FRONTEND AVANZADO

JAIME BURBANO

SQUARE 1
ACADEMY

Conceptos básicos

SQUARE 1
ACADEMY

Conceptos básicos

- Componentes
- Props drilling
- Composición de componentes
- Manejador de estados
- HTML semántico
- SEO
- Accesibilidad
- Cross browsing
- Rendering

Componentes

Los componentes son bloques de código que encapsulan la estructura interna de los elementos HTML.

HTML + CSS + JS = **Componente**

Conceptos básicos

Componentes

```
<h1>Square1 Academy</h1>

<section>
  <h2>Conceptos básicos del frontend</h2>
    <article>
      <h3>¿Qué es un componente?</h3>
      <p>
        Los componentes son bloques de código que encapsulan la estructura interna de los elementos HTML.
      </p>
      <address>By <a rel="author">Jaime Burbano</a></address>
    </article>

    <article>
      <h3>¿Qué es un Props Drilling?</h3>
      <p>
        Props drilling es una técnica para gestionar el estado de un componente.
      </p>
      <address>By <a rel="author">Jaime Burbano</a></address>
    </article>

    <article>
      <h3>¿Qué es un Composicion de componentes?</h3>
      <p>
        Los componentes de composición pueden ser con estado (stateful) o sin estado (stateless)...
      </p>
      <address>By <a rel="author">Jaime Burbano</a></address>
    </article>
</section>
```

SQUARE1 ACADEMY

Conceptos básicos

Componentes

```
<h1>Square1 Academy</h1>

<section>
  <h2>Conceptos básicos del frontend</h2>
  <article>
    <h3>¿Qué es un componente?</h3>
    <p>
      Los componentes son bloques de código que encapsulan la estructura interna de los elementos HTML.
    </p>
    <address>By <a rel="author">Jaime Burbano</a></address>
  </article>

  <article>
    <h3>¿Qué es un Props Drilling?</h3>
    <p>
      Props drilling es una técnica para gestionar el estado de un componente.
    </p>
    <address>By <a rel="author">Jaime Burbano</a></address>
  </article>

  <article>
    <h3>¿Qué es un Composicion de componentes?</h3>
    <p>
      Los componentes de composición pueden ser con estado (stateful) o sin estado (stateless)...
    </p>
    <address>By <a rel="author">Jaime Burbano</a></address>
  </article>
</section>
```

SQUARE1 ACADEMY

Conceptos básicos

Componentes

```
<h1>Square1 Academy</h1>

<section>
  <h2>Conceptos básicos del frontend</h2>
    <x-article
        title="¿Que es un componente?"
        description="Los componentes son bloques de código que encapsulan la estructura interna de los elementos HTML."
        author="Jaime Burbano"
    />

  <x-article
        title="¿Qué es un Props Drilling?"
        description="Props drilling es una técnica para gestionar el estado de un componente."
        author="Jaime Burbano"
    />

  <x-article
        title="¿Qué es un Composicion de componentes?"
        description="Los componentes de composición pueden ser con estado (stateful) o sin estado (stateless)..."
        author="Jaime Burbano"
    />
</section>
```

SQUARE1 ACADEMY

Conceptos básicos

Componentes

article.blade.php

```
@props(['title', 'description', 'author'])

<article>
  <h3>{{ $title }}</h3>
  <p>{{ $description }}</p>
  <address>By <a rel="author">{{ $author }}</a>
</article>
```

SQUARE 1 ACADEMY

Conceptos básicos

Componentes

Hay dos tipos de componentes

Stateful component

```blade
@php
    $title = "Statefull component";
    $description = "Esto es un statefull component";
    $author = "Jaime Burbano"
@endphp

<article>
  <h3>{{ $title }}</h3>
  <p>{{ $description }}</p>
  <address>By <a rel="author">{{ $author }}</a>
</address></article>
```

Stateless component

```blade
@props(['title', 'description', 'author'])

<article>
  <h3>{{ $title }}</h3>
  <p>{{ $description }}</p>
  <address>By <a rel="author">{{ $author }}</a>
</address></article>
```

SQUARE 1 ACADEMY

Props drilling

Técnica inadecuada de pasar el estado de un componente padre a un ***grandchildren***, donde el componente ***children***, no necesita ese estado.

Conceptos básicos

Props drilling

```
// main.blade.php

@php $message = "Hi, I'm the parent"; @endphp
<main>
    <x-content :message="$message"></x-content>
</main>

-----------------------------------------------------

// content.blade.php

@props(['message'])

<div>
  <p>I'm the child</p>
  <x-message :message="$message"></x-message>
</div>

-----------------------------------------------------

// message.blade.php

@props(['message'])
<div>
  <p> {{ $message }} </p>
</div>
```

SQUARE 1 ACADEMY

Conceptos básicos

Props drilling

```
// main.blade.php

@php $message = "Hi, I'm the parent"; @endphp
<main>
    <x-content :message="$message"></x-content>
</main>

-----------------------------------------------------

// content.blade.php

@props(['message'])

<div>
  <p>I'm the child</p>
  <x-message :message="$message"></x-message>
</div>

-----------------------------------------------------

// message.blade.php

@props(['message'])
<div>
  <p> {{ $message }} </p>
</div>
```

Conceptos básicos

Props drilling

```
// main.blade.php

@php $message = "Hi, I'm the parent"; @endphp
<main>
    <x-content :message="$message"></x-content>
</main>

-----------------------------------------------------
```

```
// content.blade.php

@props(['message'])

<div>
  <p>I'm the child</p>
  <x-message :message="$message"></x-message>
</div>
```

```
-----------------------------------------------------

// message.blade.php

@props(['message'])
<div>
  <p> {{ $message }} </p>
</div>
```

SQUARE1 ACADEMY

Conceptos básicos

Props drilling

```blade
// main.blade.php

@php $message = "Hi, I'm the parent"; @endphp
<main>
    <x-content :message="$message"></x-content>
</main>

-----------------------------------------------------

// content.blade.php

@props(['message'])

<div>
  <p>I'm the child</p>
  <x-message :message="$message"></x-message>
</div>

-----------------------------------------------------

// message.blade.php

@props(['message'])
<div>
  <p> {{ $message }} </p>
</div>
```

SQUARE 1 ACADEMY

Conceptos básicos

Props drilling

Problemas

- Multiples props
- Cohesión
- Baja legibilidad

```blade
// main.blade.php
@php $message = "Hi, I'm the parent"; @endphp
<main>
    <x-content
      :message="$message"
      propOne="prop1"
      propTwo="prop2"
      propThree="prop3"
      propFour="prop4"
      propFive="prop5"
    ></x-content>
</main>
-----------------------------------------------------
// content.blade.php
@props([
  'message',
  'propOne',
  'propTwo',
  'propThree',
  'propFour',
  'propFive'
])

<div>
  <p>I'm the child</p>
  <x-message
    :message="$message"
    :propOne="$propOne"
    :propTwo="$propTwo"
    :propThree="$propThree"
    :propFour="$propFour"
    :propFive="$propFive"
  ></x-message>
</div>
-----------------------------------------------------
// message.blade.php
@props([
  'message',
  'propOne',
  'propTwo',
  'propThree',
  'propFour',
  'propFive'
])

<div>
  <p> {{ $message }} </p>
</div>
```

SQUARE1 ACADEMY

Conceptos básicos

Props drilling

Problemas

- Multiples props
- Cohesión
- Baja legibilidad

```blade
// main.blade.php
@php $message = "Hi, I'm the parent"; @endphp
<main>
    <x-content
      :message="$message"
      propOne="prop1"
      propTwo="prop2"
      propThree="prop3"
      propFour="prop4"
      propFive="prop5"
    ></x-content>
</main>
```

```blade
// content.blade.php
@props([
  'message',
  'propOne',
  'propTwo',
  'propThree',
  'propFour',
  'propFive'
])

<div>
  <p>I'm the child</p>
  <x-message
    :message="$message"
    :propOne="$propOne"
    :propTwo="$propTwo"
    :propThree="$propThree"
    :propFour="$propFour"
    :propFive="$propFive"
  ></x-message>
</div>
```

```blade
// message.blade.php
@props([
  'message',
  'propOne',
  'propTwo',
  'propThree',
  'propFour',
  'propFive'
])

<div>
  <p> {{ $message }} </p>
</div>
```

SQUARE1 ACADEMY

Conceptos básicos

Props drilling

Soluciones

- Re pensar la solución pensada
- Usar slots
- Composición de componentes
- Manejador de estados

```
// main.blade.php
@php $message = "Hi, I'm the parent"; @endphp
<main>
    <x-content
      :message="$message"
      propOne="prop1"
      propTwo="prop2"
      propThree="prop3"
      propFour="prop4"
      propFive="prop5"
    ></x-content>
</main>
```

```
// content.blade.php
@props([
  'message',
  'propOne',
  'propTwo',
  'propThree',
  'propFour',
  'propFive'
])

<div>
  <p>I'm the child</p>
  <x-message
    :message="$message"
    :propOne="$propOne"
    :propTwo="$propTwo"
    :propThree="$propThree"
    :propFour="$propFour"
    :propFive="$propFive"
  ></x-message>
</div>
```

```
// message.blade.php
@props([
  'message',
  'propOne',
  'propTwo',
  'propThree',
  'propFour',
  'propFive'
])

<div>
  <p> {{ $message }} </p>
</div>
```

SQUARE1 ACADEMY

Composición de componentes

Es la composición **lógica** de componentes que permite mantener la usabilidad, modularidad y mantenibilidad de cada componente.

Conceptos básicos

Composición de componentes

CocktailCard

- CocktailTitle
- CocktailMethod
- CocktailImage
- CocktailIngredient

SQUARE 1 ACADEMY

Conceptos básicos

Manejador de estados

Son la forma de almacenar, manipular, guardar y enviar la información.

Manejador de estados

Hay dos tipos de manejadores de estado:

Backend

Frontend

- Local
- Modular
- Global

Conceptos básicos

Manejador de estados

Estado local

```
@php
    $title = "Statefull component";
    $description = "Esto es un statefull component";
    $author = "Jaime Burbano"
@endphp

<article>
  <h3>{{ $title }}</h3>
  <p>{{ $description }}</p>
  <address>By <a rel="author">{{ $author }}</a>
</address>
```

SQUARE1 ACADEMY

Conceptos básicos

Manejador de estados

Estado modular

```
@props(['title', 'description', 'author'])

<article>
  <h3>{{ $title }}</h3>
  <p>{{ $description }}</p>
  <address>By <a rel="author">{{ $author }}</a>
</article>
```

SQUARE 1 ACADEMY

Conceptos básicos

Manejador de estados

Store

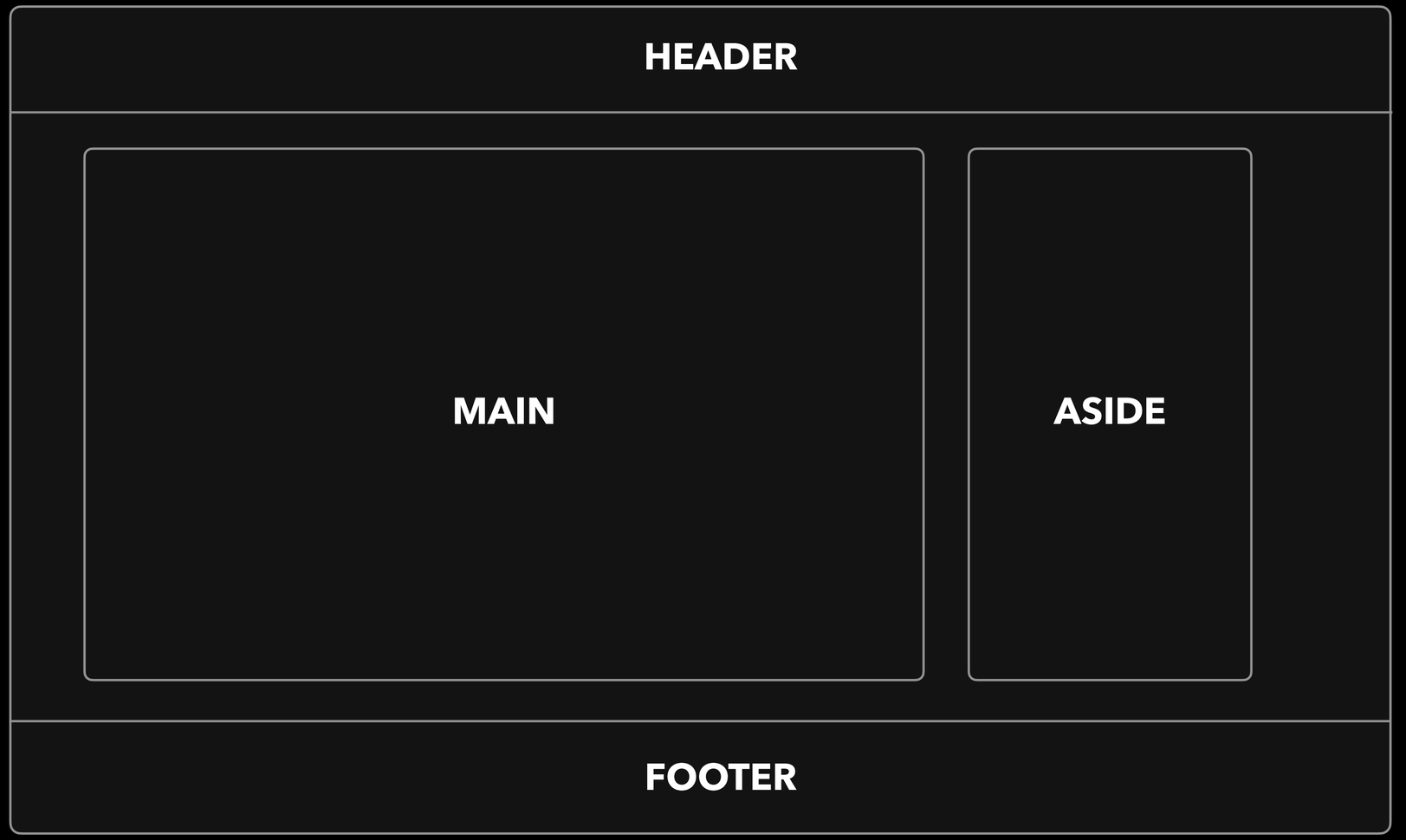
- Información consistente
- Información centralizada


```
{
   data: {
      “name”: null,
      “email”: null
   },
   actions: {
      setName: (newName) => data.name = newName,
      setEmail: (newEmail) => data.email = newEmail
   }
}
```

SQUARE 1 ACADEMY

Conceptos básicos

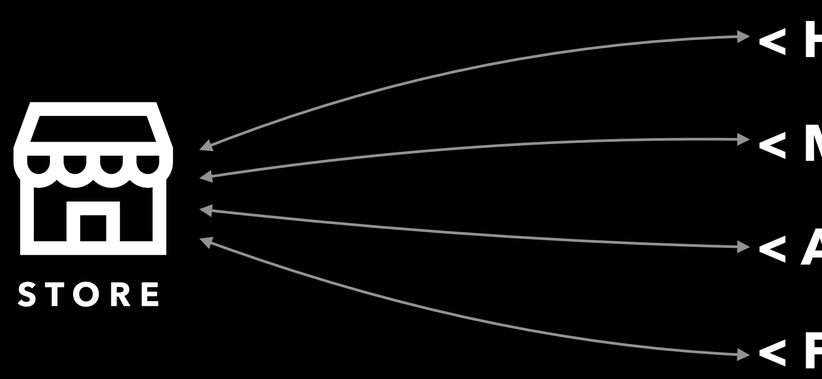
Manejador de estados



SQUARE 1 ACADEMY

Conceptos básicos

Manejador de estados



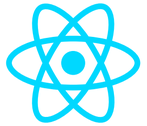
Manejador de estados

Algunos manejadores de estado

ZUSTAND

Mobx

Redux



useContext

X STATE

Conceptos básicos

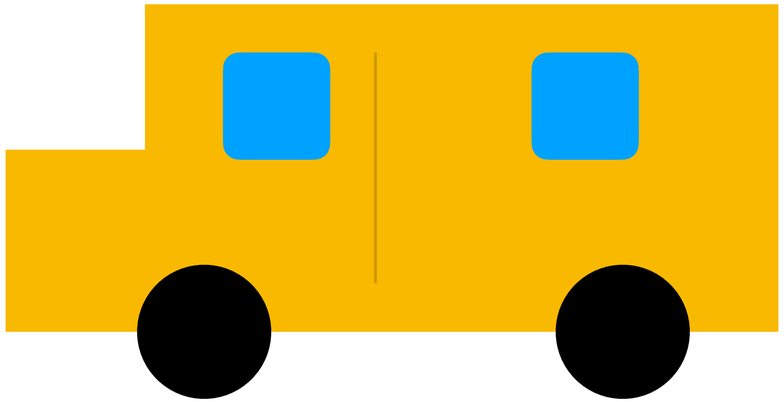
HTML Semántico

Es utilizar las etiquetas de HTML que brindan de **significado** a los elementos de un sitio web.

SQUARE 1 ACADEMY

Conceptos básicos

HTML Semántico



SQUARE 1
ACADEMY

Conceptos básicos

HTML Semántico

SQUARE 1 ACADEMY

Conceptos básicos

HTML Semántico

SQUARE 1
ACADEMY

Conceptos básicos

HTML Semántico

SQUARE 1 ACADEMY

Conceptos básicos

HTML Semántico

```
<div>
  <img src="logo.png" />
  <div>
    <a href="home">Home</a>
    <a href="nosotros">Nosotros</a>
    <a href="servicios">Servicios</a>
    <a href="contacto">Contacto</a>
  </div>
</div>
```

SQUARE 1 ACADEMY

Conceptos básicos

HTML Semántico

```html
<header>
  <img src="logo.png" alt="Square1 Academy logo" />
  <nav>
    <a href="home">Home</a>
    <a href="nosotros">Nosotros</a>
    <a href="servicios">Servicios</a>
    <a href="contacto">Contacto</a>
  </nav>
</header>
```

SQUARE1 ACADEMY

Conceptos básicos

Search Engine Optimization

Conjunto de técnicas y estrategias orientadas a mejorar la **visibilidad** y la **posicionamiento** de un sitio web en los resultados orgánicos (no pagados) de los motores de búsqueda.

SQUARE 1
ACADEMY

Conceptos básicos

Search Engine Optimization

SEO on-page

- Optimización del contenido.
- Estructura del sitio.
- Velocidad de carga.
- HTML semántico.

Conceptos básicos

Search Engine Optimization

SEO off-page

- Enlaces de calidad.
- Menciones en redes sociales.
- Relaciones públicas.
- Participación de comunidades.
- Redirecciones de sitios con autoridad y relevancia.

Conceptos básicos

Search Engine Optimization

SEO técnico

- Rastreo, indexación y rendimiento en motores de búsqueda.
- Archivo robots.tsx.
- Arquitectura del sitio.
- Mapa de sitio XML.
- Velocidad de carga.
- Etiquetas canónicas.

Conceptos básicos

Search Engine Optimization

SEO local

- Google my business.
- Información de contacto en el sitio web.
- Reseñas.
- Testimonios.
- Creación de contenido relevante.

Conceptos básicos

Search Engine Optimization

SEO móvil

- Velocidad de carga en dispositivos táctiles.
- Accesibilidad.
- Optimización de sitios web para dispositivos táctiles.

SQUARE 1 ACADEMY

Conceptos básicos

Accesibilidad

Garantizar que todos los usuarios, independientemente de sus capacidades o limitaciones, puedan utilizar un sitio web de **manera efectiva**

Accesibilidad

Discapacidades o limitaciones

- Auditivas.
- Cognitivas.
- Del habla.
- Físicas.
- Neurológicas.
- Temporales.
- Visuales.

Accesibilidad

Adicionalmente...

- Usuarios de televisores inteligentes.
- Usuarios de relojes inteligentes.
- Personas mayores.
- Limitaciones por ubicación.
- Mala conexión.

Conceptos básicos

Accesibilidad

Beneficios

- Éticas.
- Legales.
- Negocio.

Conceptos básicos

Accesibilidad

Principios de la WCAG

- Perceptible.
- Operable.
- Comprensible.
- Robusto

Calificación

- **A**: Básico
- **AA**: Intermedio
- **AAA**: Avanzado

https://www.w3.org/TR/WCAG21/#perceivable

SQUARE 1 ACADEMY

Accesibilidad

Atributos Aria

Los atributos Aria (Accesible Rich Internet Applications), dotan de significado adicionando información a las etiquetas de HTML.

- Roles ARIA.
- Propiedades ARIA.
- Estados ARIA.

Conceptos básicos

Accesibilidad

Roles Aria

Definen el comportamiento semántico de una etiqueta, cuando la semántica por defecto no es suficiente para describir su función.

SQUARE 1 ACADEMY

Conceptos básicos

Accesibilidad

Propiedades Aria

Agregan información adicional a las etiquetas que no pueden expresarse a través del HTML.

SQUARE 1 ACADEMY

Conceptos básicos

Accesibilidad

Estados Aria

Comunican el estado actual de un elemento, cuando este solamente es evidente visualmente.

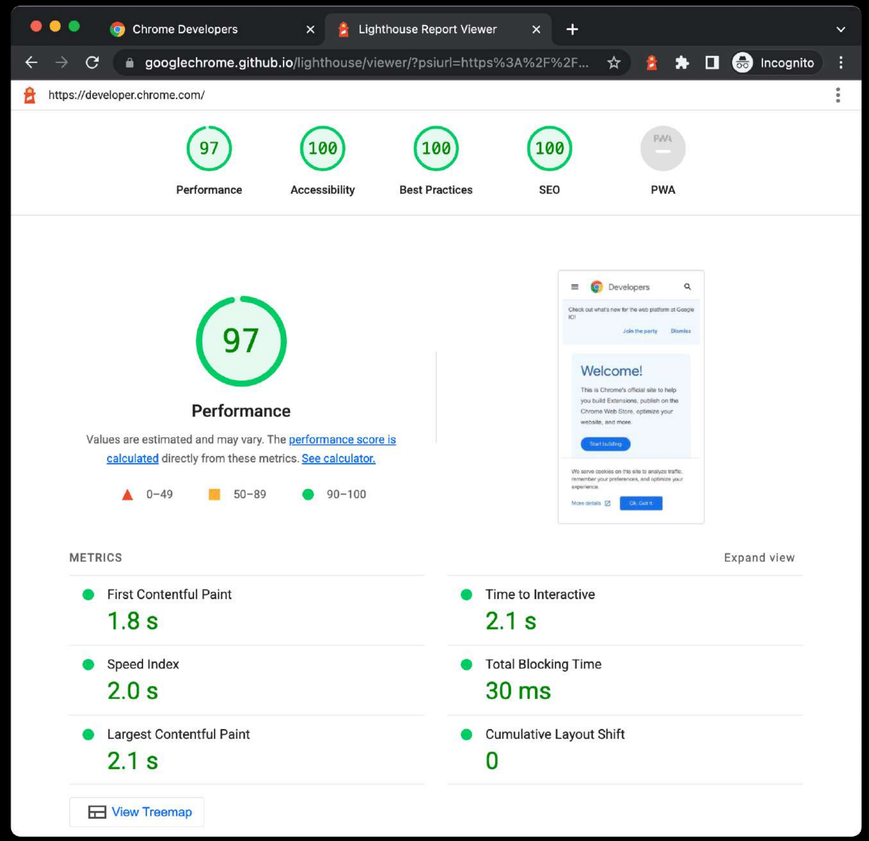
SQUARE 1 ACADEMY

Conceptos básicos

Accesibilidad

Métrica de Lighthouse

La herramienta de google Lighthouse proporciona una evaluación del performance, grado de accesibilidad, mejores prácticas y SEO de un sitio web.



SQUARE 1 ACADEMY

Crossbrowsing

Garantizar que nuestro producto funcione de manera **correcta** y **consistente** en todos los clientes posibles.

Conceptos básicos

Crossbrowsing

Chrominum base

Presto base

Webkit base

SQUARE 1 ACADEMY

Conceptos básicos

Crossbrowsing

SQUARE 1
ACADEMY

Conceptos básicos

Rendering

Proceso de transformación de código a la representación visual que se **monta** en el DOM.

Client Side Rendering

Incremental Site Regeneration

Deferred Side Rendering

Server Side Rendering

Static Site Generation

SQUARE 1 ACADEMY

Conceptos básicos

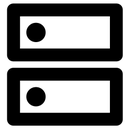
Rendering

Partes que interactúan

Usuario

Navegador

Web



Servidor

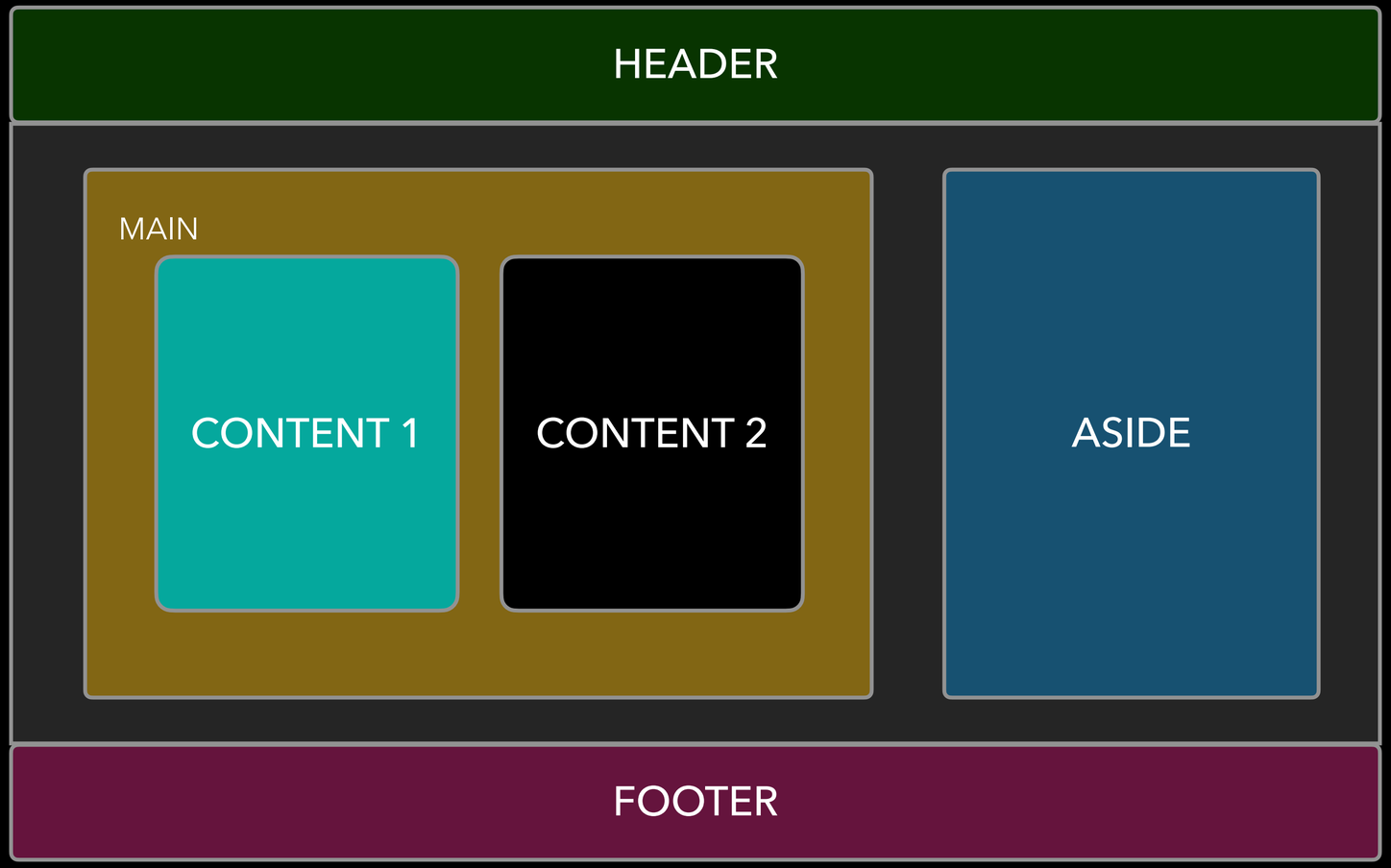
Cache

Proceso

SQUARE 1 ACADEMY

Conceptos básicos

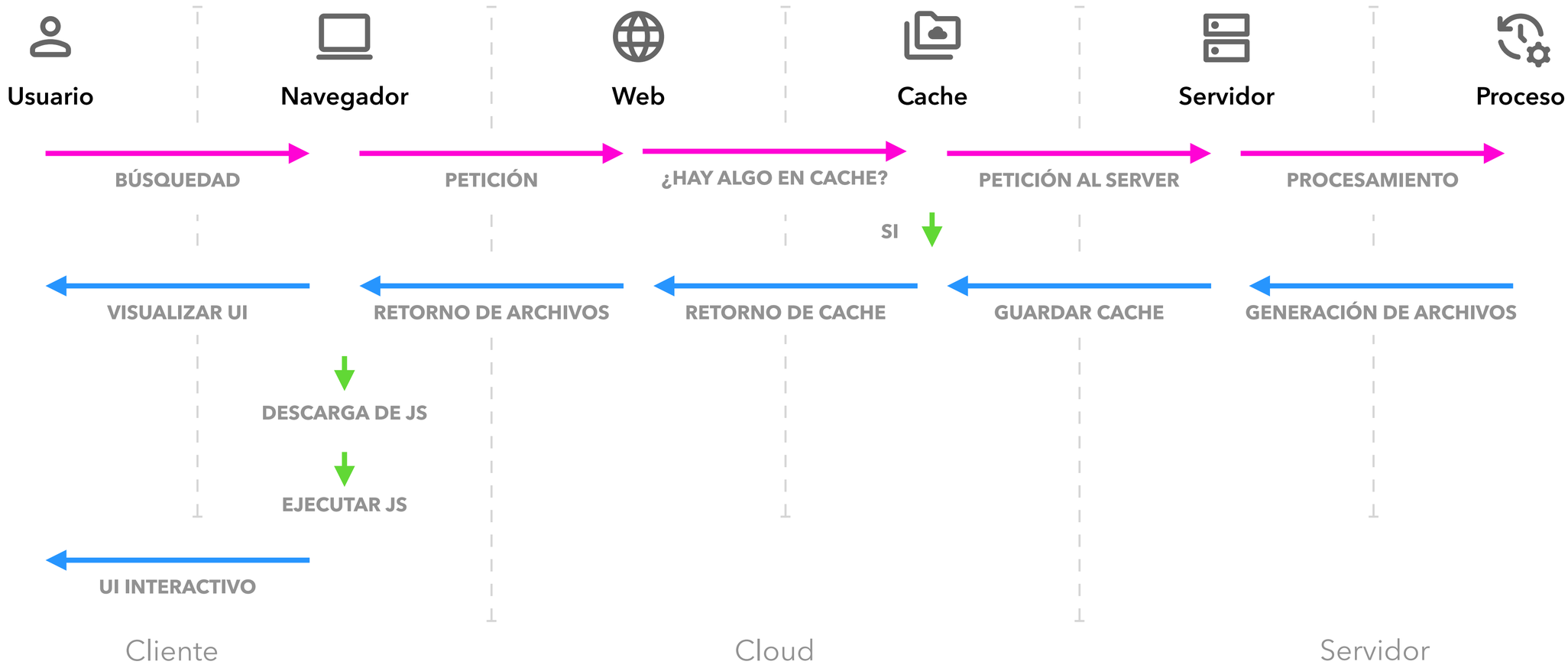
Rendering



Conceptos básicos

Rendering

Server Side Rendering - SSR



SQUARE 1 ACADEMY

Rendering

Server Side Rendering - SSR

Ventajas

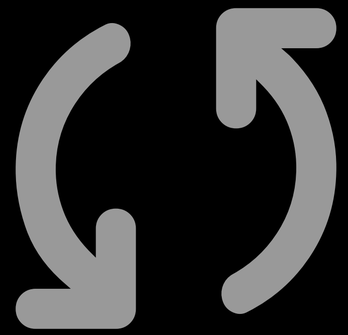
- SEO.
- Rendimiento inicial rápido. *
- Seguridad.
- Crossbrowsing.

Desventajas

- Carga en el servidor.
- Latencia inicial.
- Complejidad de desarrollo.

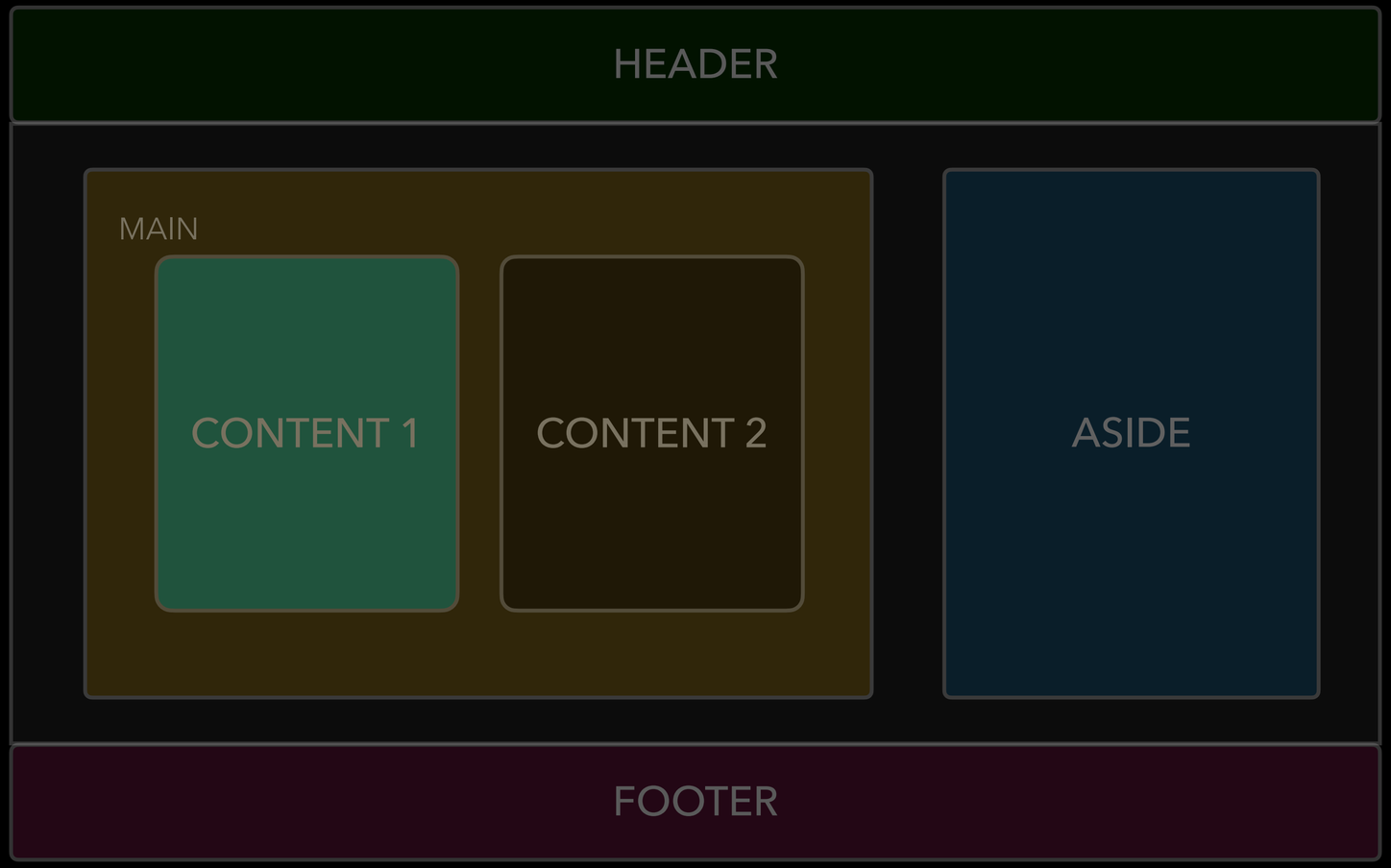
* Rendimiento inicial comparado con el Client Side Rendering y dependiendo de la implementación del SSR.

Rendering

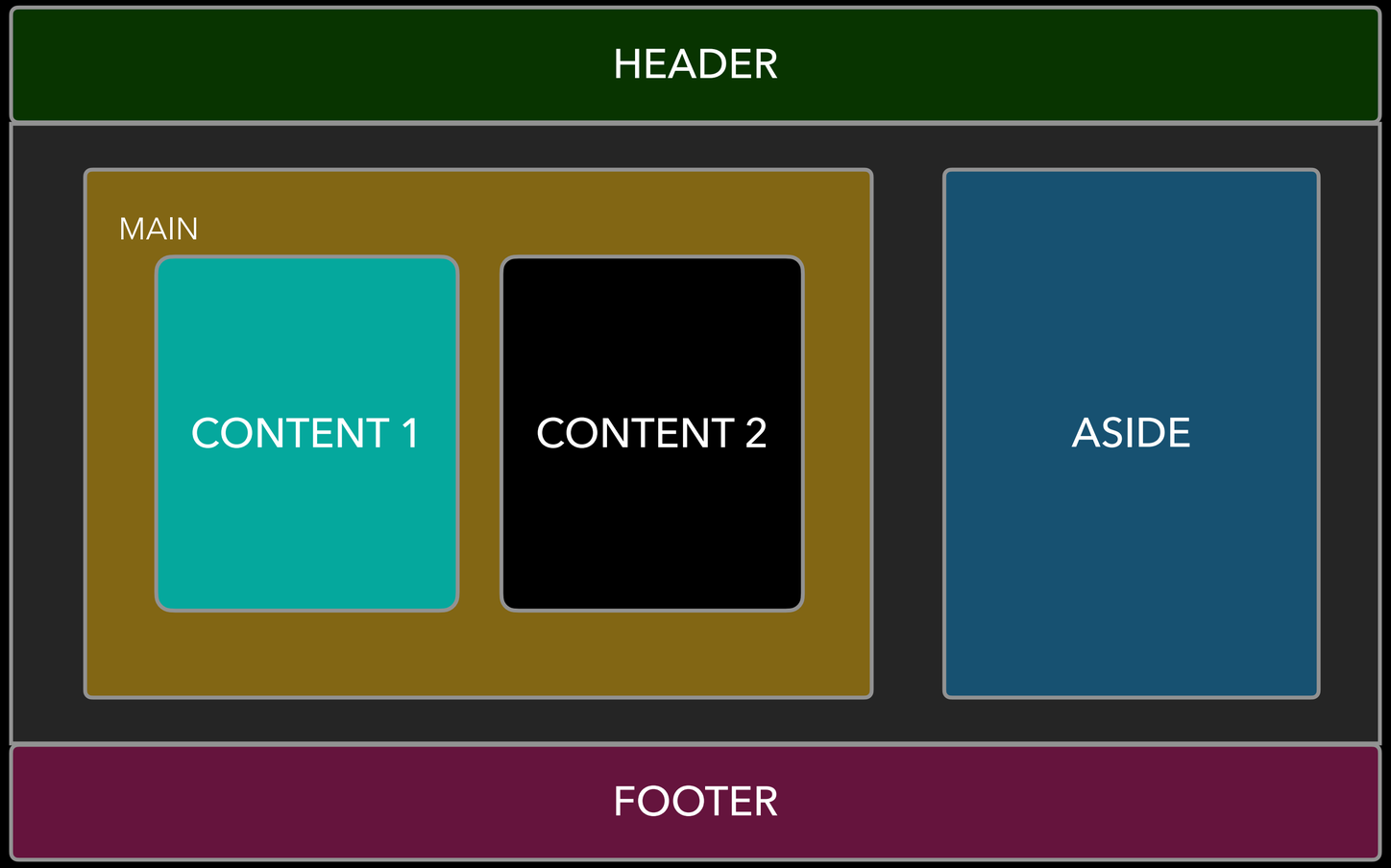


Conceptos básicos

Rendering



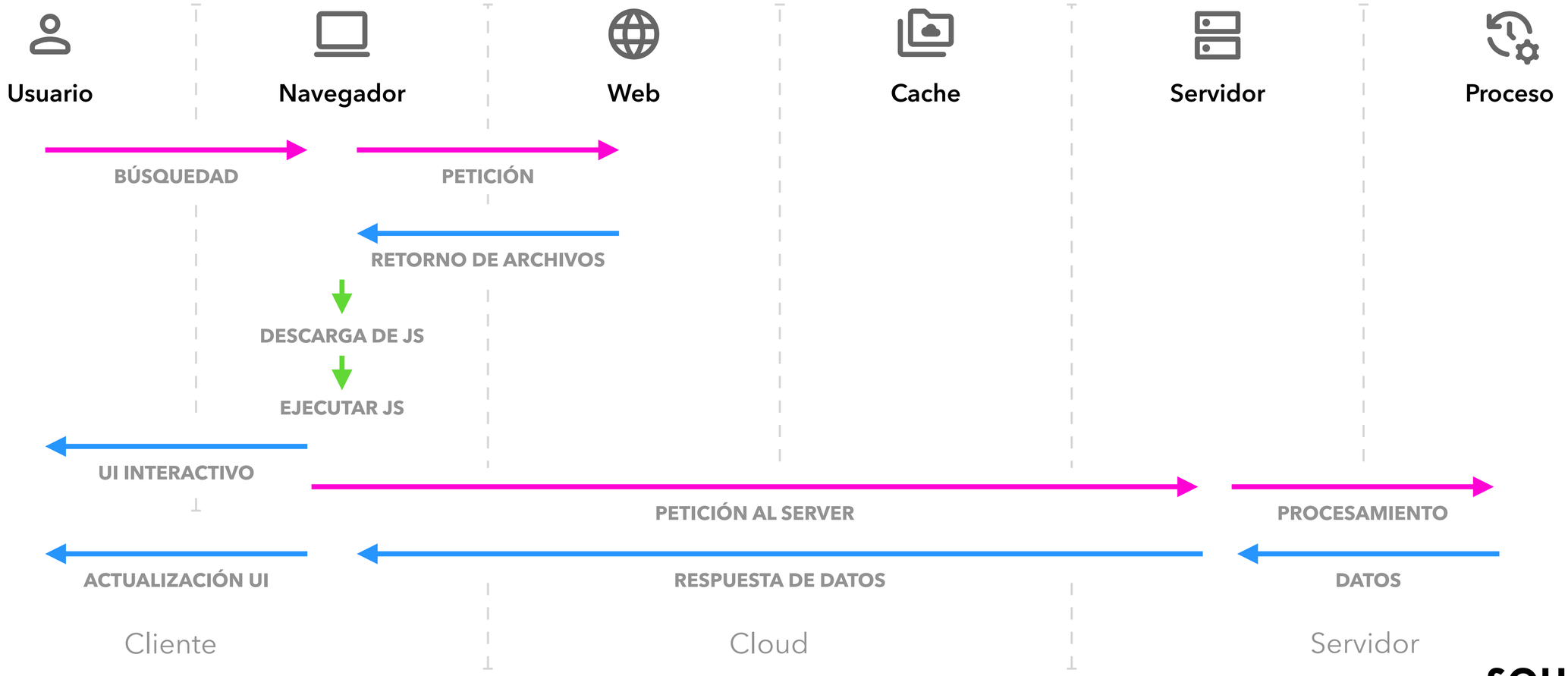
Rendering



Conceptos básicos

Rendering

Client Side Rendering - CSR



SQUARE 1 ACADEMY

Rendering

Client Side Rendering - CSR

Ventajas

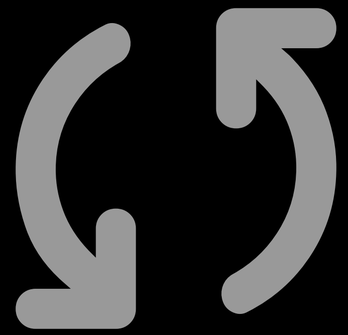
- Alto rendimiento inicial. *
- Interactividad.
- Facilidad de desarrollo.

Desventajas

- SEO.
- Tiempos de carga.

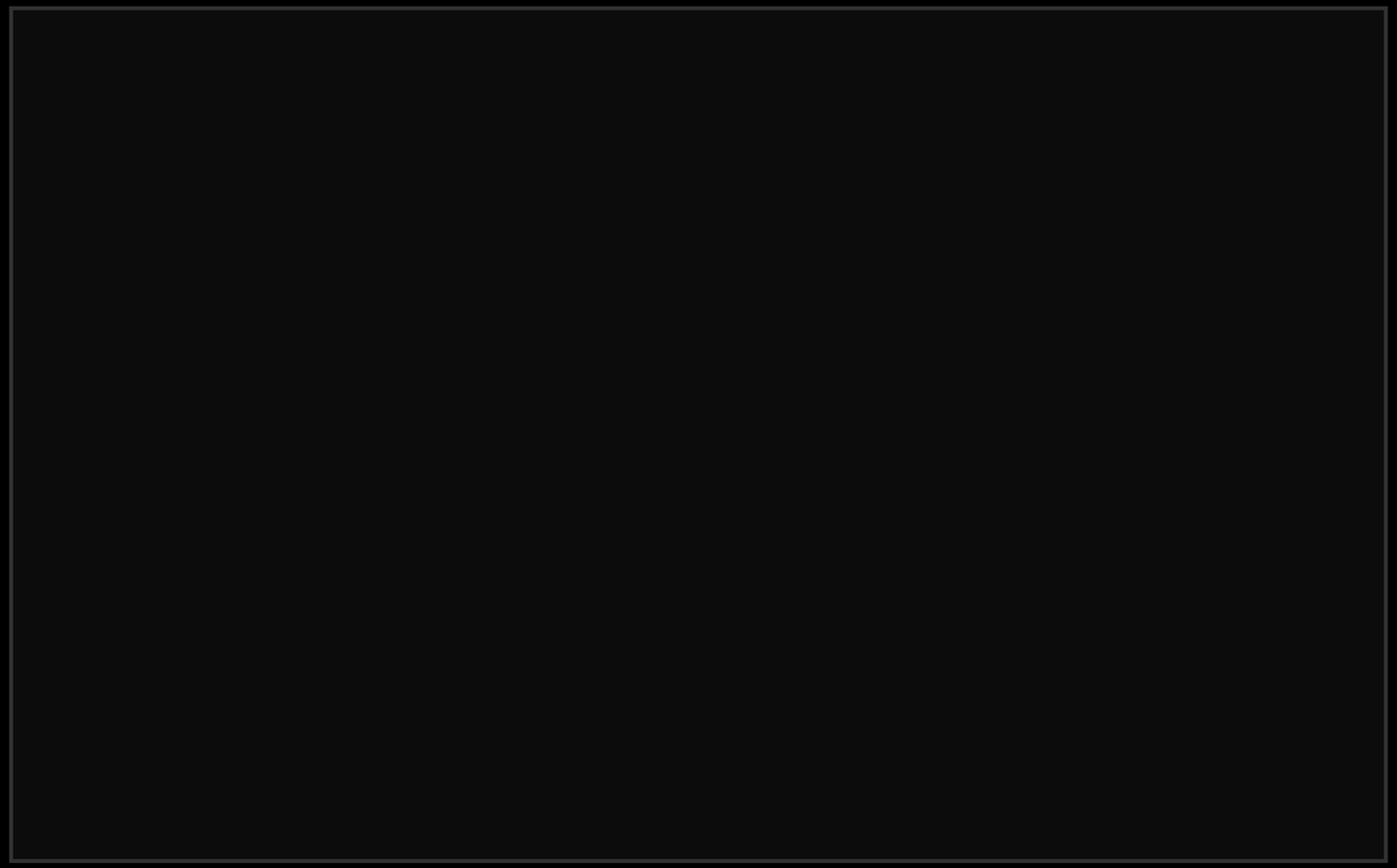
* Rendimiento inicial montando el DOM, sin la carga del Js.

Rendering



Conceptos básicos

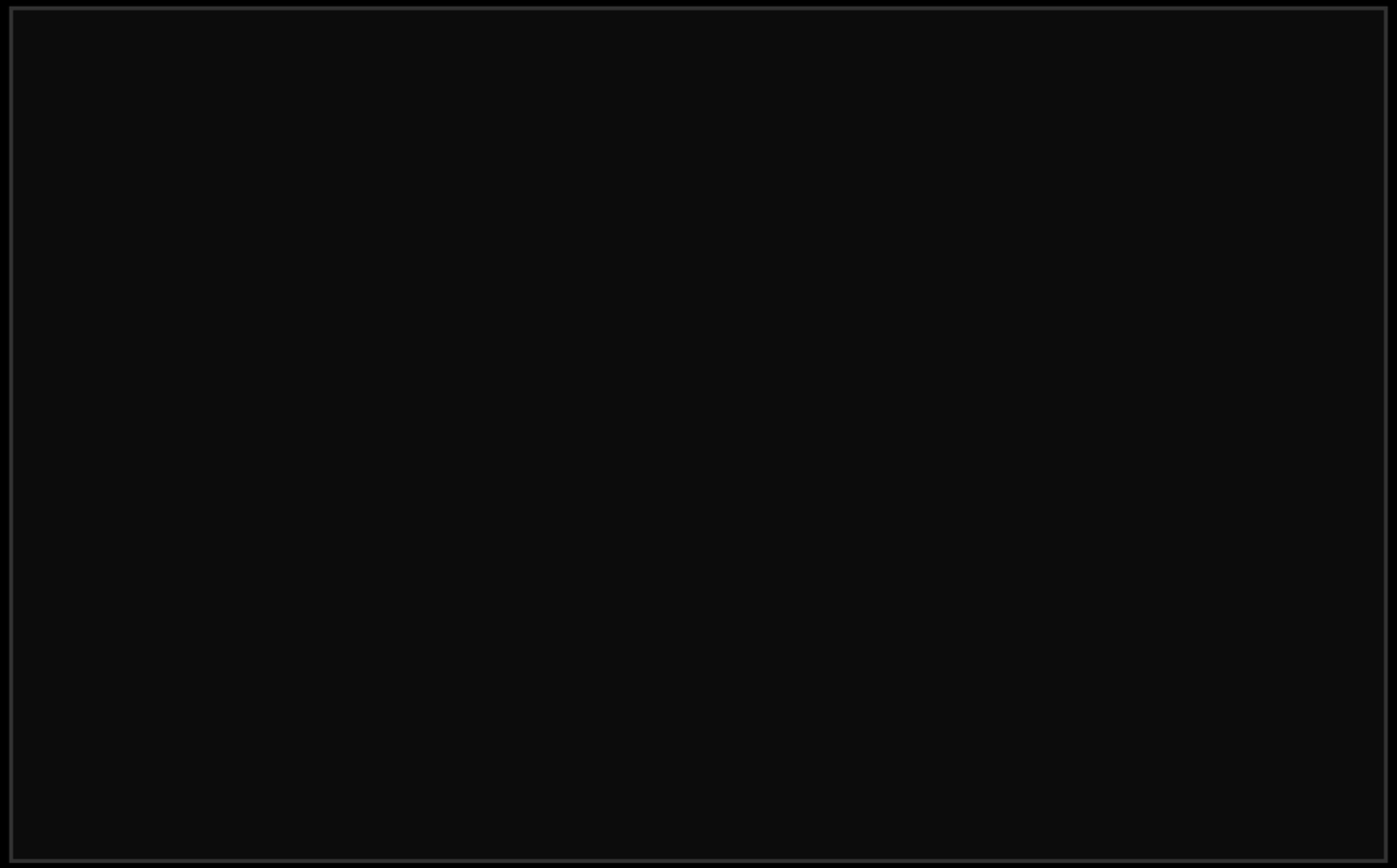
Rendering



SQUARE 1
ACADEMY

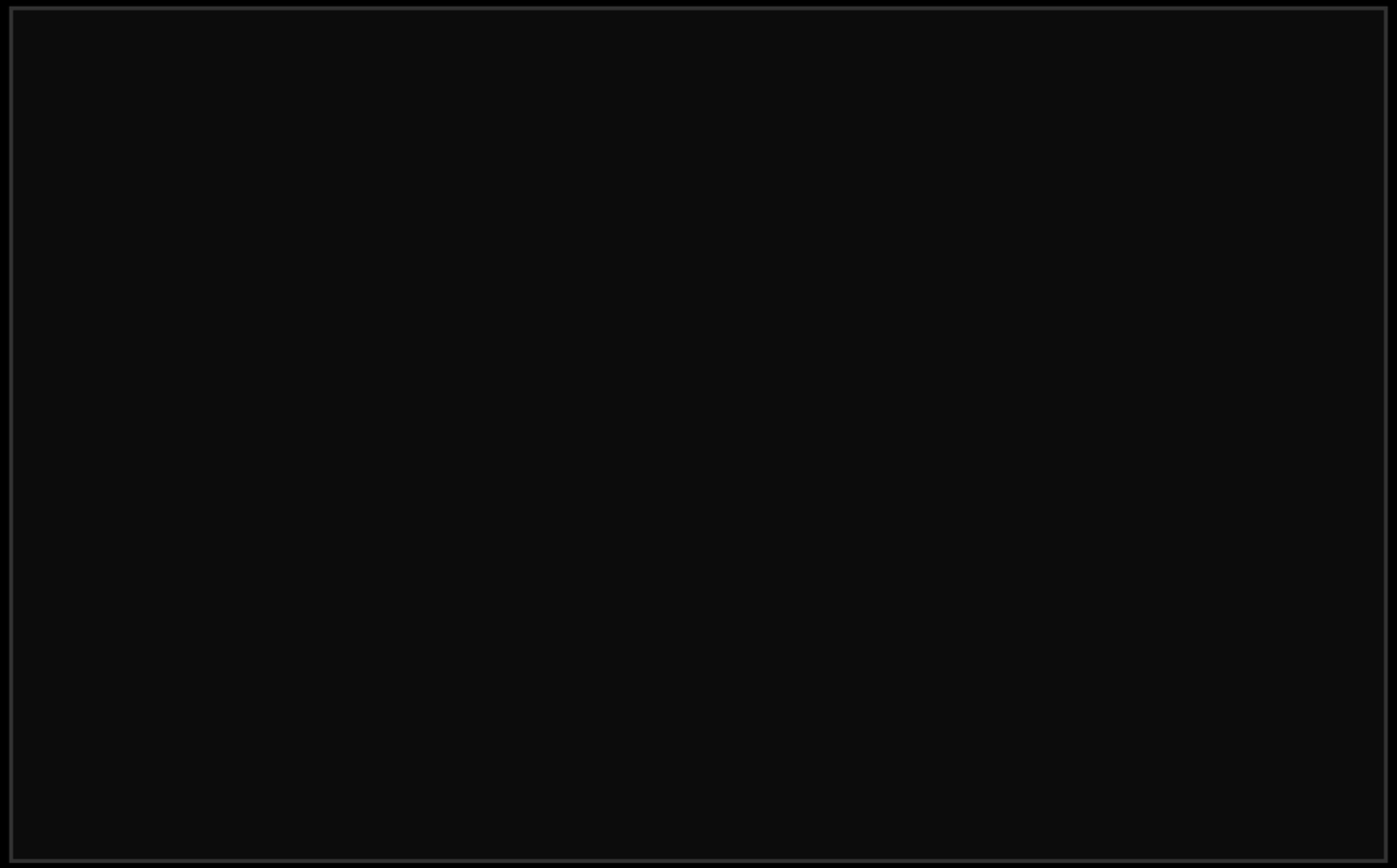
Conceptos básicos

Rendering

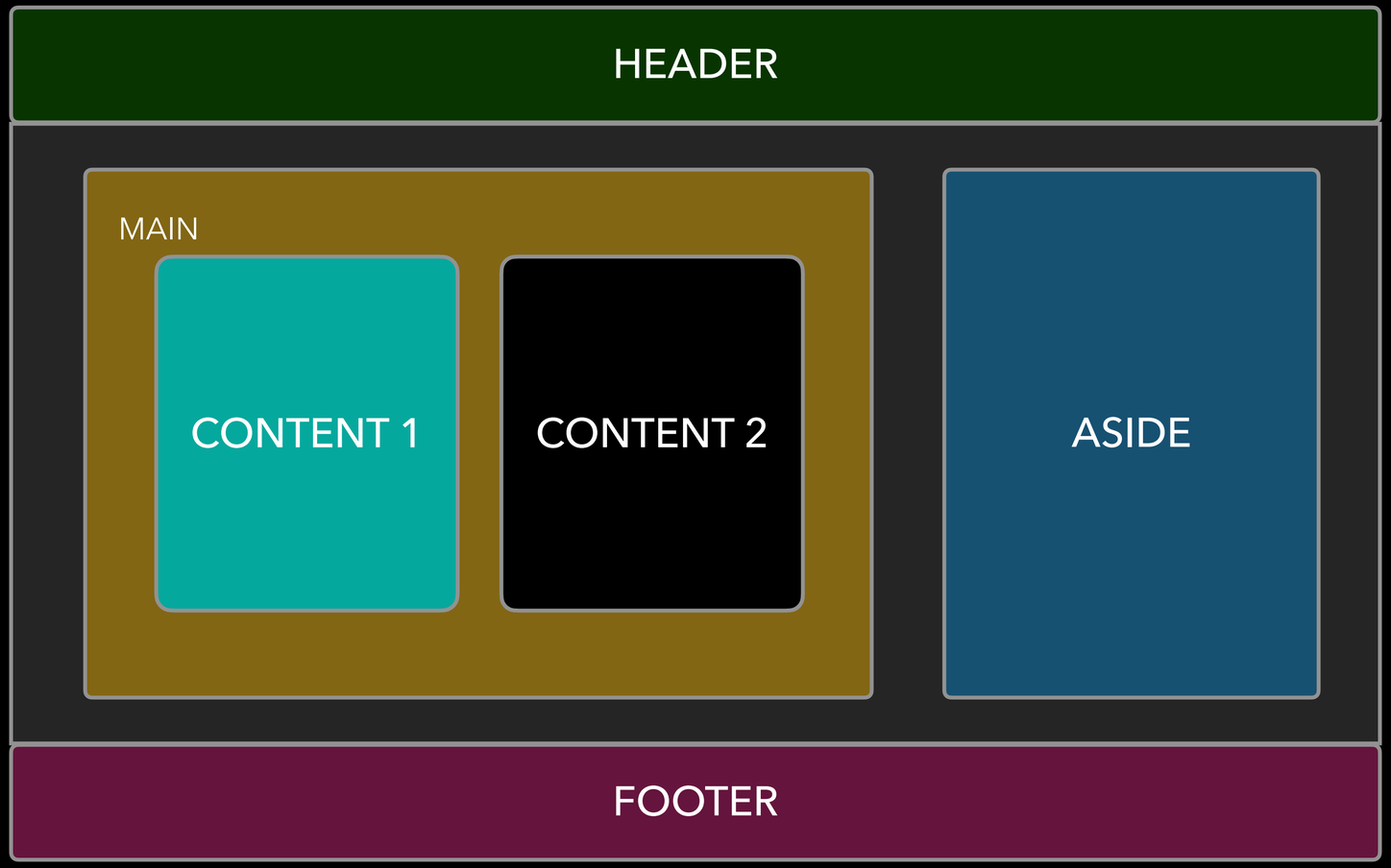


Conceptos básicos

Rendering



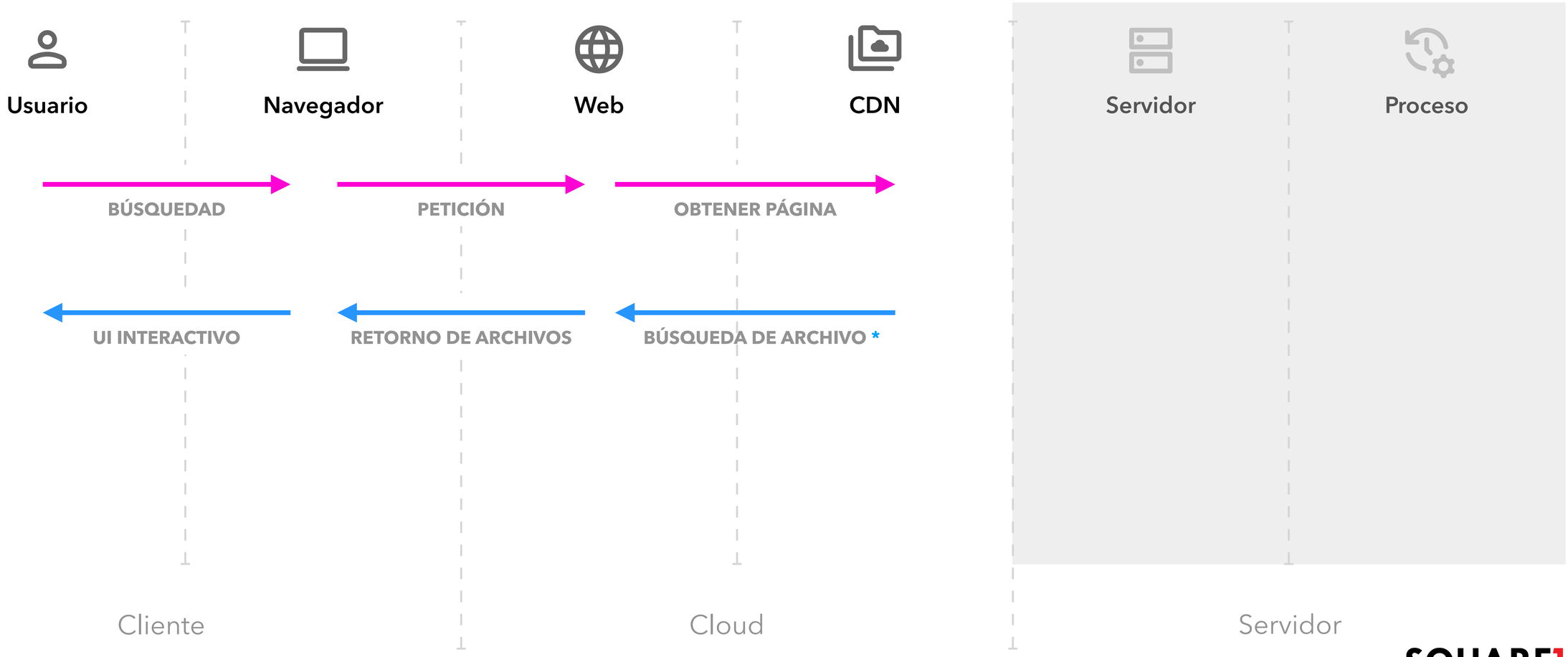
Rendering



Conceptos básicos

Rendering

Static Site Generation - SSG



* N archivos almacenados y retornados

SQUARE 1 ACADEMY

Rendering

Static Site Generation - SSG

Ventajas

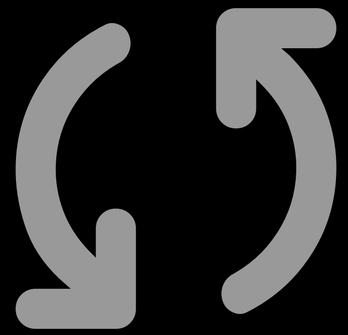
- Performance.
- Seguridad.
- Implementación del CDN.
- SEO.

Desventajas

- Interactividad.
- Contenido dinámico.
- Features avanzados.

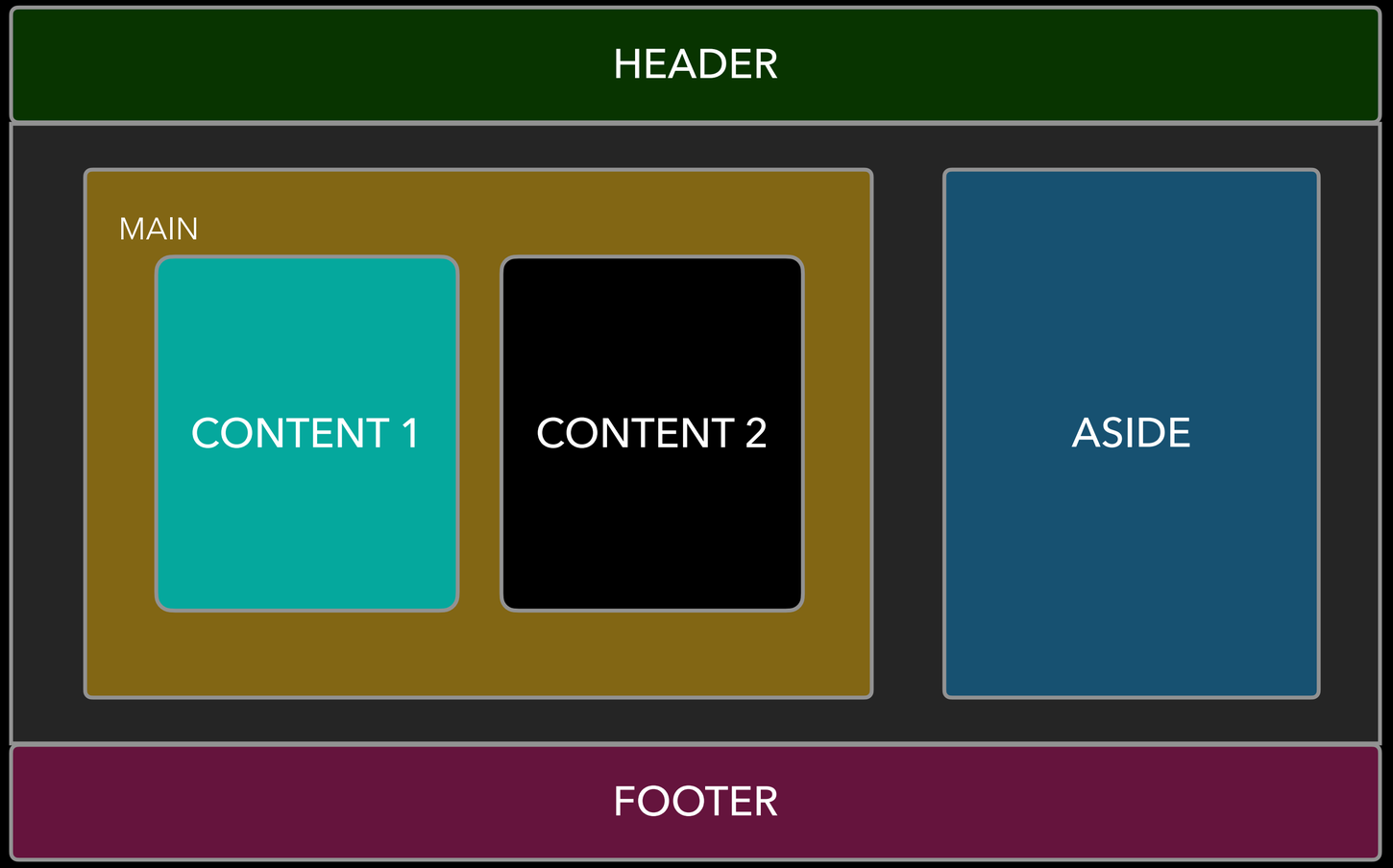
CDN: Content Delivery Network

Rendering



Conceptos básicos

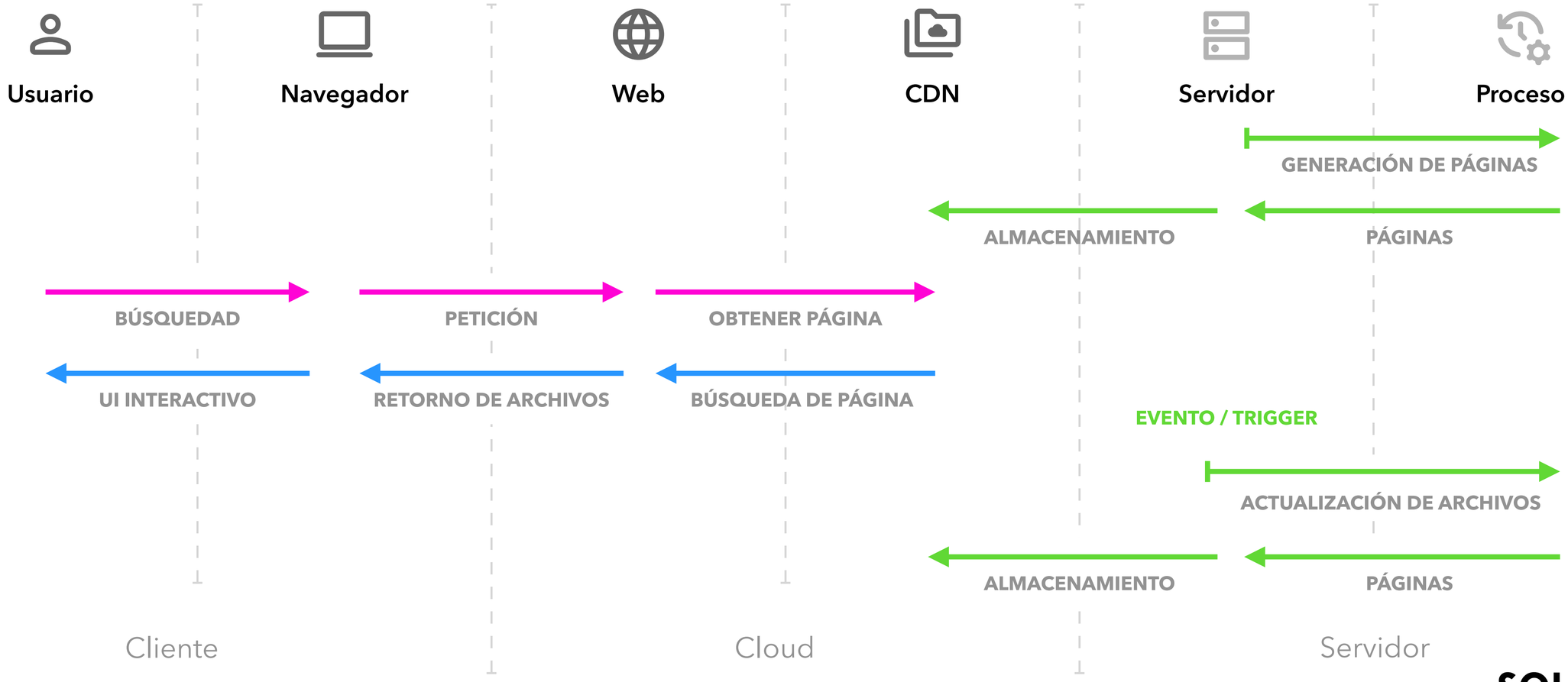
Rendering



Conceptos básicos

Rendering

Incremental Side Regeneration - ISR



SQUARE 1 ACADEMY

Rendering

Incremental Side Regeneration - ISR

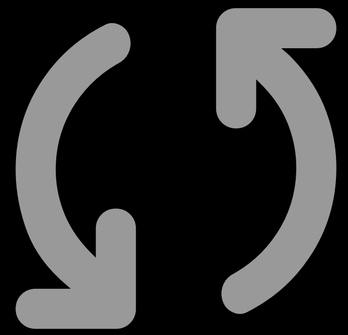
Ventajas

- Performance.
- Eficiencia de recursos.
- Contenido dinámico.

Desventajas

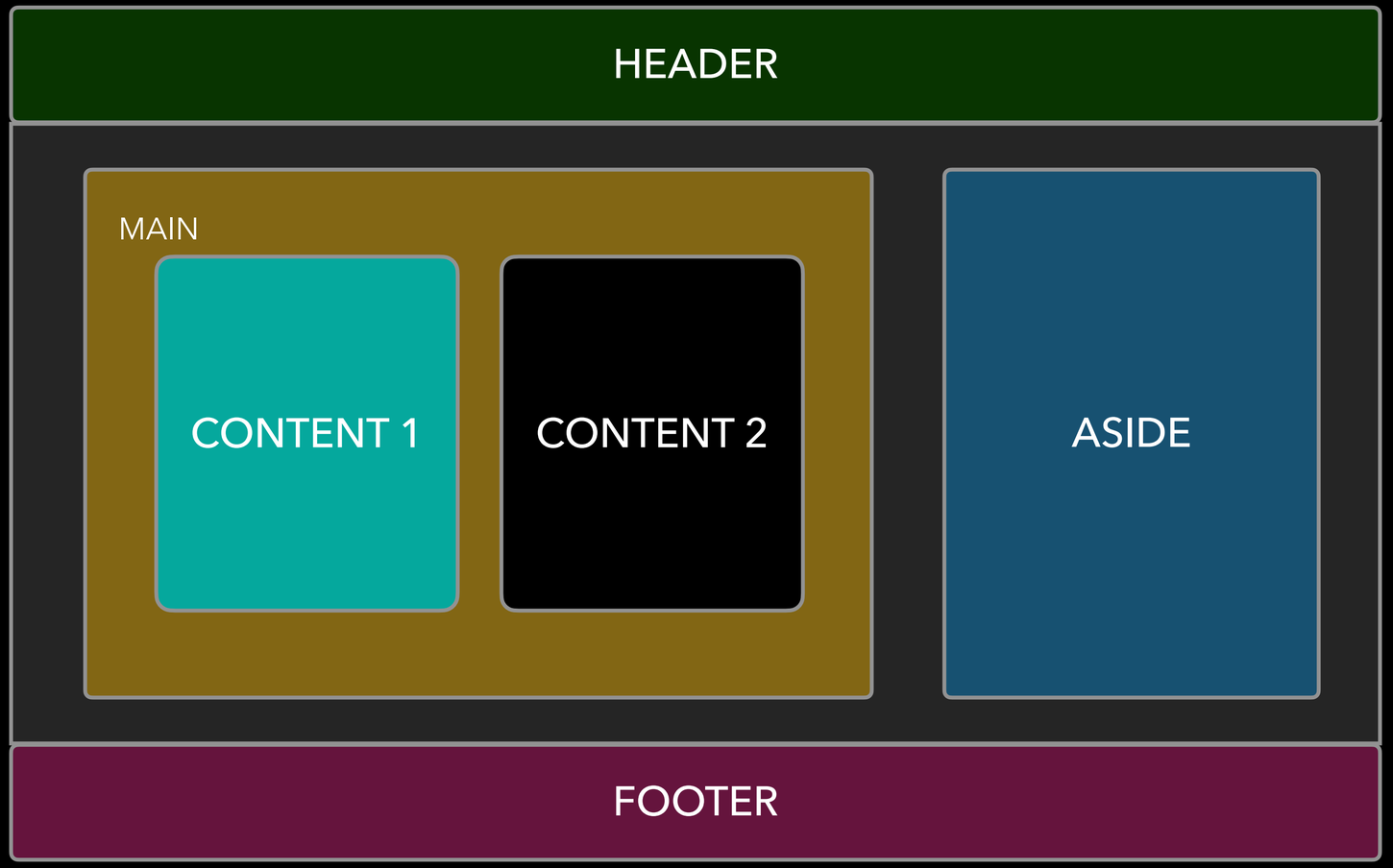
- Implementación.
- Coherencia.
- Sobrecargas en el servidor.

Rendering



Conceptos básicos

Rendering



Conceptos básicos

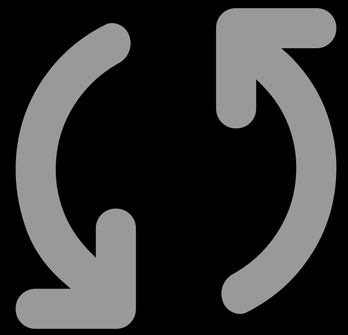
Rendering

Evento

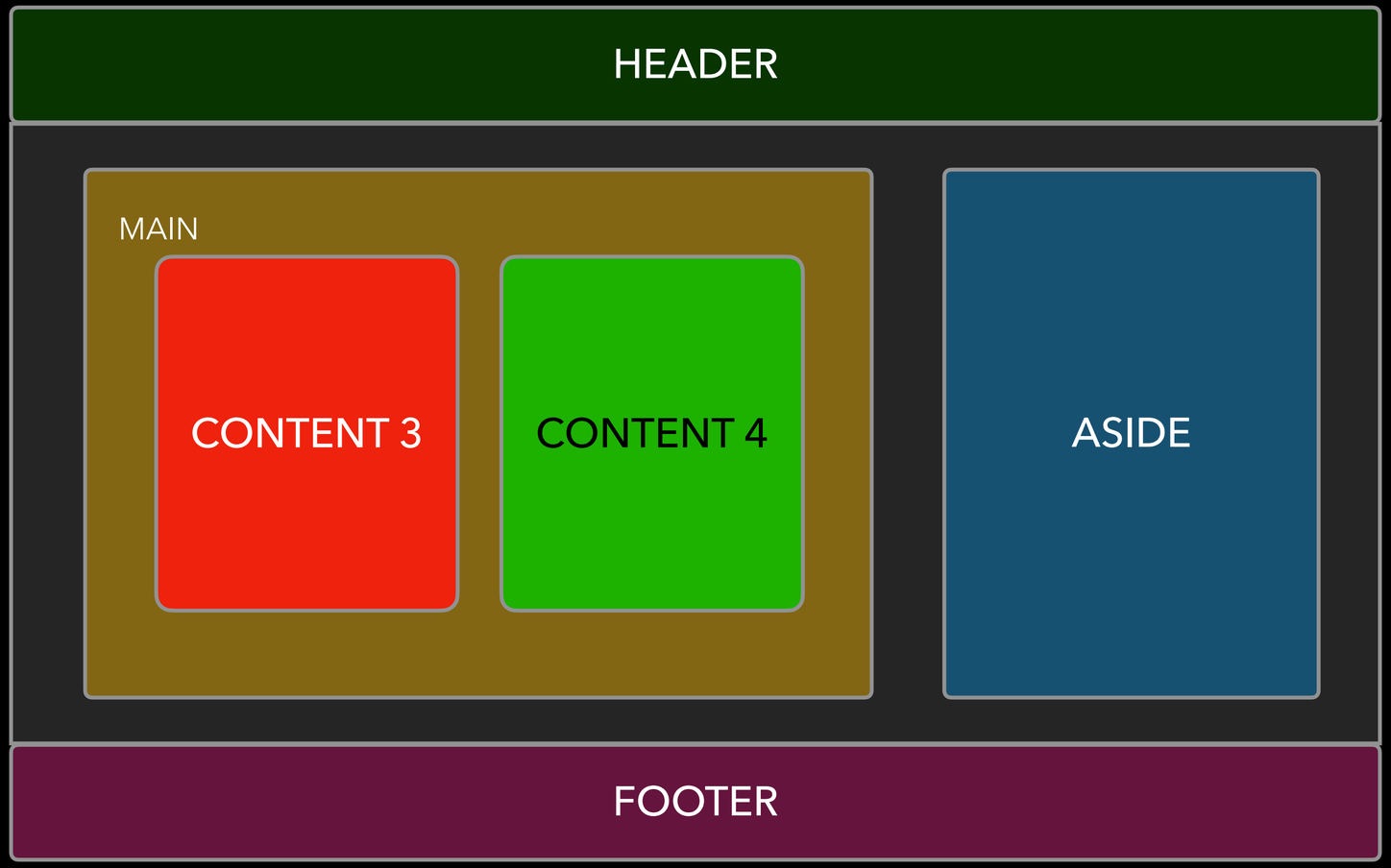
REQUEST

SQUARE1 ACADEMY

Rendering



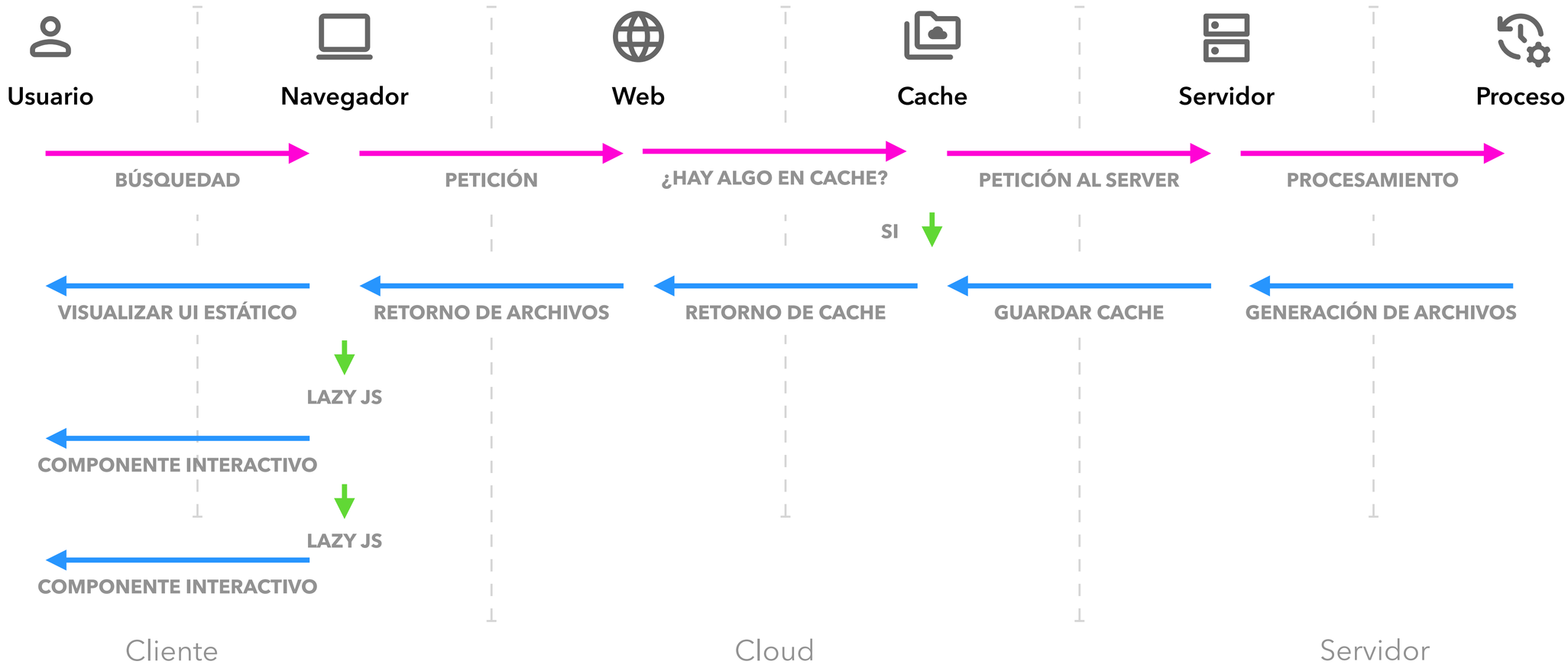
Rendering



Conceptos básicos

Rendering

Deferred Site Rendering - DSR



SQUARE 1 ACADEMY

Rendering

Deferred Site Rendering - DSR

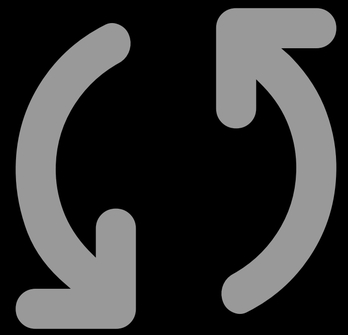
Ventajas

- Performance.
- Eficiencia de recursos.
- Optimización de recursos.

Desventajas

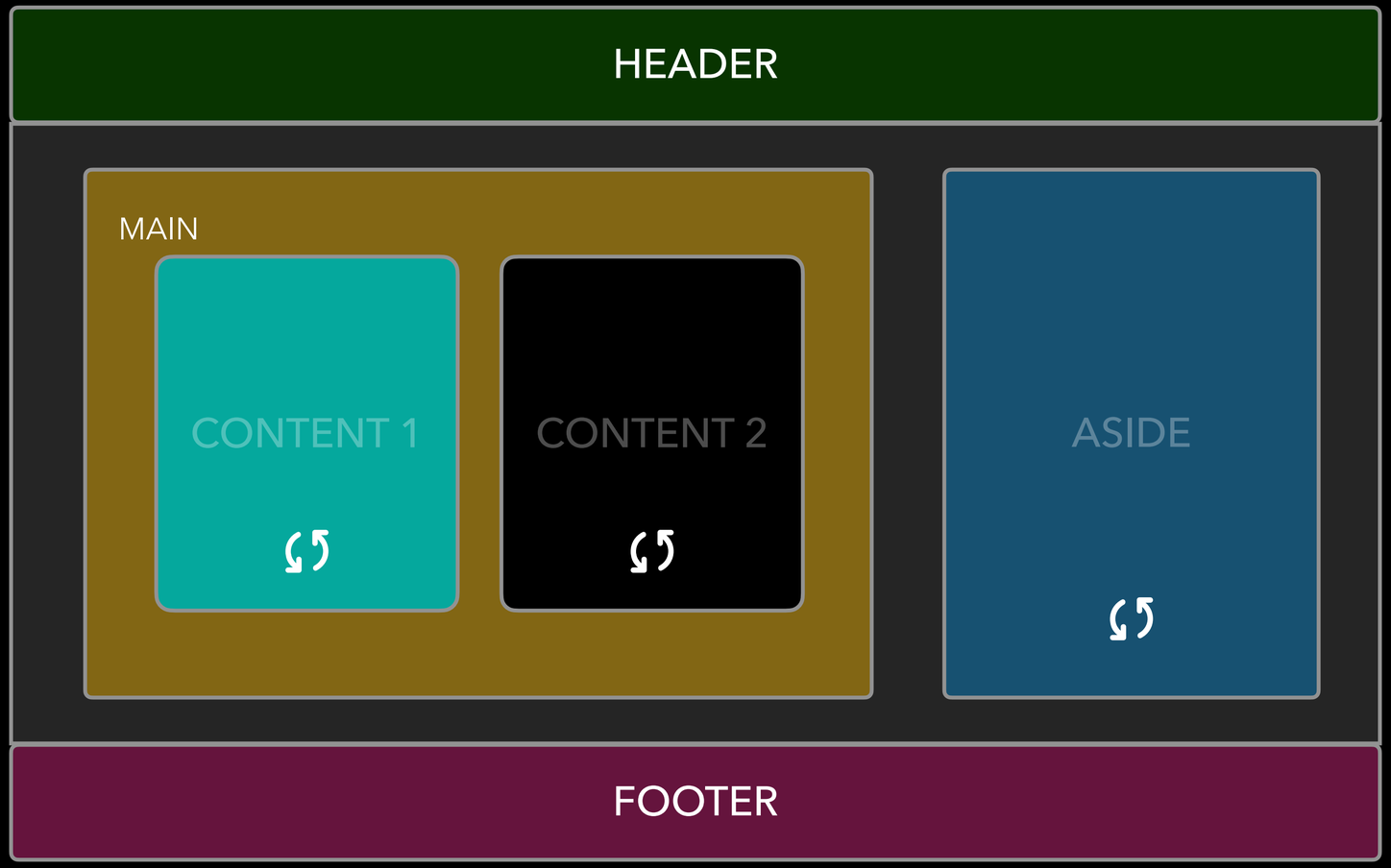
- Implementación.
- Carga.
- Accesibilidad.

Rendering



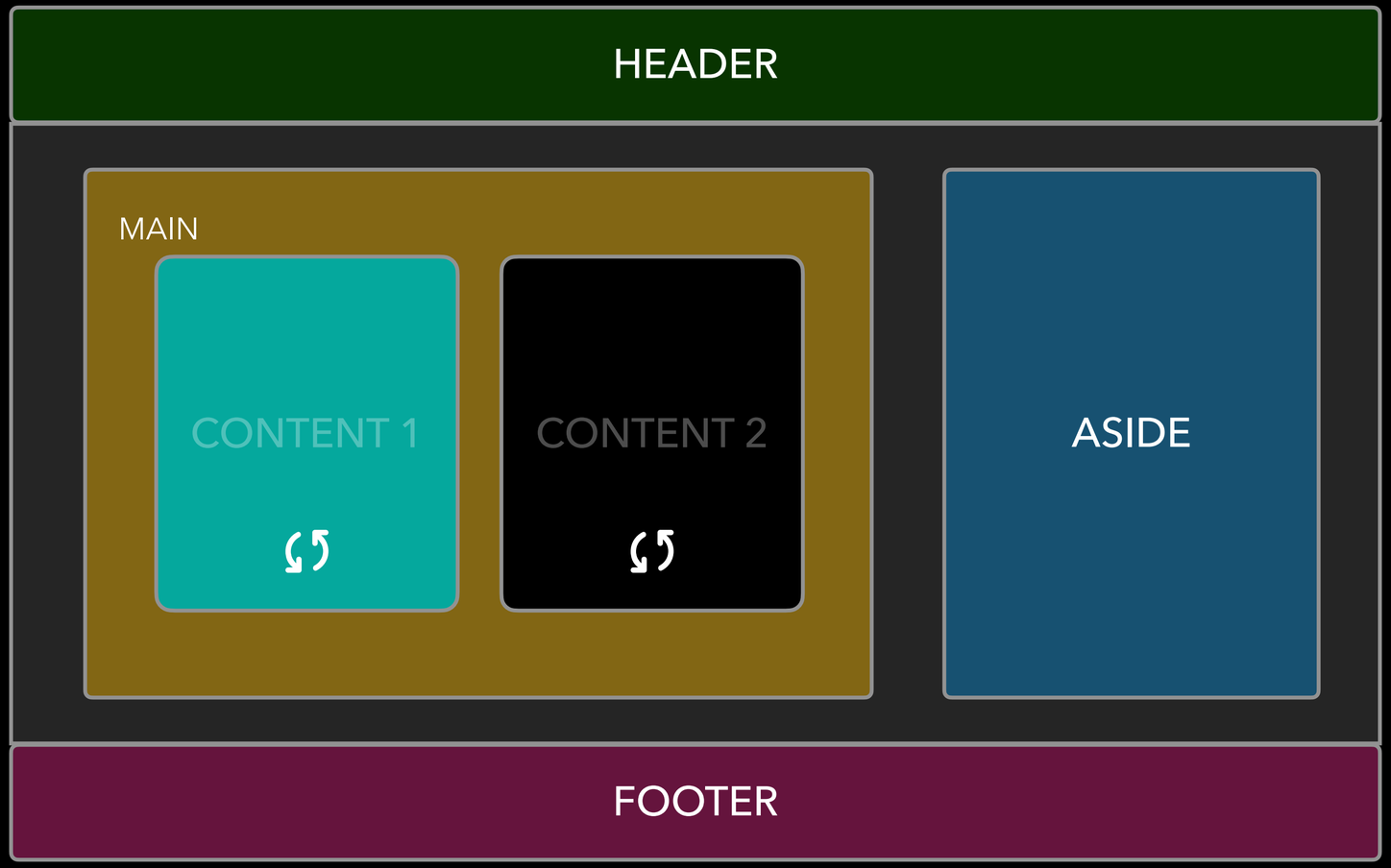
Conceptos básicos

Rendering



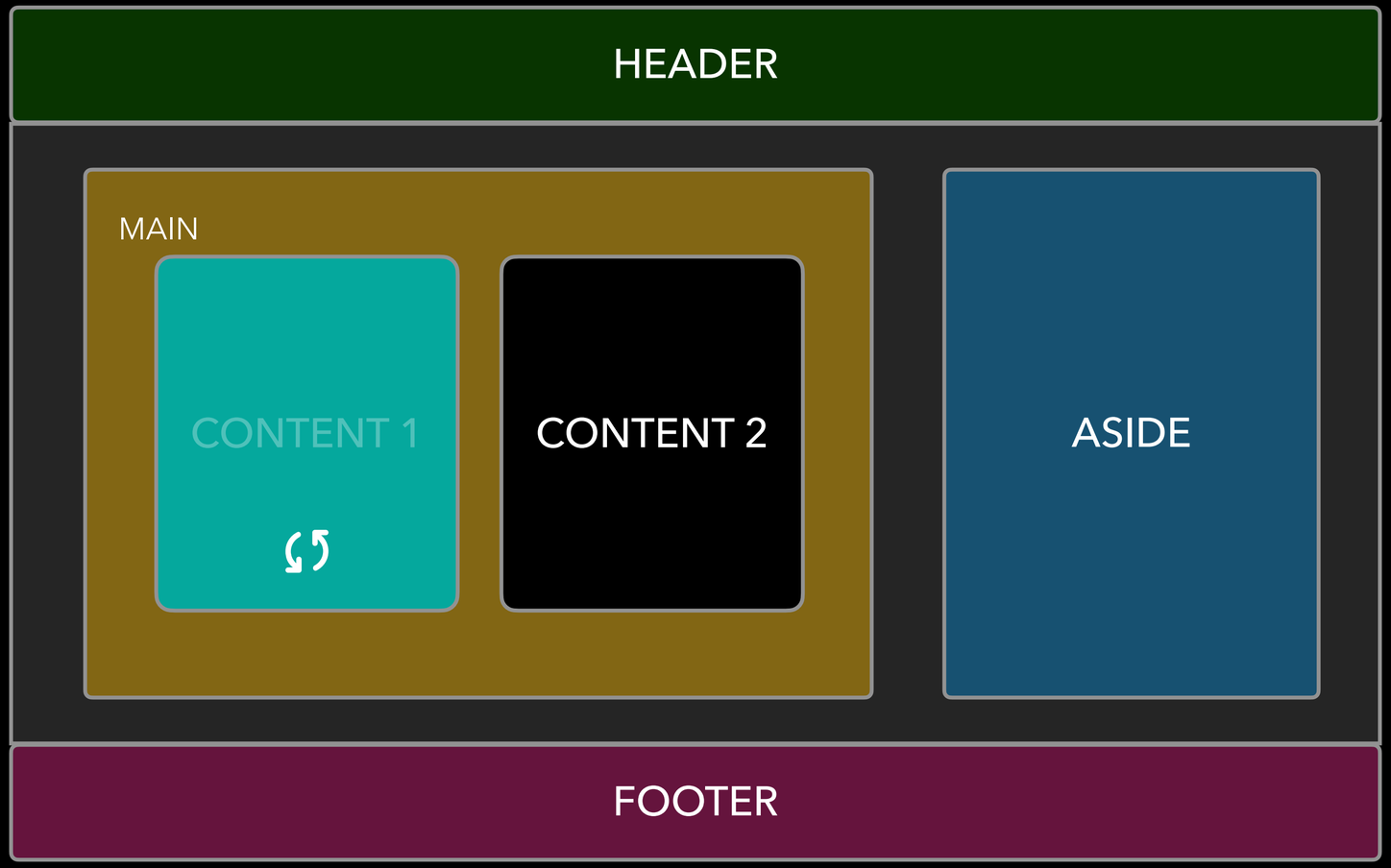
Conceptos básicos

Rendering



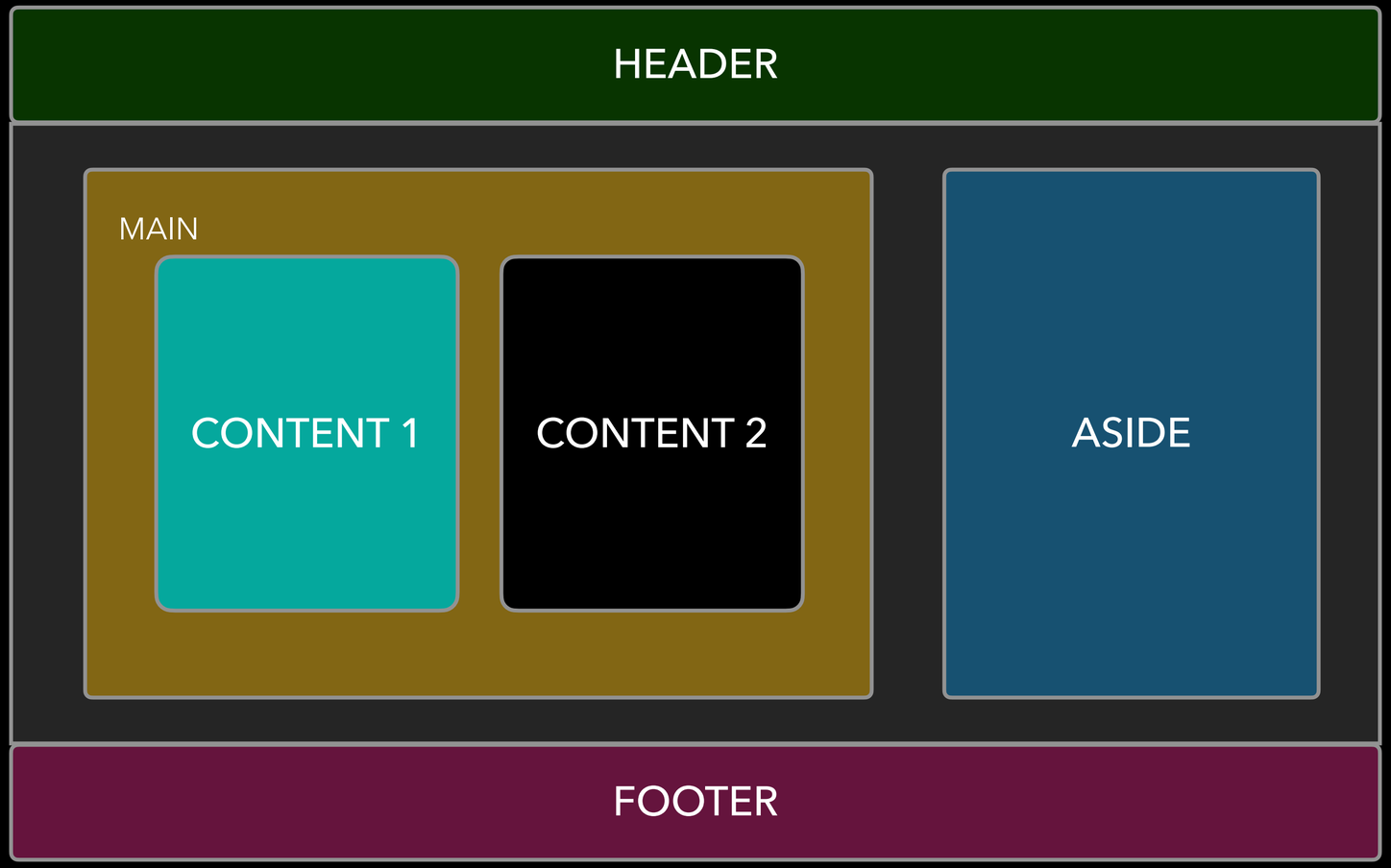
Conceptos básicos

Rendering



Conceptos básicos

Rendering



Frameworks y librerías más comunes

Frameworks y librerías más comunes

- React Js / Next Js
- Vue Js / Nuxt Js
- Astro
- Angular
- Preact

SQUARE 1
ACADEMY

Conceptos básicos

React Js - Next Js

Es una librería para la construcción de **interfaces de usuario interactivas** para web y móvil basada en CSR, creado por Facebook.

Conceptos básicos

React Js - Next Js

features

- **Componentes**
- Virtual DOM
- JSX
- Unidireccionalidad de datos

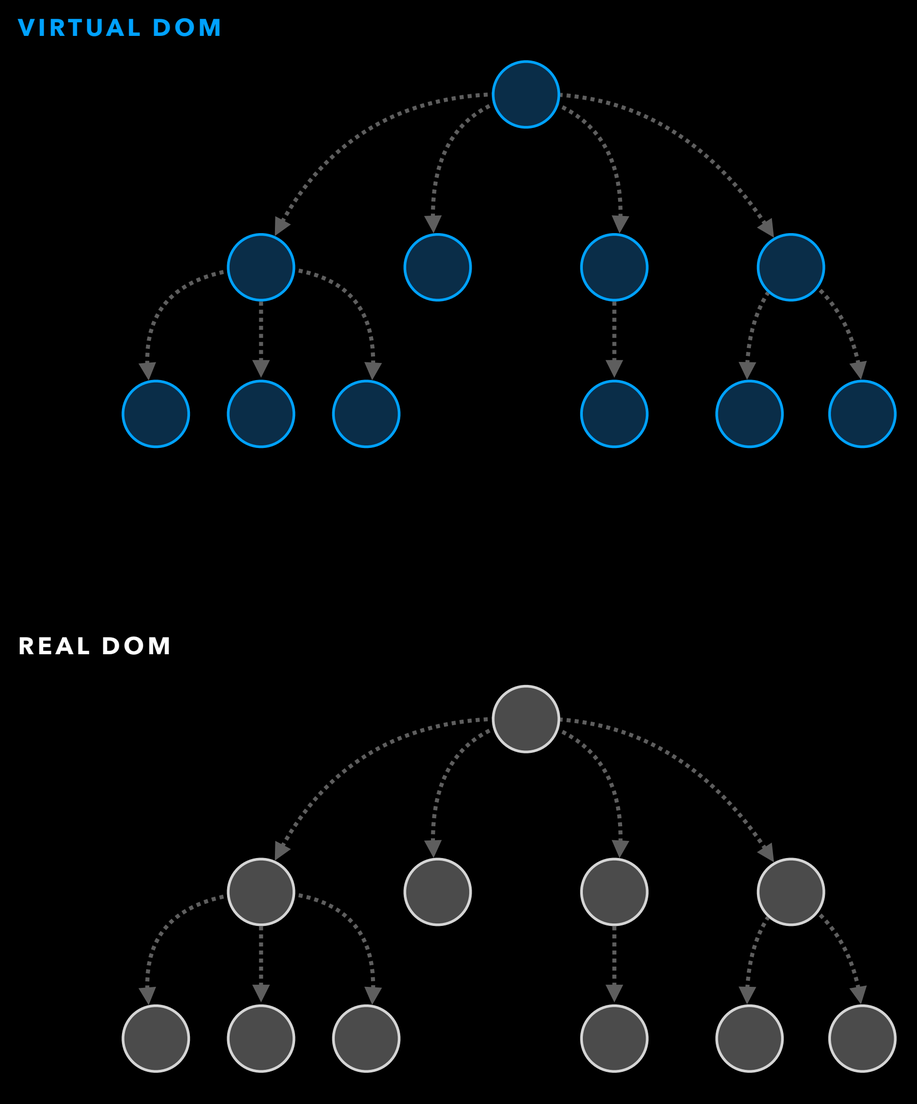
SQUARE 1
ACADEMY

Conceptos básicos

React Js - Next Js

features

- Componentes
- **Virtual DOM**
- JSX
- Unidireccionalidad de datos



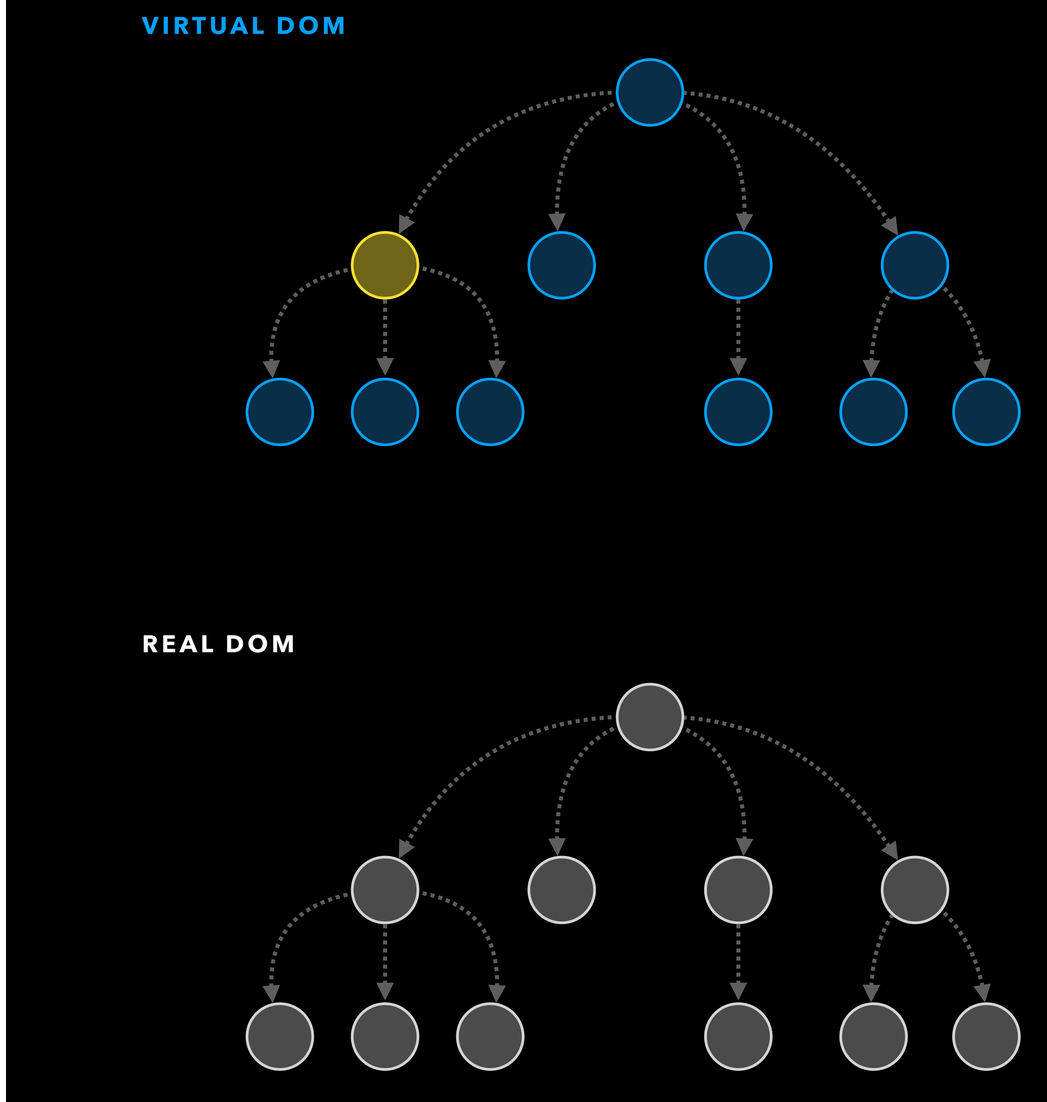
SQUARE1 ACADEMY

Conceptos básicos

React Js - Next Js

features

- Componentes
- **Virtual DOM**
- JSX
- Unidireccionalidad de datos

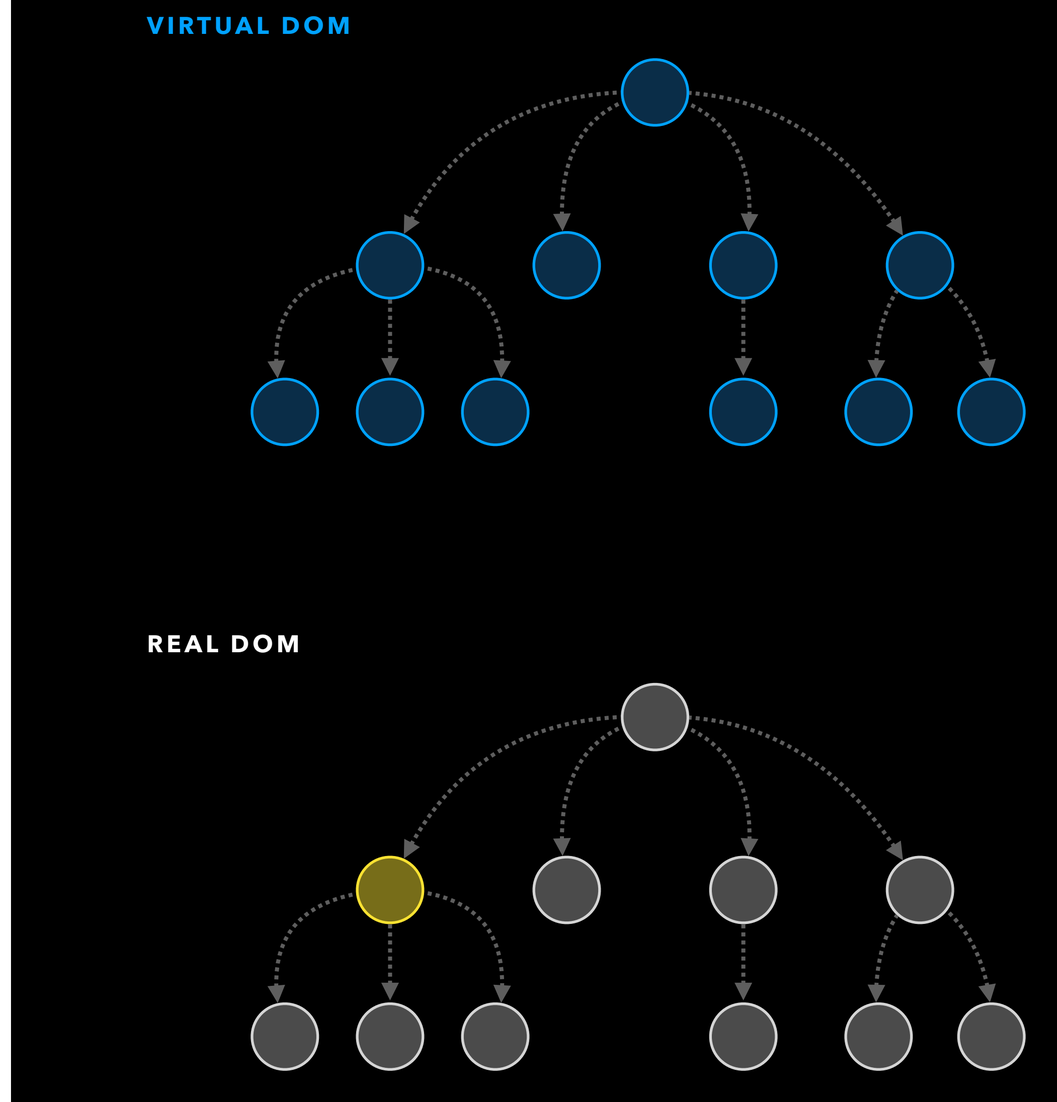


Conceptos básicos

React Js - Next Js

features

- Componentes
- **Virtual DOM**
- JSX
- Unidireccionalidad de datos

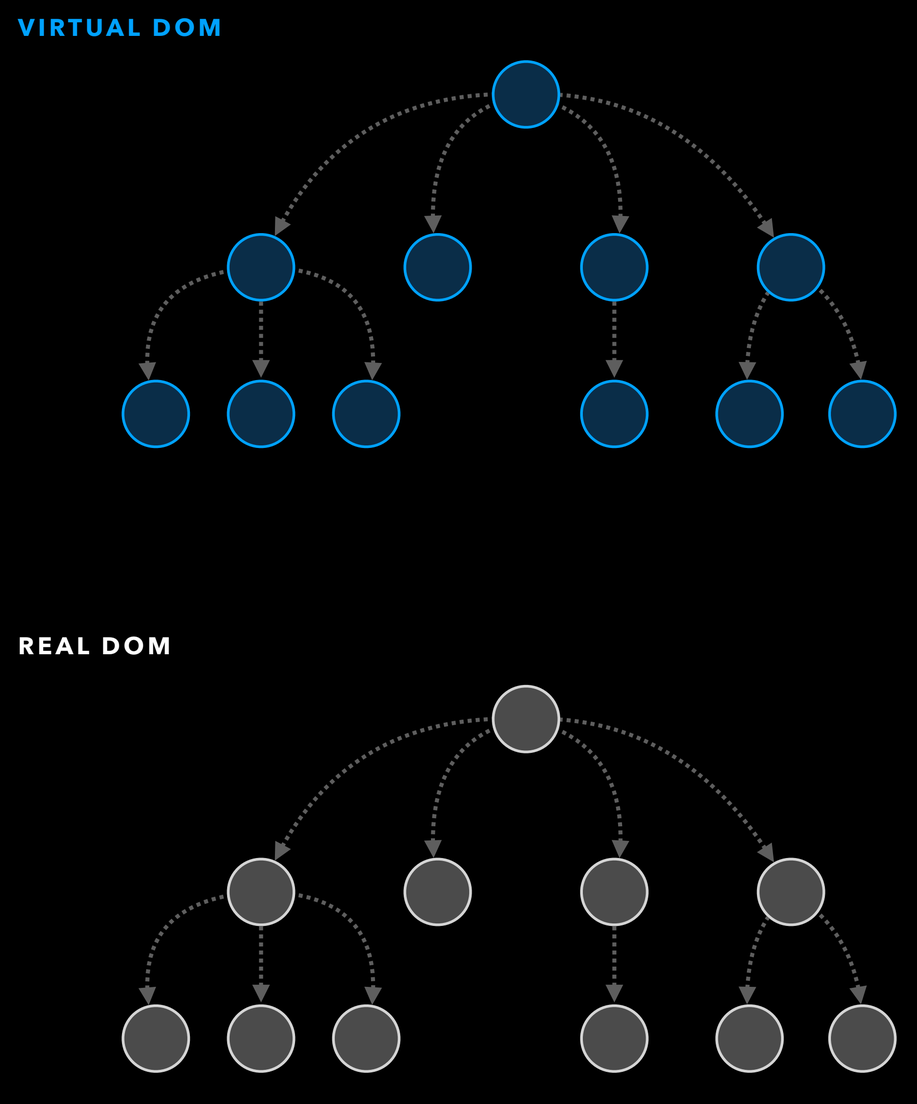


Conceptos básicos

React Js - Next Js

features

- Componentes
- **Virtual DOM**
- JSX
- Unidireccionalidad de datos



SQUARE1 ACADEMY

Conceptos básicos

React Js - Next Js

features

- Componentes
- Virtual DOM
- **JSX**
- Unidireccionalidad de datos

HTML y Javascript

```html
<div id="componente"></div>

<script>
  document.addEvenetListener("DOMContentLoaded", function(){
    const parent = document.getElementById('componente');

    const child = document.createElement("p");
    child.style.color = "white";

    const newText = document.createTextNode("Hola, soy el hijo");

    child.apendChild(newText);
    parent.apendChild(child);
  })
</script>
```

JSX

```jsx
const Componente = () => {
    return (
      <div>
        <p style={{ color: 'white' }}>Hola, soy el hijo</p>
      </div>
    )
}

export default Componente;
```

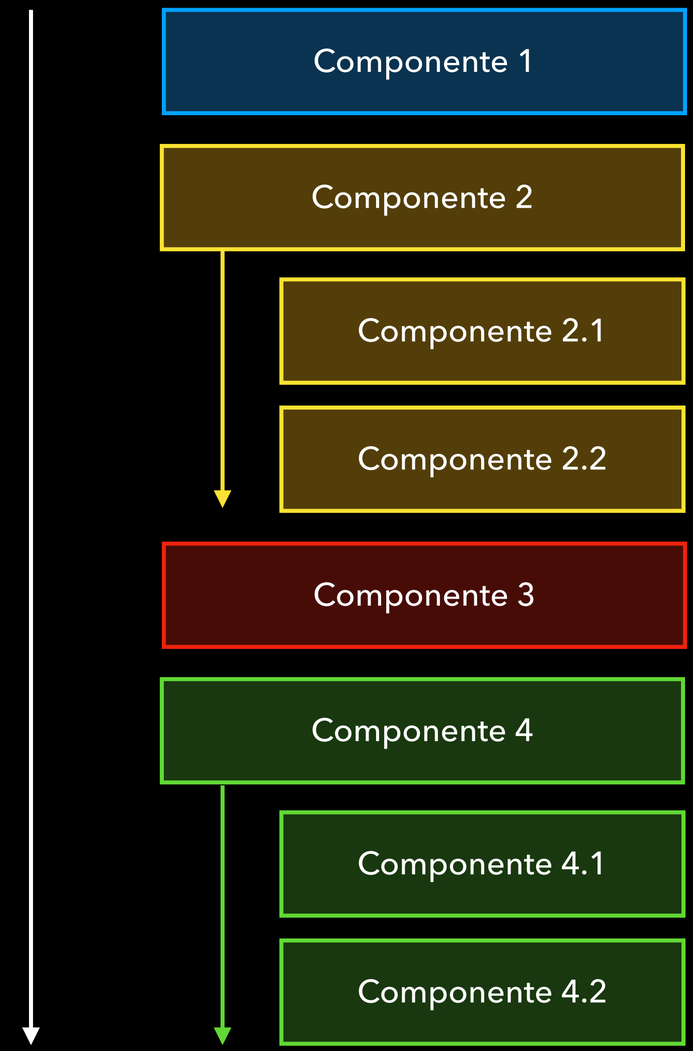
SQUARE 1 ACADEMY

Conceptos básicos

React Js - Next Js

features

- Componentes
- Virtual DOM
- JSX
- **Unidireccionalidad de datos**



SQUARE1 ACADEMY

Vue Js - Nuxt Js

Es una librería de código abierto para la construcción de **single page applications**.

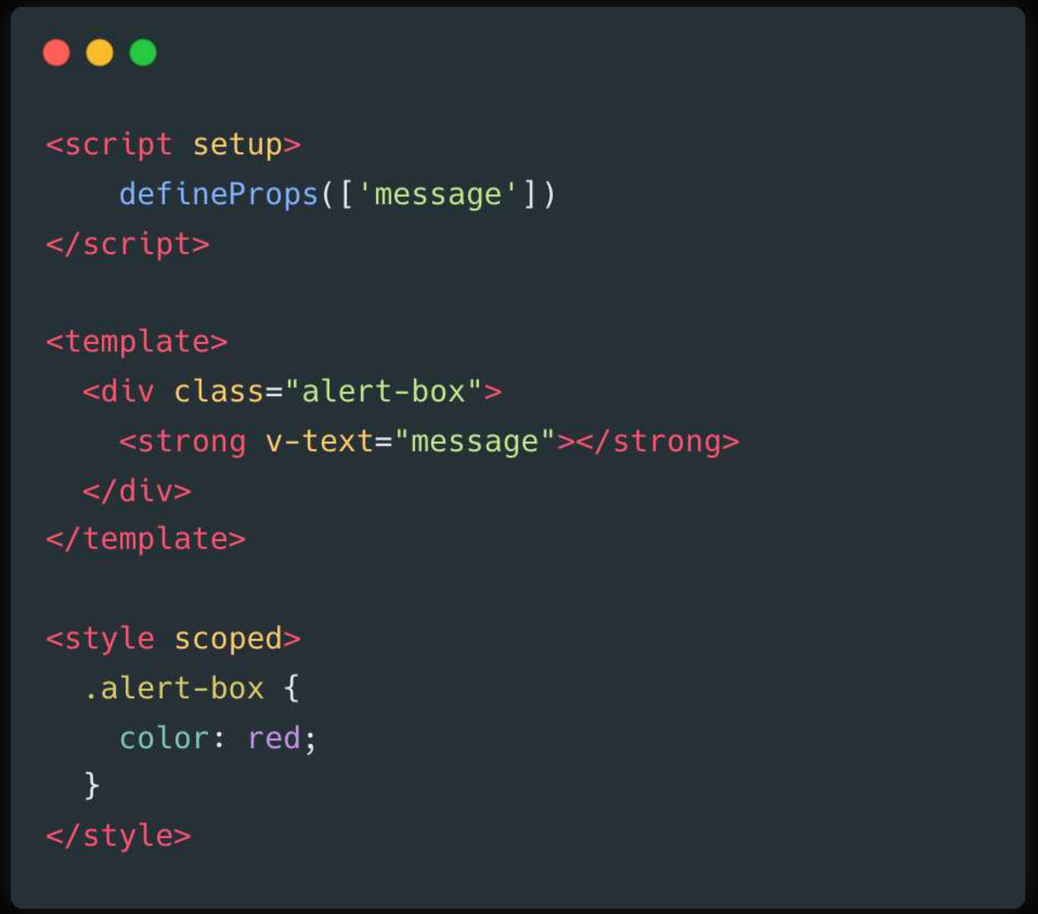
Tiene un enfoque progresivo e implementación incremental.

Conceptos básicos

Vue Js - Nuxt Js

features

- **Single File Component**
- Virtual DOM
- Directivas
- Unidireccionalidad de datos



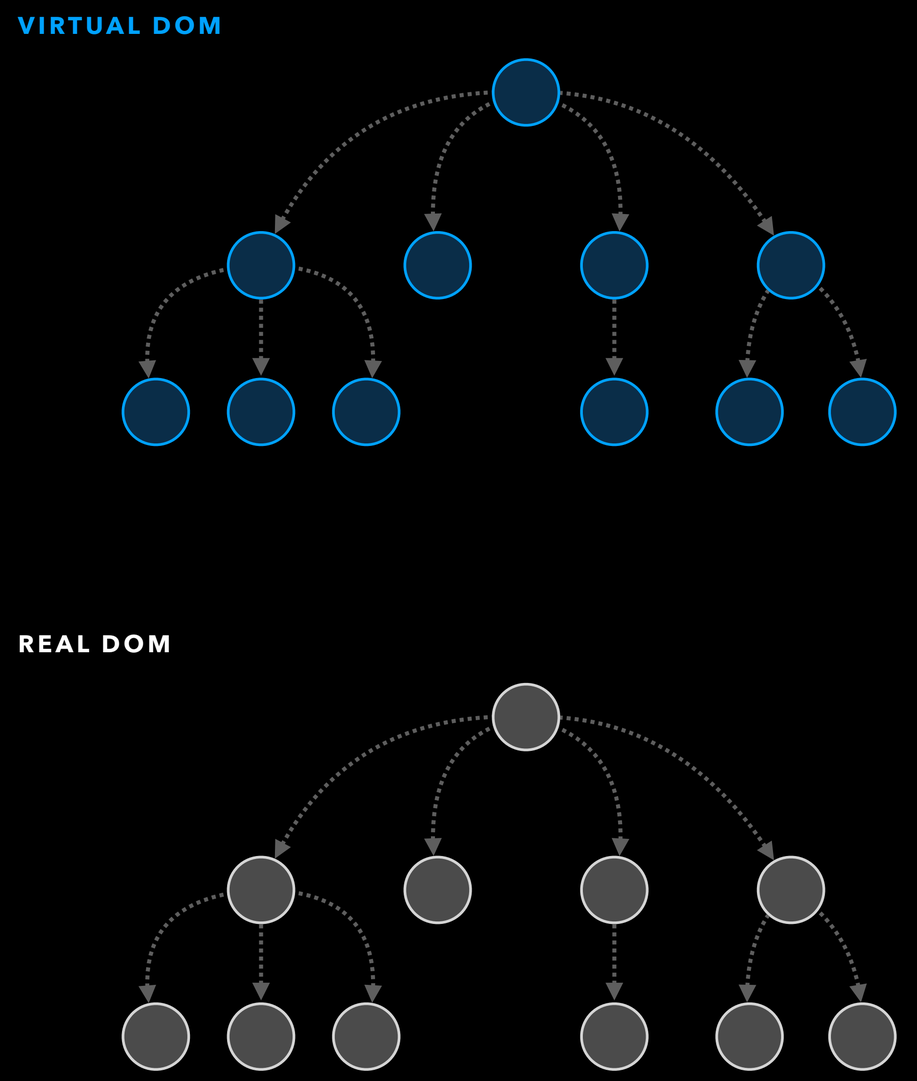
SQUARE1 ACADEMY

Conceptos básicos

Vue Js - Nuxt Js

features

- Single File Component
- **Virtual DOM**
- Directivas
- Unidireccionalidad de datos



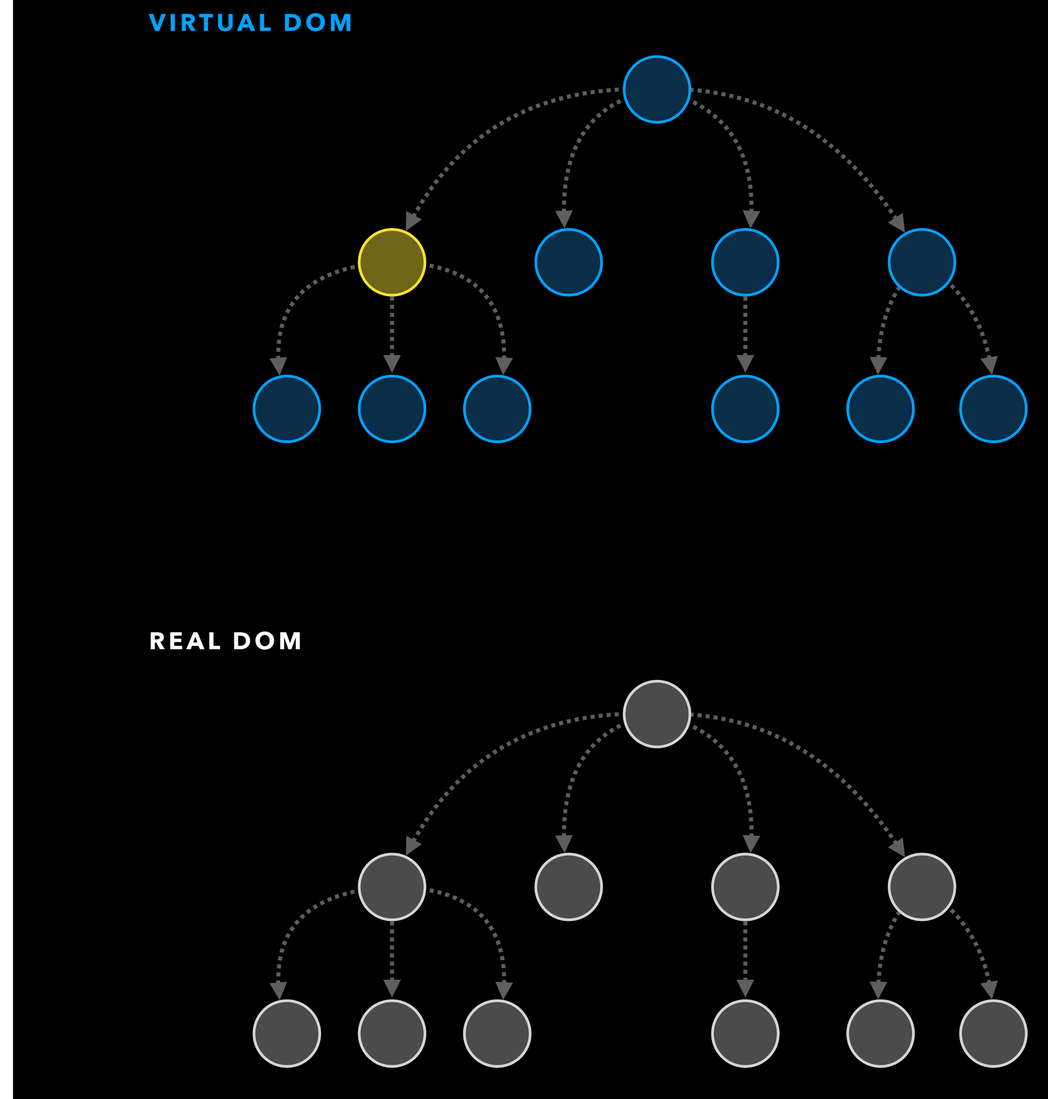
SQUARE 1 ACADEMY

Conceptos básicos

Vue Js - Nuxt Js

features

- Single File Component
- **Virtual DOM**
- Directivas
- Unidireccionalidad de datos



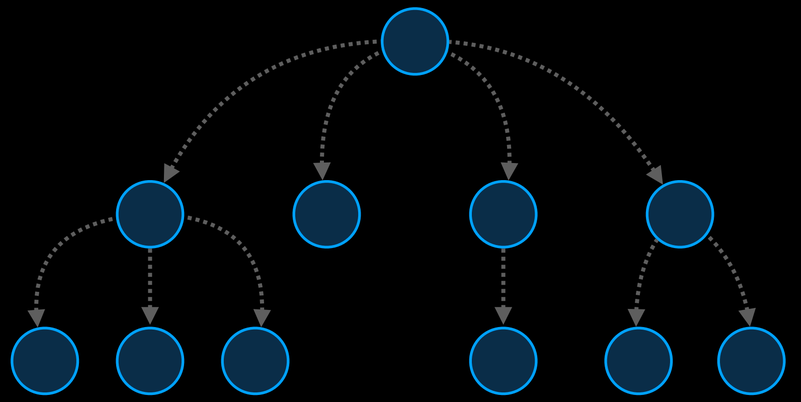
Conceptos básicos

Vue Js - Nuxt Js

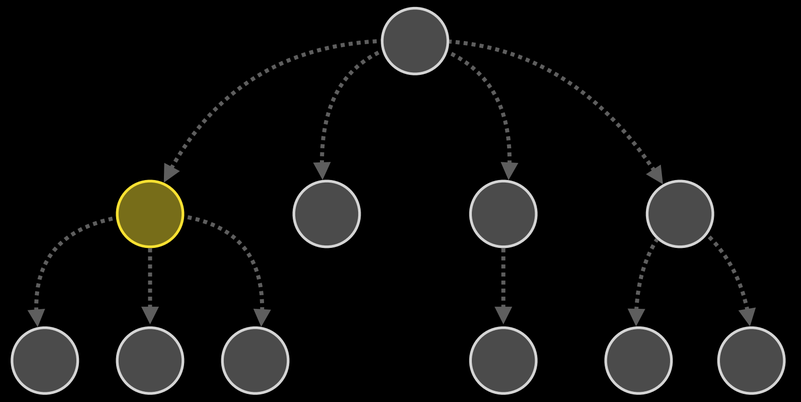
features

- Single File Component
- **Virtual DOM**
- Directivas
- Unidireccionalidad de datos

VIRTUAL DOM



REAL DOM



HIDRATACIÓN

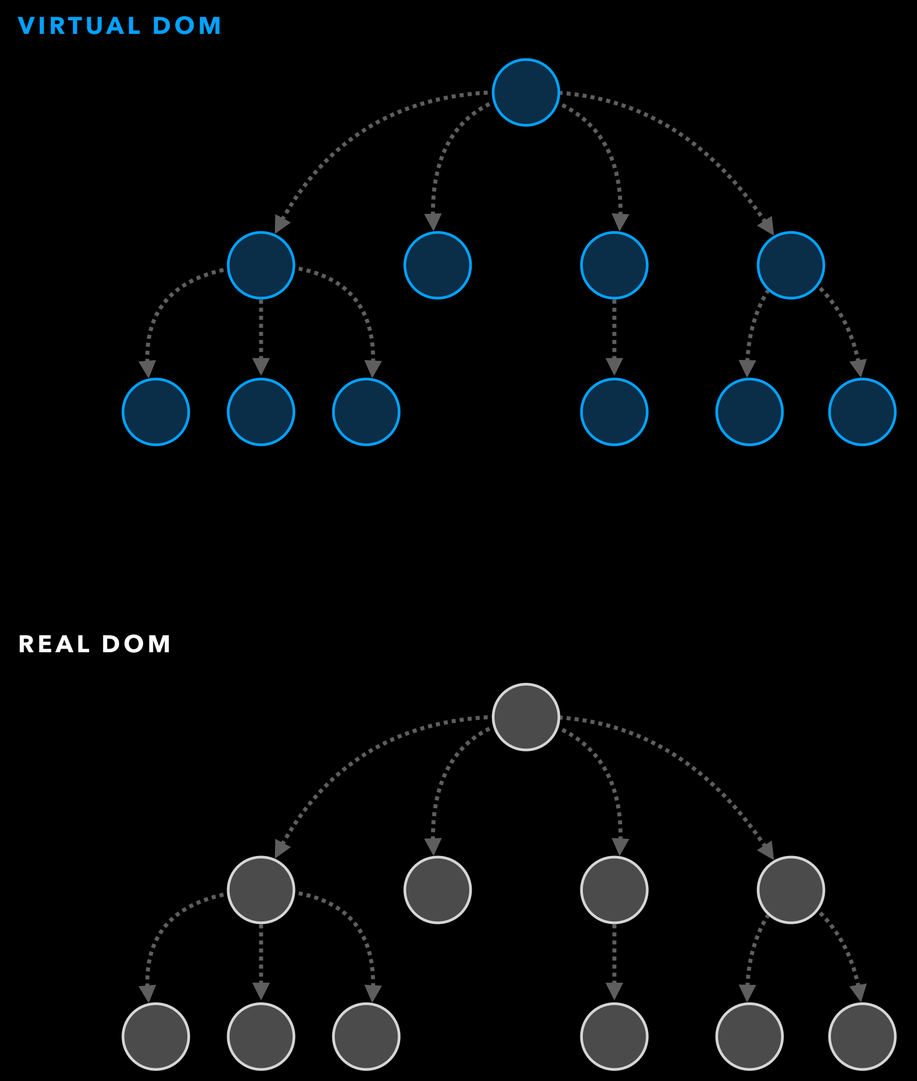
SQUARE1 ACADEMY

Conceptos básicos

Vue Js - Nuxt Js

features

- Single File Component
- **Virtual DOM**
- Directivas
- Unidireccionalidad de datos



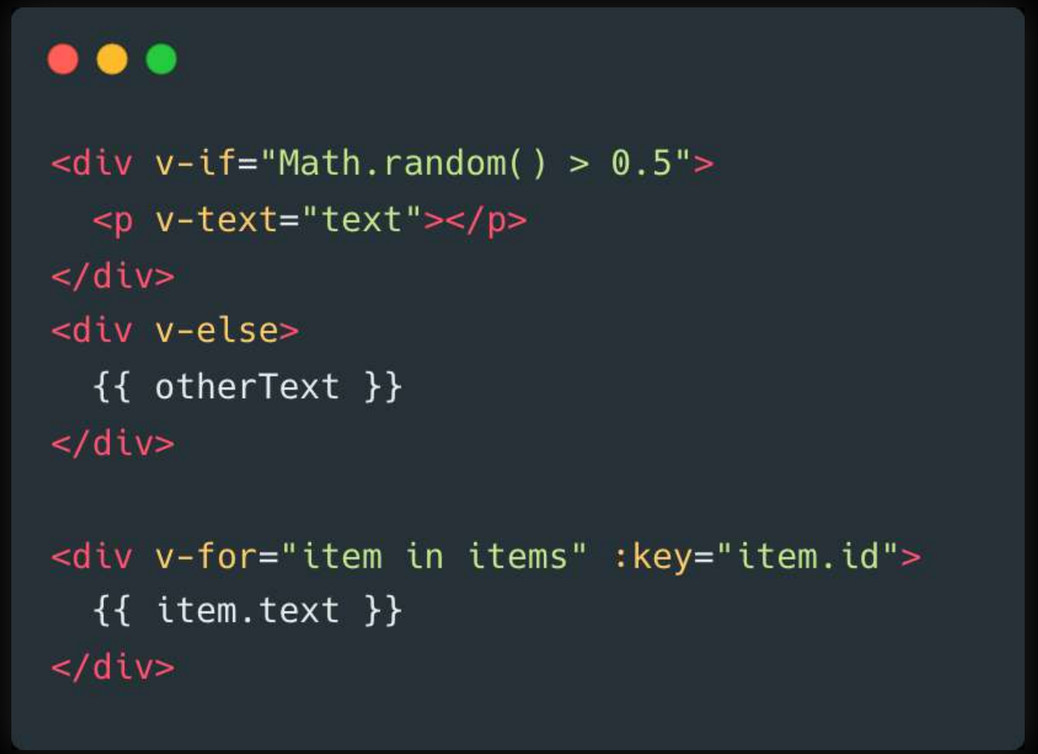
SQUARE 1 ACADEMY

Conceptos básicos

Vue Js - Nuxt Js

features

- Single File Component
- Virtual DOM
- **Directivas**
- Unidireccionalidad de datos



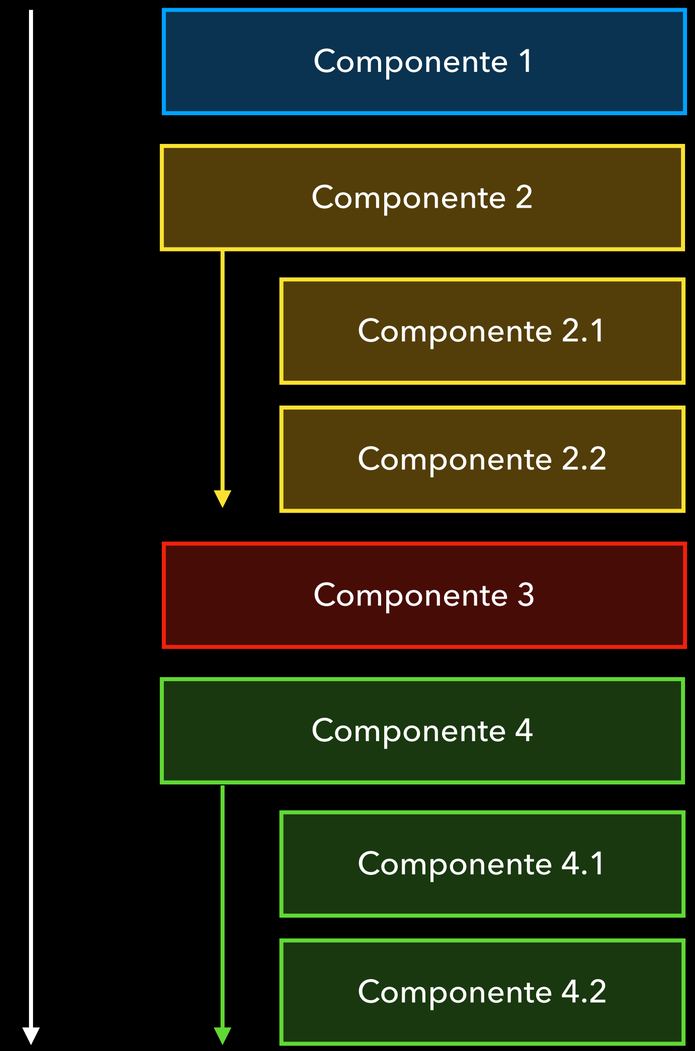
SQUARE1 ACADEMY

Conceptos básicos

Vue Js - Nuxt Js

features

- Single File Component
- Virtual DOM
- Directivas
- **Unidireccionalidad de datos**



SQUARE1 ACADEMY

Conceptos básicos

Astro

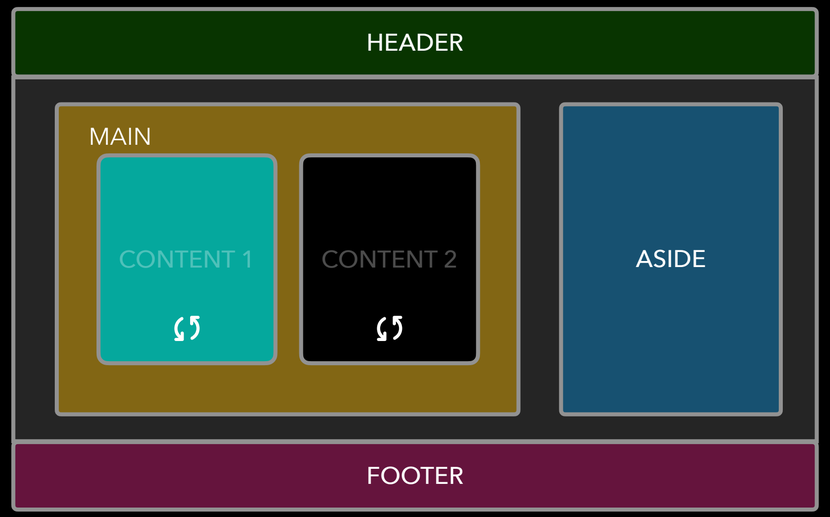
Es una framework para construir **sitios orientados a contenido**. Pionero de *island architecture*.

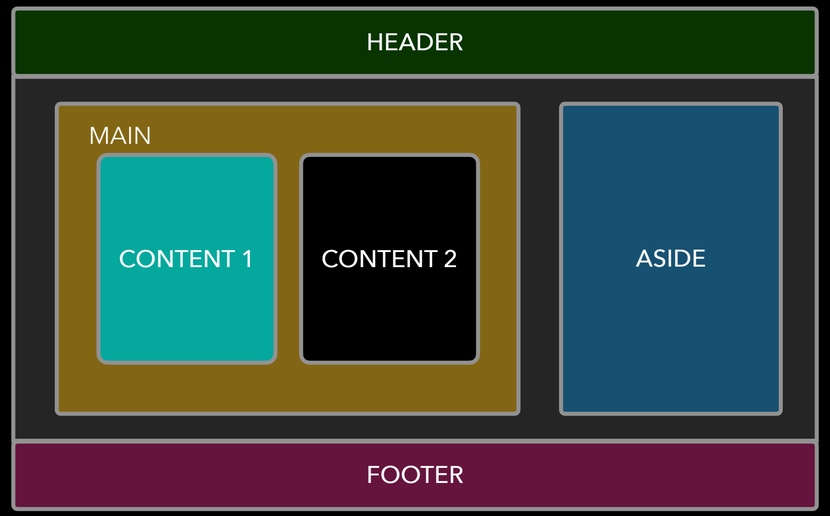
Conceptos básicos

Vue Js - Nuxt Js

features

- **Islas**
- Agnostico a la UI
- Servidor primero
- Cero Js





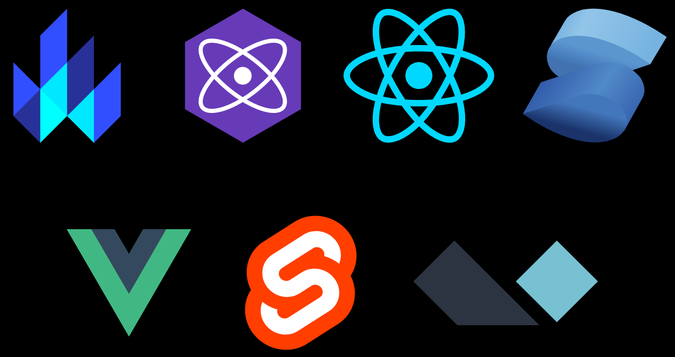
SQUARE1 ACADEMY

Conceptos básicos

Vue Js - Nuxt Js

features

- Islas
- **Agnostico a la UI**
- Servidor primero
- Cero Js



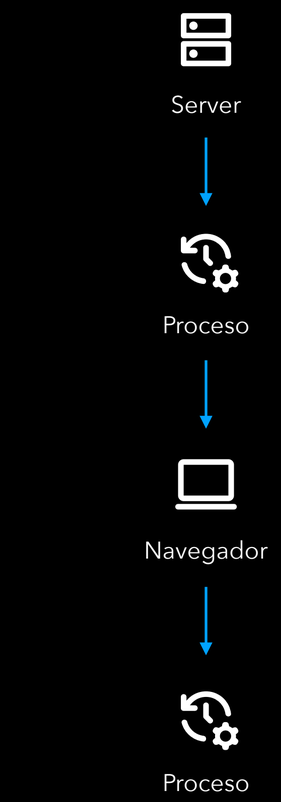
SQUARE1 ACADEMY

Conceptos básicos

Vue Js - Nuxt Js

features

- Islas
- Agnostico a la UI
- **Servidor primero**
- Cero Js



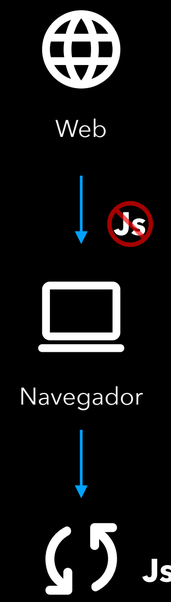
SQUARE1 ACADEMY

Conceptos básicos

Vue Js - Nuxt Js

features

- Islas
- Agnostico a la UI
- Servidor primero
- **Cero Js**



SQUARE1 ACADEMY

JAIME BURBANO

jaime@square1.io

SQUARE1 ACADEMY