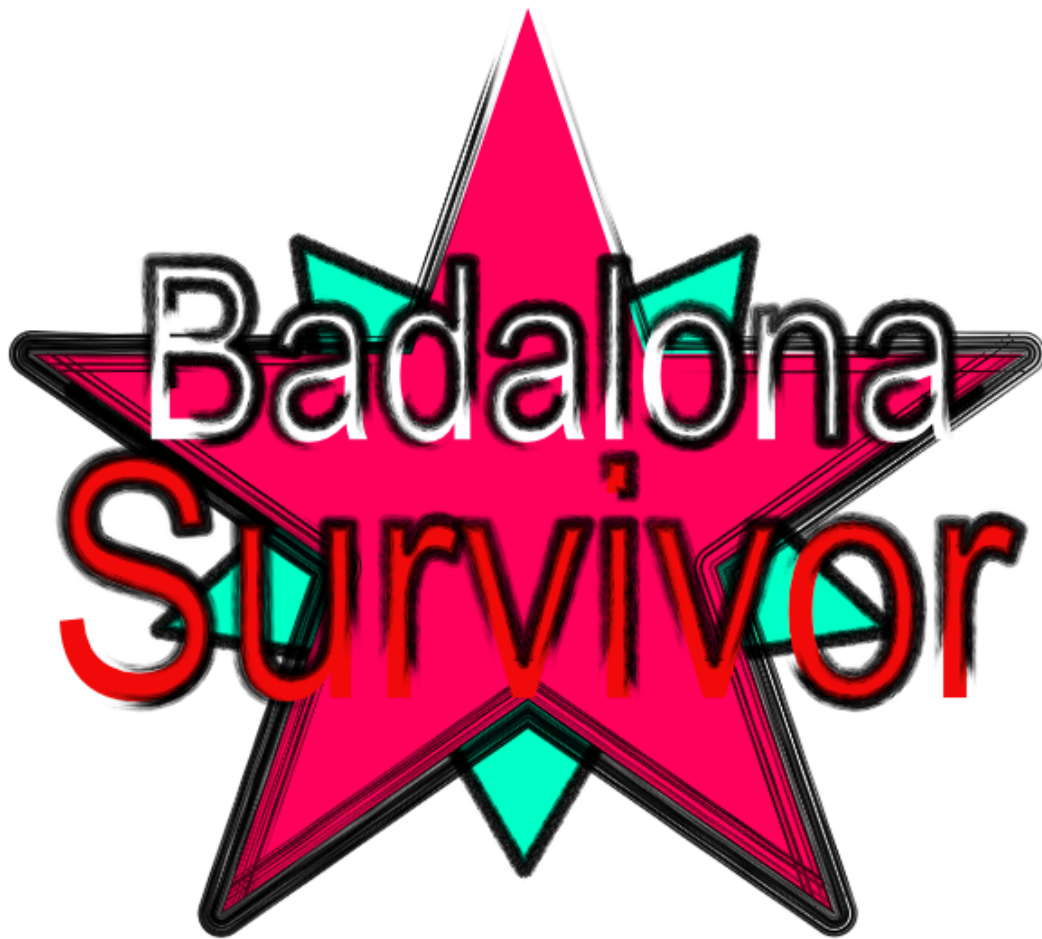


Arte 2D y Desarrollo Web Responsive

FASE 1. Proyecto Arte 2D



FASE 1. Proyecto Arte 2D	1
Nombre del videojuego	4
Badalona Survivor	4
Temática y gameplay	4
Historia	5
Desarrollo	5
Logo	7
Personajes	7
Llúcia	7
Tipo de licencia	8
Acuerdo de licencia	8
Enlace	8
Enemigo 1: Normal Knight	8
Tipo de licencia	9
Acuerdo de licencia	9
Enemigo 2: Josechiro Ashina	9
Tipo de licencia	10
Acuerdo de licencia	10
Enemigo 3: Bringer of Chaos, final boss	10
Tipo de licencia	11
Acuerdo de licencia	11
Escenario	11
Tipo de licencia	12
Acuerdo de licencia	12
Contenido en los paquetes (Package Content)	12
Licencias	13
Concept Art	15
Audio	16
Ambiente	16
Tipo de licencia	16
Acuerdo de licencia	16
Sound FX	16
Tipo de licencia	16
Acuerdo de licencia	16
Formatos de sonido	17
De dónde viene y qué es Materialize CSS	19
Material is the metaphor	19
Bold, graphic, intentional	19

Motion provides meaning	20
Instalación y forma de uso	20
Project Structure	21
HTML Setup	22
Templates	23
SASS Setup	24
Compiling SASS	24
Cómo usar Materialize	25
Color	25
Uso más adecuado/generalizado.	26
Casos de uso comunes	26
Aplicaciones Web Responsivas	26
Formularios Estilizados	27
Modales, Tarjetas y Navegación	27
Ventajas e inconvenientes	28
Ventajas	28
Desventajas	30
¿Cuándo usar Materialize CSS?	31
Extensivo ejemplos de uso y sintaxis de la herramienta.	32
Grid System (Sistema de Columnas Responsivas)	32
Botones	32
Formularios	33
Tarjetas (Cards)	33
Navbars (Barras de Navegación)	34
Modales (Ventanas Emergentes)	35
Listas Desplegables (Dropdowns)	36
Tabs (Pestañas)	36
Conclusiones	37

Nombre del videojuego

Badalona Survivor

Temática y gameplay

Badalona Survivor es un juego 2D en el que la protagonista trata de defender Badalona del ataque de enemigos malignos que buscan la destrucción de la ciudad.

El gameplay es de un juego de peleas uno contra uno, en el que el jugador tiene que derrotar al enemigo al que se enfrenta. Cuando derrotes al enemigo, avanza a otra pantalla con un enemigo mucho más fuerte, así hasta que derrotes a todos los enemigos que invaden Badalona.

Historia

Badalona se encuentra en peligro, unas sombras malignas están atacando la ciudad de la comunidad de Barcelona. Estas sombras surgen a partir del mal que origina las personas que habitan la ciudad.

Llúcia, una experta en este tema de las sombras malignas, está preparada para enfrentar los horrores que consumen Badalona.

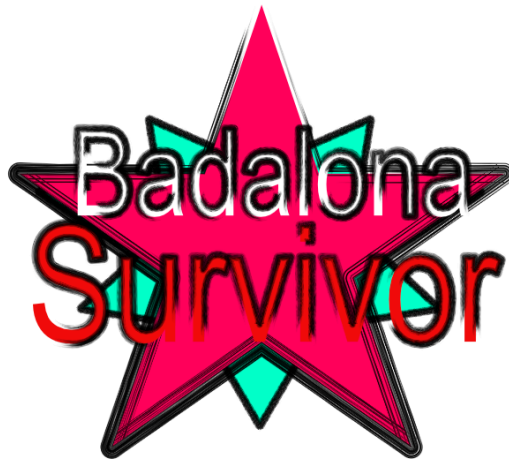
¿Podrá Llúcia librar a Badalona del MAL?

Desarrollo

Herramienta	Tarea	Justificación
Unity	Desarrollo del videojuego (escenarios, personajes, programación)	Unity es ideal para desarrollar videojuegos desde cero, ya que proporciona un entorno completo para crear escenarios, personajes y programar la jugabilidad. Ofrece herramientas específicas para el desarrollo de juegos y una Asset Store con recursos gratuitos.
Photoshop	Creación de texturas, sprites y elementos gráficos 2D	Photoshop es perfecto para diseñar y editar imágenes 2D detalladas que se utilizan en el juego, como texturas para personajes, objetos y fondos. Sus potentes herramientas de edición son esenciales para crear gráficos visualmente atractivos.
Audacity	Edición de efectos de sonido, música y diálogos	Audacity es una herramienta gratuita y fácil de usar para grabar y editar audio. Es ideal para modificar efectos de sonido, grabar diálogos y crear música ambiental, asegurando que los sonidos sean claros y bien integrados en el juego.
Premiere	Edición de videos promocionales, trailers y cinemáticas	Premiere es ideal para la creación de trailers, cinemáticas y otros videos promocionales. Sus potentes herramientas de edición de video permiten ensamblar clips, ajustar colores y agregar efectos visuales de manera profesional.

Para el desarrollo de Badalona Survivor, hemos elegido Unity como el motor principal del videojuego debido a su capacidad para manejar tanto la programación como la creación de escenarios 3D. Unity ofrece un entorno todo-en-uno para diseñar, programar e implementar todos los aspectos del juego, desde la interacción hasta la animación. Además, para tareas específicas como la creación de texturas y gráficos 2D, utilizamos Photoshop, ya que nos permite crear imágenes detalladas y adaptadas a las necesidades del juego. Para los efectos de sonido y la música, Audacity es nuestra herramienta preferida, debido a su facilidad de uso y su potente capacidad para editar y procesar audio. Finalmente, para crear trailers o cinemáticas que complementen la experiencia del juego, utilizamos Premiere, dada su robustez y versatilidad en la edición de videos.

Logo



El logo se ha creado en la herramienta Affinity. Tiene el nombre del videojuego y detrás dos estrellas.

Personajes

Llúcia

Personaje principal y protagonista.

El modelo de Llúcia está basado en un asset gratuito de la tienda de Unity AssetStore



Tipo de licencia

Extension Asset

Acuerdo de licencia

Standard Unity Asset Store EULA

[Enlace](#)

<https://assetstore.unity.com/packages/2d/characters/warrior-free-asset-195707>

Enemigo 1: Normal Knight

Primer enemigo con el que se encuentra Llúcia.

El modelo está basado en un asset gratuito de la AssetStore



Tipo de licencia

Extension Asset

Acuerdo de licencia

Standard Unity Asset Store EULA

[Enlace](#)

<https://assetstore.unity.com/packages/2d/characters/hero-knight-2-168019>

Enemigo 2: Josechiro Ashina

Segundo enemigo, aparece cuando derrotas a Normal Knight.

El modelo está basado en un asset gratis de la AssetStore.



Tipo de licencia

Extension Asset

Acuerdo de licencia

Standard Unity Asset Store EULA

[Enlace](#)

<https://assetstore.unity.com/packages/2d/characters/martial-hero-170422#content>

Enemigo 3: Bringer of Chaos, final boss

Enemigo final que aparece cuando derrotas a los enemigos anteriores.

El modelo está basado en un asset gratuito de la AssetStore.



Tipo de licencia

Extension Asset

Acuerdo de licencia

Standard Unity Asset Store EULA

[Enlace](#)

<https://assetstore.unity.com/packages/2d/characters/bringer-of-death-free-195719>

Escenario

El escenario representa a la ciudad de Badalona.

Está construido con assets gratuitos de la AssetStore.



Tipo de licencia

Extension Asset

Acuerdo de licencia

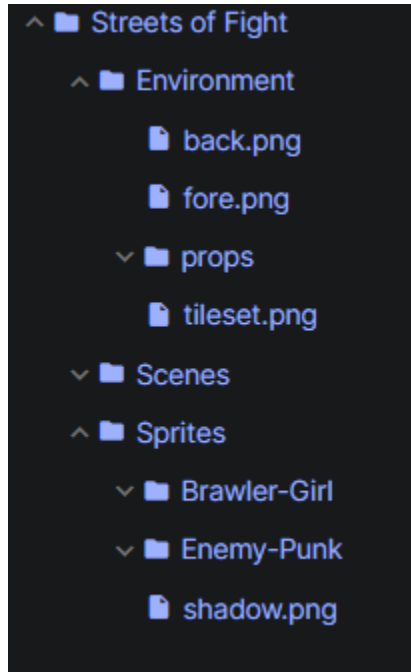
Standard Unity Asset Store EULA

[Enlace](#)

<https://assetstore.unity.com/packages/2d/characters/streets-of-fight-211512#content>

Contenido en los paquetes (Package Content)

Cuando usamos un asset, este viene con una serie de contenidos, en nuestro caso cada paquete trae una serie de imágenes vectoriales .png para poder construir los escenarios del videojuego, animaciones, personajes, etc.



Licencias

La "**Standard Unity Asset Store EULA**" (Acuerdo de Licencia de Usuario Final de la Tienda de Assets de Unity) es la licencia estándar que se aplica a la mayoría de los assets adquiridos en la Unity Asset Store, incluidos los gratuitos. Este acuerdo establece los términos y condiciones bajo los cuales puedes usar los assets en tus proyectos.

Puntos claves:

1. **Uso Personal y Comercial:** Puedes usar los assets en proyectos tanto personales como comerciales sin tener que pagar tarifas adicionales o royalties, siempre que no redistribuyas el asset como parte de un paquete o un producto que pueda ser adquirido de forma separada.
2. **Modificaciones:** Permite que modifiques los assets para que se ajusten a tus necesidades dentro de tu proyecto. Sin embargo, las modificaciones no pueden ser distribuidas o vendidas como un asset o recurso reutilizable.
3. **Prohibición de Redistribución:** No puedes redistribuir los assets (ni originales ni modificados) como parte de otros paquetes o venderlos de manera independiente. Esto incluye no vender, compartir o distribuir el asset tal cual o como parte de otro producto que se pueda volver a usar.
4. **Uso en Proyectos de Juegos:** Puedes utilizar los assets en proyectos de juegos, aplicaciones, simuladores y otros tipos de software, sin restricciones, siempre que el producto final sea tuyo y no distribuya los assets por separado.
5. **No Exclusividad:** Esta licencia es no exclusiva, lo que significa que otros usuarios también pueden usar los mismos assets.

Se van a cumplir todos los puntos de esta licencia, ya que los assets los vamos a usar para un proyecto de clase, no vamos a distribuir los assets ni individualmente ni como un paquete.

Se pueden modificar los assets dentro del proyecto, como cambiar el color o añadir, quitar partes del asset (se implementará en un futuro si hiciese falta).

Como los assets se van a usar para desarrollar un proyecto nuevo, el proyecto pasa a ser de una nueva propiedad intelectual, no los assets individuales.

Si quisiéramos en un futuro comercializar el videojuego creado a partir de los assets, se podría hacer perfectamente ya que la licencia lo permite.

Concept Art





Audio

Ambiente

Para el audio ambiente usamos un asset gratuito que trae un pack de sonidos

Tipo de licencia

Extension Asset

Acuerdo de licencia

Standard Unity Asset Store EULA

[Enlace](https://assetstore.unity.com/packages/audio/ambient/sci-fi/free-sci-fi-horror-music-276943#content)

<https://assetstore.unity.com/packages/audio/ambient/sci-fi/free-sci-fi-horror-music-276943#content>

Sound FX

Para los efectos de sonido usamos un asset que trae una gran variedad de sonidos aptos para Badalona Survivor

Tipo de licencia

Extension Asset

Acuerdo de licencia

Standard Unity Asset Store EULA

[Enlace](#)

<https://assetstore.unity.com/packages/audio/sound-fx/rpg-essentials-sound-effects-free-227708#content>

Formatos de sonido

Los sonidos utilizados en el videojuego usan el formato .wav

.wav (Waveform Audio File Format) es un tipo de archivo de audio que almacena sonidos en forma de ondas digitales. Es uno de los formatos más comunes para almacenar audio en PC y otros dispositivos debido a su calidad y simplicidad.

FASE 2: Materialize CSS



De dónde viene y qué es Materialize CSS

Creado y diseñado por Google, Material Design es un lenguaje de diseño que combina los principios clásicos del diseño de éxito junto con la innovación y la tecnología. El objetivo de Google es desarrollar un sistema de diseño que permita una experiencia de usuario unificada en todos sus productos y en cualquier plataforma.

El equipo de desarrollo está compuesto por cuatro estudiantes de la universidad de Carnegie Mellon.

Material is the metaphor

La metáfora del material define la relación entre espacio y movimiento. La idea es que la tecnología se inspira en el papel y la tinta y se utiliza para facilitar la creatividad y la innovación. Las superficies y los bordes proporcionan señales visuales familiares que permiten a los usuarios comprender rápidamente la tecnología más allá del mundo físico.

Bold, graphic, intentional

Elements and components such as grids, typography, color, and imagery are not only visually pleasing, but also create a sense of hierarchy, meaning, and focus. Emphasis on different actions and components create a visual guide for users.

Motion provides meaning

El movimiento permite al usuario establecer un paralelismo entre lo que ve en la pantalla y en la vida real. Al proporcionar tanto retroalimentación como familiaridad, permite al usuario sumergirse por completo en una tecnología desconocida. El movimiento contiene coherencia y continuidad, además de dar a los usuarios información subconsciente adicional sobre los objetos y las transformaciones.

Instalación y forma de uso

Materialize está disponible en dos versiones diferentes. Puede seleccionar la versión que desee en función de sus preferencias y conocimientos. Para empezar a utilizar Materialize, todo lo que tienes que hacer es descargar una de las opciones siguientes.

Materialize

This is the standard version that comes with both the minified and unminified CSS and JavaScript files. This option requires little to no setup. Use this if you are unfamiliar with Sass.

MATERIALIZE 

Sass

This version contains the source SCSS files. By choosing this version you have more control over which components to include. You will need a Sass compiler if you choose this option.

SOURCE 

También puede obtener la última versión utilizando NPM. Esta versión contiene los archivos fuente, así como los archivos CSS y JavaScript compilados.

```
npm install materialize-css@next
```

Project Structure

Después de la descarga, extraiga los archivos en el directorio donde se encuentra su sitio web. Tu directorio tendrá este aspecto.

Observará que hay dos conjuntos de archivos. El minificado significa que el archivo está «comprimido» para reducir los tiempos de carga. Estos archivos minificados suelen utilizarse en producción, mientras que es mejor utilizar los archivos sin minificar durante el desarrollo.

```
MyWebsite/  
|--css/  
|   |--materialize.css  
|  
|--fonts/  
|   |--roboto/  
|  
|--js/  
|   |--materialize.js  
|  
|--index.html
```

HTML Setup

A continuación sólo tienes que asegurarte de enlazar correctamente los archivos en tu página web. Generalmente es aconsejable importar los archivos javascript al final del cuerpo para reducir el tiempo de carga de la página. Siga el siguiente ejemplo para importar Materialize en su página web.

```
<!DOCTYPE html>

<html>

  <head>

    <!--Import Google Icon Font-->

    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

    <!--Import materialize.css-->

    <link type="text/css" rel="stylesheet" href="css/materialize.min.css"
media="screen,projection"/>


    <!--Let browser know website is optimized for mobile-->

    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>

  </head>


  <body>


    <!--JavaScript at end of body for optimized loading-->

    <script type="text/javascript" src="js/materialize.min.js"></script>

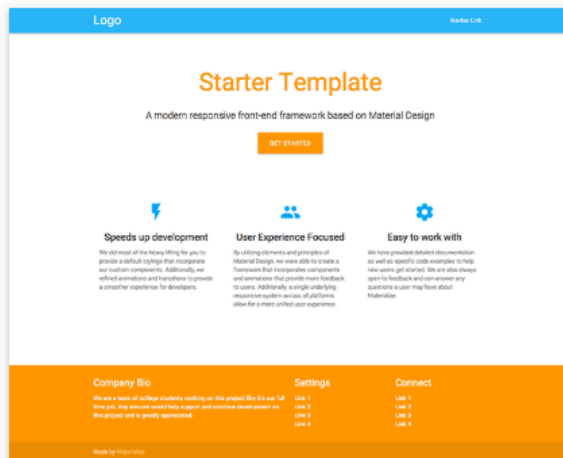
  </body>

</html>
```

Templates

Materialize tiene algunas plantillas de inicio para que puedas empezar a diseñar fácilmente tu sitio web con un tiempo de configuración mínimo. Echa un vistazo a la colección y descarga la que mejor se adapte a sus necesidades.

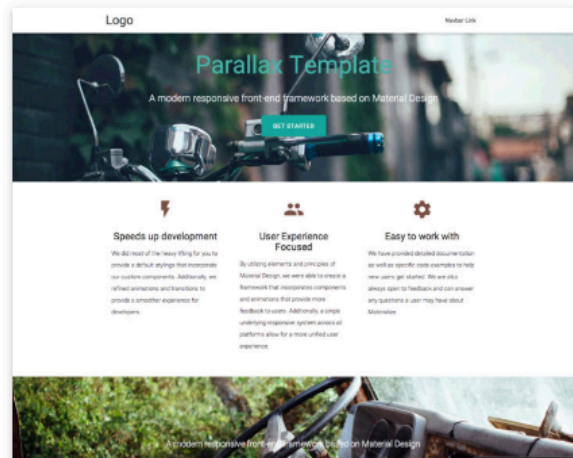
Starter Template



This is the simplest starter page with a Header, Call-to-Action, and Icon Features.



Parallax Template



This is the simplest starter page with a Header, Call-to-Action, and Icon Features.



SASS Setup

Esta opción sólo es relevante si has elegido descargar la versión SASS de materialize.

Compiling SASS

En lugar de tener una carpeta css, encontrarás que la descarga contiene muchos archivos .scss que contienen los estilos de los componentes individuales. Desafortunadamente, el navegador no puede interpretar Sass, por lo que debe hacer que su compilador Sass compile el scss/materialize.scss en un archivo CSS normal. En este punto puede enlazar este archivo recién generado en su página HTML.

```
MyWebsite/  
|--css/  
| |--materialize.css <-- compiled from scss/materialize.scss  
|  
|--js/  
| |--materialize.js  
|  
|--scss/  
| |--materialize.scss  
| |--components/  
|  
|--index.html
```

Cómo usar Materialize

Principalmente la forma de uso es añadir las clases que proporciona Material CSS a los elementos html. Aquí hay un ejemplo con el color.

Color

Materialize tiene una paleta de colores basada en los colores base del diseño de materiales. Cada uno de estos colores se define con una clase de color base y una clase opcional de aclarado u oscurecido.

Por ejemplo, si quieres cambiar el background-color, basta con añadir el nombre del color y claro/oscuro como clase al elemento.

```
<div class="card-panel teal lighten-2">This is a card panel with a  
teal lighten-2 class</div>
```

Para el text-color, solo tienes que aplicar -text a la clase.

```
<div class="card-panel">  
  <span class="blue-text text-darken-2">This is a card panel with  
dark blue text</span>  
</div>
```

Para la versión SASS puedes aplicar los colores extendiendo las clases de esta forma

```
.ilike-blue-container {  
  @extend .blue, .lighten-4;  
}
```


Para darle estilo a un botón, también se ponen las clases que Materialize nos facilita.

```
<a class="waves-effect waves-light btn">button</a>
```

Para más información puedes consultar en la documentación [oficial](#).

Uso más adecuado/generalizado.

Como ya sabemos Materialize CSS es un framework basado en Material Design de Google. Su uso más adecuado y generalizado es en proyectos donde se busca un diseño moderno, responsivo y con una estructura basada en componentes predefinidos.

Casos de uso comunes

Aplicaciones Web Responsivas

Materialize es excelente para páginas web adaptables sin necesidad de escribir mucho CSS.

```
<div class="container">
  <div class="row">
    <div class="col s12 m6 l4">
      <h5>Sección 1</h5>
      <p>Contenido aquí.</p>
    </div>
    <div class="col s12 m6 l4">
      <h5>Sección 2</h5>
      <p>Otro contenido aquí.</p>
    </div>
  </div>
</div>
```

s12: Ocupa 12 columnas en pantallas pequeñas.

m6: Ocupa 6 columnas en medianas.

l4: Ocupa 4 columnas en grandes.

Formularios Estilizados

Materialize ofrece formularios con efectos dinámicos sin necesidad de mucho CSS/JS.

```
<div class="row">
  <form class="col s12">
    <div class="input-field">
      <input id="email" type="email" class="validate">
      <label for="email">Correo Electrónico</label>
      <span class="helper-text" data-error="Email inválido" data-success="Válido"></span>
    </div>
    <button class="btn waves-effect waves-light" type="submit">Enviar</button>
  </form>
</div>
```

validate: Aplica validaciones automáticas.

helper-text: Muestra mensajes de error o éxito.

waves-effect: Efecto de onda al hacer clic en botones.

Modales, Tarjetas y Navegación

Si necesitas tarjetas, modales o una barra de navegación, Materialize te lo facilita.

```
<!-- Botón para abrir modal -->
<a class="waves-effect waves-light btn modal-trigger" href="#modal1">Abrir Modal</a>

<!-- Modal -->
<div id="modal1" class="modal">
  <div class="modal-content">
    <h4>Título del Modal</h4>
    <p>Este es un modal simple.</p>
  </div>
  <div class="modal-footer">
    <a href="#" class="modal-close waves-effect waves-green btn-flat">Cerrar</a>
  </div>
</div>

<script>
  document.addEventListener('DOMContentLoaded', function() {
    var elems = document.querySelectorAll('.modal');
    M.Modal.init(elems);
  });
</script>
```

Se activa con modal-trigger.

Se inicializa en JavaScript con M.Modal.init().

Ventajas e inconvenientes

Ventajas

1. Diseño Moderno y Atractivo

Basado en Material Design, con efectos y animaciones fluidas. Incluye elementos como botones flotantes, sombras, transiciones y tarjetas interactivas.

```
<div class="card">
  <div class="card-content">
    <span class="card-title">Título</span>
    <p>Este es un ejemplo de tarjeta con Materialize.</p>
  </div>
</div>
```

2. Responsividad Integrada

Usa un sistema de grid flexible basado en 12 columnas. Adapta automáticamente los elementos a móviles, tablets y escritorios.

Ventaja sobre Bootstrap: Es más liviano y visualmente más atractivo por defecto.

```
<div class="row">
  <div class="col s12 m6 l4">Columna 1</div>
  <div class="col s12 m6 l4">Columna 2</div>
</div>
```

3. Componentes Predefinidos

Incluye formularios, modales, botones, listas desplegables, tooltips y más. Se pueden usar sin escribir mucho código CSS o JavaScript.

Diferencia con Bootstrap: No requiere jQuery, ya que se inicializa con JavaScript puro.

```
<a class="waves-effect waves-light btn modal-trigger" href="#modal1">Abrir Modal</a>

<div id="modal1" class="modal">
  <div class="modal-content">
    <h4>Modal</h4>
    <p>Este es un modal simple.</p>
  </div>
</div>

<script>
  document.addEventListener('DOMContentLoaded', function() {
    M.Modal.init(document.querySelectorAll('.modal'));
  });
</script>
```

4. Fácil de Implementar

Se puede integrar rápidamente con CDN o instalando con NPM. No necesitas escribir mucho CSS o personalizar estilos.

<!-- Agregar en el <head> -->

```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
```

```
<script
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>
```

5. Animaciones y Efectos Nativos

Tiene efectos como "waves" (ondas al hacer clic), transiciones y sombras suaves. No necesitas bibliotecas adicionales para agregar interactividad.

```
<button class="btn waves-effect waves-light">Haz clic</button>
```

Desventajas

1. Personalización Limitada

Aunque tiene un diseño atractivo, es difícil personalizarlo sin sobrescribir muchas clases. Si quieres un estilo diferente a Material Design, es mejor usar Tailwind CSS o Bootstrap.

2. Comunidad y Actualizaciones Lentas

Última versión estable: 2018, lo que indica un desarrollo casi abandonado.

Menos comunidad y documentación en comparación con Bootstrap o Tailwind.

Alternativa recomendada: Si buscas actualizaciones constantes, usa Bootstrap 5 o Tailwind CSS.

3. Alto Uso de Clases en el HTML

Para hacer pequeños cambios en el diseño, a veces necesitas agregar muchas clases.

```
<div class="input-field">
  <input id="email" type="email" class="validate">
  <label for="email">Correo Electrónico</label>
  <span class="helper-text" data-error="Email inválido" data-success="Válido"></span>
</div>
```

Con Tailwind CSS o Bootstrap, esto podría hacerse con menos clases.

4. No es Tan Flexible Como Otros Frameworks

Está diseñado exclusivamente para Material Design, lo que limita su flexibilidad en proyectos con estilos diferentes.

Otras opciones como Tailwind CSS permite más personalización desde el inicio.

¿Cuándo usar Materialize CSS?

Úsalo si:

Quieres un diseño moderno listo para usar.

Necesitas una web rápida y responsiva sin mucho esfuerzo.

Prefieres un framework ligero y fácil de aprender.

Evítalo si:

Quieres personalizar mucho el diseño.

Necesitas un framework con actualizaciones frecuentes.

Prefieres un ecosistema con más comunidad y soporte (como Bootstrap o Tailwind).

Extensivo ejemplos de uso y sintaxis de la herramienta.

Grid System (Sistema de Columnas Responsivas)

Materialize usa un grid de 12 columnas basado en Flexbox.

```
<div class="row">
  <div class="col s12 m6 l4">Columna 1</div>
  <div class="col s12 m6 l4">Columna 2</div>
  <div class="col s12 m12 l4">Columna 3</div>
</div>
```

s12: Ocupa toda la pantalla en dispositivos pequeños.

m6: Ocupa 6 columnas en tablets.

l4: Ocupa 4 columnas en pantallas grandes.

Botones

Los botones tienen efectos de "waves" por defecto.

```
<button class="btn waves-effect waves-light">Botón Normal</button>
<a class="btn red">Botón Rojo</a>
<button class="btn-large blue">Botón Grande</button>
```

btn → Botón estándar.

btn-large → Botón grande.

waves-effect → Efecto de onda al hacer clic.

Formularios

Materialize ofrece formularios estilizados con animaciones en los inputs.

```
<div class="row">
  <form class="col s12">
    <div class="input-field">
      <input id="email" type="email" class="validate">
      <label for="email">Correo Electrónico</label>
      <span class="helper-text" data-error="Email inválido" data-success="Válido"></span>
    </div>
    <button class="btn waves-effect waves-light" type="submit">Enviar</button>
  </form>
</div>
```

input-field: Contenedor de inputs con animaciones.

validate: Agrega validaciones automáticas.

helper-text: Mensajes de error o éxito.

Tarjetas (Cards)

Las tarjetas son ideales para mostrar contenido estructurado.

```
<div class="card">
  <div class="card-content">
    <span class="card-title">Título de la Tarjeta</span>
    <p>Este es un ejemplo de tarjeta con Materialize CSS.</p>
  </div>
  <div class="card-action">
    <a href="#">Ver más</a>
  </div>
</div>
```


card: Contenedor principal.

card-content: Contenido dentro de la tarjeta.

card-action: Área para botones/enlaces.

Navbars (Barras de Navegación)

```
<nav>
  <div class="nav-wrapper blue">
    <a href="#" class="brand-logo">Mi Sitio</a>
    <ul id="nav-mobile" class="right hide-on-med-and-down">
      <li><a href="#">Inicio</a></li>
      <li><a href="#">Servicios</a></li>
      <li><a href="#">Contacto</a></li>
    </ul>
  </div>
</nav>
```

nav-wrapper: Contenedor de la barra de navegación.

right: Alinea los enlaces a la derecha.

hide-on-med-and-down: Oculta el menú en pantallas pequeñas.

Modales (Ventanas Emergentes)

```
<!-- Botón para abrir modal -->
<a class="waves-effect waves-light btn modal-trigger" href="#modal1">Abrir Modal</a>

<!-- Modal -->
<div id="modal1" class="modal">
  <div class="modal-content">
    <h4>Título</h4>
    <p>Este es un modal simple.</p>
  </div>
  <div class="modal-footer">
    <a href="#" class="modal-close waves-effect waves-green btn-flat">Cerrar</a>
  </div>
</div>

<script>
  document.addEventListener('DOMContentLoaded', function() {
    var elems = document.querySelectorAll('.modal');
    M.Modal.init(elems);
  });
</script>
```

```
M.Modal.init(document.querySelectorAll('.modal'));
```

Listas Desplegables (Dropdowns)

```
<!-- Botón para desplegar -->
<a class='dropdown-trigger btn' href='#' data-target='dropdown1'>Menú</a>

<!-- Contenido del dropdown -->
<ul id='dropdown1' class='dropdown-content'>
  <li><a href="#">Opción 1</a></li>
  <li><a href="#">Opción 2</a></li>
</ul>

<script>
  document.addEventListener('DOMContentLoaded', function() {
    var elems = document.querySelectorAll('.dropdown-trigger');
    M.Dropdown.init(elems);
  });
</script>
```

Tabs (Pestañas)

```
<div class="row">
  <div class="col s12">
    <ul class="tabs">
      <li class="tab col s3"><a href="#test1">Tab 1</a></li>
      <li class="tab col s3"><a href="#test2">Tab 2</a></li>
    </ul>
  </div>
  <div id="test1" class="col s12">Contenido 1</div>
  <div id="test2" class="col s12">Contenido 2</div>
</div>
```

Conclusiones

Materialize CSS es un framework de front-end basado en el lenguaje de diseño Material Design de Google. Su principal objetivo es proporcionar una estructura visual moderna, intuitiva y responsiva para el desarrollo de interfaces web. A lo largo de su uso, se pueden destacar las siguientes conclusiones:

Facilidad de Uso y Rápida Implementación: Al contar con una sintaxis sencilla y un conjunto de clases predefinidas, permite a los desarrolladores crear diseños atractivos sin necesidad de escribir grandes cantidades de CSS personalizado.

Diseño Responsivo y Moderno: Materialize CSS proporciona un sistema de rejilla basado en Flexbox y diversos componentes adaptables que garantizan una experiencia de usuario coherente en distintos dispositivos.

Compatibilidad con Componentes Dinámicos: El framework incluye una serie de elementos interactivos como modales, carruseles y listas desplegables, que pueden ser fácilmente inicializados con JavaScript.

Integración con JavaScript y JQuery: Si bien permite personalizar y utilizar sus componentes con JavaScript puro, Materialize también ofrece compatibilidad con jQuery para una implementación más sencilla.

Limitaciones en la Personalización: Aunque Materialize facilita el desarrollo de interfaces, su fuerte adhesión a Material Design puede representar una limitación en proyectos que requieran una mayor personalización visual.

Comunidad y Soporte: Aunque no es tan popular como otros frameworks como Bootstrap, Materialize cuenta con una comunidad activa y una documentación clara que facilita su aprendizaje y resolución de dudas.

En definitiva, **Materialize CSS** es una excelente opción para desarrolladores que buscan rapidez en la implementación de diseños modernos y responsivos, especialmente en proyectos que siguen las directrices de Material Design. Sin embargo, para aplicaciones que requieran una alta personalización, puede ser necesario modificar o complementar sus estilos predeterminados.