RESUMEN SESIÓN EN LÍNEA

Fecha de la Sesión:	09/09/2024
Competencia:	TÉCNICA
Tema Central:	CARACTERISTICAS TÉCNICAS DE LA INTERFAZ GRÁFICA
Instructor Responsable:	ILMER CUELLO

Desarrollo de la Sesión

- ✓ **COMPETENCIA: 220501095-** DISEÑAR LA SOLUCIÓN DE SOFTWARE DE ACUERDO CON PROCEDIMIENTOS Y REQUISITOS TÉCNICOS

- ✓ **RESULRADO DE APRENDIZAJE:** RAP 09 DETERMINAR LAS CARACTERÍSTICAS TÉCNICAS DE LA INTERFAZ GRÁFICA DEL SOFTWARE ADOPTANDO

Se inicia la sesión en línea a las 9:00 pm con los aprendices de las fichas: 2879674 Y 2879675, del programa ANÁLISIS Y DESARROLLO DE SOFTWARE. Posterior a la solicitud de aprobación para grabar la sesión en línea, se inicia explicando las evidencias:

- ✓ RAP09-GA5-220501095-AA1-EV04-CA Maquetación de la interfaz gráfica en HTML de la aplicación web y app móvil.

- ✓ RAP09-GA5-220501095-AA1-EV05-CA Mapa de navegación

- ✓ RAP09-GA5-220501095-AA1-EV06-CA Nociones de reglas de usabilidad y accesibilidad en aplicaciones independientes, móviles.

DOCUMENTO RESUMEN DE LA SESIÓN

ENLACE GRABACIÓN:

https://drive.google.com/file/d/1AJJP1uhlp5sWHBqNu0sxyoTITiOPGawJ/view?usp=drive_link

RAP09-GA5-220501095-AA1-EV04-CA Maquetación de la interfaz gráfica en HTML de la aplicación web y app móvil.

Los componentes técnicos clave que deben considerar al realizar la maquetación de la interfaz gráfica en HTML para aplicaciones web y móviles. La maquetación es el primer paso para estructurar visualmente una aplicación, por lo que es crucial seguir buenas prácticas desde el inicio.

1. Estructura Semántica del HTML

La base de cualquier maquetación es el uso adecuado de las etiquetas HTML. Deben asegurarse de que el código HTML sea semántico, es decir, que cada etiqueta tenga un propósito claro en cuanto a la estructura y el contenido de la página. Algunas etiquetas importantes incluyen:

- **<header>**: Define el encabezado o cabecera del sitio.
- **<nav>**: Para menús de navegación.
- **<main>**: Contiene el contenido principal de la página.
- **<section>**: Divide el contenido en secciones lógicas.
- **<article>**: Para artículos independientes dentro de la página.
- **<footer>**: Pie de página.

Una buena estructura semántica mejora la accesibilidad y el SEO, lo que es relevante tanto para aplicaciones web como móviles.

2. Diseño Responsivo (Responsive Design)

Hoy en día, las aplicaciones deben funcionar en una amplia gama de dispositivos, desde pantallas grandes de escritorio hasta móviles. Para ello, deben aplicar principios de diseño responsivo:

- **Media Queries**: Con CSS, pueden utilizar @media queries para ajustar el diseño según el ancho o el alto del dispositivo. Por ejemplo:

```
@media only screen and (max-width: 600px) {
  .container {
    flex-direction: column;
  }
}
```



Viewport Meta Tag: En aplicaciones móviles, deben agregar la siguiente etiqueta en el HTML para controlar el zoom y el tamaño de la pantalla.

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



Unidades Flexibles: Utilicen **rem**, **em**, **%** o **vh/vw** para unidades de tamaño flexibles en lugar de píxeles fijos, lo que facilita la adaptación del diseño a distintas pantallas.

3. Frameworks y Librerías CSS

Para acelerar el desarrollo, pueden usar frameworks CSS como **Bootstrap**, **Foundation** o **TailwindCSS**. Estos frameworks ofrecen componentes y clases predefinidas para crear un diseño limpio y responsivo sin tener que escribir código CSS desde cero.

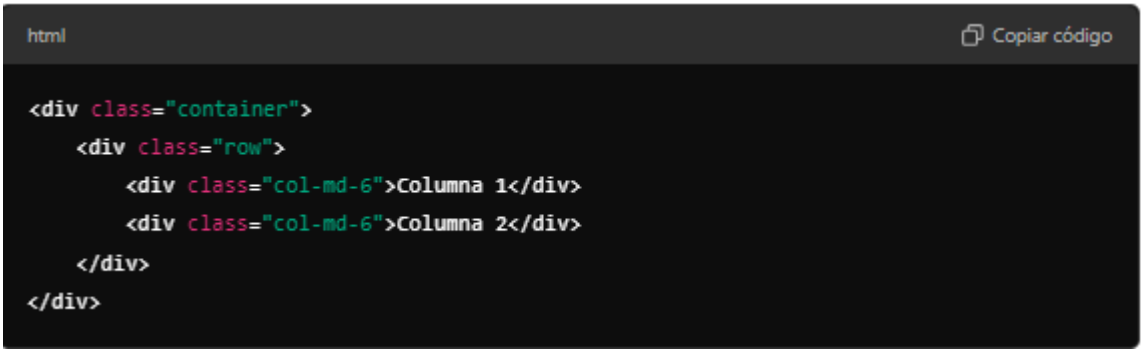
Por ejemplo, en **Bootstrap** podrían usar un sistema de cuadrícula (grid) como este:

```
<div class="container">
  <div class="row">
    <div class="col-md-6">Columna 1</div>
    <div class="col-md-6">Columna 2</div>
  </div>
</div>
```

3. Frameworks y Librerías CSS

Para acelerar el desarrollo, pueden usar frameworks CSS como **Bootstrap**, **Foundation** o **TailwindCSS**. Estos frameworks ofrecen componentes y clases predefinidas para crear un diseño limpio y responsivo sin tener que escribir código CSS desde cero.

Por ejemplo, en **Bootstrap** podrían usar un sistema de cuadrícula (grid) como este:



Esto permite un diseño que se ajusta automáticamente al tamaño del dispositivo.

4. Optimización de Imágenes y Gráficos

Las imágenes deben estar optimizadas para cargar rápidamente en dispositivos con diferentes resoluciones. Aquí algunos puntos importantes:

- **Imágenes en Formato WebP**: Este formato reduce el tamaño de las imágenes sin perder calidad visual, ideal para aplicaciones móviles.
- **Uso de SVG**: Los gráficos vectoriales SVG son ideales para íconos y logotipos porque se escalan sin perder calidad.
- **Lazy Loading**: Para mejorar el rendimiento, pueden aplicar carga diferida (lazy loading) de imágenes, especialmente en aplicaciones móviles.

<img src="imagen.jpg" loading="lazy" alt="Descripción de la imagen">

```html
<img src="imagen.jpg" loading="lazy" alt="Descripción de la imagen">
```

5. Componentes Interactivos con JavaScript

Aunque la maquetación se centra en HTML y CSS, deben tener en cuenta la integración de componentes interactivos mediante JavaScript. Para crear una experiencia fluida:

- **Menús Desplegables (Dropdowns)**, **Pestañas (Tabs)**, y **Carouseles** pueden ser implementados con frameworks como **jQuery** o las capacidades nativas de JavaScript.
- **Integración con APIs**: Las aplicaciones web móviles suelen interactuar con APIs. Por ejemplo, para cargar datos dinámicos en la interfaz, pueden usar **Fetch API** o **Axios**.

6. Ajuste de Fuentes y Tipografía

El tipo de fuente es crucial para la legibilidad, especialmente en aplicaciones móviles. Algunas recomendaciones:

- **Google Fonts**: Proveen una amplia gama de fuentes optimizadas para la web.
- **Tamaño relativo**: Utilicen tamaños relativos como em o rem para que el texto se ajuste bien en distintas pantallas.

```
body {
  font-size: 16px; /* Desktop */
}
@media only screen and (max-width: 600px) {
  body {
    font-size: 14px; /* Móvil */
  }
}
```

```css
body {
    font-size: 16px; /* Desktop */
}
@media only screen and (max-width: 600px) {
    body {
        font-size: 14px; /* Móvil */
    }
}
```

7. Pruebas en Múltiples Dispositivos
Una vez completada la maquetación, deben realizar pruebas exhaustivas en diferentes dispositivos y navegadores para asegurar que la aplicación se vea y funcione correctamente en todos los entornos. Pueden usar herramientas como **Google Chrome DevTools** para simular diferentes dispositivos móviles o servicios como **BrowserStack** para realizar pruebas más completas.

Estos son los puntos principales para considerar en la maquetación de la interfaz gráfica de aplicaciones web y móviles. Recuerden que la maquetación es solo el primer paso, luego deberán integrar funcionalidad e interactividad. Mantener un código modular y bien organizado desde el inicio facilitará futuras modificaciones y escalabilidad del proyecto.

Ejemplos de Maquetas Web
https://creately.com/diagram-community/examples-es/t/webmockups-es

RAP09-GA5-220501095-AA1-EV05-CA Mapa de navegación

Elaboración paso a paso de un mapa de navegación para una aplicación de software

Vamos a crear un **mapa de navegación** para una aplicación de software en desarrollo. El objetivo de esta actividad es representar visualmente, en forma de árbol jerárquico, cómo los usuarios navegarán a través de las distintas páginas y secciones de la aplicación. A lo largo de este proceso, organizaremos la jerarquía y distribución del contenido, de manera que sea fácil de entender y acceder para los usuarios.

Paso 1: Definir los Objetivos y Público de la Aplicación

Antes de comenzar a diseñar el mapa de navegación, es importante tener en claro:

- **El propósito de la aplicación**: ¿Qué solución ofrece? ¿Cuáles son los servicios o productos principales que los usuarios buscan?
- **El público objetivo**: ¿Qué tipo de usuarios usarán la aplicación? ¿Cuáles son sus necesidades o expectativas?

Esta información te ayudará a priorizar el contenido y organizar las páginas de manera lógica.

Paso 2: Hacer una Lista de Páginas o Secciones Principales

Haz una lista de las páginas principales que debe tener la aplicación. Estas son las primeras páginas que los usuarios verán y navegarán. Ejemplos comunes incluyen:

- **Página de inicio (Homepage)**
- **Sobre nosotros (About)**

- **Servicios o Productos (Services/Products)**
- **Contacto (Contact)**
- **Inicio de sesión/Registro (Login/Signup)**
- **Panel de usuario (User Dashboard)**

Piensa también en subpáginas dentro de cada sección. Por ejemplo, dentro de “Servicios”, podrías tener una página por cada servicio específico.

Paso 3: Establecer la Jerarquía

Organiza estas páginas de manera jerárquica. La **página de inicio** suele estar en la parte superior del árbol, con las secciones y subpáginas ramificándose hacia abajo. Pregúntate:

- ¿Qué páginas son más importantes?
- ¿Qué páginas están a un mismo nivel (es decir, están relacionadas entre sí)?
- ¿Qué subpáginas dependen de otras?

Aquí es donde puedes empezar a ver cómo se construirá tu **árbol jerárquico**.

Paso 4: Crear el Primer Boceto del Mapa de Navegación

Toma lápiz y papel, o una herramienta digital (como Lucidchart, Figma o una pizarra en línea), y comienza a dibujar tu mapa de navegación. Puedes hacerlo en forma de árbol jerárquico:

- **El nivel superior**: Representa la **página principal** (homepage).
- **El segundo nivel**: Representa las **secciones clave** del sitio, como "Productos", "Servicios", "Sobre nosotros", "Contacto".
- **El tercer nivel y siguientes**: Aquí se colocan las subpáginas, organizadas de forma que dependan de las secciones principales.

Por ejemplo:

markdown Copiar código

```
Inicio
├── Productos
|   ├── Producto 1
|   ├── Producto 2
|   └── Producto 3
├── Servicios
|   ├── Servicio 1
|   ├── Servicio 2
├── Sobre Nosotros
├── Contacto
└── Panel de Usuario
    ├── Perfil
    └── Configuración
```

Paso 5: Definir los Flujos de Navegación

Ahora debes **definir cómo los usuarios se moverán entre las páginas**. Para esto, puedes usar flechas o líneas que conecten una página con otra:

- **Navegación principal**: Cómo los usuarios se moverán entre las páginas de nivel superior (como Inicio, Productos, Servicios, etc.).
- **Navegación interna**: Cómo navegarán entre subpáginas dentro de una sección (por ejemplo, de "Producto 1" a "Producto 2").
- **Navegación transversal**: Si hay páginas que deben ser accesibles desde cualquier lugar (como el "Login" o el "Contacto"), indícalo claramente.

Paso 6: Optimizar la Experiencia del Usuario

Asegúrate de que la navegación sea lo más intuitiva posible. Algunas preguntas clave que debes hacerte:

- ¿Se puede acceder fácilmente a las páginas más importantes desde cualquier lugar del sitio?
- ¿La estructura es clara y no confunde al usuario?
- ¿El número de clics para llegar a cualquier página está limitado (preferiblemente no más de 3 clics)?
- ¿Existen accesos directos a áreas clave, como el panel de usuario o el carrito de compras (si aplica)?

Paso 7: Verificación y Pruebas

Una vez que tengas el mapa, revísalo para asegurarte de que la jerarquía sea coherente y de que no haya rutas de navegación innecesarias o confusas. Puedes pedir a compañeros o usuarios potenciales que lo revisen para obtener retroalimentación sobre la lógica de la navegación.

Paso 8: Implementación y Ajustes

Una vez aprobado, este mapa servirá como una guía para estructurar el código y los enlaces dentro de la aplicación. Al implementar el HTML y el sistema de rutas, asegúrate de respetar esta estructura. Recuerda que la maquetación inicial puede necesitar ajustes según la evolución del proyecto o la retroalimentación de los usuarios.

✓ RAP09-GA5-220501095-AA1-EV06-CA Nociones de reglas de usabilidad y accesibilidad en aplicaciones independientes, móviles.

Introducción a las Reglas de Usabilidad y Accesibilidad en Aplicaciones Móviles

Vamos a explorar las **reglas de usabilidad** que todo desarrollador debe tener en cuenta al crear aplicaciones, especialmente **móviles**, para garantizar que sean fáciles de usar, atractivas y accesibles para todos los usuarios. La **usabilidad** mide cuán efectivo, eficiente y satisfactorio es para los usuarios interactuar con una aplicación. La **accesibilidad** garantiza que las personas con diversas discapacidades puedan utilizar el software.

Paso a Paso para Comprender las Reglas de Usabilidad

Paso 1: Conocer los Principios de Usabilidad

Hay cinco principios básicos que guían la usabilidad de cualquier software:

1. **Facilidad de aprendizaje**: ¿Qué tan rápido puede un nuevo usuario aprender a usar la aplicación?
2. **Eficiencia**: ¿Qué tan rápido y fácilmente puede un usuario realizar tareas comunes?
3. **Memorabilidad**: ¿El usuario puede recordar cómo utilizar la aplicación después de un tiempo sin usarla?
4. **Prevención de errores**: ¿La aplicación ayuda al usuario a evitar errores o recuperarse fácilmente de ellos?
5. **Satisfacción**: ¿El usuario disfruta la experiencia de uso?

Estos principios te ayudarán a diseñar aplicaciones centradas en el usuario, que no solo sean funcionales, sino también placenteras.

Paso 2: Estudio de la Accesibilidad

La accesibilidad es un componente clave de la usabilidad. Significa diseñar aplicaciones que sean utilizables por personas con diferentes habilidades, incluyendo aquellos con discapacidades visuales, auditivas, cognitivas o motoras.

Aspectos clave a considerar:

- **Compatibilidad con lectores de pantalla** para personas con discapacidades visuales.
- **Subtítulos y descripciones de audio** para usuarios con discapacidades auditivas.
- **Contraste de colores adecuado** para personas con daltonismo.
- **Tamaño de los elementos táctiles** para facilitar la navegación para aquellos con dificultades motoras.

Paso 3: Investigación y Conocimiento del Usuario

Antes de comenzar a diseñar una aplicación, es crucial entender a los usuarios que la utilizarán. Pregúntate:

- ¿Quiénes son mis usuarios objetivo?
- ¿Qué tareas necesitan completar en la aplicación?
- ¿Cómo usan sus dispositivos móviles (dedos grandes, pantalla pequeña, etc.)?

Para esto, puedes realizar encuestas, entrevistas o estudios de usuarios. Los datos que obtengas guiarán decisiones clave, como el diseño de la interfaz y las funcionalidades.

Paso 4: Simplicidad en el Diseño

Uno de los principios clave de usabilidad es la simplicidad. Una aplicación debe ser:

- **Fácil de entender**: Los elementos de la interfaz deben ser intuitivos y estar organizados de manera lógica.
- **Minimalista**: Evita sobrecargar la pantalla con demasiada información o funcionalidades. Cada elemento debe tener un propósito claro.
- **Consistente**: El diseño debe ser coherente en todas las pantallas de la aplicación. Los botones, menús y acciones deben seguir patrones familiares para que los usuarios no se confundan.

Paso 5: Crear Interfaces Táctiles Intuitivas

En el caso de las aplicaciones móviles, la **interacción táctil** es fundamental. Las interfaces deben ser:

- **Responsivas**: Los botones y enlaces deben responder rápidamente al toque.
- **Tamaños adecuados**: Los elementos táctiles deben ser lo suficientemente grandes para que los usuarios puedan presionarlos fácilmente con sus dedos, sin errores.
- **Gestos intuitivos**: Usa gestos estándar como deslizamientos, pellizcos para acercar, etc. No intentes reinventar la rueda con interacciones complejas que los usuarios no entienden.

Paso 6: Optimización para Diferentes Tamaños de Pantalla

Es importante que tu aplicación funcione correctamente en **diferentes tamaños de pantallas**. Asegúrate de que:

- **El contenido se ajuste** adecuadamente en pantallas pequeñas (smartphones) y más grandes (tabletas).
- **Los textos y elementos gráficos** sean escalables, de modo que no se vuelvan demasiado pequeños en dispositivos móviles.
- **La navegación sea clara**: Usa menús desplegables o hamburguesa para ahorrar espacio en pantallas pequeñas.

Paso 7: Pruebas de Usabilidad

Una vez que has diseñado la aplicación, es crucial realizar **pruebas de usabilidad** para asegurarte de que los usuarios puedan interactuar con la aplicación de manera eficiente y satisfactoria. Realiza sesiones de prueba con usuarios reales y observa:

- **Cómo navegan por la aplicación**.
- **Cuánto tardan en completar tareas clave**.
- **Dónde encuentran dificultades o confusión**.

Con los comentarios obtenidos, puedes ajustar el diseño y mejorar la experiencia general.

Paso 8: Mantener la Eficiencia y Velocidad

En las aplicaciones móviles, la **velocidad** es fundamental. Los usuarios tienden a abandonar aplicaciones lentas o que tardan en cargar. Para garantizar una buena usabilidad:

- Optimiza el **rendimiento** y los tiempos de carga.

- Minimiza el **uso de datos** móviles.
- Asegúrate de que la aplicación funcione sin problemas en conexiones de red más lentas.

Paso 9: Feedback Visual y de Retroalimentación

Siempre que un usuario interactúe con un elemento de la aplicación, debe haber una respuesta clara. Algunos ejemplos de **feedback visual**:

- Botones que cambian de color cuando se tocan.
- Indicadores de progreso cuando una página o acción está cargando.
- Mensajes de confirmación cuando una tarea ha sido completada con éxito (como el envío de un formulario).

Esto ayuda a los usuarios a entender qué está ocurriendo en la aplicación y reduce la posibilidad de confusión.

Paso 10: Implementar Regresividad y Prevención de Errores

Permite que el usuario **deshaga acciones** si comete un error. Por ejemplo:

- Un botón de **"Atrás"** visible y claro.
- Confirmaciones antes de realizar una acción irreversible, como eliminar una cuenta o archivo.

Reducir errores mejora significativamente la usabilidad, haciendo que el uso de la aplicación sea más fluido y menos frustrante.

Resumen de Puntos Clave para Crear Aplicaciones Móviles Usables

1. **Simplicidad**: Diseñar una interfaz clara, minimalista y fácil de usar.
2. **Accesibilidad**: Incluir características que ayuden a personas con discapacidades a usar la aplicación.
3. **Consistencia**: Mantener coherencia en los elementos visuales y de interacción.
4. **Pruebas de Usabilidad**: Validar el diseño con usuarios reales antes del lanzamiento.
5. **Velocidad**: Optimizar el rendimiento para asegurar que la aplicación sea rápida y eficiente.
6. **Retroalimentación Visual**: Proporcionar respuestas claras a las acciones del usuario.

Estos conceptos son fundamentales para que las aplicaciones móviles no solo sean funcionales, sino también **placenteras y accesibles** para el mayor número de usuarios posible.

Recursos: https://acortar.link/ciQUlM

Les invitamos a visitar nuestro nuevo espacio blog, que será de gran ayuda para apoyar y complementar todos los procesos del ciclo de desarrollo de software:

1. Planeación y diseño de proyecto
2. Diseño de software
3. Desarrollo de Software
4. Gestión del ciclo de vida del software

Blog: https://toursvirtuales.co/software/