UNIVERSIDAD TECNOLOGICA DE SANTIAGO UTESA



Nombre:

Osvaldo Castillo

Matrícula:

1-20-1706

Materia:

Taller de Programacion I

Profesor:

RAUL ALEXIS TORIBIO SILVERIO

Grupo:

001

Asignación:  
Proyecto 2 (Control de inventario de Herramientas)

**DESCRIPCION DEL PROBLEMA**

En mi trabajo el uso de las herramientas es muy importante para el desarrollo del proyecto, pero no todos los proyectos requieren el uso de algunas herramientas por eso se tiene un número limitado de estas al tener un número limitados de herramientas esto genera una problemática que es que algunas veces la persona encargada de organizar las brigadas no sabe en qué proyecto se dejó o uso una herramienta. Con este proyecto busco realizar una herramienta que permita saber a qué proyecto se le designó la utilización de uno o varias escaleras.

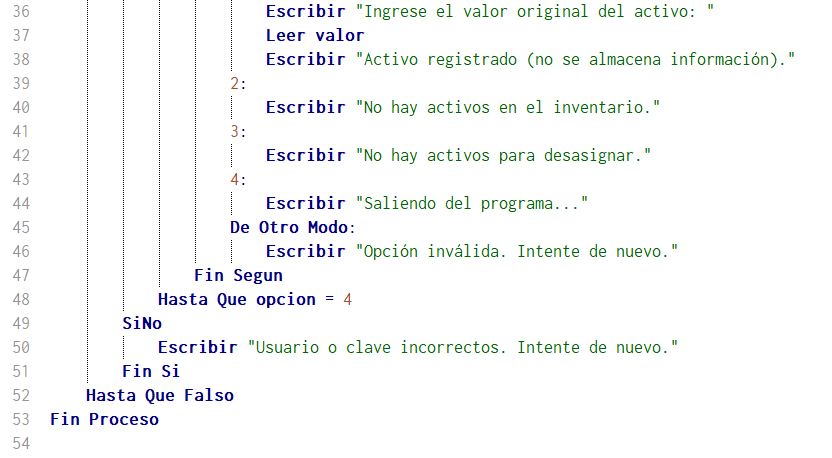
**OBJETIVO DEL PROYECTO**

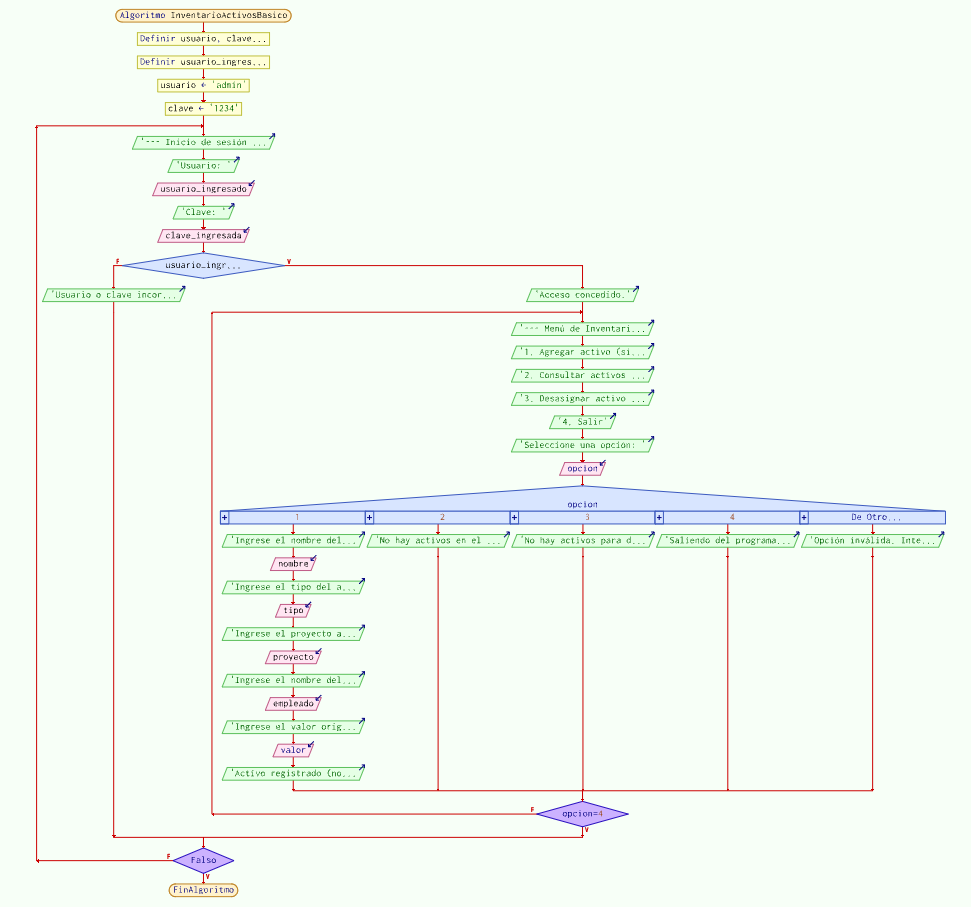
Con este proyecto busco tener un espacio donde asignatura el uso de herramientas por proyecto para así saber en qué lugar se dejaron o usaron las herramientas y no perderlas. Al implementar la herramienta de registro de uso de herramientas se busca saber dónde y cuándo se dejó una escalera en un proyecto para así evitar pérdidas de herramientas.

**RESUMEN TECNICO**

Este código es un programa que gestiona un inventario de activos. Permite agregar nuevos activos, consultar los existentes, y desasignar los que ya no se necesiten. Cada activo tiene información como su nombre, tipo, proyecto, empleado asignado, valor original, vida útil y año de compra. Además, calcula la depreciación anual de los activos dividiendo su valor original entre su vida útil.

**DISENO ESQUEMATICO**

****

****

**CODIGO FUENTE**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_ACTIVOS 100

#define MAX\_USUARIOS 10

typedef struct {

    char nombre[50];

    char tipo[20];

    char proyecto[50];

    char empleado[50];

    float valor\_original;

    int vida\_util;

    int anio\_compra;

} Activo;

Activo inventario[MAX\_ACTIVOS];

int total\_activos = 0;

char usuario[50] = "admin";

char clave[50] = "1234";

void guardarCredenciales() {

    FILE \*archivo = fopen("credenciales.txt", "w");

    if (archivo == NULL) {

        printf("Error al guardar las credenciales.\n");

        return;

    }

    fprintf(archivo, "%s\n%s\n", usuario, clave);

    fclose(archivo);

}

void cargarCredenciales() {

    FILE \*archivo = fopen("credenciales.txt", "r");

    if (archivo != NULL) {

        fscanf(archivo, "%s", usuario);

        fscanf(archivo, "%s", clave);

        fclose(archivo);

    } else {

        printf("No se encontraron credenciales guardadas. Usando valores predeterminados.\n");

    }

}

void guardarDatos() {

    FILE \*archivo = fopen("inventario.txt", "w");

    if (archivo == NULL) {

        printf("Error al guardar los datos.\n");

        return;

    }

    fprintf(archivo, "%d\n", total\_activos);

    for (int i = 0; i < total\_activos; i++) {

        fprintf(archivo, "%s %s %s %s %.2f %d %d\n", inventario[i].nombre, inventario[i].tipo,

                inventario[i].proyecto, inventario[i].empleado, inventario[i].valor\_original,

                inventario[i].vida\_util, inventario[i].anio\_compra);

    }

    fclose(archivo);

}

void cargarDatos() {

    FILE \*archivo = fopen("inventario.txt", "r");

    if (archivo == NULL) {

        printf("No se encontraron datos guardados.\n");

        return;

    }

    fscanf(archivo, "%d", &total\_activos);

    for (int i = 0; i < total\_activos; i++) {

        fscanf(archivo, "%s %s %s %s %f %d %d", inventario[i].nombre, inventario[i].tipo,

               inventario[i].proyecto, inventario[i].empleado, &inventario[i].valor\_original,

               &inventario[i].vida\_util, &inventario[i].anio\_compra);

    }

    fclose(archivo);

}

void cambiarCredenciales() {

    char clave\_actual[50];

    printf("Ingrese la clave actual: ");

    scanf("%s", clave\_actual);

    if (strcmp(clave\_actual, clave) == 0) {

        printf("Ingrese el nuevo usuario: ");

        scanf("%s", usuario);

        printf("Ingrese la nueva clave: ");

        scanf("%s", clave);

        guardarCredenciales();

        printf("Usuario y clave cambiados con éxito.\n");

    } else {

        printf("Clave incorrecta.\n");

    }

}

void agregarActivo() {

    if (total\_activos >= MAX\_ACTIVOS) {

        printf("No se pueden agregar más activos.\n");

        return;

    }

    Activo nuevo;

    getchar(); // Consumir el carácter de nueva línea pendiente

    printf("Ingrese el nombre del activo: ");

    fgets(nuevo.nombre, sizeof(nuevo.nombre), stdin);

    nuevo.nombre[strcspn(nuevo.nombre, "\n")] = '\0'; // Eliminar el salto de línea

    printf("Ingrese el tipo de activo: ");

    fgets(nuevo.tipo, sizeof(nuevo.tipo), stdin);

    nuevo.tipo[strcspn(nuevo.tipo, "\n")] = '\0';

    printf("Ingrese el proyecto asignado: ");

    fgets(nuevo.proyecto, sizeof(nuevo.proyecto), stdin);

    nuevo.proyecto[strcspn(nuevo.proyecto, "\n")] = '\0';

    printf("Ingrese el nombre del empleado al que se le asigna el activo: ");

    fgets(nuevo.empleado, sizeof(nuevo.empleado), stdin);

    nuevo.empleado[strcspn(nuevo.empleado, "\n")] = '\0';

    printf("Ingrese el valor original del activo: ");

    scanf("%f", &nuevo.valor\_original);

    printf("Ingrese la vida útil del activo (en años): ");

    scanf("%d", &nuevo.vida\_util);

    printf("Ingrese el año de compra: ");

    scanf("%d", &nuevo.anio\_compra);

    inventario[total\_activos] = nuevo;

    total\_activos++;

    guardarDatos();

    printf("Activo agregado con éxito.\n");

}

float calcularDepreciacion(Activo activo) {

    // Depreciación anual

    return activo.valor\_original / activo.vida\_util;

}

void consultarActivos() {

    for (int i = 0; i < total\_activos; i++) {

        printf("Activo %d:\n", i + 1);

        printf("Nombre: %s\n", inventario[i].nombre);

        printf("Tipo: %s\n", inventario[i].tipo);

        printf("Proyecto: %s\n", inventario[i].proyecto);

        printf("Empleado asignado: %s\n", inventario[i].empleado);

        printf("Valor original: %.2f\n", inventario[i].valor\_original);

        printf("Vida útil: %d\n", inventario[i].vida\_util);

        printf("Año de compra: %d\n", inventario[i].anio\_compra);

        // Mostrar la depreciación anual

        printf("Depreciación anual: %.2f\n\n", calcularDepreciacion(inventario[i]));

    }

}

void desasignarActivo() {

    int indice;

    consultarActivos();

    if (total\_activos == 0) {

        printf("No hay activos para desasignar.\n");

        return;

    }

    printf("Ingrese el número del activo a desasignar: ");

    scanf("%d", &indice);

    if (indice < 1 || indice > total\_activos) {

        printf("Índice inválido.\n");

        return;

    }

    for (int i = indice - 1; i < total\_activos - 1; i++) {

        inventario[i] = inventario[i + 1];

    }

    total\_activos--;

    guardarDatos();

    printf("Activo desasignado con éxito.\n");

}

void menu() {

    int opcion;

    do {

        printf("\n--- Menú de Inventario ---\n");

        printf("1. Agregar activo\n");

        printf("2. Consultar activos\n");

        printf("3. Desasignar activo\n");

        printf("4. Cambiar usuario y clave\n");

        printf("5. Salir\n");

        printf("Seleccione una opción: ");

        scanf("%d", &opcion);

        switch (opcion) {

            case 1:

                agregarActivo();

                break;

            case 2:

                consultarActivos();

                break;

            case 3:

                desasignarActivo();

                break;

            case 4:

                cambiarCredenciales();

                break;

            case 5:

                printf("Saliendo del programa...\n");

                break;

            default:

                printf("Opción no válida. Intente de nuevo.\n");

        }

    } while (opcion != 5);

}

int main() {

    char usuario\_ingresado[50], clave\_ingresada[50];

    cargarCredenciales();

    cargarDatos();

    while (1) {

        printf("--- Inicio de sesión ---\n");

        printf("Usuario: ");

        scanf("%s", usuario\_ingresado);

        printf("Clave: ");

        scanf("%s", clave\_ingresada);

        if (strcmp(usuario\_ingresado, usuario) == 0 && strcmp(clave\_ingresada, clave) == 0) {

            printf("Acceso concedido.\n");

            menu();

            break;

        } else {

            printf("Usuario o clave incorrectos. Intente de nuevo.\n");

        }

    }

    return 0;

}

**PRUEBAS DE INTEGRACION**

Para probar los resultados del código se realizaron comparaciones tomando encuentra las fórmulas expresadas en el diseño esquemático. Además, se confirmó que el código crea los ficheros que almacenan los datos de las herramientas y también el fichero de las credenciales.

Se realizaron pruebas del pseudocódigo en pseint.

**Manual de Usuario**

Este programa permite gestionar un inventario de activos mediante un sistema que ofrece opciones para agregar activos, consultar los activos existentes, desasignar activos y cambiar las credenciales de acceso. Los datos se almacenan en archivos y son recuperados al iniciar el programa.

**Funcionalidades**

1. **Inicio de sesión**
   * Al iniciar el programa, se solicita el usuario y la contraseña.
   * Las credenciales predeterminadas son:  
     **Usuario:** admin  
     **Clave:** 1234
   * Si las credenciales son correctas, se concede el acceso y se accede al menú principal.
2. **Menú de opciones** Después de iniciar sesión correctamente, se presenta el siguiente menú:
   * **1. Agregar activo**: Permite ingresar un nuevo activo al inventario.
   * **2. Consultar activos**: Muestra la lista de activos registrados, junto con su depreciación anual.
   * **3. Desasignar activo**: Elimina un activo del inventario.
   * **4. Cambiar usuario y clave**: Permite cambiar las credenciales de acceso (usuario y/o clave).
   * **5. Salir**: Cierra el programa.

**Detalles de las Opciones**

1. **Agregar Activo**
   * Se solicita información para crear un nuevo activo, como:
     + Nombre del activo
     + Tipo de activo
     + Proyecto asignado
     + Empleado asignado
     + Valor original
     + Vida útil (en años)
     + Año de compra
   * Si se alcanzó el límite de activos (100), no se podrá agregar más.
2. **Consultar Activos**
   * Muestra todos los activos registrados con sus respectivos detalles y su depreciación anual.
3. **Desasignar Activo**
   * Permite eliminar un activo. Se muestra la lista de activos y se solicita el número del activo a desasignar.
4. **Cambiar Usuario y Clave**
   * Para cambiar las credenciales, se debe ingresar la clave actual. Si es correcta, se podrá cambiar el usuario y la clave.

**Archivos de Datos**

* **credenciales.txt**: Almacena el usuario y la clave para el inicio de sesión.
* **inventario.txt**: Guarda los datos de los activos, incluyendo el nombre, tipo, proyecto asignado, empleado, valor original, vida útil y año de compra.

**Conclusión**

Para concluir, este código permite gestionar un inventario de activos dentro de una empresa, con funcionalidades clave como agregar, consultar, y desasignar activos, además de permitir la modificación de credenciales de acceso. Como mejoras, se sugiere añadir opciones para retroceder en cada parte del proceso del código.