

ISPC INNOVACIÓN EN GESTIÓN DE DATOS

INVENTARIO IPET-251

AUTORES

CABRERA JUAN GABRIEL
SANTARENA MARIA FLORENCIA

DESCRIPCIÓN

El proyecto consiste en desarrollar un sistema de gestión de inventario para los recursos educativos del IPET 251. Cuando se ejecuta el programa nos encontramos con las siguientes opciones: Ingresar (al sistema), crear usuario, modificar usuario, eliminar usuario, buscar usuario, mostrar todos los usuarios, salir. Una vez colocado el username y la contraseña se observa un menú, en donde los docentes podrán visualizar, agregar y modificar recursos en cuatro categorías: Informáticos, Equipamiento, Herramental e Insumos. El sistema permitirá a los usuarios acceder a información detallada sobre los elementos disponibles, gestionar su estado y asignación, y asegurará que solo usuarios con mayores credenciales puedan eliminar recursos, garantizando así la integridad de la base de datos.

JUSTIFICACIÓN

Este proyecto para establecer un inventario en el IPET 251 surge de una necesidad crítica: la falta de trazabilidad de los elementos y materiales empleados en las actividades educativas. Esta situación ha generado diversos inconvenientes que impactan negativamente tanto el rendimiento académico de los estudiantes como la eficiencia en la gestión de los recursos del laboratorio.

La trazabilidad de los elementos del laboratorio es esencial para garantizar un uso adecuado y seguro de los mismos. Sin un registro claro de qué materiales están disponibles, su estado y ubicación, se corre el riesgo de pérdida, deterioro o mal uso de los insumos. Esto no solo compromete la calidad de la enseñanza, sino que también puede poner en riesgo la seguridad de los estudiantes y docentes al utilizar equipos o sustancias que no han sido debidamente supervisados.

Un inventario sistemático permite una mejor planificación y gestión de los recursos. Al conocer con precisión qué materiales están disponibles y cuáles son necesarios, se puede optimizar la compra y uso de insumos, evitando desperdicios y asegurando que siempre haya lo necesario para las actividades educativas. Esto se traduce en un uso más eficiente del presupuesto destinado al laboratorio.

Por lo tanto, la creación de un inventario en el laboratorio escolar es una respuesta proactiva a la falta de trazabilidad de los elementos. Este sistema no solo mejora la gestión de recursos, sino que también garantiza la seguridad de los estudiantes, fomenta una cultura de responsabilidad y permite una evaluación constante que beneficiará el proceso educativo. Implementar un inventario es, por lo tanto, un paso esencial hacia un laboratorio más eficiente y seguro.

OBJETIVOS

OBJETIVOS GENERAL: Establecer un sistema de inventario en el IPET 251 que garantice la trazabilidad de los elementos y materiales utilizados en las actividades educativas, optimizando así la gestión de recursos del laboratorio y mejorando el rendimiento académico de los estudiantes.

OBJETIVOS ESPECÍFICOS:

Facilitar la Visualización de Recursos: Proporcionar a los docentes una interfaz sencilla para acceder a la información de los recursos disponibles, incluyendo su estado, asignación y cantidad.

Permitir la Gestión de Recursos: Implementar funcionalidades que permitan a los docentes agregar nuevos recursos a la base de datos, asegurando que la información se mantenga actualizada.

Habilitar la Modificación de Recursos: Permitir la edición de información de recursos existentes, facilitando ajustes en los registros conforme sea necesario.

Establecer Control de Acceso: Crear un sistema de credenciales que limite la eliminación de recursos a usuarios autorizados, como directores o jefes de taller, para proteger la integridad de los datos.

Optimizar la Administración de Recursos: Mejorar la eficiencia en la gestión y seguimiento de los recursos educativos, promoviendo un uso responsable y organizado dentro del IPET 251.

METODOLOGÍA

Recolección de requisitos: reunión con docentes para entender sus necesidades.

Diseño del modelo de datos: creación de un esquema de la base de datos que contemple las diferentes categorías de recursos y sus atributos.

Desarrollo: Implementación del menú de ingreso de la aplicación (ingresar al sistema, crear usuario, modificar usuario, eliminar usuario, buscar usuario, mostrar todos los usuarios, salir) y el menú de usuario con las opciones para visualizar, agregar y modificar recursos.

Desarrollo de funciones para cada opción del menú:

Visualizar: mostrar todos los elementos o detalles de un elemento específico (estado, asignación, cantidad).

Agregar/crear: permitir la incorporación de nuevos recursos a la base de datos.

Modificar: facilitar la actualización de recursos existentes y la eliminación de los mismos, restringiendo esta última acción a usuarios con mayores credenciales.

Pruebas:

Pruebas de funcionalidad: verificar que cada opción de ambos menús opere correctamente.

Pruebas de seguridad: asegurar que solo usuarios autorizados puedan eliminar recursos.

Implementación:

Despliegue del sistema en el entorno de trabajo del IPET 251.

Capacitación a los docentes sobre el uso del sistema y sus funcionalidades.

Mantenimiento:

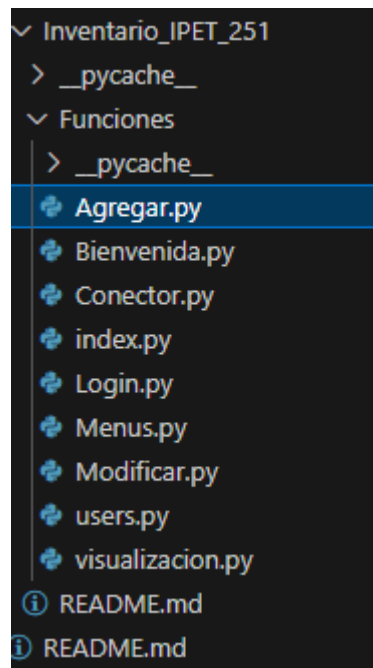
Establecimiento de un plan de mantenimiento para resolver posibles incidencias y realizar actualizaciones conforme se requiera.

CRONOGRAMA

CRONOGRAMA											
ETAPA DEL PROYECTO	ACTIVIDAD	MES									
		AGOSTO		SEPTIEMBRE				OCTUBRE			
		SEMANA 3	SEMANA 4	SEMANA 1	SEMANA 2	SEMANA 3	SEMANA 4	SEMANA 1	SEMANA 2	SEMANA 3	SEMANA 4
DIAGNOSTICO	Reuniones de equipo para la definición de requisitos y diseño	X	X								
	Identificación de elementos y materiales existentes	X	X								
	Evaluación de la situación actual del laboratorio		X	X							
IMPLEMENTACION	Diseño del formato del inventario				X	X					
	Programación de las funcionalidades del sistema.					X	X	X			
	Registro de elementos y materiales en el inventario								X		
MONITOREO Y EVALUACION	Revisión y prueba de cada módulo desarrollado.									X	X
	Ajustes y mejoras en el sistema según retroalimentación										X

A continuación se procederá a mostrar mediante algunos registros fotográficos o scripts la aplicación creada en python y las diferentes tablas de la base de datos con la que se vinculó.

Módulos que integran la aplicación



En este script podemos observar la función ‘agregar’ en donde definimos a dicha función para cada tabla del inventario

```

def agregado():
    system("cls")
    print("-----")
    print('          ALTA DE RECURSOS          ')
    print("-----")
    print(" ")
    print("          Que desea agregar          ")
    print("-----")
    print(" ")
    print('1. Recursos Informaticos')
    print('2. Herramental')
    print('3. Equipamiento')
    print('4. Insumos')
    print('0. Menu Principal')
    print(" ")
    op_ag = int(input(": "))
    match op_ag:
        case 1:
            ag_info()

        case 2:
            ag_herr()

        case 3:
            ag_eq()

        case 4:
            ag_ins()

        case 0:
            system("cls")

```

```

def agregado():
    agregado()

def ag_info():
    a = input("Tipo: ")
    b = input("Marca: ")
    c = input("Número de serie: ")
    d = input("Modelo: ")
    e = input("Número de serie secundario: ")
    f = input("Origen: ")
    g = input("Estado: ")

    try:
        conec = conexion()
        cursor = conec.cursor()
        sql = """
        INSERT INTO informaticos (tipo, marca, modelo, n_serie, n_serie_sec, origen, estado, Laboratorio_idlabor
        VALUES (%s, %s, %s, %s, %s, %s, %s, %s);
        """
        valores = (a, b, d, c, e, f, g, 1)
        cursor.execute(sql, valores)
        conec.commit()
        print(cursor.rowcount, "Registro ingresado")
        conec.close()
    except mysql.connector.Error as e:
        print(f"Error {e}")
    sleep(1)
    agregado()

```

Módulo index del código para el inventario IPET 251

```

from Bienvenida import *
from Menus import *
from Login import *
from visualizacion import *
from os import system          #Importamos el modulo os la funcion system("cls") para poder borrar la c
from time import sleep

system("cls")                  #Borrado automatico de consola
bienvenida()                  #Funcion de bienvenida a sistema

"""
prof, contra = login()        #Llamamos a la funcion login() y recuperamos el usuario y contraseña
sleep(1.5)                    #Establecemos un delay para que se pueda leer el mensaje

system("cls")
bienvenida_prof(prof)         #Llamamos a la funcion importada saludo_prof para dar una bienvenida
"""

ingreso()
#menu_principal()             #Ejecutamos el menu principal
system("cls")
#cierre(prof)                 #Llamamos a la funcion saludo_prof pero para dar una despedida

```

Modulo login

```
from users import *
from os import system          #Importamos del modulo os la funcion system("cls") para poder borrar la consola
from time import sleep        #Importamos del modulo time la funcion sleep() que nos permite realizar un sleep
from Bienvenida import bienvenida_prof

def solicitar_datos():
    id_usuario = int(input("Ingrese el DNI del Usuario: "))
    username = input("Ingrese el Nombre de Usuario: ")
    password = input("Ingrese la Contraseña: ")
    email = input("Ingrese el Correo Electrónico: ")
    return id_usuario, username, password, email

def login():
    system("cls")
    print("Ingrese sus credenciales para acceder al sistema.")
    username = input("Usuario: ")
    password = input("Contraseña: ")
    try:
        usuarios_lista = leer_usuarios("usuarios.ispc")
    except FileNotFoundError:
        system("cls")
        print("No se encontraron usuarios registrados.")
        return False
    for usuario in usuarios_lista:
        if usuario.getUsername() == username and usuario.getPassword() == password:
            system("cls")
            bienvenida_prof(username)
            sleep(1)
            system("cls")
            return True
```

Menú principal


```

from visualizacion import visualizar_elementos
from Agregar import agregado
from os import system          #Importamos del modulo os la funcion system("cls") para poder borrar la consola
from time import sleep         #Importamos del modulo time la funcion sleep() que nos permite realizar un sleep
from Modificar import modificacion
from users import *            #Importamos del modulo users sus clases y sus funciones
from Login import *

def ingreso():
    op_menu=99
    print("-----")
    print('                IPET Nº 251                ')
    print("-----")
    print('1. Ingresar al sistema')
    print('2. Crear Usuario.')
    print('3. Modificar Usuario.')
    print('4. Eliminar Usuario.')
    print('5. Mostrar listado de usuarios.')
    print('0. Salir')
    try:
        op_menu = int(input("Por favor, Ingrese el número de la opción deseada: "))
    except ValueError:
        print("Entrada inválida. Por favor, ingrese un número.")
        sleep(1)
        system("cls")
    # Volver al inicio del bucle para solicitar la opción de nuevo

    system("cls") # Limpiamos la pantalla entre opciones

```

```

match op_menu:
    case 1:
        if login():
            menu_principal()
        else:
            system("cls")
            print("Usuario o contraseña incorrecta")
            sleep(1)
            ingreso()
    case 2:
        system("cls")
        print("Crear un nuevo Usuario")
        id_usuario, username, password, email = solicitar_datos()
        usuario_nuevo = Usuario(id_usuario, username, password, email)
        agregar_usuario(usuario_nuevo)
        print("Usuario agregado exitosamente")
        sleep(1)
        system("cls")
        ingreso()
    case 3:
        system("cls")
        print("Modificar un Usuario")
        id_usuario = int(input("Ingrese el DNI del Usuario que desea modificar: "))
        actualizar_usuario(id_usuario)
        sleep(1)
        system("cls")

```

```
case 4:
    system("cls")
    print("Eliminar un Usuario")
    id_usuario = int(input("Ingrese el ID del Usuario que desea eliminar: "))
    eliminar_usuario(id_usuario)
    sleep(1)
    system("cls")
    ingreso()

case 5:
    system("cls")
    print("Mostrar todos los usuarios")
    mostrar_usuarios()
    sleep(3)
    ingreso()

case 0:
    exit

case _:
    print("Opcion Incorrecta")
    sleep(1)
    system("cls")
    ingreso()
```

Módulo users

```

import pickle          #Importamos la libreria pickle que nos permite convertir objetos a binarios

class Usuario:         #Definicion de la clase Usuario
    #Definicion de variables privadas
    __id:int
    __username:str
    __password:str
    __email:str

    def __init__(self,id:int, username:str, password:str, email:str) -> None:    #Constructor
        self.__id = id
        self.__username = username
        self.__password = password
        self.__email = email

    #Getters
    def getId(self):
        return self.__id
    def getUsername(self):
        return self.__username
    def getPassword(self):
        return self.__password
    def getEmail(self):
        return self.__email
    #Setters
    def setUsername(self, username:str):
        self.__username=username
    def setPassword(self, password:str):
        self.__password=password
    def setEmail(self, email:str):
        self.__email=email

```

```

def leer_usuarios(archivo = "usuarios.ispc"):    #Definimos una funcion para leer "usuarios" de un archivo binario, por defe
    with open(archivo,"rb") as archivo:         #Abrimos el documento guardado en archivo y con "rb" definimos que va a ser
        usuario = pickle.load(archivo)         # Almacenamos en la variable usuario la informacion leida a traves del meto
    return usuario

def agregar_usuario(usuario:Usuario, archivo = "usuarios.ispc"): #Definimos una funcion para agregar "usuarios" de un archi
    try:
        usuarios_en_lista = leer_usuarios(archivo)    #Utilizamos la funcion predefinina que internamente nos dev
    except FileNotFoundError:
        usuarios_en_lista = []
    """
    El try/except es para manejar algun error que pueda aparecer por no existir el archivo o estar en blanco
    """
    usuarios_en_lista.append(usuario)                #Agregamos a el usuario a la lista existente en el archivo

    with open(archivo, 'wb') as archivo:
        pickle.dump(usuarios_en_lista, archivo)        #Sobreescribimos la lista existente con la nueva lista que

    print(f"El Usuario: {usuario.getUsername()} ha sido agregado exitosamente")

    """
    En esta funcion utilizamos el metodo wb que sobre escribe la informacion que haya en el archivo binario
    utilizamos este metodo porque al trabajar con poca informacion va a ser mas eficiente y mas facil de
    trabajar
    """

```

BASE DE DATOS

Esquemas de la base de datos

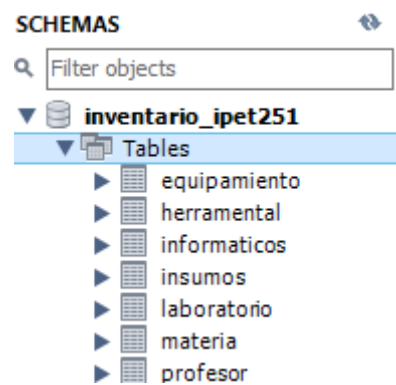


Tabla equipamiento

	idEquipamiento	tipo	uso	marca	modelo	num_serie	descripcion	fecha_calibracion	Laboratorio_idlaboratorio
▶	1	multimetro	manual	mlt	NULL	NULL	NULL	NULL	1
	2	fuentes de alimentacion v	manual	ftv	NULL	NULL	NULL	NULL	3
	3	multimetro	manual	eot	NULL	NULL	NULL	NULL	4
	4	termometro laser	manual	tls	NULL	NULL	NULL	NULL	5
	5	estacion de soldadura	electronico	sde	NULL	NULL	NULL	NULL	2
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

SQL de la tabla laboratorio

```
CREATE DATABASE IF NOT EXISTS `inventario_ipet251`;

-- -----
-- Table `inventario_ipet251`.`Laboratorio`
-- -----

USE `inventario_ipet251`;
) CREATE TABLE IF NOT EXISTS `Laboratorio` (
  `idlaboratorio` INT NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(45) NOT NULL,
  `cantidad_alumnos` INT NOT NULL,
  `metros_cuadrados` INT NOT NULL,
  PRIMARY KEY (`idlaboratorio`)
) ;
```

SQL de la tabla equipamiento

```
-- Table `inventario_ipet251`.`Equipamiento`  
-----  
CREATE TABLE IF NOT EXISTS `Equipamiento` (  
  `idEquipamiento` INT NOT NULL AUTO_INCREMENT,  
  `tipo` VARCHAR(45),  
  `uso` VARCHAR(10),  
  `marca` VARCHAR(10),  
  `modelo` VARCHAR(15),  
  `num_serie` VARCHAR(25),  
  `descripcion` VARCHAR(45),  
  `fecha_calibracion` DATETIME,  
  `Laboratorio_idlaboratorio` INT NOT NULL,  
  PRIMARY KEY (`idEquipamiento`),  
  INDEX `fk_Equipamiento_Laboratorio1_idx` (`Laboratorio_idlaboratorio` ASC) VISIBLE,  
  CONSTRAINT `fk_Equipamiento_Laboratorio1`  
    FOREIGN KEY (`Laboratorio_idlaboratorio`)  
    REFERENCES `Laboratorio` (`idlaboratorio`)  
  );
```

Carga de datos en la tabla de profesor

```
SELECT * FROM inventario_ipet251.profesor;  
INSERT INTO profesor VALUES  
( '123456', 'Agustin', 'Bracamonte', 'agustinbracamonte@gmail.com', '4'),  
( '234567', 'Maria', 'Serrezuela', 'mariaserrezuela', '3'),  
( '345678', 'Gustavo', 'Berni', 'gustavoberni@gail.com', '2'),  
( '456789', 'Teresa', 'Calo', 'teresacalo@gmail.com', '5'),  
( '567890', 'Gaston', 'Bermudes', 'gastonbermudes@gmail.com', '1');
```