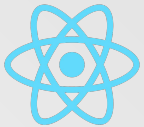
Desarrollo FrontEnd con React



Ingenieria del Software I - 2023

CONTENIDOS

01
JavaScript
Introducción al lenguaje, conceptos fundamentales

02
HTML, JS & Web Pages
Funcionamiento de una página web, html, JS scripting

03
React 101
Introducción a ReactJS, components, props, stateless components, hooks, function components

04
React App
Estructura, compartimentalización de responsabilidades.

05
Manejo de estado
Motivación, SSOT, herramientas

06
Cierre
.

01

JavaScript

“ The World's Most Misunderstood Programming Language ”

Douglas Crockford

JavaScript

Lenguaje interpretado de alto nivel

Creado especificamente para ser embebido en Netscape (circa 1995). Luego del ECMAScript 6, necesita ser “transpilado” para correr.

Principales características:

- Dynamic Typing
- Curly Bracket Syntax
- First Class Functions
- Prototype-based Object-Orientation
- No input/output

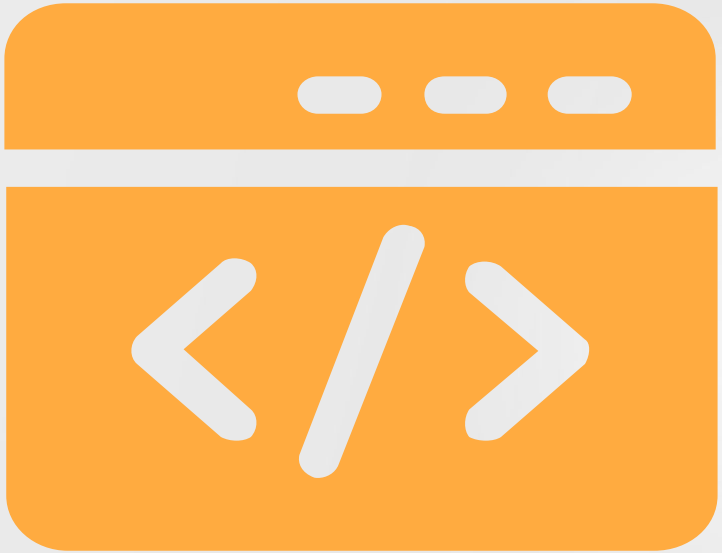
Prototypal Inheritance (objects without classes, prototype delegation)

Functional Programming (lambdas with closure)

02

HTML, JS & Web Pages

Que es una “Página Web”?



HTML estructura

CSS estilo

JS logica

DOM Environment

```
<html>
 <body>
   <h2>Hello World</h2>
   <spam id="countInfo">Count: 0</spam>
   <Button onClick="add()">Add</Button>
 </body>

 <script>
   var count = 0;
   function add() {
     count++;
     document.getElementById('countInfo').innerHTML = 'Count:' + count;
   };
 </script>

</html>
```

```
<html>
 <body>
   <h2>Hello World</h2>
   <spam id="countInfo">Count: 0</spam>
   <Button onClick="add()">Add</Button>
 </body>

 <script>
   var count = 0;
   function add() {
     count++;
     document.getElementById('countInfo').innerHTML = 'Count:' + count;
   };
 </script>

</html>
```

```
<html>
 <body>
   <h2>Hello World</h2>
   <spam id="countInfo">Count: 0</spam>
   <Button onClick="add()">Add</Button>
 </body>

 <script>
   var count = 0;
   function add() {
     count++;
     document.getElementById('countInfo').innerHTML = 'Count:' + count;
   };
 </script>
</html>
```


Microsoft
Microsoft
Windows 95

03

React 101

REACT

Libreria open-source, implementada por Facebook, cuyo objetivo es la creación de interfaces de usuario.

- Representa el View en un modelo MVC
- React está diseñado específicamente para crear componentes simples, que pueden ser combinados para crear aplicaciones web.
- Utiliza su propio DOM virtual, mediante el cual renderiza los componentes a discreción.
- Cada componente es una función que retorna JSX

State - data storing

Props - data handling

```
const Counter = ({ initialCount }) => {
  const [count, setCount] = React.useState(initialCount);

  const add = () => {
    let newCount = count + 1;
    setCount(newCount);
  }

  return (
    <div>
      <span id="countInfo">Count: {count}</span>
      <input type="Button" value="Add" onClick={add}></input>
    </div>
  );
};
```

04

React App

Contact
- id: int
- name: string
- lastname: string
- email: string PK
- phone_main: string
- phone_backup: string
- tags: ManyToMany(Tag) List[Tag]

Tag:
- id: int
- label: string

GET /contacts
Response -> 200 OK -> List[Contact]

GET /contacts?name=<str>&lastname=<str>&email=<str>
Response -> 200 OK -> List[Contact]

GET /contacts/<id>
Response -> 200 OK -> Contact
Errors:
* Id not found -> 404 Not found
* Invalid ID -> 422 Bad Entity

POST /contacts
Response -> 201 Created -> {id: int}
data -> {
"name": str REQUIRED
"lastname": str REQUIRED
"email": str REQUIRED UNIQUE
"phone_main": str REQUIRED
"phone_backup": str OPTIONAL default ""
"tags": List[int] OPTIONAL default []
}
Errors:
* Data Malformed -> 422 Bad Entity
* Email not unique -> 400 Bad request

DELETE /contacts/<id>
Response -> 200 OK -> {id: int}
Errors:
* Id not found -> 404 Not found
* Invalid ID -> 422 Bad Entity

05

State management

REDUX

Librería para el manejo de estado en aplicaciones JavaScript, basada en Flux.

Motivación: El manejo del estado en aplicaciones complejas es exponencialmente difícil.

Single Source of Truth

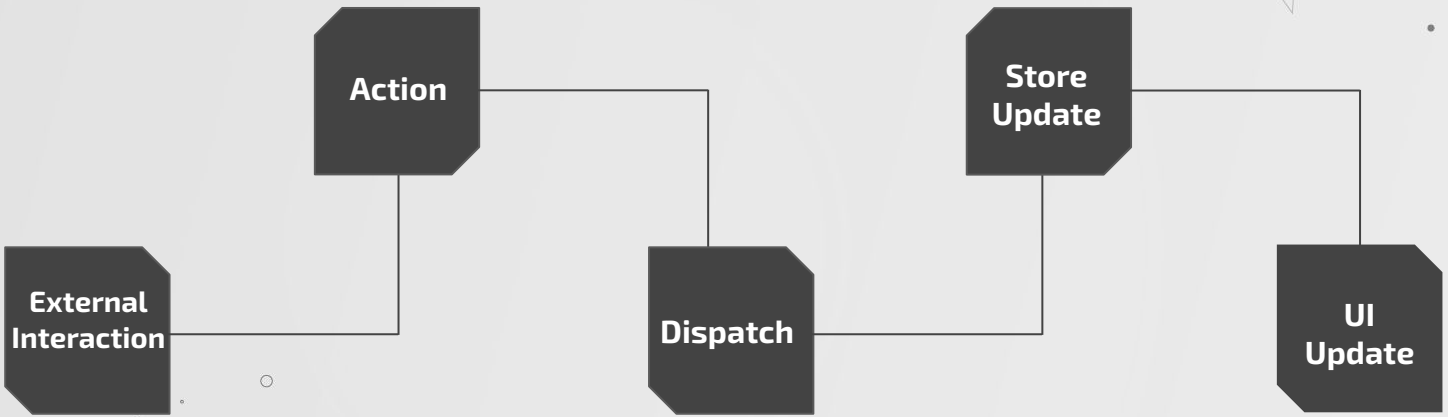
- La información debe estructurarse tal que cada dato es editado en un solo lugar. Cada posible lectura de ese dato es únicamente por referencia, lo que hace que la actualización de dicho elemento se propague completamente a lo largo del sistema.

Core Concepts

- Existe una única fuente de información: store.
- El store es inmutable.
- El store solo puede ser actualizado a través de un pedido explícito al sistema: action.
- Una función pura determina cómo se actualiza el store (reducer).


Action
Representa inequívocamente un cambio sobre un dato del store
Dispatcher
Ejecuta las acciones de manera síncrona
Reducer
Define la manera en que una acción determinada afecta el store
Store
Objeto que representa el estado de la aplicación en un momento arbitrario
App

Store lifecycle



Usuario dispara un evento (onClick)

Se crea una accion (increaseCount)

La acción es emitida por el dispatcher (COUNT_INCREASE)

El reducer ejecuta el update correspondiente

El nuevo valor del store se pasa al componente como una prop

Recursos

JavaScript: The Definitive Guide

MDN Language Overview

React

The World's Most Misunderstood Programming Language