



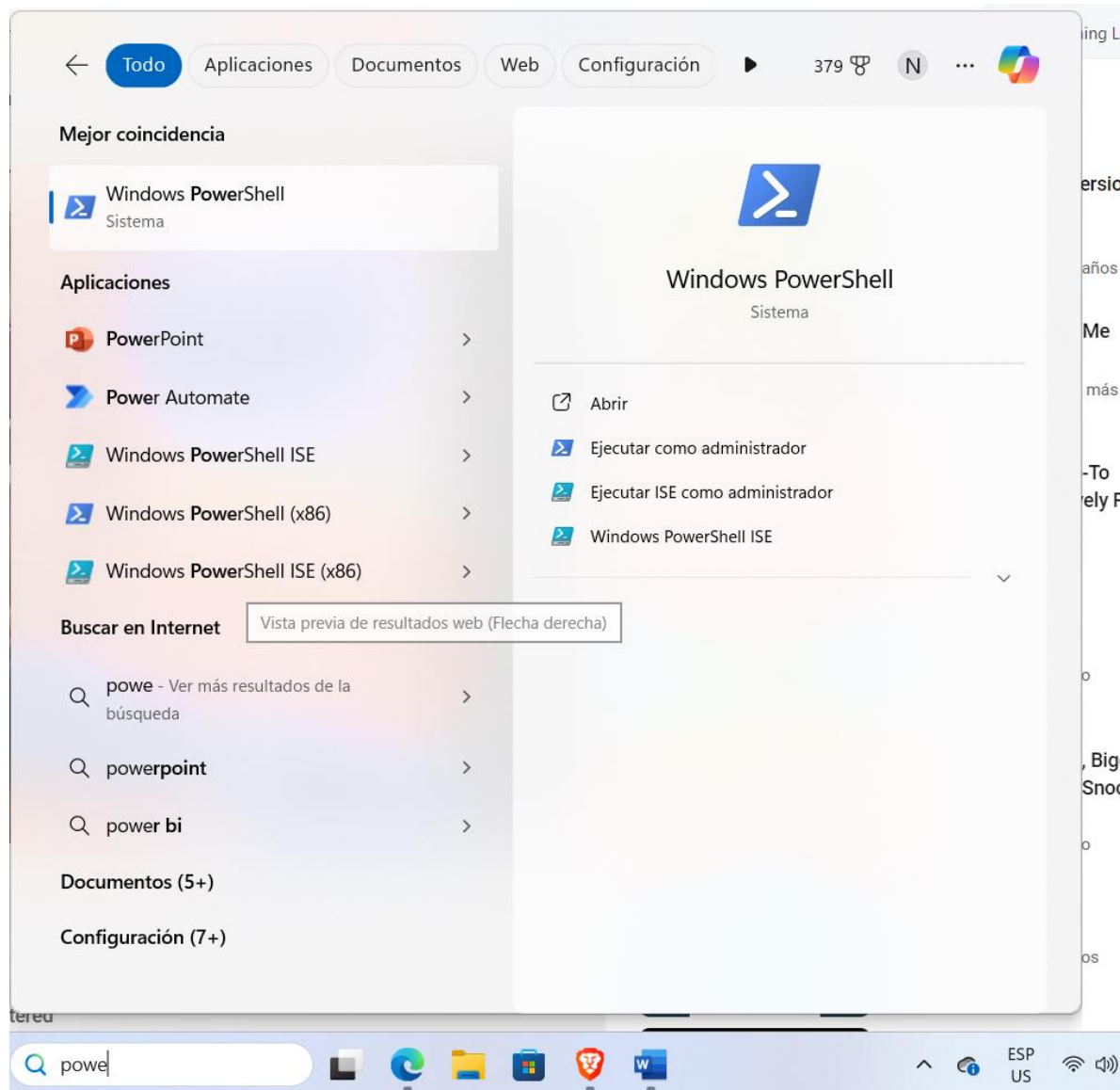
EXAMEN CICI

Jose Noe Gomez Sánchez

Taller de Desarrollo

jose gomez

Primero abrimos PowerShell en Windows como administrador



Después creamos nuestra carpeta "Examen" desde PowerShell para crear el archivo nest

```
PS C:\Examen> Set-ExecutionPolicy Bypass -Scope Process

Cambio de directiva de ejecución
La directiva de ejecución te ayuda a protegerte de scripts en los que no confías. Si cambias dicha directiva, podrías exponerte a los riesgos de seguridad descritos en el tema de la Ayuda
about_Execution_Policies en https://go.microsoft.com/fwlink/?LinkID=135170. ¿Quieres cambiar la directiva de ejecución?
[S] Sí [O] No [?] Ayuda (el valor predeterminado es "N"): 0
PS C:\Examen> nest new exam

We will scaffold your app in a few seconds...

Which package manager would you like to use? npm
CREATE exam/.eslintrc.js (688 bytes)
CREATE exam/.prettierrc (54 bytes)
CREATE exam/nest-cli.json (179 bytes)
CREATE exam/package.json (2012 bytes)
CREATE exam/README.md (3413 bytes)
CREATE exam/tsconfig.build.json (101 bytes)
CREATE exam/tsconfig.json (567 bytes)
CREATE exam/src/app.controller.ts (286 bytes)
CREATE exam/src/app.module.ts (259 bytes)
CREATE exam/src/app.service.ts (150 bytes)
CREATE exam/src/main.ts (216 bytes)
CREATE exam/src/app.controller.spec.ts (639 bytes)
CREATE exam/test/jest-e2e.json (192 bytes)
CREATE exam/test/app.e2e-spec.ts (654 bytes)

Installation in progress...
```

En mi caso debo usar el comando **Set-ExecutionPolicy Bypass -Scope Process** para que pueda ejecutar el comando “nest new” o saldrá este error

```
PS C:\> cd ".\Examen\"
PS C:\Examen> nest new exam
nest : No se puede cargar el archivo C:\Users\noe3E\AppData\Roaming\vpw\nest.ps1. El archivo C:\Users\noe3E\AppData\Roaming\vpw\nest.ps1 no está firmado digitalmente. No se puede ejecutar este script en el sistema actual. Para obtener más información acerca de la ejecución de scripts y la configuración de la directiva de ejecución, consulta about_Execution_Policies en https://go.microsoft.com/fwlink/?linkID=135170.
En línea: 1 Carácter: 1
+ nest new exam
+ ~~~~~
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
```

Creamos nuestro controlador y asignamos su nombre

```
PS C:\Examen> cd exan
PS C:\Examen\exan> nest generate controller exam
CREATE src/exam/exam.controller.ts (101 bytes)
CREATE src/exam/exam.controller.spec.ts (496 bytes)
UPDATE src/app.module.ts (332 bytes)
PS C:\Examen\exan> npm install ejs
```

Instalamos ejs

```
PS C:\Examen\exan> npm install ejs
added 7 packages, and audited 730 packages in 4s

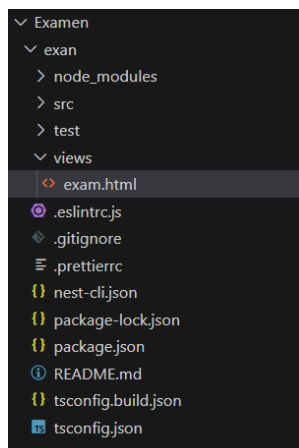
115 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Examen\exan> mkdir views

Directorio: C:\Examen\exan

Mode                LastWriteTime         Length Name
----                -
d-----          21/02/2024  08:54 a. m.          views
```

Abrimos visual estudio y creamos el html con el nombre del controlador



```

Examen
├── exan
│   ├── node_modules
│   ├── src
│   ├── test
│   └── views
│       └── exam.html
├── .eslintignore
├── .gitignore
├── .prettierrc
├── nest-cli.json
├── package-lock.json
├── package.json
├── README.md
├── tsconfig.build.json
└── tsconfig.json
```

Nos dirigimos a la carpeta “SRC” y cambiamos la dirección del “Main.ts” para dirigirlo al HTML

```
import { NestFactory } from '@nestjs/core';
import { AppModule } from './app.module';

async function bootstrap() {
  const app = await NestFactory.create(AppModule);
  await app.listen(3000);
}
bootstrap();
```

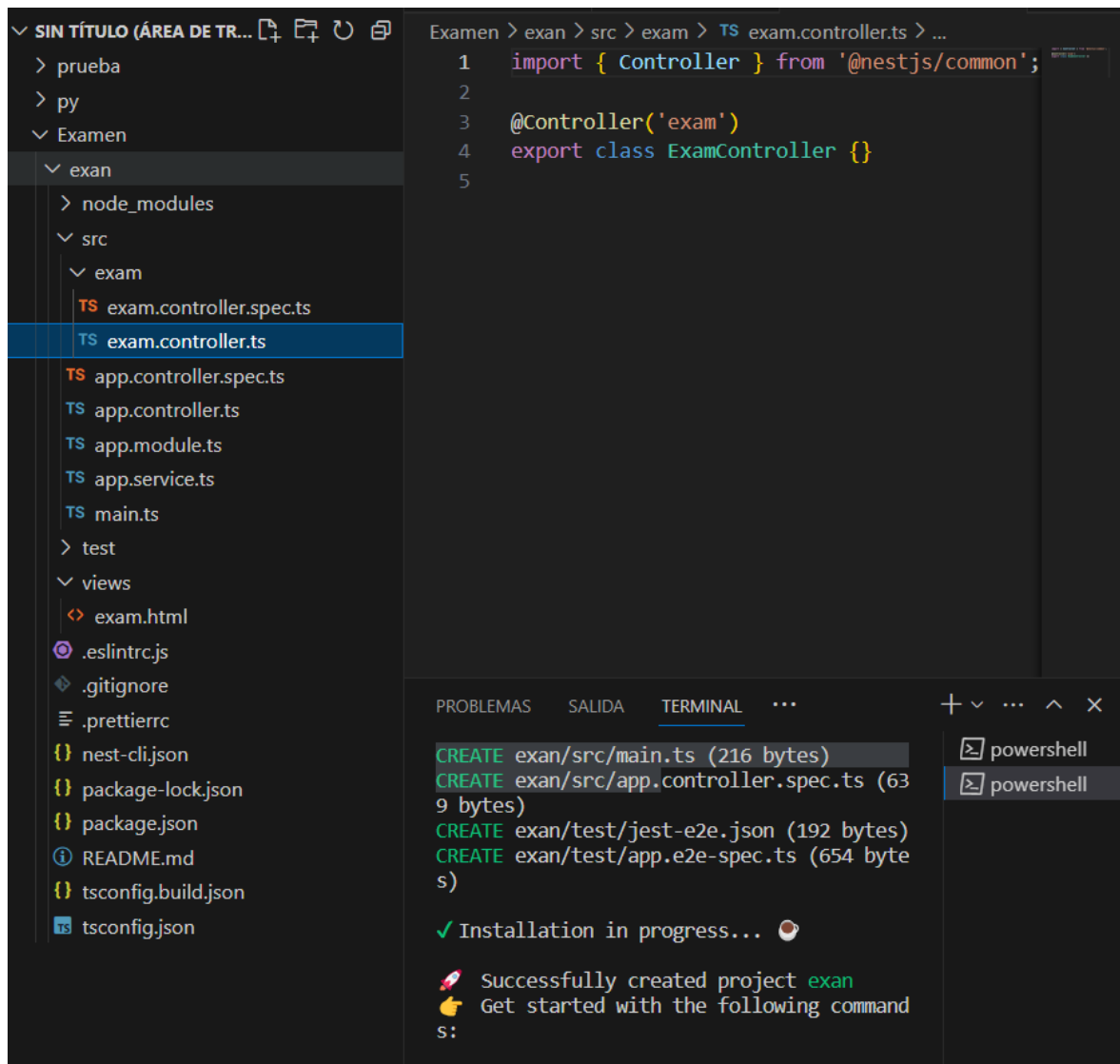
Cambiamos estas direcciones e importaciones

```
import { NestFactory } from "@nestjs/core";
import { AppModule } from "./app.module";
import { NestExpressApplication } from "@nestjs/platform-express";
import * as ejs from 'ejs';
import { join } from "path";

async function bootstrap() {
  const app = await NestFactory.create<NestExpressApplication>(AppModule);
  app.engine('html', ejs.renderFile)
  app.set('view engine', 'html')
  app.set('views', join(__dirname, '..', 'views'));

  await app.listen(3000);
}
bootstrap();
```

Después nos dirigimos a la carpeta del controlador para cambiar las rutas



Ahora agregaremos “Get y Render” a las importaciones para generar la renderización del html

```
1 import { Controller, Get, Render } from '@nestjs/common';
2 import { get } from 'http';
3
4 @Controller('exam')
5 export class ExamController {
6   @Get()
7   @Render('exam')
8   exam(){
9     return[]
10  }
11 }
12
```

En la carpeta app.module.ts nos debe actualizar a nuestra nueva dirección

```
1 import { Module } from '@nestjs/common';
2 import { AppController } from './app.controller';
3 import { AppService } from './app.service';
4 import { ExamController } from './exam/exam.controller';
5
6 @Module({
7   imports: [],
8   controllers: [AppController, ExamController],
9   providers: [AppService],
10 })
11 export class AppModule {}
12
```

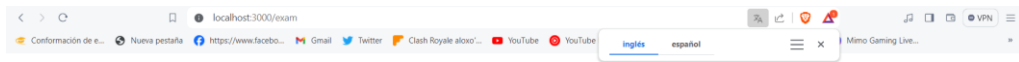
Agregamos algunas cosas a nuestro HTML

```
Examen > exam > views > exam.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <H1>Oda</H1>
10 </body>
11 </html>
```

Ejecutamos

```
[Nest] 12780 - 21/02/2024, 9:19:36 a.m. LOG [NestFactory] Starting Nest application...
[Nest] 12780 - 21/02/2024, 9:19:36 a.m. LOG [InstanceLoader] AppModule dependencies initialized +22ms
[Nest] 12780 - 21/02/2024, 9:19:36 a.m. LOG [RoutesResolver] AppController {/}: +36ms
[Nest] 12780 - 21/02/2024, 9:19:36 a.m. LOG [RouterExplorer] Mapped {/, GET} route +6ms
[Nest] 12780 - 21/02/2024, 9:19:36 a.m. LOG [RoutesResolver] ExamController {/exam}: +1ms
[Nest] 12780 - 21/02/2024, 9:19:36 a.m. LOG [RouterExplorer] Mapped {/exam, GET} route +2ms
[Nest] 12780 - 21/02/2024, 9:19:36 a.m. LOG [NestApplication] Nest application successfully started +5ms
```

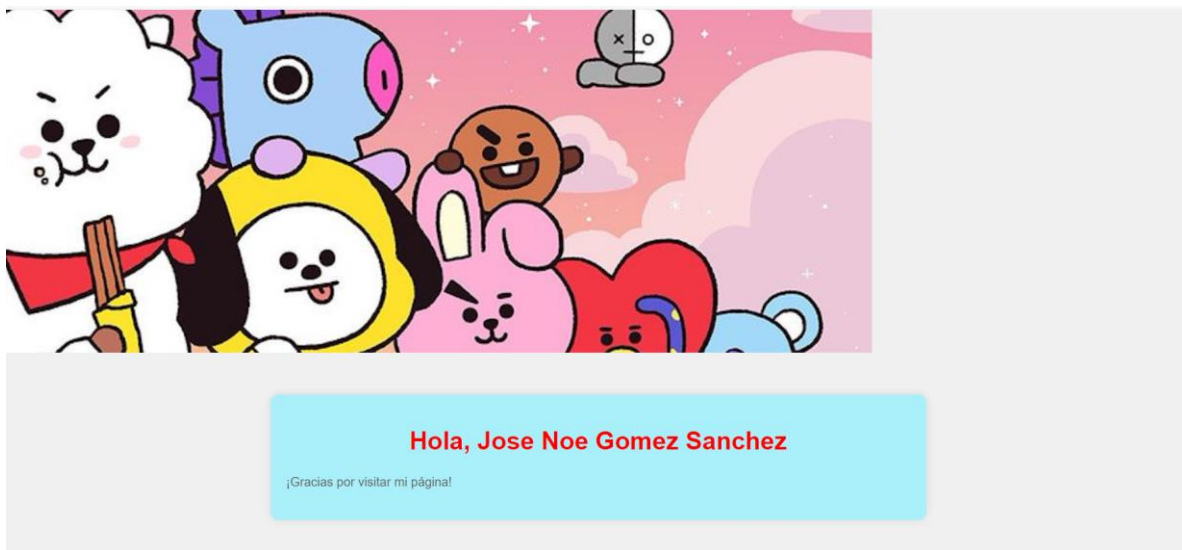
Vamos a nuestra dirección en el navegador



Oda



Damos diseño y Terminamos.



Tuve un error en mi código al usar estilos e imágenes

```
import { NestFactory } from "@nestjs/core";
import { AppModule } from "../app.module";
import { NestExpressApplication } from "@nestjs/platform-express";
import * as ejs from 'ejs';
import { join } from "path";
import * as express from 'express';
import * as path from 'path';

async function bootstrap() {
  const app = await NestFactory.create<NestExpressApplication>(AppModule);
  app.engine('html', ejs.renderFile);
  app.set('view engine', 'html');
  app.set('views', join(__dirname, '..', 'views'));
  app.use([express.static(path.join(__dirname, '..', 'views'))]);

  await app.listen(3000);
}
bootstrap();
```