



UNIVERSIDAD  
DE SANTIAGO  
DE CHILE

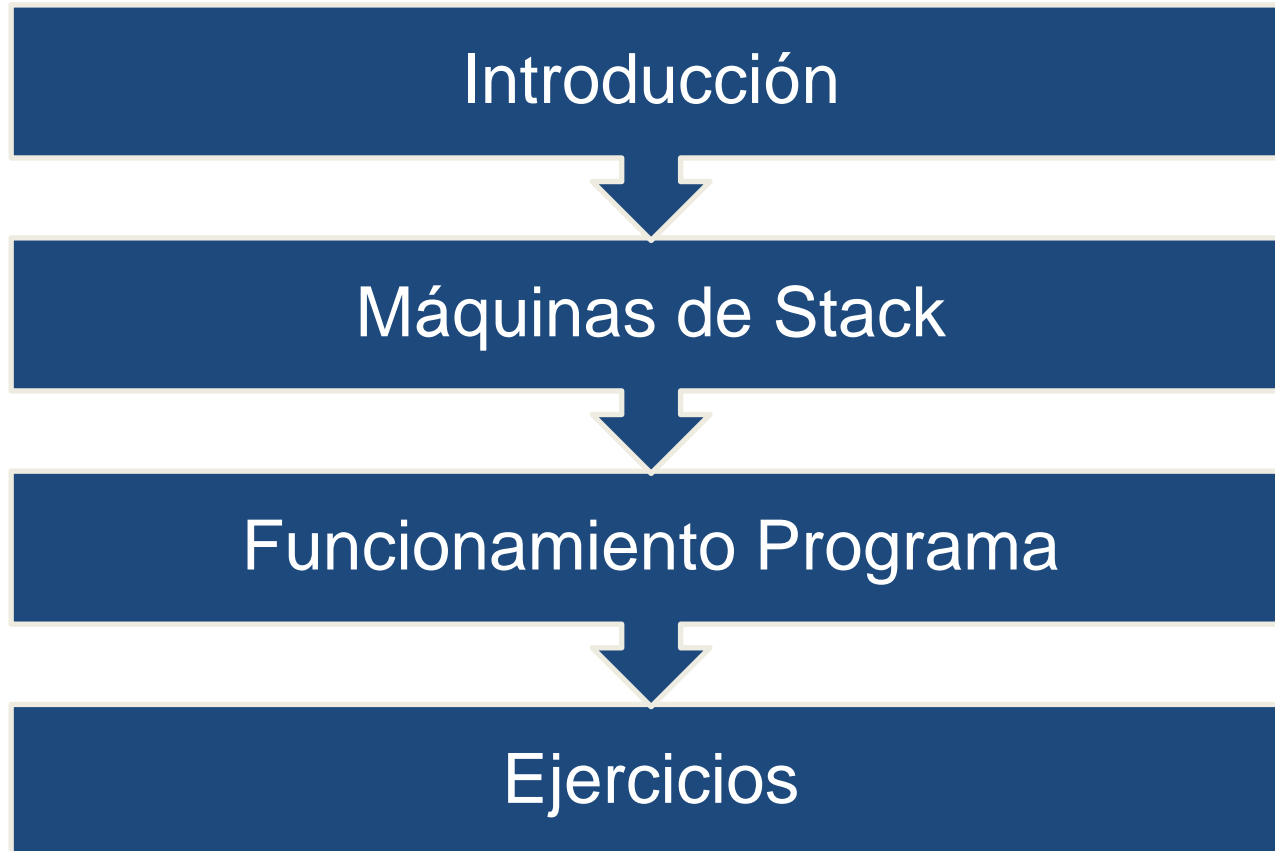
# Máquinas de Stack

Clase 05

Métodos de Programación  
2-2021



# CONTENIDO





# INTRODUCCIÓN [1/2]

Introducción

Máquinas de  
Stack

Funcionamiento  
Programa

Ejercicios

- **Objetivo:**  
Buscar un modelo simple de ejecución.
- **¿Por qué?**  
Para comprender la naturaleza del cómputo y cómo se implementa en la realidad.
- **Contexto**  
Los procesadores reales se encuentran basados en máquinas de Stack y Registros.  
No solo los procesadores sino que también las Máquinas Virtuales (e.g. Java, Python) que permiten portabilidad de código.  
Existen otras máquinas que son más avanzadas, como las máquinas de Turing, que veremos más adelante.



# INTRODUCCIÓN [2/2]

## Introducción

## Máquinas de Stack

## Funcionamiento Programa

## Ejercicios

- **¿Qué necesitamos saber antes?:**
  - Existen distintas formas de escribir las operaciones aritméticas (y no solo las aritméticas, sino que otras también)
    - $5 + 6 * (2 + 8) / 4 \rightarrow$  Esta es la notación infija
  - También existe la notación:
    - Postfija:  $5\ 6\ 2\ 8\ +\ *\ 4\ /\ +$
    - Prefija:  $+ 5\ /\ * 6\ +\ 2\ 8\ 4$
- **Formas de almacenamiento de información**
  - Dentro del procesador la información se almacena mediante conceptos los cuales son:
    - Pilas: Utiliza el concepto LIFO (*Last In First Out*)
    - Colas: Utiliza el concepto FIFO (*First In First Out*)



# MÁQUINA DE STACK [1/8]

Introducción

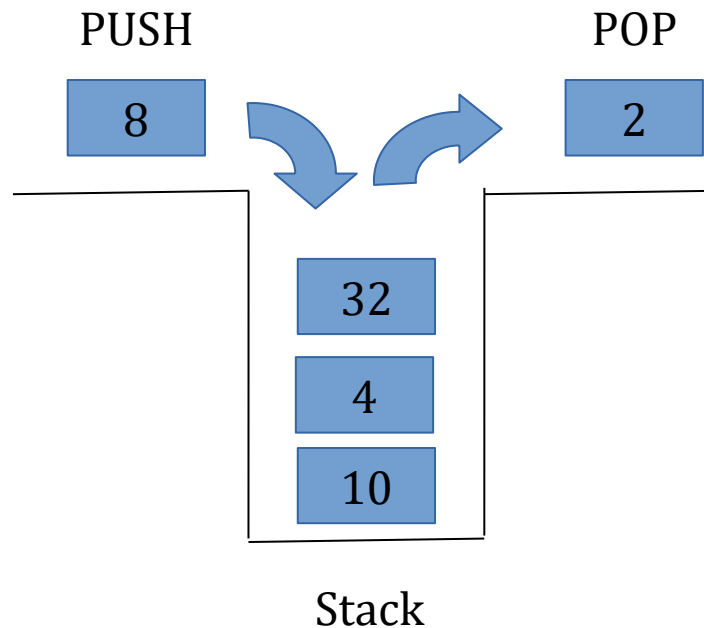
Máquinas de Stack

Funcionamiento Programa

Ejercicios

- **¿Qué es un stack?**

- Repositorio de datos auxiliar.
- Operación en LIFO: Last In First Out.
- Operaciones Básicas: PUSH(X), POP.
- Otras operaciones: Aritméticas y lógicas.





## Introducción

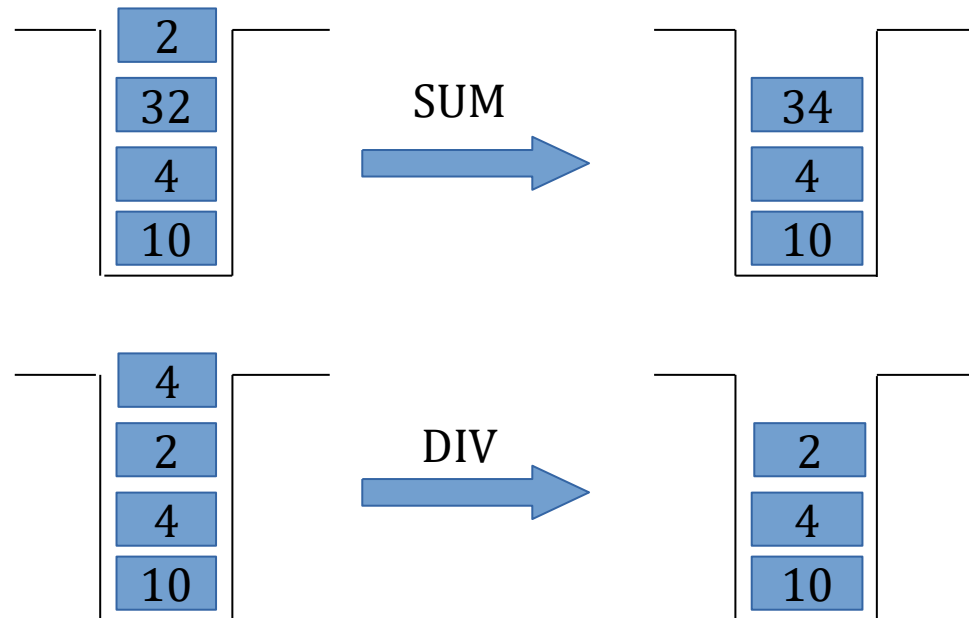
## Máquinas de Stack

## Funcionamiento Programa

## Ejercicios

# MÁQUINA DE STACK [2/8]

- **Operaciones en el Stack:**
  - SUM: Realiza 2 POP, **suma** los dos valores y efectúa un PUSH del resultado.
  - DIV: Realiza 2 POP, **divide** los valores (el primero dividido el segundo) y efectúa un PUSH del res





Introducción

Máquinas de  
Stack

Funcionamiento  
Programa

Ejercicios

# MÁQUINA DE STACK [3/8]

- Otras operaciones en el Stack que utilizaremos en el curso:
  - RES: Realiza la **resta** de los elementos que estén al tope del Stack.
  - PROD: Realiza la **multiplicación** de los elementos que estén al tope del Stack.
  - MOD: Realiza el **resto de la división** de los elementos que estén al tope del Stack.
  - RAI: Realiza la **raíz cuadrada** del elemento que esté al tope del Stack.
  - EXP: Calcula el **exponencial ( $e^x$ )** del elemento que esté al tope del Stack.
  - POT: Realiza la **potencia** para los últimos elementos del Stack (Similar a la división)



## Introducción

## Máquinas de Stack

## Funcionamiento Programa

## Ejercicios

# MÁQUINA DE STACK [4/8]

- **Ejemplo de problema:**

$$\cosh(x) = \frac{e^x + e^{-x}}{2}$$

- **¿Cómo lo pensamos?:**

Escribamos el problema normalmente:

1.  $(e^x + e^{-x}) / 2$

Ahora indicamos qué orden debe desarrollar las operaciones:

1.  $e^x \rightarrow$  Guardar el resultado (aux1)
2.  $e^{-x} \rightarrow$  Guardar el resultado (aux2)
3.  $e^x + e^{-x} \rightarrow$  Se suma lo guardado en aux1 y aux2. Se guarda el resultado (aux3)
4.  $(e^x + e^{-x})/2 \rightarrow$  Se toma lo guardado en aux3 y se divide por 2





## Introducción

## Máquinas de Stack

## Funcionamiento Programa

## Ejercicios

# MÁQUINA DE STACK [5/8]

- **Ahora pasamos los pasos a notación postfija los pasos anteriores**
  1.  $e^x \rightarrow$  Guardar el resultado (aux1)
  2.  $e^{-x} \rightarrow$  Guardar el resultado (aux2)
  3.  $e^x + e^{-x} \rightarrow$  Se suma lo guardado en aux1 y aux2. Se guarda el resultado (aux3)
  4.  $(e^x + e^{-x})/2 \rightarrow$  Se toma lo guardado en aux3 y se divide por 2
- **El orden de los pasos en postfija serán:**
  1. Lo último que se debe hacer es la división  
 $\text{aux3} / 2 \rightarrow / \text{aux3 } 2$
  1. El cálculo de aux3 es la suma de aux2 y aux 1, lo reemplazamos  
 $\text{aux3} \rightarrow \text{aux1} + \text{aux2} \rightarrow + \text{aux1 aux2}$   
 $/ \text{aux3 } 2 \rightarrow / + \text{aux1 aux2 } 2$
  1. Reemplazamos ahora aux2 y aux1  
 $\text{aux2} \rightarrow e^{-x}$   
 $\text{aux1} \rightarrow e^x$   
 $/ + e^x e^{-x} 2$



## Introducción

## Máquinas de Stack

## Funcionamiento Programa

## Ejercicios

# MÁQUINA DE STACK [6/8]

- Ejemplo de problema: Calcular la función

$$\cosh(x) = \frac{e^x + e^{-x}}{2}$$

- Instrucciones:  $/ + e^x e^{-x} 2$

PUSH (2)

PUSH (-1) *No existe el -x, si que reemplaza por -1 \* x*

PUSH (x)

MUL

EXP

PUSH (x)

EXP

SUM

DIV



## Introducción

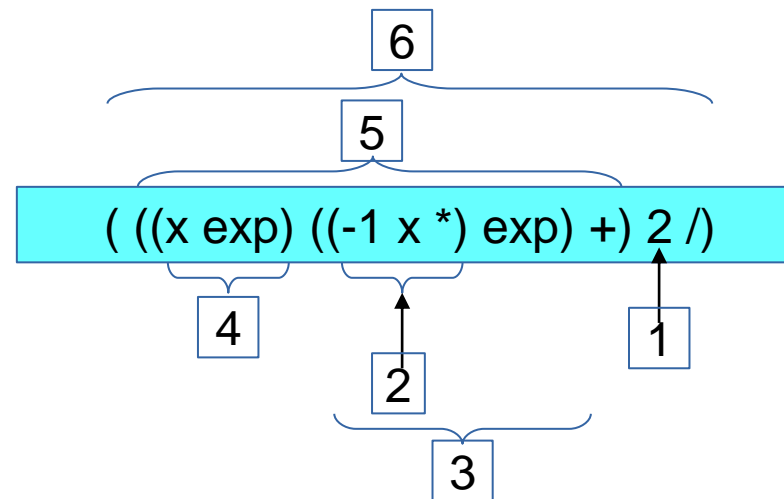
## Máquinas de Stack

## Funcionamiento Programa

## Ejercicios

# MÁQUINA DE STACK [7/8]

- Si fueran funciones sería algo así:  
$$\cosh(x) = / ( + ( \exp(x), \exp( * (-1, x) ) ), 2 )$$
- Lo anterior corresponde al programa previo.
- Regla: se ejecutan los argumentos de derecha a izquierda.
- Si se trabajara con notación Postfija, el resultado es de la siguiente forma:





# MÁQUINA DE STACK [8/8]

Introducción

Máquinas de Stack

Funcionamiento Programa

Ejercicios

## En general:

- En la práctica una máquina de stack en su formato original sirve para computar fórmulas matemáticas.
- Un lenguaje de programación que utiliza dicha forma de cómputo son: postscript o Tex.
- Lo anterior requiere mayores recursos para poder implementar algoritmos más complejos. Por ejemplo:
  - Memoria adicional.
  - Instrucciones de salto condicional.
  - Carga de datos de otros sectores.



# FUNCIONAMIENTO DEL PROGRAMA

Introducción

Máquinas de Stack

Funcionamiento Programa

Ejercicios

## En general:

- Comprende de 3 archivos:
  - Una cabecera (stack.h): Corresponde al archivo que posee las instrucciones de las operaciones de la máquina de stack.
  - El archivo principal (stack.c): Es el archivo principal (y el que se debe compilar). Este lee el archivo de instrucciones y realiza el cálculo de la máquina de stack.
  - Las instrucciones de la máquina de stack (stack.in): Es un archivo de texto plano compuesto por las instrucciones básicas enseñadas. Su estructura es la siguiente:
    - La primera línea indica la cantidad de instrucciones que se poseen.
    - La instrucción PUSH siempre debe ir acompañada de un valor numérico.
    - Las instrucciones aritméticas deben ir solo con el nombre, según lo explicado en esta presentación.
    - Las instrucciones se ejecutan de abajo hacia arriba según lo indicado en el archivo *.in*.



# EJERCICIOS

Introducción

Máquinas de  
Stack

Funcionamiento  
Programa

Ejercicios

**Ecuaciones (Pasar de notación infija a post y prefija):**

1.  $2 + 3 * 4 + 9 / 3 - 2 ** 4 / 8$

1.  $(2 + 3 * (4 + 9 / 3) - 2) ** 4 / 8$

1.  $2 + 3 * 4 + 9 / (3 - 2) ** (4 / 8)$

1.  $2 + (3 * 4) + 9 / (3 - 2 ** 4) / 8$

1.  $2 + (3 * (4 + (9 / 3) - 2) ** 4) / 8$