



## TECHNICAL TEST – FULL STACK

<b>Code:</b>	OSDENTAL-TECF-001
<b>Version:</b>	1.0
<b>Version Date:</b>	08-Nov-2024
<b>Prepared By:</b>	Daniel Alejandro Nicolalde Mendoza Enterprise Software Architect
<b>Review By:</b>	
<b>Approved By:</b>	
<b>Level of Confidentiality:</b>	Internal

## Instrucciones

- Responde las preguntas teóricas de forma breve pero precisa.
- Realiza el ejercicio práctico en el lenguaje o herramienta especificada.
- Tiempo estimado para completar la prueba: 2 horas.

## Sección 1: Preguntas Teóricas

### 1. Python

- ¿Cómo manejarías excepciones personalizadas en Python? Da un ejemplo.

---

---

---

---

- ¿Cómo manejarías un archivo de 10 GB en Python sin cargarlo completamente en memoria?

---

---

---

---

- ¿Qué es un ORM (Object-Relational Mapping)? Menciona ventajas y desventajas de usar un ORM como SQLAlchemy.

---

---

---

---

## 2. SQL Server

- ¿Qué es un índice compuesto en SQL Server, y en qué situaciones sería útil?

---

---

---

---

- Explica cómo interpretar un plan de ejecución en SQL Server. ¿Qué indicadores buscarías para optimizar una consulta?

---

---

---

---

- ¿Cuál es la diferencia entre funciones escalares y funciones con valores de tabla en SQL Server? Proporciona un caso de uso para cada una.

---

---

---

---

## 3. React

- ¿Cuál es la diferencia entre props y state en React? Da ejemplos de cuándo usarías cada uno.

---

---

---

---

- Explica cómo funciona el Virtual DOM en React y por qué mejora el rendimiento en comparación con manipular el DOM directamente.

---

---

---

---

- Explica cómo implementar carga diferida (*lazy loading*) de componentes en React y sus beneficios en el rendimiento.

---

---

---

---

## Sección 2: Preguntas Prácticas

### 4. SQL Server: Procedimientos y Consultas

Dado el siguiente esquema de tablas:

#### Empleados

```
CREATE TABLE Empleados (
```

```
    ID INT PRIMARY KEY,
```

```
    Nombre NVARCHAR(50),
```

```
    Departamento NVARCHAR(50),
```

```
    Salario DECIMAL(10, 2)
```

```
);
```

#### Departamentos

```
CREATE TABLE Departamentos (
```

```
    Departamento NVARCHAR(50) PRIMARY KEY,
```

```
    Presupuesto DECIMAL(15, 2)
```

```
);
```

Crea un procedimiento almacenado que:

- Inserte un nuevo empleado solo si el presupuesto del departamento no se excede después de agregar el salario del empleado.
- Manejo de Excepciones controladas y no controladas
- Devuelva un mensaje indicando si la operación fue exitosa o no.
  - Formato de Respuesta: STATUS\_CODE (código de respuesta), STATUS\_MESSAGE (Mensaje de la respuesta)

## 5. React: CRUD App

Crea un componente React que:

- Muestre una lista de tareas (con las propiedades: id, description, status).
- Permita agregar una nueva tarea (usando un formulario).
- Permita marcar una tarea como completada (actualizando el status).
- Utiliza *state* y simula la persistencia en el almacenamiento local del navegador.

## 6. Full Stack: Integración

Crea una API REST en Python (usando FastAPI y API ROUTER) que tenga los siguientes endpoints:

- GET /tasks: Devuelve una lista de tareas.
- POST /tasks: Recibe los datos de una nueva tarea y la agrega a la lista (almacenada en memoria o archivo).