



# Entrega 1

## IIC2513 Tecnologías y aplicaciones WEB

Primer semestre 2022 Profesor: Hernán Cabrera

Este, como todos los enunciados para el proyecto, dan guías y líneas generales para el desarrollo e implementación de su sitio WEB, pero ni entran en detalles ni puntos específicos de tal manera que ustedes demuestren su capacidad de trabajo en equipo, resolución de problemas y puedan **exponer sus ideas y alcances y así acordar los criterios de evaluación con su ayudante de seguimiento.**

***Salvo excepciones, tendrán total libertad en cuanto a la interfaz y la implementación de su trabajo.***

**Fecha límite de entrega: Miércoles 20 de abril, 23:00 horas**

### Indicaciones

Hemos visto que un sitio WEB completo (fullstack) se compone (siguiendo un patrón de diseño MVC) de una parte cliente y otra parte servidor. Ya saben de HTML, CSS, JS en el browser y DOM, todo de lado cliente, pues es el momento de avanzar en ese conocimiento y ponerlo en práctica. El objetivo de esta entrega es concretar el diseño visual y funcional de su aplicación en front-end (es decir la capa cliente, que es lo que ve y maneja el browser/navegador) y entregar una grupo de páginas estáticas HTML que serán parte de su sitio WEB. A esas páginas ustedes le incorporarán un diseño y estilo (CSS) e implementarán una navegación básica por las distintas páginas estáticas (¿Estáticas? Eso significa que sólo será por el momento algo “local”, todavía **no** iremos a un servidor, ni usaremos Heroku ni tampoco NodeJs ni KOAJs).

Así pues su sitio WEB (de sólo páginas estáticas HTML-CSS) debe incluir, entre otras cosas: imágenes, texto, secciones, instrucciones de su juego, preguntas frecuentes y una página especial de “nosotros” que relata cuál es el equipo de trabajo. Por supuesto pueden crear otras secciones con toda otra información que consideren relevante.

En particular, se deberá entregar al menos dos **tipos de archivos**: *html* y *css*. ¡Esto no significa que serán sólo dos archivos!, puede tener **varios para html y css**, que es lo recomendado. Al respecto, estos archivos a entregar deberán contemplar **al menos**:



1. Un archivo index.html como punto de entrada.
2. Un archivo con estilos para index.html.
3. Una página de inicio de su juego (aplicación WEB) que sea “amigable” es decir, de navegación obvia, con títulos, menús, sub páginas y secciones que orienten al jugador.
4. Las reglas detalladas del juego a nivel de usuario en la página, organizadas con secciones y de facilidad para lectura.
5. Una sección “acerca de..”, muchas veces conocida como “about...” donde explican qué es esto!, la versión, etc.
6. Una sección de “nosotros” donde se identifique el grupo (nombre) y se individualice a sus integrantes.
7. De ser necesario, deberán entregar también, en el directorio correspondiente, todas las imágenes y otros recursos requeridos para el despliegue correcto de su página.
8. Una barra de navegación que estará presente siempre en su página (en la parte superior) que permitirá ir a “home” o a cualquiera de las secciones que ustedes entreguen (recomendación, una sección-> una página HTML)

Adicionalmente DEBEN ENTREGAR un archivo **README.md**. (en el repositorio grupal) que incluya toda la información relativa al diseño de su HTML, consideraciones sobre las reglas (cambios que hayan implementado), detalle de las mejoras que introdujeron a su diseño visual con el HTML y CSS respecto a la entrega cero, cualquier tipo de supuesto, restricción, o información relevante para corregir su entrega.

***Todo lo anterior es lo mínimo.***

*¿Qué cosas se evaluarán positivamente para mejorar la banda (escala) del mínimo? (que es la C)*

- Programación, con un archivo JS, que incluya cierto comportamiento básico a nivel de la página, por ejemplo que haya alguna funcionalidad que permita cambiar el estilo “estándar” de la página a otro estilo, o que hay un selector que al elegir opciones genere cambios en la pantalla (por ejemplo vía programación cambie una foto, muestre u oculte mensajes, muestre información diferente ocultando o dejando visibles ciertas secciones...inventen!)
- Uso de Flex y Grid de manera adecuada.

*¿Se quedaron en blanco y quisieran ver un ejemplo de páginas más bien estáticas?*

<https://phylum.io/>

<https://mystaticself.com/>

<http://www.staggeringbeauty.com/> (Cuidado! tiene flash y script inline.. malo malo)



<https://loops.wannathis.one/>

*¿No se le ocurren buenas combinaciones de colores?*

<https://color.adobe.com/create/color-wheel>

*Pero, hay juegos basados en turnos que sirvan para algo en WEB?*

[https://blight.ironhelmet.com/#/menu/main\\_menu](https://blight.ironhelmet.com/#/menu/main_menu)

*¿Alguna guía rápida de JS-HTML-CSS?*

<https://htmlcheatsheet.com/>

<https://web.stanford.edu/group/csp/cs21/htmlcheatsheet.pdf>

<http://www.simplehtmlguide.com/cheatsheet.php>

<https://ilovecoding.org/blog/htmlcss-cheatsheet>

[https://fundamentals.generalassemb.ly/11\\_unit/dom-cheatsheet.html](https://fundamentals.generalassemb.ly/11_unit/dom-cheatsheet.html)

**La entrega de todos los archivos se hará en el repositorio de Github creado para su grupo.**

**NO se aceptarán:**

- Entregas por mail (ya sea al profesor o ayudantes)
- Entrega en otro sistema que no sea el que se ha provisto para estos efectos (el github entregado por el coordinador)

**Estructura de archivos:**

Una estructura apropiada para su repositorio tras esta entrega **podría** verse como el siguiente ejemplo:

```
-  
+-- documents/  
+-- src/  
| +-- assets/  
| | +-- styles/  
  
| | +-- imgs/  
| +-- views/
```



+-- index.html  
+-- .gitignore  
+-- README.md

## **Condiciones y restricciones**

1. En esta entrega no se espera que se entregue nada de programación. Sólo HTML y CSS junto al README
2. Se debe entregar una aplicación con estilos, es decir colores, fonts, control de despliegue (es decir ancho, alto de los elementos, etc).
3. Las reglas del juego pueden sufrir variaciones con respecto a la primera entrega, sin embargo deben señalar esto en su archivo README

## **Recomendaciones**

1. Piensen en distintas opciones de presentación y la mejor forma de abordar su tarea. No se queden con la primera idea que les venga a la cabeza.
2. Aprenderán mucho más si trabajan colaborativamente en su grupo, como equipo en lugar de repartirse el trabajo y realizarlo como unidades independientes.
3. Recuerden que hay una evaluación de pares la cual no es un castigo al que no trabaje sino más bien una protección a los que sí se esfuerzan.
4. Si hay problemas con su compañero(a) y no lo pueden resolver, comuníquese con el profesor o con el coordinador.
5. Diseñen muy bien las reglas, el entorno, los turnos, la resolución de empates y conflictos.
6. Planifiquen el trabajo para que les permita la colaboración entre los integrantes del equipo.
7. Pregunten y consulten, usen foro, colaboren entre ustedes (NO COPIEN). Los ayudantes están para apoyarlos.
8. **Trabajen con tiempo**, no esperen a último momento para **comenzar con la tarea o despejar dudas**.
9. Comiencen con una interfaz de usuario (UI) simple; **no es necesario que su aplicación se vea excepcionalmente "bonita" o "completa" desde el comienzo**. Una vez que cubran lo básico, la pueden ir mejorando.
10. No traten de resolver aún detalles específicos de integración, comunicación, consultas a BDD u otros temas de implementación ajenos al alcance de esta entrega.
11. Siempre podrán, justificadamente, cambiar alguna regla, mejorar algún aspecto del juego, etc.