‹ BoyaConf ›

https://slides.com/darking360/deck-3-4

BoyaConf

¿Quién eres? 😅

@darking360

Miguel Bolivar 🇻🇪

Ingeniero en Informática ... casi 😢

Trabajo para 🔨

Apploi

Colaboro con:

Gatsby

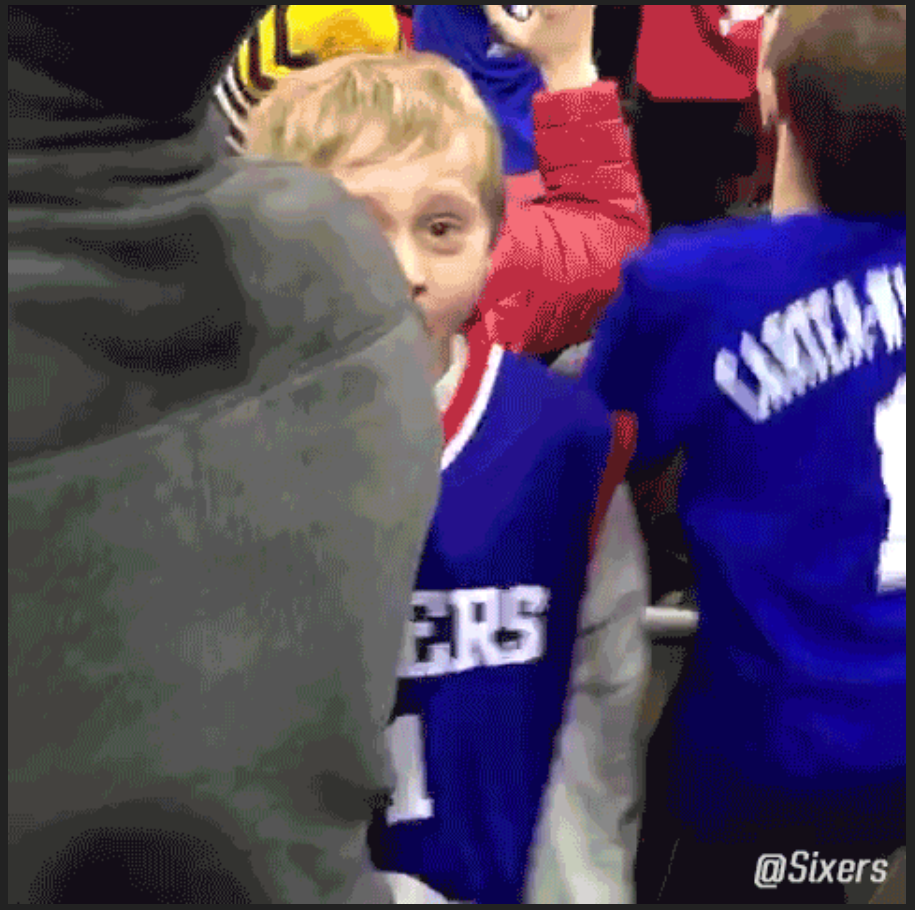
React

Agenda 📚

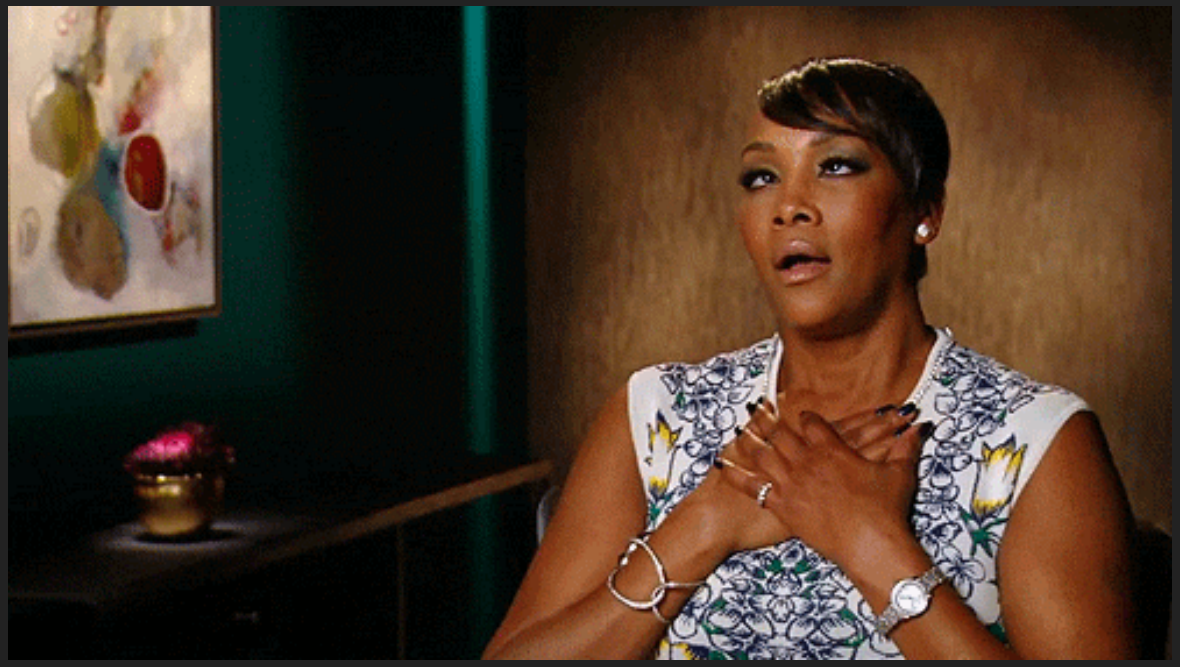
Caso de estudio 🤓

- Empezando en un nuevo trabajo 💪
  - Frontend 💅
  - Backend ⚙️
  - Moralejas 🤔

Nuevo trabajo 💻



Lo logramos 🎉



Proyecto nuevo 😱


AWARD
gifbin.com

Proyecto iniciado 😱

- Funcionando por 3 años 😱
- Mantiene a unos 10.000 usuarios 🙃

¿Cierto? 😅

La historia 🧐

- Decisiones
- Preguntas
- Contexto

⚠️ No es un silver bullet ⚠️

Iniciando la historia 🔥

GIF foto realista de **G** ☝️

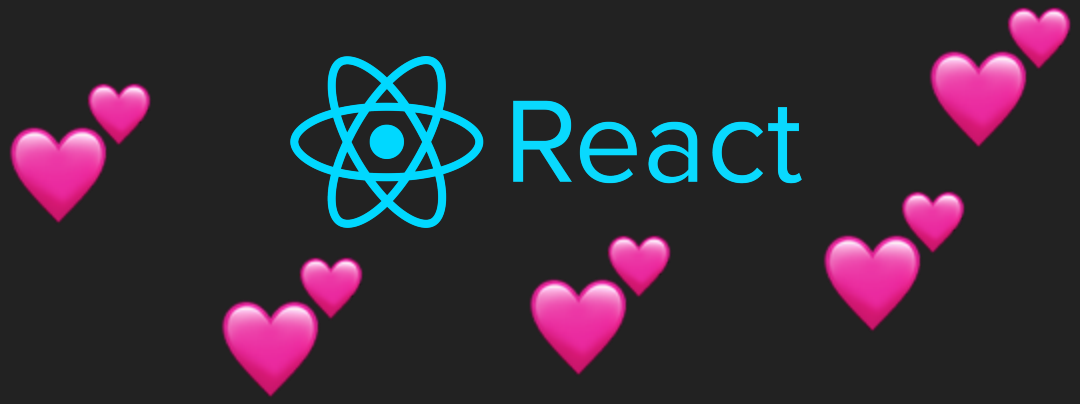
Legacy 💀

Legacy ☠️

Un sistema heredado (o sistema legacy) es un sistema informático (equipos informáticos o aplicaciones) que ha quedado anticuado pero que sigue siendo utilizado por el usuario (generalmente, una organización o empresa) y no se quiere o no se puede reemplazar o actualizar de forma sencilla.

Fuente: Dios Wikipedia ☝️

Continuando la historia 🔥



GIF foto realista de **G** ☝

Empieza la mudanza 🚚

- Componentes
- Programación reactiva

El error 💥

GIF foto realista de cuando vi la busqueda ☝️

Continuando la mudanza 🚚

React Router

Redux

Styled components

Backend ⚙️

GIF foto realista de **K** ☝️

💕

Flask


Then everything changed when the Fire Nation attacked.

Pruebas 💡

- Probar caminos normales
- Probar caminos anomalos 🌪️

Las pruebas buenas, son las que fallan 🤔

Blindando la aplicación

- Pruebas unitarias
- Pruebas end-to-end

Mocks! 

- Sobreescribimos funciones
- Cuándo fue llamada
- Con qué parametros
- Cuántas veces

¿Pero por qué?

A donde vamos 🚋

- Correos
- Firmas
- Almacenamiento externo
- Búsquedas

Separando ⛈️

Qué usamos 🤔

💕

django

```
INSTALLED_APPS  = [
    integraciones1,
    integraciones
    # ... integraciones
]
```

Exito!

Pero todo va creciendo 🏗️

- Más clientes
- Más demanda
- Más trafico
- Más problemas

¿Cómo escalar? 🧗

Delegar 💅

- Workers
- Estado de las tareas
- ¿Qué pasa si van mal?
- Reintentar tareas

Escogiendo 🧐

Servicios de terceros 💀

Celery

Rápido, probado, seguro, confiado por miles 💕

Escogimos MRQ 😅

- Más facil que Celery
- Tablero con estados
- Mecanismos de reintento
- Colas personalizadas 🤯

Colas 🐿️

mongoDB 💕 redis

¿Pero era la suficientemente bueno 🤔 ?

💡 Eureka 💡

❌ Handshake 250 milisegundos a 10 segundos 😱

Por cada solicitud 💀

Encolemos en el API y desencolemos en las integraciones 🔥

Explorando 🗺️

- Rápido entre ambas aplicaciones
- Todos los beneficios de MRQ
- Mecanismos de reintento

Crecer y crecer 👱‍♀️

Yo llegando a la empresa ☝️

Deuda técnica 💸

Es un concepto en el desarrollo de software que refleja el costo implícito de hacer trabajo adicional causado por elegir una solución fácil (limitada) ahora en lugar de utilizar un mejor enfoque que tomaría más tiempo.

Fuente: Dios Wikipedia ☝️

Variables globales 🌐

```
1  var search = {
2    query: "boyaca",
3    page: 10,
4    limit: 20
5  }
6
7  function alterSearch(newParams) {
8      search.query = newParams.query
9      search.page = newParams.page
10     search.limit = newParams.limit
11 }
12
13 // Mas funciones que mutan
14 // a "search"
```

Refactor 😱

Moralejas 🍎

Moralejas del Frontend 🤳

- Componentes
- Programación reactiva
- Composición sobre herencia 🤯

Herencia 👱‍♀️

Define a las cosas o clases, por lo que son 🤔

Composicion 🏗️

Define a las cosas o clases, por lo que hacen 🛠️

Ejemplo 🎮

```
class Player {
  health = 100;

  damage() {/* ... */}
  attack() {/* ... */}
}
```

```
class Pet {
  owner = playerReference;
  health = 100;

  damage() {/* ... */}
  animalAttack() {/* ... */}
}
```

```
class Zombie {
  zone = zoneReference;
  health = 100;

  damage() {/* ... */}
  zombieAttack() {/* ... */}
}
```

```
class CharacterTemplate {
  health = 100;
  damage() {/* ... */}
}


class Player extends CharacterTemplate {/* ... */}
class Pet extends CharacterTemplate {/* ... */}
class Zombie extends CharacterTemplate {/* ... */}
```

Composición 🚀

```
class Player extends CharacterTemplate {
  animalAttack() { /* Meow */ }
  zombieAttack() { /* Braaaaains */ }
  /*
   * Hasta el infinito (pls no)
   * */
}
```

```
function Dialog(props) {
  return (
    <FancyBorder color="blue">
      <h1 className="Dialog-title">
        {props.title}
      </h1>
      <p className="Dialog-message">
        {props.message}
      </p>
    </FancyBorder>
  );
}

function WelcomeDialog() {
  return (
    <Dialog
```

Fuente: Documentacion de React ☝️

Mutabilidad 💀

Redux

En parte 😅

Funciones puras 🔍

```
1 const add = (x, y) => x + y;
```

```
1 let x = 2;
2
3 const add = (y) => {
4   x += y;
5 };
6
7 add(4);
```

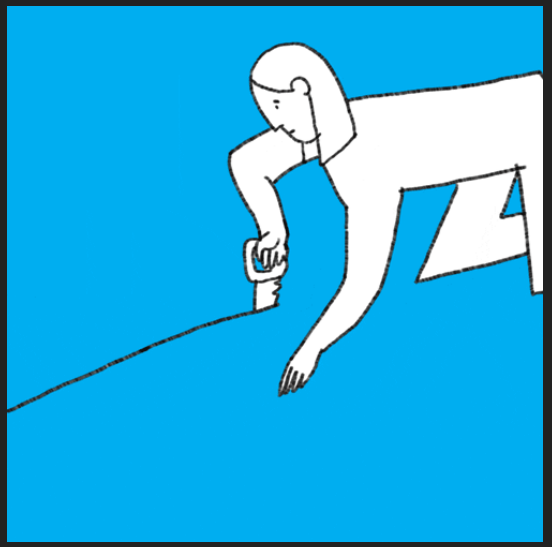
Fuente: Freecodecamp ☝️

Moralejas del Backend 

Flask

¿Por qué separar? 🧐

- ❌ Aplicaciones monolíticas
- ▶ Separación de preocupaciones
- ▶ Una no afecta a la otra



El encolado 🤯

MRQ 🤔 

Mongo y Redis 🤝

Comunicación mediante colas 🌠

Yo cuando la comunicación entre aplicaciones funcionó ☝️

Ahora sí la moraleja 😅

¿En busca de trabajo o empezando? 🙋‍♀️

- Pregunten M U C H O!
- Conozcan el proyecto

Frameworks van y vienen 🍂

Finalmente...

Se requiere algo de ingeniería para servir a 41679 usuarios 🚀

‹ BoyaConf ›

¡Muchísimas gracias! 🙏

Miguel Bolivar 🇻🇪

💕

@darking360