

Información sobre la entrega

En este examen se evalúan contenidos específicos del módulo. Cada ejercicio **debe** ser hecho con las técnicas requeridas y descritas en el enunciado, en caso contrario el ejercicio será puntuado con **0 puntos**.

Ejemplo: Escribe un programa que cree un proceso hijo con fork, el hijo escribirá “Hola mundo” y el padre esperará a que finalice el proceso hijo.

Entrega: El alumno entrega un programa que escribe “Hola mundo” (Similar al que se hace el primer día de clase de programación).

Resultado: **0 puntos**.

Ejemplo: La segunda actividad realiza una petición al API y muestra un listado de elementos.

Entrega: Una actividad que muestra unos elementos escritos con código.

Resultado: **0 puntos**.

Puedes utilizar todo el código y material que consideres sin acceder a Internet.

En caso de copia, intento, o acceso a documentación no indicada en el enunciado, conllevará una calificación en la prueba de suspenso más su correspondiente sanción.

Para poder puntuar en algún ejercicio, será requisito indispensable que compile y ejecute sin ningún tipo de error.

Formato d entrega:

- Se habilitará en el aula virtual una tarea para la entrega del examen.
- La entrega constará de un único fichero comprimido con el nombre del alumno.
- Dicho fichero, provendrá de un proyecto con los paquetes relativos a los ejercicios.
- En la cabecera de cada clase, como comentario o documentación, vuestro nombre
- completo. En caso de omitirse, se anulará la puntuación de dicho ejercicio

Penalizaciones:

Cada línea de código con mala sangría o sangría inconsistente restará **0.1 punto**.

Cada número mágico o información constante dentro del código que pueda ser modificado en el futuro y que no esté almacenada en una constante restará **0.1 puntos**. Los valores constante deben ser CONSTANTES (escritas con nombre mayúsculas)

Cada incumplimiento de reglas de estilo del lenguaje de programación tendrá una penalización de **0.05 puntos**.

Java: Clases en mayúsculas o PascalCase, métodos y variables camelCase

Python: Clases en mayúsculas o PascalCase, métodos y variables snake_case

Cada fallo que ocurre en la aplicación por situaciones no contempladas que la hagan cerrarse de forma inesperada restará **0.5 puntos**

Tiempo 1 hora

Crea un formulario para obtener la información sobre edificios históricos.

Se quiere recoger información sobre:

- Nombre. Texto
- Descripción. TextArea
- Año de construcción. Fecha
- Estilo. Texto (Opcional)

Crea un formulario para almacenar esta información en la base de datos. Si falta alguno de los campos el formulario mostrará qué campo falta y preservará los valores anteriores.

Cuando el formulario se rellene y se envíe se producirá un alta en la base de datos y se mostrará el listado de edificios insertados.

Debes crear la base de datos.

Base de datos examen, usuario examen, contraseña examen.

| Ítem | Puntos | ¿Se ha conseguido? |
|---------------------------------------|--------|--------------------|
| Procesa campos (proceso completo) | | |
| nombre | 0,5 | |
| descripción | 0,5 | |
| año | 0,5 | |
| estilo | 0,5 | |
| | | |
| Creación y conexión con base de datos | 1 | |
| Alta en base de datos | 1 | |
| Listado | 1 | |

Tiempo 1 hora

Autenticación con token

Analiza la base de datos que se adjunta con el ejercicio.

Analiza el contenido de la página enlaces.php

Visita la página enlaces.

login.php?token=<TOKEN>

Cuando se pinche el enlace de autenticación, la página de login.php hará lo siguiente:

- Cambiará el estado del usuario a activado estableciendo activado al valor 1
- Autenticará al usuario con una sesión
- Destruirá el token haciendo un DELETE

NOTA: Si el usuario ya tiene el valor activo a 1, no es necesario hacer este paso.

Áreas:

La página publica.php es pública y no requiere autenticación.

La página privada.php es privada y requiere autenticación.

Cuando el usuario esté autenticado se mostrará su nombre en toda la web.

| Ítem | Puntos | ¿Se ha conseguido? |
|--|--------|--------------------|
| Autenticación del usuario | 1,25 | |
| Gestión de la información de sesión en login y logout. Mostrar su nombre | 1,25 | |
| Borrado de token | 1,25 | |
| Activación de usuario solo en el primer acceso | 1,25 | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Tiempo 1 hora

Django - Utiliza SQLite

Registro civil.

Tenemos el modelo Persona

- nombre
- apellido1
- apellido2
- descripción

Tenemos el modelo Pareja

- integrante1
- integrante2
- fecha
- lugar: texto

Las personas en el admin se pueden listar por todos sus campos.

Las parejas en el admin muestran el apellido primero del integrante1, el apellido primero del integrante2 y entre paréntesis su fecha de unión.

Haz un listado en la parte pública de la web en el que aparezcan todas las parejas en una **tabla**:

| Apellido pareja | Lugar | Fecha |
|---------------------|--------|------------|
| Apellido1 Apellido2 | Madrid | 26-04-1984 |
| | | |
| | | |

Cada apellido te llevará al detalle de esa persona y la página mostrará sus datos:

Nombre, apellidos y descripción

| Ítem | Puntos | ¿Se ha conseguido? |
|-------------------------------------|--------|--------------------|
| Modelos | 1 | |
| Admin configurado en listados | 1 | |
| Listado de parejas | 1 | |
| Enlace usando url bien configuradas | 1 | |
| Ver detalle | 1 | |