

Cuestionario a responder:

1. En el método Crear de la clase JugadorService, ¿por qué se utiliza SCOPE\_IDENTITY() en la consulta SQL y qué beneficio aporta al código?

**R// Se utiliza para poder darle una identidad por así decirlo a cada uno de los usuarios creados ya que gracias al identity se puede que el programa me devuelva automáticamente el número mayor al valor anterior**

2. En el método Eliminar del servicio de jugadores, ¿por qué se verifica la existencia de elementos en el inventario antes de eliminar un jugador y qué problema está previniendo esta comprobación?

**R// Lo que se está previniendo es el que queden por así decirlo quedar datos basura o que ya no sirvan para nada en el programa y este no ocupe espacio por así decirlo que se pueda aprovechar en otro usuario o programa y evitar la saturación**

3. ¿Qué ventaja ofrece la línea using var connection = \_dbManager.GetConnection(); frente a crear y cerrar la conexión manualmente? Menciona un posible problema que podría ocurrir si no se usara esta estructura.

**R// De las cosas más fundamentales que aprendí acerca de esto es que gracias a su funcionalidad no tenemos que estar con la preocupación de estar poniendo connection close y evitarnos el riesgo de olvidarlo y pues este garantiza el cierre automático de este bloque**

4. En la clase DatabaseManager, ¿por qué la variable \_connectionString está marcada como readonly y qué implicaciones tendría para la seguridad si no tuviera este modificador?

**R// Este nos ayuda a evitar generar una manipulación sin querer a este connection me da a entender de que si este no se declara como readonly estaría en la posibilidad de que sin querer podría modificarla y este readonly lo evita.**

5. Si quisieras agregar un sistema de logros para los jugadores, ¿qué cambios realizarías en el modelo de datos actual y qué nuevos métodos deberías implementar en los servicios existentes?

**R//Podría meter un cronometro virtual en el cual recompensara la horas en la que el usuario pasara en el sistema por asi decirlo y sobre todo metería algún tipo de objetivos en el cual beneficie de alguna manera al jugador**

6. ¿Qué sucede con la conexión a la base de datos cuando ocurre una excepción dentro de un bloque using como el que se utiliza en los métodos del JugadorService?

**R//Lo que logramos con esto es que lo que mencionaba anteriormente que este por el using la conexión a la base de datos de cierra automáticamente sin la fatiga de estar poniendo connetion close o que se nos olvide y que se libere automáticamente al terminar este bloque.**

7. En el método ObtenerTodos() del JugadorService, ¿qué ocurre si la consulta SQL no devuelve ningún jugador? ¿Devuelve null o una lista vacía? ¿Por qué crees que se diseñó de esta manera?

**R// Lo comprobé y esta devuelve una lista vacía, porque en mi opinión que me devuelva un valor nulo o que me de este tipo de referencias puedo a llegar a intuir que mi código esta malo mientras es todo lo contrario simplemente no habían datos ak momento de presionar ese botón, mientras que si coloco que me ponga un mensaje de la lista esta vacía esa me da perfectamente la referencia de que no hay datos ingresados.**

8. Si necesitaras implementar una funcionalidad para registrar el tiempo jugado por cada jugador, ¿qué cambios harías en la clase Jugador y cómo modificarías los métodos del servicio para mantener actualizada esta información?

**R//Me di la tarea de investigar y visual contiene un comando por asi decirlo parecido al datetime que utilizamos para poder impregnar la fecha la cual se creo el usuario este es "TimeSpan" este registra todo tipo de tiempo en el sistema desde segundos a horas agregaría una nueva columna para almacenar el tiempo con el tipo de dato "TIME" esto me obligaría a actualizar un gran parte de métodos en el programa o al menos todos los que con lleva jugador.**

9. En el método TestConnection() de la clase DatabaseManager, ¿qué propósito cumple el bloque try-catch y por qué es importante devolver un valor booleano en lugar de simplemente lanzar la excepción?

**R//El propósito se ve simple por simpl hecho de que como su bloque lo dice Test ósea esta comprobando que la base de datos este conectada correctamente es como decir en pocas palabras que nos averigüe si en verdad estamos dentro de la BD para poder pasar de bloque y bueno considero de que se usa valo booleano por el simple hecho de que solo hay dos opciones para esto que este o no este conectado lo cual el valor true o false nos simplificaran varias excepciones**

10. Si observas el patrón de diseño utilizado en este proyecto, ¿por qué crees que se separaron las clases en carpetas como Models, Services y Utils? ¿Qué ventajas ofrece esta estructura para el mantenimiento y evolución del proyecto?

**R//Porque ofrece una capacidad de atención y orden a la hora de manejar el proyecto y el saber de alguna u otra mandera en donde van estar ciertas partes de código del programa. Las ventajas que esto puede traer de primeras obviamente lo visual y lo fresco que se ve al momento de pasar por varios usuarios el código estos se pueden avocar a la carpeta que mas lógica encuentren en base a bloque de código que necesitan ver y al momento de que una persona que no estuvo en sus inicios del código al hacerle mantenimiento no se le haga mas complicado.**

11. En la clase InventarioService, cuando se llama el método AgregarItem, ¿por qué es necesario usar una transacción SQL? ¿Qué

problemas podría causar si no se implementara una transacción en este caso?

**R//Es necesario para que todos los datos que agreguemos vayan de la mano, los problemas que nos podrían causar seria el hecho de agregar algo y que este por x razón de error pero las demás cosas que agregamos no hubo ningún problema el problema aquí es que si alguno da error y los demás no y aun asi se aguarde ya no tendría coherencia la información agregada y podría causar problemas al momento de querer exportar o alguna u otra cosa.**

12. Observa el constructor de JugadorService: ¿Por qué recibe un DatabaseManager como parámetro en lugar de crearlo internamente? ¿Qué patrón de diseño se está aplicando y qué ventajas proporciona?

**R//Pues si varios bloques necesitan de dt se puede crear una sola instancia para poder reutilizar en varios bloques de código y asi evitar la saturación y asi reducir el consumo de recursos. Pues investigando un poco el patron en el que se diseño se llama inyección de dependencias que consiste en proporcionar dependencias de una clase desde el exterior en lugar que la clase las cree por si misma y en nuestro caso facilitar en el proyecto el caso de migración.**

13. En el método ObtenerPorId de JugadorService, ¿qué ocurre cuando se busca un ID que no existe en la base de datos? ¿Cuál podría ser una forma alternativa de manejar esta situación?

**R// Pues al momento de ejecutar y no encuentre ningún registro el reader devuelve un false que el bloque if no se ejecuta en este caso al final del método retorna un null como se le designo que es un valor predeterminado. Pues la solución que yo pondría es la misma que la del bloque de obtener un jugador ya que como lo mencionábamos el null puede ser efectivo pero al momento de que nos de error podría llegarnos a confundir entonces lo que haría seria agregar un método en el cual si no se encuentra la ID del jugador poner un mensaje tipo jugador no encontrado.**

14. Si necesitas implementar un sistema de "amigos" donde los jugadores puedan conectarse entre sí, ¿cómo modificarías el modelo de datos y qué nuevos métodos agregarías a los servicios existentes?

**R// Pues agregaría una tabla nueva en la base de datos en donde se obtenagon los datos de en este caso 2 jugadores ya que es la lógica y pues en el bloque de jugador debajo de date time agregaría una lista que represente esta lista de amigos al que le agregaría el método de agregar o enviar solicitud de amitstas y eliminar de igual manera al final agregaría el método de inventario pero adaptado a este en el cual al de la misma manera en que me muestra los usuarios o el inventario un data grid en el cual me muestre a mis amigos por asi decirlo.**

15. En la implementación actual del proyecto, ¿cómo se maneja la fecha de creación de un jugador? ¿Se establece desde el código o se delega esta responsabilidad a la base de datos? ¿Cuáles son las ventajas del enfoque utilizado?

**R// Pues es un dato que me llamo la atención a primera vista ya que para eso se uso de date time que hace el trabajo por nosotros de delegar la hora en la que se dio click para crear al jugador, si esta responsabilidad esta delegada en la base de datos.**

16. ¿Por qué en el método getConnection() de DatabaseManager se crea una nueva instancia de SqlConnection cada vez en lugar de reutilizar una conexión existente? ¿Qué implicaciones tendría para el rendimiento y la concurrencia?

**R//Pues por lo que entiendo es para no reutilizar varias instancias de sql connetion ya que esto puede causar confusiones y problemas ya no esta diseñado para ser compartido en múltiples bloques y pues investigando me di cuenta de que cuando se crea un instancia no se establece una nueva conexión física en la base de datos esto minimiza el costo de abrir y cerrar las conexiones**

17. Cuando se actualiza un recurso en el inventario, ¿qué ocurriría si dos usuarios intentan modificar el mismo recurso simultáneamente? ¿Cómo podrías mejorar el código para manejar este escenario?

**R//Pues se podrían sobre escribir los datos de uno con el otro haciendo que uno que ya lo hizo de primero se quede en 0 por asi decirlo agregari un bloque en el cual este enfocado en mientras la operaciones realiza se bloquee par evitar que otros usuarios lo modifiquen como quien dice mientras la este y quien realice de**

**primero la modificación se quedaría con el cambio mientras que al otro mostrar el un mensaje como erro al cambio o como a veces se utiliza en lo juegos con el tema de los nombres de usuario ya usados.**

18. En el método Actualizar de JugadorService, ¿por qué es importante verificar el valor de rowsAffected después de ejecutar la consulta?  
¿Qué información adicional proporciona al usuario?

**R// El valor rows nos ayudar al valor devuelto por el execute este representa el numero de filas que fueron modificadas o eliminadas por la consulta sql en este caso rows no indica cuantos registro de la tabla fueron actualizados en pocas palabras Si rowsAffected > 0, significa que al menos una fila fue actualizada, lo que indica que el jugador con el ID especificado existía en la base de datos y la operación fue exitosa. Si rowsAffected == 0, significa que no se encontró ningún registro con el ID proporcionado, por lo que no se realizó ninguna actualización.**

19. Si quisieras implementar un sistema de registro (logging) para seguir todas las operaciones realizadas en la base de datos, ¿dónde colocarías este código y cómo lo implementarías para afectar mínimamente la estructura actual?

**R//Yo la crearía en la el apartado de database manager ya que esta clase centraliza la creación de conexiones en el sql esta me permite registra todas las operaciones sin modificar directamente las clases de las carpetas y así no es necesario modificar cada método.**

20. Observa cómo se maneja la relación entre jugadores e inventario en el proyecto. Si necesitaras agregar una nueva entidad "Mundo" donde cada jugador puede existir en múltiples mundos, ¿cómo modificarías el esquema de la base de datos y la estructura del código para implementar esta funcionalidad?

**R// Pues considerando que ya ahí el proyecto va en otro enfoque cambiaría por completo las tablas como la del todo el código se tendría que agregar algún data grid en el cual muestre los mundos una interfaz en donde se puedan crear los mundos modificar el**

**esquema a base de gustos quizás el tener mundos ya creados y que los usuarios decidan en cual ellos quieren también en ese caso se crearía un nuevo modelo o carpeta para poder manejar por así decirlo el servicio y el control de los mundos y sin duda cambiar toda la interfza grafica-**

**21. ¿Qué es un SqlConnection y cómo se usa?**

**R//YO sql connection lo asimilo como un puente como la manera de establecer la conexión de visual a mi base de datos de SQL y es el punto de entrada par poder interactuar con la base de datos y que nos permita la conexión como colocar comandos en nuestros códigos y que funciones en sql como la de insert to o delete y por lo que por así decirlo también se encarga de abrir y cerrar esta conexión**

**22. ¿Para qué sirven los SqlParameter?**

**R// se utilizar par representar parámetros dentro de una consulta de sql como un comando ejecutado para realizar o modificar en la base de datos y este nos ayuda a manejar los tipso de datos correctamente.**