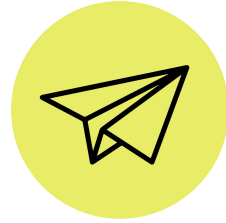


CÓDIGO PA' LANTE

Εidos
ALWAYS LEARNING

accenture



¡Les damos la bienvenida!
CLASE 11

03.

Clase de tutoría

En este encuentro nos dedicaremos a compartir recomendaciones y resolver dudas relativas al primer Proyecto.

¿Todo listo para empezar?



Consigna del proyecto

Repasamos la consigna.
¿Cómo empiezo el Proyecto?



CSS Responsive

Veremos cómo hacer
nuestra página responsive



Interacciones

Paso a paso para sumar
interacciones con JS



Consigna del Proyecto 2

Proyecto 2

Llegamos al final de nuestras clases de HTML, CSS y JS en Código Latam, sin embargo esto es solo el comienzo de sus caminos en el desarrollo web.

¡Es momento de aplicar todo lo que aprendimos en esta segunda mitad del curso!

Van a continuar trabajando sobre la página que maquetaron para el Proyecto 1, haciendo que ahora sea completamente responsive y dándole vida con un poco de JavaScript.



Etapas del proyecto

**Etapa
1**

Personalizar el
contenido

**Etapa
2**

Modificar el CSS para
hacer la página
responsive

**Etapa
3**

Aplicar JS para tener
interacciones en la
página

UPDATE...



Etapa 1: personalización

Personalizar la página que maquetaron para el Proyecto 1 **reemplazando el contenido ficticio por contenido propio.**

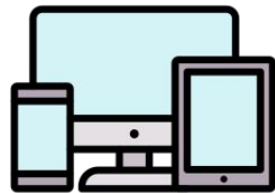


Etapas 2: responsive

Hacer que la página sea completamente responsive utilizando CSS.

En el Proyecto 1, aplicaron CSS siguiendo un diseño e indicaciones para tamaño mobile, es decir, para que el sitio se vea en celulares y tablets.

Ahora van a aplicar los media queries que vimos en la Clase 9 para que la página tenga un aspecto diferente pensado para adaptarse mejor a tamaño Desktop.



Etapas 3:

interactividad

Es momento de darle un poco de vida a la página usando JavaScript. Aplicaremos principalmente lo visto en la Clase 10: seleccionar elementos en el DOM, "escuchar eventos" y ejecutar funciones en respuesta a esos eventos.

Esta última etapa la resolveremos hoy en un livecoding para que puedan imitar los pasos.



¿Cómo empiezo?

Creando un nuevo entorno
de trabajo



Nuevo entorno de trabajo

Les recomendamos seguir estos pasos para iniciar el proyecto de forma ordenada:



- 1) Crear una nueva carpeta llamada:
“proyecto-2-cod-latam”
- 2) Copiar y pegar allí todos los archivos del Proyecto 1 (index.html, la carpeta /css con su styles.css y la carpeta /img con su contenido)
- 3) Antes que nada, realizar las correcciones indicadas en la devolución de la 1er entrega.

Nuevo entorno de trabajo



- 4) Al igual que hicimos con la 1er entrega, vamos a **iniciar un repositorio local y crear un repositorio remoto en nuestra cuenta de GitHub.**
- 5) Luego conectaremos nuestro repositorio local con nuestro repositorio remoto y a medida que vayamos trabajando, iremos actualizando los cambios.
- 6) La modalidad de entrega será a través de un formulario al igual que se hizo en la 1er entrega. Se deberá poder contar con enlace a GitHub Pages y el enlace al repositorio.



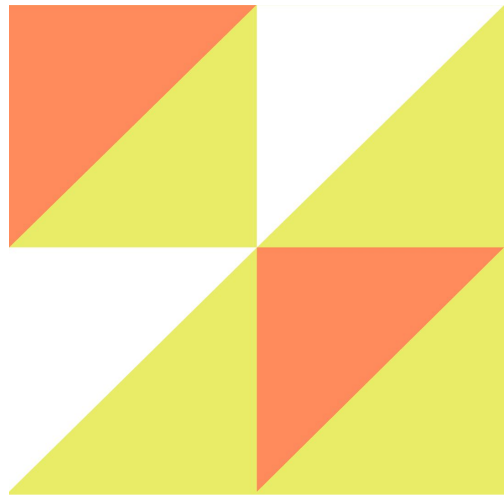
Repaso de Git y GitHub





Repaso de Git y GitHub

- 1) Dentro de la carpeta del proyecto hacer click derecho y seleccionar *"Git Bash Here"*.
- 2) Ejecutar los comandos: ***git init*** y luego ***git add .***
- 3) Luego ejecutar el comando ***git commit -m 'primer commit proyecto 2'***
- 4) Abrir el navegador e ir a su perfil de GitHub. Crear allí un repositorio público que se llame "proyecto-2-cod-latam".
- 5) Volver a la consola de git, escribir el siguiente comando para conectar los repositorios:
git remote add origin URL_DEL_REPOSITORIO
- 6) Por último, ejecutar por única vez el siguiente comando: ***git push -u origin master***

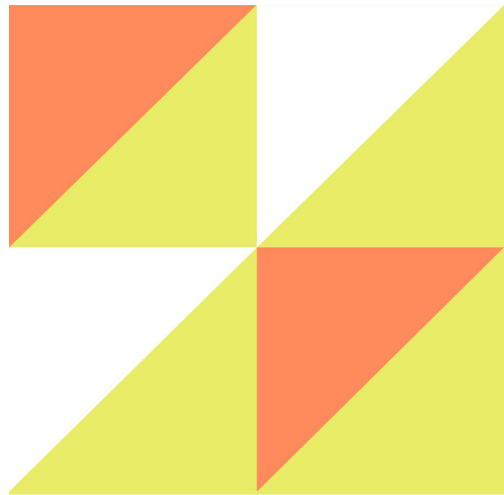




Repaso de Git y GitHub

Importante, los comandos *"git remote add origin ..."* y *"git push -u origin master"*, se ejecutan una sola vez, ya que son los que conectan el repositorio local con el remoto. Una vez conectados los repositorios, para actualizar los cambios que hagan en su código, solo tienen que escribir los siguientes comandos:

- 1) **git add .** (importante: espacio entre add y el punto)
- 2) **git commit -m "descripción de los cambios"**
- 3) **git push**





Empezamos a trabajar en la Etapa 1



Personalización

- 1) Abrir la carpeta del proyecto en el editor de código y usar LiveServer para abrir el **index.html** en el navegador.
- 2) Personalizar el contenido de la página.
 - Foto propia.
 - Reemplazar los textos con "lorem ipsum" por contenido propio.
 - Agregar su nombre completo en el <title>.
 - Modificar los enlaces de contacto por links propios.
 - En el enlace "Descargar CV" vincular un archivo PDF con su CV.
 - Cambiar paleta de colores y font





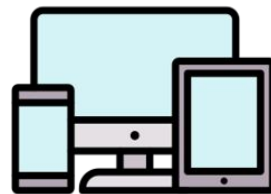
Empezamos a trabajar en la Etapa 2

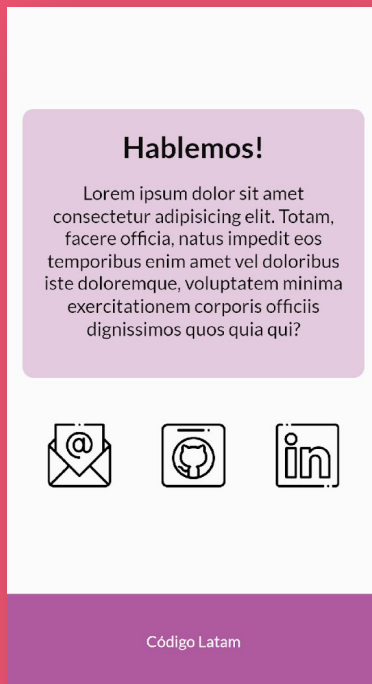




Responsive

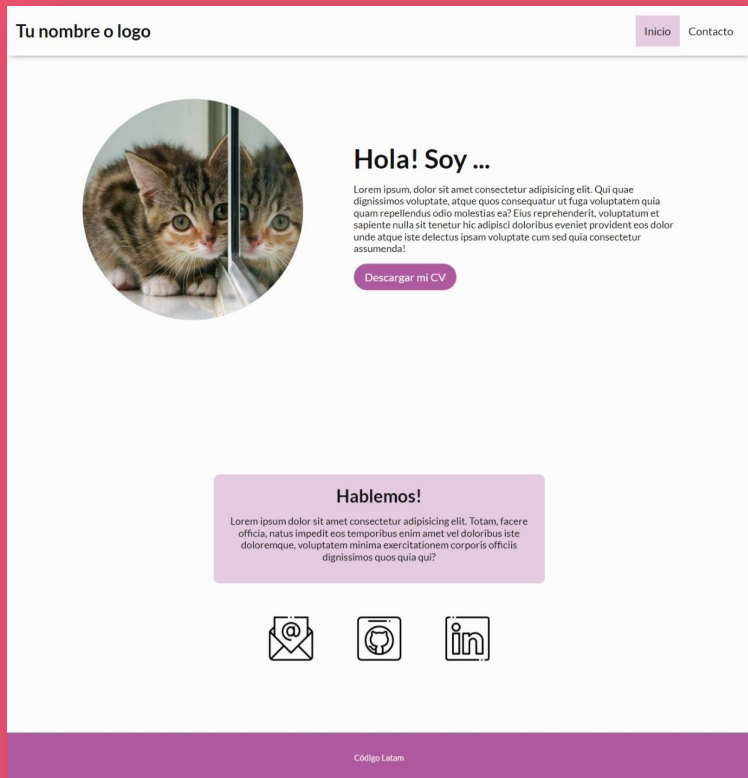
- 1) Una vez tengamos nuestro contenido personalizado, haremos cambios en el CSS para que se adapte mejor a pantallas más grandes.





Cómo se ve ahora el proyecto

Este era el diseño mobile del sitio que siguieron para el **Proyecto 1**, veamos ahora el diseño para tamaño Desktop.



Cómo debe verse en Desktop

Este es el diseño del **Proyecto 2**: deberán modificar el código CSS del Proyecto 1 para que se vea así en notebooks y monitores de escritorio. Para esto, deberán usar un **media query** que aplique estilos diferentes a partir de cierto tamaño.

styles.css

```
/* mobile */
section {
  display: flex;
  flex-direction: column;
}

h1 {
  font-size: 36px;
}

/* desktop */
@media (min-width: 1024px) {
  section {
    flex-direction: row;
  }

  h1 {
    font-size: 42px;
  }
}
```

¿Tengo que modificar todo el CSS?

No, dentro del media query solo escribirán algunos selectores y modificarán aquellas propiedades que necesiten para lograr los cambios.

Quizás solo tengan que escribir 4 o 5 selectores del media query, y cambiar alguna propiedad como flex-direction, margin, width, etc...

¿Cómo se qué propiedades modificar?

¡Paciencia! Todo esto es práctica.

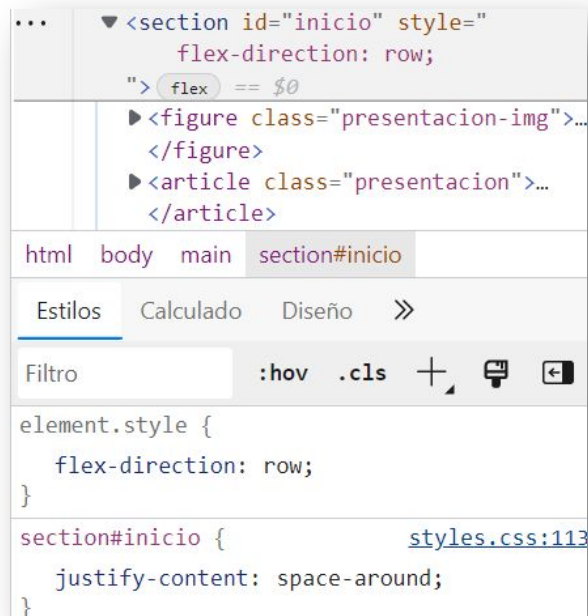


Les dejamos algunos consejos:

- Buscar las diferencias entre el diseño mobile y desktop, comenzar por las más notorias: cambios de dirección en como se organiza el contenido (horizontal/vertical)
- Trabajar sobre los elementos contenedores y modificar sus propiedades flexbox
- En la consigna se indican cambios específicos en tamaño de textos
- Revisar si es necesario modificar margins, paddings, width
- Cambios en alineación del texto
- Uso de max-width y unidades relativas (clase 5)

¡Usen mucho las devtools!

Las herramientas para desarrolladores del navegador son nuestras aliadas. Recuerden que pueden ir probando allí modificar propiedades CSS y luego pasarlas a su archivo de código.

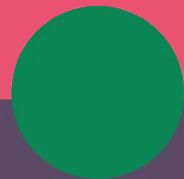




LiveCoding “CSS Responsive”

Pasemos asistencia

Etapa 3: JavaScript



Momento de agregar interactividad con JS

Algo habitual en los sitios web es que en el menú de navegación se destaque en qué sección nos encontramos.

Utilizaremos JS para aplicar una clase "active" al ítem del menú que fue clickeado y así poder agregarle estilos, como vemos en este ejemplo:



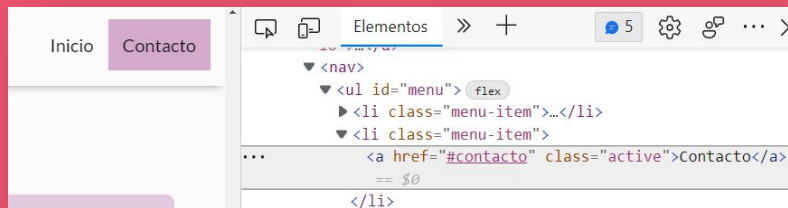


Trabajo en equipo

Antes de ver el paso a paso, es importante recordar que **HTML, CSS y JS trabajan en equipo.**

Por lo tanto, no basta solo con escribir el código JS que agrega una clase en el HTML.

Si no usamos esa clase para asignar estilos en nuestro código CSS, no veremos ningún cambio.



```
menuItems.forEach(function(item) {  
  item.addEventListener('click', function(e) {  
    const currentItem = document.querySelector('.active');  
    currentItem.classList.remove('active');  
    e.target.classList.add('active');  
  });  
});
```

Es similar a lo que hicimos en el Desafío Código de Clase 10, pero, además vamos a quitar la clase **"active"** del elemento que la tenía antes, así solo la tendrá un elemento.

JS

Primero que nada: creamos una carpeta "js" y un archivo .js, luego lo vinculamos al HTML.

- 1 – Seleccionar todos los items del menú
- 2 – Recorrer esa lista de nodos y detectar el click en cualquiera de ellos
- 3 – Cuando un elemento es clickeado, se ejecuta una función que realiza varias cosas:
 - Selecciona al elemento que ya tiene la clase active (por defecto, primero la tendrá Inicio)
 - Le quita a ese elemento la clase
 - Se la agrega al elemento que fue clickeado

```
li.menu-item:hover,  
li.menu-item a.active,  
a.btn:hover {  
  background: rgba(175, 89, 158, 0.5);  
  transition: background-color 0.3s;  
}
```

```
<nav>  
  <ul id="menu">  
    <li class="menu-item">  
      <a href="#inicio" class="active">Inicio</a>  
    </li>  
    <li class="menu-item">  
      <a href="#contacto">Contacto</a>  
    </li>  
  </ul>  
</nav>
```

CSS y HTML

Para que esto funcione, también debemos modificar el HTML y el CSS:

1 – Aprovechamos los estilos que agregamos para el :hover en los items del menu y aplicamos esos estilos también a la clase **active**.

2 – **Escribimos en el HTML la clase active solo en el primer enlace del menú**, así al abrir la página por primera vez, estará señalado.

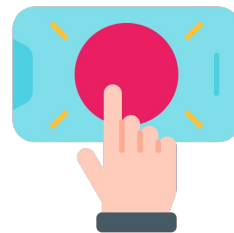


Bonus track

¿Se imaginan tener un botón en una web que cambie todos los estilos?

En el LiveCoding veremos cómo hacer esto también, Pero es opcional aplicarlo en el trabajo.

¡Empecemos!





LiveCoding JavaScript



Para tener en cuenta

Esta técnica de ***agregar una clase al <body>*** y así modificar estilos, puede aplicarse para muchas otras cosas, por ejemplo:

- Mostrar una ventana con información
- Aplicar un "dark mode" a un sitio
- Abrir y cerrar un menú de navegación oculto

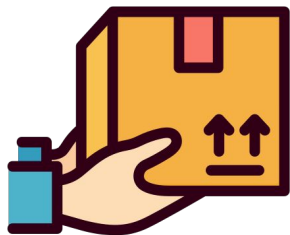
Luego del curso les sugerimos que sigan profundizando en JS y este tipo de proyectos pequeños es una gran forma de practicar 🚀



¿Cómo entregar el proyecto integrador?

Para entregar el trabajo

Te pedimos que completes el **formulario** que pasamos por Discord con tu:



- **Link de GitHub Pages:** nos servirá para ver la página terminada.
- **Link del repositorio de GitHub:** nos servirá para ver el código, hacer comentarios y correcciones.

Dudas y problemas con Git/GitHub/GitHub Pages: ¡las resolvemos en el foro de consultas!

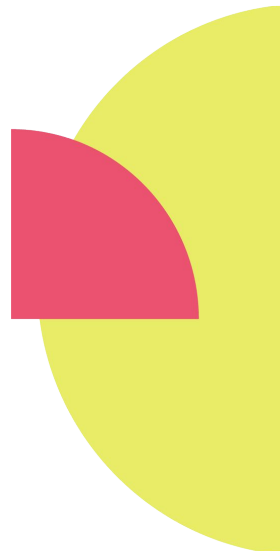
Grilla de evaluación



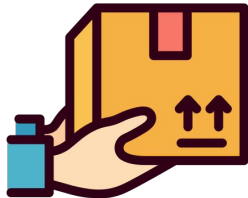
¿Aprobé?

El trabajo será corregido siguiendo una **grilla** que marca aspectos a revisar y condiciones de aprobación, esta grilla la pueden ver en el **campus**.

Para que se considere aprobado:

- Enviar ambos links (repo y GitHub Pages) por el **formulario** hasta la fecha acordada.
 - Los archivos CSS y JS deben estar vinculados al HTML correctamente.
 - Haber hecho las correcciones en HTML y CSS indicadas en la devolución del Proyecto 1.
 - La web debe poder visualizarse bien en tamaño mobile y desktop.
 - **Aplicar JavaScript para obtener la interacción en el menú de navegación que vimos en esta clase.** Si quieren agregar alguna otra interacción con JS es opcional 😊
- 

Lo más importante para aprobar es... ¡Entregarlo!



No esperamos que resuelvan en soledad, ¡estamos con ustedes!

Por eso es importante que aprovechen las instancias de resolución de dudas, devolución y correcciones.

¿Y si tengo dudas?

Parte importante del trabajo como desarrolladores también es buscar respuestas en la web, lo cual es una habilidad que iremos desarrollando.

¡A **CHEQUEAR CÓMO
VA A SER**

The slide features decorative geometric patterns on the left and right sides. These patterns are composed of various colored shapes including squares, triangles, and semi-circles in shades of yellow, green, pink, and orange. The central area is white and contains the main text.

Muchas gracias

CÓDIGO
PA' LANTE

Organizan



Socio estratégico



Con el apoyo de

