

## ALGORITMOS Y ESTRUCTURAS DE DATOS

### Examen Evaluativo 2

Nombre: \_\_\_\_\_ Cuenta: \_\_\_\_\_

UNAH

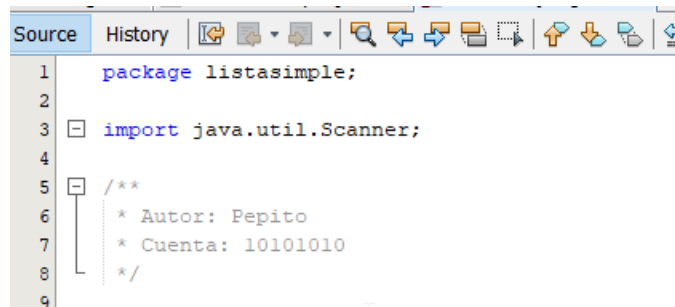
Ing. en Sistemas

## Instrucciones

1. Desarrolle, empleando la herramienta vista en clase, los ejercicios abajo descritos en un proyecto para los ejercicios de LISTAS y otro para el de LISTAS DOBLES. Asegurarse de que exportan el proyecto para simplificar su revisión (podrá castigarse la falta de este aspecto).

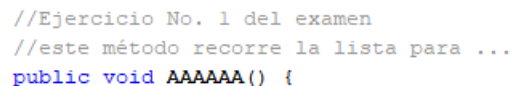
### 2. Documentación

- a. Cada archivo del proyecto deberá estar identificado con su nombre y número de cuenta



```
1 package listasimple;
2
3 import java.util.Scanner;
4
5 /**
6  * Autor: Pepito
7  * Cuenta: 10101010
8  */
9
```

- b. Cada método solicitado (u otros creados que no son parte del TAD), deben de identificar a que número de ejercicio corresponde y además agregar una descripción de lo que hace; indicando por lo menos los argumentos que recibe y la salida del mismo.



```
//Ejercicio No. 1 del examen
//este método recorre la lista para ...
public void AAAAAA() {
```

***\*\*esto es obligatorio y podrá castigarse con puntos la falta de éste aspecto.***

3. Al finalizar, suba al espacio en la plataforma un archivo comprimido con los dos proyectos, en el caso de que la plataforma fallara por alguna razón, entonces enviarlo al correo electrónico.
4. Este examen es de carácter individual y está sujeto a las regulaciones académicas de la Universidad y de la clase.
5. Usted podrá entregar el examen hasta 1 hora después de la hora límite indicada, pero esto será un castigo directo a su calificación de hasta 25pts del valor obtenido en el examen.

## Ejercicios LISTAS SIMPLES Y ORDENAMIENTO (65%)

---

En la clase de Algoritmos, sección 4pm, se ha planificado que los estudiantes realicen una guía de ejercicios cada semana (durante 7 semanas), por lo que se quiere un programa que permita almacenar la información de cada nota que obtiene un estudiante cada semana. Se le pide que, a través de listas, usted maneje la información antes descrita y que además pueda:

1. (5%) Toda la interacción debe hacerse a través de un menú para el usuario.
2. (5%) Asegurarse de tener en funcionamiento:
  - Método mostrar datos de la lista
  - Tener un método que llene al menos 5 datos de prueba de la S1 y 5 de la S2 y que se llamado cuando arranca el programa.
3. (10%) Guardar la información: **número de cuenta, nota y la semana** a la cual pertenece dicha nota. Algunas consideraciones:
  - No debe permitir ingresar dos notas para el mismo estudiante en la misma semana.
  - Las semanas se identifican con números que van de la 1 a la 7 (no se permite otro valor).
4. (5%) En cualquier momento se puede corregir la nota de un estudiante de una semana específica, esto es normal ya que puede haber reclamos o revisiones a las mismas.
5. En cualquier momento, poder sacar
  - (10%) listado de notas para un estudiante específico, por ejemplo, el estudiante con número de cuenta 20171030524:

Semana	Nota (%)
1	95
2	90
3	100
4	80
Promedio	91

- (20%) Hacer un cuadro resumen de los trabajos entregados por semana, por ejemplo, si ya se hubieran entregado las primeras cuatro guías se tendría:

Semana	Cantidad Entregaron	Promedio
1	25	90%
2	21	85%
3	26	96%
4	19	78%

*\*\* Los cuadros presentados, a nivel de valores, colores y diseño, solo es para ejemplificar... lo importante es obtener la información*

6. (10%) Dado que los datos pueden ser ingresados en cualquier momento o ser modificados, hacer un método que ordene los mismos según la semana (ascendentemente) y la nota (descendentemente).

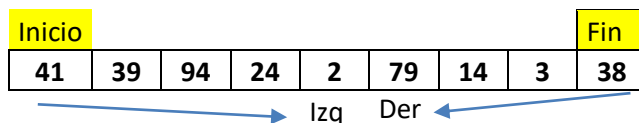
**Nota:**

1. No se permite el uso de las librerías de Java para manejo de LISTAS, debe seguir los lineamientos vistos en clase del TAD.

## Ejercicios LISTA DOBLE (35%) – HACER EN OTRO PROYECTO

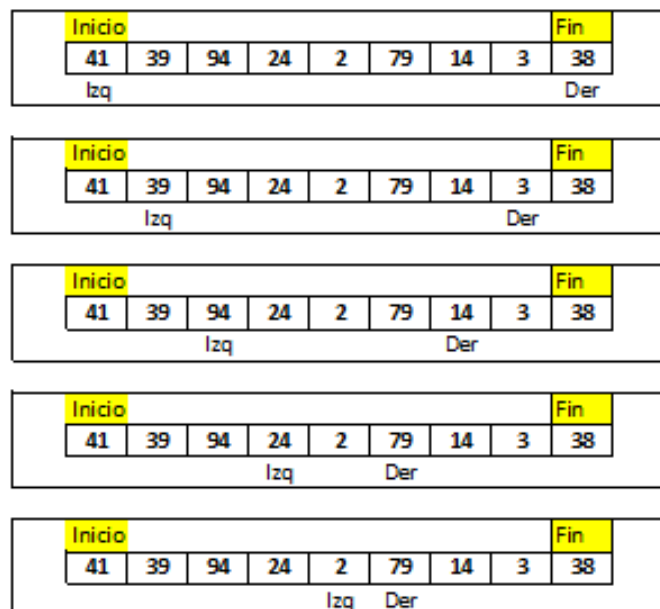
1. (5%) Toda la interacción debe hacerse a través de un menú para el usuario, debe contar con al menos las operaciones: ingresar al inicio, mostrar, mostrar al revés.
2. (5%) Hacer un método que llene la lista doble con N números aleatorios entre 1 y 100, usar el método insertar al inicio. Una vez que los haya insertado que muestre la lista de números generados.
3. (25%) hacer un método que, en el mismo ciclo, mueva dos punteros auxiliares (izquierda y derecha) hasta que estos llegan a la posición más cerca posible. Tome en cuenta los siguientes aspectos
  - 3.1. Siempre se mueve primero izquierda y luego derecha cuidando siempre que no se encuentren.
  - 3.2. Mientras no se encuentren deberá ir sumando los elementos por los que van pasando, al final deberá mostrar la suma de ambas partes del arreglo.

Ejemplo, si la lista tuviera un número de elementos impar:



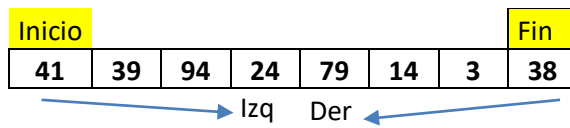
SumaIzquierda:  $41+39+94+24+2 = 200$   
SumaDerecha:  $38+3+14+79 = 134$

Ejemplo de las iteraciones realizadas del ejemplo anterior:



**\*\* Termina: no se puede mover derecha porque "chocaría" con izquierda**

Ejemplo de una lista con una cantidad de elementos par



SumaIzquierda:  $41+39+94+24 = 198$   
SumaDerecha:  $38+3+14+79 = 134$

3.3. No se puede usar una función que “cuente” la cantidad de nodos, por tanto, al desconocer la cantidad de elementos que hay, el ciclo debe avanzar hasta que se cumpla la condición antes mencionada.

**Nota:**

1. No se permite el uso de las librerías de Java para manejo de LISTAS DOBLES, debe seguir los lineamientos establecidos.