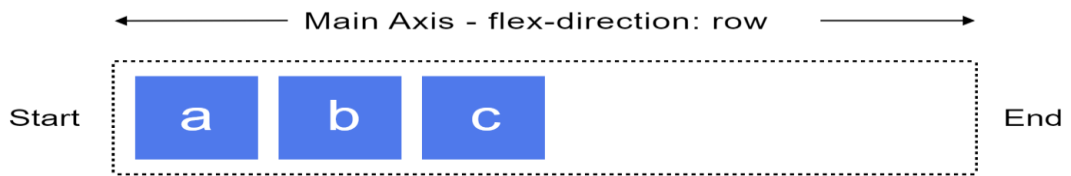


PROPIEDADES DE JUSTIFY-CONTENT

- **flex-start:** Es el valor predeterminado que tendremos al ejecutar “justify-content”, básicamente se refiere a como su propio nombre lo indica, que los elementos se coloquen al principio.



```
.error_wrapper {  
  display: flex;  
  justify-content: flex-start;  
  align-items: center;  
  height: 100vh;  
  flex-direction: column;  
}
```

- **flex-end:** Con esta propiedad haremos que todos nuestros elementos se agrupen al final de nuestro contenedor flex.



SCALER
Topics

```
.error_wrapper {  
  display: flex;  
  justify-content: flex-end;  
  align-items: center;  
  height: 100vh;  
  flex-direction: column;  
}
```

- **center:** Lo utilizaremos para centrar nuestros elementos justo en el medio de nuestro contenedor flex.

justify-content: center



```
.error_wrapper {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  height: 100vh;  
  flex-direction: column;  
}
```

- **space-between:** Nuestros elementos se distribuirán de manera uniforme a lo largo de todo nuestro contenedor flex, una de sus principales características es que el primer elemento estará al principio del todo, y el último elemento al final de la línea.

With justify-content: space-between



```
.error_wrapper {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  height: 100vh;  
  flex-direction: column;  
}
```

OBJETIVO PRIMORDIAL

Cuando hablamos de objetivo primordial nos referimos nuestro objetivo principal, o bien podríamos decir que es el propósito que queremos lograr en nuestro proyecto. En nuestro objetivo primordial debemos definir el resultado que deseamos obtener al ejecutar el programa.

Cabe destacar que nuestro objetivo primordial debe ser claro, conciso y alcanzable, ya que el mismo debe guiarnos durante el proceso de desarrollo y nos servirá como parámetro de evaluación para determinar el éxito de nuestro proyecto.

En resumen, nuestro objetivo principal se refiere a la meta principal de nuestro proyecto de desarrollo.

WIREFRAMES DE BAJA FIDELIDAD

Los Wireframes low fidelity son la representación visual de comunicar nuestra "gran idea" de una manera simplificada y no tan profundizada, lo cual nos permitirá definir la estructura y funcionalidad de la interfaz de usuario de una forma rápida y sencilla, además facilita la comunicación entre miembro del equipo de diseño y los desarrolladores porque permite que ambos tengas una imagen del objetivo final que se quiere lograr. También nos facilita la detección temprana de problemas de usabilidad o accesibilidad, lo cual nos evitará problemas y pérdidas de tiempo a futuro, ya que no ahorra la necesidad de invertir tiempo y esfuerzo en características que podrían no ser bien aceptadas por el usuario final.

Los Wireframes de baja fidelidad se caracterizan por ser sencillos y rápidos de crear ya que no necesariamente deben estar hechos a través de herramientas digitales, podemos realizar un wireframes de baja fidelidad simplemente con papel y lápiz, ya que su principal foco no son los detalles visuales como colores, tipografías o imágenes.

GIT

Git es un sistema de control de versiones de código fuente, la utilizaremos para llevar un registro de los cambios en el código fuente de nuestro proyecto, también nos permite a varios desarrolladores trabajar en el mismo proyecto de manera simultánea, en pocas palabras podríamos imaginar que Git es una maquina del tiempo para nuestro código ya que entre sus principales características nos permite:

- Guardar un historial de todos los cambios que realizan en nuestros archivos.
- Volver a versiones anteriores del código por si algo sale mal.
- Trabajar en equipo de forma organizada, fusionando los cambios de diferentes personas sin problemas.
- Registra cada modificación en nuestro código, permitiéndonos rastrear errores y experimentar con nuevas ideas sin miedo a perder trabajo previo.
- Permite crear "ramas" de desarrollo para trabajar en nuevas funcionalidades sin afectar la versión principal del proyecto. Luego, podemos fusionar los cambios de forma segura.

-Nuestro código se almacenara en un repositorio (local y/o remoto), lo que sirve como respaldo en caso de fallos en nuestro equipo.

-Git define un flujo de trabajo claro y estructurado, lo que mejora la productividad y la calidad del código.

Dejare algunos comandos básicos de Git que pueden sernos de utilidad para dar nuestros primeros pasos:

Comandos básicos:

git init: Inicializa un nuevo repositorio Git en tu proyecto.

git clone <url>: Clona un repositorio existente desde una URL (por ejemplo, desde GitHub).

git add <archivo>: Añade un archivo al "área de preparación" (staging area), listo para ser incluido en el próximo commit.

git commit -m "mensaje": Crea un nuevo commit con los cambios en el área de preparación, incluyendo un mensaje descriptivo.

git status: Muestra el estado del repositorio, incluyendo archivos modificados, añadidos y pendientes de commit.

git log: Muestra el historial de commits, incluyendo información sobre autor, fecha y mensaje.

git diff: Muestra las diferencias entre la versión actual de un archivo y la versión en el último commit.

Ramas (branches):

git branch: Lista las ramas existentes.

git branch <nombre>: Crea una nueva rama.

git checkout <nombre>: Cambia a una rama diferente.

git merge <nombre>: Fusiona los cambios de otra rama a la rama actual.

git branch -d <nombre>: Elimina una rama (si ya ha sido fusionada).

Trabajo remoto:

git remote add origin <url>: Añade un repositorio remoto (generalmente llamado "origin").

git push origin <rama>: Envía tus commits a la rama especificada del repositorio remoto.

git pull origin <rama>: Descarga los cambios de la rama especificada del repositorio remoto y los fusiona con tu rama local.

SITEMAPS

Un sitemap es un archivo que enumera todas las páginas importantes de una web, sirve de guía a los motores de búsqueda para determinar qué páginas deben añadir a su índice y cuáles deben dejar fuera, Tener un sitemap es una parte fundamental del SEO de tu web. Gracias al trabajo que realiza buscadores como por ejemplo "Google" pueden descubrir tus páginas más rápido y de manera más eficiente.

Hay dos tipos de sitemaps web:

Sitemap XML: Son sitemaps escritos en un formato específico para los rastreadores de los motores de búsqueda.

Sitemap HTML: Son sitemaps que parecen páginas normales y ayudan a los usuarios a navegar por tu web.

Sitemap XML:

Los sitemaps XML son el formato preferido de los motores de búsqueda, como Google. Contienen tres tipos de información:

- La lista de todas las URL que quieres indexar.
- El atributo "lastmod" que informa de cuándo fue la última vez que se actualizaron las URL.
- El atributo "hreflang" que informa sobre las variaciones locales de las URL.

Este es el aspecto de un sitemap XML:

```
▼<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  ▼<url>
    <loc>https://chamberlaincoffee.com/pages/12-month-gift-subscription-voucher</loc>
    <lastmod>2021-12-16T09:12:33-08:00</lastmod>
    <changefreq>weekly</changefreq>
  </url>
  ▼<url>
    <loc>https://chamberlaincoffee.com/pages/6-month-gift-subscription-voucher</loc>
    <lastmod>2021-12-16T09:12:09-08:00</lastmod>
    <changefreq>weekly</changefreq>
  </url>
  ▼<url>
    <loc>https://chamberlaincoffee.com/pages/about</loc>
    <lastmod>2021-02-15T06:44:51-08:00</lastmod>
    <changefreq>weekly</changefreq>
  </url>
```

Sitemap HTML


En su momento, los sitemaps HTML eran una forma popular de mejorar la navegación de una web y añadir enlaces a todas tus páginas desde un solo lugar, Aunque ya no son tan populares como antes, algunos profesionales de la comunidad SEO siguen afirmando que

son importantes, Los sitemap HTML pueden mejorar tu entrelazado interno y proporcionar otra capa de navegación para webs complejas con muchas páginas. No deben utilizarse como sustituto de una buena navegación (como menús, enlaces a pie de página, migas de pan, categorías, etc.).

John Mueller, Search Advocate de Google, afirmó en Mastodon:

"Si crees que necesitas un sitemap HTML, es mejor que dediques tiempo a mejorar la arquitectura de tu web".

Este es un ejemplo de sitemap HTML del Grupo H&M:

[H&M Group](#) [About us](#) [Brands](#) [Join us](#) [Sustainability](#) [Investors](#) [Corporate governance](#) [Media](#) 

Sitemap

[Home](#) > [Sitemap](#)

About us

- [At a Glance: H&M Group](#)
- [Our way](#)
- [Our values](#)
- [Inclusion and diversity](#)
 - [People](#)
 - [Business](#)
 - [Community](#)
- [Markets and expansion](#)
 - [Market overview](#)
 - [Store count per brand](#)
- [History](#)
 - [The '40s & the '50s](#)
 - [The '60s & '70s](#)
 - [The '80s & '90s](#)
 - [The '00s](#)

Sustainability

- [Sustainability reporting](#)
- [How we report](#)
 - [Materiality](#)
 - [Reporting frameworks and indexes](#)
 - [Vision and strategy](#)
 - [Goals and ambition](#)
 - [Transparency](#)
 - [Supply chain](#)
 - [Responsible purchasing practices](#)
- [Innovation](#)
 - [Awards and recognitions](#)
 - [Memberships and collaborations](#)
 - [Stakeholder engagement](#)

Investors

- [Annual & Sustainability Report 2021](#)
- [Financial calendar](#)
- [Reports and presentations](#)
- [The share](#)
 - [Shareholders](#)
 - [Dividend](#)
 - [Share buyback](#)
 - [Debt Financing](#)
 - [Sustainability-Linked Finance](#)
 - [Five year summary](#)

Corporate governance

- [About corporate governance](#)
- [Corporate governance report](#)
- [Financial calendar](#)

Nombra tres valores que se pueden usar con flex-direction, incluido el valor predeterminado que usa flexbox cuando se crea.

- **Flex-direction: row;** Este es el valor que vendrá predeterminado, su función es alinear los elementos se colocan en fila de forma horizontal.

```
.contenedor {  
  display: flex;  
  flex-direction: row; /* Valor predeterminado */  
}
```

- **flex-direction: column;** Su función es alinear los elementos en columna de forma vertical.

```
.contenedor {  
  display: flex;  
  flex-direction: column;  
}
```

Flex-direction: row-reverse; Similar a row en su alineación horizontal, pero los elementos se colocan en línea de derecha a izquierda, útil para diseños que requieren un orden inverso.

```
.contenedor {  
  display: flex;  
  flex-direction: row-reverse;  
}
```