



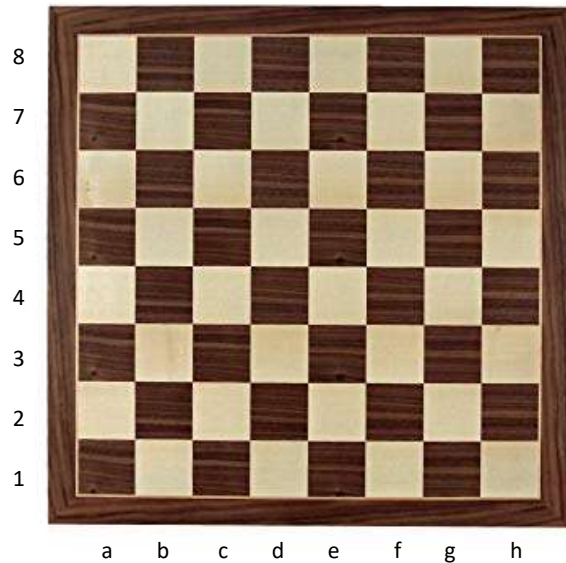
# ANÁLISIS, DISEÑO Y CODIFICACIÓN DE ALGORITMOS Y DIAGRAMAS DE FLUJO

## Competencias a desarrollar:

Analizo, diseño y codifico, algoritmos y diagramas de flujo en PSeINT.  
Creo estructuras de Programación, mediante la implementación lógica y funcional en el manejo de: Variables, Condicionales SI, Bucles con Para, Mientras, **Vectores**, entre otros.

Docente: Joan Carlos Ayala Benavides  
Facultad de Ingenierías  
Programa Ingeniería de Sistemas  
Introducción a la programación

1. Considere el siguiente tablero de un Ajedrez 8x8 (sin fichas): (1.5 pts)



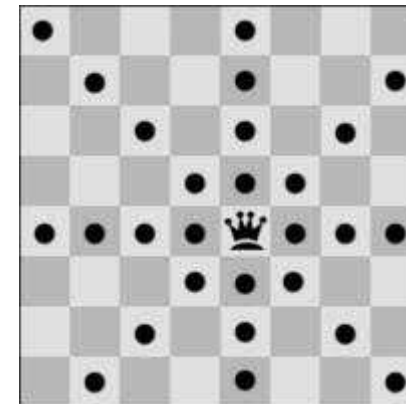
**Fuente:** <https://www.amazon.es/Aquamarine-Games-Tablero-Computid-FD101917/dp/B00EE15XIU>

Escriba un algoritmo que reciba por teclado la posición de una de las Reinas, y de acuerdo a esa coordenada muestre por pantalla al usuario, el número de posiciones que tiene la Reina para moverse en las diferentes direcciones. Tenga en cuenta que en el tablero solo estará presente la Reina.

El sistema debe dar la siguiente información una vez el usuario digite la coordenada de la Reina:

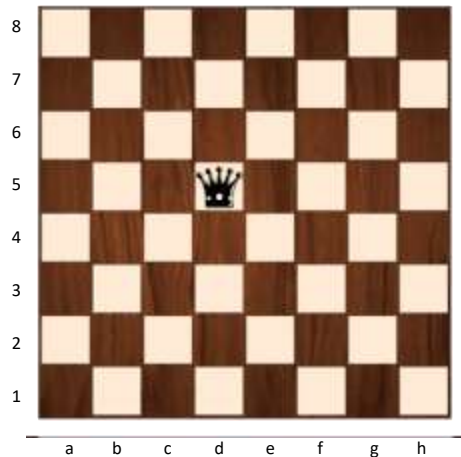
- Posiciones a la derecha
- Posiciones a la izquierda
- Posiciones hacia arriba
- Posiciones hacia abajo
- Posiciones diagonal derecha hacia arriba
- Posiciones diagonal izquierda hacia arriba
- Posiciones diagonal derecha hacia abajo
- Posiciones diagonal izquierda hacia abajo

Tenga en cuenta los movimientos de la Reina en un tablero de Ajedrez:



**Fuente:** <https://www.ajedrezdetorneo.com/movimiento-piezas-ajedrez/>

Considere el siguiente ejemplo:



Fuente: [https://es.wikipedia.org/wiki/Dama\\_\(ajedrez\)](https://es.wikipedia.org/wiki/Dama_(ajedrez))

De acuerdo a la imagen, la Reina se encuentra en la coordenada (dx, 5y). Desde esa posición, el sistema me debe mostrar el siguiente reporte:

Posición	Nro. de movimientos
Posiciones a la derecha	4
Posiciones a la izquierda	3
Posiciones hacia arriba	3
Posiciones hacia abajo	4
Posiciones diagonal derecha hacia arriba	3
Posiciones diagonal izquierda hacia arriba	3
Posiciones diagonal derecha hacia abajo	4
Posiciones diagonal izquierda hacia abajo	3

2. Cree un algoritmo que imprima por pantalla los 20 primeros términos de la sucesión **Fibonacci**. (Consulte y lea sobre esta serie) (0.1 pts)

3. Escriba un algoritmo que solicite un número por pantalla, y visualice el **Factorial** de dicho valor ingresado. (Consulte el factorial de un número) (0.1 pts)

4. Escriba un algoritmo que reciba por pantalla N números y deje de solicitar números hasta que la SUMA de los ya ingresados sea mayor a 100. (0.1 pts)

El algoritmo debe tener las siguientes validaciones:

- Únicamente aceptará 3 veces el número 3.
- Únicamente aceptará 2 veces el número 2.
- Únicamente aceptará 8 veces **consecutivas** el mismo número (Para el caso de números diferentes a 2 y 3).

5. Considere el siguiente menú de opciones: (0.1 pts)

```

::: MENÚ PRINCIPAL :::
1. Ingresar palabra
2. Mostrar palabra
3. Mostrar palabra (invertida)
4. Salir
.: Digite opción:
> |

```

Fuente: PSeInt

Diseñe y escriba un algoritmo que cumpla las siguientes acciones de acuerdo a cada opción de menú:

- **Opción 1:** Debe solicitar al usuario el número N de letras que contiene la palabra. Una vez ingresado el número de letras, debe solicitar la **palabra** en un **vector** de N cantidad de letras. Tenga en cuenta que cada **LETRA** debe guardarse en una posición del **VECTOR**.

**Considere el siguiente ejemplo:**

Supongamos que usted tiene en mente ingresar la palabra CASA, la cual está contenida de 4 letras: C-A-S-A. Cuando digite la opción 1, el sistema le solicitará inicialmente el número de letras que contiene la palabra tal como se muestra a continuación:

```
Cuántas letras tiene la palabra que  
desea ingresar:  
> 4|
```

**Fuente:** PSeInt

Una vez ingresado el número N de letras, el sistema le solicitará la nueva palabra pero LETRA x LETRA, tal como se muestra a continuación:

```
Digite la letra 1:  
> C  
Digite la letra 2:  
> A  
Digite la letra 3:  
> S  
Digite la letra 4:  
> A|
```

Técnicamente, la palabra quedará almacenada en el vector de la siguiente manera:

Palabra[4]			
C	A	S	A
1	2	3	4

- **Opción 2:** Esta opción debe permitir al usuario visualizar la palabra ingresada en el vector. De acuerdo al ejemplo anterior, imprimirá en pantalla el siguiente resultado:

```
::: La palabra ingresada es:  
C  
A  
S  
A  
Presione una tecla para continuar.
```

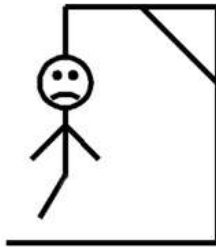
- **Opción 3:** Esta opción debe permitir al usuario visualizar la palabra ingresada en el vector pero de forma invertida. De acuerdo al ejemplo anterior, imprimirá en pantalla el siguiente resultado:

```
Mostrar palabra invertida  
A  
S  
A  
C  
Presione una tecla para continuar.
```

- **Opción 4:** Esta opción debe permitir al usuario salir del sistema.

Adios, Vuelve pronto.

- ▶ 6. Considere el juego del ahorcado: (1.5 pts)



A \_ M \_ I \_ N \_ I \_ S \_ \_ A \_ I \_ O \_ N

Fuente: <https://www.escuelaenlanube.com/juego-el-ahorcado/>

Diseñe y cree un algoritmo que permita jugar al Ahorcado. Tenga en cuenta las siguientes opciones de menú:

```

::: MENÚ PRINCIPAL DEL JUEGO :::
1. Ingresar palabra secreta
2. Jugar
3. Configuraciones
4. Salir
.: Digite opción:
>

```

Fuente: PSeInt

A continuación se describen las acciones de cada opción de menú:

- **Opción 1:** Esta opción debe permitir crear la Palabra secreta. Debe solicitar al usuario el número N de letras que contiene la palabra. Una vez ingresado el número de letras, debe solicitar la **palabra** en un **vector** de N cantidad de letras. Tenga en cuenta que cada **LETRA** debe guardarse en una posición del **VECTOR**.
- **Opción 2:** Esta opción de inicio al juego. Debe validar que esta opción esté disponible SI y SOLO SI, la palabra ya ha sido ingresada en la opción 1. Las reglas del juego consisten en adivinar la palabra antes que terminen las vidas del jugador. Recuerde que las vidas y tamaño de la palabra son variables que se configuran en la opción 3, pero SI no lo hace, estas tendrán valores por defecto. (Ver opción 3.)

Una vez entre a esta opción, el sistema le solicitará al usuario que ingrese letras **hasta que** el jugador haya adivinado el total de las letras de la palabra, o, si antes de hacerlo sus vidas se han terminado. **Recuerde** que una letra puede estar más de una vez en la palabra ingresada.

Al finalizar, deberá mostrar un mensaje informando si el jugador ganó o perdió. Además, visualizará por pantalla el siguiente reporte:

- **Total aciertos**

- **Total desaciertos**

- **Opción 3:** Esta opción debe permitir configurar el nivel de juego.

```

::: NIVEL DE JUEGO :::
1. Principiante
2. Intermedio
3. Avanzado
.: Digite nivel:

```

Esta opción debe modificar las **variables** de configuración, las cuales por defecto, al iniciar el juego, estarán en modo **Principiante**.

Las variables tendrán los siguientes valores de acuerdo a cada nivel:

- Principiante:
  - tamaño\_max\_palabra <- 10
  - vidas <- 6
- Intermedio:
  - tamaño\_max\_palabra <- 15
  - vidas <- 6
- Avanzado:
  - tamaño\_max\_palabra <- 20
  - vidas <- 4

- **Opción 4:** Esta opción debe permitir al usuario salir del sistema.

*Adios, Vuelve pronto.*

- 7. La entidad Financiera XYZ desea modificar el algoritmo que actualmente funciona en el sistema de sus cajeros automáticos. Se desean que la entrega de efectivo se realice en base a los siguientes valores de moneda: (1.5 pts)



Usted es el líder el equipo de Desarrollo de Software, y tiene la misión de diseñar y codificar el nuevo algoritmo. El nuevo sistema debe permitir la entrega de efectivo de acuerdo a los nuevos valores de clasificación del dinero dispuestos por el Banco.

Cree un programa que solicite por pantalla al usuario, el valor o monto a retirar, y luego muestre la entrega del dinero discriminada por monedas de cambio:

```

::: RETIROS EN EFECTIVO :::
.: Digite el valor a retirar ($):
>

```

Una vez el usuario ingrese el valor a retirar, el sistema deberá visualizar en pantalla la entrega del efectivo en las monedas cargadas en el cajero automático.

Considere los siguientes ejemplos:

- Si el monto a retirar por el usuario es **\$6**, el sistema visualizará en pantalla los siguientes valores:  
5  
1
- Si el monto a retirar por el usuario es **\$15**, el sistema visualizará en pantalla los siguientes valores:  
10  
5
- Si el monto a retirar por el usuario es **\$80**, el sistema visualizará en pantalla los siguientes valores:  
50  
20  
10
- Si el monto a retirar por el usuario es **\$36**, el sistema visualizará en pantalla los siguientes valores:  
20  
10  
5  
1
- Si el monto a retirar por el usuario es **\$125**, el sistema visualizará en pantalla los siguientes valores:

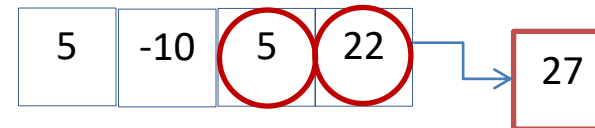
100  
20  
5

- Si el monto a retirar por el usuario es **\$100**, el sistema visualizará en pantalla los siguientes valores:  
100
- Etc.

8. Escriba un programa que solicite N números enteros en un vector. Si el nuevo valor a ingresar es igual a la SUMA del último y penúltimo valor ingresado, el sistema deberá solicitar nuevamente el número HASTA QUE este sea diferente de la SUMA en mención. (0.1 pts)

Considere el siguiente ejemplo:

Suponga que ha ingresado los siguientes valores en el vector N[100]:

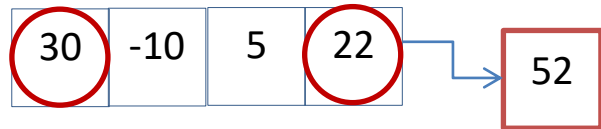


Ahora, usted desea ingresar el número 27 al vector. En este caso el sistema le mostrará por pantalla un mensaje informándole que NO es posible agregar ese número, ya que la suma del último y penúltimo número existentes en el vector hasta el momento, es igual al valor que se desea ingresar, por lo tanto, deberá ingresar otro número.

9. Escriba un programa que solicite N números enteros en un vector. Si el nuevo valor a ingresar es igual a la SUMA del primer y último valor ingresado, el sistema deberá solicitar nuevamente el número HASTA QUE este sea diferente de la SUMA en mención. (0.1 pts extra)

Considere el siguiente ejemplo:

Suponga que ha ingresado los siguientes valores en el vector N[100]:



Ahora, usted desea ingresar el número 52 al vector. En este caso el sistema le mostrará por pantalla un mensaje informándole que NO es posible agregar ese número, ya que la suma del primer y último número existentes en el vector hasta el momento, es igual al valor que se desea ingresar, por lo tanto, deberá ingresar otro número.

#### Recomendaciones:

1. Lea, analice e interprete con detalle cada ejercicio planteado.
2. Evalúe en cada ejercicio cuando conviene o no aplicar VECTORES.
3. Puede usar todas las estructuras de control trabajadas y estudiadas en clase: Condicional SI, Ciclos con Para, Mientras, Repetir, entre otros.
4. La entrega del desarrollo de estos ejercicios deberá hacerla a través de la Plataforma Moodle (UVIRTUAL), la cual estará disponible hasta el día martes 23 de abril 23:55 Hrs.
5. Tenga en cuenta que el día miércoles 24 de abril, se realizará el examen práctico evaluativo de segundo corte.
6. Recuerde estar al día en la entrega de sus actividades.
7. Tenga en cuenta el uso correcto de ortografía en los textos que ve el usuario. (No aplica para variables y/o palabras reservadas del lenguaje)

#### Inquietudes:

E-mail: [jcayala@iucsmag.edu.co](mailto:jcayala@iucsmag.edu.co)