

Web Components

Dirección General de Sistemas de Información y Comunicaciones

Áreas de Gobierno Tecnológico de SSII y del Dato



Servicio Andaluz de Salud
**Consejería de Sanidad, Presidencia
y Emergencias**

Versión



Normativa front sección Web Components Vigente

Versión: 1.0.0

Estado: ACTIVO

Entrada en vigor desde: 1 jun 2023

Obligado cumplimiento desde: Sept-2023



Normativa front sección Web Components PRE-RELEASE

Versión: 1.0.0

Estado: RELEASE

Entrada en vigor desde: 28 nov 2022

Obligado cumplimiento desde: -

Tabla de Contenido

- Dirección General de Sistemas de Información y Comunicaciones
 - Áreas de Gobierno Tecnológico de SSII y del Dato
 - 1. Introducción
 - 2. Qué son los Web Components
 - 3. Especificaciones
 - 3.1 Custom Elements
 - 3.2 HTML Templates
 - 3.3 Shadow DOM
 - 4. Comunicación de Web Components
 - 5. Ciclo de vida de un Web Component
 - 6. Soporte de navegadores
 - 7. Pautas
 - 7.1 Pautas de categorización
 - 7.2 Pautas de nomenclatura
 - 8. Caso práctico
 - Átomos
 - Átomos no funcionales
 - Átomos funcionales
 - Moléculas
 - Organismos
 - Plantillas
 - Páginas

Cumplimiento Normativo



Las normas expuestas son de obligado cumplimiento. La STIC podrá estudiar los casos excepcionales los cuales serán gestionados a través de los responsables del proyecto correspondiente y autorizados por el Área de Gobernanza de la STIC. Asimismo cualquier aspecto no recogido en estas normas deberá regirse en primera instancia por las guías técnicas correspondientes al esquema nacional de seguridad y esquema nacional de interoperabilidad según correspondencia y en su defecto a los marcos normativos y de desarrollo software establecidos por la Junta de Andalucía, debiendo ser puesto de manifiesto ante la STIC.

La STIC se reserva el derecho a la modificación de la norma sin previo aviso, tras lo cual, notificará del cambio a los actores implicados para su adopción inmediata según la planificación de cada proyecto.

En el caso de que algún actor considere conveniente y/o necesario el incumplimiento de alguna de las normas y /o recomendaciones, deberá aportar previamente la correspondiente justificación fehaciente documentada de la solución alternativa propuesta, así como toda aquella documentación que le sea requerida por la STIC para proceder a su validación técnica.

Contacto Arquitectura: l-arquitectura.stic@juntadeandalucia.es

Histórico de cambios

Los cambios en la normativa vendrán acompañados de un registro de las modificaciones. De este modo se podrá realizar un seguimiento y consultar su evolución. Ordenándose de mas recientes a menos recientes, prestando especial cuidado a las cabeceras de la tablas dónde se indican las fechas de entrada en vigor y versión.

Versión	Pre-release	Adopción	Activa	Retiro	Alcance
v01r00	15 jun 2022				Versión inicial del documento

1. Introducción

Este apartado se centra en conocer el estándar y la filosofía de WebComponents haciendo referencia a sus especificaciones, el ciclo de vida de los componentes, navegadores que soportan esta tecnología, así como las limitaciones que se pueden encontrar al usar WebComponents. Finalmente se definen una serie de pautas a tener en cuenta a la hora de crear o categorizar WebComponents y un caso práctico donde plasmar dichas pautas.

2. Qué son los Web Components

Un **componente web** es la forma en la que se une el marcado HTML, los estilos CSS y la funcionalidad JavaScript hacia un mismo fin.

Cuando se habla de un **Web Component** se hace referencia a un estándar que abarca a ese conjunto de características nativas HTML, CSS y JavaScript que buscan una mecánica general para crear un modelo estándar, abierto y resolutivo de componentes que sirva para la mayoría de los problemas desde un enfoque general.

NOTA: Cuando se habla del estándar Web Component se tendrá de referencia la versión V1 del mismo, considerada como estándar abierto.



Composición WebComponents

Los objetivos principales del uso de WebComponents son los siguientes:

- Aislamiento: Contar con elementos aislados del exterior aporta ventajas como la protección o seguridad respecto a comportamientos no deseados propiciados por elementos de un contexto externo al elemento. Esto permite que nuevas piezas puedan usarse en el contexto de una web sin que repercuta en otras ya existentes. También facilitará las tareas de desarrollo ya que bastará con centrarse en una funcionalidad concreta y dejar de lado agentes externos.
 - Reutilización: Al trabajar con elementos aislados y funcionalidad acotada, la capacidad de reutilización aumenta, lo que implica unos desarrollos más ágiles y unos mantenimientos más ligeros.
 - Interoperabilidad: Los WebComponents hacen uso de las tecnologías abiertas y estándares web generales. Esto facilita notablemente el intercambio de información entre sistemas diferentes, es decir, no existe una dependencia a nivel tecnológico que nos limite el poder hacer uso de un determinado componente en una determinada aplicación desarrollada con un determinado framework o librería.
-

3. Especificaciones

Haciendo referencia a las características web relacionadas con HTML, CSS y JavaScript mencionadas previamente, los WebComponents están basados una serie de especificaciones:

3.1 Custom Elements

Cuando se habla de custom elements o elementos personalizados se hará de una característica de los WebComponents que permite crear etiquetas HTML personalizadas a las que se atribuyen un determinado marcado HTML, estilo y funcionalidad.

Los elementos pueden heredar de otro elemento HTML nativo o desde otro elemento personalizado. Por ejemplo, si interesa crear un componente basado en una etiqueta `<button>`, es posible extender del elemento nativo de HTML y así ya contar con toda su funcionalidad teniendo la posibilidad de eliminarla, modificarla o extenderla.

3.2 HTML Templates

Cuando se habla de templates o plantillas se hace referencia a elementos que contienen tanto HTML como CSS y que de forma inicial no son visibles en la web. Estas plantillas son reconocidas por el navegador como `<template>` y `<slot>` y, a partir de JavaScript, es posible acceder al código interno y manipularlo, clonarlo en numerosas partes del sitio web y finalmente hacerlas visibles cuando se desee.

3.3 Shadow DOM

Antes de detallar lo que es Shadow DOM, es necesario conocer el significado de DOM. El DOM (**D**ocument **O**bject **M**odel) es la composición de elementos HTML que dan lugar a una página web. Esta estructura se representa en forma de árbol de tal manera que es posible conocer elementos padres o hijos de cada elemento.

Shadow DOM o DOM oculto, es un sistema que permite tener oculto o en la sombra un fragmento del DOM, es decir, se encarga de encapsular parte del DOM para que no interfiera en otras partes de la página. El principal caso de uso está relacionado con el aislamiento de los estilos para evitar tanto alteraciones del Shadow en el DOM principal como alteraciones del DOM principal en el Shadow.

A la hora de crear un Shadow DOM, existen algunas características:

- Encapsulación: Shadow DOM presenta dos modos de encapsulación
 - Abierto: Es posible acceder (mediante JavaScript) al Shadow DOM desde el contexto principal de la página.
 - Cerrado: No es posible acceder al Shadow DOM.
- Focalización: Shadow DOM tiene la posibilidad de gestionar el foco (:focus)
 - Delegar foco: El foco se delega al Shadow DOM.
 - No delegar foco: El foco no se delega al Shadow DOM
- Contenerización: Shadow DOM solo se puede atribuir a los siguientes elementos
 - Custom Elements
 - Cabeceras (h1, h2, h3, h4, h5 y h6)
 - Etiquetas semánticas (header, footer, article, aside, nav, section y nav)
 - Contenedores (div, span, blockquote, p)
 - Body

Puede encontrarse más información al respecto en los siguientes enlaces:

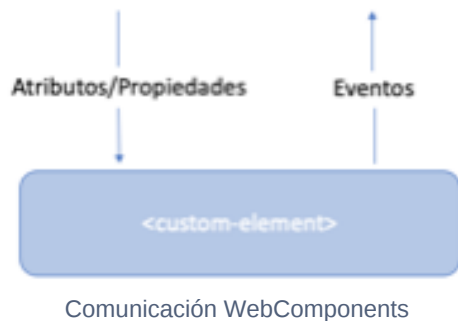
- Documentación oficial de lit ([Shadow DOM](#))
- [Shadow DOM v1: Self-Contained Web Components](#)
- [Using shadow DOM](#)

4. Comunicación de Web Components

Los WebComponents generalmente necesitan contar con información de entrada y también poder emitir información fuera de su contexto. En este caso existen atributos y propiedades para la entrada de información y eventos para su salida.

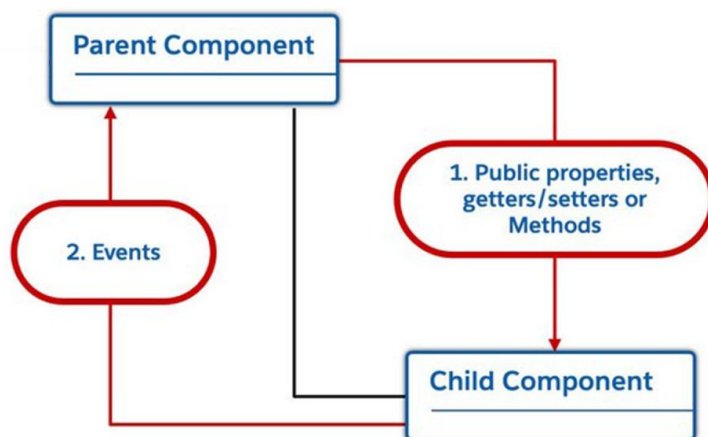
- Atributos: Se utilizan para pasar información desde el exterior al interior del componente. La información puede ser de tipo texto (string) o carecer de valor e indicar si se trata de un valor verdadero o falso (boolean).
- Propiedades: Pueden ser miembros públicos o privados del componente. En cuanto a la información que pueden recibir del exterior, es posible almacenar datos más complejos respecto a los atributos.

- **Eventos** : A partir de eventos es posible enviar información al exterior del componente. Puede ser tanto a partir de eventos nativos como de eventos personalizados.



De manera más visual, la siguiente imagen indica cómo debe hacerse la comunicación entre componentes desarrollados con web components (lit), de modo que:

- 1) El componente padre modifica los atributos del componente hijo para comunicarse con el.
- 2) El componente hijo usa (genera) eventos para comunicarse con el componente padre.



Puede encontrarse más información al respecto en la siguiente página de la documentación oficial de lit ([Events](#)).

5. Ciclo de vida de un Web Component

A la hora de cargar una página que hace uso de Web Components hay que tener en cuenta que existen varias fases o estados por los que pasa el componente, lo que se conoce como el ciclo de vida del mismo. Cada estado tiene un comportamiento nativo sin embargo es posible personalizarlo.

A continuación, se muestran las fases del ciclo de vida de un Web Component teniendo en cuenta el estándar (existen algunas librerías que incluyen estados adicionales que ofrecen un mayor grado de gestión del componente):

- Creación: El componente se crea y se crea una instancia del mismo. El que esté creado no quiere decir que sea visible.
- Conexión: El componente creado en memoria se incrusta en el DOM y pasa a formar parte de la página.
- Desconexión: El componente se desacopla del DOM y deja de formar parte de la página.
- Cambio de propiedad: Los componentes pueden tener atributos asociados. Esta fase tiene lugar cada vez que alguno de ellos sufre una variación.

Puede encontrarse más información al respecto en la siguiente página de la documentación oficial de lit ([Lifecycle](#)).

6. Soporte de navegadores

Debido a que los navegadores sufren cambios y actualizaciones en sus funcionalidades, no sería correcto indicar cuáles de ellos presentan impedimentos ante cualquiera de las especificaciones de WebComponents. Por esta razón, es necesario determinar esa compatibilidad mediante una herramienta online:

- **Custom Elements**: <https://caniuse.com/?search=custom%20elements>
- **Shadow DOM**: <https://caniuse.com/?search=shadow%20dom>
- **HTML Templates**: <https://caniuse.com/?search=html%20templates>

La herramienta muestra un listado de navegadores a lo largo de las columnas, si las especificaciones previamente indicadas están implementadas en la versión concreta del mismo, o cuando empezarán a funcionar:

IE	Edge	Firefox	Chrome	Safari	Opera	Safari on iOS	Opera Mini	Chrome for Android	UC Browser for Android	Samsung Internet
			91			12.5				
	92	91	92	13.1		14.4				
	93	92	93	14.1	78	14.8				
11	94	93	94	15	80	15	all	94	12.12	15.0
		94	95	TP						
		95	96							
			97							

Hay que tener en cuenta que si se tiene un alto porcentaje de usuarios que utilizan una versión de un navegador que no soporta WebComponents y sus especificaciones, haya que buscar alternativas o abstenerse a utilizarla hasta que ese porcentaje se reduzca. Afortunadamente la mayor parte de los navegadores relevantes soportan estas especificaciones y además están dentro de la estrategia silenciosa de actualización denominada [Evergreen Browsers](#).

Un navegador como por ejemplo Internet Explorer 11 que no presentan compatibilidad con este tipo de estándares, sin embargo, es posible darle soporte haciendo uso de [polyfill](#) (implementa características más comunes en navegadores antiguos u obsoletos). A pesar de ello, **es recomendado no tener en cuenta la compatibilidad con IE11 en cuanto a la accesibilidad desde navegadores ya que cuanta con un soporte limitado y próximo a desaparecer para principales sistemas operativos** :

- <https://docs.microsoft.com/es-es/lifecycle/announcements/internet-explorer-11-end-of-support>

📌 Importante

La aplicación de escritorio de Internet Explorer (IE) 11 dejará de ser compatible con ciertos sistemas operativos a partir del 15 de junio de 2022. Se recomienda a los clientes que se pasen a Microsoft Edge con el modo IE. El modo IE permite la compatibilidad con versiones anteriores y será compatible al menos hasta 2029. Además, Microsoft notificará la retirada del modo IE con un año de antelación. Haga clic [aquí](#) para obtener una lista de los sistemas operativos compatibles con Internet Explorer, incluidas las versiones afectadas por este anuncio.

Fin del soporte de IE11

7. Pautas

En esta sección se mostrarán las pautas asociadas a Web Components en función de su naturaleza, en este caso en base a su categorización y nomenclatura:

7.1 Pautas de categorización

Partir de un sistema de diseño proporciona una gran ventaja a la hora de desarrollar componentes, mejorar el ciclo de vida del desarrollo de productos.

Cuando los desarrolladores toman como referencia el diseño a la hora de definir componentes de la interfaz, se agiliza la identificación de los mismos, su categorización y el desarrollo de los mismos ya que es posible intuir el comportamiento asociado a ellos a partir de los diseños. El resultado son repositorios de componentes limpios, con división clara en base a su funcionalidad y negocio, reutilización, composición y una mejor comunicación con los diseñadores.

Al igual que en la etapa de diseño, en esta etapa de desarrollo de Web Components también se hace uso del modelo atómico, pero en este caso toma más relevancia el ámbito de cada fase del modelo atómico:

- A la hora de definir átomos se hace más hincapié en la separación de átomos funcionales de los no funcionales, esto servirá de más ayuda para el desarrollador. Por otro lado, a partir de átomos con funcionalidad de UI se podrá considerar a los elementos como Web Components.
- En la fase de diseño se hace referencia a la lógica de negocio a nivel de plantilla y página, sin embargo en este caso el negocio entra en juego a nivel de organismos.

Pauta de Categorización (PC)	Ámbito	Criterios de verificación	Criterios de aceptación
PC1 – Definición de átomos	PC1.1 - No funcional	PC1.1.1 - Carencia de funcionalidad	Carentes de funcionalidad, únicamente aportación visual
		PC1.1.2 - Indivisible	Indivisibles en cualquier otro elemento

	PC1.2 - Funcional UI	PC1.1.3 - Solo añaden características	Utilizados para enriquecer otros elementos en forma de propiedades (colores, dimensiones, fuentes, tipografía...)
		PC1.1.4 - Carencia de eventos	No envía ni recibe eventos
		PC1.2.1 - Componentes abstractos	Presentan funcionalidad UI pero no presentan utilidad o coherencia por si solos. Principalmente elementos HTML básicos como <input>, <h1>, <label>, <p>...
		PC1.2.2 - Pueden presentar estado UI	Pueden presentar estados a nivel de UI como deshabilitado, seleccionado, enfocado...
		PC1.2.3 - Indivisible	Indivisibles en cualquier otro elemento funcional
		PC1.2.4 - Capaz de emitir eventos	Capacidad de emitir eventos no de recibirlos
PC2 – Definición de moléculas	PC2.1 - Funcional UI	PC2.1.1 - Puede estar compuesta por átomos y otra molécula más simple	El componente puede estar formado por alguna de las siguientes alternativas: - combinación de más de un componente átomo - combinación de uno o más componentes átomos y un componente molécula más simple. Un a molécula simple puede combinarse con otros átomos para dar lugar a otra molécula más compleja con propósito único Todas las alternativas de combinación tiene que tener coherencia funcional de UI.
		PC2.1.2 - Componente funcional UI con propósito simple	Presentan funcionalidad UI con un único propósito
		PC2.1.3 - Pueden presentar estado UI	Pueden presentar estados a nivel de UI como deshabilitado, seleccionado, enfocado...
		PC2.1.4 - Capaz de emitir eventos	Capacidad de emitir eventos y suscribirse a los de los elementos que lo forman.
PC3 – Definición de organismos	PC3.1 - Funcional de negocio	PC3.1.1 - Composición de átomos, moléculas u otros organismos	El componente puede estar formado por alguna de las siguientes combinaciones: - al menos dos moléculas con o sin átomos - composición de uno o varios organismos y uno o varios(as) átomos/moléculas. Todas las alternativas de combinación tienen que tener coherencia funcional de negocio.
		PC3.1.2 - Componente funcional UI y de negocio	Componente funcional y de negocio: - Lógica de negocio intrínseca de la aplicación - Se conoce el origen del estado
		PC3.1.3 - Capaz de emitir eventos	Capacidad de emitir eventos y suscribirse a los eventos de los elementos que lo forman.
PC4 – Definición de plantillas	PC4.1 - Funcional de negocio	PC4.1.1 - Composición de componentes	Distribución de los componentes átomos, moléculas y organismos, así como la aplicación del estilado.
PC5 – Definición de páginas	PC5.1 - Funcional de negocio	PC5.1.1 - Gestión de eventos y comunicación	Es capaz de orquestar y gestionar los eventos de todos los elementos que dan lugar a la página (átomos, moléculas, organismos y plantillas)
		PC5.1.2 - Gestión de navegación	Determina la navegación interna y externa de la página
		PC5.1.3 - Gestión de estado	Determina el estado global de la página

7.2 Pautas de nomenclatura

A la hora de definir el custom element de un Web Component será necesario tener en cuenta una serie de pautas:

Pauta de Nomenclatura (PN)	Criterio de verificación
PN-1 Nombre compuesto	PN-1.1 Al menos formado por dos palabras

PN-2 Separación con guion	PN-2.1 La separación de las palabras se realiza mediante en carácter guion “-”
PN-3 Minúsculas	PN-3.1 Las palabras que forman el nombre serán minúsculas
PN-4 Único	PN-4.1 No puede haber más de un custom element con el mismo nombre

8. Caso práctico

Teniendo en cuenta las pautas de categorización, a continuación se muestra un ejemplo donde se hace uso de ellas para la categorización de componentes según su funcionalidad y negocio:

- **Átomos**

Dentro de esta categoría se va a hacer una subdivisión en función del ámbito que pueden tener los átomos. Existirán los átomos no funcionales y por otro lado los funcionales. Éstos últimos ya considerados Web Components:

- **Átomos no funcionales**

 Font-size-large Font-size-med Font-size-small Font-weight-normal Font-weight-strong	• PC1.1.1: Los elementos no presentan funcionalidad. En el ejemplo se puede comprobar que por ejemplo ni los colores ni los tamaños de letra o el grosor de la misma no proporciona ningún tipo de funcionalidad por sí solos. • PC1.1.2: Los elementos no se podrán dividir en ningún otro elemento. En el ejemplo, ni colores ni tamaños de letra ni grosores se podrán dividir en otros elementos. • PC1.1.3: Los elementos añaden se utilizan para añadir características o propiedades a otros elementos. ◦ Colores: Por ejemplo pueden indicar el estado de otro tipo de elementos como por ejemplo un botón. Si un botón es de color rojo puede indicar que su naturaleza es distinta a la de un botón de color verde. ◦ Tamaños o grosores de letra: Cuando a un elemento como un texto se le atribuye un
---	---

tamaño de letra o un grosor distinto, puede indicar que uno presenta más relevancia que el otro.

- PC1.1.4: Los elementos guardan relación con eventos. En el ejemplo, ni colores ni tamaños de letras o grosores pueden emitir o recibir eventos.

• Átomos funcionales



- **PC1.2.1:** Los componentes presentan funcionalidad pero no coherencia o utilidad por sí solos en la UI, además deben extender o redefinir elementos HTML básicos. En base al ejemplo:
 - 1) Componente botón: Un botón simple tiene funcionalidad sin embargo no presenta una utilidad o coherencia en la interfaz por sí solo, en ese caso necesita de otro elemento para conseguirlo. El componente surgiría a partir de un elemento `<button>`.
 - 2) Componente icono: Un icono simple tiene la funcionalidad de representar algo. En el ejemplo un icono que simula una persona corriendo puede estar relacionado con el deporte, sin embargo, hasta que no se combina con otro elemento no tiene coherencia en la interfaz por sí solo, así como estar acompañado de un texto o incrustado en un botón. El componente surgiría a partir de un elemento HTML `<i>` o `<span>`.
 - 3) Componente texto: Un texto, ya sea un título, subtítulo, párrafo... tiene la funcionalidad de representar algo sin embargo hasta que no se enlaza con otro elemento puede no tener coherencia en la interfaz. El componente podría surgir a partir de elementos HTML como `<h1>`, `<h2>`, `<label>`, `<p>`...
- **PC1.2.2:** Los componentes pueden tener estados a nivel de UI. En base al ejemplo, un botón 1) puede estar deshabilitado,

seleccionado o tener otro aspecto cuando el usuario posiciona el cursor sobre el mismo. Del mismo modo, un icono 2) o texto 3) puede tener un estado por ejemplo al cambiar el color de los mismos.

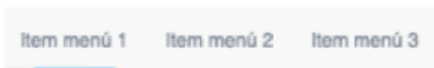
- **PC1.2.3:** Los componentes no pueden dividirse en otros componentes funcionales. En este caso ni botones 1) ni iconos 2) ni textos 3) se pueden descomponer en elementos con funcionalidad.
- **PC1.2.4:** Los componentes pueden emitir pero no recibir eventos. Por ejemplo un componente botón 1) puede emitir un evento cuando se haga clic sobre él, sin embargo el componente no recibirá ningún evento del exterior.

• Moléculas

1)



2)



3)

● Título input
Subtítulo input

4)



- **PC2.1.1:** El componente puede estar formado por una combinación de átomos y ninguna molécula
 - 1) Botón llamada = componente botón + componente icono llamada
 - 2) Menú de pestañas = componente botón + componente botón + componente botón
 - 3) Ítem menú simple = componente texto + componente subtexto + componente figura
- El componente puede estar formado por una combinación de átomos y una molécula más simple
 - 4) Ítem menú complejo: componentes átomos (texto + subtexto + descripción + enlace) + componente molécula (icono gota + botón)
- **PC2.1.2:** El componente presentan funcionalidad UI y la unión de sus elementos **tiene un propósito** :

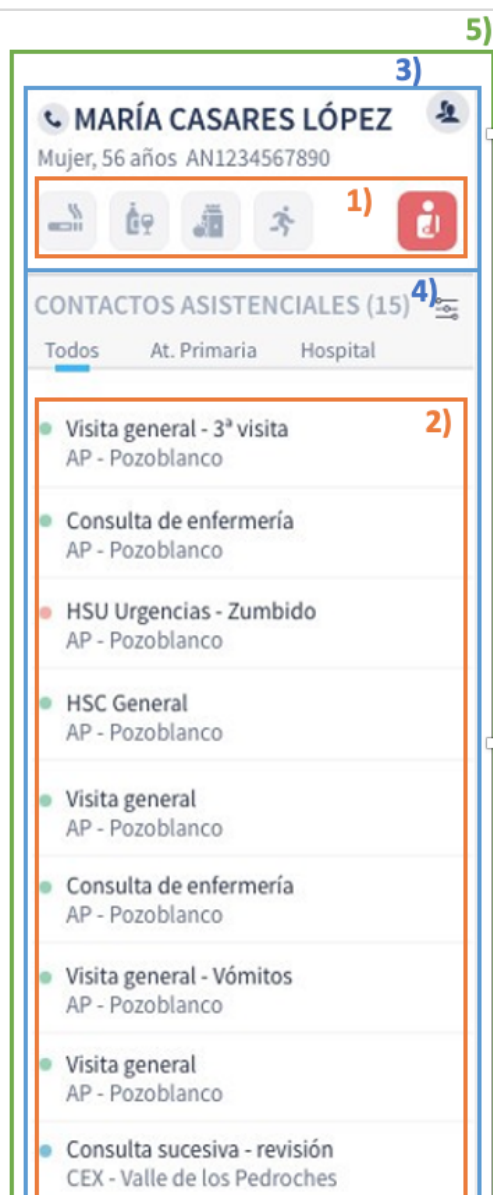
- 1) La unión del icono y el botón implica que se realizará algún tipo de acción relacionada con información que transmita el icono, en este caso un teléfono
- 2) La unión de componentes botones da lugar a un componente considerado como menú de pestañas donde se puede seleccionar cualquiera de las opciones haciendo clic sobre ellas.
- 3) La unión de los átomos da lugar a un ítem que guarda una coherencia en la UI como por ejemplo el ítem de un listado.
- 4) La gota de agua y el botón ya dan lugar a un componente molécula coherente en la UI. Si se combina con otros componentes átomos como textos título, texto subtítulo, enlace... Da lugar a otro componente más complejo que tiene coherencia en la UI como por ejemplo el ítem de un listado.

- **PC2.1.3:** Pueden **presentar estado UI** . Por ejemplo las moléculas formadas por componentes botones siguen pudiendo tener estado deshabilitado. En el caso del componente 3) también podrá asociarse un estado cuando el usuario se posicione sobre ella.

- **PC2.1.4:** Cada molécula citada es capaz de **emitir eventos** hacia los componentes que las contengan. También pueden escuchar los posibles eventos enviados por los elementos que las componen.

- 1) Al hacer clic sobre el botón, la molécula recibe el evento que a la vez lo puede emitir al exterior notificando al componente donde esté contenida.
- 2) El componente molécula recibe un evento por cada componente botón que se seleccione. Esa selección puede ser emitida al exterior.

• Organismos



- PC3.1.1:** Un componente organismo puede estar compuesto por dos o más molécula con o sin componentes átomos. Si se fijan en las áreas naranjas se pueden encontrar los siguientes ejemplos:
 - 1) Listado de componentes moléculas que guardan una coherencia funcional de negocio.
 - 2) Listado de componentes moléculas que dan lugar a un listado que carga información asociada al negocio.
- Un componente organismo puede estar compuesto por uno o varios componentes organismos y ninguno, uno/a o varios/as átomos/moléculas. Si se fijan en las áreas azules y verdes se pueden encontrar los siguientes casos:
 - 3) Conjunto de componentes átomos como el nombre de la persona y la descripción. Por otro lado moléculas como los botones que incluyen iconos. Finalmente se combina con otro componente organismo simple 1).
 - 4) Conjunto de componentes átomos como el título "Contactos...", componentes moléculas como el botón con icono situado a su derecha y el menú de pestañas y el organismo del listado de ítems 2).
 - 5) Unión de dos organismos que da lugar a una sección lateral de una interfaz y relacionado con información de una persona así como los contactos asociados a ella.
- PC3.1.2:** Un componente organismo presenta funcionalidad de interfaz y de negocio. Partiendo de los ejemplos previos:
 - 1) En este caso nos puede indicar que una persona tiene características como ser fumadora, consumidora de alcohol, practica deporte... Esa información es específica del ámbito de la aplicación y se conoce el origen de esa información como por ejemplo un servicio de un API.
 - 2) En este caso nos puede indicar los contactos asociados a una persona. También se conoce el origen de la misma.
 - 3) Combina la información de negocio del componente 1) y además se le atribuye otra información como datos personales o información de contacto.

- 4) Combina la información de negocio del componente 2) y además se incluyen otras características de negocio como categorización y filtrado de de contactos.
 - 5) Combina la información de negocio presente en los componentes 3) y 4) teniendo en cuenta datos personales de la persona y sus contactos asistenciales.
- **PC3.1.3:** Los componentes citados son capaces de emitir eventos los cuales podrán ser escuchados desde los componentes padres, ya sean otros organismos o la propia página de la aplicación. A su vez cada uno es capaz de gestionar los eventos emitidos por los elementos que los forman.
 - El componente 2) puede recibir eventos de los componentes internos como por ejemplo el clic sobre cualquiera de los ítems del listado. Ese evento se puede propagar desde el componente 2) hacia su componente contenedor, en este caso el 4). El componente 4) escuchará los eventos de 2) y de otros componentes como por ejemplo el menú de pestañas. En este punto, el componente puede orquestar los eventos para entablar comportamientos lógicos o permitir flujos de comunicación. Se puede imaginar que el usuario hace clic en otra pestaña del menú, ese evento lo recibirá el componente 4) y éste puede aplicar cambios sobre el componente 2) para que el listado se cargue aplicando un determinado filtro.

Por otro lado, 4) seguirá pudiendo propagar los eventos hacia el exterior, en este caso el componente 5) que este a su vez podrá reenviar la información a la página final

• Plantillas

- **PC4.1.1:** A partir de diferentes componentes ya sean átomos, moléculas u organismos, es posible crear componentes de carácter plantilla

Encabezado 1

Encabezado 2

TÍTULO SECCION

Item menú 1Item menú 2Item menú 3

Título input

Subtítulo input

Título input

Subtítulo input

Título input

Subtítulo input

Título input

Subtítulo input

o layout de tal manera que determinen la distribución de los componentes y apliquen estilado para llevarlo a cabo. En este caso se tendría un ejemplo de plantilla formado por una sección lateral a la izquierda y una sección de contenido a la derecha.

• Páginas

MARÍA CASARES LÓPEZ

Mujer, 56 años AN1234567890

CONTACTOS ASISTENCIALES (15)

TodosAt PrimariaHospital

Visita general - 3ª visita

AP - Pozoblanco

Consulta de enfermería

AP - Pozoblanco

HSU Urgencias - Zumbido

AP - Pozoblanco

HSC General

AP - Pozoblanco

Visita general

AP - Pozoblanco

Consulta de enfermería

AP - Pozoblanco

Visita general - Vómitos

AP - Pozoblanco

Visita general

AP - Pozoblanco

Consulta sucesiva - revisión

CEX - Valle de los Pedroches

PROGRAMAS/PROCESOS ASISTENCIALES DE SALUD/CARTERA

Planificación familiar

12 nov 2013

Última visita: 11 jul 2018

Mostrar visita

Diabetes

12 nov 2013

Última visita: 11 jul 2018

Mostrar visita

Embarazo

12 nov 2013

Última visita: 11 jul 2018

Mostrar visita

Cáncer cérvix/útero

12 nov 2013

Última visita: 11 jul 2018

Mostrar visita

PROBLEMAS DE SALUD ACTIVOS (3)

HIPERTENSIÓN ESENCIAL NEOM

DIABETES MELLITUS TIPO I NEOM

DEMATITIS ATÓPICA, NEUMONIA CONGENITA

PRESCRIPCIONES ACTIVAS (3)

PARACETAMOL 1G / CLORFENAMINA 2,80 MG 10 COMPRIMIDOS

RECUBIERTO CON PELÍCULA

1 / 12 horas

15 ene 2020 - 03 feb 2020

ANTALGIN 550 MG 10 COMPRIMIDOS RECUBIERTO CON PELÍCULA

1 / 12 horas

04 feb 2020 - 22 feb 2020

ANGELICA ARCHANGELICA 10ML, MANZANILLA 20ML, CARDIO MARIANO 10ML, CELIDONIA 10ML, REGALIZ 10ML, MELISSA OFFICINALIS HOJAS 10ML / ... (POR ML) 100ML SOLUCION ORAL/ GOTAS

1 / 4 horas

01 dic 2019 - 10 mar 2020

ALERGIAS ACTIVAS (3)

PARACETAMOL

IBUPROFENO

AMOXICILINA

CONTRAINDICACIONES ACTIVAS (3)

ZOLMITRIPTAN

ACEITE DE SOJA-ACEITE DE OLIVA

ALPRAZOLAM

- **PC5.1.1:** En la página entra en juego la gestión de eventos. Ésta será el destino final de los eventos emitidos por los componentes que la forman. A partir de ellos se puede establecer comunicación y desencadenar diferentes comportamientos.
- **PC5.1.2:** En la página se gestiona la navegación, es decir, si se imagina que él hace clic en un botón del menú lateral, la página recibe el evento y acto seguido navega hacia otra página distinta o aplica un scroll para que el foco de la ventana se centre en un sección determinada (en caso de haber scroll).
- **PC5.1.3:** En la página se gestiona el estado general. En este caso el estado general podría estar relacionado con un identificador asociado a la persona de la que se está consultando la información.

Este estado será transversal para los componentes de negocio que dan lugar a la página.