

REACT PARA PRINCIPIANTES

TEMA 1: FUNDAMENTOS

Material obtenido de Open Webinars

TEMAS DEL CURSO

REACT PARA

PRINCIPIANTES



1. **FUNDAMENTOS:** qué saber antes de iniciarse con React
2. **SETUP:** crear un proyecto React desde cero.
3. **RENDERIZADO:** cómo aprovechar las capacidades de renderizado de React
4. **PROPS & STATE:** comunicación de componentes

ÍNDICE DEL TEMA 1

INTRODUCCIÓN A REACT. 1 - FUNDAMENTALS



Design systems



Modern Javascript



React anatomy



React Ecosystem

SISTEMAS DE DISEÑO

ÍNDICE

INTRODUCCIÓN A REACT. 1 - FUNDAMENTALS

Design systems



History

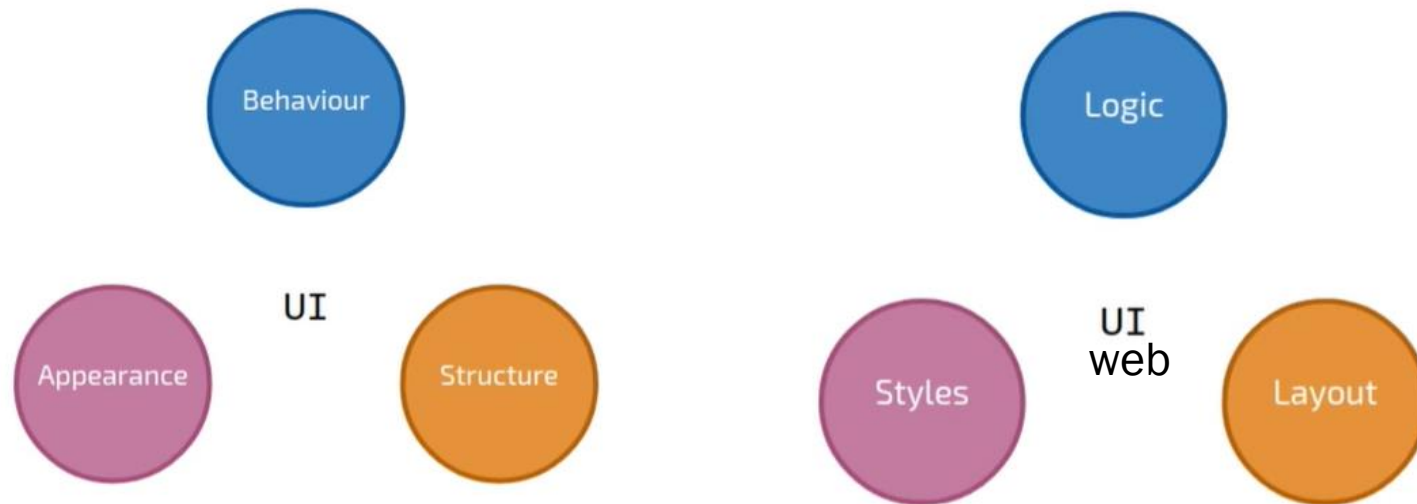


Encapsulation



Atomic Design

SISTEMAS DE DISEÑO: HISTORIA



- Cualquier interfaz de usuario es una combinación de estos 3 factores:
 - Estructura/Layout: relación entre las partes
 - Comportamiento/Lógica: cómo responde a nuestras acciones y cómo interaccionan las partes
 - Apariencia/Estilos: estética, usabilidad



Inicialmente... —————> Algo más evolucionado...

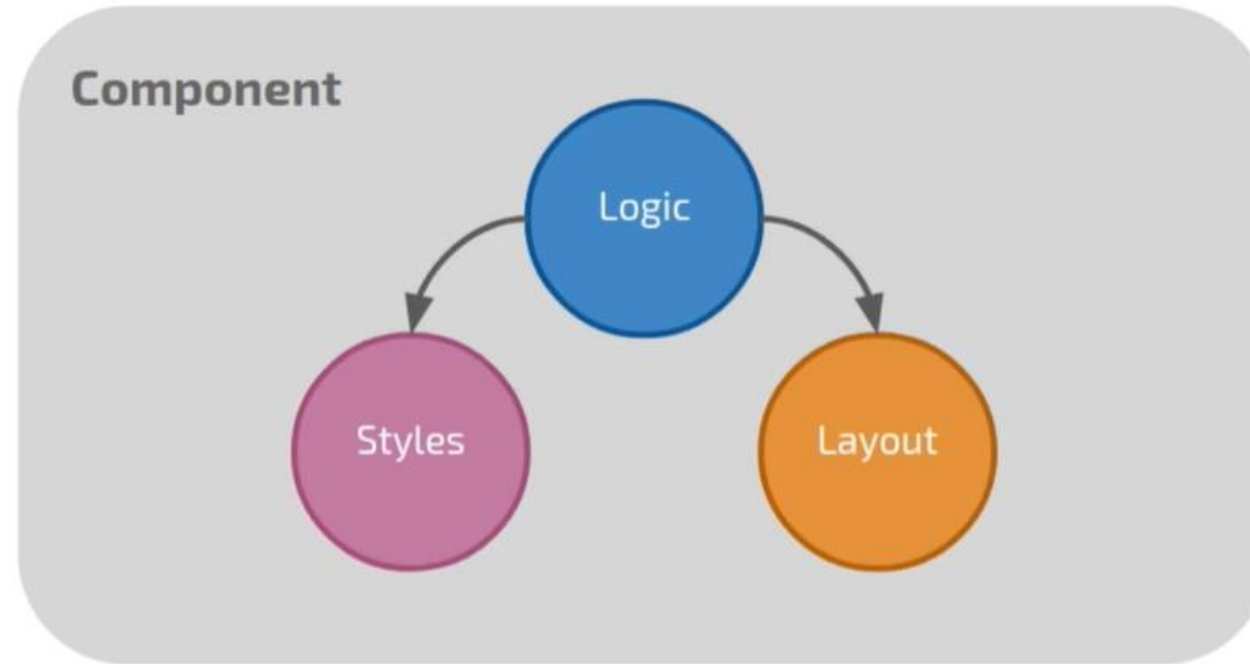
FRONT END LÓGICA TRADICIONAL

“ We're not designing pages.
We're designing systems of components.”

Stephen Hay - Author of Responsive Design Workflow

ARQUITECTURA BASADA EN COMPONENTES

- Evolución del sistema de páginas tradicional
- Tendremos componentes encapsulados y reutilizables



SISTEMAS DE DISEÑO: ENCAPSULACIÓN

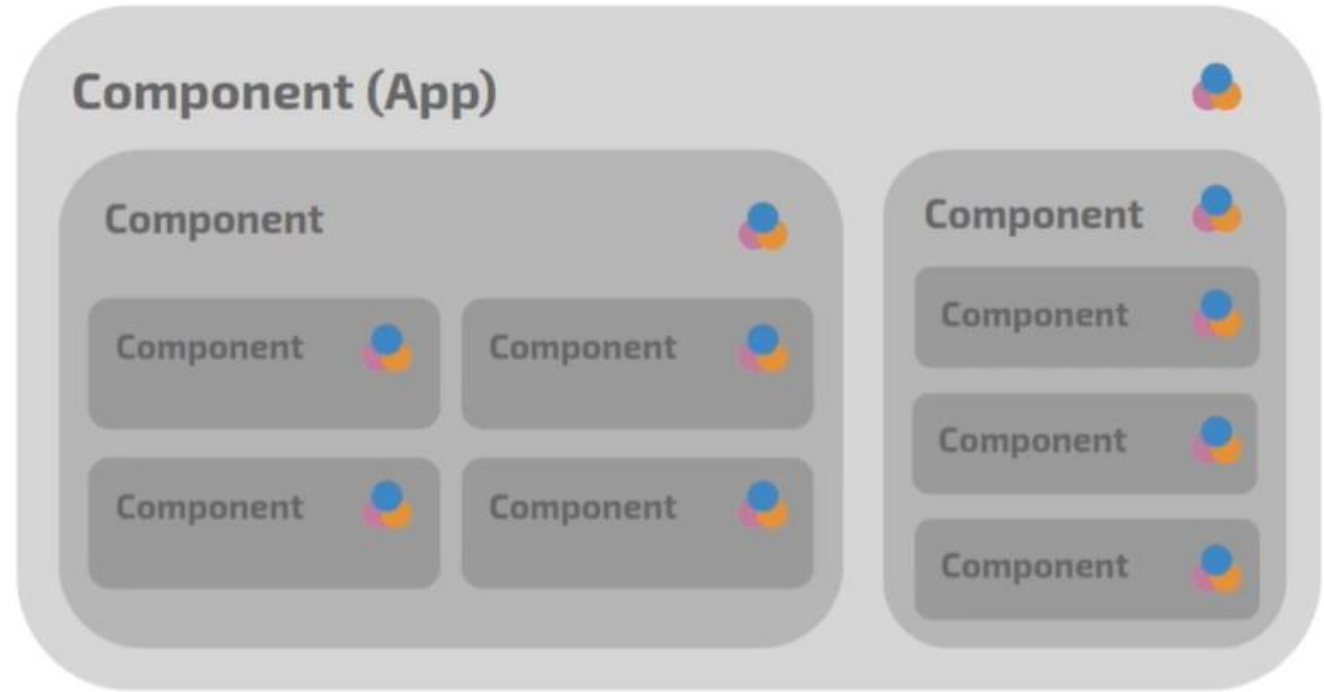
- Nuestra unidad lógica a partir de ahora va a ser el componente
- El usuario verá páginas pero nosotros trabajaremos con componentes encapsulados

ESTRUCTURA DE ARQ. BASADA EN COMPONENTES

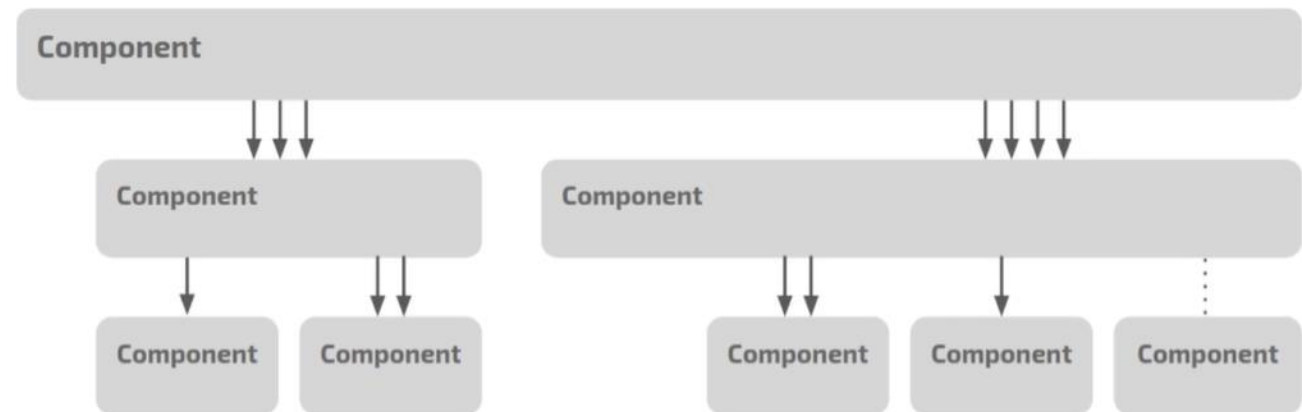
- Ahora tendremos más bien una estructura horizontal donde cada capa tiene un trocito de estructura



ARQUITECTURA BASADA EN COMPONENTES



Similar al DOM



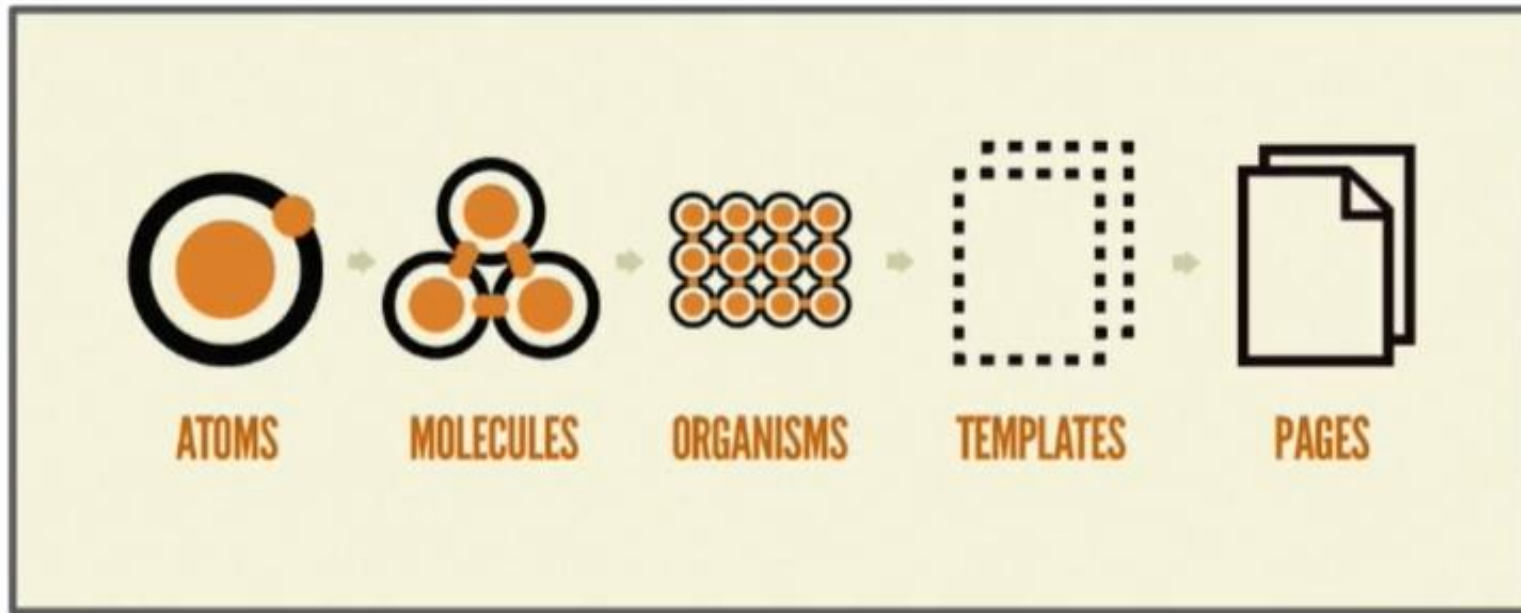


Image from atomicdesign.bradfrost.com

ATOMIC DESIGN

- Va más allá de la arquitectura de componentes, es una sofisticación.
- Intentamos alcanzar este patrón de diseño usando componentes

ATOMIC DESIGN: ÁTOMOS

ejemplo

- Cada parte es un átomo

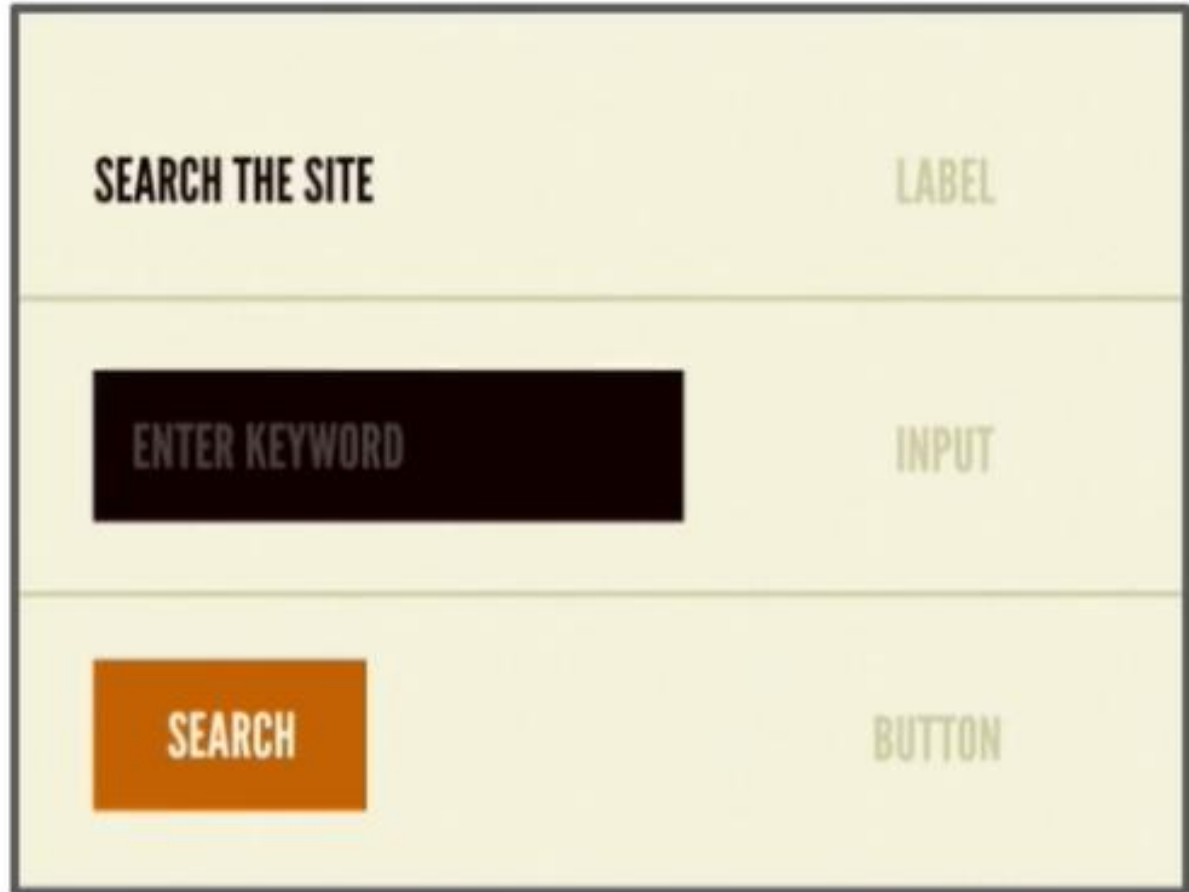


Image from atomicdesign.bradfrost.com

ATOMIC DESIGN: MOLÉCULA ejemplo

- Combinamos átomos en una molécula

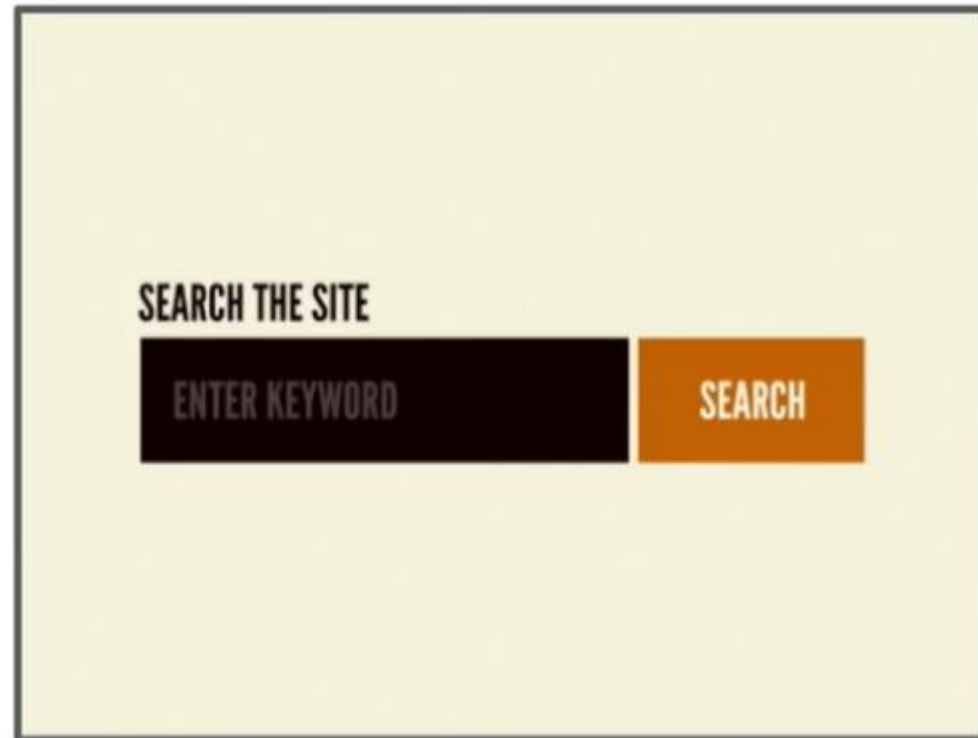


Image from atomicdesign.bradfrost.com

ATOMIC DESIGN: ORGANISMO

ejemplo

- Incluimos la molécula en nuestro organismo cabecera



ATOMIC DESIGN: TEMPLATE

ejemplo

- Juntamos organismos y tenemos templates

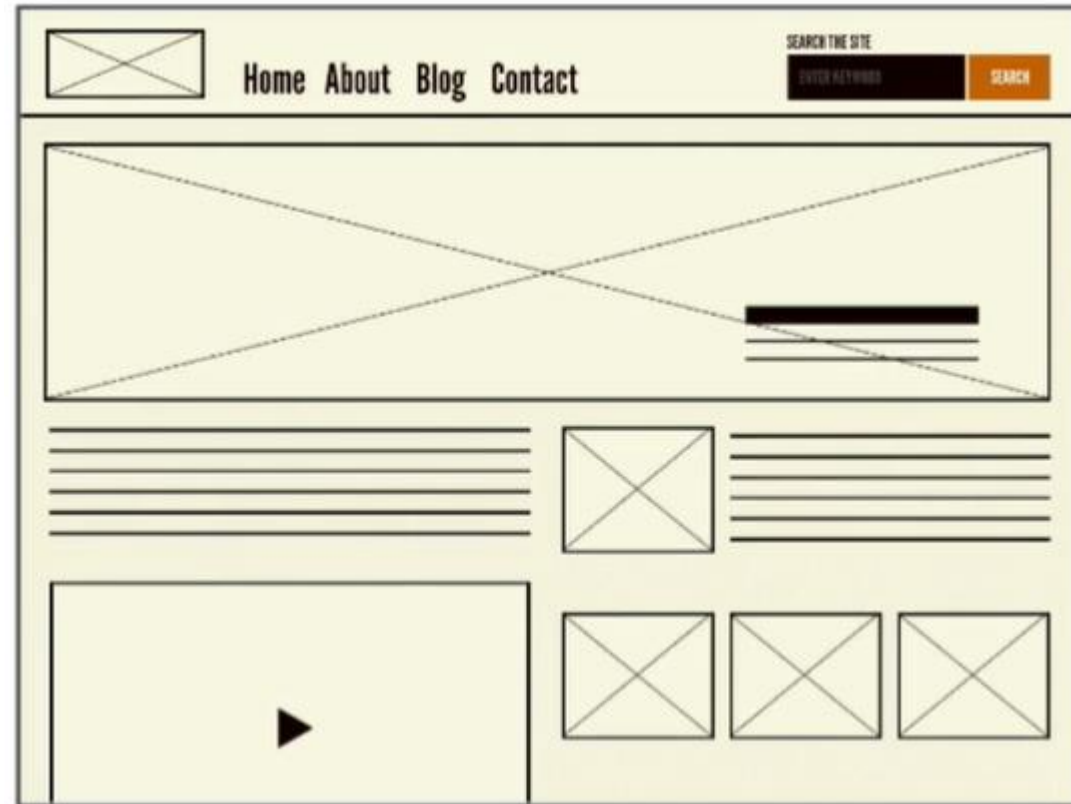


Image from atomicdesign.bradfrost.com

ATOMIC DESIGN: PÁGINA ejemplo

- Una instancia de una plantilla (o template) es una página

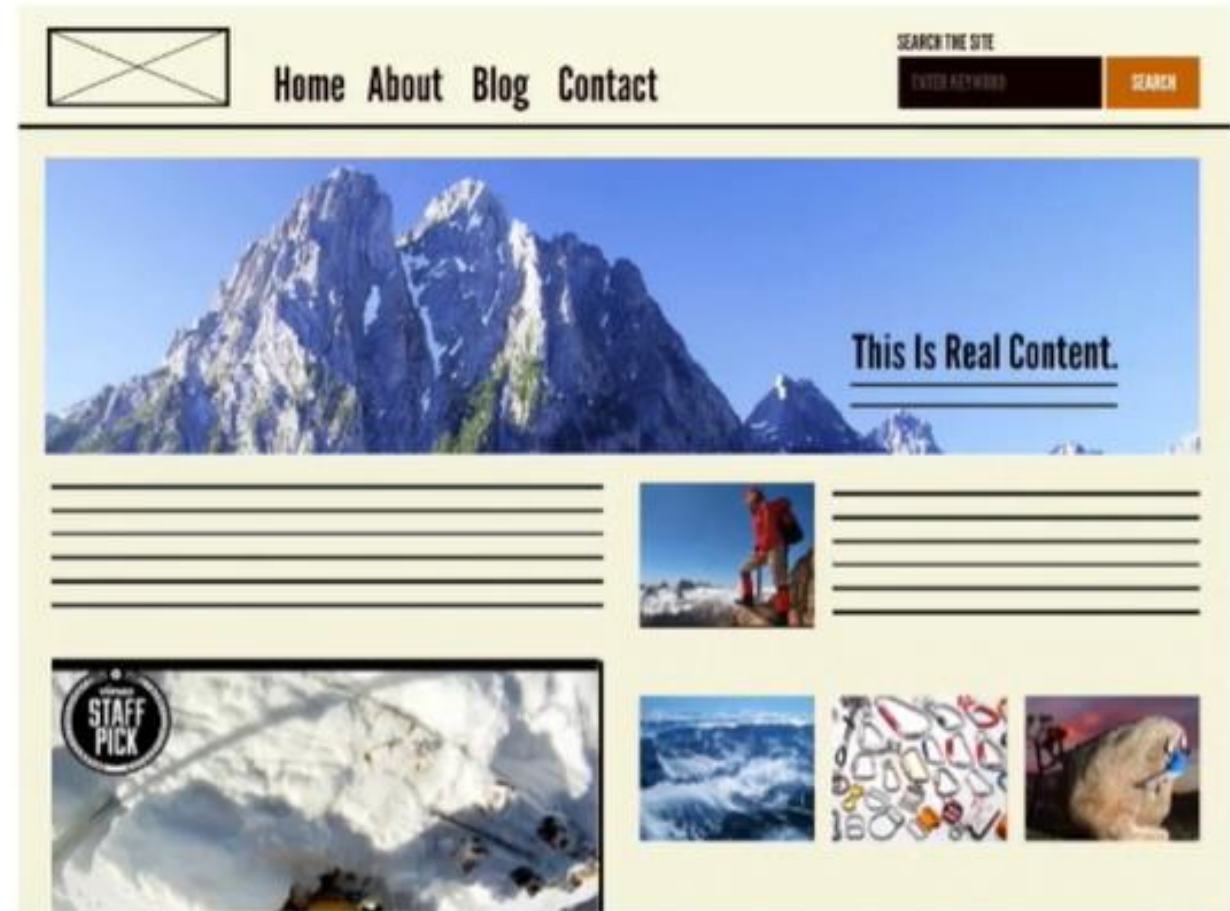
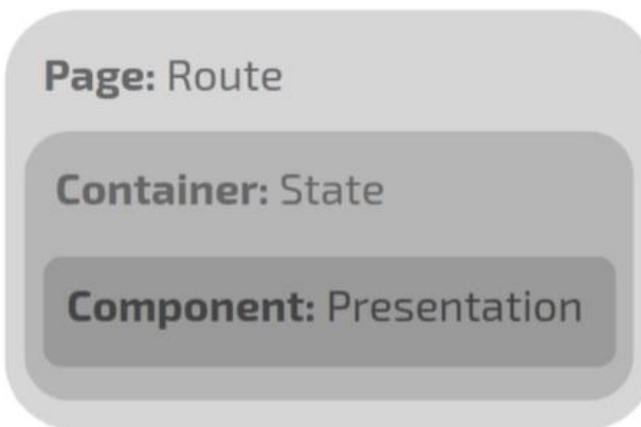
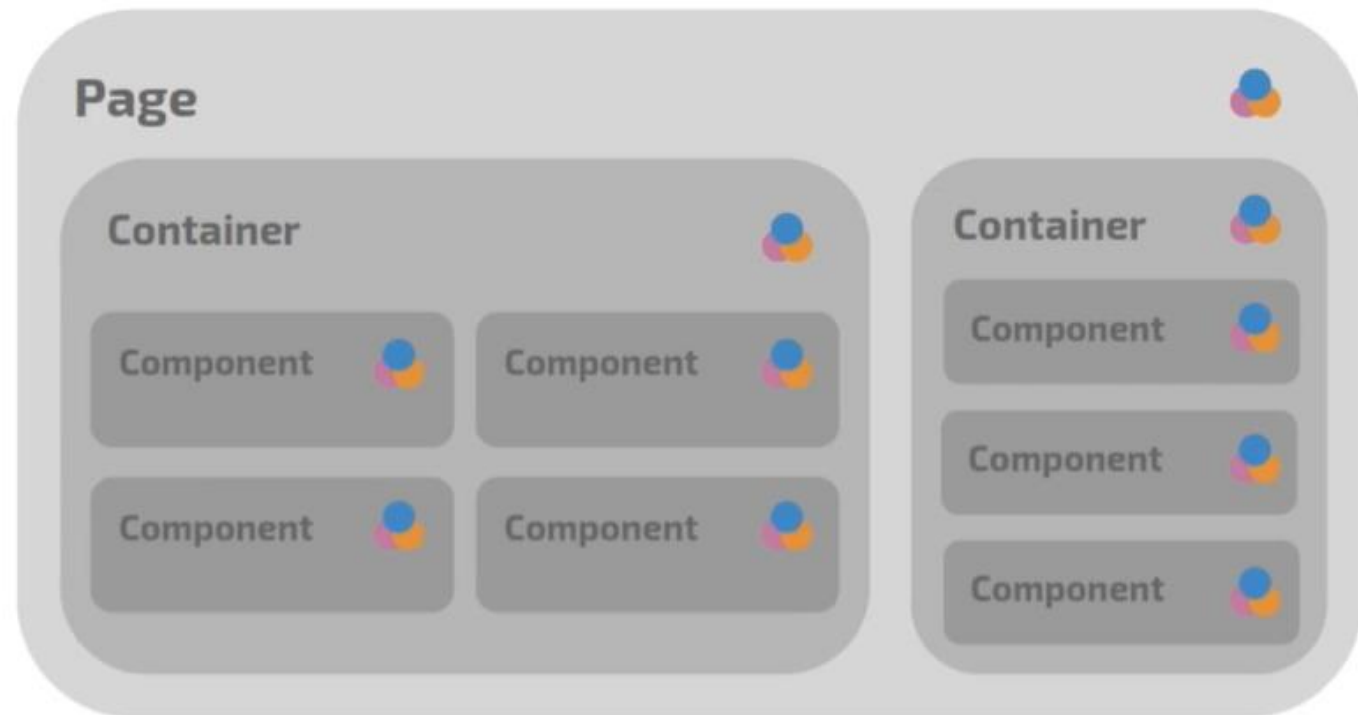


Image from atomicdesign.bradfrost.com

ATOMIC DESIGN EN REACT

- Simplifica un poco el atomic design pero sigue la misma filosofía



- Las rutas las gestionan las páginas
- El estado lo gestionan los contenedores
- Los componentes se encargan de la presentación

JAVASCRIPT AVANZADO PARA REACT

INTRODUCCIÓN A REACT. 1 - FUNDAMENTALS



Identifiers



Arrow Functions



String literals



Destructuring



Classes



Modules

ANATOMÍA DE REACT



Reconciliation



Renderization

Reconciliación: Proceso mediante el cual intenta preparar los datos para la posterior renderización

RECONCILIACIÓN REACT

TRADICIONALMENTE..

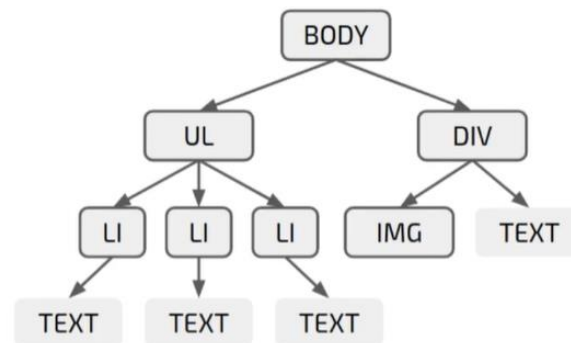


View

v1

```
<body>
  <ul>
    <li>One </li>
    <li>Two </li>
    <li>Three </li>
  </ul>
  <div class="my-class">
    </script>
    Some image description
  </div>
</body>
```

DOM

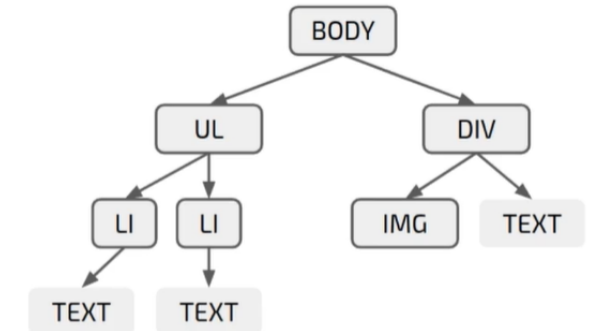


View

v2

```
<body>
  <ul>
    <li>One </li>
    <li>Two </li>
  </ul>
  <div class="my-class">
    </script>
    Some image description
  </div>
</body>
```

DOM



Este repintado es costoso si el árbol es muy grande.
Por ello aparece el concepto de VIRTUAL DOM

RECONCILIACIÓN REACT



El Virtual DOM es una copia del DOM original.

En el virtual DOM puedo hacer estrategias más sofisticadas y solo ataco al DOM nativo cuando sea necesario.

Evito sobrecargar el DOM

RECONCILIACIÓN REACT

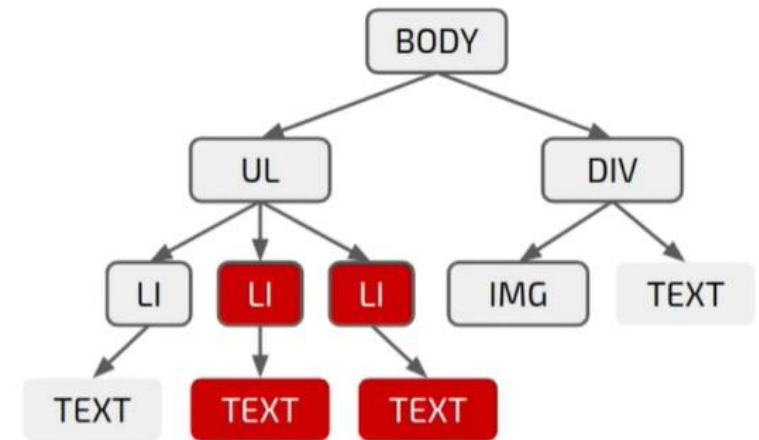
- Actualización menos costosa

View

v1

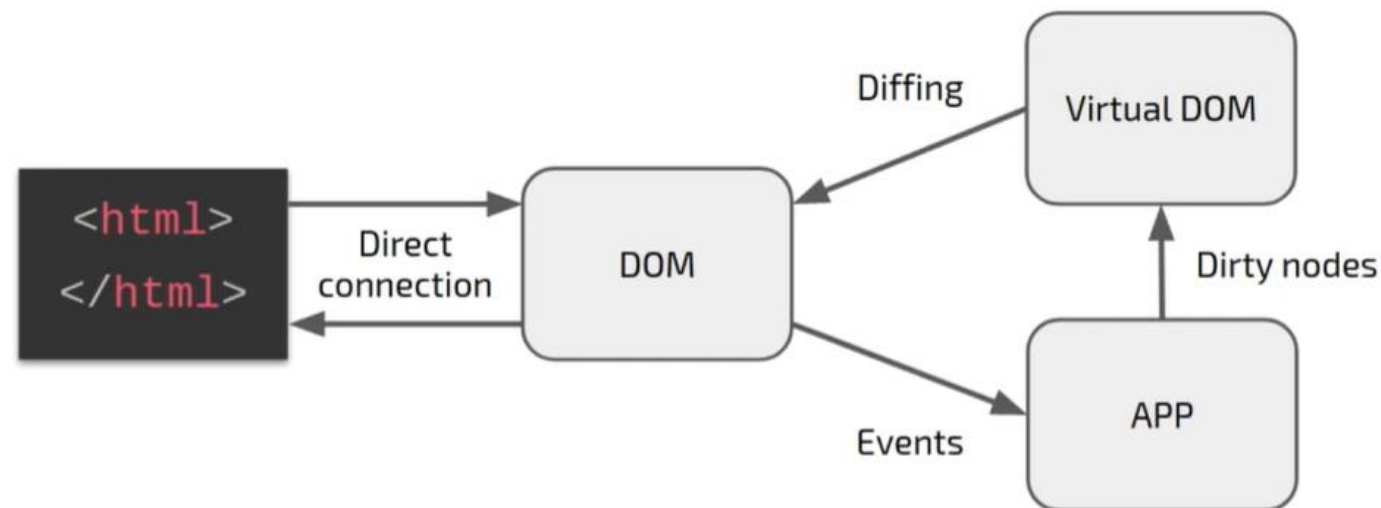
```
<body>
  <ul>
    <li key="1">One </li>
    <li key="2">Two </li>
    <li key="3">Three </li>
  </ul>
  <div class="my-class">
    </script>
    Some image description
  </div>
</body>
```

VIRTUAL DOM

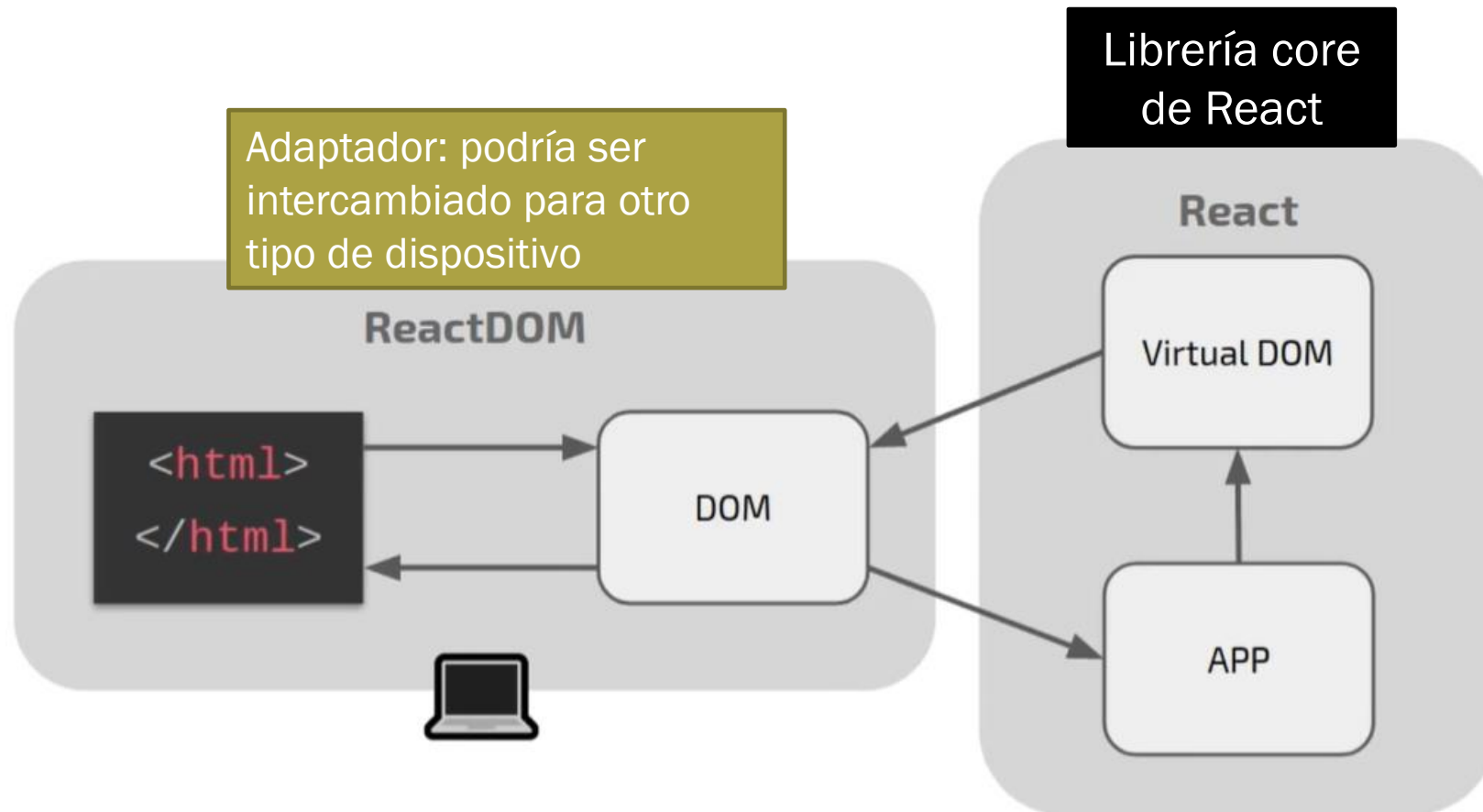


RECONCILIACIÓN REACT

- Por cada evento del DOM nativo, la app hará un cálculo para obtener los **Dirty Nodes** (nodos posiblemente afectados por los cambios) y se los pasa al Virtual DOM para que localice los puntos diferentes entre el renderizado anterior y el siguiente y los devuelva al DOM (**Diffing**).

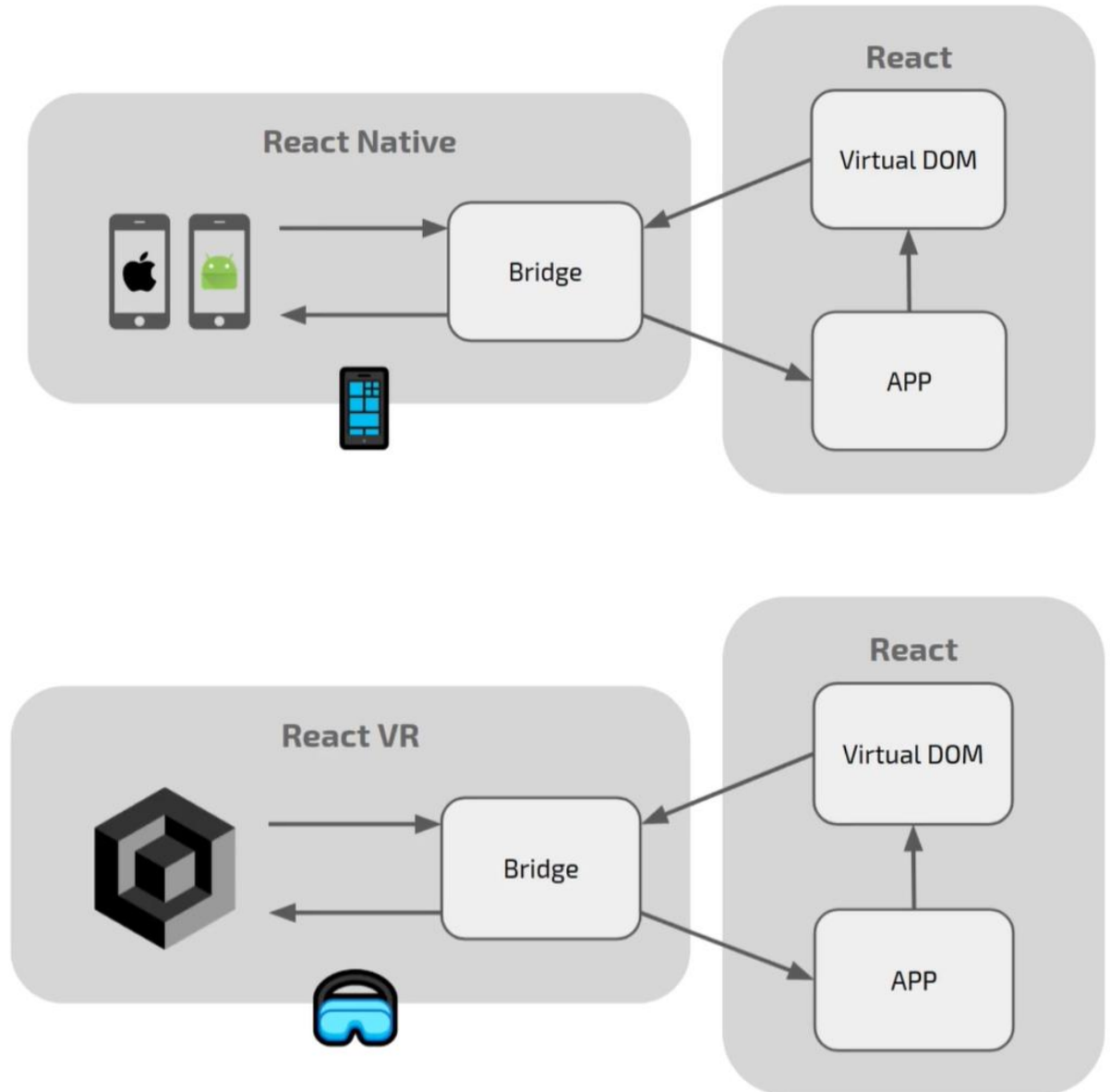


RENDERIZACIÓN REACT

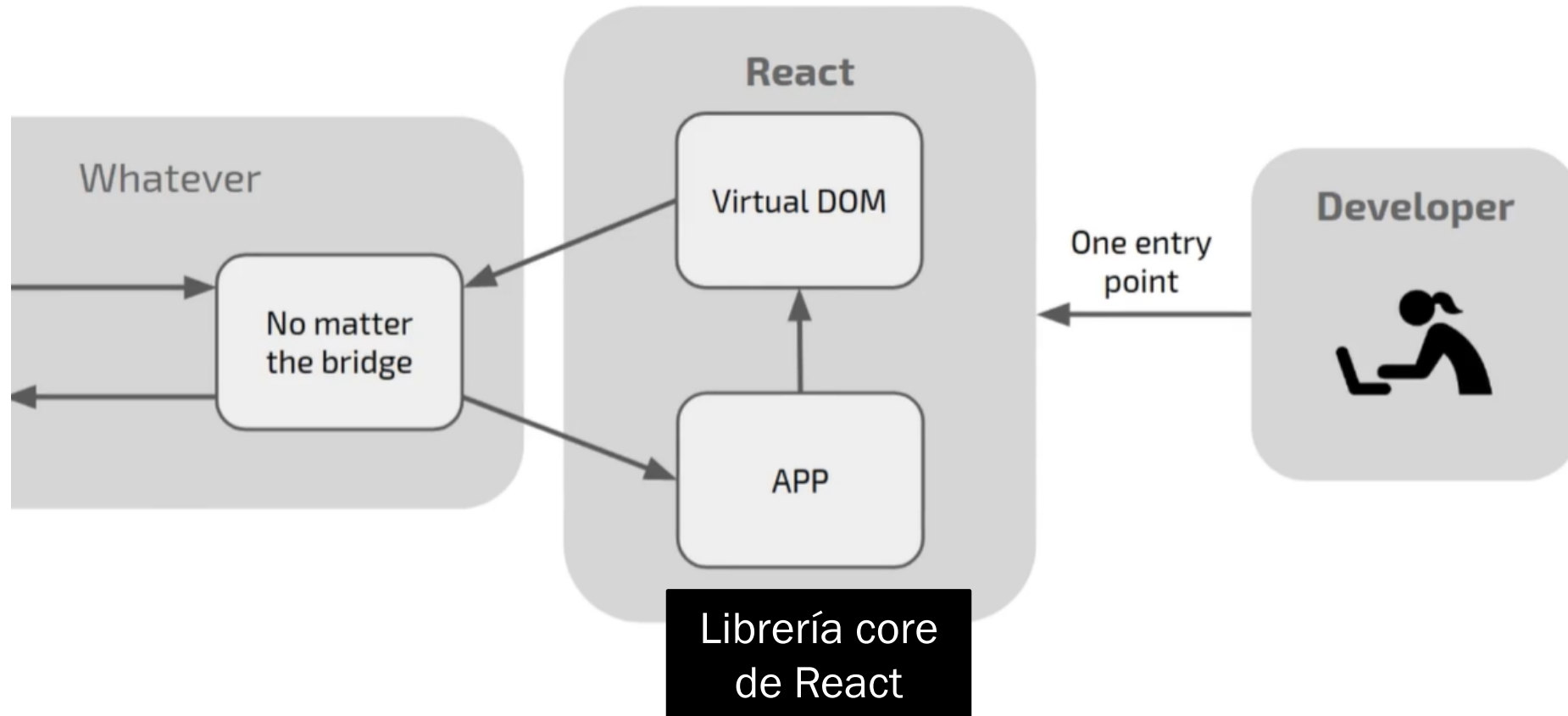


OTROS ADAPTADORES DE RENDERIZADO:

REACT NATIVE Y REALIDAD VIRTUAL

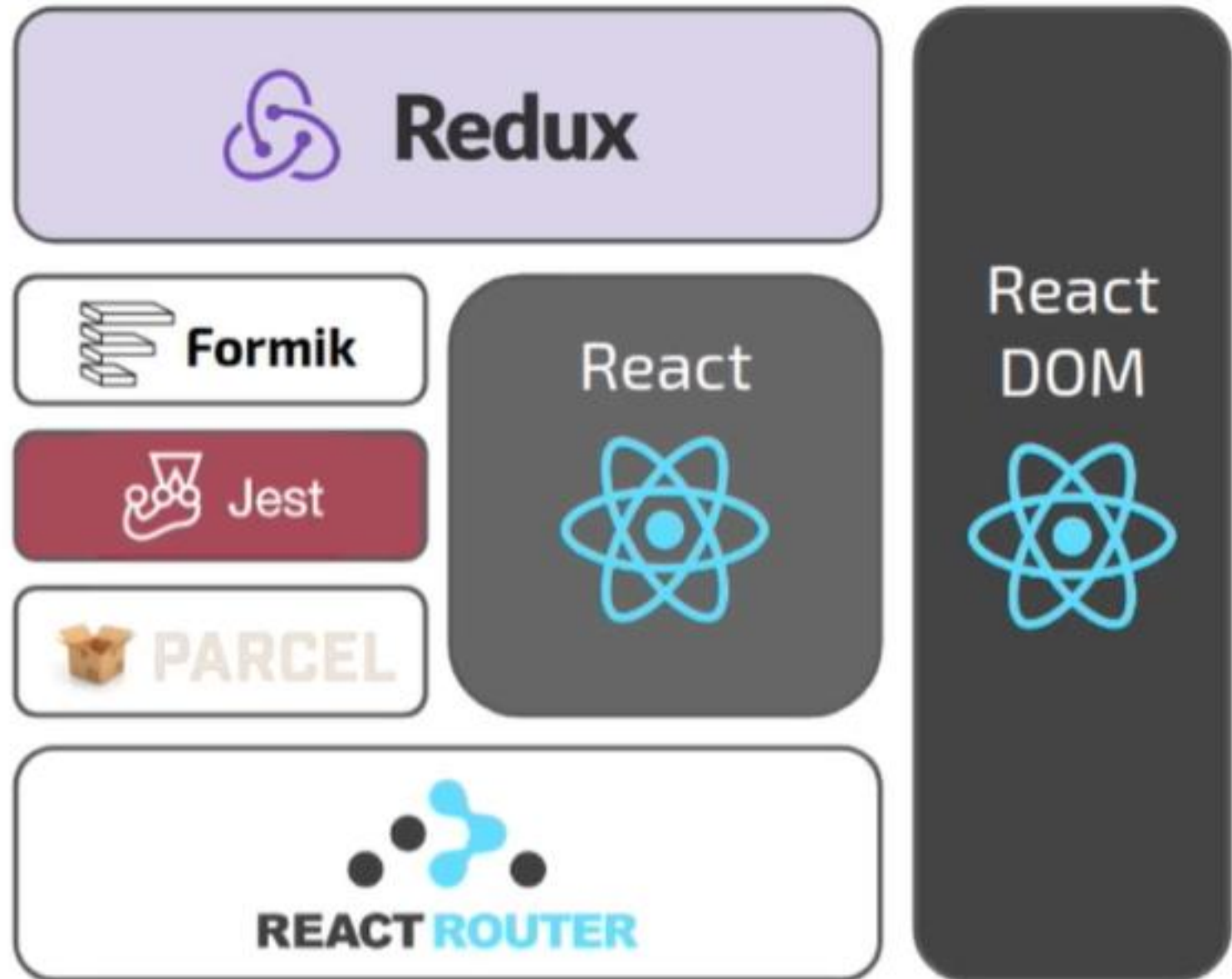


RENDERIZACIÓN REACT



ECOSISTEMA REACT TÍPICO

- React Core
- Renderizado: p.ej. React DOM
- Routing: p.ej. Librería React Router
- Gestión del estado: p. ej. Librería Redux
- Gestión de formularios: p. ej. Formik
- Tests de componentes: p. ej. Librería Jest
- Empaquetado de componentes: p.ej. Parcel





FIN DEL TEMA 1

A CONTINUACIÓN: TEMA 2 SETUP