



# MODULO 1

## Ejercicio 05\* Responsive design

### Descripción breve

Un **framework** es una herramienta de desarrollo **web** que, por lo general, se define como una aplicación o conjunto de módulos que permiten el desarrollo ágil de aplicaciones mediante la aportación de librerías y/o funcionalidades ya creadas.

Carlos Arias  
carlos.arias@alumnos.udg.mx



## Contenido

Contenido .....	2
Bibliografía .....	10

## Contenido

- ¿Qué es un framework?

Son un conjunto de herramientas, estilos y librerías dispuestas a través de una estructura o esqueleto base, para el desarrollo de aplicaciones web más escalables y sencillas de mantener.

- ¿Qué es un diseño responsivo en el desarrollo web?

Es una técnica de diseño web que busca la correcta visualización de una misma página en distintos dispositivos. Desde ordenadores de escritorio a tablets y móviles.

- ¿Cuáles son las características del diseño responsivo?



- ¿Qué funciones tiene los framework de diseño web?

En el caso del diseño web, los frameworks nos dan una serie de clases CSS y funciones de JavaScript que podemos aplicar directamente en nuestra página, además de ofrecer un esqueleto basado en columnas que facilita mucho el construir una web que se adapte a cualquier dispositivo.

- ¿Cuáles son las ventajas e inconvenientes de usar un framework?

Ventajas:

Estructura y organización del código predeterminada. Los frameworks proporcionan tanto un esqueleto como una forma de trabajar. Por lo tanto, evitan tener que realizar un análisis sobre dónde situar los diferentes archivos de la aplicación (recursos, controladores, vistas, modelos, etc.).

Reutilización del código. Evitar duplicidad de código. En el desarrollo de una aplicación existen ciertos apartados que suelen repetirse, como la conexión con la base de datos, validación de formularios, páginas de estilos, etc. Con la utilización de un framework ahorraremos tiempo en desarrollar funcionalidades que ya están cubiertas y podremos enfocarnos en el funcionamiento de la aplicación más que en cómo llevarla a cabo.

Agilidad y rapidez en el desarrollo. Precisamente gracias a la reutilización de código mencionada anteriormente, conseguimos mayor rapidez en el desarrollo, ya que no perderemos tiempo en desarrollar funcionalidades nuevas.

Menor coste en el desarrollo. El coste es un parámetro que está directamente relacionado con la rapidez y agilidad. Acabar antes un proyecto implica que la dedicación es menor y por lo tanto el coste del proyecto también disminuye. Al igual que la rapidez en el desarrollo, esta ventaja beneficia tanto al cliente como al desarrollador.

Buenas prácticas de desarrollo con el uso de patrones. La mayoría de frameworks están basados en patrones de diseños, que nos indican pautas sobre cómo solucionar un problema específico que ya ha ocurrido con anterioridad. El patrón de diseño más popular es MVC (Modelo-Vista-Controlador), que nos ayuda a separar la capa de datos de la lógica del negocio de la interfaz con el usuario.

Minimizar errores y mayor facilidad para solucionarlos. Como el framework ya incorpora código implementado por otros programadores, los posibles errores que este pueda tener siempre serán menores que al desarrollarlo desde cero. Además, en caso de que hubiera un error, lo más probable es que ya haya sido solucionado por la comunidad.

Facilidad a la hora de encontrar una librería o código que ya cubra funcionalidades de tu desarrollo. Resulta más fácil encontrar herramientas (utilidades, librerías) adaptadas al framework que para un desarrollo propio.

Facilita la colaboración con otros desarrolladores. Tanto si son compañeros de tu equipo como de GitHub, leer el código desarrollado por otra persona puede resultar complejo. Sin embargo, si ya sabes qué estructura va a seguir el código y cómo se

organiza, resultará más fácil comprenderlo y poder aplicarle nuevos cambios. Lo que nos lleva a la siguiente ventaja.

Facilita el mantenimiento. Si todos los miembros de un equipo trabajan de la misma forma, en el momento que haya que actualizar la aplicación o realizar algún evolutivo, tardaremos menos tiempo y el coste será menor.

#### Desventajas:

Tiempo de aprendizaje. Antes de empezar a utilizar un framework debemos familiarizarnos con él, con cómo se estructuran sus archivos, con la forma en la que se comunican los componentes, etc. Por lo tanto, tendremos que invertir tiempo en superar la curva de aprendizaje para poder comenzar un nuevo desarrollo utilizando el framework.

Versiones inestables. El hecho de que los frameworks sean tan populares provoca que estén en constante actualización para cumplir con las nuevas tecnologías y las nuevas políticas de seguridad. Por ello, si en el desarrollo surgen incompatibilidades con otras librerías o se detectan errores de seguridad, la elección de una versión muy reciente del framework podría ralentizarnos.

Menor rendimiento. Los frameworks consumen, en general, más recursos que una aplicación creada desde cero y orientada al rendimiento. En aplicaciones muy exigentes, un framework puede resultar poco apropiado.

Código sin utilizar. Si la aplicación es pequeña o no requiere mucha funcionalidad, probablemente estaremos desaprovechando mucho código que viene implementado en el framework, por lo que la aplicación ocupará más espacio del que realmente necesitaría.

Elección del framework. Puede resultar complicado elegir cuál es el framework en el que te vas a especializar, ya que existe una gran variedad, y cada uno posee sus propias características. Por lo que, antes de lanzarse a aprender un framework nuevo, hay que dedicar tiempo a estudiar cuáles son los que tienen una base más sólida, ya que puede ocurrir que tras comenzar a desarrollar con un framework poco usado, se quede desactualizado al poco tiempo.

- ¿Qué es bootstrap y por qué es tan usado?

Bootstrap es un framework front-end utilizado para desarrollar aplicaciones web y sitios mobile first, o sea, con un layout que se adapta a la pantalla del dispositivo utilizado por el usuario.

Una de las características principales de Bootstrap es permitir que la adaptación de la página se realice según el tipo de dispositivo utilizado.

- ¿Qué es LESS?

Less (Leaner Style Sheets) es una ampliación retrocompatible del lenguaje CSS o, en otras palabras, un preprocesador del lenguaje de hoja de estilo. Eso quiere decir que cualquier código CSS es automáticamente también un código Less válido (pero no al revés).

- ¿Qué ventajas tiene LESS?

Utilizando Variables: Todo programador está familiarizado con el uso de variables y LESS nos da una gran ventaja con ellas, y solo es necesario poner un @ para poder utilizarlas ya sea para poner un color, una medida e incluso texto – @Variable: Valor;

Anidando: Otra ventaja que nos ofrece LESS es el poder anidar etiquetas, clases, id , así como las pseudoclasas y el lugar de estar escribiendo cada especificación por separado.

Utilizar Mixins: Ahora pasamos a los Mixin con los cuales se agrupan varias propiedades y se les asigna un identificador, también es posible pasarle valores a través de parámetros. Gracias a los Mixins se ahorra mucho tiempo y líneas escritas ya que al compilarlo cada uno se genera automáticamente con el valor que hemos pasado por el parámetro.

Operaciones y Funciones: Después de ver las grandes cosas que ofrece LESS te sorprenderás con lo que sigue “Operaciones y Funciones”, así es, LESS nos permite realizar diversas operaciones y ciertas funciones.

- ¿Qué es Sass?

Un preprocesador CSS es una herramienta que nos permite generar, de manera automática, hojas de estilo, añadiéndoles características que no tiene CSS, y que son propias de los lenguajes de programación, como pueden ser variables, funciones, selectores anidados, herencia, etcétera.

- ¿Cuáles son las principales diferencias de Less y Sass?

SASS se puede usar con dos formas de sintaxis diferentes: la forma original, a menudo llamada "indented syntax" (sintaxis de indentación), y una variante más nueva, que está más orientada a los requisitos de CSS y se denomina Sassy CSS (SCSS); es decir, CSS estilo SASS. Con la versión 3 de SASS, SCSS se ha

establecido como sintaxis oficial. La mayor diferencia entre ambas: el uso de paréntesis y del punto y coma.

La sintaxis original de SASS trabaja con sangrías y saltos de línea, un enfoque de estilo YAML. Para completar una línea de código, basta con hacer un salto de línea, es decir, pulsar la tecla Intro. La indentación funciona fácilmente con el tabulador. Al cambiar la posición en la tipografía, se forman agrupaciones, los llamados bloques de declaración, lo que no es posible en CSS. En este caso, deben utilizarse llaves para las agrupaciones y punto y coma para las declaraciones de propiedad. Y eso es exactamente lo que SCSS necesita.

Algunos usuarios prefieren el fácil manejo del SASS original, en el que no es necesario prestar atención a la correcta configuración de los separadores cuando se mueven fragmentos de código fuente y, en general, generar un código más limpio y claro. En general, la "sintaxis sangrada" se conforma con menos caracteres y líneas. Los seguidores de SCSS, por otro lado, prefieren aceptar este esfuerzo adicional, ya que se asemeja más a la sintaxis que ya conocen de CSS.

SCSS es un superconjunto de CSS, lo que asegura que el código CSS también funciona en SCSS, pero no al revés. Las funciones de SASS siguen estando plenamente incluidas, razón por la cual se hace más fácil trabajar con ambos idiomas al mismo tiempo. Además, para las personas que ya trabajan con CSS y se han acostumbrado a la sintaxis, el cambio es mucho más sencillo. Aunque SASS soporta ambas sintaxis, tú tienes que decidir cuál escoger para cada proyecto: para distinguir entre los diferentes formatos, se les da a los archivos la extensión .sass o .scss.

- ¿Qué son los grid system?

960.gs es un framework orientado a maquetar sitios web (o incluso aplicaciones) con el mínimo esfuerzo posible, pero limitados a (de ahí el nombre) 960 píxeles de ancho.

- ¿Cuáles son las librerías o frameworks basados en grid system más populares y menciona sus principales características?

Skeleton: Esta framework CSS con una cuadrícula que se ajusta al tamaño de diferentes resoluciones de manera limpia. Viene en un ligero paquete con sólo los archivos necesarios y básicos.

320 and up: Esta framework CSS se ha diseñado para la técnica Mobile First. Tiene elementos básicos para interfaces de usuario.

Base: Al igual que Skeleton, esta framework CSS nos ofrece un juego de características básicas.

Gumby Framework: Basada en la conocida framework 960, Gumby Framework ofrece opciones para personalizar el paquete de acuerdo al tipo de cuadrícula que requieras.

Wirefy: Esta framework CSS fue creada con la intención de desarrollar wireframes interactivas. Sin embargo, también se puede emplear para otro tipo de proyectos.

Amazium: Esta completa framework CSS es similar a Bootstrap y ofrece opciones para modificar elementos de interfaz de usuario, así como un poderoso sistema de cuadrículas.

Bootstrap: Es una de las framework CSS más populares y una de las más usadas. Ha sido creado por los desarrolladores de Twitter y ofrece todas las herramientas para crear aplicaciones web. Además, puedes elegir sólo las características que necesitemos dependiendo del proyecto que estés desarrollando.

Golden Grid System: Esta framework CSS ofrece un máximo de 16 columnas que funciona en cualquier resolución, no importa que tan alta sea. Luego, esas 16 columnas se convierten en 8 para tabletas y en 4 para teléfonos móviles.

Proportional Grids: Esta framework CSS ofrece una manera simple de crear cuadrículas con un medianil fijo.

Responsive Grid System: Esta es otra framework CSS sencilla donde puedes crear columnas de manera rápida y fácil.

- ¿Qué son los diseño sensibles?

El diseño web adaptativo o sensible (en inglés, Responsive Web Design) es una técnica de diseño y desarrollo web que mediante el uso de estructuras e imágenes fluidas, así como de media-queries en la hoja de estilo CSS, consigue adaptar el sitio web al entorno del usuario.

- ¿Menciona 20 frameworks o librerías para diseño web?

FICTOAN

Avalanche

Beauter

Vanilla Framework

Bulma

Milligram

InvisCss

Look



unlimitedGrid

Vital

PowerToCSS

Kouto Swiss

Furtive CSS

Webplate

Fluidity

Schema

Emerald

Bijou

Spark

Typebase

Hoisin

Cute Grids

Light but Mighty

- ¿Qué es Single-Page Applications (SPA) donde se utiliza y cuales son sus principales beneficios?

Single Page Application es una aplicación web de una sola pagina, es decir, la interacción de la aplicación es en una pagina. En una SPA el HTML, JavaScript, CSS se carga una sola vez.

Ejemplo YouTube, Netflix, Gmail, etc.

Agiliza la lógica de negocio al procesarse en lado del cliente como bien podría ser cálculos de impuestos, sumatorias, etc.

Agiliza la navegación ya que todo el contenido inicial (css, js, imágenes, etc) es cargado de golpe, lo demás será el contenido que se quiera visualizar.

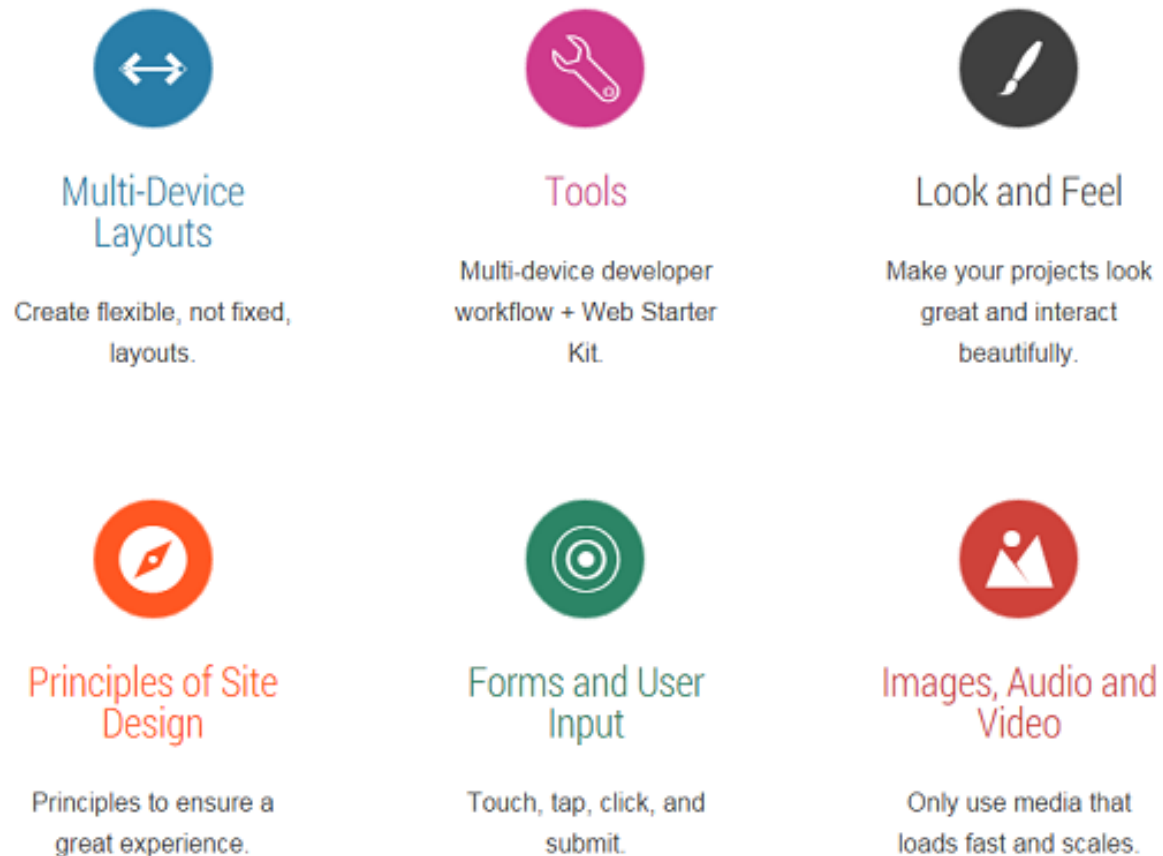
Ayuda abaratar los costos de servidor al disminuir el número de requests realizados.

- ¿Qué es Font Awesome?

Te trata de una tipografía con 369 caracteres, que en lugar de estar formada con el abecedario, está formada por íconos. Además, cada versión nueva agrega diferentes íconos a la colección.

- ¿Qué es Web Starter Kit (google), cuales son sus principales funciones?

Es un framework que ofrece un esqueleto sólido para desarrollar sitios web. Usa JavaScript, CSS y HTML, tiene una base responsive, sincronización multidispositivo y optimización de rendimiento con soporte para SASS y para la herramienta gulp.js



- ¿Qué significa diseño Pixel Perfect?

Cuando hablamos del pixel perfect en diseño web o app nos referimos, como comentaba anteriormente, a todas las praxis a tener en cuenta a la hora de diseñar para maximizar la nitidez y la fidelidad, para conseguir un producto que sea fácil percibir, entender y con el que sea sencillo interactuar.

## Bibliografía

<https://www.illusionstudio.es/que-es-un-framework-web>

<https://www.40defiebre.com/que-es/disenio-responsive>

<https://indexdesarrollo.com/frameworks-en-diseno-web/>

<https://www.tithink.com/es/2018/08/29/framework-o-librerias-ventajas-y-desventajas/>

<https://rockcontent.com/es/blog/bootstrap/>

<https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/less-css-tutorial/>

<https://yosoy.dev/ventajas-del-uso-de-less/>

<https://openwebinars.net/blog/que-es-sass-ventajas-desventajas-y-ejemplos-de-desarrollo/>

<https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/sass/>

<https://www.genbeta.com/desarrollo/960-grid-system-un-framework-para-css>

<https://blog.aulaformativa.com/10-framework-css-con-sistema-de-cuadriculas/>

<https://www.ictea.com/cs/index.php?rp=%2Fknowledgebase%2F2346%2FiQue-es-el-diseno-web-sensible-o-adaptativo.html>

<https://speckyboy.com/responsive-lightweight-css-frameworks/>

<https://medium.com/@programacionje/que-es-spa-single-page-application-4dbd3694fac9>

<https://anexsoft.com/que-son-las-single-page-application-spa-ventajas-y-desventajas>

<https://galernaestudio.com/font-awesome-mas-facil-diseno-web/>

<https://blog.ida.cl/desarrollo/funcionalidades-google-web-starter-kit/>

<https://www.linkedin.com/pulse/pixel-perfect-en-los-peque%C3%B1os-detalles-radica-la-cereceda-vicente/?originalSubdomain=es>