P2. Actividades de aprendizaje de la competencia 220501095 - Diseñar la solución de *software* de acuerdo con procedimientos y requisitos técnicos

Con base en los requisitos de *software* especificados tanto funcionales como no funcionales, en esta competencia se empieza a construir los prototipos del modelo necesarios para representar la solución del sistema desde el punto de vista del diseño, ergonomía y usabilidad con base en los siguientes documentos: diagramas de clase, documentos de casos de uso o historias de usuario, diagrama de componentes y también un documento de informe con los colores institucionales del cliente a manejar (paleta de colores) unido a una descripción de las tecnologías a utilizar.

3.1.1 Actividad de aprendizaje GA5-220501095-AA1 - Elaborar el prototipo navegable del software aplicando estándares de usabilidad y accesibilidad.

Esta actividad se centra en el estudio de la construcción de los prototipos del *software* con base en la definición de las reglas de navegación de las pantallas de visualización de cada funcionalidad y acordes con los requerimientos del sistema guiados por las buenas prácticas en mantenibilidad, diseño y usabilidad tendientes a satisfacer una gran experiencia de usuario.

Duración: 136 horas.

Materiales de formación: para el desarrollo de esta actividad es importante la lectura y análisis del material de formación: "**Diseño de aplicaciones web - Diseño y construcción de *frontend*".**

Evidencias:

A continuación, se describen las acciones y las correspondientes evidencias que conforman la actividad de aprendizaje:

P8. Diseño aplicación web

El **Diseño de Aplicaciones Web** es el proceso de planificación, estructuración y creación de interfaces para aplicaciones que se ejecutan en un navegador web. Implica la combinación de diseño gráfico, experiencia de usuario (UX), desarrollo front-end y, en algunos casos, desarrollo back-end para garantizar que la aplicación sea funcional, atractiva y fácil de usar.

¿Para qué sirve?

El diseño de aplicaciones web es fundamental para:

1. **Brindar una experiencia de usuario (UX) óptima**
    - Permite que los usuarios naveguen de manera intuitiva.
    - Mejora la accesibilidad y la usabilidad.
2. **Garantizar una interfaz visual atractiva (UI)**
    - Utiliza principios de diseño gráfico para hacer la aplicación visualmente atractiva.
    - Aplica conceptos como colores, tipografías y disposición de elementos.
3. **Optimizar la funcionalidad y rendimiento**

- Asegura que la aplicación web cargue rápido y funcione correctamente en diferentes dispositivos.
- Reduce errores y mejora la interacción del usuario.

4. **Hacer que la aplicación sea responsive**
   - Permite que la aplicación se adapte a diferentes tamaños de pantalla (móviles, tablets y computadoras).
5. **Facilitar el mantenimiento y escalabilidad**
   - Un buen diseño facilita futuras actualizaciones y mejoras en la aplicación.

Ejemplos de Aplicaciones Web

- **Google Drive** (almacenamiento en la nube)
- **Facebook** (red social)
- **Trello** (gestión de proyectos)
- **Gmail** (correo electrónico)

P9. Sistema de navegación en plataformas software

Un sistema de navegación permite que un usuario se pueda mover libremente por las diferentes secciones que soporta una plataforma.

En busca de la información.

Buscar información en línea enfrenta varios problemas y desafíos, como la sobrecarga de datos, la dificultad para evaluar la calidad y la credibilidad de las fuentes, y el riesgo de sesgo de confirmación. La estrategia de búsqueda suele ser flexible: a medida que se encuentra nueva información, los usuarios pueden ajustar y redirigir su búsqueda, adaptando su enfoque según los resultados obtenidos. Este proceso iterativo permite refinar la investigación pero también puede complicar la navegación eficiente a través de la vasta cantidad de recursos disponibles.

1. **Navegar por sitios web**

Buscar información en línea no solo involucra una actividad racional, sino también emocional. La "experiencia" de búsqueda incluye factores emocionales, conductuales y cognitivos, que influyen en cómo las personas interactúan con la información. En lugar de ser vistas como racionales y sistemáticas, las personas experimentan la búsqueda de manera más compleja y subjetiva. La arquitectura de la información en los sitios web, con secciones como "Inicio", "Servicios", "Nosotros", "Acerca de" y "Contactos", debe evaluarse cuidadosamente para mejorar la experiencia del usuario y facilitar una navegación efectiva.

P10. Patrón tradicional para el diseño de navegación de un sitio Web

El diseño de la navegación es crucial para la experiencia del usuario en un sitio web. Los diseñadores deben comprender los patrones de búsqueda, emociones y reacciones de los usuarios para tomar decisiones informadas sobre la estructura de navegación. Este conocimiento les permite elegir la opción de navegación más adecuada para el sitio, mejorando así la interacción y satisfacción del usuario.

Navegar por aplicaciones móviles

La navegación en aplicaciones móviles debe ser intuitiva, predecible y fácil de recordar, tanto para usuarios nuevos como recurrentes. La simplicidad en la navegación es esencial para que los usuarios puedan moverse cómodamente entre secciones. Sin embargo, las limitaciones de las pantallas pequeñas y la necesidad de priorizar el contenido sobre los elementos de la interfaz presentan desafíos. Existen varios patrones de navegación para abordar estos problemas, pero todos enfrentan ciertos problemas de usabilidad.

P11. En la imagen muestra un **diagrama de navegación web**, que representa la estructura jerárquica de un sitio web.

Análisis de la Imagen

- **Página Principal (Home)**: Es el punto de entrada principal del sitio.
- **Enlaces a secciones clave**:
    - **Contacto**: Información de contacto o formulario de comunicación.
    - **Empresas**: Información sobre empresas o perfiles de usuario.
    - **Ofertas**: Sección con promociones o productos destacados.
    - **Servicios**: Página con los servicios que se ofrecen.
    - **Blog**: Espacio con artículos o noticias.

Importancia de una buena estructura de navegación

1. **Mejora la experiencia del usuario (UX)** al facilitar la búsqueda de información.
2. **Optimiza el SEO** al organizar correctamente el contenido.
3. **Aumenta la retención de usuarios**, ya que encuentran lo que buscan rápidamente.

P12. Las páginas web han de ser hoy más eficientes e intuitivas aún para poder sobrevivir en un mercado online con una enorme competencia, ya sea en el PC de escritorio, la tablet o el teléfono móvil. El sobrecargado diseño de la década de los años 2000 ya es historia: la nueva máxima es la concentración en lo esencial.

Exactamente, hoy en día el diseño web debe enfocarse en la **eficiencia, la usabilidad y la experiencia del usuario (UX)**.

P13. **¿Qué es Wireframing?**

El **wireframing** es el proceso de diseñar una estructura visual básica de una página web o aplicación antes de agregar diseño y funcionalidad. Es un **boceto** que muestra la disposición de los elementos clave como menús, botones, imágenes y contenido, sin enfocarse en colores o estilos visuales.

¿Para qué sirve el Wireframing?

1. **Organizar la información**: Define cómo se distribuirá el contenido en la pantalla.
2. **Optimizar la experiencia del usuario (UX)**: Ayuda a diseñar una navegación intuitiva.
3. **Facilitar la comunicación**: Sirve como guía para diseñadores y desarrolladores.
4. **Ahorrar tiempo y recursos**: Detecta problemas antes de pasar a la fase de diseño y desarrollo.

Tipos de Wireframes

1. **Wireframes de baja fidelidad**
    - Dibujos a mano o en herramientas básicas.
    - Representación simple con formas geométricas.
    - Se enfocan solo en la estructura.
2. **Wireframes de media fidelidad**
    - Incluyen más detalles y texto real.
    - Se diseñan con herramientas digitales como Balsamiq o Figma.
3. **Wireframes de alta fidelidad**
    - Se parecen más al producto final.
    - Incluyen iconos, fuentes y tamaños de texto más definidos.
    - Se crean con herramientas avanzadas como **Adobe XD, Sketch o Figma**.

Ejemplo de un Proceso de Wireframing

1. **Definir objetivos**: ¿Qué necesita el usuario en esta pantalla?
2. **Dibujar bocetos**: Diseño rápido en papel o herramienta digital.
3. **Probar y recibir feedback**: Ajustes antes de pasar al diseño final.
4. **Convertirlo en prototipo**: Agregar interactividad con herramientas como **Figma o InVision**.

Herramientas Populares para Wireframing

✅ **Figma** – Colaboración en tiempo real.
✅ **Balsamiq** – Ideal para bocetos rápidos.
✅ **Adobe XD** – Prototipado y diseño en alta fidelidad.
✅ **Sketch** – Exclusivo para Mac, enfocado en diseño UI/UX.

P18. Principios Básicos de Usabilidad y Calidad del Software

La usabilidad y la calidad del software son fundamentales para garantizar que los usuarios puedan interactuar de manera eficiente, efectiva y satisfactoria con un sistema o aplicación.

P19. 🔷 Principios de Usabilidad

La usabilidad se refiere a la facilidad con la que los usuarios pueden utilizar un software o aplicación para alcanzar sus objetivos. Se basa en los siguientes principios clave:

1. Facilidad de Aprendizaje

📌 El software debe ser intuitivo, permitiendo a los nuevos usuarios entender rápidamente cómo utilizarlo.
✅ Uso de elementos familiares (íconos, menús estándar).
✅ Diseño coherente con otras aplicaciones similares.

2. Eficiencia

📌 Los usuarios deben poder completar sus tareas con rapidez y sin complicaciones.
✅ Optimización de flujos de trabajo.
✅ Minimización de clics o pasos innecesarios.

3. Memorabilidad

📌 Los usuarios ocasionales no deberían necesitar reaprender cómo funciona el software cada vez que lo usan.
✅ Interfaz intuitiva y consistente.
✅ Uso de patrones de diseño estándar.

4. Prevención de Errores

📌 El sistema debe evitar que los usuarios cometan errores o proporcionar soluciones rápidas cuando ocurran.
✅ Confirmaciones antes de realizar acciones importantes.
✅ Mensajes de error claros y soluciones sugeridas.

5. Satisfacción del Usuario

📌 La experiencia debe ser agradable, generando confianza y comodidad en el uso del software.
✅ Diseño atractivo y accesible.
✅ Respuesta rápida del sistema a las acciones del usuario.

🔷 **¿Por qué son importantes estos principios?**

✓ Un software usable y de alta calidad mejora la productividad.
✓ Reduce costos de soporte y mantenimiento.
✓ Aumenta la satisfacción del usuario y la lealtad a la marca.
✓ Mejora la competitividad en el mercado.

P20. **"Facilidad de aprendizaje"**, explica que la facilidad de aprendizaje de un sistema o herramienta puede medirse en función del tiempo que los usuarios tardan en usarla correctamente y en realizar operaciones básicas y comunes. Esto sugiere que una interfaz intuitiva y bien diseñada puede mejorar la experiencia del usuario al reducir el tiempo de aprendizaje.

"Satisfacción subjetiva", explica que la satisfacción subjetiva se refiere a qué tan satisfechos quedan los usuarios después de usar un sistema, influenciada por el diseño presentado. Se denomina subjetiva porque varía según la percepción individual de cada usuario. Esto indica que la experiencia del usuario no solo depende de la funcionalidad del sistema, sino también de su presentación y facilidad de uso.

"Frecuencia y gravedad de errores", explica que los mensajes de error son esenciales para guiar a los usuarios cuando enfrentan problemas dentro de un sistema. Se menciona que la frecuencia y gravedad de los errores pueden afectar la experiencia del usuario. Como ejemplo, se describe un error en un sistema de facturación donde se intenta generar una factura sin asociar al menos un producto o servicio. Esto resalta la importancia de proporcionar mensajes de error claros y efectivos para evitar confusiones y mejorar la usabilidad del sistema.

"Facilidad y eficiencia de uso", explica que la facilidad y eficiencia de uso pueden medirse con base en el tiempo que un usuario tarda en completar una operación utilizando una herramienta de trabajo. Esto sugiere que un sistema bien diseñado debe minimizar el tiempo y esfuerzo requeridos para realizar tareas, optimizando así la productividad del usuario.

"Facilidad de recordar el funcionamiento", explica que la facilidad de recordar el funcionamiento de un sistema se refiere a la capacidad del usuario para recordar cómo utilizarlo tras un periodo sin uso. Un diseño intuitivo y consistente ayuda a que los usuarios retomen su uso sin necesidad de reaprendizaje significativo, mejorando así la experiencia y eficiencia del sistema.

P.21 **Ejemplo de Usabilidad: Diseño Responsivo**

1. **Versión de escritorio (izquierda)**
    - Se despliega un **menú horizontal completo** con todas las categorías visibles (Rostro, Cuerpo y cabello, Ofertas, etc.).

- El usuario puede acceder directamente a cualquier sección sin necesidad de hacer clic extra.
- Se aprovecha el espacio amplio de la pantalla para mostrar información en un formato más distribuido.

2. **Versión móvil (derecha)**
   - El menú se **compacta en un ícono de hamburguesa (≡)**, ubicado en la parte superior izquierda o derecha, optimizando el espacio reducido de la pantalla.
   - El diseño se **adapta verticalmente**, colocando los elementos uno debajo del otro para una mejor legibilidad y navegación con el dedo.
   - Los **elementos destacados** (como las rebajas y promociones) se mantienen visibles, priorizando la información clave para el usuario móvil.

Beneficios de este enfoque

- **Consistencia:** el contenido es el mismo en ambas versiones, evitando pérdida de información.
- **Adaptabilidad:** la interfaz se ajusta al tamaño de pantalla (ordenador, tablet o móvil).
- **Usabilidad mejorada:** facilita la navegación al usuario, sin importar el dispositivo que use.
- **Experiencia centrada en el usuario:** garantiza que las promociones y botones importantes (ej. “Disfrutar”) estén siempre visibles.

👉 Este tipo de diseño se denomina **Responsive Web Design (RWD)**, y es una práctica fundamental en el desarrollo actual porque mejora la experiencia del usuario y la accesibilidad del sitio.

Ver diapositiva 22

P23. Calidad del software se refiere al grado en que un software cumple con los requisitos funcionales y no funcionales, asegurando su eficiencia, confiabilidad y facilidad de uso. Se evalúa mediante diferentes características clave, como:

Características de la calidad del software

1. **Funcionalidad** – El software debe cumplir con los requisitos especificados y responder correctamente a las necesidades del usuario.

2. **Fiabilidad** – Debe ser estable y funcionar sin fallos en diferentes condiciones de uso.
3. **Usabilidad** – Debe ser fácil de aprender, entender y operar.
4. **Eficiencia** – Optimiza el uso de recursos como tiempo de respuesta y memoria.
5. **Mantenibilidad** – Permite realizar modificaciones y actualizaciones de manera sencilla.
6. **Portabilidad** – Puede adaptarse a diferentes plataformas y entornos sin grandes cambios.

Factores que influyen en la calidad del software

- **Pruebas y validaciones** para detectar errores y mejorar el rendimiento.
- **Buenas prácticas de desarrollo**, como metodologías ágiles o modelos como CMMI e ISO 25010.
- **Retroalimentación de los usuarios** para mejorar la experiencia y resolver problemas detectados.

Un software de alta calidad reduce costos de mantenimiento, mejora la experiencia del usuario y aumenta la satisfacción general con el producto.

P24.Actividades de Aseguramiento de Calidad de Software

En nuestra labor de acompañamiento a los clientes, apoyamos en diferentes etapas del ciclo de vida del software para garantizar que los productos entregados sean confiables, seguros y cumplan con las necesidades del negocio. Algunas de las principales actividades son:

1. **Revisión a los elementos del plan de proyecto**
   Se analizan cronogramas, recursos, riesgos y entregables para asegurar que la planificación sea realista y esté alineada con los objetivos del negocio.
2. **Inspección y verificación de requerimientos de negocio y técnicos del sistema**
   Se valida que los requerimientos estén claros, completos, consistentes y trazables, evitando ambigüedades que afecten el desarrollo posterior.
3. **Análisis y validación de diseños**
   Se revisan los diagramas, arquitecturas y modelos de software para garantizar que respondan a los requerimientos y se ajusten a estándares de calidad y buenas prácticas.
4. **Técnicas de diseño de pruebas**
   Se definen los métodos y tipos de pruebas (unitarias, integración, regresión, aceptación) que permitirán comprobar el correcto funcionamiento del sistema.
5. **Inspecciones de código**
   Se revisa el código fuente para identificar defectos tempranamente, verificar el cumplimiento de estándares de programación y mejorar la mantenibilidad del software.

6. **Verificación del uso del sistema por medio de pruebas funcionales y no funcionales**
   Se validan funcionalidades frente a lo solicitado por el cliente y también aspectos como rendimiento, seguridad, usabilidad y compatibilidad.
7. **Control de calidad en el despliegue del sistema**
   Se supervisan las actividades de instalación, configuración y puesta en marcha para asegurar que el sistema funcione correctamente en los entornos productivos.
8. **Seguimiento y control a las actividades de soporte**
   Se verifican las respuestas a incidentes, tiempos de atención y soluciones aplicadas, garantizando la continuidad del servicio.
9. **Verificación y soporte al proceso de administración de cambios**
   Se controla que cualquier modificación al sistema esté documentada, probada y autorizada, reduciendo riesgos en la operación.
10. **Evaluación del estado de madurez de los procesos de control de calidad de software**
    Se mide el nivel de adopción de prácticas de calidad y se identifican áreas de mejora frente a modelos de referencia como CMMI o TMMi.
11. **Mejoramiento continuo del proceso de control de calidad de software**
    A partir de métricas, lecciones aprendidas y retroalimentación, se ajustan los procesos para elevar progresivamente los estándares de calidad.

P29. La **usabilidad** es una característica clave en la calidad del software que determina qué tan fácil y eficiente es para los usuarios interactuar con un sistema. Un software con buena usabilidad permite que los usuarios realicen tareas de manera intuitiva, rápida y con el menor número de errores posible.

> Una buena usabilidad requiere que el sitio sea fácil de navegar, tenga un diseño decente, sea coherente y útil para el visitante. El usuario deben disponer de autonomía, deben de tener el control del sitio.

P30. La usabilidad es clave en el éxito de cualquier software, ya que influye directamente en la productividad y la satisfacción del usuario.

Principales Características de la Usabilidad

1. **Facilidad de aprendizaje** – Qué tan rápido los usuarios pueden entender y comenzar a utilizar el sistema.
2. **Eficiencia de uso** – Permite realizar tareas con rapidez y sin esfuerzo innecesario.
3. **Memorabilidad** – Los usuarios pueden recordar cómo usar el sistema tras un periodo sin utilizarlo.

4. **P31.Prevención y manejo de errores** – Reduce la posibilidad de cometer errores y proporciona mensajes claros cuando ocurren.
5. **Satisfacción del usuario** – Proporciona una experiencia agradable y sin frustraciones.

Factores que mejoran la Usabilidad

- **Diseño intuitivo** con interfaces claras y bien organizadas.
- **Navegación sencilla** que guíe al usuario de manera natural.
- **Retroalimentación inmediata** para que los usuarios comprendan el resultado de sus acciones.
- **Accesibilidad** para que personas con diferentes capacidades puedan usar el sistema sin problemas.

La usabilidad es clave en el éxito de cualquier software, ya que influye directamente en la productividad y la satisfacción del usuario.

P32. Buenas prácticas para aplicaciones móviles

1. **Definir con simplicidad y claridad la estructura de los contenidos**
   La organización del contenido debe ser **intuitiva y jerárquica**, siguiendo el flujo natural de los procesos de negocio que soporta la aplicación. Esto facilita que el usuario identifique rápidamente dónde se encuentra la información y cómo acceder a ella, reduciendo la curva de aprendizaje.
2. **Establecer un diseño "limpio"**
   Una interfaz debe evitar la sobrecarga visual. Para ello se recomienda:
   - Usar una **tipografía legible** y consistente.
   - Incluir imágenes y gráficos que aporten valor, no que distraigan.
   - Mantener un **equilibrio entre espacio en blanco y contenido**, lo que ayuda a guiar la atención del usuario hacia lo importante.
3. **Ceder el control al usuario**
   La aplicación debe garantizar que el usuario tenga siempre **claridad y autonomía** sobre sus acciones. Esto implica:
   - Señalar en qué parte de la aplicación se encuentra.
   - Permitir navegar fácilmente hacia otras secciones.
   - Evitar bloqueos o pasos obligatorios innecesarios.
   - Ofrecer confirmaciones o mensajes claros cuando se ejecuta una acción importante (ej. eliminar, guardar, comprar).

P33.

1. **Facilitar la interacción**
   El sistema debe permitir que el usuario realice diferentes acciones de manera **fácil, rápida e intuitiva**. Esto significa reducir la cantidad de pasos innecesarios, ofrecer botones o enlaces claros, y mantener una lógica de navegación coherente. Cuando la interacción es sencilla, aumenta la satisfacción del usuario y se mejora la experiencia global.

2. **Simplificar y sintetizar**
   Bajo la premisa de que *"todo lo que no es necesario sobra"*, el diseño debe eliminar elementos superfluos que distraigan o confundan al usuario. La simplicidad se traduce en interfaces más limpias, con mensajes claros y operaciones directas, lo que facilita la comprensión y evita la sobrecarga cognitiva.

3. **Adaptar para todo tipo de dispositivos**
   Un diseño usable debe ser **responsivo**, es decir, ajustarse a distintos tamaños y características de dispositivos (computadores, tablets, móviles). De esta forma, se garantiza que el contenido y las funcionalidades estén disponibles de manera accesible sin importar desde dónde acceda el usuario.

👉 Estos tres principios trabajan en conjunto para lograr que el usuario se sienta cómodo, seguro y satisfecho al interactuar con un sistema digital.

P34. Leer las diapositivas

DESCARGA e INSTALA Balsamiq Mockups GRATIS

https://www.youtube.com/watch?v=cZd505X8PFQ&t=99s

Aprende a utilizar balsamiq Mockups paso a paso con ejemplos

https://www.youtube.com/watch?v=6vXjEd6BWgo&t=17s

Como DESCARGAR e INSTALAR Figma para PC

https://www.youtube.com/watch?v=GhVFlof4FdE&t=28s

Figma tutorial español - Como diseñar una pantalla de Login

https://www.youtube.com/watch?v=va_4klLW7R4