

ACÀMICA

# Agenda

---

Daily

Programo: DOM

Buenas prácticas

Break

Programamos: Datos al usuario + DOM

Programan: Javascript al wireframe (Parte 1)

Cierre



# DOM

Document Object Model (DOM) es la estructura de objetos que genera el navegador cuando se carga un documento. Hoy veremos cómo manipularlo y generar interacción con los elementos de la aplicación web.

# Daily



Daily



## Sincronizando...

### Bitácora



¿Cómo te ha ido?  
¿Obstáculos?  
¿Cómo seguimos?

### Challenge



¿Cómo te ha ido?  
¿Obstáculos?  
¿Cómo seguimos?

# DOM



# DOM

**Document Object Model** o DOM es la innovación que ha influido en el desarrollo de las páginas web dinámicas y aplicaciones más complejas. DOM nos permite acceder y manipular las páginas.



# Local Storage





## Local Storage

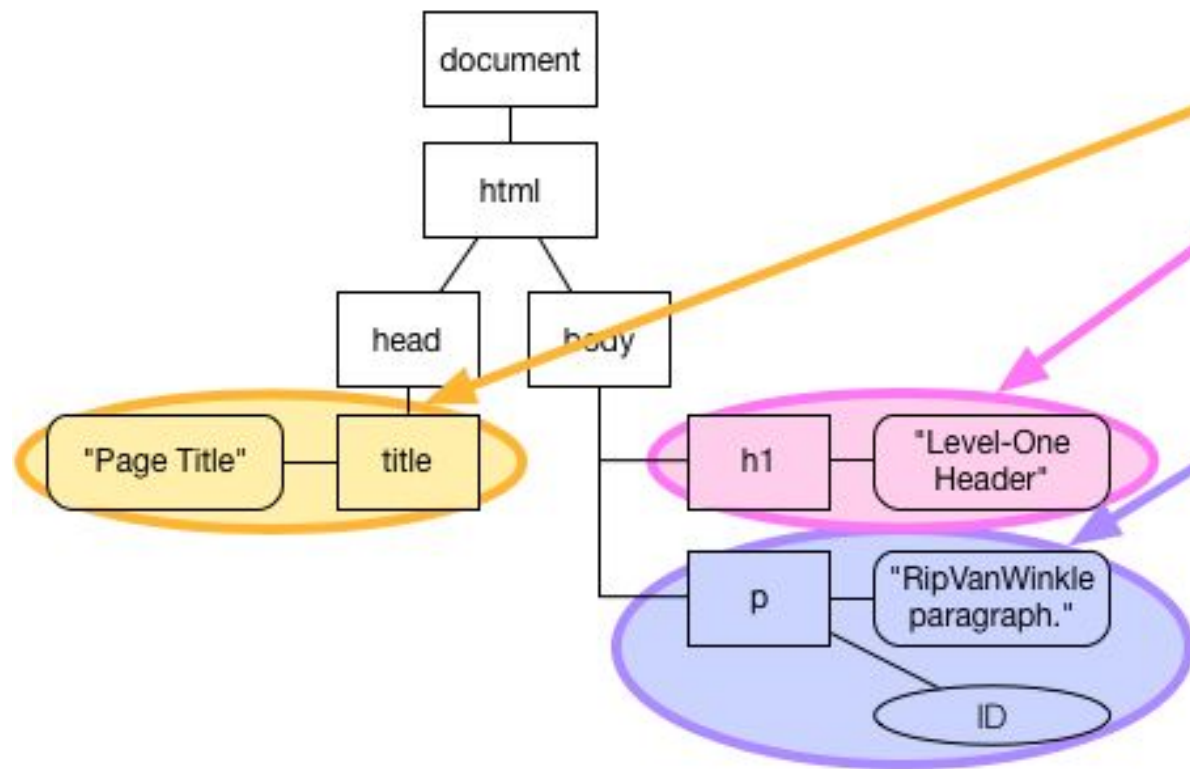
**Local Storage** es una funcionalidad que nos proveen los navegadores para guardar información y poder recuperarla en el momento que necesitemos.

Aunque el usuario reinicie el navegador la información que almacenemos allí seguirá quedando de manera persistente.

El usuario tiene la facultad de borrar la información que almacenemos pero debe ir a la configuración de su navegador para poder hacerlo.



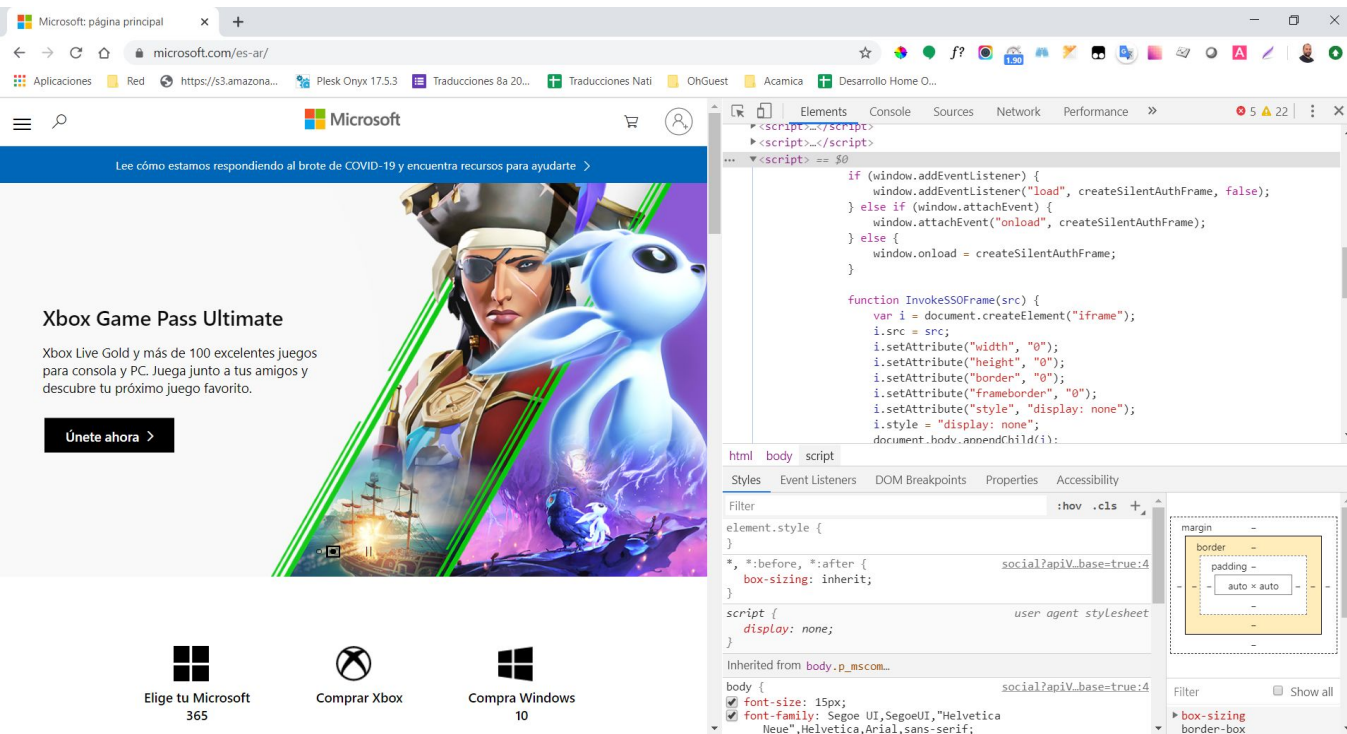
DOM tree for index.html



index.html

```
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>Level-One Header</h1>
    <p id="RipVanWinkle">
      RipVanWinkle paragraph.
    </p>
  </body>
</html>
```

# Encuentra la manipulación del DOM por todas partes:



The screenshot shows the Microsoft website with a promotion for Xbox Game Pass Ultimate. The page features a large image of a character and a blue rabbit. Below the image, there is a button that says "Únete ahora >".

The Chrome DevTools console is open, showing the following JavaScript code:

```
<script></script>
<script></script>
<script> == $0

if (window.addEventListener) {
  window.addEventListener("load", createSilentAuthFrame, false);
} else if (window.attachEvent) {
  window.attachEvent("onload", createSilentAuthFrame);
} else {
  window.onload = createSilentAuthFrame;
}

function InvokeSSOFrame(src) {
  var i = document.createElement("iframe");
  i.src = src;
  i.setAttribute("width", "0");
  i.setAttribute("height", "0");
  i.setAttribute("border", "0");
  i.setAttribute("frameborder", "0");
  i.setAttribute("style", "display: none");
  i.style = "display: none";
  document.body.appendChild(i);
}
```

The Styles pane shows the default user agent styles for the `body` element:

```
body {
  font-size: 15px;
  font-family: Segoe UI, Segoe UI, "Helvetica Neue", Helvetica, Arial, sans-serif;
}
```

The DOM Breakpoints pane shows the `body` element selected.

addEventListener  
attachEvent  
onload  
createElement  
setAttribute  
appendChild  
style  
src

# Programo

mentores/as



# Programo

Veamos en vivo como manipular el DOM mediante Javascript.



# Buenas prácticas



## Reduce el acceso al DOM

La parte más lenta de una aplicación es el acceso DOM, ya que cada acceso consume muchos recursos.

Utilízalo todas las veces que lo necesites, pero no abuses del recurso. Puedes ahorrar muchos accesos al DOM resolviendo problemáticas con HTML y CSS.

## **addClass vs style**

Cambia tus estilos agregando o quitando clases, ya que tiene menos consumo de procesador que al utilizar styles inline.



# Animaciones

Utiliza, en la medida que puedas, animaciones en CSS para optimizar tu aplicación al máximo.

Las animaciones realizadas en CSS son procesadas por la GPU (Unidad de Procesamiento de Gráficos), mientras que las realizadas en JS son procesadas por el CPU (Unidad Central de Procesamiento)

## Local Storage

Evita guardar información sensible en este tipo de almacenamiento. Contraseñas, tarjetas de crédito o cualquier otro dato que consideres sensible no debe almacenarse en los navegadores.

A close-up photograph of a white ceramic cup filled with a latte. The surface of the milk is decorated with intricate latte art, featuring a central heart shape surrounded by concentric, wavy lines. The cup sits on a matching white saucer. In the background, a white napkin and a silver spoon are visible, though they are out of focus. The overall lighting is soft and even.

**¡BREAK!**

---

# Programamos

todos/as



JAVASCRIPT

# Programamos

Programemos la siguiente estructura:

Pide al usuario estos 4 datos:

- Nombre y apellido
- Background color
- Font color
- Font size

Dibuja un cuadrado con el nombre y apellido en el centro y aplica las propiedades que solicitaste modificando el DOM.

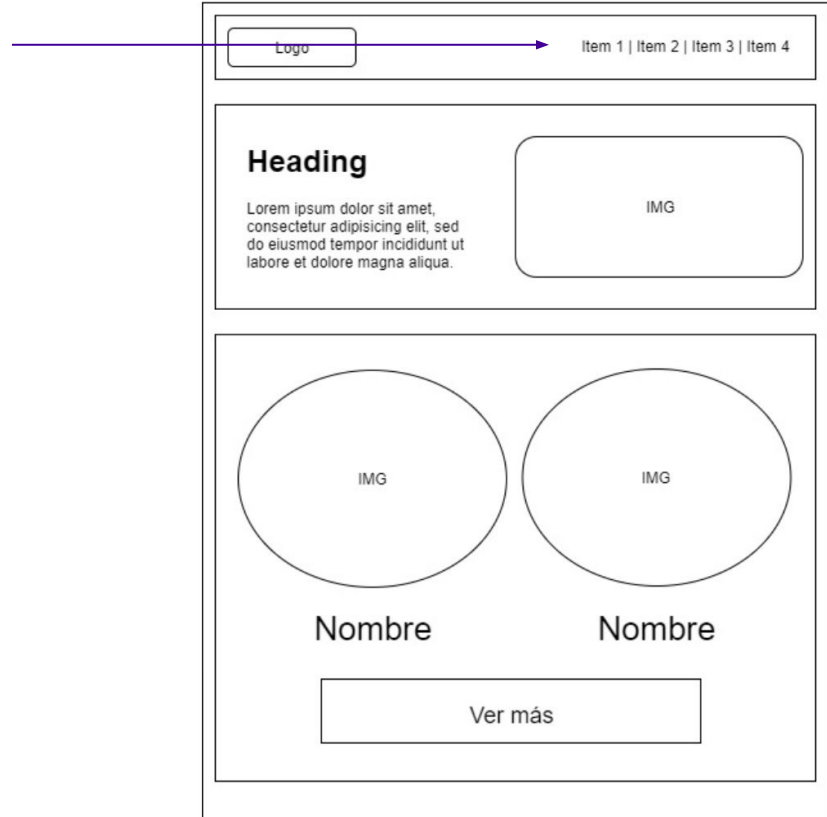
# Programan

estudiantes



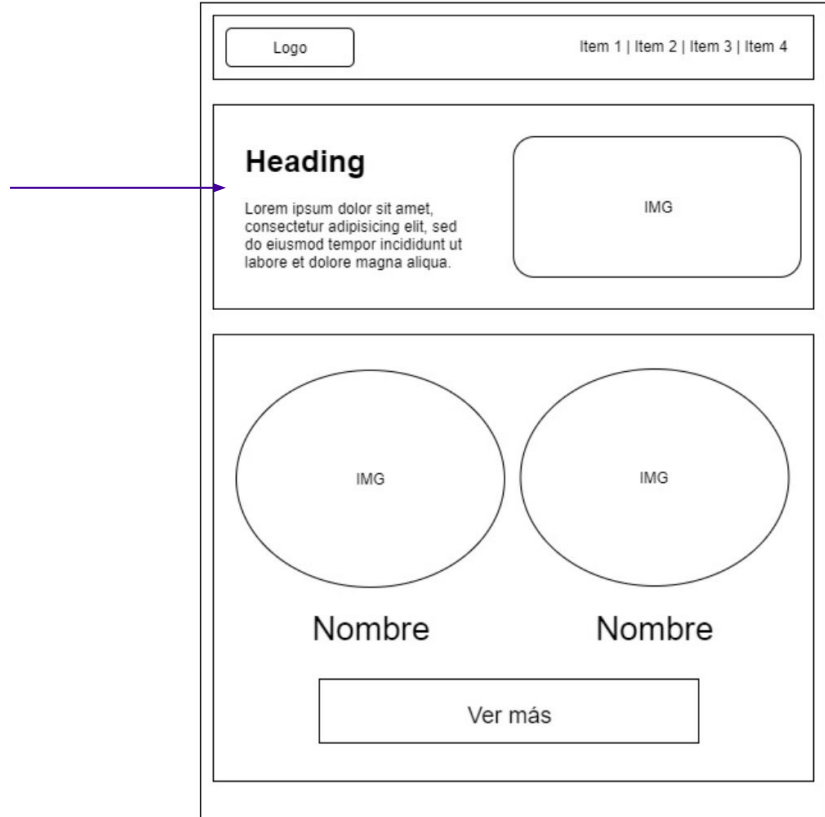
# Agregar ítem

Agrega un nuevo ítem al menú de navegación desde Javascript.



# Agregar negrita

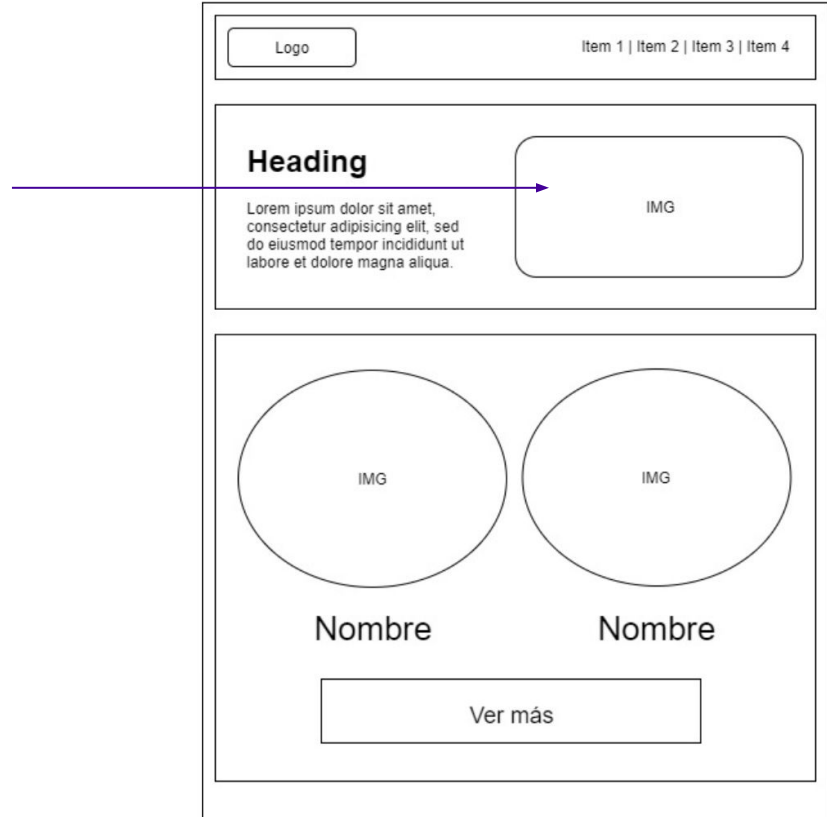
Agrega una negrita a una sola palabra dentro de tu párrafo.





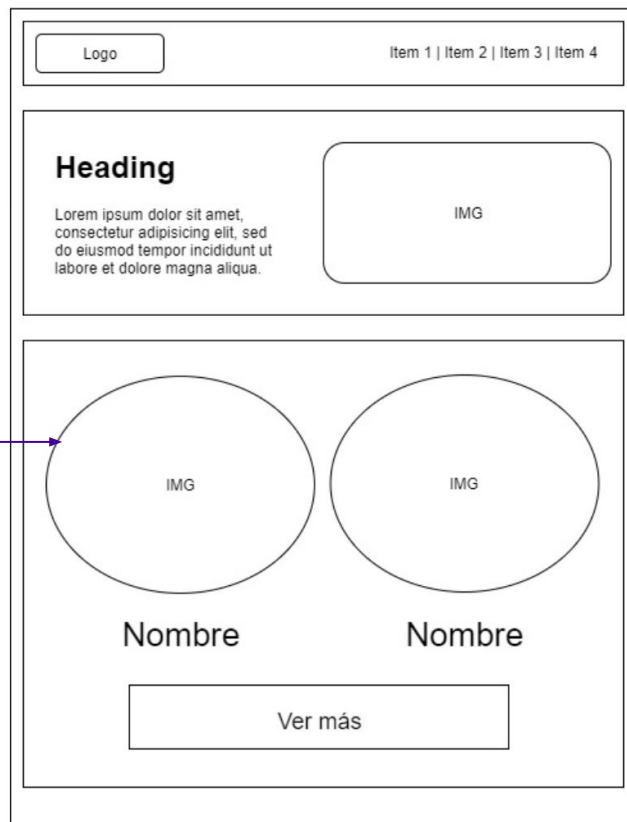
# Cambiar imagen

Luego de 10 segundos cambia la imagen. Intercala entre al menos 2 imágenes diferentes.



# Invertir el orden

Invierte el orden de estas dos tarjetas.



# Para la próxima

---

- 1) Termina el ejercicio del encuentro de hoy.
- 2) Lee la bitácora 26 y carga las dudas que tengas al Trello.
- 3) Resuelve el challenge.

En el encuentro que viene uno/a de ustedes será seleccionado para mostrar el ejercicio de hoy y otro/a mostrará cómo resolvió el challenge de la bitácora. De esta manera, ¡aprendemos todos/as de (y con) todos/as, así que vengan preparados/as!

ACÀMICA