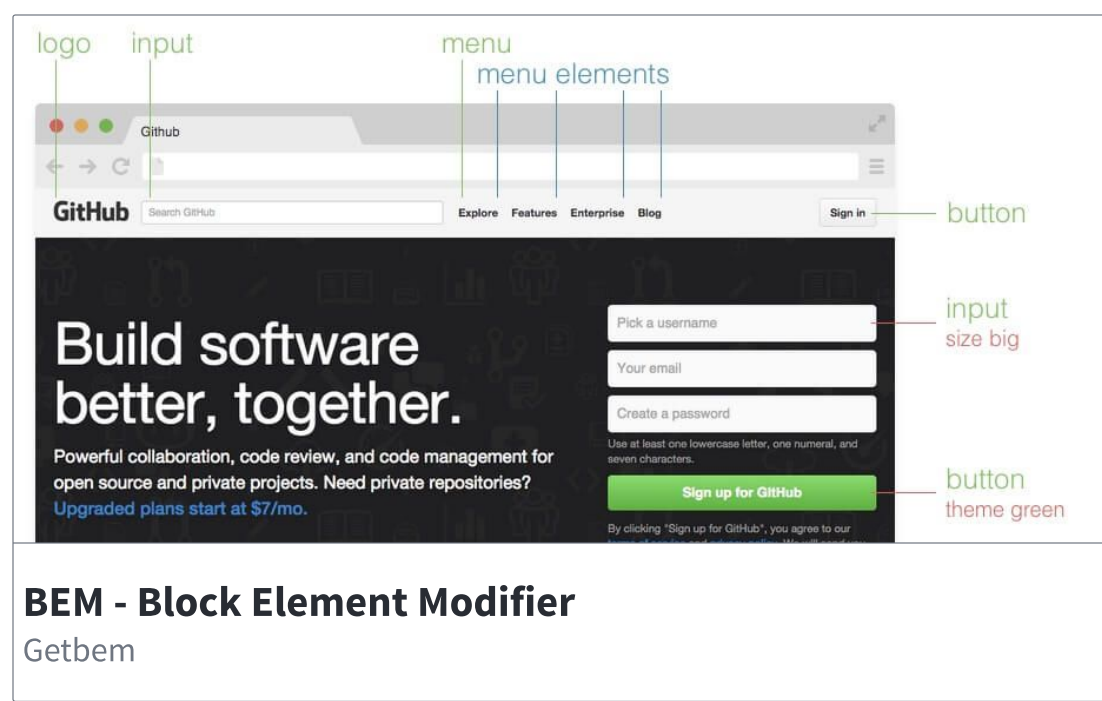


## Trabajar BEM con bootstrap

[illegible][illegible]

**Podemos poner bloques dentro de bloques? la respuesta es si**



DOCUMENTACION

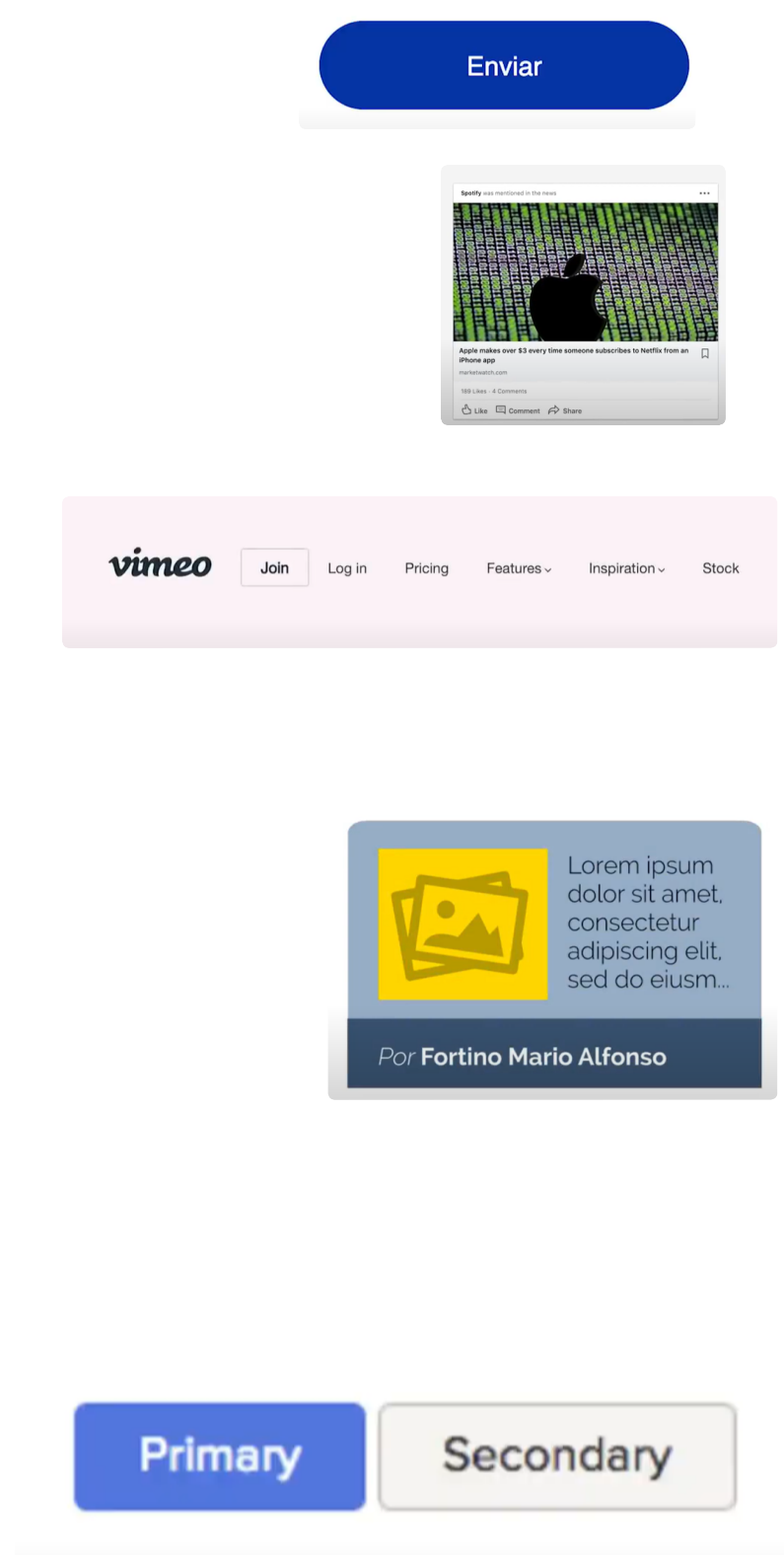


## BEM - Block Element Modifier

Getbem

## BEM - Block Element Modifier

Getbem



**BEM es mas recomendable para proyectos grandes**

La metodología BSC se encuentra en constante evolución, pero ha demostrado su efectividad al establecer una buena práctica en el desarrollo de plataformas.

En el tiempo que llevamos estudiando hemos percibido las siguientes ventajas:

- **Añade especificidad.** Al ser un indicador único para cada área, lo que permite reducir y hacer menos repetitivas.
- **Aumenta la independencia.** Los bloques se pueden mover o cualquier parte del documento, sin afectar sus estilos.
- **Permite la flexibilidad metodológica.** Se puede cambiar cualquiera de las tres perspectivas al afectar a las áreas.
- **Evita la reutilización.** Es posible crear una serie de bloques de código dentro de un proyecto hacia otro, con lo que se evita la existencia de dependencias mayores entre la construcción de cada uno de los bloques interrelacionados.
- **Entrega información.** Permite la fácil implementación en una consola o programación avanzada como el caso de Java, así como la construcción a la vez de mostrar los datos generados por un control específico al poder usar cualquier herramienta de desarrollo para realizar pruebas de integración.

Si embargo, existen algunas limitaciones en contra de esta metodología. Las más comunes son:

- Las conexiones pueden ser muy largas.
- Algunas personas les puede tomar tiempo aprender la metodología.
- El sistema de organización puede ser difícil de implementar en un proyecto pequeño.

- La metodología BSC se encuentra en constante evolución, pero ha demostrado su efectividad al establecer una buena práctica en el desarrollo de plataformas.
- En el tiempo que llevamos estudiando hemos percibido las siguientes ventajas:
- **Añade especificidad.** Al ser un indicador único para cada área, lo que permite reducir y hacer menos repetitivas.
  - **Aumenta la independencia.** Los bloques se pueden mover o cualquier parte del documento, sin afectar sus estilos.
  - **Permite la flexibilidad metodológica.** Se puede cambiar cualquiera de las tres perspectivas al afectar a las áreas.
  - **Evita la reutilización.** Es posible crear una serie de bloques de código dentro de un proyecto hacia otro, con lo que se evita la existencia de dependencias mayores entre la construcción de cada uno de los bloques interrelacionados.
  - **Entrega información.** Permite la fácil implementación en una consola o programación avanzada como el caso de Java, así como la construcción a la vez de mostrar los datos generados por un control específico al poder usar cualquier herramienta de desarrollo para realizar pruebas de integración.
- Si embargo, existen algunas limitaciones en contra de esta metodología. Las más comunes son:
- Las conexiones pueden ser muy largas.
  - Algunas personas les puede tomar tiempo aprender la metodología.
  - El sistema de organización puede ser difícil de implementar en un proyecto pequeño.

La metodología BSC se encuentra en constante evolución, pero ha demostrado su efectividad al establecer una buena práctica en el desarrollo de plataformas.

En el tiempo que llevamos estudiando hemos percibido las siguientes ventajas:

- **Añade especificidad.** Al ser un indicador único para cada área, lo que permite reducir y hacer menos repetitivas.
- **Aumenta la independencia.** Los bloques se pueden mover o cualquier parte del documento, sin afectar sus estilos.
- **Permite la flexibilidad metodológica.** Se puede cambiar cualquiera de las tres perspectivas al afectar a las áreas.
- **Evita la reutilización.** Es posible crear una serie de bloques de código dentro de un proyecto hacia otro, con lo que se evita la existencia de dependencias mayores entre la construcción de cada uno de los bloques interrelacionados.
- **Entrega información.** Permite la fácil implementación en una consola o programa de gestión como es el caso de Ana, así como la construcción a la vez de mostrar los datos generados para un control específico al poder usar cualquier herramienta de gestión de recursos para la gestión de la información.

Si embargo, existen algunas limitaciones en contra de esta metodología. Las más comunes son:

- Las conexiones pueden ser muy largas.
- Algunas personas les puede tomar tiempo aprender la metodología.
- El sistema de organización puede ser difícil de implementar en un proyecto pequeño.

- La metodología BSC se encuentra en constante evolución, pero ha demostrado su efectividad al establecer una buena práctica en el desarrollo de plataformas.
- En el tiempo que llevamos estudiando hemos percibido las siguientes ventajas:
- **Añade especificidad.** Al ser un indicador único para cada área, lo que permite reducir y hacer menos repetitivas.
  - **Aumenta la independencia.** Los bloques se pueden mover o cualquier parte del documento, sin afectar sus estilos.
  - **Permite la flexibilidad metodológica.** Se puede cambiar cualquiera de las tres perspectivas al afectar a las áreas.
  - **Evita la reutilización.** Es posible crear una serie de bloques de código dentro de un proyecto hacia otro, con lo que se evita la existencia de dependencias mayores entre la construcción de cada uno de los bloques interrelacionados.
  - **Entrega información.** Permite la fácil implementación en una consola o programa de gestión como es el caso de Ana, así como la construcción a la vez de mostrar los datos generados para un control específico al poder usar cualquier herramienta de gestión de recursos para la gestión de la información.
- Si embargo, existen algunas limitaciones en contra de esta metodología. Las más comunes son:
- Las conexiones pueden ser muy largas.
  - Algunas personas les puede tomar tiempo aprender la metodología.
  - El sistema de organización puede ser difícil de implementar en un proyecto pequeño.

Cabe destacar que es posible aplicar esta metodología en diferentes lenguajes y contextos. Por ejemplo, destacando la estructuración en HTML5, en CSS y sus preprocesadores (como SASS o LESS) y en programación con JavaScript, considerando su utilización en frameworks como Angular JS o ReactJS.

**.bloque** **.bloque\_\_elemento**

**.bloque\_\_elemento**



```
.button { }
```

```

var result
  webpack.config({ mode: 'production' })
})

```

```

var result = webpack.config({
  mode: 'production',
  entry: './src/index.js'
})

```

```

var result
  webpack.config({ mode: 'production' })
})

```

```

var result = webpack.config({
  mode: 'production',
  entry: './src/index.js'
})

```

**Para proyectos pequeños  
tal vez caería mejor  
OOCSS**

**.bloque--modificador**

## .bloque\_\_elemento--modificador

Modificadores, Ojo siempre tiene que ir debajo del bloque o elemento en el archivo CSS.

HTML

Modificadores, Ojo siempre tiene que ir debajo del bloque o elemento en el archivo CSS.

HTML

```

<button class="button">
  <div>
    </div>
  </button>

<article class="post">

```

[illegible]

```
button--primary ()
button--secondary ()
```

Para aplicar el modificado, antes tengo que poner el bloque o el elemento y ya despues pongo el modificado

```
button class="button button--primary"
  Primary
</button>

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

```

public class Article {
    private String titre;
    private String contenu;
    private String auteur;
    private Date date;

    // Constructeur par défaut
    public Article() {}

    // Constructeur avec paramètres
    public Article(String titre, String contenu, String auteur, Date date) {
        this.titre = titre;
        this.contenu = contenu;
        this.auteur = auteur;
        this.date = date;
    }

    // Getters et Setters
    public String getTitre() {
        return titre;
    }

    public void setTitre(String titre) {
        this.titre = titre;
    }

    public String getContenu() {
        return contenu;
    }

    public void setContenu(String contenu) {
        this.contenu = contenu;
    }

    public String getAuteur() {
        return auteur;
    }

    public void setAuteur(String auteur) {
        this.auteur = auteur;
    }

    public Date getDate() {
        return date;
    }

    public void setDate(Date date) {
        this.date = date;
    }
}

```

[illegible][illegible][illegible]

The image consists of three panels illustrating code used in a video analysis project.

- Left Panel:** A snippet from a file named `new_link_color.py`. It shows a dictionary definition: `color = {'name': 'brown'}`.
- Middle Panel:** A snippet from a file named `main.py`. It defines several variables: `margin = 5`, `pading = 5`, `box_size = border-horiz`, `box_v`, `font-family: serif`, and `align=left; top=0;`.
- Right Panel:** A text box containing the following text: "Aquí la logica que sigue el autor del vídeo, es que aunque 'nav\_img' tiene dentro de 'nav' aunque no lo está directamente, ya por esa condición debe ser un elemento de 'nav', pero estoy suponiendo que no está siguiente realmente las reglas de BEM". To the right of this text is a link: [Consultar más información sobre documentación de este asy muy](#).

