

## Comandos de Dotnet, GitHub y VS Code para crear Proyectos y ejecutar acciones

Dotnet Command	Description
dotnet -h	Para Ayuda, ver todas las opciones de dotnet
dotnet new -l	Ver todas las plantillas de Proyectos .net
dotnet new sln	Crear una Nueva Solution
dotnet new webapi -o API	Crear nuevo Proyecto web Api y darle el nombre <b>API</b>
dotnet sln -h	Para ver las opciones de la Solution, por ejemplo: add, para agregar proyectos a la Solution
dotnet sln add API/	Agregar el nuevo proyecto de nombre <b>API</b> a la Solution
dotnet sln list	Para ver todos los proyectos dentro de la Solution
code .	Para abrir la Solution en VS code
dotnet run dotnet watch run	Para ejecutar la Aplicación, tenemos que asegurarnos que estemos en el directorio correcto
En VS .Net: Generate Assets for Build and Debug OmniSharp restart (Si no funciona) ctrl+shift+p	Esto es para que se nos genere la carpeta .vscode en nuestra solución, que nos permite ejecutar código c# y reconocer errores
dotnet tool install --global dotnet-ef --version 6.0.0 (update)	Instalar tools Globalmente
dotnet tool update --global dotnet-ef	Actualizar tools Globalmente
dotnet add package Microsoft.EntityFrameworkCore.Design --version 6.0.0	Instalar EntityFramework.Design sirve para que el tools funcione
dotnet ef -h	Para ver las opciones de Entity Framework: database, dbContext, migrations
dotnet ef migrations add NombreMigracion -o Data/Migrations	Para generar una nueva Migración, -o es Origen, Data es el directorio donde está el DbContext y Migrations es el directorio en donde se van a generar los archivos de migraciones.
dotnet ef database update	Actualiza o Crea la Base de Datos en base a las migraciones.
dotnet new classlib -o Core	Crear un Proyecto de Librería de clases llamado Core
dotnet new classlib -o Infrastructure	Crear un Proyecto de Librería de clases llamado Infrastructure.
dotnet sln add Core/	Agregar Proyecto Core a la solución
dotnet sln add Infrastructure/	Agregar Proyecto Infrastructure a la solución
dotnet sln list	Para ver los proyectos agregados a la solución
<b>Dependencias</b>	
cd API	

dotnet add reference ../Infrastructure	Agregar dependencias de Proyecto Infrastructure a API
cd ..	Volvemos a la Solución
cd Infrastructure	
dotnet add reference ../Core/	Agregar dependencias de Proyecto Core a Infrastructure
cd ..	Volvemos a la Solución
dotnet restore	Reiniciamos para que todas las referencias tomen efecto
Movemos la carpeta Entities de API a Core	
Movemos la carpeta Data de API a Infrastructure	
Corregimos todas las dependencias	
Movemos el paquete EntityFrameworkCore de API.csproj y creamos un ItemGroup en Infrastructure.csproj y lo pegamos	
dotnet build	Para construir nuevamente el Proyecto
Git	
git status	Para Ver el Status de Git
git init	Para Inicializar Git
Crear archivo .gitignore	Para indicar lo que no vamos a subir a Git
git add .	Agregamos todos los cambios
git commit -m "primer Commit"	Agregar Commit con Mensaje para subir los cambios
git remote add origin <a href="https://github.com/carlospiedrab/skinet.git">https://github.com/carlospiedrab/skinet.git</a>	
git push -u origin master	
git push -u origin main (mejor Opción)	
Borrar las Anteriores Migraciones	
dotnet ef migrations remove -p Infrastructure -s API	Borrar las Anteriores Migraciones
Nueva Migration seleccionando el proyecto	
dotnet ef migrations add InitialCreate -p Infrastructure -s API -o Data/Migrations	

## Comandos de Docker en VS Code

Docker Command	Description
docker --version	Para ver la versión de Docker
docker build -t <docker hub id>/globaltours .	Para generar la Imagen de Docker
docker run -p 8080:80 -d <docker hub id>/globaltours	Crear el container de Docker
docker ps	Lista de Containers activos
docker stop #id	Detener un Container en base al Id
docker start #id	Iniciar un Container en base al Id
docker push <docker hub id>/globaltours	Push a Docker Hub

## Comandos de Heroku con Docker en VS Code

Heroku Command	Description
heroku login	Conectarse a Heroku
heroku container:login	Enlazar el login con contenedores heroku
docker build -t registry.heroku.com/globaltours/web .	Hacer el build de Docker container a la aplicación en Heroku
heroku container:push -a globaltours web	Push del contenedor a la aplicación
heroku container:release -a globaltours web	Hacer Release a la aplicación
heroku logs --tail --app globaltours	Ver los posibles errores al desplegar

## Comandos de Kubernetes en VS Code

Heroku Command	Description
Crear el Kubernetes File	
kubectl version	Versión de Kubernetes
kubectl apply -f platform-depl.yaml	Crear
kubectl get deployments	Obtener Despliegues
kubectl get pods	
kubectl get services	