



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**EMPLEO DE SOFTWARE SIMBÓLICO
PARA EL DISEÑO DE MATERIAL
DIDÁCTICO EN UN PROYECTO DE
MEJORA EDUCATIVA**

MATERIAL DIDÁCTICO

Que para obtener el título de
Ingeniero en Computación

P R E S E N T A

Mario Francisco Martínez Venegas

ASESOR DE MATERIAL DIDÁCTICO

Fís. Salvador Enrique Villalobos Pérez



Ciudad Universitaria, Cd. Mx., 2019

Contenido

INTRODUCCIÓN.....	3
Objetivos.....	3
Objetivos generales.....	3
Objetivos particulares.....	3
Marco conceptual.....	3
Front End.....	3
HTML 5.....	4
Animate.css.....	8
Flex Box.....	9
Bootstrap 4.....	25
Maple.....	31
Ventajas del cómputo simbólico.....	33
Desventajas del cómputo simbólico.....	33
DESARROLLO.....	34
Ciclo de vida del software.....	34
Modelos del ciclo de vida.....	35
Modelo empleado.....	37
Diseño e implementación de una página web.....	39
Versionado.....	47
Contenido académico.....	52
RESULTADOS.....	53
Estructura de la página.....	53
Detalles técnicos.....	53
jQuery.....	53
CSS3.....	53
Animate.css.....	54
Smooth Scroll.....	54
CONCLUSIONES.....	55
REFERENCIAS.....	57
Bibliografía.....	57
Ciberografía.....	57
APÉNDICE A.....	58
index.html.....	58

APÉNDICE B.....	70
Mockup – Versión 1.....	70
Mockup – Versión 2.....	70
Mockup – Versión 3.....	71
Mockup – Versión final.....	72
GLOSARIO.....	74

INTRODUCCIÓN

Objetivos

Objetivos generales

- Desarrollar e implementar un medio educativo para el apoyo a profesores y alumnos en la enseñanza/aprendizaje.

Objetivos particulares

- Crear material didáctico que se relaciona con materias de ciencias básicas como Probabilidad y Estadística, Cálculo Vectorial, Cálculo Diferencial, etc., mediante el uso de herramientas de cálculo simbólico como Maple para que el personal docente facilite la comprensión hacia los alumnos de diversos temas.
- El medio educativo podrá ser utilizado sin importar el dispositivo con el cual se desee visualizar su contenido, ajustándose a todo tipo de resolución.
- El material disponible dentro del medio educativo podrá ser consultado por el público en general.
- Adquirir experiencia básica al manipular el software Maple mediante el seguimiento de los ejemplos que son mostrados.
- La estructura del medio educativo estará conformada por un diseño original.

Marco conceptual

Front End

El Front-End es la parte del desarrollo web que se enfoca en el usuario, con lo que se puede interactuar y todo lo que vemos mientras navegamos en las diferentes páginas web en las que ingresamos. Cada web, busca causar una buena impresión y agradar al usuario, para este fin hace uso de HTML5, CSS y JavaScript. Causar una buena experiencia al usuario, inmersión y usabilidad, son algunos de los propósitos que busca un buen Front End. Hoy día existe una gran variedad de frameworks, preprocesadores y librerías que nos ayudan a cumplir dichos propósitos.

Dentro del Front End, la creatividad es el recurso más valioso, ya que se tienen que tomar en cuenta fuentes, colores, imágenes, sonidos, videos, animaciones, eventos y todos los recursos de los cuales disponga para crear sitios visualmente agradables y que se vean bien en todos los dispositivos y resoluciones¹.

HTML 5

HTML es el lenguaje de marcado estándar para crear páginas web. Sus siglas son el acrónimo de Lenguaje de Marcado de Hiper Texto. Describe la estructura de las páginas web usando marcado. Ya que HTML5 no es una nueva versión del antiguo lenguaje de etiquetas, y tampoco una mejora de esta ya antigua tecnología, HTML5 se define como un nuevo concepto para la elaboración de páginas web y aplicaciones en la que se combinan diferentes dispositivos móviles, almacenamiento en la nube y diversos trabajos de red.

La historia de HTML comenzó tiempo atrás con una simple versión de HTML propuesta para crear la estructura básica de páginas web, organizar su contenido y compartir información. El lenguaje y la web nacieron por un solo motivo; comunicar información por medio de texto.

HTML5 provee básicamente tres características: estructura, estilo y funcionalidad, es considerado el producto de la combinación de HTML, CSS y JavaScript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y JavaScript hace el resto⁵.

Las etiquetas HTML etiquetan partes de contenido como “head (encabezado)”, “p (párrafo)”, “Table (tabla)”, etc. Los navegadores no visualizan las etiquetas HTML, pero las usan para renderizar el contenido de la página.

¹ <https://devcode.la/blog/frontend-y-backend/> consultado en Mayo 13, 2018 14:09 hrs.

⁵ GAUCHAT, JUAN DIEGO. *El gran libro de HTML5, CSS3 y Javascript*. MARCOMBO, S.A., 2012, España, p. 1.

La estructura básica de una página web está compuesta mediante las siguientes líneas de código.

- `<! DOCTYPE html>` define que el documento es HTML5.
- El elemento `<html>` es la raíz de una página HTML.
- El elemento `<head>` contiene meta información sobre el documento.
- El elemento `<title>` especifica un título para el documento.
- El elemento `<body>` contiene lo visual del contenido de la página.
- El elemento `<h1>` define un encabezado largo.
- El elemento `<p>` define un párrafo.

Etiquetas HTML

Las etiquetas HTML son nombres de elementos delimitados por pico paréntesis: `<nombre_etiqueta>` El contenido va aquí...`</nombre_etiqueta>`. Las etiquetas HTML normalmente vienen en pares como `<p>` y `</p>`. La primera etiqueta en un par es la etiqueta de inicio, la segunda etiqueta es la etiqueta de cierre. La etiqueta de cierre está escrita como la de inicio, pero con una diagonal insertada antes del nombre de la etiqueta.

La siguiente imagen muestra la visualización de la estructura de una página Web:

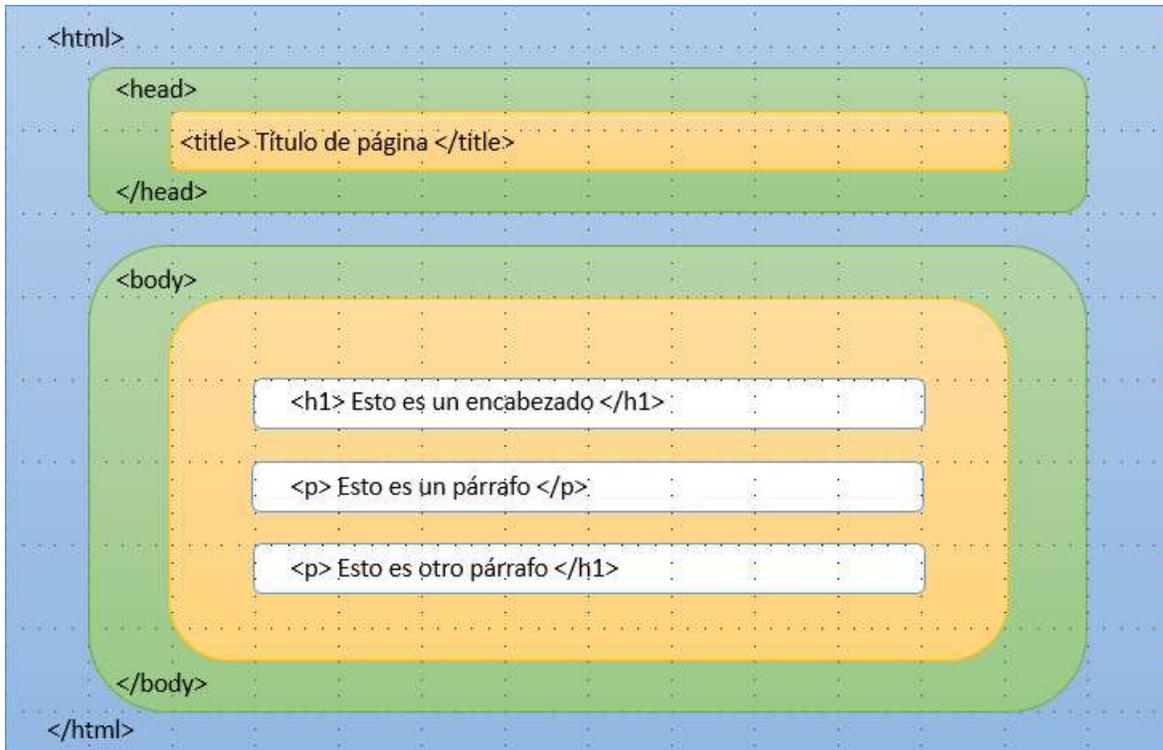


Imagen 1. Estructura de una página web.

El propósito de un navegador web es leer documentos HTML y mostrarlos, estos navegadores no muestran las etiquetas HTML, pero las utilizan para determinar cómo mostrarán el documento. El contenido dentro de la sección `<body>` es mostrada en el navegador.

La declaración `<!DOCTYPE>` representa el tipo de documento, y ayuda a los navegadores a mostrar las páginas web correctamente. Esta declaración podría sólo aparece una vez en la parte superior de la página (antes de cualquier etiqueta HTML). De esta forma, el modo estándar del navegador es activado y las incorporaciones de HTML5 son interpretadas siempre que sea posible, o ignoradas en caso contrario. La declaración `<!DOCTYPE>` para HTML5 es:

```
1 <!DOCTYPE html>
```

Luego de declarar el tipo de documento, se comienza a construir la estructura HTML. La estructura tipo árbol de este lenguaje tiene su raíz en el elemento <html>. Este elemento es el que envuelve al resto del código.

Encabezados HTML

Los encabezados HTML son definidos con las etiquetas <h1> a <h6>, <h1> define el encabezado más importante y por el contrario, <h6> define el encabezado menos importante.

Párrafos HTML

Los párrafos HTML están definidos con la etiqueta <p>.

Enlaces HTML

Los enlaces HTML están definidos con la etiqueta <a>, por ejemplo:

```
1 <a href="mipaginaweb.com">Esto es un enlace</a>
```

El destino del enlace es especificado en el **atributo href**. Los atributos son utilizados para brindar información adicional sobre los elementos HTML.

Imágenes HTML

Las imágenes HTML son definidas con la etiqueta . El archivo de origen (**src**), el texto alternativo (**alt**), el ancho (**width**) y la altura (**height**) son brindados como atributos, por ejemplo:

```

```

Elementos de Formato HTML

El lenguaje HTML también define elementos especiales para definir texto con un significado especial. HTML usa elementos como y <i> para salidas con formato,

como texto en negritas (***bold***) o texto en itálicas (***italic***). Los elementos de formato fueron diseñados para mostrar tipos de texto especial. La etiqueta **** de HTML define texto en negritas, sin ninguna importancia. El elemento **** define texto “fuerte”, es decir, con semántica agregada de importancia. El elemento **** define texto enfatizado, al igual que **** denota importancia en el texto.

Estilos HTML con CSS

CSS es el acrónimo de Cascading Style Sheets – Hojas de Estilo en Cascada. Describe cómo los elementos HTML van a ser mostrados en pantalla, papel o cualquier otro medio de visualización. CSS ahorra mucho trabajo, puede controlar el diseño de múltiples páginas web en una sola hoja de estilos. CSS puede ser agregado a los elementos HTML de tres maneras:

- En línea: utilizando el atributo de estilo en los elementos HTML. Se ocupa para aplicar un único estilo en un solo elemento HTML.
- Interna: utilizando el elemento `<style>` en la sección `<head>` del documento HTML.
- Externa: utilizando un archivo css externo. Con una hoja de estilos externa, se puede cambiar el diseño de un sitio web completo cambiando un solo archivo. Para utilizar una hoja de estilos externo, se debe de agregar un enlace en la sección `<head>` de una página HTML (`<link rel="stylesheet" href="ubicacionArchivo.css">`)

La forma más común para agregar CSS, es manteniendo archivos CSS por separado⁶.

[Animate.css](#)

El código del proyecto involucra el uso de una de las librerías que permiten crear animaciones CSS de manera sencilla. El archivo se llamada `Animate.css`⁷, el cual

⁶ <https://www.w3schools.com/html/>

⁷ <https://daneden.github.io/animate.css/>

contiene un gran número de animaciones para ser aplicadas en los distintos elementos HTML con los que cuenta el proyecto. Al entrar al sitio oficial de la librería podemos encontrar tres elementos principales; el primero (nombre de Animate.css) que refleja la animación seleccionada, la segunda es el tipo de animación y finalmente la tercera es el botón para dar inicio a la animación. La siguiente imagen muestra estos tres elementos:

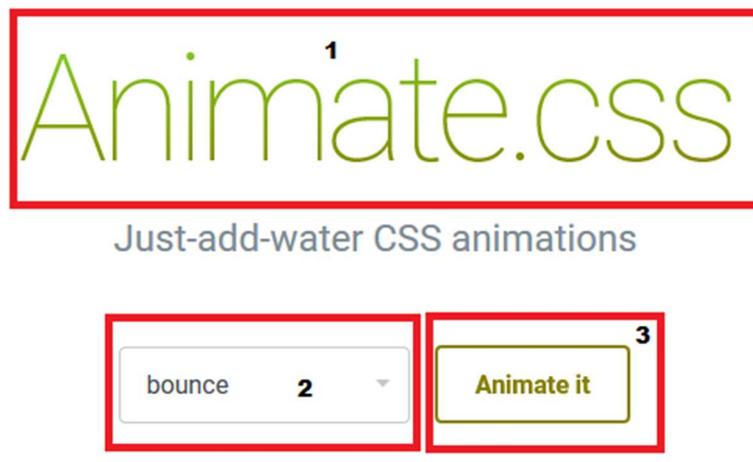


Imagen 2. Elementos de la página oficial de la librería Animate.css.

Flex Box

FlexBox es un módulo de diseño para CSS3 (Cascading Style Sheets - Hojas de Estilo en Cascada) que resuelve la forma en el que creamos sitios con *Responsive Design* o Diseño Responsivo, es decir, sitios que se adaptan a las distintas resoluciones de pantallas que presentan los dispositivos que hay en el mercado ya que, mediante el uso de cajas flexibles evitamos el uso de la propiedad **float**.

Antes de CSS3 había cuatro formas de crear una caja: *block*, *inline*, *table* y *position*. Ahora la caja nueva, caja flexible o Flex-Box se crea de una forma muy sencilla. También puede definirse como un conjunto o sistema de propiedades para la creación de interfaces web.

FlexBox tiene una estructura que es muy simple de comprender, está constituido por un padre (Contenedor-Flex) y sus hijos (Elementos-Flex), a continuación, se describen los conceptos antes mencionados:

- **Contenedor-Flex:** Es el elemento “padre” que contiene a los elementos “hijos”, para definirlos se usa la propiedad *display: flex|inline-flex*.
- **Elemento-Flex:** Se refiere a los elementos hijos que tendrán un comportamiento automático dependiendo de lo que se defina en el elemento padre.
- **Ejes:** FlexBox está constituido por dos ejes:
 - **Eje principal:** Es el eje que define la posición horizontal de los **elementos-flex**.
 - **Eje secundario:** Es el eje que define la posición vertical de los **elementos-flex**.

Dentro de las características principales que ofrece el trabajar con Flex-Box, se encuentran las de crear diseños flexibles escribiendo una cantidad menor de código, alinear los diferentes elementos de manera vertical y horizontal de acuerdo a los elementos que se mencionan líneas atrás, y finalmente el reordenamiento del contenido sin tocar el código HTML.

Existen sitios que pueden ser de utilidad, si el objetivo es conocer más acerca de Flex-Box o incluso, si deseamos hacer uso del código de colores en su valor hexadecimal, por ejemplo:

- flexbox.help
- the-echoplex.net/flexyboxes
- css-trick.com/box-sizing
- paletton.com

Las 10 Reglas esenciales de Flex-Box

Regla 1 de Flex-Box:

Flex-box necesita un contenedor padre y por lo menos un contenedor hijo (la altura del contenedor va a definir la altura de los hijos).

Regla 2 de Flex-Box

El *Flex-Container* tiene dos ejes. Los elementos se muestran de arriba hacia abajo y de izquierda a derecha, como se muestra en la imagen.

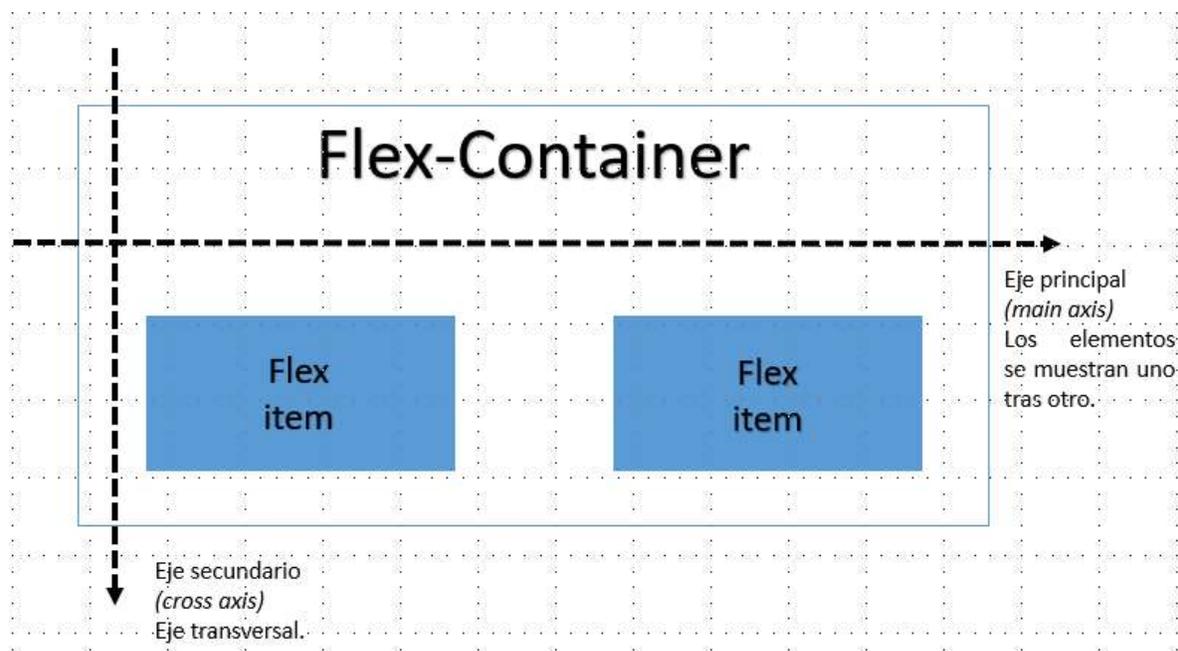


Imagen 3. Flex-Container.

Regla 3 de Flex-Box

Se puede modificar el eje principal con la propiedad: *flex-direction*.

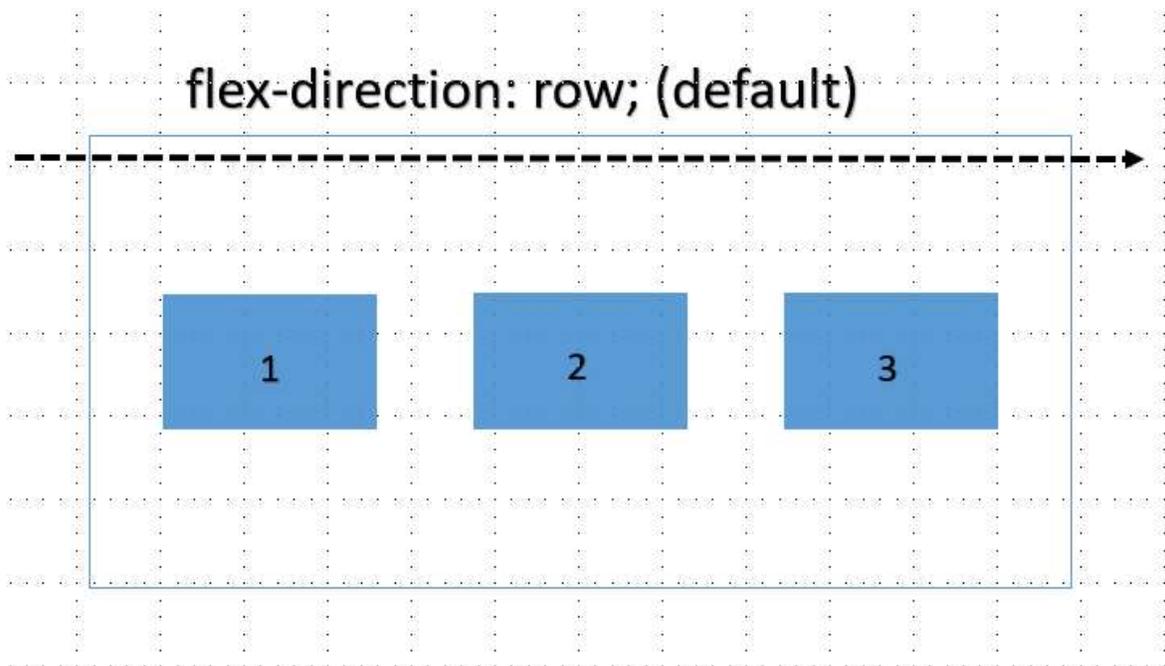


Imagen 4. Propiedad flex-direction: row.

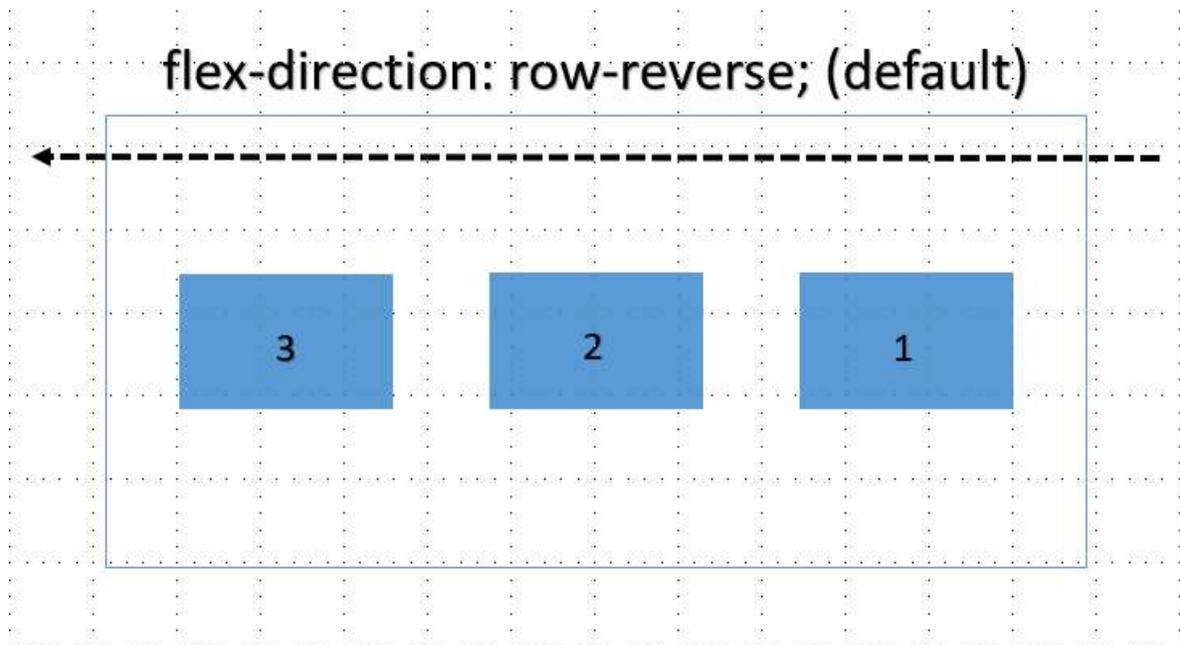


Imagen 5. Propiedad flex-direction: row-reverse.

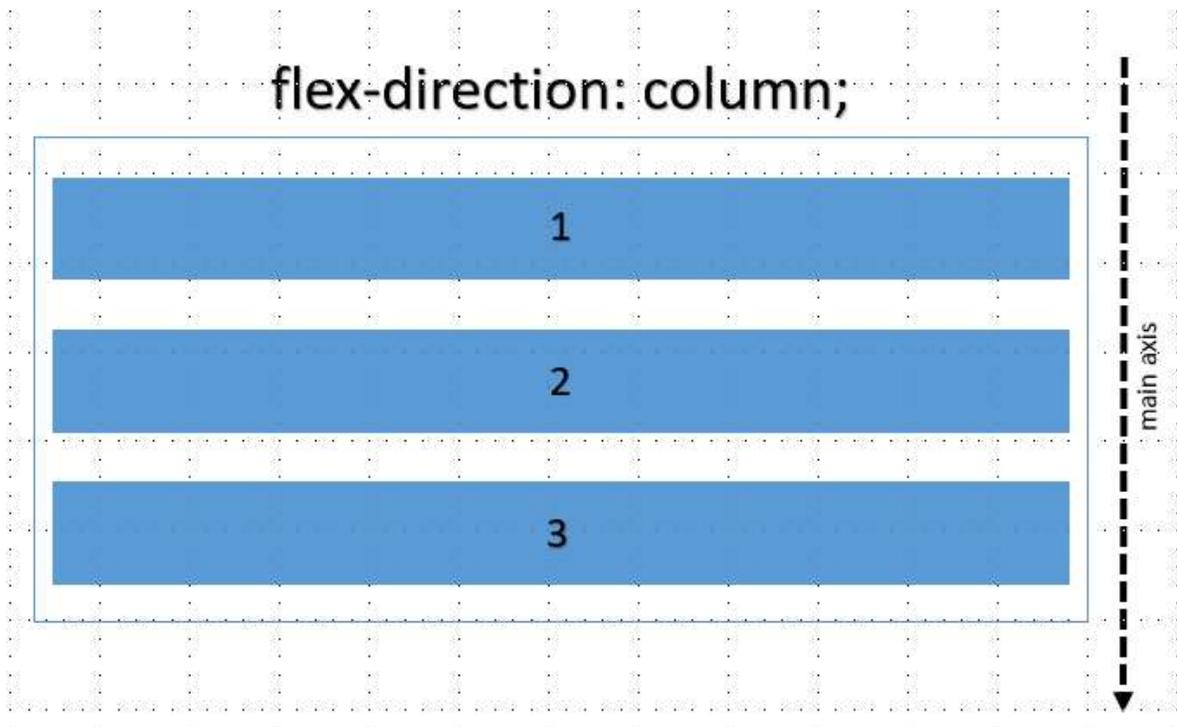


Imagen 6. Propiedad flex-direction: column.

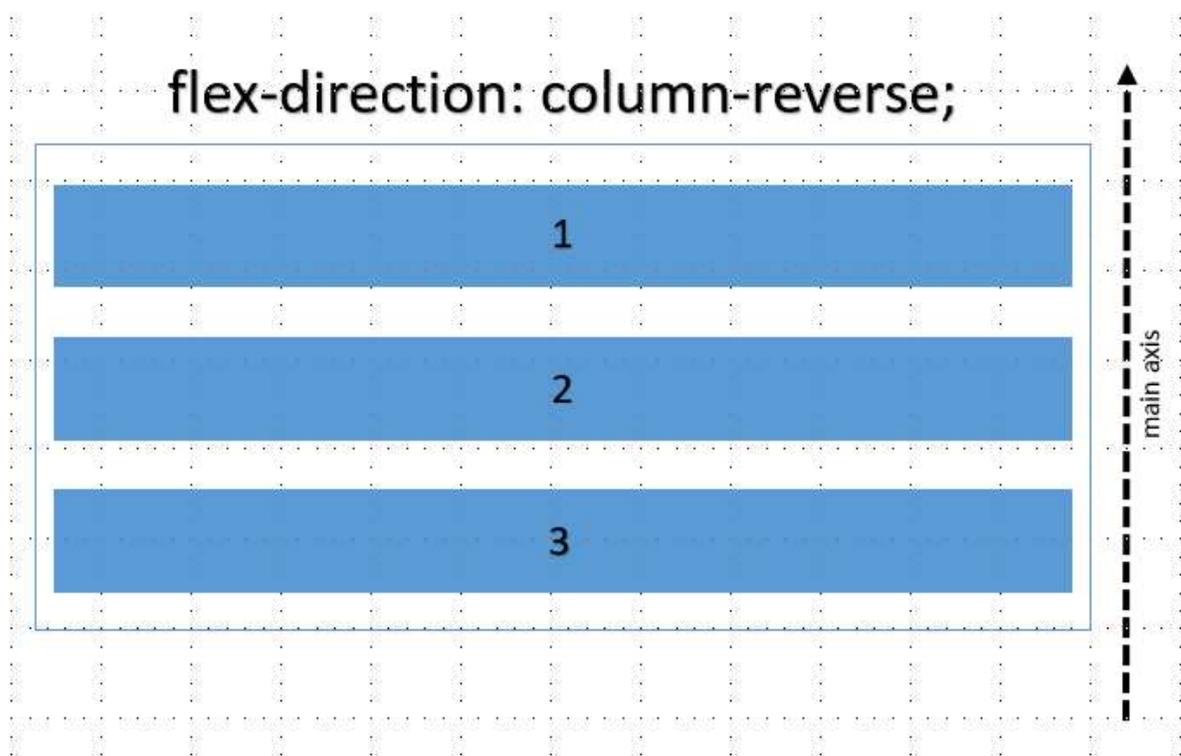


Imagen 7. Propiedad flex-direction: column-reverse.

Regla 4 de Flex-Box

Se permite el salto de columnas si el elemento ya no tiene espacio mediante la propiedad *flex-wrap*.

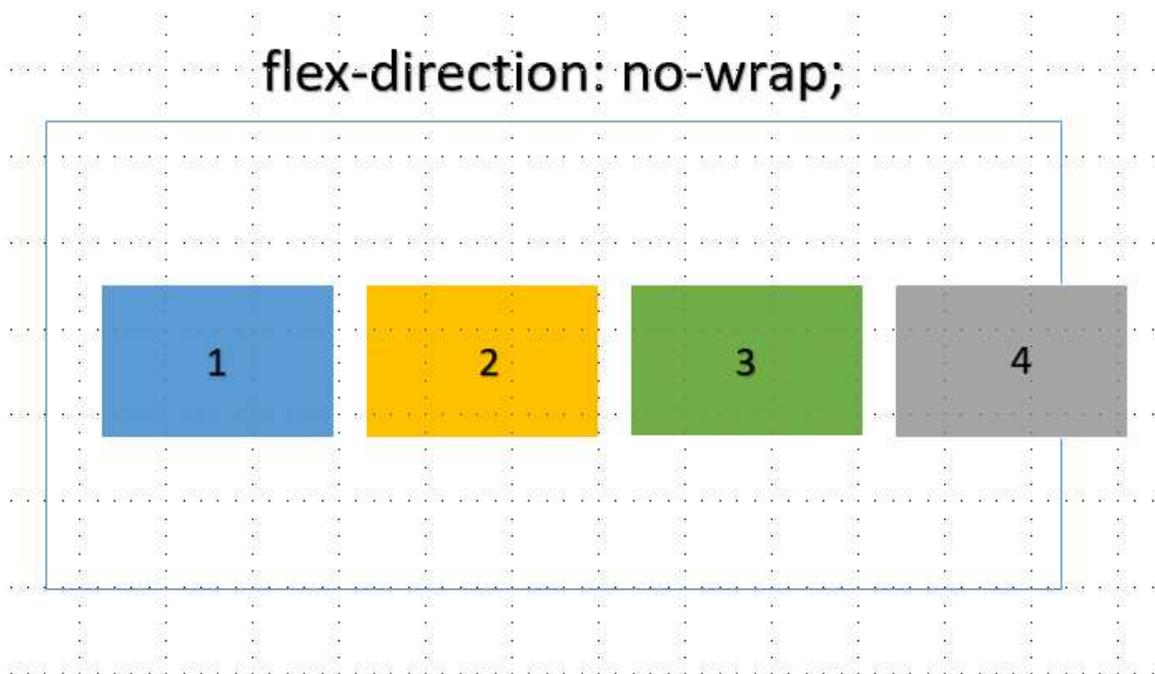


Imagen 8. Propiedad flex-wrap: no-wrap.

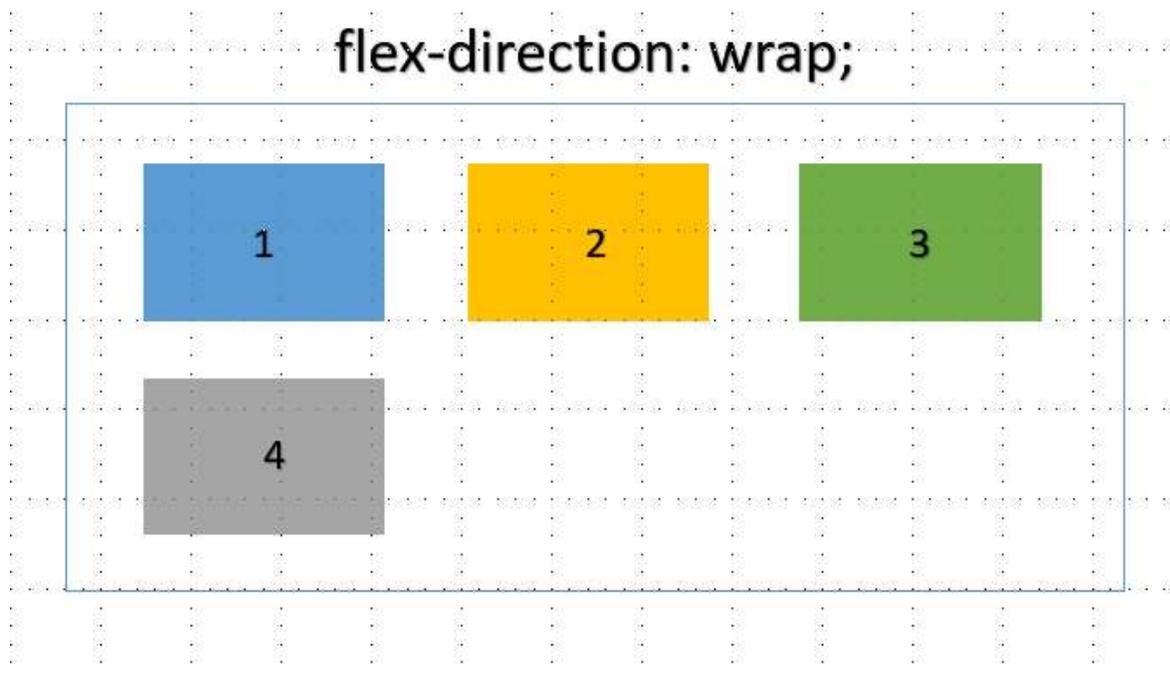


Imagen 9. Propiedad flex-wrap: wrap.

`flex-direction: wrap-reverse;`

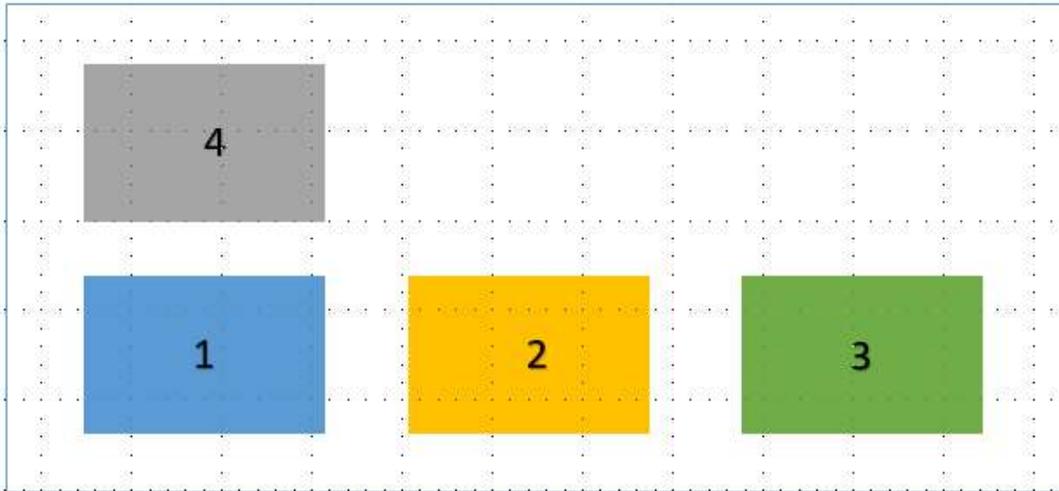


Imagen 10. Propiedad flex-wrap: wrap-reverse.

Regla 5 de Flex-Box

Se pueden alinear los elementos en el eje principal (main axis) con la propiedad *justify-content*.

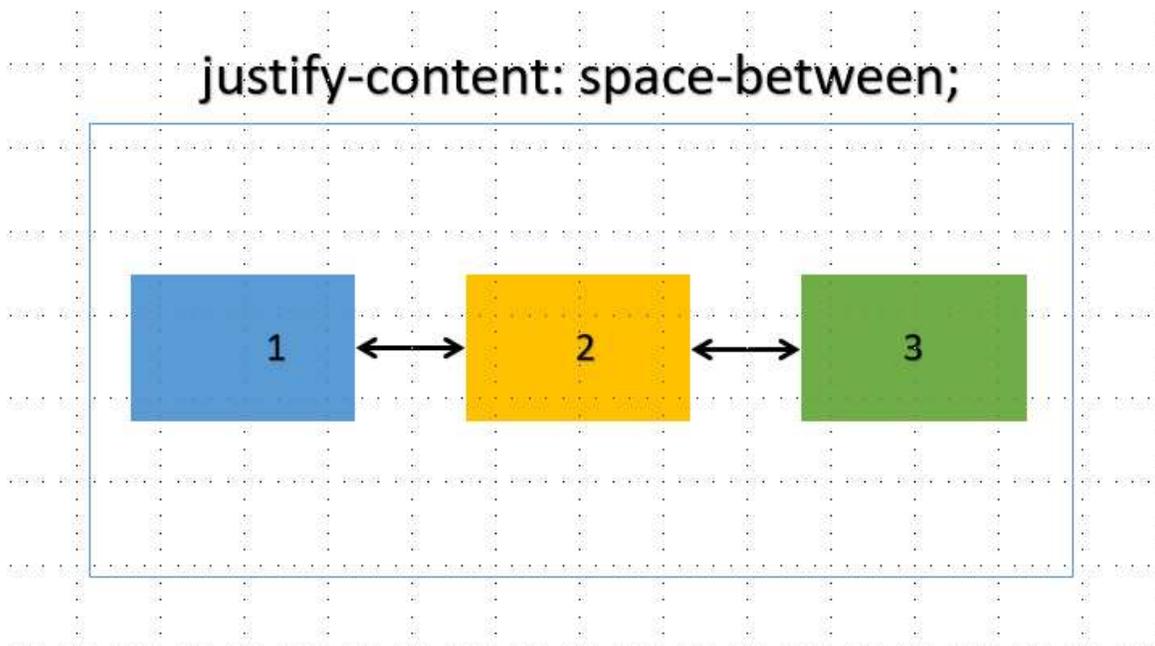


Imagen 11. Propiedad *justify-content: space-between*.

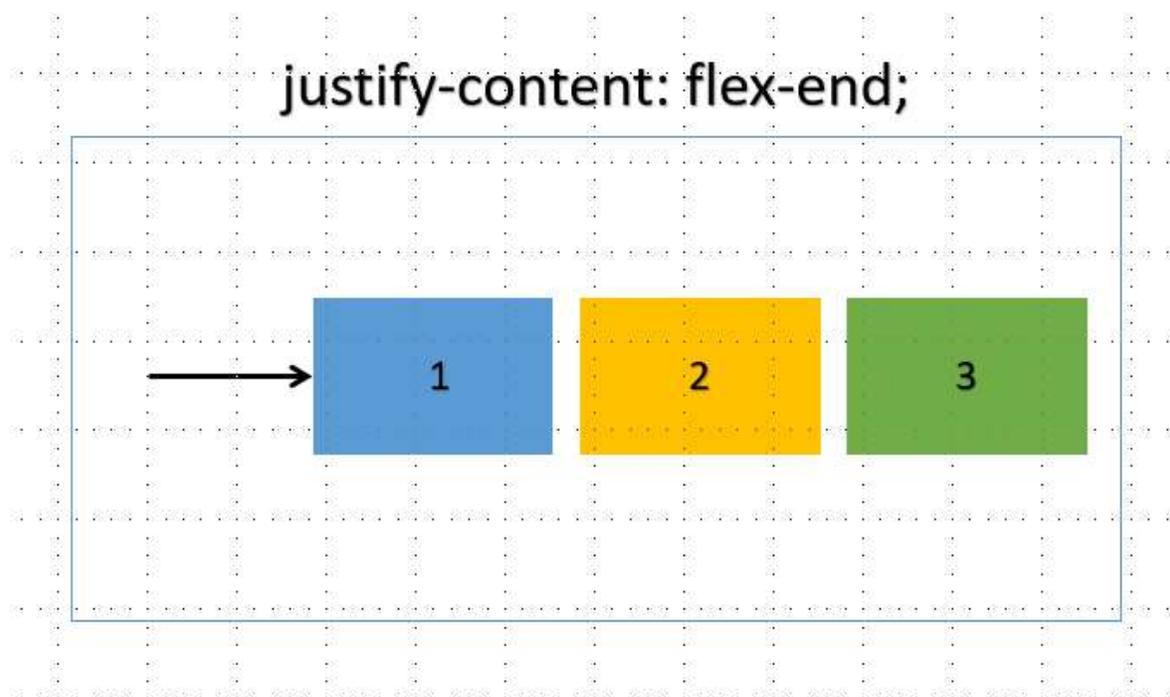


Imagen 12. Propiedad *justify-content: flex-end*.

`justify-content: flex-start;`

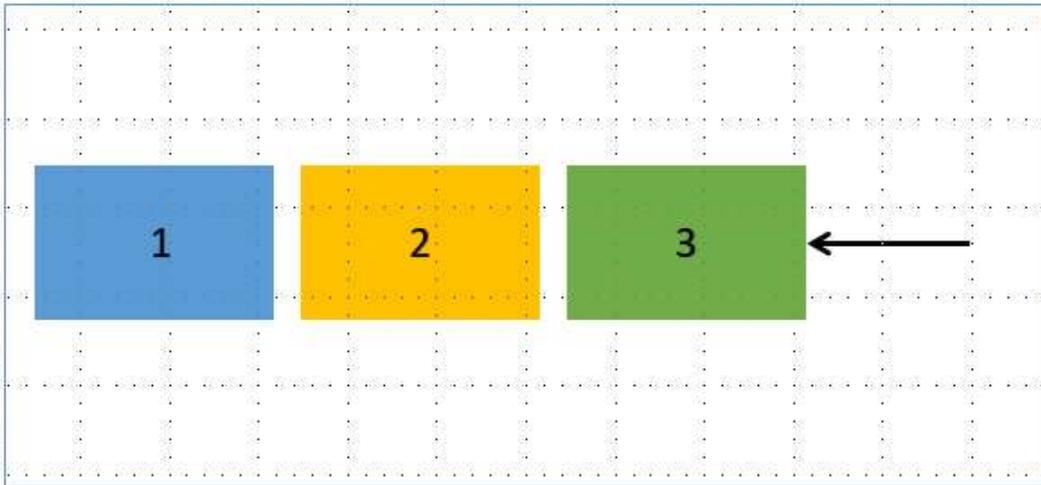


Imagen 13. Propiedad `justify-content: flex-start.`

`justify-content: center;`

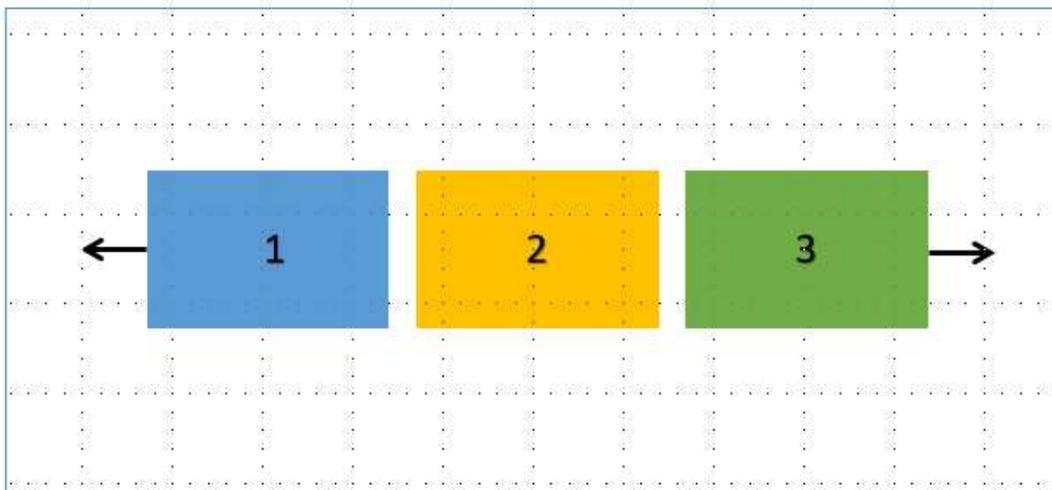


Imagen 14. Propiedad `justify-content: center.`

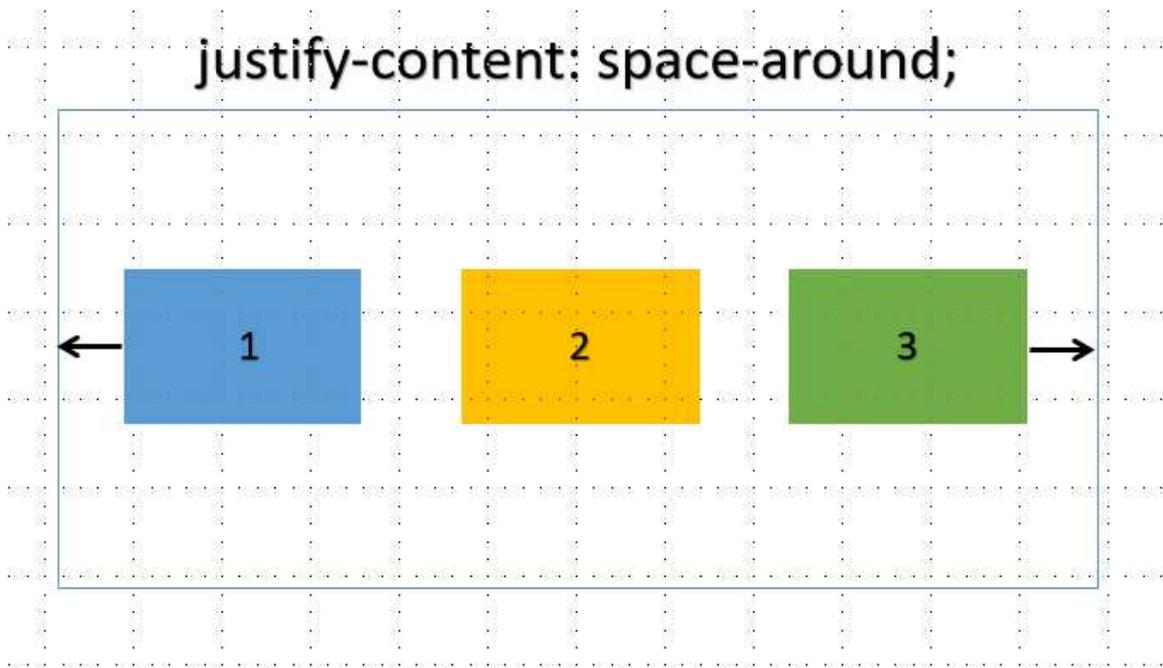


Imagen 15. Propiedad `justify-content: space-around`.

Regla 6 de Flex-Box

Se pueden alinear los elementos en el eje secundario con `align-items`.

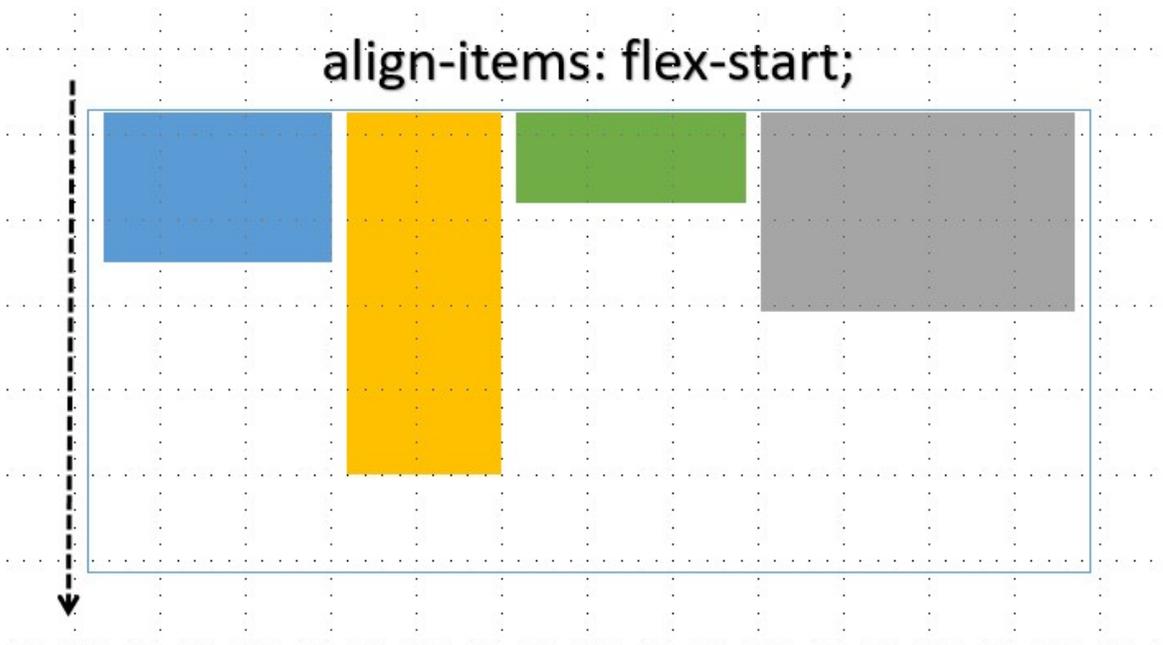


Imagen 16. Propiedad `align-items: flex-start`.

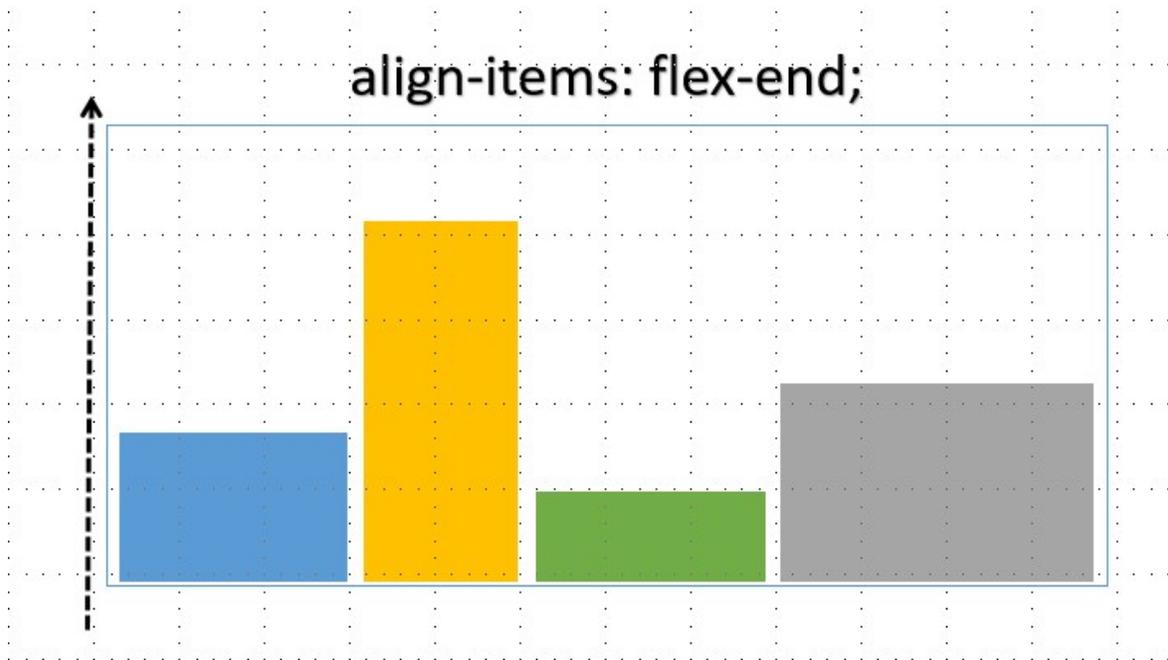


Imagen 17. Propiedad align-items: flex-end.

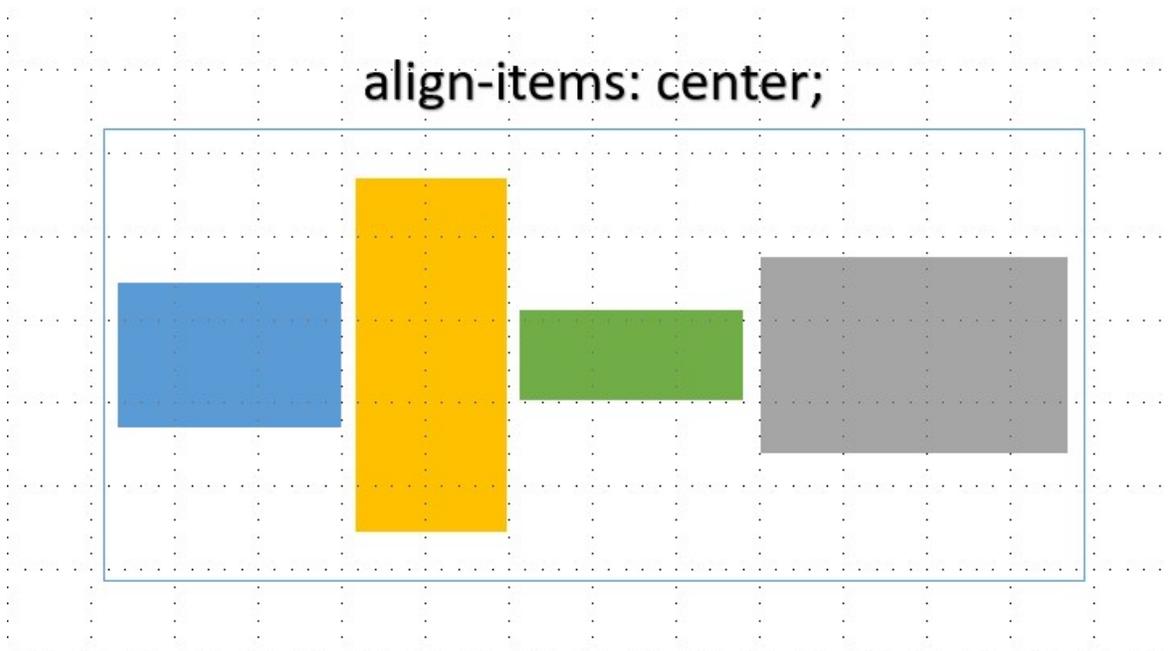


Imagen 18. Propiedad align-items: center.

`align-items: stretch; (default)`

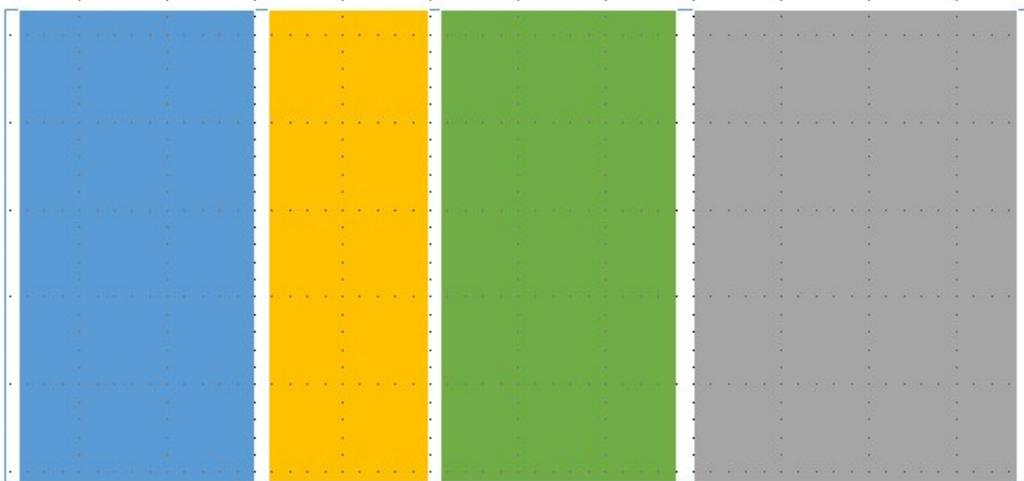


Imagen 18. Propiedad `align-items: stretch;`

`align-items: baseline;`

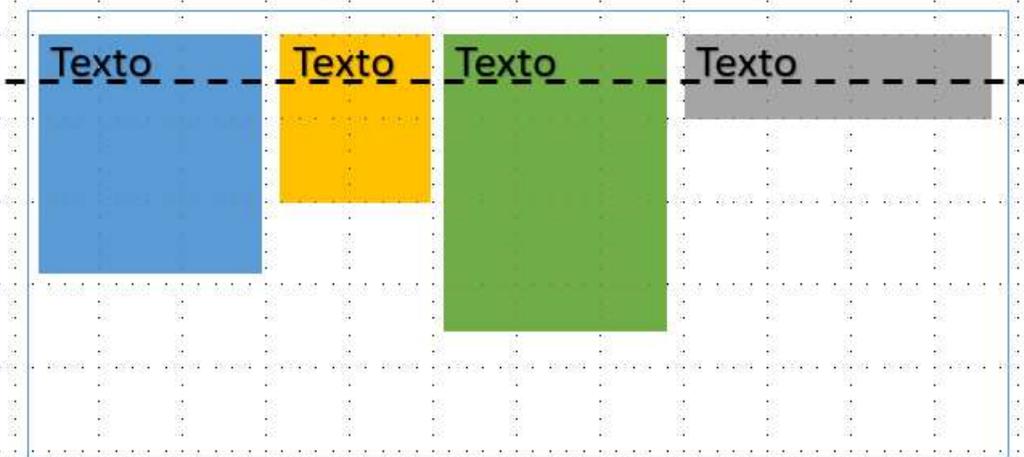


Imagen 20. Propiedad `align-items: baseline;`

Regla 7 de Flex-Box.

Se pueden alinear los elementos hijos de forma individual en el eje secundario o eje transversal (cross axis) mediante la propiedad *align-self*.

{ Esta propiedad sobrescribe de forma individual el valor definido en el padre para la alineación en el eje secundario de los hijos. }

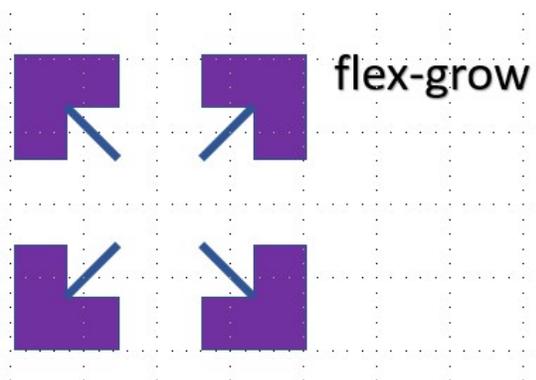
Regla 8 de Flex-Box

Los hijos que son flexibles ignoran propiedades como *float*, *clear*, *vertical-align*.

Regla 9 de Flex-Box

Se puede modificar el tamaño de los hijos mediante el uso de la propiedad *flex-grow*, *flex-shrink*, *flex-basis*.

{ Por defecto, el tamaño de los hijos se define por el tamaño de su contenido. }

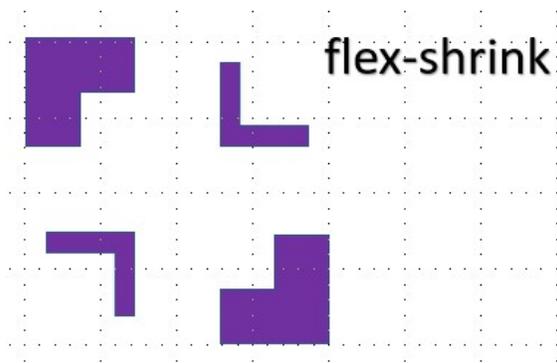


Esta propiedad define el tamaño que va a crecer un hijo en relación a sus demás hermanos **cuando hay espacio disponible en el contenedor** donde se encuentra.

El valor por defecto es 0.

Imagen 21. Flex-grow

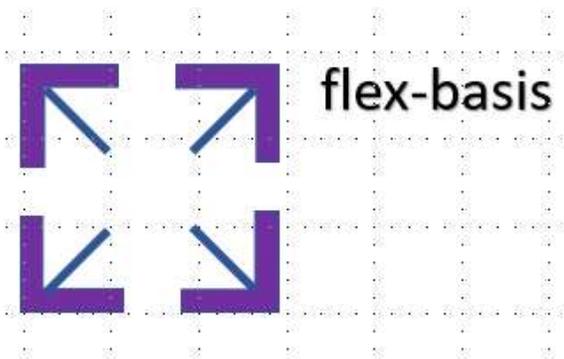
Si hay espacio, y dos hijos flexibles van a crecer, la distribución del espacio disponible es a proporciones iguales.



Esta propiedad define el tamaño de reducción de un hijo en relación a sus demás hermanos cuando no haya suficiente espacio en el contenedor donde se encuentra.

El valor por defecto es 1.

Imagen 22. Flex-shrink.



Esta propiedad define el tamaño inicial del hijo. Su valor por defecto es "auto". Cuando *flex-basis* es igual a 0, y se tiene *flex-grow* a 1, entonces todos los elementos hijos reciben las proporciones de crecimiento iguales, ya que con *flex-basis: 0*, se dispone del

100% del contenedor.

Imagen 23. Flex-basis.

Regla 10 de Flex-Box

La regla anterior se puede resumir con la propiedad *flex*. La sintaxis es la siguiente:

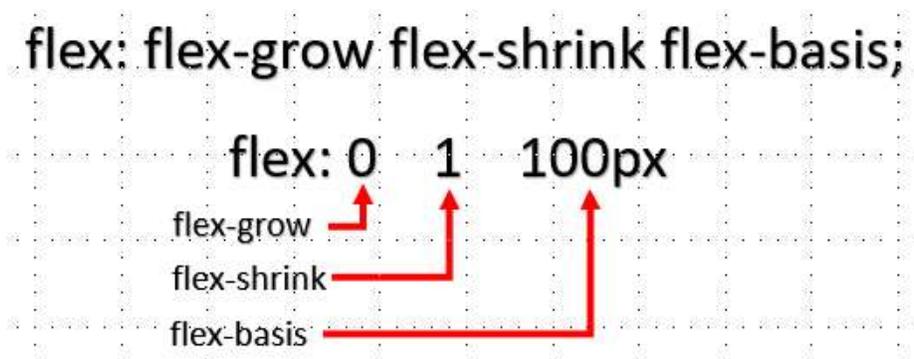


Imagen 24. Atajo flex-grow, shrink y basis.

`flex: flex-grow flex-basis;`

`flex: 0 100px`

A diagram illustrating the shorthand `flex: 0 100px`. The text `flex: 0 100px` is shown. Below it, the text `flex-grow` has a red arrow pointing to the `0` in the shorthand. The text `flex-basis` has a red arrow pointing to the `100px` in the shorthand. Below that, the text `flex-shrink: se omite` is shown.

Imagen 25. Omisión flex-basis.

`flex: flex-grow;`

`flex: 1;`

A diagram illustrating the shorthand `flex: 1;`. The text `flex: 1;` is shown. Below it, the text `flex-grow` has a red arrow pointing to the `1` in the shorthand. Below that, the text `flex-basis: 0;` is shown.

Imagen 26. Atajo flex-grow.

Finalmente, existe la posibilidad de reordenar los hijos flexibles con la propiedad `order`. Todos tienen por defecto el valor de `order = 0`.

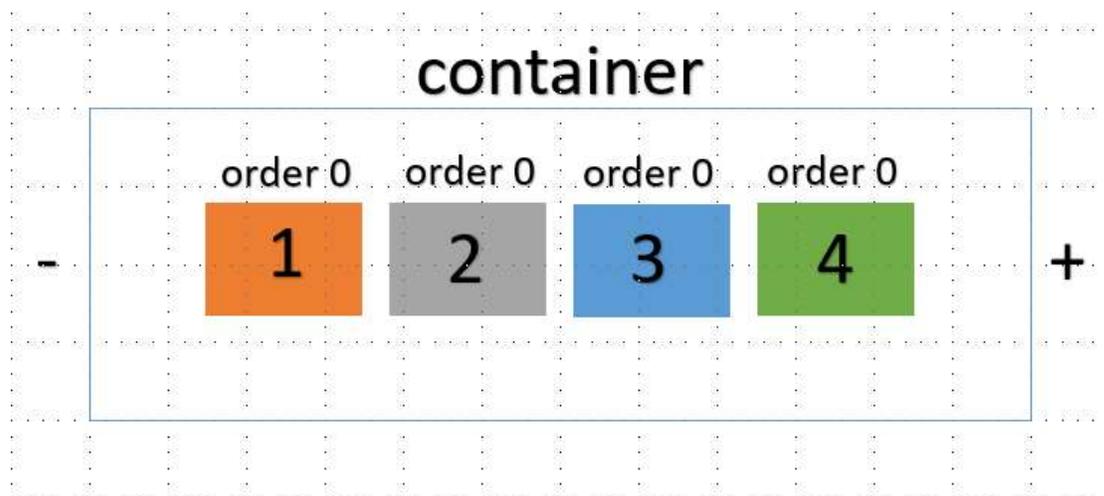


Imagen 27. Comportamiento de la propiedad `order`.

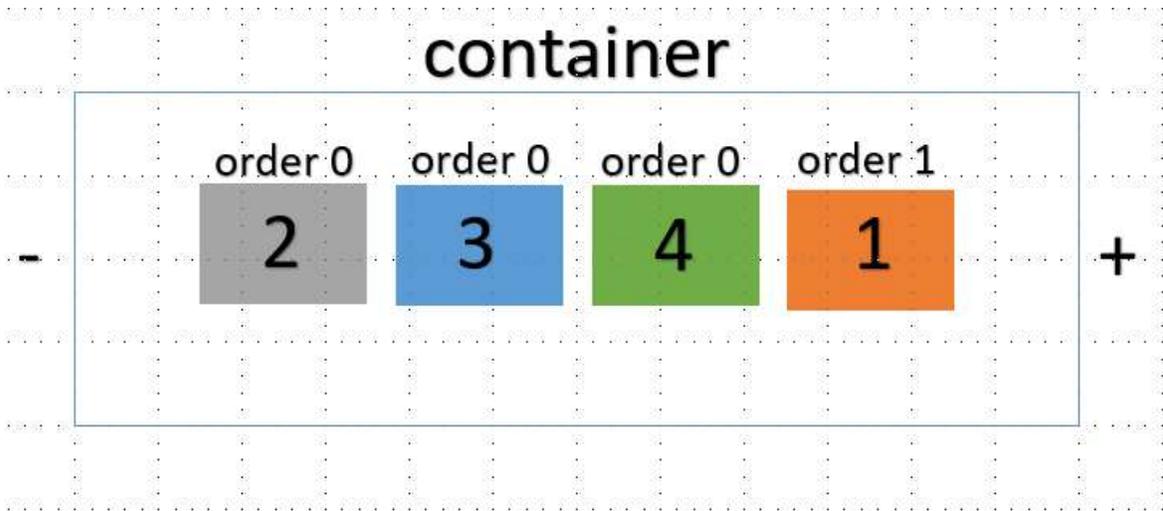


Imagen 28. Comportamiento de la propiedad *order* con valor 1.

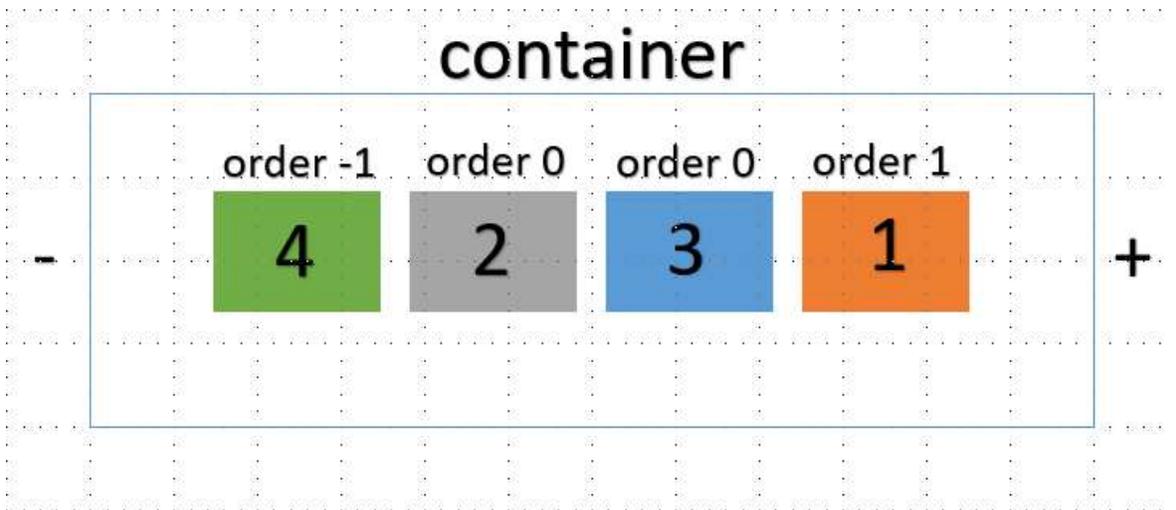


Imagen 29. Comportamiento de la propiedad *order* con valor -1.

Bootstrap 4

Es un framework HTML, CSS y Javascript de código abierto. Es un esquema (esqueleto o patrón) para el desarrollo y/o implementación de una aplicación que facilita el desarrollo rápido de páginas web responsive, utilizando la filosofía *mobile first*.

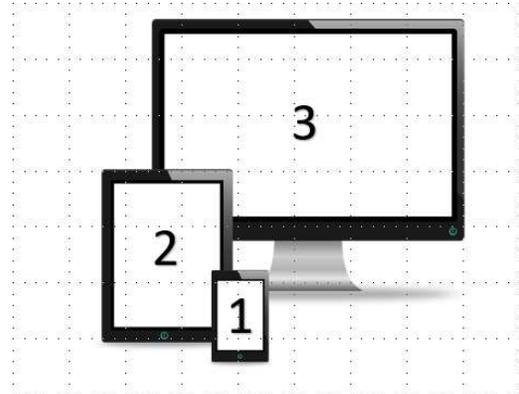


Imagen 30. Diferentes resoluciones.

Se basa en la concepción de un sitio web iniciando desde resoluciones de pantalla pequeñas como las que presenta un teléfono celular. En esta filosofía se obliga a insertar elementos que el usuario realmente necesita.

¿Qué es un framework?

Un framework es un conjunto de herramientas que liberan al diseñador o desarrollador, de crear funciones frecuentes durante la creación o maquetación de un proyecto web, en otras palabras, es un conjunto de herramientas que facilitan la creación de aplicaciones web.

¿Qué incluye Bootstrap?

El framework incluye tipografía, un sistema de columnas, tablas, acordeones, imágenes, botones, formularios, menús, barras de navegación, sliders, pantallas modales, y muchas otras funciones más.

¿Por qué Bootstrap?

La utilización de Bootstrap es sin duda alguna por su rapidez, ya que contiene código *CSS*, *HTML* y *JS* escrito y listo para usar. Otra de las razones es porque Bootstrap tiene la característica de ser personalizable. Su versión cuatro, fue específicamente desarrollada en *Mobile First* y *Responsive*, aunque no hay soporte para el navegador IE8, es compatible con los otros navegadores. Hace uso de unidades *rem* y *EM*, además de los ya conocidos *pixeles*. Se mejoró el sistema de columnas y soporta *Flex-Box*. También se agregaron otros componentes. Bootstrap fue desarrollado por los creados de Twitter.

Sistemas de Columnas de Bootstrap 4.

¿Qué es el sistema de columnas?

Es una forma de crear columnas flexibles, de tal manera que se pueda crear “columnas” para los sitios web. Bootstrap 4 consta de un total de 12 columnas. Se muestran diferentes números de columnas dependiendo del espacio de la pantalla del dispositivo del usuario. Así como *Flex-Box*, *Bootstrap 4* consta de 10 reglas indispensables.

REGLA 1

Existen tres elementos que conforman el sistema:

- Contenedor
- Filas
- Columnas

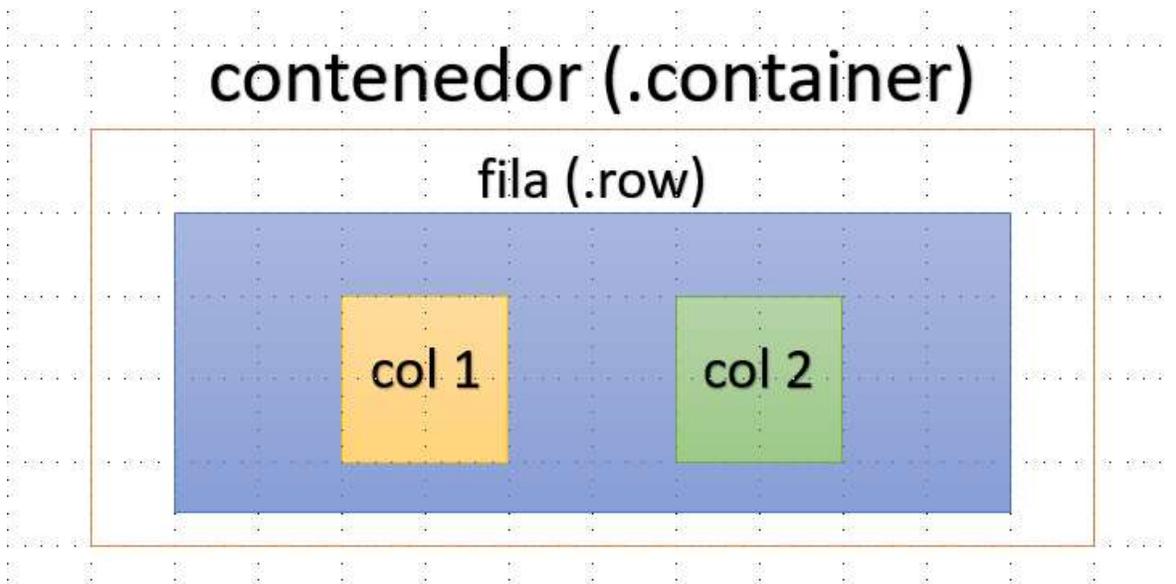


Imagen 31. Estructura de Bootstrap.

Donde *.container* → nombre de la clase y *.row* → nombre de la fila que se encuentra dentro del contenedor (*container*). El contenedor puede ser cualquier etiqueta de las que conforman el lenguaje HTML, normalmente son etiquetas *<div>*.

REGLA 2

El sistema de columnas se crea o inicia utilizando un contenedor. Existen dos tipos de contenedores:

- *container* (no ocupa el 100% de la pantalla) y
- *container-fluid* (ocupa el 100% de la pantalla [ancho])

REGLA 3

Las filas tienen que ser hijos directos del contenedor, es decir:

```
1 <div class="container">
2   <div class="row"></div>
3   <div class="row"></div>
4 </div>
5
```

REGLA 4

Las columnas se dividen en un máximo de 12, y son hijos directos de una fila, es decir:

```
1 <div class="container">
2   <div class="row">
3     <div class="col-xs-#_col"></div>
4   </div>
5
6 </div>
7
```

Donde `#_col` es el número de columnas que se van a ocupar.

12 → 100%

lg → vista de escritorio

6 → 50%

md → vista de tableta

xs → vista móvil

REGLA 5

Se se supera el número máximo de 12 columnas, las columnas sobrantes saltarán a la siguiente fila.

REGLA 6

El contenido tiene y debe de ir dentro de las columnas.

REGLA 7

Se pueden definir diferentes anchuras de columnas para diferentes tamaños de pantalla (asignando diferentes clases), es decir:

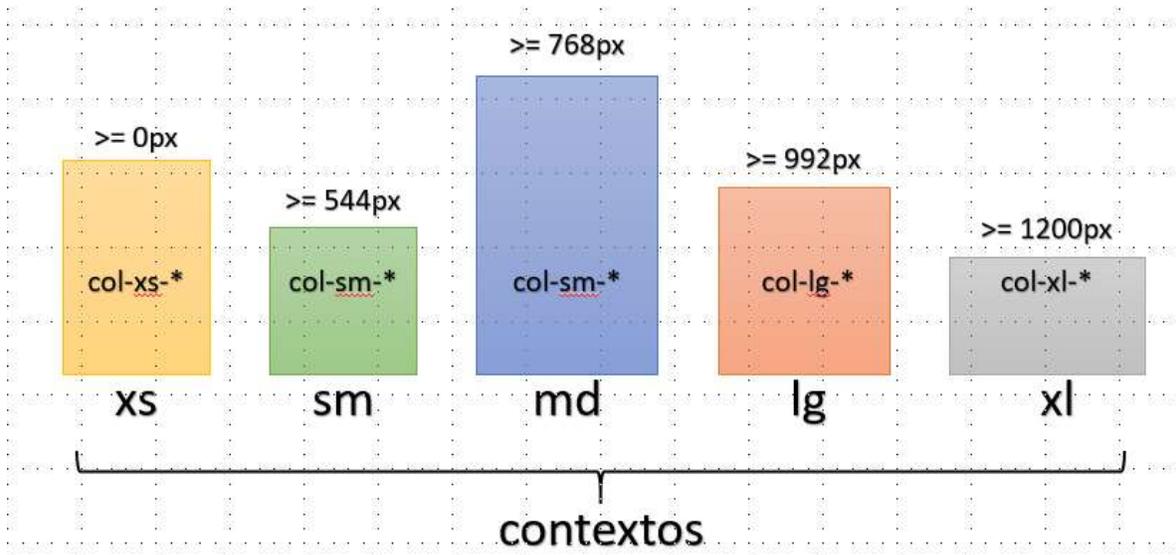


Imagen 32. Contextos para las diferentes resoluciones.

REGLA 8

Las clases para las columnas se heredan (hacia la derecha). La propiedad que hereda es la última que se define en el código css.

REGLA 9

Se pueden mover columnas con la propiedad *offset*. Por ejemplo, `col-xs-2 offset-xs-#_col`.

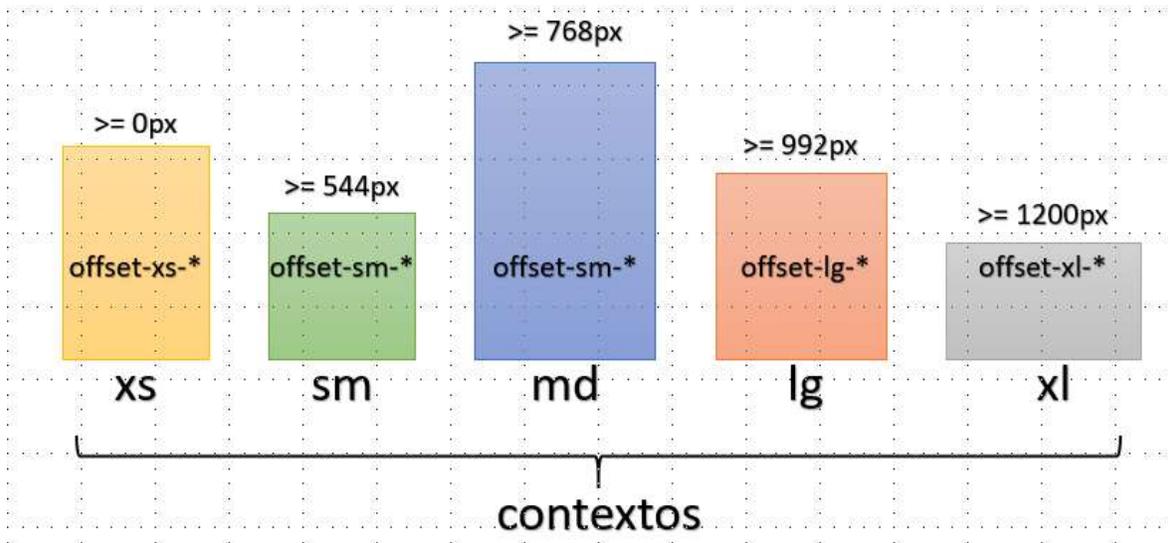


Imagen 33. Notación para propiedad *offset*.

REGLA 10

Es posible llevar a cabo una anidación de columnas, es decir:

```
1 <div class="container">
2   <div class="row">
3     <div class="col-xs-6">
4       <h2>TITULO</h2>
5       <div class="col-xs-7">
6         Sub columna 1(columna anidada)
7       </div>
8     </div>
9   </div>
10
11 </div>
```

Clases para *Margin* y *Padding*

Tipo	Dirección	Tamaño
m = margin	a = all	0 = 0
p = padding	t = top	auto = auto
	r = right	1 = 1rem (15 px)
	b = bottom	2 = 1.5rem
	l = left	3 = 3rem
	x = left y right	
	y = top y bottom	

Su sintaxis es la siguiente:

*Tipo-dirección-tamaño; p-y-1 o m-x-1 o p-all-auto*⁸.

Maple

Maple es una potente herramienta, tecnológicamente avanzada, que incorpora algoritmos simbólicos propios reconocidos en todo el mundo. Es Herramienta matemática estándar para el cálculo analítico que permite realizar algebra simbólica, cálculo numérico, resolver ecuaciones diferenciales, gráficos y animaciones.

Cualquiera que sea el área científica o técnica en la que se esté trabajando, ya sea en el ámbito de la enseñanza, en el de investigación o en desarrollo, Maple es un entorno ideal que cubre todos los aspectos necesarios.

Incorpora herramientas suficientemente flexibles para ajustarse a todas las necesidades de cálculo: desde la resolución de sistemas de ecuaciones diferenciales hasta el modelado de complejos problemas de ingeniería⁹.

Maple es un sistema de cálculo simbólico o algebraico. Ambas expresiones hacen referencia a la habilidad que posee Maple para trabajar con la información de la misma manera que lo haríamos nosotros cuando llevamos a cabo cálculos matemáticos analíticos. Mientras que los programas matemáticos tradicionales requieren valores numéricos para todas las variables, Maple mantiene y manipula los símbolos y las expresiones. Estas capacidades simbólicas permiten obtener soluciones analíticas exactas de los problemas matemáticos: por ejemplo, se

⁸ Apuntes del curso *Bootstrap 4*

⁹<https://www.addlink.es/productos/maple> consultado en Marzo 12, 2018 16:43 hrs.

pueden calcular límites, derivadas e integrales de funciones, resolver sistemas de ecuaciones de forma exacta, encontrar soluciones de ecuaciones diferenciales, etc.

Como complemento a las operaciones simbólicas existe un amplio conjunto de rutinas gráficas que permiten visualizar información matemática compleja, algoritmos numéricos que dan soluciones en precisión arbitraria de problemas cuya solución exacta no es calculable y un lenguaje de programación completo y comprensible que permite al usuario crear sus propias funciones y aplicaciones. Internamente Maple se estructura en tres partes. En primer lugar, está el núcleo, formado por rutinas escritas y compiladas en lenguaje C, donde se realizan la mayor parte de los cálculos básicos hechos por el sistema. La segunda parte es un conjunto de librerías, donde se encuentra la mayoría de los comandos de Maple, y que están escritas en su propio lenguaje de programación (interpretado no compilado), lenguaje que permite al usuario crear sus propios comandos y añadirlos a la librería estándar (es por tanto un sistema extensible). Y finalmente la interfaz del programa a través de la cual es posible comunicarse con el sistema¹⁰.

A lo largo de la historia, Maple ha tenido 17 lanzamientos, comenzando con su primer versión Maple 6, liberada en el año 2000. La versión más actual y la que se encuentra en el mercado es Maple 2017, la cual se liberó el pasado año¹¹.

El cálculo simbólico, es un área científica que se refiere al estudio y desarrollo de algoritmos y software para la manipulación de expresiones matemáticas, el uso de Maple facilita en gran medida la creación de scripts para la creación de funciones y visualizar de manera gráfica su comportamiento, de tal manera que sea mucho más fácil de comprender lo que se está resolviendo, ya sean ecuaciones lineales, no lineales, diferenciales, derivadas, integrales, series, etc., es decir, desde matemática elemental hasta matemática avanzada.

¹⁰ https://www.uoc.edu/in3/emath/docs/UV_Maple.pdf consultado en Marzo 12, 2018 16:58 hrs.

¹¹ <https://www.maplesoft.com/products/maple/history/> consultado en Marzo 12, 2018 17:03 hrs.

Ventajas del cómputo simbólico

- Resolución de problemas matemáticos de manera inmediata.
- Visualización de problemas matemáticos mediante funciones de graficación.
- Los scripts se pueden reutilizar para adecuarlos a problemas similares.
- Manejo de grandes cifras enteras y decimales.
- Creación de modelos matemáticos aplicables a la vida cotidiana en múltiples áreas.
- Manejo de lenguajes de programación.
- Se adecúa a las necesidades de quien usa las herramientas.

Desventajas del cómputo simbólico

- Algunas herramientas consumen grandes recursos de cómputo.
- Interfaces poco amigables de algunas herramientas.

DESARROLLO

Ciclo de vida del software

El término ciclo de vida del software describe el desarrollo de software, desde la fase inicial hasta la fase final. El propósito de este ciclo es definir las distintas fases intermedias que se requieren para validar el desarrollo de la aplicación, es decir, para garantizar que el software cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo: se asegura de que los métodos utilizados son apropiados.

El ciclo de vida básico de un software consta de los siguientes procedimientos:

- **Definición de objetivos:** define la finalidad del proyecto y su papel en la estrategia global.
- **Análisis de los requisitos y su viabilidad:** recopila, examina y formula los requisitos del cliente y examina cualquier restricción que se pueda aplicar.
- **Diseño general:** requisitos generales de la arquitectura de la aplicación.
- **Diseño en detalle:** definición precisa de cada subconjunto de la aplicación.
- **Programación (programación e implementación):** implementación de un lenguaje de programación para crear las funciones definidas durante la etapa de diseño.
- **Prueba de unidad:** prueba individual de cada subconjunto de la aplicación para garantizar que se implementaron de acuerdo con las especificaciones.
- **Integración:** garantiza que los diferentes módulos se integren con la aplicación. Este es el propósito de la prueba de integración que está cuidadosamente documentada.
- **Validación:** garantiza que el software cumple con las especificaciones originales.
- **Documentación:** sirve para documentar información necesaria para los usuarios del software y para desarrollos futuros.
- **Implementación**

- **Mantenimiento:** comprende todos los procedimientos correctivos (mantenimiento correctivo) y las actualizaciones secundarias del software (mantenimiento continuo).

Modelos del ciclo de vida

Modelo en cascada

El modelo de ciclo de vida en cascada se comenzó a diseñar en 1966 y se terminó alrededor de 1970. Se define como una secuencia de fases donde al final de cada una de ellas se reúne la documentación para garantizar que cumple las especificaciones y los requisitos antes de pasar a la fase siguiente¹²:

Una de sus ventajas, además de su planificación sencilla, es la de proveer un producto con un elevado grado de calidad sin necesidad de un personal altamente calificado. Se pueden considerar como inconvenientes: la necesidad de contar con todos los requerimientos (o la mayoría) al comienzo del proyecto, y, se han cometido errores y no se detectan en la etapa inmediata siguiente, es costoso y difícil volver atrás para realizar la corrección posterior¹³.

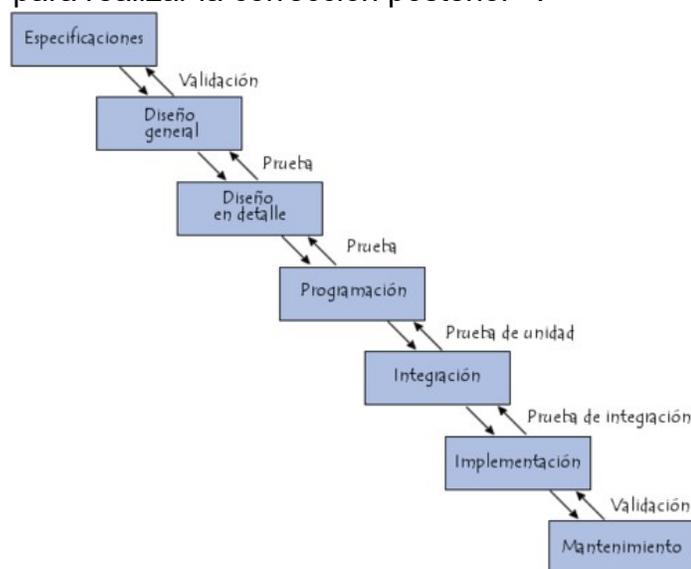


Imagen 34. Modelo en cascada.

¹² <https://es.ccm.net/contents/223-ciclo-de-vida-del-software>

¹³ <https://ingsw.pbworks.com/f/Ciclo+de+Vida+del+Software.pdf>

Ciclo de vida en V

Este ciclo fue diseñado por Alan Davis, y contiene las mismas etapas que el ciclo de vida en cascada. A diferencia de aquél, a éste se le agregaron dos sub-etapas de retroalimentación entre las etapas de análisis y mantenimiento, y entre las de diseño y debugging.

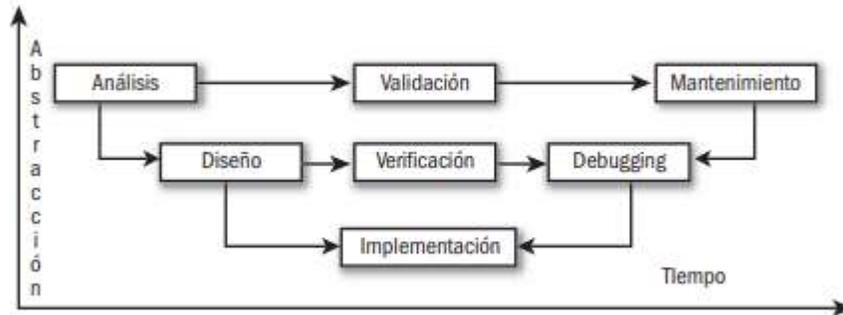


Imagen 35. Modelo en V.

Las ventajas y desventajas de este modelo son las mismas que el ciclo en cascada, con el agregado de los controles cruzados entre etapas para lograr una mayor corrección.

Ciclo de vida iterativo

Este ciclo de vida también es derivado del ciclo de vida en cascada, este modelo busca reducir el riesgo que surge entre las necesidades del usuario y el producto final por malos entendidos durante la etapa de solicitud de requerimientos.

Es la iteración de varios ciclos de vida en cascada. Al final de cada iteración se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto. El cliente es quien luego de cada iteración, evalúa el producto y lo corrige o propone mejoras. Estas iteraciones se repetirán hasta obtener un producto que satisfaga al cliente.

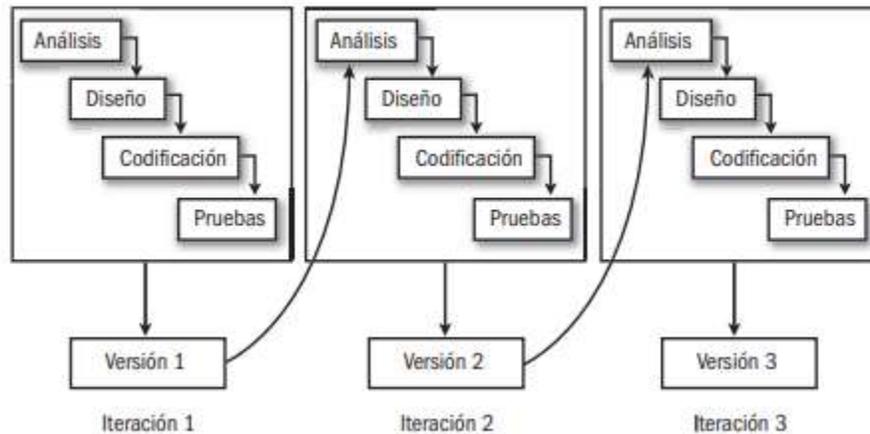


Imagen 36. Modelo iterativo.

Modelo empleado

El modelo en el que se basó el desarrollo del medio educativo fue en el iterativo, ya que se implementaron 4 iteraciones, las cuales corresponden a una versión diferente de dicho medio.

La primera fase, *análisis*, corresponde al análisis de requerimientos, los cuales están expresados dentro de los *objetivos* del proyecto, son los requisitos con los que debe cumplir el proyecto para ser aceptado, pueden considerarse parte de los criterios de aceptación.

La fase de *diseño*, corresponde a la forma en la que se va a presentar visualmente el medio educativo, engloba todos los elementos que relacionan la interfaz del usuario y la posterior experiencia del usuario (UI/UX). En el apéndice B se pueden apreciar los distintos *mockups* relacionados a esta fase. Además, en el siguiente apartado, *diseño e implementación de una página web*, se puede apreciar de forma general, la estructura básica (diseño) de una página web.

La fase de *codificación*, corresponde a trasladar cada uno de los *mockups* a una forma en la que las computadoras puedan entenderlo y representarlo, para el diseño web, existen varias herramientas y lenguajes que permiten llevar a cabo este objetivo. Para este proyecto el lenguaje utilizado es HTML en su versión 5, el cual es brevemente descrito dentro del *marco conceptual*. Sin embargo, no basta sólo con hacer uso del lenguaje correcto para poder codificar algún diseño, sino que, se requiere de la combinación de diferentes tecnologías para lograr una extraordinaria interfaz para el usuario y con ello atrapar al usuario con una extraordinaria experiencia.

Finalmente tenemos la fase de *pruebas*, la cual corresponde a implementar la validación de todos los componentes con los que interactúa el usuario, desde cómo se visualiza una imagen, un ícono, la tipografía hasta el correcto funcionamiento de botones, enlaces a otros sitios, animaciones, sonidos, etc.

El resultado obtenido en cada versión se puede visualizar en el apartado de *Versionado*, en la página 45. La iteración final se logra hasta que se considera que el diseño actual está en condiciones para ser presentada al cliente o usuario final, y cumple con todos o con la mayoría de los requerimientos que se recabaron al inicio del proyecto.

Diseño e implementación de una página web

A pesar del hecho de que cada diseñador crea sus propios diseños, en general se puede identificar cada una de las secciones en cada sitio web estudiado, y sin importar la complejidad del sitio, todas siguen el mismo diseño común. La siguiente imagen muestra cada una de las secciones de un sitio web.

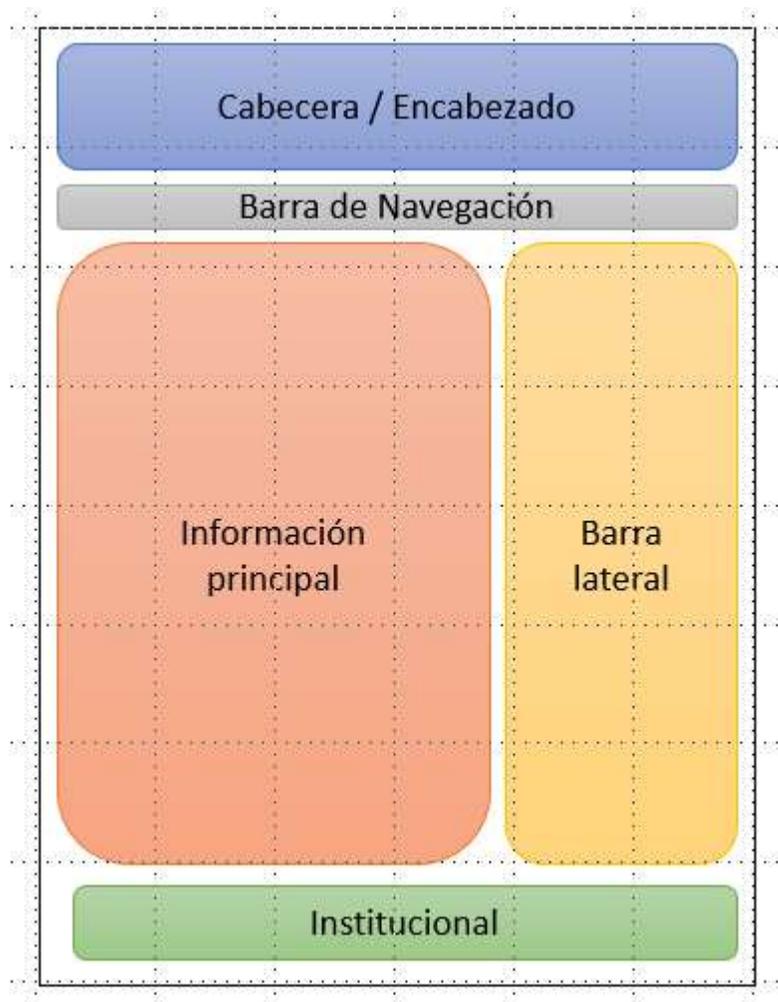


Imagen 37. Organización de una página web.

En la parte superior, la cual se describe como **Cabecera**, se encuentra el espacio donde usualmente se ubica el logo, título, subtítulos y una breve descripción del sitio web o la página. Inmediatamente debajo, se puede observar la **Barra de Navegación** en la cual casi todos los desarrolladores ofrecen un menú o lista de enlaces cuyo propósito es facilitar la navegación a través de todo el sitio por muy

básico o complejo que éste sea. Los usuarios son guiados desde esta barra hacia las diferentes páginas o documentos que normalmente pertenecen al mismo sitio web.

El contenido más importante de una página web se encuentra, en casi todo el diseño, ubicado en el centro. Esta es la sección que representa información y enlaces valiosos. En la mayoría de las veces esta puede estar dividida en varias filas y columnas. En la Imagen 37, se visualiza sólo dos columnas: **Información principal** y **Barra lateral**, pero esta sección es extremadamente flexible y normalmente los diseñadores la adaptan acorde a las necesidades insertando más columnas, dividiendo cada columna entre bloques más pequeños o generando diferentes distribuciones y combinaciones. El contenido que se presenta en esta parte del diseño es usualmente de alta prioridad. La **Barra lateral** podría mostrar una lista de enlaces redirigiendo a otra información importante dentro de la sección anterior, hacia cada uno de esos ítems.

En la base del diseño web, siempre podemos encontrarnos con una barra más, la cual es llamada **Institucional** o **Footer** o pie de página, esta es el área en donde se muestra información acerca del sitio web, el autor o la empresa, además de algunos enlaces con respecto a reglas, términos y condiciones, redes sociales, contacto y toda información adicional que el desarrollador considere importante compartir. La sección **Institucional** es un complemento de la **Cabecera** y es parte de lo que se considera hoy día dentro de la organización esencial de una página web.

HTML 5 considera esta organización básica y provee nuevos elementos para diferenciar y declarar cada una de las secciones. Con esto, podemos decirle al navegador para qué es cada sección. La siguiente imagen muestra la organización de la página web mediante el uso de sus etiquetas:

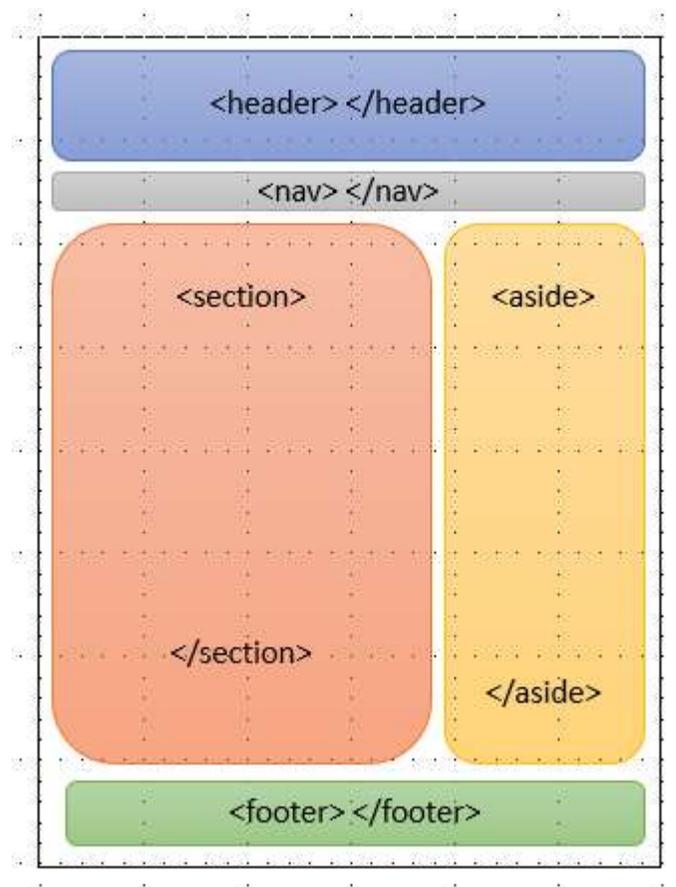


Imagen 38. Organización de una página web con elementos HTML 5.

<header>

Uno de los nuevos elementos que se incorporaron en HTML 5 es `<header>`. El elemento `<header>` no debe ser confundido con `<head>` usado antes para construir la cabecera del documento. Del mismo modo que `<head>`, la intención de `<header>` es proveer información introductoria (títulos, subtítulos, logos), pero difiere con respecto a `<head>` en su alcance. Mientras que el elemento `<head>` tiene el propósito de proveer información acerca de todo el documento, `<header>` es utilizado sólo para el cuerpo o secciones específicas dentro del cuerpo:

```

1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="iso-8859-1">
5     <meta name="description" content="Ejemplo de HTML5">
6     <meta name="keywords" content="HTML5, CSS3, JavaScript">
7     <title>Este texto es el título del documento</title>
8     <link rel="stylesheet" href="misestilos.css">
9   </head>
10  <body>
11  <header>
12    <h1>Este es el título principal del sitio web</h1>
13  </header>
14  </body>
15 </html>

```

`<nav>`

Siguiendo con el ejemplo, la siguiente sección es la **Barra de Navegación**. Esta barra es generada en HTML5 con el elemento `<nav>`:

```

1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="iso-8859-1">
5     <meta name="description" content="Ejemplo de HTML5">
6     <meta name="keywords" content="HTML5, CSS3, JavaScript">
7     <title>Este texto es el título del documento</title>
8     <link rel="stylesheet" href="misestilos.css">
9   </head>
10  <body>
11  <header>
12    <h1>Este es el título principal del sitio web</h1>
13  </header>
14  <nav>
15    <ul>
16      <li>Elemento 1</li>
17      <li>Elemento 2</li>
18      <li>Elemento 3</li>
19      <li>Elemento 4</li>
20    </ul>
21  </nav>
22  </body>
23 </html>

```

Como se puede apreciar, el elemento `<nav>` se encuentra dentro de las etiquetas `<body>` pero es ubicado después de la etiqueta de cierre de la cabecera (`</header>`), no dentro de las etiquetas `<header>`. Esto es porque `<nav>` no es parte

de la cabecera sino una nueva sección. Entre las etiquetas <nav> hay dos elementos que son utilizados para crear una lista. El propósito del elemento es definir la lista. Anidado entre las etiquetas se encuentran varias etiquetas con diferentes textos representando los elementos del menú. Las etiquetas , son usadas para definir cada ítem de la lista.

<section>

Como se explicó anteriormente, la columna **Información principal** contiene la información más relevante del documento y puede ser encontrada en diferentes formas. El elemento en HTML5 que especifica estas secciones se llama <section>:

```
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="iso-8859-1">
5     <meta name="description" content="Ejemplo de HTML5">
6     <meta name="keywords" content="HTML5, CSS3, JavaScript">
7     <title>Este texto es el título del documento</title>
8     <link rel="stylesheet" href="misestilos.css">
9   </head>
10  <body>
11  <header>
12    <h1>Este es el título principal del sitio web</h1>
13  </header>
14  <nav>
15    <ul>
16      <li>Elemento 1</li>
17      <li>Elemento 2</li>
18      <li>Elemento 3</li>
19      <li>Elemento 4</li>
20    </ul>
21  </nav>
22  <section>
23  |
24  </section>
25  </body>
26 </html>
```

Al igual que la **Barra de navegación**, la columna **Información Principal** es una sección aparte, es por ello que la etiqueta <section> va debajo de la etiqueta de cierre </nav>.

`<aside>`

La columna llamada **Barra lateral** se ubica normalmente al lado de la columna **Información principal**. Esta es una sección que contiene datos relacionados con la información principal pero que no son relevantes o igual de importantes.

```
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="iso-8859-1">
5     <meta name="description" content="Ejemplo de HTML5">
6     <meta name="keywords" content="HTML5, CSS3, JavaScript">
7     <title>Este texto es el título del documento</title>
8     <link rel="stylesheet" href="misestilos.css">
9   </head>
10  <body>
11    <header>
12      <h1>Este es el título principal del sitio web</h1>
13    </header>
14    <nav>
15      <ul>
16        <li>Elemento 1</li>
17        <li>Elemento 2</li>
18        <li>Elemento 3</li>
19        <li>Elemento 4</li>
20      </ul>
21    </nav>
22    <section>
23    </section>
24    </section>
25    <aside>
26      <p>Contenido mensaje 1</p>
27      <p>Contenido mensaje 2</p>
28    </aside>
29  </body>
30 </html>
```

El elemento `<aside>` podría estar ubicado del lado derecho o izquierdo de la página, esta etiqueta no tiene una posición predeterminada por lo que cada diseñador tiene total libertad para su acomodo.

`<footer>`

Finalizando la organización o estructura elemental de un documento HTML5, nos encontramos con un elemento específico para cerrar el diseño; `<footer>`:

```

1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4  |   <meta charset="iso-8859-1">
5  |   <meta name="description" content="Ejemplo de HTML5">
6  |   <meta name="keywords" content="HTML5, CSS3, JavaScript">
7  |   <title>Este texto es el título del documento</title>
8  |   <link rel="stylesheet" href="misestilos.css">
9  </head>
10 <body>
11 <header>
12 |   <h1>Este es el título principal del sitio web</h1>
13 </header>
14 <nav>
15 <ul>
16 |   <li>Elemento 1</li>
17 |   <li>Elemento 2</li>
18 |   <li>Elemento 3</li>
19 |   <li>Elemento 4</li>
20 </ul>
21 </nav>
22 <section>
23 |
24 </section>
25 <aside>
26 |   <p>Contenido mensaje 1</p>
27 |   <p>Contenido mensaje 2</p>
28 </aside>
29 <footer>
30 |   Todos los derechos reservados; 2017-2018
31 </footer>
32 </body>
33 </html>

```

La sección llamada **Institucional** será definida por las etiquetas <footer>. Esto es debido a que la barra representa el final (o pie) del documento y esta parte normalmente es utilizada para compartir información general sobre el autor o la organización que está detrás del proyecto.

<article>

El diseño anterior es el más común y representa una organización esencial para sitios web de hoy día, estos sitios normalmente presentan información relevante dividida en partes que comparten similares características. El elemento <article> permite que identifiquemos cada una de estas partes:

```

1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4  |   <meta charset="iso-8859-1">
5  |   <meta name="description" content="Ejemplo de HTML5">
6  |   <meta name="keywords" content="HTML5, CSS3, JavaScript">
7  |   <title>Este texto es el título del documento</title>
8  |   <link rel="stylesheet" href="misestilos.css">
9  </head>
10 <body>
11 <header>
12 |   <h1>Este es el título principal del sitio web</h1>
13 </header>
14 <nav>
15 |   <ul>
16 |     <li>Elemento 1</li>
17 |     <li>Elemento 2</li>
18 |     <li>Elemento 3</li>
19 |     <li>Elemento 4</li>
20 |   </ul>
21 </nav>
22 <section>
23 |   <article>
24 |     Texto del mensaje 1
25 |   </article>
26 |   <article>
27 |     Texto del mensaje 2
28 |   </article>
29 </section>
30 <aside>
31 |   <p>Contenido mensaje 1</p>
32 |   <p>Contenido mensaje 2</p>
33 </aside>
34 <footer>
35 |   Todos los derechos reservados; 2017-2018
36 </footer>
37 </body>
38 </html>
39 </html>

```

Las etiquetas `<article>` se encuentran ubicadas dentro del elemento `<section>`. En el ejemplo pertenecen a esta sección, son hijos, del mismo modo que cada elemento dentro de las etiquetas `<body>` es hijo del cuerpo. Al igual que cada elemento hijo del cuerpo, las etiquetas `<article>` son ubicadas una sobre otra¹⁴.

¹⁴ GAUCHAT, JUAN DIEGO, *Op. Cit.*, pp. 9-20

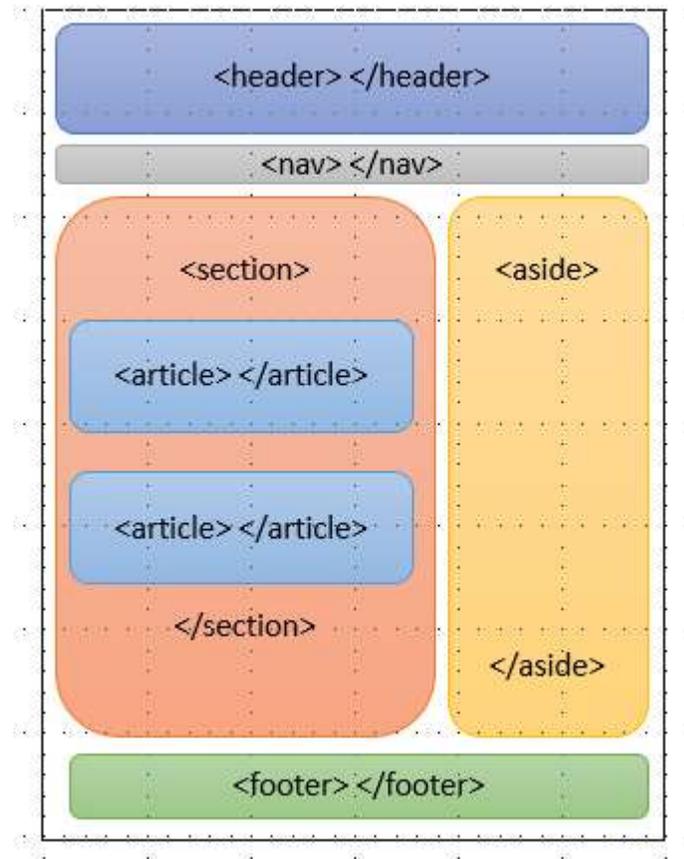


Imagen 39. Representación visual de etiquetas `<article>`.

Versionado

El proyecto se llevó a cabo mediante la modificación de la versión inicial, obteniendo 4 maquetas diferentes. La primera versión consta de un diseño muy simple, un listado sobre los distintos artículos que conforman el proyecto. La siguiente imagen muestra el diseño inicial para el proyecto, con una organización y estructura básica de HTML5:

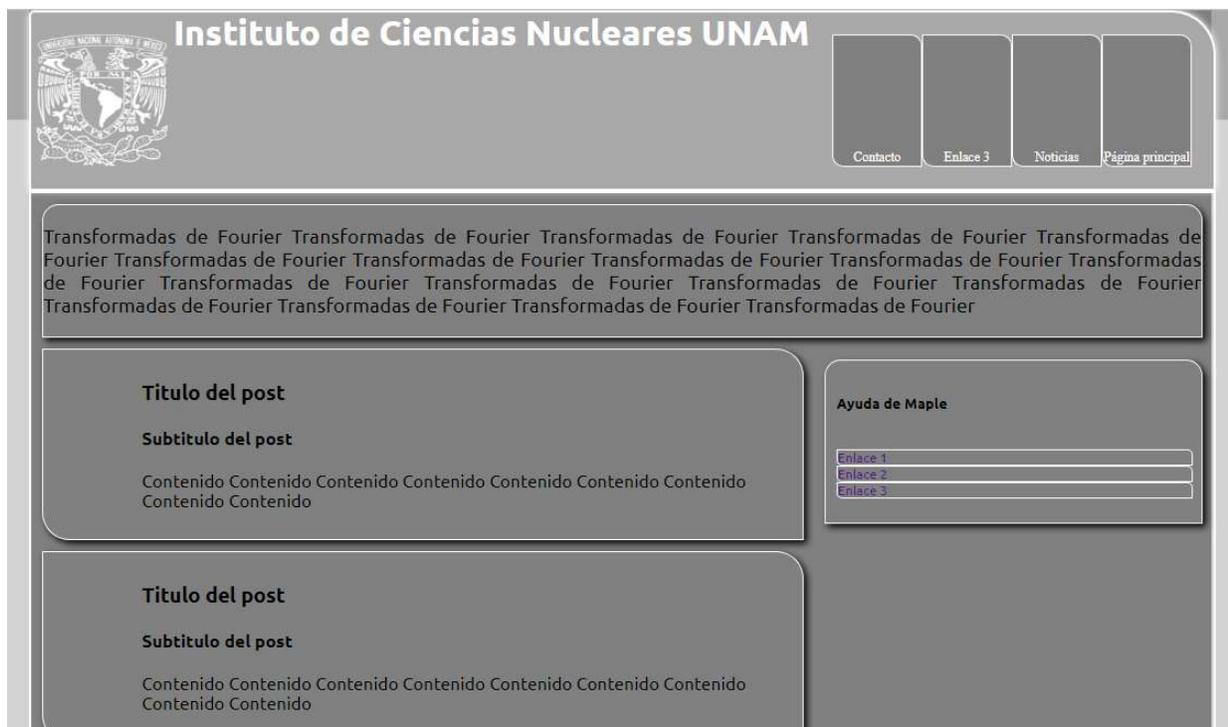


Imagen 40. Diseño inicial con organización elemental.

Debido a la simplicidad del diseño anterior, se realizaron modificaciones considerables para mejorar la apariencia del sitio, dando como resultado el diseño dos, el cual puede apreciarse mediante la siguiente imagen:

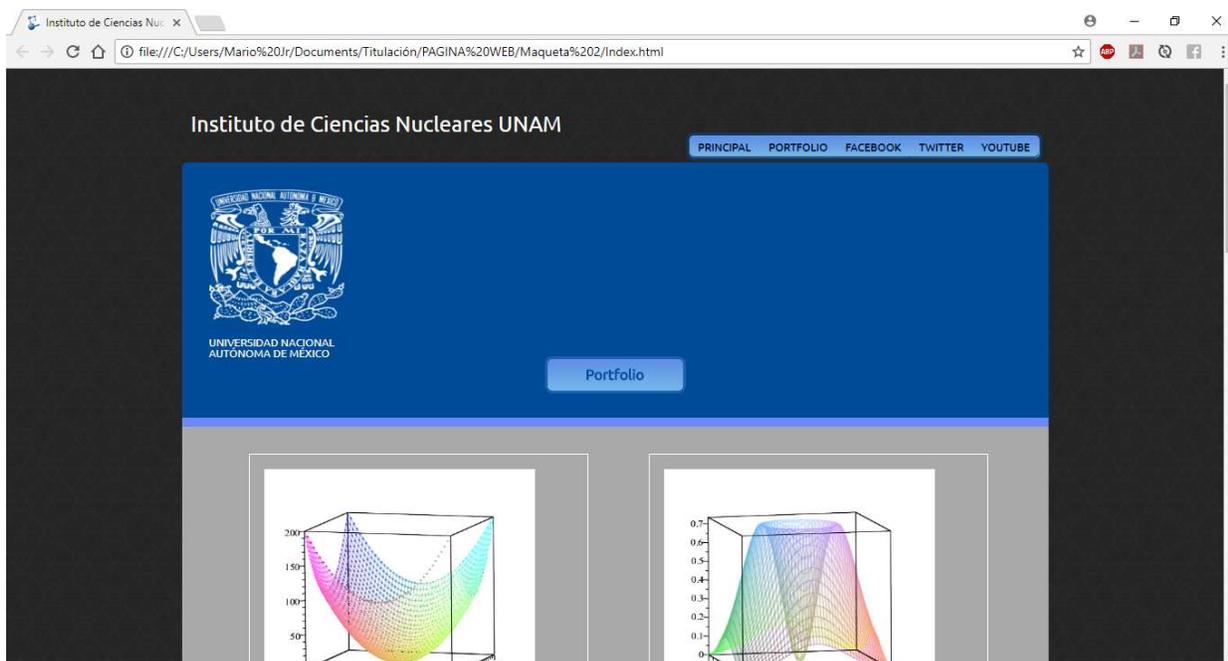


Imagen 41. Segundo diseño sin utilización de frameworks.

Para la tercera versión, el proyecto toma un giro radical, ya que está basado en un sistema de rejillas utilizando Flex Box, el diseño se implementó en Adobe Photoshop para posteriormente llevarlo a la escritura de código. Adicionalmente, se agregó como fondo, una animación de partículas con las que se pueden interactuar mediante la utilización del curso así como la adición del botón “ir arriba”. La siguiente imagen permite apreciar dichos cambios:



Imagen 42. Tercer diseño con integración de partículas y botón “ir arriba”.

Finalmente, tenemos el cuarto y último diseño, en el que se utiliza JavaScript, JQuery, Bootstrap 4, Animate.css, así como Flex Box y Responsive Design. Las tecnologías anteriores, permiten realizar sitios de manera más rápida, con presentación más profesional y, sobre todo responsivos, lo que quiere decir que se adaptan a las diferentes resoluciones que presentan los diversos dispositivos que hay en el mercado.

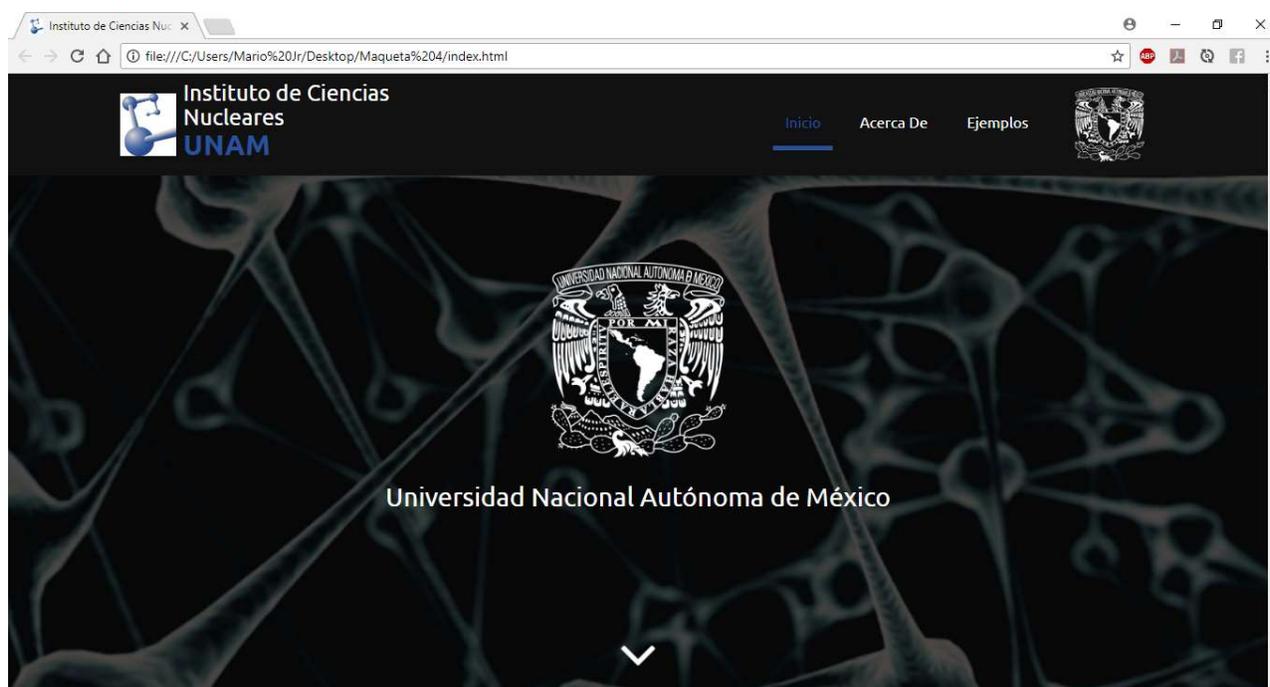


Imagen 43. Encabezado del proyecto final.



Imagen 44. Sección objetivo del proyecto final.

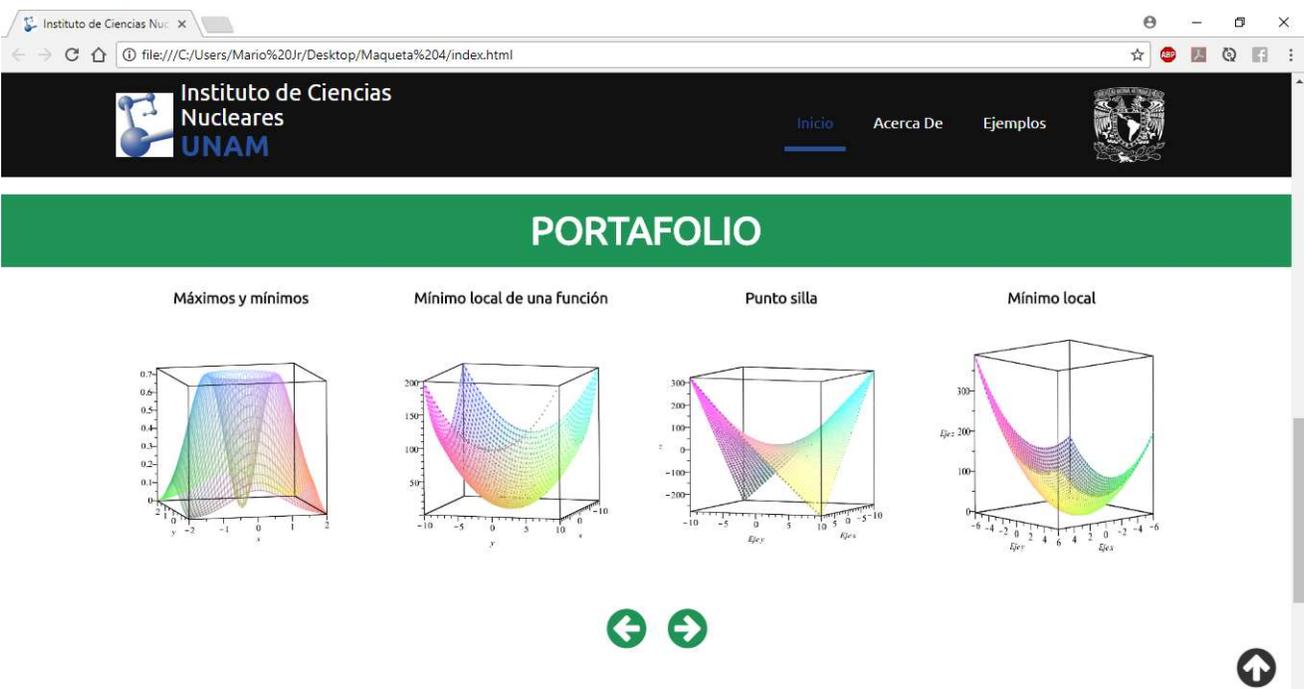


Imagen 45. Sección portafolio con efecto carrusel.

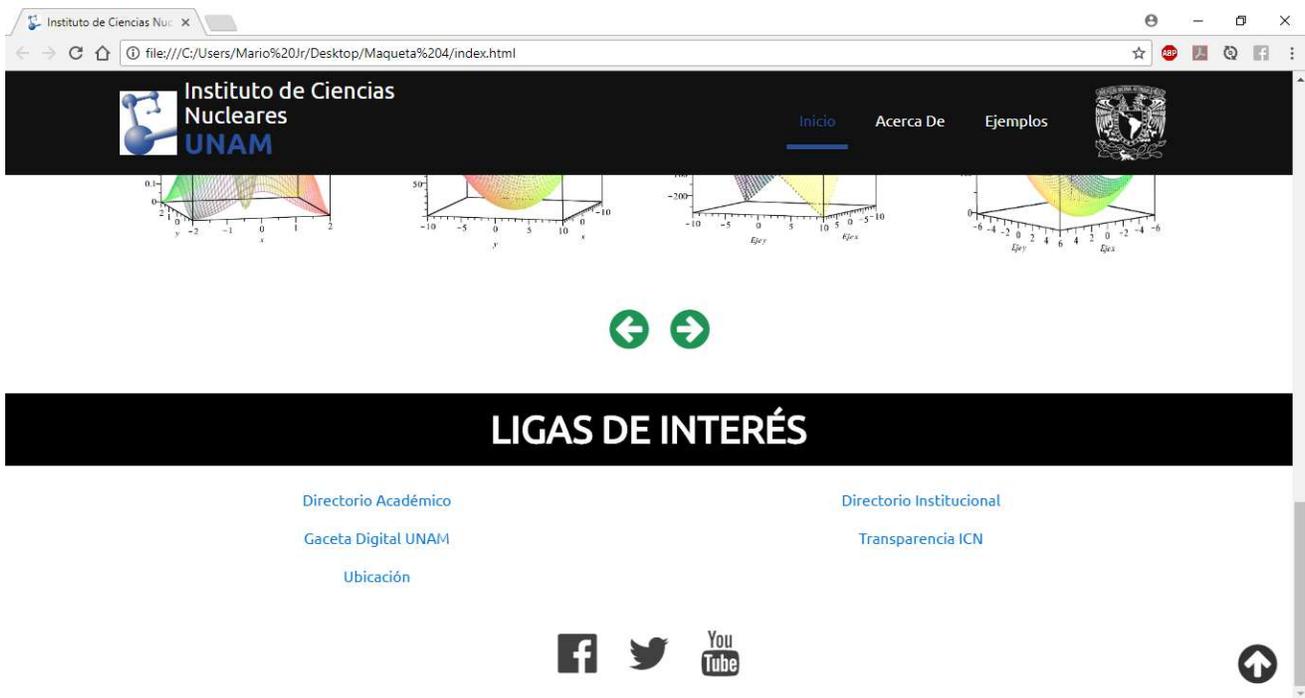


Imagen 46. Sección footer del diseño final.

Contenido académico

El proyecto final consta de 3 secciones importantes, la cuales corresponden a cada una de las ramas de estudio; Cálculo Vectorial, Matemáticas Aplicadas y Probabilidad y Estadística. Estas secciones se pueden visualizar seleccionando la pestaña de “Acerca De”, la cual lleva directamente al contenido que describe cada una de las ramas.

Si lo que se desea es visualizar el contenido disponible para cada sección, se puede visualizar desde la pestaña “Ejemplos”. En la parte inferior de la página se encuentra un botón que dice “Ver todos los ejemplos”, botón que al presionarlo redirige a un listado por sección con su respectivo contenido.

RESULTADOS

Estructura de la página

El resultado final para este proyecto queda reflejado con el código HTML que se muestra en el apéndice A, el cual contiene 281 líneas. Cabe mencionar que para que se pueda apreciar todo el contenido correctamente se debe hacer uso de todos los archivos que se enlazan en la página principal.

Detalles técnicos

El proyecto fue creado con herramientas de código abierto como lo son los editores de texto *Brackets* y *Atom*, así como la utilización del plugin *Emmet*, el cual permite la escritura de código HTML de una manera más rápida mediante el uso de atajos para escritura de etiquetas con una gran variedad de atributos.

Las animaciones que se encuentran presentes en el proyecto fueron posibles utilizando librerías como *JQuery*, *CSS3*, *Animate.css* y *Smooth Scroll*. Frameworks como Bootstrap 4 y FlexBox. A continuación, se describe brevemente las características de cada librería.

JQuery

Es una biblioteca multiplataforma de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción a páginas web¹⁵.

CSS3

CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

¹⁵ <https://es.wikipedia.org/wiki/JQuery> consultado Mayo 05, 2018 14:32 hrs.

La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "**<style>**".

CSS es un lenguaje utilizado para dar estética a un documento HTML (colores, tamaños de las fuentes, tamaños de elemento, con css podemos establecer diferentes reglas que indicarán como debe presentarse un documento¹⁶.

Animate.css

Es una pequeña librería que nos ayudará a disponer de un gran número de animaciones CSS3, las cuales pueden probarse desde su página oficial: <https://daneden.github.io/animate.css>¹⁷.

Smooth Scroll

Es una herramienta que permite realizar desplazamientos dentro de una página web de una manera elegante y suave mediante la modificación de la velocidad de desplazamiento. Esta librería hace uso de JavaScript¹⁸.

¹⁶ <https://www.ecured.cu/CSS3> consultado Mayo 05, 2018 14:37 hrs.

¹⁷ <https://devcode.la/tutoriales/crear-animaciones-css3-con-animatecss/> consultado Mayo 05, 2018 14:44 hrs.

¹⁸ <http://www.ribosomatic.com/articulos/smooth-scroll-desplazate-con-estilo-por-una-pagina-web/> consultado Mayo 05, 2018 14:45 hrs.

CONCLUSIONES

De acuerdo con los objetivos planteados y los resultados obtenidos, es posible confirmar el cumplimiento de dichos objetivos ya que se implementó un medio educativo cuyo contenido es para apoyar tanto a profesores, alumnos y público en general, estando al alcance de los mismos sin importar el área en la que las personas se especialicen.

Este proyecto fomenta el uso de las Tecnologías de la Información para uso académico con la ejemplificación de problemas realizados con software de cálculo simbólico, mostrando una gran variedad de escenarios en las que éste puede ser aplicado. Además, debido al incremento exponencial para el acceso a internet mediante un dispositivo móvil, este medio educativo de aprendizaje puede ser visualizado desde cualquier dispositivo con acceso a internet, ya que se adapta a las diferentes resoluciones de los dispositivos que se encuentran en el mercado. Así mismo, asegura la correcta visualización del contenido en cada uno de los diferentes navegadores o *browsers*, por lo que no está limitado a utilizar uno en específico.

Los códigos que se alojan en cada sección están explicados claramente, con la finalidad de orientar al lector a adaptarlos a sus necesidades. Este proyecto no pretende ser una guía del software utilizado sino una referencia de lo que se puede hacer con el mismo y en qué ramas.

El resultado final, refleja la originalidad con la que se creó el medio educativo, además de utilizar tecnologías que actualmente son muy cotizadas en el entorno laboral. Cabe destacar que el propósito de realizar un diseño desde cero, sin la necesidad de utilizar plantillas gratuitas que fácilmente se pueden encontrar en la internet, es poner en marcha los conocimientos adquiridos durante nuestra formación como ingenieros, buscar los recursos y no esperar a que estos lleguen por sí solos, buscar soluciones precisas para satisfacer las necesidades del cliente, hacer uso de las tecnologías más recientes y saberlas aplicar, indagar en la documentación las diferentes maneras para solucionar un mismo problema. Es cierto que las materias impartidas dentro de la facultad, brindan un cierto grado de conocimiento, sin embargo, no son suficientes para desempeñar un buen papel una

vez que entramos al mundo laboral, por lo que realizar investigación, se vuelve fundamental.

Otro de los puntos importantes por lo que no se utilizó una plantilla creada por alguien más, es mostrar el ingenio que conlleva la integración de diferentes tecnologías para convivir en un mismo ambiente, con las que podemos llegar a crear un medio con un aspecto y funcionalidad tan bueno, como algo que ya se vende en el mercado.

Dos de los objetivos educacionales importantes que tiene la carrera de ingeniería en computación, es “Los egresados identifican las perspectivas y oportunidades de negocio y pueden usufructuarlas con innovación, creatividad y emprendimiento, utilizando sus conocimientos de ingeniería en computación” y “Los egresados comprenden la importancia de la relación entre la teoría, la práctica y el aprendizaje continuo, pudiendo continuar su educación mediante cursos disciplinares, diplomados o estudios de posgrado”, entre otros. De tal manera que, “El egresado de Ingeniería en Computación tendrá sólidas bases científicas y fundamentos tecnológicos, que le permitan comprender, analizar, diseñar, organizar, producir, operar y dar soluciones prácticas a problemas relacionados con las áreas de Organización de Sistemas Computacionales, Ingeniería en Software y Tecnologías de Información¹⁹.”.

Gracias a estas afirmaciones, la aplicación de los conocimientos adquiridos en materias básicas como *Computación para Ingenieros*, hasta materias de competencias profesionales como *Ingeniería de Software*, *Administración de Proyectos de Software*, *Seguridad Informática*, *Criptografía*, *Sistemas Operativos*, *Lenguajes de Programación*, *Inteligencia Artificial*, etc., permitieron crear el medio educativo que durante todo el trabajo se expone. Todo el desarrollo, involucra algo de las materias anteriormente mencionadas, y con ello, la teoría se pone en práctica, y al realizar esto, generamos conocimiento, el cual podemos aplicar en la resolución de problemas.

¹⁹ http://www.ingenieria.unam.mx/programas_academicos/licenciatura/computacion.php consultado Mayo 15, 2019 14:11 hrs.

REFERENCIAS

Bibliografía

- GAUCHAT, JUAN DIEGO. *El gran libro de HTML5, CSS3 y Javascript*. MARCOMBO, S.A., 2012, España, p. 1.

Ciberografía

- <https://es.wikipedia.org/wiki/Framework>
- <https://devcode.la/blog/frontend-y-backend/>
- <https://www.addlink.es/productos/maple>
- <https://www.maplesoft.com/products/maple/history/>
- https://www.uoc.edu/in3/emath/docs/UV_Maple.pdf
- <https://es.ccm.net/contents/223-ciclo-de-vida-del-software>
- <https://ingsw.pbworks.com/f/Ciclo+de+Vida+del+Software.pdf>
- <https://abalozz.es/que-es-un-preprocesador-de-css/>
- <https://aprende-web.net/librerias/>
- <https://librosweb.es/libro/css/capitulo-1.html>
- https://www.ecured.cu/Navegador_web
- http://www.ingenieria.unam.mx/programas_academicos/licenciatura/computacion.php
- <https://www.w3schools.com/html/>
- http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf
- https://es.wikipedia.org/wiki/Navegador_web
- <https://aprende-web.net/librerias/>
- <https://tecnologia-informatica.com/que-es-hardware-y-software/>
- <http://interactivepython.org/runestone/static/pythoned/Introduction/QueEsProgramacion.html>
- <https://conceptodefinicion.de/pagina-web/>
- <https://www.definicionabc.com/tecnologia/codigo-programacion.php>
- <https://developer.mozilla.org/es/docs/DOM>

APÉNDICE A

index.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
```

```
<meta http-equiv="x-ua-compatible" content="ie=edge">
```

```
<title>Instituto de Ciencias Nucleares UNAM</title>
```

```
<link rel="shortcut icon" href="favicon.ico">
```

```
<!-- Carga de fuentes -->
```

```
<link rel="stylesheet" href="css/font-awesome.min.css">
```

```
<!-- Carga de archivos para galeria de imágenes -->
```

```
<link rel="stylesheet" href="css/owl.carousel.min.css">
```

```
<!-- Carga de archivos css -->
```

```
<link rel="stylesheet" href="css/bootstrap.css">
```

```
<link rel="stylesheet" href="css/animate.min.css">
```

```
<link rel="stylesheet" href="css/estilos.css">
```

```
</head>
```

```
<body>
```

```

<section class="bienvenidos">

  <header class="encabezado navbar-fixed-top" role="banner" id="encabezado">

    <div class="container">

      <a href="http://www.nuclecu.unam.mx/" target="_blank" class="logo">

      </a>

      <a class="Instituto" href="http://www.nuclecu.unam.mx/" target="_blank">

        <h2>Instituto de Ciencias<br>Nucleares <br>

        <span class="unam">UNAM</span>

        </h2>

      </a>

      <button type="button" class="boton-menu hidden-lg-up" data-toggle="collapse"
data-target="#menu-principal" aria-expanded="false">

        <i class="fa fa-bars" aria-hidden="true"></i>

      </button>

      <!-- <button type="button" class="boton-buscar" data-toggle="collapse" data-
target="#bloque-buscar" aria-expanded="false">

        <i class="fa fa-search" aria-hidden="true"></i>

      </button> -->

      <a class="enlace-unam" href="https://www.unam.mx"></a>

      <form action="#" id="bloque-buscar" class="collapse">

```

```
<div class="contenedor-bloque-buscar">
  <input type="text" placeholder="Buscar">
  <input type="submit" value="Buscar">
</div>
</form>

<nav id="menu-principal" class="collapse">
  <ul>
    <li class="active"><a href="index.html">Inicio</a></li>
    <li><a href="acerca.html">Acerca de</a></li>
    <li><a href="ejemplos.html">Ejemplos</a></li>
  </ul>
</nav>
</div>
</header>

<div class="escudo-encabezado text-xs-center">
  <div class="container">
    <a href="https://www.unam.mx" target="_blank"></a>
    <h3 class="wow bounceInUp" data-wow-delay="0.3s">Universidad Nacional
Autónoma de México</h3>
  </div>
</div>

<div class="flecha-bajar text-xs-center">
```

```
<a data-scroll href="#Objetivo"><i class="fa fa-angle-down" aria-  
hidden="true"></i></a>
```

```
</div>
```

```
</section>
```

```
<section class="AcercaDe p-b-1" id="Acercade">
```

```
<p class="h1 text-xs-center font-weight-bold" id="Objetivo">OBJETIVO</p>
```

```
<div class="container">
```

```
<div class="row">
```

```
<div class="col-xs-12 col-md-6 col-xl-7 wow bounceIn" data-wow-delay="0.3s">
```

```
<p class="text-justify ">Este proyecto fue realizado con el propósito de facilitar la  
utilización de herramientas de cálculo simbólico;
```

```
como Maple, para la resolución de problemas matemáticos, así como la  
visualización del comportamiento de diversas funciones.
```

```
</p>
```

```
</div>
```

```
<div class="Maple text-xs-center col-xs-12 col-md-6 col-xl-5">
```

```

```

```
</div>
```

```
</div>
```

```
</div>
```

```
</section>
```

```
<section class="hacer-maple">
```

```
<div class="container p-y-1">
```

```

<br>

<p class="h2 text-xs-center">¿Qué puedes hacer con <span class="font-weight-
bold">Maple</span>?</p>

<div class="row p-y-2">

  <ul class="text-xs-center col-xs-12 col-sm-6 col-lg-3">

    <li class="col-md-3 wow bounceIn" data-wow-delay="0.5s"><i class="fa fa-
exchange" aria-hidden="true"></i>

      <p class="h5">Cálculo Vectorial</p>

      <a href="ejemplos.html"><button type="button" name="button" class="btn btn-
secondary btn-bajar">Ver ejemplos</button></a>

    </li>

  </ul>

  <ul class="text-xs-center col-xs-12 col-sm-6 col-lg-3">

    <li class="col-md-3 wow bounceIn" data-wow-delay="0.7s"><i class="fa fa-cube"
aria-hidden="true"></i>

      <p class="h5">Funciones en 3D</p>

      <a href="ejemplos.html"><button type="button" name="button" class="btn btn-
secondary">Ver ejemplos</button></a>

    </li>

  </ul>

  <ul class="text-xs-center col-xs-12 col-sm-6 col-lg-3">

    <li class="col-md-3 wow bounceIn" data-wow-delay="0.9s"><i class="fa fa-rocket"
aria-hidden="true"></i>

      <p class="h5">Matemática Aplicada</p>

      <a href="ejemplos.html"><button type="button" name="button" class="btn btn-
secondary">Ver ejemplos</button></a>

    </li>

```

```
</ul>
```

```
<ul class="text-xs-center col-xs-12 col-sm-6 col-lg-3">
```

```
  <li class="col-md-3 wow bounceIn" data-wow-delay="1.1s"><i class="fa fa-bar-chart" aria-hidden="true"></i>
```

```
    <p class="h5">Probabilidad y Estadística</p>
```

```
    <a href="ejemplos.html"><button type="button" name="button" class="btn btn-secondary">Ver ejemplos</button></a>
```

```
  </li>
```

```
</ul>
```

```
</div>
```

```
</div>
```

```
</section>
```

```
<section class="portfolio text-xs-center p-y-1" id="Portfolio">
```

```
<p class="h1 text-xs-center font-weight-bold">PORTAFOLIO</p>
```

```
<div class="container text-xs-center">
```

```
<div class="owl-carousel text-xs-center">
```

```
<a href="#">
```

```
<h4>Máximos y mínimos</h4>
```

```

```

```
</a>
```

```
<a href="#">
```

```
<h4>Mínimo local de una función</h4>
```

```

```

```
</a>
```


<h4>Punto silla</h4>

<h4>Mínimo local</h4>

<h4>Mínimo relativo de una función 2D</h4>

<h4>Aproximación de serie trigonométrica de Fourier de función par</h4>

<h4>Aproximación de serie trigonométrica de Fourier de función impar</h4>

<h4>Función continua con intervalo finito</h4>

<h4>Función periódica con intervalo finito</h4>

<h4>Función intervalo simétrico</h4>

<h4>Función de Densidad</h4>

<h4>Mínimo relativo de una función 2D</h4>

<h4>Mínimo relativo de una función 2D</h4>

<h4>Punto silla</h4>


```
<a href="#">
  <h4>Punto silla y mínimo local</h4>
  
</a>
</div>
</div>
</section>
```

```
<footer class="piedepagina text-xs-center p-y-1" role="contentinfo">
  <p class="h1 text-xs-center font-weight-bold">LIGAS DE INTERÉS</p>
  <div class="container p-y-1">
    <div class="row">
      <ul class="text-xs-center col-xs-12 col-md-6">
        <li><a href="#">Directorio Académico</a></li>
      </ul>
      <ul class="text-xs-center col-xs-12 col-md-6">
        <li><a href="#">Directorio Institucional</a></li>
      </ul>
      <ul class="text-xs-center col-xs-12 col-md-6">
        <li><a href="#">Gaceta Digital UNAM</a></li>
      </ul>
      <ul class="text-xs-center col-xs-12 col-md-6">
        <li><a href="#">Transparencia ICN</a></li>
      </ul>
      <ul class="text-xs-center col-xs-12 col-md-6">
```

```

        <li><a href="#">Ubicación</a></li>

    </ul>

</div>

</div>

<div class="container text-xs-center">

    <ul class="redes-sociales">

        <li><a href="#"><i class="fa fa-facebook-official" aria-hidden="true"></i></a></li>

        <li><a href="#"><i class="fa fa-twitter" aria-hidden="true"></i></a></li>

        <li><a href="#"><i class="fa fa-youtube" aria-hidden="true"></i></a></li>

    </ul>

</div>

</footer>

<a data-scroll href="#encabezado" class="ir-arriba"><i class="fa fa-arrow-circle-up" aria-
hidden="true"></i></a>

<!-- Carga de archivos JS -->

<script src="js/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<script src="js/owl.carousel.min.js"></script>

<script type="text/javascript">

    $($('.owl-carousel').owlCarousel({

        loop: true,

        margin: 0,

        nav: true,

        autoWidth: false,

```

```
navText: ['<i class="fa fa-arrow-circle-left" title="Anterior"></i>', '<i class="fa fa-  
arrow-circle-right" title="Siguiente"></i>'],
```

```
responsive: {
```

```
  0: {
```

```
    items: 1
```

```
  },
```

```
  500: {
```

```
    items: 2,
```

```
    margin: 20
```

```
  },
```

```
  800: {
```

```
    items: 3,
```

```
    margin: 20
```

```
  },
```

```
  1000: {
```

```
    items: 4,
```

```
    margin: 20
```

```
  }
```

```
}
```

```
})
```

```
</script>
```

```
<script src="js/wow.min.js"></script>
```

```
<script src="js/smooth-scroll.min.js"></script>
```

```
<script type="text/javascript">
```

```

new WOW().init();

/*-----
Iniciamos smoothScroll (Scroll Suave)
-----*/

smoothScroll.init({
    speed: 1000, // Integer. How fast to complete the scroll in milliseconds
    offset: 110, // Integer. How far to offset the scrolling anchor location in pixels
});

/*-----
OCULTAR Y MOSTRAR BOTON IR ARRIBA
-----*/

$(function () {
    $(window).scroll(function () {
        var scrolltop = $(this).scrollTop();
        if (scrolltop >= 50) {
            $(".ir-arriba").fadeIn();
        } else {
            $(".ir-arriba").fadeOut();
        }
    });
});

});

</script>

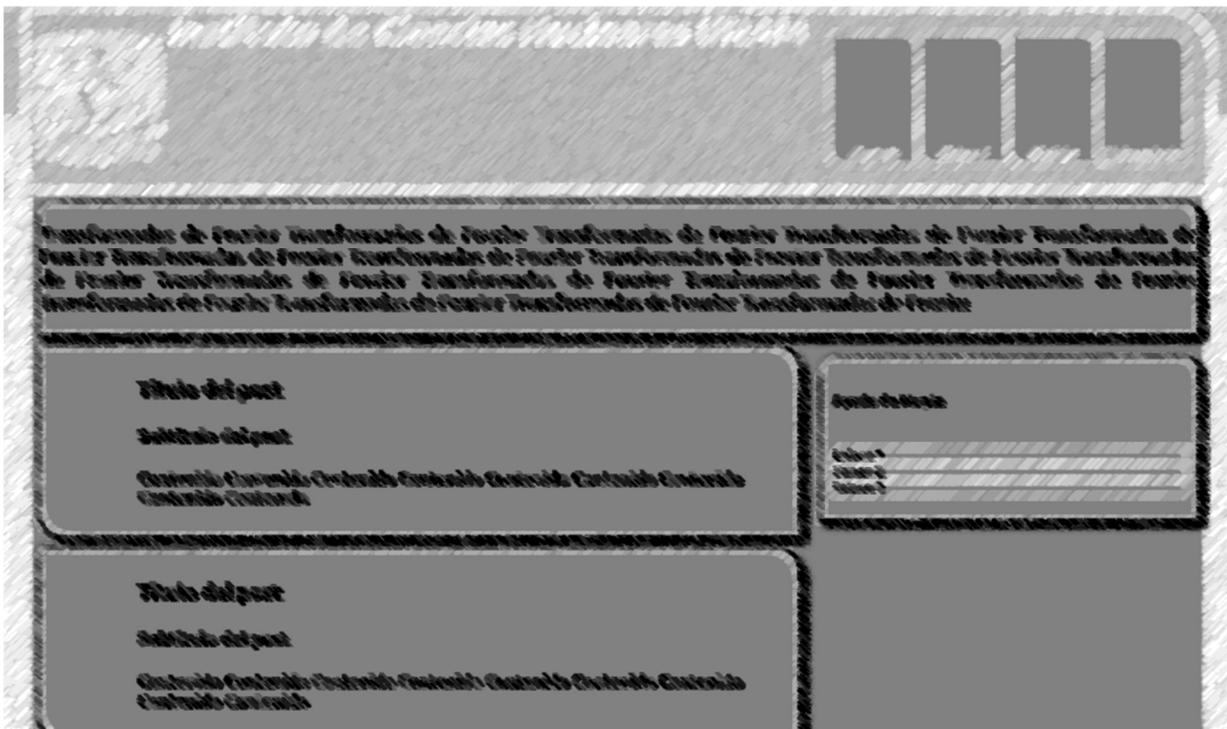
</body>

</html>

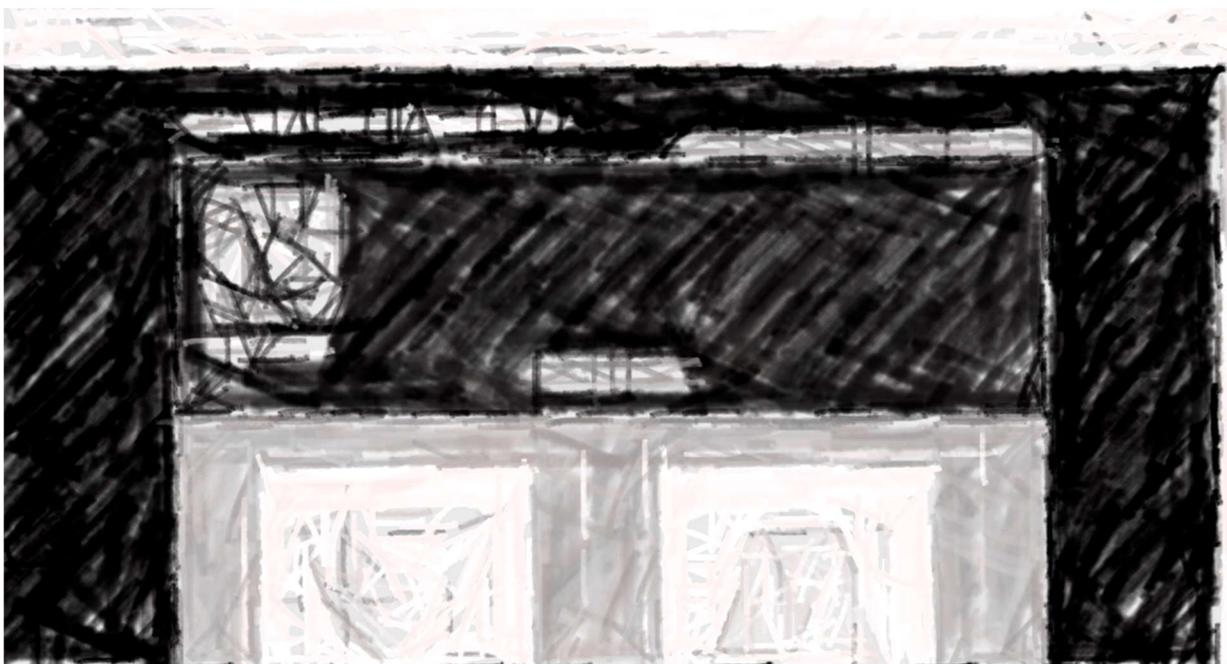
```

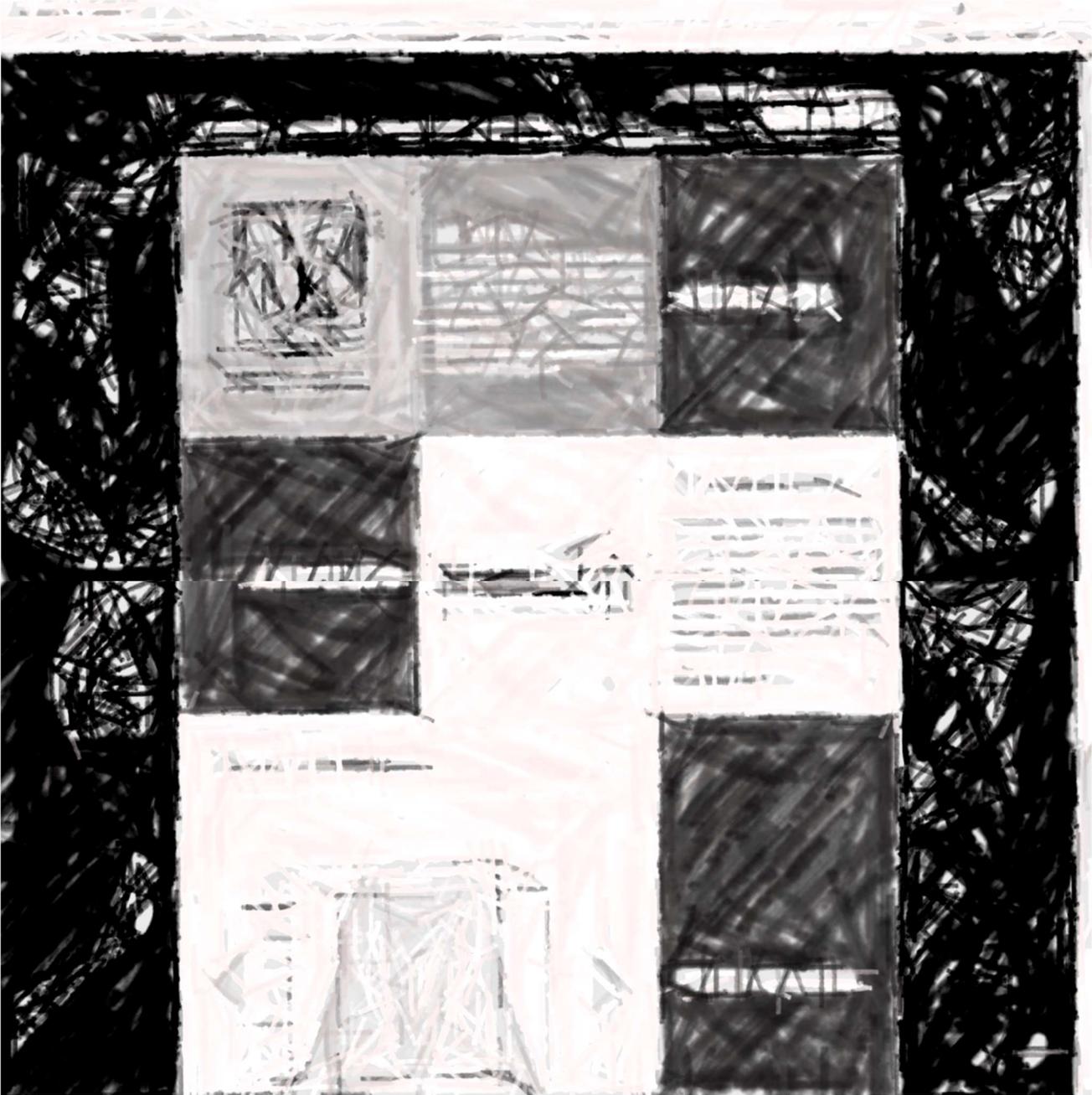
APÉNDICE B

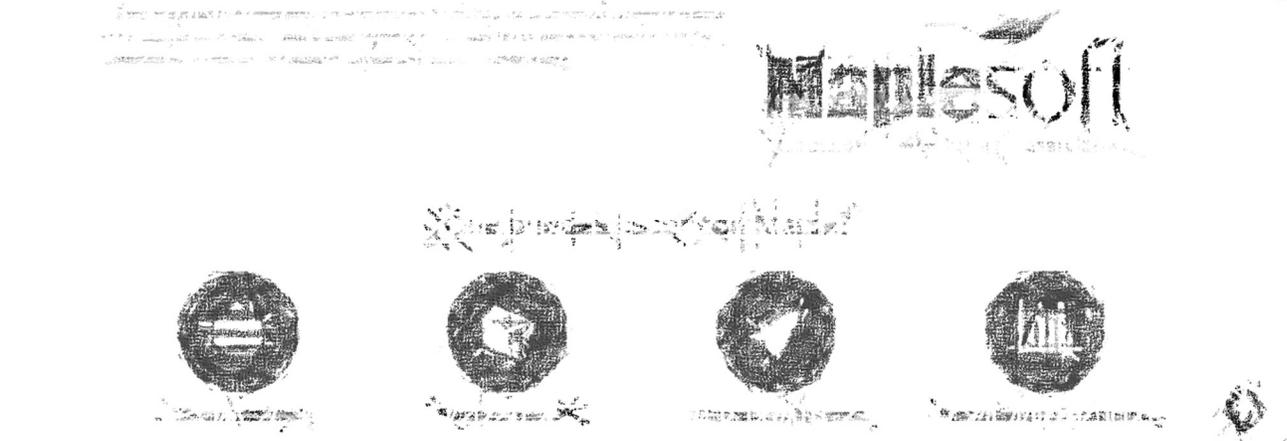
Mockup – Versión 1

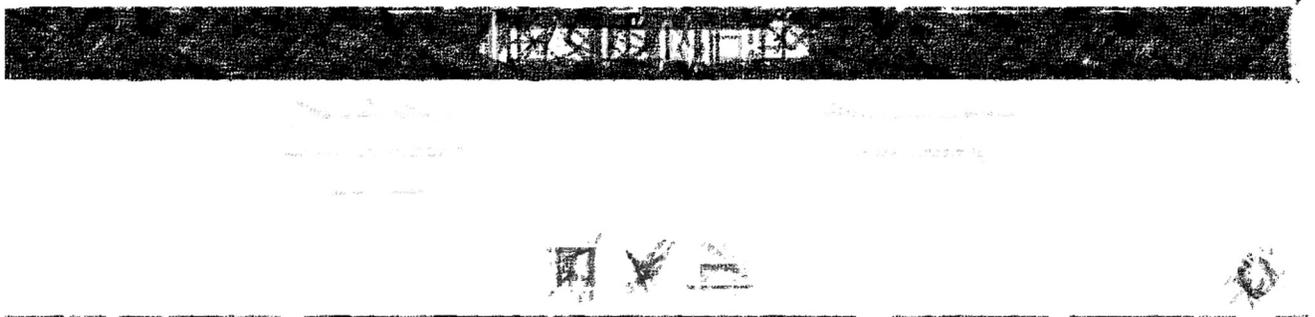
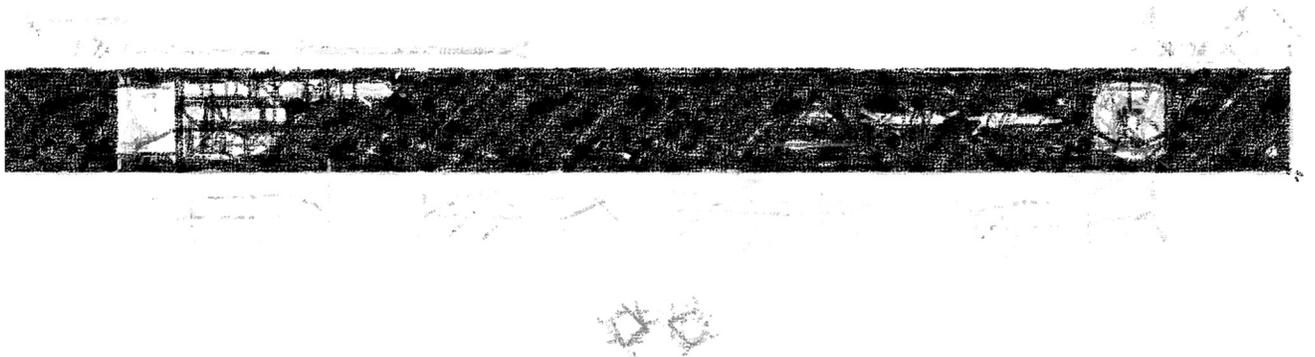


Mockup – Versión 2









GLOSARIO

A

Algoritmo: describe la solución a un problema en términos de los datos requeridos para representar el caso del problema y el conjunto de pasos necesarios para producir el resultado pretendido.

C

Código: es un conjunto de líneas de texto con los pasos que debe seguir la computadora para ejecutar un programa, puede ser binario (interpretable únicamente por computadoras) o como código fuente (interpretable por seres humanos).

D

Debugging: es el proceso de identificar y corregir errores de programación.

DOM (Document Object Model): Se define como Modelo de Objetos del Documento. Proporciona una representación estructural del documento, permitiendo la modificación de su contenido o su presentación visual. Esencialmente, comunica las páginas web con los scripts (conjunto de instrucciones) o los lenguajes de programación.

F

Framework: aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. Conjunto de componentes que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web.

L

Librería: es un archivo o conjunto de archivos que se utilizan para facilitar el desarrollo web.

M

Mockup: borrador, maqueta o bosquejo.

N

Navegador: es un software, aplicación o programa que permite el acceso a la Web, interpretando la información de distintos tipos de archivos y sitios web para que estos puedan ser visualizados por el usuario.

P

Página Web (Web Page): documento de tipo electrónico, el cual contiene información digital, la cual puede venir dada por datos visuales y/o sonoros, o una mezcla de ambos, a través de textos, imágenes, gráficos, audio y/o videos.

Programación: es el proceso de tomar un algoritmo y codificarlo en una notación, un lenguaje de programación, de modo que pueda ser ejecutado por una computadora.

S

Software: secuencia de instrucciones que son interpretadas y/o ejecutadas por un procesador para la implementación de una tarea en particular.