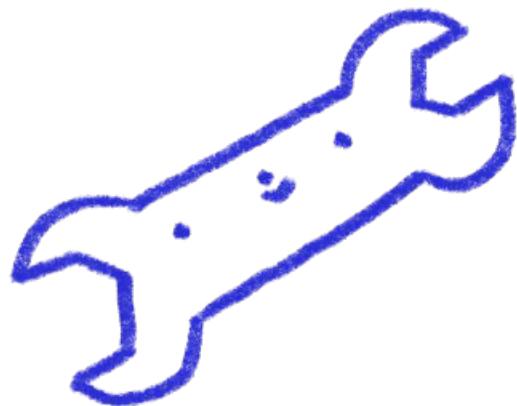


ARQUITECTURAS

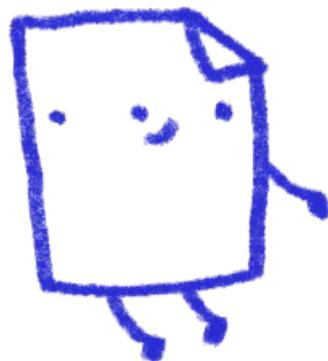
CSS





@TEFFCYDE

CSS TIENE UNA
SINTAXIS MUY SIMPLE



V ESTO PUEDE SER UN
ARMA DE DOBLE FILO

NOS ENFRENTAMOS A
PROBLEMAS DIVERSOS

¡OH, NO!
ME METÍ
EN PROBLEMAS
OTRA VEZ

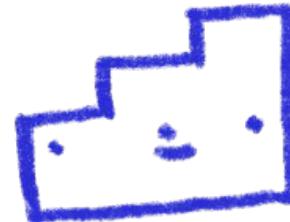


NECESITAMOS HACER
QUE EL CSS SEA:

LIMPIO



ESCALABLE



MANTENIBLE



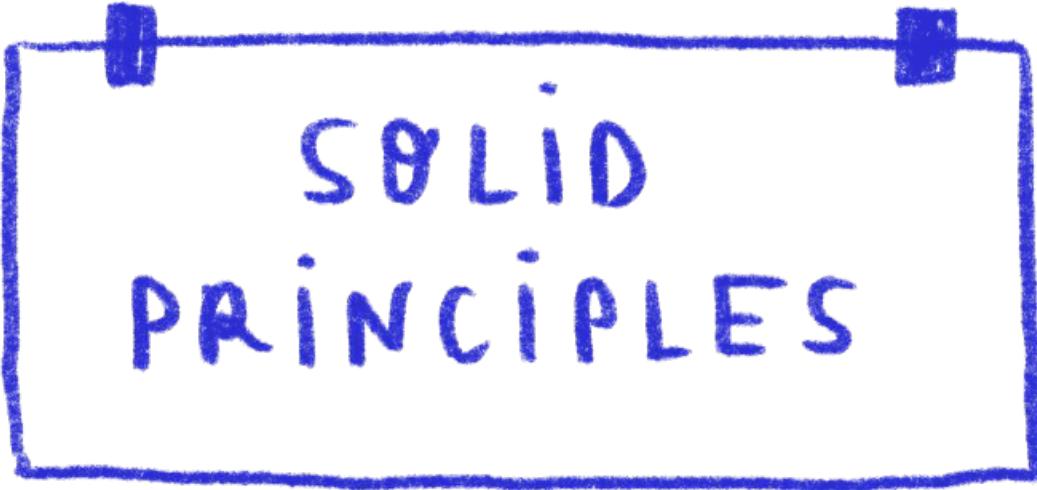
PODEMOS TOMAR EN
CUENTA LOS PRINCIPIOS
DE LA PROGRAMACIÓN
ORIENTADA A OBJETOS

SOLID
DRY
IMMUTABILITY

ARQUITECTURA

CÓMO SE DIVIDE EL
CÓDIGO EN ARCHIVOS
Y CARPETAS.

Y CÓMO TRABAJAMOS CON
NUESTROS SELECTORES
(NOMBREAMIENTO DE CLASES).



SOLID
PRINCIPLES

SOLID

SINGLE RESPONSABILITY
PRINCIPLE

"UNA CLASE DEBE TENER UNA
SOLA RAZÓN PARA CAMBIAR".

Robert C. Martin



```
.button {  
    border-radius: 2rem;  
}  
  
.button--secondary {  
    background-color: gray;  
    color: black;  
}
```



```
<button class="button button--secondary"></button>
```

SOLID

OPEN-CLOSED PRINCIPLE

"LAS ENTIDADES DE SOFTWARE
DEBEN ESTAR ABIERTAS PARA
EXTENSIÓN, PERO CERRADAS
PARA MODIFICACIÓN".

No



```
.button {  
    border-radius: 2rem;  
}  
  
.button--secondary {  
    @extend .button--sm;  
    background-color: gray;  
    color: black;  
}
```



 solidio

LISKOV SUBSTITUTION

“UN OBJETO PUEDE SER
REEMPLAZADO POR UN
SUBOBJETO SIN ROMPER
EL PROGRAMA”.

Barbara Liskov

```
● ● ●  
.button {  
    border-radius: 2rem;  
}
```



```
.button--secondary {  
    @extend .button;  
    background-color: gray;  
    color: black;  
}
```



```
<button class="button--secondary"></button>
```

SOLID

INTERFACE SEGREGATION

NØ SE DEBE OBLIGAR
A NINGÙN CÓDIGO A
DEPENDER DE MÉTODOS
QUE NØ UTILIZA.



```
→ .button {  
    border-radius: 2rem;  
}  
  
→ .button--secondary {  
    @extend .button;  
    background-color: gray;  
    color: black;  
    border-radius: 1.4rem;  
}
```

SOLID DEPENDENCY INVERSION

MÓDULOS DE ALTO NIVEL
NO DEBEN DEPENDER DE
MÓDULOS DE BAJO NIVEL.
AMBOS DEBERÍAN DEPENDER
DE ABSTRACCIONES.

Nº



```
.card div .title {  
    font-size: 2rem;  
}
```



```
.text-md {  
    font-size: 2rem;  
}
```

DRY

DON'T REPEAT YOURSELF

REDUCIR LA
REPETICIÓN DE LA
INFORMACIÓN

No

```
● ○ ●  
.card {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  border: 1px solid gray;  
  width: 100%;  
  height: 20rem;  
}  
  
.card div {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  padding: 2rem;  
}
```



```
● ○ ●  
.center {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}  
  
.card {  
  @extend .center;  
  border: 1px solid gray;  
  width: 100%;  
  height: 20rem;  
}  
  
.card div {  
  @extend .center;  
  padding: 2rem;  
}
```

IMMUTABILITY

OBJETO CUYO ESTADO
NO PUEDE SER MODIFICADO
UNA VEZ CREADO



```
.card p {  
    color: black;  
}  
  
.gray {  
    color: gray !important;  
}
```



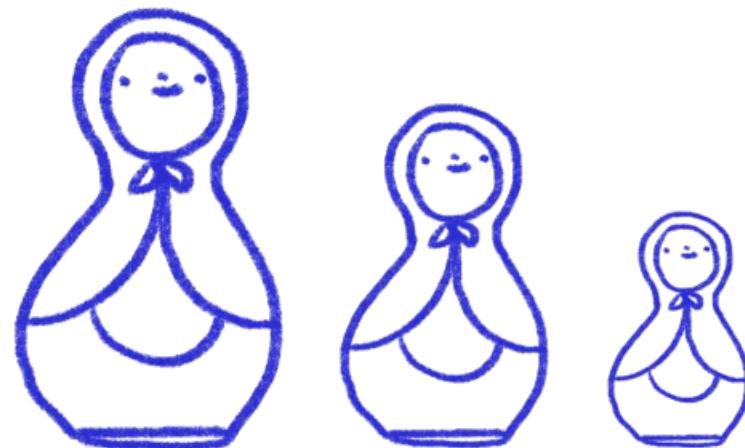
```
<div class="card">  
    <p class="gray"></p>  
</div>
```

LO QUE VEREMOS
EN ESTE CURSO:

PROYECTOS REALES



METODOLOGIA
BEM



B → BLOQUE

E → ELEMENTO

M → MODIFICADOR

- CÓMO FUNCIONA
- CÓMO USARLO
- POR QUÉ Y PARA QUÉ
- EJEMPLOS PRÁCTICOS
- ¿QUÉ HACER EN ESTOS CASOS?
- PROBLEMAS COMUNES
- BEM + SASS
- RECOMENDACIONES



HOLA!

SOY EL COHETE
REUTILIZABLE
DISEÑADO Y
FABRICADO
POR SPACEX:

FALCON 9

Y HOY, JUNTO
CON PLATZI,
APRENDEREMOS
Sobre UNA DE
LAS METODOLÓGÍAS
MÁS USADAS EN CSS:

BEM

¡COMENZEMOS!

¿ CÓMO FUNCIONA BEM ?

BLOQUE

CONTENEDOR PRINCIPAL

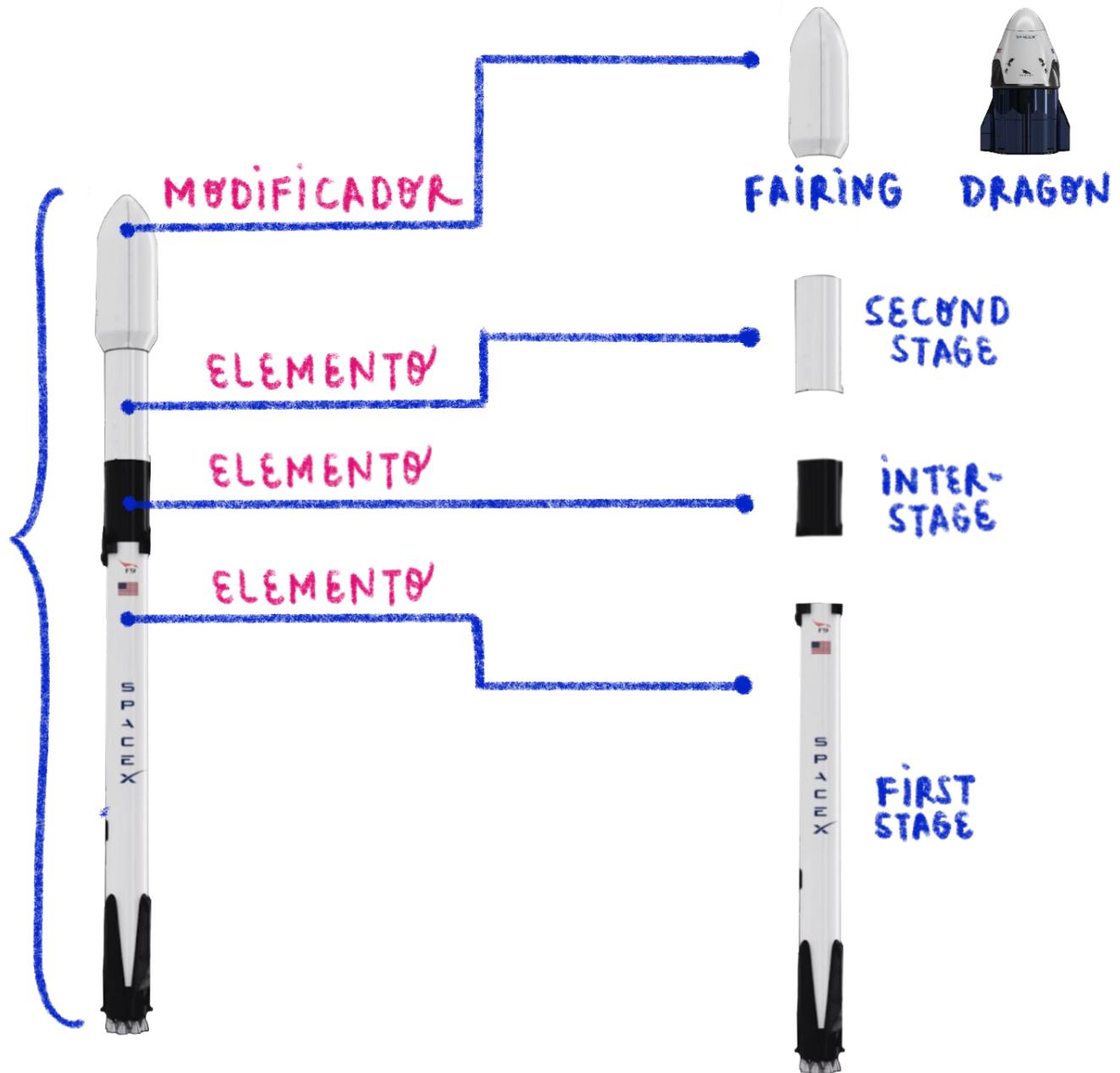
ELEMENTO

PARTES INTERNAS

MODIFICADOR

VARIACIONES

FALCON 9
BLOCK



BLØBLUE

CARD, BUTTON,
FORM, MENU,
HEADER...

ELEMENTS

ICON, TEXT,
ITEM, IMAGE,
INPUT, BUTTON...

MODIFICADOR

ACTIVE, BIG,
RIGHT, SECONDARY,
RED...

¡CÓMO USAR BEM EN CSS?



- [bloque]
- [bloque]__[elemento]
- [bloque]--[modificador]
- [elemento]--[modificador]
- [bloque]__[elemento]--[modificador]

EJEMPLO CON FALCON 9



- .falcon9 { ... }
- .falcon9--fairing { ... }
- .falcon9--dragon { ... }
- .falcon9__first-stage { ... }
- .falcon9__interstage { ... }
- .falcon9__second-stage { ... }

EJEMPLO CON FALCON9



```
<rocket class="falcon9 falcon9--dragon">
  <stage class="falcon9__first-stage"></stage>
  <stage class="falcon9__interstage"></stage>
  <stage class="falcon9__second-stage">
</stage>
```

EJEMPLO DE LA VIDA REAL



- .button { ... }
- .button--active { ... }
- .button__icon { ... }
- .button__text { ... }

EJEMPLO DE LA VIDA REAL



```
<button class="button button--active">  
  <i class="button__icon"></i>  
  <span class="button__text"></span>  
</button>
```

GUIONES BAJOS --
SEPARAN EL
BLOQUE DEL ELEMENTO.

GUIONES MEDIOS --
SEPARAN EL
BLOQUE O ELEMENTO
DEL MODIFICADOR.

NOTA: BEM PERMITE
CAMBIAR ESTA
NOMENCLATURA

[bloque]__[elemento]-[modificador]

LO MÁS IMPORTANTE ES
SER CONSISTENTE EN
TODO EL PROYECTO.

¿POR QUÉ Y PARA QUÉ USAR GEM?

PERO, TEFF, ¿PARA QUÉ
SEGUIR ESTA METODOLÓGIA
SI EL RESULTADO EN CSS
ES EL MISMO? ¡SÓLO
ESTAMOS HACIENDO MÁS
TRABAJO!

10

PARA TENER UN CSS
MÁS FÁCIL DE LEER,
ENTENDER, MANTENER
Y ESCALAR.

20

PARA ORGANIZAR LAS
CLASES DE CSS EN
MÓDULOS INDEPEN-
DIENTES.

3.

PARA EVITAR
SELECTORES
ANIDADOS

CASOS PRACTICOS

ELEMENT[®]

BLOQUE



KID'S SPACEX SPACESUIT ONESIE

\$45.00



ELEMENT[®]

KID'S SPACEX SPACESUIT ONESIE

\$45.00

ELEMENT[®]

BLØRUE : ?

ELEMENTOS : ?

MØDIFICADØRES : ?

BLOQUE: CARD

ELEMENTOS: IMAGE, TEXT

MODIFICADORES: (NO TIENE)

BLØBUE: CARD

ELEMENTOS: IMAGE, TEXT

MØDIFICADØRES: (NØ TIENE)



```
<div class="card">
  
  <p class="card__text"></p>
  <p class="card__text"></p>
</div>
```

BLOQUE



SPACEX

MENS

WOMENS

KIDS

ACCESSORIES

PREMIUM

ACCOUNT

SEARCH

CART (0)

SPACEX

KIDS

PREMIUM

ACCOUNT

ELEMENTOS

MODIFICADORES

BLØRUE : ?

ELEMENTOS : ?

MØDIFICADØRES : ?

BLOQUE : NAVBAR

ELEMENTOS : LOGO, ITEMS

MODIFICADORES : GRAY



```
<nav class="navbar">
  <ul class="navbar">
    <li class="navbar__item"><i class="navbar__icon"></i></li>
    <li class="navbar__item">MENS</li>
    <li class="navbar__item">WOMENS</li>
    <li class="navbar__item">KIDS</li>
    <li class="navbar__item">ACCESSORIES</li>
    <li class="navbar__item">PREMIUM</li>
    <li class="navbar__item navbar__item--gray">ACCOUNT</li>
    <li class="navbar__item navbar__item--gray">SEARCH</li>
    <li class="navbar__item">CART (0)</li>
  </ul>
</nav>
```

FALCON 9

MÓDIFICADOR
MÓDIFICADOR

Falcon 9, the world's first orbital class reusable rocket, is manufactured and operated by SpaceX for the reliable and safe transport of people and payloads to Earth orbit and beyond.

[Learn more about Falcon 9 >](#)

FALCON 9

Falcon 9, the world's first orbital class reusable rocket, is manufactured and operated by SpaceX for the reliable and safe transport of people and payloads to Earth orbit and beyond.

[Learn more about Falcon 9 >](#)

BLOQUE



ELEMENTO



ELEMENTO ELEMENTO

BLOQUE

MÓDIFICADOR

ELEMENTO



FALCON 9

MØDIFICADOR

MØDIFICADOR

Falcon 9, the world's first orbital class reusable rocket, is manufactured and operated by SpaceX for the reliable and safe transport of people and payloads to Earth orbit and beyond.

[Learn more about Falcon 9 >](#)

MØDIFICADOR

! BLOQUE

FALCON 9

Falcon 9, the world's first orbital class reusable rocket, is manufactured and operated by SpaceX for the reliable and safe transport of people and payloads to Earth orbit and beyond.

[Learn more about Falcon 9 >](#)

VEHICLE HEIGHT

70m / 229.6ft

VEHICLE DIAMETER

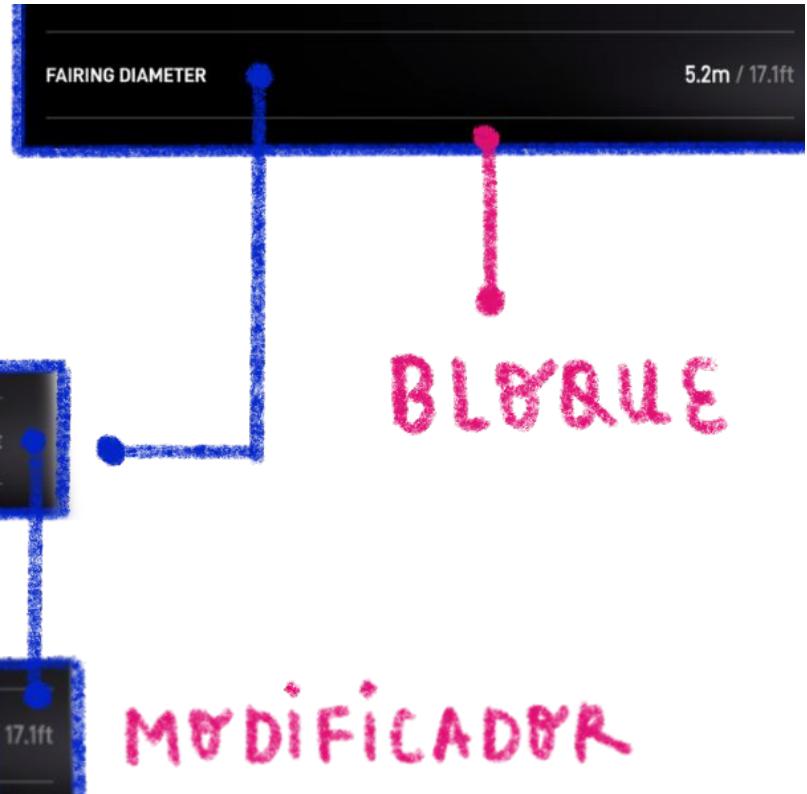
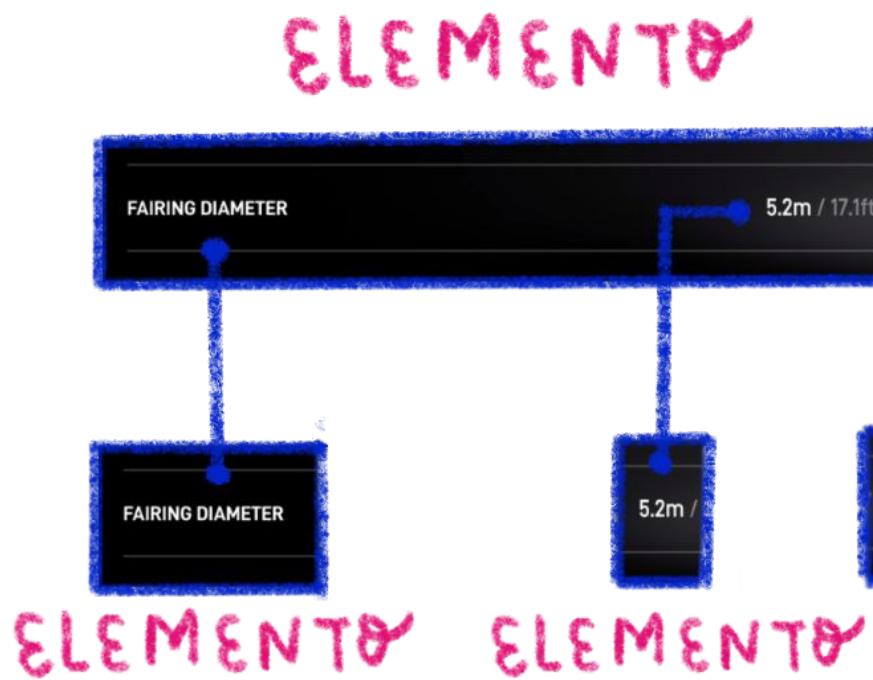
3.7m / 12ft

FAIRING HEIGHT

13.1m / 43ft

FAIRING DIAMETER

5.2m / 17.1ft





ELEMENTS

BLØBUE : ?

ELEMENTOS : ?

MØDIFICAÐPRES : ?

BLØBUE : ?

ELEMENTOS : ?

MØDIFICAÐPRES : ?

BLØRUE : SECTION

ELEMENTOS : TEXT, IMAGE

MØDIFICADØRES : PRIMARY,
SECONDARY- SEMIBOLD,
SECONDARY- BOLD

BLØRUE : LIST

ELEMENTOS : ITEM, TEXT

MØDIFICADØRES : GRAY



```
<section class="section">
  <div>
    <h2 class="section__text section__text--primary">FALCON 9</h2>
    <p class="section__text section__text--secondary-semibold">Falcon 9, the world's...</p>
    <p class="section__text section__text--secondary-bold">Learn more about Falcon 9</p>
    <ul class="list">
      <li class="list__item">
        <span class="list__text">VEHICLE HEIGHT</span>
        <div>
          <span class="list__text">70m</span>
          <span class="list__text list__text--gray">/229.6 ft</span>
        </div>
      </li>
      ...
    </ul>
  </div>
</section>
```

CASOS CON LO
QUE DEBES Y
NO DEBES HACER
CON BEM

CASB 1

LOR RUE SÍ :)

DEJAR LA CLASE PRINCIPAL
CARD Y AÑADIR OTRA CLASE
CON MODIFICADOR



```
<div class="card card--flat"></div>
```

LÓGUE N° 1

USAR LA CLASE CON
MODIFICADOR COMO
CLASE PRINCIPAL



```
<div class="card--flat"></div>
```

CASE 2

LØ RUE SÍ :)

NØ REPRESENTAR LOS
NIVELES DE PROFUNDIDAD
DE HTML CON BEM.



```
<div class="card">
  
  <p class="card__text">
    <i class="card__icon">
      </p>
</div>
```

LÓGUE N° 1

REPRESENTAR LOS NIVELES
DE PROFUNDIDAD DE
HTML CON BEM.



```
<div class="card">
  
  <p class="card__text">
    <i class="card__text__icon">
    </p>
</div>
```

CASE 3

LÓ QUE SÍ :)

INCLUIR LA CLASE
EN UN HIJO QUE
NECESITA ESTILOS



```
<div class="card">
    
</div>

<style>
    .card { ... }
    .card__image { ... }
</style>
```

LÓGUE N° 1

OMITIR LA CLASE
EN UN HIJO QUE
NECESITA ESTILOS



```
<div class="card">
    
</div>

<style>
    .card { ... }
    .card img { ... }
</style>
```

3 PROBLEMAS COMUNES
CON BEM Y CÓMO
SOLUCIONARLOS

CASB 1

PROBLEMA: TENGO UN
COMPONENTE A QUE YA
TIENE SUS PROPIAS CLASES
Y DESEO AÑADIRLO A UN
NUEVO COMPONENTE B.
¿DEBES AGREGAR UNA NUEVA
CONVENCIÓN PARA EL COMPON-
ENTE A QUE ESTÁ DENTRO
DE B?



```
# Componente A:
```

```
<button class="button button--active"></button>
```

```
# Componente B:
```

```
<div class="card">
  
</div>
```

SOLUCIÓN: PUEDES DEJAR
EL COMPONENTE A CON
LAS CLASES QUE YA ESTABAN
ESTABLECIDAS ASÍ NO SEAN
COHERENTES CON EL NUEVO
COMPONENTE B,
POR EJEMPLO:



```
<div class="card">
  
  <button class="button button--active"></button>
</div>
```

CASE 2

PROBLEMA: EN MI ESTRUCTURA
DE HTML TENGO PADRES, HIJOS,
NIETOS, TATARANIETOS, ETC;
PERO BEM SÓLO DEJA USAR
3 NÍVELES. ¿QUÉ HAGO CON
LOS ELEMENTOS NIETOS Y
SUS DESCENDIENTES?



```
<div class="card">
  
  <div class="card__footer">
    <p class="card__footer__text">
      <i class="card__footer__text__icon">
        <p>
      </div>
    </div>
```

SOLUCIÓN: COMO BEM NO
REPRESENTA LOS NIVELES
DE TU ESTRUCTURA DE
HTML. LO IDEAL EN ESTE
CASO SERÍA TENER:



```
<div class="card">
  
  <div class="card__footer">
    <p class="card__text">
      <i class="card__icon">
        <p>
      </div>
    </div>
```

CASE 3

PROBLEMA: QUIERO UTILIZAR LA PROPIEDAD DISPLAY: NONE PARA OCULTAR DESDE JS UN COMPONENTE DE CARD Y, UN COMPONENTE DE BOTÓN. ¿DEBO CREAR UNA CLASE PARA CADA COMPONENTE SIGUIENDO SU PROPIA ESTRUCTURA DE BEM (CARD--HIDDEN Y BUTTON--HIDDEN) ?



```
<div class="card card--hidden">
    
</div>
<button class="button button--hidden"></button>

<style>
    .card--hidden {
        display: none;
    }
    .button--hidden {
        display: none;
    }
</style>
```

SOLUCIÓN: PUEDES CREAR UNA CLASE INDEPENDIENTE A LA ESTRUCTURA DE BEM PARA APLICAR PROPIEDADES GENERALES, ASÍ REDUCIRÁS LA CANTIDAD DE LÍNEAS DE JS YA QUE NO TENDRÁS QUE USAR UNA CLASE ESPECÍFICA PARA CADA COMPONENTE.



```
<div class="card hidden">
  
</div>
<button class="button hidden"></button>

<style>
  .hidden {
    display: none;
  }
</style>
```

BEM + SASS





```
<div class="card">
  
  <button class="button button--active"></button>
</div>
```



```
.card {
  ...
}
.card__image {
  ...
}
.button {
  ...
}
.button--active {
  ...
}
```



```
.card {
  &__image {
    ...
  }
}
.button {
  &--active {
    ...
  }
}
```

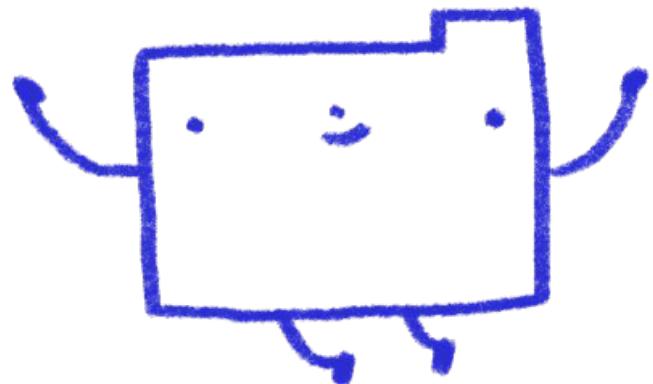
RECOMENDACIONES

→ LOS PROYECTOS QUE USAN BEM SON:

- FÁCILES DE LEER
- MUY INTUITIVOS
- PERMITEN EVITAR SELECTORES ANIDADOS

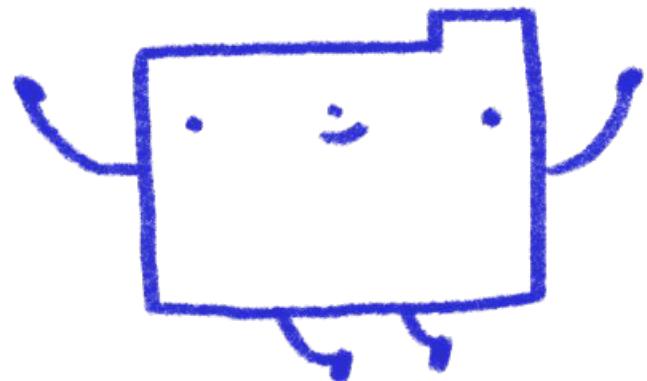
→ BEM PERMITE PERSONALIZAR SUS REGLAS Y NOMENCLATURAS

¡TIEMPO DE PROYECTO!



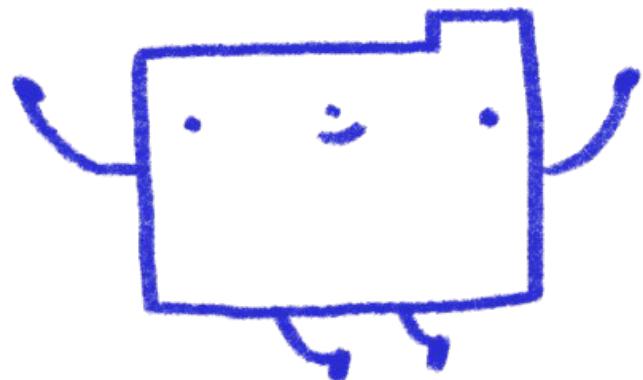
BEM
EN UN PROYECTO DE
HTML, CSS Y JS

¡TIEMPO DE PROYECTO!



BEM
EN UN PROYECTO DE
VUE JS

¡TIEMPO DE PROYECTO!



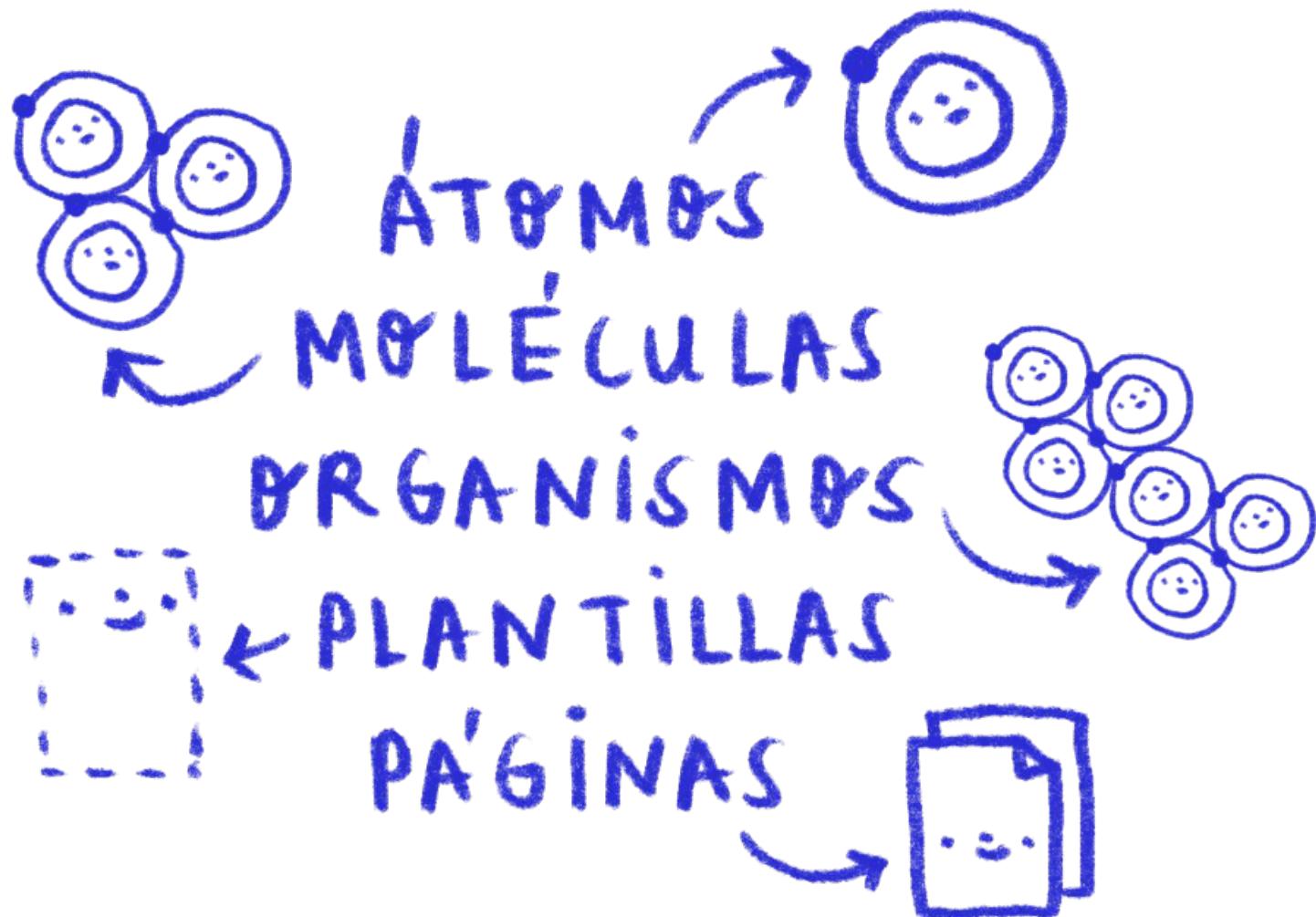
BEM Y STYLED COMPONENTS
EN UN PROYECTO DE
REACT

ATOMIC DESIGN



BRAD FRØST
ENCONTRÓ,
INSPIRACIÓN
EN LA
QUÍMICA





ATÓMOS

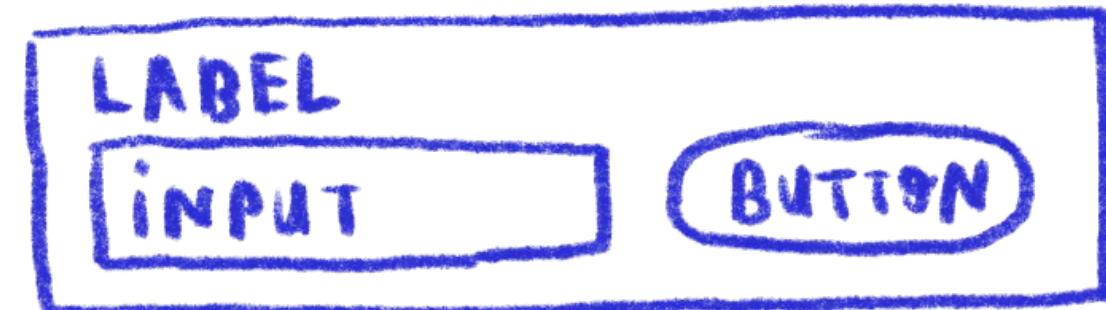
ELEMENTOS DE INTERFAZ
QUE NO SE PUEDEN DIVIDIR
MÁS Y SIRVEN COMO BLOQUES
DE CONSTRUCCIÓN

LABEL

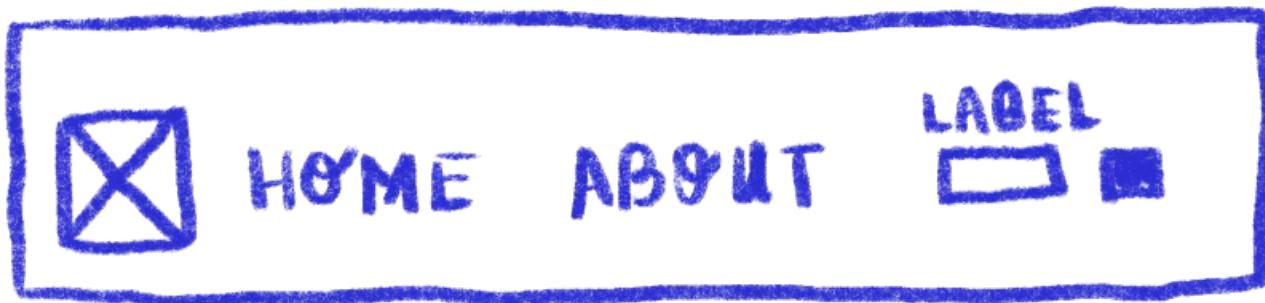
INPUT

BUTTON

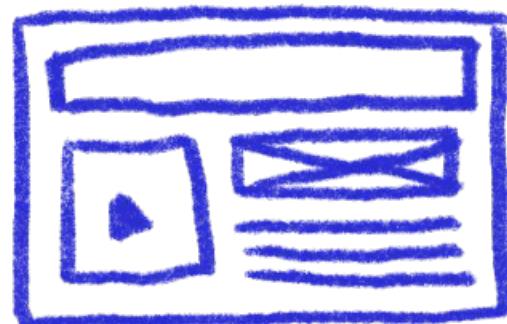
MOLÉCULAS
SON COLECCIONES DE
ÁTOMOS QUE FORMAN
COMPONENTES DE
INTERFAZ DE USUARIO
RELATIVAMENTE SIMPLES



ORGANISMOS
SØN CØMPØNENTES
RELATIVAMENTE CØMPLØJØS
QUE FØRMAN SECCIONES
DISCRETAS DE INTERFAZ



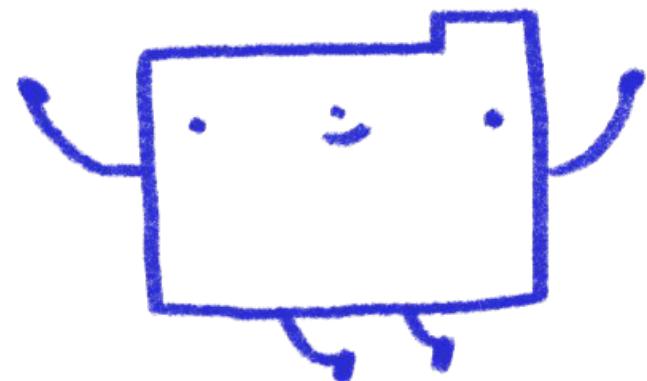
TEMPLATES
COLLOCAN COMPONENTES
DENTRO DE UN DISEÑO Y
DEMUESTRAN LA ESTRUCTURA
DE CONTENIDO SUBYACENTE
AL DISEÑO



PÁGINAS
APLICAN CONTENIDO
REAL A LAS PLANTILLAS

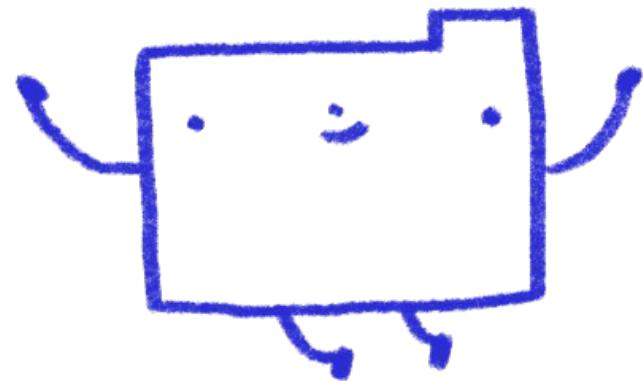


¡TIEMPO DE PROYECTO!



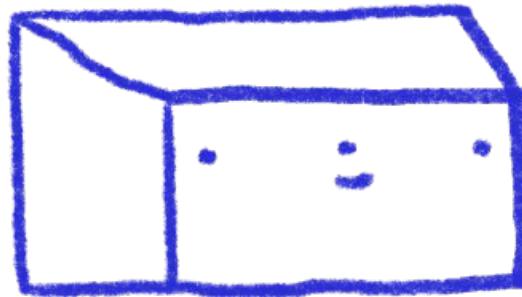
ATOMIC DESIGN
EN UN PROYECTO DE
REACT

J TIEMPO DE PROYECTO!



ATOMIC DESIGN
EN UN PROYECTO DE
VUE JS

METODOLÓGIA
BBCSS



PRINCIPIOS

- SEPARAR LA ESTRUCTURA
DEL DISEÑO
- SEPARAR CONTENEDOR
DE CONTENIDOS.

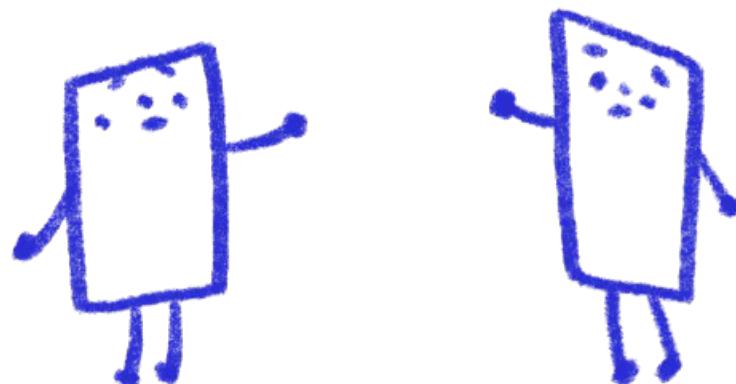
- PROPIEDADES DE ESTRUCTURA:

HEIGHT WIDTH
MARGIN PADDING

- PROPIEDADES DE APARIENCIA:

COLOR FONT

SEPARAR LA
ESTRUCTURA
DEL DISEÑO



SIN CSS



```
.card-gray {  
    width: 10rem;  
    height: 15rem;  
    background-color: gray;  
}
```

```
.card-white {  
    width: 10rem;  
    height: 15rem;  
    background-color: white;  
}
```

SIN BOSS



```
<div class="card-gray"></div>
<div class="card-white"></div>
```

CONCEPTS



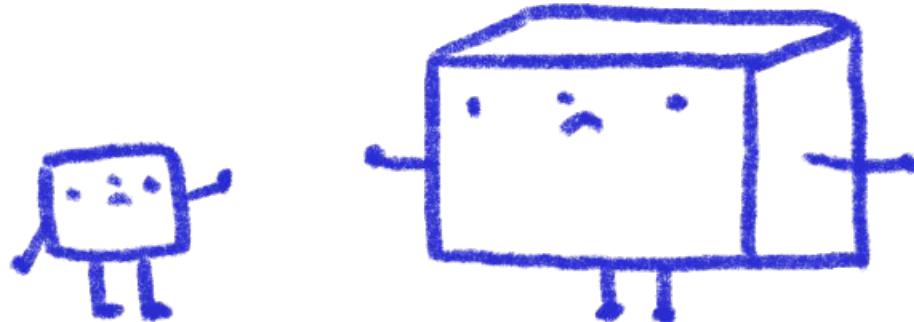
```
.card {  
    width: 10rem;  
    height: 15rem;  
}  
  
.gray {  
    background-color: gray;  
}  
  
.white {  
    background-color: white;  
}
```

CONCEPTS



```
<div class="card gray"></div>
<div class="card white"></div>
```

SEPARAR
CONTENEDOR
DE CONTENIDO



SIN CSS

```
<div>  
  <p>Hol&iacute;</p>  
</div>
```

```
div p {  
  font-size: 1.2rem;  
  color: gray;  
}
```

CONCEPTS



```
.text-sm {  
    font-size: 1.2rem;  
}
```

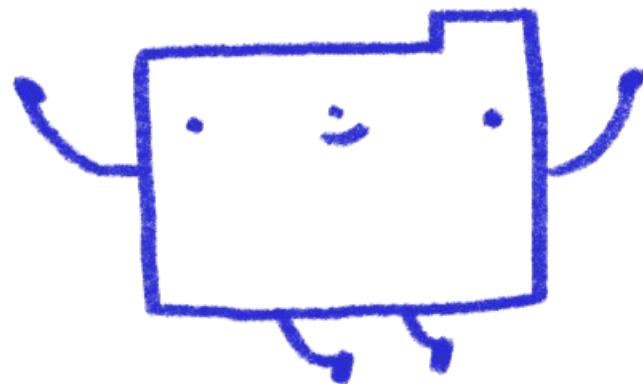
```
.text-gray {  
    color: gray;  
}
```

CON 80 CSS



```
<div>
  <p class="text-sm text-gray">Holi</p>
</div>
```

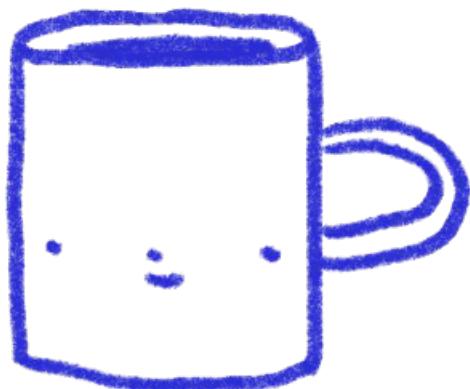
¡TIEMPO DE PROYECTO!



BBCSS Y SASS
EN UN PROYECTO DE
ANGULAR

METODOLOGÍA

SMACSS



S → SCALABLE

M → MODULAR

A → ARCHITECTURE

C → CASCADING

S → STYLE

S → SHEET

CATEGORIAS:

1. BASE
2. LAYOUT
3. MODULES
4. STATE
5. THEME

BASE (BASE)

- ESTILOS PREDETERMINADOS
- SELECTORES DE ELEMENTOS & PSEUDOC - CLASES
 - NØ CLASES
 - NØ ID
- TAMAÑOS, COLORES...



```
html {  
    margin: 0;  
}  
  
a {  
    text-decoration: none;  
}  
  
button {  
    color: gray;  
    border-radius: 2rem;  
}
```

LAYOUT (DISEÑO &
ESTRUCTURA
DE PÁGINA)



ESTILOS PARA LAS
SECCIONES PRINCIPALES

HEADER , FOOTER ...

SE USAN CON ID (& CLASES)



```
#footer {  
    background-color: gray;  
}  
  
#footer .l-center {  
    text-align: center;  
}
```

MODULE (MÓDULO)

FRAGMENTOS ÚNICOS
QUE PUEDE CONTENER:

- PREFIJOS
- ELEMENTOS
SECUNDARIOS



```
.section {  
    background-color: gray;  
}  
  
.section--title {  
    font-size: 1.2rem;  
}
```

STATE (ESTADO)

REPRESENTA
SITUACIONES
DINÁMICAS



is - SELECTED



```
.is-selected {  
    background-color: gray;  
}  
  
.is-visible {  
    visibility: visible;  
}
```

THEME (TEMA)

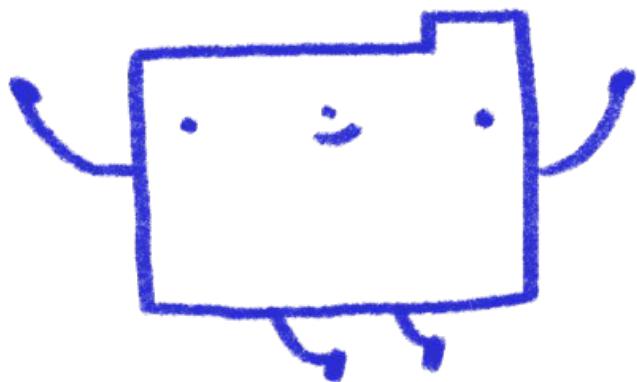
COLORES PRIMARIOS,
FORMAS, BORDES,
SOMBRA S Y DEMÁS.

SE REPITEN EN
TODO EL SÍTIOS.



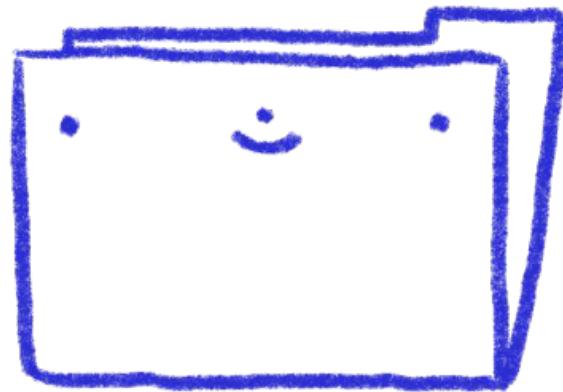
```
.text-xs {  
    font-size: 1.2rem;  
}  
  
.text-sm {  
    font-size: 1.4rem;  
}  
  
.text-md {  
    font-size: 1.6rem;  
}  
  
.text-lg {  
    font-size: 2rem;  
}
```

¡TIEMPO DE PROYECTO!



SMACSS Y SASS
EN UN PROYECTO DE
ANGULAR

ARQUITECTURA iTCSS



INVERTED
TRIANGLE

ITCSS ES UNA COLECCIÓN
DE PRINCIPIOS Y MÉTRICAS

LA IDEA ES VISUALIZAR EL
CSS COMO UN TRIÁNGULO
INVERTIDO EN CAPAS.

- PRINCIPIOS BÁSICOS
- MÉTRICAS CLAVE
- BENEFICIOS
- CAPAS
- RESULTADOS

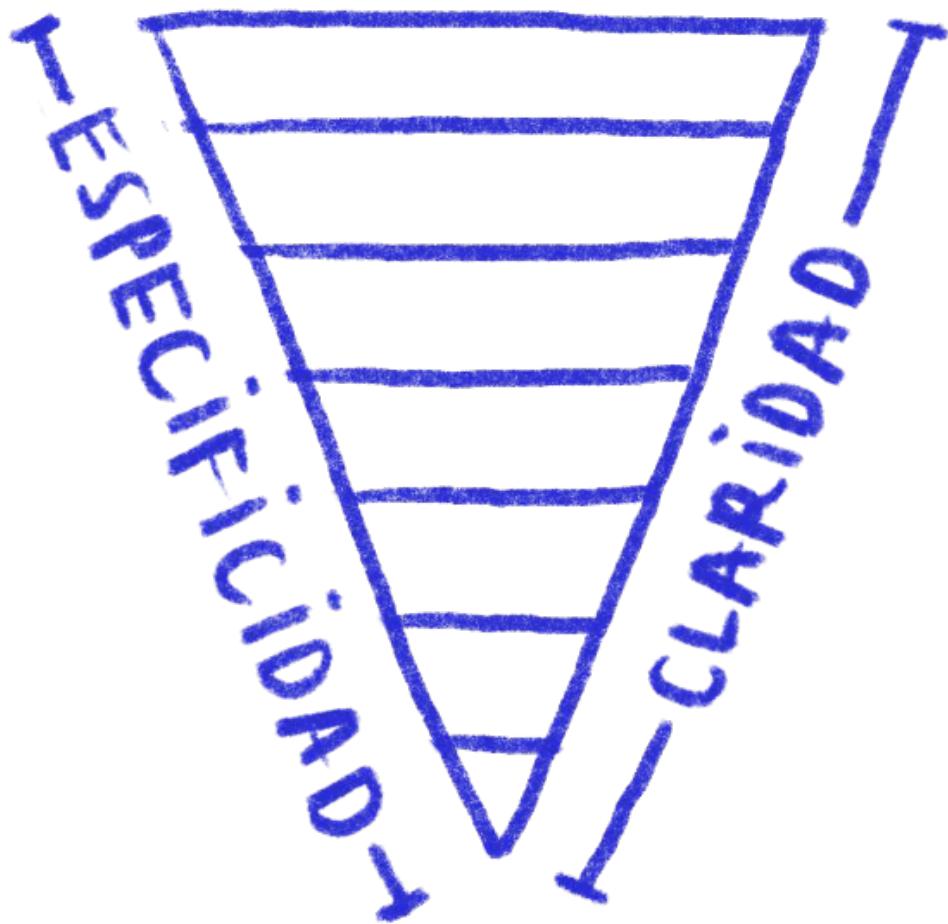
PRINCIPIOS BÁSICOS

1. SIN ID EN CSS
2. PATRÓN COMPONENT UI
3. ARQUITECTURA BASADA EN CLASES.

MÉTRICAS

1. DE ESTILOS GENÉRICOS
A ESTILOS ESPECÍFICOS.
2. DE BAJA ESPECIFICIDAD
A ALTA ESPECIFICIDAD
3. DE LARGO ALCANCE A
LOCALIZADO

MAGNITUD



Magnitud

ALCANCE E IMPACTO
EN CANTIDAD DE ELEMENTOS

Especificidad

FUERZA DEL SELECTOR
Y SUS PROPIEDADES

Claridad

MENSAJE ABSTRACCIÓN EN LA
SEMÁNTICA E IMPACTO DEL SELECTOR

BENEFICIOS

- ESTILOS GLOBALES SE COMPARTEN DE MANERA MÁS EFICAZ Y EFICIENTE
- DISMINUCIÓN DE BATALLAS POR ESPECIFICIDAD
- MENOS REDUNDANCIA

CAPAS

SETTINGS

TOOLS

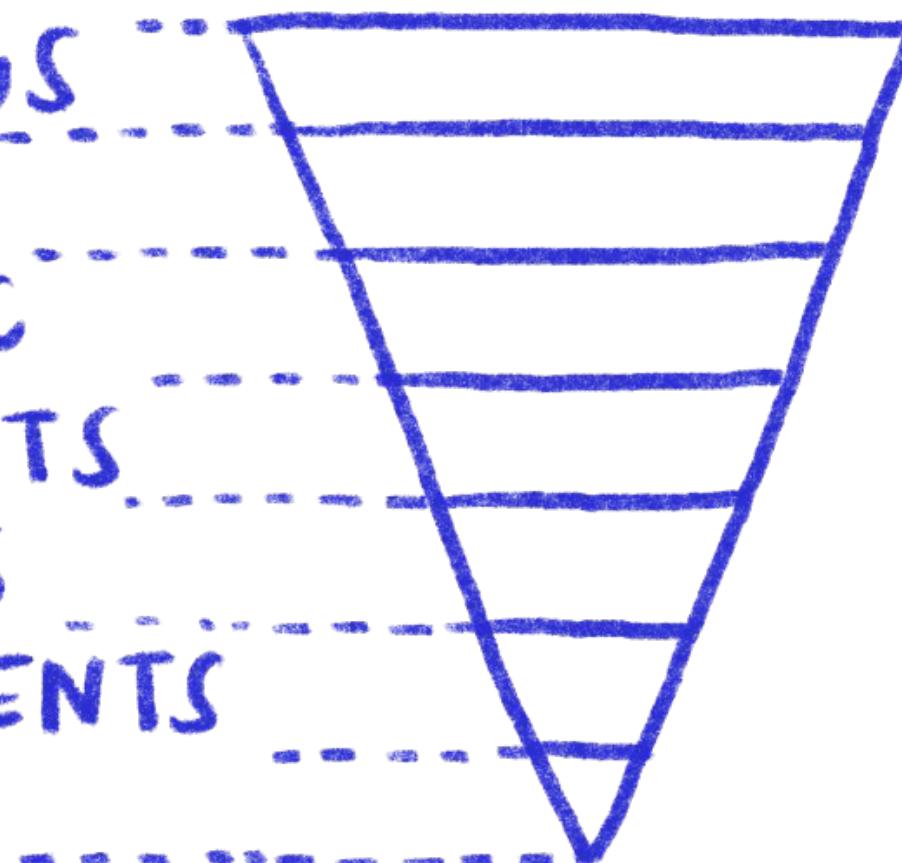
GENERIC

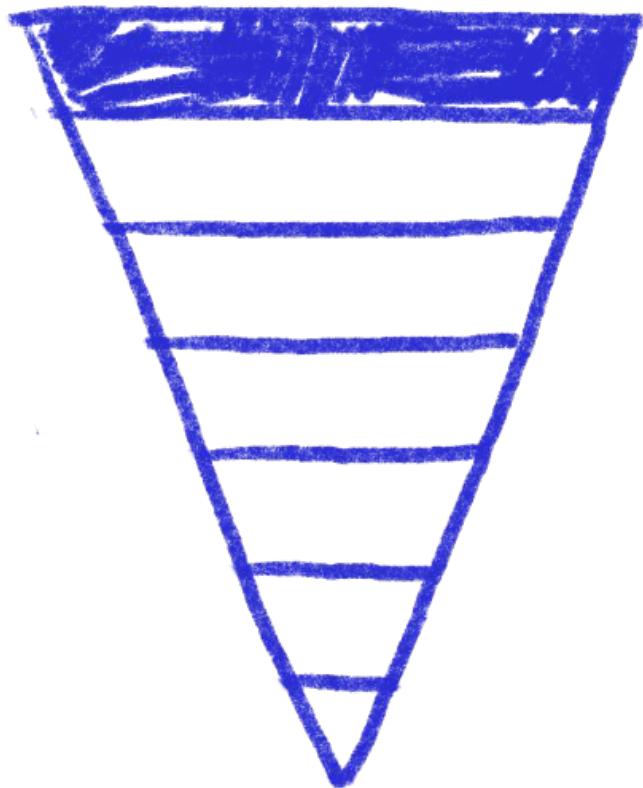
ELEMENTS

OBJECTS

COMPONENTS

TRUMPS





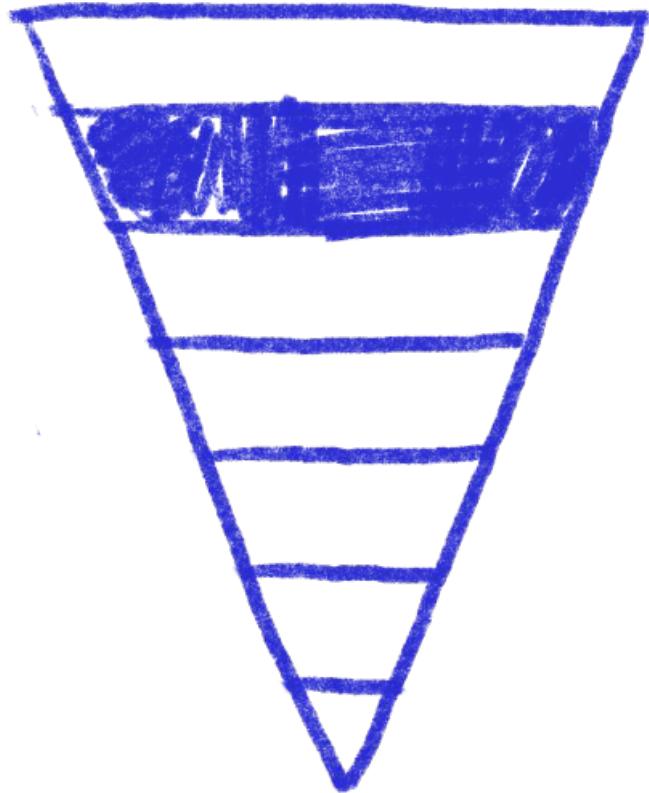
SETTINGS
AJUSTES

VARIABLES
GLOBALES

→ COLORES
→ TAMAÑOS



```
/* main.scss file */
@import "settings.fontsize";
@import "settings.colors";
```



TOOLS
HERRAMIENTAS

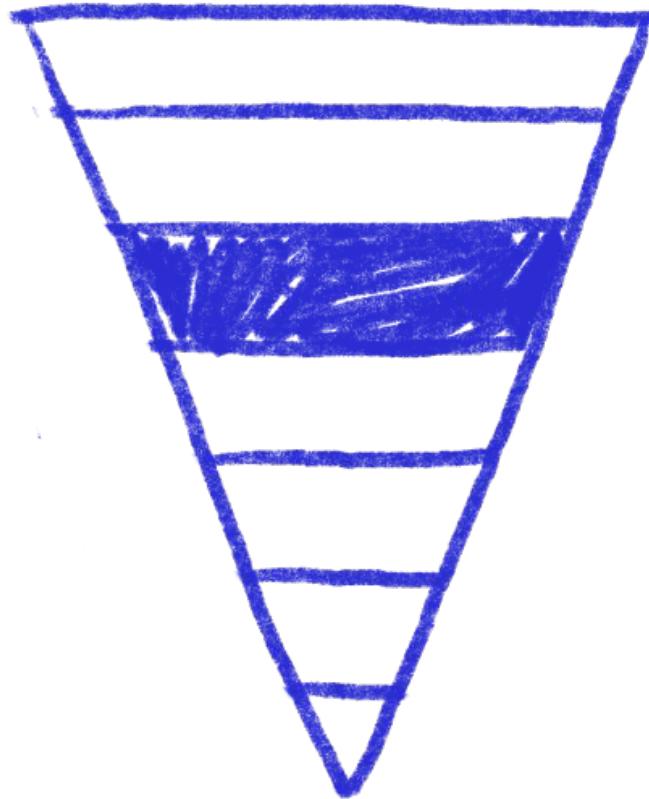
FUNCIONES
GLÓBALES Y
MIXINS



SÓLO SI
SE USAN
PREPROCE-
SADORES



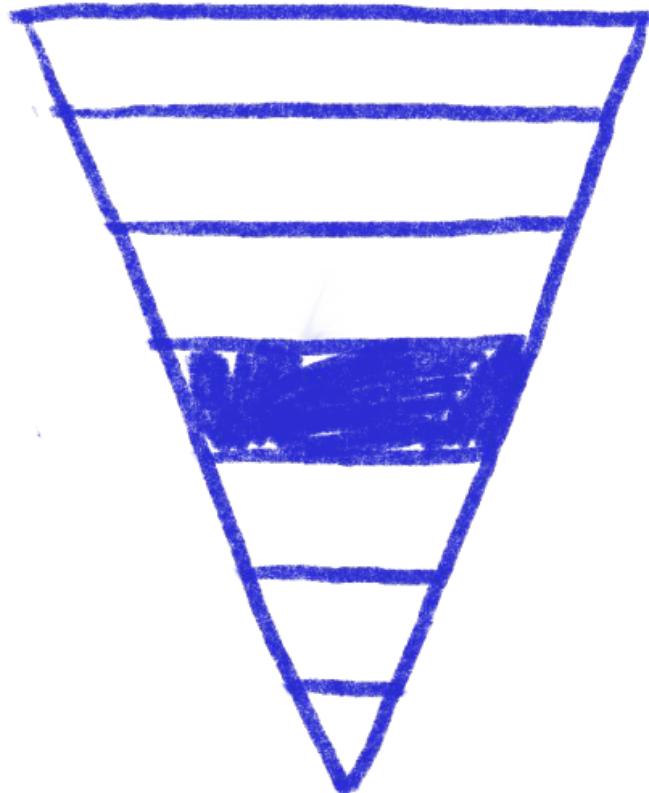
```
/* main.scss file */
@import "tools.functions";
@import "tools.mixins";
```



GENERIC
GENÉRICO
CSS COMÚN



```
/* main.scss file */
@import "generic.reset";
@import "generic.normalize";
```



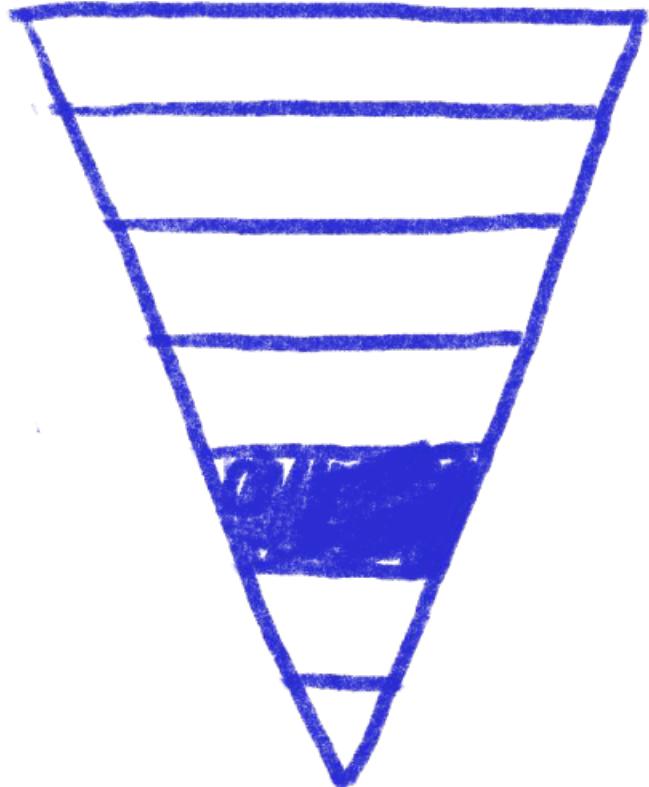
ELEMENTS
ELEMENTO

ESTILOS DE
SELECTOR DE
ELEMENTOS.

NAV
FORM



```
/* main.scss file */
@import "elements.forms";
@import "elements.table";
```



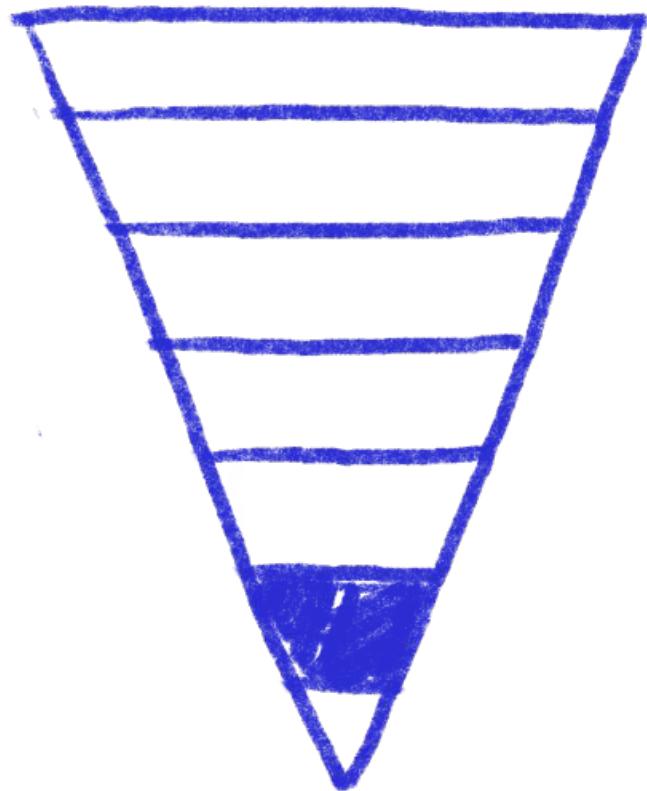
OBJECTS
OBJETOS

SELECTORES
BASADOS EN
CLASES PARA
OBJETOS

→ CONTENEDOR



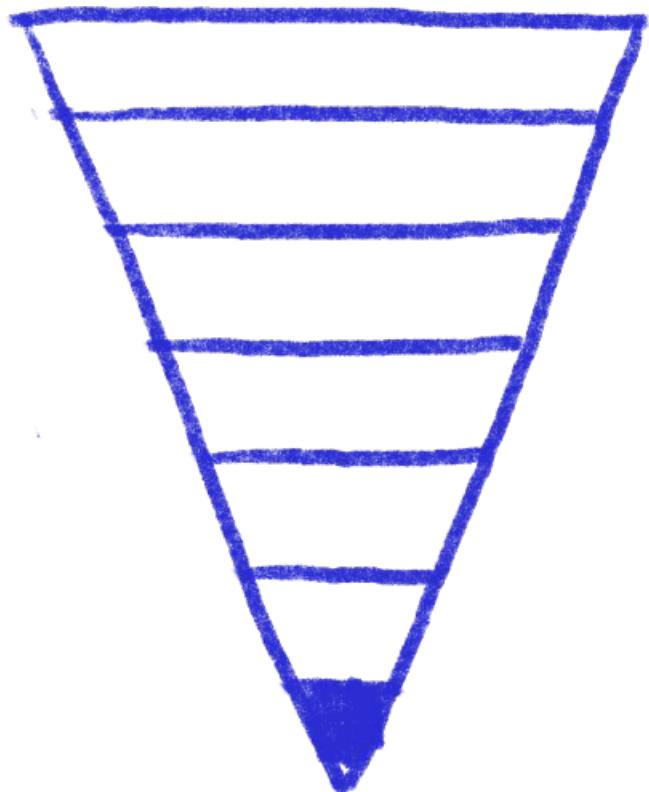
```
/* main.scss file */
@import "objects.container";
@import "objects.grid";
```



COMPONENTS
COMPONENTES
ESTILOS PARA
CADA COMPONENTE



```
/* main.scss file */
@import "components.site-nav";
@import "components.buttons";
@import "components.carousel";
```



TRUMPS
TRIUNFOS

ESTILOS DE
UTILIDAD
Y AYUDA

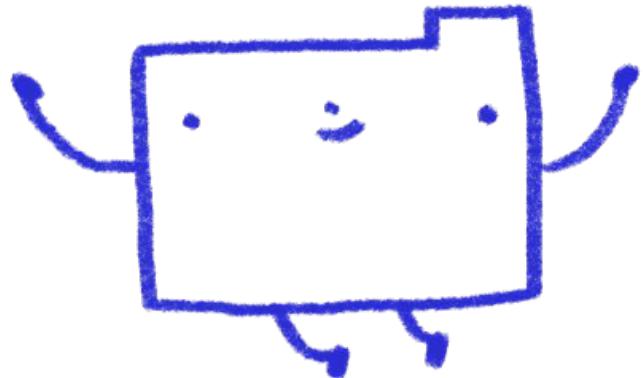
PUEDE ANULAR
ESTILOS

→ ! IMPORANT



```
/* main.scss file */
@import "trumps.clearfix";
@import "trumps.utilities";
```

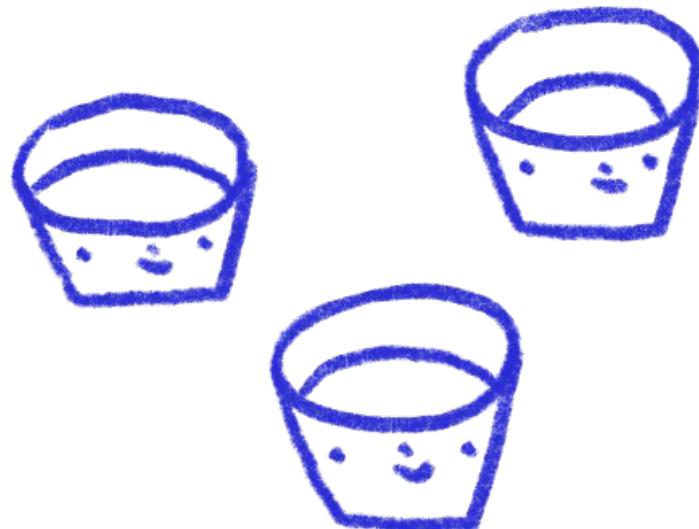
¡TIEMPO DE PROYECTO!



ITCSS
EN UN PROYECTO DE
VUE JS (NO ÉXITO)

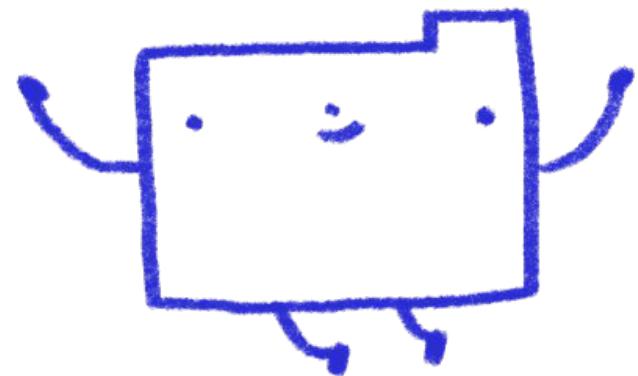
FRAMEWORKS

CSS



REVISEMOS EL
STATE OF CSS

¡TIEMPO DE PROYECTO!



TAILWIND CSS
EN UN PROYECTO DE
VUE JS

DIFERENCIAS ENTRE
METODOLÓGIAS Y
ARQUITECTURAS

¿QUÉ USAR?

LA ARQUITECTURA

DEPENDE DE:

- EL EQUIPO
- EL PROYECTO
- EL TIEMPO
- SI HAY O NO DISEÑO