

Sustento de Arquitectura Implementada en API

- El API está basada en Web API Net 5.0, en su estructura tiene carpetas Models, Controllers y Data, en cuales en la carpeta Models se encuentran las dos entidades que tiene una estructura similar a la base de datos(Eschema); mientras que en la carpeta Controllers se encuentran los endpoints de cada entidad con la cual realizaremos las diferentes operaciones de mantenimiento a la base de datos y por último en la carpeta Base se encuentra el archivo encargado de realizar las diferentes consultas a la base de datos utilizando los procedimientos almacenados que se encuentran alojada en el gestor de base de datos.
- Esta API cuenta con dos entidades, que están representadas en sus respectivos archivos Model.; cuyos nombres de estas entidades son:
 - Categoria
 - Producto
- Esta API estará realizando consultas a una base de datos denominada EJERCICIO_TECNICO_CANNVIA que se encuentra alojada en un gestor de base de datos basada en SQL SERVER. Esta base de datos contara con dos tablas denominadas:
 - Categoria
 - Producto
- Los esquemas utilizados en esta API son:
 - **Categoria:**

```
{
    Id          : integer,
    Nombre      : string,
    Descripcion : string
}
```
 - **Producto:**

```
{
    Id          : integer,
    Nombre      : string,
    Descripcion : string,
    Unidades    : integer,
    IdCategoria : integer
}
```
- **Nota:** Tanto en el campo de Nombre y Descripcion, estos no pueden ser datos nulos.

- Esta Api cuenta con 13 Endpoints, que serán utilizados para las siguientes acciones:

- Obtener todos los datos de la tabla Categoria que se encuentran activos, utilizando método GET.

GET -> /api/Categorias

- Obtener los datos de tabla Categoria según su Id representativo (índice), utilizando método GET.

GET -> /api/Categorias/{id}

- Agregar datos a la tabla Categoria. Enviando su respectivo esquema. Utilizando método POST.

POST -> /api/Categorias

- Actualizar un dato de la tabla Categoria, donde necesitaremos su Id representativo (índice). Enviando su respectivo esquema. Utilizando método PUT.

PUT -> /api/Categorias/{id}

- Eliminar un dato de la tabla Categoria. Utilizando método DELETE, donde necesitaremos su Id representativo (índice).

DELETE -> /api/Categorias/{id}

- Eliminación lógica de un dato de la tabla Categoria, donde necesitaremos su Id representativo (índice). Utilizando método DELETE.

DELETE -> /api/Categorias/Delete/{id}

- Obtener todos los datos de la tabla Producto que se encuentran activos, utilizando método GET.

GET -> /api/Productos

- Obtener los datos de tabla Producto según su Id representativo (índice), utilizando método GET.

GET -> /api/Productos/{id}

- Obtener los datos de tabla Producto según su tipo de Categoria al que pertenecen mediante un idcategoria (índice). Utilizando método GET.

GET -> /api/Productos/Categoria/{idcategoria}

- Agregar datos a la tabla Producto. Enviando su respectivo esquema. Utilizando método POST.

POST -> /api/Productos

- Actualizar un dato de la tabla Producto, donde necesitaremos su Id representativo (índice). Enviando su respectivo esquema. Utilizando método PUT.

PUT -> /api/Productos/{id}

- Eliminar un dato de la tabla Producto. Utilizando método DELETE, donde necesitaremos su Id representativo (índice).

DELETE -> /api/Productos/{id}

- Eliminación lógica de un dato de la tabla Producto, donde necesitaremos su Id representativo (índice). Utilizando método DELETE.

DELETE -> /api/Productos/Delete/{id}