

1. Normas

- El examen es individual.
- Se podrá hacer uso del código de las prácticas realizadas durante el curso.
- Se puntuará la corrección de los resultados, pero también la “calidad” del código (legibilidad, comentarios, seguimiento del patrón de diseño proporcionado por Django, etc.).
- Se entregará en moodle como resultado del examen el fichero producido por el comando `git archive --format zip --output ../examenX.zip master` (donde X será una letra que identificará el examen)
- Se incluirá un fichero de texto (con el nombre `memoria.txt`) describiendo como se ha realizado cada uno de los apartados descritos a continuación.
- La hora límite para entrega de los ficheros será fijada por el profesor. A partir de dicha hora se penalizará con un punto cada minuto de retraso.

Enunciado

Un hospital desea implementar un servicio de gestión de medicamentos. Este servicio viene soportado por la base de datos correspondiente al siguiente esquema relacional:

paciente(id , nombreP)
medico(id,nombreM)
receta(id,medico(id)↑,paciente(id)↑,)
donde ↑ indica clave extranjera

Recordad que Django añade automáticamente el atributo *id* que funciona como clave primaria.

Actividades a realizar

1. Cread un proyecto llamado **proyecto** en Django. Dentro del proyecto crear una aplicación llamada **aplicacion**. Los datos se persistirán en una base de datos llamada **examen** creada en postgres.
2. Cread el modelo de datos (fichero **models.py**) y configurad el proyecto para que los modelos puedan ser accedidos usando el sistema de administración de Django (<http://localhost:8000/admin/>).
3. Escribid un script (llamado **poblar.py**) que inserte los datos siguientes en la base de datos usando la API ofrecida por Django.

```
medico(1,'medico1')
medico(2,'medico2')
medico(3,'medico3')
medico(4,'medico4')
```

```
paciente(1,'paciente1')
paciente(2,'paciente2')
```

```
receta(1,1,1)
receta(2,2,1)
receta(3,1,2)
receta(4,2,2)
receta(5,3,2)
```

4. Despliega el proyecto en Heroku y puebla la base de datos usando el fichero **poblar.py**, así mismo activa el interfaz de administración usando como nombre de usuario y password **alumnodb**. Nota: incluir en el fichero **memoria.txt** el URL en el cual la aplicación está accesible en Django. IMPORTANTE: En cuando acabéis de implementar este punto, y siempre antes de la hora de finalización del examen, mostrárselo a vuestro profesor.
5. Usando el template **receta.html**, crea una página web accesible en la dirección `$PROJECT_URL/aplicacion/receta/` que devuelva una lista con las últimas 3 recetas expedidas. Nota: **template.html** no debe modificarse. Si la base de datos está vacía se debe informar al usuario a través de la variable **error**.
6. Crea un test (a ejecutarse con la orden `python manage.py aplicacion.test --keepdb`) que:
 - borre todos los medicos, pacientes y recetas
 - cree el médico (1,'medico1')
 - cree el paciente (1,'paciente1')

- cree la receta (1,1,1)
- cree la receta (2,1,1)
- cree la receta (3,1,1)
- cree la receta (4,1,1)
- acceda a la vista relacionada con el URL `$PROJECT_URL/aplicacion/receta/`
- compruebe que las recetas devueltas son las correctas.

2. Normas de calificación

Para obtener 5 puntos (aprobar) es necesario que tras ejecutar los comandos:

```
dropdb -U alumnodb -h localhost examen
createdb -U alumnodb -h localhost examen
python manage.py makemigrations aplicacion
python manage.py migrate
python manage.py createsuperuser
python ./poblar.py
```

Se pueda acceder al interfaz de administración `http://localhost:8000/admin/` y obtener un listado con los datos enumerados en el punto (3) del apartado anterior. Igualmente es necesario que los datos se persistan en una base creada en postgres.

Puntuación 7 puntos si se satisfacen todos los puntos del apartado anterior y es posible acceder al interfaz de administración de la aplicación desplegada en heroku usando el username/password `alumnodb`. Nota: el código desplegado en Heroku debe ser IDÉNTICO al subido a moodle, en caso contrario la puntuación de este apartado será nula.

Puntuación 8 puntos si se satisfacen todos los puntos del apartado anterior y la página `$PROJECT_URL/aplicacion/receta/` funciona correctamente. Todas las variables usadas en el template `receta.html` se deben suministrar.

Puntuación 10 puntos si se satisfacen todos los puntos del apartado anterior y el test se satisface todos los requerimientos solicitados.

A. Templates

```
<html>
  <head>
  </head>
<body>

{% if error %}
```

```

        {{ error }}
{% endif %}
<table>
{% for receta in recetas %}
    <tr>
        <th>{{ receta.id }}</th>
        <td>{{ receta.paciente.nombreP }}</td>
        <td>{{ receta.medico.nombreM }}</td>
    </tr>
{% endfor %}
</table>
</body>
</html>

```