

PyConAr 2021

@pyconar

EL DESAFÍO DE

ENSEÑAR

PYTHON A NIVEL
UNIVERSITARIO

Lic. FELIPE MORALES, INSTITUTO DE INGENIERÍA Y AGRONOMÍA, UNAJ, 2021

AGENDA

- *DATOS DEL AUTOR*
- *CONTEXTO*
- *PROBLEMÁTICA*
- *TESTIMONIOS*
- *DESAFÍOS*
- *CONCLUSIONES*

DATOS DEL AUTOR

Graduado en Informática (Universidad Nacional de La Plata, Argentina)

Docente (UNLP, UNAJ e IPAP).

Socio fundador de Centro de Residentes y Estudiantes de Benito Juárez (La Plata).

Socio fundador de Angras.

Miembro fundador y ex-presidente de Distrito Informático de Gran La Plata (DILP).

Voluntario en Comunidad La Plata de PMIBA.

Administrador de grupo Telegram PyAr Docentes desde el PyConAr 2019.

Experiencia en:

Desarrollo de software y sistemas informáticos

Docencia

Gestión de proyectos

Gerenciamiento de operaciones

Donde:

Farmacity (CABA), Julasoft S.A., Cooperativa de Crédito La Plata y otros.

Ministerio de Economía (Bs.As), Ministerio de Cultura y Educación (Bs.As.)

Distrito Informático de Gran La Plata

Comunidad La Plata de PMIBA.

Lic. FELIPE MORALES, INSTITUTO DE INGENIERÍA Y AGRONOMÍA, UNAJ, 2021

CONTEXTO

*Dictado de clases a 3 comisiones de la materia **Fundamentos de Informática** en la Universidad Nacional Arturo Jauretche (Florencio Varela, Bs. As.).*

Destinatarios: Alumnos de ingeniería

Modalidad: Teórico-Práctica

Concepto áulico: Aula-Laboratorio-Taller-Virtual

Duración: Cuatrimestral (en ambos cuatrimestres)

Estudiantes por Comisión (en promedio): 30

Antigüedad: 9 años.

Total alumnos: 1620 (9 años * 30 estudiantes * 3 comisiones * 2 cuatrimestres)

PROBLEMÁTICA DETECTADA

Bajo interés en aprender

Estudiantes de carreras no informáticas

Baja comprensión de texto

Desinterés por lo desconocido

Bajo nivel de aprobados

Infraestructura técnica y edilicia acotada

Modalidad virtual (años 2020 y 2021)

TESTIMONIOS DE ALUMNOS QUE ESTUDIAN CARRERAS INFORMÁTICAS:

“Yo quiero programar cosas en la web”

“Buenísimo profe, con esto ya puedo hacer sistemas”

“No entiendo porque los programas no se hacen solos”

“El programa hace lo que quiere. Yo no tengo la culpa”

“Programar en una pantalla negra no es programar, es solo escribir”

“A mí no me gusta empezar una oración en minúscula”

TESTIMONIOS DE ALUMNOS QUE **NO** ESTUDIAN CARRERAS INFORMÁTICAS:

“¿Para qué vamos a programar?”

“Esto no me va a salir”

“Esto no sirve, es como matemática, una pérdida de tiempo”

“Funciones no vimos en el secundario”

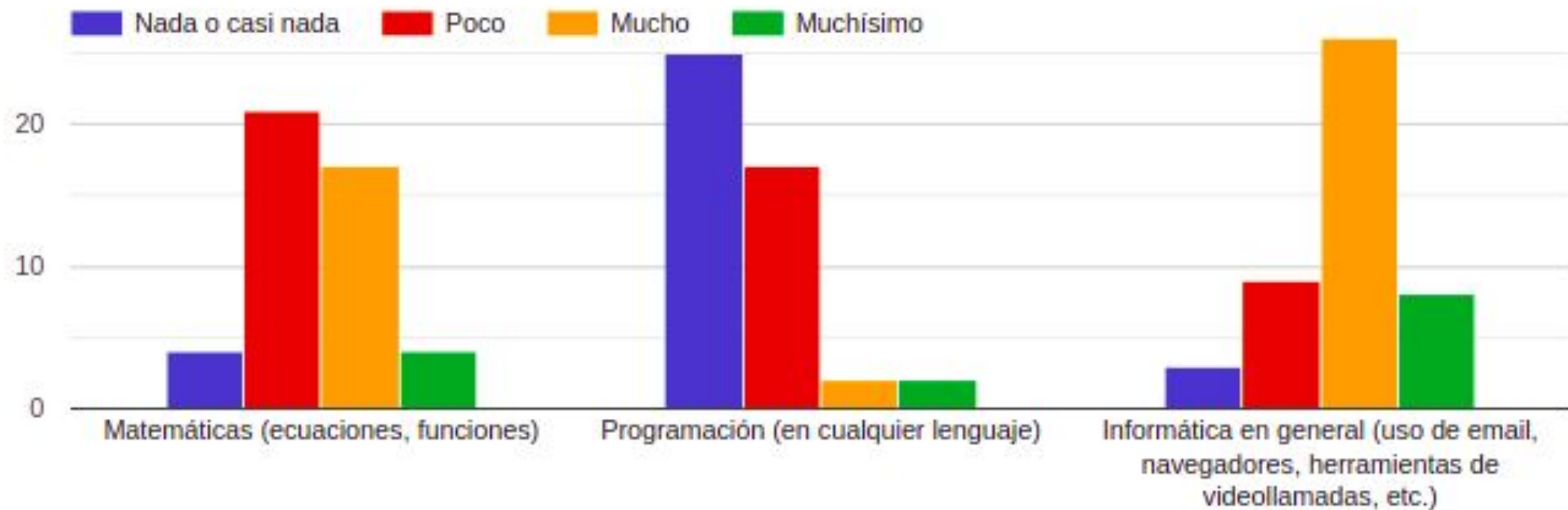
“¿En serio vamos a programar un juego? Es imposible”

“Casi seguro que la voy a tener que recursar”

“No entiendo nada”

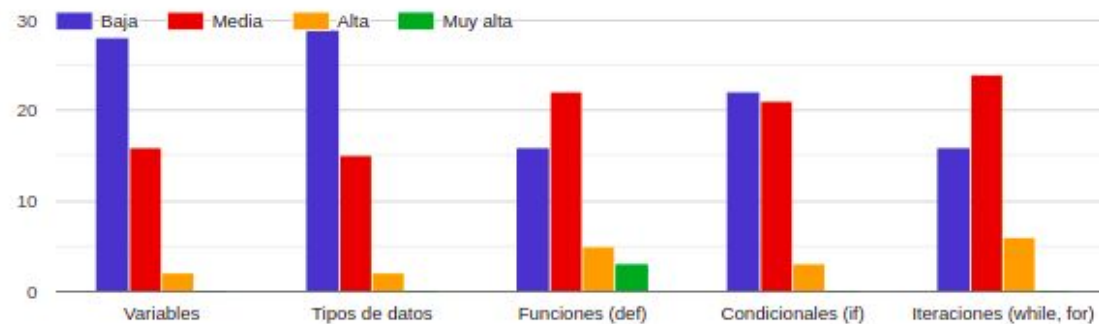
DATOS ENCUESTA 2021.C2:

1. Conocimiento previos al momento de empezar la cursada

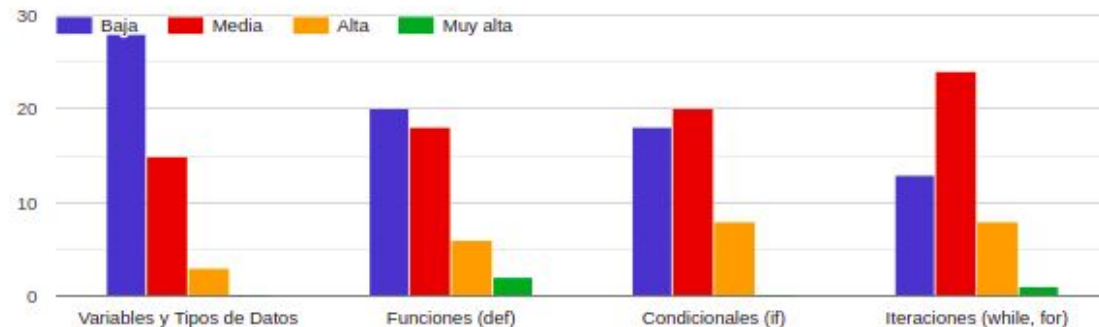


DATOS ENCUESTA 2021.C2:

2. Dificultad para entender la teoría:



3. Dificultad para hacer los trabajos prácticos:



DESAFÍOS:

- 1. ARRAIGO MATEMÁTICO*
- 2. SELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN*
- 3. DINÁMICA DE APRENDIZAJE*
- 4. “APRENDER A ANDAR EN BICICLETA”*
- 5. ANÉCDOTAS y NOVEDADES*

ARRANQUEMOS ENTONCES



Lic. FELIPE MORALES, INSTITUTO DE INGENIERÍA Y AGRONOMÍA, UNAJ, 2021

1. ARRAIGO MATEMÁTICO

- 1. DESCRIPCIÓN > MODELADO**
- 2. TEORÍA DE CONJUNTOS > TIPOS DE DATOS**
- 3. FUNCIONES > FUNCIONES**
- 4. ECUACIONES > CASOS DE PRUEBAS**
- 5. MATRICES > LISTAS, LISTAS DE LISTAS**

2. LENGUAJE FÁCIL DE APRENDER

"Hola Mundo" en distintos lenguajes

C++

```
#include <iostream>
int main() {
    cout<<"Hola, mundo"<<endl;
    return 0;
}
```

F#

```
printf "Hola, mundo\n"
```

C#

```
class HolaMundo {
    static void Main() {
        System.Console.WriteLine("Hola, mundo");
    }
}
```

Javascript

```
console.log("Hola, mundo");
```

Java

```
class HolaMundo {
    static public void main(String args[]) {
        System.out.println("Hola, mundo");
    }
}
```

Kotlin

```
fun main(args : Array<String>) {
    println("Hola, mundo")
}
```

Visual Basic .Net

```
Class HolaMundo
    Public Shared Sub Main()
        WriteLine("Hola, mundo")
    End Sub
End Class
```

Python 3

```
print("Hola, mundo")
```

Ruby

```
puts "Hola, mundo"
```

Pascal

```
program HolaMundo;
begin
    writeln('Hola, mundo');
end.
```

C

```
#include <stdio.h>
int main() {
    printf("Hola, mundo");
    return 0;
}
```


2. LENGUAJE EXPERIMENTADO

Python Timeline

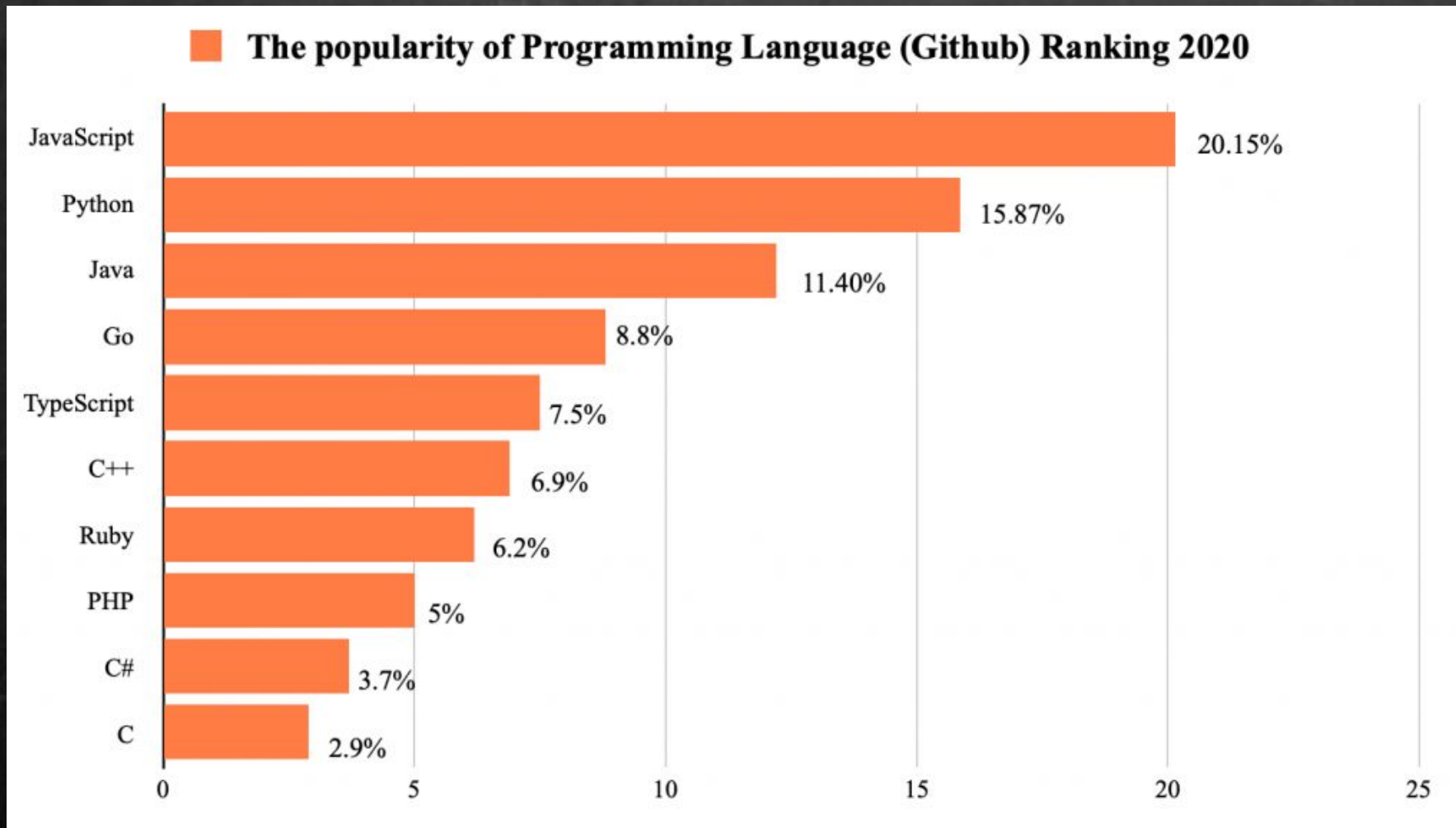


Python was created by Guido van Rossum

3. LENGUAJE (USOS)



3. LENGUAJE (POPULARIDAD)



2. SELECCIÓN DEL LENGUAJE (SIMPLICIDAD)

Lenguaje de alto nivel, fácil de aprender

Expresivo y legible

Sintaxis elegante y tipado dinámico y fuerte

Multiparadigma

Interpretado

Multiplataforma

2. SELECCIÓN DEL LENGUAJE (SOFTWARE LIBRE)

Libertad 0: Libertad de *usar* el programa, con cualquier propósito.

Libertad 1: Libertad de *estudiar* cómo funciona el programa, y *adaptarlo* a necesidades propias

Libertad 2: Libertad de *distribuir* copias

Libertad 3: Libertad de *mejorar*, y *hacer públicas* las mejoras a los demás, de modo que toda la comunidad se beneficie.

3. DINÁMICA DE APRENDIZAJE

- 1. ¿Primero la teoría o la práctica? Pros y contras*
- 2. Aportes de los estudiantes: ¿es válido? o ¡Es válido!*
- 3. Repaso al comienzo de TODAS las clases*
- 4. Entrega de ejercicios puntuales de los trabajos prácticos*
- 5. Evaluación permanente*
- 6. Propuesta de desarrollo conjunto*

4. “APRENDER A ANDAR EN BICICLETA”

1. *¿Cómo se enseña?*
2. *¿Cómo se aprende?*
3. *El docente como “creador de puentes”*
4. *Practicar, practicar, practicar*
5. *Programar es como “andar en bicicleta”*

5. ANÉCDOTAS Y NOVEDADES

1. *Avances tecnológicos como aliados*
2. *Anécdotas como disparadores*
3. *Facilitadores de la interacción*
4. *Mecanismo de inclusión*
5. *Generadores de debate*

CONCLUSIONES

*El docente como **facilitador** de aprendizaje permanente.*

Ser tan *DOCENTE* como *PROGRAMADOR!*

APRENDER MIENTRAS ENSEÑAMOS!

TIP: EL ZEN DE PYTHON (TIM PETERS)

HERMOSO ES MEJOR QUE FEO.

EXPLÍCITO ES MEJOR QUE IMPLÍCITO.

SIMPLE ES MEJOR QUE COMPLEJO.

COMPLEJO ES MEJOR QUE COMPLICADO.

SENCILLO ES MEJOR QUE ANIDADO.

ESCASO ES MEJOR QUE DENSO.

LA LEGIBILIDAD CUENTA.

LOS CASOS ESPECIALES NO SON LO SUFICIENTEMENTE ESPECIALES PARA ROMPER LAS REGLAS.

LO PRÁCTICO LE GANA A LA PUREZA.

LOS ERRORES NO DEBE PASAR EN SILENCIO. A MENOS QUE SEAN SILENCIADOS.

EN CARA A LA AMBIGÜEDAD, RECHAZAR LA TENTACIÓN DE ADIVINAR.

DEBE HABER UNA - Y PREFERIBLEMENTE SÓLO UNA - MANERA OBVIA DE HACERLO.

AUNQUE ESA MANERA PUEDE NO SER OBVIA EN UN PRIMER MOMENTO A MENOS QUE SEAS HOLANDÉS.

AHORA ES MEJOR QUE NUNCA.

AUNQUE "NUNCA" ES A MENUDO MEJOR QUE "AHORA MISMO".

SI LA APLICACIÓN ES DIFÍCIL DE EXPLICAR, ES UNA MALA IDEA.

SI LA APLICACIÓN ES FÁCIL DE EXPLICAR, PUEDE SER UNA BUENA IDEA.

LOS ESPACIOS DE NOMBRES SON UNA GRAN IDEA ¡HAY QUE HACER MÁS DE ESO!

EL DESAFÍO DE

~~ENSEÑAR~~

APRENDER

PYTHON

A NIVEL UNIVERSITARIO

Grupo de Telegram: <https://t.me/pyardocentes>



MUCHAS GRACIAS!!!

felipe.morales.querol@gmail.com

*«El maestro ideal es aquel que se pone en el papel de un puente por el cual invita a sus alumnos a cruzar y que luego de haberlos ayudado en el cruce, se desploma con alegría, alentandolos a crear sus propios puentes» **Nikos Kazantzakis***