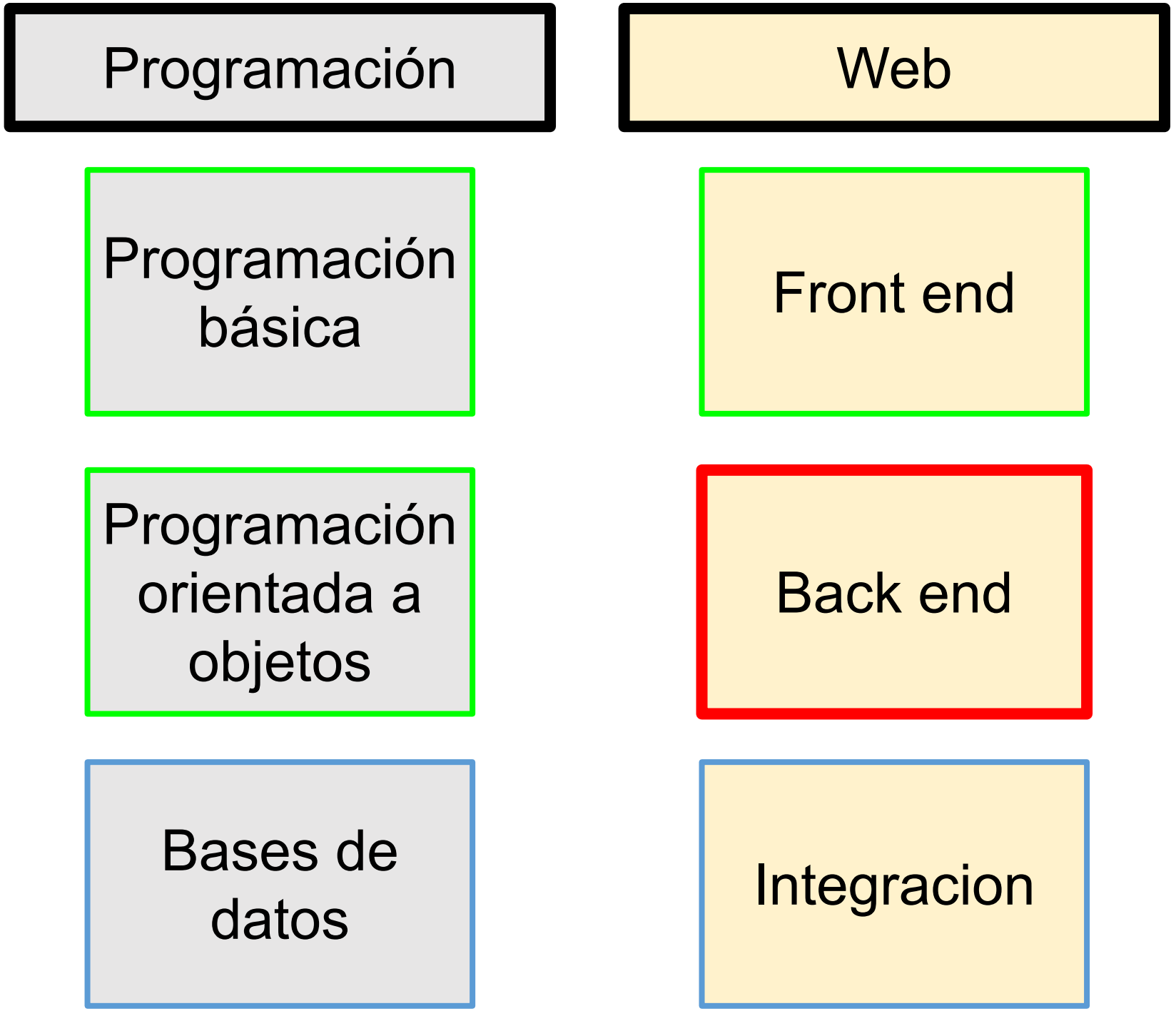
CFP
Programador full-stack


Programación
Web
Programación básica
Front end
Programación orientada a objetos
Back end
Bases de datos
Integracion

¿Qué es un servidor web?

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor realizando:

- conexiones bidireccionales y/o unidireccionales
- conexiones síncronas o asíncronas con el cliente
- generando o cediendo una respuesta

¿Qué hace un servidor web?

- La versión más simple de servidores web sirven archivos estáticos
- Simplemente envía el archivo que le pidieron en un response HTTP.
- Estos archivos son HTML, CSS, JS, imágenes, audio, etc.

DNS


Domain Name Registrar
Examples: GoDaddy, Network Solutions, Register.com, GKG.net
Visitor Request
Example: www.ateamsystems.com
DNS Servers
Examples: NS1.ATEAMSERVERS.COM
NS2.ATEAMSERVERS.COM
NS3.ATEAMSERVERS.COM
NS4.ATEAMSERVERS.COM
Mail Server
Web Server

HTTP

- HTTP: Hypertext Transfer Protocol, ‘protocolo de transferencia de hipertextos’
- Y su variante segura HTTPS
- Los usamos todo el tiempo para navegar por Internet
- Permite hacer pedidos de recursos a un servidor

HTTP y el navegador

El navegador es un cliente HTTP

Solicita archivos al servidor, lee y dibuja esos archivos

Elige que otros archivos descargar (CSS, img, js, etc)

Hay códigos de respuesta:

- 200: OK
- 404: Not found
- 500: Internal server error

Nest: un framework backend

Nest o NestJS es un framework para construir aplicaciones Node.JS eficientes, y escalables

Usa progressive JavaScript y TypeScript

Combina OOP (Object Oriented Programming), FP (Functional Programming), and FRP (Functional Reactive Programming)

https://docs.nestjs.com/

Webserver en Nest

Vamos a hacer un web server de un sitio estático en Nest

Crear el proyecto

Instalar Nest CLI Tool:

```
npm i -g @nestjs/cli
```

Crear proyecto NEST:

```
nest new cfp-demo
```

Entrar a la carpeta y commitear el código inicial:

```
cd cfp-demo
git add .
git commit -m "Initial Project commit"
//conectar un remoto en git y pushear
```

Conectar un remoto en git y pushear

Crear un proyecto en Github

Después:

Opción A:

//se supone que el repositorio está vacío (sin commits)

git remote add origin URL_DE_GITHUB

git push -u origin master

Opción B:

- Hacer un clone
- Mover los archivos ahi (incluir .gitignore)

Iniciar el proyecto

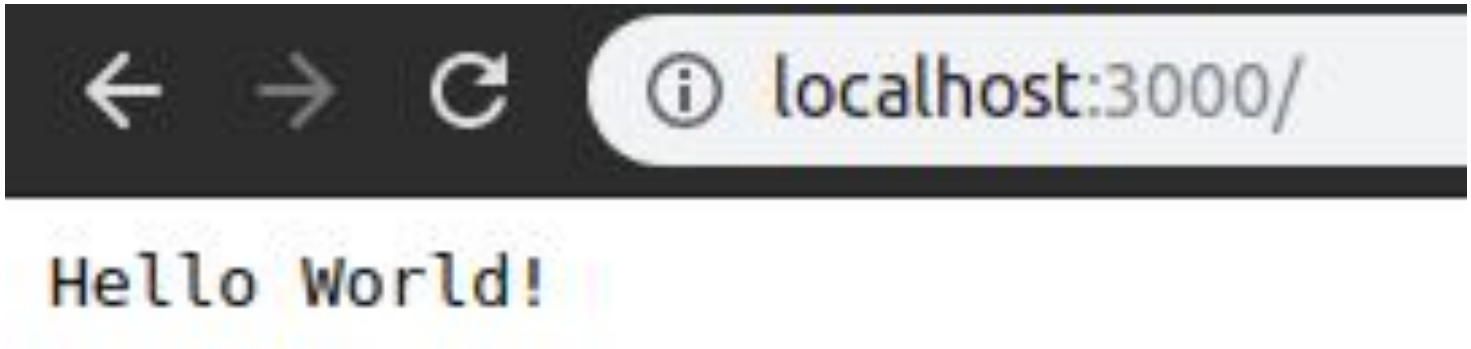
Iniciar el servidor:

```
npm run start:dev
```

Verlo en el navegador

http://localhost:3000

Ver el Hello World



API (para el futuro)

En el archivo app.controller.ts vemos una linea:

```
@Controller()
```

Este archivo hace un “return” que es lo que vemos en el navegador

Esto vamos a usarlo más adelante

Vamos a cambiarla a

```
@Controller('api')
```


localhost:3000/api
Hello World!

¿Qué está pasando?

- localhost: nombre del servidor o URL, en este caso localhost es un alias/apodo para nuestra PC
- 3000: puerto, un “oido” en la PC asignado a un programa en particular
- /api: endpoint, ruta que el servidor NEST sabe que ejecutar

Servir archivos estáticos

```
npm install --save @nestjs/serve-static
```

En app.module.ts agregar en `imports`

```
[
    ServeStaticModule.forRoot({
      rootPath: join(__dirname, '..', 'client'),
    }),
  ],
```

Crear el sitio estático

Crear los archivos estáticos en la carpeta "`client`", a la misma altura de "`src`"

Crear:

- un index.html
- una hoja de estilos en css/estilos.css

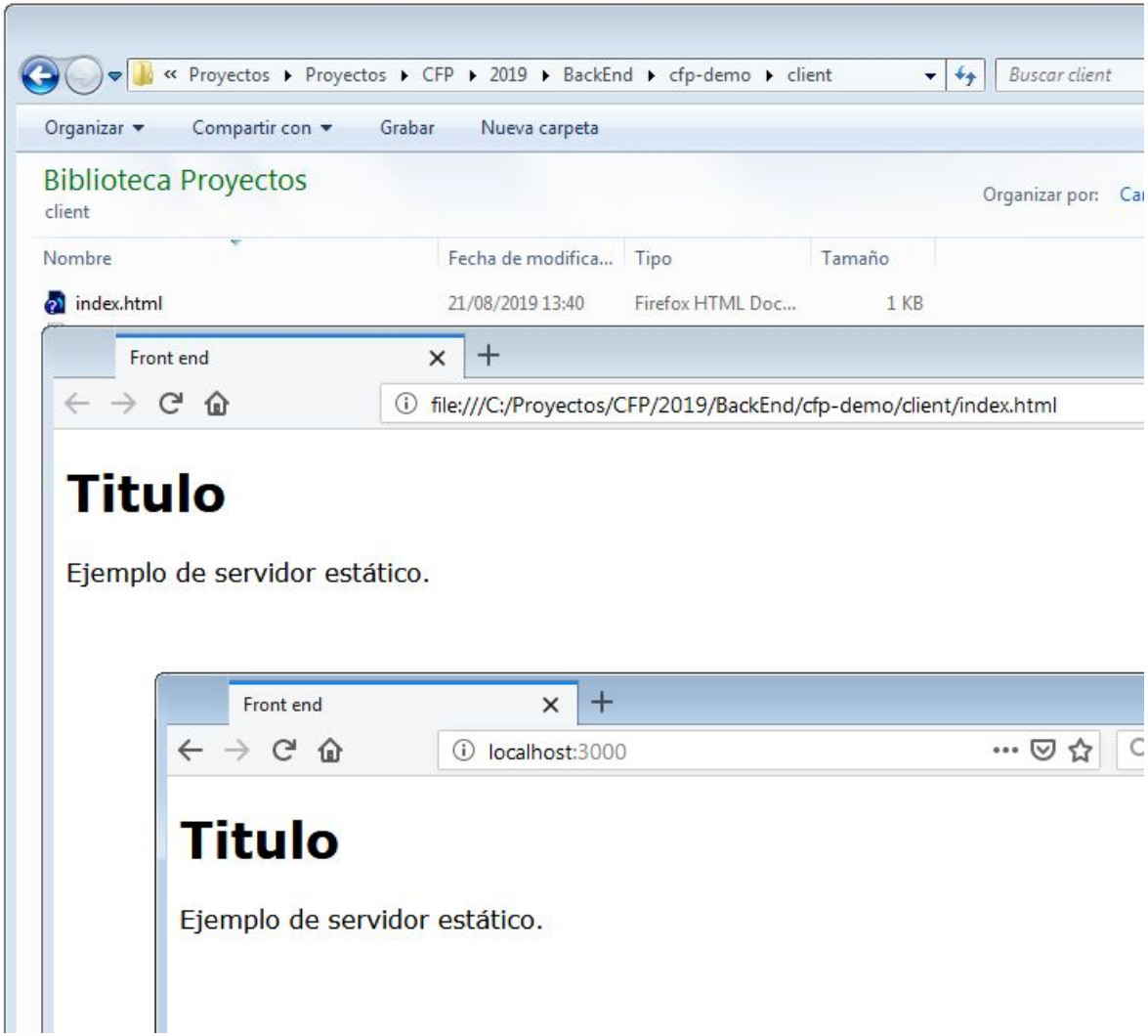
Documentación oficial del framework:

- https://docs.nestjs.com/
- https://docs.nestjs.com/recipes/serve-static

Ejemplo:

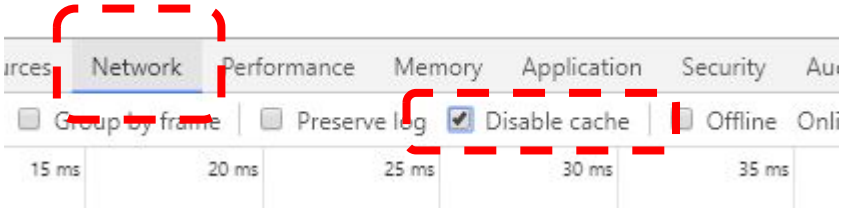
https://github.com/jdottori/nest-cfp-demo/blob/master/cfp-documentation/1-staticfiles.md

Rutas


« Proyectos ▸ Proyectos ▸ CFP ▸ 2019 ▸ BackEnd ▸ cfp-demo ▸ client
Buscar client
Organizar
Compartir con
Grabar
Nueva carpeta
Biblioteca Proyectos
client
Organizar por:
Nombre
Fecha de modifica...
Tipo
Tamaño
index.html
21/08/2019 13:40
Firefox HTML Doc...
1 KB
Front end
file:///C:/Proyectos/CFP/2019/BackEnd/cfp-demo/client/index.html
Titulo
Ejemplo de servidor estático.
Front end
localhost:3000
Titulo
Ejemplo de servidor estático.

Cache y Refresco

- A partir de ahora nuestra página web va a estar en un servidor.
- El navegador puede usar caché para acelerar la navegación, mostrándonos a veces una versión vieja de la página (o de algunos archivos CSS, JS, …)
- Usar siempre Ctrl+F5 para actualizar los archivos para forzar a borrar el cache o con el inspector deshabilitar el cache



CFP
Programador full-stack

Crear un sitio web

Crear un mini sitio web con dos o tres archivos HTML y que se naveguen entre sí