*2DA ENTREGA DEL PROYECTO FINAL*

*Hacer entrega del proyecto (dos páginas a elección) adaptado a la vista mobile y la vista desktop agregando animaciones. Para el mismo podrán optar entre el uso correcto del framework o sin el uso del mismo. La entrega se deberá cargar a un repositorio de github público.*

**Componentes:**

1. Estructura avanzada de la web
2. Estilo avanzado de la web
3. Repositorio en Github

*Estructura avanzada de la web*

* **Formato:** Archivos HTML
* **Objetivo del desafío:** 
  + El estudiante deberá realizar una estructura del HTML prolija, limpia, fácil de leer y que no tenga errores en sus atributos o en sus valores.
  + El estudiante deberá agregar elementos HTML según su necesidad en armar contenedores o elementos web determinados, en base al framework elegido y la documentación del mismo.

**Incluir**:

* **Maquetado de la web:** Las estructuras maquetan a la web en base al framework elegido, haciendo usos de clases utilitarias para armar grillas, elementos web y estilos propios del framework, además del HTML de contenido. En caso de no elegir framework, los elementos deben respetar una cierta maqueta propia.
* **Páginas:** Todas las páginas tienen el contenido estructurado y el estilo linkeado. En caso de elegir un framework también tiene que tener agregadas las diferentes librerías de Javascript y CSS pertinentes al framework.

*Estilo avanzado de la web*

* **Formato:** Archivos CSS
* **Objetivo del desafío:**
  + El estudiante deberá crear archivos de CSS para darle estilo a su web.
  + El estudiante deberá agregar transformaciones, animaciones y/o transiciones para otorgarle dinamismo a la web en elementos que tengan interacción con el usuario.
  + El estudiante deberá hacer uso de selectores de CSS para poder darle su estilo a los elementos que ya vienen con su propia identidad del framework.
  + En caso de no haber usado el framework para maquetar, deberá aparecer la implementación para que el diseño se adapte a las vistas mobile y desktop.

**Incluir:**

* **Estilo avanzado:** Se le mejorarán los elementos interactivos con variaciones en sus diferentes estados, ya sea de la mano de transformaciones, transiciones y/o animaciones.
* **Estilo del Framework:** No todos los elementos del framework van a tener una estética que condice con el sitio en el que son implementados, por lo que se usará CSS para darles un estilo acorde.
* **Estructura de la web:** Usa etiquetas no sólo para armar contenido, sino para armar los elementos que van a conformar el layout de la web, los contenedores, etc.

*Repositorio en Github*

* **Formato:** Link al repositorio en Github donde está hosteado el proyecto
* **Objetivo del desafío:**
  + El estudiante deberá utilizar git de forma correcta para versionar su proyecto.
  + El estudiante hará uso de Github para brindar acceso al proyecto versionado.

**Incluir:**

* Se envían en el repositorio todos los archivos necesarios para visualizar correctamente la web.
* En el repositorio se muestran los commit que el estudiante usó para actualizar/versionar su código.

*RÚBRICAS DE EVALUACIÓN*

*2DA ENTREGA DEL PROYECTO FINAL*

| **Estructura avanzada de la web** | | | |
| --- | --- | --- | --- |
| **Criterios** | **Bajo**  Falta más profundización. Es confuso. | **Correcto**  Acorde pero con errores puntuales. | **Óptimo**  Es claro y pertinente. |
| Código prolijo | * El uso de tabulaciones y/o nuevas líneas marca de forma correcta la jerarquía de padre/hijo pero hay inconsistencias. * Se usan demasiados comentarios. * Comentan demasiado código para uso futuro o porque ya no lo va a usar. | * Uso de nuevas líneas y tabulaciones de manera prolija y consistente. * Usa los comentarios para documentar secciones de su HTML/CSS. | * Tabulaciones correctas y ordenadas, denotando jerarquía entre los elementos. |
| Tags HTML | * Errores mínimos a la hora de escribir tags. * Tags semánticos usados pero con problemas para comprender cual sirve para cada caso. * Falta de h1 en algunas páginas. * Más de un h1 en algunas páginas. * Las imágenes no tienen alt. | * El HTML no contiene errores en los atributos y tags. * Crea tags que envuelven a otros innecesariamente, ya sea porque no cumplen ninguna función o no se usan. * Todas las imágenes tienen alt. | * El nesting es óptimo, usando la menor cantidad de tags posibles. Uso de tags semánticos correcto y estructuración de la página desde el HTML. * El alt de las imágenes es pertinente y descriptivo. |
| Estilo en el HTML | * Usa nombres poco legibles para las clases. | * Clases correctas pero redundantes o irrelevantes. | * Nombres de clases consistentes, usa BEM. |
| 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 como 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 a todas las secciones en su navegación. * Las fotos están bien cargadas pero algunas son de relleno. * Hay enlaces que llevan a diferentes páginas y tiene como volver a la home. * No se usan rutas absolutas para los archivos de la web, sino relativas. | * Las páginas tienen enlaces funcionales. * Las imágenes tienen rutas relativas y correctas. * Las páginas están interconectadas correctamente. * Todas las imágenes son pertinentes al sitio y no hay de relleno. |
| Contenido | El contenido tiene errores ortográficos.  Hay contenido que no corresponde a la página donde está ubicado.  El contenido está desorganizado.  Hay poco contenido o está incompleto en algunas de las páginas.  La organización del contenido es equitativa pero mal distribuida a lo largo de la página. | El contenido no tiene faltas ortográficas o gramaticales.  Las páginas tienen una cantidad contenido apropiado y está bien distribuida.  El contenido es apropiado a la sección en la que está ubicado.  Imágenes demasiado pesadas o mucho más grandes del tamaño que corresponde. | La información está correctamente estructurada, usando los tags correctos para cada tipo de contenido, ya sean tablas, listas, titulares, párrafos o imágenes.  El contenido no está distribuido monótonamente y tiene varios niveles de lectura.  Las imágenes tienen un tamaño apropiado al que ocupan en el contenido. |

| **Estilo avanzado 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 | * Tabulaciones erráticas y poco predecibles. | * Uso de tabulaciones consistente. * Declaraciones de reglas y de espacios entre los elementos estructuradas correctamente. * Métodos de tabular poco convencionales. | * Uso de tabulaciones y correcto. * Hace buen uso de los selectores para evitar repetir código. |
| Entendimiento del CSS | * Errores a la hora de hacer selectores. * Usa reglas de CSS no pertinentes al elemento seleccionado. * No recicla código y lo repite. * Los selectores son innecesariamente precisos, previniendo una óptima reutilización del código. | * Recicla código de forma poco óptima. * Usa demasiados elementos en su selector. * Usa reglas de CSS pertinentes al selector. | * Expande sobre elementos que ya había creado con clases que los modifican. * Genera estilos que son fáciles de cambiar o transformar para diferentes tamaños de dispositivo. |
| Código de la estructura visual o layout | * Logra el layout de web haciendo uso de reglas CSS ineficientes como floats, o position. * Usa grids o flex para estructuras que se resuelven fácilmente con box-modelling. | * Uso de br apropiado, separando los párrafos de texto. * Los layouts son correctos pero repite el código incluso aunque las estructuras sean iguales. | * A través del uso de clases especiales o *helper*, logra layouts diferentes reutilizando código y no reinventando el layouts similares. * Utilización de flex y grid pertinente al tipo de layout a generar. |
| Diseño de la estructura visual o layout | * El diseño del layout de la web no es consistente a lo largo de las páginas. * Elementos de misma jerarquía son inconsistentes página a página. * La ubicación de elementos de navegación cambia de lugar. * El layout de la web no es intuitivo o fácil de navegar. | * El diseño del layout de la web es consistente a lo largo de algunas páginas pero no en todas. * El layout del sitio web es navegable. * Los elementos de la misma jerarquía, son consistentes a lo largo de las diferentes páginas. | * El diseño del layout es consistente página a página. * Los estilos definidos para los elementos se mantienen consistentes a lo largo de las páginas. * La interfaz web planteada por el layout es intuitiva y navegable. * Los elementos se pueden ver de manera óptima y no se pierde la información. |
| Diseño web atractivo | * Uso de transiciones muy lentas. * Uso de transiciones exageradas. * Elementos fuera que no pertenecen a sección sueltos a lo largo de la página. * Animaciones muy largas y molestas para la navegación. * Elementos no interactivos con transiciones que dan a entender lo contrario. * Uso de colores chillantes o con mucho contraste entre sí. * El texto no es legible. * La página ocupa todo el ancho del navegador lo que dificulta la lectura. * No hay una paleta de colores. | * Transiciones apropiadas y decorativas. * Solo los elementos interactivos tienen transiciones que les dan estilo. * Uso de colores correcto pero el texto no es legible. * La paleta de colores varía a lo largo de las páginas. * Hay un elemento contenedor pero es demasiado grande o demasiado chico. * El elemento contenedor no está centrado o alineado a nada. * Los elementos respetan la paleta pero tienen un diseño diferente página a página. | * Transiciones y animaciones apropiadas y decorativas moderadamente. * El contraste entre los colores es apropiado. Hay una paleta de colores y se respeta a lo largo de las páginas del sitio web. * El texto es legible. * Hay un elemento que evita que el contenido vaya hasta los bordes de la pantalla y está alineado. * No se genera overflow-x |
| Utilización de Frameworks | * Trae la librería en su proyecto y no hace uso de las herramientas. * Arruina el box-modelling o el sistema de grillas del framework con estilos desde su hoja de estilo. * No usa las clases de responsive ya dadas por la librería. * Usa los estilos por defecto de la librería en toda la web, no hay identidad propia a lo largo de las páginas. * Utiliza un theme descargado y realiza cambios mínimos. * Usa el framework solo para reutilizar componentes (carousel, menu hamburguesa, modal, etc) | * Recrea las clases que ya trae su librería innecesariamente. * Utiliza el framework pero hace uso de pocos elementos. * Superpone los estilos de su librería para darle su propio estilo. | * Maqueta reutilizando clases que nos provee el framework. * Personaliza el framework haciendo uso de las variables del mismo y no de superponer las clases previamente definidas por el framework haciendo uso de la cascada. |
| Media queries & Responsive | * El sitio web no es usable en dispositivos más pequeños que desktop. * Contiene elementos completamente fuera de cuadro que no se adaptan a los cambios de tamaño. * Contiene imágenes o bloques que superan el ancho del padre. * El texto es ilegible, ya sea porque es demasiado grande o muy chico. * El elemento contenedor no se adapta a diferentes tamaños. * Usando la librería de bootstrap no usa img-fluid. | * Hace uso de las columnas de bootstrap o de media queries propios para lograr responsividad. * Hay demasiados queries porque no usa los breakpoints provistos por bootstrap o el framework que eligió. * No usa un framework y hay demasiados queries por no definir sus propios breakpoint. * Hace selectores muy específicos para los cambios de tamaño de los elementos, como pueden ser las tipografías | * Uso de unidad rem a lo largo de los tamaños tipográficos. * El sitio web cuenta con una buena navegación en numerosos tamaños, en particular en mobile, laptop y desktop. * Usa los breakpoint de su framework para generar media query para los diferentes tamaños con los que trabaja. * Ante la falta de un framework, genera sus propios breakpoint y limita sus media query a esos breakpoint. |