

La Web Orientada a Componentes

El nacimiento de un Paradigma

Javier Vélez Reyes

@javiervelezreye

Javier.velez.reyes@gmail.com

Noviembre 2015

{codemotion}

La Web Orientada a Componentes

Autor

¿Quién Soy?



Licenciado por la UPM desde el año 2001 y doctor en informática por la UNED desde el año 2009, Javier conjuga sus labores como profesor e investigador con la consultoría y la formación técnica para empresa. Su línea de trabajo actual se centra en la innovación y desarrollo de tecnologías para la Web. Además realiza actividades de evangelización y divulgación en diversas comunidades IT y es coordinador de varios grupos de ámbito local como NodeJS Madrid o Madrid JS. Forma parte del programa Polymer Polytechnic Speaker y es mentor del capítulo de Madrid de Node School.



javier.velez.reyes@gmail.com



@javiervelezreye



linkedin.com/in/javiervelezreyes



gplus.to/javiervelezreyes



jvelez77



javiervelezreyes



youtube.com/user/javiervelezreyes



Javier Vélez Reyes

@javiervelezreye

Javier.velez.reyes@gmail.com

1 *Introducción*

- Qué Son Los Componentes Web
- Por Qué Componentes Web
- Cómo Operan Los Componentes Web

La Web Orientada a Componentes

Introducción



Qué Son Los Componentes Web

Etiquetas de Autor

“

La tecnología de Componentes Web proporciona un mecanismo para construir nuevas etiquetas de autor personalizadas que incluyen una semántica, un comportamiento funcional y una lógica de presentación propia.

”

La Web Orientada a Componentes

Introducción



Por Qué Componentes Web

La Reutilización como Objetivo

R

reutilización

La Web Orientada a Componentes

Introducción

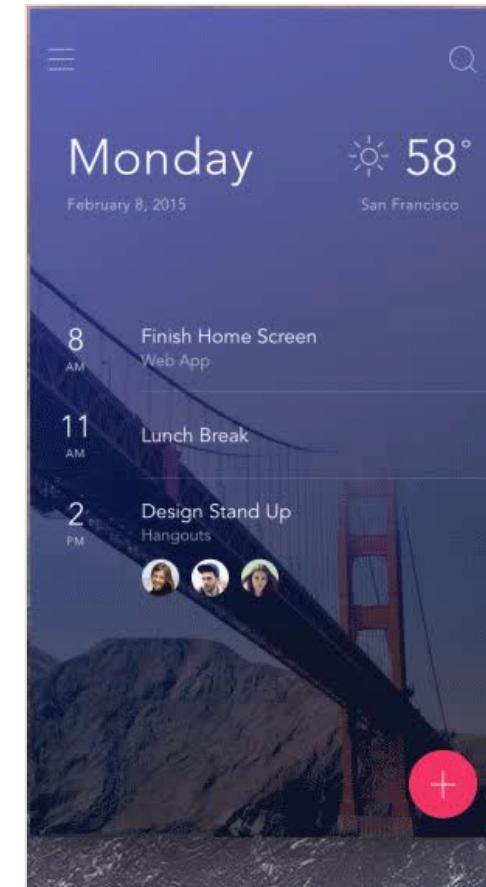
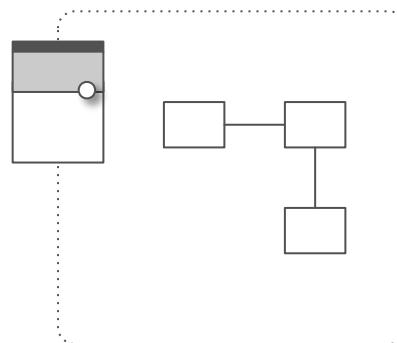


Por Qué Componentes Web

La Reutilización como Objetivo



*Se encapsulan en forma de componentes reutilizables **modelos de interacción** de aplicación recurrente en distintos contextos de uso*



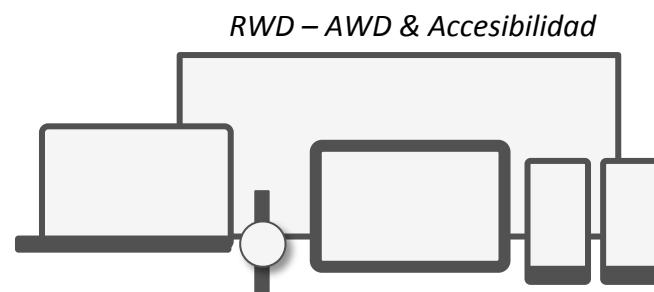
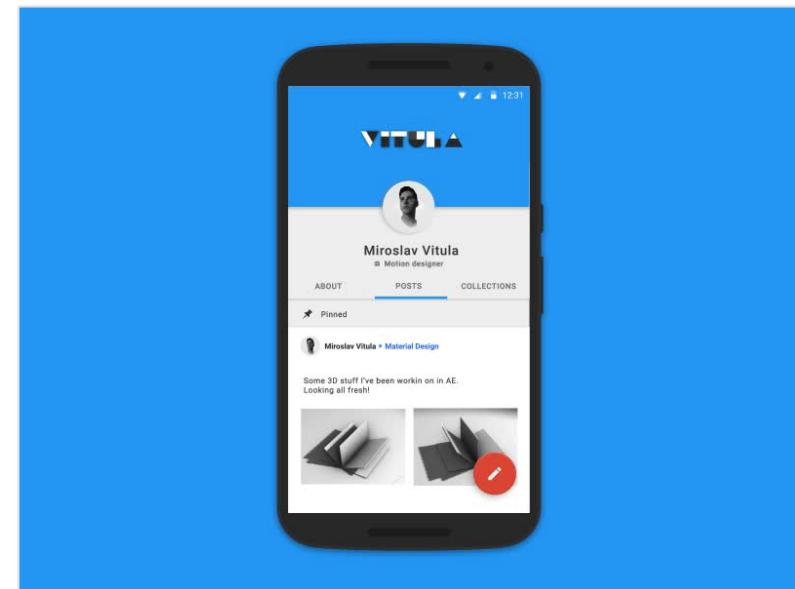
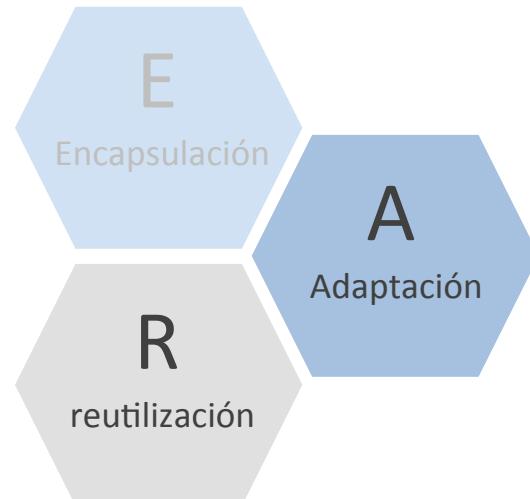
La Web Orientada a Componentes

Introducción



Por Qué Componentes Web

La Reutilización como Objetivo



Cada componente es responsable de gestionar el **espacio visual** que ocupa el contenido en relación con las condiciones del **dispositivo**

La Web Orientada a Componentes

Introducción

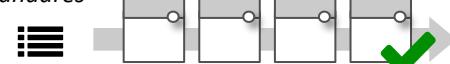


Por Qué Componentes Web

La Reutilización como Objetivo



Estándares



*La homogeneidad visual es imprescindible para articular un **proceso compositivo** adecuado. La estandarización es la forma de obtenerlo*

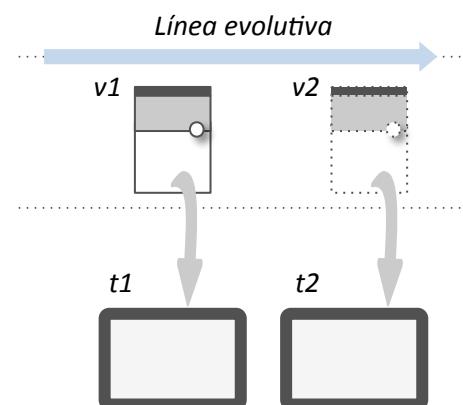
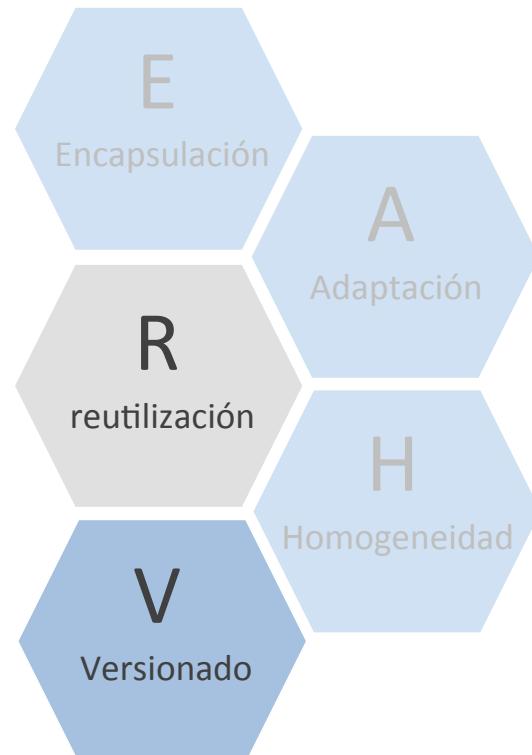
La Web Orientada a Componentes

Introducción



Por Qué Componentes Web

La Reutilización como Objetivo



*Las actividades de mantenimiento evolutivo son gestionadas a través del **cambio centralizado** de los componentes Web*

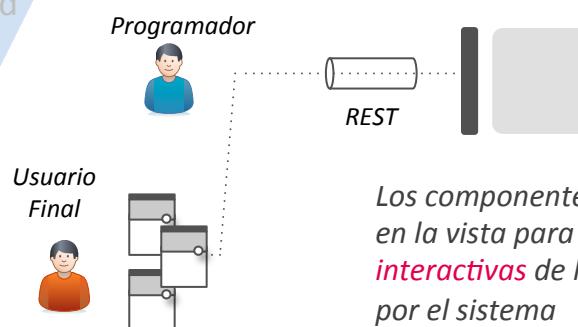
La Web Orientada a Componentes

Introducción



Por Qué Componentes Web

La Reutilización como Objetivo



*Los componentes web están centrados en la vista para ofrecer **proyecciones interactivas** de las capacidades ofrecidas por el sistema*

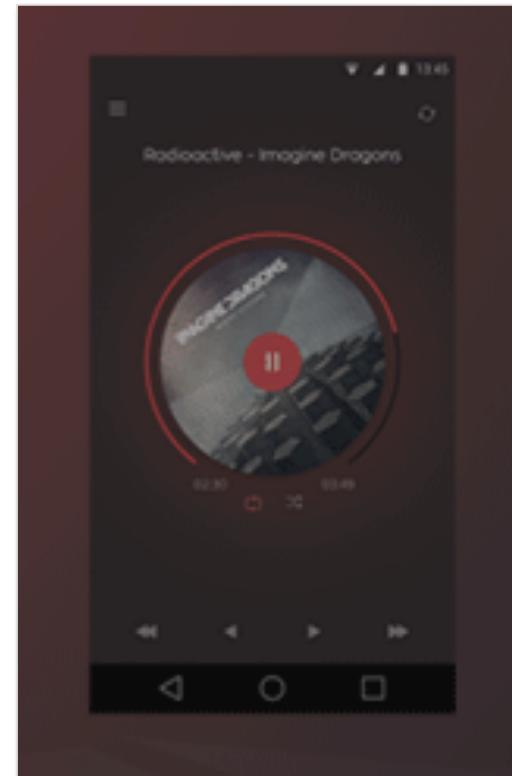
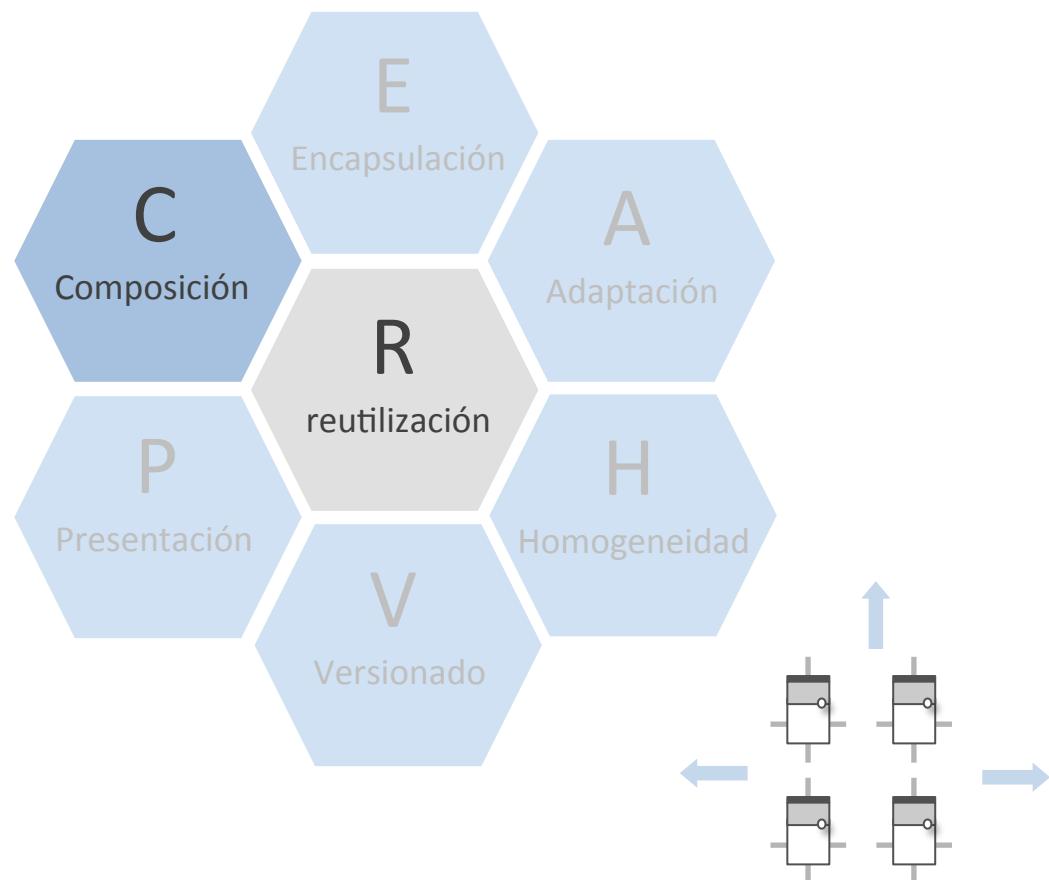
La Web Orientada a Componentes

Introducción



Por Qué Componentes Web

La Reutilización como Objetivo



Cada componente opera visualmente sin imponer fronteras de contexto

La Web Orientada a Componentes

Introducción



Cómo Operan los Componentes Web

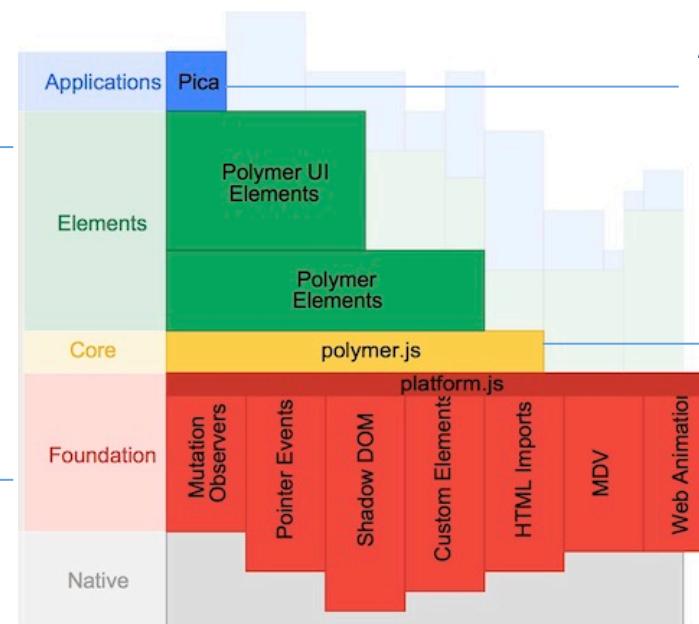
Perspectiva General

Elementos

Componentes estándar dedicados a dar respuesta a necesidades recurrentes de control o UI

Plataforma

Incluye todas las nuevas capacidades *estándar* que extienden la Web como plataforma de orientación a componentes



Aplicaciones

Aplicaciones orientadas a componentes al servicio de los usuarios de la Web

Polymer

Capacidades funcionales sobre la plataforma que *simplifican* el proceso de desarrollo de Componentes Web

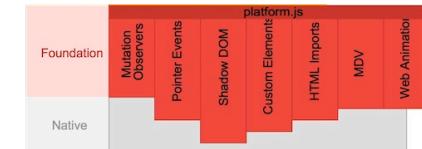
La Web Orientada a Componentes

Introducción



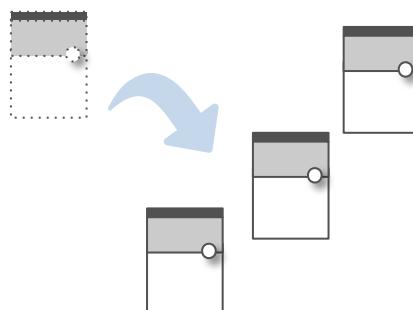
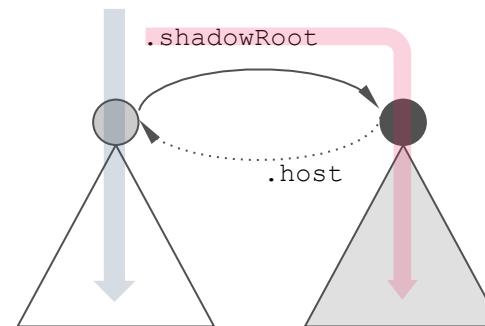
Cómo Operan los Componentes Web

4 Estándares Web



Shadow DOM

Ofrece un modelo de *encapsulación* que permite aislar el contenido interno del componente de aquél en la página donde éste es renderizado



HTML Templates

Ofrece un modelo de *renderización* basado en plantillas inertes de código HTML que sólo son activadas cuando se renderiza el componente

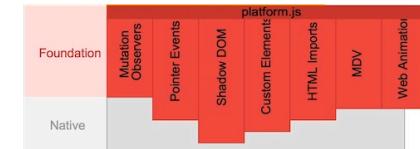
La Web Orientada a Componentes

Introducción



Cómo Operan los Componentes Web

4 Estándares Web



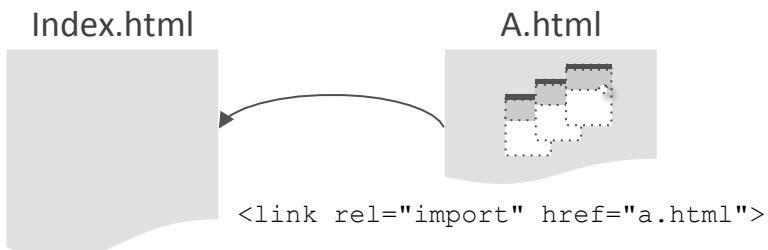
</> *Custom Elements*

Ofrece un modelo de *extensibilidad* que permite a los desarrolladores definir sus propias etiquetas o redefinir semánticamente las etiquetas del estándar HTML

Estándar
HTML

<google-maps>


WC
Etiquetas de autor



HTML Imports

Ofrece un modelo de *modularización* y empaquetamiento basado en la posibilidad de incluir ficheros de código HTML dentro de otros ficheros HTML

La Web Orientada a Componentes

Introducción



Cómo Operan los Componentes Web

El Framework Polymer

Sistema de Data-Binding

El sistema de data binding de polymer mantiene **sincronizada** la vista con los cambios en el estado del componente y recíprocamente

```
<dom-module id="wc-test">  
  <template>  
    <div>{{x}}</div>  
  </template>  
  
  <script>  
    Polymer ({  
      is: 'wc-test',  
      ready: function () {  
        this.x = 7;  
      }  
    });  
  </script>  
</dom-module>
```



```
<dom-module is="dom-repeat" items="{{its}}">  
  <template>  
    <div>  
      <div>{{item}}</div>  
    </div>  
  </template>  
</dom-module>
```

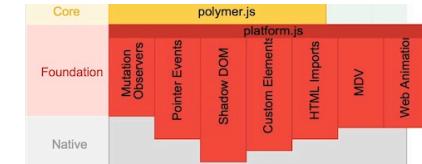


```
<dom-module is="dom-if" items="{{exp}}">  
  <template>  
    <div>  
      <div>{{item}}</div>  
    </div>  
  </template>  
</dom-module>
```



```
...
```

```
<dom-module id="wc-test">  
  <template>  
    <div>{{x}}</div>  
  </template>  
  
  <script>  
    Polymer ({  
      is: 'wc-test',  
      ready: function () {  
        this.x = 7;  
      }  
    });  
  </script>  
</dom-module>
```



Motor de Plantillas

Polymer también ofrece **un motor de renderizado basado en plantillas con lógica para representación de contenido iterativa y condicional**

La Web Orientada a Componentes

Introducción

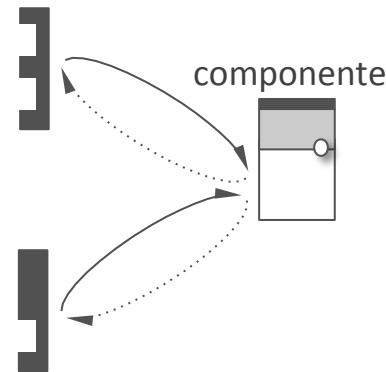


Cómo Operan los Componentes Web El Framework Polymer

Behaviors

*Los comportamientos de Polymer son **rasgos parciales** definidos de manera externa que pueden incorporarse como contribuciones a los componentes de forma declarativa*

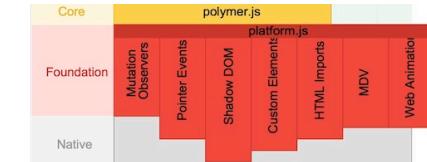
Rasgos recurrentes



```
this.$  
this.$$ (selector)  
this.fire (event, ctx)  
this.async (fn, [ms])  
Polymer.dom (node)  
...
```

Utilidades

*Polymer también ofrece un abanico de facilidades y **funciones utilitarias** que vienen a simplificar el proceso de desarrollo de componentes*



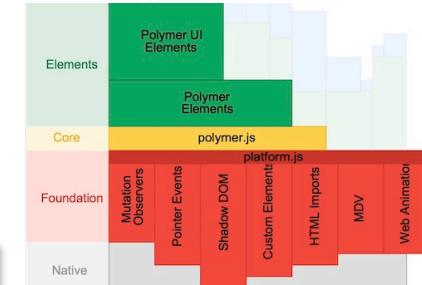
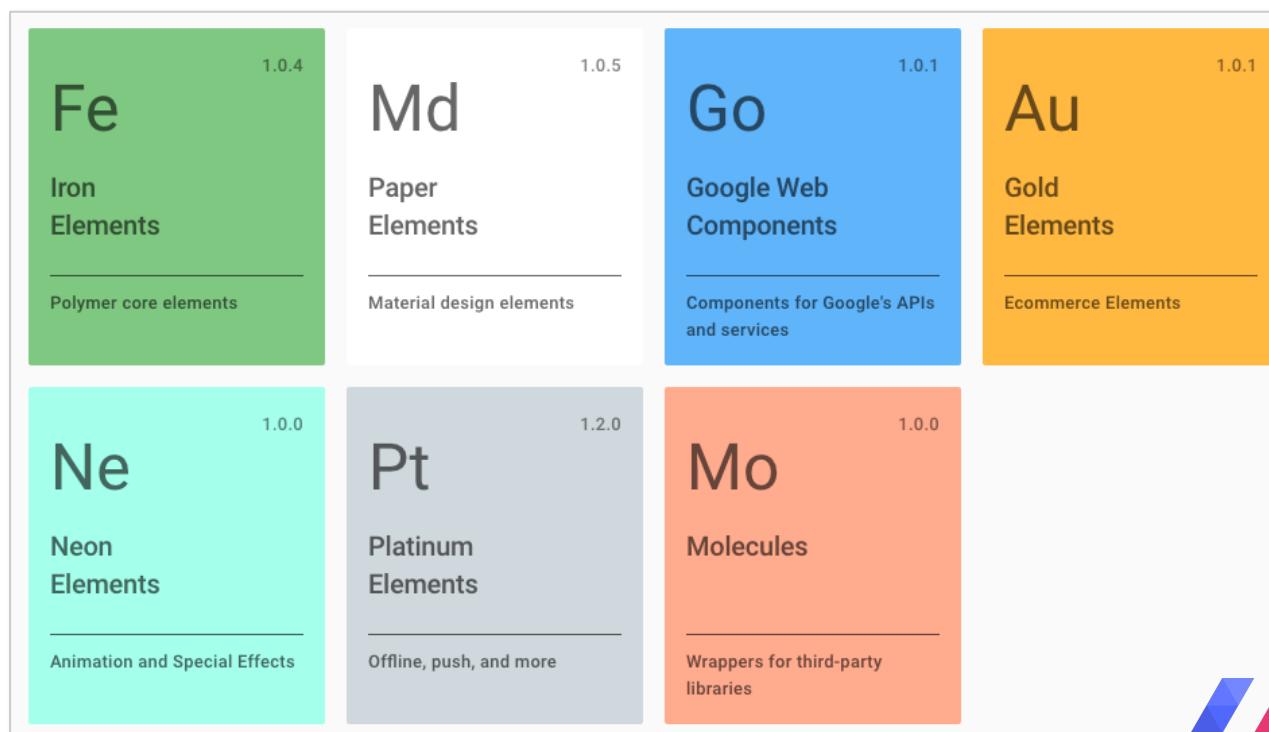
La Web Orientada a Componentes

Introducción



Cómo Operan los Componentes Web

Ecosistema de Componentes Web



Google
Polymer Project

La Web Orientada a Componentes

Introducción



Cómo Operan los Componentes Web

Aplicaciones de Componentes Web

I. Plantilla



Es buena idea – aunque no imprescindible – partir de una **plantilla de código** para el componente

yo polymer:element my-element
gitgub/seed-element.org

Flujo de

Desarrollo

II. Desarrollo

Polymer Starter Kit



El desarrollo y diseño queda asistido por la **estructura** de aplicación definida en el **Polymer Starter Kit**

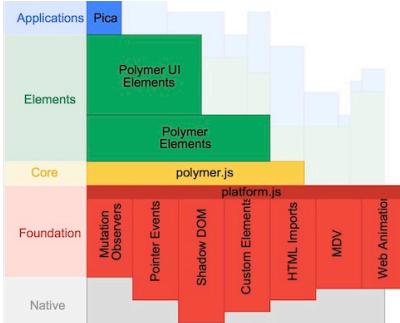
III. Despliegue

Una vez desarrollada la aplicación de componentes se **compacta** con gulp o grunt



IV. Publicar

Se empaqueta la aplicación como un **modulo bower** y se **publica** en los repositorios de componentes



La Web Orientada a Componentes

Introducción



Cómo Operan los Componentes Web

Aplicaciones de Componentes Web

I. Buscar



Existen *repositorios* de componentes Web que resuelven problemas de aparición recurrente

www.customelements.io
elements.polymer-project.org



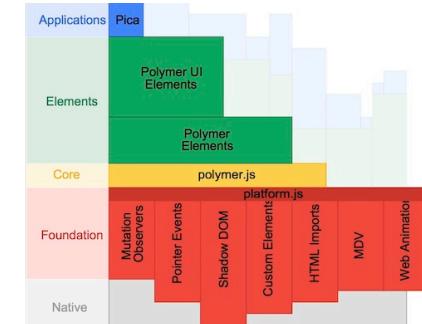
Flujo de Explotación

III. Explorar

Una vez que el componente se ha instalado, se *carga* en la página y se *instancia* como una etiqueta estándar



```
<link rel="import" href="paper-toolbar.html">  
  
<paper-toolbar class="tall">  
  <div id="progressBar" class="bottom fit"></div>  
</paper-toolbar>
```



II. Instalar

\$ bower install paper-toolbar



Con bower se *instalan* los componentes desde el repositorio con un simple comando

Javier Vélez Reyes

@javiervelezreye

Javier.velez.reyes@gmail.com

2 *Arquitecturas Orientadas a Componentes Web*

- Hacia una Web de Componentes
- Modelo de Componente Web
- Arquitecturas Orientadas a Componentes Web

La Web Orientada a Componentes

Arquitecturas Orientadas a Componentes Web

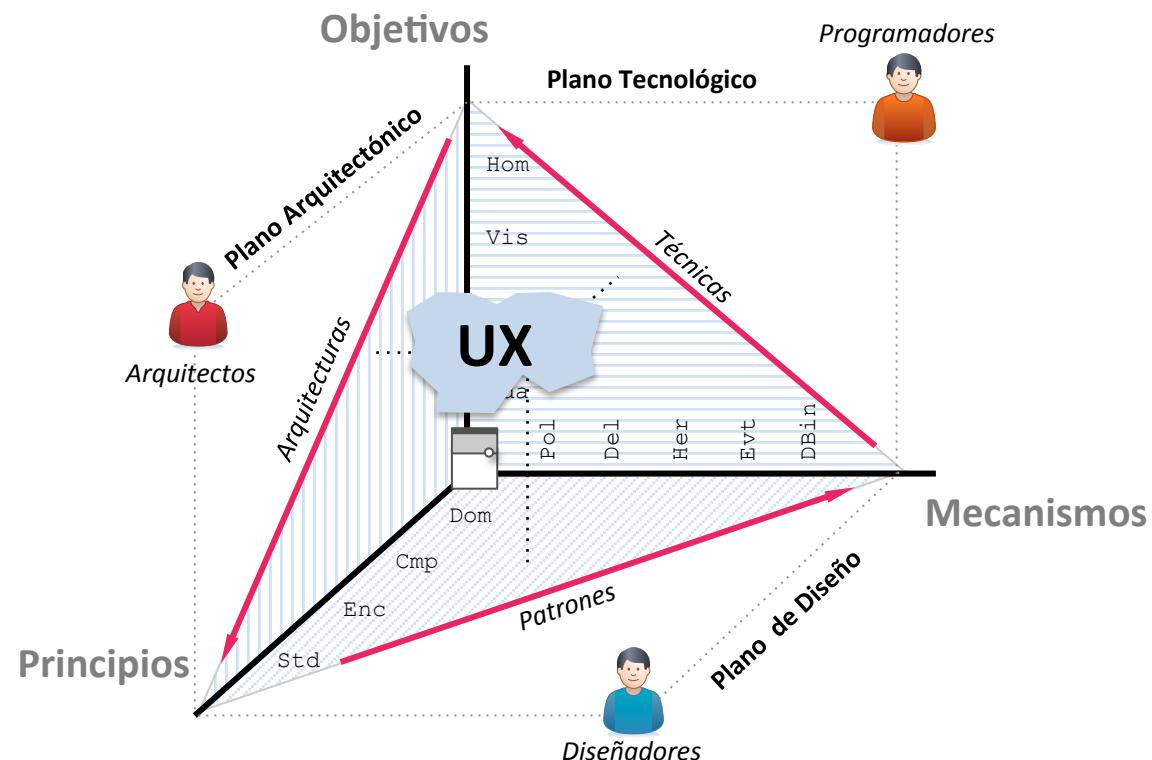
La Web Orientada a Componentes

Arquitecturas Orientadas a Componentes Web



Hacia Una Web De Componentes

Un Nuevo Paradigma Por Derecho Propio



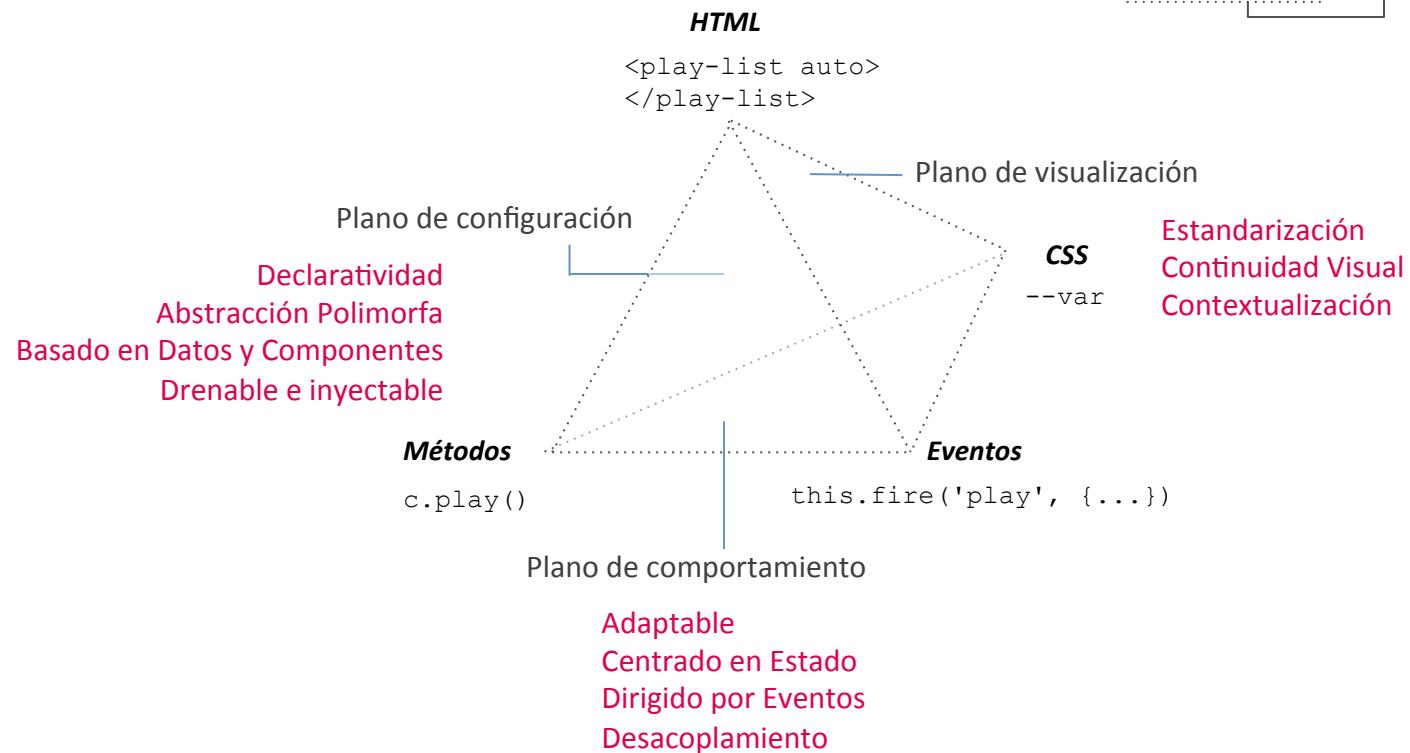
La Web Orientada a Componentes

Arquitecturas Orientadas a Componentes Web



Modelo de Componente Web

El Componente Web como Fachada



La Web Orientada a Componentes

Arquitecturas Orientadas a Componentes Web



Arquitecturas Orientadas a Componentes Web

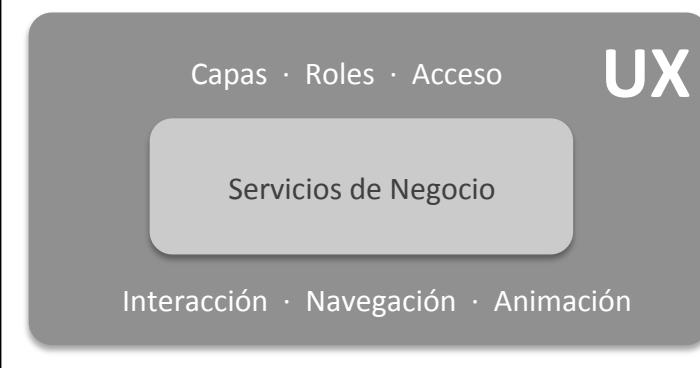
Problemas de Composición & Coordinación

Composición

Espacio

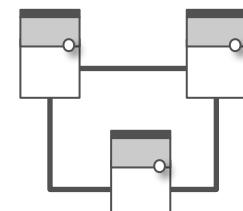


Cómo se adaptan los componentes para que encajen en el contexto arquitectónico de uso



Coordinación

Cómo se acompaña el funcionamiento de los componentes para que soporten un modelo de interacción



Tiempo

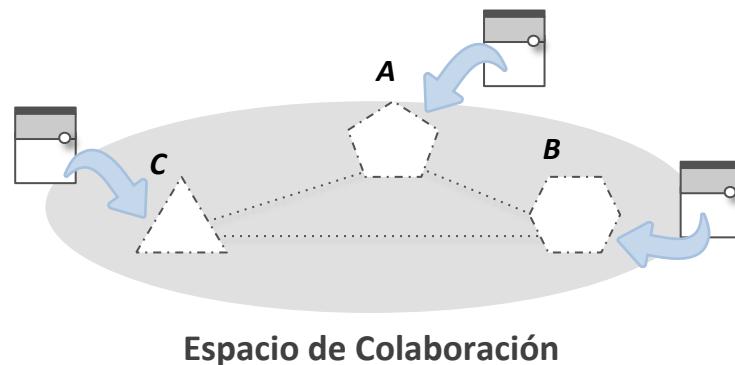
La Web Orientada a Componentes

Arquitecturas Orientadas a Componentes Web



Composición en Componentes Web

El Problema de Composición



Espacio de Colaboración

2 Aspectos Complementarios

- Comportamiento
- Presentación

Comportamiento

Delegación
Intercesión
Extensión
Adición

Solapamiento

Entrelazado

Inserción

Sustitución

Yuxtaposición

Presentación

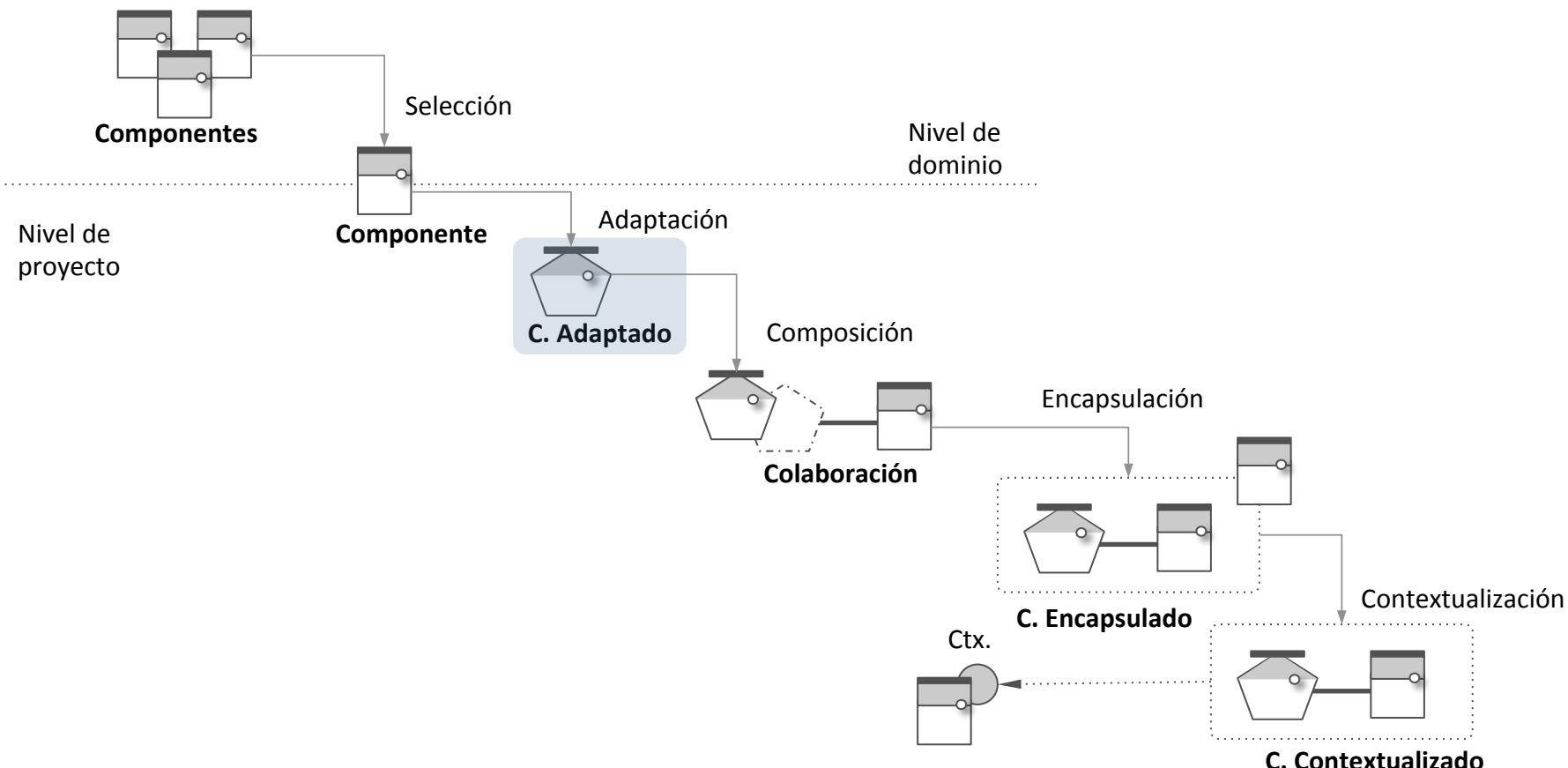
La Web Orientada a Componentes

Arquitecturas Orientadas a Componentes Web



Composición en Componentes Web

El Problema de Composición



La Web Orientada a Componentes

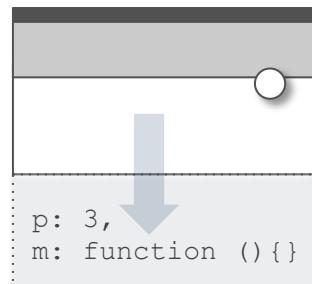
Arquitecturas Orientadas a Componentes Web



Composición en Componentes Web

El Problema de Composición. Adaptación del Comportamiento

Adición



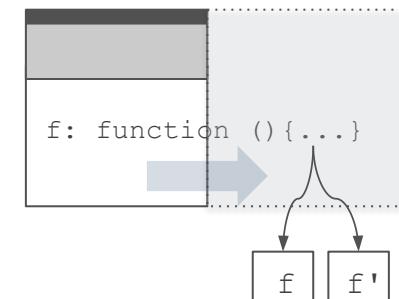
```
Polymer({  
  is: 'super-element',  
  behaviors: [SuperBehavior]  
});
```

La adición extiende un componente **horizontalmente** añadiendo nuevas responsabilidades a cada capacidad

La adición extiende un componente **verticalmente** añadiendo nuevas capacidades definidas en otro componente que opera como rasgo de comportamiento

```
var before = function (c, k, fn) {  
  var gn = c[k];  
  c[k] = function () {  
    var args = [].slice.call(arguments);  
    var out = fn.apply(this, args);  
    return gn.apply(this, args.concat(out));  
  };  
  return core;  
};
```

Intercesión



La Web Orientada a Componentes

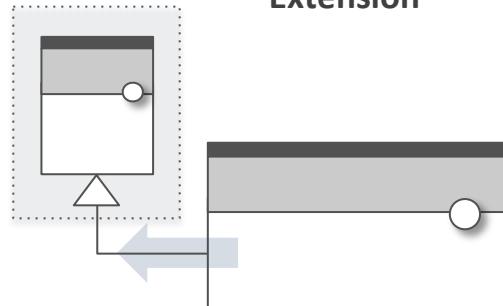
Arquitecturas Orientadas a Componentes Web



Composición en Componentes Web

El Problema de Composición. Adaptación del Comportamiento

Extensión



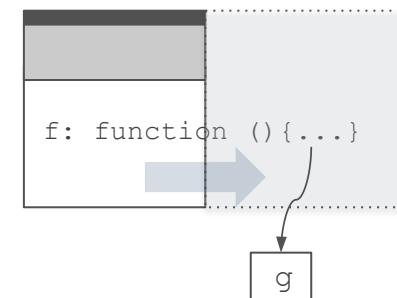
```
Polymer({  
  is: 'super-element',  
  extends: [SuperBehavior]  
});
```

La extensión extiende un componente por herencia de las capacidades en otros componente. **Solo etiquetas estándar!**

La delegación descansa la responsabilidad de una capacidad en otro componente **delegado**

```
var delegate = function (c, d, k, ctx) {  
  var context = ctx || d;  
  c[k] = function () {  
    var args = [].slice.call(arguments);  
    var out = d[k].apply(context, args);  
    return out === d ? this : out;  
  };  
  return c;  
};
```

Delegación



La Web Orientada a Componentes

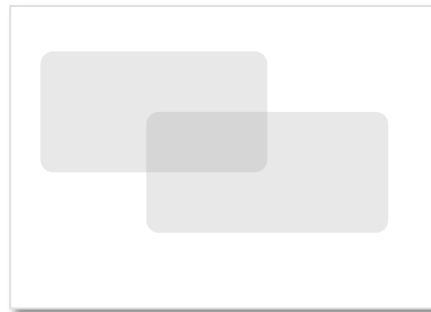
Arquitecturas Orientadas a Componentes Web



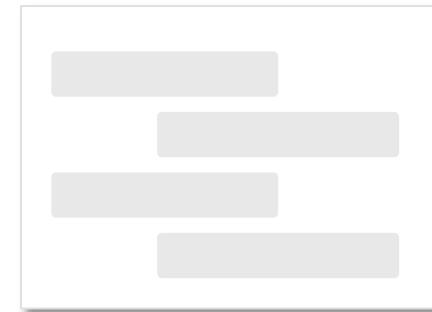
Composición en Componentes Web

El Problema de Composición. Adaptación de la Presentación

Solapamiento



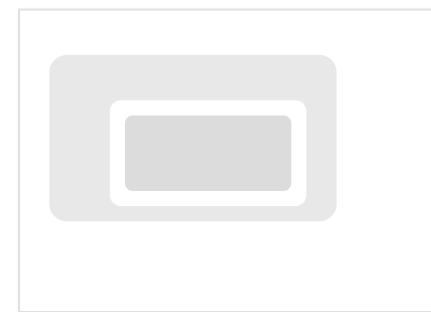
Entrelazado



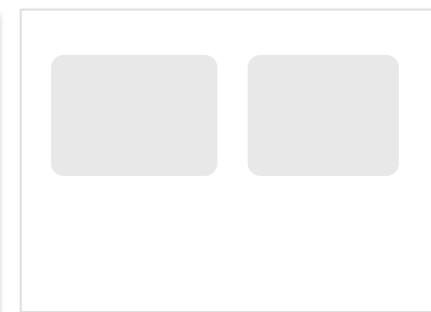
Inserción



Sustitución



Yuxtaposición



La Web Orientada a Componentes

Arquitecturas Orientadas a Componentes Web



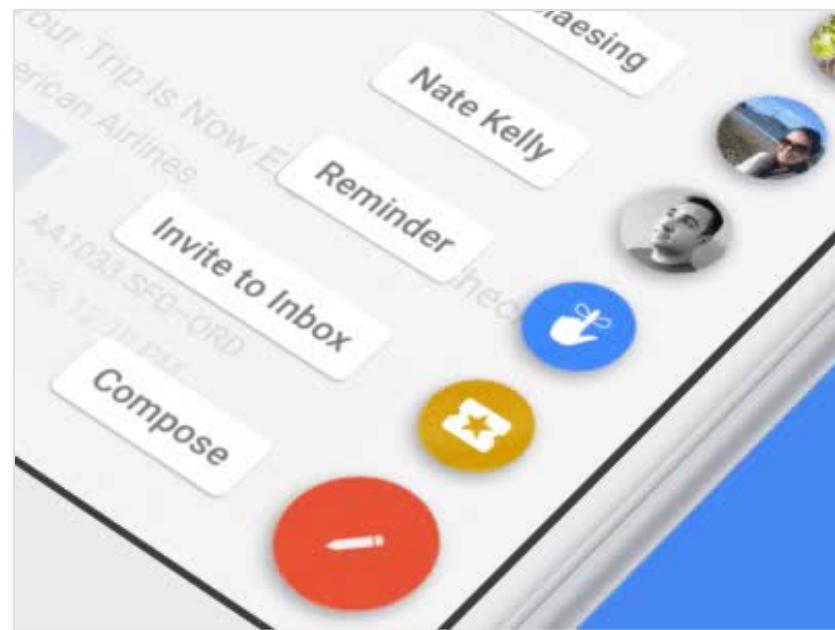
Composición en Componentes Web

Ejemplo

Visualmente existe una **adaptación solapante** en la que el botón se superpone a la vista del componente

Funcionalmente, el componente tiene que reaccionar a los **eventos de tap** disparados por el nuevo componente incrustado

```
<wc-overlap>
  <wc-listen target on="tap" do="{!! fn !!}">
    <wc-A></wc-A>
  </wc-listen>
  <wc-B ext></wc-B>
</wc-overlap>
```



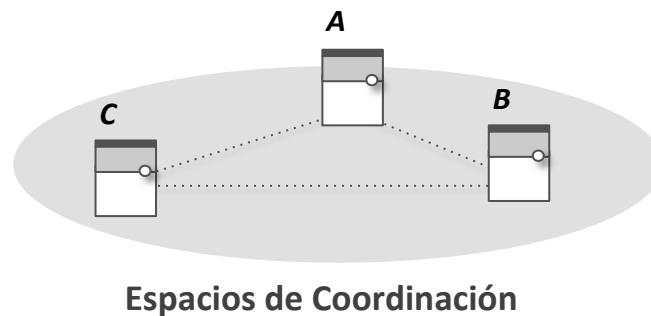
La Web Orientada a Componentes

Arquitecturas Orientadas a Componentes Web



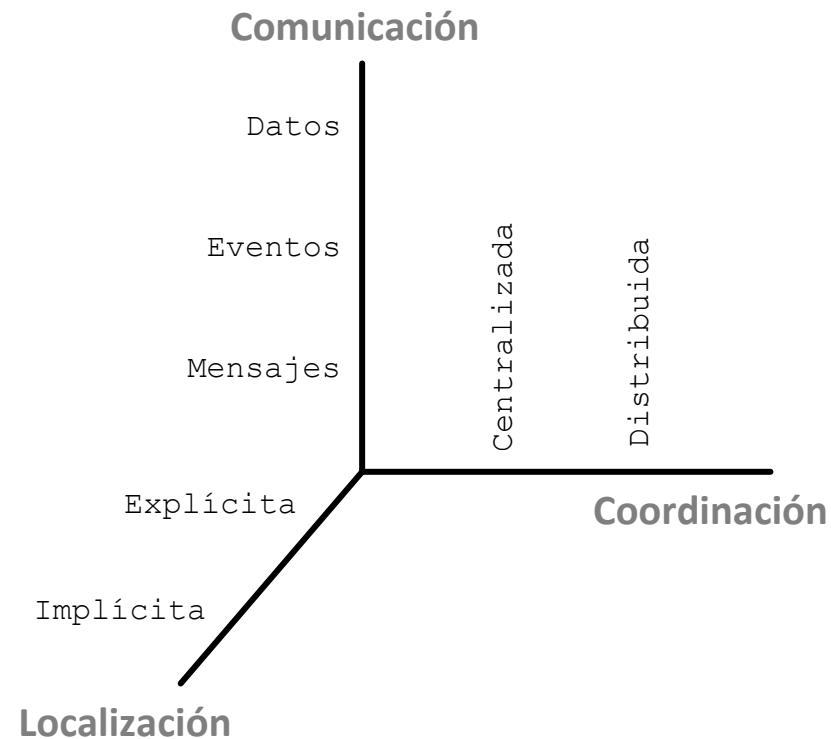
Composición en Componentes Web

El Problema de Coordinación



3 Problemas Complementarios

- Localización
- Comunicación
- Coordinación



La Web Orientada a Componentes

Arquitecturas Orientadas a Componentes Web

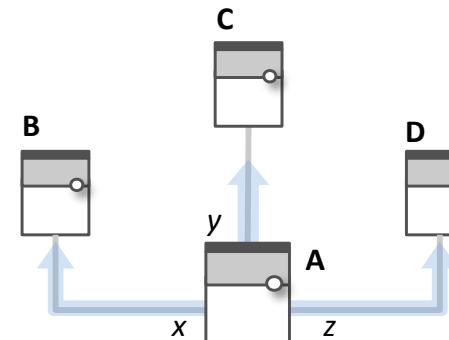


Coordinación en Componentes Web

El Problema de Coordinación. Localización



Localización Explícita



Localización Implícita

Patrones de Creación

- wc-factory
- wc-provider
- wc-context
- wc-loader

Patrones de Descubrimiento

- wc-recruiter
- wc-scatter-gather

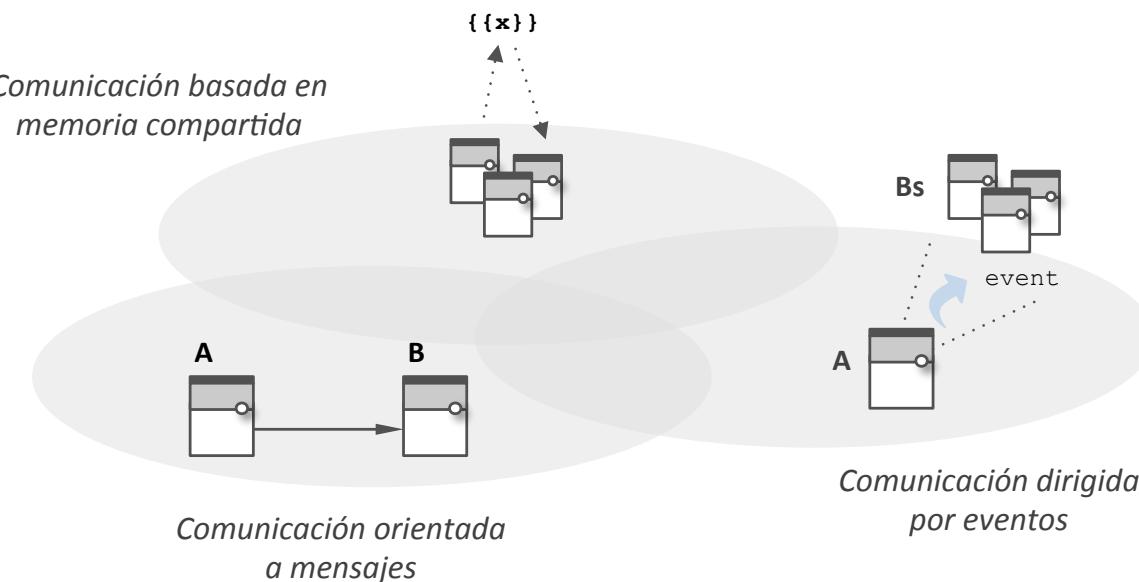
La Web Orientada a Componentes

Arquitecturas Orientadas a Componentes Web



Coordinación en Componentes Web

El Problema de Coordinación. Comunicación



Patrones de Enrutamiento

- wc-router
- wc-switch

Patrones de Correlación

- wc-bus
- wc-rebound
- wc-split
- wc-join
- wc-sort

Patrones de Distribución

- wc-ajax
- wc-rest
- wc-web-socket

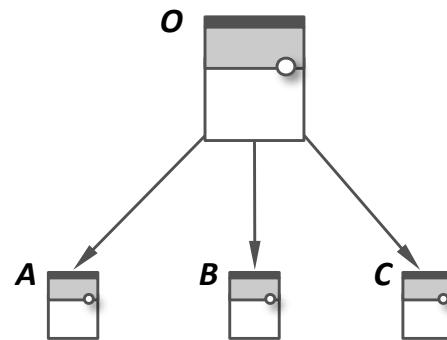
La Web Orientada a Componentes

Arquitecturas Orientadas a Componentes Web

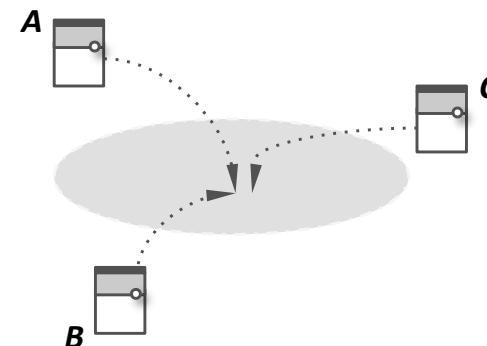


Coordinación en Componentes Web

El Problema de Coordinación. Coordinación



Coordinación Centralizada



Coordinación distribuida

Patrones de Control

- wc-orchestrator
- wc-chain
- wc-pipe
- wc-balancer
- wc-broker

Patrones de Sincronización

- wc-mutex
- wc-semaphore
- wc-channel

Patrones de Planificación

- wc-ward
- wc-worker
- wc-task
- wc-scheduller
- wc-branch
- wc-chreck
- wc-timer

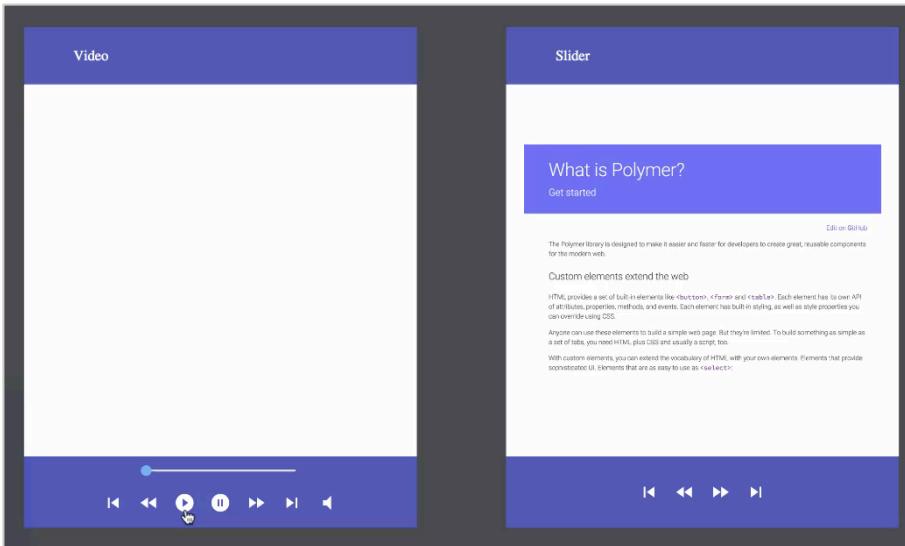
La Web Orientada a Componentes

Arquitecturas Orientadas a Componentes Web



Coordinación en Componentes Web

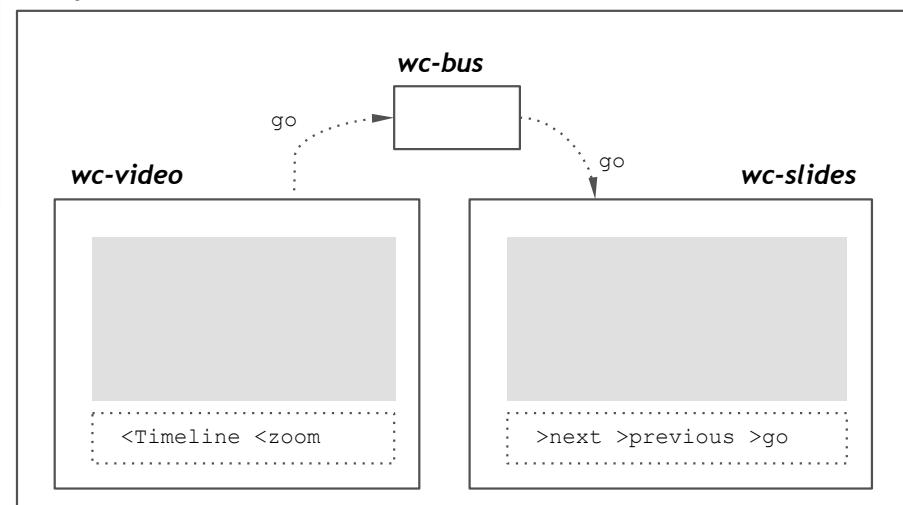
Ejemplo I. Sincronización Video & Transparencias



```
<wc-speech>
  <wc-video> ... </wc-video>
  <wc-slides> ... </wc-slides>
</wc-speech>
```

- El video envía eventos
- Las transparencias escuchan
- Los recursos se sincronizan

wc-speech



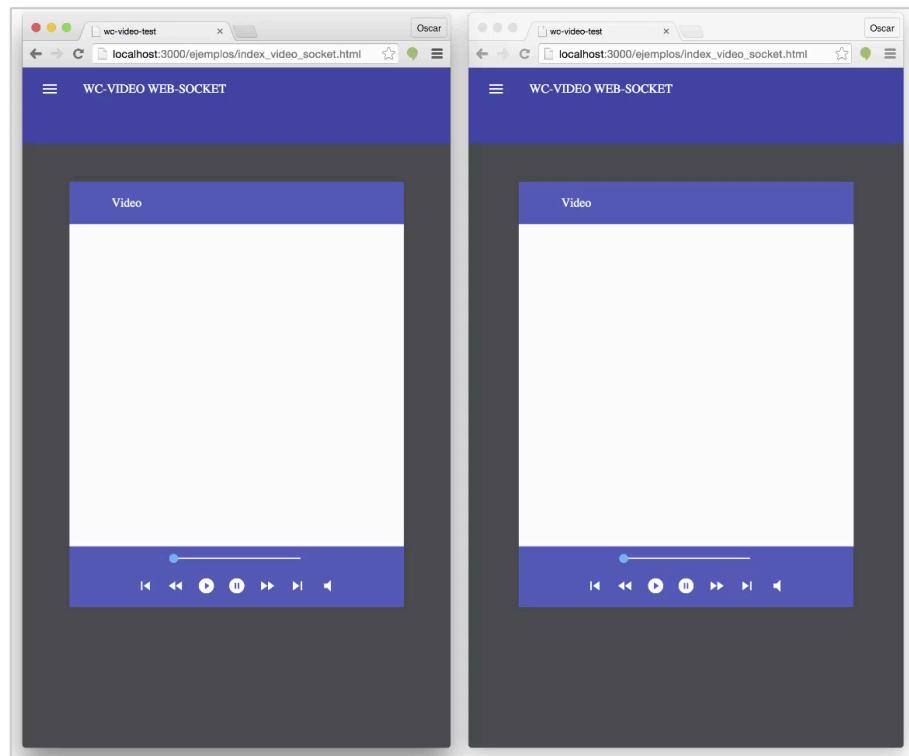
La Web Orientada a Componentes

Arquitecturas Orientadas a Componentes Web

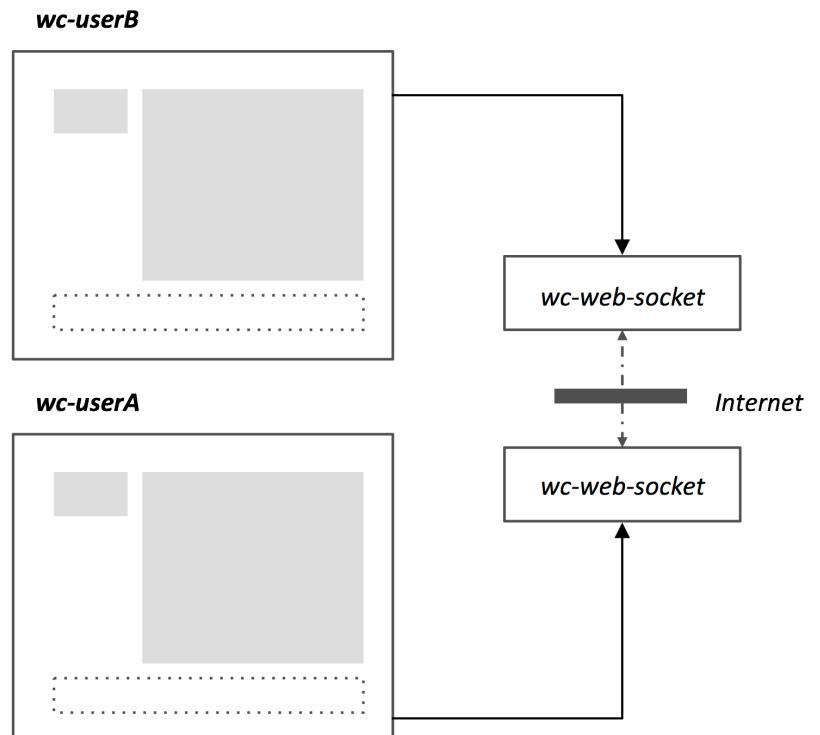


Coordinación en Componentes Web

Ejemplo II. Video Compartido



- Sincronización remota
- Chat o video
- Arquitectura de eventos



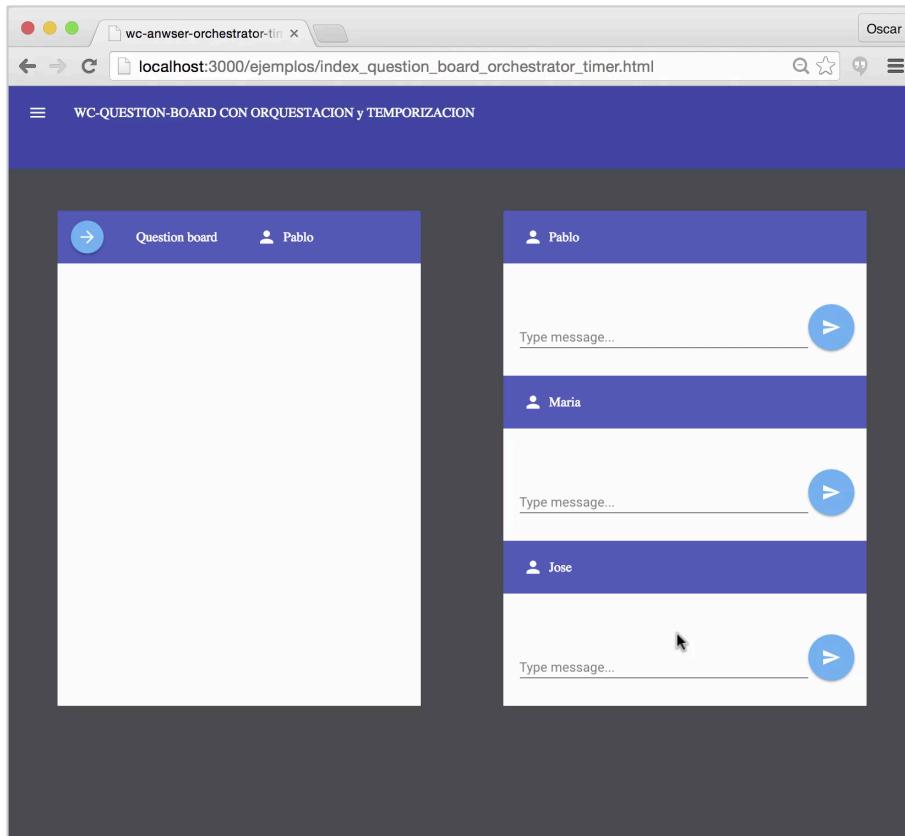
La Web Orientada a Componentes

Arquitecturas Orientadas a Componentes Web



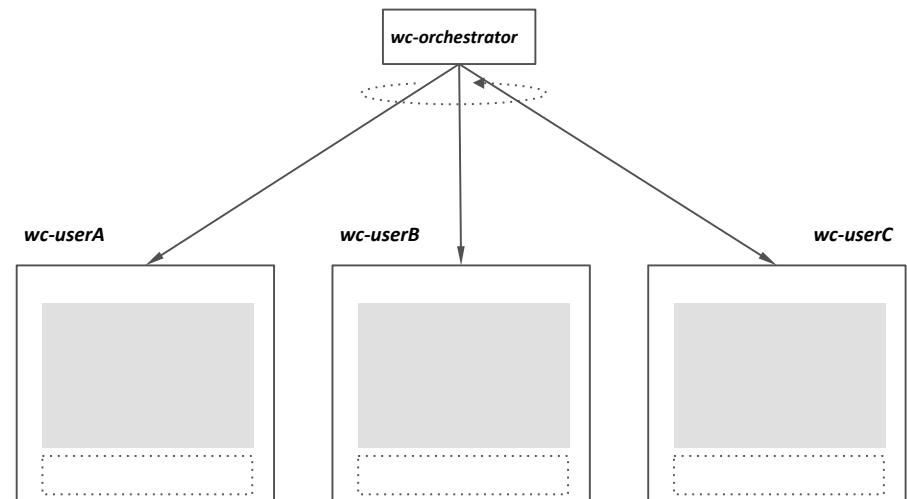
Coordinación en Componentes Web

Ejemplo III. Control de Turnos



- Control de turnos
- Intervención circular

```
<wc-orchestrator>
  <wc-agent id="A"> ... </wc-agent>
  <wc-agent id="B"> ... </wc-agent>
  <wc-agent id="C"> ... </wc-agent>
</wc-orchestrator>
```



Javier Vélez Reyes

@javiervelezreye

Javier.velez.reyes@gmail.com

3

Buenas Prácticas en el diseño de Componentes Web

- Dialectos de negocio · Composición ascendente
- Ausencia de contexto · Gateways · Diseño Desde Front
- Proyecciones · Granularidad · Declaratividad
- Composición · Enrutamiento

La Web Orientada a Componentes

Buenas Prácticas en el Diseño de Componentes Web

La Web Orientada a Componentes

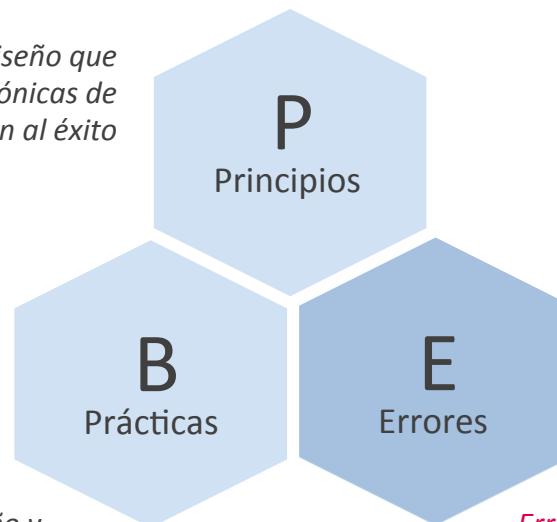
Buenas Prácticas en el Diseño de Componentes Web



Introducción

Principios, Buenas Prácticas & Errores Comunes

Principios de diseño que prescriben formas canónicas de proceder que conducen al éxito



Buenas prácticas de diseño y desarrollo que de manera no sistemática se identifican en base a la experiencia

Errores comunes que se han identificado como fuente de conflictos futuros en soluciones construidas

La Web Orientada a Componentes

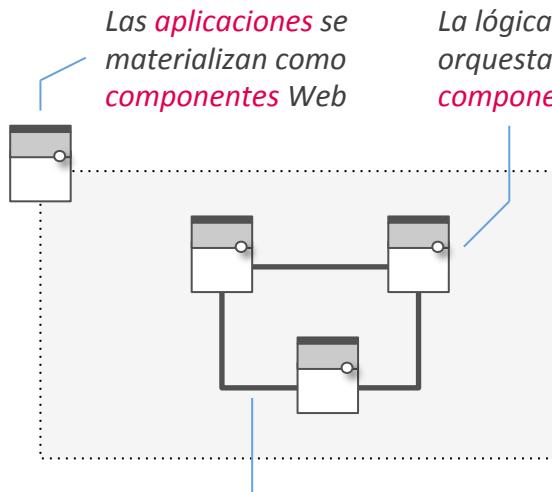
Arquitecturas Orientadas a Componentes Web



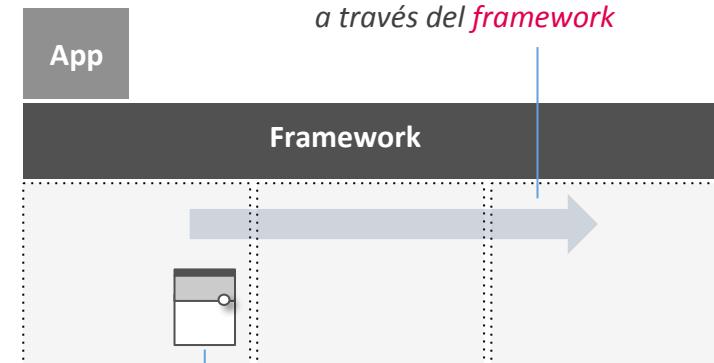
Confundir Tipos de Arquitecturas de Componentes Web

Arquitecturas Basadas & Orientadas a Componentes Web

Arquitecturas Orientadas a Componentes Web



Un modelo de interacción recurrente se *encapsula* para su reutilización



Los *componentes* resuelven problemas puntuales como elementos *autónomos*

Arquitecturas Basadas en Componentes Web

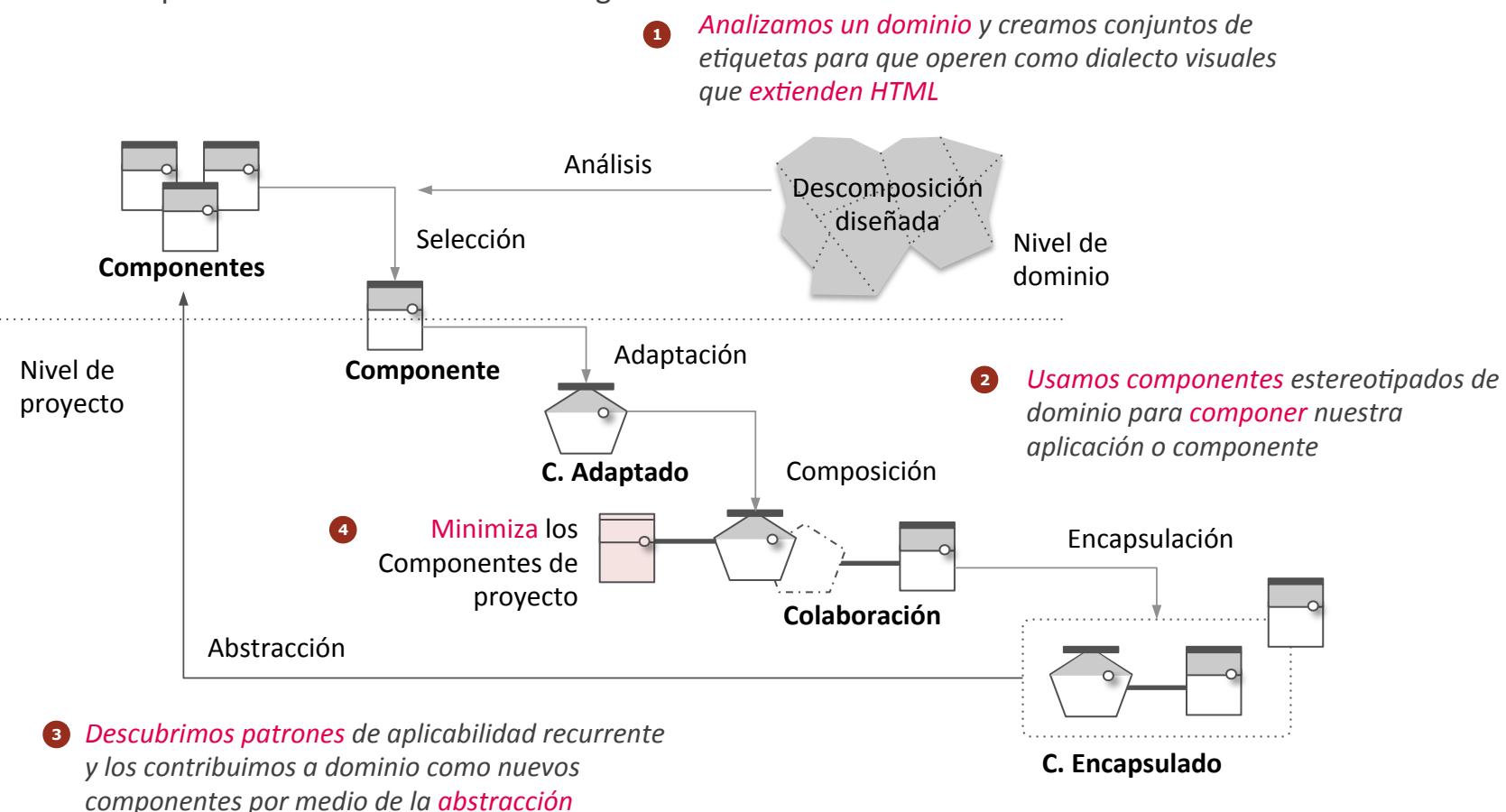
La Web Orientada a Componentes

Buenas Prácticas en el Diseño de Componentes Web



Creer Que Esto Va de Crear Aplicaciones

Componentes como Dialectos de Negocio



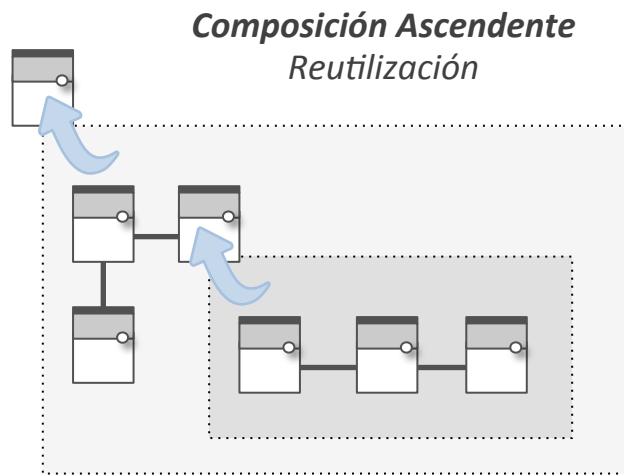
La Web Orientada a Componentes

Buenas Prácticas en el Diseño de Componentes Web



Confundir Reutilización y Modularización

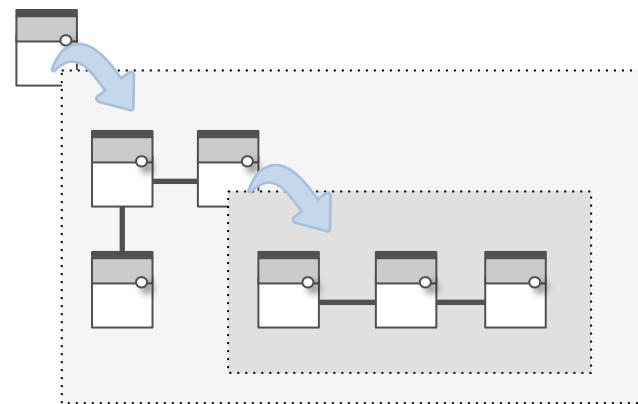
Composición Ascendente y Descomposición Descendente



Cada componente surge como una encapsulación natural motivada por una búsqueda de la **reutilización**



La estrategia de descomposición acopla al contexto los componentes particular para el que se diseñan



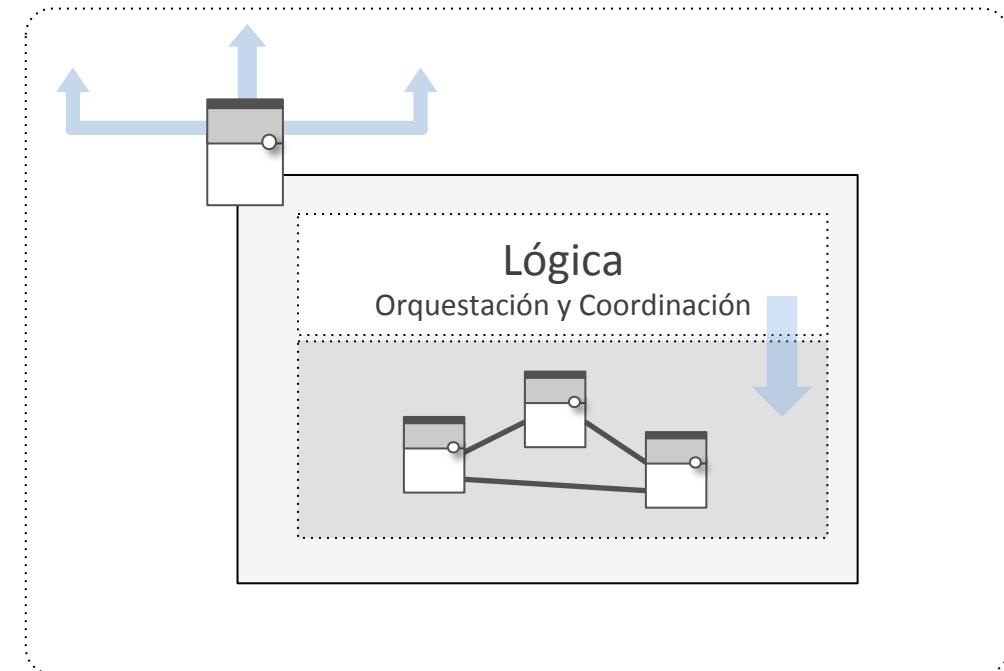
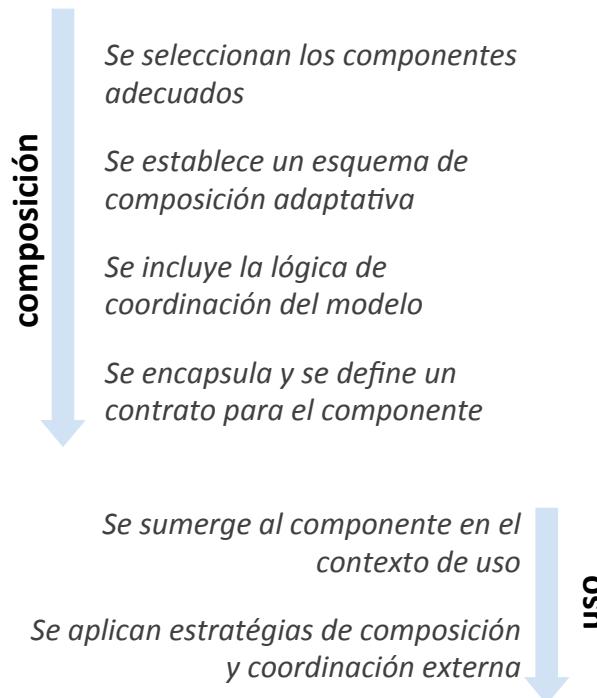
La Web Orientada a Componentes

Buenas Prácticas en el Diseño de Componentes Web



Asumir La Existencia de Contexto

Piensa Localmente y Diseña Hacia Dentro



El **contexto** de uso es **desconocido** durante la fase de diseño y coordinación



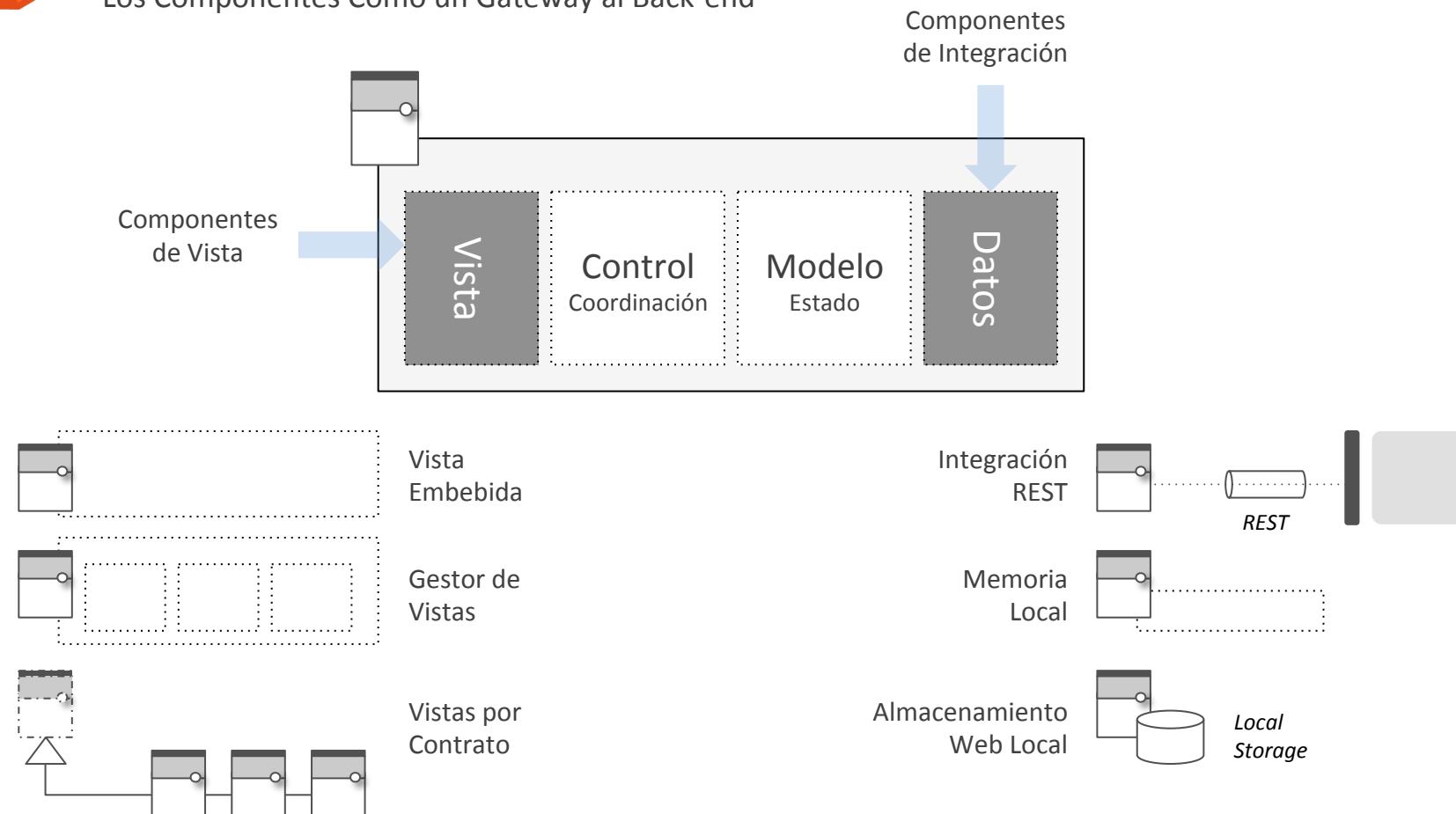
La Web Orientada a Componentes

Buenas Prácticas en el Diseño de Componentes Web



Pensar que los Componentes Son Sólo una Cuestión de Vista

Los Componentes Como un Gateway al Back-end



La Web Orientada a Componentes

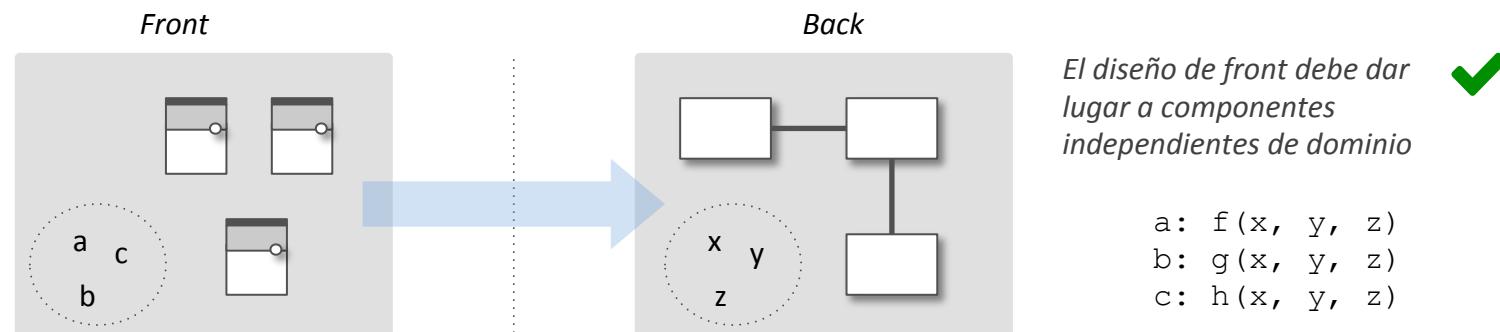
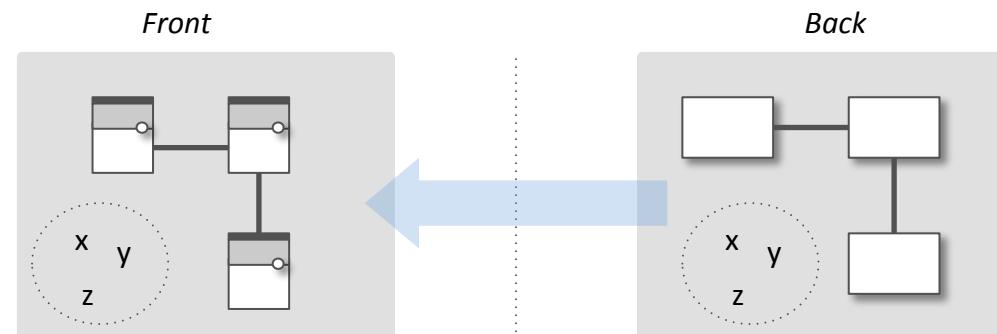
Buenas Prácticas en el Diseño de Componentes Web



Dirigir el Diseño De Componentes desde el Back End

Los Componentes Como Entidades Centradas en la Vista

✗ *El diseño de los componentes de front corresponden a proyecciones parciales de entidades de negocio*



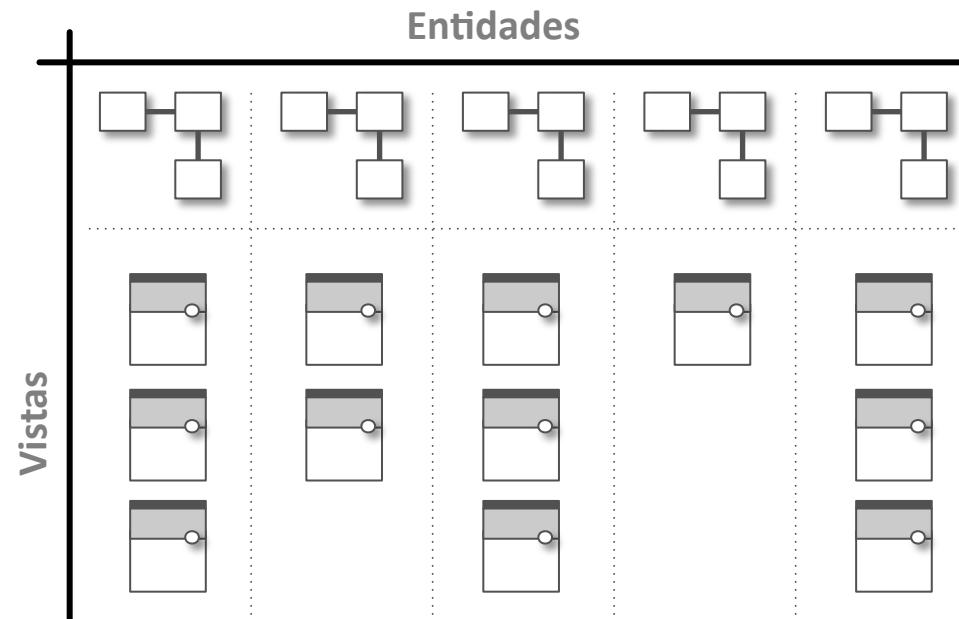
La Web Orientada a Componentes

Buenas Prácticas en el Diseño de Componentes Web



Olvidar que Esto es una Solución de Front

Los Componentes Como Proyecciones Centrados en la Interacción



Para los *componentes de negocio* debe partirse desde entidades o *agregados de entidades*

Los componentes responden a *proyecciones* de entidades relacionadas con los *modelos de interacción*

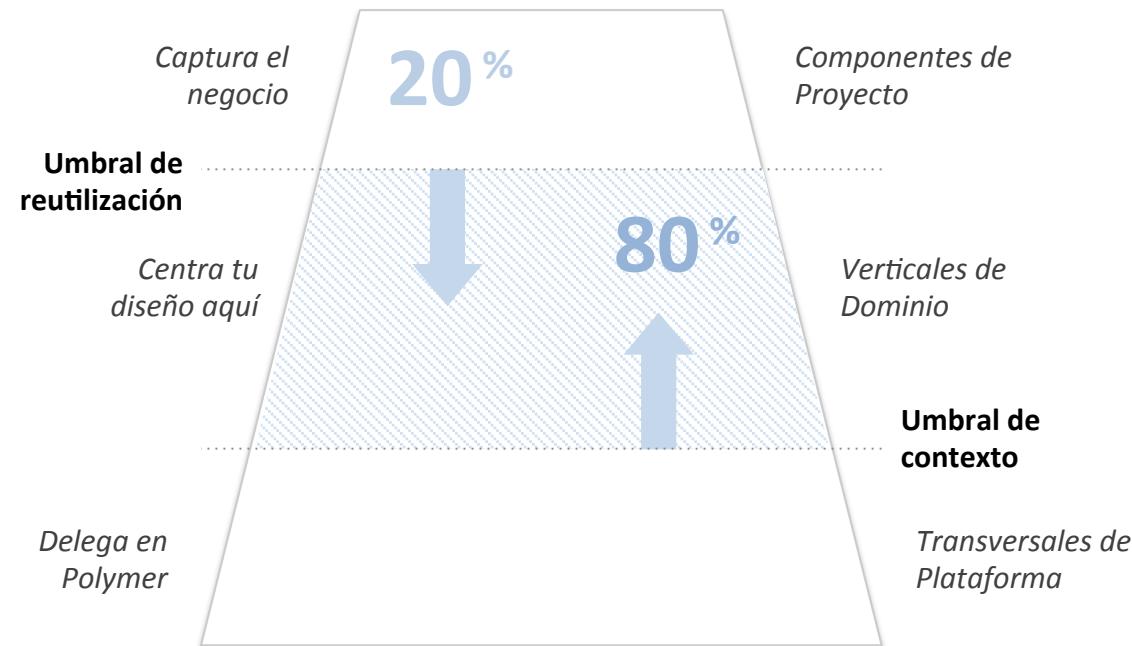
La Web Orientada a Componentes

Buenas Prácticas en el Diseño de Componentes Web



Centrar el Diseño en el Producto

Los Componentes Como Artefactos Transversales



La Web Orientada a Componentes

Buenas Prácticas en el Diseño de Componentes Web



Olvidar que Esto va de Composición Declarativa

Los Componentes Como Soluciones Centradas en HTML



```
<google-map map="{map}" .....  
    latitude="37.779"  
    longitude="-122.3892">  
</google-map>
```

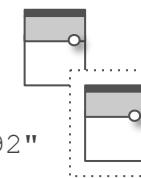
OUT

IN



```
<google-map-directions map="{map}" .....  
    startAddress="San Francisco"  
    endAddress="Mountain View">  
</google-map-directions>
```

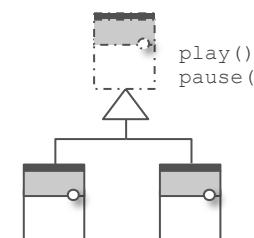
Delegación



```
<google-map fitToMarkers>  
  <google-map-marker  
    latitude="37.779"  
    longitude="-122.3892"  
    draggable="true"  
    title="Go Giants!">  
  </google-map-marker>  
</google-map>
```

Agregación

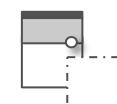
```
<wc-player>  
  <video> ... </video>  
</wc-player>
```



```
<wc-player>  
  <audio> ... </audio>  
</wc-player>
```

Polimorfismo

```
<wc-list>  
  <div> ... </div>  
  <div> ... </div>  
  <div> ... </div>  
</wc-list>
```



Genericidad

La Web Orientada a Componentes

Buenas Prácticas en el Diseño de Componentes Web



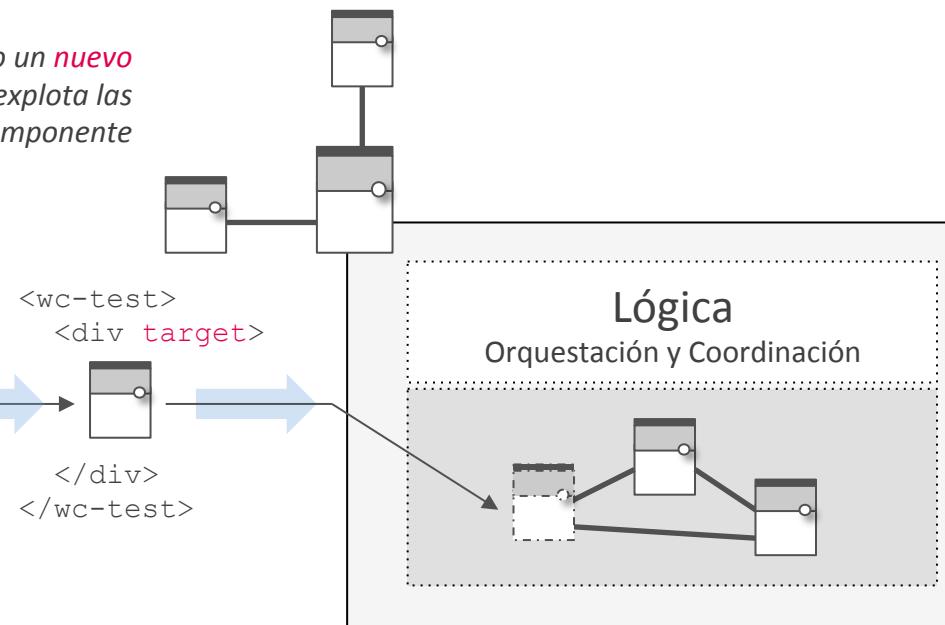
Pensar Que La Construcción Compositiva Es Cosa de Terceros

Los Componentes Como Un Esquema de Composición

En el contexto de uso un **nuevo esquema** explota las capacidades del componente

El esquema se completa en la configuración **integrando componentes** desde el contexto

Ocasionalmente puede ser necesario realizar **adaptaciones dinámicas**



Definimos un **esquema abstracto** de composición que usa componentes de terceros

La implementación se realiza por **composición interna** de otros componentes

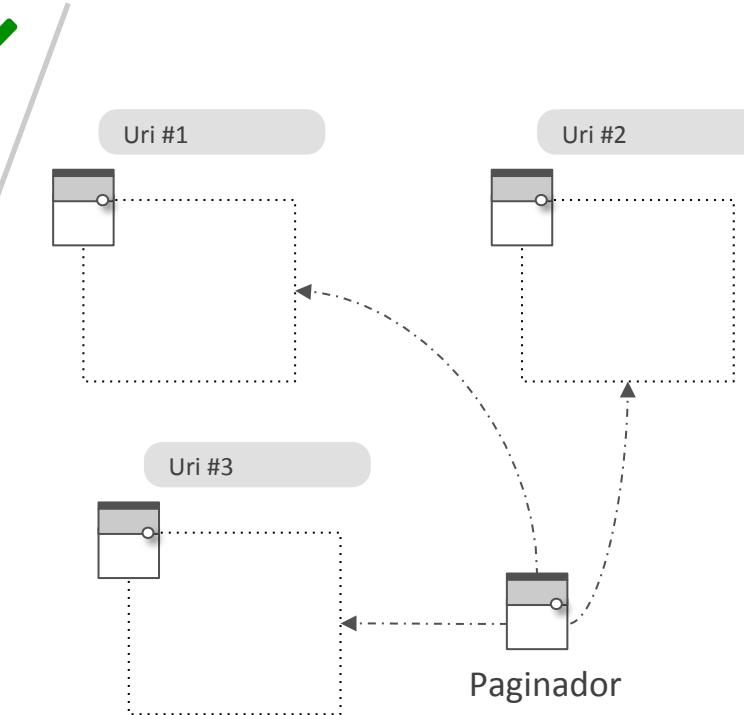
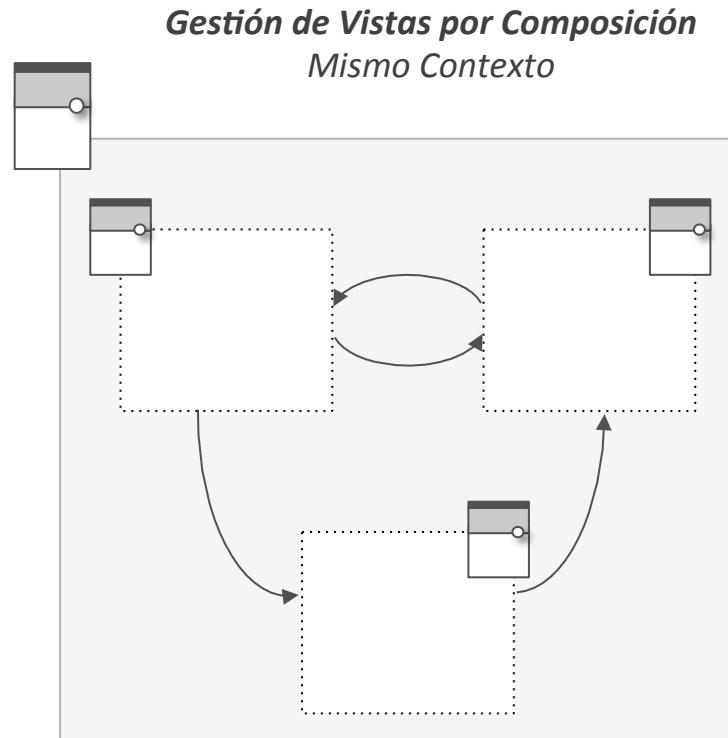
La Web Orientada a Componentes

Buenas Prácticas en el Diseño de Componentes Web



Abusar del Enrutamiento

Los Componentes Como Modelos de Interacción



Javier Vélez Reyes

@javiervelezreye

Javier.velez.reyes@gmail.com

4 *Llamada a la Acción*

- Unión de Comunidades & Sinergia de Perfiles
- Librerías de Componentes Transversales
- Hoja de Ruta

La Web Orientada a Componentes

Componentes Web en la Práctica

La Web Orientada a Componentes

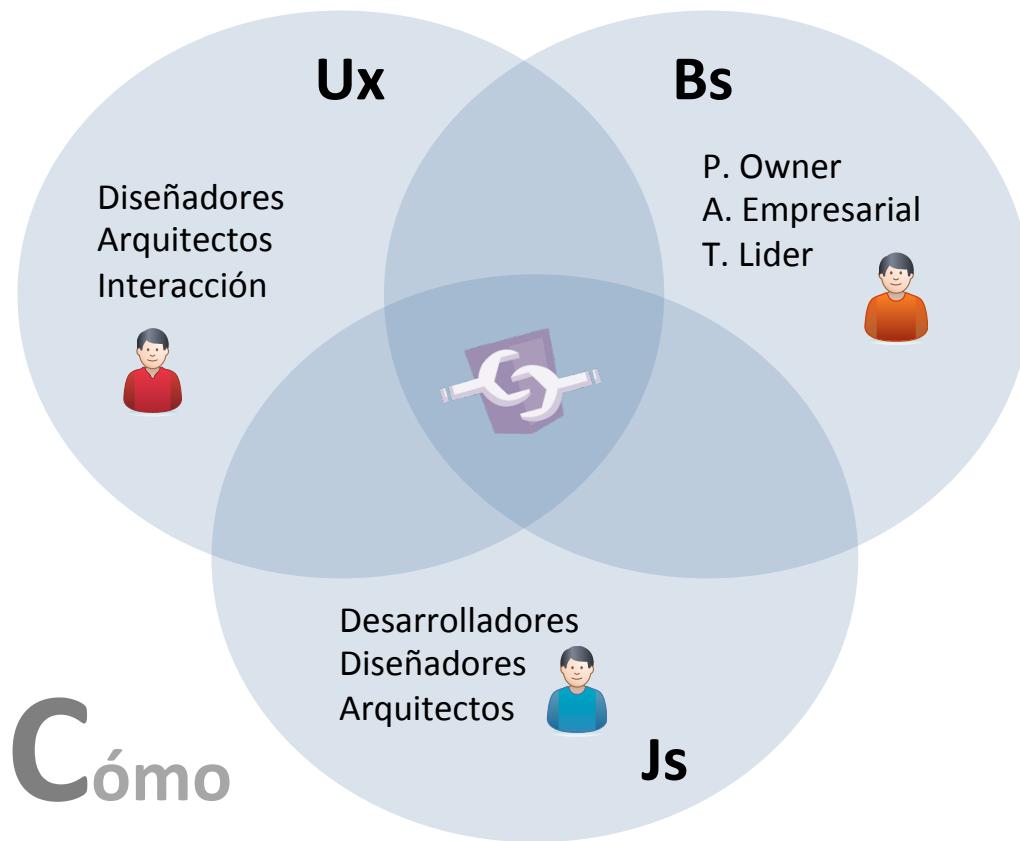
Buenas Prácticas en el Diseño de Componentes Web



Unión de Comunidades

Sinergia de Perfiles

Qué



Por qué

Cómo

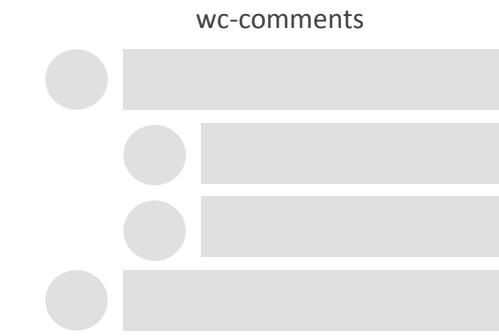
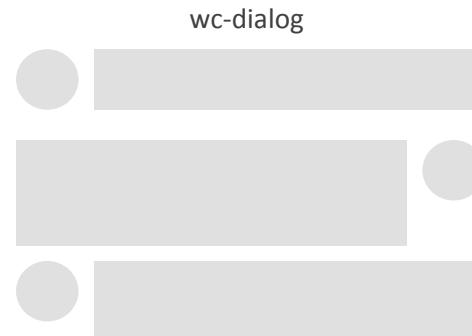
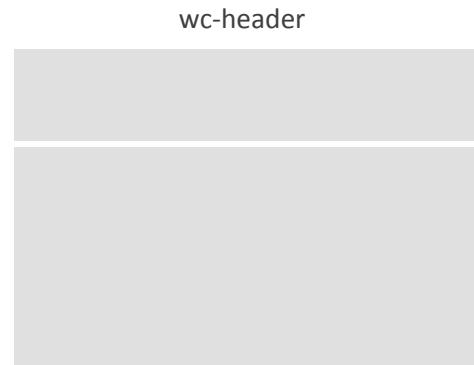
La Web Orientada a Componentes

Buenas Prácticas en el Diseño de Componentes Web



Librerías de Componentes Transversales

Modelos de Layout



La Web Orientada a Componentes

Buenas Prácticas en el Diseño de Componentes Web



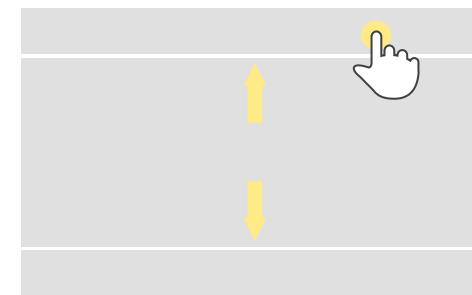
Librerías de Componentes Transversales

Modelos de Interacción

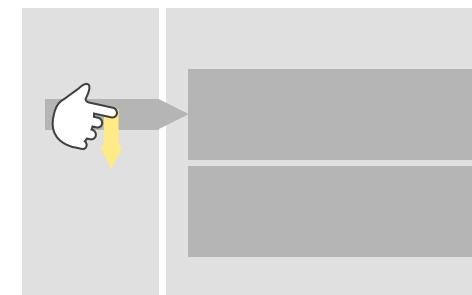
wc-pages



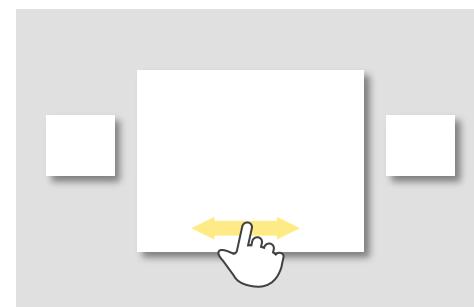
wc-pages



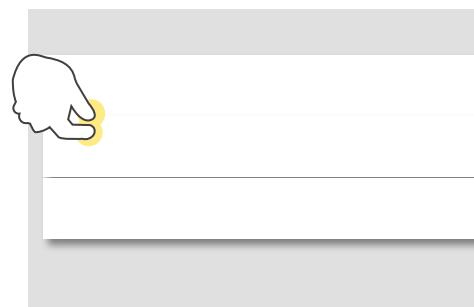
wc-index



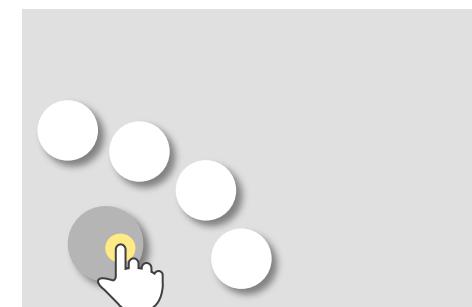
wc-focus



wc-list-in



wc-circular-fab



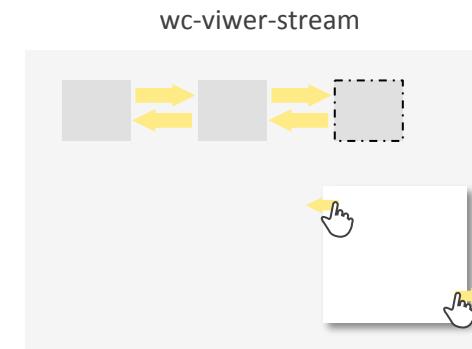
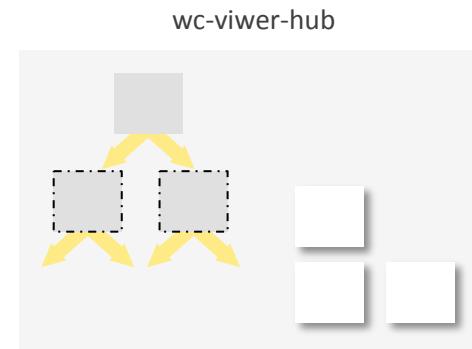
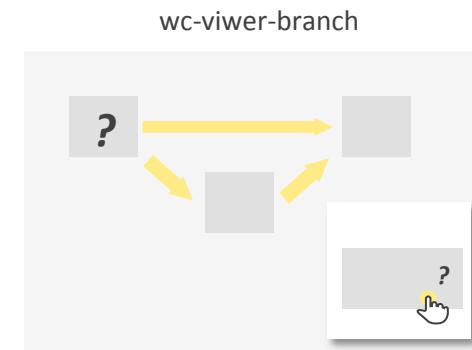
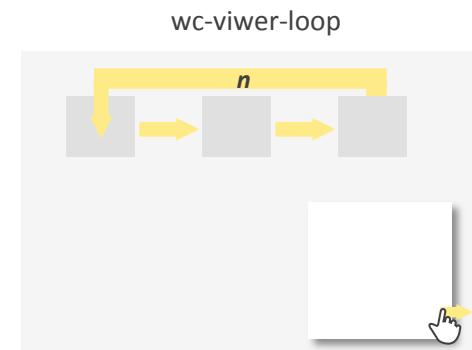
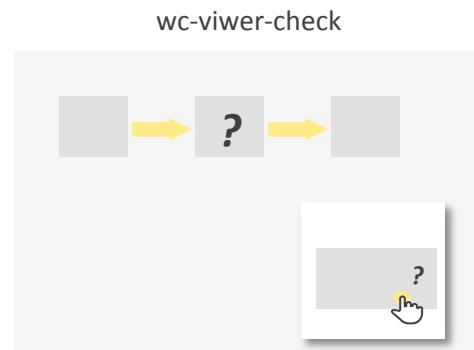
La Web Orientada a Componentes

Buenas Prácticas en el Diseño de Componentes Web



Librerías de Componentes Transversales

Modelos de Navegación



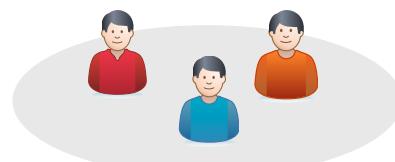
La Web Orientada a Componentes

Buenas Prácticas en el Diseño de Componentes Web



Librerías de Componentes Transversales

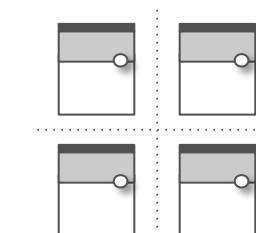
Hoja de Ruta



Grupo de Trabajo



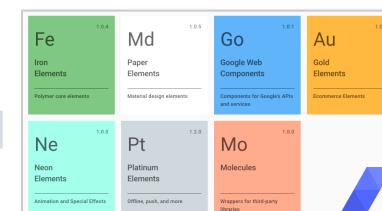
Análisis de Problemas



Diseñar Componentes



Reportar



Implementar

La Web Orientada a Componentes

Preguntas



Licenciado por la UPM desde el año 2001 y doctor en informática por la UNED desde el año 2009, Javier conjuga sus labores como profesor e investigador con la consultoría y la formación técnica para empresa. Su línea de trabajo actual se centra en la innovación y desarrollo de tecnologías para la Web. Además realiza actividades de evangelización y divulgación en diversas comunidades IT y es coordinador de varios grupos de ámbito local como NodeJS Madrid o Madrid JS. Forma parte del programa Polymer Polytechnic Speaker y es mentor del capítulo de Madrid de Node School.



javier.velez.reyes@gmail.com



@javiervelezreye



linkedin.com/in/javiervelezreyes



gplus.to/javiervelezreyes



jvelez77



javiervelezreyes



youtube.com/user/javiervelezreyes



La Web Orientada a Componentes

El nacimiento de un Paradigma

Javier Vélez Reyes

@javiervelezreye

Javier.velez.reyes@gmail.com

Noviembre 2015

{codemotion}