

{desafío}
latam_

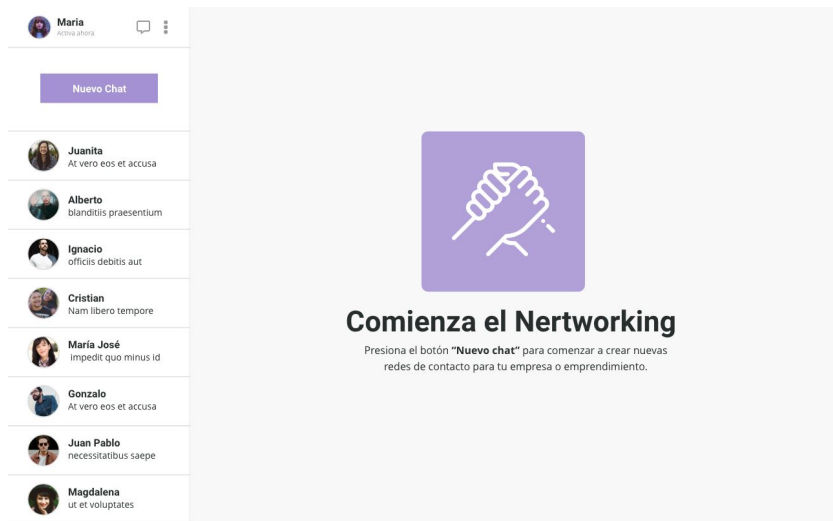
Flexbox _

Parte I



Proyecto

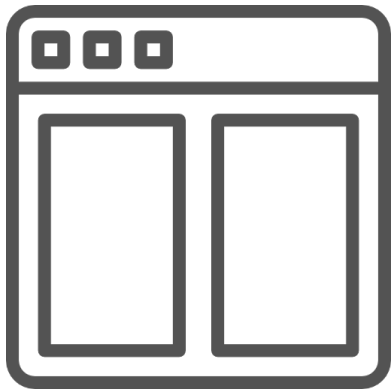
Maquetar vista principal APP



- Maquetar vista principal de una aplicación web llamada **“OK Talk”**.
- Esta tiene como objetivo conectar a usuarios a modo de generar contactos entre empresas y emprendedores.
- Tendremos a disposición 3 mockups, imágenes y guía de estilos.

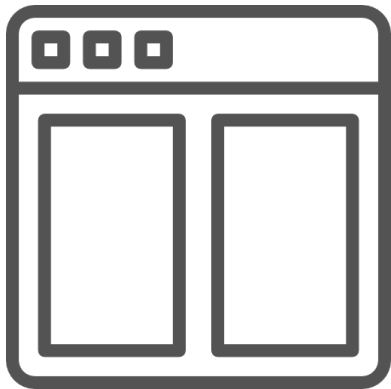
Conociendo *Flexbox*

¿Qué es flexbox?



- *Flexbox* es una herramienta de CSS que nos ayudará a crear layouts de manera más eficiente, ya que podremos distribuir el espacio y alinear el contenido de diferentes maneras dentro de un sitio o aplicación web
- Flexbox es similar a las propiedades de bloque como los elementos flotantes (*float*).

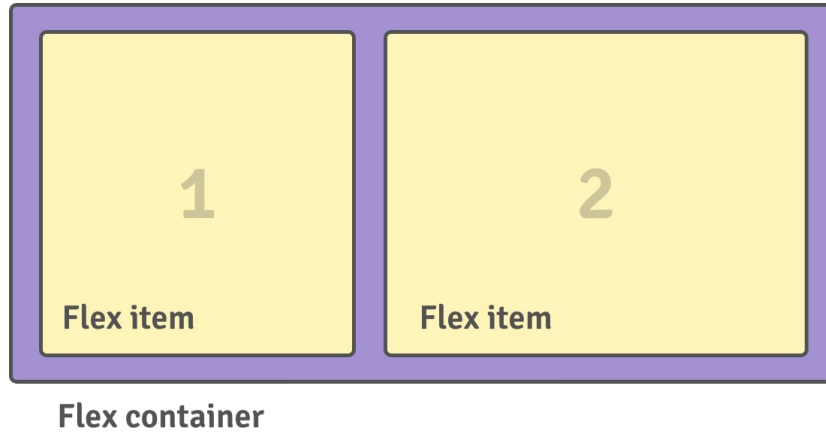
¿Qué es flexbox?



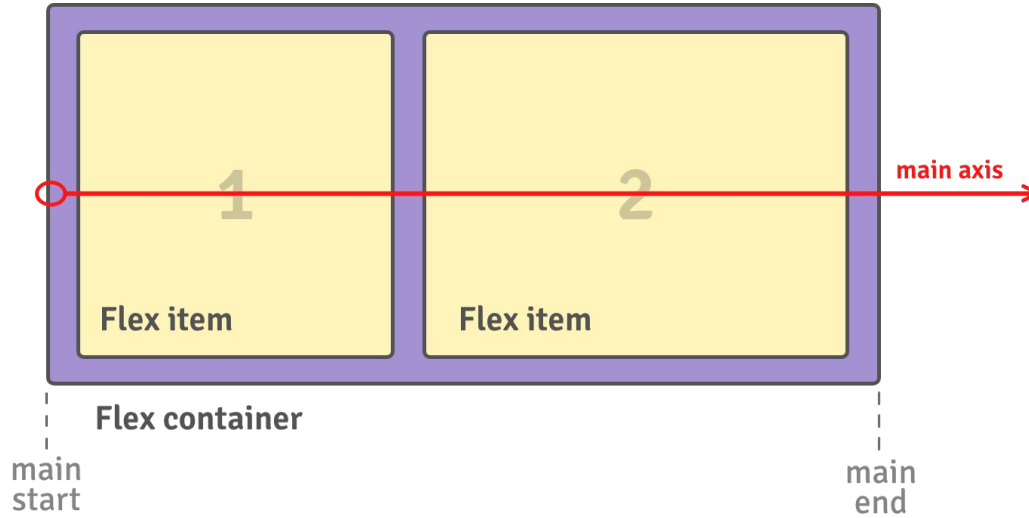
- *Flexbox* es agnóstico a la dirección del flujo normal de los elementos.
- Esto último sucede porque flexbox tiene un “modelo de cajas *flexible*”.

Modelo de cajas flexible

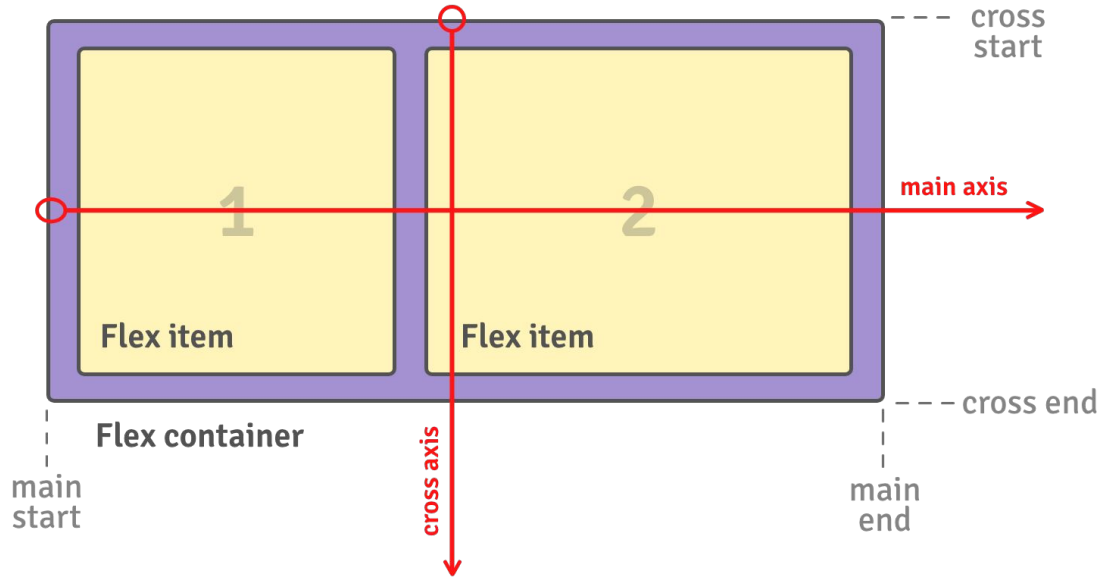
Contenedores e ítems



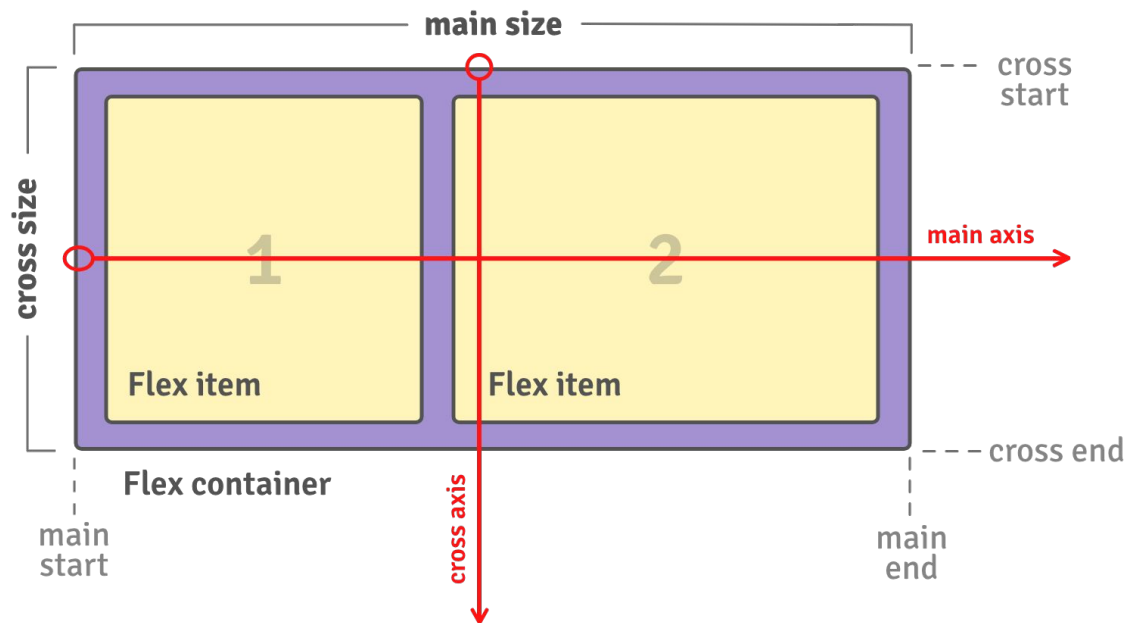
Main axis



Cross axis



Tamaños



Propiedades de contenedor

Display: flex

```
.container {  
  display: flex;  
}
```

- Esta propiedad define a un contenedor cómo flexible y a todos sus hijos directos.

Flex-direction

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

- Propiedad que nos permitirá definir si la dirección de los elementos será definida por el **main axis** o por el **cross axis**.
- Por defecto la dirección de los elementos es **row**.
- Podemos usar como valores: **row**, **row-reverse**, **column** y **column-reverse**.

Flex-wrap

```
.container {  
  display: flex;  
  flex-wrap: wrap;  
}
```

- Propiedad que nos ayudará a definir el ajuste de los elementos contenidos por el contenedor.
- Por defecto (***nowrap***) los elementos se ajustan en una línea siguiendo el main axis.
- Si deseamos cambiar eso podremos usar el valor ***nowrap***.

Flex-flow

```
.container {  
  display: flex;  
  flex-flow: column nowrap;  
}
```

- Versión reducida las propiedades **flex-direction** y **flex-wrap**.

Justify-content

```
.container {  
  display: flex;  
  justify-content: center;  
}
```

- Esta propiedad alinea los elementos flexibles a lo largo del main axis del contenedor.
- Esto significa que podremos distribuir de mejor manera el espacio y desborde.
- Los valores que podremos usar son: ***flex-start, flex-end, center, space-between, space-around, y space-evenly.***

Align-items

```
.container {  
  display: flex;  
  align-items: center;  
}
```

- Esta propiedad nos ayudará a alinear los elementos siguiendo al **cross axis**.
- Por defecto el valor de los elementos es **stretch**, que estira a los ítems respetando los anchos mínimos y máximos.
- Podemos usar valores como: **flex-start**, **flex-end**, **center**, **stretch** y **baseline**.

Align-content

```
.container {  
  display: flex;  
  align-content:  
  flex-start;  
}
```

- Propiedad que nos ayudará a alinear el contenido de nuestro contenedor.
- En esencia esta propiedad es similar a **justify-content**, que alinea los ítems dentro del **main axis**, pero en vez de usar este eje usa el **cross axis**.
- Esta propiedad no funcionará si los elemento se encuentran en una sola línea.

Propiedades de ítems

Order

```
.item-1 {  
  order: 2;  
}  
  
.item-2 {  
  order: 1;  
}
```

- Esta propiedad es usada para escoger el orden los elementos dentro de un contenedor *flexbox*.
- Lo interesante de **order** es que podemos decidir qué ítem irá primero y cuál último.

Flex-grow

```
.item-1 {  
  flex-grow: 1;  
}  
  
.item-2 {  
  flex-grow: 2;  
}
```

- Esta propiedad tiene la facultad de aumentar el tamaño de un elemento sólo si tiene el espacio suficiente para crecer.
- Su valor inicial es 0 y no acepta números negativos.

Flex-shrink

```
.item-1 {  
  flex-shrink: 1;  
}  
  
.item-2 {  
  flex-shrink: 2;  
}
```

- Esta propiedad funciona de manera similar a flex-grow cambiando el tamaño proporcionalmente al espacio disponible que tenga.
- Su valor inicial es 1 y no acepta números negativos.

Flex-basis

```
.item-1 {  
  flex-basis: 250px;  
}  
  
.item-2 {  
  flex-basis: 10vw;  
}
```

- Propiedad que nos permitirá definir la longitud inicial de un elemento.
- Podemos usar diferentes tipos de unidades de medida como **px** o **%**, así como también keywords como **auto**.

Flex

```
.item-1 {  
  order: 3;  
  flex: 1 1 300px;  
}
```

- Versión reducida que une a *flex-grow*, *flex-shrink* y *flex-basis* en una sola propiedad.

Align-self

```
.item-1 {  
  flex: 1 1 300px;  
  align-self: flex-end;  
}
```

- Propiedad que permite alinear un elemento de manera independiente siguiendo al **cross axis**.
- Los valores que podemos usar son los mismo que **align-items**.

{desafío}
latam_

*Academia de
talentos digitales*

www.desafiolatam.com