



Universidad  
**CENFOTEC**  
SOMOS LO QUE SABEMOS

**DOCUMENTACIÓN DE CÓDIGO  
PROYECTO PROGRAMACIÓN  
VERTICAL\_STACKING.PY**

Profesora: Stephanie Delgado Brenes

Curso: SOFT-01 Principios de programación 1  
Grupo: SCV3

Estudiantes:

Moisés Reyes Sevilla  
Raulin González Segura  
Johnny Víquez Quirós

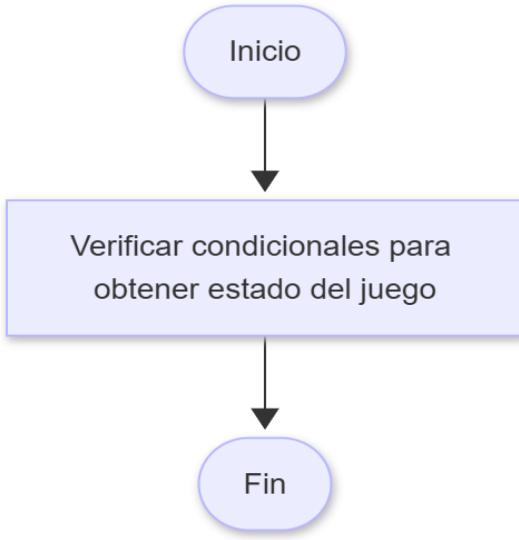
Cartago, Costa Rica  
Octubre, 2025

## ENUNCIADO

1. Un programa que reciba del usuario un dato de entrada con un valor numérico entero positivo y lo evalúe, si el número es menor que 20 el programa debe imprimir un mensaje indicando que el juego se encuentra en curso, si el número es igual a 20 el programa debe imprimir un mensaje indicando que el jugador ha perdido la partida. Si el dato ingresado por el usuario es mayor que 20, o un número entero negativo, o un número con decimales, o cualquier otro valor que no represente un número del todo, el programa debe imprimir un mensaje indicando que el dato de entrada no es válido.

El número recibido por este programa como dato de entrada representa el apilamiento vertical de las piezas alcanzado durante la partida en curso, por lo que un valor de 20 o más es indicador de que el juego ha finalizado porque las piezas se han apilado hasta el límite superior del tablero.

## DIAGRAMA GENERAL



## PSEUDOCÓDIGO

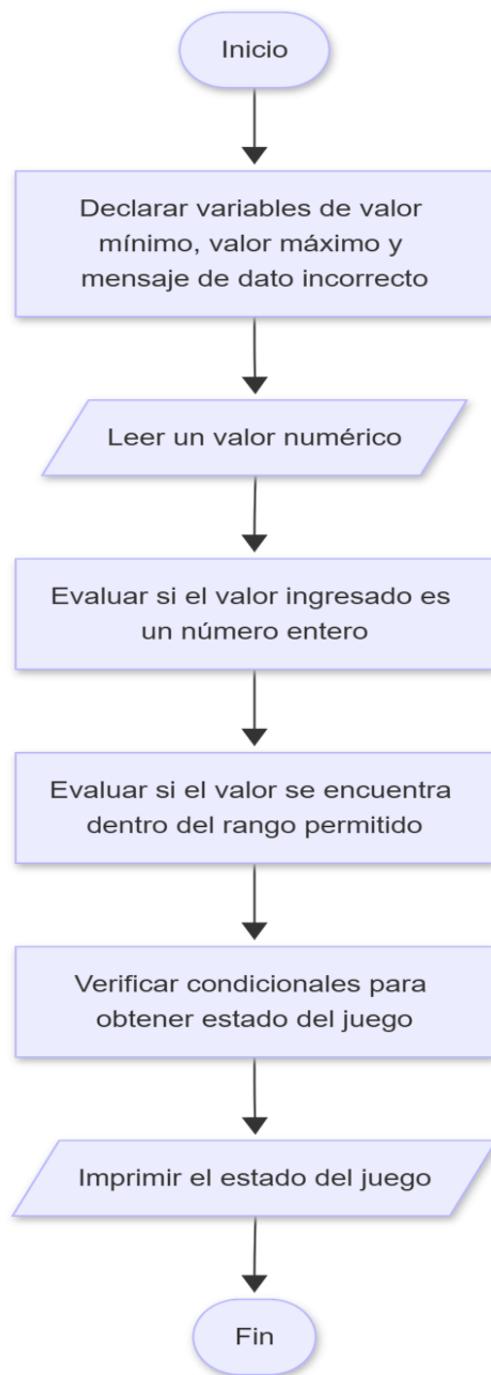
1. Declarar variables de valor mínimo, valor máximo y mensaje de dato incorrecto
2. Leer un valor numérico
3. Evaluar si el valor ingresado es un número entero
4. Evaluar si el valor se encuentra dentro del rango permitido
5. Verificar condicionales para obtener estado del juego
6. Imprimir el estado del juego

### flowchart TB

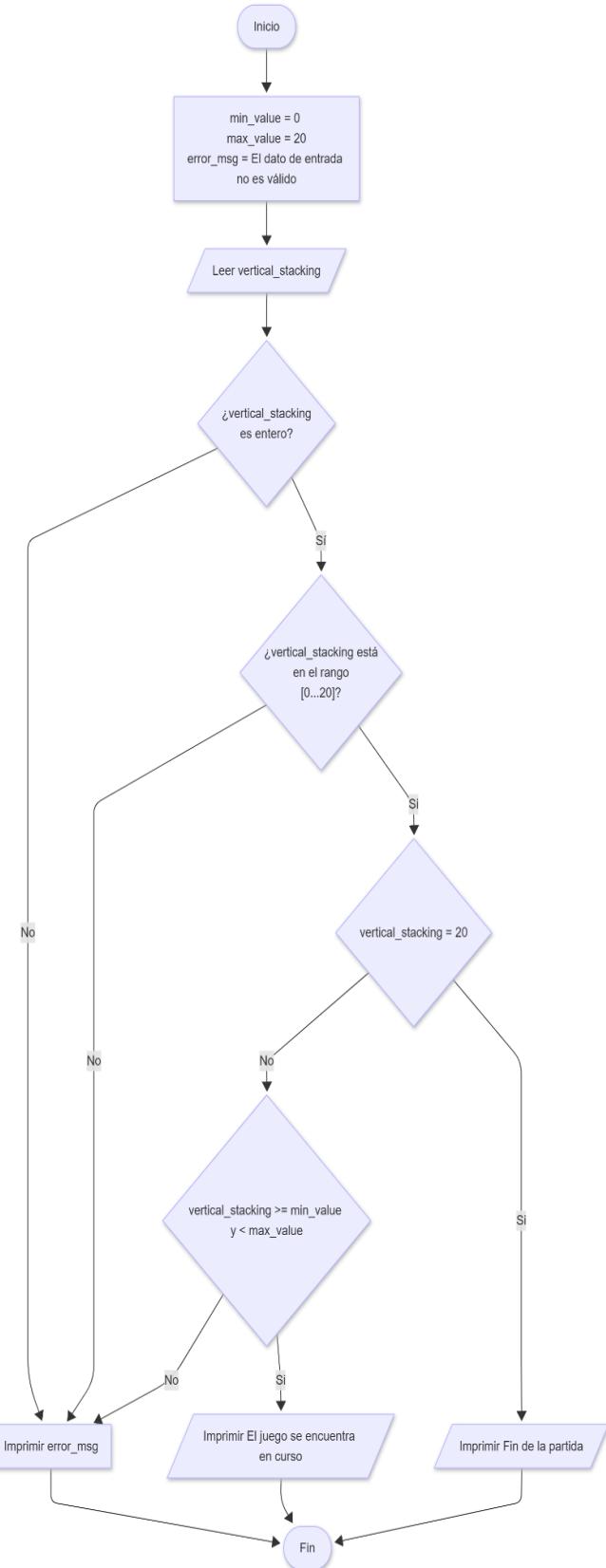
```
A([Inicio]) --> B[Verificar condicionales para obtener estado del juego]
B --> C([Fin])
```

## TABLAS DE VARIABLES

## DIAGRAMA EXPLICATIVO



## DIAGRAMA DE FLUJO



## CASOS DE PRUEBA

| Caso | Entradas | Resultado esperado                | Resultado obtenido                | Encargado    | Fecha      |
|------|----------|-----------------------------------|-----------------------------------|--------------|------------|
| 1    | 10       | "El juego se encuentra en curso"  | "El juego se encuentra en curso"  | Moisés Reyes | 2025-10-10 |
| 2    | 20       | "Fin de la partida"               | "Fin de la partida"               |              |            |
| 3    | -7       | "El dato de entrada no es válido" | "El dato de entrada no es válido" |              |            |
| 4    | 21       | "El dato de entrada no es válido" | "El dato de entrada no es válido" |              |            |
| 5    | "a"      | "El dato de entrada no es válido" | "El dato de entrada no es válido" |              |            |
| 6    | 2.5      | "El dato de entrada no es válido" | "El dato de entrada no es válido" |              |            |

## CÓDIGOS MERMAID

**flowchart TB**

```

A(["Inicio"]) --> B["Declarar variables de valor
mínimo, valor máximo y mensaje de dato
incorrecto"]
B --> C["Leer un valor numérico"]
C --> D["Evaluar si el valor ingresado es un número
entero"]
D --> E["Evaluar si el valor se encuentra dentro del
rango permitido"]
E --> F["Verificar condicionales para obtener estado
del juego"]
F --> G["Imprimir el estado del juego"]
G --> H([Fin])

```

**flowchart TB**

```

A(["Inicio"]) --> B[min_value = 0
max_value = 20
error_msg = El dato de entrada no es válido]
B --> C["Leer vertical_stacking"]
C --> D{vertical_stacking
es entero?}
D -- No --> F[Imprimir error_msg]
F --> G("Fin")
D -- Sí --> H{vertical_stacking está
en el rango
[0...20]?"}
H -- No --> F
I -- No --> K{vertical_stacking >= min_value y <
max_value}
K -- No --> F
K -- Si --> L[Imprimir El juego se encuentra en
curso]
L --> G
H -- Si --> I{vertical_stacking = 20}
I -- Si --> J[Imprimir Fin de la partida]
J --> G

```