**Resumen completo de lo que hemos trabajado:**

**1. Sistema de autenticación (Login)**

* **Creación del archivo login.php:**
  + Se implementó un sistema de autenticación que utiliza sesiones para validar usuarios.
  + Se configuró la verificación segura de contraseñas mediante password\_verify.
* **Gestión de errores:**
  + Mensajes claros de error para usuario o contraseña incorrectos.
* **Redirección:**
  + Usuarios autenticados son redirigidos al index.php.

**2. Control de acceso**

* **Protección de páginas:**
  + Implementamos validaciones en index.php y otras páginas para verificar si el usuario está autenticado.
  + Redirección automática a login.php para usuarios no logueados.
* **Control de visibilidad:**
  + Usuarios no autenticados ven un menú limitado.
  + Usuarios autenticados tienen acceso a todas las opciones del menú.

**3. Diseño del menú lateral (sidebar)**

* **Integración con AdminLTE:**
  + El menú lateral se diseñó utilizando la plantilla **AdminLTE**, asegurando un diseño moderno y funcional.
* **Opciones del menú creadas:**
  + Inicio
  + Registrar Compra
  + Registrar Proponente
  + Agregar Documento
  + Estados de Compra:
    - Adjudicados
    - Cancelados
    - Abiertas
* **Dinamismo:**
  + Opciones del menú visibles según el estado del usuario (logueado o no).

**4. Formulario de Registro de Compra**

* **Estructura:**
  + Un formulario diseñado en HTML que sigue los estándares de diseño de AdminLTE.
* **Campos básicos:**
  + Incluye todos los campos necesarios para registrar una compra (como nombre, descripción, monto, entre otros).
* **Diseño responsivo:**
  + Compatible con dispositivos móviles y escritorio.

**5. Diseño y funcionalidad del index.php**

* **Integración con el menú lateral:**
  + Se añadieron las opciones del menú con enlaces funcionales.
* **Protección mediante sesiones:**
  + El archivo verifica si el usuario está logueado antes de cargar las opciones completas.
* **Inicio visual:**
  + Un diseño limpio y profesional utilizando **AdminLTE** para la página principal.

**6. Mejoras en diseño y usabilidad**

* **Plantilla AdminLTE:**
  + Se configuraron estilos y funcionalidades específicas (como botones, barras de navegación, íconos, etc.).
* **Font Awesome:**
  + Se usaron íconos para mejorar la presentación visual del menú.
* **Colores y branding:**
  + Personalización de colores para alinearse con los requerimientos del proyecto.

**7. Lógica adicional**

* **Gestión de redirección post-login:**
  + Una vez logueado, el usuario es redirigido a la página principal con acceso completo.
* **Sesión segura:**
  + Las variables de sesión se utilizan para mantener al usuario autenticado durante su navegación.
* **Cerrar sesión:**
  + Se implementó un botón para cerrar sesión, destruyendo las variables de sesión y redirigiendo al login.

**Resumen de la creación y configuración de todos los archivos del sistema**

**1. login.php**

* **Propósito:**
  + Permitir el inicio de sesión de usuarios mediante validación contra la base de datos.
* **Funcionalidades principales:**
  + Validación del usuario y la contraseña con password\_verify.
  + Inicio de sesión utilizando sesiones PHP.
  + Redirección a index.php si el login es exitoso.
  + Manejo de errores para usuarios o contraseñas incorrectas.
* **Interfaz de usuario:**
  + Un formulario simple con campos para usuario y contraseña, diseñado con estilos CSS personalizados.

**2. logout.php**

* **Propósito:**
  + Gestionar el cierre de sesión.
* **Funcionalidades principales:**
  + Destruir la sesión actual del usuario.
  + Redirigir al usuario de vuelta al login.php.

**3. index.php**

* **Propósito:**
  + Página principal del sistema, mostrando el menú lateral y opciones de navegación.
* **Funcionalidades principales:**
  + Comprobar si el usuario está autenticado antes de mostrar el contenido completo.
  + Mostrar un menú limitado para usuarios no autenticados.
  + Integración del diseño con **AdminLTE** para proporcionar una interfaz profesional.
* **Menú lateral (sidebar):**
  + Opciones creadas:
    - Inicio.
    - Registrar Compra.
    - Registrar Proponente.
    - Agregar Documento.
    - Adjudicados, Cancelados, Abiertas (estados de compra).

**4. formulario\_compra.php**

* **Propósito:**
  + Permitir el registro de nuevas compras en el sistema.
* **Estructura:**
  + Formulario con campos como:
    - Nombre de la compra.
    - Descripción.
    - Monto.
    - Proveedor.
    - Fecha.
* **Diseño:**
  + Totalmente responsivo, siguiendo la plantilla de **AdminLTE**.

**5. registrar\_proponente.php**

* **Propósito:**
  + Página para registrar nuevos proponentes.
* **Estructura:**
  + Formulario con campos como:
    - Nombre del proponente.
    - Empresa.
    - Contacto.
* **Diseño:**
  + Integración con AdminLTE para mantener consistencia visual.

**6. agregar\_documento.php**

* **Propósito:**
  + Permitir a los usuarios cargar documentos relacionados con las compras.
* **Estructura:**
  + Formulario con campos como:
    - Título del documento.
    - Archivo adjunto.
* **Funcionalidad adicional:**
  + Validación para el tipo y tamaño de archivo permitido.

**7. adjudicados.php, cancelados.php, abiertas.php**

* **Propósito:**
  + Páginas para gestionar el estado de las compras.
* **Estructura:**
  + Listas dinámicas que muestran las compras filtradas por su estado:
    - **Adjudicados:** Compras asignadas a un proveedor.
    - **Cancelados:** Compras anuladas.
    - **Abiertas:** Compras que aún están en proceso.

**8. conexion.php**

* **Propósito:**
  + Establecer la conexión con la base de datos.
* **Estructura:**
  + Configuración de las credenciales de la base de datos.
  + Manejo de errores si la conexión falla.

**9. ver\_registros.php**

* **Propósito:**
  + Mostrar todas las compras registradas en el sistema en una tabla.
* **Estructura:**
  + Tabla con columnas como:
    - ID de la compra.
    - Nombre.
    - Descripción.
    - Estado.
    - Opciones (editar o eliminar).
* **Diseño:**
  + Tabla estilizada con **AdminLTE**.

**10. Archivos de diseño (CSS y librerías externas)**

* **AdminLTE:**
  + Plantilla principal para todo el diseño del sistema.
* **Font Awesome:**
  + Uso de íconos para mejorar la experiencia visual.
* **Bootstrap y otros estilos personalizados:**
  + Adaptación responsiva y mejoras en el diseño.

**11. Archivos adicionales**

* **cerrar.php:**
  + Archivo simple para cerrar la sesión del usuario.
* **Archivos de plantilla:**
  + Se incluyeron recursos como estilos, scripts y configuraciones de AdminLTE para facilitar el diseño y la interacción.
* **Integración con AdminLTE**:
  + Se utilizó la estructura y clases CSS de la plantilla AdminLTE para dar al formulario un diseño profesional y moderno.
* **Seguridad contra inyección SQL**:
  + Se empleó prepared statements (bind\_param) para evitar inyecciones SQL en la consulta de login.
* **Mensajes de error estilizados**:
  + Los errores (usuario no encontrado o contraseña incorrecta) se muestran con un diseño de alerta (alert-danger) de AdminLTE.
* **Estilización de campos de entrada**:
  + Los campos de usuario y contraseña tienen iconos de Font Awesome para un mejor aspecto visual.
  + Se usaron input-group y input-group-append para mostrar los iconos dentro del diseño de los campos.
* **Botón estilizado**:
  + Se agregó un botón de inicio de sesión estilizado con las clases btn btn-primary btn-block para un diseño atractivo y funcional.
* **Diseño responsivo**:
  + El formulario está centrado y es compatible con dispositivos móviles gracias al sistema de diseño fluido de AdminLTE.

22/11/2024

**1. Funcionalidades Principales**

El sistema está diseñado para gestionar y visualizar registros relacionados con compras en un entorno intuitivo, seguro y accesible. Las funcionalidades clave incluyen:

* **Listado de Registros**:
  + Visualización paginada de registros provenientes de la base de datos.
  + Cada registro muestra información clave como número de compra, descripción, fecha de publicación y estado.
* **Búsqueda Dinámica**:
  + Un campo de búsqueda permite filtrar registros con base en palabras clave ingresadas por el usuario.
  + Los resultados de búsqueda se integran con la paginación, conservando la estructura de navegación.
* **Visualización de Detalles**:
  + Cada registro incluye un botón de acción que permite al usuario visualizar los detalles específicos en una nueva vista.
* **Autenticación y Control de Sesiones**:
  + Uso de variables de sesión ($\_SESSION) para garantizar que solo usuarios autenticados accedan a funciones específicas como registrar compras o proponentes.

**2. Tecnologías Utilizadas**

El proyecto utiliza herramientas y tecnologías ampliamente conocidas para garantizar un desarrollo robusto y un diseño moderno:

* **Backend**:
  + Lenguaje PHP para lógica del servidor y manejo de bases de datos.
  + Conexión a base de datos MySQL para almacenar y gestionar los registros.
* **Frontend**:
  + Estructuración en HTML5.
  + Estilos y diseño responsivo utilizando:
    - **Bootstrap**: Para botones, tablas, paginación y diseño general.
    - **AdminLTE**: Plantilla que proporciona un diseño profesional para paneles de administración.
    - **Font Awesome**: Para íconos en menús y botones.
* **Gestión de Bases de Datos**:
  + Consultas SQL para obtención y manipulación de registros.
  + Paginación y búsqueda implementadas mediante LIMIT, OFFSET y LIKE.

**3. Mejoras Aplicadas**

A lo largo del desarrollo se han aplicado las siguientes optimizaciones:

* **Paginación Mejorada**:
  + Botones estilizados usando clases Bootstrap (pagination, page-item, page-link).
  + Resaltado dinámico de la página activa.
  + Deshabilitación de botones "anterior" y "siguiente" si el usuario se encuentra al inicio o al final de la lista.
* **Formulario de Búsqueda**:
  + Campo persistente que retiene el valor ingresado después de realizar la búsqueda.
  + Integración con paginación, garantizando que el filtro aplicado permanezca al navegar entre páginas.
* **Diseño Responsivo**:
  + Ajuste de tablas para adaptarse a pantallas pequeñas.
  + Uso de clases de diseño flexible (Bootstrap Grid System).
  + Botones redondeados con efectos de sombra y transiciones al interactuar.
* **Separación del Código**:
  + Fragmentación en módulos (archivos PHP separados) para mejorar legibilidad y mantenimiento:
    - Conexión a la base de datos.
    - Lógica de autenticación y control de sesión.
    - Lógica para visualización y búsqueda.

**4. Seguridad y Validaciones**

Se han incorporado mecanismos de seguridad para proteger los datos y evitar posibles ataques:

* **Validaciones en Entradas del Usuario**:
  + Escapado de datos (htmlspecialchars) para prevenir inyecciones de código y ataques XSS.
  + Validación en el lado del servidor para los parámetros de búsqueda y paginación.
* **Control de Sesiones**:
  + Autenticación con variables de sesión para funciones restringidas.
  + Validación constante de que el usuario está logueado antes de mostrar opciones sensibles.

**5. Pendientes o Propuestas de Mejora**

El sistema puede beneficiarse de futuras actualizaciones:

1. **Seguridad Mejorada**:
   * Implementar hash de contraseñas con password\_hash y verificar con password\_verify.
   * Usar consultas preparadas (PDO o mysqli\_prepare) para proteger aún más las consultas SQL.
2. **Experiencia del Usuario**:
   * Reemplazar la paginación tradicional por carga asíncrona (AJAX) para evitar recargas completas de la página.
   * Mejora en los estilos de tablas y botones con sombras más dinámicas y transiciones avanzadas.
3. **Rendimiento**:
   * Optimización de consultas SQL, especialmente en tablas con muchos registros.
   * Indexación en la base de datos para acelerar búsquedas y filtrados.
4. **Documentación y Pruebas**:
   * Crear una guía de usuario para explicar cómo operar el sistema.
   * Realizar pruebas de estrés en la base de datos y pruebas de usabilidad con usuarios finales.

**6. Próximos Pasos**

* **Personalización de Reportes**:
  + Generar reportes PDF con librerías como TCPDF o DOMPDF para exportar datos clave.
* **Módulos Adicionales**:
  + Un módulo para gestión de usuarios (roles y permisos).
  + Un dashboard estadístico con gráficos que resuma la actividad del portal.
* **Preparación para Producción**:
  + Migrar el sistema a un servidor seguro.
  + Configuración de HTTPS y protección contra accesos no autorizados.

25/11/2024

El resumen de los cambios realizados es el siguiente:

1. **Corrección del conteo de registros**: Se ajustó la consulta para que, cuando haya un término de búsqueda (searchTerm), se cuenten únicamente los registros que coinciden con ese término. Si no hay búsqueda, se cuentan todos los registros.
2. **Paginación**: Se mejoró la lógica de paginación, calculando correctamente el número total de páginas en función de los registros encontrados (ya sea con o sin filtro de búsqueda).
3. **Formulario de búsqueda**: El formulario ahora funciona correctamente para enviar el término de búsqueda a través de POST o GET, asegurando que el término se mantenga al cambiar de página.
4. **Visualización de registros**: Los registros encontrados se muestran en una tabla, y cada uno incluye botones para editar y generar un reporte en PDF.
5. **Paginación en la interfaz**: Se agregó la paginación en la interfaz, permitiendo navegar entre las páginas de resultados, manteniendo el término de búsqueda seleccionado en las diferentes páginas.

 **Añadir un botón para subir archivo:**

* En la lista de compras (donde se muestran los registros), se ha agregado un nuevo botón con el ícono de **subir archivo**.
* El botón redirige al usuario a la página subir\_doc.php, pasando el id\_pcompra como parámetro en la URL para saber qué compra está asociada al archivo que se va a subir.

 **Modificar la tabla de registros de compras:**

* En la columna de **Acciones** de la tabla, se agregó el siguiente código para el botón de subir archivo:
*  Este botón tiene el ícono fa-upload de Font Awesome y está estilizado con las clases de Bootstrap.

 **Crear la página subir\_doc.php:**

* La página subir\_doc.php ahora recibe el id\_pcompra desde la URL (pasado por el botón) y muestra un formulario para que el usuario suba un archivo PDF.
* El formulario incluye:
  + Un campo para ingresar el nombre del documento.
  + Un campo para seleccionar el archivo PDF a subir.
* También se añadió un botón para **volver** a la lista de compras.

 **Lógica de subida de archivos:**

* En subir\_doc.php, se incluye la lógica para recibir el archivo, almacenarlo en el servidor (en una carpeta uploads/), y asociarlo con el registro de compra mediante el id\_pcompra.

 **Actualización del código en PHP para insertar nuevos campos**:

* Se adaptó el archivo formulario\_compra.php para incluir los nuevos campos requeridos:
  + **Partida Presupuestaria**.
  + **Modalidad de Adjudicación**.
  + **Provincia de Entrega**.
  + **Usuario Registrado**.
* Se agregó el campo **Usuario Registrado** para capturar automáticamente al usuario que está logueado en el sistema, utilizando la variable de sesión $\_SESSION['usuario'].

 **Unificación del manejo de la base de datos**:

* Se utilizó exclusivamente **PDO** para realizar las operaciones con la base de datos, eliminando mezclas con mysqli que podrían causar conflictos.

 **Inserción y redirección tras el guardado**:

* Los datos capturados del formulario ahora se almacenan en la tabla wp\_portalcompra, incluyendo los nuevos campos.
* Una vez insertados, el sistema redirige automáticamente al formulario de documentos (form/index.php) con el ID de la compra creada.

 **Mejora en la validación de usuario**:

* Se verificó que solo los usuarios autenticados puedan interactuar con la página. Si no están logueados, son redirigidos al login.

 **Código organizado y funcional**:

* Se corrigieron errores en la lógica de inserción y organización del código para que sea más legible, eficiente y consistente.

 **Modificación del campo "Fecha de Presentación"**  
Cambiamos el tipo del campo de date a text para permitir que el usuario escriba la fecha manualmente. Se añadió un atributo placeholder con el texto **"Ingrese la fecha de presentación"**.

 **Solución al mensaje del navegador sobre autocompletado de tarjetas de crédito**  
Identificamos que el mensaje aparece porque:

* El formulario no está utilizando HTTPS.
* Cambiar el tipo de campo puede confundir al navegador.  
  Para resolverlo, sugerimos:
* Especificar autocomplete="off" en el campo o formulario.
* Usar el tipo date para evitar la interpretación incorrecta del campo.
* Aclarar el contexto del formulario con nombres descriptivos para los campos.

 **Recomendaciones de mejora**

* Configurar un certificado SSL para habilitar HTTPS y evitar problemas de seguridad y compatibilidad.
* Revisar el propósito del campo para usar el tipo más apropiado (text o date).

26/11/2024

 **Centrar el Título y los Logos:**  
Ajustamos la posición del título en el encabezado del documento para que quedara centrado entre los dos logos. Se utilizaron las coordenadas adecuadas para mantener el equilibrio visual.

 **Excluir el Campo ID:**  
Modificamos el código para omitir la impresión del campo ID al generar el contenido del reporte.

 **Firma del Responsable:**  
Agregamos un espacio al final del documento con el texto *"Firma del Responsable:"* acompañado de una línea horizontal para permitir que el responsable firme el documento.

 **Footer Personalizado:**  
Creamos un footer que contiene la nota específica:

 Se centró el footer y se ajustó para que no tuviera márgenes adicionales innecesarios.

 **Eliminar la Línea del Footer:**  
Modificamos la configuración de TCPDF para que la línea predeterminada del footer fuera invisible, configurando su color en blanco (255, 255, 255).

 **Optimización del Margen:**  
Ajustamos los márgenes del footer y otros elementos para un diseño más limpio y alineado.