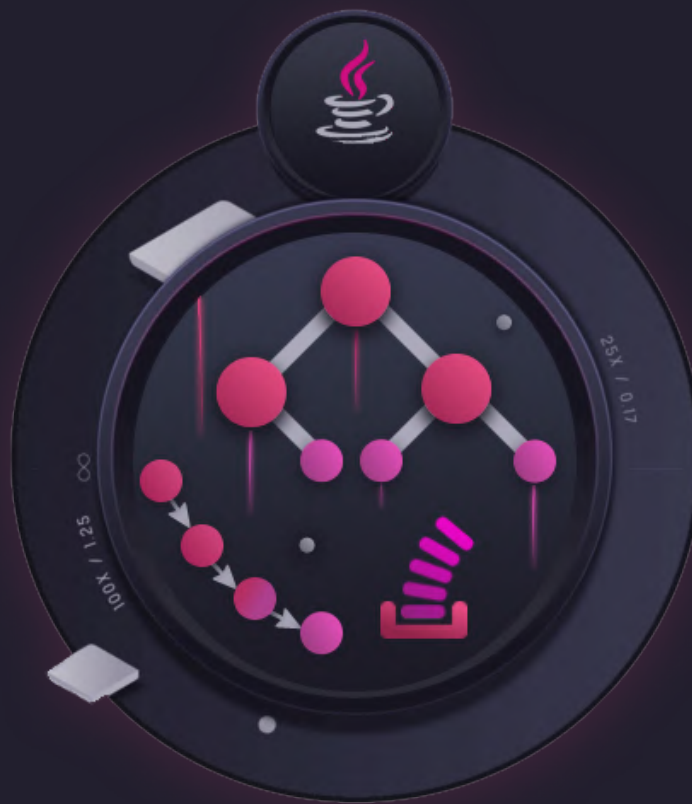


Guía de Problemas



Kevin Sotillo


Índice

Introducción	1
Ejercicios básicos para practicar Java	2
Diagramas de Clases	8
Vectores y Matrices	13
Algoritmos de Ordenamiento y Búsqueda	16
Listas Enlazadas	18
Pilas	21
Colas	24
Hash Tables	28
Árboles	32
Bibliografía	38

Introducción

En nuestra vida diaria como programadores comúnmente tenemos que hacer frente a problemas para poder administrar cierta información. Las estructuras de datos son colecciones que nos ayudan a poder organizar y manejar dicha información de manera bastante eficaz y eficiente. Estas estructuras, si bien quizás no tengamos que hacerlas nosotros mismos en el trabajo del día a día, es muy probable que las utilicemos en distintas instancias, directa o indirectamente, por ello es que es necesaria su comprensión, para poder entender como funcionan desde su implementación y que ventajas ofrecen con respecto a otro tipo de soluciones. A su vez, típicamente en entrevistas de trabajo de grandes compañías se pide al postulante que resuelva cierto tipo de problemas que requieren o son fáciles de resolver haciendo uso de estas herramientas, por lo que quizás si el argumento anterior a este no te convenció, el poder trabajar en alguna FANG (Alguna compañía como Facebook, Amazon, Netflix o Google) pueda ser incentivo para ti.

La presente guía está realizada con una selección de más de 120 problemas de las bibliografías citadas al final y con problemas propuestos por mi persona, con la finalidad de ayudar a los estudiantes a afirmar los conceptos de la Programación Orientada a Objetos, los diagramas de clases (UML), el uso de arreglos, algoritmos de ordenamientos, y las estructuras de datos dinámicas: Lista, Pilas, Colas, Hash Tables y Árboles.

Si bien todos los problemas están propuestos para que sean realizados en Java , los mismos pueden ser llevados a cabo fácilmente en cualquier otro lenguaje de programación que soporte la Programación Orientada a Objetos. Lo importante es que sean alcanzables para el lector y puedan ayudar a que este obtenga una mejor comprensión y manejo de estas utilidades.



Ejercicios básicos para practicar Java



1. Desarrolle un programa que imprima todos los números elevados cuadrados desde uno hasta un número indicado por el usuario.
2. Desarrolle un programa que pueda determinar si un número ingresado por el usuario es primo o no.
3. Wikipedia define los primos relativos de la siguiente manera:

En matemáticas, los números primos relativos son dos números enteros a y b que no tienen ningún factor primo en común. Dicho de otra manera, si no tienen otro divisor común más que 1 y -1. Equivalentemente son primos relativos, si y sólo si, su máximo común divisor (MCD) es igual a 1. Dos números primos relativos no tienen porqué ser primos absolutos de forma individual, 14 y 15 son compuestos, sin embargo son coprimos, pues su $MCD = 1$.

Por ejemplo, 6 y 19 son primos relativos, pero 6 y 27 no lo son porque ambos son divisibles por 3.

Dados dos números ingresados por el usuario, determine si son, o no, números primos relativos

4. Según el buen Wikipedia: *En matemáticas, la sucesión o serie de Fibonacci es la siguiente sucesión infinita de números naturales:*

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597...

La sucesión comienza con los números 0 y 1; a partir de estos, «cada término es la suma de los dos anteriores», es la relación de recurrencia que la define. A los elementos de esta sucesión se les llama números de Fibonacci.

5. Desarrolle un programa que calcule e imprima los primeros 10 números de la serie de Fibonacci.
6. Desarrolle un programa que calcule e imprima los números de la serie de Fibonacci menores a un número determinado por el usuario.
7. Realice un programa que calcule la potencia de un número, a través de las estructuras iterativas. La base y potencia deben ser ingresadas por el usuario. Una vez mostrado el resultado, se le debe preguntar al usuario si desea realizar otra operación. En caso afirmativo, se solicita de nuevo los datos y se vuelve a calcular.
8. Desarrolle un programa que, dado un año ingresado por el usuario, indique a que siglo pertenece. El primer siglo va desde el año 1 hasta el año 100 (incluido), el segundo desde el año 101 hasta el año 200 (incluido), etc.
9. Desarrolle un programa que determine si un string es palíndromo o no. Un string es palíndromo cuando no cambia luego de haberlo invertido.
10. Realizar un juego, donde se lanza un dado de 6 caras. Si el jugador consigue un 1 o 3, gana, si saca un 2 pierde, y si obtiene 4, 5 o 6, vuelve a lanzar. Se debe de investigar cómo generar un número aleatorio, entre 1 y 6.



11. Realizar un juego, donde se lanza un dado de 6 caras 10 veces. De acuerdo a los números obtenidos, el jugador irá acumulando puntos, que el sistema deberá ir mostrando en cada turno. Las reglas con los puntos son:

- ☐ 1: Gana 10 puntos.
- ☐ 2: Gana 25 puntos.
- ☐ 3: Pierde tantos puntos como sea el triple de lanzamientos realizados.
- ☐ 4: No consigue puntos.
- ☐ 5: Gana tantos puntos como sea el doble de lanzamientos restantes
- ☐ 6: Pierde 20 puntos.

Al final se le debe mostrar al usuario, dependiendo de la cantidad de puntos logrados los siguientes mensajes:

Si tiene más de 50 puntos: ¡Que grande sos papá! 😄 Sos dueño de este juego, ya le puedes decir a las demás personas: ¿Y vos? ¿Cuántas copas tenés? 🏆🏆🏆

Si tiene más de 0 puntos: Che vas re bien pero ponele más ganas boludo que el desarrollador no se partió el lomo programando esta poronga para que vengas a pasar pena 😓😓😓

Si tiene menos de 0 puntos: Noooooooooo amigo pero ¿Qué haces pibe? La estás re cagando loco, ponele más ánimo, poneleeeeeeeee 😞😞😞😞😞

Finalmente, se le debe preguntar al usuario si desea volver a empezar.

12. La Unidad de Radiología de una Clínica en Caracas, ofrece varios Estudios a sus clientes. Los Estudios son de tres tipos:

Estudio	Descripción	Valor en \$. x Estudio
U	Ultrasonido	100
T	Tomografía	130
R	Resonancia	160

Cuando el Cliente, solicita un Estudio en la Unidad de Radiología, se le solicitan los siguientes datos:

- ☐ Cédula de Identidad
- ☐ Edad
- ☐ Sexo (F=Femenino; M=Masculino)
- ☐ Si pertenece a un seguro
- ☐ El tipo de estudio (uno solo por cliente) que se le va a efectuar.

Cuando el cliente es Femenino y mayor de 55 años, o Masculino mayor de 65 años (Tercera Edad), el Sistema les otorga un descuento del 25%. Si pertenece a un seguro se le dará un descuento del 80%. A todos los clientes cuyo último dígito de cédula sea impar se le otorgará un 5% de descuento adicional.



Al final de la ejecución del programa debe de mostrarse:

1. Para cada cliente, el programa deberá desplegar la información del recibo con los siguientes datos:

- ☐ El número de su cédula de identidad
- ☐ Su edad
- ☐ Su sexo
- ☐ El tipo de estudio escogido
- ☐ El monto neto a pagar, habiéndole aplicado el descuento para los casos que

aplique.

2. Al final del día, luego de haber registrado todos los clientes.

- ☐ La cantidad de Clientes en total.
- ☐ La cantidad de Clientes por tipo de Estudio.
- ☐ El Monto Total de Descuentos Otorgados.
- ☐ El Monto Total Neto Facturado.
- ☐ El Promedio de Pago de todos los Clientes.
- ☐ El promedio de pago por tipo de estudio.
- ☐ Cantidad de descuento otorgado por seguro.

- 13.** La cuenta de instagram @creadas_ es una cuenta de manualidades que para San Valentín 🧡 está promocionando muchos regalos para parejas y amigos, debido a su crecimiento exponencial a lo largo de los últimos meses ellos proyectan que en promedio van a vender unos \$7.5 por pedido, y que la cantidad de pedidos va a ser más de 100. Es por ello que, para llevar un debido registro de todas las ventas llevadas a cabo con toda su respectiva información y precio final, se le ha pedido a usted que desarrolle un programa en Java ☕ que pueda cumplir con las siguientes necesidades:

El programa debe de pedir al usuario que ingrese:

- ☐ Nombre.
- ☐ Apellido.
- ☐ Cuenta de Instagram (sin el @).
- ☐ Nombre de la persona a quien se le va a dar el regalo.
- ☐ Apellido de la persona a quien se le va a dar el regalo.
- ☐ Cuenta de Instagram a quien se le va a dar el regalo (sin el @).
- ☐ Dirección de envío.
- ☐ Distancia en Km por envío.
- ☐ Mensaje que dedica la persona.

Una vez se tenga esos datos se procede a procesar el pedido del cliente, por lo que el usuario debe de escoger de los siguientes:

- ☐ Caja con flores: \$8

☞ El regalo tiene 5 flores, pero puede tener flores adicionales, si desea flores adicionales se pregunta el total y por cada una es \$0.5 extra.



- Cajita con decoración simple: \$5
 - ☞ La decoración puede ser con pintura luminiscente, lo cual tiene un costo adicional de \$3.
- Álbum con fotos de 12 páginas + 1 rosa de cartulina: \$15
 - ☞ Si desea fotos extra, entonces se cobra por páginas, se debe preguntar cuántas páginas extras con fotos desea y por cada una es \$1 más.
 - ☞ También puede ser rosas de cartulina extra, por cada una sería \$0.5 más.
- Mini pancarta con banderines + 1 rosa de cartulina: \$4.
 - ☞ La pancarta tiene la opción de incluir fotos, por cada foto a incluir son \$0.25 más.
 - ☞ De igual forma, pueden pedirse rosas de cartulina extra, por cada una sería \$0.5 más.
- Tarjeta explosiva en forma de corazón con 6 fotos + 1 rosa de cartulina: \$3.
 - ☞ Si desea fotos extra, entonces se cobra por páginas, se debe preguntar cuántas páginas extras con fotos desea y por cada una es \$1 más.
 - ☞ También puede ser rosas de cartulina extra, por cada una sería \$0.5 más.

Es posible escoger de un mismo producto cierta cantidad, es decir si se pidió el último producto cuatro rosas extras, entonces la cantidad sería ese mismo producto la cantidad de veces especificada por la cantidad.

Las políticas de la cuenta de Instagram con respecto al envío funcionan de la siguiente manera: para los envíos menores a 5 km el envío sale gratis, para aquellos menores a 10 km sale en 3\$, para aquellos menores a 15 km sale en 5\$. Si sobrepasa los 15 km entonces deberá decir que se le enviará por alguna agencia.

Los dueños del emprendimiento digital tienen unas políticas bastante particulares con respecto a los descuentos aplicados a las ventas realizadas, los cuales son:

- Si el nombre de la persona y apellido del cliente es más largo que el de la persona al que va dirigido el regalo, entonces se aplica un descuento del 5%. Si da la casualidad que tienen la misma cantidad de caracteres entonces ese descuento es del 15%
- Si el mensaje que la persona dedica es mayor a 300 caracteres, entonces se le aplica un descuento del 10% porque la persona puede escribir algo más largo que un tweet para su pareja 💕
- Si el nombre y el apellido del cliente son iguales a los de la persona al que va dirigido el regalo, entonces significa que se lo va a regalar a sí mismo, en ese caso se exonera el 100% del costo total y el mensaje que iba a dedicar la persona se reemplaza por "Sabemos lo importante que es quererse y valorarse a sí mismos, pero nunca está de más que alguien también lo haga, en este caso ese alguien más queremos ser nosotros, ¡Feliz 14F! 🥰🥰🥰🥰"
- Si el subtotal de lo pedido es un múltiplo de 5 entonces se le otorga un descuento del 25%.



Luego de haber registrado un pedido debe de salir un resumen de lo pedido de la siguiente manera:

```
----- RESUMEN -----  
----- @creadas_ -----  
  
De parte de Kevin Sotillo (@ksotillo) para Samán Excentrico (@samanexcentrico), con el siguiente mensaje:  
  
De parte de un fiel seguidor para los tqm <3  
  
Regala: Caja con flores ($8 o 13.824.00 Bs)  
Se envía a: La Universidad Metropolitana  
Con un costo de envío de: $5 o 8.640.00 Bs  
Un subtotal de: $25 o 43.200.00 Bs  
Un descuento de: $6.25 o 10.800.00 Bs  
Un total de: $23.75 o 41.040.00 Bs
```

Una vez finalizada la orden, se pregunta si se quiere volver a registrar otro pedido, de no ser así, se procede a dar la siguiente información como el resumen de todas las ventas registradas:

- ☐ Cantidad de pedidos hechos a lo largo del día
- ☐ Total de descuentos dados a lo largo del día.
- ☐ Total en ventas hechas a lo largo del día.
- ☐ Total del pedido más caro.
- ☐ Máxima distancia registrada para envío de un pedido.
- ☐ Cantidad de caracteres del mensaje más largo.
- ☐ Promedio de descuento por pedido.
- ☐ Promedio de ventas a lo largo del día, y si ese monto supera al promedio esperado o no.

Algunas consideraciones a tomar en cuenta:

- ☐ Los montos al final deben de mostrarse tanto en dólares como en bolívares.
- ☐ El monto en bolívares debe de ser calculado a la tasa del BCV del día.
- ☐ Pueden escogerse múltiples productos por cada registro.



Diagramas de Clases



1. Diseñe las clases necesarias para modelar una biblioteca. Donde tres tipos de materiales disponibles, que tienen en común el nombre del material, la cota, cantidad de ejemplares disponibles, los materiales pueden ser libros los cuales poseen un autor y un año de publicación, cintas de videos las cuales pueden estar en formato DVD o Blu-ray y por últimos mapas los cuales pueden ser sacados de la biblioteca o no. Se podrá retirar un libro, una cinta de video o un mapa (si está disponible) en calidad de préstamo, comprar libros, incorporar nuevos libros y desincorporar libros de la biblioteca.
2. Los productos que comercializa una determinada empresa son libros y CD. Cada producto viene caracterizado por un código que lo identifica de manera única, el precio base del mercado, y el IVA que se le aplica que es diferente en cada caso, el 8% para los libros y el 12% para los CD. Además, las características propias de cada tipo de producto son para los Libros: título, autor(es), año de publicación, editorial, ISBN y para los CD: título, intérprete, año de publicación.

Por otro lado, esta empresa aplica un 20% de descuento a los libros y un 10% a los CD. Por tanto, el precio de venta de cada producto se calculará como: $\text{precio base} + \% \text{IVA} - \text{dto}$. Realice el diseño de este sistema.
3. Se desea crear una aplicación para representar la fauna de un país. Se representarán los mamíferos, peces y aves. De los mamíferos se considerarán si viven en el agua o en la tierra. Además de proveer la flexibilidad de poder expandir dicha lista de animales para mejoras de la aplicación a futuro. Además, el sistema deberá guardar un historial de los primeros 100 animales clasificados. Realizar el diagrama de clases con las relaciones necesarias para poder crear dicha aplicación. Deberá especificar el tipo de relación entre las clases.
4. Se desea crear una aplicación para dibujar figuras geométricas. Con la aplicación se podrá dibujar: cuadriláteros, círculos y triángulos. Entre los cuadriláteros deberá representar cuadrados y rectángulos. Además de proveer la flexibilidad de poder expandir dicha lista de figuras geométricas para mejoras de la aplicación a futuro. El sistema deberá guardar un historial de las primeras 50 figuras creadas. Realizar el diagrama de clases con las relaciones necesarias para poder crear dicha aplicación. Deberá especificar el tipo de relación entre las clases.
5. Una Universidad desea elaborar un sistema de reservaciones de espacio físico. La universidad está formada por Aulas, Laboratorios, y Auditorios. Por otra parte, las Aulas pueden ser Aulas Sencillas o Aulas Multimedia. El sistema debe contener información sobre: identificación, capacidad y ubicación de cada espacio físico Igualmente se deberá tener información acerca de las reservaciones de las cuales se conocerá: fecha y hora de reservación, identificación del espacio reservado y nombre de la persona que reserva. Se quiere además mantener un registro con cada una de las reservaciones realizadas. Elabore el diagrama de clases para representar el nuevo sistema.
6. Cuando se visita una ferretería, es común hallar todo tipo de materiales y herramientas. Todas las herramientas poseen un conjunto de características comunes, que son de interés no solo para el dueño del negocio sino también para los clientes como, por ejemplo: el nombre, una pequeña descripción, el costo, algún código de fábrica asociado, el nombre del fabricante, etc. En algunas ferreterías, las herramientas las clasifican en tres grandes grupos, manuales, eléctricas y motoras.



Dependiendo del uso futuro por parte de los clientes, dichos tipos de herramientas se clasifican a su vez en: de albañilería, de plomería, o de electricidad.

☐ Las herramientas eléctricas tienen como característica principal el voltaje con el cual trabajan, incluso existen herramientas eléctricas que no solo trabajan con la corriente proveniente del tendido eléctrico, sino que pueden operar en base a la energía almacenada en baterías.

☐ Las herramientas motoras funcionan a base de un motor que posee cierta potencia, poseen además un pequeño tanque para el combustible y hay algunas que hasta necesita aceite. Con la avanzada tecnología actual, se han desarrollado herramientas que pueden trabajar a base de electricidad y a base de energía motora. Y ya se encuentran disponibles en mucha ferretería a pesar de su elevado costo.

- 7.** El periódico el “El informante”, desea construir un sistema informático para automatizar algunos de sus procesos, llevar un control de asignaciones y sacar ciertas estadísticas que son requeridas para el funcionamiento de la empresa. El periódico está dividido por secciones, cada sección tiene asignado un jefe de redactores y un grupo de redactores que están a su mando, los jefes de redactores son los que supervisan y deciden los artículos que serán publicados, mientras que los redactores son los que escriben los artículos, sean publicados o no. Para cualquiera de ellos se deberá conocer: nombre, cédula, sección en la que escribe.

☐ Un artículo en el periódico está compuesto por un título, un resumen, un cuerpo, cero o varias imágenes asociadas al artículo, una fecha de publicación (en caso de haber sido aprobado por el jefe de redactores correspondiente), una fecha de creación (fecha en la cual el redactor creo el artículo) y por su puesto el redactor que lo hizo. Además de esto, el periódico puede publicar anuncios comerciales de diversas clases, en diferentes secciones del periódico.

☐ El sistema debe tener la capacidad de guardar por anuncio comercial los siguientes datos, nombre y cedula de la persona que solicito su publicación, sección del periódico donde se publicara el anuncio, título del anuncio y cuerpo del anuncio, el cual debe ser entregado en las oficinas de la empresa como una imagen.

☐ También es posible solicitar al periódico la publicación de un anuncio clasificado, el cliente debe proporcionar los siguientes datos para el anuncio, precio, fecha de publicación, cantidad de días que debe aparecer la publicación, tipo de articulo vendido (vivienda o vehículo), si es vehículo debe indicar, marca, modelo, año color y Kilometraje y si es vivienda debe indicar m2 , cantidad de cuartos, cantidad de baños, cantidad de puestos de estacionamientos y si acepta ley de política habitacional para el pago. Elaborar el diagrama de clase.

- 8.** Usted ha sido contratado para realizar el diagrama de clases del siguiente sistema:

☐ Esta red está compuesta por hospitales, los cuales están identificados por su nombre, ubicación, capacidad, instalaciones especiales: quirófanos, salas de terapias, consultorios; de las cuales se debe conocer su identificación, ubicación y con qué cuentan. El administrar un hospital requiere saber cuál quirófano se usa a diario y personal médico y de enfermeras que lo uso, además del motivo.



☐ En estos hospitales hay un gran número de enfermeras, doctores, personal administrativo, de limpieza y de vigilancia; de todos ellos se conocen su nombre, edad, grado de instrucción.

☐ En el caso de los médicos la especialización, turno en el cual laboran y consultorio. De las enfermeras turno en el cual laboran, al igual que el personal de limpieza y de vigilancia.

☐ De cada hospital se desea conocer cantidad de pacientes atendidos a diario, además de sus nombres, edades, sexo, motivo de consulta y doctor que lo atendió; si fue necesario hospitalizarlo el motivo del mismo.

☐ Pero como sabemos manejan además un inventario de medicinas, de las cuales se debe tener el nombre y la cantidad en existencia. También cuentan con ambulancias de las cuales es necesario saber su placa y fecha en la cual se le hizo el último chequeo.

- 9.** Se desea crear una aplicación para representar el funcionamiento de un Banco. El Banco Nueva Era, posee varias sucursales en todo nuestro país. y una red extensa de cajeros electrónicos. Cada agencia o sucursal, cuenta con una taquilla externa que es atendida por un cajero externo, y obviamente sus taquillas internas que son atendidas de igual manera por cajeros. De las sucursales y cajeros se conoce la ubicación y un código único que los identifica. Cada taquilla es atendida por un cajero, y en ellas se pueden realizar diferentes tipos de operaciones, como lo son consulta, retiro, depósito o transferencia.

Una de las peticiones es que se maneje una aplicación que contenga las cuentas registradas en ese Banco, y permita reportar las mismas, donde tan solo se señale el status: activa, inactiva. Las cuentas están identificadas con un número y necesariamente se debe conocer el saldo, hay que almacenar los movimientos que se realizan y mostrarlos bien sea en un reporte por pantalla o en reporte en papel, que se emite para los clientes. Las cuentas pueden ser: Corrientes, Ahorro, FAL. Las operaciones que se pueden hacer para cada una de ellas son: consultar, depositar, retirar y transferir operación para la cual se necesita además el número de cuenta al cual se va a transferir.

Realizar el diagrama de clases con las relaciones necesarias para poder implementar dicha aplicación. Debe especificar el tipo de relación entre clases.

- 10.** Los nuevos dueños de una prestigiosa Farmacia, te quieren contratar para que desarrolles un sistema para llevar un control de sus productos (máximo de 100), sus clientes (máximo 100) y sus proveedores (máximo 20), sin embargo, antes de hacer eso, se necesita que usted realice el diseño de este sistema a partir de los siguientes requerimientos:

Esta farmacia, además de medicamentos, vende otro tipo de productos como lo son alimentos y cosméticos, donde cada uno de estos tiene un código de producto, un nombre, un precio y una cantidad en existencia. Los medicamentos tienen componentes: nutricionales o químicos, de los alimentos se conoce si son para personas diabéticas o no, y los cosméticos se caracterizan por tener una fecha de vencimiento. Por otro lado, se debe llevar un registro de los clientes, por lo que se tiene que conocer nombre, apellido, sexo, cédula, dirección y teléfono de cada uno de ellos. Los clientes pueden entrar a la farmacia y comprar todo tipo de productos, por lo que tienen una bolsa donde caben un total de 10 productos de cualquiera sea su categoría.



Además de los clientes y los productos, se desea llevar control de los proveedores, que son los que venden esos productos a la farmacia, de los cuales se lleva un registro de su RIF, nombre de empresa proveedora y tipo de producto que vende.

Realice el diagrama de clases correspondiente al sistema, indicando las relaciones entre las clases, atributos y métodos.

- 11.** El sistema de reserva de vuelos es un sistema que permite al usuario hacer consultas y reservas de vuelos, además de poder comprar los billetes aéreos de forma remota, sin la necesidad de recurrir a un agente de viajes humano. Se desea que el sistema de reservas sea accesible a través de la World Wide Web.

○ El sistema actualmente tiene un Terminal de Servicio de Reserva en donde se presenta un mensaje de bienvenida describiendo los servicios ofrecidos junto con la opción para registrarse por primera vez, o si ya se está registrado, poder utilizar el sistema de reserva de vuelos. Este acceso se da por medio de la inserción de un login previamente especificado (dirección de correo electrónico del usuario) y una contraseña previamente escogida y que debe validarse. Una vez registrado el usuario, y después de haberse validado el registro y contraseña del usuario, se pueden seleccionar las siguientes actividades:

- ☞ Consulta de vuelos.
- ☞ Reserva de vuelos.
- ☞ Compra de billetes.

- La consulta de vuelos se puede hacer de tres maneras diferentes:

- ☞ Horarios de Vuelos.
- ☞ Tarifas de Vuelos.
- ☞ Información de Vuelo

○ La consulta según horario muestra los horarios de las diferentes aerolíneas que dan servicio entre dos ciudades. La consulta según tarifas muestra los diferentes vuelos entre dos ciudades ordenados por su costo. La información de vuelos se utiliza principalmente para consultar el estado de algún vuelo, incluyendo información de si existen asientos disponibles y, en el caso de un vuelo para el mismo día, si éste está en hora. Se pueden incluir preferencias en las búsquedas, como fecha y horario deseado, categoría de asiento, aerolínea deseada y si se desean sólo vuelos directos. La reserva de vuelo permite al cliente hacer una reserva para un vuelo particular, especificando la fecha y horario, bajo una tarifa establecida. Es posible reservar un itinerario compuesto de múltiples vuelos, para uno o más pasajeros, además de poder reservar asientos.

○ La compra permite al cliente, dada una reserva de vuelo previa y una tarjeta de crédito válida, adquirir los billetes aéreos. Los billetes serán posteriormente enviados al cliente, o estarán listos para ser recogidos en el mostrador del aeropuerto antes de la salida del primer vuelo. Es necesario estar previamente registrado con un número de tarjeta de crédito válida para poder hacer compras de billetes, o bien proveerla en el momento de la compra. Además de los servicios de vuelo, el usuario podrá en cualquier momento leer, modificar o cancelar su propio registro, todo esto después de haber sido el usuario validado en el sistema.



Vectores y Matrices



1. Elabore un programa que construya un vector de N-casillas de números enteros y busque el mayor de los números contenidos en el vector.
2. Elabore un programa, que reciba dos objetos matrices A y B, ambas cuadradas de dimensión 3 y calcule:
 - ☐ La Suma de A y B
 - ☐ La multiplicación de A y B
3. Hacer programa que trabaje con una matriz MAT de números float y devuelva un vector, en el cual se almacenen las sumas de cada diagonal de la matriz. Solamente se almacenarán la diagonal principal y las que se encuentran por encima de ella.
4. Una matriz esparcida es una matriz en la cual la mayoría de los elementos son 0. Es un desperdicio almacenar estas matrices completas puesto que muy pocos elementos contienen información significativa. Una forma eficiente de representar una matriz esparcida $m \times n$ es un arreglo $k \times 3$, donde K es el número de elementos significativos. En ese arreglo se almacena el índice de la fila, el índice de la columna y el valor de los elementos distintos a cero de la matriz original. Esto se denomina una representación reducida. Escriba un método que dada una matriz esparcida de dimensión $m \times n$, genere la representación reducida de dicha matriz.
5. Escribir un método que intercambie la fila i-ésima por la j-ésima de un arreglo de dos dimensiones, $m \times n$.
6. Escribir un programa que permita visualizar el triángulo de Pascal. En el triángulo de Pascal, cada número es la suma de los dos números situados encima de él. Este problema se debe resolver utilizando un arreglo de una sola dimensión.
7. Se dice que una matriz tiene un punto de silla si alguna posición de la matriz es el menor valor de su fila y, a la vez, el mayor de su columna. Escribir un programa que tenga como entrada una matriz de números reales y calcule la posición de un punto de silla (si es que existe).
8. Escribir un programa en el que se genere aleatoriamente un arreglo de 20 números enteros. El arreglo ha de quedar de tal forma que la suma de los 10 primeros elementos sea mayor que la suma de los 10 últimos elementos. Mostrar el arreglo original y el arreglo con la distribución indicada.
9. Dado un arreglo de enteros, encuentre el par de elementos adyacentes que tengan el producto más grande y retorne el resultado de ese producto. Por ejemplo: Dado el arreglo { 3, 6, -2, -5, 7, 3 }, el resultado retornado por el programa debería de ser 21, dado que 7 y 3 dan ese producto.
10. El rey Julien recibió estatuas suyas de diferentes tamaños como un regalo por su cumpleaños. Dado que a él le gusta las cosas perfectas, quiere que se organicen desde la más pequeña hasta la más alta, de manera que cada estatua sea mayor a la anterior por exactamente un metro. Muy probablemente necesite añadir estatuas adicionales para cumplir con su cometido. Ayúdalo a encontrar el mínimo número de estatuas requeridas. Por ejemplo, para las siguientes estatuas: { 6, 2, 3, 8 }, se requeriría de las estatuas de tamaño 4, 5 y 7 metros.



11. Dado un arreglo de strings, retornar otro arreglo que contenga todos los strings más largos. Por ejemplo: dado el arreglo { "aba", "aa", "ad", "vcd", "aba" } el programa debería de retornar { "aba", "vcd", "aba" } dado que son los string más largos.

12. Dada una secuencia de numeros enteros en un arreglo, determine si es posible obtener una secuencia estrictamente ascendente, sin remover más de un elemento del arreglo.

Por ejemplo, dado el arreglo { 1, 3, 2, 1 } la respuesta debería de ser negativa dado que el arreglo no contiene ningún elemento que pueda eliminarse y que dicho arreglo quede como una secuencia ascendente. Por otro lado, dado el arreglo { 1, 3, 2 } es posible quitar el tres para que quede una secuencia ascendente { 1, 2 }. De igual forma, es posible eliminar el dos para obtener una secuencia ascendente de { 1, 3 }.

Nota: la secuencia a_0, a_1, \dots, a_n es considerada estrictamente ascendente si y sólo si se cumple que: $a_0 < a_1 < \dots < a_n$. Una secuencia que sólo contiene un único elemento también es considerada una secuencia estrictamente ascendente.

13. Un grupo de unimatanos decidieron luego de terminar su semana 12 irse a un plan chill en el Hotel Humboldt de Caracas, dónde pasarán la noche. Cada uno de las habitaciones tiene un coste distinto, y algunas de ellas están gratis por una promoción exclusiva, pero existe el rumor de que todas las habitaciones que están gratis también están embrujadas. Como nuestro grupo de unimatanos creen en el horóscopo y son bien supersticiosos, se rehusan a quedarse en cualquiera de esas habitaciones gratuitas, o cualquiera de las habitaciones que estén debajo de ellas.

Dada una matriz rectangular de enteros, dónde cada valor representa el costo de una habitación, se te ha otorgado la tarea de retorn el suma total de las habitaciones que son apropiadas para los unimatanos, es decir, sumar todos los valores que no aparecen debajo de un cero.

Por ejemplo, para la matriz: { { 0, 1, 1, 2 },
 { 0, 5, 0, 0 },
 { 2, 0, 3, 3 } }

El resultado debería de ser 9, dado que:

	1	1	2
	5		
2		3	3

Hay varias habitaciones embrujadas, por lo que no se contarán así como las habitaciones debajo de ellas. Por lo tanto, la respuesta es: $1 + 5 + 1 + 2 = 9$



Algoritmos de Ordenamiento y Búsqueda



1. Escriba el algoritmo de Selección Sort recursivamente.
2. Modifique el algoritmo de Inserción para que ordene el arreglo indirectamente. Esto requiere un arreglo que contenga los índices del arreglo que se desea ordenar. El algoritmo hace los cambios en el arreglo de índices y deja los datos originales sin modificar.
3. Escriba el algoritmo de Inserción recursivamente.
4. Modifique cada uno de los algoritmos vistos para que ejecuten el ordenamiento en orden decreciente (de mayor a menor).
5. Modifique el algoritmo de la Burbuja para que se detenga en cuanto el arreglo esté completamente ordenado.
6. Modifique el algoritmo QuickSort para que tome como pivote el último elemento en lugar del primer elemento de la secuencia.
7. Modifique el algoritmo QuickSort para que tome como pivote la mediana entre el primer, el del medio y el último elemento en lugar del primer elemento de la subsecuencia.
8. Dado un vector cualquiera, que contiene objetos con distintos atributos, defina un método que implemente el Mergesort sobre el vector, para cualquiera de los atributos del mismo. El método debe aceptar como parámetros el vector, el número del atributo por el cual se quiere ordenar (clave) y el orden que se desea (ascendente o descendente).
9. Dado un vector x de n elementos reales, donde n es impar, diseñar un método que calcule y devuelva la mediana de ese vector. La mediana es el valor tal que la mitad de los números son mayores que él y la otra mitad son menores. Escribir un programa de prueba.



Listas Enlazadas



- 1.** Dada una lista simple, se desea crear una nueva lista con los elementos en orden inverso. Utilice los métodos vistos en clase.
- 2.** Modifique la pregunta 1 para que a medida que va creando la lista nueva vaya destruyendo la lista original.
- 3.** Dadas tres listas simples, crear una nueva lista que contenga la suma de los elementos de las tres listas uno a uno, es decir los tres primeros, los tres segundos y así sucesivamente. Las listas no necesariamente son de la misma longitud
- 4.** Dada una lista simple y un número entero, se desea crear una nueva lista solo con aquellos elementos que son múltiplos de ese número dado.
- 5.** Dada una lista y dos valores enteros N y V. Cambiar todas las ocurrencias del valor V en la lista por el valor N. No debe crear una nueva lista solo trabajar sobre la misma lista original.
- 6.** Escriba un método para ordenar una lista.
- 7.** Modifique algún algoritmo de ordenamiento para ser aplicado a una lista.
- 8.** Escriba un método que dado convierta a la lista en un vector, es decir, dada una lista ingrese los datos de los nodos en un vector.
- 9.** Realice un algoritmo que dado un vector construya una lista.
- 10.** Escriba un método que, dado una lista, convierta la lista en un arreglo (vector) lo ordene a través del método selectionSort y luego devuelva una lista con los elementos del vector ordenado.
- 11.** Dada una lista doble ordenada ascendentemente, escriba un método que permita insertar un nuevo nodo manteniendo el orden.
- 12.** Escribir un programa para obtener una lista doblemente enlazada con los caracteres de una cadena leída desde el teclado. Cada nodo de la lista tendrá un carácter. Una vez que se haya creado la lista, ordenarla alfabéticamente y escribirla en pantalla.
- 13.** Utilizar una lista doblemente enlazada para controlar una lista de pasajeros de una línea aérea. El programa principal debe ser controlado por menú y permitir al usuario visualizar los datos de un pasajero determinado, insertar un nodo (siempre por el final) y eliminar un pasajero de la lista. A la lista se accede por dos variables, una referencia al primer nodo y la otra al último nodo. Realizar el sistema completo.
- 14.** Utilice una lista circular doblemente enlazada para simular la lista de reproducción de un sistema de audio, desarrolle algoritmos para agregar una canción a la lista, eliminarla y repetir la canción. Debe mostrar al Usuario cual es la canción que se reproduce y lista de reproducción si así lo desea el mismo. Realizar el sistema completo.



- 15.** El departamento de atención al cliente de la compañía de software MICROEVILSOFT posee una lista circular doblemente enlazada con todos los técnicos que trabajan en atención al cliente. El departamento de atención al cliente está dividido en grupos según los diferentes programas a los cuales se le presta soporte técnico, estos grupos son: Procesador de Palabras, Hoja de Cálculo, Presentación, Base de Datos, Sistema Operativo y Juegos. Cada grupo posee dos empleados. Cada integrante del grupo de soporte técnico de MICROEVILSOFT está obligado a atender un máximo de 8 clientes diarios. Si un técnico no está disponible al momento, este está en la obligación de devolver la llamada al cliente para prestarle sus servicios.

La compañía posee una lista Simple ordenada por hora de todas las llamadas que se han hecho al departamento de atención al cliente durante el día. De estas llamadas, se conocen los siguientes datos: el nombre de la persona, el número de teléfono donde encontrarlo, el programa de software con problemas y el problema que tiene con dicho software.

Se desea implementar un programa en Java ☕ el cual, dada las dos listas, les asigne llamadas a los técnicos correspondientes, tomando en cuenta la cantidad de clientes diarios que puede atender cada técnico. Desarrolle el sistema por completo.

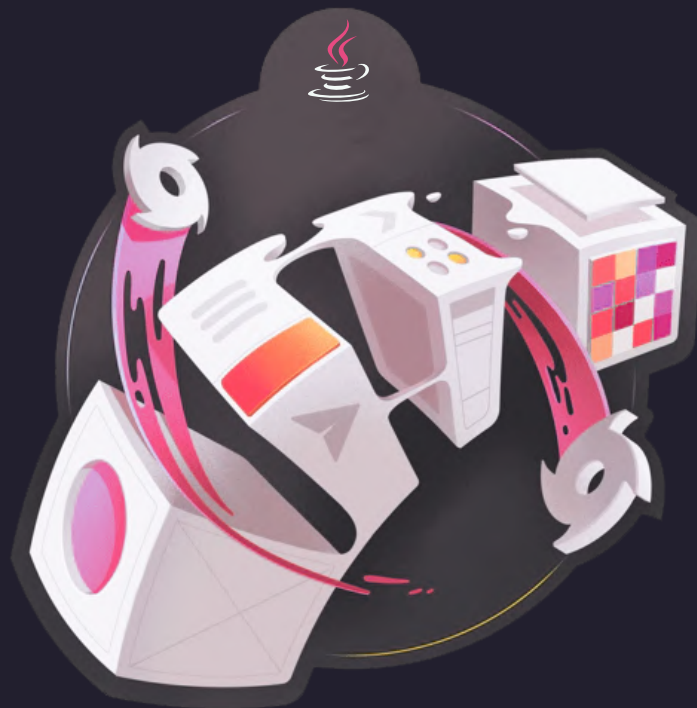
- 16.** Escriba un método recursivo para imprimir toda la información de una lista.
- 17.** Considere que se tiene una lista de números enteros, ordenados crecientemente, como la que se muestra a continuación. Observe que faltan algunos números para tener todos los valores comprendidos entre el primero, y el último. Escriba un método que complete la lista, de manera tal que la misma, una vez modificada, almacene todos los valores a partir del número del primer nodo hasta el número del último. Para el ejemplo la lista contiene a los números {7,8,10,13} y la lista debe guardar los siguientes números {7,8,9,10,11,12,13}.
- 18.** Su empresa ha sido contratada para desarrollar un programa para contralar las asistencias de los alumnos de una escuela. A su departamento se le ha encargado el desarrollo de un módulo que lee de las listas con los datos de los de los alumnos de un archivo de texto y que luego guarda esta información en una lista doblemente enlazada. Usted deberá implementar métodos que permitan:

- ☐ Promedio de notas de la sección, global y por materias
- ☐ Generar cortes de notas y boletas
- ☐ Determinar promedio de asistencia, ausencia y personas con reposo en la sección

Nota: El programa está siendo desarrollado orientado a objetos y usted solo conoce la implementación de la lista doble y de la clase alumnos, por lo cual sus métodos deberán estar allí.



Pilas



1. Desarrolle un método que permita buscar un elemento dentro de una pila.
2. Dada una pila de enteros, desarrolle los metodos necesarios que permitan conocer el mínimo y máximo de la pila.
3. Desarrolle la implementación de una cola utilizando dos pilas. Dichas implementación debe de soportar todas las funciones de una cola normal (push, peek, pop y isEmpty)
4. El profesor de estructura de datos desea elaborar un quiz, donde cada estudiante tenga una pregunta diferente, para ello ha creado una lista con la información de los estudiantes y una pila de preguntas a realizar, las cuales pueden estar repetidas. Se desea formar una estructura nueva donde aparezcan los estudiantes con una pregunta asignada sin repetirse las preguntas.
5. Decimos que una pila P es un sombrero de otra pila Q, si todos los elementos de P están en Q, en el mismo orden y en las posiciones más próximas a la cima. La pila nula se considera un sombrero de cualquier pila.
6. Desarrollar el código necesario para trabajar una Pila extensión de un vector y de una lista simple.
7. Se tiene una lista con los datos de los clientes de una compañía de telefonía celular, los cuales pueden aparecer repetidos en la lista, si tienen registrado más de un número telefónico. La compañía para su próximo aniversario desea enviar un regalo a sus clientes, sin repetir regalos a un mismo cliente. Los regalos se encuentran almacenados en una pila de regalos. Se desea elaborar un programa en Java ☕ que permita generar una nueva estructura donde los clientes aparezcan sólo una vez con sus regalos asignados.
8. Hacer un método que reciba una lista y una pila, el método deberá modificar la lista original, eliminando las posiciones indicadas por cada nodo de la pila.
9. Una pila A está contenida en una pila B, si todos los elementos de la pila A están en la pila B. Elabore un método en Java ☕ que verifique si una pila A está contenida en una pila B.
10. Escribir un método, copiarPila(), que copie el contenido de una pila en otra. El método tendrá dos argumentos de tipo pila, uno para la pila fuente y otro para la pila destino. No puede usar Estructuras de Datos auxiliares y solo poder usar los métodos, peek, pop y push.
11. Se quieren determinar las frases que son palíndromo, para lo cual se ha de seguir la siguiente estrategia: considerar cada línea de una frase; añadir cada carácter de la frase a una pila y, a la vez, a lista enlazada circular por el final; extraer carácter a carácter, simultáneamente de la pila y de la lista circular el primero, su comparación determina si es palíndromo o no. Escribir un programa que lea líneas y determine si son palíndromo.
12. El llamado a subrutinas normalmente implicar guardar todos los datos que se encuentran disponibles para el procesador en una pila y una vez que la subrutina acaba, los datos se obtienen de la misma y se devuelven a donde estaban. Desarrolle un método que permita simular el llamado a subrutina, guardando los datos en que se encuentren disponibles en el contexto actual en una pila, y que luego los devuelva al mismo contexto.



13. Diseñe una estructura similar a una pila que permite guardar elementos en la pila y sacar los elementos que más se repitan en la misma. Es decir, suponga que se tiene la pila { 5, 5, 6, 7, 5, 7 }, al hacer pop debería de retornarse 5, si se hace nuevamente retornaría 5 de nuevo, luego retornaría 7, y así sucesivamente, retornando siempre el elemento que más se repite.

14. Dadas dos secuencias llamadas pushed and popped, ambas con valores distintos retorne true si y sólo si el resultados de una secuencias de operaciones push y pop pudieron ser hechas en una pila inicialmente vacía. Por ejemplo:

Dados pushed = { 1,2,3,4,5 }, popped = { 4,5,3,2,1 } el resultado sería true, dado que podría darse la siguiente secuencia de operaciones sobre una pila: push(1), push(2), push(3), push(4), pop() -> 4, push(5), pop() -> 5, pop() -> 3, pop() -> 2, pop() -> 1

Por otro lado, dados pushed = {1,2,3,4,5}, popped = {4,3,5,1,2} el resultado sería false, dado que 1 no puede ser eliminado antes de 2.

15. Dada una lista llamada target y un entero n, en cada iteración se debe leer un número de una lista definida como list = { 1, 2, 3, ..., n }. Se debe de construir una pila con todos los elementos de target usando las siguientes operaciones:

- ☐ Push: lee un nuevo elemento desde el principio de la lista y lo guarda en la pila.
- ☐ Pop: elimina el último elemento de la pila.
- ☐ Si la pila ya tiene todos los elementos que tiene la lista target, entonces deja de leer elementos.

Se debe de retornar las operaciones hechas para construir la pila. Por ejemplo:

Datos target = [1,3], n = 3, la lista list creada sería list={ 1, 2, 3 } y el resultado a retornar debería de ser ["Push","Push","Pop","Push"] ya que se lee el número 1 de list se guarda en la pila, luego se guarda el número 2 y luego se elimina al no estar en la lista target, finalmente, se procede a guardar el número 3.

Por otro lado, dados target = [1,2], n = 4, la lista creada sería list = { 1, 2, 3, 4 } y el resultado a retornar debería de ser ["Push","Push"] ya que sólo se deben de leer los primeros dos elementos y detenerse.

16. Diseñe una pila con la operación Increment. Implemete la clase CustomStack con los siguientes métodos:

- ☐ CustomStack(int maxSize) la cual inicializa el objeto con maxSize el cual es máximo número de elementos en la pila, por lo cual no se pueden guardar más elementos en la misma si se llega a dicho límite.
- ☐ void push(int x) el cual añade x al tope de la pila si no ha llegado al límite de tamaño.
- ☐ int pop() el cual elimina y retorna el tope de la pila o -1 si la misma está vacía.
- ☐ void increment(int k, int val) el cual incrementa a los primeros k elementos de la pila por el valor de val (si val es 100, se incrementa los elementos en 100). Si hay menos de k elementos en la pila, entonces se incrementan todos los elementos de la pila.



Colas



1. Escribir un programa que invierta el contenido de una cola. Usted puede utilizar estructuras de datos auxiliares para hacerlo.
2. Usted debe realizar una optimización del control de las colas en los bancos, para ello emplee la estructura de dato cola de manera tal que reciba los datos de personas en una fila de un banco, estos son, nombre y conjunto de transacciones a realizar. La idea tras esta optimización es conocer el tiempo estimado de permanencia de cualquier persona en la cola, si se sabe que los tiempos estimados para cada tipo de transacción son:

Transacción	Tiempo Promedio (min)
Retiro	4
Depósito	2
Consulta	3.5
Actualización	5
Pago	2

3. Dada una pila y una cola que contienen letras almacenadas, en donde una letra puede estar contenida en ambas estructuras, elabore un método, que genere una lista con la información contenida en ambas, sin repetir las letras, que están en ambas.
4. Elabore un método que verifique si una pila A está contenida en una pila B. Se dice que una pila está contenida en otra pila, si todos los elementos de la primera pila están en la segunda pila.
5. Suponga que tiene ya codificados los métodos que implementan las operaciones del Cola. Escribir un método para crear una copia de una cola determinada. Las operaciones que se han de utilizar serán únicamente las del Cola.
6. Se tiene una pila de enteros positivos. Con las operaciones básicas de pilas y colas escribir método para poner todos los elementos que son par de la pila en la cola.
7. Escriba un método que invierta los elementos de una cola, solo puede usar una estructura auxiliar.
8. Escriba un método para ordenar una cola, puede usar Listas y Pilas auxiliares. *Nota: No utilice el método de ordenación de la clase Lista.*
9. Dada una cola de números, desarrolle la implementación de los métodos necesarios para encontrar el máximo y el mínimo.
10. Desarrolle la implementación de una cola circular doblemente terminada (deque). Su implementación debería de poder soportar las siguientes operaciones:



- ☐ `MiDequeCircular(k)`: Constructor, establece el tamaño de la cola en k.
- ☐ `insertarPrimero()`: Añade un elemento al inicio de la cola. Retorna true si la operación fue exitosa.
- ☐ `insertarUltimo()`: Añade un elemento al final de la cola. Retorna true si la operación fue exitosa
- ☐ `eliminarPrimero()`: Elimina un elemento al inicio de la cola. Retorna true si la operación fue exitosa.
- ☐ `eliminarUltimo()`: Elimina un elemento al final de la cola. Retorna true si la operación fue exitosa.
- ☐ `obtenerPrimero()`: Obtiene el elemento al inicio de la cola. Si la cola está vacía retorna -1.
- ☐ `obtenerUltimo()`: Obtiene el elemento al final de la cola. Si la cola está vacía retorna -1.
- ☐ `isEmpty()`: Revisa si la cola está vacía o no.
- ☐ `isFull()`: Revisa si la cola está full o no.

11. Desarrolle la implementación de una pila usando dos colas. La implementación de dicha pila debe de soportar todas las funciones de una pila normal. Sólo pueden usarse operaciones estandar de una cola.
12. Desarrolle un método que permita eliminar todos los elemento duplicados en una cola.
13. Implemente la función `queue_sum` el cual recibe una cola y retorna la suma de los elementos en ella. No debe de usarse estructuras auxiliares y la cola recibida por parametros debe de tener los elementos en el mismo orden de como se recibió en el momento en el resultado es retornado por la función.
14. Se tiene una cola de impresión donde se almacenan documentos* que se deben imprimir. En la medida en que llega un nuevo, trabajo éste se encola. Cuando la impresora se libera se toma un trabajo de la cola y se imprime. Utilice la clase previamente implementada para realizar una clase `ColaImpresion` con un atributo de la clase cola para simular el sistema previamente descrito.
*Desarrolle la clase Documento.
15. Modifique la pregunta anterior para que el sistema de impresión cuente con dos impresoras y varias colas de impresión, en donde el sistema asignará a la impresora libre el trabajo que más tiempo lleve en espera entre los posibles a procesar.
16. El supermercado Esperanza quiere simular los tiempos de atención al cliente a la hora de pasar por la caja. Los supuestos de los que se parte para la simulación son los siguientes:
 - ☐ Los clientes forman una única fila. Si alguna caja está libre, el primer cliente de la fila es atendido. En el caso de que haya más de una caja libre, la elección del número de caja por parte del cliente es aleatoria.
 - ☐ El número de cajas del que se dispone para atención a los clientes es de tres, salvo que haya más de 20 personas esperando en la fila; entonces se habilita una cuarta caja, que se cierra cuando no quedan clientes esperando. El tiempo de atención de cada una de las cajas está distribuido uniformemente: la caja 1 entre 1,5 y 2,5 minutos; la caja 2 entre 2 y 5 minutos, la caja 3



entre 2 y 4 minutos. La caja 4, cuando está abierta, tiene un tiempo de atención entre 2 y 4,5 minutos.

☐ Los clientes llegan a la salida en intervalos de tiempo distribuidos uniformemente, con un tiempo medio de 1 minuto.

El programa de simulación se debe realizar para 7 horas de trabajo. Se desea obtener una estadística con los siguientes datos:

☐ Clientes atendidos durante la simulación.

☐ Tamaño medio de la fila de clientes.

☐ Tamaño máximo de la fila de clientes.

☐ Tiempo máximo de espera de los clientes.

☐ Tiempo en que está abierto la cuarta caja.



Hash Tables




1. Uno de los usos más extendidos de las Hash Table es la de guardar los nombre de usuarios y contraseñas. Desarrolle un programa que, haciendo uso de Hash Table, imite la funcionalidad de registro y login en un sistema. Cuando un usuario se registre debe de guardarse sus datos en el sistema, mientras que cuando un usuario intente loguearse busque los datos ingresados por el usuario, si se encuentran el programa debe de indicar que la autenticación fue exitosa, de lo contrario debe de indicarse cuál es el problema, ya sea que el usuario o la contraseña estén mal, o bien el usuario no existe en el registro.

2. Desarrolle la implementación de un HashSet. Un HashSet es parecido a un HashTable pero no admite elementos repetidos. Dicha implementación debe de contar con los siguientes métodos:

- ☐ void agregar(llave): Inserta el valor llave en el HashSet
- ☐ bool contiene(llave): Retorna si el valor en llave existe o no el HashSet
- ☐ void eliminar(llave): Eliminar el valor perteneciente a llave en el HashSet. Si no existe la llave, entonces el programa debería de alertar al usuario sobre el error.

3. En una Universidad necesitan crear un nuevo sistema que permita la carga de datos de los estudiantes de manera fácil y efectiva, actualmente se tienen todos los datos en archivos CSV, cada fila en el archivo cuenta con los siguientes atributos:

- ☐ Nombre
- ☐ Apellido
- ☐ Edad
- ☐ Cédula
- ☐ Carrera

La universidad le ha pedido que desarrolle un programa escrito en Java  que emule dicha carga de datos a una Hash Table, para probar su implementación y luego hacer la carga real en la base de datos. El programa debe de leer los datos fila por fila en el CSV e irlos cargando en la Hash Table, finalmente debe de mostrar una tabla con todos los datos cargados.

4. Se requiere que su equipo de trabajo construya un programa en Java que lea un archivo de texto contentivo del resumen de una investigación y busque en él la frecuencia con la que aparezcan cada una de las palabras claves registradas en el repositorio del sistema. Como resultado del análisis, una estadística es mostrada en la pantalla en la que aparezca:

- ☐ Nombre del trabajo.
- ☐ Autores.
- ☐ Para cada una de las palabras clave se debe de obtener la frecuencia con la cual aparece en el resumen.
- ☐ Número de palabras del resumen.
- ☐ Tiempo empleado en el análisis.

El programa debe de cumplir con cada uno de los siguientes requerimientos:

- ☐ Agregar resumen. Un resumen está contenido en un archivo de texto, tal y como se muestra en el el texto de ejemplo al final del ejercicio. Esta funcionalidad consiste en:



☞ Cargar el archivo en el que se encuentra el resumen y agregarlo a una tabla de dispersión. Es decir, se debe diseñar una función hash que genere una clave única que permita tanto almacenar como buscar dicho resumen en la tabla con una complejidad lo más cercana a $O(1)$. Dicha clave debe ser generada a partir del título de la investigación.

☞ Por cada cada resumen, se guarda: título, autores, cuerpo del resumen y palabras claves.

☞ El programa deberá validar que no se introduzcan dos veces al mismo resumen.

○ Analizar resumen. Para la ejecución de esta funcionalidad, la secuencia de pasos es la siguiente:

☞ El sistema muestra la lista de las investigaciones guardadas, preferiblemente en orden alfabético.

☞ El usuario selecciona la investigación.

☞ El sistema analiza el texto correspondiente al resumen y se imprime en pantalla las estadísticas correspondientes, tal y como se describió en el planteamiento del problema.

○ Buscar Investigaciones. Dada una palabra clave, ver listado de las investigaciones relacionadas. Esta operación debe ser lo más cercano a $O(1)$.

○ Salir del sistema: se deben guardar la información de los resúmenes de manera que el usuario pueda acceder a dicha información en una siguiente oportunidad.

Es necesario que al comenzar el sistema, un conjunto de palabras claves, así como de investigaciones deben ser precargadas.

En la siguiente página se deja un resumen de un trabajo de investigación como ejemplo.



GraphQL vs REST: una comparación desde la perspectiva de eficiencia de desempeño.

Autores

Christian Guillén Drija
Reynaldo Quintero
Asher Kleiman

Resumen

El estilo arquitectónico conocido como Transferencia de Estado Representacional (REST según sus siglas en inglés) ha modelado y estandarizado la comunicación cliente-servidor a través del protocolo de transferencia de hipertexto (HTTP). Según Roy Fielding, creador de REST, este estilo consiste en un conjunto de restricciones arquitectónicas que enfatizan la escalabilidad de las interacciones entre los componentes, así como la generalidad de las interfaces utilizadas, el despliegue independiente de dichos componentes, la aplicación de componentes intermediarios para la reducción de la latencia, la preponderancia de la seguridad y la encapsulación de los sistemas Legacy. Como consecuencia, REST se ha convertido en el estilo más aplicado en la comunicación con una Interfaz de Programación de Aplicaciones (API). Por otra parte, GraphQL es una tecnología basada en un lenguaje de consultas de datos para API's que se ha constituido en una alternativa al estilo REST. En este trabajo de investigación, se realizó una comparación de REST y GraphQL como mecanismos arquitectónicos desde una perspectiva de calidad, específicamente centrada en la Eficiencia de Desempeño, en el marco del estándar ISO/IEC 9126 e ISO/IEC 25000. Las subcaracterísticas de Eficiencia de Desempeño evaluadas fueron: Comportamiento Temporal, Utilización de Recursos y Capacidad. Para cada una de dichas subcaracterísticas, se aplicaron métricas, a saber: para el Comportamiento Temporal se aplicaron las métricas Tiempo de Respuesta y Rendimiento; para la Utilización de Recursos, se aplicaron las métricas: Utilización de la Memoria, Utilización de Caché, Overfetching y Underfetching; y para Capacidad, se aplicaron las métricas Capacidad y Velocidad Bajo Estrés. Para el desarrollo de este trabajo se seleccionó un caso de estudio que permitiera evidenciar las características de los dos mecanismos. Asimismo, para la obtención de datos cuantitativos que permitiesen establecer comparaciones, se diseñaron dos API's que con base en el caso de estudio escogido, respondieran fielmente a las restricciones que caracterizan a los dos mecanismos arquitectónicos. Los resultados obtenidos, una vez ejecutados los casos de prueba, permiten inferir que REST posee una mayor rapidez de respuesta que GraphQL. Debido a la naturaleza dinámica de las peticiones que pueden realizarse en las que se puede especificar con mayor granularidad la información requerida a través de consultas anidadas, GraphQL parece ser más eficiente en lo relativo al underfetching y al overfetching. En general, GraphQL debería utilizarse cuando se requiere menor utilización de la memoria, cuando los recursos que la API maneja están estrechamente relacionados y cuando no sea necesario utilizar la totalidad de la información obtenida. Por otra parte, REST debería utilizarse en caso de necesitarse mayor velocidad de respuesta y cuando se requiera utilizar la totalidad de la información de los recursos manejados.

Palabras Claves: REST, GraphQL, cliente-servidor, mecanismo arquitectónico, calidad de software.



Árboles



1. Implemente la Clase ABB (Árbol Binario de Búsqueda) con los siguientes métodos:

- ☐ estaVacio()
- ☐ insertar(Nodo o)
- ☐ buscar(Object Clave)
- ☐ recuperar(Object Clave), devuelve un Nodo.

2. Escriba un método llamado crecer que tome como argumento un árbol de enteros y un valor entero n. Debe devolver un árbol binario de búsqueda, pero sustituyendo las hojas con el valor n por nodos que tengan como hoja izquierda $n/2$ (división entera) y como hoja derecha $n/2$ si n es par y $(n \% 2)$ si n es impar. Debe retornar NULL si el argumento no es un árbol o es un árbol, de un sólo nodo.

3. Elabore un método de que reciba una matriz de números y genere un árbol binario de búsqueda a partir de los elementos de la matriz.

4. Dos árboles binarios de búsquedas son similares, si ambos son vacíos o si ambos no vacíos, tienen subárboles izquierdos y derechos similares. Elabore un método que permita indicar si dos árboles binarios de búsquedas son similares.

5. Dado un árbol binario de búsqueda, cuyo contenido sean números enteros. Elabore el método sumar el cual suma todos los valores contenidos en el árbol.

6. Elabore un método que, dado un árbol binario de búsqueda, determine la cantidad de nodos que este posee.

7. La profundidad de un árbol binario de búsqueda, es la mayor cantidad de nodos que hay desde la raíz hasta el último nodo hijo de cada subárbol. Implemente un método que determine la profundidad de un árbol binario de búsqueda.

8. Se dice que el nivel de un nodo es la longitud del camino que lo conecta al nodo raíz. Desarrolle un método que, dado un árbol binario de búsqueda y una clave de búsqueda, determine el nivel de este nodo.

9. Se dice que un árbol Binario de búsqueda balanceado es aquel árbol que la misma profundidad para cada subárbol. Elabore un método que determine si un árbol binario de búsqueda es balanceado o no.

10. Implemente el método podar() el cual, elimina todas las hojas de un árbol binario de búsqueda.

11. Se dice que un nodo es papá de hoja cuando los dos subárboles son hojas. Desarrolle un método que determine si un nodo es papá de hoja.

12. Modifique el método de la pregunta 10 para que reciba un valor de tipo entero y elimine las hojas basados en estos criterios:

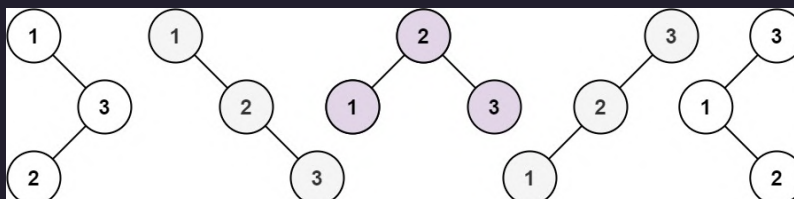
- ☐ Si el valor del papá de la(s) hoja(s) es el mismo que el valor pasado por parámetro elimine todas las hojas.



○ Si no se cumple lo anterior, se debe evaluar si el contenido del nodo es par, si lo es se debe eliminar el hijo izquierdo.

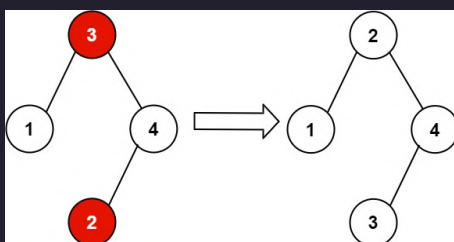
○ Si no se cumple ninguna de las dos condiciones previas, se debe eliminar el hijo derecho.

- 13.** Elabore un método que reciba un árbol binario de búsqueda y una pila, de números (ambas estructuras se encuentran desordenadas) y genere un vector con los elementos de ambas estructuras en forma ordenada.
- 14.** Dado un árbol binario de búsqueda que contenga números enteros. Desarrolle un método que imprima el mayor de todos los valores, y el promedio de los mismos.
- 15.** Dado un árbol binario de búsqueda realice los métodos de recorrido en inorden, postorden y preorden
- 16.** Dado un entero n , desarrolle una función que retorne el número de Árboles Binarios de Búsqueda estructuralmente únicos que tienen exactamente n nodos de valores únicos que van desde 1 hasta n .
- 17.** Por ejemplo: dado $n = 3$, se debería de retornar 5 ya que ese es el número de Árboles Binarios de Búsqueda estructuralmente únicos posibles que se pueden crear, como bien se puede apreciar en la siguiente imagen:

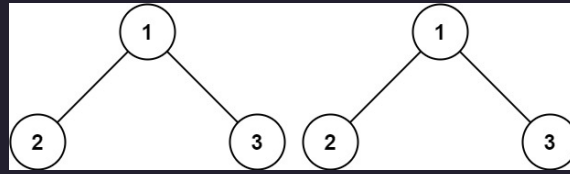


- 18.** Modifique el ejercicio anterior para que no retorne el número de posibles casos, sino todos los posibles árboles en sí, es decir, que retorne las estructuras creadas, en una lista.
- 19.** Dada la raíz de un Árbol Binario, determine si dicho árbol es un Árbol Binario de Búsqueda válido.
- 20.** Dada la raíz de un Árbol Binario de Búsqueda, donde dos de los nodos dentro del árbol fueron cambiados por error, desarrolle un método llamado `recoverTree()` que permita arreglar de nuevo el árbol sin cambiar su estructura.

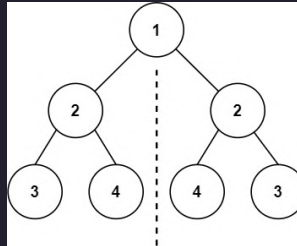
A continuación se muestra un árbol que sería dado como input de este ejercicio y cuál sería el valor a obtener:



- 21.** Dada las raíces de dos Árboles Binarios p y q, escriba una función que permita validar si son iguales o no. Se considera que dos Árboles Binarios son iguales si son estructuralmente iguales y si los nodos tienen el mismo valor. A continuación se muestran dos árboles binarios iguales:

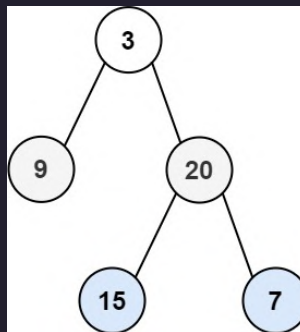


- 22.** Dada la raíz de un Árbol Binario, valide si dicho árbol es simétrico o no. A continuación se muestra un ejemplo de un árbol simétrico:



- 23.** Dada la raíz de un Árbol Binario, desarrolle una función que retorne el orden transversal por nivel de los valores de sus nodos en forma de una matriz (debe de ser de derecha a izquierda por cada nivel).

Por ejemplo: Dado el siguiente Árbol Binario:



El valor a retornar sería la siguiente matriz: $[[3], [9, 20], [15, 7]]$

- 24.** Modifique el ejercicio anterior para que retorne el orden transversal por nivel en zigzag de los valores de los nodos en forma de una matriz (debe de ser de izquierda a derecha, luego de derecha a izquierda en el siguiente nivel, y así alternando entre cada nivel)

El valor a retornar, tomando el Árbol señalado anteriormente sería: $[[3], [20, 9], [15, 7]]$

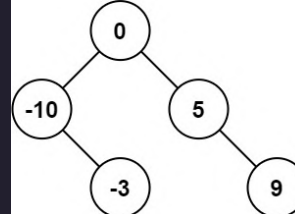
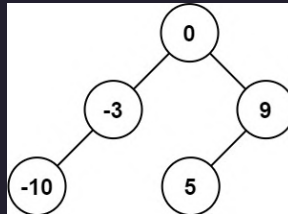
Modifique el ejercicio número 23 para que en vez de retornar el orden desde arriba hacia abajo, retorne el mismo orden pero de abajo hacia arriba, es decir, de izquierda a derecha, nivel por nivel desde las hojas hasta la raíz.



El valor a retornar, tomando el Árbol señalado en el ejercicio 23 sería: `[[15,7],[9,20],[3]]`

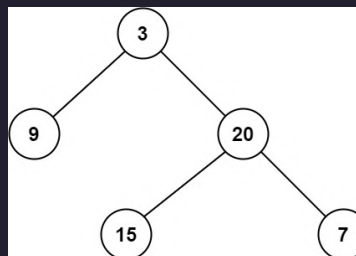
- 25.** Dada una lista de números enteros dónde los elementos están ordenados en orden ascendente, desarrolle una función que lo convierta en un Árbol Binario de Búsqueda balanceado.

Por ejemplo: dada la siguiente lista: `nums = [-10,-3,0,5,9]`. Los posibles Árboles resultantes (pueden haber varias posibles soluciones) son:



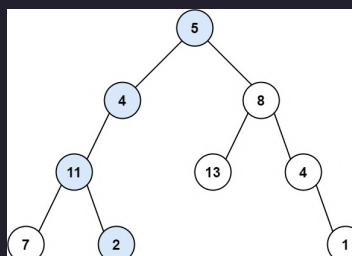
- 26.** Dado un Árbol Binario, desarrolle un método que permita obtener su profundidad mínima. La profundidad mínima es el número de nodos a lo largo de la ruta más corta desde el nodo raíz hasta el nodo hoja más cercano.

A continuación se muestra un Árbol Binario de Búsqueda cuya profundidad mínima es 2:



- 27.** Dada la raíz de un Árbol Binario y un número entero llamado `targetSum`, desarrolle un método que retorne `true` si el árbol posee una ruta desde la raíz hasta una hoja, de manera tal de que sumando todos los valores a lo largo de la ruta, el resultado de igual a `targetSum`, de lo contrario deberá retornar `false`.

Por ejemplo, dado el siguiente árbol y `targetSum = 22`:

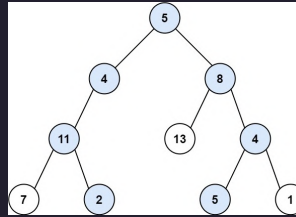


La ruta resaltada en color azul muestra que si hay una ruta cuya suma todos los valores es equivalente a 22, por lo tanto la función retorna `true`.



- 28.** Modifique el ejercicio anterior para que retorne no un booleano, sino una lista con todas las posibles rutas desde la raíz hasta una hoja que puedan obtener el valor de targetSum.

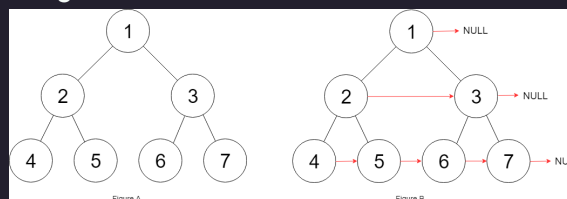
Por ejemplo, dado el siguiente árbol y targetSum = 22:



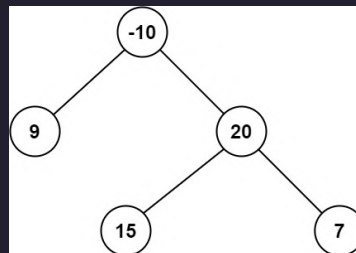
Se retorna la siguiente lista que contiene las dos rutas que permiten obtener dicho resultado:
[[5,4,11,2],[5,8,4,5]]

- 29.** Por políticas de negocios de una empresa se le ha pedido que haga un ajuste a la implementación de los Árboles Binarios, ahora cada uno de sus nodos debe de tener apuntadores a sus nodos que se encuentra en la derecha. Entonces, se le ha dado un árbol binario perfecto, dónde todas las hojas están el mismo nivel, y cada hoja padre tiene dos nodos hijos, a cada nodo debe de añadirse la propiedad Node derecha; desarrolle un algoritmo que permita establecer para cada nodo en el árbol dado su nodo derecho, si no hay nodo derecho, entonces el apuntador debe de establecerse en null.

A continuación se muestra un árbol (figura A) cuyos nodos no poseen apuntadores, la función a desarrollar procede a establecer cada apuntador al nodo que está justo a la derecha de cada uno, obteniendo así al árbol en la figura B.



- 30.** Por políticas de negocios aún más extrañas de una empresa, se le ha pedido que dada la raíz de u árbol, retorne la ruta con mayor suma dentro de los valores de los nodos del árbol. Una ruta en un árbol binario de búsqueda es una secuencia de nodos dónde cada par de los nodos adyacentes en una secuencia tiene una conexión que los relaciona. Un nodo puede sólo aparecer una única vez en una ruta. La ruta no necesariamente tiene que pasar por la raíz. A continuación se muestra un ejemplo, dado el siguiente árbol:



La ruta con la suma mayor entre todas las rutas posibles da un total de 42, dado que el camino 15 => 20 => 7 da la operación 15 + 20 + 7 = 42, retornando dicho valor como resultado.



Bibliografía

1. O. Cairó y S. Guardati, Estructuras de Datos, Monterrey México: McGraw-Hill.
2. L. Joyanes Aguilar y I. Zahonero Martínez, Estructuras de datos en Java, Aravaca (Madrid): McGraw-Hill, 2008.
3. C. Canelón, Programateca, Caracas: Universidad Metropolitana.
4. José R. Quevedo, Guía de Problemas, Caracas: Universidad Metropolitana.
5. Problems - LeetCode. (2021). Recuperado de <https://leetcode.com/problemset/all/>
6. CodeSignal. (2021). Recuperado de <https://app.codesignal.com>

