|  |
| --- |
| **Evaluación y Prueba Técnica: Node JS, Typescript y Oracle** |

**Evaluación Técnica:**

1. ¿Qué es TypeScript y cuáles son sus principales ventajas sobre JavaScript estándar?  
   Es un lenguaje de programación que se basa en javascript, es robusto y flexible. Sus ventajas sobre javascript sería que ayuda a detectar errores a tiempo, es intuitivo, tiene una escritura estática y que typescript le da soporte a los módulos.
2. Explica las ventajas de utilizar un ORM (Object-Relational Mapping) como TypeORM o Sequalize en lugar de consultas SQL directas.

Simplifica el desarrollo, da un mejor mantenimiento al código, es más claro, aporta mayor seguridad, realiza mapeos entre las estructuras de los datos y previene la inyección SQL.

1. ¿Explica qué es la programación asincrónica y cómo se implementa en Node JS?

La programación asincrónica se basa en la ejecución de tareas y no espera a que se ejecute uno tras otro para que pueda continuar.

Para implementar en Node Js se implementan 3 métodos tales como: las promesas, los callbacks y el async/await.

**Prueba Técnica:**

En esta prueba técnica, se solicita crear las API REST utilizando Node.js y Oracle para gestionar la información de seguros. Deberás diseñar y desarrollar una serie de endpoints para realizar operaciones CRUD (crear, leer, actualizar y eliminar) sobre las tablas proporcionadas a continuación:

**Tabla de Coberturas:**

|  |  |  |
| --- | --- | --- |
| **ID\_Cobertura** | **Nombre\_Cobertura** | **Descripcion** |
| 1 | Seguro de Vida | Cobertura que protege en caso de fallecimiento. |
| 2 | Seguro de Salud | Cobertura médica para gastos de salud. |
| 3 | Seguro de Automóvil | Cobertura para daños a vehículos y terceros. |

**Tabla de Seguros:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID\_Seguro** | **ID\_Cobertura** | **Nombre\_Seguro** | **Descripcion** | **Precio** | **Moneda** | **Duracion** |
| 1 | 1 | Vida Total | Cobertura completa de seguro de vida. | 500 | PEN | Anual |
| 2 | 2 | Salud Básico | Cobertura médica básica. | 200 | USD | Mensual |
| 3 | 3 | Auto Terceros | Cobertura para daños a terceros. | 300 | PEN | Anual |

**Tabla de Clientes:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID\_Cliente** | **Nombre\_Cliente** | **Direccion** | **Correo\_Electronico** | **Telefono** | **Fecha\_Nacimiento** | **Sexo** |
| 1 | Juan Pérez | Calle 123, Lima | juan.perez@example.com | 999888777 | 1985-03-15 | M |
| 2 | María Gómez | Jr. Paz 451, Tacna | maria.gomez@example.com | 999999888 | 1990-11-22 | F |
| 3 | Carlos López | Calle 8, Ica | carlos.lopez@example.com | 999888999 | 1988-07-10 | M |

**Tabla de Pólizas:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID\_Poliza** | **ID\_Cliente** | **ID\_Seguro** | **Fecha\_Inicio** | **Fecha\_Vencimiento** | **Estado de la Póliza** |
| 1 | 1 | 1 | 2023-01-01 | 2024-01-01 | Activo |
| 2 | 2 | 2 | 2023-02-15 | 2023-03-15 | Activo |
| 3 | 3 | 3 | 2023-03-10 | 2024-03-10 | Activo |

**Tabla de Pagos:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID\_Pago** | **ID\_Poliza** | **Fecha\_Pago** | **Total\_ Pago** | **Moneda** |
| 1 | 1 | 2023-01-05 | 500 | PEN |
| 2 | 2 | 2023-02-20 | 200 | USD |
| 3 | 3 | 2023-03-12 | 300 | PEN |

**Tabla de Pagos Detalle:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID\_Detalle\_Pago** | **ID\_Pago** | **Monto\_Pago\_Detalle** | **Metodo\_Pago** | **Moneda** |
| 1 | 1 | 280 | Tarjeta de Crédito | PEN |
| 2 | 1 | 220 | Transferencia Bancaria | PEN |
| 3 | 2 | 110 | Tarjeta de Crédito | USD |
| 4 | 2 | 90 | Tarjeta de Débito | USD |
| 5 | 3 | 300 | Tarjeta de Crédito | PEN |

**Donde:**

**Tabla de Coberturas:**

Esta tabla almacena información sobre los diferentes tipos de coberturas que ofrece la compañía de seguros. Cada registro representaría una cobertura específica, como seguro de vida, seguro de salud, seguro de automóvil, entre otros. Los campos que contiene son:

|  |  |
| --- | --- |
| ID\_Cobertura: | Un identificador único para cada tipo de cobertura (clave primaria). |
| Nombre\_Cobertura: | El nombre descriptivo de la cobertura. |
| Descripcion: | Una breve descripción que detalla qué está cubierto por esa póliza. |

**Tabla de Seguros:**

En esta tabla se registran los detalles de los diferentes seguros disponibles para los clientes. Cada seguro estaría asociado a una cobertura específica de la tabla anterior. Los campos en esta tabla son:

|  |  |
| --- | --- |
| ID\_Seguro: | Un identificador único para cada seguro (clave primaria). |
| ID\_Cobertura: | Un identificador que establece la relación con la tabla de Coberturas (clave foránea). |
| Nombre\_Seguro: | El nombre o título del seguro. |
| Descripcion: | Información adicional que describe la póliza en detalle. |
| Precio: | El costo del seguro para el período de cobertura. |
| Moneda: | La moneda del costo del seguro para el período de cobertura (PEN=Moneda Peruana o USD=Dólar). |
| Duracion: | Indica la duración de la póliza (por ejemplo, anual, mensual, trimestral, semestral, etc.). |

**Tabla de Clientes:**

En esta tabla se almacenan los datos de los clientes que adquieren los seguros. Cada registro representa un cliente distinto. Los campos que contiene son:

|  |  |
| --- | --- |
| ID\_Cliente: | Un identificador único para cada cliente (clave primaria). |
| Nombre\_ Cliente: | El nombre del cliente. |
| Direccion: | La dirección postal del cliente. |
| Correo\_Electronico: | La dirección de correo electrónico del cliente. |
| Número\_Telefono: | El número de teléfono del cliente. |
| Fecha\_Nacimiento: | La fecha de nacimiento del cliente. |
| Sexo: | El sexo del cliente (M=Masculino o F=Femenino). |

**Tabla de Pólizas:**

Esta tabla se utiliza para registrar las pólizas individuales que se han vendido a los clientes. Los campos incluidos son:

|  |  |
| --- | --- |
| ID\_Poliza: | Un identificador único para cada póliza (clave primaria). |
| ID\_Cliente: | Un identificador que enlaza la póliza con un cliente específico de la tabla de Clientes (clave foránea). |
| ID\_Seguro: | Un identificador que enlaza la póliza con el tipo de seguro correspondiente de la tabla de Seguros (clave foránea). |
| Fecha\_Inicio: | La fecha en que la póliza comienza a ser válida. |
| Fecha\_Vencimiento: | La fecha en que la póliza expira o vence. |
| Estado\_Poliza: | Indica si la póliza está activo, cancelado, vencido o anulado. |

**Tabla de Pagos:**

En esta tabla se registran los pagos realizados por los clientes para mantener sus pólizas activas. Los campos son:

|  |  |
| --- | --- |
| ID\_Pago: | Es el identificador único (clave primaria) del pago. Se utiliza para identificar de manera exclusiva cada registro en la tabla. |
| ID\_Poliza: | Es una clave externa (foreign key) que hace referencia al identificador único de una póliza en la tabla "Polizas". Esto establece una relación entre el pago y la póliza a la que pertenece. |
| Fecha\_ Pago: | Es la fecha en la que se realizó el pago. Se almacena como un tipo de dato DATE. |
| Total\_Pago: | Es el monto total pagado en la transacción. Se almacena como un número decimal con una precisión de 10 dígitos y 2 decimales. |
| Moneda\_Pago: | Es la moneda del monto total pagado en la transacción. |

**Tabla de Detalle Pagos:**

|  |  |
| --- | --- |
| ID\_Detalle\_Pago: | Es el identificador único (clave primaria) del detalle de pago. Cada registro en esta tabla tiene un ID único para identificarlo de manera exclusiva. |
| ID\_Pago: | Es una clave externa (foreign key) que hace referencia al identificador único de un pago en la tabla "Pagos". Esto establece una relación entre el detalle de pago y el pago al que pertenece. |
| Monto\_Pago\_Detalle: | Representa el importe que se pagó nivel de detalle. La suma de todos los montos de pago detalle debe ser la suma total del total del pago. |
| Metodo\_Pago: | Es una cadena que indica el método utilizado para realizar el pago, por ejemplo, "Tarjeta de Crédito", "Transferencia Bancaria", "Tarjeta de Débito", etc. |
| Moneda\_Pago: | Es la moneda del monto pagado a nivel de detalle en la transacción. |

**Requerimientos:**

Crear las API REST con los siguientes endpoints:

* Endpoint para obtener la lista de todos los seguros disponibles, incluyendo información sobre sus coberturas asociadas.
* Endpoint para obtener los detalles de un seguro específico, incluyendo información sobre los clientes que tienen una póliza activa para ese seguro.
* Endpoint para agregar un nuevo cliente junto con sus datos personales.
* Endpoint para registrar una póliza de un nuevo cliente de un seguro que haya adquirido.
* Endpoint para registrar un nuevo pago para una póliza específica. El pago se registra en las Tablas Pagos y pagos Detalle. Para mantener la integridad de los datos se debe usar transacciones en Node JS.
* Implementar la lógica de negocio para validar los datos ingresados en los endpoints y garantizar la integridad de la base de datos.
* Gestionar las respuestas de error personalizadas con códigos de estado adecuados.

**Requerimientos Técnicos:**

Tener en cuenta las tecnologías necesarias a usar para desarrollar las API REST:

* Node.js (17+).
* Typescript (ES2021+).
* TypeORM o Sequelize (para definir las entidades y relaciones necesarias en la base de datos Oracle para reflejar las tablas de seguros).
* Hapi o Express (para definir las rutas y manejar las solicitudes HTTP entrantes).
* Oracle (para el almacenamiento y gestión de datos).

**Notas:**

* Para realizar la prueba, el candidato deberá tener acceso a un servidor Oracle (local o remoto) con las tablas mencionadas (Coberturas, Seguros, Clientes, Pólizas, Pagos y Pagos Detalle) creadas previamente.
* Puede tomar como referencia el script adjunto a este documento, que se le ha proporcionado para la creación de las tablas e inserción de la información en la Base de Datos.

**Criterios de Evaluación:**

* Configuración del entorno de desarrollo con Node.js y Typescript.
* Implementación de rutas y controladores para las funcionalidades requeridas (CRUD y consultas).
* Uso de TypeORM o Sequelize para realizar operaciones CRUD y consultas a la base de datos.
* Implementación de transacciones para las operaciones de registro de pagos, garantizando la integridad de los datos en caso de errores.
* Uso de buenas prácticas de codificación, incluyendo la modularización del código y la implementación de patrones de diseño apropiados.
* Manejo adecuado de errores y excepciones en la API.

Debes proporcionar el código fuente completo de las API REST desarrolladas, junto con las instrucciones para ejecutarla y probarla localmente.

¡Buena Suerte!