

APLICACIONES DE LAS PILAS DENTRO DE ESTRUCTURA DE DATOS



ÍNDICE

➤ 1

*FUNCIONES PRINCIPALES
DE UNA PILA
(INTRODUCCION)*

➤ 2

*APLICACIONES DE LAS
PILAS (INDICE)*

➤ 3

*LLAMADAS A
SUBPROGRAMAS*

➤ 4

*PASO DE PROGRAMAS
RECURSIVOS A ITERATIVOS.*

➤ 5

*EQUILIBRADO DE
SÍMBOLOS.*

➤ 6

*TRATAMIENTOS DE
EXPRESIONES ARITMÉTICAS.*

➤ 7

*LA PILA COMO UN
ADAPTADOR.*

➤ 8

ORDENACIÓN RÁPIDA



INTRODUCCIÓN



Para saber que aplicaciones tiene las pilas dentro de la estructura de datos primero hay que recordar que acciones se puede hacer o llevar a cabo dentro de una pila las cuales son las siguientes acciones:



APLICACIONES PRINCIPALES

- Crear: se crea la pila vacía. (constructor)
- Tamaño: regresa el número de elementos de la pila. (size)
- Apilar: se añade un elemento a la pila (push)

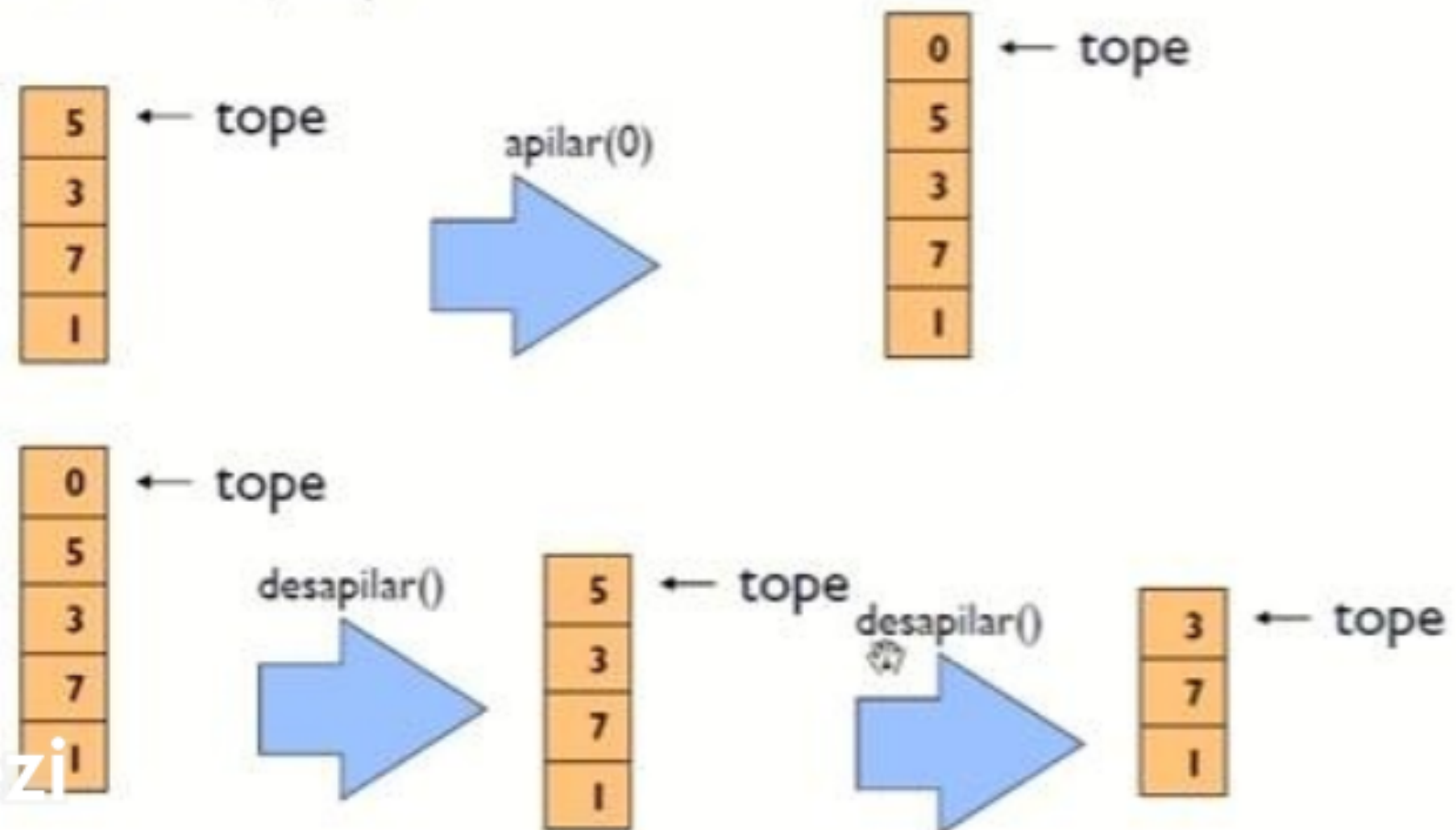
Desapilar: se elimina el elemento frontal de la pila (pop)

- Cima: devuelve el elemento que esta en la cima de la pila. (top o peek)
- Vacía: devuelve cierto si la pila está sin elementos o falso en caso de que contenga uno (empty).

EJEMPLO VISUAL

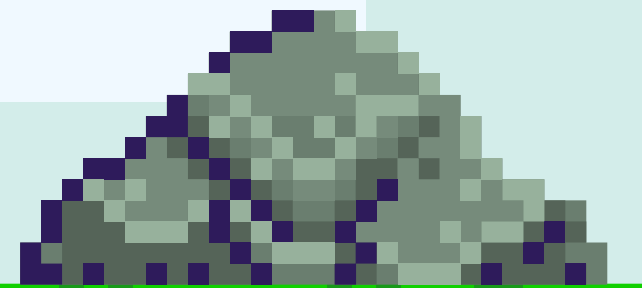
Comportamiento del Modelo Pila

► Pila de Ejemplo:



APLICACIONES DE LAS PILAS

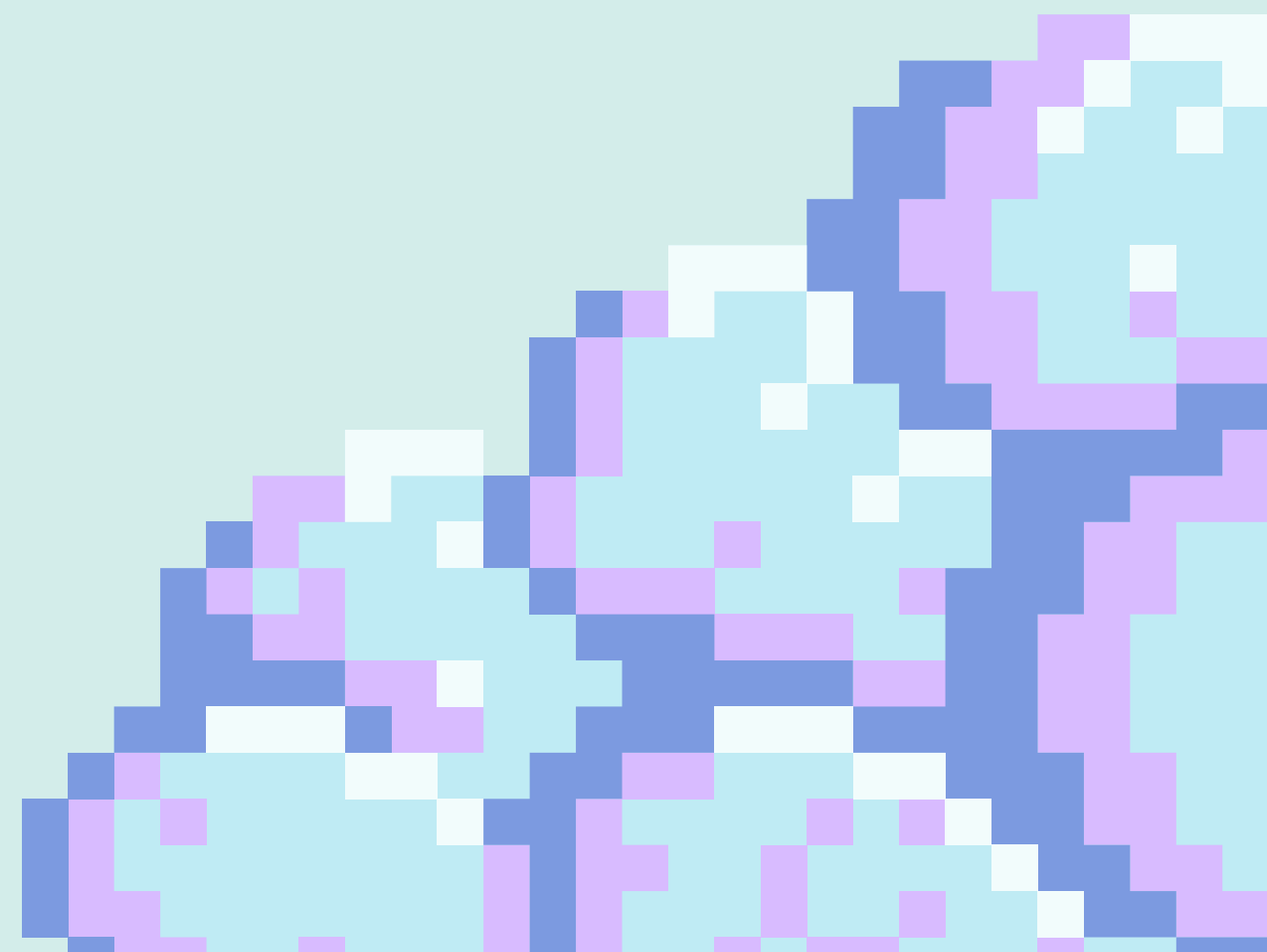
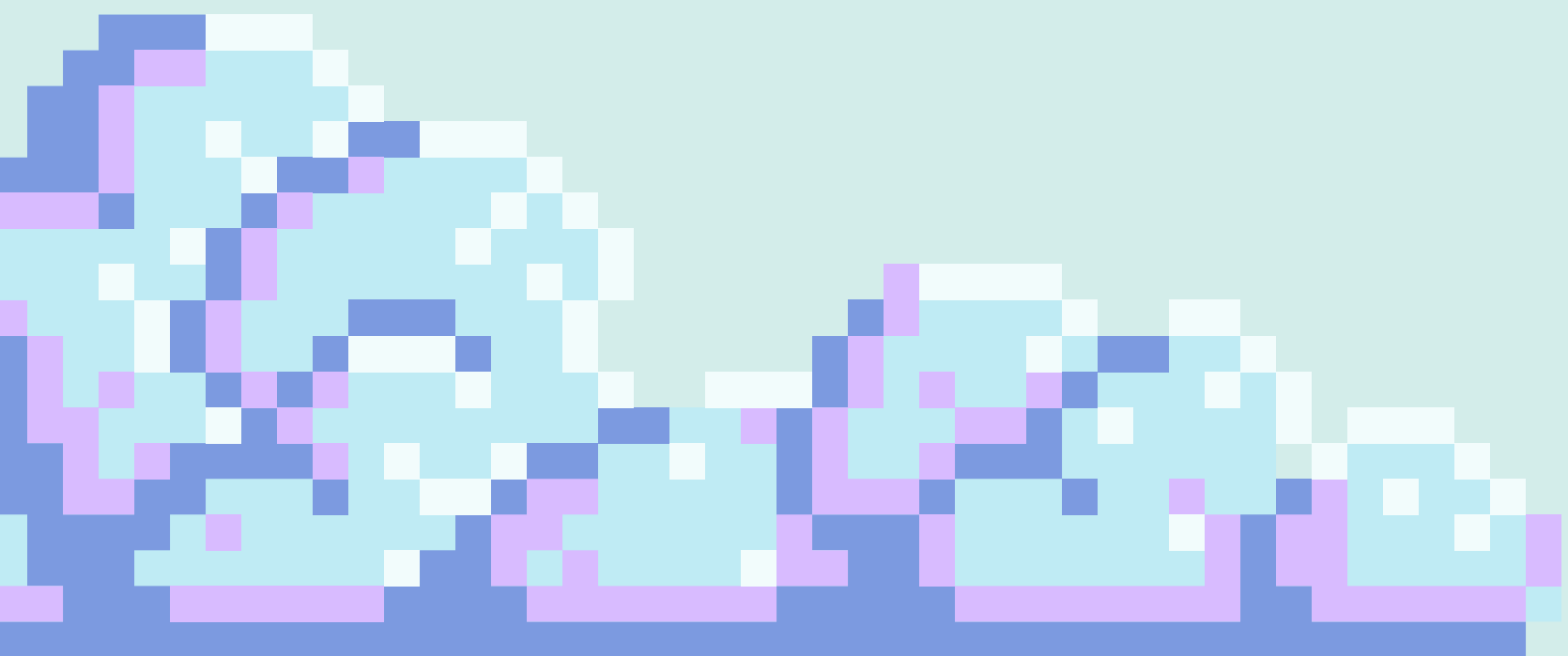
-
- LLAMADAS A SUBPROGRAMAS.
 - PASO DE PROGRAMAS RECURSIVOS A ITERATIVOS.
 - EQUILIBRADO DE SÍMBOLOS.
 - TRATAMIENTOS DE EXPRESIONES ARITMÉTICAS.
 - LA PILA COMO UN ADAPTADOR.
 - ORDENACIÓN RÁPIDA



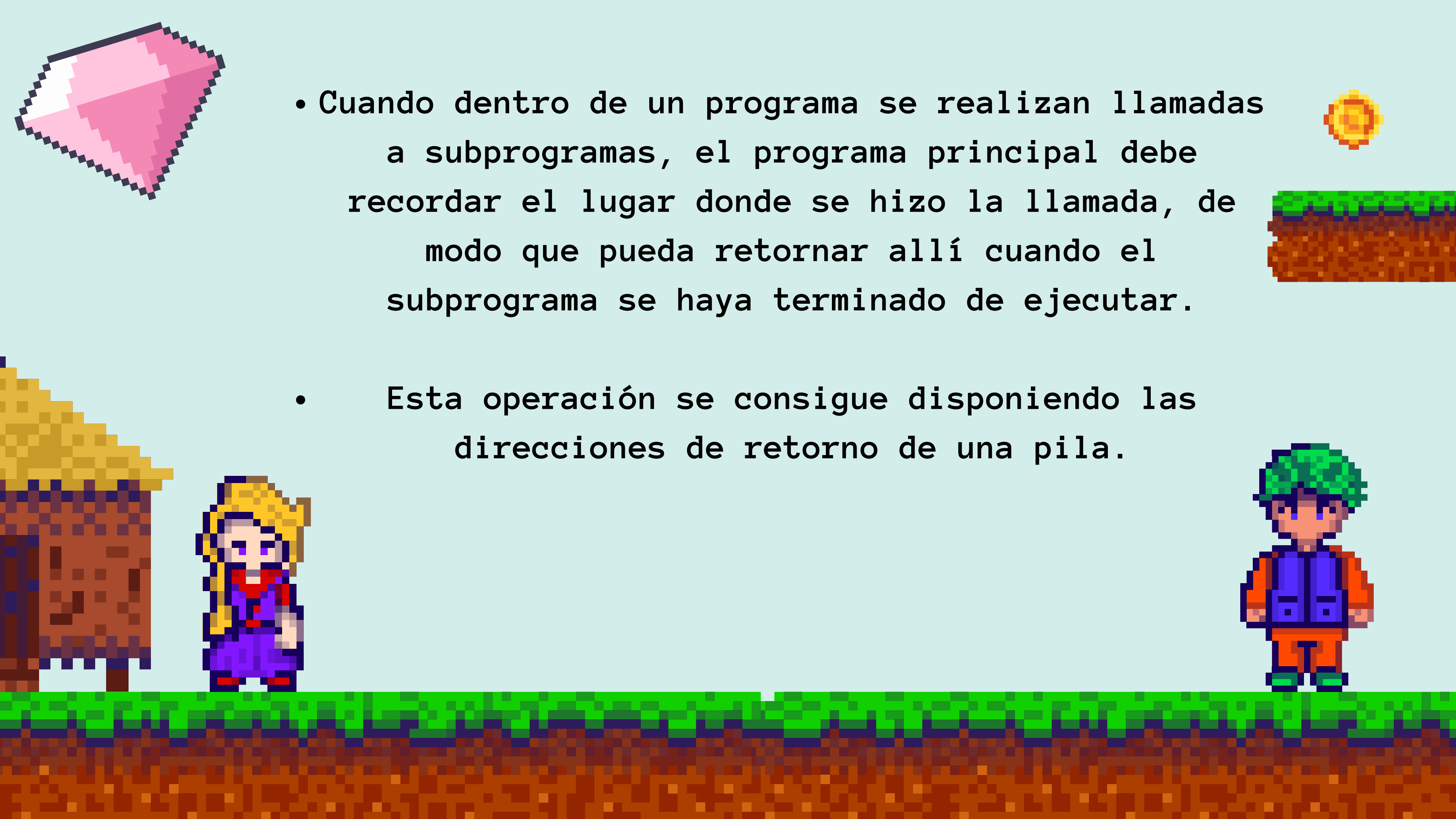


LLAMADAS A SUBPROGRAMAS.

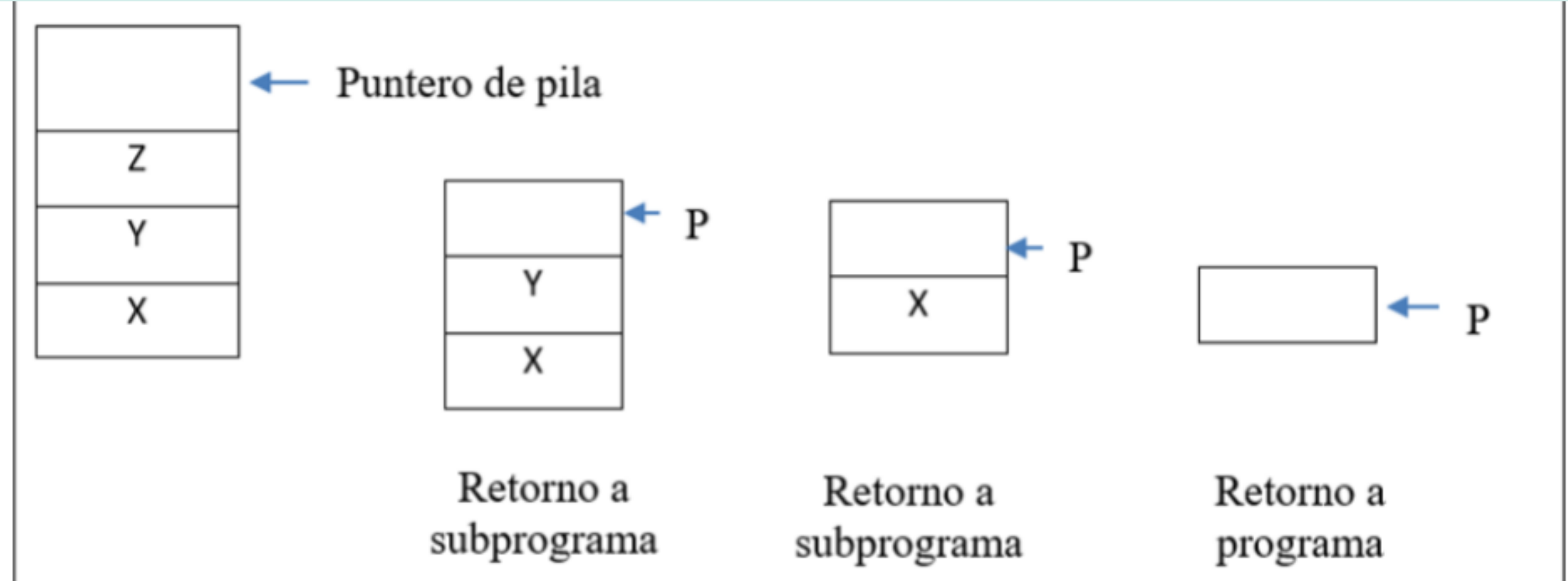
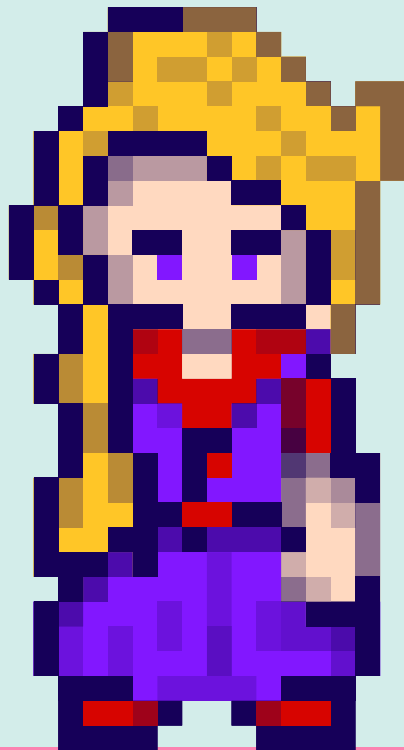
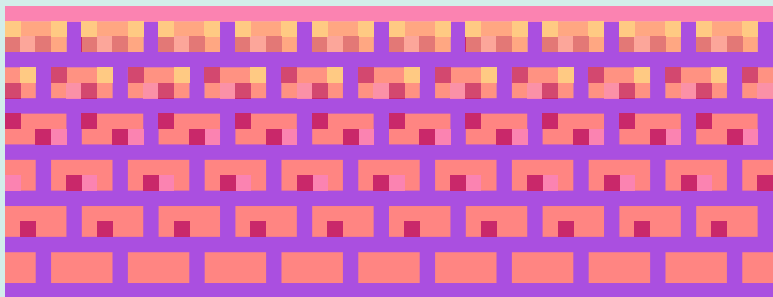
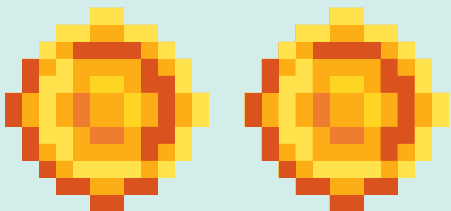
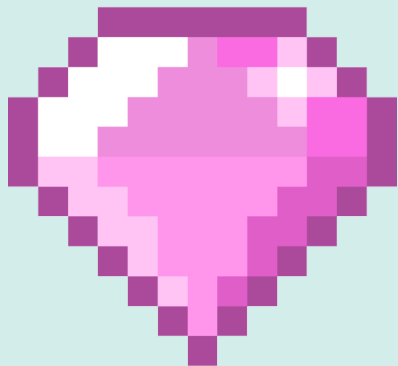
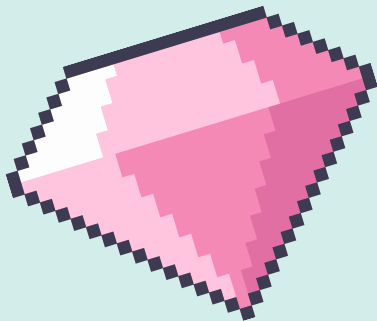
START



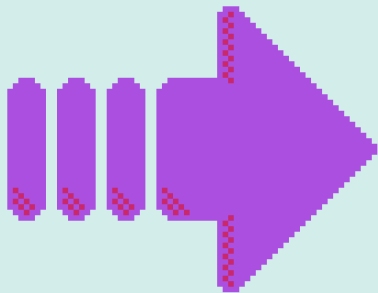
- Cuando dentro de un programa se realizan llamadas a subprogramas, el programa principal debe recordar el lugar donde se hizo la llamada, de modo que pueda retornar allí cuando el subprograma se haya terminado de ejecutar.
- Esta operación se consigue disponiendo las direcciones de retorno de una pila.



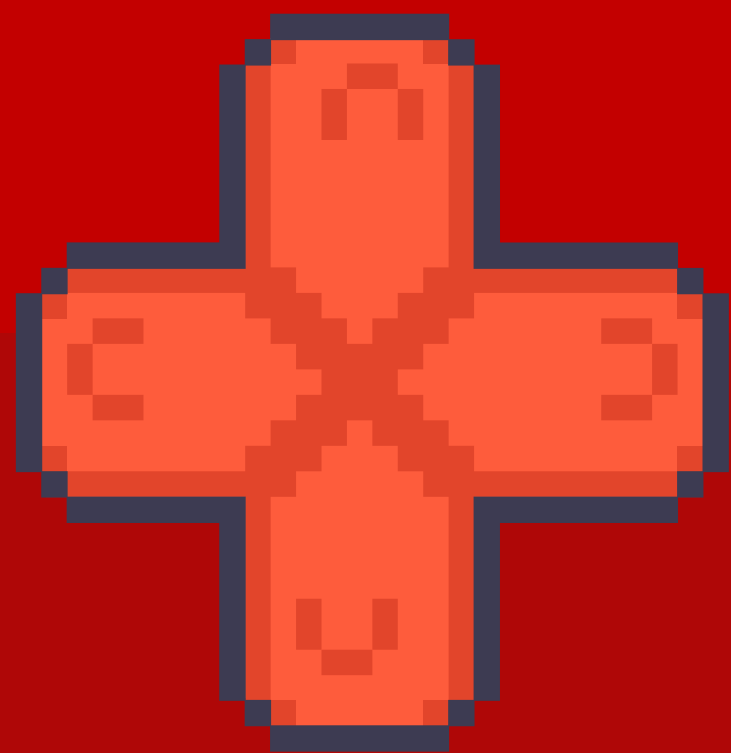
SOLUCIÓN GRAFICA



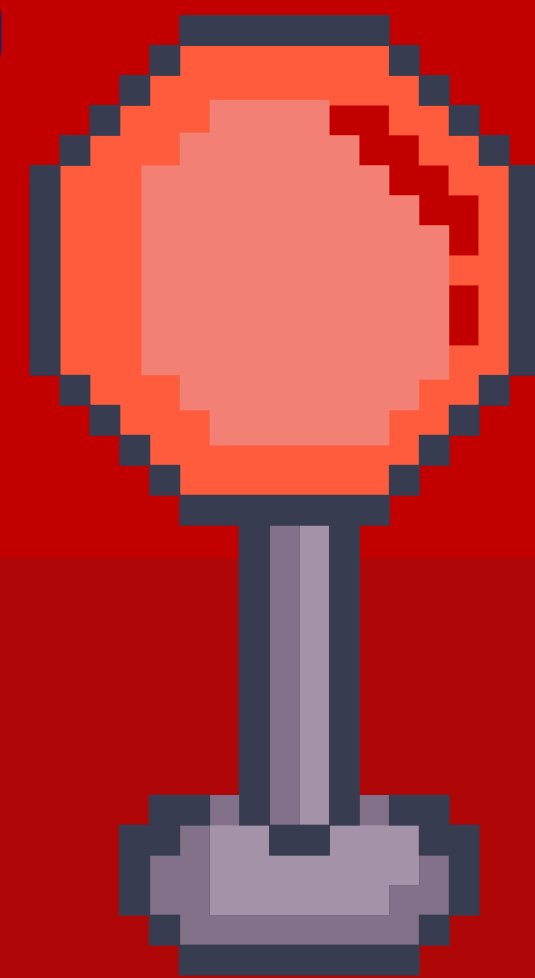
AVANZA AL
SIGUIENTE NIVEL



PASO DE PROGRAMAS ITERATIVOS A RECURSIVOS



START



EXPLICACION

Para simular un programa recursivo es necesario la utilización de pilas, ya que se está llamando continuamente a subprogramas que a la vez vuelven a llamarse a si mismo.

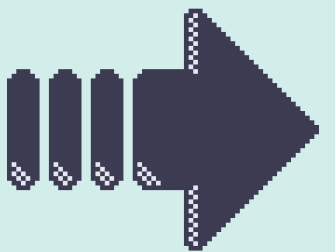


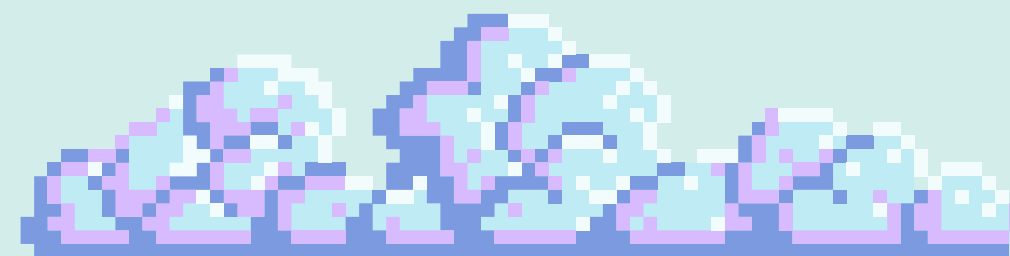
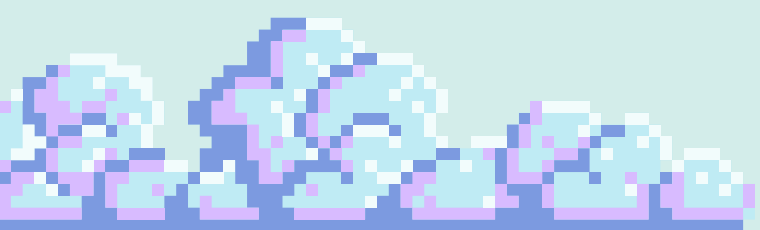
SI DESEAS CONVERTIR UN ALGORITMO RECURSIVO EN UNO ITERATIVO, BÁSICAMENTE DEBES REALIZAR LO SIGUIENTE:

- DETERMINAR EL CASO BASE DE LA RECURSIÓN. CUANDO UNA RECURSIÓN ALCANZA EL CASO BASE, SE DETIENE. CADA ALGORITMO RECURSIVO DEBE ACERCARSE A ÉL EN CADA PASO, LO QUE EVITA QUE SE REPITA INDEFINIDAMENTE.
- IMPLEMENTAR UN BUCLE QUE ITERARÁ HASTA QUE EL CASO BASE ES ALCANZADO. PUEDE SE CUALQUIER ESTRUCTURA, COMO UN FOR, WHILE O DO WHILE, SEGÚN CONVENGA.
- REALIZAR UN PROGRESO HASTA EL CASO BASE. SE SUELEN UTILIZAR VARIABLES ACUMULADORAS QUE SE ACTUALIZAN EN CADA ITERACIÓN.



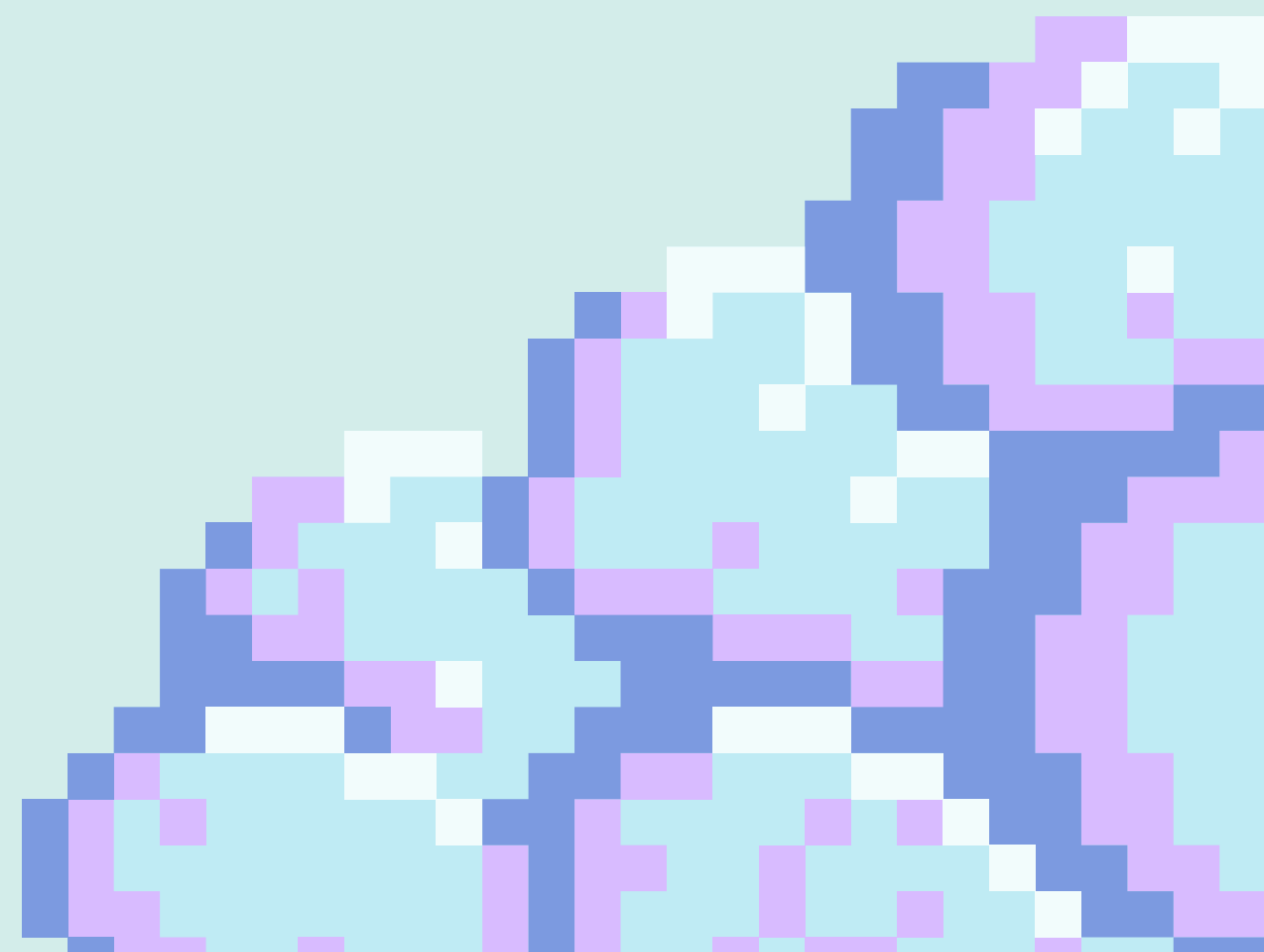
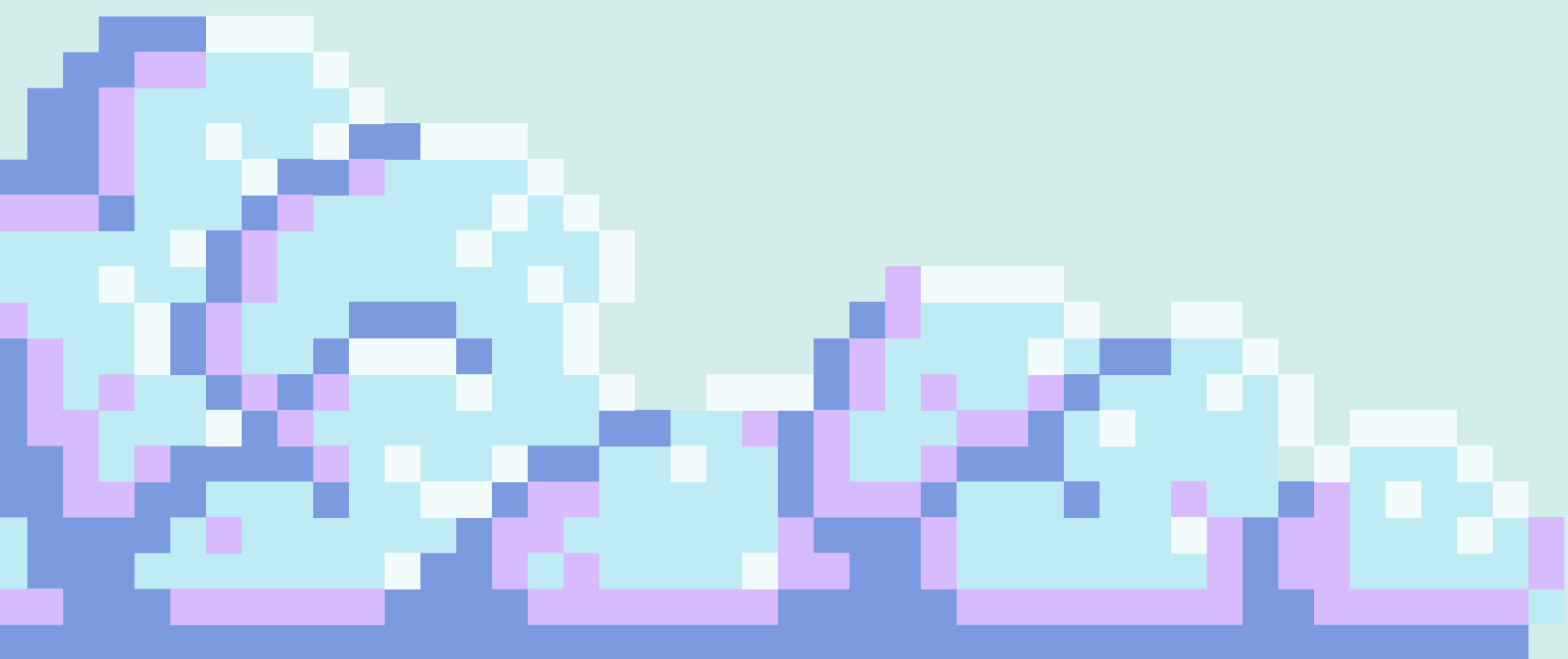
AVANZA AL
SIGUIENTE NIVEL





EQUILIBRADO DE SÍMBOLOS.

START



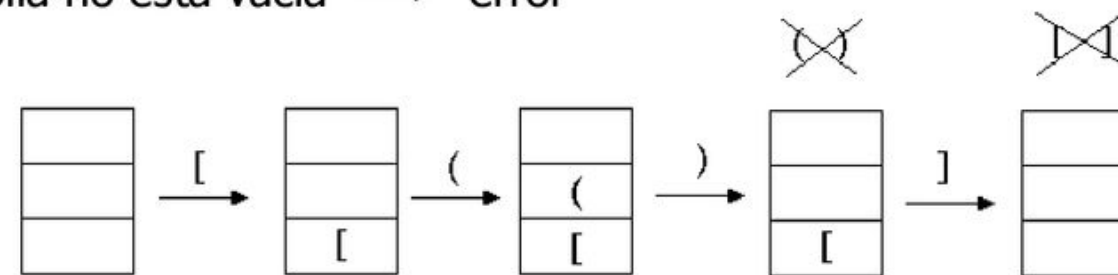
- Las pilas se pueden usar para verificar un programa en busca de símbolos balanceados (como {}, (), []), es decir, el símbolo de cierre debe coincidir con el símbolo de apertura visto más recientemente.



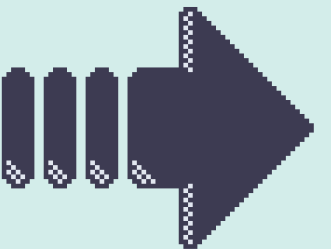


Equilibrado de símbolos

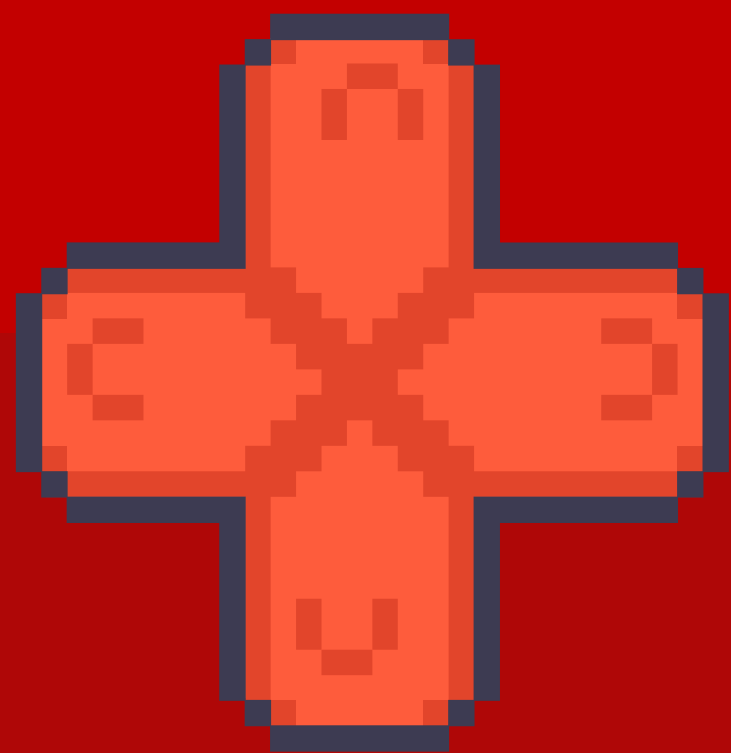
- Se van leyendo los caracteres. Cuando se encuentra un elemento clave (paréntesis, corchete...) se trata según su tipo:
 - Si es de apertura: se mete en la pila.
 - Si es de cierre:
 - Si la pila está vacía → error.
 - Si la pila no está vacía:
 - Si la cima es el correspondiente símbolo de apertura se extrae.
 - Si no lo es → error.
- Si al final la pila no está vacía → error



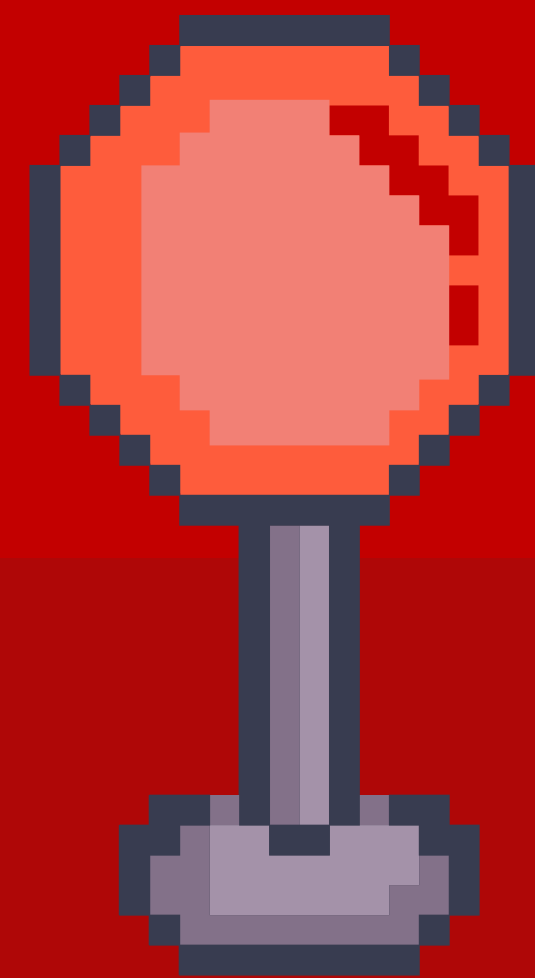
AVANZA AL
SIGUIENTE NIVEL



TRATAMIENTOS DE EXPRESIONES ARITMÉTICAS



START



EXPLICACION

Una expresión aritmética esta formada por operandos y operadores, como por ejemplo $(a+b) - c*d$. En este caso $+$, $-$, $*$ son los operadores y a , b , c , d los operandos.

Los operandos pueden ser valores, variables o incluso otras expresiones,





Pilas



Notación Prefija, Infija y Postfija

- **Expresión aritmética:**

- Formada por operandos y operadores: $A*B / (A+C)$
- Operandos: variables que toman valores enteros o reales
- Operadores:

| | | |
|---------------------------|-----|--------------------------|
| Paréntesis | () | Nivel mayor de prioridad |
| Potencia | ^ | |
| Multiplicación / División | * / | |
| Suma / Resta | + - | Nivel menor de prioridad |

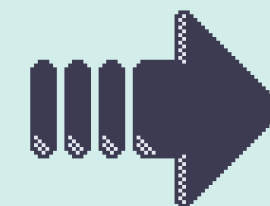
- **En caso de igualdad de prioridad**

- Son evaluados de izquierda a derecha (se evalúa primero el que primero aparece) $\Rightarrow 5*4/2 = (5*4)/2 = 10$
- Cuando aparecen varios operadores de potenciación juntos la expresión se evalúa de derecha a izquierda $\Rightarrow 2^3^2 = 2^{(3^2)} = 2^9 = 512$

- **Notación Infija**

- Es la notación ya vista que sitúa el operador entre sus operandos.
- Ventaja: Es la forma natural de escribir expresiones aritméticas
- Inconveniente: Muchas veces necesita de paréntesis para indicar el orden de evaluación: $A*B/(A+C) \neq A*B/A+C$

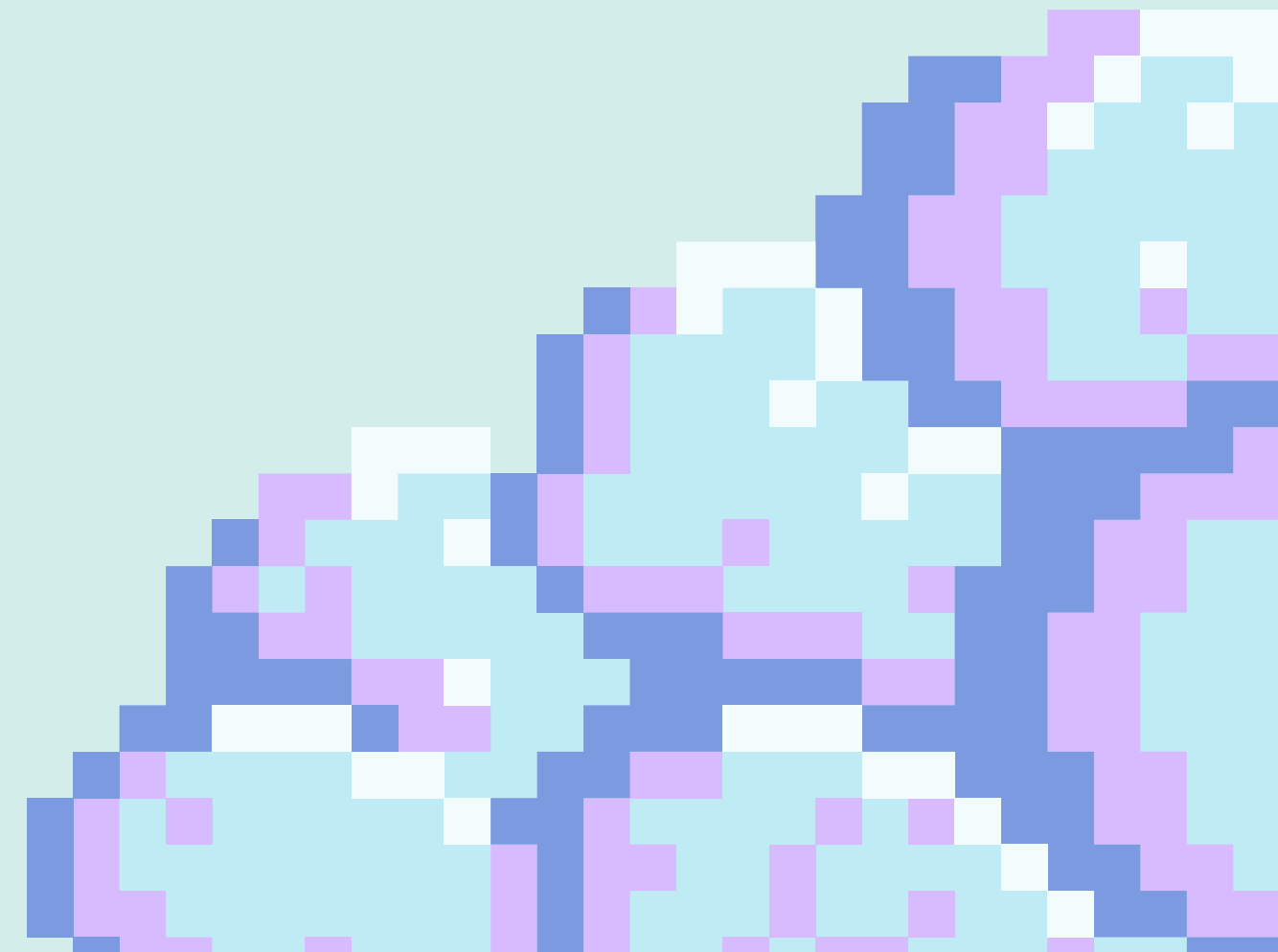
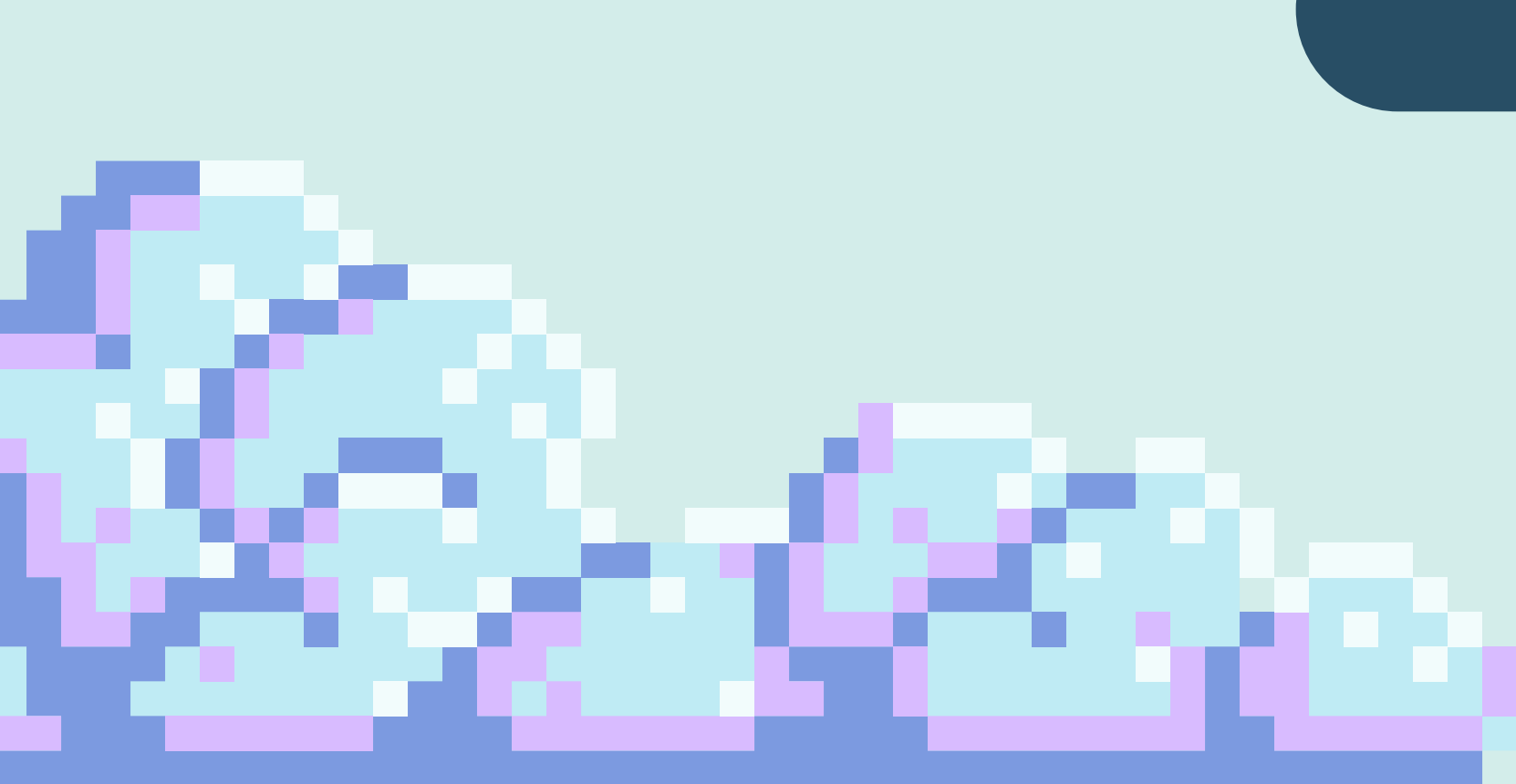
AVANZA AL
SIGUIENTE NIVEL



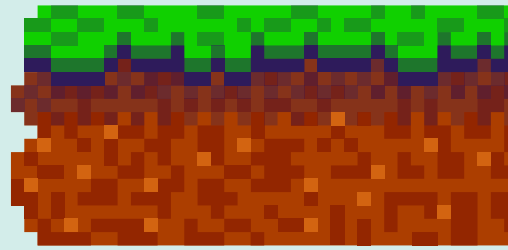
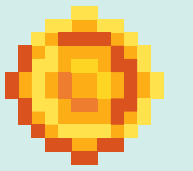
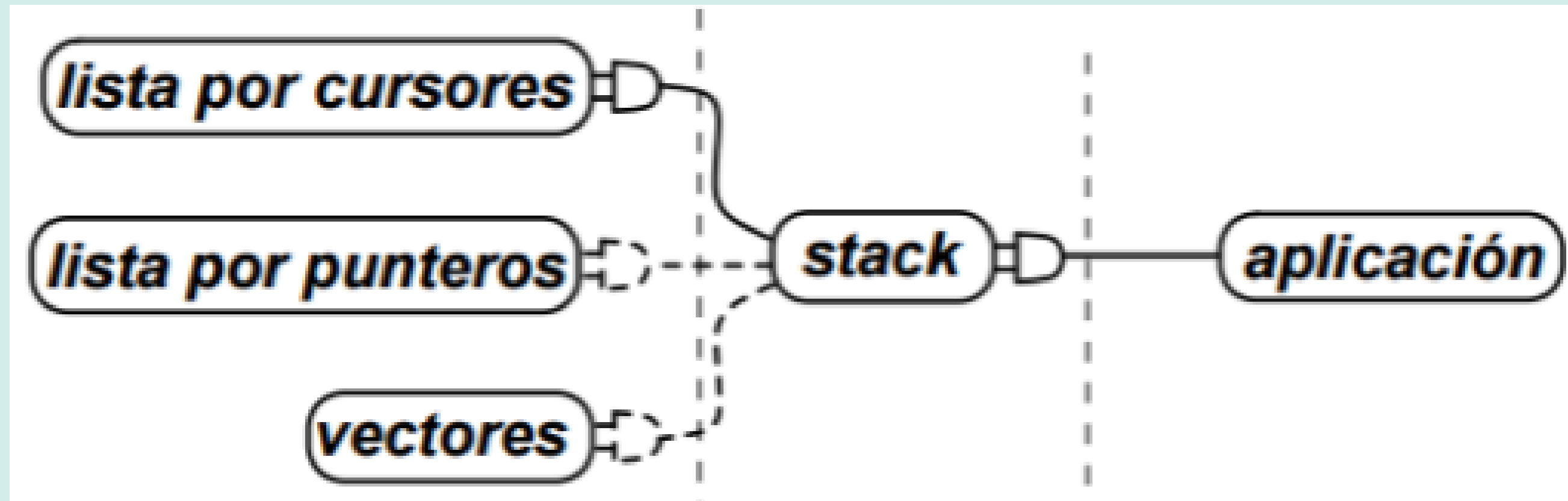


LA PILA COMO UN ADAPTADOR.

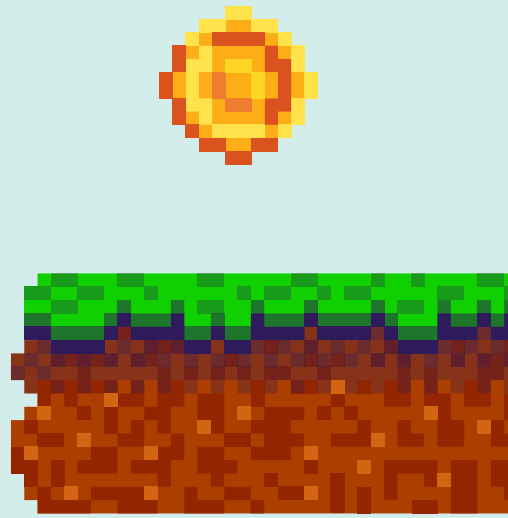
START



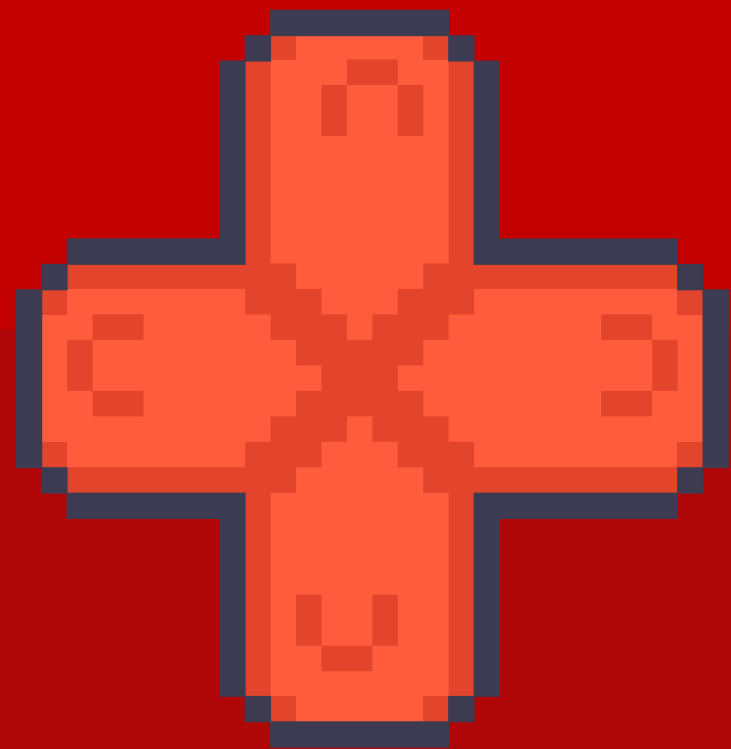
- Una pila deriva de una lista pero con la declaracion private, de esta forma el usuario de clase stack no puede usar metodos de la clase lista



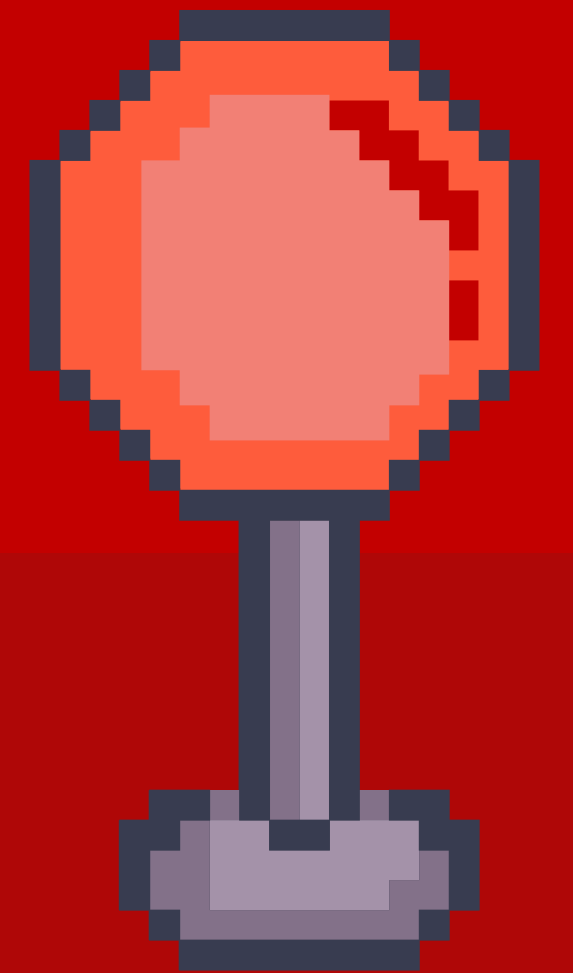
- Como la pila en si misma no contiene iteradores no hay necesidad de clases anidadas ni sobrecarga de operadores, de manera que la única diferencia con la interfaz STL es el uso de templates. Una interfaz compatible con STL



ORDENACIÓN RÁPIDA



START



EXPLICACION

El método de ordenamiento Quick Sort es actualmente el más eficiente y veloz de los métodos de ordenación interna, es también conocido con el nombre del método rápido

Este método es una mejora sustancial del método de intercambio directo y recibe el nombre de Quick Sort por la velocidad con que ordena los elementos del arreglo.



INTEGRANTES DEL EQUIPO



RODRIGO
MARTINEZ RAMOS
EXPOSITOR



ROBERTO RAFAEL
GUZMAN LINO
EXPOSITOR



DAVID ALEJANFRO
CASTRO RAMON
EXPOSITOR



JUAN PABLO
CARRILLO AVILA
EXPOSITOR

