

Examen DWES

Información general:

<https://github.com/JorgeDuenasLerin/docencia-23-24/>

Todo lo que no cumpla los requisitos o no sea entregado según formato no será evaluado.

Entrega:

apellido1-apellido-nombre

\- ejercicio1y2y3

\- código

\- entrega.txt <- información de qué ejercicios se han realizado.

\- ejercicio4

2 horas por trimestre

Revisa la base de datos db.sql

Usuario examen, password examen y contraseña examen.

Trimestre 1

Ejercicio 1

Gestión de formulario de inserción para la tabla `pedidos`

Características:

- Las flores irán en un select con información sacada de la base de datos, su stock parecerá entre paréntesis.
- Todos los campos son obligatorios
- La fecha debe ser posterior a hoy
- Debe haber suficiente stock
- Cuando se guarde un pedido se actualizará el stock y se redirigirá a la página de éxito (Por hacer)

Puntos:

- 1.00 puntos: sacar flores con stock en select ejemplo: “Rosa(100)”.
- 1.00 puntos: mantener todos los campos menos de texto y fechas
- 1.00 puntos: mantener selección de flor
- 0.50 puntos: validar fecha
- 1.00 puntos: validar stock

- 1.00 puntos: registrar pedido
- 1.00 puntos: actualizar stock
- 1.00 puntos: redirigir a éxito

Ejercicio 2

Listado para usuarios autenticados ordenado por fecha de entrega.

Crea un login.php, logout.php y un pedidos.php

Puntos:

- 1.00 puntos. login.php con base de datos
- 0.25 puntos. proteger pedidos.php
- 0.25 puntos. logout.php
- 1.00 puntos. listado ordenado (recuerda que necesitas los puntos anteriores)

Trimestre 2

Ejercicio 1

Recupera contraseña. Completa el script para recuperar contraseña

Puntos 3.5 puntos:

- 0.25 puntos. Sacar listado de tokens de base de datos
- 1.50 puntos. Verificar token antes de pedir contraseña
- 1.50 puntos. Actualizar contraseña del usuario y consumir token
- 0.25 puntos. Redirigir a página del login

Ejercicio 2

Django con API

Desarrolla un sistema de gestión de mantenimiento para edificios utilizando Django, que permita registrar edificios, técnicos de mantenimiento y las actividades de mantenimiento realizadas. El sistema debe proporcionar un API para consultar la información sobre los mantenimientos realizados.

Entidades:

- Edificio: Representa los edificios que requieren mantenimiento. Cada edificio tiene un nombre, una dirección y un contacto principal.
 - Edificio tendría campos como nombre, dirección, contacto (email)

- Técnico: Refleja los profesionales técnicos que realizan las tareas de mantenimiento. Cada técnico tiene un nombre, una especialidad y un número de contacto.
 - Técnico incluiría nombre, especialidad, contacto (email).
 - La especialidad es de tipo cadena, no hace falta entidad extra.
- Mantenimiento: Detalla las actividades de mantenimiento realizadas. Incluye la fecha del mantenimiento, la descripción del trabajo realizado, el edificio en el que se realizó el mantenimiento y el técnico que lo ejecutó.
 - Mantenimiento contendría fecha, descripción, una clave foránea a Edificio y otra a Técnico.

Administración para todas las entidades.

Parte pública:

Listado de mantenimientos realizados con paginación cada 4 y ver detalle

Funcionalidades del API:

- Consulta de mantenimientos: Permite consultar las actividades de mantenimiento realizadas.

Puntos:

- 1.0 puntos entidades
- 1.0 puntos administración
- 1.0 puntos template base (con estilos)
- 1.0 puntos listado paginado
- 1.0 puntos detalle
- 1.5 puntos API rest