



# MIDIENDO LA PERFORMANCE DE NUESTRO CÓDIGO

Jornadas de Calidad de Software y Agilidad 2023  
FCEQyN - UNaM | FaCENA - UNNE | FRRe - UTN

# .self()

Martín Rey

Docente en FCEQyN - UNaM hace 13+  
años en las áreas de Ingeniería de  
Software | Auditoría Informática |  
Programación | Datos

Nodocente en FCE - UNaM en la  
Dirección de Tecnologías para la  
Gestión trabajando con Software +  
Datos + *otras cosas*





## Vamos a conocernos

Una serie de preguntas para saber  
un poco más de Uds.





“Mi código funciona, a veces  
incluso fuera de mi máquina, y  
con eso es suficiente”

¿SENSACIONES?

# AGENDA

01

Intro

03

Demos

02

Algunas  
propuestas

04

Cierre (?)



01

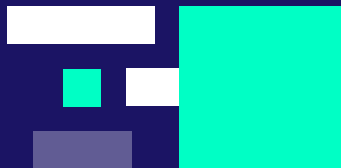
# INTRODUCCIÓN



# Rendimiento?

En términos simples vamos a hablar de rendimiento a partir de analizar el funcionamiento de un bloque de código o aplicación en términos algunas métricas que sean de interés según el problema que **se haya detectado**.

# Aspectos a tener en cuenta



## Velocidad

Tiempo de ejecución

## Consumo de memoria

Uso de este recurso



## Consumo de CPU / GPU

Uso de este recurso

## Respuesta

Velocidad que ve el usuario





# ¿Por dónde empezamos?

Esta actividad (**profiling**) del código puede realizarse de manera incremental sobre código ya existente o podemos ir automatizando mediciones sobre código que está en desarrollo.



02

ALGUNAS PROPUESTAS

# ¿Qué herramientas podemos usar?

Hay diferentes alternativas para medir performance del código. Dependiendo del lenguaje y objetivo de la medición se va a poder optar por una u otra.

# Tres instancias

## Performance

Opciones de medición

Herramientas  
propias del  
lenguaje

Librerías  
específicas

Aplicaciones  
de dominio  
específico



# Herramientas propias del lenguaje

En casi todos existe una forma de tomar un *timestamp* antes y después de ejecutar algún bloque de código. Se obtiene la diferencia y así sabemos cuánto demora.

Es la opción vía *fuerza bruta*.

# Librerías específicas de profiling

En este caso son componentes que realizan la medición de uno o más aspectos del código en forma automática.

Es una forma un poco más *elegante* pero tiene sus problemas.

# Aplicaciones de dominio específico

Pueden ser desde un monitor de calidad del producto hasta una aplicación que permite ver detalles de la ejecución de algún bloque.

Dependiendo del caso algunas pueden ir en un *pipeline* de CI/CD

# ¿Qué vamos a usar?

`time()` +  
decorators

Python en  
algunos casos  
de  
optimización

scalene

Librería de  
Python que  
realiza  
profiling

pgadmin

Una  
aproximación  
para SQL





03

DEMOS

## ¿Qué ejemplos?

Para cada herramienta un  
caso para probar

## Repo

Espacio con todo el contenido  
y ejemplos para ejecutarlos  
(algunos online)





04

CIERRE

# Cierre

El rendimiento de nuestro código es un aspecto más de su calidad.

Con las herramientas vistas podemos implementar diferentes evaluaciones a lo largo de un proyecto de desarrollo y optimizar el producto.

Incluso, parte de estas evaluaciones se podrían implementar mediante pruebas y pasar a estar automatizadas en los flujos de integración y despliegue.



Preguntas?



# Muchas gracias!

CREDITS: This presentation template was created by  
Slidesgo, including icons by Flaticon, and  
infographics & images by Freepik.

Contacto: [martinrey@fceqyn.unam.edu.ar](mailto:martinrey@fceqyn.unam.edu.ar)