**Parte I**

1. ¿Cómo encontraría usted el número faltante dado un arreglo no ordenado que

contenga una secuencia de enteros del 1 al 100, y le falte un número?

* var datos=[];
* for (let index = 1; index <= 100; index++) {
* datos.push(index)
* }
* function removeItemFromArr( arr, item ) {
* return arr.filter( function( e ) {
* return e !== item;
* } );
* };
* mat= Math.floor(Math.random() \* (100 - 1) + 1)
* console.log("delete",mat)
* datos=removeItemFromArr( datos,mat);
* datos=datos.sort(function(){return Math.random()-0.5})
* console.log(datos)
* for(i=1;i<=100;i>i++){
* var cont=0
* for (let index = 0; index < datos.length; index++) {
* if(datos[index]==i){
* cont=+1;
* }
* }
* if(cont==0)
* console.log("no esta el"+i)
* }

1. ¿Cómo encontraría usted el primer número que aparezca duplicado en un arreglo de

enteros desordenados?

* const array = [1,3,14,3,3,4,5,6,2,3,7,8,5,9];
* for (let x = 0; x < array.length; x++) {
* for (let i = 0; i < array.length-x-1; i++) {
* if(array[i] >array[i+1]){
* var tmp = array[i+1];
* array[i+1] = array[i];
* array[i] = tmp;
* }
* }
* }
* console.log(array);
* const numbers = array
* let number = 0;
* for (let i = 0; i <= numbers.length; i++) {
* if (numbers[i + 1] == numbers[i]) {
* number = numbers[i];
* break;
* }
* }
* console.log(number);

1. ¿Cómo reversaría usted el orden de un arreglo de enteros?

* var datos = [1,2,3,4,5,6,7,8,9,10,11,12,13];
* let longitudDelArreglo = datos.length;
* for (let x = 0; x < longitudDelArreglo / 2; x++) {
* let temporal = datos[x];
* let indiceContrario = longitudDelArreglo - x - 1;
* datos[x] = datos[indiceContrario];
* datos[indiceContrario] = temporal;
* }
* console.log(datos);

1. ¿Cómo intercambiaría usted el valor de dos variables que contienen enteros, sin

utilizar una tercera variable?

* var a = 30;
* var b = 40;
* a = a + b
* b = a - b
* a = a - b
* console.log(a)
* console.log(b)

1. ¿Cómo verificaría usted que una cadena de texto es palindrome?

* var palabra= 'arenera' ;
* i=0
* j =palabra.length-1
* while(palabra[i]==palabra[j] && i<j){
* i=i+1
* j=j-1
* }
* console.log(i)
* console.log(j)
* if(i>=j){
* console.log("palindroma")
* }else{
* console.log("no espalindroma")
* }

1. ¿Cómo reversaría usted una cadena de texto utilizando recursión?

* const reverseString = (str) => {
* return str ? reverseString(str.substr(1)) + str[0] : str;
* };
* console.log(reverseString('prueba'));

1. ¿Cómo contaría usted el número de consonantes en una cadena de texto?

* var palabraIntroducida = "eeqqqq";
* var Cont= 0;
* for (var i = 0 ; i < palabraIntroducida.length ; i++) {
* switch(palabraIntroducida.charAt(i)) {
* case "a":
* Cont++;
* break;
* case "e":
* Cont++;
* break;
* case "i":
* Cont++;
* break;
* case "o":
* Cont++;
* break;
* case "u":
* Cont++;
* break;
* }
* }
* console.log(Cont)

**Parte II**

1. Si se quiere emplear el proceso TDD (Test Driven Development) en la empresa,

seleccione la opción que mejor represente el flujo de trabajo de los métodos TDD.

* Requisitos → Arquitectura de software → Pruebas unitarias → Desarrollo de

software → Revisión de código → Pruebas funcionales → Soporte

1. ¿Qué tipo de lenguaje es Ruby, y en qué se diferencia con Java?

* Es un lenguaje de backend, a diferencia de java es de sintaxis mas simple reduciendo bastante el código no es necesario compilar el código, las variables son privadas, se accede a ellas usando métodos, las variables no son tipadas, no es necesario poner paréntesis en las llamadas de los métodos.

1. ¿Qué son los microservicios?

* Es un componente individual diseñado para un solo propósito el cual debería tener su propia base de datos sin depender de otros servicios.

1. ¿Por qué serían beneficiosos los microservicios para una organización?

* Cuando se tiene una organización con experiencia en orquestación la mantención es mucho más fácil, si se cae algún microservicio este no afectara la productividad de los demás.

1. ¿Por qué no serían beneficiosos los microservicios para una organización?

* Si no se tiene un grupo bien estructurado y sin experiencia al momento de la orquestación puede haber problemas, al principio en la implementación de la estructura se verán altos costos a comparación de un servicio monolítico.

1. ¿Cuál es la diferencia entre SOA y Microservicios?

* SOA está más orientado a productos bancarios con protocolo XML con peticiones empaquetadas más pesadas.
* Microservicios trabaja con aplicaciones RESTfull (con JSON o XML) bajo los protocolos https se usa para cosas más pequeñas con peticiones livianas con tiempos de respuesta menores.

1. ¿Por qué podría ser más compleja la comunicación por una red en una arquitectura

por microservicios?

* Se debe a los saltos que pueden haber entre cada componente haciendo que halla mas latencia haciendo que el tiempo de respuesta sea más elevado entonces se debe tener una arquitectura definida de como va a ser el comportamiento de la aplicación.

1. Proponga un método de comunicación para una arquitectura por microservicios.

* Lo ideal es tener un orquestador que se encargue de tener mapeado donde se encuentra cada microservicio y como se llama adicional debería tener un Gateway que seria la puerta de entrada donde todo el llamado o todo el consumo que se le hace a una API tiene que llegar primero al Gateway para que este llame al orquestador y luego este llame al microservicio y se ejecute la acción.

1. ¿Qué es y cómo funciona Docker?

* Es un contenedor portable para aplicaciones de software que puedan ejecutarse en cualquier entorno o maquina con Docker instalado facilitando así los despliegues.

1. ¿Qué es y cómo funciona Kubernetes?

* Es un orquestador de contenedores, encargado de gestionar el ciclo de vida de los contenedores de una aplicación, manejando los clúster, servicios Discovery, servicios de red y load-balancing, monitoreo, chequeo de estado de salud entre otros.

**Parte III**

<https://github.com/jhorman10/Prueba3A>