



inicio

# Temario

04

## Grid

- ✓ Grid
- ✓ Grid + flexbox
- ✓ Implementación
- ✓ Diseño responsive
- ✓ Media queries
- ✓ Mobile first
- ✓ Grids+Flex+@Media

05

## - Pseudoclases y Pseudoelementos - BEM - GIT

- ✓ Pseudoclases
- ✓ Pseudoelementos
- ✓ BEM
- ✓ Git
- ✓ Comandos básicos
- ✓ Repositorios
- ✓ Ramas

06

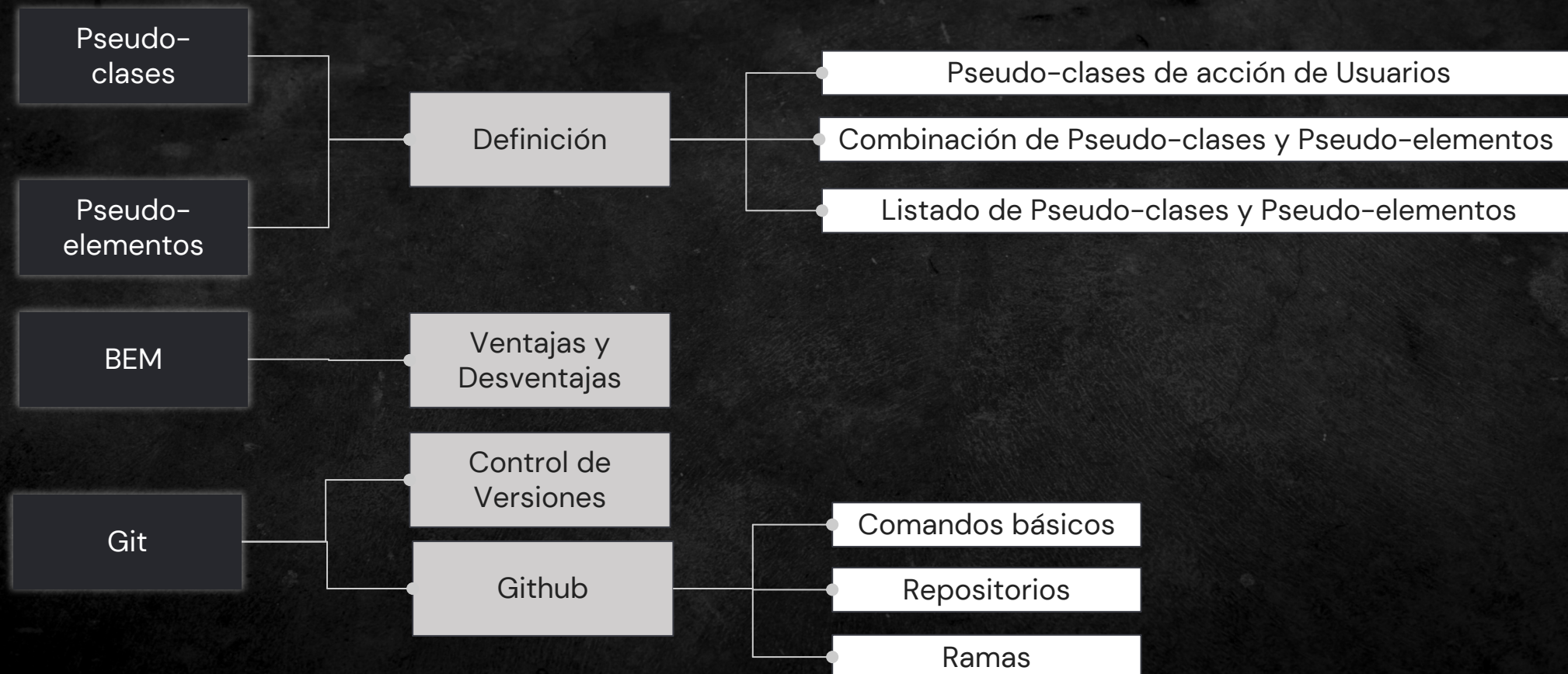
## Frameworks CSS + Bootstrap

- ✓ Librerías y Frameworks
- ✓ Bootstrap
- ✓ Estructura de Bootstrap



# Mapa de conceptos

## - Pseudoclases y Pseudoelementos - BEM - GIT



# Definición

**Pseudo-clases:** Modifica el aspecto o comportamiento de un elemento existente en función de su estado (hover, active, first-child, etc.).

**:pseudo-class-name**

---

**Pseudo-elementos:** Añade contenido o estilos a partes específicas de un elemento, como el principio, el final o un marcador. Genera nuevo contenido.

**::pseudo-element-name**



# ¿Cuándo emplearlo?

**Pseudo-clases:** Para crear efectos interactivos, cambiar el estilo de elementos en función de su estado, etc..

---

**Pseudo-elementos:** Para añadir decoraciones, generar contenido antes o después de un elemento, crear layouts más complejos, etc..

# Pseudoclases de acción de Usuario

“Algunas pseudoclases solo intervienen cuando el usuario interactúa con el documento de alguna manera. Estas **pseudoclases de acción de usuario**, que también reciben el nombre de **pseudoclases dinámicas**, actúan como si se añadiese una clase al elemento cuando el usuario interactúa con él” (mdn).

## Ejemplo



```
1  nav:hover{  
2    font-size: 2rem;  
3  }
```

Modificar el tamaño de la fuente al pasar el mouse



# Pseudoclases + Pseudoelementos

```
selector #id .clase [atributo] :pseudoclase ::pseudoelemento {  
    propiedad : valor ;  
    propiedad : valor  
}
```

“Si quieres poner en negrita la primera línea del primer párrafo, puedes encadenar los selectores `:first-child` y `::first-line`. Añade al ejemplo anterior el CSS siguiente. Queremos que se seleccione la primera línea del primer elemento `<p>` que esté dentro de un elemento `<article>`.”

```
1 article p:first-child::first-line {  
2     font-size: 120%;  
3     font-weight: bold;  
4 }
```

# BEM [Block, Element, Modifier]

## Bloques, Elementos, Modificadores

Metodología que te ayuda a organizar tu código CSS de una manera más estructurada y escalable.



```
1 .button{...}  
2 .button__text{...}  
3 .button--disabled{...}  
4 .button__text--large{...}
```

- **Bloques**: Se nombran con un nombre único y en minúsculas.
- **Elementos**: Se unen al bloque con dos guiones bajos (\_\_).
- **Modificadores**: Se unen al bloque o elemento con dos guiones (--).



# BEM

[Block, Element, Modifier]

## Ventajas

---

**Mayor legibilidad:** El código es más fácil de entender y mantener.

**Reutilización:** Los bloques y elementos pueden reutilizarse en diferentes partes de la aplicación.

**Escalabilidad:** Es fácil agregar nuevos componentes y modificar los existentes sin afectar a otras partes del código.

**Colaboración:** Facilita el trabajo en equipo, ya que todos los miembros del equipo siguen las mismas convenciones de nomenclatura.



# BEM

[Block, Element, Modifier]

## Desventajas

### Nomenclatura larga:

- **Mayor cantidad de código:** A medida que aumenta la complejidad del proyecto, los nombres de las clases pueden volverse muy largos, lo que puede dificultar la lectura y escritura del código.
- **Mayor posibilidad de errores:** Con nombres tan largos, es más fácil cometer errores tipográficos, lo que puede generar problemas en el estilo.

### Curva de aprendizaje:

- **Tiempo de adaptación:** Puede llevar un tiempo acostumbrarse a la sintaxis y a la forma de pensar en BEM, especialmente para desarrolladores que están empezando.



# BEM

[Block, Element, Modifier]

## Desventajas

### Rigidez:

- **Restricciones:** La estructura rígida de BEM puede limitar la creatividad en algunos casos, ya que puede ser difícil encontrar la mejor manera de nombrar ciertos elementos.

### Sobre-ingeniería:

- **Proyectos pequeños:** En proyectos pequeños, puede resultar excesivo utilizar BEM, ya que la complejidad de la metodología puede no justificarse.

### Confusión con otras metodologías:

- **Combinación con otras:** Al combinar BEM con otras metodologías (como OOCSS o SMACSS), puede surgir cierta confusión en cuanto a la nomenclatura y la estructura del CSS.



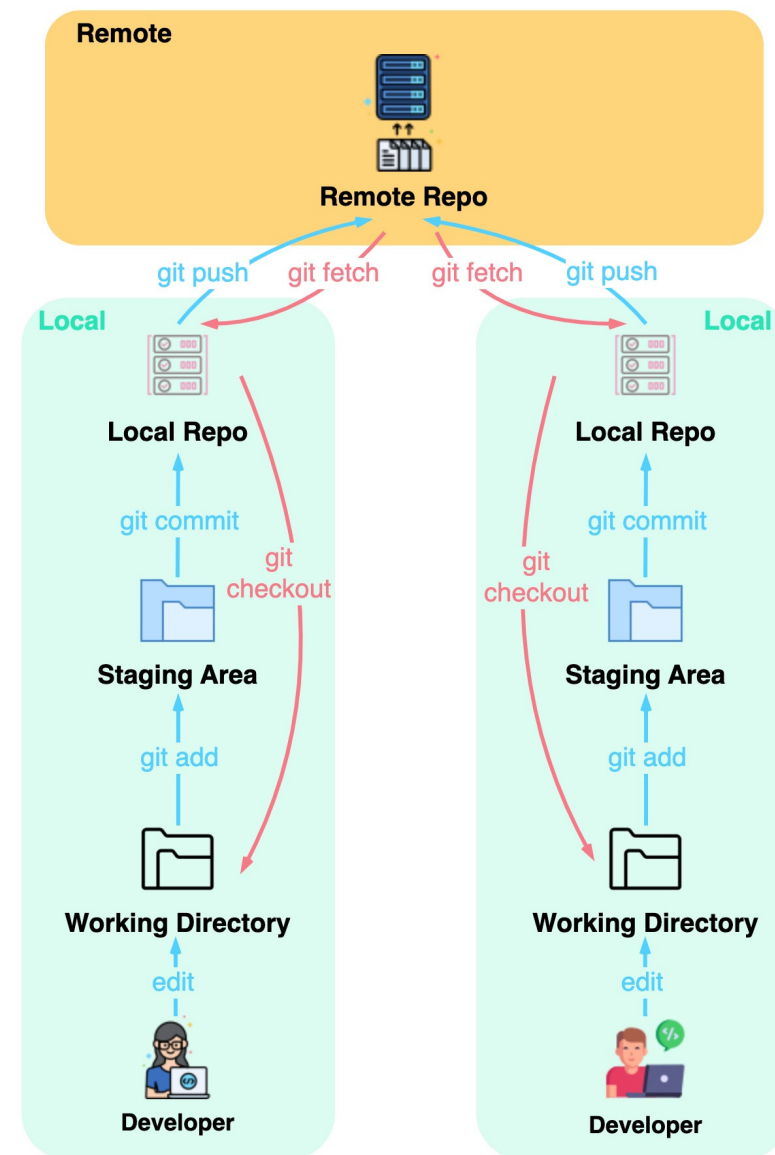
Git es un **sistema de control de versiones** distribuido, gratuito y de código abierto, diseñado para gestionar todo, desde proyectos pequeños hasta muy grandes, con velocidad y eficiencia.



<https://git-scm.com/>

## How does Git Work?

[blog.bytebytego.com](https://blog.bytebytego.com)







Configurar GIT desde GitBash.

---

```
$ git config --global user.name "usuarioGitHub"
```

```
$ git config --global user.email "emailUsuarioGitHub"
```

```
$ git config --list
```

Repositorio es el espacio (carpeta) donde se almacena, organiza , mantiene y difunde la información.

---

```
$ git init nombreDeMiRepositorio
```



**Status** chequea el estado del repositorio

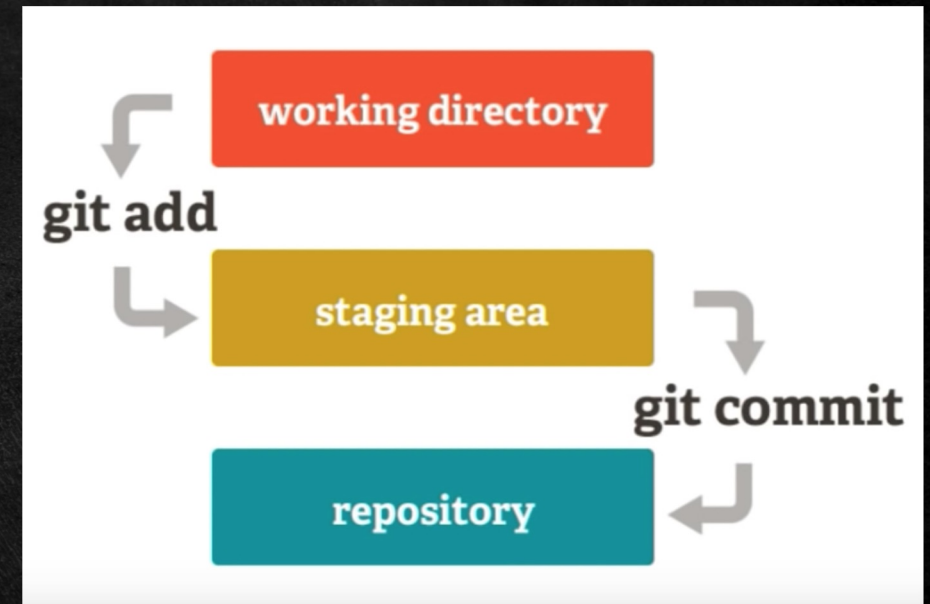
```
$ git status
```

**Add** agrega el o los archivos al Staging Area

```
$ git add .
```

**Commit** envía los ajustes al repositorio local.

```
$ git commit -m "comentario referente al asunto del commit"
```

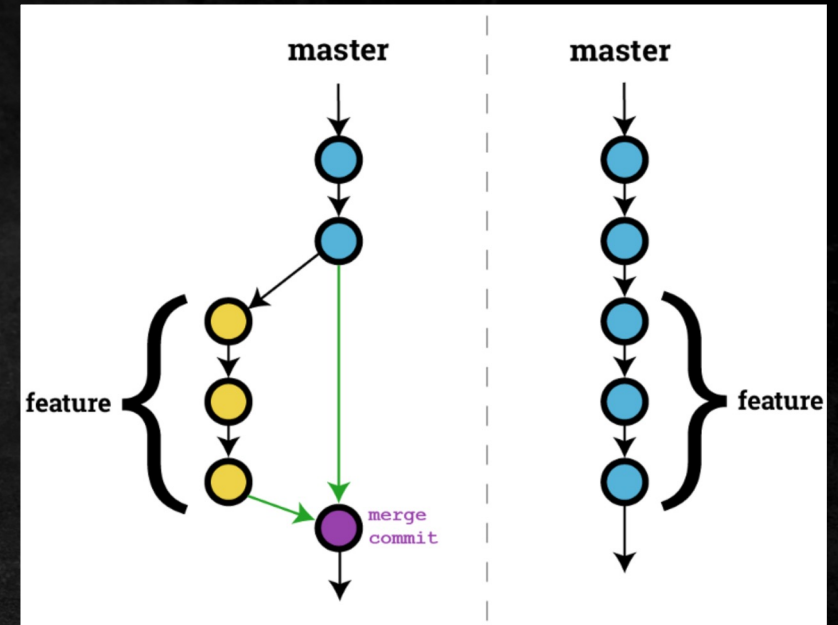






**Log** presenta el historial de las versiones.

```
$ git log
```



## RAMAS (Branch)

- **branch** permite ver todas las ramas y crear una nueva rama  

```
$ git branch nombreDeLaRama
```

```
$ git branch // ver todas las ramas
```

```
$ git branch -D ramaAeliminar
```
- **checkout** permite navegar entre ramas.  

```
$ git checkout nombreDeLaRamaDestino
```

```
$ git status // ver rama actual
```



**Ejercicio** Crear un usuario en GitHub + crear un repositorio + vincularlo a VSC + realizar un git push

---

