P7. 9.1 Usabilidad

La usabilidad se refiere a la facilidad con la que los usuarios pueden interactuar con un sistema, aplicación o producto digital para alcanzar sus objetivos de manera eficiente, efectiva y satisfactoria. Un sistema con alta usabilidad debe cumplir con los siguientes principios:

- **Facilidad de aprendizaje:** Los nuevos usuarios deben poder entender y utilizar el sistema con rapidez.
- **Eficiencia:** Los usuarios experimentados deben poder realizar tareas de manera rápida y fluida.
- **Memorabilidad:** Los usuarios ocasionales deben recordar fácilmente cómo utilizar el sistema sin necesidad de reaprendizaje.
- **Prevención de errores:** El diseño debe minimizar la posibilidad de cometer errores y, en caso de que ocurran, permitir su rápida corrección.
- **Satisfacción del usuario:** La experiencia debe ser agradable y no generar frustración.

9.2 Accesibilidad

La accesibilidad se refiere a la capacidad de un sistema para ser utilizado por personas con diferentes habilidades y condiciones, incluyendo aquellas con discapacidades visuales, auditivas, motoras o cognitivas. Para garantizar un sistema accesible, se deben considerar los siguientes aspectos:

- **Cumplimiento de estándares:** Implementación de normas como las **WCAG (Web Content Accessibility Guidelines)** para garantizar la inclusión de todos los usuarios.
- **Compatibilidad con tecnologías de asistencia:** Soporte para lectores de pantalla, navegación mediante teclado, comandos de voz, entre otros.
- **Contraste y legibilidad:** Uso adecuado de colores, tipografías y tamaños de fuente para facilitar la lectura.
- **Opciones de personalización:** Permitir ajustes en la interfaz, como cambio de colores, ampliación de texto y activación de subtítulos.
- **Navegación clara y sencilla:** Diseño intuitivo que facilite la interacción sin barreras innecesarias.

Ambos conceptos son fundamentales para garantizar una experiencia de usuario inclusiva y eficiente, permitiendo que todas las personas, independientemente de sus capacidades, puedan interactuar con el sistema de manera efectiva.

P8. La **interfaz de usuario (UI)** en el contexto de **usabilidad** y **accesibilidad** en el diseño de una página web se describe como el conjunto de elementos visuales, estructuras y controles que permiten a los usuarios interactuar de manera efectiva, eficiente y satisfactoria con un sitio web, independientemente de sus capacidades o limitaciones. A continuación te explico cómo se describe desde ambos enfoques:

✅ **1. En términos de *Usabilidad***

La interfaz debe permitir que el usuario logre sus objetivos de forma rápida, sencilla y sin errores.

Características clave:

- **Claridad**: Los elementos deben ser comprensibles (botones, menús, formularios).
- **Consistencia**: Uso coherente de colores, íconos y tipografía en toda la página.
- **Navegación intuitiva**: Menús visibles, rutas claras, jerarquía visual clara.
- **Feedback inmediato**: Confirmaciones visuales o auditivas ante las acciones del usuario (por ejemplo: mensajes de error o éxito).
- **Minimización de esfuerzo**: El usuario debe hacer la menor cantidad de clics posible para alcanzar su objetivo.

Ejemplo:

Un botón de "Agregar al carrito" debe ser claramente visible, con un diseño coherente con el resto del sitio, y al hacer clic debe mostrar un mensaje de confirmación.

✅ **2. En términos de *Accesibilidad***

Se refiere a que la interfaz sea usable para **personas con discapacidades** visuales, auditivas, motrices o cognitivas.

Características clave:

- **Compatibilidad con lectores de pantalla** (uso correcto de etiquetas HTML como aria-label, alt en imágenes).
- **Contraste de colores adecuado** para usuarios con baja visión o daltonismo.
- **Navegación por teclado** (todas las funciones accesibles sin necesidad de usar un mouse).
- **Textos ampliables** sin pérdida de funcionalidad o diseño.
- **Lenguaje claro y comprensible**.

Ejemplo:

Un campo de formulario con una etiqueta visible y asociada correctamente (<label for="email">Correo electrónico</label>) permite que usuarios con lectores de pantalla lo comprendan sin dificultad.

✍️ **En resumen:**

La interfaz de usuario se describe como un sistema visual y funcional diseñado para facilitar la interacción entre el usuario y la página web, garantizando que sea comprensible, eficiente y accesible para todos, incluyendo personas con discapacidades.

Leer La diapositiva P9, P10 y P11

P12. Esta actividad se orienta a realizar las prácticas que permiten elaborar la interfaz gráfica y el mapa de navegación de una aplicación teniendo en cuenta las reglas de usabilidad y accesibilidad.

Plan de trabajo del aprendiz

Actividad: Elaborar la interfaz gráfica y el mapa de navegación de una aplicación con base en requerimientos, aplicando reglas de usabilidad y accesibilidad.

---

1. Revisión y análisis de requerimientos

Tareas:

- Leer y comprender la lista de requerimientos funcionales y no funcionales.
- Identificar las pantallas que necesita el sistema.
- Hacer un listado preliminar de las vistas (ejemplo: Inicio, Login, Registro, Inventarios, Reportes, Contacto).

---

2. Definición del tipo de aplicación

Tareas:

- Determinar si la aplicación será web o independiente (de escritorio/móvil).
- Justificar la elección (ejemplo: “Se escoge aplicación web porque se requiere acceso desde distintos dispositivos”).

---

3. Adaptación a la imagen institucional del cliente

Tareas:

- Investigar la identidad visual del cliente (colores, logotipo, tipografía, estilo gráfico).
- Definir la paleta de colores y tipografías a usar.

---

4. Diseño de prototipos de interfaz gráfica

Tareas:

- Crear prototipos de baja o media fidelidad para cada pantalla identificada.
- Incluir botones, formularios, menús y secciones necesarias.
- Aplicar buenas prácticas de usabilidad:
    - Botones y menús claros.
    - Jerarquía visual (títulos, subtítulos, texto).
    - Evitar sobrecarga de información.
- Aplicar criterios de accesibilidad:
    - Contraste de colores adecuado.

- Texto alternativo para imágenes.
- Navegación con teclado.

Entregable: Prototipos en herramienta digital (Figma, Adobe XD, Balsamiq, Mockflow) o en HTML/CSS (si se pide algo más práctico).

---

5. Diseño del mapa de navegación

Tareas:

- Crear un diagrama que muestre la relación entre las pantallas.
- Ejemplo: Inicio → Login → Panel principal → Módulo de inventarios → Detalle del producto.
- Puede hacerse en: Draw.io, Lucidchart, Miro, PowerPoint o incluso en papel digitalizado.

Entregable: Mapa de navegación en formato imagen o PDF.

---

6. Validación y retroalimentación de prototipos

Tareas:

- Presentar prototipos y mapa de navegación al instructor o compañeros.
- Recolectar comentarios (claridad, accesibilidad, estética, usabilidad).
- Ajustar los diseños con base en la retroalimentación.

P13. Lo que tendría que realizar el aprendiz, en esta evidencia, es **complementar la lista de requerimientos con una sección de documentación más formal**, aplicando los elementos del estándar **IEEE 830** y relacionándolos con los prototipos diseñados.

En concreto, debería:

1. **Crear una nueva sección en el documento** donde se describa cada prototipo.
   Para cada prototipo debe incluir:
   - **Perspectiva del producto:** Cómo encaja el prototipo dentro del sistema completo, cuál es su propósito.
   - **Funciones del producto:** Qué funcionalidades específicas representa el prototipo (ejemplo: login, registro, gestión de inventario).
   - **Características de los usuarios:** Qué tipo de usuarios van a interactuar con ese prototipo (administrador, cliente, mesero, proveedor, etc.).
   - **Restricciones:** Limitaciones técnicas, de diseño, de seguridad o de recursos que deben cumplirse.
   - **Requisitos funcionales:** Acciones que debe realizar el prototipo (ejemplo: validar credenciales, mostrar listado, generar reporte).
   - **Requisitos no funcionales:** Aspectos de rendimiento, usabilidad, accesibilidad, confiabilidad o seguridad asociados al prototipo.

2. **Relacionar cada requisito con su prototipo**.
    - Para cada requisito de la lista ya elaborada, el aprendiz debe indicar en qué prototipo se representa y cómo se evidencia en el diseño.
    - Esto puede hacerse con una tabla de trazabilidad donde se muestre el requisito → prototipo.
3. **Ejemplo de estructura que podría seguir el aprendiz:**

Prototipo: Pantalla de Inicio de Sesión

- Perspectiva del producto: Permite a los usuarios autenticarse para acceder al sistema.
- Funciones del producto: Validar credenciales, mostrar mensajes de error.
- Características de los usuarios: Administradores, empleados y clientes registrados.
- Restricciones: Debe usar cifrado SSL, tiempos de respuesta menores a 3 segundos.
- Requisitos funcionales:
    - RF01: El sistema debe permitir ingresar usuario y contraseña.
    - RF02: El sistema debe validar las credenciales.
- Requisitos no funcionales:
    - RNF01: El sistema debe responder en menos de 3 segundos.
    - RNF02: La interfaz debe ser accesible en dispositivos móviles.

¿Quieres que te prepare un **formato en tabla** que el aprendiz pueda usar directamente para documentar cada prototipo con base en IEEE 830?

P14. Formato de documentación de prototipos (IEEE 830)

Elemento	Descripción del prototipo
Nombre del prototipo	(Ejemplo: Pantalla de inicio de sesión)
Perspectiva del producto	(Explica el rol del prototipo dentro del sistema, su objetivo principal)
Funciones del producto	(Lista de funcionalidades principales que representa el prototipo)
Características de los usuarios	(Tipo de usuario que interactúa con el prototipo: cliente, administrador, empleado, proveedor, etc.)
Restricciones	(Limitaciones técnicas, de diseño, de seguridad o de compatibilidad)

Elemento	Descripción del prototipo
Requisitos funcionales	(Lista de requisitos que se evidencian en el prototipo, por ejemplo: RF01, RF02, etc.)
Requisitos no funcionales	(Condiciones de calidad: usabilidad, rendimiento, accesibilidad, seguridad, etc.)
Relación requisito-prototipo	(Tabla o listado que muestre qué requisito de la lista general se representa en este prototipo)

P15. Ejemplo aplicado

Elemento	Descripción del prototipo
Nombre del prototipo	Pantalla de inicio de sesión
Perspectiva del producto	Este prototipo permite a los usuarios autenticarse en el sistema para acceder a sus funcionalidades.
Funciones del producto	- Ingresar credenciales. - Validar usuario y contraseña. - Mostrar mensajes de error.
Características de los usuarios	- Administradores. - Empleados. - Clientes registrados.
Restricciones	- La contraseña debe estar encriptada. - Tiempo de respuesta máximo de 3 segundos.
Requisitos funcionales	- RF01: El sistema debe permitir ingresar usuario y contraseña. - RF02: El sistema debe validar credenciales.
Requisitos no funcionales	- RNF01: El sistema debe ser accesible desde dispositivos móviles. - RNF02: Debe responder en menos de 3 segundos.
Relación requisito-prototipo	RF01 → Campo usuario y contraseña. RF02 → Botón "Iniciar sesión". RNF01 → Diseño adaptable. RNF02 → Optimización en el backend.

P21. ¿Qué es HTML5?

HTML5 (**HyperText Markup Language 5**) es la última versión del lenguaje de marcado utilizado para estructurar y presentar contenido en la web. Es una evolución de HTML4 y XHTML, e introduce nuevas características que mejoran la experiencia de desarrollo y usuario en aplicaciones web modernas.

HyperText, es un texto que enlaza con otros contenidos, que pueden ser otro texto u otro archivo.

Markup, marca o etiqueta.

Languaje, es decir, tiene sus normas, tiene su estructura y una serie de convenciones que nos sirven para definirla.

Podemos definir HTML5 como un estándar que sirve para definir la estructura y el contenido de una página Web. Es un lenguaje de etiquetas, utilizado para la estructuración y la presentación de contenido en los sitios web.

P22. Cuando escribimos código en HTML, escribimos etiquetas HTML. Todas las etiquetas HTML están hechas con un número de **partes específicas**, incluyendo:

El carácter **"menor que" (<)** es un símbolo especial en **HTML** porque se usa para definir etiquetas.

En **HTML**, la **palabra o carácter que determina qué etiqueta se está escribiendo** se llama **nombre de la etiqueta**.

El **nombre de la etiqueta** es la palabra clave que identifica el propósito de un elemento en la estructura del documento HTML.

Los **atributos HTML** proporcionan información adicional sobre una etiqueta y se escriben en la forma:

nombre_atributo="valor"

El carácter **"mayor que" (>)** en **HTML** se usa para cerrar etiquetas y definir jerarquías dentro del código.

P23. En la actualidad existen 142 etiquetas HTML disponibles que permiten la creación de varios elementos. **Una etiqueta** le indica al navegador dónde empieza y dónde termina cada elemento, mientras que un **atributo** describe las características.

Partes de una etiqueta HTML

Una etiqueta HTML está compuesta por las siguientes partes:

1. **Etiqueta de apertura** (<etiqueta>)
    - Indica el inicio de un elemento en el documento HTML.
    - Ejemplo: abre un párrafo.
2. **Contenido**
    - Es el texto o elementos que están dentro de la etiqueta.
    - Ejemplo: En Hola, mundo! , el contenido es "Hola, mundo!".
3. **Etiqueta de cierre** (</etiqueta>)
    - Indica el final del elemento.

- Siempre lleva una barra inclinada (/).
- Ejemplo: </p> cierra un párrafo.

4. **Atributos**
   - Proporcionan información adicional sobre la etiqueta.
   - Se escriben dentro de la etiqueta de apertura.
   - Ejemplo: Haz clic aquí (el atributo href define el enlace).

La imagen muestra la estructura básica de una **etiqueta HTML** y desglosa sus partes principales:

1. **Etiqueta de apertura (<h1 class="principal">)**:
   - **Nombre de etiqueta**: h1 → Indica que es un encabezado de nivel 1.
   - **Atributo**: class → Un atributo que permite asignar una clase para aplicar estilos con CSS o comportamientos con JavaScript.
   - **Valor**: "principal" → El valor asignado al atributo class.
2. **Contenido**:
   - Texto entre la etiqueta de apertura y la de cierre: Título destacado.
3. **Etiqueta de cierre (</h1>)**:
   - Indica el fin del contenido del encabezado.
4. **Etiquetas de apertura y cierre**:
   - Se señala que toda etiqueta HTML suele tener una etiqueta de apertura (<h1>) y una de cierre (</h1>) si no es una etiqueta auto-cerrada.

P24 y P25 Leer la diapositiva

P26. La imagen muestra una comparación entre el uso de y el uso de **etiquetas semánticas** en HTML5 para estructurar una página web.

Análisis de la imagen

1. **Lado izquierdo (con)**
   - Usa con id o class para definir secciones como cabecera, menu, articulo, pie, etc.
   - No hay significado semántico, solo una estructura basada en contenedores genéricos.
2. **Lado derecho (con etiquetas semánticas de HTML5)**
   - Usa <header> en lugar de .
   - Usa <nav> en lugar de .

- Usa <section> para agrupar contenido.
- Usa <article> en lugar de .
- Usa <footer> en lugar de .

Ventajas del uso de etiquetas semánticas

✅ **Mejor accesibilidad**: Los navegadores y lectores de pantalla entienden mejor el contenido.
✅ **SEO optimizado**: Los motores de búsqueda comprenden mejor la estructura de la página.
✅ **Código más claro**: Hace que el HTML sea más fácil de leer y mantener.

P27.

Vamos a mostrar la **estructura de una etiqueta HTML** y sus componentes principales:

1. **Etiqueta de apertura:**
   - Se inicia con < y termina con >.
   - Contiene el nombre de la etiqueta.
   - Ejemplo: <html>, <body>.
2. **Atributos:**
   - Modifican el comportamiento de una etiqueta.
   - Se escriben dentro de la etiqueta de apertura en formato nombre="valor".
   - Ejemplo: <html lang="es">.
3. **Contenido:**
   - Es el texto, imágenes o elementos dentro de una etiqueta.
   - Ejemplo: <h1>Hola mundo</h1>.
4. **Etiqueta de cierre:**
   - Se escribe con </ seguido del nombre de la etiqueta y >.
   - Indica el final del elemento.
   - Ejemplo: </html>, </body>.

Estos son los fundamentos esenciales de HTML para estructurar correctamente una página web.

P28.

Toda la página web debe ir encerrada entre las etiquetas
<html> ..... </html>
Estas etiquetas delimitan el principio y el final de la página. Dentro de esta etiqueta hay otras dos etiquetas que son imprescindibles. Estas son:

<head> ..... </head>
<body> ..... </body>
Estas dos etiquetas deben ir siempre en este orden, y una detrás de otra, y ambas incluidas dentro de la etiqueta <html> ..... </html>.

Etiquetas principales de una página HTML

1. **<html> ... </html>**
    - Es la etiqueta raíz que envuelve todo el contenido de la página.
2. **<head> ... </head>**
    - Contiene metadatos como el título de la página, enlaces a hojas de estilo (CSS), scripts (JavaScript), y configuraciones importantes.
3. **<body> ... </body>**
    - Contiene el contenido visible de la página, como textos, imágenes, enlaces, formularios, etc.

P29 al P34 Leer diapositiva

P35. Análisis de lo que se realiza en la imagen:

1. **Etiqueta <html> (línea 1)**
    - Indica el inicio del documento HTML.
2. **Sección <head> (líneas 2 a 5)**
    - **<title>**: Define el título que aparecerá en la pestaña del navegador:
      → *"esto aparecerá en la pestaña del navegador"*
    - **<link rel="stylesheet" href="estilo.css" type="text/css">**:
      Vincula una hoja de estilos externa llamada estilo.css para dar formato visual a la página.
3. **Sección <body> (línea 7 en adelante)**
    - Contiene todo el contenido visible de la página.

🔷 **Contenido visible del sitio web:**

🔷 **<header> (líneas 8–11):**

- **<h1>**: Título principal que da la bienvenida:
  → *"Bienvenido a mi sitio web"*

- **<nav>**: Barra de navegación con dos enlaces:
    - link1.html (texto: "link1")
    - link2.html (texto: "link2")

🔷 **<section> (líneas 12–21):**

- Contiene dos <article>:
    - El primero dice: *"Esto es un artículo"*
    - El segundo dice: *"Esto es otro artículo"*

🔷 **<footer> (líneas 23–25):**

- Muestra el pie de página con el texto:
  → *"este es el pie de mi página"*

✅ **Resumen breve:**

En la imagen se está creando una página web básica con HTML5. Se estructura con encabezado, navegación, contenido dividido en artículos y un pie de página. Además, se enlaza una hoja de estilos externa para aplicar diseño visual.

P36. La imagen muestra un ejemplo de cómo se integran **HTML y CSS** para estructurar y dar estilo a una página web. Vamos a explicarlo paso a paso:

🔷 **Archivo HTML (ejemplo_2_1.html) – Estructura del contenido**

```
<!DOCTYPE html>
<html lang="es">
<head>
 <meta charset="utf-8" />
 <title>Ejemplo 2.1. Hojas de estilo</title>
 <link rel="stylesheet" type="text/css" href="css/ejemplo_2_1.css" />
</head>
<body>
 <h1>Introducción</h1>
 <p>Excel es un programa del tipo Hoja de Cálculo.</p>
 <p class="resaltado">Microsoft es la propietaria de Excel.</p>
</body>
</html>
```

✳️ **Elementos clave:**

- <link rel="stylesheet"...>: conecta el archivo CSS externo.
- <h1>: título principal.
- <p>: párrafos de texto.
- class="resaltado": clase usada para aplicar estilo especial al segundo párrafo.

🔷 **Archivo CSS (ejemplo_2_1.css) – Estilos visuales**

```
h1 {
 text-align: center;
}
p {
 font-family: arial;
 font-size: 16px;
}
.resaltado {
 color: blue;
}
```

🎨 **Estilos aplicados:**

- h1: se centra el texto.
- p: todos los párrafos usan fuente Arial y tamaño de 16px.
- .resaltado: cualquier elemento con esta clase será azul.

🔶 **Resultado en el navegador:**

Se muestra:

- Un título centrado **"Introducción"** en negrita.
- Un primer párrafo normal: “Excel es un programa del tipo Hoja de Cálculo.”
- Un segundo párrafo en **color azul**: “Microsoft es la propietaria de Excel.”

P37. Leer la diapositiva

P38. Las **reglas o comandos CSS**, que son las instrucciones que los navegadores utilizan

para dar estilo a las páginas web.

Estructura de una regla CSS

Una regla CSS está compuesta por:
✅ **Selector**: Indica el elemento HTML al que se aplicará el estilo.
✅ **Declaración**: Contiene una o más propiedades con sus respectivos valores.

La declaración tiene la siguiente sintaxis:

selector { propiedad: valor; propiedad: valor; }

Ejemplo de una regla CSS

```
p {
  color: #ff0000; /* Hace el texto rojo */
  font-family: Arial; /* Usa la fuente Arial */
}
```

📌 **Explicación**:

- El **selector** p indica que la regla se aplicará a los párrafos (<p>).
- La **propiedad** color cambia el color del texto.
- La **propiedad** font-family define la fuente del texto.

Esta estructura básica permite personalizar el diseño de las páginas web de manera eficiente

P39. Leer la diapositiva

P40. La imagen muestra un ejemplo práctico de **diseño web con HTML y CSS**, donde se están aplicando estilos visuales a tres bloques ( ) mediante identificadores únicos (id). A la derecha, se ve una vista previa del resultado en un navegador.

🔍 **¿Qué se está realizando en la imagen?**

🧱 **En la parte HTML (superior izquierda):**

Se crean tres contenedores con diferentes contenidos:

html

CopiarEditar

```
<div id="first">Soy el primer div.</div>
<div id="second">Soy el segundo div.</div>
<div id="third">Soy el tercer div.</div>
```

Cada tiene un id único: first, second, y third.

🎨 **En la parte CSS (inferior izquierda):**

Se definen estilos visuales diferentes para cada div, además de estilos generales aplicables a todos los div.

🔴 **#first:**

- Fondo rojo (red)
- Ancho automático (auto)
- Fuente Arial

🔵 **#second:**

- Fondo azul verdoso (#19A5BD)
- Ancho fijo de 200px
- Fuente Times New Roman

🟠 **#third:**

- Fondo naranja claro (rgb(253, 143, 89))
- Ancho del 50% del contenedor padre
- Fuente Courier

📐 **div (para todos):**

- Margen externo de 10px
- Relleno interno de 5px

📺 **En la parte derecha (vista previa):**

- Se visualizan los tres div con sus colores de fondo, anchos y fuentes aplicados correctamente.
- Se observa separación entre ellos gracias al margin.
- El texto dentro de cada div tiene espacio interno gracias al padding.

✅ **Resumen explicativo:**

En la imagen se está diseñando la apariencia de tres elementos usando CSS. Cada uno tiene un estilo personalizado en cuanto a color de fondo, ancho y tipo de fuente. Además, se aplican márgenes y rellenos a todos los div, lo que mejora la presentación visual. Esta práctica demuestra cómo separar el contenido (HTML) de la presentación (CSS) para lograr páginas web organizadas y atractivas.

INSTALAR Y CONFIGURAR VISUAL STUDIO CODE EN WINDOWS

https://www.youtube.com/watch?v=X_Z7d04x9-E

CREA UN FORMULARIO DE INCIO DE SESION CON HTML Y CSS

https://www.youtube.com/watch?v=XPrs9nqiYOc

Cómo Crear una Página Web COMPLETA con HTML y CSS desde cero

https://www.youtube.com/watch?v=8-RC-Q7Wtzc

CREAR UN FORMULARIO DE INICIO DE SESIÓN CON HTML Y CSS

Index.html

```
<!DOCTYPE html>
<html lang="es">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Documento</title>
   <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="formulario">
   <h1>Inicio de Sesión</h1>
   <form action="post" method="post">
      <div class="username">
         <input type="text" required>
         <label for="nombre de usuario">Nombre de Usuario</label>
      </div>
      <div class="username">
         <input type="password" required>
         <label for="contrasena">Contraseña</label>
      </div>
      <div class="recordar">Olvido su contraseña</div>
      <input type="submit" value="Iniciar">
      <div class="registrarse">
         Quiero hacer el <a href="#">registro</a>
      </div>
   </form>
  </div>
</body>
</html>
```

Style.css

```
body {
   margin: 0;
   padding: 0;
   font-family: montserrat;
   background: linear-gradient(150deg,#6c3483,#58d68d);
```

```
    height: 100vh;
}

.formulario{
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    width: 400px;
    background: white;
    border-radius: 10px;
}

.formulario h1{
    text-align: center;
    padding: 0 0 20px 0;
    border-bottom: 1px solid silver;
}

.formulario form{
    padding: 0 40px;
    box-sizing: border-box;
}

form .username{
    position: relative;
    border-bottom: 2px solid #adadad;
    margin: 30px 0;
}

.username input{
    width: 100%;
    padding: 0 5px;
    height: 40px;
    font-size: 16px;
    border: none;
    background: none;
    outline: none;
}
 .username label{
    position: absolute;
    top: 50%;
    left: 5px;
    color: #adadad;
    transform: translateY(-50%);
    font-size: 16px;
    pointer-events: none;
    transition: .5s;
 }
.username span::before{
    content: " ";
    position: absolute;
```

```
    top: 40px;
    left: 0;
    width: 100%;
    height: 2px;
    background: #6c3483;
    transition: .5s;
}
.username input:focus ~ label,
.username input:focus ~ label{
    top: -5px;
    color: #6c3483;
}
.username input:focus ~ span::before,
.username input:focus ~ span::before{
    width: 100%;
}
.recordar{
    margin: -5px 0 20px 5px;
    color: blue;
    cursor: pointer;
}
.recordar:hover{
    text-decoration: underline;
}
input[type="submit"]{
    width: 100%;
    height: 50px;
    border: 1px solid;
    background: #6c3483;
    border-radius: 25px;
    font-size: 18px;
    color: white;
    cursor: pointer;
    outline: none;
}
input[type="submit"]:hover{
    border-color: purple;
    transition: .5s;
}
.registrarse{
    margin: 30px 0;
    text-align: center;
    font-size: 16px;
    color: black;
}
.registrarse a{
    color: blue;
    text-decoration: underline;
}
.registrarse a:hover{
    text-decoration: underline;
}
```