

**Estado** Terminados  
**Iniciado** martes, 11 de marzo de 2025, 13:38  
**Completado** martes, 11 de marzo de 2025, 13:58  
**Duración** 20 mins 15 segundos  
**Calificación** **30.00** de un total de 30.00 (**100%**)

**Pregunta 1**

Completada

Puntúa 1.00 sobre 1.00

¿Por qué WSGI no es ideal para WebSockets?

- ☐ a. Porque cada request se maneja y cierra, impidiendo la comunicación constante
- ☐ b. Porque WSGI rechaza todas las peticiones HTTP
- ☐ c. Porque WSGI es estrictamente para bases de datos
- ☐ d. Porque WSGI se usa en Java, no en Python

**Pregunta 2**

Completada

Puntúa 1.00 sobre 1.00

A diferencia de HTTP tradicional, un WebSocket:

Seleccione una:

- ☐ a. Se basa en largas colas de peticiones HTTP GET sin respuesta.
- ☐ b. Es idéntico al polling, pero con cabeceras adicionales.
- ☐ c. No permite uso de canales full-duplex (solo unidireccional).
- ☐ d. Evita la apertura y cierre repetidos de conexiones para cada mensaje, permitiendo intercambio continuo de datos.

**Pregunta 3**

Completada

Puntúa 1.00 sobre 1.00

¿Para qué sirve la clave "type" dentro de los diccionarios enviados con group\_send en un AsyncWebsocketConsumer?

- ☐ a. Es un campo opcional sin efecto alguno
- ☐ b. Para indicar el nombre de la variable global a compartir
- ☐ c. Para asignar un ID único a cada mensaje
- ☐ d. Para especificar el método que manejará el evento, p. ej. "chat\_message"

## Pregunta 4

Completada

Puntúa 1.00 sobre 1.00

¿Cuál NO es una característica clave de ASGI?

- ☒ a. Es totalmente síncrono y evita la concurrencia
- ☐ b. Diseñado para permitir asincronía con servidores como Uvicorn
- ☐ c. Soporta varios protocolos como HTTP y WebSockets
- ☐ d. Separa la capa de aplicación de la capa de servidor

## Pregunta 5

Completada

Puntúa 1.00 sobre 1.00

¿Qué se debe agregar en INSTALLED\_APPS para habilitar el uso de WebSockets con Django Channels?

- ☐ a. 'django.contrib.sessions'
- ☐ b. 'gunicorn'
- ☒ c. 'channels'
- ☐ d. 'rest\_framework'

## Pregunta 6

Completada

Puntúa 1.00 sobre 1.00

¿Qué ocurre si se lanza un error dentro de receive() sin manejarlo?

- ☐ a. La conexión WebSocket se mantiene, pero el cliente no recibe datos
- ☐ b. Se activa la callback on\_error() para manejos personalizados
- ☒ c. Podría cerrar la conexión y se llama disconnect() con un código de cierre
- ☐ d. El consumer ignora el error y continua

## Pregunta 7

Completada

Puntúa 1.00 sobre 1.00

¿Qué configuración se requiere para utilizar Redis u otro backend en la capa de canales (Channel Layer)?

- ☐ a. Configurar `STATIC_ROOT` en `settings.py`
- ☒ b. Definir `CHANNEL_LAYERS` con el backend y los hosts
- ☐ c. Agregar `'django.middleware.csrf.CsrfViewMiddleware'` en `MIDDLEWARE`
- ☐ d. Usar el archivo `wsgi.py` en lugar de `asgi.py`

## Pregunta 8

Completada

Puntúa 1.00 sobre 1.00

Al iniciar la conexión con un WebSocket en el navegador, ¿qué evento se produce cuando se establece la comunicación correctamente?

- ☐ a. `onmessage`
- ☒ b. `onopen`
- ☐ c. `onclose`
- ☐ d. `onerror`

## Pregunta 9

Completada

Puntúa 1.00 sobre 1.00

Al usar Channels, además de `'channels'` en `INSTALLED_APPS`, ¿cuál es un paso clave para que Django reconozca la aplicación asíncrona?

- ☐ a. Usar el comando `python manage.py collectstatic`
- ☒ b. Asignar la ruta de ASGI en `ASGI_APPLICATION` (por ejemplo, `'mi_proyecto.asgi.application'`)
- ☐ c. Eliminar `'django.contrib.admin'` de `INSTALLED_APPS`
- ☐ d. Configurar la variable `DEBUG` en `False`

## Pregunta 10

Completada

Puntúa 1.00 sobre 1.00

¿Qué beneficio ofrece ASGI en la práctica?

- ☒ a. Permite escalabilidad y manejo de miles de conexiones concurrentes
- ☐ b. Limita las aplicaciones a un solo proceso
- ☐ c. Hace que la comunicación sea más lenta en tiempo real
- ☐ d. Dificulta manejar miles de conexiones simultáneas

## Pregunta 11

Completada

Puntúa 1.00 sobre 1.00

¿Por qué ASGI promueve la flexibilidad en aplicaciones Python modernas?

- ☒ a. Porque facilita la integración de varias tecnologías (WebSockets, HTTP2, etc.)
- ☐ b. Porque requiere cerrar la conexión en cada petición
- ☐ c. Porque funciona solo con un único protocolo, HTTP
- ☐ d. Porque elimina por completo la necesidad de usar un framework

## Pregunta 12

Completada

Puntúa 1.00 sobre 1.00

Cuando se invoca `socket.close()`, ¿qué evento se espera que ocurra tras cerrar la conexión?

- ☐ a. `onrelease`
- ☐ b. `onstop`
- ☒ c. `onclose`
- ☐ d. `onfinish`

## Pregunta 13

Completada

Puntúa 1.00 sobre 1.00

¿Qué propiedad o variable indica el estado de un WebSocket en un cliente JavaScript?

- ☒ a. `socket.readyState`
- ☐ b. `socket.state`
- ☐ c. `socket.connectionStatus`
- ☐ d. `socket.mode`

## Pregunta 14

Completada

Puntúa 1.00 sobre 1.00

¿Cuál de los siguientes NO es un valor posible de la propiedad `readyState` en un WebSocket del lado del cliente?

- ☐ a. 2 (CLOSING)
- ☐ b. 0 (CONNECTING)
- ☐ c. 1 (OPEN)
- ☒ d. 4 (EXPIRED)

## Pregunta 15

Completada

Puntúa 1.00 sobre 1.00

¿Qué instrucción se debe ejecutar dentro del método `connect()` para aceptar de forma explícita la conexión?

- ☐ a. `await self.open()`
- ☒ b. `await self.accept()`
- ☐ c. `self.accept_connection()`
- ☐ d. `self.conectar()`

## Pregunta 16

Completada

Puntúa 1.00 sobre 1.00

¿Qué método se llama al establecerse la conexión WebSocket en un AsyncWebsocketConsumer?

- ☐ a. receive()
- ☒ b. connect()
- ☐ c. accept()
- ☐ d. disconnect()

## Pregunta 17

Completada

Puntúa 1.00 sobre 1.00

¿Qué ocurre si no se llama a await self.accept() dentro del método connect()?

- ☐ a. Se envía un mensaje de error al cliente
- ☐ b. Se produce un error en el método disconnect()
- ☐ c. Se inicia la comunicación aunque sea de solo lectura
- ☒ d. El WebSocket se cierra automáticamente sin establecer la comunicación

## Pregunta 18

Completada

Puntúa 1.00 sobre 1.00

¿Cómo podemos cerrar de manera asíncrona el WebSocket desde el servidor?

- ☐ a. await self.finish()
- ☐ b. self.disconnect\_force()
- ☒ c. await self.close()
- ☐ d. self.close\_connection()

## Pregunta 19

Completada

Puntúa 1.00 sobre 1.00

¿Por qué se suele llamar 'django.setup()' al inicio de asgi.py?

- ☐ a. Para evitar el uso de ProtocolTypeRouter
- ☐ b. Para configurar la base de datos a nivel de shell
- ☐ c. Para añadir automáticamente las rutas de URL
- ☒ d. Para inicializar Django e informar a la aplicación de las settings antes de cargar la aplicación ASGI

## Pregunta 20

Completada

Puntúa 1.00 sobre 1.00

¿Cuál de las siguientes opciones describe mejor qué es un WebSocket?

Seleccione una:

- ☐ a. Una variante de SSE (Server-Sent Events) que requiere abrir conexiones constantemente.
- ☒ b. Un protocolo que permite conexiones bidireccionales y persistentes entre cliente y servidor.
- ☐ c. Una tecnología que no soporta intercambio de datos en tiempo real.
- ☐ d. Un simple patrón de request-response igual que HTTP, pero con menos cabeceras.

## Pregunta 21

Completada

Puntúa 1.00 sobre 1.00

A diferencia de HTTP tradicional, un WebSocket:

Seleccione una:

- ☐ a. No permite uso de canales full-duplex (solo unidireccional).
- ☒ b. Evita la apertura y cierre repetidos de conexiones para cada mensaje, permitiendo intercambio continuo de datos.
- ☐ c. Se basa en largas colas de peticiones HTTP GET sin respuesta.
- ☐ d. Es idéntico al polling, pero con cabeceras adicionales.

## Pregunta 22

Completada

Puntúa 1.00 sobre 1.00

¿Cuál de estas acciones NO se recomienda para conservar la conexión WebSocket activa y detectar inactividad?

- ☐ a. Realizar reintentos de conexión si onclose indica un cierre inesperado
- ☐ b. Enviar pings o mensajes de keep-alive periódicamente
- ☒ c. Reducir el tamaño de los mensajes de forma drástica
- ☐ d. Manejar onerror para reconectar automáticamente

## Pregunta 23

Completada

Puntúa 1.00 sobre 1.00

¿Cuál de las siguientes opciones describe mejor qué es un WebSocket?

Seleccione una:

- ☐ a. Una tecnología que no soporta intercambio de datos en tiempo real.
- ☐ b. Una variante de SSE (Server-Sent Events) que requiere abrir conexiones constantemente.
- ☐ c. Un simple patrón de request-response igual que HTTP, pero con menos cabeceras.
- ☒ d. Un protocolo que permite conexiones bidireccionales y persistentes entre cliente y servidor.

## Pregunta 24

Completada

Puntúa 1.00 sobre 1.00

A diferencia de HTTP tradicional, un WebSocket:

Seleccione una:

- ☐ a. Es idéntico al polling, pero con cabeceras adicionales.
- ☐ b. Se basa en largas colas de peticiones HTTP GET sin respuesta.
- ☒ c. Evita la apertura y cierre repetidos de conexiones para cada mensaje, permitiendo intercambio continuo de datos.
- ☐ d. No permite uso de canales full-duplex (solo unidireccional).



## Pregunta 25

Completada

Puntúa 1.00 sobre 1.00

Dentro de un AsyncWebsocketConsumer, si se desea manejar un evento llamado "chat\_message", ¿cómo se nombra el método que lo procesará?

- ☐ a. `def chat_message_event(self, event):`
- ☐ b. `async def on_chat_message(self, event):`
- ☒ c. `async def chat_message(self, event):`
- ☐ d. `def chat_message(self):`

## Pregunta 26

Completada

Puntúa 1.00 sobre 1.00

¿Cuál es la forma correcta de asignar un alias de ruta WebSocket a un consumer asíncrono?

- ☒ a. `path("ws/chat/", MiConsumer.as_asgi())`
- ☐ b. `path("ws/chat/", MiConsumer.as_view())`
- ☐ c. `path("ws/chat/", MiConsumer.as_consumer())`
- ☐ d. `path("ws/chat/", MiConsumer)` en `routing.py`

## Pregunta 27

Completada

Puntúa 1.00 sobre 1.00

¿Cuál de las siguientes opciones describe mejor qué es un WebSocket?

Seleccione una:

- ☐ a. Un simple patrón de request-response igual que HTTP, pero con menos cabeceras.
- ☐ b. Una variante de SSE (Server-Sent Events) que requiere abrir conexiones constantemente.
- ☐ c. Una tecnología que no soporta intercambio de datos en tiempo real.
- ☒ d. Un protocolo que permite conexiones bidireccionales y persistentes entre cliente y servidor.

## Pregunta 28

Completada

Puntúa 1.00 sobre 1.00

A diferencia de HTTP tradicional, un WebSocket:

Seleccione una:

- ☐ a. Se basa en largas colas de peticiones HTTP GET sin respuesta.
- ☒ b. Evita la apertura y cierre repetidos de conexiones para cada mensaje, permitiendo intercambio continuo de datos.
- ☐ c. Es idéntico al polling, pero con cabeceras adicionales.
- ☐ d. No permite uso de canales full-duplex (solo unidireccional).

## Pregunta 29

Completada

Puntúa 1.00 sobre 1.00

En un AsyncWebsocketConsumer, ¿cómo se envía un mensaje de texto al cliente?

- ☒ a. `await self.send(text_data=texto)`
- ☐ b. `self.broadcast_message(texto)`
- ☐ c. `await self.send_json({"msg": texto})`
- ☐ d. `await self.send(texto)`

## Pregunta 30

Completada

Puntúa 1.00 sobre 1.00

Selecciona la principal ventaja de usar WebSocket en una aplicación en tiempo real:

Seleccione una:

- ☐ a. No soporta comunicación bidireccional, solo unidireccional desde servidor a cliente.
- ☐ b. Reinicia la conexión cada vez que el servidor necesita enviar un mensaje.
- ☒ c. Permite mantener una sola conexión abierta, reduciendo latencia y sobrecarga de cabeceras.
- ☐ d. Requiere múltiples conexiones simultáneas para enviar y recibir datos.

◀ EXAMEN DE LA UNIDAD 1

Ir a...