Programación Front End

UNIDAD 1



Tabla de CONTENIDOS

Semana 1:
Introducción a la programación Front End y los lenguajes e programación..... **03**

Semana 2:
Tutorial interactivo de HTML: identificación de elementos a través de un *template*..... **05**

Semana 3:
Tutorial práctico: convertir un diseño en *HTML*..... **07**

Semana 4:
Guía práctica de CSS3: aplicación de estilos según un diseño..... **09**

Semana 5:
Guía de Git y GitHub para desarrolladores: implementación de políticas de respaldo..... **11**

Semana 6:
Sitio web con HTML5 y CCS5..... **13**



Semana 1:
Introducción a la programación Front End y los lenguajes de programación

La programación Front End se centra en el desarrollo de la parte visual e interactiva de un sitio web, utilizando lenguajes específicos que permiten crear experiencias de usuario ricas y funcionales.



Los lenguajes fundamentales para el desarrollo Front End son *HTML, CSS* y JavaScript. *HTML* proporciona la estructura básica de las páginas web, *CSS* se encarga del estilo y la presentación, y JavaScript permite la interactividad y el comportamiento dinámico. Juntos, estos lenguajes forman la base sobre la cual se construyen sitios web modernos, accesibles y visualmente atractivos.

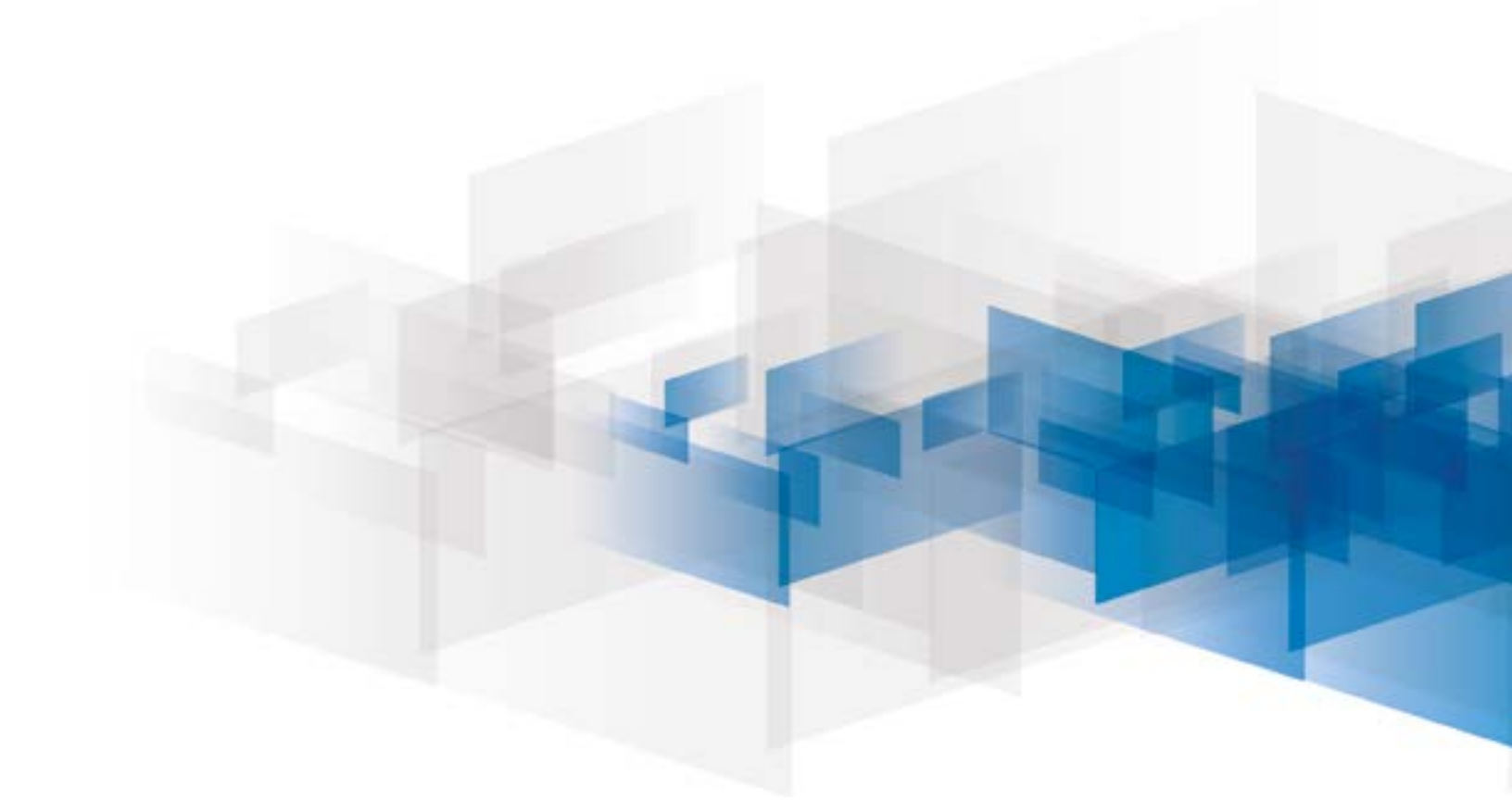
Aprender a programar en Front End es esencial para cualquier aspirante a desarrollador web, ya que estas habilidades son necesarias para crear interfaces que no solo sean estéticamente agradables, sino también funcionales y fáciles de usar. Con el dominio de *HTML, CSS* y JavaScript, los estudiantes pueden transformar un diseño estático en una aplicación web interactiva, mejorando la experiencia del usuario y haciendo que el contenido sea accesible en una amplia variedad de dispositivos y navegadores. La programación Front End también sienta las bases para el aprendizaje de tecnologías y marcos más avanzados, como React, Angular y Vue.js, que son utilizados en el desarrollo de aplicaciones web modernas y sofisticadas.

En resumen, ten presente:

1. La programación Front End es fundamental para crear experiencias de usuario atractivas y funcionales en sitios web.

2. *HTML, CSS* y JavaScript son los pilares del desarrollo Front End, permitiendo estructurar, diseñar y dinamizar páginas web.

3. Dominar los lenguajes de programación Front End es esencial para cualquier desarrollador web que desee crear sitios modernos y responsive.



Semana 2:
Tutorial interactivo de HTML: identificación de elementos a través de un *template*

HTML (*HyperText Markup Language*) es el lenguaje de marcado estándar utilizado para estructurar y presentar contenido en la web. A través del uso de etiquetas y atributos, *HTML* define elementos como encabezados, párrafos, enlaces, imágenes y formularios, permitiendo la creación de páginas web accesibles y funcionales. Las etiquetas estructuran el contenido, mientras que los atributos proporcionan información adicional que mejora la funcionalidad y el diseño del sitio. *HTML* es fundamental en el desarrollo web, sirviendo como la base sobre la cual se construyen las experiencias web modernas junto con *CSS* y JavaScript.



Aprender a programar en Front End es esencial para cualquier aspirante a desarrollador web, ya que estas habilidades son necesarias para crear interfaces que no solo sean estéticamente agradables, sino también funcionales y fáciles de usar. Con el dominio de HTML, CSS y JavaScript, los estudiantes pueden transformar un diseño estático en una aplicación web interactiva, mejorando la experiencia del usuario y haciendo que el contenido sea accesible en una amplia variedad de dispositivos y navegadores. La programación Front End también sienta las bases para el aprendizaje de tecnologías y marcos más avanzados, como React, Angular y Vue.js, que son utilizados en el desarrollo de aplicaciones web modernas y sofisticadas.

En resumen, ten presente:

1. La programación Front End es fundamental para crear experiencias de usuario atractivas y funcionales en sitios web.

2. *HTML, CSS* y JavaScript son los pilares del desarrollo Front End, permitiendo estructurar, diseñar y dinamizar páginas web.

3. Dominar los lenguajes de programación Front End es esencial para cualquier desarrollador web que desee crear sitios modernos y responsive.

Semana 3:
Tutorial práctico: convertir un diseño en *HTML*

El tutorial práctico “convertir un diseño en HTML” guía a los desarrolladores a transformar un diseño gráfico en una página web funcional usando *HTML.* Este proceso comienza con la interpretación del diseño, donde se identifican los distintos elementos y su disposición en la página. Luego, se procede a construir la estructura del *HTML,* creando contenedores para los diferentes componentes como encabezados, párrafos, imágenes y enlaces, asegurando que la estructura sea semántica y accesible.



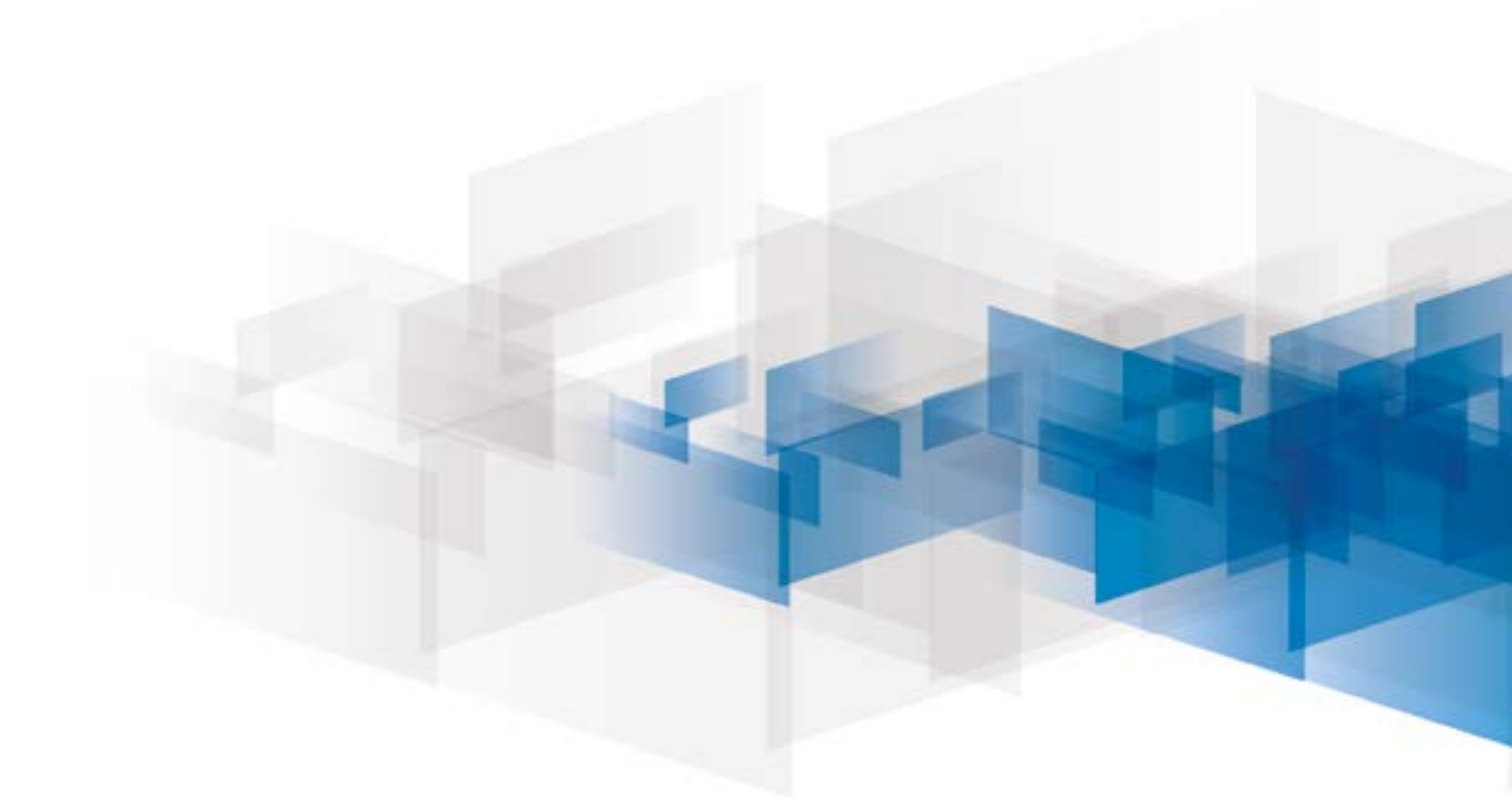
Además, el tutorial enfatiza la importancia de utilizar etiquetas HTML correctas y anidar adecuadamente los elementos, para mantener la coherencia y legibilidad del código. Se enseñan prácticas recomendadas para organizar el código, como el uso de comentarios y una estructura limpia, lo que facilita el mantenimiento y las futuras modificaciones. Al finalizar el tutorial, los desarrolladores tendrán una base sólida para convertir cualquier diseño en una página HTML bien estructurada y funcional.

En resumen, ten presente:

1. Convertir un diseño gráfico en *HTML* implica interpretar correctamente los elementos del diseño y crear una estructura *HTML* semántica y accesible.

2. El uso adecuado de etiquetas *HTML* y la organización limpia del código son cruciales para mantener la coherencia y facilitar el mantenimiento de la página web.

3. Este proceso proporciona a los desarrolladores las habilidades necesarias para transformar diseños en páginas web funcionales y bien estructuradas.



Semana 4:
Guía práctica de CSS3: aplicación de estilos según un diseño

La “guía práctica de CSS3: aplicación de estilos según un diseño” proporciona un enfoque detallado sobre cómo transformar un diseño visual en una página web estilizada utilizando CSS3. Esta guía comienza con la identificación de los elementos de diseño y sus propiedades visuales, como colores, tipografía, márgenes y alineaciones. Luego, se explica cómo aplicar estos estilos a través de *CSS*, utilizando selectores, clases e identificadores de manera efectiva para mantener el código organizado y modular.



Además, la guía destaca la importancia de usar características avanzadas de CSS3, como flexbox, grid layout, y animaciones, para lograr un diseño responsive y dinámico, que se adapta a diferentes dispositivos y tamaños de pantalla. Al seguir esta guía, los desarrolladores aprenderán a aplicar estilos de manera precisa y eficiente, replicando fielmente el diseño original en la web y asegurando una experiencia de usuario coherente y atractiva.

En resumen, ten presente

1. La guía enseña a transformar un diseño visual en estilos CSS3 precisos, abarcando colores, tipografía, márgenes y alineaciones.

2. El uso de características avanzadas de CSS3, como flexbox y grid layout, permite crear diseños responsive y adaptables a distintos dispositivos.

3. La organización y modularidad en el uso de selectores, clases e identificadores es clave para mantener un código CSS limpio y eficiente.

Semana 5:
Guía de Git y GitHub para desarrolladores: implementación de políticas de respaldo

La “guía de Git y GitHub para desarrolladores: implementación de políticas de respaldo” proporciona una comprensión detallada sobre cómo utilizar Git y GitHub para gestionar y respaldar proyectos de desarrollo de manera efectiva. La guía cubre la configuración de repositorios, la creación de ramas y la implementación de políticas de *commit* para asegurar que todos los cambios en el código estén bien documentados y sean recuperables.



Se enfatiza la importancia de los respaldos regulares y el uso de pull requests para revisar y validar cambios antes de integrarlos en la rama principal.

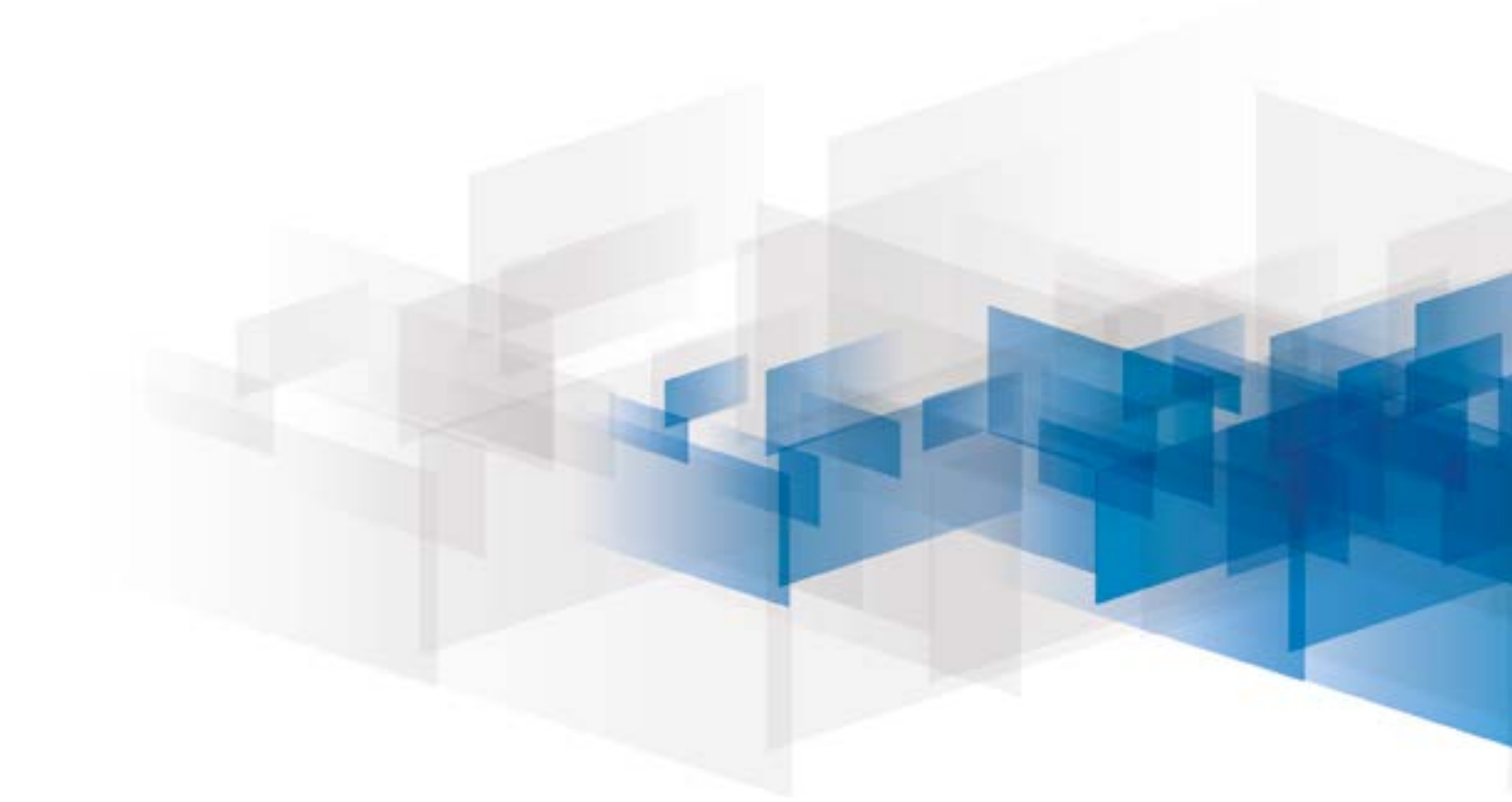
Además, la guía destaca las mejores prácticas para colaborar en equipo utilizando GitHub, como la asignación de revisores, la creación de issues para rastrear errores y la utilización de acciones de GitHub para automatizar tareas de integración y despliegue continuo. Al seguir esta guía, los desarrolladores podrán mantener un historial claro y organizado de sus proyectos, facilitando la colaboración y asegurando que el código esté siempre protegido contra pérdidas y errores.

En resumen, ten presente:

1. Implementar políticas de *commit* y crear ramas en Git y GitHub es crucial para mantener un historial de cambios claro y recuperable.

2. Los respaldos regulares y el uso de *pull requests* son esenciales para validar y documentar los cambios antes de integrarlos en la rama principal.

3. La colaboración en equipo se optimiza con la asignación de revisores, el seguimiento de errores a través de *issues* y la automatización de tareas con acciones de GitHub.



Semana 6:
Sitio web con HTML5 y CCS5

La evaluación sumativa 1, titulada “sitio web con HTML5 y CSS3”, tiene como objetivo que los estudiantes apliquen sus conocimientos en la creación de una página web completa utilizando HTML5 y CSS3.

Esta evaluación práctica requiere que los estudiantes diseñen y estructuren una página web que sea semántica y accesible, haciendo uso de las etiquetas y elementos propios de HTML5. Además, deben aplicar estilos CSS3 para lograr un diseño visual atractivo y responsivo, incorporando propiedades como colores, fuentes, *layout* y técnicas avanzadas como flexbox y grid.



A lo largo de esta evaluación, los estudiantes deben demostrar su habilidad para crear una estructura de HTML5 bien organizada y estilizarla adecuadamente con CSS3. El enfoque está en la creación de un diseño cohesivo y funcional que no solo sea estéticamente agradable, sino también accesible y adaptable a diferentes dispositivos. Esta evaluación no solo mide el conocimiento técnico de los estudiantes, sino también su capacidad para implementar buenas prácticas de desarrollo web y atención al detalle en la construcción de un sitio web.

En resumen, ten presente:

1. La evaluación requiere la creación de una página web completa y semántica utilizando HTML5, destacando la estructura y accesibilidad del contenido.

2. Se deben aplicar estilos CSS3 para lograr un diseño visual atractivo y *responsive*, utilizando propiedades y técnicas avanzadas como flexbox y grid.

3. La evaluación mide tanto el conocimiento técnico como la capacidad para implementar buenas prácticas de desarrollo web y atención al detalle.

