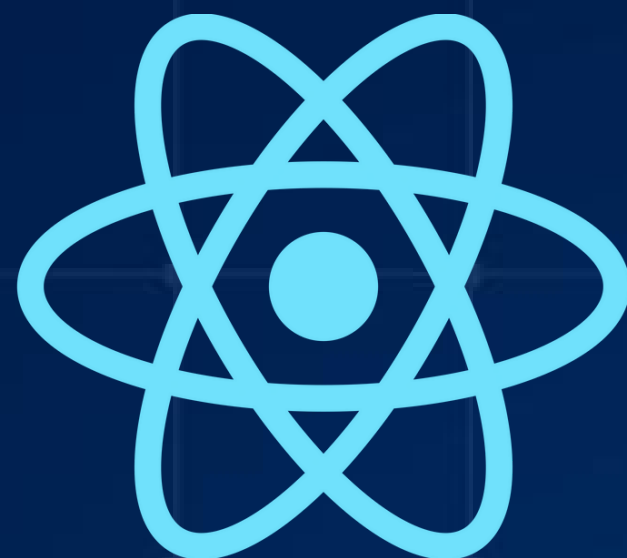


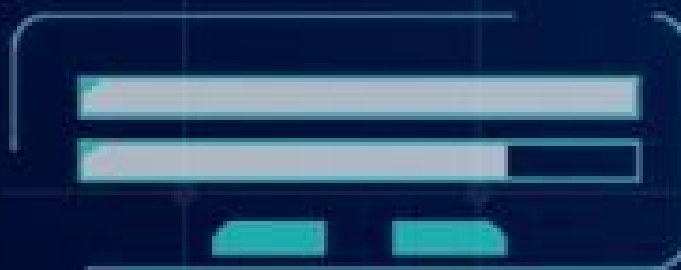


Escuela
Nacional de
Estudios
Superiores

IECAGto
Instituto Estatal de Capacitación



DESARROLLO WEB CON REACT



Instructor

Experiencia previa



Uriel Infante

Full Stack Developer - Inventer



Egresado del Instituto Tecnológico de León



8 años de experiencia en desarrollo web y móvil con cómputo en la nube.



AWS Community Builder



Experiencia de 8 años en el sector Fintech



Objetivo del curso

**Adquirir el conocimiento necesario
sobre desarrollo web y ReactJS
para mejorar mis competencias en
el mundo laboral**

Temario

Fundamentos de desarrollo web

Javascript

Aplicaciones con React

Hooks y navegabilidad

Consumo de Web APIs

Herramientas en la nube

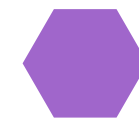
Sitios estáticos y SSR

Aplicaciones en tiempo real

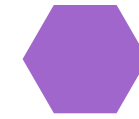


Fundamentos de desarrollo web

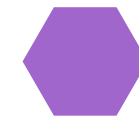
Características del desarrollo web



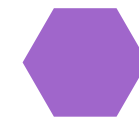
Acceso desde cualquier dispositivo conectado a Internet



No importa el sistema operativo



Facilidad de actualizaciones

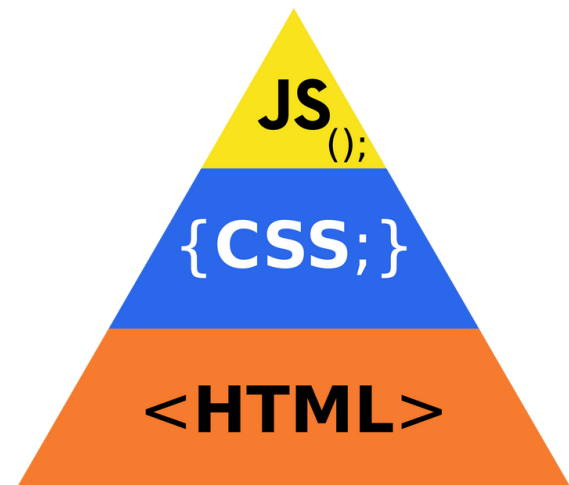


Centralización de información en servidores

Evolución del desarrollo web

Sitios estáticos

Administrados por un webmaster, consistían en HTML puro y poca interacción.



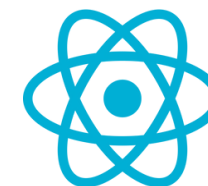
Sitios dinámicos del lado del servidor

Experiencias interactivas, conocida como la web 2.0.



Sitios dinámicos modernos

Se consolida la idea de la separación de back-end y front-end y surgen frameworks del lado del cliente que se conectan a APIs



SSR, SSG, CSR

Generación de sitios estáticos (SSG) y Renderizados del lado del servidor (SSR) nuevamente con la separación de front-end y back-end.



Sitio web estático (tradicional)

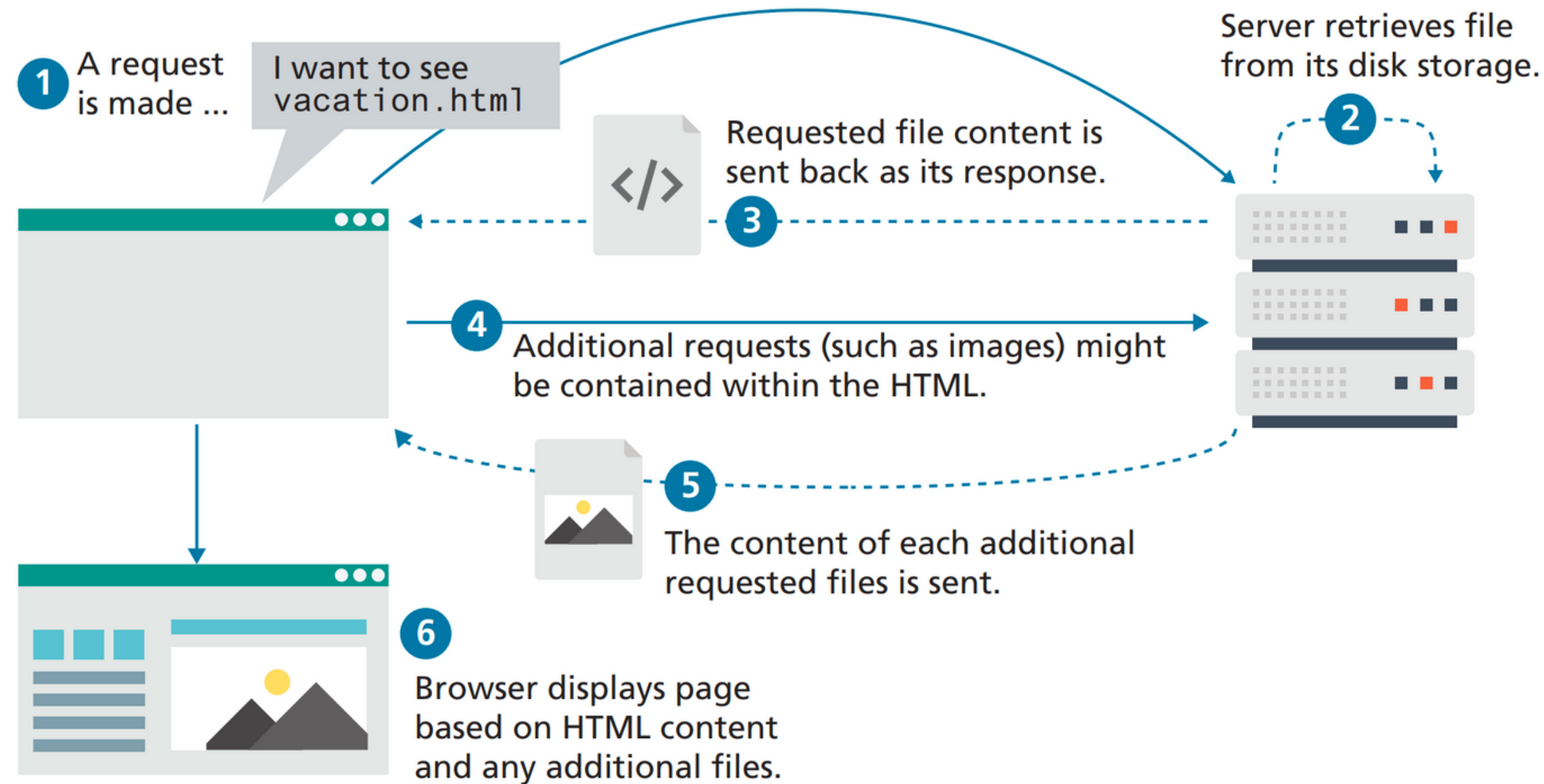


FIGURE 1.6 Static website (first generation)

Sitio web dinámico (tradicional)

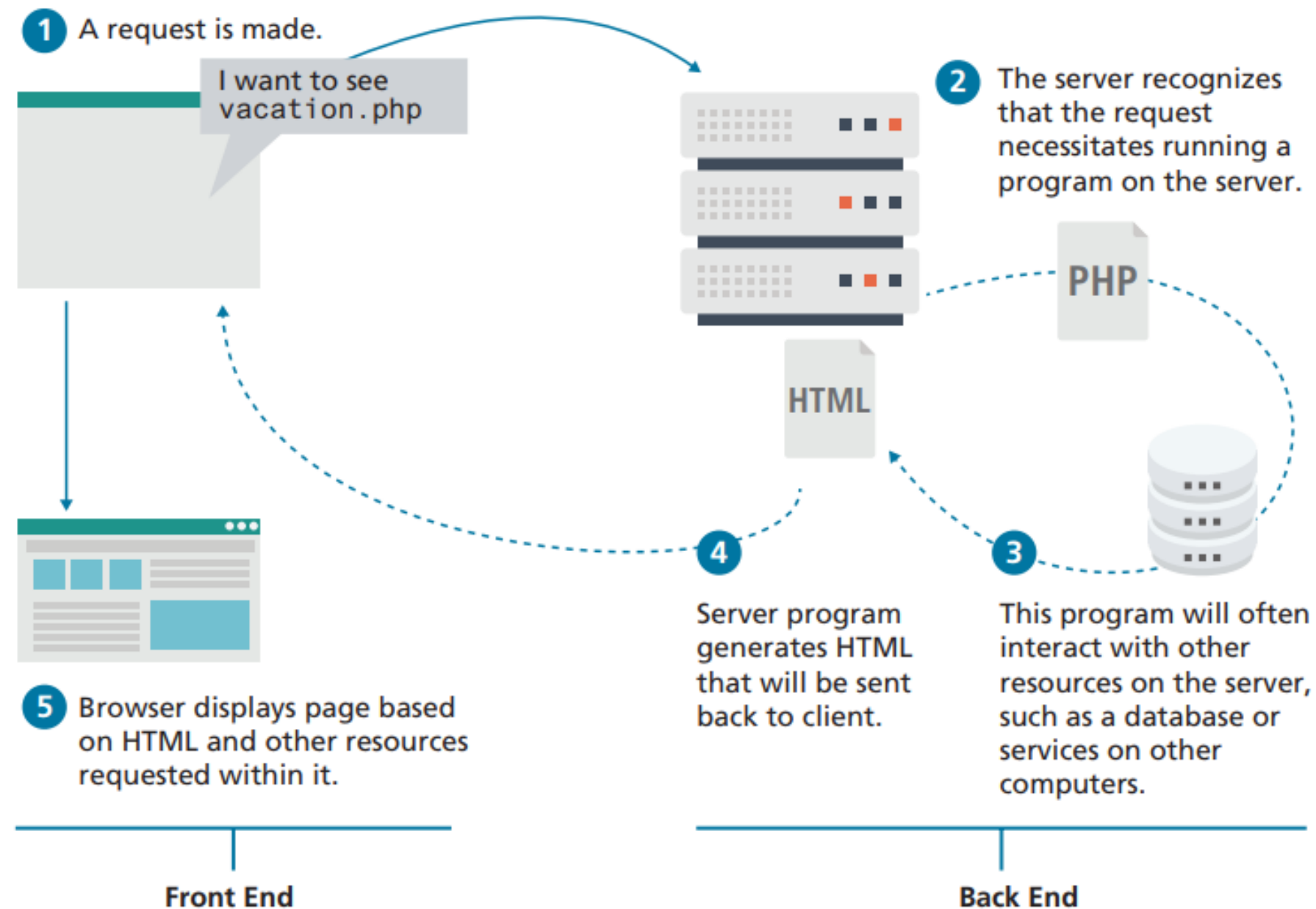
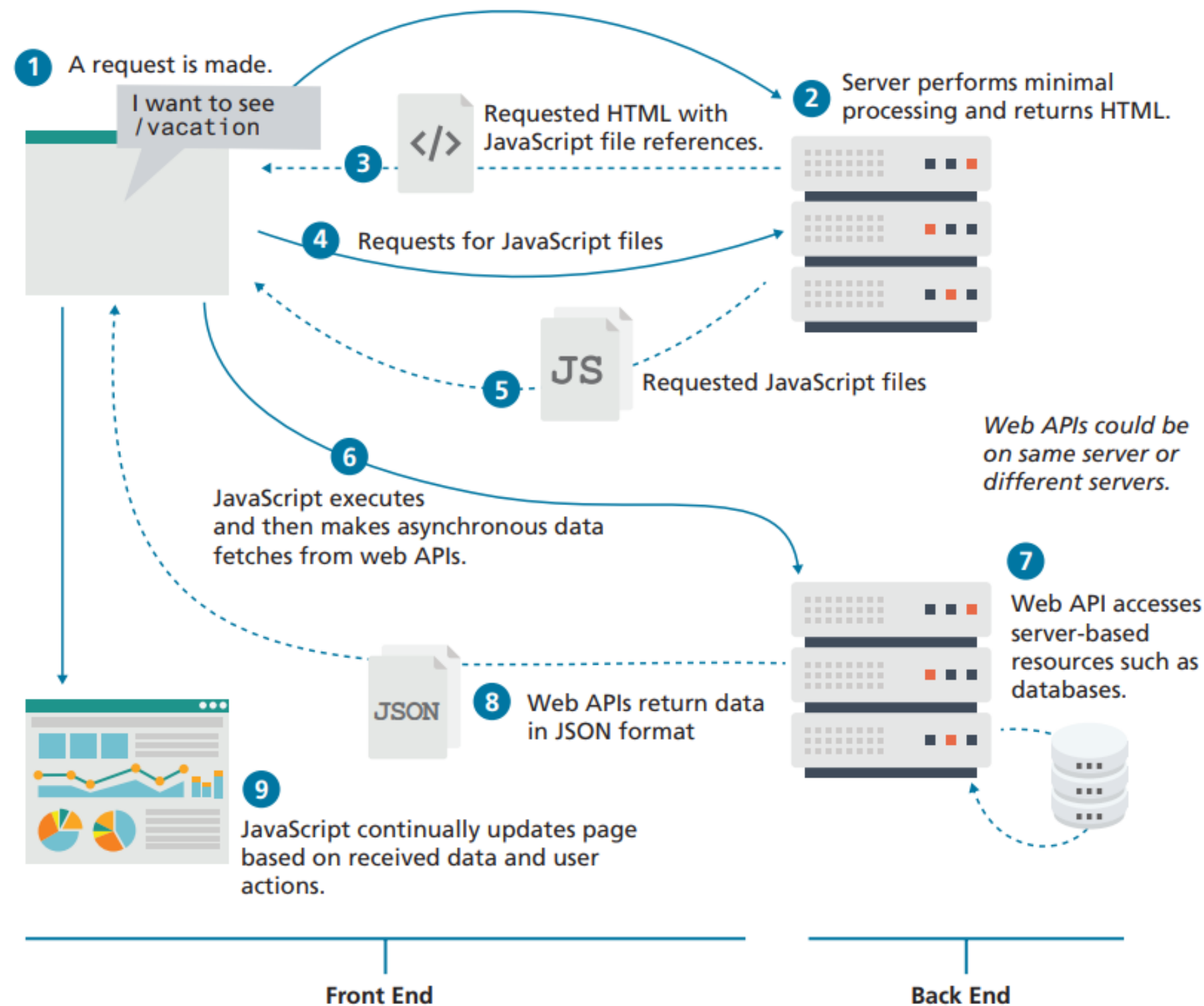


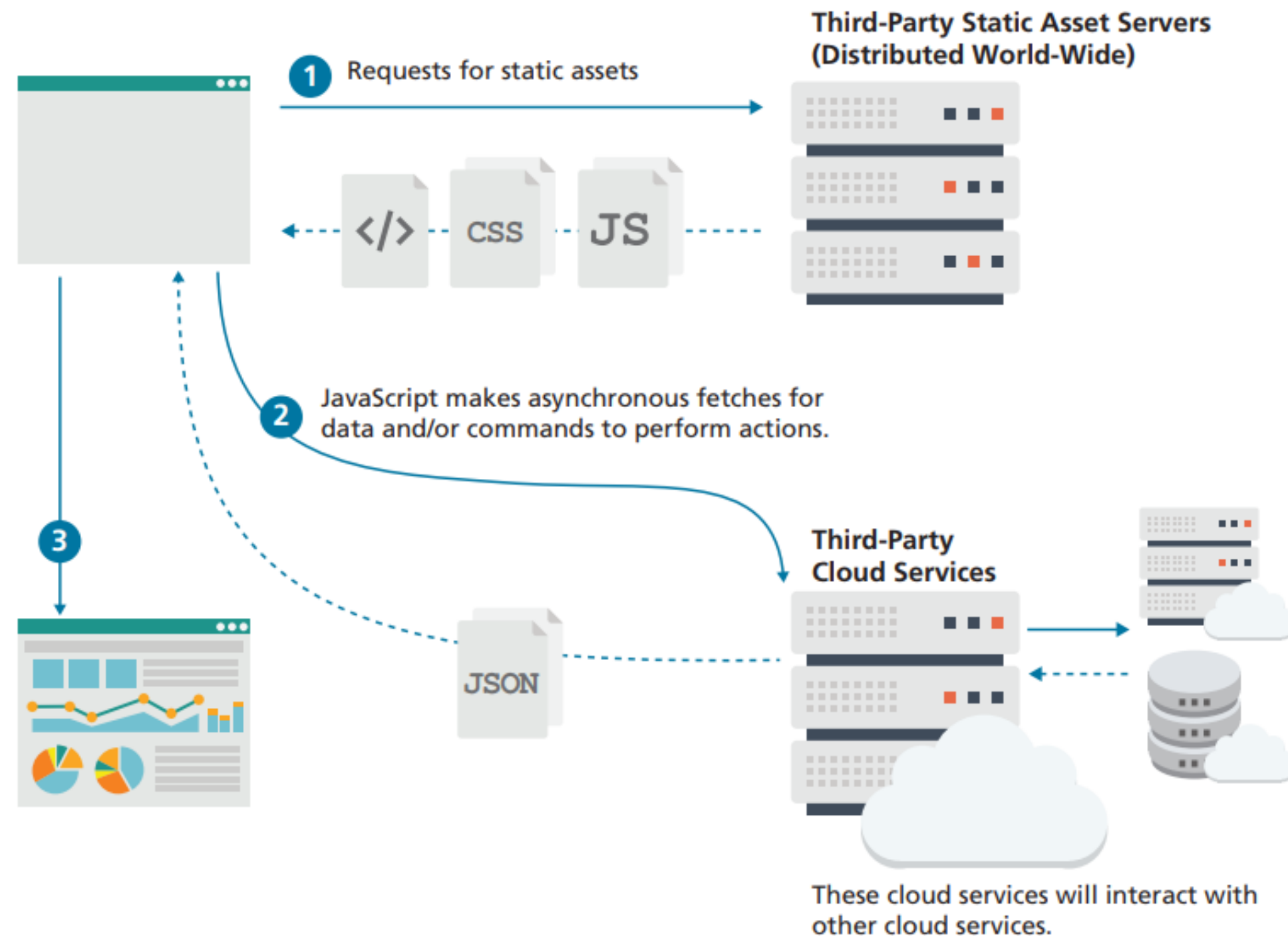
FIGURE 1.7 Dynamic Server-Side website

Sitio web dinámico (moderno)



- Se introdujo la separación de front-end y back-end.
- El servidor original solo devuelve HTML y JavaScript para ejecutar en el sitio.
- El contenido dinámico se genera a partir de llamadas a APIs Web.
- Se establece JSON como formato de comunicación estándar.

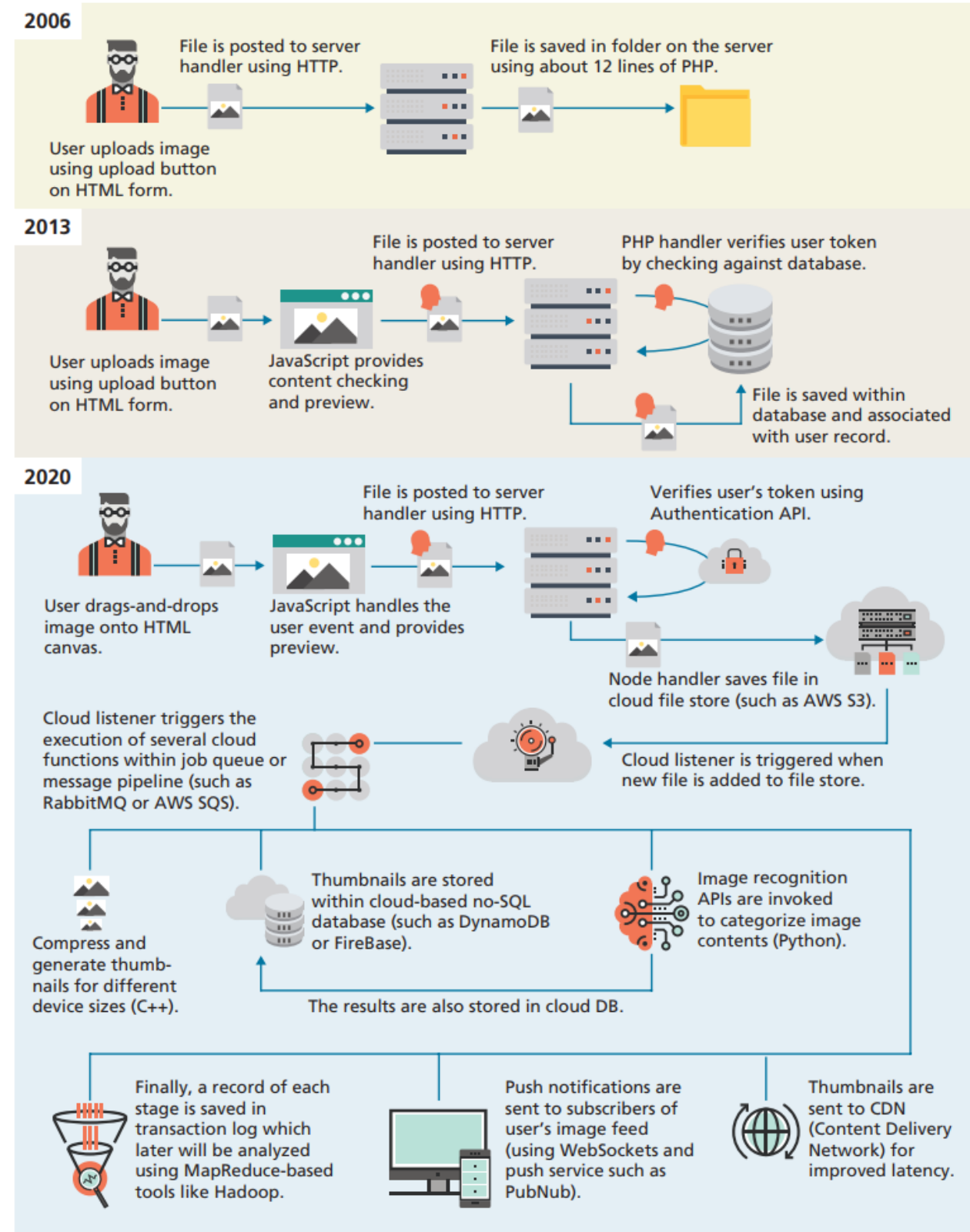
Sitio web estático (moderno)



- Potenciados por el crecimiento del cómputo en la nube.
- Generalmente se distribuyen a partir de redes de contenido como Cloudflare o Amazon Cloudfront.
- El contenido es cargado en tiempo de compilación y entregado al usuario a mayor velocidad.

FIGURE 1.10 Static websites today

Evolución de complejidad en el desarrollo web



Cómputo en la nube

Uno de los avances más importantes en la tecnología ha sido la implementación de cómputo en la nube.

Consiste en el uso de una red de servidores remotos conectados a internet para almacenar, administrar y procesar datos, servidores, bases de datos, redes y software.

Actualmente existen muchas empresas que ofertan tecnología en la nube como Google, Amazon Web Services, Microsoft Azure, etc.



La importancia de Javascript

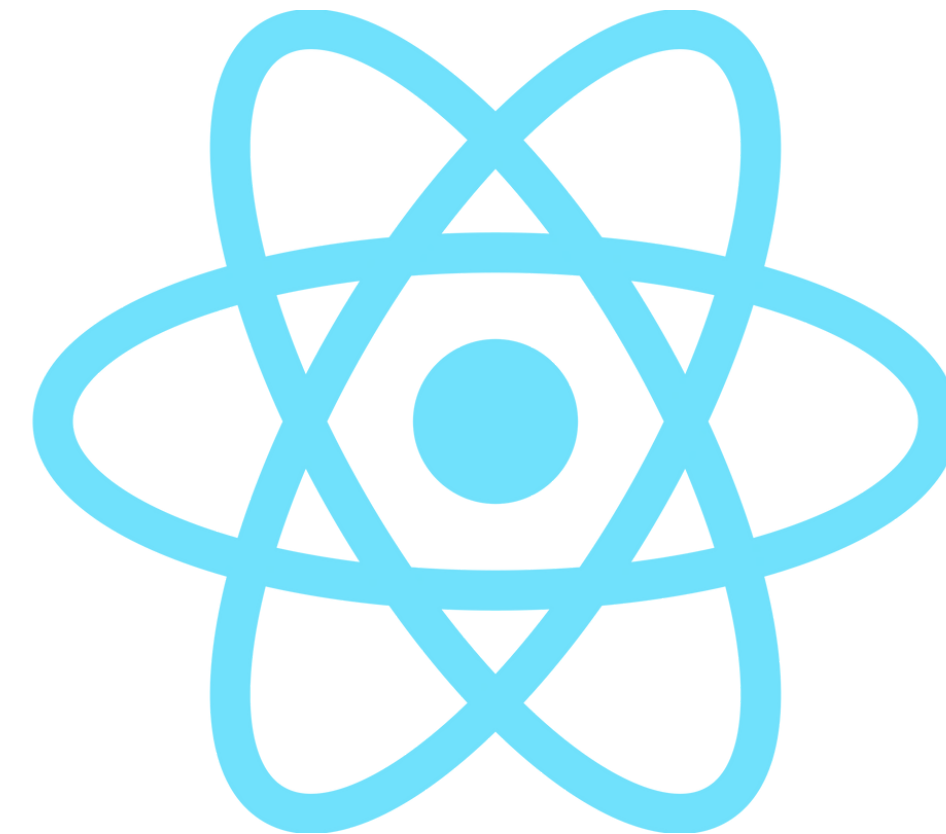
Javascript es uno de los lenguajes más importantes en el desarrollo web moderno por varias razones:

- Interactividad en el lado del cliente.
- Manipulación del DOM.
- Desarrollo de apps completas (front-end, back-end, base de datos, IaaS).
- Comunicación asíncrona.
- Comunidad extendida alrededor del mundo.



¿Qué es React?

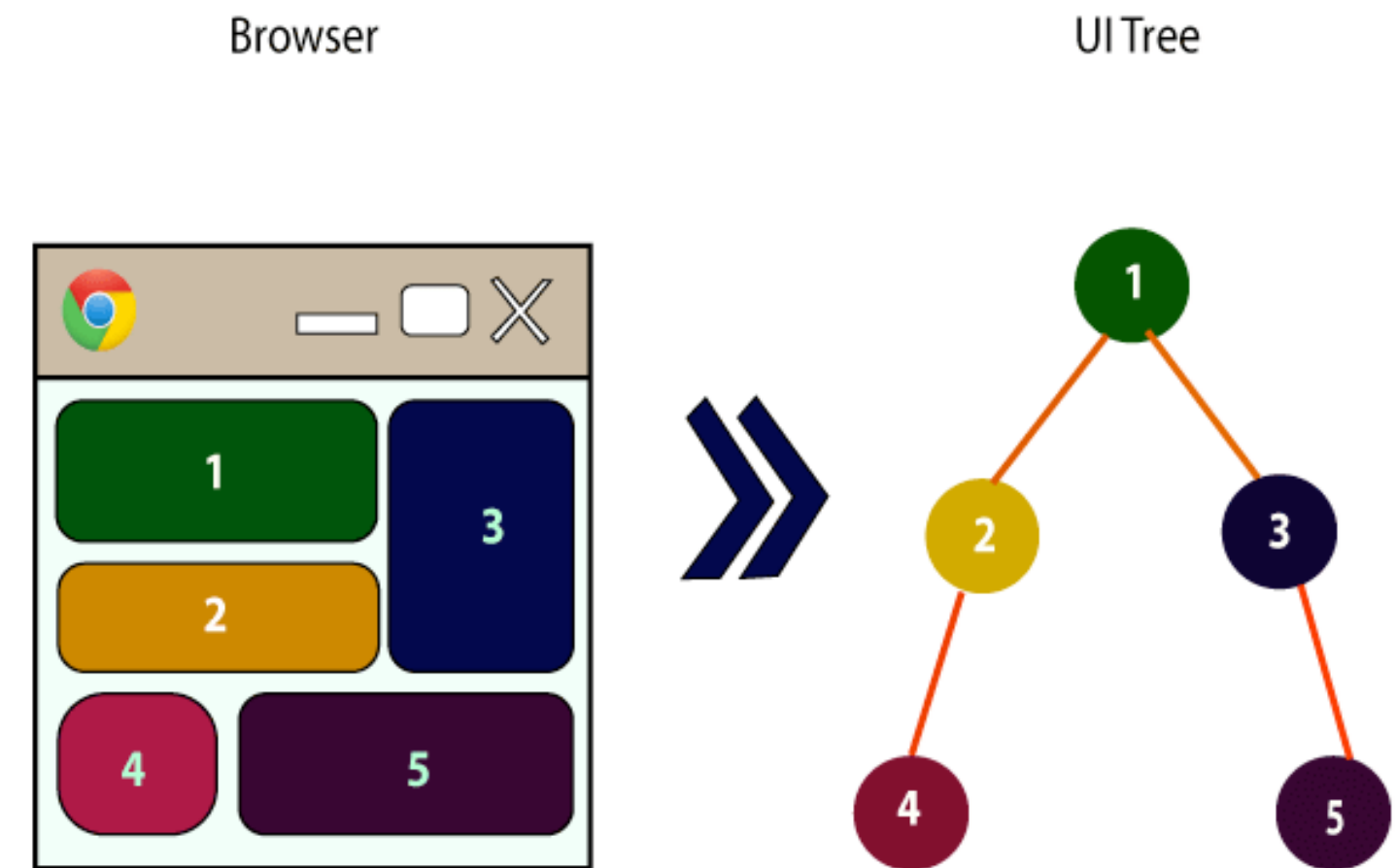
Es una biblioteca de JavaScript de código abierto y mantenida por Facebook. Se utiliza para construir interfaces de usuario (UI) interactivas y reutilizables en aplicaciones web.



Características de React

Componentes reutilizables: Promueve un enfoque basado en componentes reutilizables y autónomos. Cada componente puede tener su propio estado interno y propiedades, y se pueden combinar para crear interfaces de usuario más complejas.

DOM Virtual: Documento de Objeto Modelo Virtual, sirve para optimizar la manipulación y actualización del DOM real. El Virtual DOM es una representación en memoria de la estructura de la interfaz de usuario.



Características de React

JSX: (JavaScript XML), una extensión de sintaxis que permite escribir componentes de React de manera similar a la combinación de JavaScript y HTML. JSX permite mezclar lógica y estructura de la interfaz de usuario en un solo archivo, lo que facilita la comprensión y el mantenimiento del código.

```
11 ReactDOM.render((  
12   <BrowserRouter>  
13     <Switch>  
14       <Route path="/login" component={Login} />  
15       <ProtectedRoute exact={true} path="/" component={Login} />  
16       <ProtectedRoute path="/settings" component={Settings} />  
17       <ProtectedRoute component={Dashboard} />  
18     </Switch>  
19   </BrowserRouter>  
20 ), document.getElementById('root'));  
21
```

Principio de unidireccionalidad: sigue el principio de unidireccionalidad de datos, conocido como flujo de datos unidireccional o patrón unidireccional (one-way data flow). Esto significa que los datos fluyen en una sola dirección, desde el componente principal hasta los componentes secundarios. Esta arquitectura facilita la comprensión y el mantenimiento del estado de la aplicación, evitando problemas de sincronización y actualización no deseados.

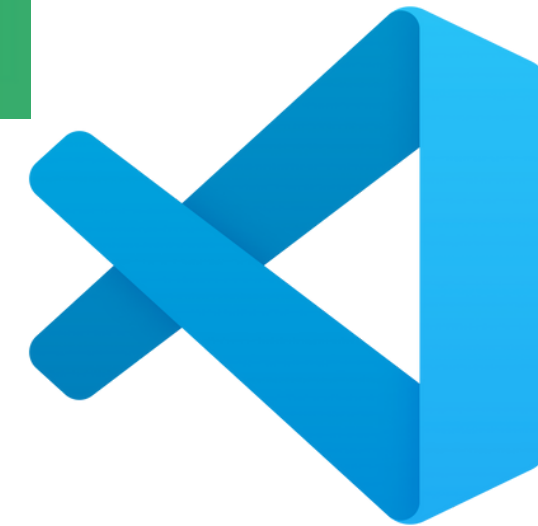
Preparación del entorno



Instalar Node (versión 18.16.0 de preferencia)

<https://nodejs.org/es>

Como alternativa se puede instalar Node Version Manager (NVM)



Contar con un entorno de desarrollo IDE.

- Visual Studio Code
- WebStorm
- Sublime Text
- Vim

Herramientas deseadas



Cuenta de GitHub

<https://github.com>



Práctica - Creación del primer sitio con React

Configuración de la página

Crear un archivo index.html en el cual incluir la siguiente configuración.

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5    <meta charset="utf-8" />
6    <title>Ejemplo de React</title>
7  </head>
8
9  <body>
10   <!-- Target container -->
11   <div id="root"></div>
12   <!-- React library & ReactDOM (Development Version)-->
13   <script src="https://unpkg.com/react@18/umd/react.development.js">
14   </script>
15   <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js">
16   </script>
17   <script>
18     // Pure React and JavaScript code
19   </script>
20 </body>
21
22 </html>
```

Elementos de React

React es una librería diseñada para modificar el DOM evitando la complejidad de hacerlo manualmente.

Nos permite la creación de representaciones del DOM a partir de código.



```
1 React.createElement("h1", { id: "recipe-0" }, "Baked Salmon");  
2
```

ReactDOM

ReactDOM es la biblioteca encargada de renderizar los componentes creados en código como elementos HTML.



```
1  const root = ReactDOM.createRoot(document.getElementById("root"));
2  root.render(React.createElement("h1", { id: "hola-mundo" }, "¡Hola mundo2!"));
```

Componentes dinámicos y con hijos

ReactDOM es la biblioteca encargada de renderizar los componentes creados en código como elementos HTML.



```
1  const form = React.createElement("form", { id: "todo-form", onSubmit: submit }, [  
2    React.createElement("input", { type: "text", id: "todo-input", key: "input" }),  
3    React.createElement("button", { type: "submit", key: "button" }, "Agregar"),  
4  ])
```


Ejercicio sugerido

Usando la biblioteca de React con ReactDOM, crear un sitio que permita capturar recetas de cocina y mostrarlas con actualización de la lista cada que se añade una nueva. Cada receta debe contener al menos los siguientes atributos:

- Nombre de la receta
- Lista de ingredientes
- Tiempo de preparación

Opcional:

- Hacer que la lista persista si recargo la página (Solo con herramientas del navegador).
- Agregar una imagen a cada receta



Escuela
Nacional de
Estudios
Superiores

IECAGto[®]
Instituto Estatal de Capacitación



¡GRACIAS!

