

# Geolocation App - .Net Core 9

Este proyecto implementa una API basada en el principio de arquitectura limpia utilizando .NET Core 9. El propósito de la aplicación es registrar la ubicación de un visitante, determinar la moneda de su país y almacenar los datos en una base de datos.

## Requisitos Previos

Antes de comenzar, asegúrate de tener instalados los siguientes componentes:

- [Visual Studio 2022](#) con las cargas de trabajo:
  - Desarrollo para ASP.NET y web
  - Herramientas para contenedores
- [.NET Core SDK 9.0](#)
- [SQL Server](#) para la base de datos
- Una herramienta para la gestión de dependencias, como [NuGet](#).

## Instalación

Sigue los pasos para configurar y ejecutar el proyecto localmente:

### 1. Clonar el repositorio

```
git clone git@github.com:abhuanco/CleanArchitecture.git
cd CleanArchitecture
```

### 2. Configurar la base de datos y endpoints de la api

Crea una base de datos en SQL Server y actualiza las cadenas de conexión en el archivo `appsettings.Development.json` ubicado en la carpeta `GeolocationApp.Api`:

```
""json { "ConnectionStrings": { "DefaultConnection": "Server=localhost,1433;Database=GeolocationApi2;User Id=sa;Password=Root@00m;Encrypt=True;Trust Server Certificate=True" } "GeolocationSetting": { "GeolocationApiUrl": "https://xxxxxx/geolocate", "CurrencyApiUrl": "https://xxxxxxx/currencies" } }
```

### 2. Restaurar dependencias

Ejecuta el siguiente comando para restaurar los paquetes necesarios:

```
dotnet restore
```

### 3. Aplicar migraciones de la base de datos

Aplica las migraciones para crear las tablas en la base de datos configurada:

```
dotnet ef database update --project ./GeolocationApp.Infrastructure --startup-project ./GeolocationApp.Api
```

- La importación manual de la base de datos también está disponible en la ruta `./GeolocationApp/GeolocationApp.Infrastructure/Script SQL/Visits.sql`

### 4. Ejecutar la aplicación

Inicia la aplicación de API ejecutando:

```
dotnet run --project GeolocationApp.Api
```

La API estará disponible en `https://localhost:5103/scalar/v1`.

### 5. Ejecutar la aplicación React

Inicia creando una copia a partir de `.env.example` en `GeolocationApp/ClientApp/`

```
cp .env.example .env
```

instalar los módulos en `GeolocationApp/ClientApp/`

```
npm install
```

Inicia la aplicación Web ejecutando:

```
dotnet run --project GeolocationApp.Web
```

La API estará disponible en `http://localhost:5176`. esperar al proxy en caso de no responder

## Uso

La API incluye los siguientes endpoints principales:

### 1. Registrar ubicación y moneda en caso de estar disponible

- POST /api/Visit
- GET /api/Visit
- GET /api/Visit/{id}
- PUT /api/Visit/{id}
- DELETE /api/Visit/{id}

## Pruebas unitarias

---

1. Debe ir a la solucion ./GeolocationApp y ejecutar el comando

```
dotnet test
```

## Contribución

---

Si deseas contribuir al proyecto, sigue estas pautas:

1. Haz un fork del repositorio.
2. Crea una rama para tu funcionalidad: `git checkout -b feature/nueva-funcionalidad` .
3. Realiza tus cambios y confirma los commits.
4. Envía un pull request para revisión.

## Licencia

---

Este proyecto está bajo la Licencia MIT. Consulta el archivo `LICENSE` para más detalles.