Universidad Galileo

Carrera: Técnico universitario en desarrollo Full stack

Curso: Estructura y análisis de datos

Sede: Virtual

Resúmenes de las clases.

Nombre: Maidellin Suset Alvarado Cayax

Carné: 24011377

Sección: T

Fecha de entrega: 07/21/2024

SESION 1

05/03/2024

En esta sesión fue una introducción a que veremos en la clase durante todo el semestre algunos temas son:

- Algoritmos
- Estructura de datos
- Lectura de los datos en una computadora.
- Continua Aprendizaje.

Algoritmos:

Son una serie de instrucciones por lo general es mejor si son las menores posibles para que sea útil, los clasificamos en base a su estructura, iterativo, o recursivo.

Es un proceso o una función en matemática la conocemos como una máquina.

Un algoritmo iterativo, es un bucle o ciclo. Iteramos diferentes valores repetidas veces hasta que una condición nos lo permita.

Un recursivo es cuando: Es un ciclo infinito, por ejemplo, factoriales, Fibonacci. Se llama a sí mismo, en una recursividad no hay condiciones, sino que el dato se llama a sí mismo. No se ve mucho en el ámbito de la programación, pero es de utilidad que lo sepamos.

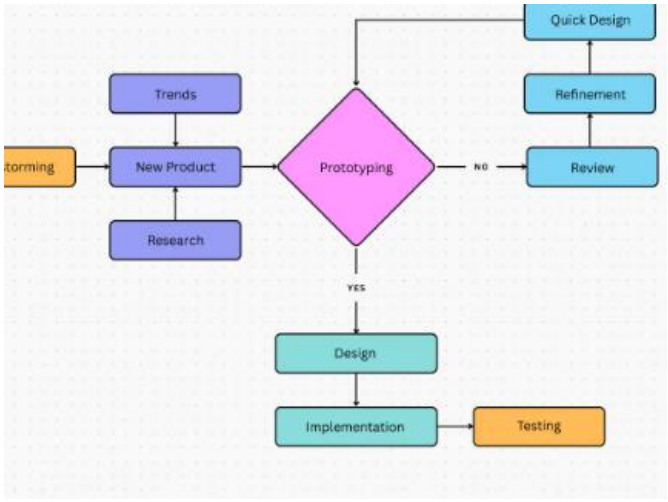
El tiempo que toma el algoritmo se mide en función de cuantas instrucciones están en un algoritmo, el tiempo que se toma algoritmo se puede expresar en gráficas, como lineales, cuadráticas, exponencial.

Cuando finalicemos este curso tendremos podremos realizar páginas, ponerles interfaces, hacer una página interactiva, en base a código de páginas web.

SESION 2

12/03/2024

En esta sesión vimos el diagrama de flujo es una representación visual de un proceso o algoritmo, se representan con figuras gráficas, como círculos que significan inicios, decisiones que son rombos, procesos que son cuadrados.



Es una forma estándar de entender los diagramas, para personas alrededor de todo el mundo. Es una herramienta para principiantes de la programación. Mas adelante cuando seamos más experimentados ya los utilizaremos, pero poco.

COMPLEJIDAD EN UN ALGORITMO

Para ver su complejidad de un algoritmo, las podemos medir con la notación big 0, esta notación es usada por programadores para medir el tiempo que se toma el algoritmo en finalizar, con esta notación evaluamos los pasos lógicos que se tomaron en el algoritmo, excepciones, resolución de problemas, resultado etc.

La complejidad Big 0 Ayuda a identificar áreas del código que podrían ser optimizadas para mejorar el rendimiento.

La complejidad se identifica como contaste, exponencial, cuadrática, logarítmica etc.

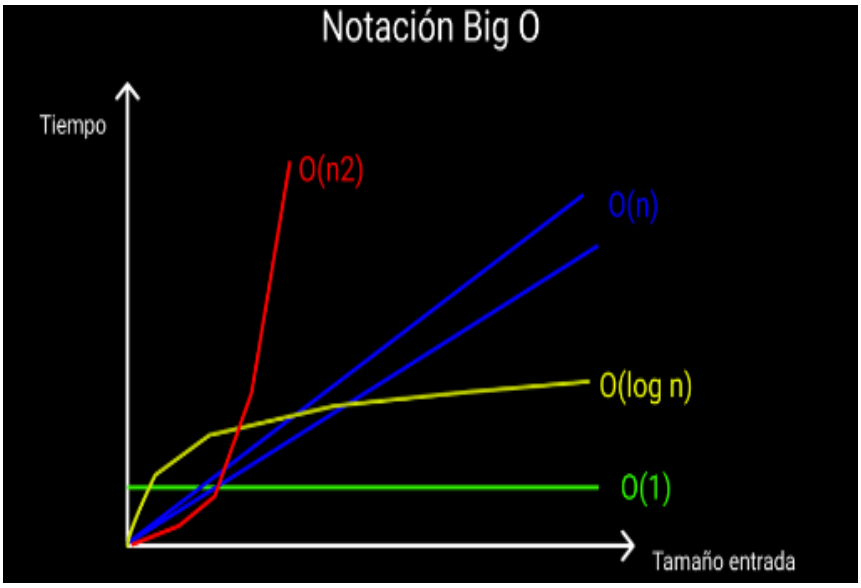
En la constante la operación toma el mismo tiempo sin importar el tamaño de la entrada.

En la logarítmica el tiempo de ejecución crece en proporción al logaritmo del tamaño de la entrada.

La lineal el tiempo de ejecución crece directamente en proporción al tamaño de la entrada.

La cuadrático crece proporcionalmente al cuadrado del tamaño de la entrada.

En la exponencial la ejecución aumenta exponencialmente con el tamaño de la entrada. Este es muy tardado es mejor evitarlos.



En conclusión, esta notación nos ayudara a elegir el mejor algoritmo para nuestros programas y con la notación o tenderemos una idea del tiempo que está invirtiendo el usuario en nuestra plataforma. Nos puede ayudar a optimizar nuestro código.

SESION 3

19/03/2024

En esta clase vimos el pseudocidogo, es una forma de describir algoritmos utilizando un lenguaje informal y semiestructurado que se asemeja al lenguaje de programación real, Se utiliza principalmente para planificar y diseñar algoritmos antes de ponerlo en código para tenemos una mejor idea de que es lo que haremos

El pseoudocidog es el lenguaje al nivel que nosotros entendamos, por ejemplo, las computadoras entienden lenguaje binario, pero nosotros con palabras. Primero lo pasamos a palabras y después lo pasamos a lenguaje de programación.

Nosotros no podemos hablar lenguaje binario por eso debemos aprender el lenguaje de programación para darle instrucciones la computadora.

Método de análisis de datos:

Analizamos los datos analizando el disco duro y la memoria RAM , y el tiempo que se toma en procesar los datos, la memoria RAM es muy rápida pero no guarda ningún registro a menos que la computadora este encendida. Y el disco duro que es un sistema que guarda incluso cuando la computadora esta apagada

Lo puedo medir a base a casos, peor o mejor de los casos,

En base a la experiencia, en esta en base al uso de nuestra pagina web veremos si es óptima o no.

Tipos de estructura de datos:

Cuando hablamos de estructura de datos en la computadora hablamos de 0 y 1, se llaman bit regularmente se dice true, o false. Y también en Byte, que son 8 bits se guarda mas que true o false

Los tipos de datos:

Enteros representan números enteros, positivos o negativos.

Reales representan números con decimales. Normalmente se dividen en float (32 bits) y double (64 bits).

Caracteres son como letras, números y símbolos. Se utilizan comúnmente para manejar texto y caracteres individuales.

Booleanos son valores lógicos Verdadero (true) o Falso (false). Se utilizan para condiciones lógicas y evaluaciones de control

SESION 4

02/04/2024

En esta clase vimos los arrays, es una lista indexada, o por como entran al sistema. Es una lista de datos que guarda múltiples valores, esto es bueno para muchos datos y que te permiten almacenar colecciones ordenadas de datos y acceder a ellos mediante un índice numérico.

Se recorre con un ciclo, porque son muchos datos los elementos de un array pueden ser de diferentes tipos, boleanos, string, int, al menos en java script así funciona.

Con un array podemos guardar cualquier valor, para crear un array esta es la estructura:

```
String Nombres [];
Nombres = new String[5];

Nombres[0] = "Samantha";
Nombres[1] = "Javier";
Nombres[2] = "Marina";
Nombres[3] = "Ana";
Nombres[4] = "John";
```

Los array son útiles para crear un objeto, una instancia objeto es una plantilla para otros objetos, los datos se ingresan en un arreglo, para evitar la creación de muchas variables,

Para correr este array utilizamos un ciclo for permite repetir un bloque de código un número específico de veces. Es útil cuando sabes exactamente cuántas veces necesitas ejecutar una serie de instrucciones. Los arrays empiezan a correr desde 0, pero para la computadora el 0 si tiene un valor y termina cuando lo decidimos. La propiedad del array que es length que es el tamaño abstracto en un array. También podemos ordenarlos por el menor al mayor, por orden alfabético de reversa etc.

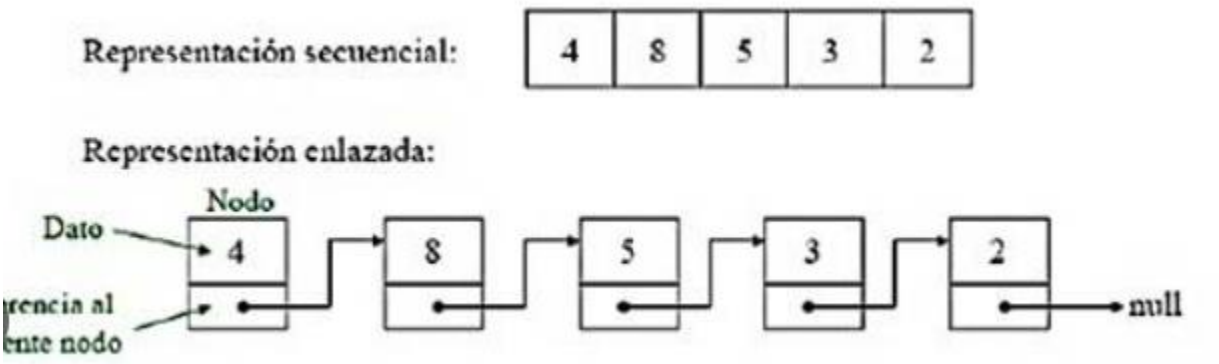
En conclusión, los arrays nos ayudan a instanciar objetos con diferentes propiedades, como el mismo modelo de carro, pero diferente año y diseño. Este solo es un ejemplo básico del uso de los arrays.

SESION 5

09/04/2024

En esta clase vimos las listas enlazadas que son similares al los array pero ahora vemos los datos no con índices sino con punteros y nodos cada nodo contiene dos partes principales: el dato que queremos almacenar y una referencia (enlace) al siguiente nodo en la lista. Los array son un poco más difíciles porque no hay direcciones a donde apuntar pero en las listas si tenemos punteros y enlaces. Las listas se utilizan mas en C++ pero lo implementaremos en java script.

Las listas enlazadas funcionan por medio de enlaces como explicaba en el párrafo anterior, es una lista que apunta a otro valor y este apunta a otro y el otro al otro y así sucesivamente.



Implementar y entender listas enlazadas es fundamental para la programación, especialmente cuando se trabaja con datos dinámicos. Que siempre van cambiando.

Existen listas, dobles, circulares y simples, pero nosotros nos centraremos en las simples

La eficiencia de las operaciones en una lista enlazada depende del tipo de operación y la implementación específica. Las inserciones y eliminaciones pueden ser rápidas en comparación con otras estructuras de datos como los arrays, especialmente en listas largas.

SESION 6

16/04/2024

En esta sesión veremos las Pilas, en la que todas las inserciones se realizan por un mismo medio tope o encima de la pila. Las pilas son útiles en muchas aplicaciones, como la implementación de sistemas de historial porque ordena del mas reciente o mas antiguo los registros de búsqueda.

Con las pilas podemos borrar, crear, push y pop en una pila basada en un array tienen un tiempo de complejida lineal lo que las hace muy fáciles de usar.

Ejemplo de una creación de una pila en Java script

```
class Stack {
  constructor() {
    this.items = []; // Array para almacenar los elementos de la pila}
```

Métodos de la pila:

- **pop()**: Elimina y devuelve el último elemento añadido a la pila.
- **peek()**: Devuelve el último elemento añadido a la pila sin eliminarlo.
- **isEmpty()**: Verifica si la pila está vacía.
- **size()**: Devuelve el número de elementos en la pila
- clear(): Vacía la pila, eliminando todos los elementos.
- printStack(): Imprime los elementos de la pila en formato de cadena para propósitos de visualización.

SESION 7

23/04/2024

En esta clase tuvimos una pequeña practica de java script donde utilizamos html para hacer un pequeño juego de preguntar y respuestas

Creamos un contenedor utilizamos código de html y ccss

Botones, animaciones como hover, background color y hover en ellos botones, bordes con border-radium

Creamos un array de preguntas con una variable y otra con la variable respuestas con estos dos arrays.

SESION 8

30/04/2024

En programación y estructuras de datos, un árbol binario es una estructura de datos jerárquica en la cual cada nodo puede tener hasta dos nodos hijos, conocidos como hijo izquierdo y hijo derecho y las hojas. Estos árboles tiene niveles.

SESION 9

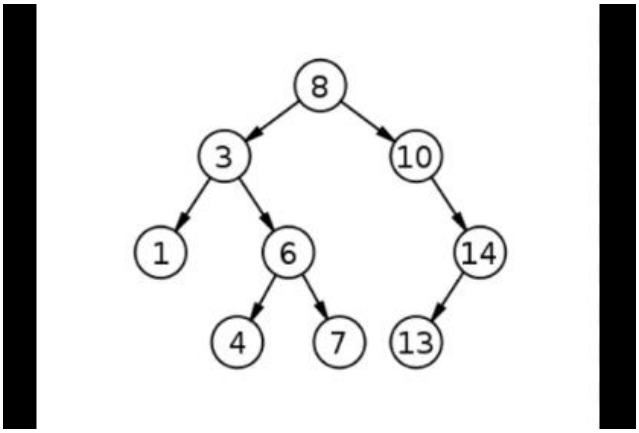
07/05/2024

En esta clase vemos los árboles binarios de búsqueda, se utilizan para reducir el tiempo de búsqueda, lo que trata de hacer es crear la estructura de árbol balanceada, para almacenar los datos de forma ordenada.

Los descendientes izquierdos deben ser menores al padre y los derechos mayores a los padres.

En esta clase analizamos los arboles y sus datos si son mayores los hijos para acomodarlos a los padres acomodar del lado correcto.

- Algunas condiciones son los datos para insertar no deben ser iguales
- El nuevo nodo será nodo hoja
- Buscar al nodo padre del nodo a agregar
- El orden sin importa al agregar los datos



Eliminación de nodo

Hay tres casos para eliminar, si es hoja desconectamos el nodo hoja del nodo padre y ya esta

Si es un nodo hijo buscar el padre conectar el hijo con el nodo del padre que vamos a borrar y liberar el nodo

Si es nodo con dos hijos, localizamos el nodo sucesor copiamos la información y lo eliminamos según sea necesario

Eliminación Predecesor: Es el método uno ala izquierda y toda ala derecha.

Eliminación sucesora Es el método uno ala derecha y toda ala izquierda.

Un árbol binario de búsqueda en código se ve mas o menos así

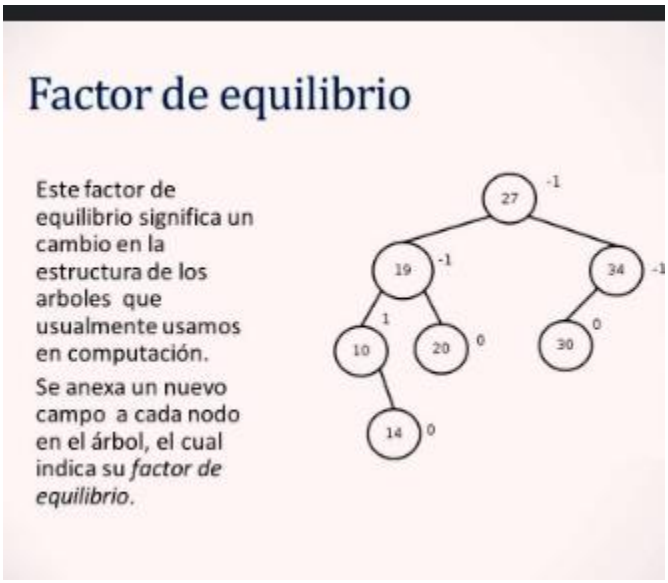
```
class Node {
  constructor(value) {
    this.value = value;
    this.left = null;
    this.right = null;
  }
}
```

```
function insert(root, value) {
  if (root == null) {
    return new Node(value);
  }

  if (value < root.value) {
    root.left = insert(root.left, value);
  } else {
    root.right = insert(root.right, value);
  }

  return root;
}
```

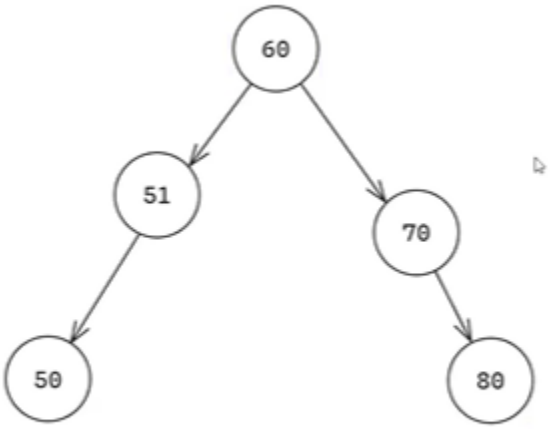
SESION 10

14/05/2024

En esta clase vimos un poco de Arboles AVL que son arboles balanceados donde su altura difiere en 1, en este árbol tenemos que lograr un equilibrio para buscar más rápido los datos, para ordenar el árbol lo balanceamos.



Para balancear los árboles tenemos reglas, las alturas difieren de 1.



Para insertar un dato siguiendo algunas de las reglas, tendremos que encontrar la posición correcta para insértalo, cuando lo tengamos se calcula el factor de balance desde la raíz hasta el nodo insertado.

Si algún nodo supera el umbral de 1 hay un desequilibrio se debe realizar una rotación para equilibrarlo. Depende del peso que tengamos ahí determinaremos a donde nos balancearemos.

SESION 11

21/05/2024

En esta sesión hablamos un poco de git hub y como crear un repositorio,

En esta clase instalamos git hub CLI que es una herramienta de línea de comandos que te permite interactuar con Git, un sistema de control de versiones distribuido ampliamente utilizado. Git CLI proporciona una forma poderosa y flexible de gestionar versiones del código.

Investigando un poco sobre este programa encontré sus ventajas de utilizar Git CLI:

Control y flexibilidad: La línea de comandos te ofrece un control granular sobre cada acción que realizas en Git.

Eficiencia: Para usuarios avanzados y aquellos cómodos con la línea de comandos, Git CLI puede ser más rápido que las interfaces gráficas para tareas repetitivas.

Integración: Se integra fácilmente con flujos de trabajo de desarrollo y herramientas de automatización a través de scripts y herramientas de línea de comandos.

También vimos Git hub Pages que ofrece GitHub para alojar sitios web estáticos directamente desde tus repositorios de GitHub. Es gratuito de publicar tu sitio web personal, proyecto asociada con tu repositorio en GitHub.

SESION 12

28/05/2024

En esta clase vimos un poco de Tablas dinámicas. Una tabla dinámica en JavaScript es una tabla en HTML que se crea, actualiza o modifica dinámicamente utilizando JavaScript en respuesta a eventos o datos específicos que implican filas y columnas, la actualización de celdas con datos nuevos, o la eliminación de elementos según sea necesario.

Puedes agregar, eliminar o actualizar celdas y filas en respuesta a eventos como la carga de datos o acciones del usuario.

Puedes agregar funcionalidades como ordenación, filtrado y paginación para permitir que los usuarios naveguen y manipulen los datos visualizados de manera más eficiente.

Seguimos en el tema de rotación de árboles binarios balanceos, en la clase anterior vemos que podemos equilibrarlos por medio de métodos, uno de estos es:

Rotación simple a la derecha: Identificamos el desbalance, se mueve al posición correcta y se convierte en el subárbol

Es una técnica fundamental en la teoría de árboles binarios de búsqueda balanceados como los árboles AVL. Su aplicación permite mantener el árbol balanceado y garantizar operaciones eficientes incluso cuando el árbol experimenta cambios dinámicos en su estructura.

Rotación simple ala izquierda: Es un proceso similar al de la derecha, la rotación hacia la izquierda preserva las relaciones de orden en el árbol binario de búsqueda, asegurando que los elementos a la izquierda de cualquier nodo sean menores y los elementos a la derecha sean mayores.

SESION 13

04/06/2024

En esta clase vemos Algoritmos de Ordenamiento que nos permite organizar los elementos de una lista hay muchos tiempos de ordenamiento

Buble short: Evalúa la lista por pares compara cada elemento con su suceror y los comparar para luego intercambiarlos

El proceso se repite tantas veces como sea necesario hasta que no se requieran más intercambios en una pasada completa. Esto asegura que los elementos queden ordenados de menor a mayor al final del proceso. Para esto se utiliza la variable i este es para el bucle externo.

Para el bucle interno Itera sobre los elementos no ordenados (desde el inicio hasta n-i-2). Compara cada elemento con su siguiente (lista[j] y para el bucle de salida si en una pasada completa no se realizan intercambios significa que la lista está ordenada y se puede salir del bucle.

Selection short:

Recorre todo el arreglo hasta que cumple con todos los criterios de ordenamiento impuestos

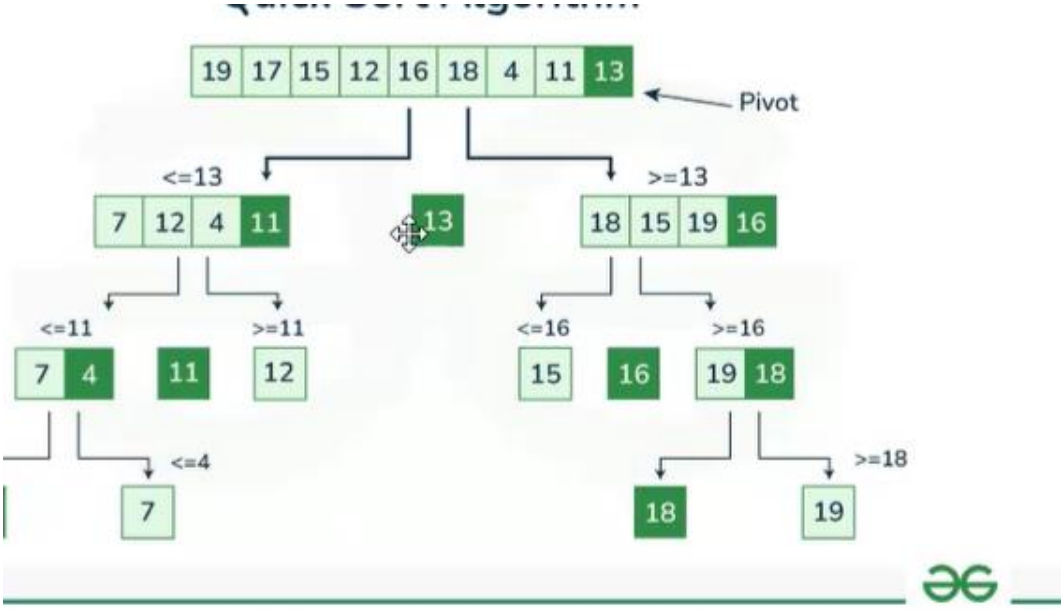
El algoritmo divide la lista en dos partes: una parte ordenada y otra parte desordenada. En cada iteración, busca el elemento más pequeño (o más grande) en la parte desordenada y lo coloca al inicio de la parte ordenada.

En cada pasada, encuentra el mínimo elemento de la parte desordenada y lo intercambia con el primer elemento de la parte desordenada esto se repite hasta que la lista esta ordenada.

```
function selectionSort(arr) {
  if(arr.length < 2) return arr;
  let min;
  let index;

  for (let i = 0; i < arr.length; i++) {
    min = arr[i]
    for (let j = i+1; j < arr.length; j++) {
      if (min > arr[j]) {
        min = arr[j];
        index = j;
      }
    }
    if(idx) {
      [arr[i], arr[index]] = [arr[index], arr[i]]
      index = undefined;
    }
  }
  return arr;
}

selectionSort(nums);
```

En el insertion short: Es un método simple y eficiente para ordenar pequeñas cantidades de elementos es uno a la vez, colocando cada nuevo elemento en su lugar correcto en la parte ordenada de la lista. El algoritmo divide la lista en dos partes: una parte ordenada y otra parte desordenada. Inicialmente, la parte ordenada contiene el primer elemento de la lista y la parte desordenada el resto.

En cada iteración, el algoritmo toma el primer elemento de la parte desordenada y lo compara con los elementos en la parte ordenada. Mueve los elementos mayores hacia la derecha para abrir espacio y luego inserta el elemento en la posición correcta. Esto lo va repitiendo hasta que todos los elementos desordenados estén en la parte ordenada.



Por ultimo vimos Quick short que es uno de los métodos más eficientes para ordenar arreglos grandes de elementos. Utiliza la estrategia de dividir y conquistar, lo que implica dividir repetidamente el arreglo en subarreglos más pequeños hasta que cada subarreglo contenga un solo elemento, lo cual está inherentemente ordenado. Luego, combina recursivamente los subarreglos en el orden correcto.

Yo investigue un poco más y te muestro como funciona con esta explicación:

Selecciona un elemento del arreglo como pivote. Generalmente, se elige el primer elemento, el último elemento o un elemento al azar.

Reorganiza los elementos del arreglo de manera que todos los elementos menores que el pivote estén a su izquierda y todos los elementos mayores estén a su derecha.

Aplica recursivamente el mismo proceso a los subarreglos de elementos menores y mayores al pivote.

Los subarreglos ordenados se combinan para formar el arreglo final.

```
const quickSort = (nums) => {
    if (nums.length < 2) return nums;

    const pivot = nums[0];
    const smaller = [];
    const bigger = [];
      for(let i = 1; i < nums.length; i++) {
      if(nums[i] < pivot) smaller.push(nums[i]);
      else bigger.push(nums[i]);
    }

    return [...quickSort(smaller), pivot, ...quickSort(bigger)];
  }
```

SESION 14

11/06/2024

En esta clase vimos los objetos en java script. Java Script es un lenguaje no tipado así que no debemos decláralo por defecto js lo hace siempre hay tipado pero en este caso no lo vemos porque lo hace internamente.

Los objetos son estructuras que permiten almacenar datos y funciones. A diferencia de otros tipos de datos simples como números y cadenas de texto, los objetos en JavaScript pueden contener múltiples valores y tipos, así como métodos así que no debemos de preocuparnos por decláralo.

Un objeto tiene propiedades: Son variables dentro del objeto que almacenan valores.

Y también tiene métodos: que son funciones definidas dentro del objeto que pueden realizar operaciones en sus datos.

Por ejemplo:

```
let persona = {
  nombre: 'Juan',
  edad: 30,
  esEstudiante: false,
  saludar: function() {
    console.log(` Hola, soy ${this.nombre} y tengo ${this.edad} años. ` );
  }
}
```

Objetos Json:

Es una sintaxis similar a la de los objetos y matrices en JavaScript, pero está estrictamente formateado como texto plano.

Los datos en JSON están estructurados en pares de clave: valor, donde la clave siempre es una cadena (string) y el valor puede ser cualquier tipo de dato válido en JSON (números, cadenas, booleanos, objetos, arrays o null).

```
{
   "nombre": "Juan",
   "edad": 30,
   "esEstudiante": false,
   "direcciones": [
      }
```

Nada mas sin mas.

Pero porque tenemos dos formas similares de hacer lo mismo, JSON se utiliza principalmente para intercambiar datos entre un servidor y un cliente web, ya que es fácilmente legible por humanos y fácil de analizar y generar por las máquinas.

Es común en servicios web y API para enviar datos desde un servidor al cliente o viceversa.

SESION 15

18/06/2024

En esta clase uniremos varios conceptos que ya vimos.

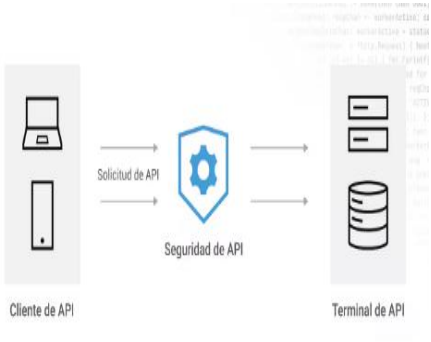
Empezamos con frontend vs Backend: Lo que sucede por detrás es el backend y lo que yo veo e interactuar y que interpreta mi navegador es mi Frontend.

Si quiero ver datos, como números e información y conocimiento que es que hago con los datos y la información. El conocimiento es la unión de datos e información.

Cuando nosotros tenemos mas conocimientos poderemos utilizar bien la información y los datos que nos dan, los datos usualmente se manejan con el Bakcend.

Para obtener datos deberemos tener acceso a una base de datos, y necesito algo que conecte los dos y con eso hablamos de API es un intermediario de los datos y la aplicación, los puedo editar, guardar etc. API maneja permisos, estructura, estadísticas.

Existe dos tipos que son SOAP y REST.



Ahora gracias a Json usamos solo HTTP. También vimos que un CRUD (Create, Read, Update, Delete) es una operación estándar utilizada en bases de datos y aplicaciones para gestionar la creación, lectura, actualización y eliminación de datos. Cuando se combina con una API nos permite realizar estas operaciones de manera remota a través de solicitudes HTTP.

SESION 16

25/06/2024

En esta sesión realizamos código para la tarea de To-do list

SESION 17

02/0/2024

Fetch:

En esta clase vimos introducción a fetch, es una función de JavaScript que se utiliza para realizar solicitudes HTTP/HTTPS asíncronas desde el navegador web hacia un servidor web. Es parte del estándar moderno de JavaScript conocido como Fetch API, que proporciona una interfaz para recuperar recursos (como archivos, datos JSON, XML, entre otros) de forma asíncrona a través de la red.

Podemos asignarle un link a una variable con fetch, es una API moderna para solicitudes HTTP como mencione anteriormente.

También aprendimos CEO en Google para saber como se posiciona nuestra página en Google.

Request y response

Request: información enviada desde el cliente al servidor

Response: Información devuelta por el servidor al cliente

Token: Es una cadena de caracteres para la autenticación y autorización.

También hablamos un poco de java script que es secuencial, asíncrono, no bloqueador, simultaneo.t

SESION 18

09/07/2024

En esta clase creamos una cuenta Giphy Developers que es la plataforma y API proporcionada por Giphy, que permite a los desarrolladores integrar contenido de GIFs animados en sus aplicaciones, sitios web y servicios.

Y aprendimos como plasmarlas en una pagina web.

SESION 19

16/07/2024

En esta clase tuvimos resolución de dudas. Y lo que veremos en el siguiente semestre en esta clase. Como scrum, react , metodologías de entrega de un proyecto de inicio a fin.