

# Módulo 1: Ejercicio de evaluación final

---

## Recursos

- Antes de empezar, tenéis que crear un nuevo repositorio desde GitHub Classroom usando [este enlace](#). Una vez creado, debéis clonarlo en vuestro ordenador y en la carpeta creada debéis empezar a trabajar en el ejercicio (recuerda descargar este PDF a tu ordenador para que el enlace funcione).
- A continuación debéis descargar e incluir en el proyecto el [starter kit de Adalab](#).
- También necesitaréis varias imágenes e iconos que podéis descargar desde [esta carpeta](#).
- Y por último aquí tenéis los [diseños en zeplin](#) para guiaros en la maquetación. Si no puedes permisos para ver el proyecto, pídeselo a tu profesora.

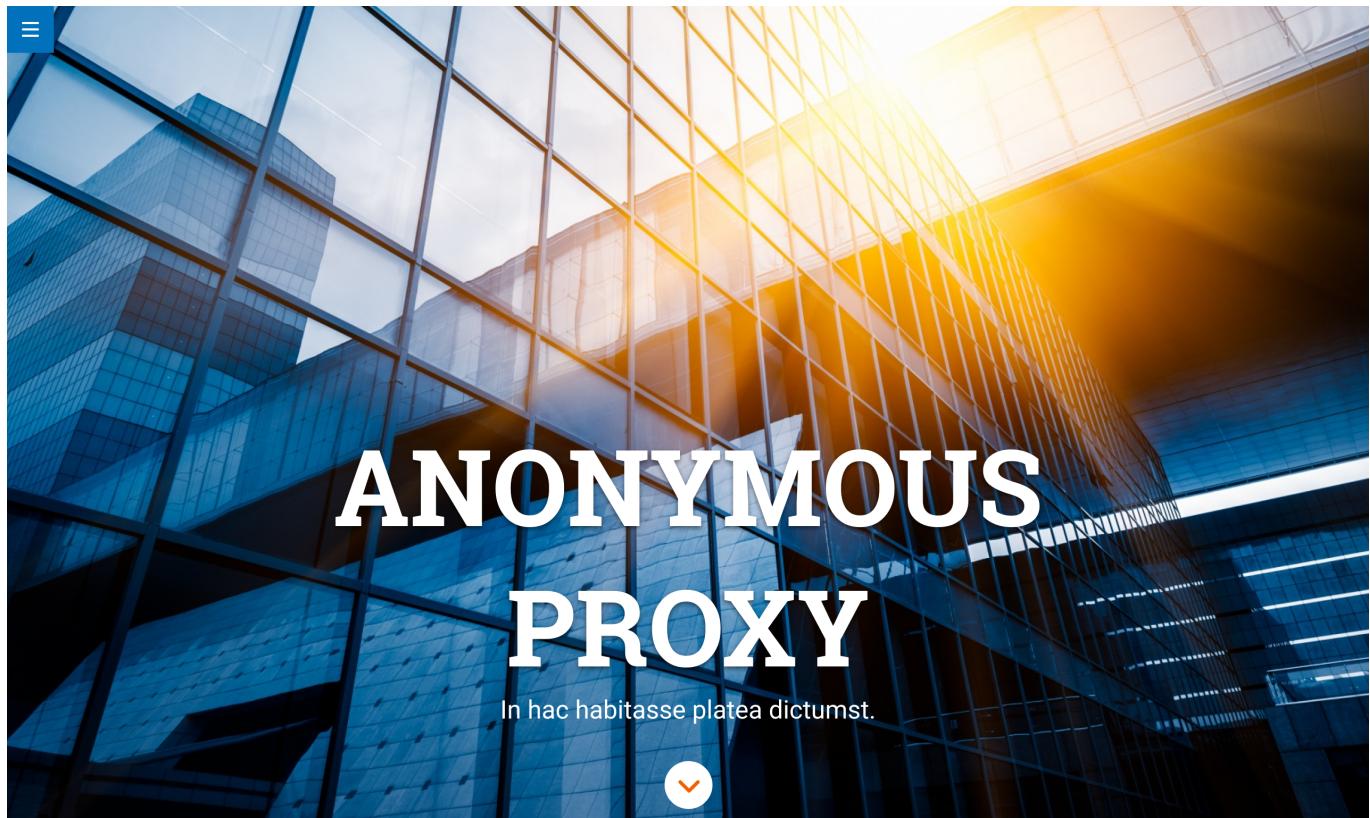
## Enunciado

El ejercicio consiste en desarrollar una página web de acuerdo a un diseño dado. Hay que resolver varios puntos:

- Usar Sass.
- Usar flexbox y CSS Grid.
- Usar media queries.
- Resolver algunas interacciones usando transiciones.

## Maquetación

En primer lugar debéis realizar la maquetación sobre un diseño dado:



Insomnia Tips

## Looking Through A Window

Donec accumsan, nulla ut volutpat porttitor, quam odio tempus felis, at luctus ex diam nec lectus. Vivamus semper sodales rutrum.

Go

## 3 Reasons To Purchase

### Ut sed metus

Lorem ipsum dolor sit amet,  
consectetur.

### Curabitur ac erat

Mauris mattis urna quis lorem molestie.

### Morbi dictum

Aliquam in dolor et dolor tincidunt.

More Reasons



©2022

we ❤ components

ARTICLES

BOOKS

EVENTS

REASONS

TIPS

TWITTER

INSTAGRAM

MASTODON.SOCIAL

1. El botón de hamburguesa (en la esquina superior izquierda) debe estar fijo en la parte superior de la pantalla y **no** debe desaparecer al hacer scroll. El icono de la hamburguesa debe ser un enlace a la página de Adalab. Este menú de hamburguesa no despliega ningún submenú.
2. Primer módulo (Anonymous proxy): debe estar maquetado con **flexbox** y debe ocupar el alto de la ventana del navegador.
3. Segundo módulo (Looking Through A Window): se puede maquetar usando las propiedades de CSS que se deseen.
4. Tercer módulo (3 Reasons To Purchase): los 3 elementos del listado deben estar maquetados con CSS **Grid** en todos los tamaños de pantalla.
5. Cuarto módulo (footer): se debe maquetar usando **flexbox**. Todos los textos de la columna "ARTICLES" y todos los textos de la columna "TWITTER" deben ser enlaces a la página de Adalab.

## Interacción

En total, hay varias interacciones que resolver:

1. El botón de flecha del módulo hero debe enlazar a la sección "3 Reasons To Purchase".
2. El botón de flecha del footer debe enlazar al inicio de la página.
3. Todos los links del pie deben ir a <https://adalab.es>.
4. En el **hover** de los botones ("Go" y "3 Reasons To Purchase") se debe incluir una transform que dejamos a vuestra elección (BONUS-> hacer la transform con transition, investigar la propiedad transition)
5. BONUS: hacer una pequeña animación en el botón del footer.(investigar la propiedad animate )

## Entrega

La fecha límite de entrega es:

- Martes, 11 de octubre a las 23:00h.

Solo debéis hacer commits y merges en la rama master de vuestro repositorio hasta la fecha límite. Si después del ejercicio queréis seguir trabajando sobre el ejercicio, lo podéis hacer en otra rama y no debéis mergearla hasta que los profesores os lo indiquen.

La evaluación solo se considerará terminada cuando:

- Esté publicada en GitHub Pages y esté funcionando, para lo cual tendréis que subir el código, también a la carpeta **docs/** del repositorio.
- El enlace a GitHub Pages esté en la página principal del repositorio, en la parte superior, al lado de la descripción.

## Normas

Este ejercicio está pensado para que lo realices de forma individual en clase, pero podrás consultar tus dudas con las profesoras y tus compañeras si lo consideras necesario. Ellas no te darán directamente la solución de tu duda, pero sí pistas para poder solucionarla. Aún facilitando la comunicación entre compañeras, durante la prueba no debes copiar código de otra persona ni acceder a su portátil. Confiamos en tu responsabilidad.

La evaluación es una buena oportunidad para conocer cómo estás progresando, saber qué temas debes reforzar durante las siguientes semanas y cuáles dominas. Te recomendamos que te sientas cómoda con el ejercicio que entregues y no envíes cosas copiadas que no entiendas.

Si detectamos que has entregado código que no es tuyo, no entiendes y no lo puedes defender, pasarás directamente a la re-evaluación del módulo. Tu objetivo no debería ser pasar la evaluación sino convertirte en programadora, y esto debes tenerlo claro en todo momento.

Una vez entregado el ejercicio realizarás una revisión del mismo con la profesora (25 minutos), que se asemenjará a una entrevista técnica: te pedirá que expliques las decisiones tomadas para realizarlo y te propondrá realizar cambios *in situ* sobre tu solución.

Es una oportunidad para practicar la dinámica de una entrevista técnica donde te van a proponer cambios sobre tu código que no conoces a priori. Si evitas que otras compañeras te den pistas sobre la dinámica de feedback, podrás aprovecharlo como una práctica y pasar los nervios con la profesora en lugar de en tu primera entrevista de trabajo.

Al final tendrás un feedback sobre aspectos a destacar y a mejorar en tu ejercicio, y sabrás qué objetivos de aprendizaje has superado de los listados a continuación.

## Criterios de evaluación

Vamos a listar los criterios de evaluación de este ejercicio. Si no superas al menos el 80% de estos criterios o no has superado algún criterio clave (marcados con asterisco \*) te pediremos que realices una re-evaluación con el fin de que termines el curso mejor preparada y enfrentes tu primera experiencia profesional con más seguridad. En caso contrario, estás aprendiendo al ritmo que hemos pautado para poder afrontar los conocimientos del siguiente módulo.

### General

- Usar una estructura adecuada de ficheros y carpetas para un proyecto web, y enlazar bien los distintos ficheros \*.
- Uso de control de versiones **con ramas** para manejar un proyecto de código.

### HTML

- Tener el código perfectamente indentado \*.
- Crear código HTML con sintaxis correcta, bien estructurado \*.
- Usar etiquetas HTML semánticas adecuadas para cada pieza de contenido \*.

### CSS / Sass

- Tener el código perfectamente indentado \*.
- Crear código Sass con sintaxis correcta, bien estructurado \*.
- Usar algunas características de Sass como variables, anidación y parciales.
- Usar código CSS que usa de forma intensiva selectores de clase. No usar selectores de etiqueta ni de id \*.
- Usar selectores de clase en inglés \*.

- Usar el modelo de caja de CSS de forma adecuada para especificar tamaño, relleno y márgenes\* .
- Usar estilos de texto y fondo para distintos tipos de elementos.
- Usar flexbox de forma adecuada para organizar elemento en cajas flexibles\* .
- Usar media queries para que los diseños se ajusten a distintos tamaños de dispositivo\* .
- Usar posicionamiento para emplazar elementos fijos y absolutos en la pantalla.
- Usar CSS grid para emplazar elementos usando una rejilla.
- Usar transiciones CSS para dotar de dinamismo a un proyecto web.

## Issues

- Haber resuelto las issues de la evaluación intermedia antes del inicio de la evaluación.

## ¡Al turrón!