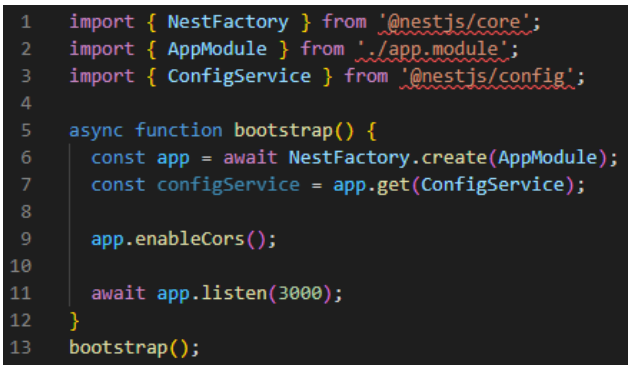
Desarrollo dispositivos móviles:

GUÍA PRÁCTICA: CREACIÓN DE HEALTH-API CON NESTJS Y SUPABASE:

Paso	Descripción tarea	Observaciones
1	Verifica que tengas instaladas las herramientas necesarias: • PostgreSQL • DBeaver • NodeJS	
2	Crea un repositorio nuevo en GitHub. Nombre del repositorio: **health-api**	El Repositorio fue creado al inicio del semestre
3	Clona el repositorio en tu computadora: $ git clone **REPO_URL**	Toda la información de Git/GitHub la encuentras en tus apuntes o en la guía práctica que está disponible en TAU.
4	Instala el CLI de **NestJS**: $ npm i -g @nestjs/cli	
5	Crea un nuevo proyecto (estructura): $ nest new nest-supabase-auth .	Elige la opción de npm
6	Ingresa al repositorio y ábrelo con tu editor de código favorito.	En clase estamos usando Visual Studio Code, pero puedes usar el de tu preferencia.
7	Instala las dependencias requeridas para este proyecto: Instalación de dependencias de forma individual: $ npm install @nestjs/config $ npm install @nestjs/jwt $ npm install @nestjs/passport $ npm install passport $ npm install passport-jwt $ npm install bcrypt $ npm install pg $ npm install supabase $ npm install @supabase/supabase-js	

	Instalación de dependencias en conjunto: $ npm install @nestjs/config @nestjs/jwt @nestjs/passport passport passport-jwt bcrypt pg supabase **Descripción de paquetes:** • @nestjs/config: Para manejar variables de entorno. • @nestjs/jwt: Para manejar autenticación con tokens. • @nestjs/passport y passport-jwt: Para autenticación basada en JWT. • bcrypt: Para encriptar contraseñas. • pg: Cliente de PostgreSQL para conectar con Supabase. • supabase: SDK oficial para interactuar con Supabase.	
8	Configura tus variables de entorno: Crea un archivo **.env** en la raíz del repositorio con el siguiente contenido: SUPABASE_URL=**supabase_url** SUPABASE_KEY=**supabase_anon_key** JWT_SECRET=**clave_secreta_jwt** Reemplaza **supabase_url**, **supabase_anon_key** y **clave_secreta_jwt**, por los valores de tu cuenta personal de Supabase.	
9	Revisa que el archivo src/main.ts tenga esta estructura:  ``` import { NestFactory } from '@nestjs/core'; import { AppModule } from './app.module'; import { ConfigService } from '@nestjs/config'; async function bootstrap() { const app = await NestFactory.create(AppModule); const configService = app.get(ConfigService); app.enableCors(); await app.listen(3000); } bootstrap(); ```	

10	Genera un módulo, servicio y controlador para autenticación: $ nest generate module auth $ nest generate service auth $ nest generate controller auth	
11	Edita src/auth/auth.service.ts Descarga el archive auth.service.ts disponible en TAU y agrega las líneas de código que hagan falta en el mismo archivo de tu proyecto.	
12	Edita src/auth/auth.controller.ts Descarga el archive auth.controller.ts disponible en TAU y agrega las líneas de código que hagan falta en el mismo archivo de tu proyecto.	
13	Edita src/auth/auth.module.ts Descarga el archive auth. module.ts disponible en TAU y agrega las líneas de código que hagan falta en el mismo archivo de tu proyecto.	
14	Corre el servidor: $ npm run start	
15	Prueba las rutas de login y register con ThunderClient o Postman: http://localhost:3000/auth/register http://localhost:3000/auth/login	
16	Verifica que los usuarios queden registrados en Supabase	
17	No olvides versionar los cambios en GitHub (STAGING)	