
Notas del Curso de Introducción al Desarrollo Backend

Autor

Wilson Eduardo Jerez Hernández

Profesor

Facundo García Martoni

Platzi

Desarrollo Backend con Go
Curso de Introducción al Desarrollo Backend

Índice

	Página
1. Te doy la bienvenida al desarrollo backend	3
2. Yin y Yang de una aplicación: frontend y backend	3
3. Framework vs. librería	3
4. Cómo se conecta el frontend con el backend: API y JSON	4
5. El lenguaje que habla Internet: HTTP	4
6. ¿Cómo es el flujo de desarrollo de una aplicación web?	4
7. El hogar de tu código: el servidor	4
8. Proyecto: diseño y bosquejo de una API	5
9. Proyecto: diseñando los endpoints de los Tweets	5
10.proyecto: diseñando los endpoints para los usuarios	6
11.Qué lenguaje y framework escoger para backend	6

1. Te doy la bienvenida al desarrollo backend

el mundo esta lleno de aplicaciones web. ellas nacen en la web.

2. Yin y Yang de una aplicación: frontend y backend

la carroceria es el acara visible del auto, y otro componente impotarte es el motor. ahora piensa en aplicaciones. igual que los carros tienen lo mismos componente relativamente”. en tecnologia la correceria: es el frontend y el motor es el backend. python es un lenguaje para onstruir backend. las tecnologias principales del frontend son:

1. html
2. css
3. JavaScript

esos son los basicos. css estilos. JS interacción. html una estrcutura. cada lenguaje tiene sus derivados piensa como si fuera codigo estrito por demas personas. en backend se pueden utilizar lenguajes de progamación como:

1. JavaScript
2. php
3. Java
4. Go
5. Rust
6. Ruby
7. python

y seguramente hay muchos mas. y cada uno de estos lenguajes tienen sus derivados.

3. Framework vs. librería

¿como se construye un auto? comenzamos por el motor, tenemos que empezar por un lado. seguramente hay una guía para hacer el motor, a esto en programación se le llama libreria. las reglas para contruuir el motor, se le llama Framework, en español un marco de trabajo. el Framework usualmente es un conjunto de librerias de paosos y recetas para contruuir un motor.

4. Cómo se conecta el frontend con el backend: API y JSON

API Y JSON. una API es solo una sección de motor que permite que el frontend se comuniquen con el backend. dos grandes estándares son **SOAP** o simple object access protocol, y **REST** o representational state transfer. este primer estándar mueve la información con **XML** extensible markup language parecido a html. soap a quedado un poco en el olvido :'. por culpa de JSON por ser superior. un JSON es como un diccionario de python. los diccionarios en python son iguales a los objetos en JS.

5. El lenguaje que habla Internet: HTTP

<https://developer.mozilla.org/es/docs/Web/HTTP/Status>

cliente: dispositivos. **servidor:** una computadora del día encendida 24/7 protocolo de transferencia de hipertexto. el cliente hace la petición en idioma HTTP. 200 es ok. dentro del rango de los 400 es que algún recurso no es encontrado. si está en el rango de los 500 hay un error en el código.
tarea investigar los status code

6. ¿Cómo es el flujo de desarrollo de una aplicación web?

el lugar en el que comienzas a trabajar es tu editor de texto. git sistema de control de versiones. browser (navegador). en el servidor se coloca una aplicación para que esté disponible para todos. esto se denomina deploy (pasar de el entorno de trabajo a un repositorio). normalmente se hace un push a github antes. desde github se hace **CI/CD**. este proceso lo que hace es testarlo. y si todo está ok se envía a el servidor. el servidor guarda el código es un servidor. normalmente son de paga pero se pueden conseguir algunas de manera gratuita. la nube es una forma popular de llamar a muchos servidores.

7. El hogar de tu código: el servidor

los servidores están en datacenters. al hecho de guardar una aplicación en un servidor se le denomina hosting. hay distintos tipos de hosting. entendamos 3 conceptos

1. iaaS (infraestructura como servicio). es una opción cuando quieras tener control como la CPU, la RAM y la SSD, esta
 - a) AWS (Amazon Web Services)
 - b) Microsoft Azure
 - c) Digital Ocean (económico)vas a encontrar dos tipos de iaaS en internet
 - a) VPS (Virtual Private Server). para rendimiento superior.
 - b) Shared hosting. alojamiento compartido. es más barato.
2. PaaS (plataforma como servicio). se encarga de actualizar, tu relación. just deploy. opciones:

- a)* google aee engine. servicio de google cloud
 - b)* firebase. un poco mas complejas.
 - c)* heroku.
- 3. sass (software como servicio). no reinventar la rueda. no code. es una aplicacion que un proveedor te presta para que puedas hacer funcionar tu servicio.
 - a)* google docs.
 - b)* slack.
 - c)* wordpress.

hay que elegir una de ellas cuando se publica tu aplicacion

8. Proyecto: diseño y bosquejo de una API

cosas fundamentales twitter, por ejemplo un twit, si fuéramos a crear un clon de twitter, tengo que ser capaz de crear usuarios de publicar twits de actualizar usuarios y actualizar twits, de borrar un usuario y de borrar un twits ademas ver a usuarios y ver twits **API** aplicacion, program, interface. una api se construye de varias maneras por ejemplo en python

1. fastapi
2. Django
3. flask

endpoint/route/path es simplemente una section de la url de tu Proyecto. normalmente seria `http://twiteer.com/api/...` lo que esta despues de dominio seria el endpoint

9. Proyecto: diseñando los endpoints de los Tweets

los endpoints de los Tweets. las reglas que nos proporcionan **CRUD** create Read Update Delete. los endpoints para los Tweets serian:

1. `/Tweets -¿show` all Tweets
2. `/post -¿publish` a Tweet
3. `/Tweets / Tweet.id -¿show` a tweet
4. `/Tweets/Tweet.id/update -¿update` a Tweets
5. `/Tweets/tweet.id/delete -¿delete` a tweet

10. proyecto: diseñando los endpoints para los usuarios

hay dos modelos:

1. Tweets
2. users

a los modelos se los conoce como tablas en sql. cada uno de estos modelos contiene registros. los endpoints para los usuarios serian

1. /users -¿show all users
2. /signup -¿registrer a user.
3. / users/user.id -¿show a user.
4. /user/user.id/update -¿update a user.
5. /user/user.id/delete -¿delete a user.

cada vez que entremos a un endpoint el servidor nos responde con un archivo de JASON es proplabe de lusca de la sigueinete manera:

```
{
  "user.id": 121
  "username": "facmartoni
  "email": "x@x.com"
```

lectura recomendada: <https://elbauldelprogramador.com/buenas-practicas-para-el-diseno-de-una-api-re>

11. Qué lenguaje y framework escoger para backend

python tiene 3 framework:

1. Django. es el mas robusto y mas grande, tienes un panel de administración bastante completo. si vas a trabajar con muchos datos. comunidad enorme.
2. flask. aplicaciones simples, muy flexive. algo personalizable.
3. fast api. mucho mas rapida.

JavaScript:

1. express. simple.
2. nest. complejidad mayor.

php:

1. laravel.es el mas facil.
2. sympony. mas dificil. mas escaleble mas compleja.

java:

1. spring.

Go:

1. gin. por excelencia mas popular.
2. beego. crecimiento.

ruby:

1. rails