



## Entrenamiento de Python-Django - 3

### Sesiones

Resumen: ¡O cómo un simple proyecto Django puede convertirte en cinturón negro en gestión de permisos y usuarios!

Versión: 1

# Contenido

I	Reglas generales	2
II	Reglas específicas de hoy	3
III	Ejercicio 00	4
IV	Ejercicio 01	5
V	Ejercicio 02	7
VI	Ejercicio 03	8
VII	Ejercicio 04	10
VIII	Ejercicio 05	11
Ejercicio IX	06	12
X	Presentación y evaluación por pares	13

# Capítulo I

## Reglas generales

- Su proyecto debe realizarse en una máquina virtual.
- Su máquina virtual debe tener todo el software necesario para completar su proyecto. Estos software deben configurarse e instalarse.
- Puede elegir el sistema operativo que utilizará para su máquina virtual.
- Debe poder utilizar su máquina virtual desde una computadora del clúster.
- Debe utilizar una carpeta compartida entre su máquina virtual y su máquina host.
- Durante tus evaluaciones utilizarás esta carpeta para compartir con tu repositorio.

Sus funciones no deben cerrarse inesperadamente (fallo de segmentación, error de bus, doble liberación, etc.), salvo por comportamientos indefinidos. Si esto ocurre, su proyecto se considerará no funcional y recibirá un 0 en la evaluación.

Te animamos a crear programas de prueba para tu proyecto, aunque este trabajo no tenga que entregarse ni se calificará. Esto te permitirá evaluar fácilmente tu trabajo y el de tus compañeros. Estas pruebas te resultarán especialmente útiles durante la defensa. De hecho, durante la defensa, puedes usar tus propias pruebas o las del compañero que estás evaluando.

Envía tu trabajo al repositorio Git asignado. Solo se calificará el trabajo en el repositorio Git. Si Deepthought se encarga de calificar tu trabajo, lo hará después de la evaluación por pares. Si se produce un error en alguna sección de tu trabajo durante la calificación de Deepthought, la evaluación se detendrá.


## Capítulo II

### Reglas específicas de hoy

- Debes utilizar la versión LTS de django.
- Puedes usar la versión de Bootstrap que prefieras. Esta versión debe ser compatible con el Versión LTS de Django.
- Debes utilizar Python3.

## Capítulo III

### Ejercicio 00

	Ejercicio
Ejercicio 00: Sesiones anónimas.	
Directorio de entrega: ex/	
Archivos a entregar: Su proyecto, su archivo requirements.txt Funciones permitidas:	
Todas las funcionalidades de Django, aleatorias, django-bootstrap*	

Estás iniciando un proyecto de sitio web que permitirá a los usuarios publicar "Consejos profesionales de vida".

Después de crear su proyecto, una aplicación y configurar una página de inicio en esta URL /, implemente el siguiente sistema de sesión anónima:


Un usuario web que visite su sitio puede obtener un nombre de usuario. Este nombre se elige aleatoriamente de una lista de 10 nombres que definirá en el archivo settings.py de su proyecto, y su validez solo dura 42 segundos después de su definición.

Este nombre debe ser persistente y permanecer idéntico durante 42 segundos. Transcurridos estos 42 segundos, debe redefinirse como se explicó anteriormente.

- Resalte el comportamiento implementado añadiendo un mensaje en un elemento <nav> del encabezado de la página de inicio. Este mensaje será un simple "¡Hola usuario!", reemplazando al usuario por el nombre aleatorio hasta el final de su validez.
- El nombre debe aparecer instantáneamente en las páginas visitadas: si necesita recargar el página para hacerla aparecer o cambiar, el ejercicio no está completo!

## Capítulo IV

### Ejercicio 01

	Ejercicio
Ejercicio 01: Creación de usuarios.	
Directorio de entrega: ex/	
Archivos a entregar: Su proyecto, su archivo requirements.txt Funciones permitidas:	
Todas las funcionalidades de Django, django-bootstrap*	

Ahora que se ha implementado el sistema de sesión anónima, es momento de agregar la gestión de usuarios con la creación de una interfaz de autenticación.

En este ejercicio, crearás esta interfaz paso a paso:

Empiece añadiendo una página de registro y una de inicio de sesión. Añada un enlace a cada una de estas páginas en el elemento de navegación. Para ello, añada todas las URL, plantillas y vistas necesarias. etc...

También agregará el nombre del sitio al comienzo de su navegación y lo convertirá en un enlace a su página de inicio.

- Crea formularios que insertarás en estas páginas siguiendo estas especificaciones:

- Formulario de inscripción:

Insertarás 3 campos diferentes, entre ellos:

- Un campo para el nombre de usuario.

- Un campo para la contraseña.

- Un campo para la confirmación de la contraseña.

La validación se llevará a cabo solo si cada campo está completado correctamente, el nombre dado no está ya en uso y ambos campos de contraseña están completados de manera estrictamente idéntica.

- Un formulario de inicio de sesión:

Debes insertar 2 campos diferentes, entre ellos:

- Un campo para el nombre de usuario.

- Un campo para la contraseña.

La validación se realizará sólo si cada campo está relleno y la pareja de usuario/contraseña utilizada ya existe en la base de datos y puede ser identificada.

- Ahora, implementa el comportamiento de tus páginas:

- En caso de validación de datos:

Registro: Creación y activación de un nuevo usuario con la información proporcionada. A continuación, conexión con este usuario y redirección a la página de inicio.

Conexión: Conexión con el usuario que coincida con la información especificada, y redirección hacia la página de inicio.

- En caso de no validación:

En cualquier caso, volverás a mostrar la página con el formulario que contiene la información que has introducido previamente (menos la contraseña).

Registro: Mostrar uno (o varios) mensajes de error explicando el error (campo obligatorio, nombre de usuario existente, contraseña incorrecta).

Iniciar sesión: Mostrar un mensaje de error con el formulario (campo obligatorio o información de inicio de sesión incorrecta).

- Finalmente, cuando el usuario haya iniciado sesión:

- Los enlaces a las páginas de registro e inicio de sesión se reemplazan por un enlace de cierre de sesión (funcional, por supuesto) que cierra la sesión del usuario y lo redirige a la página de inicio después de hacer clic.

- El nombre de la sesión anónima en el elemento nav se reemplaza por el nombre de usuario.

- El visitante no debe poder acceder a las páginas de registro e inicio de sesión. Si intenta acceder a ellas, será redirigido a la página de inicio.

- Por supuesto, las contraseñas quedarán ocultas al escribirlas.




En otro ejercicio, tendrás que crear tu propia clase de usuario para reemplazar la clase predeterminada proporcionada.

Puede ser conveniente (pero no necesario) anticipar este cambio en su proyecto para no tener que reescribir manualmente cada llamada a su clase de usuario en su código.

## Capítulo V

### Ejercicio 02

	Ejercicio
Ejercicio 02: ¡Nuestros consejos!	
Directorio de entrega: ex/	
Archivos a entregar: Su proyecto, su archivo requirements.txt Funciones permitidas:	
Todas las funcionalidades de Django, django-bootstrap*	

Ha llegado el momento de implementar la funcionalidad principal de este proyecto: ¡un sistema de propinas!

Para ello será necesario concebir un modelo Tip con:

- Un campo de contenido con... el contenido de la sugerencia (en formato de texto, por supuesto).
- Un campo de autor (el usuario que creó la sugerencia).
- Un campo de fecha (fecha de creación de la sugerencia).

Los consejos deberán estar enumerados en la página de inicio y deberán mostrarse todos los atributos indicados en esta sección.


Una vez iniciada la sesión, el usuario debe poder crear una sugerencia con el formulario configurado en la página de inicio.  
(¡use un formulario modelo!)

Si el usuario no ha iniciado sesión, el formulario no debe aparecer y el visitante no debe estar  
Capaz de crear una propina.



## Capítulo VI

### Ejercicio 03

	Ejercicio
Ejercicio 03: Votos.	
Directorio de entrega: ex/	
Archivos a entregar: Su proyecto, su archivo requirements.txt Funciones permitidas:	
Todas las funcionalidades de Django, django-bootstrap*	

Ahora que tu sistema de propinas está listo, sería genial poder recopilar las mejores. ¡Sí, es hora de configurar un sistema de votos positivos y negativos!

Para ello, deberás modificar tu modelo de Tip para poder implementar estas funcionalidades.

También debes implementar una forma de eliminar los consejos que se consideren poco interesantes.

Debes agregar los siguientes elementos a los consejos que aparecen en tu página de inicio:

- Un botón de eliminar
- Un botón de voto positivo
- Un botón de voto negativo
- El número de votos positivos
- El número de votos negativos

Por el momento, la única restricción que implementará será la necesidad de iniciar sesión para votar o eliminar. Si el usuario no ha iniciado sesión, ninguna de estas acciones será posible.

Cada una de estas acciones recargará la página.

Obviamente, un consejo recién creado tendrá 0 votos a favor y 0 votos en contra.

Advertencia:

- Debe ser imposible votar a favor o en contra del mismo consejo al mismo tiempo.
- Un usuario solo puede votar a favor o en contra del mismo consejo una vez. Al hacer clic dos veces en el mismo botón cancela el voto positivo o negativo.
- Si un usuario vota negativamente sobre un consejo que previamente había votado positivamente, el voto positivo se reemplaza por el voto negativo (y viceversa).




Vigila atentamente la implementación de tu sistema de votos positivos. Una inconsistencia o imperfección (que el número de votos no disminuya tras cancelar una votación) invalidará el ejercicio. Deberías considerar usar campos "mucho a muchos" en lugar de "contadores" para implementar tu sistema de votos.

En esta etapa, cada acción solo debería funcionar si el usuario ha iniciado sesión.  
Debe poder mostrar o deshabilitar el elemento que el usuario no podrá utilizar.

## Capítulo VII

### Ejercicio 04

	Ejercicio
Ejercicio 04: Uso primario de autorizaciones.	
Directorio de entrega: ex/	
Archivos a entregar: Su proyecto, su archivo requirements.txt Funciones permitidas:	
Todas las funcionalidades de Django, django-bootstrap5	

Ahora nuestro proyecto es un poco más sustancial, nos damos cuenta de que cualquiera puede hacer cualquier cosa. siempre que hayan iniciado sesión. Quizás haya llegado el momento de añadir algunas pequeñas restricciones.

El usuario ahora necesita una autorización para eliminar consejos si desea... eliminar consejos.


Activa la interfaz de administración que viene con Django por defecto y utiliza un superusuario para modificar los permisos y resaltar el comportamiento de su proyecto.

Por supuesto, se necesita más que simplemente eliminar el botón "Eliminar" de la página (aunque podría hacerlo). Encuentre la manera correcta de restringir esta funcionalidad verificando correctamente los permisos.

NB: Por supuesto, el autor de un tip puede eliminar su tip sin ninguna autorización. Esta será la única excepción a la regla.

## Capítulo VIII

### Ejercicio 05

	Ejercicio
Ejercicio 05: Autorización personalizada.	
Directorio de entrega: ex/	
Archivos a entregar: Su proyecto, su archivo requirements.txt Funciones permitidas:	
Todas las funcionalidades de Django.	

Dado el carácter negativo del voto negativo, su uso debería reservarse para casos específicos. usuarios.


Implementar el mismo control que el ejercicio anterior para los votos negativos. Dado que este no es un permiso predeterminado y sería una pena que utilizaras otro permiso que no fuera adecuado, ¡crea tu propio permiso para restringir el uso del voto negativo!

Al igual que en el ejercicio anterior, no se trata solo de eliminar el botón de voto negativo. Puedes intentarlo, ¿no? Aunque el éxito no está garantizado.

NB: al igual que con la supresión de una sugerencia, el autor de una sugerencia siempre puede votar negativamente sus propias sugerencias si cree que son tontas o algo así.

## Capítulo IX

### Ejercicio 06

	Ejercicio
Ejercicio 06: Automatización y reputación Directorio	
de entrega: ex/ Archivos a	
entregar: Su proyecto, su archivo requirements.txt Funciones permitidas: Todas las	
funcionalidades de Django, django-bootstrap*	

Ahora que contamos con la mayoría de las herramientas necesarias para ejecutar este proyecto, sería bueno no tener que gestionar manualmente esos permisos. Algunos sitios han empezado a otorgar privilegios con una escala de reputación. Esto otorga a los usuarios que lo merecen el derecho a usar nuevas funciones. De eso se trata este ejercicio. Deberás establecer un sistema de reputación basado en los votos que hayan recibido los consejos publicados por un usuario.

Reemplaza la clase de usuario estándar por tu propia clase de usuario. En esta clase, implementarás una forma en que las autorizaciones para votar negativamente y eliminar sugerencias dependan de la reputación del usuario. La reputación de este usuario dependerá de la cantidad de votos positivos y negativos que reciban sus sugerencias.

Cada usuario nuevo empieza con 0 puntos de reputación. Cada voto positivo a sus consejos le otorgará 5 puntos de reputación. Un voto negativo le restará 2. Si se elimina un consejo, la influencia de sus votos en la reputación del usuario desaparece.

Un usuario desbloqueará su autorización para votar negativamente (en otros consejos que no sean el suyo) con 15 puntos de reputación. Se necesitarán 30 puntos de reputación para obtener la autorización para eliminar consejos. Si la reputación de un usuario disminuye, puede perder las autorizaciones que le otorgaba su reputación.

También modificarás la apariencia del nombre del usuario, mostrando su reputación en un lateral (entre paréntesis) cuando aparezca en tus páginas.

## Capítulo X

# Presentación y evaluación por pares

Entrega tu tarea en tu repositorio de Git como de costumbre. Solo se evaluará el trabajo dentro de tus repositorios durante la defensa. No dudes en verificar los nombres de tus carpetas y archivos para asegurarte de que sean correctos.



El proceso de evaluación se realizará en el computador del evaluado.  
grupo.