**Evaluación - Desarrollador de Software**

Hola, continuamos con el proceso de selección realizamos esta prueba para conocer más sobre tus skills y tu talento.

**Bloque 1: Responde las siguientes preguntas:**

1. ¿Puedes encontrar el (los) error(es) en este código? Y cómo lo corregirías (Agregar imagen del código)

***<?php***

*// se desea conectar a un servidor mysql en la IP 10.20.30.40*

*// desde el puerto 3306 e imprimir los codigos en una lista de html*

*// el usuario es user, la clave key y la base de datos testdb*

*// y la tabla laptops*

ini\_set('display\_errors', 0);

ini\_set('display\_startup\_errors', 0);

error\_reporting(0);

$mysqli **=** **new** *mysqli*("localhost", "testdb", "user", "key");

$mysqli**->**set\_charset('utf-8');

$mysqli**->**query('SET timezone = "-05:00"');

$tabla **=** "laptops"

$qry **=** $mysqli**->**query('SELECT **\***.codigo FROM $tabla');

**while**($row **==** $qry**->**fetch\_row())

{

    $lista **.=** "<li>$row['codigo']</li>";

}

echo "</ul>";

1. ¿Qué consideraciones tienes en cuenta para proteger un proyecto desarrollado de ciberataques?
2. Si fueses el líder técnico de un proyecto, describe como harías el despliegue en la nube de un desarrollo en Laravel (Sistema de inventarios), puedes usar los servicios cloud que consideres necesarios y/o modificar el proyecto para que se ajuste a estos servicios.
3. Si fueses líder técnico cómo abordarías el proyecto (Sistema de inventarios) desde la ingeniería de requisitos.

**Bloque 2: Establecer el diagrama relacional de base de datos y codificar en SQL:**

En este siguiente caso, las empresas alquilan diferentes cantidades de equipos y cada empresa es atendida por un asesor comercial, usted deberá implementar las columnas adicionales que considere necesarias para el diagrama y redactar la sentencias SQL (para la creación de las tablas, relaciones y vista de reporte)

Ver excel “Datos prueba técnica – DFSS”

**Bloque 3: Demo Técnica:**

Desarrollar una aplicación web usando PHP o Laravel en el backend y Vue o React en frontend para hacer consultas por Cliente (qué equipos tiene) y mostrarlo en una tabla. (Usar la base de datos anterior)

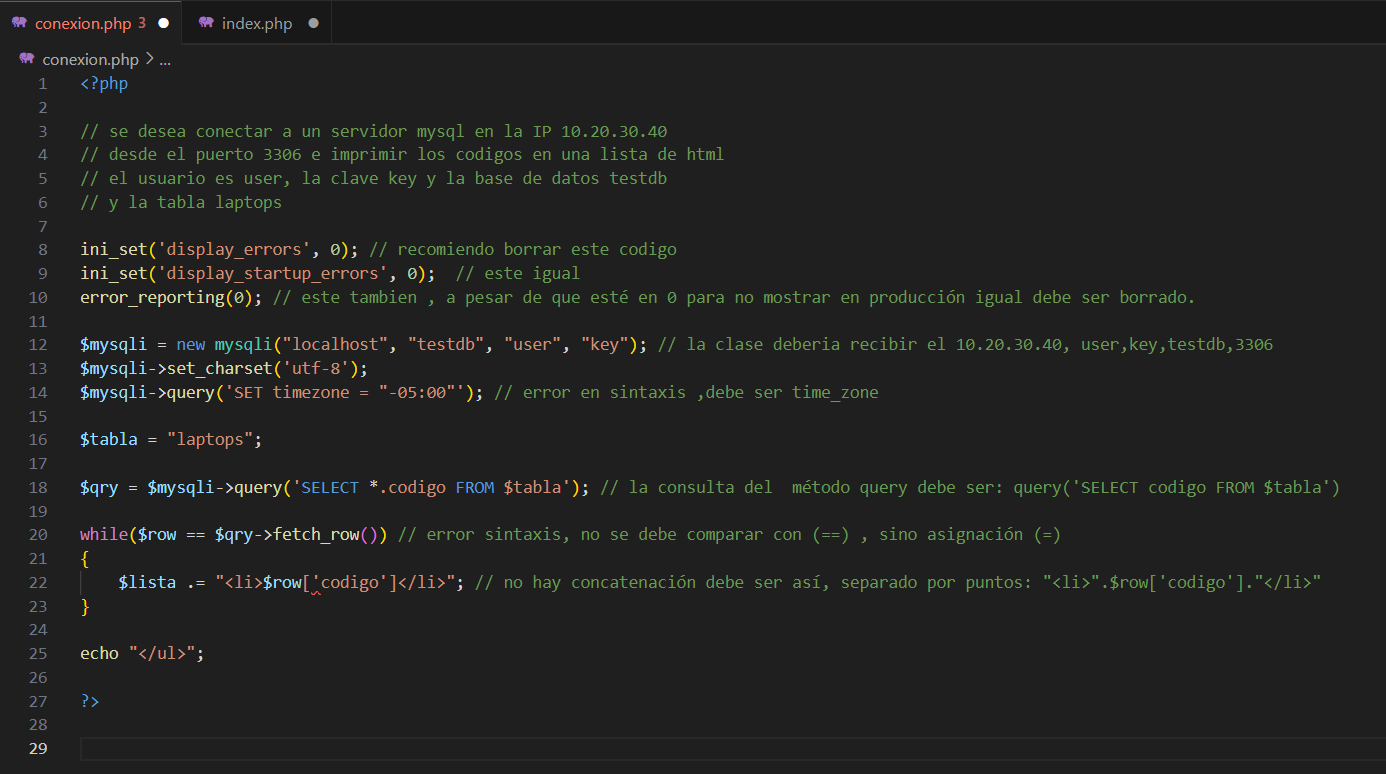
El diseño del tabla y buscador, queda a su criterio, siempre que se usen las tecnologías mencionadas.

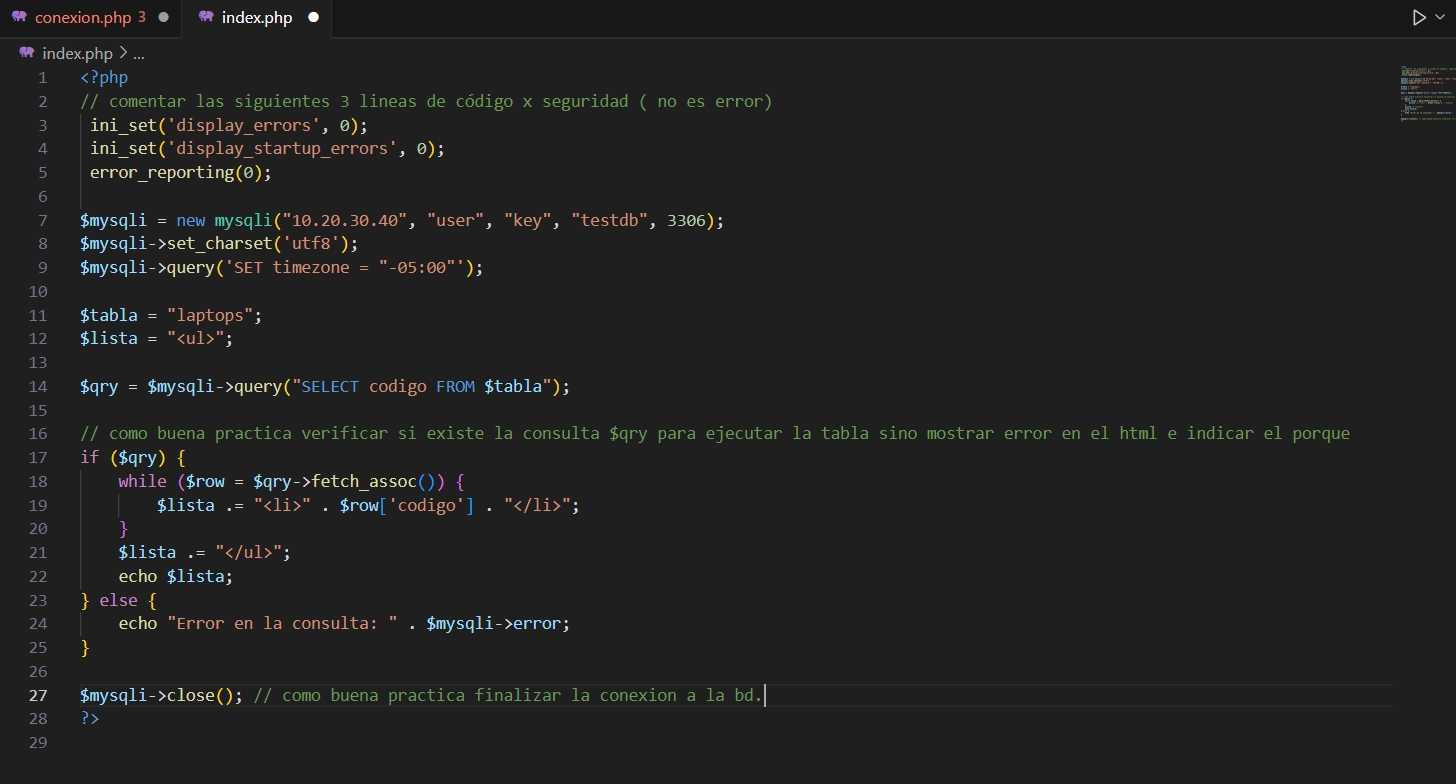
El código del desarrollo tendrá que ser subido a un repositorio público para su revisión, se tendrá en cuenta buenas prácticas y uso de recursos actuales y deberá enviarse el link.

**RESULTADOS:**

**1 . BLOQUE:**

**1.** Errores encontrados

****

Errores resueltos

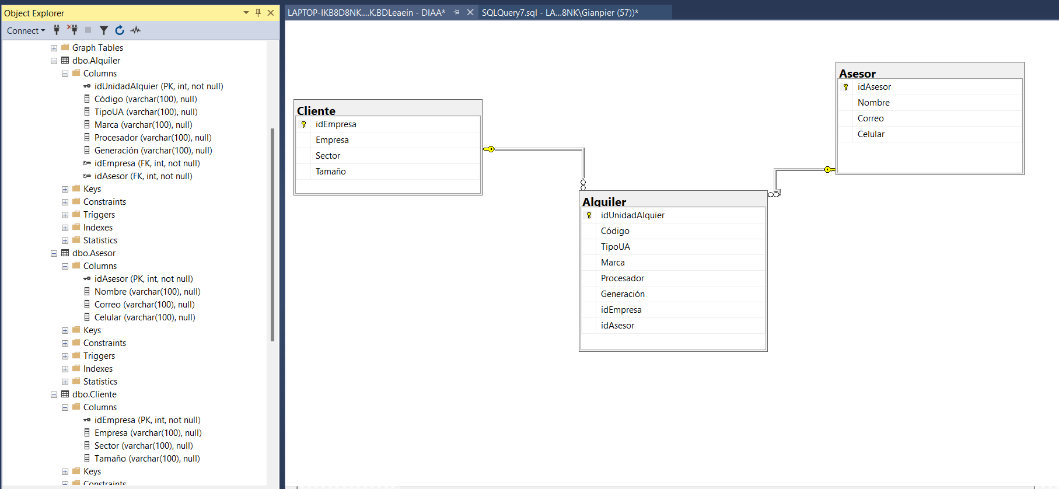
2. Ante un ataque el proyecto web debe tener filtros en los formularios o en los inputs para que la persona atacante no le permita ingresar scripts o querys que afecten a la base de datos. Por eso es importante el máximo de caracteres de cada input. Es sumamente importante que las contraseñas se envíen encriptadas a la bd y que una vez recibida la bd la desencripte para la validación y luego encripte nuevamente (seguridad).

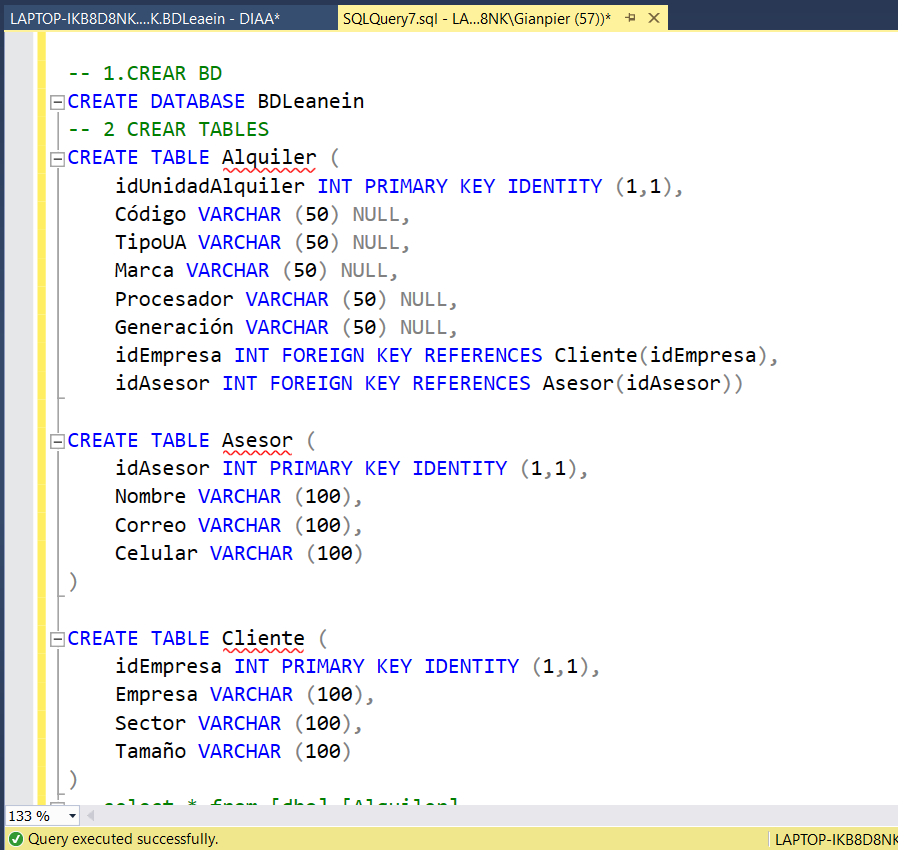
Usar procedimientos almacenados en vez de consultas directas a la bd es prioridad para la seguridad de los datos . Los métodos post y get si es posible deben tener token y evitar que la persona atacante haga consultas por la url sin antes haberse autenticado. En el FRONT proteger las rutas a usuarios autorizados , es importante tener en la bd una tabla de acceso para establecer que usuario tiene ciertos permisos en las rutas o módulos (URL). Y por último recomendaría como buena práctica que las cookies, localstorage o sessionStorage esten encriptados para el usuario.

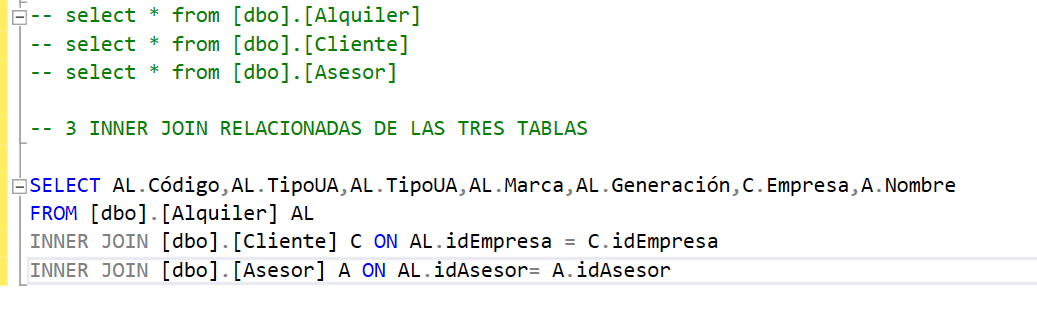
3. En sí todo va a depender de los requerimientos que quiera el cliente o la empresa, es prioritario realizar un análisis general de que tecnologías se usarán y armarlo en un prototipo visual tanto de la bd, front,back y arquitectura cloud, ya después se ve que proveedor de nube se va a utilizar AWS, AZURE , Google Cloud , etc. Yo preferible recomendaría Azure ya que tiene un servicio llamado Azure Devops en el cual conozco como se maneja los repositorios muy similar a github, en Azure se puede usar casi todos los tipos de base de datos MYSQL , SQL, POSTGRESQL , ETC. En realidad es una elección, si es que es una implementación que se hace desde 0 , pero si es una mejora de algún sistema ya hecho entonces se debería verificar a que base de datos apunta la conexión y configurar los ips , querys, puertos , etc. Si el sistema de inventario está hecho de laravel, recomendaría usar apis protegidas con tokens en cada solicitud que se envian (POST,GET,DELETE,UPDATE,PUT). Con AzureDevops se puede gestionar los avances del equipo de desarrollo, usando ramas, clonaciones por consola, cambios , etc.

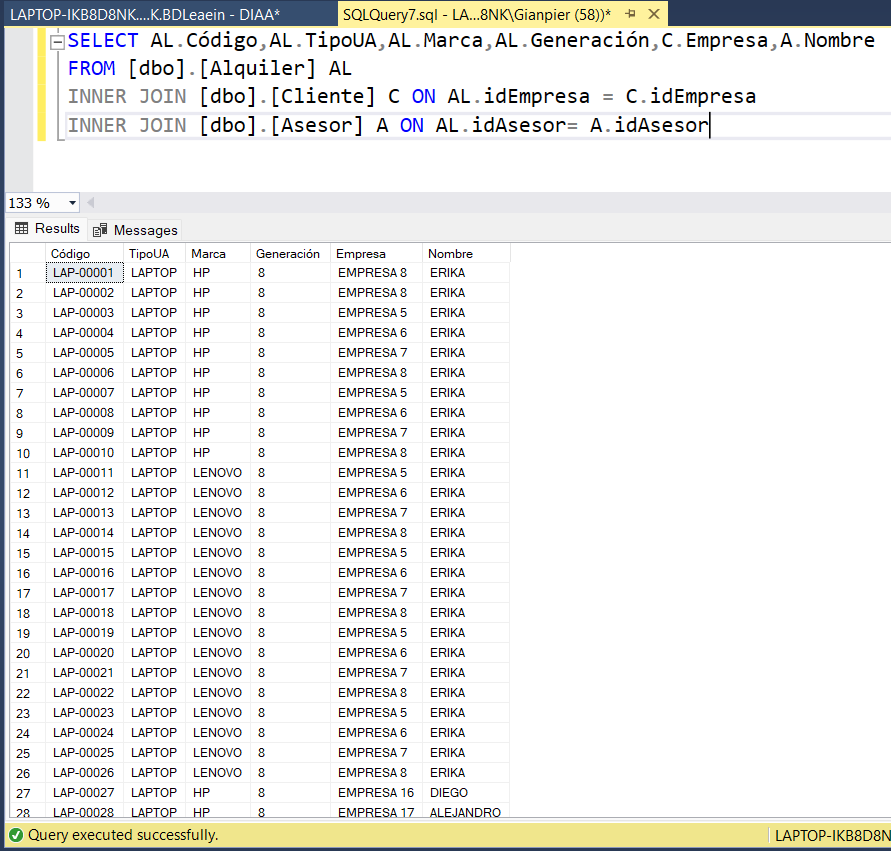
4. Muy importante conocer bien el modelo de negocio del cliente o empresa y el alcance del proyecto a implementar, recomendaría que se haga una documentación de los requerimientos del sistema de inventario, un ejemplo muy simple: login (usuario y contraseña), roles(usuarios,permisos), productos(stock,categoría), ventas(productos,cliente,usuario), etc. Con documentación escrita y bien hecha el equipo de desarrollo tendrá idea del alcance del sistema . Otro punto, sería tener estructurado las responsabilidades de las áreas del front y del back . Comprometer al equipo de desarrollo a las reuniones con los stakeholders ayuda. Finalmente, los testing o pruebas de desarrollo es crítico de realizar para la presentación final o subida a producción.

BLOQUE 2:









BLOQUE 3: https://react-prueba.gianpierovasquez.cloud/