

Proyecto integrador primero semestre



Integrantes CodeTitans:

Leandro Agustin Gonzalez

1. Descripción del Proyecto:

- **Nombre del proyecto:** Base de Datos de Clientes.
- **Objetivo:** El proyecto consiste en un sistema simple para gestionar una base de datos de clientes. Permite agregar, eliminar, consultar y ordenar registros de clientes, incluyendo su documento, nombre, saldo y estado de actividad.
- **Lenguaje:** PSeint.

2. Descripción del Algoritmo Principal:

- **Proceso principal:** BaseDeDatos.
- **Pasos:**
 - **Inicio:** Se muestra el logo del programa mediante `mostrarLogo()`.
 - **Definir la cantidad de registros:** Se llama al subalgoritmo `cantidadDeRegistros()`, que solicita al usuario la cantidad de registros que tendrá la base de datos (hasta 10).
 - **Inicializar los arreglos:** Se declaran los arreglos `documento`, `nombres`, `saldo`, y `activo` para almacenar los datos de los clientes.
 - **Mostrar el menú principal:** Se llama al subalgoritmo `menuPrincipal()`, el cual se encarga de mostrar el menú de opciones al usuario y de gestionar la ejecución de las diferentes funciones del sistema.
 - **Fin:** El programa finaliza.

3. Subalgoritmos y Funciones:

Aquí se describe cada uno de los subalgoritmos del código:

3.1. Menú Principal:

- **Subalgoritmo:** `menuPrincipal`
- **Descripción:** Muestra el menú principal del programa al usuario. Ofrece las opciones de consultar, ordenar, eliminar, ingresar, ver información sobre los autores y salir del sistema.
- **Parámetros:**
 - `documento`: Arreglo que almacena los documentos de los clientes.
 - `nombres`: Arreglo que almacena los nombres de los clientes.

- saldo: Arreglo que almacena el saldo de los clientes.
- activo: Arreglo que almacena el estado de actividad de los clientes.
- cantRegistros: Cantidad de registros de la base de datos.

3.2. Consultar Registros:

- **Subalgoritmo:** consultar
- **Descripción:** Permite al usuario consultar registros de clientes por documento o listar todos los registros.
- **Parámetros:** Los mismos que menuPrincipal.
- **Funcionamiento:**
 - Presenta las opciones de consulta por documento, lista de todos los registros o salir.
 - Si el usuario selecciona "Consultar por documento", se le pide ingresar el documento y se busca en la base de datos.
 - Si el documento se encuentra, se muestra la información del registro.
 - Si el documento no se encuentra, se le ofrece al usuario la opción de registrar el cliente o salir.
 - Si el usuario selecciona "Listar todos los registros", se llama al subalgoritmo listar para mostrar la lista de registros.
 - Si el usuario selecciona "Salir", se regresa al menú principal.

3.3. Ordenar Registros:

- **Subalgoritmo:** ordenar
- **Descripción:** Ordena los registros de la base de datos por documento, de menor a mayor o de mayor a menor, según la elección del usuario.
- **Parámetros:** Los mismos que menuPrincipal.
- **Funcionamiento:**
 - Llama al subalgoritmo disponibilidad para determinar la cantidad de registros válidos.
 - Si hay al menos dos registros, presenta las opciones de ordenar de menor a mayor o de mayor a menor.

- Llama al subalgoritmo ordenados para ordenar los registros según la opción elegida.
- Si hay un solo registro, muestra un mensaje indicando que no hay suficientes registros para ordenar.
- Si no hay registros, muestra un mensaje indicando que la base de datos está vacía.

3.4. Eliminar Registros:

- **Subalgoritmo:** eliminar
- **Descripción:** Permite al usuario eliminar un registro de la base de datos.
- **Parámetros:** Los mismos que menuPrincipal.
- **Funcionamiento:**
 - Llama al subalgoritmo disponibilidad para determinar la cantidad de registros válidos.
 - Si no hay registros, se ofrece al usuario la opción de ingresar un nuevo registro o regresar al menú principal.
 - Si hay registros, se muestra la lista de registros y se le pide al usuario que ingrese el número del registro que desea eliminar.
 - Si el número ingresado es válido, se solicita confirmación para eliminar el registro.
 - Si se confirma la eliminación, se desplazan los datos del registro eliminado hacia arriba en los arreglos, se actualiza la cantidad de registros válidos y se muestra una lista de los registros restantes.
 - Si se cancela la eliminación, se regresa al menú de eliminación.

3.5. Ingresar Registros:

- **Subalgoritmo:** ingresar
- **Descripción:** Permite al usuario ingresar un nuevo registro de cliente a la base de datos.
- **Parámetros:** Los mismos que menuPrincipal.
- **Funcionamiento:**
 - Llama al subalgoritmo disponibilidad para determinar si hay espacio disponible en la base de datos.

- Si la base de datos está llena, se ofrece al usuario la opción de eliminar un registro o regresar al menú principal.
- Si hay espacio disponible, se solicitan los datos del cliente (documento, nombre, saldo y estado de actividad) al usuario.
- Los datos se validan para asegurar que sean correctos y que no existan registros duplicados (se utilizan los subalgoritmos validador y ordenados).
- Si los datos son válidos, se almacenan en los arreglos correspondientes.
- Se ordenan los registros por documento de menor a mayor.
- Se ofrece al usuario la opción de ingresar otro registro o regresar al menú principal.

3.6. Subalgoritmos de Utilidad:

- **Subalgoritmo:** disponibilidad
- **Descripción:** Cuenta la cantidad de registros válidos en la base de datos, considerando que los arreglos pueden tener elementos vacíos.
- **Parámetros:**
 - documento: Arreglo que almacena los documentos de los clientes.
 - cantRegistros: Cantidad de registros de la base de datos.
- **Retorno:** La cantidad de registros válidos.
- **Subalgoritmo:** validador
- **Descripción:** Valida si un nuevo registro, con un valor específico (documento o nombre), ya existe en la base de datos.
- **Parámetros:**
 - documento: Arreglo que almacena los documentos de los clientes.
 - nombres: Arreglo que almacena los nombres de los clientes.
 - cantRegistros: Cantidad de registros de la base de datos.
 - regValidar: El valor a validar (documento o nombre).
 - vectorValidar: Indica si se debe validar el documento (1) o el nombre (2).

- **Retorno:** Verdadero si el registro ya existe, Falso si no existe.
- **Subalgoritmo:** ordenados
- **Descripción:** Ordena los arreglos de datos (documento, nombres, saldo y activo) según el orden solicitado (menor a mayor o mayor a menor) y la cantidad de registros válidos.
- **Parámetros:**
 - documento, nombres, saldo, activo: Los arreglos de datos.
 - cantRegistros: Cantidad de registros de la base de datos.
 - nOrden: Indica el tipo de orden: 1 para menor a mayor, 2 para mayor a menor.

3.7. Subalgoritmos de Menú:

- **Subalgoritmo:** menuOrdenar, menuEliminar, menuConsultar, menuIngresar
- **Descripción:** Estos subalgoritmos muestran el logo del programa y el nombre de la sección del menú actual (ordenar, eliminar, consultar, ingresar).

3.8. Subalgoritmo de Visualización:

- **Subalgoritmo:** listar
- **Descripción:** Muestra la lista de todos los registros de la base de datos, incluyendo el número del registro, el documento, el nombre, el saldo y el estado de actividad.
- **Parámetros:**
 - documento, nombres, saldo, activo: Los arreglos de datos.
 - cantRegistros: Cantidad de registros de la base de datos.

3.9. Subalgoritmo para mostrar el logo:

- **Subalgoritmo:** mostrarLogo
- **Descripción:** Se encarga de mostrar el logo del programa al inicio, generando un efecto de movimiento hacia arriba.

4. Consideraciones:

- **Manejo de errores:** El código implementa algunos mecanismos de validación y manejo de errores, como la comprobación de que el documento sea un número y la confirmación de la eliminación de un registro. Sin embargo, se pueden mejorar las validaciones para manejar otros posibles errores (por ejemplo, que el usuario ingrese un nombre que no sea solo letras).
- **Eficiencia:** Se puede evaluar la eficiencia del código, especialmente la función de ordenamiento, y considerar alternativas más eficientes si es necesario.
- **Modularidad:** El código está bastante modularizado, con diferentes subalgoritmos para cada función. Se puede seguir mejorando la modularidad para facilitar la comprensión y el mantenimiento del código.