



PROCESO DIRECCIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

GUÍA DE APRENDIZAJE BASE DE DATOS

IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- **Denominación del programa de formación:** Análisis Y Desarrollo De Software.
 - **Código del programa de formación:** 228118
 - **Nombre del proyecto:**
 - **Fase del proyecto:**
 - **Actividad de proyecto:**
 - **Competencia:** Diseñar la solución de software de acuerdo con procedimientos y requisitos técnicos
 - **Resultados de aprendizaje alcanzar:** Elaborar los artefactos de diseño del software siguiendo las prácticas de la metodología seleccionada.
-
- **2. PRESENTACIÓN**

En esta guía se aplica las habilidades técnicas: desarrollar soluciones de software de acuerdo con los métodos de diseño y desarrollo Según a la Medida.

Los artefactos de diseño del software son documentos o entregables que describen los aspectos técnicos y funcionales de un sistema o aplicación, pero nos enfocamos en artefactos de código.

Debes leer detenidamente el diseño curricular como mapa de comprensión de los contenidos de la competencia y lo que concierne al resultado de aprendizaje, pero se aclara que la elección de equipo de ejecución serán artefactos realizados mediante html, css y javascript

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

3.1 Actividades de Reflexión inicial.

3.1.1. Observar la siguiente imagen y mediante un análisis, resolver los siguientes interrogantes:

Los artefactos de diseño del software son documentos o entregables que describen los aspectos técnicos y funcionales de un sistema o aplicación. En el contexto de una metodología de desarrollo de software, donde pueden incluir:

- Diagramas de clases: Representan la estructura de las clases y sus relaciones.
- Diagramas de secuencia: Muestran la interacción entre los objetos y cómo se realiza una tarea específica.
- Modelos de datos: Describen la estructura de los datos que se manejan en el sistema.
- Especificaciones técnicas: Describen detalladamente cómo se construirá el sistema.
- Plan de pruebas: Describen las pruebas que se realizarán para garantizar la calidad del software.



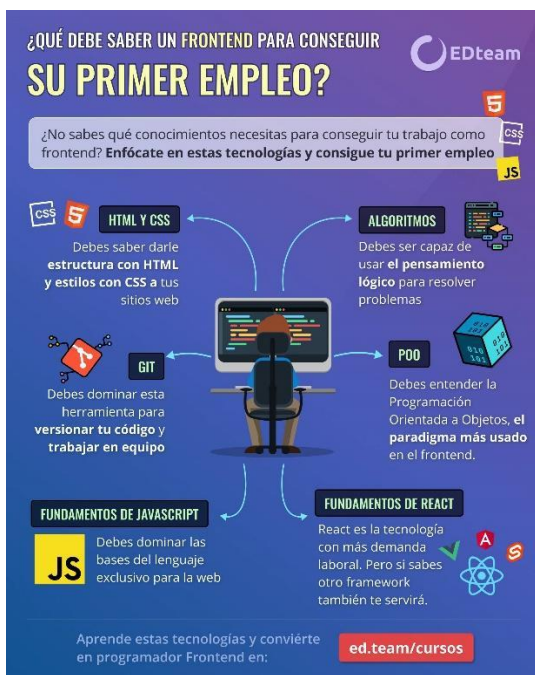
Estos son solo algunos ejemplos de artefactos de diseño. La selección de artefactos puede variar dependiendo de la metodología seleccionada. Por ejemplo, en una metodología ágil, los artefactos pueden ser más simples y se enfocan en la colaboración y la comunicación continua con los equipos de desarrollo y los clientes. Mientras que, en una metodología más tradicional, como el Proceso Unificado de Desarrollo de Software, puede haber una documentación más detallada y rigurosa.

Interrogantes:

3.1.2 Realizar el siguiente cuestionario de diseño de artefactos y consignar en el informe semanal de la formación.

1. ¿Qué es un caso de uso en el desarrollo de software?
2. ¿Cuál es la importancia de la documentación técnica en el diseño de software?
3. ¿Cómo se utiliza un diagrama de clases en el desarrollo de software orientado a objetos?
4. ¿Qué es un modelo de arquitectura de software y cómo se utiliza en el proceso de diseño?
5. ¿Cómo se puede mejorar la calidad del software a través de la revisión de código?
6. ¿Cuál es la diferencia entre un diagrama de secuencia y un diagrama de colaboración en UML?
7. ¿Cómo se asegura la escalabilidad en el diseño de software?
8. ¿Qué es un prototipo en el desarrollo de software y para qué se utiliza?
9. ¿Cómo se integran los requisitos del usuario en el proceso de diseño de software?
10. ¿Cómo se puede mejorar la eficiencia y el rendimiento en el desarrollo de software?

3.1.3 Analizar la infografía y realizar conclusiones de conocimientos que debe tener un FRONTEND



3.1.4 Solucionar el cuestionario de artefactos de desarrollo de software y consignarlos en el informe semanal de la formación.

1. ¿Qué es un artefacto de desarrollo de software?
2. ¿Cuáles son algunos ejemplos comunes de artefactos de desarrollo de software?



3. ¿Cuál es la importancia de los artefactos de desarrollo de software en el proceso de desarrollo de software?
4. ¿Qué diferencia hay entre un artefacto y un producto de software?
5. ¿Cómo se almacenan y gestionan los artefactos de desarrollo de software en un proyecto de software?
6. ¿Qué papel desempeñan los artefactos de desarrollo de software en la gestión de la configuración de software?
7. ¿Cómo se utilizan los artefactos de desarrollo de software en la integración continua y el despliegue continuo?
8. ¿Cómo se asegura la calidad de los artefactos de desarrollo de software?
9. ¿Cómo se asegura la trazabilidad de los artefactos de desarrollo de software a lo largo del ciclo de vida del software?

3.1.5 realizar mapa mental de los frontend

Información la analizas del link entregado en esta guía

<https://www.trabajofrontend.com/blog/5-tipos-desarrolladores-frontend>

Nota:

Todas estas actividades deben ir en el informe final del trimestre.

3.2. Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje

Dentro de la competencia y el enfoque queda el resultado de aprendizaje se estará trabajando artefactos de desarrollo de software, según la línea de profundización del centro, para ello los temas a contextualización son:

Los artefactos de diseño del software son documentos que describen y describen cómo se desarrollará un software en términos de su funcionalidad, estructura y comportamiento. En el caso de HTML, CSS y JavaScript, algunos de los artefactos de diseño más comunes incluyen:

1. Wireframes: Es una representación básica y simplificada de cómo se verá una página web. Se utiliza para definir la estructura de la página y cómo se distribuirán los elementos en la pantalla.
2. Maquetas: Son versiones más detalladas de los wireframes, que incluyen más información visual y estética sobre cómo se verá una página web.
3. Especificaciones técnicas: Describen los requisitos técnicos específicos para el desarrollo de una aplicación web, incluyendo detalles sobre la funcionalidad, la estructura de la página y la interacción del usuario con la página.
4. Diagramas de flujo: Muestran cómo se desarrollarán las interacciones y los procesos en una aplicación web, incluyendo cualquier navegación, procesamiento de datos y eventos de usuario.
5. Uso de plantillas y mejoras de ellas para paginas web
6. Pruebas de usuario: Describen cómo se probarán las funcionalidades y se evaluará la experiencia del usuario en la aplicación web.
7. El desarrollo de la parte frontal (front-end) de una aplicación web implica conocimientos en varias áreas, tales como:



8. HTML: Lenguaje de marcado para crear estructuras y contenido en una página web.
9. CSS: Lenguaje de estilo para dar formato y diseño a las páginas web.
10. JavaScript: Lenguaje de programación que permite añadir interactividad y dinamismo a las páginas web.
11. DOM (Document Object Model): Modelo de objetos que representa el contenido y estructura de una página web.

3.2.1 Responder los siguientes interrogantes

Nota:

La solución de estas actividades debe ir consignadas en el informe final de trimestre.

3.3 Actividades de apropiación del conocimiento (Conceptualización y Teorización).

El éxito de aprender algoritmos de programación depende de muchos factores, pero algunos de los más importantes son los siguientes:

- **Comprensión:** Es fundamental comprender los conceptos subyacentes a los algoritmos para poder aplicarlos de manera efectiva.
- **Práctica:** La práctica constante es esencial para mejorar las habilidades y el conocimiento de los algoritmos.
- **Conocimiento de lenguajes de programación:** Es importante tener un conocimiento sólido de los lenguajes de programación en los que se desea aplicar los algoritmos.
- **Resolución de problemas:** La capacidad de resolver problemas es un aspecto clave para el éxito en la programación de algoritmos, ya que permite aplicar los conceptos de manera efectiva.
- **Enfoque:** Mantener un enfoque concentrado en el objetivo y la resolución del problema es crucial para el éxito en la programación de algoritmos.
- **Comunicación:** La capacidad de comunicar efectivamente los algoritmos y los resultados obtenidos es importante para trabajar en equipo y para la revisión y mejora continua del código.
- **Colaboración:** Trabajar en colaboración con otros programadores y recibir retroalimentación es un aspecto importante para el éxito en la programación de algoritmos.

En resumen, el éxito en la programación de algoritmos depende de una combinación de comprensión, práctica, conocimiento de lenguajes de programación, resolución de problemas, enfoque, comunicación y colaboración.

Realizar las siguientes actividades conocimiento:

GFPI-F-135 V01

3.3.1 Realizar taller básico de HTML



- 3.3.2 Resolver Hacer taller intermedio de HTML
- 3.3.3 Hacer el taller Básico de CSS.
- 3.3.4 Desarrollar el taller intermedio de CSS.
- 3.3.5 Realizar taller básico de conocimientos de sentencia JavaScript.

Realizar actividades de desempeño.

- 3.3.6 Elaborar taller de interfaces web

Realizar actividades de producto:

- 3.3.7 Crear 5 Interfaces web no funcional del proyecto formativo

Valor total de actividad: 40%

4. ACTIVIDADES DE EVALUACIÓN

- 4.1.1 Realizar Prueba de conocimientos CSS,html (teoría). Valor calificable 15%
- 4.1.2 Realizar Prueba de conocimientos sentencias JavaScript para paginas web (habilidades y desarrollo de codificación) valor calificable 30%.
- 4.1.3 Realizar Prueba de conocimientos web de JavaScript, html, css para páginas web funcional (habilidades y desarrollo de codificación) valor calificable 25%.

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento: 4.1.1 Realizar Prueba de conocimientos CSS,html (teoría). Valor calificable 15% 4.1.2 Realizar Prueba de conocimientos sentencias JavaScript para paginas web (habilidades y desarrollo de codificación) valor calificable 25%. 4.1.3 Realizar Prueba de conocimientos web de JavaScript, html, css para páginas web funcional (habilidades y desarrollo de codificación) valor calificable 25%. Evidencias de Desempeño 3.3.1 Realizar taller básico de HTML	Elaborar los artefactos de diseño del software siguiendo las prácticas de la metodología. * interpreta el informe de análisis identificando las características del software a diseñar. * define las plataformas tecnológicas a emplear en el desarrollo de acuerdo con las condiciones del software a desarrollar. * define los entregables de diseño siguiendo los conceptos y principios de orientación a objetos. * crea el diagrama de clases de acuerdo con los requisitos,	Prueba de Conocimiento Listas de chequeo desempeño Listas de chequeo producto Cuestionarios.



3.3.2	Resolver Hacer taller intermedio de HTML	aplicando buenas prácticas de diseño orientado a objetos.	
3.3.3	Hacer el taller Básico de CSS.	* incorpora patrones de diseño	
3.3.4	Desarrollar el taller intermedio de CSS.	propendiendo en mejores prácticas para la codificación y mantenibilidad del software.	
3.3.5	Realizar taller básico de conocimientos de sentencia JavaScript.	* define la arquitectura del software dando cumplimiento a los requisitos funcionales y no funcionales.	
Evidencias de Producto:		* crea la vista de componentes para visualizar el software en fases avanzadas del ciclo de vida.	
3.3.1	Crear 5 Interfaces web no funcional del proyecto formativo	* crea la vista de despliegue del software para determinar condiciones de la implantación de la solución informática.	

4. GLOSARIO DE TÉRMINOS

CSS

- Selector: un patrón utilizado para seleccionar elementos HTML a los que se aplicarán estilos CSS.
- Propiedad: un atributo que se aplica a un selector para definir su estilo visual.
- Valor: el valor asignado a una propiedad CSS.
- Declaración: una combinación de propiedad y valor que se aplica a un selector.
- Clase: una etiqueta que se utiliza para agrupar elementos HTML que comparten un estilo común.
- ID: un atributo único que se utiliza para identificar un elemento HTML en particular.
- Pseudo-clase: un selector que se aplica a un elemento solo cuando se encuentra en un estado específico, como cuando el mouse está sobre él.
- Pseudo-elemento: un selector que se aplica a una parte específica de un elemento, como el primer párrafo dentro de un contenedor.
- Herencia: el proceso por el cual los estilos aplicados a un elemento también se aplican a sus elementos secundarios.
- Box Model: un modelo de caja que describe cómo se renderiza un elemento HTML, incluyendo su ancho, alto, margen y relleno.

HTML



- Elemento: una etiqueta que se utiliza para marcar contenido en HTML, como un encabezado o un párrafo.
- Atributo: un valor que se utiliza para modificar el comportamiento o estilo de un elemento HTML, como el color o la alineación.
- Etiqueta: un término genérico que se utiliza para describir tanto los elementos HTML como los atributos.
- Anidación: el proceso de colocar un elemento dentro de otro para crear una estructura jerárquica de contenido.
- DOM (Modelo de Objeto de Documento): una representación programática del documento HTML que se puede manipular con JavaScript.
- Enlace (Link): un elemento que se utiliza para crear hipervínculos entre páginas web.
- Atributo href: un atributo de enlace que especifica la URL de destino del enlace.
- Imagen (Image): un elemento que se utiliza para mostrar imágenes en una página web.
- Atributo src: un atributo de imagen que especifica la ubicación de la imagen.
- Lista (List): un elemento que se utiliza para mostrar una lista de elementos relacionados.

JavaScript

- JavaScript: Lenguaje de programación interpretado, utilizado en el desarrollo web y otros entornos.
- Variable: Espacio en memoria que se utiliza para almacenar datos.
- Función: Bloque de código que se puede llamar y ejecutar en cualquier momento.
- Array: Estructura de datos que permite almacenar múltiples elementos en una sola variable.
- Objeto: Entidad que agrupa datos y funciones relacionadas.
- Condicionales: Estructuras de control que permiten ejecutar diferentes bloques de código según una condición.
- Ciclos: Estructuras de control que permiten repetir bloques de código varias veces.
- Eventos: Acciones del usuario o del navegador que se pueden detectar y manejar en el código.
- DOM: Document Object Model, interfaz de programación que permite acceder y manipular elementos de una página web.
- AJAX: Asynchronous JavaScript and XML, técnica de programación que permite actualizar partes de una página web sin tener que recargarla completamente.
- JSON: JavaScript Object Notation, formato de intercambio de datos que se utiliza para transmitir información estructurada a través de la red.
- Callback: Función que se pasa como argumento a otra función y se ejecuta después de que se completa una tarea asíncrona.

6. REFERENTES BIBLIOGRÁFICOS

"HTML and CSS: Design and Build Websites" de Jon Duckett

"JavaScript and JQuery: Interactive Front-End Web Development" de Jon Duckett

"Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" de Jennifer Niederst Robbins

"CSS: The Definitive Guide: Visual Presentation for the Web" de Eric A. Meyer y Estelle Weyl

"JavaScript: The Definitive Guide: Activate Your Web Pages" de David Flanagan



"Eloquent JavaScript: A Modern Introduction to Programming" de Marijn Haverbeke

"Head First HTML and CSS: A Learner's Guide to Creating Standards-Based Web Pages" de Elisabeth Robson y Eric Freeman

"CSS Secrets: Better Solutions to Everyday Web Design Problems" de Lea Verou

"JavaScript: The Good Parts" de Douglas Crockford

"CSS Grid Layout: The Essential Guide" de Rachel Andrew.

HTML:

Duckett, J. (2011). HTML and CSS: Design and Build Websites. John Wiley & Sons.

W3C (World Wide Web Consortium). (2022). HTML Standard. <https://html.spec.whatwg.org/multipage/>

JavaScript:

Flanagan, D. (2011). JavaScript: The Definitive Guide. O'Reilly Media.

Eloquent JavaScript. (2021). 3rd Edition. <https://eloquentjavascript.net/>

CSS:

Meyer, E. A. (2017). CSS: The Definitive Guide. O'Reilly Media.

W3C (World Wide Web Consortium). (2022). Cascading Style Sheets (CSS) Standard. <https://www.w3.org/Style/CSS/Overview.en.html>

Algunas referencias bibliográficas para HTML, JavaScript y CSS son:

MDN Web Docs: <https://developer.mozilla.org/en-US/>

W3Schools: <https://www.w3schools.com/>

CSS Tricks: <https://css-tricks.com/>

A List Apart: <https://alistapart.com/>

Smashing Magazine: <https://www.smashingmagazine.com/>

Además, algunos fabricantes de navegadores y plataformas web también ofrecen documentación y recursos útiles:

Google Developers: <https://developers.google.com/>

Mozilla Developer Network: <https://developer.mozilla.org/en-US/>

Apple Developer: <https://developer.apple.com/>

Microsoft Developer Network: <https://developer.microsoft.com/>



Bootstrap: <https://getbootstrap.com/>

<https://desarrolladoresweb.org/curso-css-css3-completo-desde-0/>

<https://www.ejemplode.com/18-css/>

<https://www.w3schools.com/css/default.asp>

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Lilliana Uribe Gonzalez	INSTRUCTOR A		Febrero 12 DE 2023

8. CONTROL DE CAMBIOS

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)		INSTRUCTOR A INSTRUCTOR			Actualización