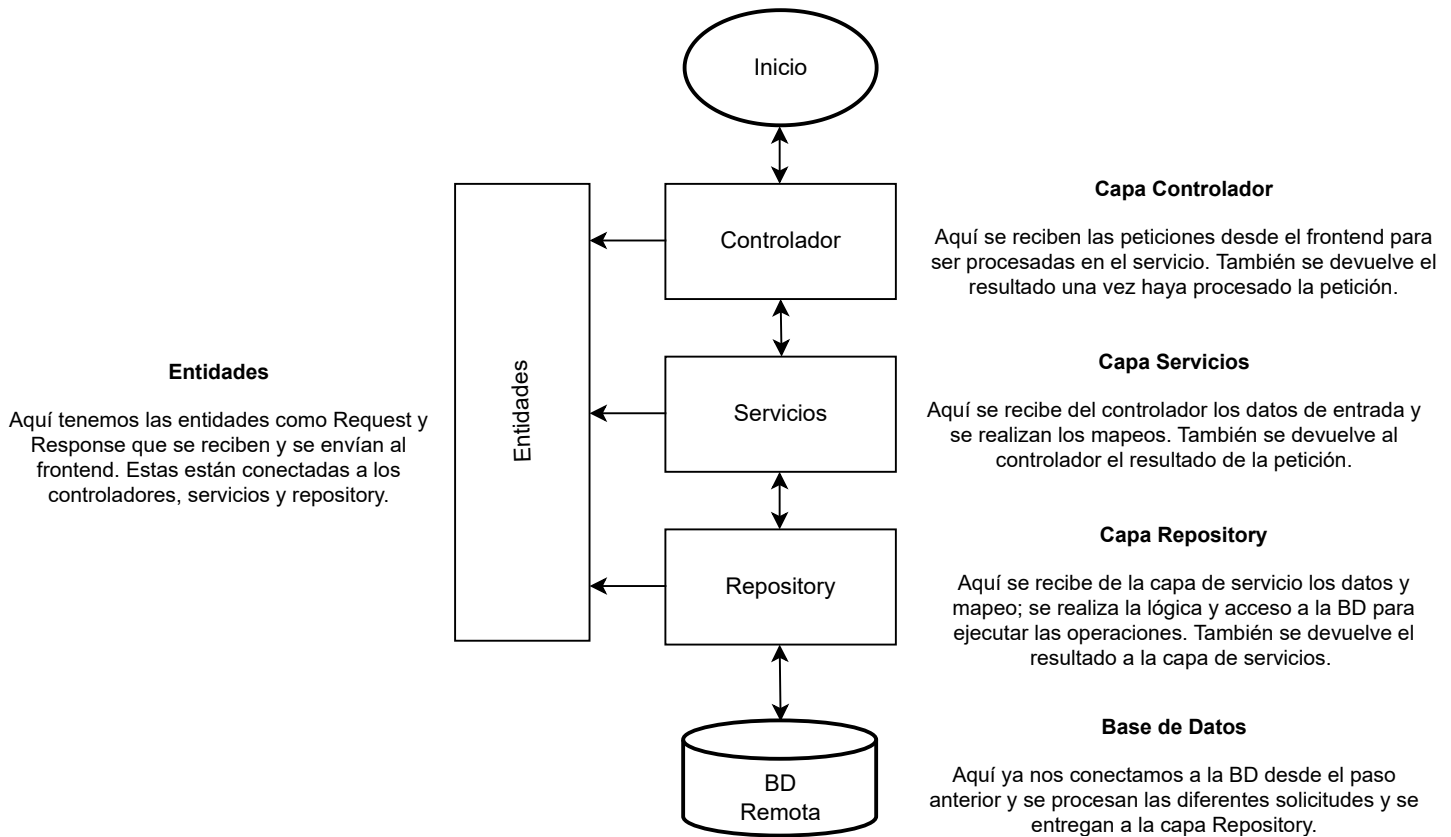


BACKEND

El proyecto del Backend se desarrolló en Visual C# con .Net Core conectado a la base de datos remota de SQL Server. El API consta de 3 capas y se implementó el patrón de diseño de Inyección de dependencias.



Documentación del API

La documentación de los métodos del API se encuentran en Swagger. Los métodos creados fueron:

DatosVehiculos

POST: /api/v1/datosvehiculos/conteo : Aquí ejecutamos la Petición login de los servicios de la documentación de la API de pruebas de conteo y recaudo de <http://23.102.103.53:5200/documentation/>. Primero obtenemos el token del login, luego con ese token ejecutamos la consulta del método de conteo y obtenemos los datos, finalmente realizamos el insert en la tabla ConteoVehiculos en el esquema a mi nombre Eduardo en la BD Remota.

POST: /api/v1/datosvehiculos/recaudo : Aquí ejecutamos la Petición login de los servicios de la documentación de la API de pruebas de conteo y recaudo de <http://23.102.103.53:5200/documentation/>. Primero obtenemos el token del login, luego con ese token ejecutamos la consulta del método de recaudo y obtenemos los datos, finalmente realizamos el insert en la tabla RecaudoVehiculos en el esquema a mi nombre Eduardo en la BD Remota.

Recaudos

GET: /api/v1/recaudos: Aquí obtenemos la información guardada en la base de datos, donde se puede realizar filtros en la consulta.

POST : /api/v1/recaudos: Aquí se ejecuta el método post del Login que utiliza la seguridad del JSON Web Token (JWT), donde podemos ver que en los métodos del controlador tenemos la línea `Authorize` debido a que no se alcanzó a realizar la funcionalidad en el frontend, pero este método genera el token que sería utilizado para validar los usuarios y si tienen permiso de acceder o no.

Reporte

GET: /api/v1/reporte: Aquí realizamos todo el procesamiento de la información para generar el reporte, acá se reciben parámetros de fecha inicio y fecha fin en el formato YYYY/MM/DD.