Métodos de gestión de sesiones en AtlasNode

1. JWT (JSON Web Tokens) en cookies httpOnly

- Funcionamiento:
  - El servidor genera un token firmado que se guarda en una cookie httpOnly tras el login.
  - Este token contiene los datos del usuario (como ID, rol, expiración) y se envía automáticamente en cada petición.

- Ventajas:
  - Rápido y sin necesidad de consultar la base de datos.
  - No depende del almacenamiento en el servidor (es independiente del estado del servidor).
  - Ideal para entornos serverless.

- Desventajas:
  - No se puede invalidar un token una vez emitido.
  - No permite gestionar múltiples sesiones activas.
  - Poca trazabilidad.

2. Tabla de sesiones (session tokens persistentes)

- Funcionamiento:
  - Se genera un token único aleatorio por sesión y se guarda en una tabla `Session` asociada al usuario.
  - Se almacena una cookie con ese token. En cada petición se consulta la tabla para validar.

- Ventajas:
  - Se pueden revocar sesiones individualmente.
  - Permite saber cuántas sesiones tiene un usuario, desde qué IP, etc.
  - Ideal para control administrativo.

- Desventajas:
  - Requiere hacer una consulta a la base de datos en cada petición.
  - Depende del almacenamiento en el servidor para funcionar correctamente.

3. Enfoque híbrido: JWT + Tabla de sesiones

- Funcionamiento:
  - Se genera un JWT y se guarda en cookie, y además se registra en una tabla `Session`.
  - En cada request se valida el JWT y se consulta la existencia de la sesión en la tabla.

- Ventajas:
  - Combina lo mejor de ambos mundos: velocidad + control.
  - Se pueden forzar cierres de sesión.

- Se pueden consultar logs y gestionar dispositivos activos.
- Compatible con arquitectura actual de AtlasNode.

- Desventajas:
  - Requiere algo más de infraestructura de validación (BD + verificación del token).
  - No es completamente independiente del estado del servidor.

Tabla comparativa

Característica	JWT solo	Tabla de sesiones	JWT + sesiones
Autenticación rápida sin BD	✅	❌	✅
Revocar sesión manualmente	❌	✅	✅
Auditoría y trazabilidad de sesiones	❌	✅	✅
Control de múltiples sesiones activas	❌	✅	✅
Logout forzado desde el panel admin	❌	✅	✅
Protección por rol directamente en token	✅	❌	✅
Requiere acceso a BD por request	❌	✅	✅
Compatible con serverless	✅	⚠️	⚠️