

DESAFIO **LATAM**

REPASEMOS DE NUEVO CONCEPTOS CLAVES

¿Cuál es la diferencia entre una clase y un objeto?

¿Qué hace el operador new?

¿Qué significa instanciar?

¿Para qué sirven los getters?

¿Qué hace `def x=(x)`?

Qué hace attr_reader :x?

¿Si existe la clase Monster, que devuelve `Monster.class`?

¿Si `a = Monster.new`, que devuelve `a.class`?

RUBY

OBJETOS Y ARCHIVOS



¿POR QUÉ ESTAMOS TRABAJANDO CON ARREGLOS Y ARCHIVOS?

Porque nos darán las abstracciones necesarias para obtener resultados de una base de datos y mostrarlas en pantalla

REPASEMOS CON
PREGUNTAS !!

¿CUÁL ES EL ERROR?

```
class T
  def foo
    m = 5
  end

  def bar
    m += 1
  end
end
```

```
p1 = T.new
p1.foo
p1.bar
```

¿CUÁL ES EL ERROR?

```
class T
  def foo
    @m = 5
  end

  def bar
    @m += 1
  end
end
```

```
p1 = T.new
p1.foo
p1.bar
p1.m
```

¿CUÁL ES EL ERROR?

```
class T
  attr_reader :m
  def foo
    @m = 5
  end

  def bar
    @m += 1
  end
end
```

```
p1 = T.new
p1.foo
p1.bar
p1.m = 10
```


¿CUÁL ES EL ERROR CON EL SIGUIENTE CÓDIGO?

```
class Store
  attr_accessor :name
  def initialize(name)
    name = name
  end
end
```

```
store = Store.new("S1")
store.name
```

VEAMOS UN EJEMPLO CON ARCHIVOS

Crear la clase **Product**, cada producto tiene stock asociados a cada color, donde son 3 los posibles colores, azul, rojo, y verde

```
Producto1, 10, 15, 21  
Producto2, 20, 0, 3  
Producto3, 4, 8, 0
```

Se pide:

- Leer el archivo y crear un arreglo de productos con la información.
- Mostrar el stock de cada productos en color rojo.
- Mostrar el total de stock rojo (la suma de todos los rojo de cada producto).
- Mostrar el stock total de un producto.
- Mostrar el stock total de todos los productos.

Crear la clase **Student**, cada alumno tiene un nombre y notas del 0 al 10

```
Juan Pablo, 10, 7, 9  
Magdalena, 9, 9, 9  
Francisca, 8, 6, 7
```

Se pide:

- Leer el archivo y crear un arreglo de alumnos con la información de sus notas
- Crear un método para calcular el promedio
- Mostrar el promedio de notas de todos los alumnos.

EL SIGUIENTE EJERCICIO REQUIERE TRABAJAR CON FECHAS

`require 'date'`

Se tiene un archivo con los pagos a ciertos proveedores

Empresa1	2016-05-21	2017-08-10	2017-08-31	2017-09-18	2017-09-28
Empresa2	2017-06-16	2017-07-05	2017-07-20	2017-12-04	2017-12-19
Empresa3	2017-05-12	2017-05-30	2017-06-29	2017-09-27	2017-10-19
Empresa4	2017-07-06	2017-10-10	2017-12-07	2017-12-15	2017-12-29

Se pide:

- Crear el objeto Empresa y almacenar los pagos como arreglo
- Crear un método que muestre todos los pagos antes de una fecha
- Crear un método que muestre todos los pagos registrados posteriores al del día de hoy

EJERCICIO EN CLASE:

- Crear la clase **Product**, cada producto tiene stock asociados a cada color, donde son 3 los posibles colores, azul, rojo, y verde

Se pide:

- Leer el archivo y crear un arreglo de productos con la información.
- Pedirle al usuario que ingrese un producto nuevo.
- Agregar al archivo los datos guardando con ellos el dato nuevo.

DADO EL ARCHIVO CON PELÍCULAS

- Crear la clase **Movie**, cada película tiene nombre, fecha, estudio, categoría y votos de personas que la vieron.

Se pide:

- Leer el archivo y crear un arreglo con la información de las películas.
- Iterar el arreglo y mostrar los nombres de las películas.

AGRUPANDO POR UNA CARACTERÍSTICA



```
class Movie
  attr_accessor :name, :date, :studio, :category, :votes
  def initialize(name, date, studio, category, votes)
    @name = name
    @date = date
    @studio = studio
    @category = category
    @votes = votes
  end
end
```

```
movies = []
file = File.open("movies.txt")
data = file.readlines()
file.close

data.each_slice(5) do |movie_data|
  movies << Movie.new(*movie_data)
end

movies.group_by(&:studio).each do |studio, group|
  puts studio
  puts group.count
end
```

DESAFÍO

Crear la clase carta con pinta y número
pedir un input al usuario, si el usuario escribe “jugar”
generar 5 cartas al azar
Si el usuario escribe “guardar”, guardar las cartas en un archivo
Si el usuario escribe “mostrar” mostrar las 5 cartas en pantalla
Si el usuario escribe “leer” cargar las cartas desde el archivo
Si el usuario escribe “salir”, terminar el programa.