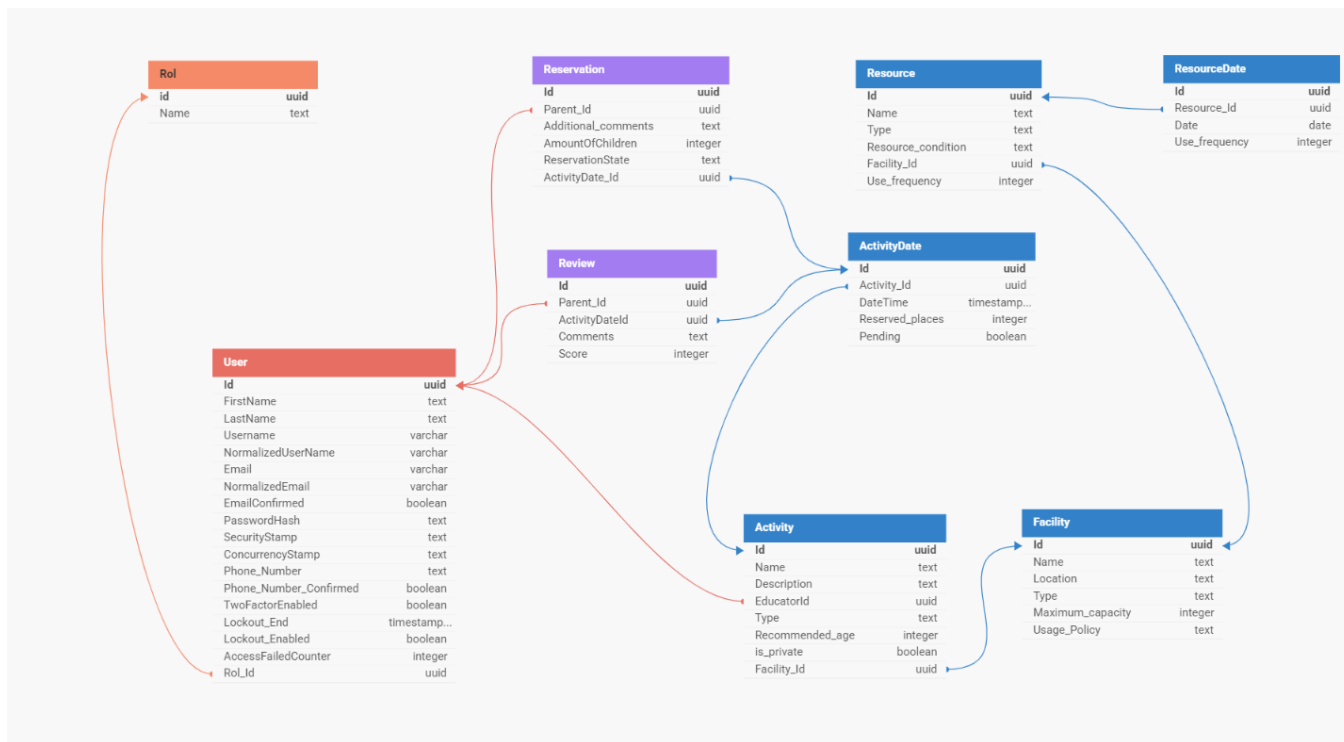


Figura 15: Esquema de clases definidas

Solución propuesta:

A continuación se describe la conformación de la solución propuesta en diferentes aspectos de la misma.

Arquitectura Empleada:

El sistema implementa el patrón de arquitectura Domain-Driven Design (DDD), estructurado en las siguientes capas:

- **Capa de Dominio:** La capa de Dominio es el núcleo de la aplicación y contiene la lógica de negocio fundamental. Es independiente de las preocupaciones técnicas y de infraestructura.
 - **Contenido:**
 - **Entidades:** Clases que representan objetos de negocio con identidad única y comportamiento.
 - **Enums:** Tipos de datos que definen un conjunto de constantes con nombre, utilizados para representar valores discretos.
 - **Specifications:** Objetos que encapsulan reglas de negocio o criterios de búsqueda que pueden ser reutilizados y combinados.
 - **Dependencias:** No depende de otras capas. Es autónoma y define el comportamiento del negocio.
- **Capa de Aplicación:** La capa de Aplicación coordina las operaciones de la aplicación, orquestando la lógica de negocio y las interacciones con la infraestructura.
 - **Contenido:**

- Interfaces de Servicios: Definiciones de servicios que encapsulan lógica de aplicación.
 - Interfaces de Repositorios y Unit of Work: Definiciones para acceder y manipular datos.
 - Commands y Queries: Representan acciones que la aplicación debe realizar, separando las operaciones de escritura (commands) de las de lectura (queries).
 - Handlers: Implementaciones que manejan la ejecución de commands y queries.
 - Validators: Validan los datos antes de procesar commands y queries.
 - DTOs y Responses: Objetos de transferencia de datos y respuestas de endpoints.
- Dependencias: Depende del Dominio para utilizar entidades y especificaciones. No depende directamente de la Infraestructura, sino que define interfaces que la Infraestructura implementa.
- Capa de Infraestructura: La capa de Infraestructura es responsable de proporcionar las implementaciones concretas necesarias para interactuar con sistemas externos, como bases de datos, servicios de mensajería, y otros recursos de infraestructura.
 - Contenido:
 - Implementaciones de Servicios y Repositorios: Clases que implementan las interfaces definidas en la capa de Aplicación. Estas clases manejan la lógica de acceso a datos y otras operaciones que requieren interacción con sistemas externos.
 - Unit of Work y Repository Factory: Gestionan las transacciones y la creación de repositorios, asegurando que las operaciones de datos sean consistentes y atómicas.
 - DbContext y Migraciones: Configuran el acceso a la base de datos utilizando Entity Framework, y gestionan los cambios en el esquema de la base de datos a través de migraciones.
 - Configuraciones de Dependencias: Configura el contenedor de inyección de dependencias, especificando qué implementaciones concretas deben ser utilizadas para las interfaces definidas en la capa de Aplicación
 - Dependencias: Depende de la capa de Aplicación para las interfaces que implementa. No debería depender de la capa de Dominio directamente, manteniendo así la separación de preocupaciones.
- Capa de WebApi: La capa de WebApi actúa como el punto de entrada para las solicitudes HTTP, exponiendo la funcionalidad de la aplicación a través de endpoints que los clientes pueden consumir.
 - Contenido:
 - Endpoints: Controladores que manejan las solicitudes HTTP. Cada endpoint se asocia con un command o query de la capa de Aplicación, siguiendo el patrón CQRS.
 - Configuraciones Globales: Archivos como appsettings.json que contienen configuraciones globales para la aplicación, como cadenas de conexión y configuraciones de servicios.
 - Seeder: Inicializa datos en la base de datos, útil para entornos de desarrollo o pruebas.

- Program.cs: Configura y arranca la aplicación, incluyendo la configuración de middleware, servicios, y el contenedor de inyección de dependencias.
- Program.cs: Configura y arranca la aplicación, incluyendo la configuración de middleware, servicios, y el contenedor de inyección de dependencias.

Descripción de los Módulos:

Los módulos que componen la aplicación se describen a continuación:

- Dominio (Domain): Define las entidades y especificaciones del negocio.
- Aplicación (Application): Implementa la lógica de negocio a través de patrones CQRS y especificaciones.
- Infraestructura (Infrastructure): Gestiona la persistencia de datos y la interacción con la base de datos.
- Interfaz Web (WebApi): Proporciona la interfaz de usuario a través de endpoints y maneja las solicitudes HTTP.

Patrones Implementados:

Los patrones implementados para mejorar la eficiencia y calidad del código son los siguientes:

- Repository Pattern: Para el acceso a datos desacoplado.
- Specification Pattern: Para construir consultas complejas de manera flexible.
- CQRS (Command Query Responsibility Segregation): Para separar las operaciones de lectura y escritura.
- Repository Factory Pattern: Para la creación dinámica de repositorios, facilitando la extensibilidad y el mantenimiento.
- Unit of Work Pattern: Para gestionar transacciones de manera eficiente, asegurando la consistencia de los datos.

Mantenimiento del Sistema:

Como exponentes de las buenas prácticas de programación, se ha implementado el uso de pruebas unitarias para asegurar la calidad del código. Además, no se ha excluido el uso de comentarios y documentación (docstrings) en todo el código.

Se ha trabajado con GitHub como herramienta de control de versiones para la gestión del código fuente, y se ha hecho uso de herramientas CASE como Jira para la gestión de tareas y planificación.

Algunas de las recomendaciones a seguir para el mantenimiento de esta aplicación lo constituyen el mantenimiento y actualización de las dependencias del proyecto, revisión y actualización de pruebas unitarias con cada cambio significativo y la documentación de cualquier cambio en la arquitectura o en las especificaciones del sistema.

El sistema desarrollado cumple con todos los requisitos funcionales y no funcionales planteados, proporcionando una solución robusta y escalable para la gestión de un parque infantil. La arquitectura empleada permite un fácil mantenimiento y extensibilidad del sistema, asegurando su adaptabilidad a futuras necesidades y mejoras.