

TITA MEDIA

PRUEBA DE CONOCIMIENTO

Nombre		Fecha	DD/MM/YYYY
Cargo			

COMPONENTE TEÓRICO.

1. Indique con sus propias palabras qué es Ingeniería del Software.
2. Indique para usted que es un API **REST** y **SOAP**, relacione tres elementos que las diferencie, cuál de las dos es más usada y por qué.
3. Indique para usted que es un ORM, cuales conoce, por qué se utilizan y qué beneficios traen.
4. En los lenguajes de programaciones orientados a objetos que son las interfaces y por qué son utilizadas.
5. Que resulta de ejecutar el siguiente fragmento de código

```
String a = "5";  
String b = "10";  
System.out.println(a + Integer.parseInt(b));
```

COMPONENTE LÓGICO.

6. Teniendo el siguiente vector de cadenas

```
[ "abc", "ujk", "zzy", "ahj", "aaz", "oip" ]
```

Realice un algoritmo que permita organizar los elementos del vector de menor a mayor, teniendo en cuenta que el valor de cada elemento (cadena) será la suma de la posición de cada letra en el alfabeto.

Ejemplo: el valor de la cadena "abc" es 6 ya que la posición de cada letra es 1, 2 y 3 y la suma de estos números es 6.

7. Dado el siguiente vector

```
[ 1, 4, 6, 7, 8, 8, 127, 89 ]
```

Realice un algoritmo que permita realizar lo siguiente:

- Dividir el vector en múltiples vectores según un valor dado. Ejemplo: dado el valor 3, el vector será dividido en tres vectores, quedando de la siguiente forma [1,4,6], [7,8,8], [127,12,89]

NOTA: Cada subvector debe quedar con la misma cantidad de elementos, por lo tanto, el sistema debe validar que el valor dado permita dividir el vector con la misma cantidad de elementos cada uno.

- (Opcional, Bonus si lo realiza) Luego de realizar la división del vector, el sistema debe sumar los elementos de cada subvector e indicar cual es el de mayor valor.

COMPONENTE PRÁCTICO (TECNOLOGÍA JAVA SPRING BOOT)

8. Realizar una aplicación que le permite a un cliente pagar las deudas que posee con sus bancos, se deben soportar las siguientes necesidades:

- ❖ Inicio de sesión de usuario.
- ❖ Se debe mostrar el listado de bancos que tiene el usuario.
- ❖ Se debe mostrar el listado de deudas según el banco seleccionado.
- ❖ Permitir al cliente pagar la deuda.

Aspectos funcionales:

- ❖ Debe existir una vista que permita al usuario iniciar sesión.
- ❖ Debe existir una vista que permita al usuario ver el listado de sus bancos.
- ❖ El sistema debe permitir al usuario seleccionar un Banco y en otra vista mostrar las deudas que posee en ese banco.
- ❖ El sistema debe permitir al usuario seleccionar una deuda y mostrar cuánto ha pagado, cuando cuotas debe, cuanto es el dinero total que debe.
- ❖ El sistema debe permitir al usuario pagar una deuda, el usuario podrá indicar cuantas cuotas quiere pagar o si quiere pagar su deuda en su totalidad. Posteriormente el sistema debe actualizar para mostrar el nuevo saldo que debe, cuántas cuotas debe o si la deuda ya fue cancelada.

Aspectos técnicos:

- ❖ Aplicación Backend con tecnología Java Spring Boot.
- ❖ Puede usar su tecnología de preferencia para el Frontend.
- ❖ El backend debe exponer un API REST.
- ❖ El Frontend debe tener un aspecto prudente (organizado, limpio), no es necesario un diseño que demande mucho tiempo.

- ❖ Puede usar la base de datos que desee.
- ❖ Puede estructurar las tablas de la base de datos como desee.
- ❖ En la base de datos ya debe existir la información de usuarios, bancos y deudas. (No es necesario que realice código para el registro de esta información)

Entregables:

- ❖ Manual instructivo de despliegue y configuración de la aplicación (lo necesario para iniciar la aplicación)
- ❖ Archivo de despliegue.
- ❖ Código fuente (indicar el IDE el cual se realizó)
- ❖ Script de base de datos.