

# SERIES DE TIEMPO

## SELECCIÓN 0

# LECCIÓN 0

Applied Mathematics and Actuary Training



## PRESENTACIÓN

LECCIÓN 0 – Introducción

# P R E S E N T A C I O N E S

## Preguntas Básicas

Nombre / Apodo

Ocupación (explicado)

Background (académico / laboral)

Conocimientos en programación / estadística

Hobbies/Datos Curiosos

# PRESENTACIÓN EMANUEL MEJÍA

FES Acatlán UNAM – Licenciatura en Actuaría

IPADE Business School – Maestría en Dirección de Empresas para Ejecutivos con Experiencia (MBA/MEDEX)

University of California, Berkeley – Master of Information and Data Science

1

## Certificados:

- SOA – Exámenes P, FM , MFE
- HBS – Credential of Readiness
- Scrum Alliance – Certified Scrum Master
- GMAC – GMAT Focus (Perc 98)
- ETS – TOEFL iBT (Score 115/120)
- BMV Educación – Asesor de Estrategias de Inversión

2

## Experiencia de más de 10 años

### *Desarrollo de emprendimientos de tecnología*

- 2014                      Grupo Converse de México
- 2015 - Actual        Firedrop
- 2020 - Actual        Laboratorios Verktann



# CONTACTO

## Correo electrónico:

- [emanuelmejia@berkeley.edu](mailto:emanuelmejia@berkeley.edu)
- [emanuel@firedrop.mx](mailto:emanuel@firedrop.mx)

## Redes:

- [www.linkedin.com/in/emanuel-mejia-firedrop](https://www.linkedin.com/in/emanuel-mejia-firedrop)
- <https://github.com/emanuelmejia>





# LECCIÓN 0

Applied Mathematics and Actuary Training



## DINÁMICA DE CLASE

LECCIÓN 0 - Introducción

# DINÁMICA DE CLASE

...los tiempos han cambiado... y también para dar clases

- El maestro es un **facilitador** del aprendizaje.
- Los **medios electrónicos** permiten obtener información **actualizada** de cualquier índole (validarla).
- Curso en **zoom**: Destinar un **espacio** para que se puedan concentrar en su aprendizaje.
- No hay necesidad de desconectarse de sus **dispositivos móviles**.
- Habrá presentaciones en Power Point y se harán ejercicios prácticos (comprobar instalaciones).
- Las presentaciones, scripts y cuadernos **se enviarán** posterior a la clase,
- Se exhorta a los alumnos a que **hagan todos los ejercicios** en conjunto con el instructor y en su caso tomar notas.
- Los **comentarios y sugerencias** son bienvenidos.



# LECCIÓN 0

Applied Mathematics and Actuary Training



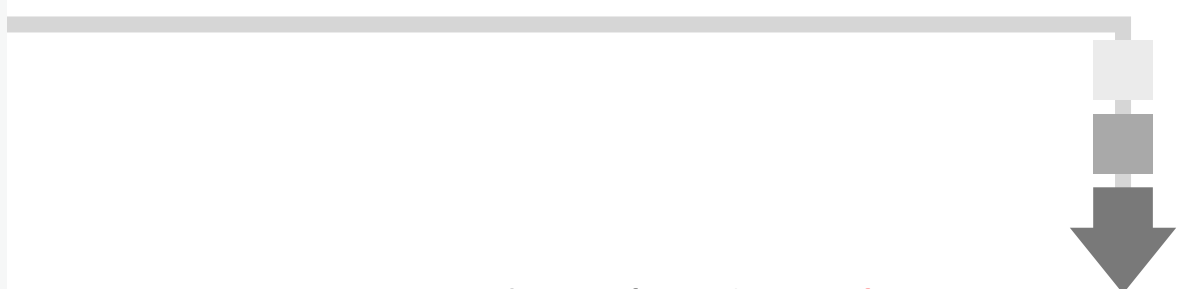
## REPASO DE PROGRAMACIÓN

LECCIÓN 0 – Introducción



# ¿QUÉ ES UN PROGRAMA?

Gran diversidad de tareas  
que deben ser ejecutadas  
con precisión

- 
- Manipulación de **números** y **operaciones matemáticas**.
  - Registrar **nuevas entradas** de datos.
  - **Guardar datos** de forma segura.
  - **Elaborar gráficos** que apoyen a la toma de decisiones.
  - **Comunicarnos con servidores** u otros sistemas computacionales.

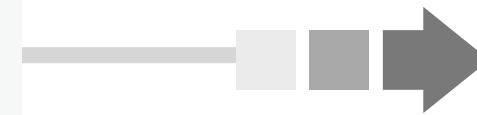
Para estas tareas hacemos uso de DIVERSOS computadores

## PROGRAMA

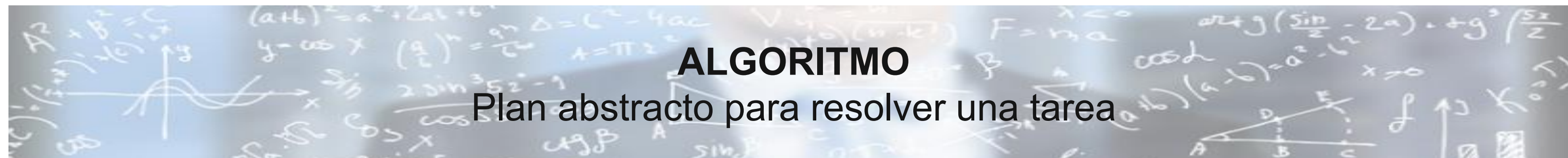
Conjunto de INSTRUCCIONES para ejecutar un trabajo en un LENGUAJE que sea comprensible para una computadora

# PROGRAMA VS ALGORITMO

Pensar en la manera de resolver un problema



- Pasos a Seguir
- El método a ejecutar
- Puede explicarse en español.



## Algoritmo

Conjunto de pasos para resolver un problema

## Programa

Instrucciones codificadas en un lenguaje de programación

# SENTENCIAS DE PROGRAMACIÓN

Un lenguaje de programación  
es MÁS PRECISO  
que un lenguaje humano



- No hay lugar para ambigüedades
- La computadora no adivina a lo que nos referimos

Cada lenguaje tiene una  
SINTAXIS ESTRUCTA



- Existen reglas para escribir instrucciones
- Sentencias de Programación o Declaraciones de Programa
- Algunos son más amigables, pero no hay excepción

Existe una  
Interpretación ÚNICA



- La frase `for(i in 1:length(palabras))` tiene una única interpretación en lenguaje R
- Cualquier computadora que hable “el lenguaje” hará lo mismo cuando vea esta línea.

No puede intercambiarse  
EL ORDEN de las palabras



- Un humano puede deducir el significado
- Una computadora no podrá ejecutarlo

# DIFERENTES LENGUAJES

El mismo algoritmo puede  
codificarse en varios  
lenguajes de programación

- El algoritmo es abstracto – Es **diferente de la forma de comunicarlo** a la computadora
- Cada lenguaje de programación tiene **diferente sintaxis**.
- Elegir un lenguaje específico tiene **ventajas y desventajas**.

## Algoritmo

Conjunto de pasos para resolver un problema

Programa en  
C++

Programa en  
Python

Programa en  
R



## EJEMPLO DE ALGORITMO

Encontrar la palabra más larga en una lista de palabras

Inventamos un algoritmo SIMPLE, con los siguientes **PASOS**:

1. Iniciamos con una “palabra” ELEGIDA de longitud 0.
2. Pasamos a la SIGUIENTE PALABRA de la lista.
3. Si esta nueva palabra tiene una LONGITUD MAYOR a la ELEGIDA, la convertiremos en nuestra NUEVA ELEGIDA. De lo contrario, permanecemos con la misma.
4. REPETIMOS los pasos 2 y 3 con TODAS LAS PALABRAS de la lista.
5. Al finalizar, nuestra palabra ELEGIDA, será la de mayor longitud.

## PROGRAMA CODIFICADO

Encontrar la palabra más larga en una lista de palabras

Traduciendo el algoritmo anterior para implementarlo en lenguaje R mediante las siguientes líneas de código:

```
1 oracion <- readline("Ingresa palabras separadas por espacios: ")
2 palabras <- unlist(strsplit(oracion, " "))
3 elegida <- ""
4 for (i in 1:length(palabras)){
5   if(nchar(palabras[i]) > nchar(elegida)){
6     elegida <- palabras[i]
7   }
8 }
9 print(elegida)
```

- Las **líneas 1 y 2** obtienen los datos y los guardan en una variable llamada palabras
- La **línea 3** corresponde al **paso 1** del algoritmo
- Las **líneas 4-8** corresponden a los **pasos 2-4**.
- La **línea 9** imprime el resultado correspondiente al **paso 5**.

# SERIES DE TIEMPO