



1. Prueba Técnica

1.1. Responda las siguientes preguntas de acuerdo a sus conocimientos. Si no conoce la respuesta, indíquelo:

- 1.1. ¿Qué es html, css y javascript?
- 1.2. ¿Para maquetar un email utilizaría divs o utilizaría tablas y por qué?
- 1.3. ¿Son las etiquetas
 la única manera de separar secciones de texto?
- 1.4. ¿Cuántos tipos de CSS existen?
- 1.5. El posicionamiento absoluto nos permite colocar un elemento en un punto exacto, y se quede ahí aunque desplacemos la página. La anterior afirmación es verdadera o falsa, argumente su respuesta.
- 1.6. ¿Qué propiedad CSS nos permite ocultar las viñetas de una lista?
- 1.7. ¿Qué es 'float' en CSS? Explíquelo con sus palabras.
- 1.8. ¿Qué recogen los selectores CSS 'clase' y 'ID' cuando se emiten? ¿Cómo los identificas en css?
- 1.9. Utilizando Sass explique o escriba la estructura de cómo le daría estilos al siguiente bloque de código:

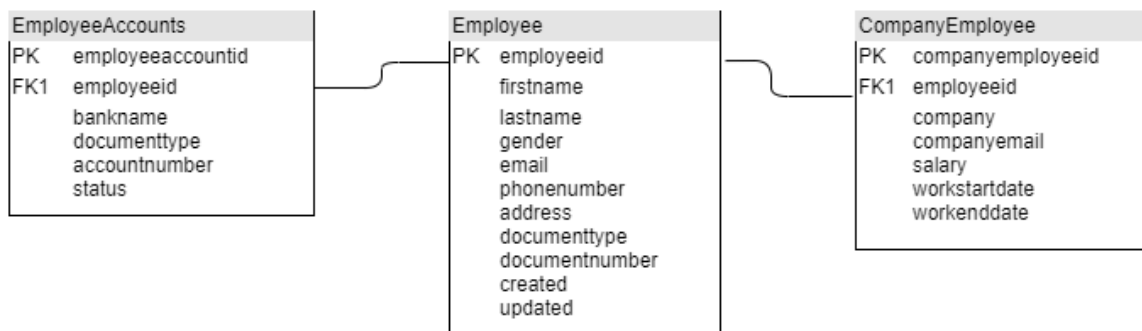
```
<div class="content">
  Lista de enlaces:
  <ul>
    <li>
      <a href="#">Enlace 1</a>
    </li>
    <li>
      <a href="#">Enlace 2</a>
    </li>
    <li>
      <a href="#">Enlace 3</a>
    </li>
  </ul>
</div>
```

- 1.10. ¿Qué significa la palabra clave 'this' en JavaScript?
- 1.11. ¿Cómo se puede cambiar el estilo/clase de un elemento en JavaScript?
- 1.12. ¿Qué es un callback?
- 1.13. ¿Qué es una API?
- 1.14. ¿Qué beneficios existen de JSON sobre XML?

- 1.15. ¿Qué protocolo utilizan los servicios web RESTful?
- 1.16. ¿Debería almacenarse los token de acceso JWT en el servidor o la base de datos?
¿Por qué?
- 1.17. ¿Cuáles son las diferencias entre los comandos 'delete' y 'truncate'?
- 1.18. Explique el uso de la palabra clave Join y sus distintos tipos
- 1.19. ¿Cuál es la diferencia entre las cláusulas having y where?
- 1.20. Explique el uso de la palabra clave Group by y nombre con cuales funciones se usa más a menudo.

2. Prueba Practica

Lo que se busca en esta sección es medir que tal se desempeña aplicando los conocimientos que ha adquirido, para esto deberá crear una pequeña aplicación cuya función sea realizar un CRUD utilizando SQL para la parte de base de datos, node para crear la parte del backend, es decir, APIS que conecten con la base de datos y realice toda la lógica y para la parte del frontend deberá utilizar react. A continuación, se describe más a detalle las partes del ejercicio:



2.1. En base al anterior diagrama crear los scripts para:

- Crear una base de datos
- Crear cada una de las tablas del diagrama
- Insertar datos en cada una de las tablas
- Eliminar un registro específico de la tabla employee
- Una consulta que asocie las tres tablas
- Una consulta que filtre los registros de la tabla employee por una parte del firstname. Por ejemplo, si escribo las letras "Ar" me debe traer todos los registros que tengan "ar" en su firstname.
- Una consulta que ordene los registros por lastname de forma alfabética ascendente.
- Una consulta que agrupe y me diga cuantos empleados son de la misma compañía, por ejemplo, de la compañía Advance hay cinco empleados, de la compañía IBM hay un empleado y así sucesivamente.

2.2. Teniendo en cuenta la base de datos creada anteriormente cree cuatro endpoints que se conecten con dicha base de datos para:

- Listar los empleados.
- Crear un empleado.
- Actualizar los datos de un empleado.
- Eliminar un empleado.

NOTA: Cada vez que se cree o se actualice un empleado se debe guardar la fecha en que se realizó la operación en los campos created y updated específicamente.

2.3. Teniendo en cuenta los endpoints creados anteriormente impleméntelos en la parte del frontend, la cual usted diseñara a su gusto utilizando react y el framework Material-ui (<https://material-ui.com/>) las pantallas a desarrollar son las siguientes:

- Listado de los empleados. La pantalla debe tener:
 - Botón para crear empleado
 - Lista de empleados con: Documento, nombre, apellido y teléfono
 - Para cada empleado debe existir un botón para editar y otro para eliminar el empleado
 - Si se solicita eliminar un usuario, mostrar una alerta con el mensaje “¿Está seguro de eliminar este registro?”
- Pantalla para crear un empleado con los campos:
 - Nombre (input text) - **Required**
 - Apellido (input text) - **Required**
 - Gender (select with options ‘Hombre’, ‘Mujer’, ‘Otro’) - **Required**
 - Email (input mail) Validar que sea campo de correo - **Required**
 - Dirección (input text) – **Not Required**
 - Teléfono (input text) - **Required**
 - Tipo de documento (select with options ‘Cédula’, ‘Cédula de extrangería’, ‘Otro’) – **Required**
 - Documento (input text) - **Required**
- Pantalla para editar los datos del empleado con los mismos campos.

2.4. Teniendo en cuenta sus conocimientos en html y css maquetar un mail, el cual debe quedar lo más parecido posible o idéntico a la imagen de referencia “Mail.png”. Tener presente lo siguiente:

- a. El mail debe tener las siguientes dimensiones:
 - Ancho: 600px
 - Alto: Se deja a su consideración
- b. Referencias de los colores utilizados:
 - #06E2B3
 - #F9F9F9
 - #146FD1

- c. Enlace de la fuente utilizada:

<https://fonts.googleapis.com/css2?family=Nunito+Sans:wght@300;400;700;900&display=swap>

NOTA: Las imágenes e iconos para realizar el mail se encuentran en el archivo comprimido enviado junto con la descripción de la prueba.

- 2.5. Con el mail creado en el punto anterior intente enviarlo al email del empleado una vez se cree el registro, es decir, una vez el usuario oprima el botón de 'guardar empleado' se debe enviar al correo del empleado el email maquetado.