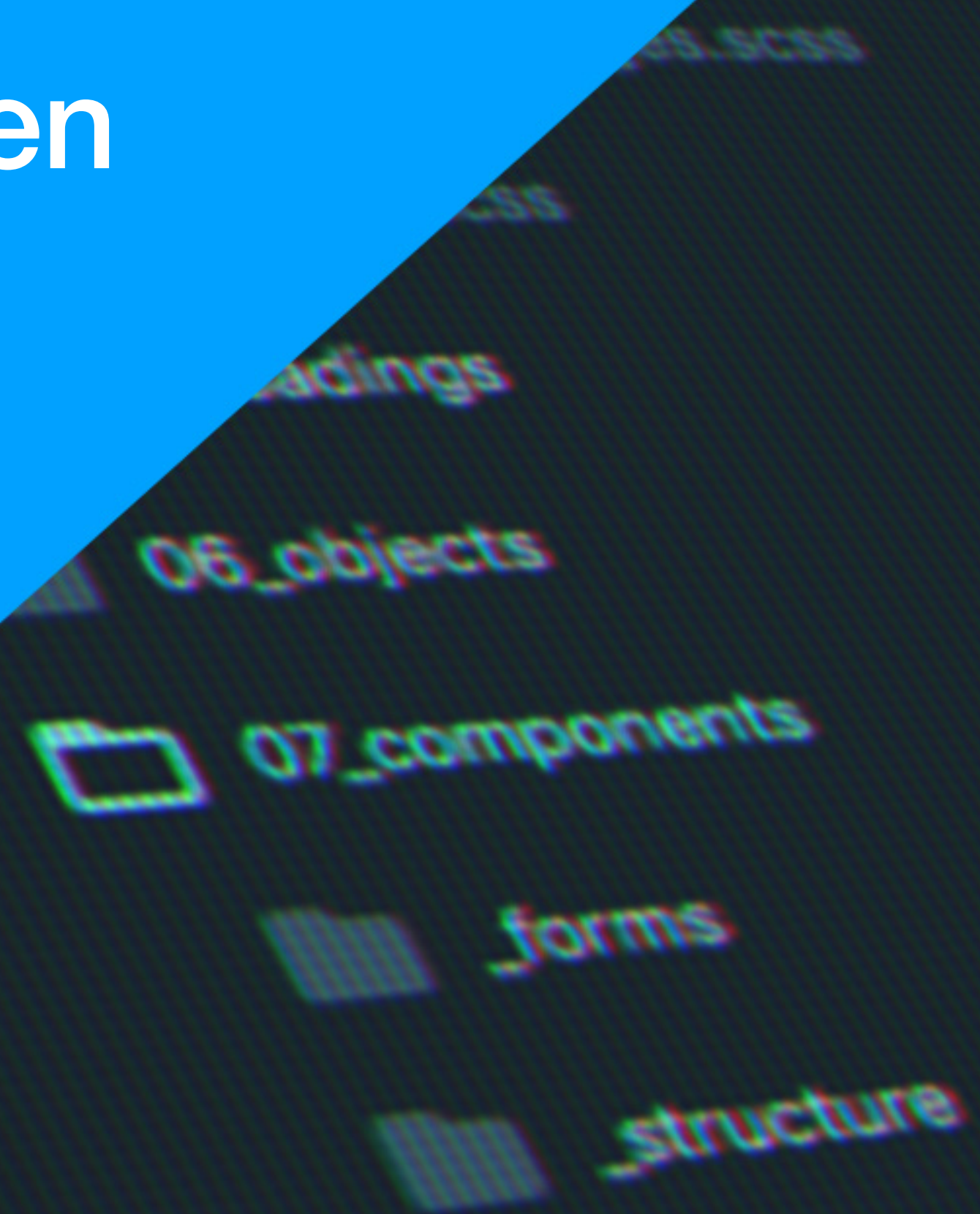


Desarrollo de arquitecturas en CSS basadas en ITCSS

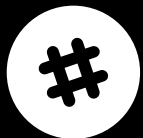
Juan Carlos Ruiz Carrión



Juan Carlos Ruiz Carrión

UI / Front-end developer

Dev:



@klaufel

Party hard:



@klaufel



Klaufel#2373





Fer

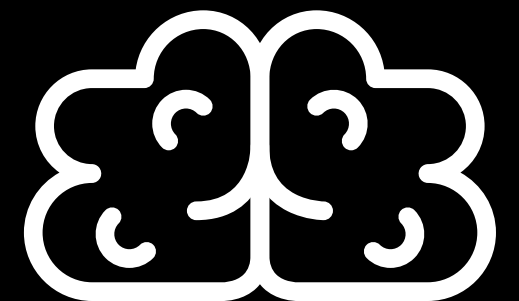


Edu



Juan Carlos

Conociendo ITCSS

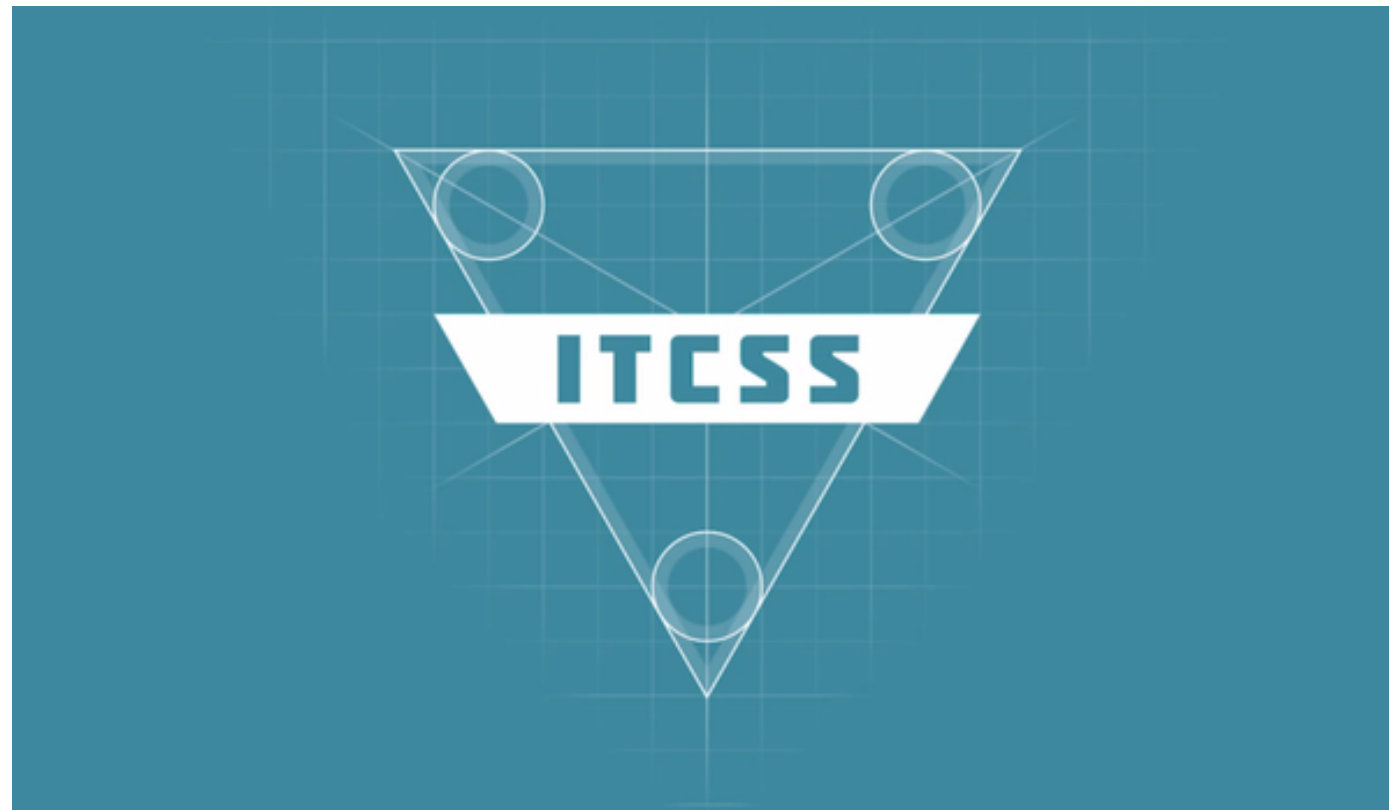




Harry Roberts

@csswizardry

¿Qué es ITCSS?



Inverted Triangle architecture for CSS

(Arquitectura de triángulo invertido)

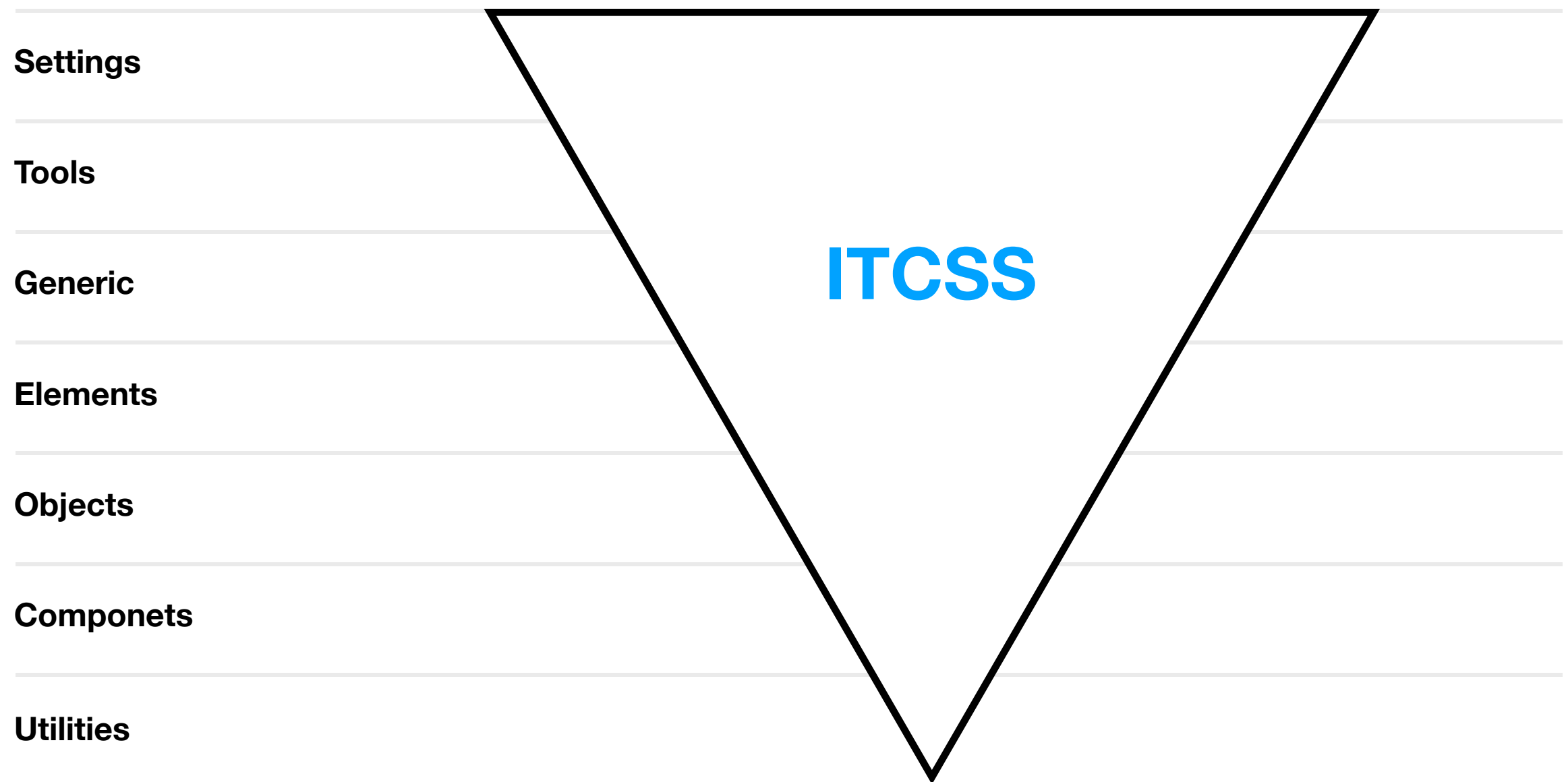
¿Qué es ITCSS?

- Una propuesta de arquitectura CSS
- Estructura para organizar las hojas de estilo
- Arquitectura basada en la especificidad

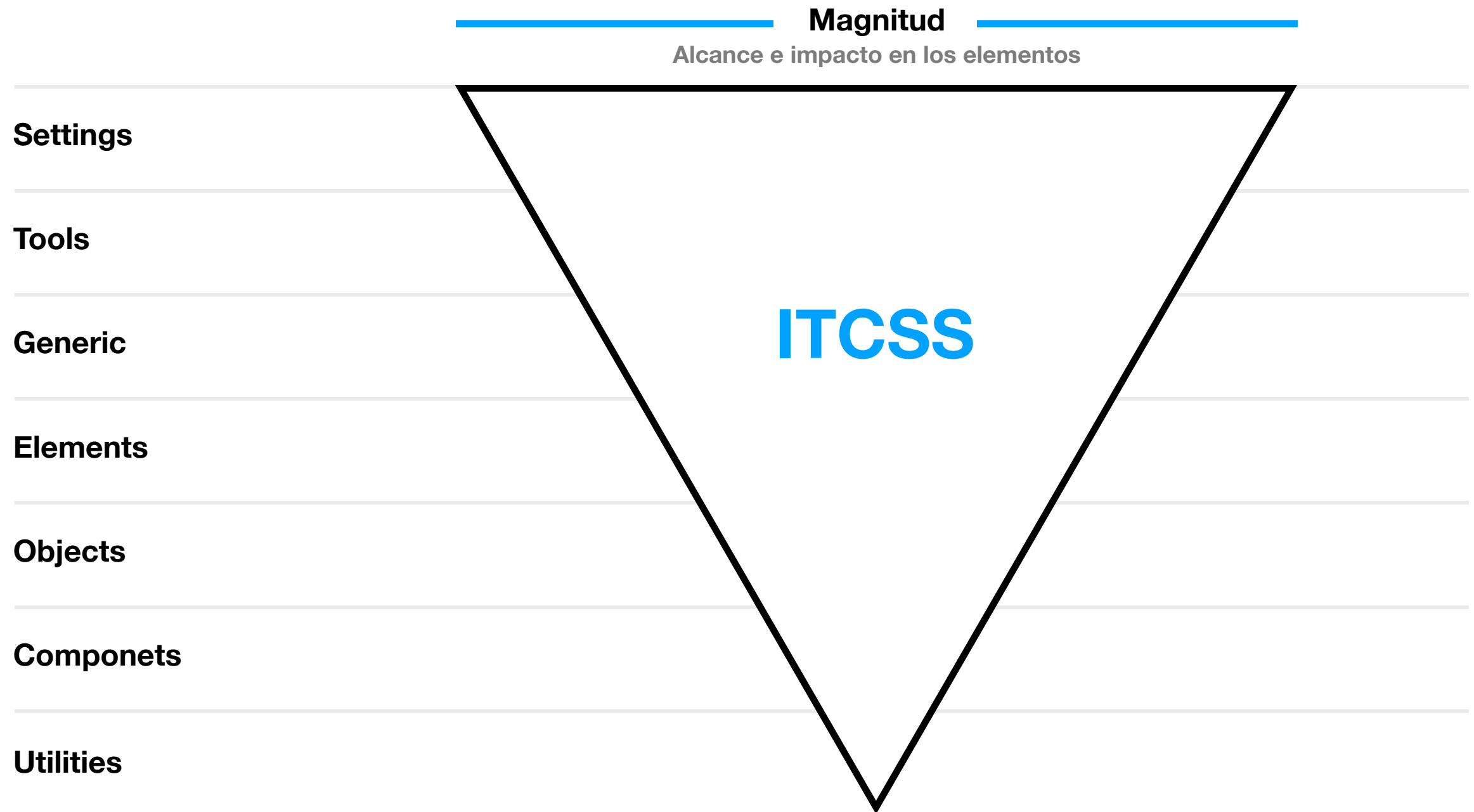
¿Qué es ITCSS?

“ITCSS nos va ayudar a escribir un CSS mucho más sano, flexible, reutilizable, comprensible y manejable.”

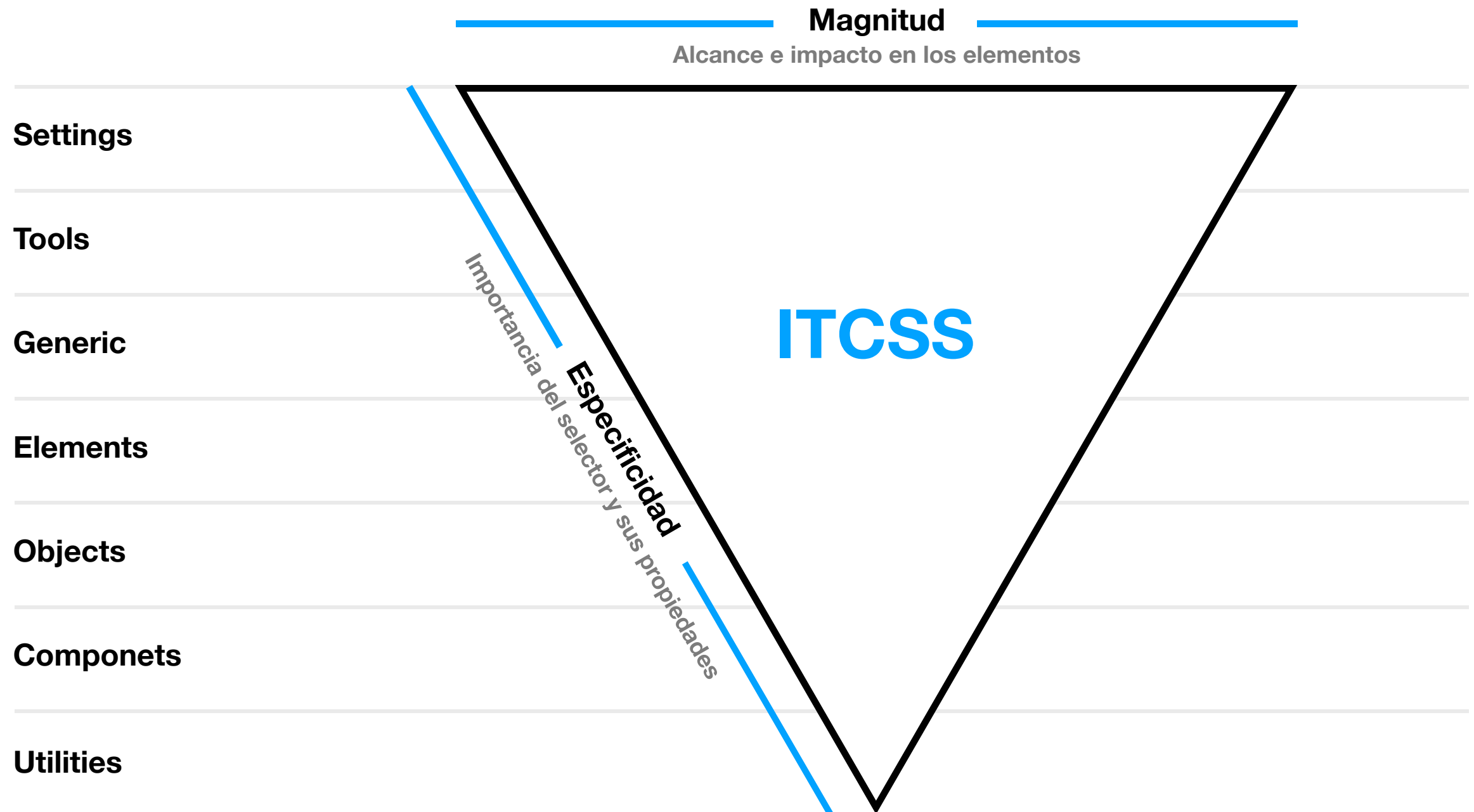
¿Qué es ITCSS?



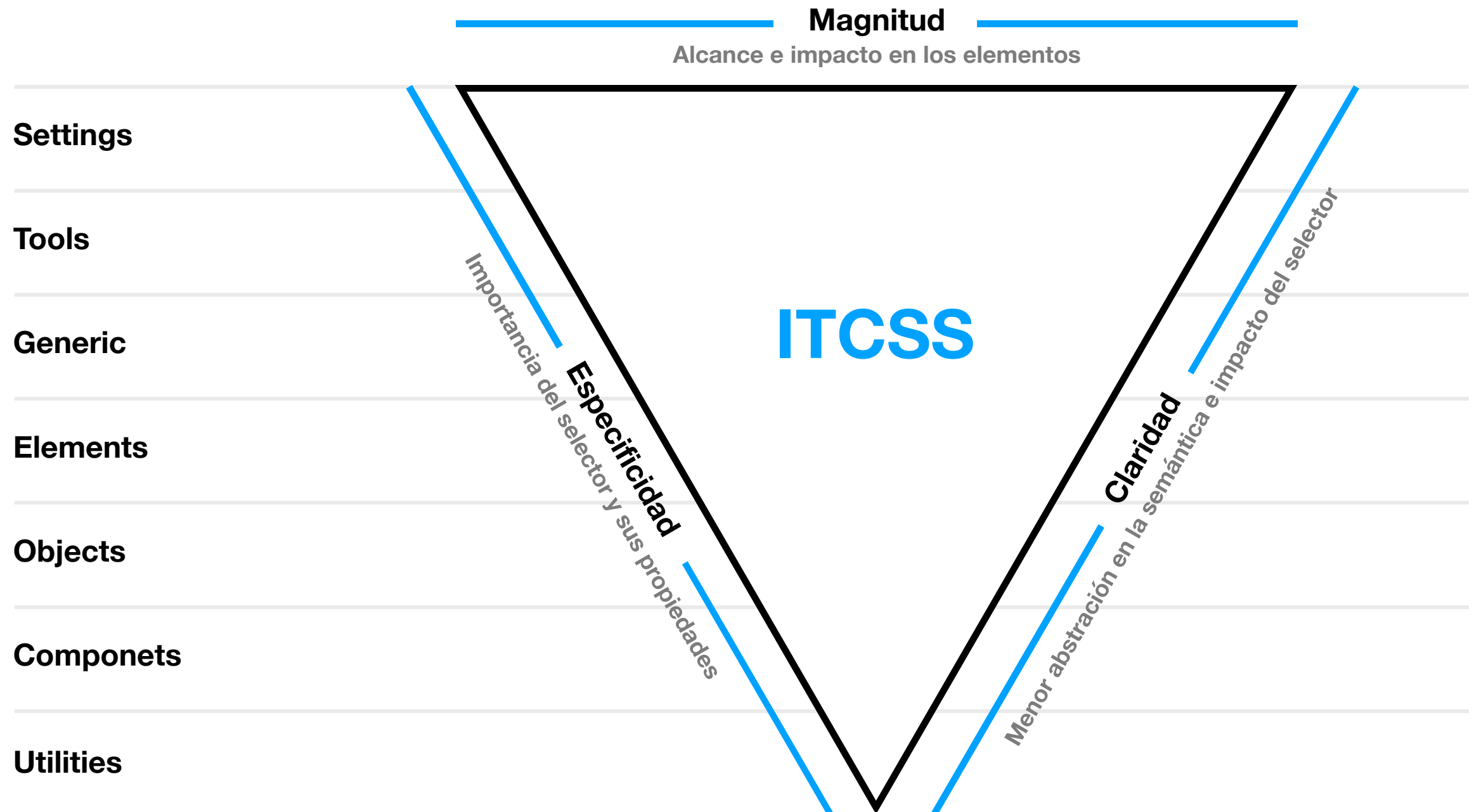
¿Qué es ITCSS?



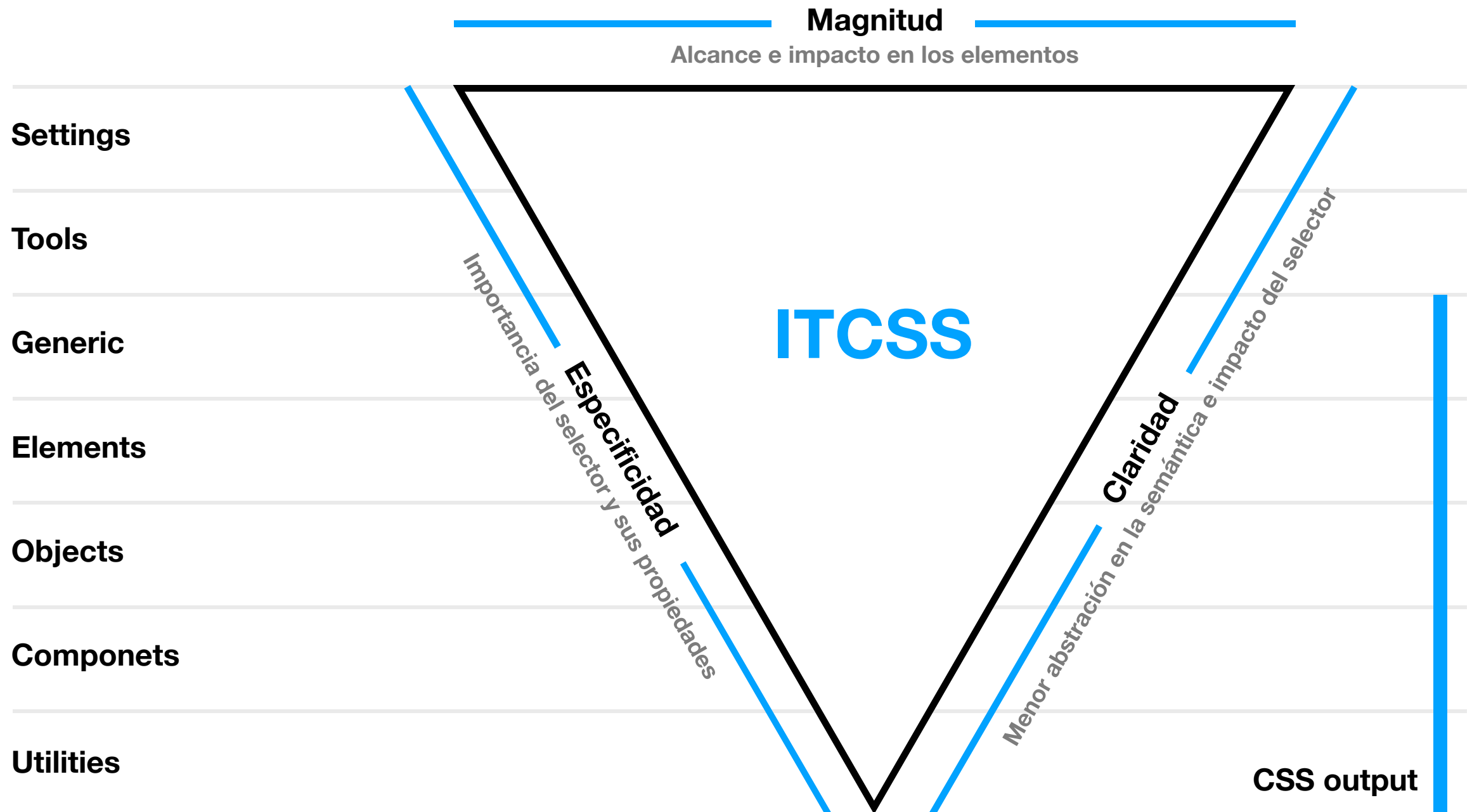
¿Qué es ITCSS?



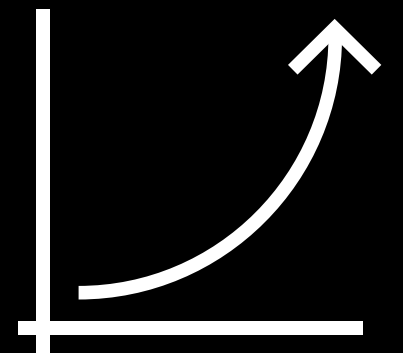
¿Qué es ITCSS?



¿Qué es ITCSS?



Especificidad CSS



La especificidad de los selectores CSS

¿Cómo decide el navegador qué estilos debe aplicar a un determinado elemento de la página?

0 | 0 | 0 | 1

```
element { }
```

```
::pseudo-element { }
```

0 | 0 | 1 | 0

`.class { }`

`[attribute] { }`

`:pseudo-class { }`

0 | 1 | 0 | 0

```
#identifier { }
```

1 | 0 | 0 | 0

```
property: value !important;
```

```
style="property: value;"
```


0 | 0 | 0 | 0

>

+

*

~

@media

Calculando la especificidad

0010 < 00012

Calculando la especificidad

0010 < 00012



0 | 0 | 1 | 0 > 0 | 0 | 0 | 12

Selectores anidados

¡No podemos predecir el futuro!

```
.section article .header h1 { }
```

```
.navigation .dropdown ul li { }
```

```
.form form .form-group input { }
```

Selectores anidados

Intenta evitarlos siempre que sea posible :)

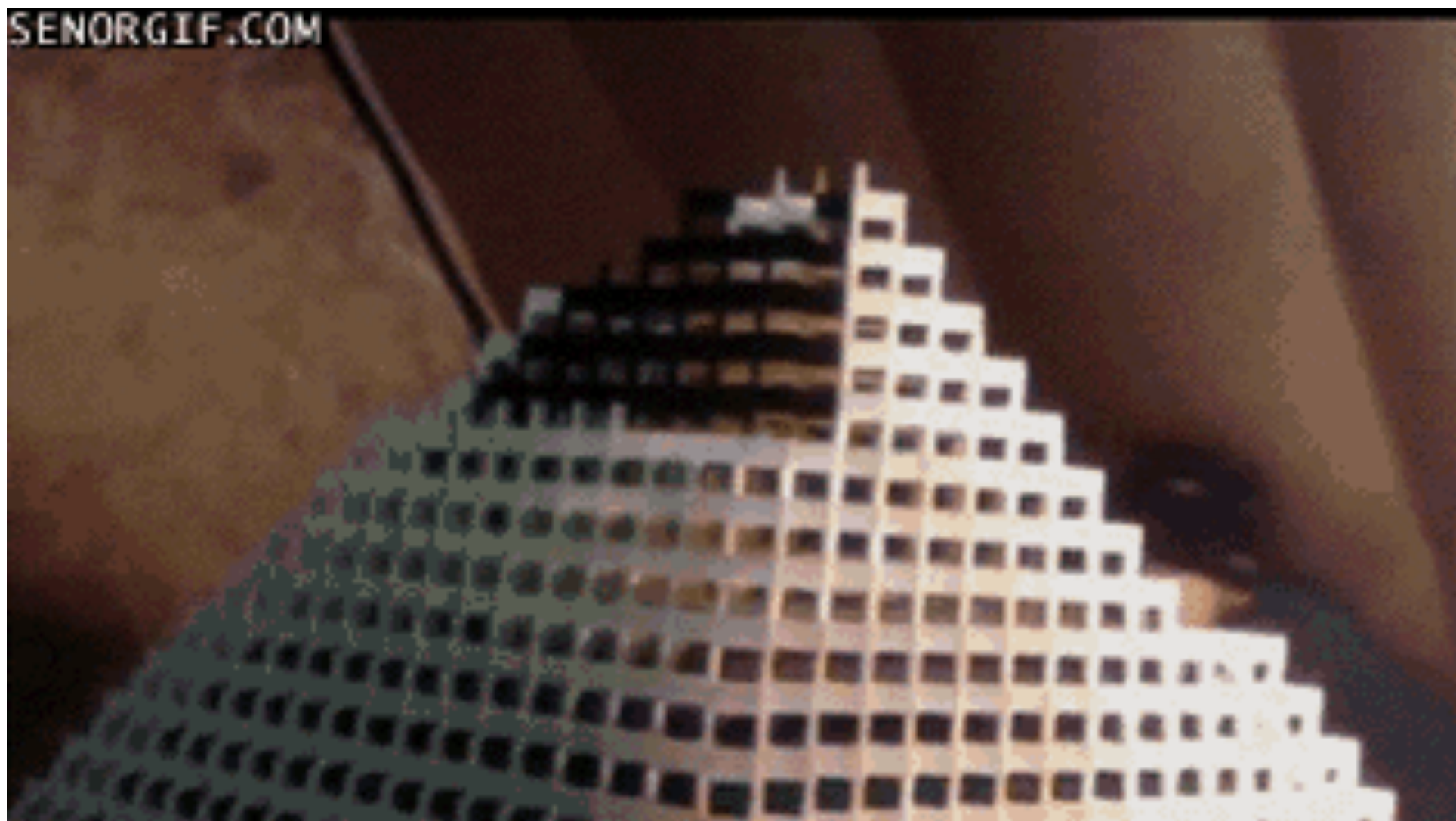


Gráfico de especificidad



Gráfico de especificidad



Gráfico de especificidad

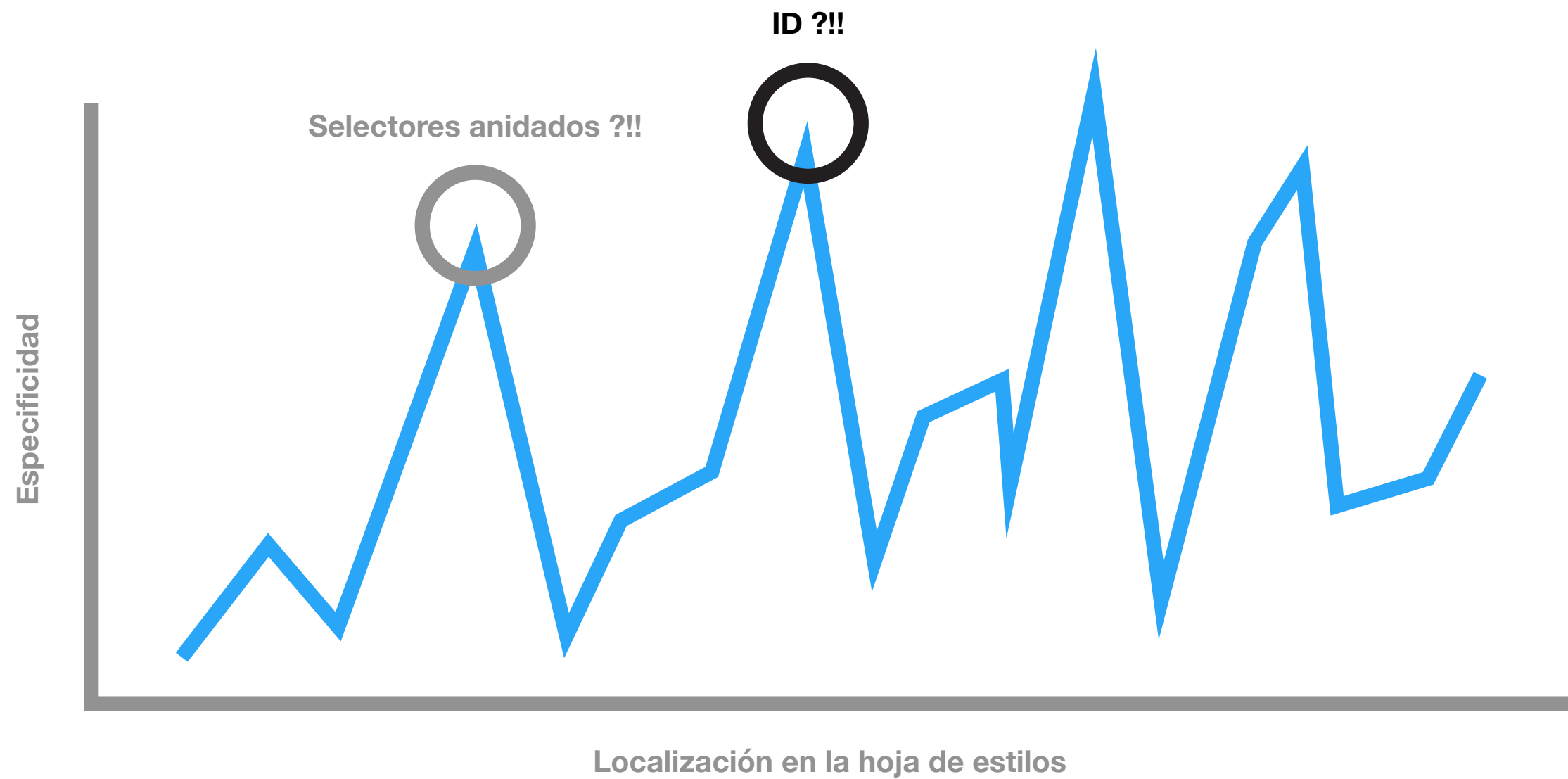


Gráfico de especificidad

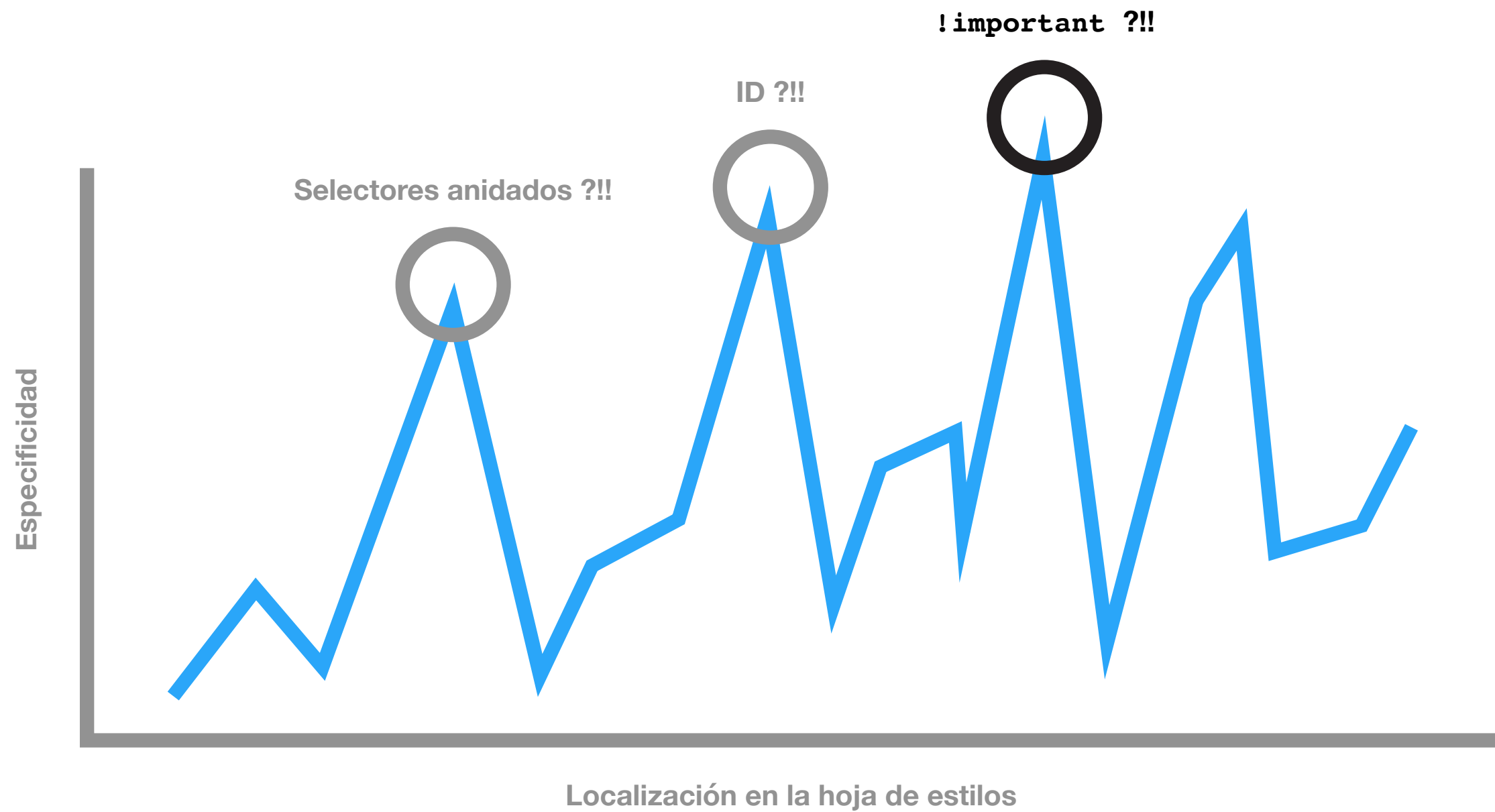
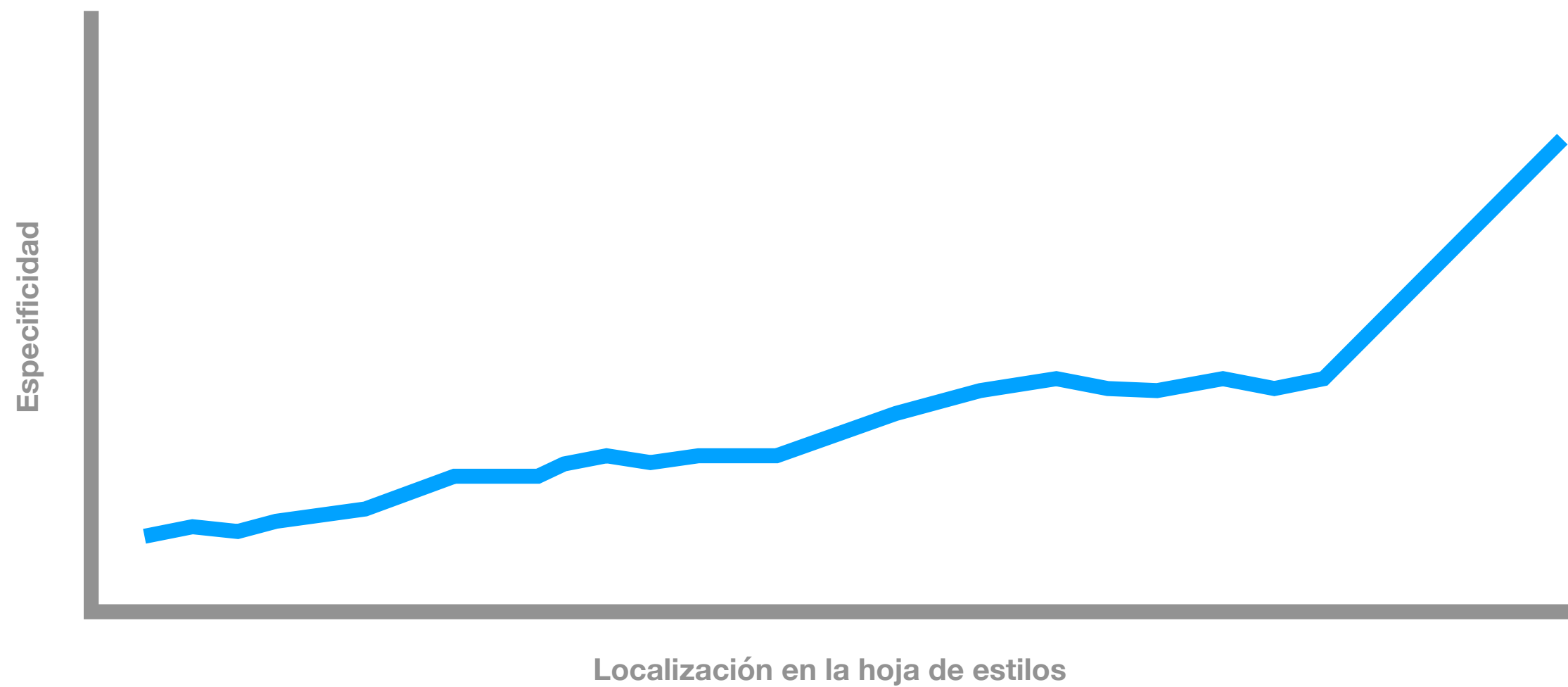
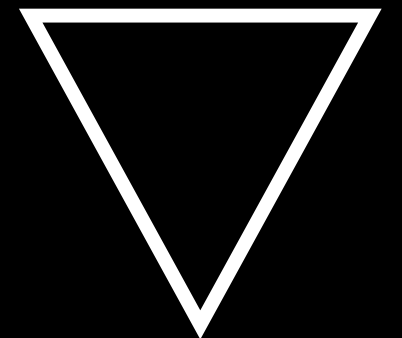


Gráfico de especificidad ITCSS



Capas del triángulo invertido



01 - Settings

(no genera CSS)

- Configuración de la arquitectura
- Variables globales
- Definición de fuentes, colores, ...
- Breakpoints

02 - Tools

(no genera CSS)

- Funciones globales
Conversión unidades, ...
- Mixins globales
Media Queries
Font sizes, transitions, ...

03 - Generic

- Normalización de estilos para navegadores (Normalize)
- Modelo de caja (box-sizing)
- Espaciado de elementos genéricos

04 - Elements

- Apariencia de los elementos HTML
(headings, listas, enlaces, ...)

05 - Objets

- Patrones repetitivos, simples y reusables
- Heredables por los componentes
- Propiedades asociadas a la distribución (contenedor, retícula, secciones)

06 - Components

- Componentes visuales reusables
- Semántica asociada a su apariencia, no a su función ni contenido
- Generalmente, es la capa que más CSS va a generar

07 - Utilities

!important no es una opción

- Clases de ayuda diseñadas para las excepciones
- Sobrescribe los atributos de las capas superiores (espaciado, color alternativo, modificaciones dependiendo del viewport, ...)

Niveles de especificidad en las capas

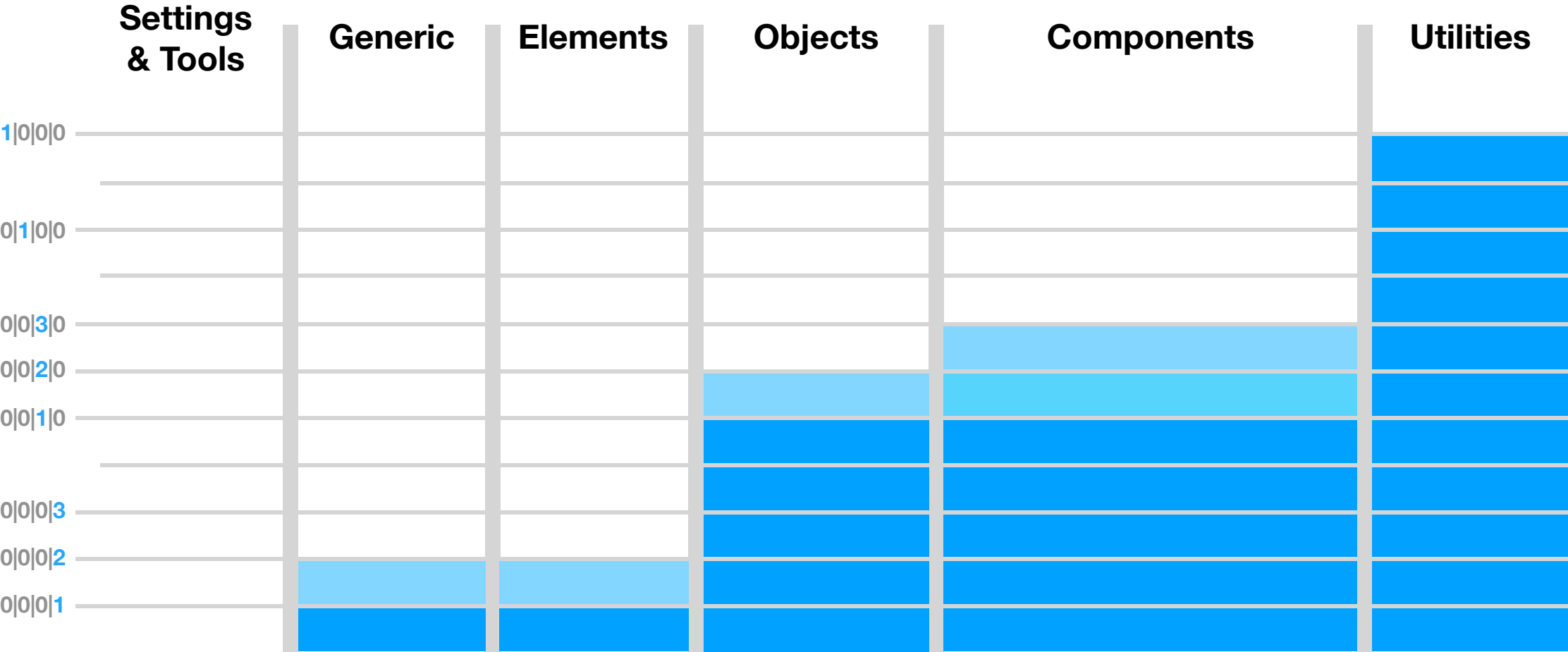


Gráfico de especificidad Amazon España

<https://www.amazon.es/>



Gráfico de especificidad con ITCSS Zinkcare

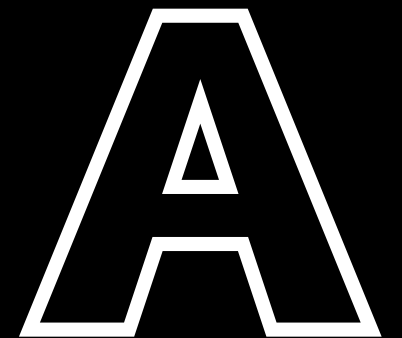
<https://www.zinkcare.com/>



Capas del
triángulo
invertido



Nomenclatura ITCSS+BEM



BEM

Block Element Modifier

```
.block { }
```

```
.block__element { }
```

```
.block--modifier { }
```

ITCSS + BEM

BEMIT

BEMIT namespaces

Se añadirá como prefijo la inicial del
nombre de la capa

```
.c-block { }
```

```
.c-block__element { }
```

```
.c-block--modifier { }
```

BEMIT namespaces

Los namespaces también se extienden
a variables, functions y mixins

```
$s-var: true ! default;
```

```
@function t-rem($_rem__unit) { }
```

```
@mixin t-mq($_mq__name) { }
```

BEMIT namespaces

```
$c-header__width: 100% !default;
```

```
$c-header__spacing: $s-global__spacing !default;
```

```
.c-header {  
  width: $c-header__width;  
  padding: 0 $c-header__spacing;  
}
```


BEMIT sufijos

Se utiliza para definir un estilo
dependiendo del viewport
@[breakpoint]

<code>.o-layout\@sm { }</code>	<code>class="o-layout@sm"</code>
<code>.u-1/4\@sm { }</code>	<code>class="u-1/4@sm"</code>
<code>.u-hidden\@print { }</code>	<code>class="u-hidden@print"</code>

Clases de estado

Se utilizarán en caso de que un componente cambie su estado

`.is-[name]`

`.has-[name]`

Maquetación con BEMIT

```
<section class="o-section o-section--tiny">

  <article class="c-post">

    <header class="c-post__header">

      <h1 class="c-post__title">Minions ipsum tank yuuu!</h1>

    </header>

    <p>Do <span class="u-hidden@sm">potatoooo</span> bappleees</p>

    <a class="c-post__button c-button">Bananaaaa!</a>

  </article>

</section>
```

Metodología de nomenclatura

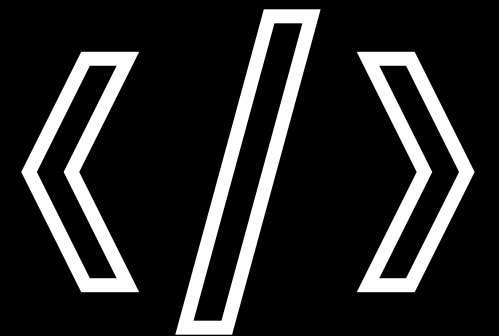
“Utilizar una metodología para nombrar las clases va a hacer que nuestro código sea mucho más fácil de leer y nos va a ayudar a trabajar de una manera mas eficiente.”

Metodología de nomenclatura

Tus compañeros de equipo lo agradecerán



¿Por dónde empezar?



¿Por dónde
empezar?

Estructuración ITCSS de las hojas de estilo

¿Por dónde
empezar?

Estructuración ITCSS

src/

01_settings/

- settings.scss
- 01_settings__config.scss
- 02_settings__variables.scss
- 03_settings__breakpoints.scss
- 04_settings__colors.scss
- 05_settings__fonts.scss

02_tools/

- tools.scss
- 01_tools__mixins.scss
- 02_tools__media-queries.scss

03_generic/

- generic.scss
- 01_generic__normalize.scss
- 02_generic__box-sizing.scss
- 03_generic__global.scss
- 04_generic__print.scss

04_elements/

- elements.scss
- 01_elements__headings.scss
- 02_elements__paragraphs.scss
- 03_elements__anchors.scss
- 04_elements__images.scss

05_objects/

- objects.scss
- 01_objects__container.scss
- 02_objects__grid.scss
- 03_objects__section.scss
- 04_objects__table.scss

06_components/

- components.scss
- 01_components__button.scss
- 02_components__input.scss
- structure/
 - 01_components__header.scss
 - 02_components__footer.scss

07_utilities/

- utilities.scss
- 01_utilities__widths.scss
- 02_utilities__colors.scss
- 03_utilities__trumps.scss
- 02_utilities__debug.scss

style.scss

¿Por dónde
empezar?

Estructuración ITCSS

```
// Settings
@import "01_settings/settings";

// Tools
@import "02_tools/tools";

// Generic
@import "03_generic/generic";

// Elements
@import "04_elements/elements";

// Objects
@import "05_objects/objects";

// Components
@import "06_components/components";

// Utilities
@import "07_utilities/utilities";
```

¿Por dónde
empezar?

Harry Roberts



inuitcss

<https://github.com/inuitcss/inuitcss>

¿Por dónde
empezar?



HaitiCSS

<https://github.com/haiticss/haiticss>

(Daniel Fornells)



iotaCSS

<https://github.com/iotacss/iotacss>

(Dimitris Psaropoulos)

¿Por dónde
empezar?

Refactorizando que es gerundio

Refactoriación

Adopción gradual de nuestra arquitectura

```
@import "01_settings/settings";  
@import "02_tools/tools";  
  
.s-refactor {  
  @import "03_generic/generic";  
  @import "04_elements/elements";  
  @import "05_objects/objects";  
  @import "06_components/components";  
  @import "07_utilities/utilities";  
}
```

Refactoriación

Adopción gradual de nuestra arquitectura

```
<section class="container">

  <ul id="listadoDeProductos">

    <li class="list-item">

      <img class="img_max_width" src="" />

      <a style="text-decoration: none !important;" href="">Bananaaaa!</a>

    </li>

  </ul>

</section>
```

Refactoriación

Adopción gradual de nuestra arquitectura

```
<section class="container">

  <div class="s-refactor">

    <ul class="c-product-list">

      <li class="c-product-list__item">

        <img class="c-product-list__image u-block@md" src="" />

        <a class="c-product-list__link" href="">Bananaaaa!</a>

      </li>

    </ul>

  </div> <!-- / ITCSS Refactor -->

</section>
```

Agradecimientos



Daniel Fornells

@danifornells



HaitiCSS

<https://github.com/haiticss/haiticss>

Referencias

Specificity calculator

<https://specificity.keegan.st/>

CSS Stats

<http://cssstats.com/>

Specificity graph

<https://jonassebastianohlsson.com/specificity-graph/>

CSS Wizardry (Harry Roberts)

<https://csswizardry.com/>

CSS Performance test by @benfrain

<https://github.com/benfrain/css-performance-tests>



¡Muchas gracias!

Frontendly greetings!!!

