

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Curso DESARROLLO DE APLICACIONES WEB Y MÓVILES - II PAO

2025

Evaluación: Examen Mejoramiento

Código de Honor - (0.0 punto(s))

Yo, en mi calidad de estudiante de la ESPOL, declaro que he sido informado y conozco las normas que rigen a la ESPOL, en particular el Código de Ética y el Reglamento de Disciplina.

Acepto con pleno conocimiento este compromiso de honor por el cual reconozco y estoy consciente de que la presente evaluación está diseñada para ser resuelta de forma individual; que sólo puedo comunicarme con la persona responsable de la recepción de la evaluación; que no haré consultas indebidas o no autorizadas por el evaluador; ni usaré dispositivos electrónicos.

Acepto el presente compromiso, como constancia de haber leído y aceptar la declaración anterior y me comprometo a seguir fielmente las instrucciones que se indican para la realización de la presente evaluación.

- Verdadero
- Falso

JAVASCRIPT - AJAX - (30.0 punto(s))

Planteamiento del problema

Una empresa de logística desea mejorar su panel interno de gestión de pedidos, actualmente, los pedidos se consultan recargando toda la página, lo que obviamente genera lentitud y mala experiencia de usuario.

Al área de sistemas se le solicita una solución, para la cual proponen usar JavaScript y AJAX, ya que esta combinación permite consultar los pedidos de un cliente sin recargar toda la página y así solo mostrará el resultado directamente en una sección de la pantalla.

La empresa ya cuenta con:

- Un backend que expone una API REST
- Una página HTML con un formulario básico
- Estilos CSS ya definidos (formatos y estilos propios de la empresa)
- El backend tiene el siguiente endpoint:
 - GET /api/pedidos?cliente=ID_CLIENTE
- La API devuelve una respuesta en formato JSON con la siguiente estructura:

○

```
[  
  { "id": 1, "producto": "Laptop", "estado":  
  "Entregado" },  
  { "id": 2, "producto": "Mouse", "estado": "Pendiente"  
 }  
]
```

Requerimiento funcional de desarrollo:

Se espera que, cuando el usuario (interno) del sistema ingrese el ID del cliente y presione el botón consultar:

1. Se debe realizar una petición AJAX usando fetch().
2. No se debe recargar la página.
3. Los pedidos deben mostrarse dinámicamente en una lista HTML.
4. Si no existen pedidos, se debe mostrar un mensaje claro al usuario indicando aquello.
5. Se debe manejar correctamente una respuesta por error del servidor.

Restricciones

- No debe de modificar el HTML ni CSS.
- No puede usar librerías externas.
- No puede usar frameworks.

Entrega solicitada:

- Solo debe de entregar el código JavaScript que presente la solución solicitada.
- Debe de agregar explicaciones y/o comentarios.

BOOTSTRAP - MAQUETACIÓN RESPONSIVA - (30.0 punto(s))

Planteamiento del problema.-

Una empresa de servicios educativos desea actualizar la sección “Nuestros cursos” de su sitio web institucional, para esto, el área de diseño ha definido que la maquetación debe ser totalmente responsive, es decir adaptable a cualquier tipo de pantalla, sin importar el tipo de dispositivo o sistema operativo.

La empresa ya cuenta con:

- Un archivo HTML base funcional
- Bootstrap incluido mediante el CDN del respectivo Framework
- Contenido textual e imágenes disponibles

Requerimiento funcional de desarrollo:

El área técnica solicita que el desarrollo solo sea la estructura HTML con clases Bootstrap, sin agregar lógica adicional.

Se pide que, al programar, siga el enfoque mobile-first utilizando exclusivamente Bootstrap

Desarrolle el código HTML necesario para construir una sección de cursos que cumpla con las siguientes condiciones:

1. La sección debe estar contenida dentro de un container.
2. Debe incluir un título centrado con el texto: **Cursos disponibles**.
3. Los cursos deben mostrarse usando el componente Card de Bootstrap.
4. El diseño debe comportarse así:
 - 1 columna en pantallas pequeñas
 - 2 columnas en pantallas medianas
 - 4 columnas en pantallas grandes
5. Cada card debe incluir:
 - Imagen superior
 - Título del curso
 - Breve descripción
 - Botón “Ver más”

6. Todas las cards deben mantener la misma altura.

Restricciones:

- No escribir CSS personalizado.
- No usar estilos en línea.
- No usar JavaScript.
- No modificar el CDN de Bootstrap.
- No usar librerías externas.
- No puede usar otros Framework

Entrega solicitada

- Entregue únicamente el código HTML correspondiente a la sección solicitada.
- Debe de agregar explicaciones y/o comentarios.

DJANGO - LISTADO DE REGISTROS - (40.0 punto(s))

Planteamiento del problema

El taller MecanBici cuenta con un sistema web interno que opera dentro de su intranet, en el sistema se registra todas las reparaciones realizadas a las bicicletas de los clientes como parte del proceso diario.

La gerencia del taller necesita visualizar el historial completo de reparaciones para poder analizar el funcionamiento del negocio, evaluar costos, identificar los tipos de reparaciones más frecuentes y apoyar la toma de decisiones administrativas.

El sistema ya cuenta con:

- Una base de datos denominada **DBMECANICA**
- Una tabla llamada **Reparaciones**
- Un proyecto Django correctamente configurado y funcional al 100%
- El modelo **Reparacion** definido y sincronizado con la base de datos

Dado que el sistema opera exclusivamente dentro de la intranet de la empresa, la gerencia solicita que se muestre información crítica y consolidada sobre las reparaciones registradas, sin necesidad de realizar modificaciones estructurales al sistema existente.

Información del modelo.- El modelo **Reparacion** contiene los siguientes campos:

- id
- cliente
- bicicleta
- tipo_reparacion
- costo
- fecha

Nota: Este es el conjunto de datos de información mínima necesaria para analizar el historial de reparaciones realizadas por el taller.

Requerimiento funcional de desarrollo:

Desarrolle el código necesario en Django para cumplir con lo siguiente:

1. Crear una vista que consulte todos los registros de la tabla **Reparaciones**.
2. Enviar los datos consultados a un template.
3. Mostrar la información en una tabla HTML.
4. En la tabla se deben mostrar las 6 columnas del modelo.
5. La página debe cargarse correctamente sin errores.

Restricciones:

- No puede usar SQL directo.

- No puede usar consultas raw.
- No puede usar Django Admin.
- No puede usar JavaScript.
- No puede usar librerías externas.
- Debe usar exclusivamente el ORM de Django.

Entrega solicitada:

- Código de la vista
- Código del template
- No se solicita models.py ni settings.py.
- No se solicita estilos CSS.