

# **Introducción al pensamiento computacional**



Primeros pasos

# Acerca de mi

Christian Poveda

## Estudios

- Pregrado en física  
*Algoritmos para modelar formación de galaxias*
- MEng en Ingeniería de Sistemas & Computación  
*Extensiones sobre el sistema de tipos de un leng. de programación*

## Áreas de Investigación

- Lenguajes de programación (Diseño, teoría, implementación)
- Compiladores
- Sistemas de tipos

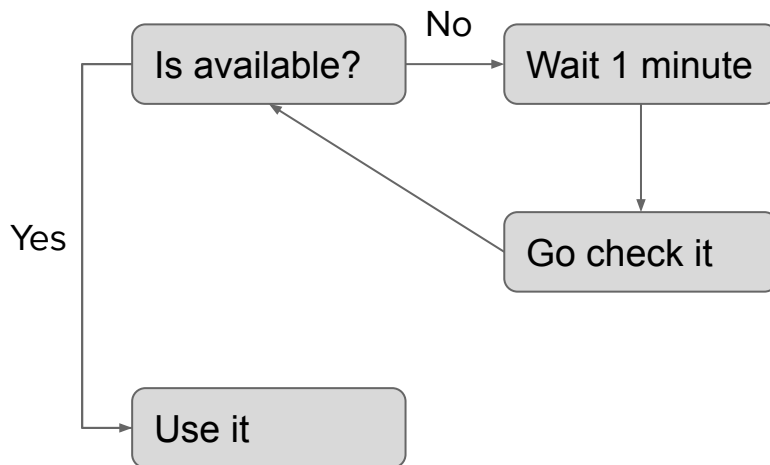
## Otros intereses

- Criptografía
- Computación gráfica

# Sobre el curso: Habilidades

## Algorítmica

Aprender a formalizar  
problemas y sus  
soluciones



## Programación

Lograr que un  
computador los resuelva  
por nosotros

```
def try_use():  
    if is_available():  
        use_it()  
    else:  
        wait_minutes(1)  
        try_use()
```

# **Logística del curso**

**<https://christianpoveda.github.io/ipc>**

# Algoritmo

Es una secuencia **finita** de **instrucciones** realizables por un **computador** para resolver un **problema**.

# ¿Qué es un computador?

<https://youtu.be/ZoqMiFKspAA>

# Cosas que puede hacer un computador

Cosas geniales:

- Análisis de sentimiento
- *Physically based rendering*
- Optometría objetiva

Cosas no tan geniales:

- Vigilancia masiva
- *Deepfakes*

**¿Que  
computador  
vamos a usar  
hoy?**





**WE WANT YOU!**

# Nuestro primer problema

- Organizarse de a parejas.
- Cada estudiante debe pensar en un número de 1 a 10.
- Turnándose, deben tratar de adivinar el número de su compañero.
- Su compañero solo puede dar 3 tipos de respuesta:
  - Ese es mi número.
  - Mi número es más alto.
  - Mi número es más bajo.
- El primero en adivinar el número de su compañero, gana.

# Nuestro primer problema

- ¿Qué hacemos si ahora el número de nuestro compañero está entre 1 y 100 o entre 1 y 1000?
- Junto a su compañero, escriban un algoritmo para adivinar el número del otro tratando de hacer el menor número de preguntas posible.
- Prueben el algoritmo entre ustedes mismos.

**Ahora adivinen  
mi número y  
cuenten cuantos  
intentos  
necesitan**

**¿Qué otro  
problema  
similar hemos  
resuelto en  
nuestro día a  
día?**

# Búsqueda binaria: Solución de libro

1. Llevaremos cuenta de dos números:
  - la cota superior (**sup**)
  - la cota inferior (**inf**)
2. Al comienzo **inf** es 0 y **sup** es 100.
3. Proponer como candidato la mitad entre **sup** e **inf**
4. Si adiviné, termino mi algoritmo.
5. Si el número a adivinar es mayor a mi candidato:
  - actualizar **inf** al número siguiente a mi candidato.
6. Si el número a adivinar es menor a mi candidato:
  - actualizar **sup** al número anterior a mi candidato.
7. Repetir desde 3.

# Búsqueda binaria: ¿Por qué funciona?

Supongamos que el número a adivinar es 15:

inf	sup	candidato
1	100	50
1	49	25
1	24	12
13	24	18
13	17	15

¿Cuántos números estamos descartando en cada paso?