



Entrega 2

IIC2513 Tecnologías y aplicaciones WEB

Segundo semestre 2023 Profesor: Hernán Cabrera

Los enunciados de proyecto dan instrucciones y aspectos genéricos para la construcción de su sitio WEB, pero sin entrar en detalles ni puntos específicos de tal manera que ustedes demuestren su capacidad de análisis, trabajo en equipo, resolución de problemas y puedan **exponer sus ideas y así acordar los criterios de evaluación con su ayudante de seguimiento**.

LEAN CON ATENCIÓN EL ENUNCIADO COMPLETO

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

Fecha límite de entrega: miércoles 27 de septiembre, 23:59 horas

Indicaciones

Hemos visto que un sitio WEB se compone, al menos, de una parte, cliente, llamada front-end, que se accede por medio de un browser y otra parte de back-end, que reside en algún servidor en alguna parte del mundo.

En las clases de cátedra, ayudantías y cápsulas hemos visto conceptos relevantes de HTML, CSS, JS en el browser y DOM y los seguiremos profundizando junto con un manejo introductorio de REACT.

El objetivo de esta entrega es transformar su diseño inicial (la entrega anterior) en un conjunto de archivos HTML con CSS y algo de JavaScript-DOM y de módulos React, que den vida de manera visual y funcional de su aplicación en **front-end** (es decir la capa cliente, que es lo que ve y maneja el usuario en el browser/navegador). Para esto, deberán entregar las páginas estáticas¹ 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.

¹ Por “estáticas” nos referimos a que su entrega sólo será, por el momento, algo de carácter “local”, todavía no iremos a un servidor remoto, ni usaremos NodeJs ni KOAJs.



Adicionalmente, diseñarán y entregarán los primeros módulos REACT que permitirán dar vida de manera dinámica a su parte cliente.

Su sitio WEB debe incluir, entre otras cosas: imágenes, texto, secciones, instrucciones de su aplicación, 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 tres **tipos de archivos**: *html*, *css* y *jsx* (React). ¡Esto no significa que serán sólo tres archivos! En rigor, deberán tener **varios para html y jsx**, 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 `.css`
3. La posibilidad de navegar entre distintas páginas HTML (usen la etiqueta de anchor para ello)
4. Una página de inicio de su aplicación WEB, que sea “amigable” es decir, de navegación obvia, con títulos, menús, subpáginas y secciones que orienten al usuario.
5. Al menos una página con diseño modular que le permita usar de manera preliminar React.
6. Las consideraciones, guía y reglas detalladas para el uso de su aplicación, a nivel de usuario como de administrador (al menos esos niveles se han de considerar).
7. Una sección “acerca de..”, muchas veces conocida como “about...” donde explican qué es esto!, la versión, etc.
8. Una sección de “nosotros” donde se identifique el grupo (nombre) y se individualice a sus integrantes.
9. De ser necesario, deberán entregar también, en el directorio correspondiente, todas las imágenes y otros recursos requeridos para el despliegue correcto de su sitio.
10. 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: cada sección diseñada debiese corresponder a una página HTML)
11. Deberán entregar Javascript de lado cliente (archivos `.js`) que permitan hacer algunas acciones básicas sobre sus páginas y que sean sólo de carácter local, por ejemplo, usar elementos HTML como un tag *selection* y que ante la selección de una opción cambie algo en la página, o el uso de botones que realicen acciones locales, etc. Recuerden que esto es parte integral de vuestro diseño de front.

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 descrito anteriormente es lo mínimo que se espera en su entrega.

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

- Uso de Flex y Grid de manera adecuada.
- Un diseño atractivo de las páginas (cuiden uso de colores, textos, ortografía, etc.)
- Un diseño modular adecuado (cuando corresponda) para el uso de React. Por ejemplo, la selección de elementos del juego (personajes, recursos, etc.) al estilo “cartas” o bien ya los primeros prototipos de su tablero.
- Que su página sea responsiva, es decir que se pueda ajustar a diferentes tipos de tamaños de dispositivos (pantallas). Para una guía sobre eso pueden ver:

https://developer.mozilla.org/es/docs/Web/CSS/Media_Queries/Using_media_queries

Importante: Usen lo que ya conocen de ingeniería de software y otros cursos en lo relativo a diseño de software. En particular HAGA SU TRABAJO ORIENTADO A LA REUTILIZACIÓN DE CÓDIGO, A CÓDIGO MODULAR Y A BUENAS PRÁCTICAS DE PROGRAMACIÓN

🔗 **GUIA DE ESTILO QUE USAREMOS EN EL CURSO:** Léanla, es **obligatorio** su uso. Aplicará tanto para JavaScript como para React

<https://github.com/airbnb/javascript>

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

<https://phylum.io/>

<https://mystaticself.com/>

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

¿No se le ocurren buenas combinaciones de colores?

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

¿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



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 (no olviden los módulos React en su folder "components")

```
-  
+-- documents/  
+-- src/  
| +-- components  
| +-- assets/  
| | +-- styles/  
  
| | +-- imgs/  
| +-- views/  
+-- index.html  
+-- .gitignore  
+-- README.md
```

Condiciones y restricciones

1. En esta entrega no se espera que se entregue nada de lado servidor, sólo funcionamiento de lado cliente (front)
2. Se debe entregar una aplicación con estilos, es decir colores, fonts, control de despliegue (es decir ancho, alto de los elementos, etc).
3. La parte React será sólo preliminar, el objetivo es que incorporen desde ya en su diseño el manejo de módulos y de creación de contenido dinámico cuando toque, en entregas futuras, juntar front con back
4. Las reglas de negocio de su aplicación 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 entrega. No se queden con la primera idea que le 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. Planifiquen el trabajo para que les permita la colaboración entre los integrantes del equipo.
6. Pregunten y consulten, usen foro, colaboren entre ustedes (**NO COPIEN**). Los ayudantes están para apoyarlos.
7. **Trabajen con tiempo**, no esperen a último momento para **comenzar con la tarea o despejar dudas**.
8. 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.
9. No traten de resolver aún detalles específicos de integración, comunicación, consultas a BDD u otros temas de implementación ajenos al alcance de esta entrega.
10. Siempre podrán, justificadamente, cambiar alguna regla de negocio, mejorar algún aspecto de su aplicación, etc.
11. Pueden usar ChatGpt.. PERO, deben indicar qué usaron de ChatGpt (que es lo mismo que hacer una "cita" de bibliografía) y deben señalar el código entregado y qué modificaron ustedes para su entrega, es decir, deben mostrar que valor agregaron.

Feliz 18, dicho sea de paso 😊

