

¡Javascript!

guayerd beta hub



¡Buen Comienzo!

¿nos presentamos?



Ana Romero

TUTORA

- Hola, soy psicóloga 🧠 y me gusta escuchar a las personas, conocer sus historias y acompañarlas para que puedan superar sus dificultades.
- Un detalle personal: amo a los animales, sobre todo a mi perro Teodoro 🐶, que es un personaje.
- En mi tiempo libre disfruto leer 📖 y mirar series 📽.



Eliana Alvarez

FACILITADORA

- Soy Full-stack Developer.
- Me gusta leer, ver series y las plantas 🌵
- Soy toda una Lady Plant ☀️



Julio Avantt!

MENTOR

- Soy coder, profesor, creador de contenido, YouTuber, humorista en tiempos difíciles.
- Me apasiona aprender, comprender, enseñar, compartir conocimientos.
- Mi gata se llama Gata.

¡Ahora es su turno!

ESTUDIANTES



Podrán contarnos un poco sobre ustedes
ingresando a este link:

<https://www.menti.com/aletyywz92xu>

Para una buena clase

RESPETO

Generar un clima de confianza



PARTICIPACIÓN

Ser protagonistas de nuestro aprendizaje



COLABORACIÓN

Aprender con y de otras personas



MICRÓFONO APAGADO

Cuando no hablamos



CÁMARA PRENDIDA

Para conectarnos



COMPROMISO

Asistencia, atención y puntualidad



Temas de las primeras clases

Clase 1

HTML - CSS

Clase 3

GIT

Clase 5

Introducción a Javascript

Clase 7

Estructuras de control II

Clase 2

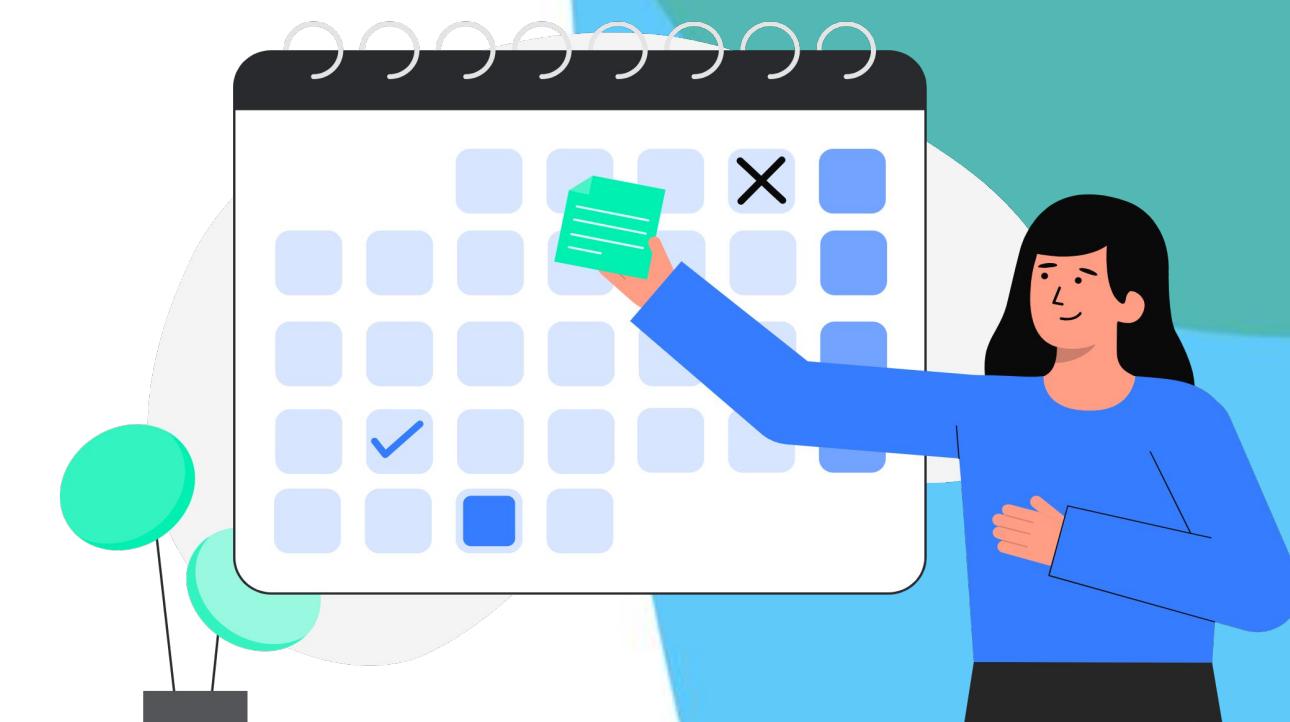
Flexbox - RWD -
Bootstrap

Clase 4

GITHUB

Clase 6

Estructuras de control I



Proyecto

¿Cuál será nuestro proyecto?



Proyecto

Nuestro proyecto será un **e-commerce** dinámico que obtendrá y actualizará datos desde una **API**.

Proyecto

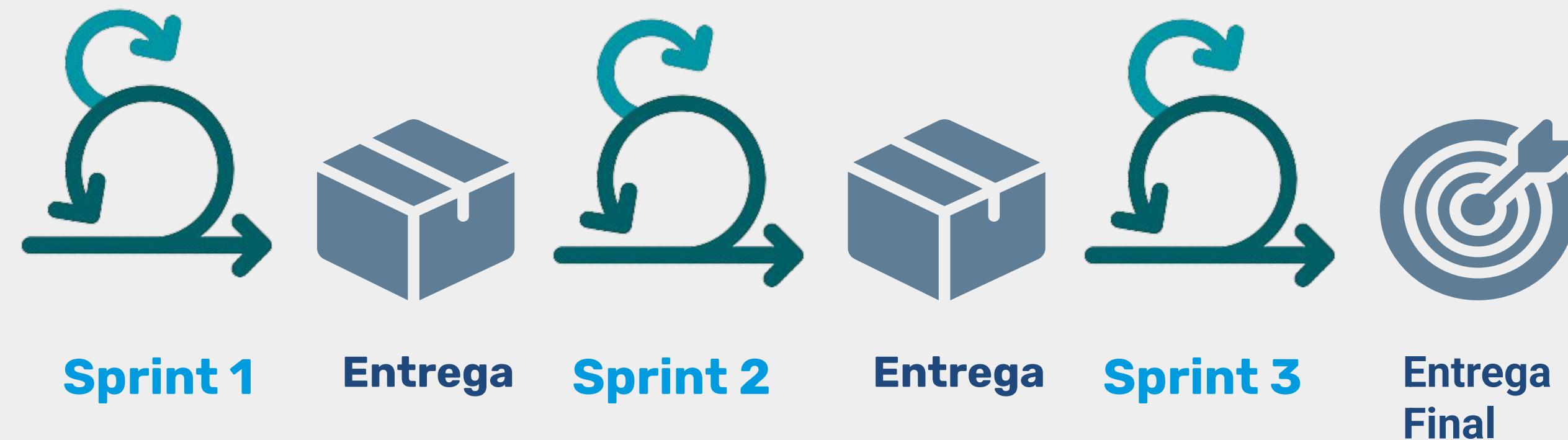
En este proyecto implementarás buenas prácticas de desarrollo y aprenderás Javascript usando:

- | | |
|------------------------|----------------------------------|
| Valores | Validación de formularios |
| Variables | Desestructuración |
| Coerción de tipos | Ternarios |
| Condiciones | Programación asincrónica |
| Estructuras de control | Pedidos a APIS |
| Bucles | DOM |
| Funciones | Manipulación del DOM |
| Objetos | Programación orientada a objetos |
| Arrays | Métodos |
| Eventos | Almacenamiento local |

Sprints project

30 CLASES

3 SPRINTS - 3 ENTREGABLES - 1 PROYECTO



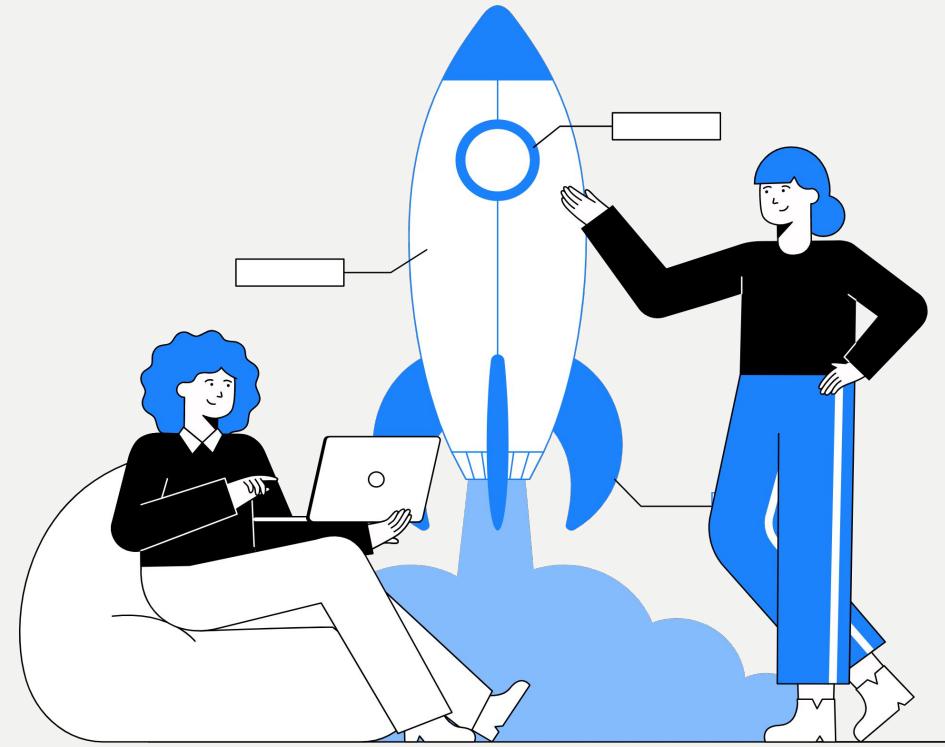
Colores de slides

Cambiamos el color de los bordes de las slides para diferenciar teoría, práctica y ceremonias ágiles dentro de la clase.

Ceremonias
ágiles

Teoría

Práctica



Ceremonias ágiles

Reuniones de equipo para garantizar que todos estén alineados en la planificación y desarrollo de un proyecto.



Actualiza a todos los miembros del equipo sobre qué se finalizó ayer, en qué trabajarán el día de hoy y si hay algún tipo de obstáculo. Son de corta duración.



Espacio de reflexión acerca de lo que resultó en el sprint pasado o qué se podría mejorar para la próxima vez.



Al finalizar el sprint se realiza una presentación del trabajo entregado, se comprueba si se alcanzó el objetivo y si hay que hacer cambios.

Agenda del día



01

**Introducción y
taller de trabajo en
equipo**

Objetivos del curso.

02

HTML y CSS

Etiquetas.

Atributos.

Selectores.

Propiedades.

Ejercitación

03

**Taller de trabajo en
equipo**

Instalar

Visual Studio Code

Es un editor de código. Este editor de código fue desarrollado por Microsoft para Windows, macOS y Linux.

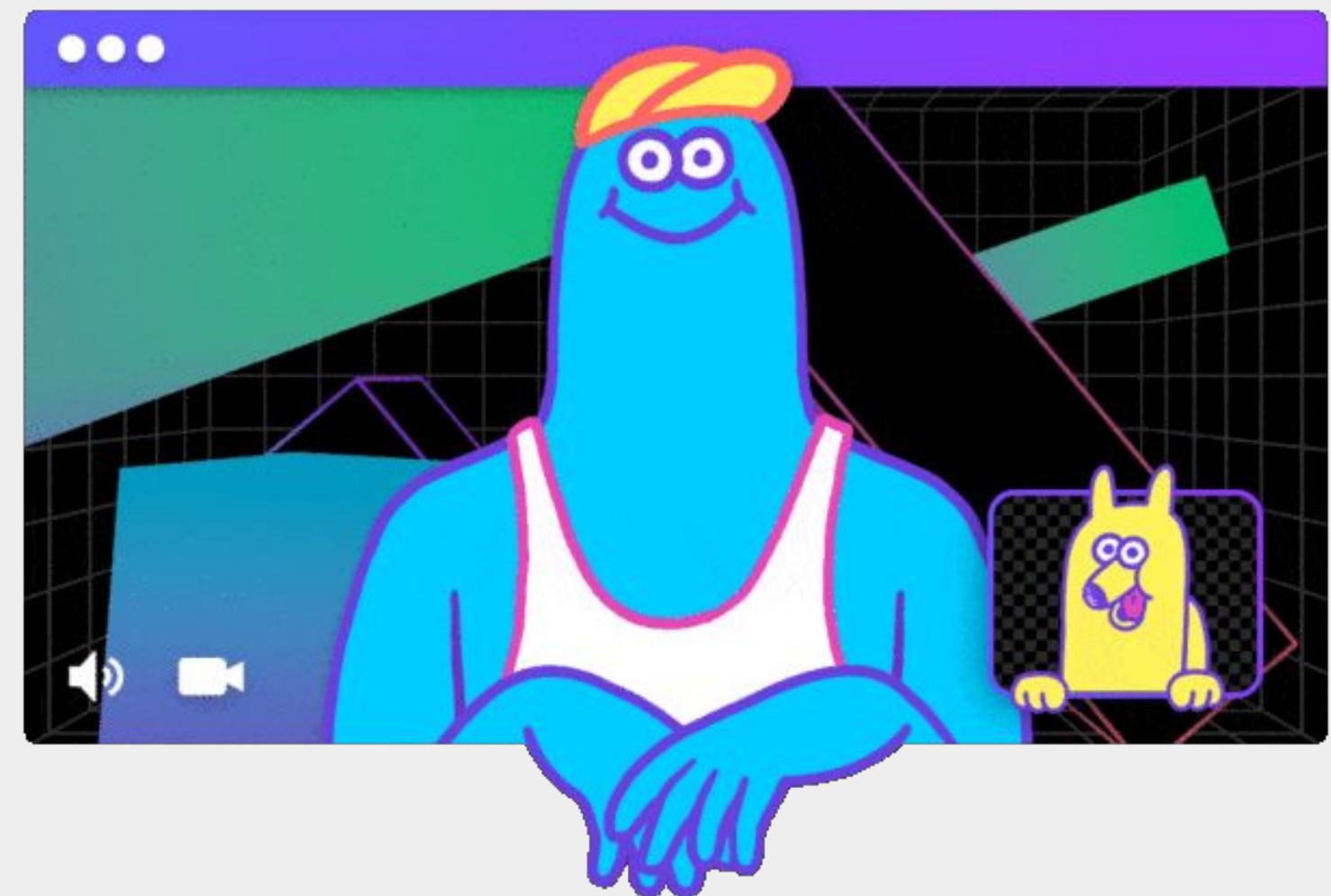
¿Qué ofrece? Una consola integrada, atajos y comandos que te ayudan a codear más rápido y extensiones muy útiles.

LINK DE DESCARGA PARA WINDOWS:

<https://code.visualstudio.com/docs/?dv=win64user>

HTML

¿Qué es? ¿Como funciona?



SPRINT 1

HTML



HTML es el lenguaje que permite estructurar las partes de nuestra página web y declarar sus elementos y atributos.

La traducción de las siglas es “lenguaje de marcado de hipertexto”.

Estructura básica HTML

El código HTML está armado a través de etiquetas anidadas.



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>JAVASCRIPT</title>
  </head>
  <body>
    <button>Aceptar</button>
  </body>
</html>
```

Aceptar →

Vemos un botón en pantalla del navegador.

Etiqueta HTML

Etiqueta button

Tiene etiqueta de apertura y cierre.



```
<button>Aceptar</button>
```

Etiqueta input

Sólo tiene una etiqueta que se autocierra.



```
<input />
```

Etiquetas anidadas



```
<body>
  <input />
  <button>Aceptar</button>
</body>
```

Atributos en etiquetas HTML

Son propiedades (con valores) de nuestros elementos o etiquetas HTML.

Ejemplo: placeholder y type



```
<input type="text" placeholder="Nombre" />
```

Nuestra primera web

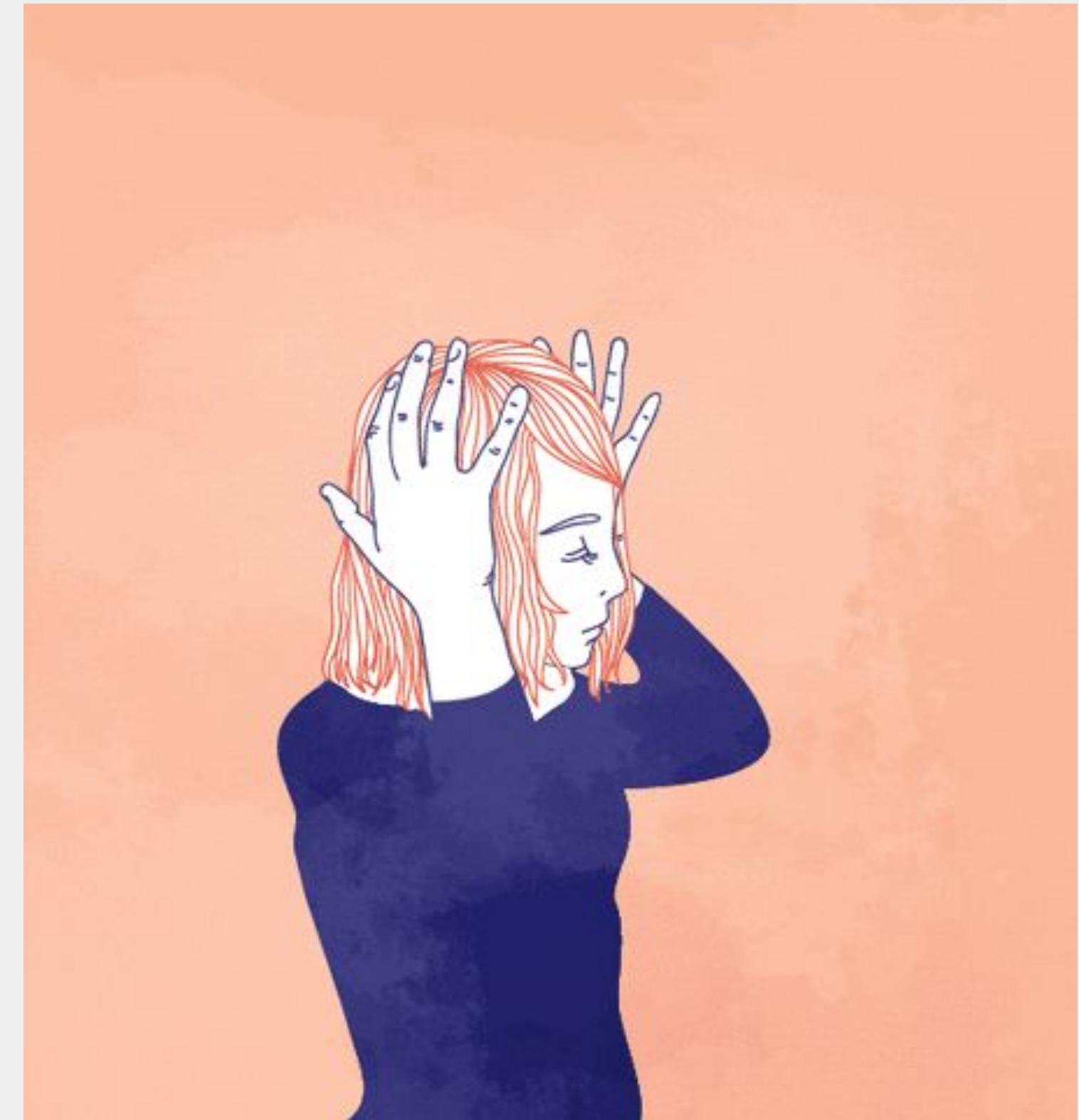
¿Qué son estas nuevas etiquetas?

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
  <p>Mi gato es muy gruñón.</p>
</body>
</html>
```

- **<!DOCTYPE html>** — el tipo de documento. Debe ser incluido para que todo funcione correctamente.
- **<html></html>** — Este elemento encierra todo el contenido de la página entera.
- **<head></head>** — Este elemento actúa como un contenedor de todo aquello que quieras incluir en la página HTML que *no* es contenido visible por los visitantes de la página.
- **<title></title>** — Establece el título de tu página, que es el título que aparece en la pestaña o en la barra de título del navegador cuando la página es cargada, y se usa para describir la página cuando es añadida a los marcadores o como favorita.
- **<body></body>** — Encierra *todo* el contenido que deseas mostrar a los usuarios web que visiten tu página, ya sea texto, imágenes, videos, juegos, pistas de audio reproducibles, y demás.

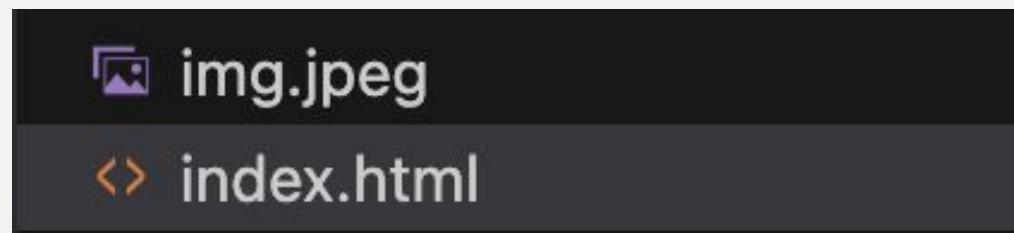
Archivos

Inserción de archivos en HTML.

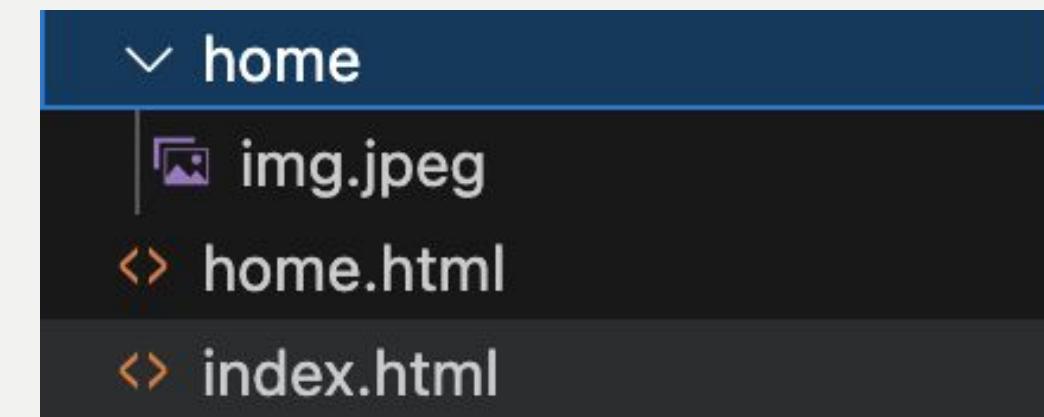


¿Cómo accedo al archivo que necesito, según el nivel de carpeta en el que se encuentra?

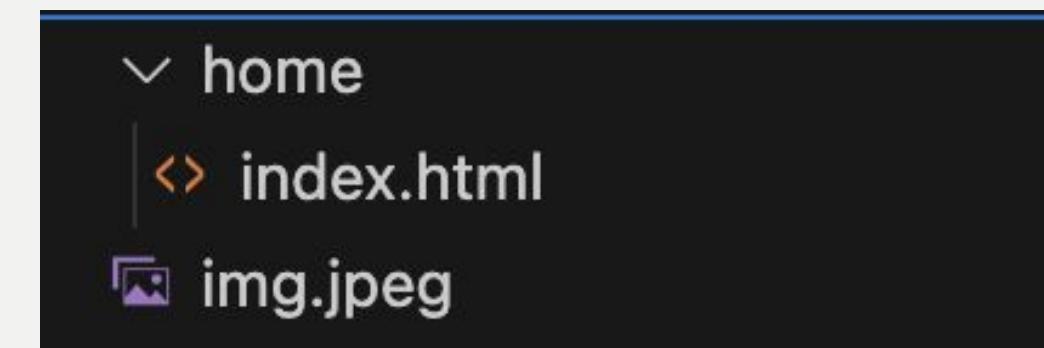
- Si está en el mismo nivel → ./img.jpeg



- Si está en un nivel superior → /home/img.jpeg



- Si está en un nivel inferior → ../img.jpeg



CSS

Cascading Style Sheet



SPRINT 1

CSS

**CSS es un lenguaje que da estilos
y diseño visual a una página web.**

**CSS significa hojas de estilo en
cascada.**



Sintaxis CSS

Una regla CSS consiste de un selector y un bloque de declaración. Dentro del bloque de declaración vamos a tener una propiedad y su respectivo valor. Los tipos de valores serán definidos según la propiedad que queremos cambiar.



El selector apunta al elemento HTML que se quiere estilar. El bloque de declaración está encerrado por {} y contiene una o más declaraciones separadas por un punto y coma (;).

Selector de elemento

Selector CSS por nombre del elemento HTML:

```
p {  
    text-align: center;  
    color: red;  
}
```

¡Todos los párrafos van a ser afectados por el estilo!

¡Yo también!

¡Y yo!

Selector de ID

Usa el atributo id de un elemento HTML para **seleccionar un elemento específico**. El id de un elemento es único dentro de una página, por lo tanto el selector por id es utilizado para seleccionar un único elemento.

```
#para1 {  
    font-weight: bold;  
    color: red;  
}
```

¡Hello World!

Este párrafo no va a ser afectado por el estilo.

```
<p id="para1">¡Hello World!</p>  
<p>Este párrafo no va a ser afectado por el estilo.</p>
```

Selector de clase

Selecciona un elemento HTML con un atributo de clase específico. Para seleccionar elementos con una clase específica se debe escribir un punto (.) seguido del nombre de la clase.

```
.paral {  
    font-weight: bold;  
    color: red;  
}
```

¡Hello World!

Este párrafo no va a ser afectado por el estilo.

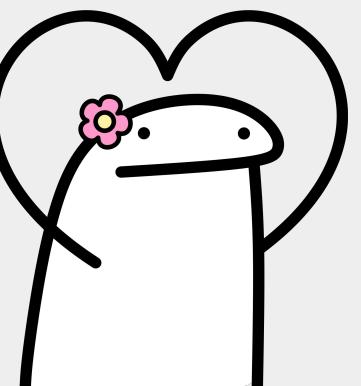
```
<p class="paral">¡Hello World!</p>  
<p>Este párrafo no va a ser afectado por el estilo.</p>
```

¿Como agregamos CSS al HTML?

Cuando un navegador lee una hoja de estilos formatea el documento HTML de acuerdo a la información que hay en la hoja de estilo.

Hay tres formas de insertar CSS:

- Estilos en línea.
- Estilos externos.
- Estilos internos.



Insertar CSS: Estilos externos

Con una hoja de estilos externa se puede cambiar el diseño de un sitio web cambiando solo un archivo.

Cada página HTML debe incluir una referencia al archivo de la hoja de estilos externa dentro de `<link>`, dentro del `<head>`.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="styles.css">
<title>Document</title>
```

Insertar CSS: Estilos internos

Una hoja de estilos internas puede ser usada si una página HTML tiene un único estilo. El estilo interno es definido dentro del elemento **<style>** dentro del **<head>**.

```
<!DOCTYPE html>
<html>
<head>
    <style>
        body {
            background-color: linen;
        }
        h1 {
            color: maroon;
            margin-left: 40px;
        }
    </style>
</head>
</html>
```

Insertar CSS: Estilos en línea

El estilo en línea puede ser usado para aplicar un estilo único a un único elemento.

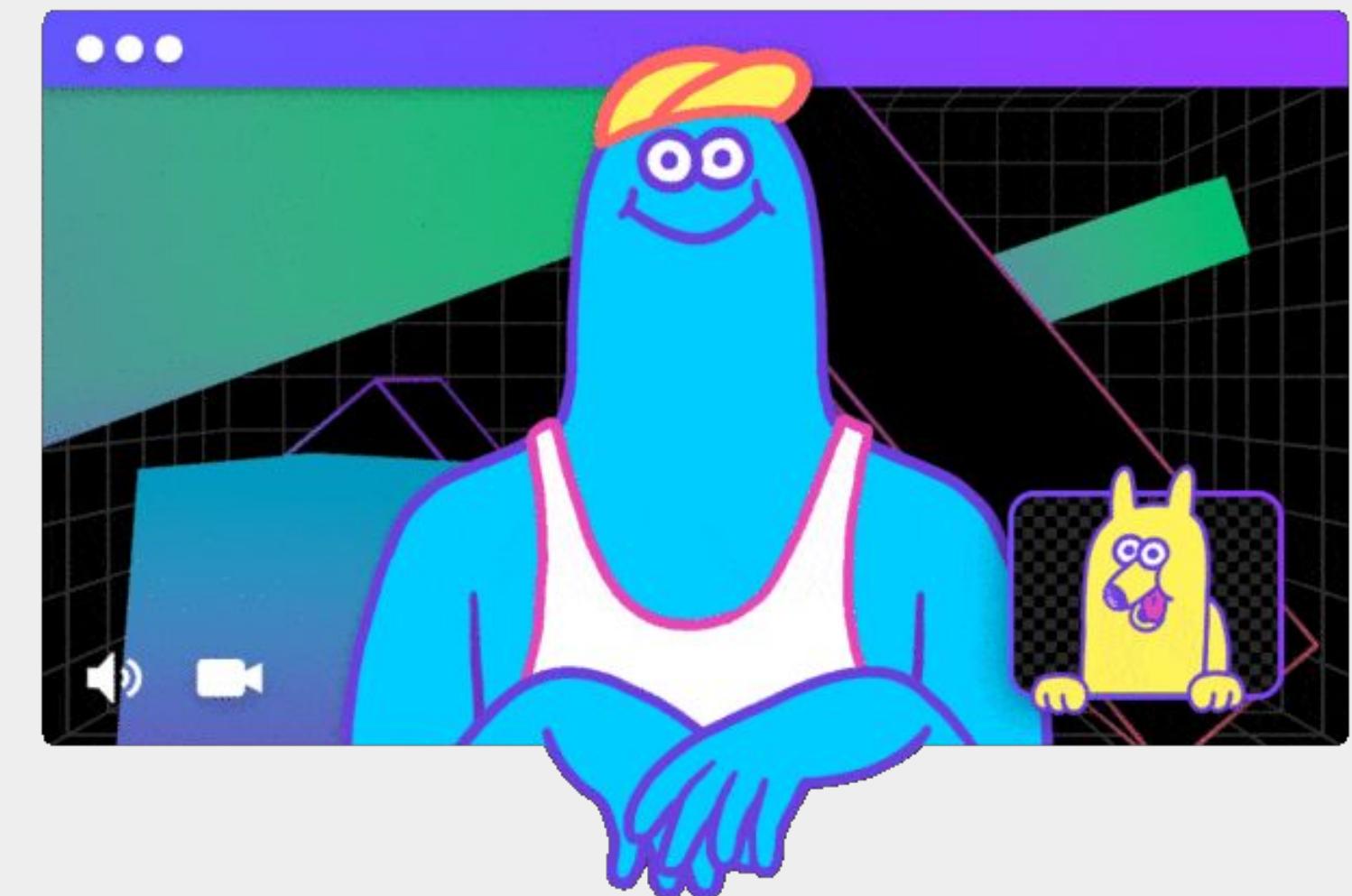
Para usarlo agrega el atributo style al elemento. El atributo style puede contener cualquier propiedad CSS.

```
<!DOCTYPE html>

<html>
<body>
    <h1 style="color:blue;text-align:center;">Soy un encabezado.</h1>
    <p style="color:red;">Soy un párrafo.</p>
</body>
</html>
```

Atributos CSS

Los más utilizados



Modelo de cajas

¿Que és?



El modelo de cajas

En css, el término “modelo de cajas” es usado cuando hablamos sobre diseño y disposición. El modelo de cajas es básicamente que los elementos html están envueltos por una caja. Consiste de contenido, padding, margin y borders.



Modelo de cajas

Contenido: Es donde se visualiza el contenido de nuestra página como el texto y las imágenes. Puede ser definido y/o cambiado por las propiedades **width (ancho)** y **height (alto)**.

Padding: Es el área que hay alrededor de un elemento y el borde.

Border: Es el borde que envuelve el padding y el contenido. Podemos modificarlo mediante border.

Margin: Es el espacio que hay entre un elemento y el siguiente. Contiene al border, padding y contenido. Afecta el espacio que va a tener el elemento en la página.

Width y Height de un elemento

Width: es el ancho de un elemento.

Height: es el alto de un elemento



IMPORTANTE:

Cuando se setean las propiedades **height** y **width** de un elemento CSS, estás seteando el **width** y **height** del **CONTENIDO**. Para calcular **height** y **width** de un elemento se debe incluir el **padding** y los **bordes**.



¡Manos a la obra!

SPRINT 1

Ejercicio

Crear una primera versión
de un e-commerce con
HTML y CSS sin Bootstrap.

1.

Crear una carpeta de nombre optativo.

2.

Crear un index.html, vincular un archivo
styles.css.

3.

Crear una primera versión de nuestro
e-commerce.

¡Trabajo en
equipo!

guayerd beta hub



En la escala de
pinturas...
¿Cómo les fue
trabajando en
grupo?



**¿Por qué es importante el
trabajo en equipo?**

Trabajo en equipo

Un equipo efectivo es aquel que se destaca por su capacidad para **colaborar de manera armoniosa y eficiente** para lograr objetivos comunes

2. Características de un equipo efectivo



Características

- Comunicación abierta y efectiva
- Confianza mutua
- Roles y responsabilidades definidas
- Colaboración activa
- Resolución de conflictos asertiva
- Flexibilidad y adaptabilidad
- Motivación y compromiso compartido

**¿Es lo mismo estar agrupados
a ser un grupo?**

Estar agrupados vs ser un grupo

- Ser un grupo implica organización, pertenencia, fines específicos.
- Varias personas en la fila de un supermercado no son un grupo.
- Los alumnos de una clase, podrían serlo.

Grupos

Individualismo

La confianza
es accesoria

Se busca competir
y destacar

Los problemas se
evitan

GRUPOS VS. EQUIPOS

Equipos

Interdependencia

La confianza
es un pilar

Se coopera y se
busca pertenecer

Los problemas se
afrontan y resuelven

3. Desafíos del trabajo en equipo



Desafíos que pueden surgir

- Comunicación deficiente
- Diferencias de opinión
- Falta de confianza
- Falta de liderazgo claro
- Distribución desigual de tareas



¿Cómo enfrentar esos desafíos?

- **Comunicación deficiente**

1. Fomentar una comunicación abierta y transparente.
2. Establecer canales claros de comunicación y retroalimentación.
3. Escuchar activamente a todos los miembros y respetar sus ideas.

- **Diferencias de opinión**

1. Valorar la diversidad de perspectivas y enfoques.
2. Buscar puntos en común y trabajar hacia soluciones consensuadas.
3. Establecer un ambiente de respeto y escucha activa.

¿Cómo enfrentar esos desafíos?

- **Falta de confianza**
 1. Construir confianza a través de la transparencia y la honestidad.
 2. Cumplir con los compromisos y demostrar responsabilidad.
 3. Fomentar un ambiente de apoyo y trabajo en equipo.
- **Falta de liderazgo claro**
 1. Designar un líder o establecer roles de liderazgo compartido.
 2. Definir claramente las responsabilidades del líder.
 3. Proporcionar capacitación y apoyo para el desarrollo de habilidades de liderazgo.

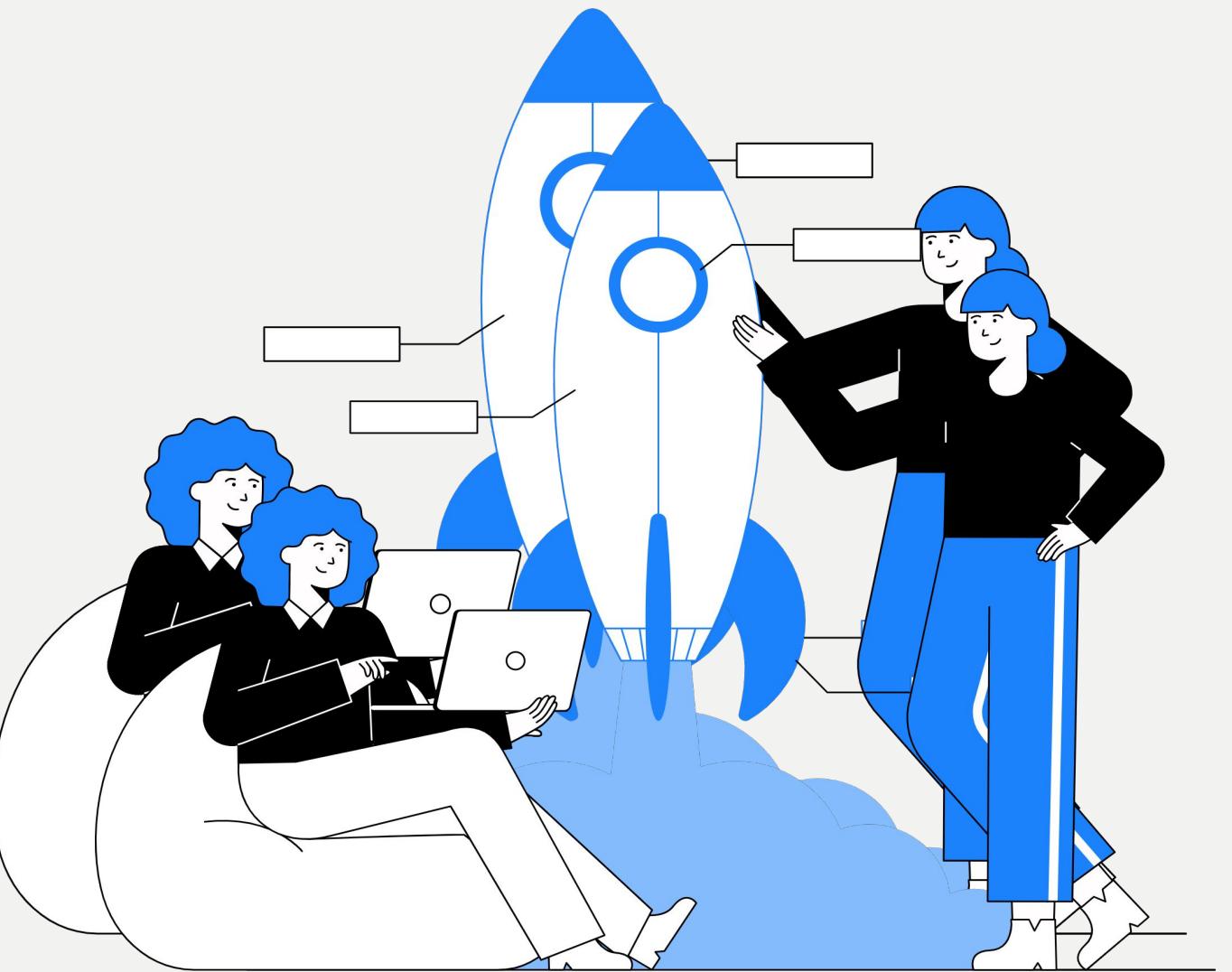
¿Cómo enfrentar esos desafíos?

- **Distribución desigual de tareas**
 1. Asignar tareas de manera equitativa y considerar las habilidades de cada miembro.
 2. Fomentar la colaboración en la asignación de tareas y objetivos.
 3. Establecer un sistema de seguimiento para garantizar la responsabilidad y el progreso.

¡Video debate!



4. Roles y responsabilidades



Roles dentro de un grupo

- La **interacción entre los miembros** del grupo da lugar a la aparición de ROLES.
- Un **rol es el papel que cada persona tiene dentro del grupo**.
- Es importante destacar que cualquier persona puede **ocupar diferentes roles en distintos momentos**.
- La salud y dinamismo de un grupo se basa en la **movilidad y flexibilidad de los roles**.



Importancia de definirlos

Evitar la duplicación de esfuerzos

- Al asignar tareas específicas a cada miembro, se evita la repetición de acciones y se maximiza la eficiencia del equipo.

Fomentar la especialización

- Al tener roles definidos, cada miembro puede enfocarse en sus áreas de experiencia y habilidades, lo que eleva la calidad del trabajo realizado.

Importancia de definirlos

Facilitar la coordinación

- Los roles claros permiten una mejor coordinación entre los miembros del equipo, asegurando que cada uno sepa qué se espera de ellos.

Promover la responsabilidad

- Al conocer sus responsabilidades, los miembros del equipo son más responsables de sus acciones y resultados.

Sentido de
identidad
colectiva



Nosotros en lugar de ellos

Roles
diferenciados

Todos los miembros **conocen su trabajo**

Modelos
estructurados
de comunicación

Líneas de **comunicación**

Normas

Reglas sociales que guían a los miembros en función de lo que hay que hacer y lo que no

Resumen: Tips para el trabajo en equipo

Comunicación abierta y efectiva

Confianza y respeto

Aprendizaje continuo

Roles y tareas claras

Resolución de conflictos

Apoyo mutuo

Colaboración efectiva

Flexibilidad y adaptabilidad

Buenas relaciones

¡Juguemos en equipo!



Nos dividiremos en equipos 

para intentar resolver este desafío



¡Llegamos al final!
Momento de retro