

GESTIÓN DE DESARROLLO DE SOFTWARE

Clase 1

Prof. Natalia Lucero

¿QUÉ ES EL DESARROLLO DE SOFTWARE?

El desarrollo de software es una serie de actividades en la ciencia de la computación dedicadas a la creación, diseño y lanzamiento de software. Este proceso es crucial, no solo en términos de desarrollo técnico, sino también para mejorar la usabilidad y la experiencia del usuario. Si eres diseñador de UI o UX, juegas un papel vital en esta cadena, contribuyendo a la creación de una interfaz de usuario y una experiencia integral.



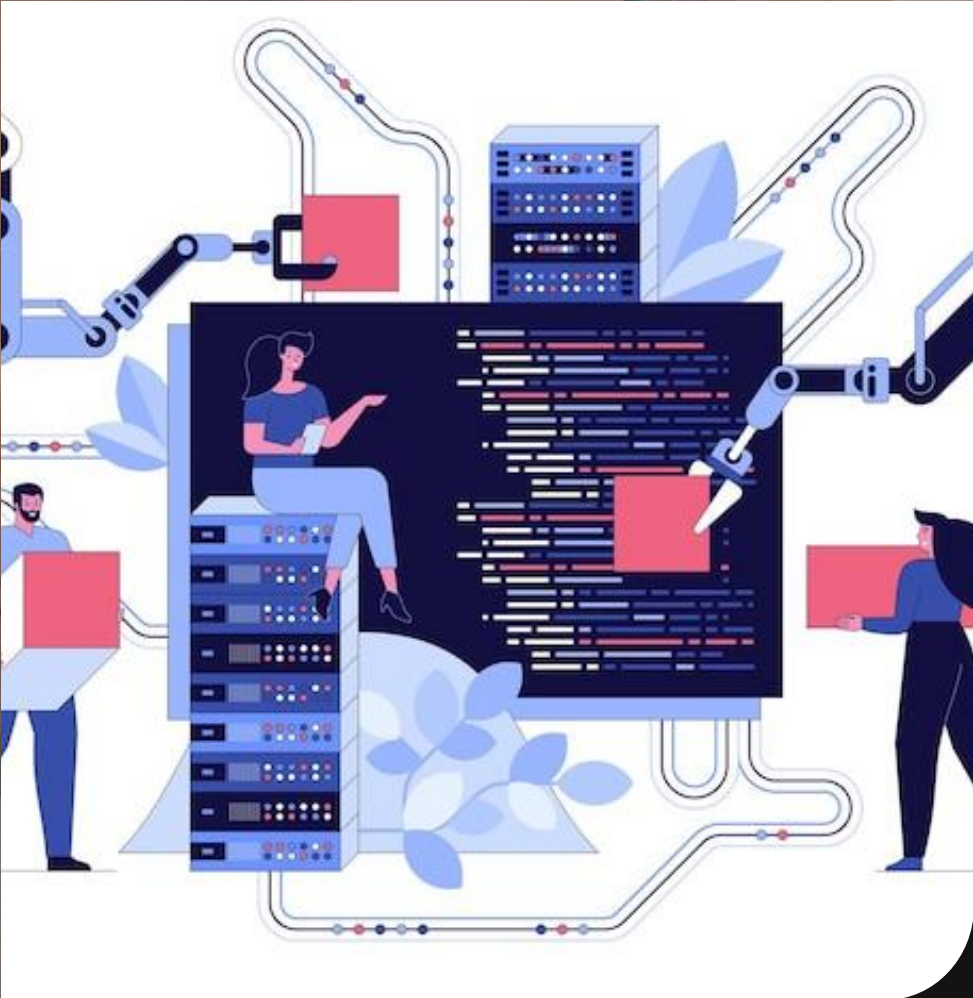
DENTRO DEL DESARROLLO, EXISTEN
DISTINTAS ÁREAS, CADA UNA CON UN
ENFOQUE DIFERENTE, AL IGUAL QUE
SUCEDER EN EL CAMPO DEL DISEÑO.
ALGUNOS EJEMPLOS DE ESTAS ÁREAS
INCLUYEN:

- **Desarrollo web**
- **Seguridad**
- **Aplicaciones móviles**
- **Software de escritorio**
- **Videojuegos**





ASÍ COMO EN EL DISEÑO PUEDES TRABAJAR CON MODA, PRODUCTOS O GRÁFICOS, EN EL CAMPO DEL DESARROLLO PUEDES ESPECIALIZARTE EN HERRAMIENTAS ESPECÍFICAS O PLATAFORMAS SEGÚN TU PREFERENCIA Y HABILIDADES.



¿QUÉ LENGUAJES DE PROGRAMACIÓN SE UTILIZAN?

Los lenguajes de programación son fundamentales para que los humanos podamos dar instrucciones a las computadoras. Se comunican a través de una sintaxis específica que cada máquina puede interpretar para ejecutar diferentes funciones. Algunos de los lenguajes más comunes incluyen:

- **JavaScript**
- **Python**



¿CÓMO FUNCIONAN ESTOS LENGUAJES EN LA PRÁCTICA?

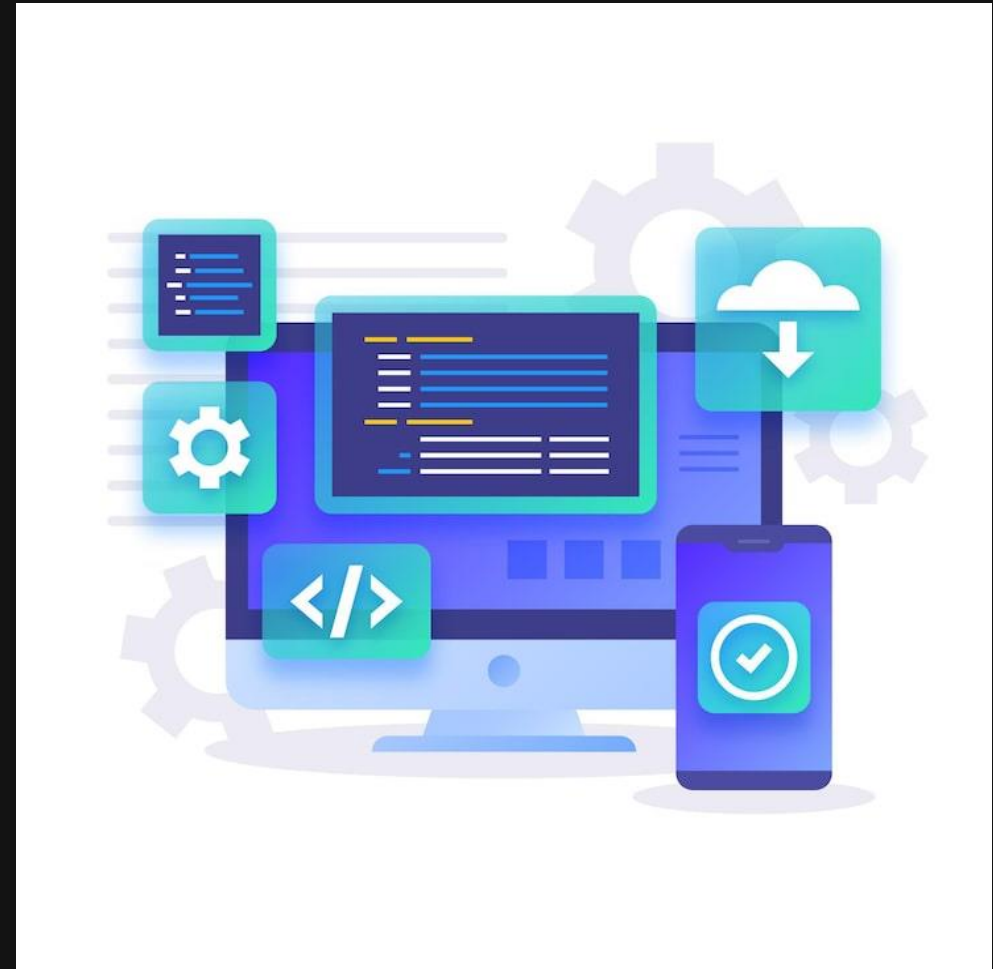
Para entender mejor la aplicación de estos lenguajes, utilizamos un ejemplo básico: verificar si un número es mayor a otro. Este es un ejemplo sencillo pero efectivo para visualizar cómo los lenguajes operan a nivel básico.

Para ilustrar:

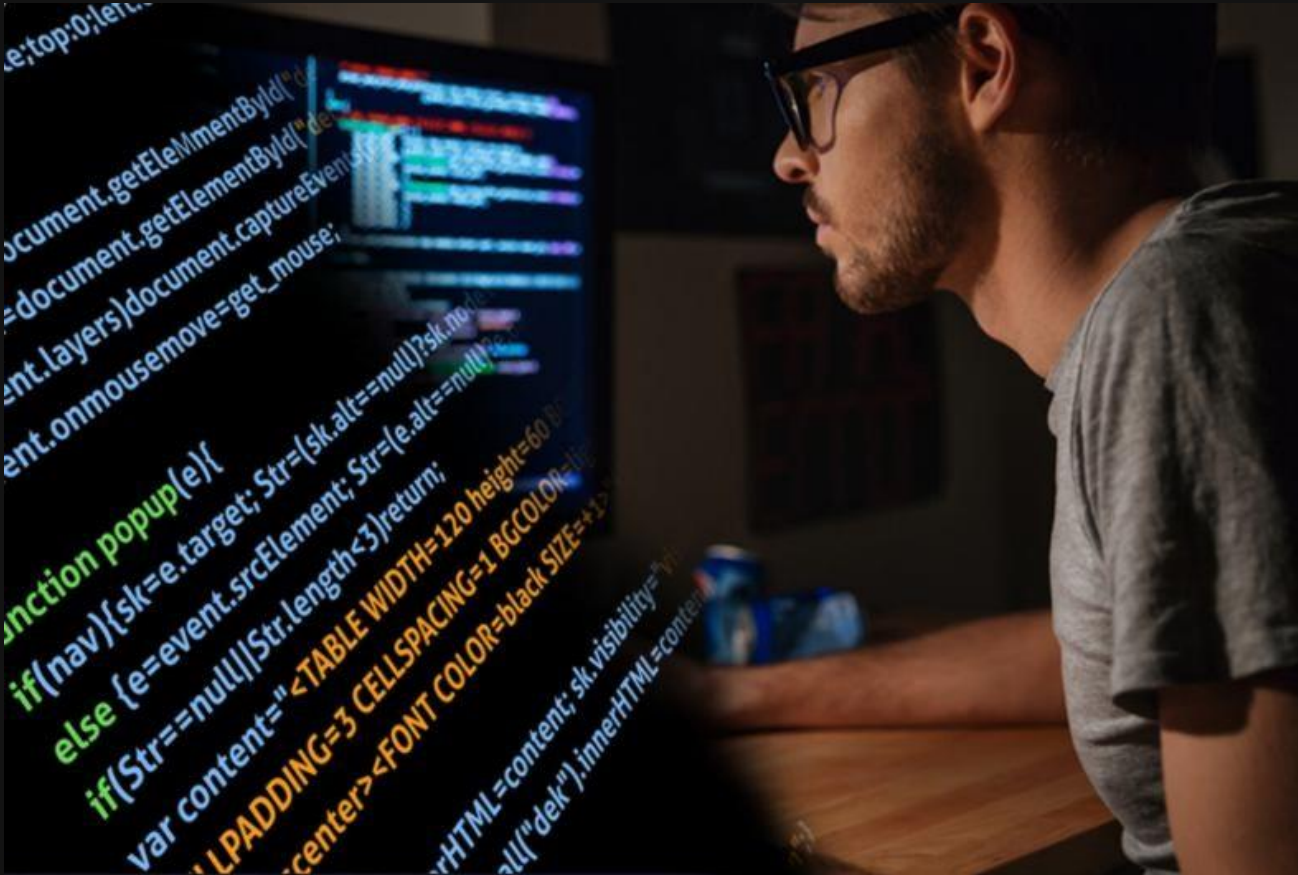
- **Variables:** Se utilizan para almacenar valores.
- **Condiciones:** Permiten ejecutar acciones específicas si se cumplen ciertos criterios, como mostrar un mensaje cuando una variable es mayor que otra.

¿QUÉ ES UN PROCESO DE DESARROLLO DE SOFTWARE?

Un proceso de desarrollo de software es un conjunto de actividades y métodos que conducen al diseño, desarrollo y entrega de un software. Compuesto por diferentes etapas, cada una de ellas tiene como objetivo garantizar que el producto final cumpla con los requisitos específicos y los estándares de calidad del software . Los pasos de un proceso de desarrollo de software incluyen etapas de diseño, desarrollo, prueba, implementación y mantenimiento del software.



¿POR QUÉ ES IMPORTANTE EL PROCESO DE DESARROLLO DE SOFTWARE?



El proceso de desarrollo de un producto de software requiere un enfoque sistemático para crear un producto final de alta calidad que satisfaga tanto las necesidades del usuario como la estrategia y los objetivos del negocio. El proceso en sí mismo es beneficioso por varias razones clave:

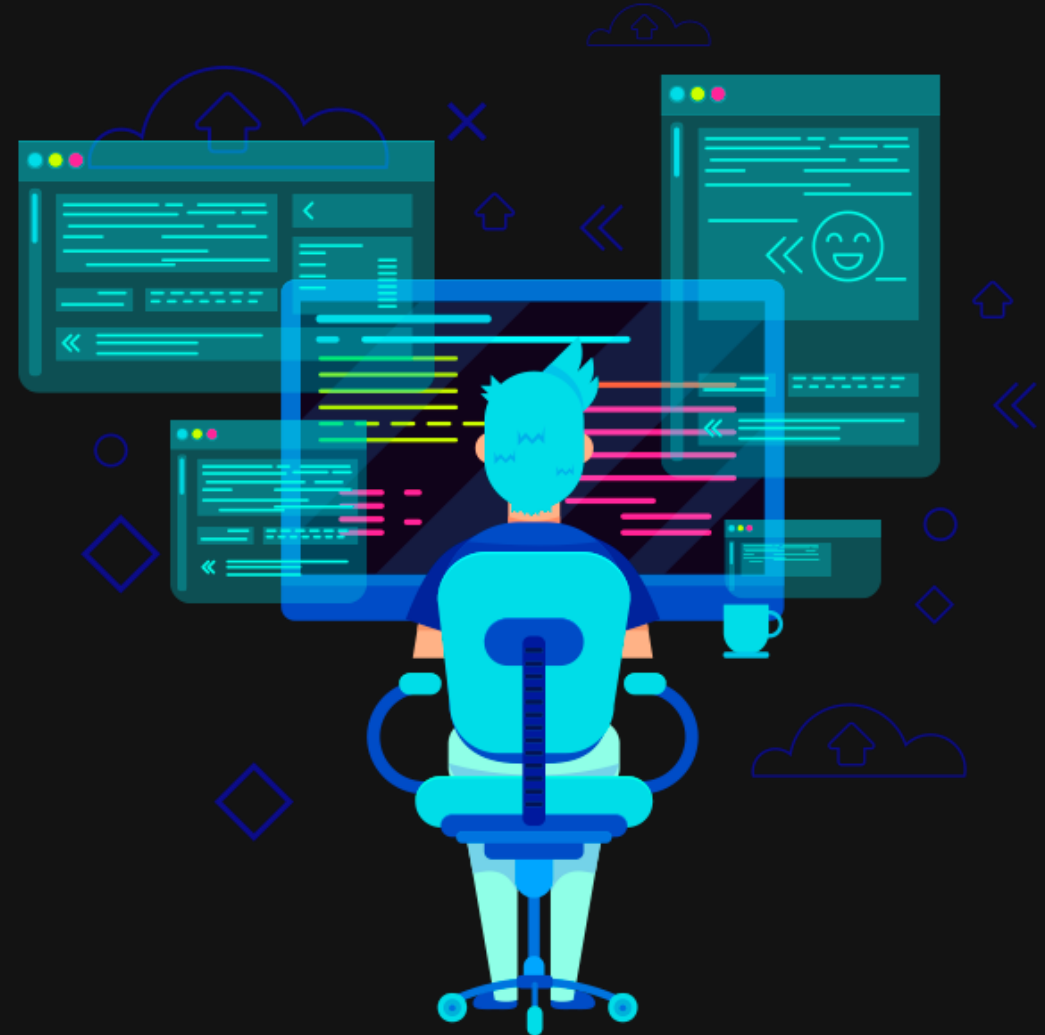
- Reduce la probabilidad de errores, retrasos y costos excesivos durante el proceso de desarrollo.
- Permite una mejor colaboración y comunicación sobre un producto.
- Garantiza que todos los aspectos del desarrollo estén cuidadosamente planificados y ejecutados.
- Ayuda a mantener la consistencia durante todo el desarrollo para garantizar un producto final estable.
- Permite a los equipos cumplir con los estándares de cumplimiento normativo.

Si bien el proceso en sí es esencial, la ruta que pueden tomar los equipos puede variar al optar por metodologías de gestión de proyectos específicas, como Agile, Scrum, Kanban u otras.

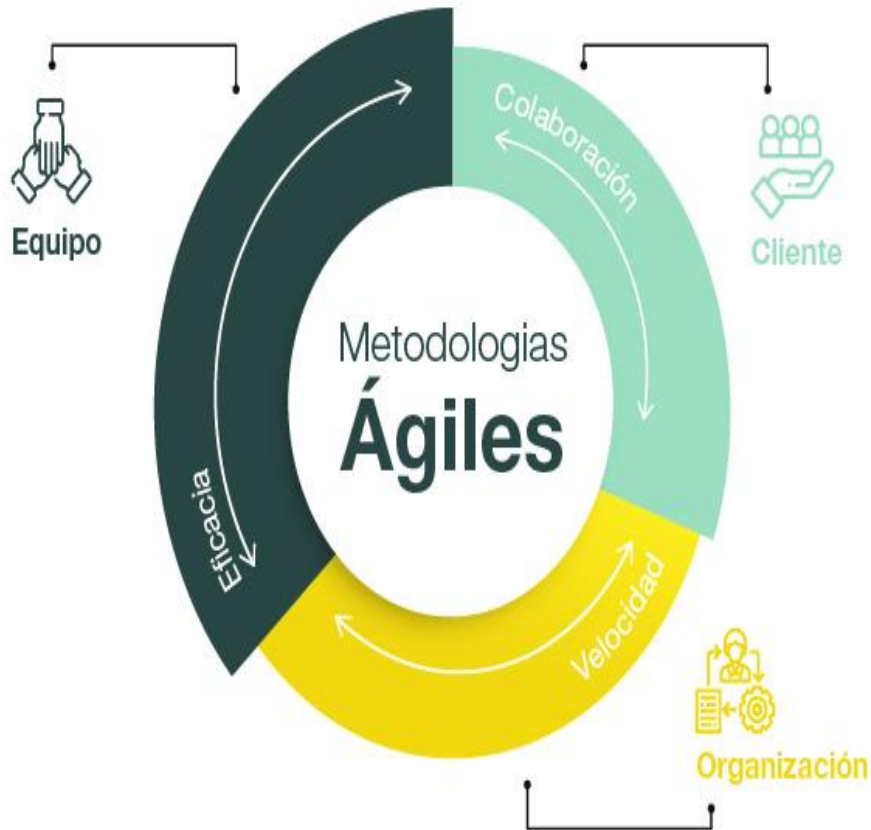
LOS TRES TIPOS DE SOFTWARE INFORMÁTICO

hay tres tipos de software de computadora que debe conocer:

- **Software de sistema** : software base que se encuentra entre el hardware y el software de aplicación. Es responsable de supervisar, controlar y operar la RAM, el procesador y los dispositivos de salida de una computadora. Algunos ejemplos incluyen sistemas operativos como Windows o iOS.
- **Software de utilidad** : software que se ejecuta en segundo plano y se encarga de realizar tareas específicas que ayudan a mantener el equipo funcionando correctamente. Algunos ejemplos incluyen antivirus, programas de limpieza de disco y programas de compresión de archivos.
- **Software de aplicación** : aplicaciones o programas codificados que realizan tareas específicas según las necesidades del usuario. Algunos ejemplos son procesadores de texto, hojas de cálculo, navegadores y software gráfico.



PROCESO DE DESARROLLO DE PRODUCTOS DE SOFTWARE ÁGIL



El desarrollo de software ágil es un enfoque moderno que prioriza la adaptación y la innovación. En un entorno ágil, el desarrollo de productos se realiza mediante un proceso iterativo e incremental donde se incorporan las opiniones de los clientes en cada iteración para brindar una experiencia óptima.

Es importante saber que Agile, en sí mismo, no implica pasos prácticos que una empresa pueda seguir para desarrollar software funcional. Más bien, Agile es una mentalidad que una organización adopta basándose en el manifiesto Agile, generalmente utilizando un marco de trabajo alineado con dichos principios.

El método de desarrollo de productos ágil más popular y eficaz es Scrum. Scrum descompone tareas complejas en sprints incrementales y se basa en la retroalimentación constante para perfeccionar el producto hasta lograr el resultado deseado.



¿QUÉ ES UN REQUERIMIENTO?

Un requerimiento es una descripción clara y detallada de una necesidad, característica o funcionalidad que un sistema debe cumplir para satisfacer las expectativas de los usuarios y las metas del proyecto. En el contexto del desarrollo de software, los requerimientos sirven como guía para diseñar, implementar y validar soluciones tecnológicas.

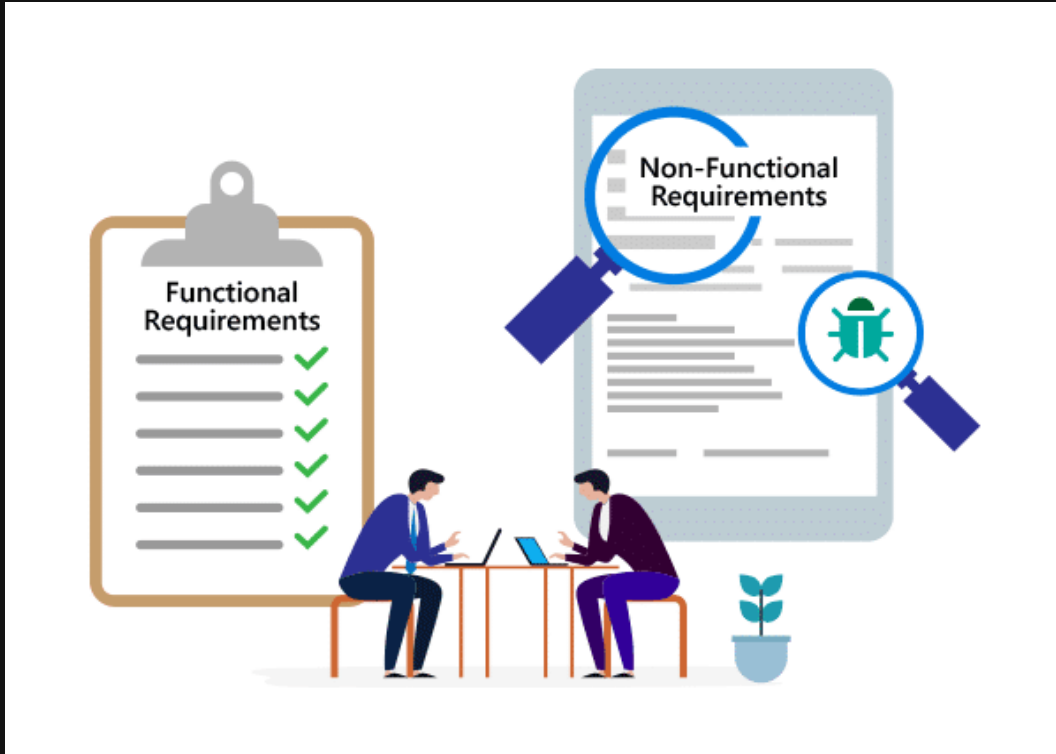
Los requerimientos son esenciales porque establecen un entendimiento común entre los stakeholders y el equipo de desarrollo, reduciendo riesgos y garantizando que el producto final cumpla con las expectativas definidas.



¿QUÉ ES LA INGENIERÍA DE REQUERIMIENTOS?

La ingeniería de requerimientos es el proceso estratégico de identificar, analizar, documentar y gestionar las necesidades y expectativas de los stakeholders en un proyecto de desarrollo de software. Se trata de traducir las ideas y objetivos de negocio en especificaciones claras y funcionales que guíen a los desarrolladores en la creación de soluciones tecnológicas efectivas.

TIPOS DE REQUERIMIENTOS



En la ingeniería de requerimientos, es fundamental clasificar las necesidades de un proyecto para garantizar claridad y enfoque durante el desarrollo. Los principales tipos de requerimientos son:

- **Requerimientos funcionales:** Describen **qué debe hacer el sistema**. Incluyen funciones, procesos y características específicas que deben implementarse para cumplir con los objetivos del proyecto. Ejemplo: "El sistema debe permitir a los usuarios registrarse mediante correo electrónico y contraseña."
- **Requerimientos no funcionales:** Detallan **cómo debe operar el sistema**. Se enfocan en atributos de calidad como rendimiento, seguridad, usabilidad y escalabilidad. Ejemplo: "El sistema debe procesar 100 solicitudes por segundo con un tiempo de respuesta menor a 2 segundos."
- **Requerimientos del dominio:** Especifican **características propias del contexto o sector del proyecto**. Incluyen estándares, normativas y restricciones específicas del entorno. Ejemplo: "El software debe cumplir con las normativas GDPR para la gestión de datos en Europa."

Identificar y documentar correctamente estos tipos de requerimientos asegura que el producto final sea funcional, eficiente y alineado con las expectativas del cliente.

PRINCIPIOS DE LA INGENIERÍA DE REQUERIMIENTOS



La ingeniería de requisitos se fundamenta en una serie de principios clave que aseguran un **proceso estructurado y efectivo** para definir y gestionar las necesidades de un proyecto. Estos principios son esenciales para minimizar riesgos y garantizar el éxito de las soluciones tecnológicas. A continuación, se explican los más relevantes:

Entender el contexto del negocio: Identificar objetivos, necesidades y restricciones para alinear los requisitos con el valor que el sistema debe aportar.

Involucrar a los stakeholders: Integrar perspectivas de todas las partes interesadas para definir requisitos claros y evitar malentendidos.

Definir requisitos claros y medibles: Especificar requisitos concretos y cuantificables para facilitar la validación.

Priorización basada en valor: Enfocar recursos en los requisitos más relevantes y estratégicos.

Gestión del cambio: Controlar modificaciones para adaptarse a cambios en el negocio sin afectar el proyecto.

Trazabilidad de los requisitos: Relacionar cada requisito con su fuente para garantizar seguimiento y cumplimiento.

Validación constante: Revisar y validar requisitos regularmente para evitar entregas que no cumplan expectativas.

Documentación adecuada: Mantener documentación clara y accesible para reducir errores y mejorar la comunicación.

Al aplicar estos principios, las empresas pueden garantizar un proceso de desarrollo más eficiente y orientado al éxito, logrando que el software entregado sea una solución funcional, alineada con las metas estratégicas del negocio y capaz de superar las expectativas de los usuarios



¿POR QUÉ ES IMPORTANTE LA INGENIERÍA DE REQUERIMIENTOS?



La ingeniería de requerimientos es esencial porque establece las bases para el éxito de cualquier proyecto de desarrollo de software o sistema. Sin un proceso claro y estructurado para identificar, analizar y gestionar los requisitos, los proyectos corren un alto riesgo de fallar en términos de tiempo, presupuesto o satisfacción del cliente. A continuación, algunas razones clave que resaltan su importancia:

1. Evita malentendidos y errores costosos
2. Alinea el proyecto con los objetivos del negocio
3. Optimiza el uso de recursos
4. Facilita la comunicación entre las partes
5. Soporta la gestión del cambio
6. Mejora la calidad del producto final

DISEÑO RESPONSIVO



CREANDO UNA SECCIÓN DE FILAS Y COLUMNAS PARA NUESTRO SITIO RESPONSIVO



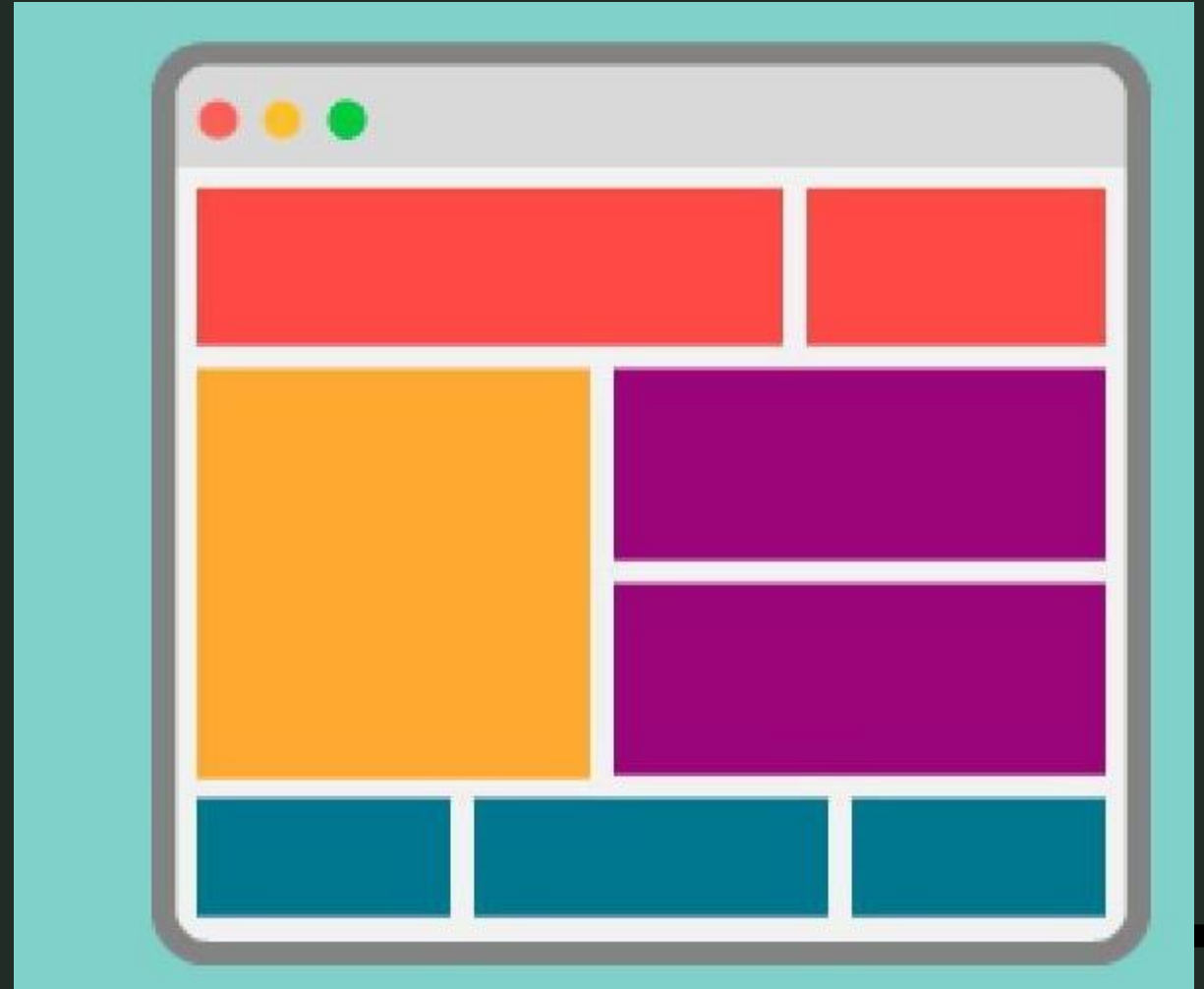
LA VISTA DE UNA PÁGINA WEB VA A VARIAR CON EL
DISPOSITIVO Y SERÁ MENOR EN UN TELÉFONO MÓVIL QUE
EN UNA PANTALLA DE COMPUTADORA.



Los primero sitios web

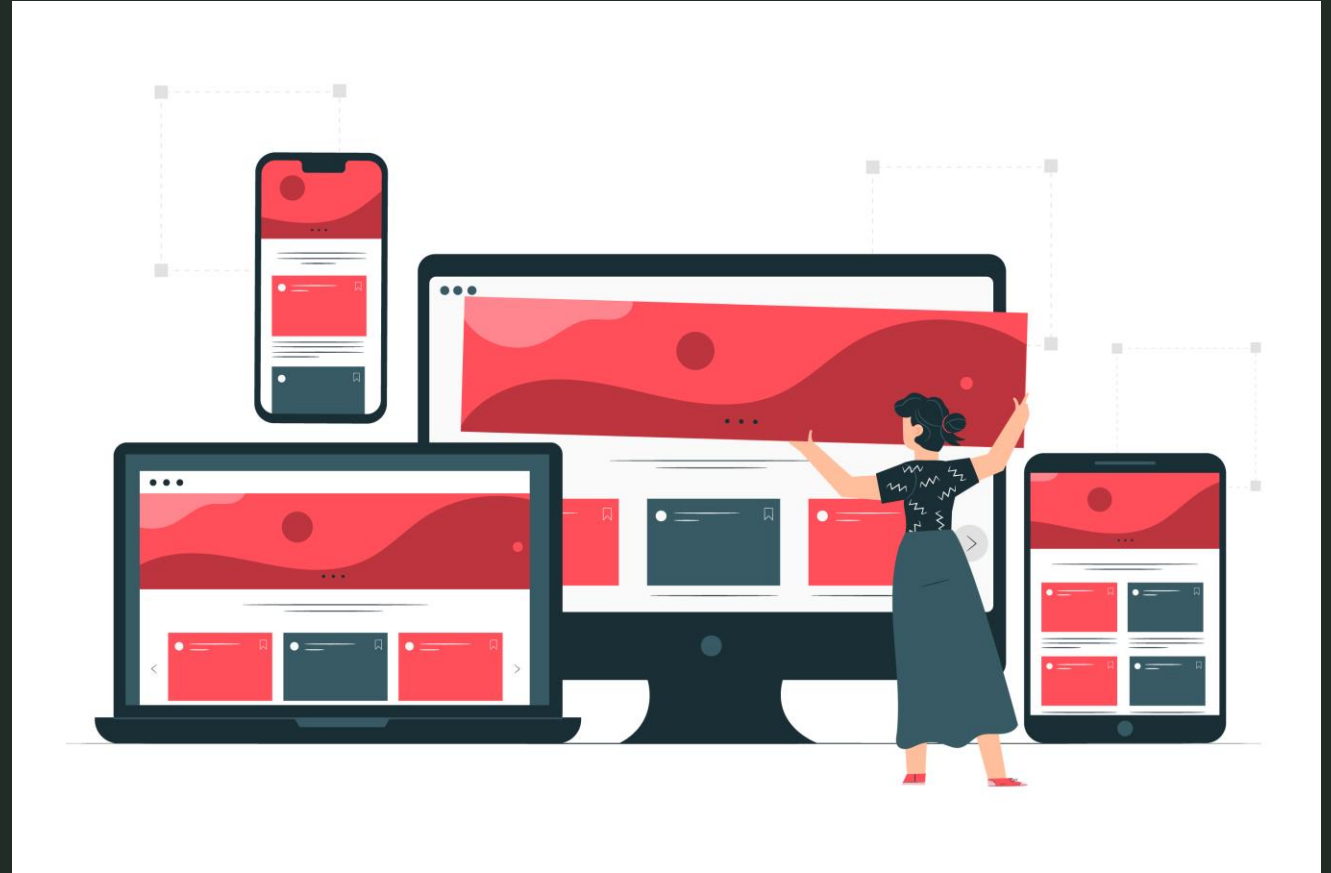
Antes de las tabletas y teléfonos móviles, las páginas web estaban diseñadas sólo para pantallas de ordenadores, y era común que las páginas web tuvieran un diseño estático y un tamaño fijo.

Debido a esto era muy difícil acceder a los sitios desde Tablet o teléfonos.

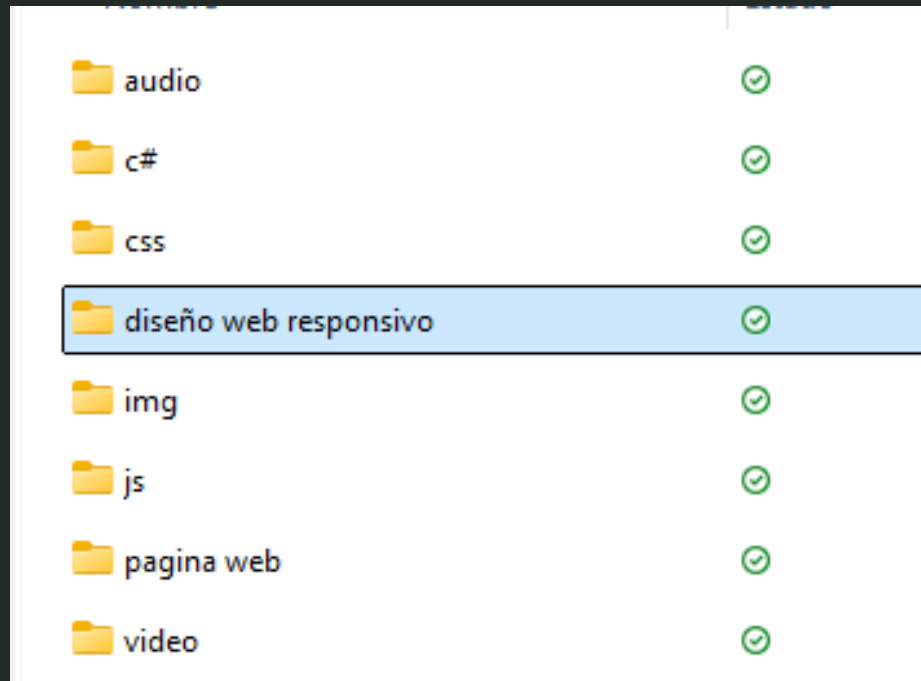


La solución

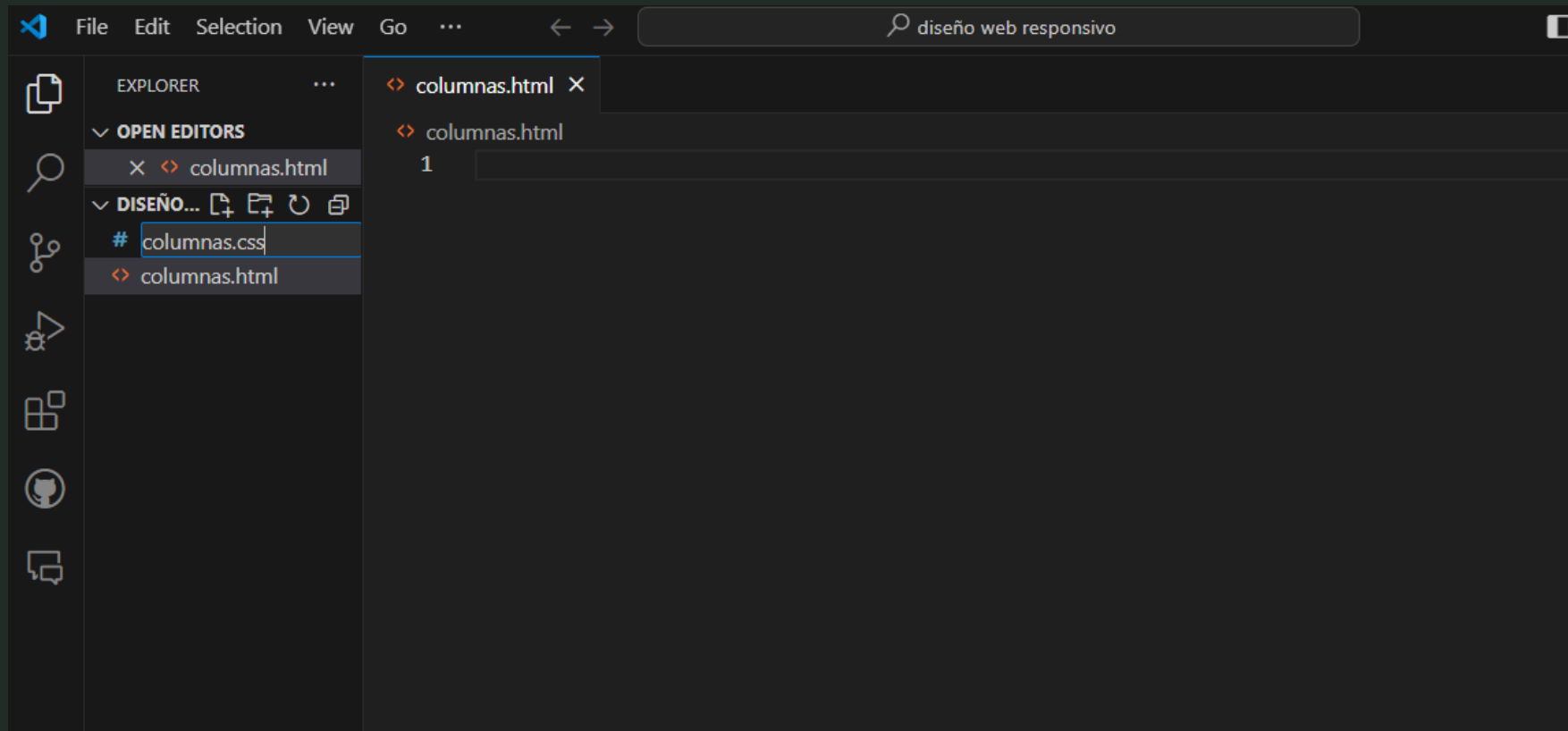
Para solucionar este inconveniente que se tenía, los navegadores de estos dispositivos redujeron las páginas web enteras para ajustarlas a las pantallas.



En nuestra carpeta html y css crearemos una carpeta con el nombre Diseño web responsivo



Luego en VSC abriremos carpeta y crearemos un index.html, un archivo con el nombre columnas.html y y otro con el nombre columnas css.

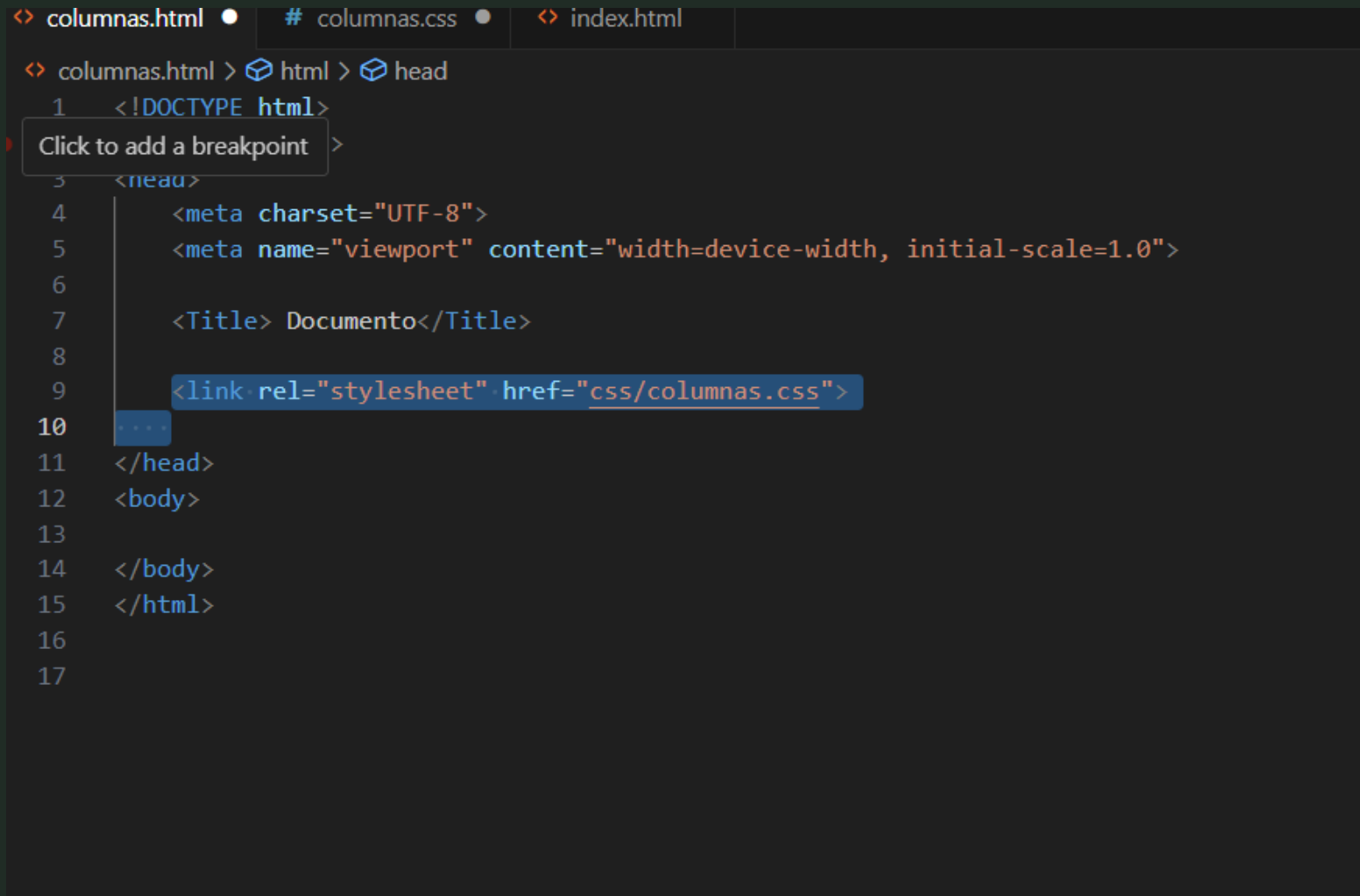


Añadiremos una estructura html con la etiqueta viewport

```
<> columnas.html ● # columnas.css ● <> index.html
<> columnas.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6
7      <Title> Documento</Title>
8
9
10 </head>
11 <body>
12
13 </body>
14 </html>
15
16 |
```



Vamos colocar en nuestra hoja columna.html el enlace para nuestra hoja de style columna.css



```
<> columnas.html • # columnas.css • <> index.html
<> columnas.html > html > head
1 <!DOCTYPE html>
Click to add a breakpoint >
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6
7   <Title> Documento</Title>
8
9   <link rel="stylesheet" href="css/columnas.css">
10  ...
11 </head>
12 <body>
13
14 </body>
15 </html>
16
17
```



Por ultimo añadiremos estilos generales a nuestra hoja de estilo columnas .css

```
*{  
  margin: 0px;  
  padding:0px;  
  list-style: none;  
  text-decoration: none;  
  font-family: sans-serif;  
}
```



Guardamos y visualizamos cambios

```
<> columnas.html ● # columnas.css ● <> index.html
# columnas.css > *
1  *{
2    margin: 0px;
3    padding:0px;
4    list-style: none;
5    text-decoration: none;
6    font-family: sans-serif;
7
8  }
```



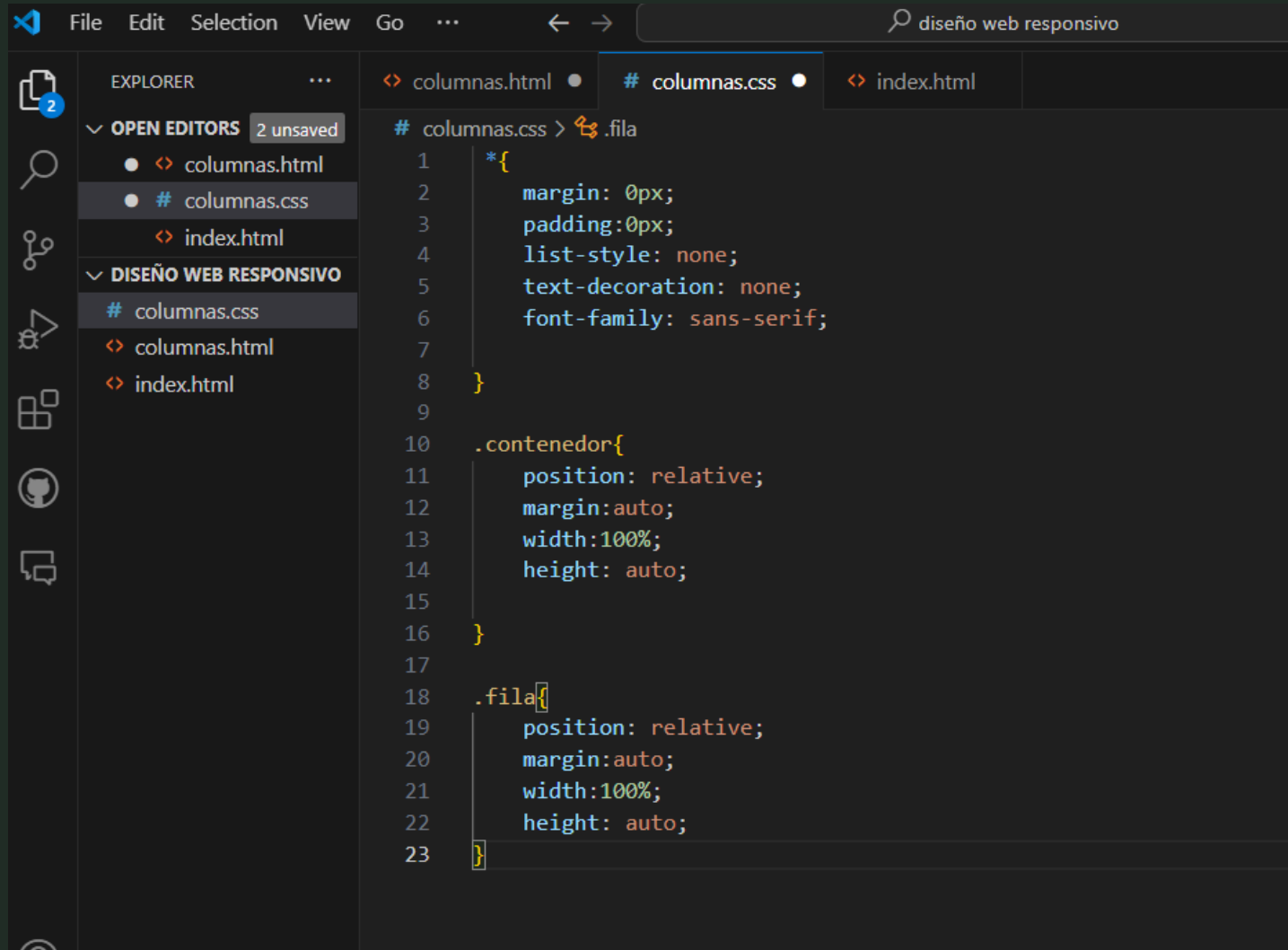
Añadiremos también un un clase con el nombre de contenedor y una clase filas con las mismas propiedades del contenedor.

```
.contenedor{  
    position: relative;  
    margin:auto;  
    width:100%;  
    height: auto;  
}
```

```
.fila{  
    position: relative;  
    margin:auto;  
    width:100%;  
    height: auto;  
}
```



Guardamos y visualizamos cambios



The screenshot shows the Visual Studio Code interface with the following components:

- Explorer Panel:** Shows the file structure with 'columnas.html', 'columnas.css', and 'index.html'. The 'DISEÑO WEB RESPONSIVO' section is expanded, showing the same three files.
- Open Editors:** A list of open files including 'columnas.html', 'columnas.css', and 'index.html'. The 'columnas.css' file is highlighted.
- Main Editor:** Displays the content of 'columnas.css'. The code is as follows:

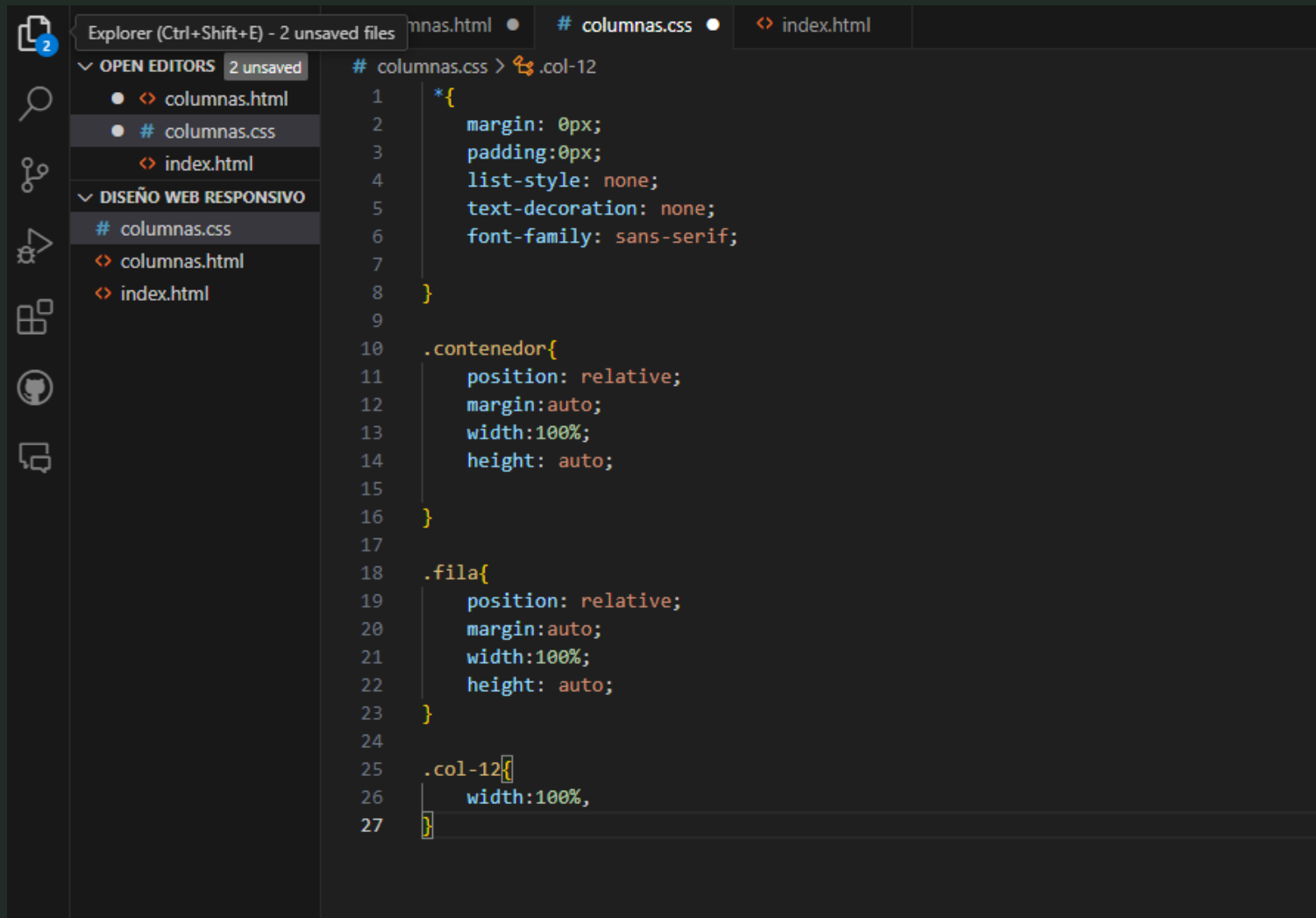
```
# columnas.css > .fila
1  *{
2      margin: 0px;
3      padding: 0px;
4      list-style: none;
5      text-decoration: none;
6      font-family: sans-serif;
7
8  }
9
10 .contenedor{
11     position: relative;
12     margin: auto;
13     width: 100%;
14     height: auto;
15 }
16
17
18 .fila{
19     position: relative;
20     margin: auto;
21     width: 100%;
22     height: auto;
23 }
```

Por último crearemos la clase columna

```
.col-12{  
  width:100%,  
}
```



Guardamos y visualizamos



The screenshot shows the Visual Studio Code interface with three files open: `columnas.html`, `columnas.css`, and `index.html`. The `columnas.css` file is active, displaying the following CSS code:

```
1  *{
2      margin: 0px;
3      padding: 0px;
4      list-style: none;
5      text-decoration: none;
6      font-family: sans-serif;
7  }
8
9
10 .contenedor{
11     position: relative;
12     margin: auto;
13     width: 100%;
14     height: auto;
15 }
16
17
18 .fila{
19     position: relative;
20     margin: auto;
21     width: 100%;
22     height: auto;
23 }
24
25 .col-12{
26     width: 100%;
27 }
```

The left sidebar shows the Explorer view with the file structure:

- OPEN EDITORS (2 unsaved)
 - columnas.html
 - columnas.css
 - index.html
- DISEÑO WEB RESPONSIVO
 - columnas.css
 - columnas.html
 - index.html

Actividad Grupal:

- 1- Realizar el cuestionario para la asistencia en el campus .
- 2- Dentro de la carpeta html y css , crear una nueva carpeta con el nombre Diseño web responsivo que contenga los archivos index.html, columnas.html y columnas.css, añadir la etiqueta viewport y el enlace a la hoja de style de nombre columnas.css en la hoja columna. html.
- 3-Dentro de la hoja de estilo columna.css, añadiremos los atributos generales y las clases de contenedor, fila y columna.
- 4- Realizar una 1 captura por grupo y enviar con nombre del grupo y alumnos que participaron