

Buenas prácticas y Arquitectura CSS

upna

Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa





María Barrios

Frontend Developer & Manager

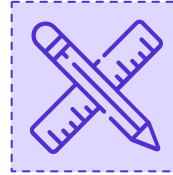
JAKALA



¿Qué vamos a ver?



Web semántica



**Buenas prácticas
CSS**



Arquitectura ITCSS



Hablemos un poco de *semántica*

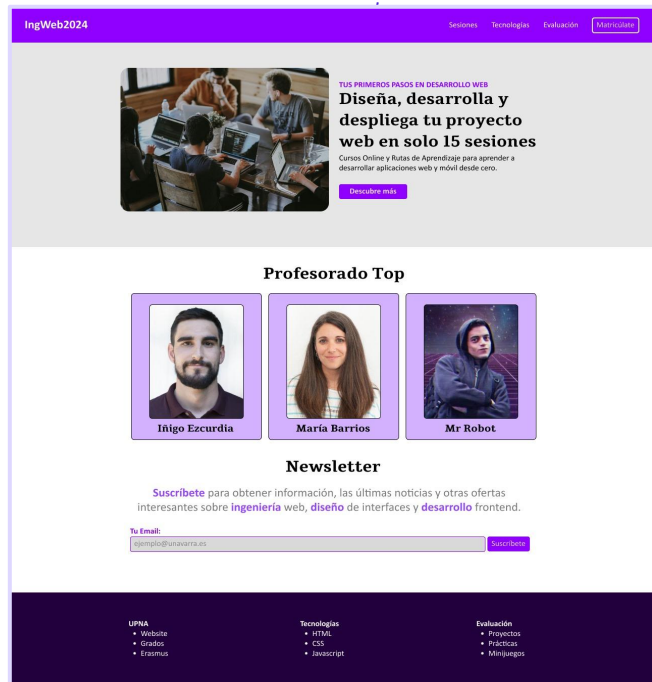


¿Qué es una web semántica?

Una web con toda la **información organizada de manera que no sólo las personas puedan entender, sino también las máquinas**: ordenadores, móviles, videoconsolas, etc.

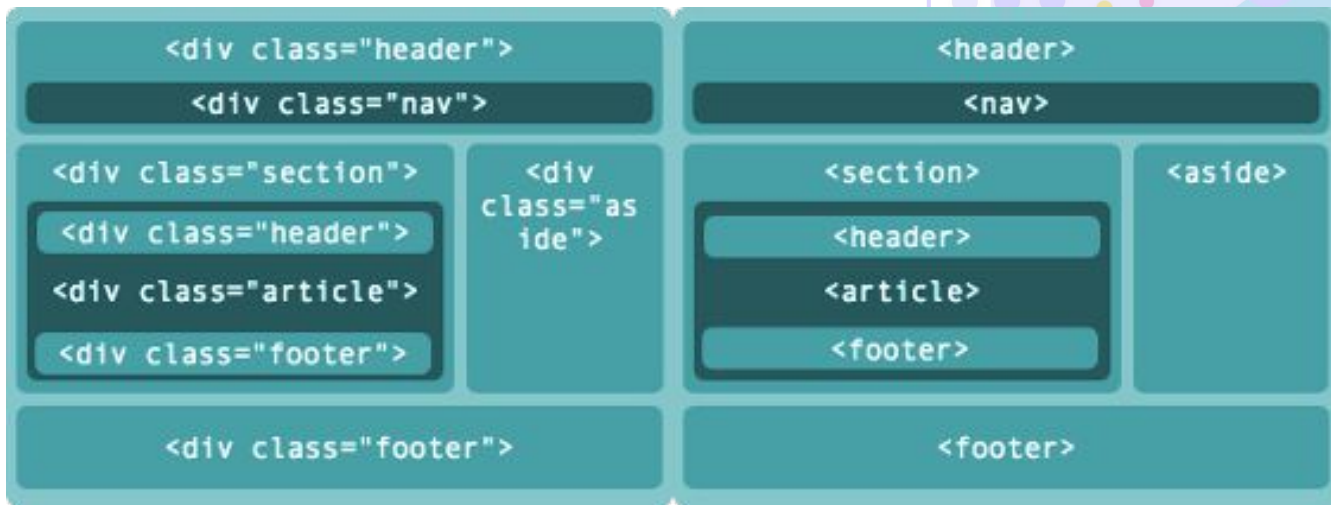
Destinada a **mejorar el contenido, mejorando la experiencia de los resultados** y al mismo tiempo la **experiencia de los usuarios**.

- Ayuda a que la web sea accesible para personas con discapacidades
- Facilita la legibilidad y la mantenibilidad del código.
- Favorece que los robots o crawlers sean capaces de interpretarlo e indexar nuestro contenido correctamente en sus buscadores (googlebot).



¿Cómo hacemos una web semántica?

HTML define una serie de **etiquetas** que se utilizan para **marcar los diferentes elementos** que componen una página:

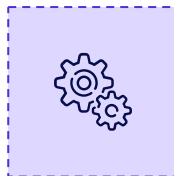


Cosicas a tener en cuenta



Usar etiquetas de sección para dar contexto

`<header>`, `<main>`,
`<footer>`, `<article>`,
`<section>`, `<aside>`, `<nav>`



Usar etiquetas de “acción” para su fin

Evitar `<div>` con atributo `onClick` para elementos de acción sustituyendo a `<button>`



Cada etiqueta tiene su rol, utilízalas bien.

El navegador ya las interpreta con todas sus características asociadas.

- `` `` para listados
- `<h1>` `<h2>``<h*>` para títulos con la jerarquía esperada
- `<p>` para párrafos, etc.



Uiss... yo no he puesto esos estilos



User Agent

El **navegador web** le aplica sus propios estilos y se verá reflejado en el inspector como *user agent stylesheet*.

```
h1 {                                     user agent stylesheet
  display: block;
  font-size: 2em;
  margin-block-start: 0.67em;
  margin-block-end: 0.67em;
  margin-inline-start: 0px;
  margin-inline-end: 0px;
  font-weight: bold;
}
```

Cada **navegador** tiene sus propios estándares. Por eso **es bueno** dentro de los proyectos, **resetear dichos estilos**.



¿Cómo podemos evitar esto?

Mediante **ficheros de reseteo o normalización de estilos**:

`reset.css`, `reboot.css`, `normalize.css`

reset.css: para **resetear todos los estilos** de los diferentes navegadores.

Fichero de Eric Meyer

reboot.css: configurar unos estilos iniciales, que son comunes a cualquier proyecto para **dejarlos como nos gustaría que los navegadores lo hiciesen por defecto**.

Ejemplo reboot de Bootstrap.

normalize.css: para que los navegadores **muestren todos los elementos de forma más coherente y acorde con los estándares modernos**. Se dirige solo a los estilos que necesitan normalizarse.

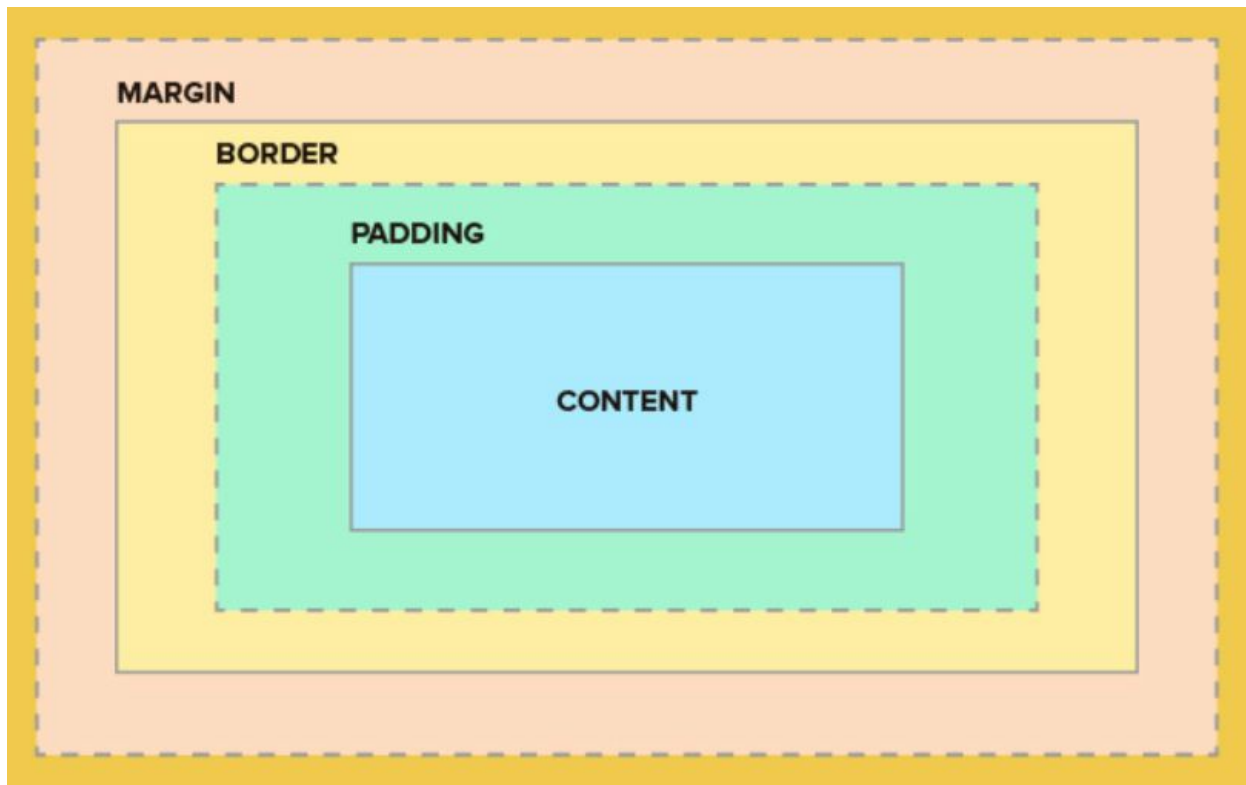
De Nicolas Gallagher.

custom.css: podemos **crear nuestro propio fichero** de normalización /

reseteo.



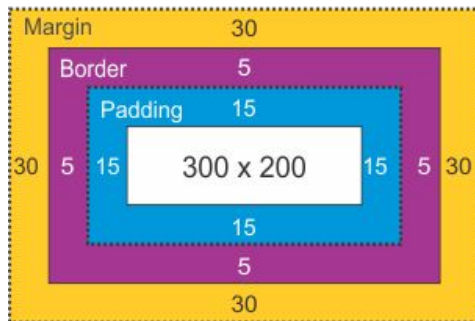
Modelo caja



¡IMPORTANTE! La propiedad Box-sizing

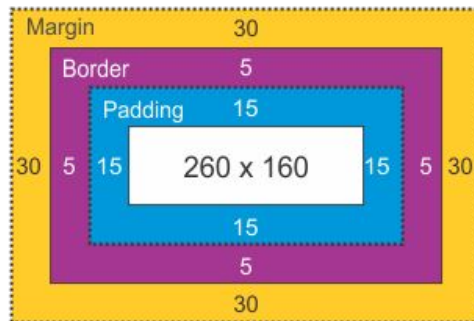


Box-sizing: content-box;



```
div{  
  width: 300px;  
  height: 200px;  
  padding: 15px;  
  border: 5px solid grey;  
  margin: 30px;  
  -moz-box-sizing: content-box;  
  -webkit-box-sizing: content-box;  
  box-sizing: content-box;  
}
```

Box-sizing: border-box;



```
div{  
  width: 300px;  
  height: 200px;  
  padding: 15px;  
  border: 5px solid grey;  
  margin: 30px;  
  -moz-box-sizing: border-box;  
  -webkit-box-sizing: border-box;  
  box-sizing: border-box;  
}
```

Variables CSS

Nos permite **almacenar valores que se repiten en diferentes clases** (*colores, tamaños*) en un **único lugar y ser utilizadas** posteriormente en **diferentes clases**, ofreciendo de este modo **mantenibilidad, coherencia**.



Cómo se declara

```
::root {  
  --color-light: #E8F8F5  
  --size-s: 12px  
}
```



Cómo se usa

```
.my-component {  
  background-color: var(--color-light);  
  margin-bottom: var(--size-s);  
}
```



¿Unidades absolutas? No more pixels

Em y rem son dos unidades de CSS relativas al font-size existente.



Tiene referencia el tamaño de la fuente del elemento **donde se usa** esa unidad.



Tiene referencia el tamaño de la fuente del elemento definido en el **elemento root.**



Escalado proporcional a la **configuración del usuario del navegador.**

Utilizar en todos los elementos que requieran tamaño **margins, paddings, position:absolute, width, height, font-size, border**, etc.

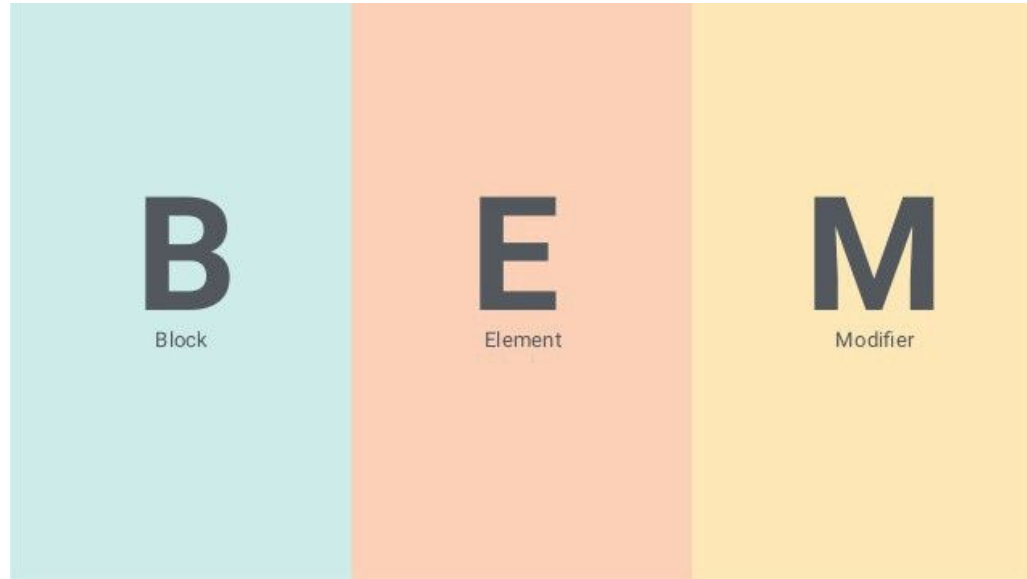




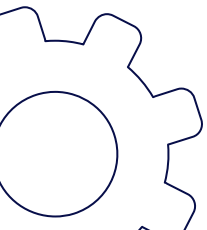
**Y el naming,
¿qué tal ha
quedado?**



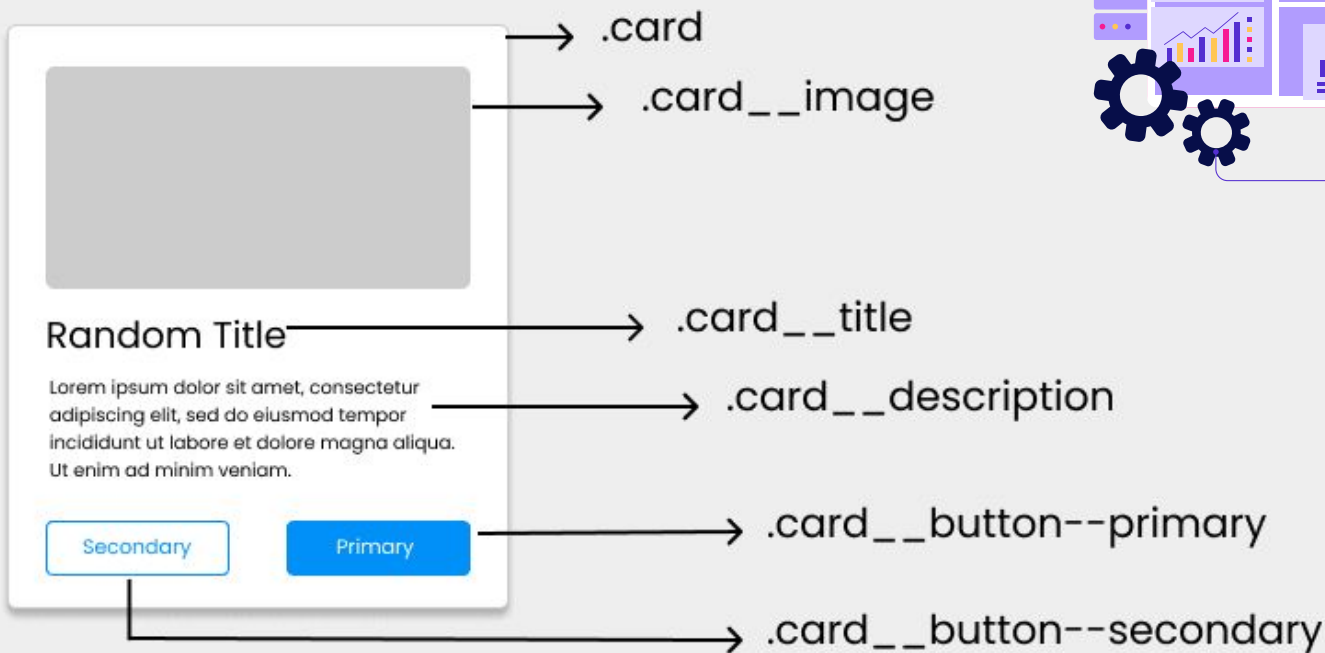
Notación BEM



block_element--modifier



Notación BEM



Propiedades CSS organizadas Outside In

1. Layout properties



position, display, flex-properties...

2. Box model properties



width, height, margin, padding...

3. Visual properties



color, background, border, box-shadow...

4. Typography properties

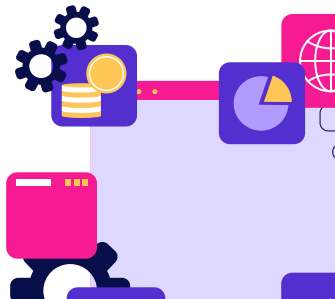


font-size, font-weight, text-align, text-transform...

5. Misc properties



cursor, overflow, z-index





**Un `main.css`
para
gobernarlos
a todos..**





Objetivos de una buena arquitectura CSS



Predecible

Reglas hacen lo que se espera.
Añadir o actualizar una regla **no**
tiene consecuencias
inesperadas



Reutilizable

Reglas **abstractas y desacopladas**.
Se podrán **crear componentes** de
manera **más rápida y sencilla**



Mantenible

Capacidad de **realizar cambios**
sin tener que refactorizar el
código existente.

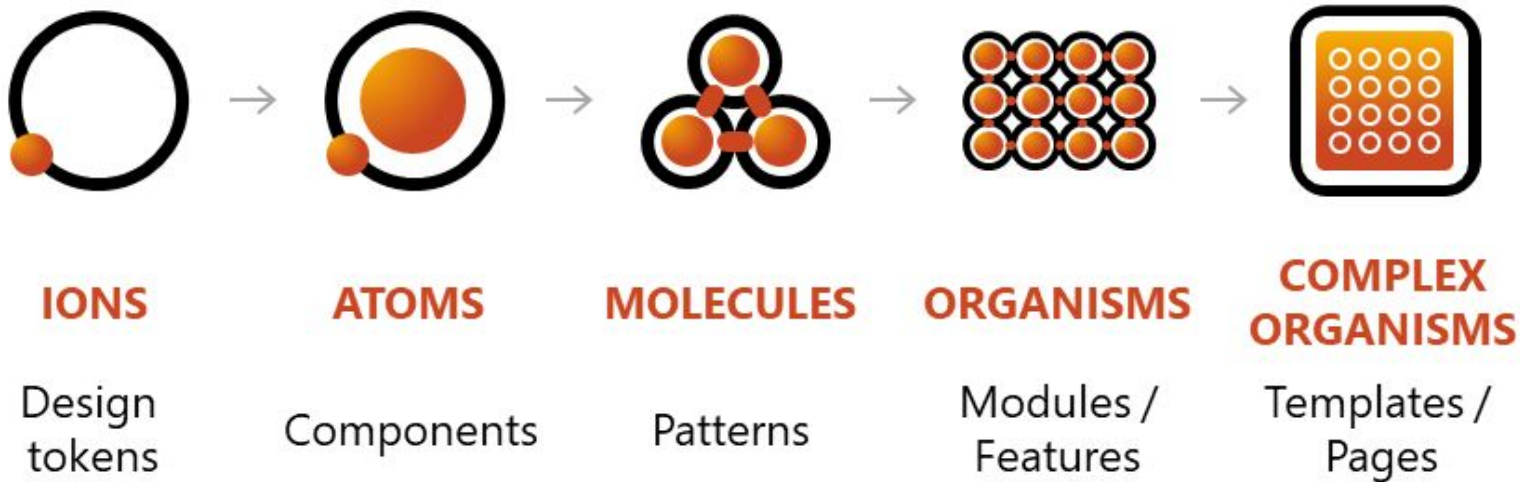


Escalable

A medida que **aumenta el**
tamaño del CSS sigue siendo
fácil de mantener.

Atomic Design

Metodología de diseño que está justo en la **intersección entre diseño y desarrollo**, viene a unir ambas disciplinas bajo un concepto: **diseñar por componentes**.



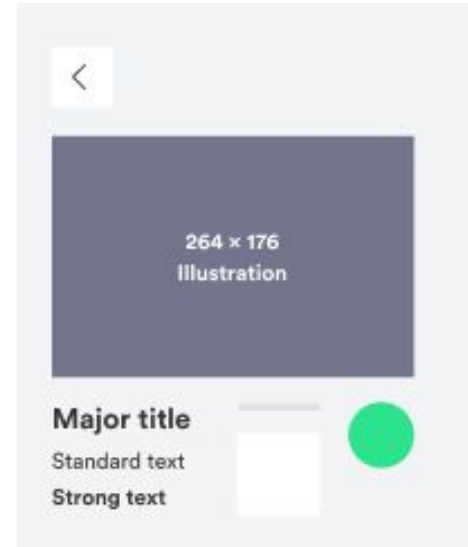
Atomic Design

Atoms



Elementos UI que **por sí solos tienen alguna funcionalidad: botones, avatares, inputs de formulario...**

Base Unit [BU]	8x8
Spacing Unit [SU]	16x16
Margin Unit [MU]	24x24
Tap Unit [TU]	48x48



Atomic Design

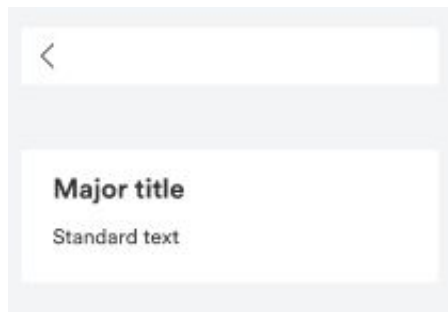
Molecules



COMPONENTS

Buttons, Input Box,
Search Box, Menu

Es la **unión** de diferentes **átomos** para **formar un elemento UI más complejo**, como **cards, formularios...**



Atomic Design

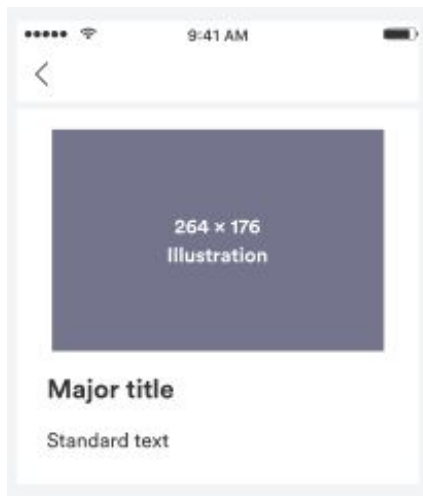
Organisms



PATTERNS

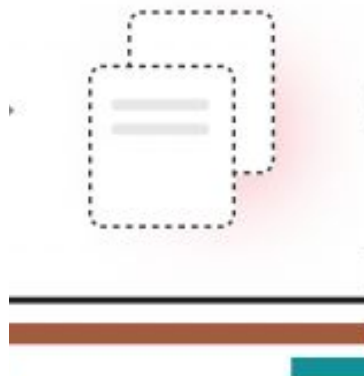
Navbar, Rating
Widgets, Cards

Elementos de UI **formados por moléculas** (*que a su vez son átomos*) y que, por encima de todo, **pueden repetirse**.



Atomic Design

Templates



TEMPLATES

Skeleton of a screen
/Page/Modal

Unión de **distintos organismos** que forman una página o aplicación. Se trata del “**esqueleto**”.



Atomic Design

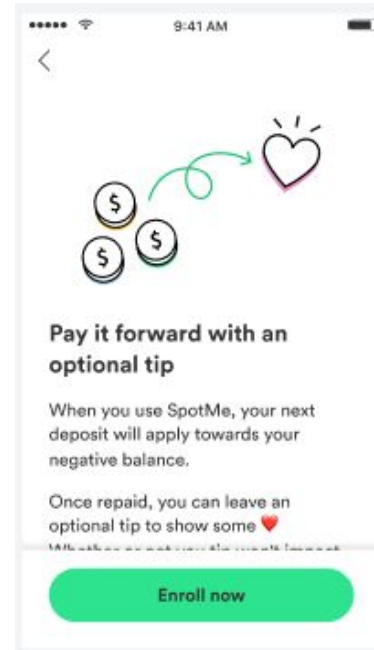
Pages/Screens



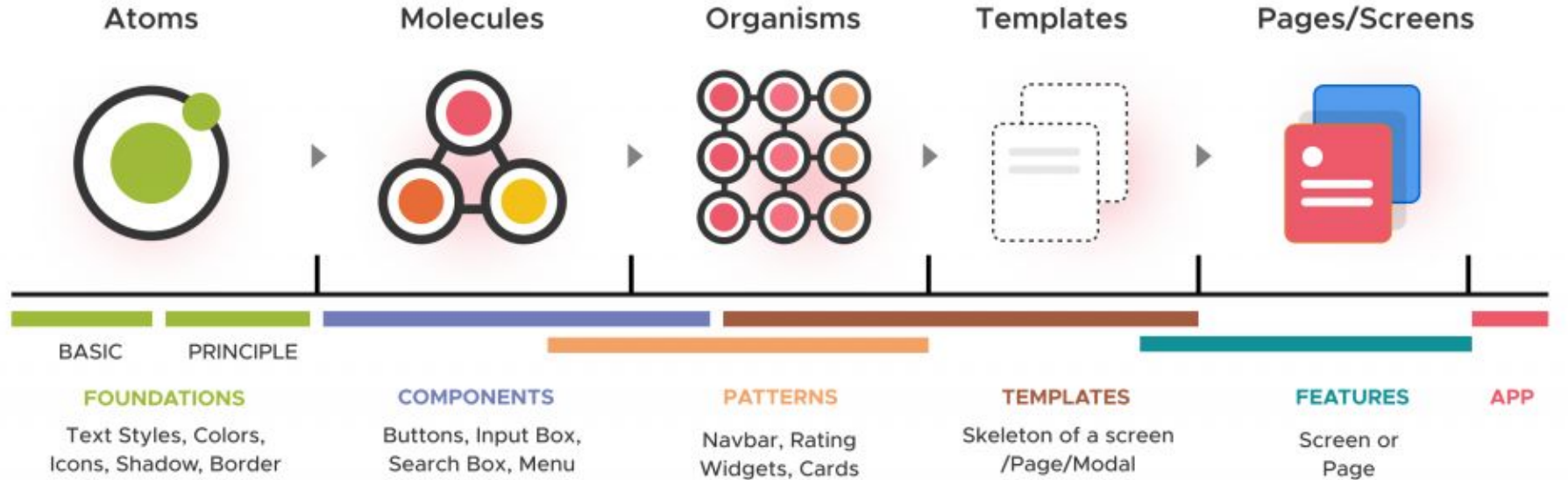
FEATURES

Screen or
Page

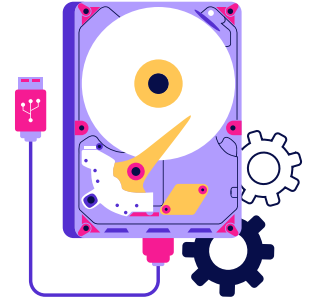
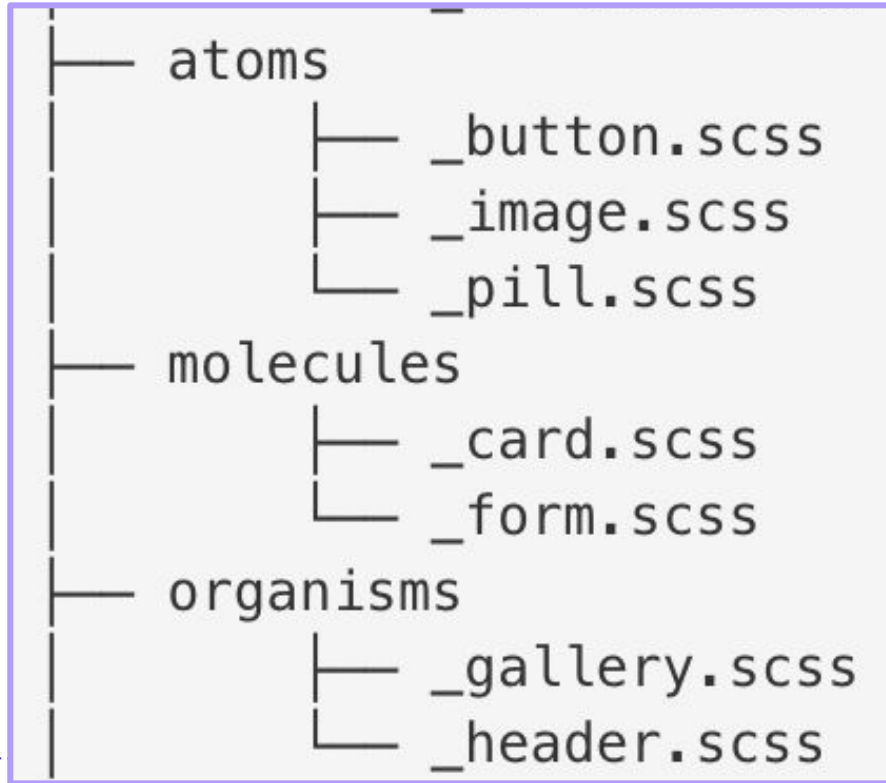
Diseño final que ya contiene las imágenes y otros detalles que hacen que ese archivo ya esté **listo para el desarrollo**



Atomic Design



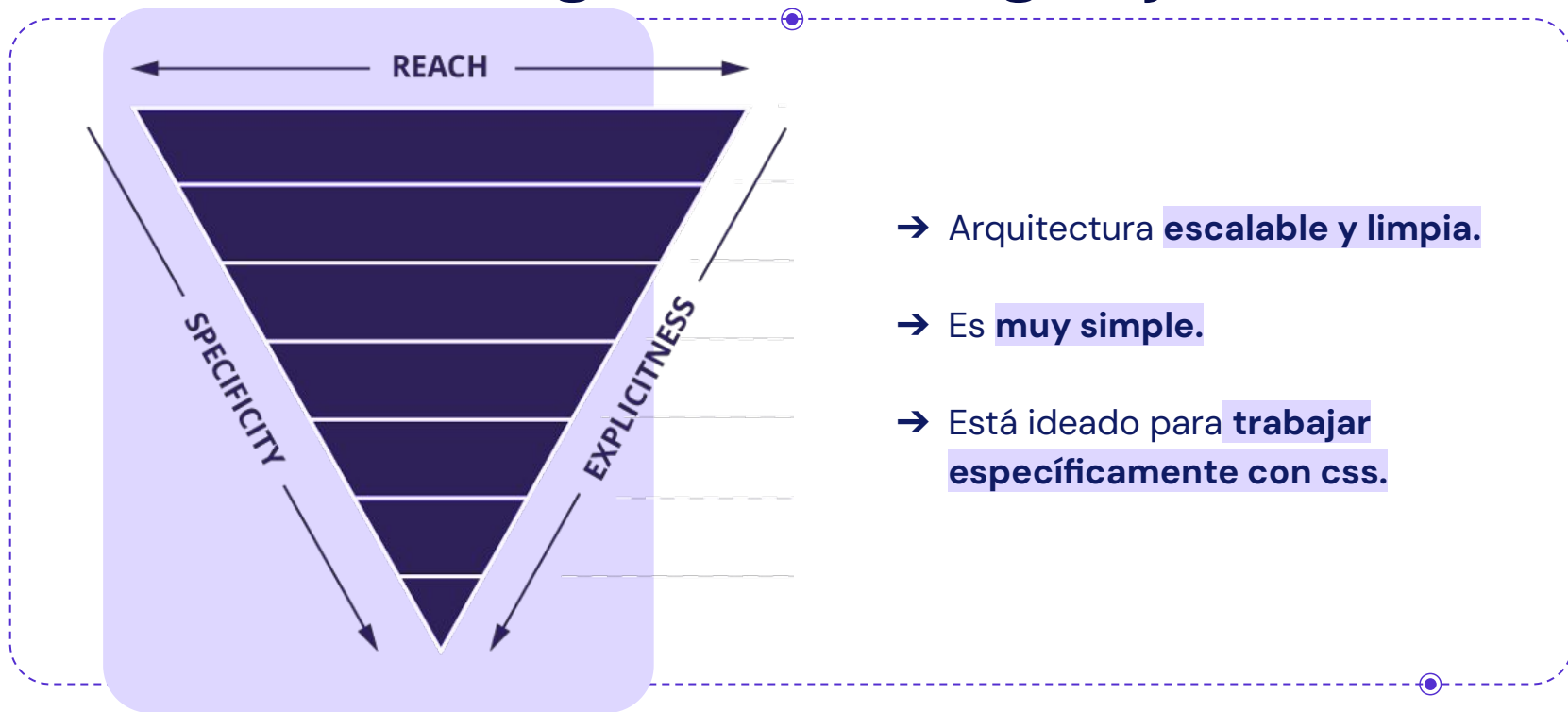
Atomic Design



"ARQUITECTURA ITCSS"

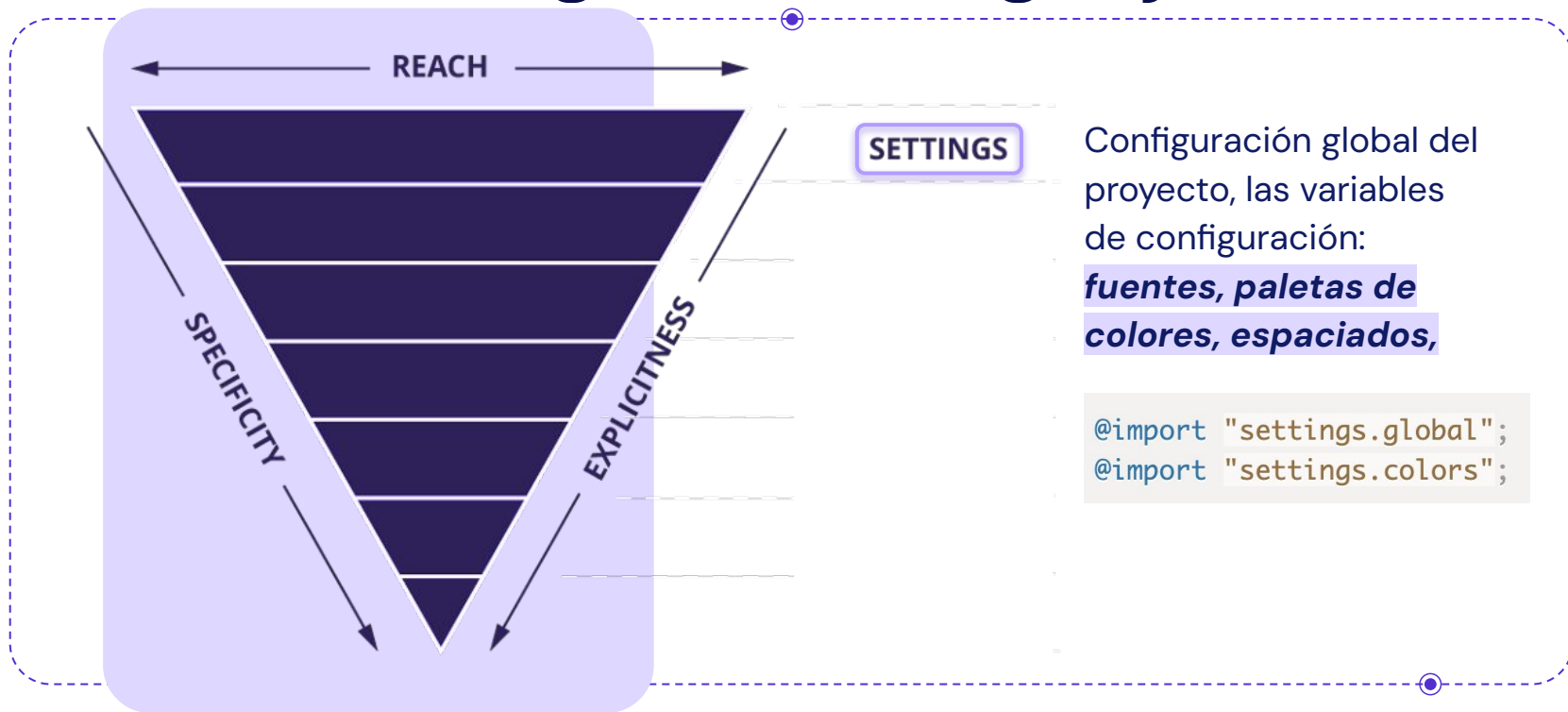


Inverted Triangle Cascading Style Sheet



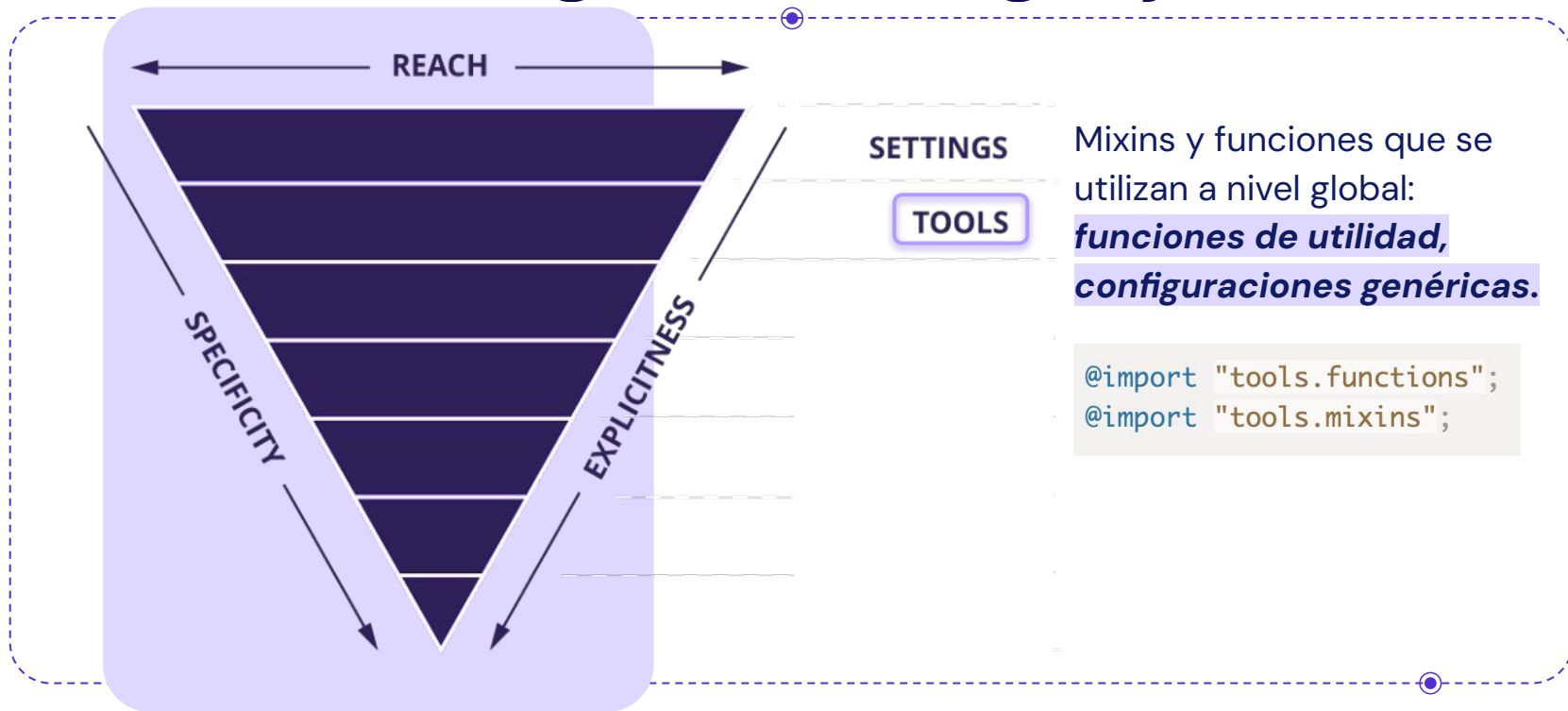


Inverted Triangle Cascading Style Sheet



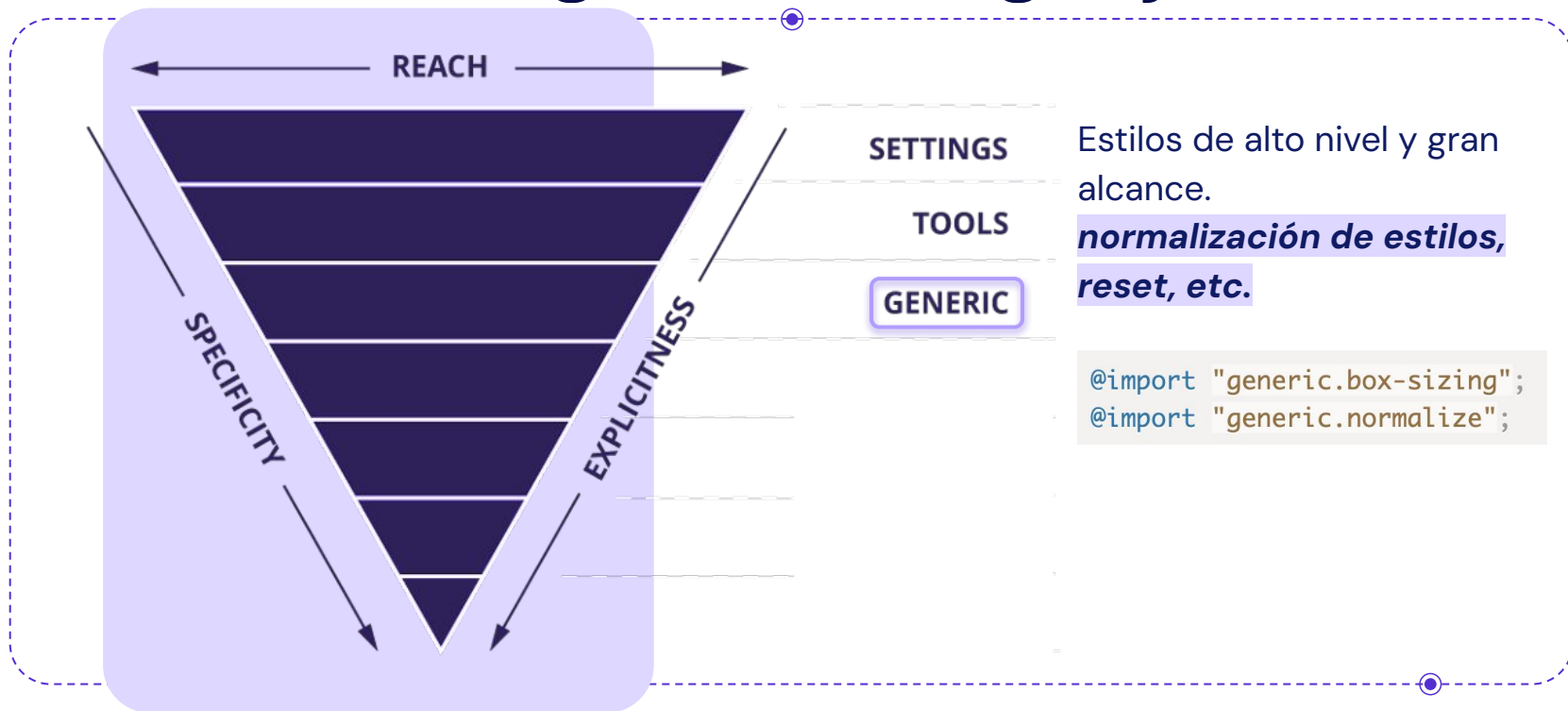


Inverted Triangle Cascading Style Sheet



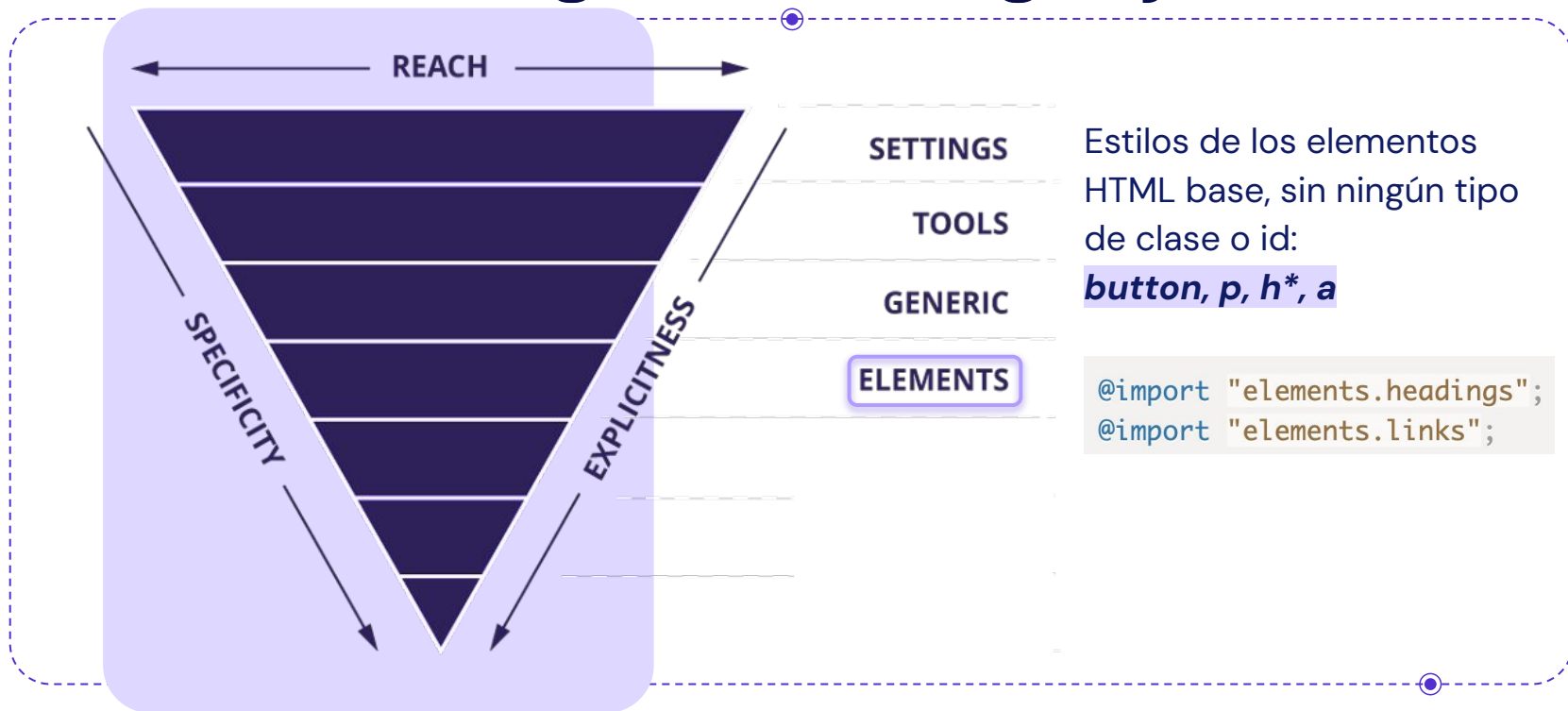


Inverted Triangle Cascading Style Sheet



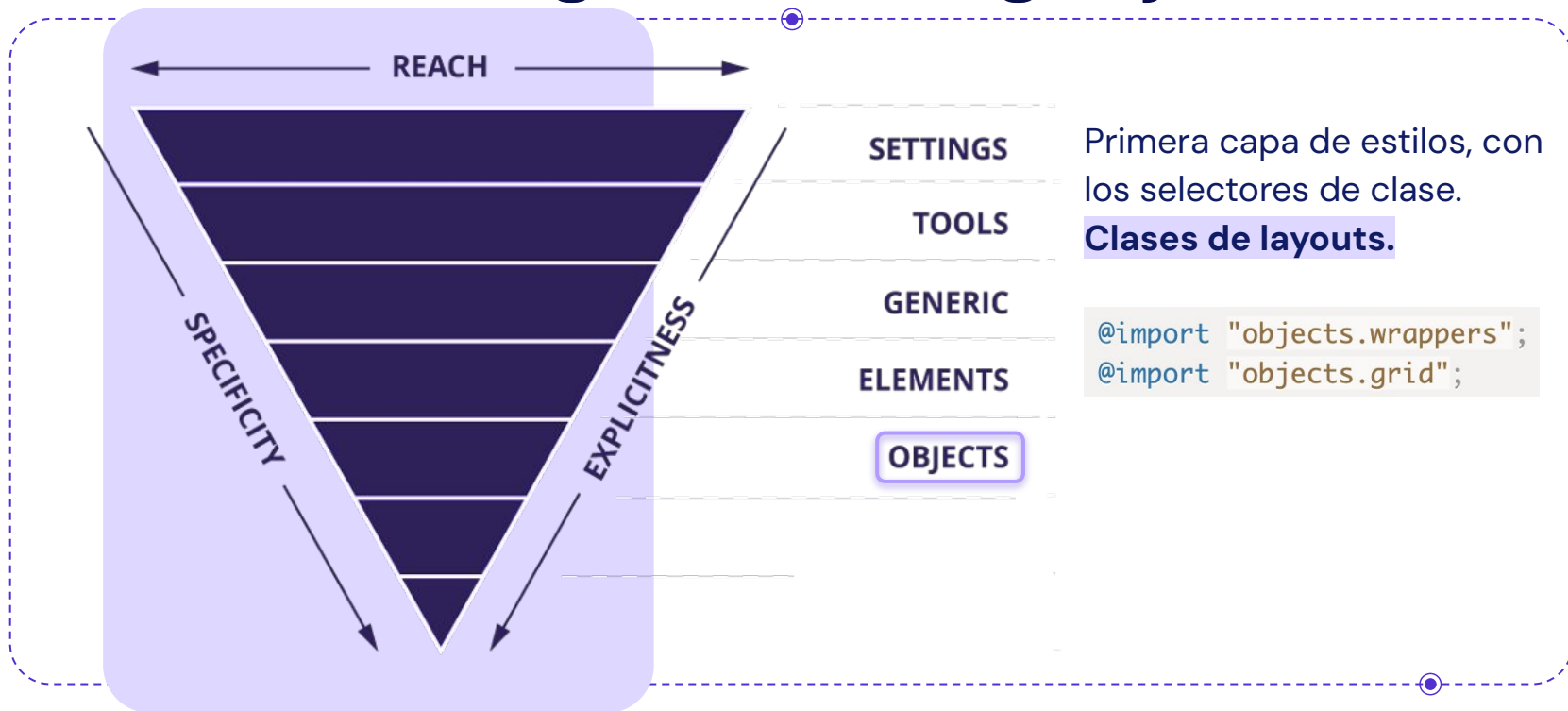


Inverted Triangle Cascading Style Sheet



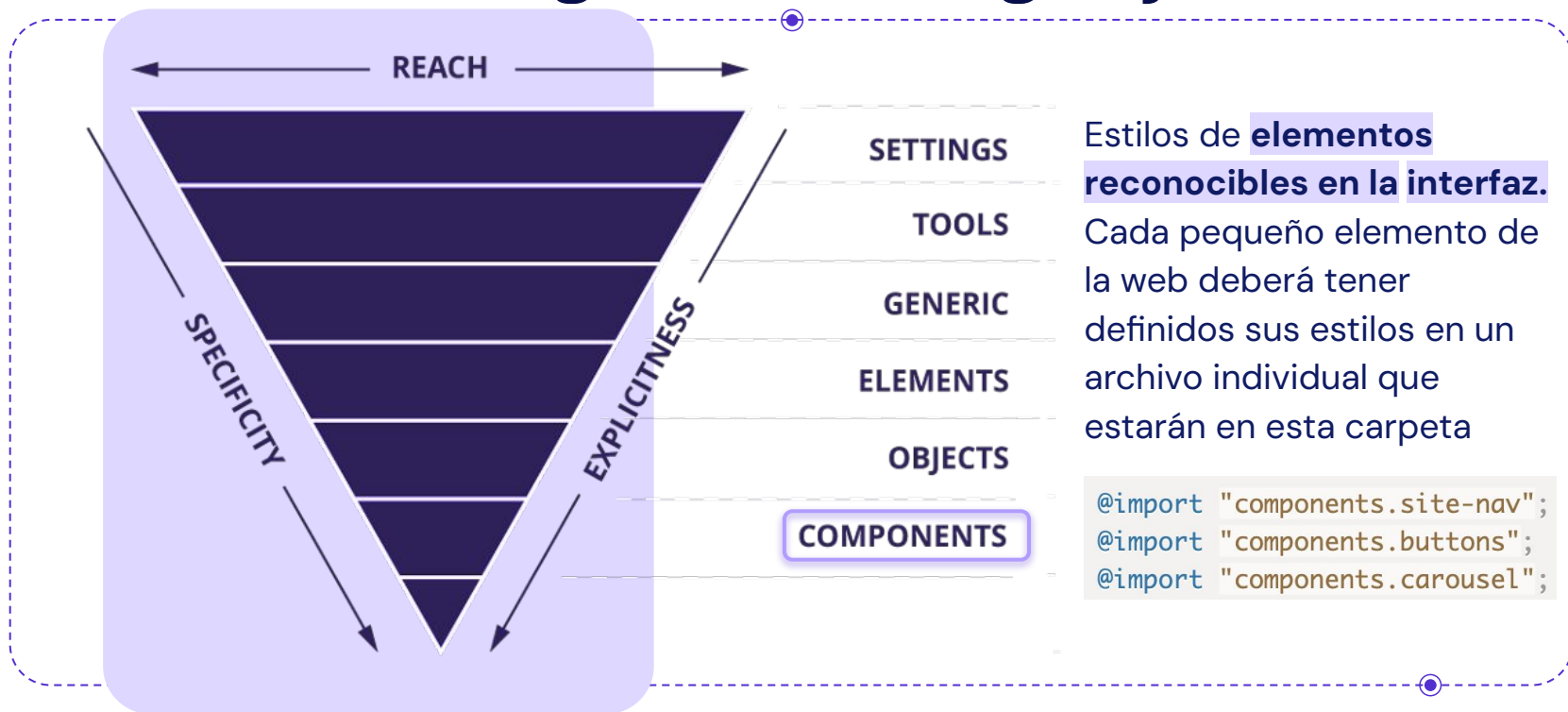


Inverted Triangle Cascading Style Sheet



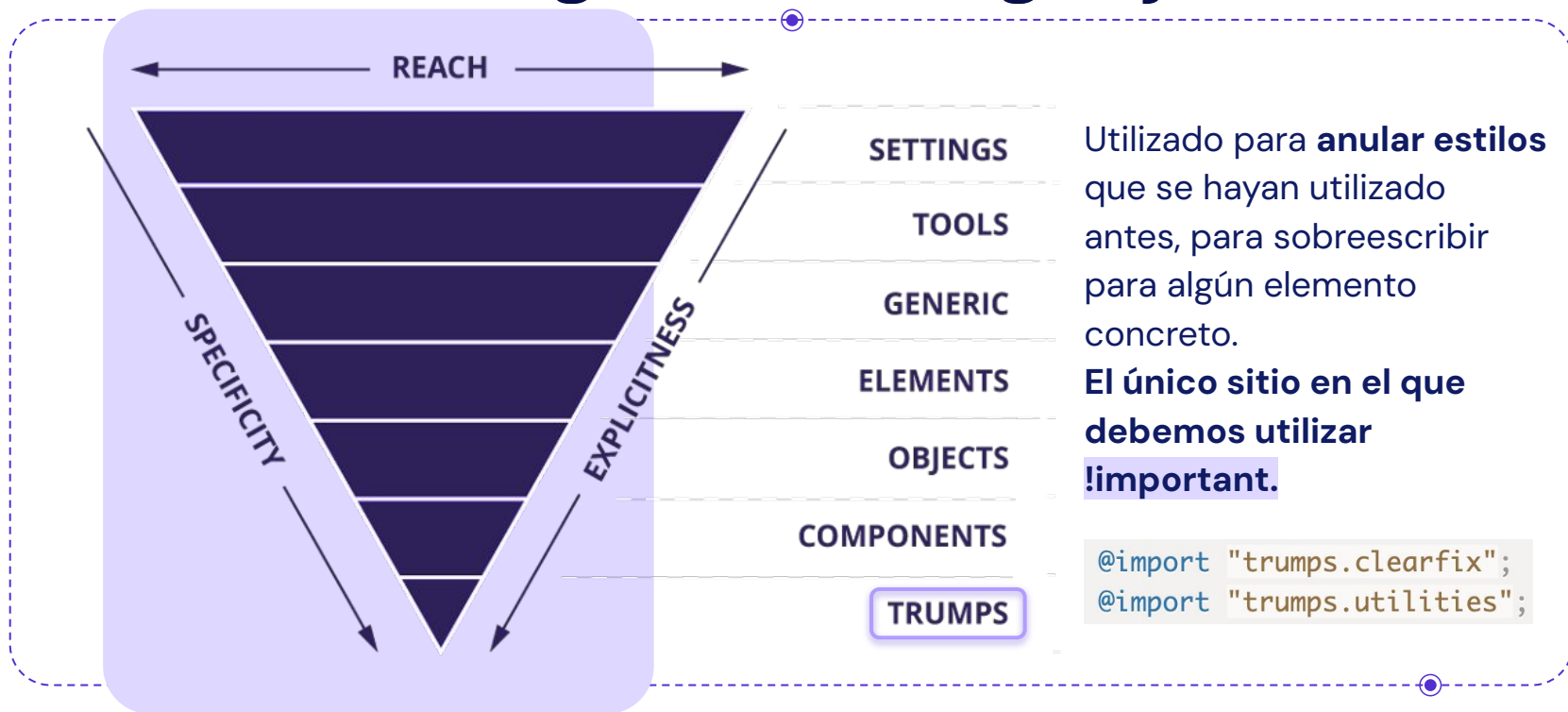


Inverted Triangle Cascading Style Sheet



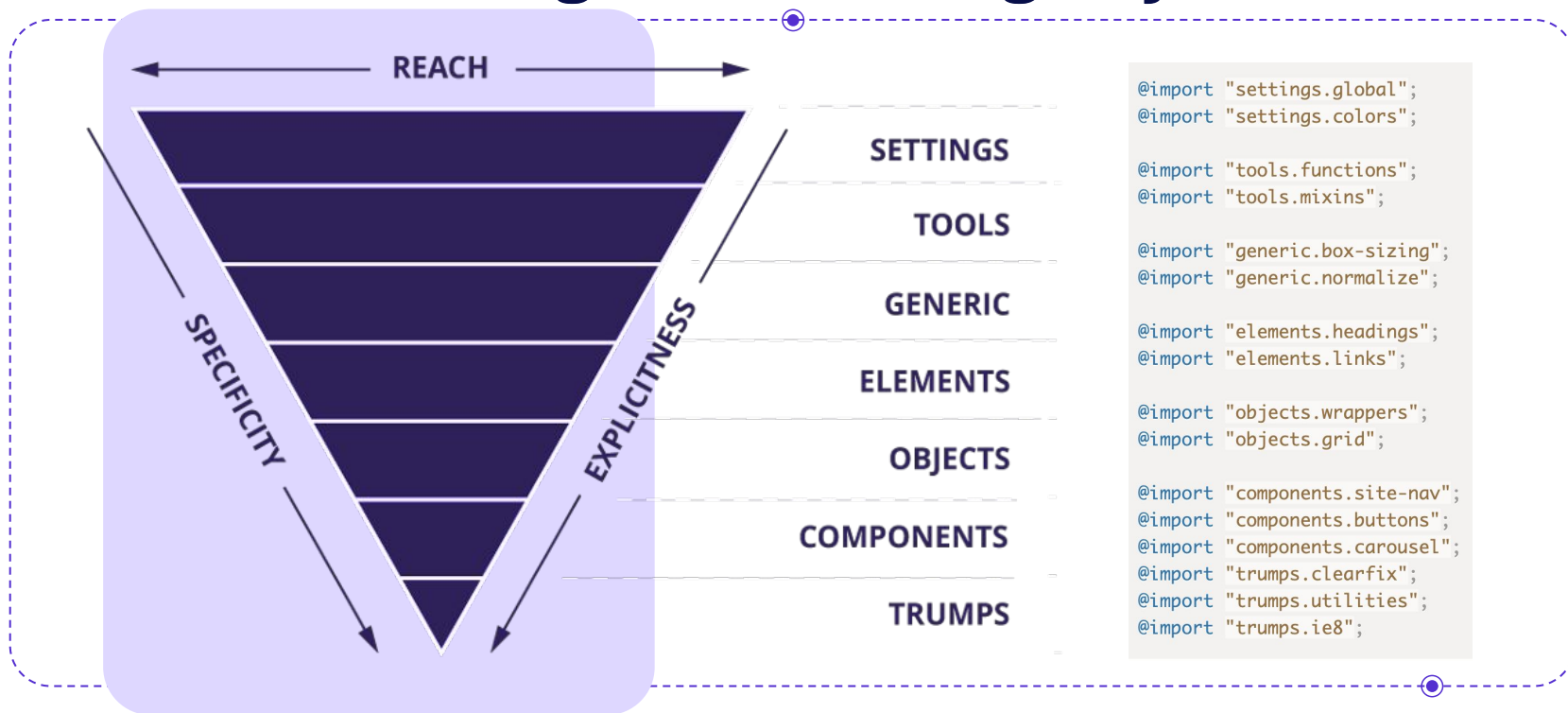


Inverted Triangle Cascading Style Sheet





Inverted Triangle Cascading Style Sheet





**¿Hace un
refactoring?**





Apúntante a la DXP School

