*1RA ENTREGA DEL PROYECTO FINAL*

Deberás hacer entrega de tu proyecto donde aparezca: WIREFRAME/PROTOTIPO (wireframe de todas tus páginas para vista mobile + desktop) + HTML (uso de etiquetas contenedoras + etiquetas multimedia + etiquetas de texto) + CSS (con modificadores de texto, colores, listas y box modeling ).

**Componentes:**

1. Prototipo de la web
2. Estructura Inicial de la Web
3. Estilo inicial de la web

*Prototipo de la web*

* **Formato:** Archivo PDF o de Imagen
* **Objetivo del desafío:** El estudiante deberá prototipar la web para tener una idea clara del resultado al que quiere llegar.

**Incluir:**

* **Wireframe/Prototipo:** crear una versión que muestre cómo se verá el sitio cuando esté productivo.
* **Estructura de la web prototipada:** crear una estructura donde se organizan los elementos que van a estar en su web. El nivel de detalle no es importante, sino más bien las posiciones que los elementos van a tener y su tamaño aproximado.

*Estructura inicial de la web*

* **Formato:** Archivos HTML
* **Objetivos del desafío:**
  + **Maquetar la web:** El estudiante deberá utilizar los tags, en especial los semánticos, para desde el código describir la estructura de la web.
  + **Estructuración del contenido:** El estudiante deberá llevar su contenido a la estructura HTML haciendo uso de los tags que corresponden para el contenido a insertar.

**Incluir**:

* **Etiquetas semánticas:** utilizar los tags semánticos de HTML5.
* **Contenido:** agregar etiquetas que van a servir para denotar dónde va a haber contenido como imágenes, párrafos y titulares.
* **Páginas:** incluir las secciones de su sitio ya maquetadas con la estructura propia de cada página.

*Estilo inicial de la web*

* **Formato:** Archivo CSS
* **Objetivo del desafío:**
  + El estudiante deberá comenzar a darle estilo básico a una página de su sitio web (ej: index.html) aplicando con CSS las propiedades vistas hasta el momento para modificar textos, encabezados, img, colores, background y box modeling**\***.
  + El documento debe estar linkeado en las páginas del proyecto.

**Incluir:**

* **Estilo básico:** agregar al CSS las propiedades vistas previamente que sean pertinentes al diseño inicial que está planteando el estudiante.

**\*box modeling:** para los programas de los días **sábados**, no corresponde la entrega de box modeling en la primera entrega del proyecto final, se incluye en la segunda.

*RÚBRICAS DE EVALUACIÓN*

*1RA ENTREGA DEL PROYECTO FINAL*

| **Desafío: Prototipo de la web** | | | |
| --- | --- | --- | --- |
| **Criterios** | **Bajo**  Falta más profundización. Es confuso. | **Correcto**  Acorde pero con errores puntuales. | **Óptimo**  Es claro y pertinente. |
| Estructura clara | * No condice con lo que **podría** ser una web.  -El diseño del prototipo es difuso o no hay una navegación clara. * No se notan las diferentes secciones o elementos. | * La estructura del sitio condice con lo que será el sitio vivo y las secciones que podría contener. | * El prototipo representa el modelo vivo y se intuye una navegación clara de la que espera un usuario que navega un sitio web. |
| Diseño Simple | * No se intuyen secciones marcadas y definidas, sino que todo parece ser un gran torrente de información. * Se detiene demasiado en los detalles, tratando de emular lo que va a ser la web, cuando debería ser más representativo. | * Las secciones se identifican claramente y son diferenciables, pero tienen muy poco o nulo contenido. * El diseño del prototipo utiliza un formato del que luego se puede partir a un sitio web funcional. | * Las secciones son identificables y tienen una cantidad de contenido apropiada para esta etapa del diseño. * El diseño es simple y da espacio a que el diseño futuro pueda crecer a partir de él. |

| **Desafío: Estructura inicial de la web** | | | |
| --- | --- | --- | --- |
| **Criterios** | **Bajo**  Falta más profundización. Es confuso. | **Correcto**  Acorde pero con errores puntuales. | **Óptimo**  Es claro y pertinente. |
| Código prolijo | * Hace poco uso de tabulaciones, de manera errática, sin diferencias entre “padres” e “hijos”. * Los elementos comienzan en una columna (ya sea por la tabulación o por falta de nueva línea) y terminan en otra. * Uso de nuevas líneas erráticas. | * El uso de tabulaciones y/o nuevas líneas marca de forma correcta la jerarquía de padre/hijo pero hay inconsistencias; ya sea que es es errático, usa diferentes niveles de tabulaciones, usa diferentes tamaños de tabulaciones, o los elementos no cierran en el mismo nivel en el que empiezan. | * El uso de tabulaciones es óptimo para declarar las jerarquías padre/hijo. * Los elementos padre comienzan y terminan en la misma columna. * No hay inconsistencias en el uso de nuevas líneas o HTML. |
| Comentarios  <!-- --> | * Errores de tipeo a la hora de escribir correctamente el tag de comentario, o los nestea. * Se usan (en demasía) para ocultar código que el estudiante no quiere que el explorador muestre. | * Se usan demasiados comentarios (seguramente por la falta de tags semánticos). | * El uso de comentarios es correcto: se usan de forma pragmática para saber dónde comienzan y/o terminan secciones. |
| Tags HTML | * Tags mal escritos o deprecados en HTML5 (center, font, blink, etc). * Atributos mal escritos o deprecados (align, border, etc). * Repetición de atributos en un mismo tag. * Mal nesting. * Uso nulo de tags semánticos. * Tags hijo o padre que no están permitidos (li dentro de p, p dentro de ul, etc). * Repite IDs. | * Nesting acorde en general. * Errores mínimos a la hora de escribir tags. * Tags semánticos usados pero con problemas para comprender cuál sirve para cada caso. | * Buen nesting, entiende claramente el uso del mismo para darle jerarquía y estructura a los elementos. * Los tags y atributos están bien escritos. * Usa los tags semánticos de forma correcta para estructurar la página desde el HTML, entendiendo el uso de cada uno. |
| Estilo en el HTML | * Mal linkeo de hojas de estilo o fuera del head. * Usa el tag style para agregar estilo a la página. * Usa el atributo style para darle estilo al elemento. * Usa nombres para las clases de forma inconsistente (en algunos elementos kebab-case en otros camelCase y en otros todojunto) * Utiliza clases que no son descriptivas para lo que hace en el estilo. * Usa caracteres prohibidos en el atributo clase. -Usa IDs como si fuesen clases. | * Hojas de estilo bien linkeadas y en las mismas se encuentra todo el CSS. * Usa demasiadas clases innecesariamente, no las re-utiliza ni organiza correctamente. * Declara clases de CSS en el HTML pero éstas no aparecen en las hojas de estilo. | * Hojas de estilo bien linkeadas y en las mismas se encuentra todo el CSS. * Reutiliza clases para no re-definir reglas de CSS para tags con estilos iguales (como colores, fondos, etc). * Nombres de clases consistentes, ya sea camelCase o kebab-case. |
| Funcionalidad | * La web tiene enlaces rotos. * Las fotos no se cargan por errores en la ruta. * No hay enlaces para navegar por las diferentes páginas. * El usuario queda atrapado al no tener cómo volver a la home por falta de enlaces. * Se usan rutas absolutas para archivos de la web, incluso haciendo uso del protocolo file:// | * La web tiene enlaces que representan secciones aún no desarrolladas, por lo que estos no llevan al lugar correspondiente. * Las fotos están bien cargadas pero son de relleno (no tienen un propósito o funcionan como “placeholder”). * Hay enlaces que llevan a diferentes páginas y permiten retornar a la home. * Se usan rutas relativas para los archivos de la web. | * Los enlaces a las secciones funcionan. * Las páginas están correctamente interconectadas y el usuario puede navegar entre ellas. * Las imágenes están bien cargadas y el contenido es pertinente a la web (no simulado o placeholder). * El enlace a la home es el logo. * Las rutas relativas son correctas. |

| **Desafío: Estilo inicial de la web** | | | |
| --- | --- | --- | --- |
| **Criterios** | **Bajo**  Falta más profundización. Es confuso. | **Correcto**  Acorde pero con errores puntuales. | **Óptimo**  Es claro y pertinente. |
| Código limpio y prolijo | * El uso de tabulaciones es poco o es errático. * Usa diferentes formas de escribir las reglas de CSS: mezclando algunos la forma en la que escribe los selectores, algunos poniendo todas las líneas a sola línea. | * Es consistente el uso de tabulaciones y las declaraciones de reglas. * Métodos de tabulaciones poco convencionales o de una forma poco predecible para otro que mire el código. | * Consistente en el uso de tabulaciones, declaraciones de reglas y de espacios entre los elementos. |
| Entendimiento del CSS | * Usa solo IDs para dar estilo a los elementos. * Errores de CSS usando reglas que no corresponden al elemento, como por ejemplo un p con list-style-none. * Errores en los selectores:   + # en vez de .   + Errores al hacer selectores de tags   + Nesting de CSS   + Nula selección de etiquetas por anteponer un símbolo * Nombres de clases con poca consistencia o no son predecibles. | * Muestra algunos errores a la hora de hacer selectores o equivoca los estilos. * Le da estilo a elementos a través de selectores e IDs según lo requiera. * Las selecciones son correctas, pero poco óptimas, usando demasiados elementos para una regla de CSS. * Usa solo IDs para armar los selectores de CSS por desconocimiento de las clases. * No recicla código. * Hace uso de demasiados elementos. * Mal uso de comentarios. | * Recicla de forma óptima. * Usa clases para armar * Las reglas de CSS que utiliza son correctas para el elemento en el que las utiliza. * El código está dividido en secciones o porciones de código predecibles. * Le da estilo a elementos a través de selectores e IDs según lo requiera. |
| Diseño de la web | * Estructura demasiado desordenada y/o errática. (Aclaración: Dada la fase de aprendizaje del estudiante, aún no hay herramientas de CSS para estructurar el contenido de forma correcta). | * Uso de elementos para reflejar el diseño de la web prototipado originalmente desordenados por el tipo de tags elegidos. | * Uso correcto de los elementos básicos para reflejar el diseño web prototipado originalmente. |