


.NET Core - ACCENTURE- 26MAY21

[Área personal](#)[Mis cursos](#)[.NET-ACC-26MAY21](#)[Actividad 9 - Implementar disparar salvos](#)[Avisos](#)

ACTIVIDAD 9 - IMPLEMENTAR DISPARAR SALVOS

[Su progreso ?](#) [Video aclaraciones actividad 9](#) [Video actividad 9](#) [email task9](#) [web-mostrar-juego-4](#)

Implementar disparar salvos

Esta actividad es muy parecida a la anterior ya que la estructura de los datos es similar y las reglas también. La única cosa extra es el seguimiento del turno, si es el turno # 4, no quieres que un jugador pueda cambiar el historial construyendo y enviando un nuevo salvo para el turno # 3. Se debe tener en cuenta que a diferencia de posicionar los barcos en el que se recibe un listado acá solo se recibirá un salvo.

Se debe Implementar un método en el controlador GamePlayersController que pueda recibir una lista de objetos de salvos, con ubicaciones, guardarlos en la base de datos y devolver el estado 201 Created si no hay problemas.

- La url deberá ser `/api/gamePlayers/{id}/salvos`
- El cuerpo de la solicitud debe poder transformarse en una lista de Salvos
- Se debe responder con código de estado 403 Forbidden si:
 - No existe el gamePlayer con el id indicado, "No existe el juego"
 - El usuario autenticado no se encuentra en el juego al que se quieren agregar los barcos, "El usuario no se encuentra en el juego"
 - Si el jugador intenta adelantarse al turno: "No se puede adelantar el turno", es decir, un jugador dispara un salvo por lo que se encuentra en el turno 1, el oponente se encontraría en el turno 0, si el jugador intenta disparar de nuevo quedaría en el turno 2 cuando el oponente aún no se encuentra en el turno 1, considerar quien inicio el juego.
- Si no existen problemas insertar los salvos en el objeto gamePlayer del id indicado y actualizar el objeto gamePlayer.

Importante: recuerda modificar el método `findById` para incluir los salvos.

Nota: para ver si un player se está adelantando en turno hay que saber el turno del oponente por lo que una buena idea sería tener un método en la clase GamePlayer que retorne al oponente (si existe) llamado `GetOpponet()`, ya que el **oponente** de un GamePlayer **es uno de los GamePlayers de la entidad Game del GamePlayer en cuestión o_O** (si haces ésto necesitaras modificar el método `findBy` para incluir al oponente, es decir incluir el Game con sus GamePlayers).

Para probar el controller se puede usar:

```
axios.post("https://localhost:5001/api/gamePlayers/21/salvos", [{id:0,turn:0, locations: [{id:0, location: 'A1'}, {id:0, location: 'A2'}, {id:0, location: 'A3'}, {id:0, location: 'A4'}, {id:0, location: 'A5'}] }]).then(response=> {console.log("salvos disparados!!")}).catch(error => {console.log("error, código de estatus: " + error.response.status + " mensaje: " + error.response.data)});
```

-



Al terminar descargar el paquete [web-mostrar-juego-4](#) y copiar su contenido en la carpeta wwwroot. Una vez copiado el contenido en la carpeta ejecutar el proyecto y colocar en el navegador la url *<https://localhost:5001/index.html>*, una vez allí iniciar sesión ir a un juego y se podrán posicionar los salvos, probar el disparar varios salvos haciendo click sobre la celda del tablero de salvo.



Actividad 8 - Implementar el posicionar barcos

Actividad 10 – Implementar golpeados y hundidos



Ir a...



MindHub

📁 Resumen de retención de datos

[Reiniciar tour para usuario en esta página](#)