

# Experimentos

Muestreo, análisis y gráficos

David Alejandro González Márquez

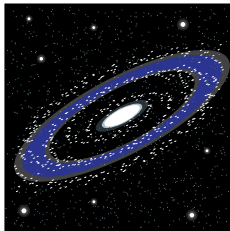
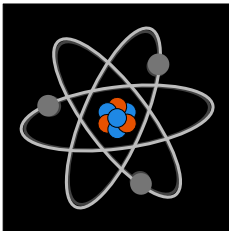
Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

## Agenda

- 1 ¿Cómo medir tiempos?
- 2 ¿Cómo armar experimentos?
- 3 ¿Qué muestran los gráficos?
- 4 Ejemplos

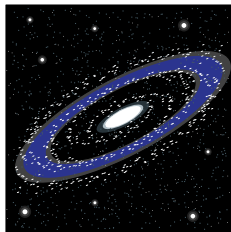
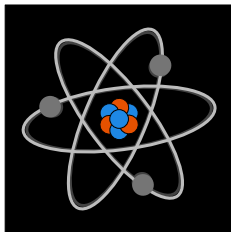
## ¿Qué medir?

Valores muy chicos o muy grandes



## ¿Qué medir?

Valores muy chicos o muy grandes



Relación entre valores sin importar la magnitud



## ¿Qué medir?

- **Tiempo** → (numerito seguido de su unidad)

Podemos medir en unidades de tiempo, como microsegundos. El problema es que para eventos que suceden muy rápidamente la precisión de los relojes es insuficiente.

## ¿Qué medir?

- **Tiempo** → (numerito seguido de su unidad)

Podemos medir en unidades de tiempo, como microsegundos. El problema es que para eventos que suceden muy rápidamente la precisión de los relojes es insuficiente.

- **Ticks de Reloj** → (numerito sin unidad)

Para obtener el contador de ticks en Intel utilizamos la instrucción `rdtsc`.

## ¿Qué medir?

- **Tiempo** → (numerito seguido de su unidad)

Podemos medir en unidades de tiempo, como microsegundos. El problema es que para eventos que suceden muy rápidamente la precisión de los relojes es insuficiente.

- **Ticks de Reloj** → (numerito sin unidad)

Para obtener el contador de ticks en Intel utilizamos la instrucción `rdtsc`.

- **Rendimiento** → (numerito en términos de porcentaje)

El rendimiento se obtiene a partir de una relación entre valores. Estos deben ser comparables.

## ¿Qué valores se espera obtener?

- **Tiempo**

Un número que represente un tiempo, si es muy grande entonces serán segundos, horas, días. Si es muy chico serán microsegundos, nanosegundos.

¿Es razonable demorar 3 segundos para ejecutar 1000 instrucciones?



## ¿Qué valores se espera obtener?

- **Tiempo**

Un número que represente un tiempo, si es muy grande entonces serán segundos, horas, días. Si es muy chico serán microsegundos, nanosegundos.

¿Es razonable demorar 3 segundos para ejecutar 1000 instrucciones?

- **Rendimiento**

Denota la diferencia de rendimiento entre dos implementaciones. Se expresa como porcentajes.

120 %, puede significar que A es 20 % más eficiente en tiempo que B.

120 %, puede significar que A demora un 20 % más que B.

10 %, puede significar que A demora el 10 % de B.

## ¿Qué valores se espera obtener?

- **Tiempo**

Un número que represente un tiempo, si es muy grande entonces serán segundos, horas, días. Si es muy chico serán microsegundos, nanosegundos.

¿Es razonable demorar 3 segundos para ejecutar 1000 instrucciones?

- **Rendimiento**

Denota la diferencia de rendimiento entre dos implementaciones. Se expresa como porcentajes.

120 %, puede significar que A es 20 % más eficiente en tiempo que B.

120 %, puede significar que A demora un 20 % más que B.

10 %, puede significar que A demora el 10 % de B.

- **Ticks de Reloj**

Son valores enteros muy grandes. Nos va a interesar la relación entre estos.

Dos implementaciones que demoran 190231359147543 y 192125445767335 ticks, tienen igual rendimiento.

## ¿Qué puede afectar la medición?

- Ruido del sistema

Nuestra aplicación **no corre sola** en el sistema, la interacción con otras aplicaciones genera ruido.

## ¿Qué puede afectar la medición?

- Ruido del sistema

Nuestra aplicación **no corre sola** en el sistema, la interacción con otras aplicaciones genera ruido.

- Datos de entrada

Puede existir alguna **característica especial** de nuestros datos de entrada que generen una medición no esperada.

## ¿Qué puede afectar la medición?

- Ruido del sistema

Nuestra aplicación **no corre sola** en el sistema, la interacción con otras aplicaciones genera ruido.

- Datos de entrada

Puede existir alguna **característica especial** de nuestros datos de entrada que generen una medición no esperada.

- Sistema Operativo

El sistema operativo y su **accionar afecta** nuestra medición. Debemos controlarlo.

## ¿Qué puede afectar la medición?

- Ruido del sistema

Nuestra aplicación **no corre sola** en el sistema, la interacción con otras aplicaciones genera ruido.

- Datos de entrada

Puede existir alguna **característica especial** de nuestros datos de entrada que generen una medición no esperada.

- Sistema Operativo

El sistema operativo y su **accionar afecta** nuestra medición. Debemos controlarlo.

- El propio procesador

Los **mecanismos de optimización** del rendimiento de un procesador afectan la medición, aumentando y disminuyendo la frecuencia de trabajo de las distintas partes.

## ¿Cómo evitar *outliers*?

Sean las siguientes muestras de datos,

```
> a = c(32, 55, 32, 54, 65, 32, 33, 54, 78, 2093486723)
```

```
> b = c(32, 55, 32, 54, 65, 32, 33, 54, 78)
```

## ¿Cómo evitar outliers?

Sean las siguientes muestras de datos,

```
> a = c(32, 55, 32, 54, 65, 32, 33, 54, 78, 2093486723)
```

```
> b = c(32, 55, 32, 54, 65, 32, 33, 54, 78)
```

```
> mean(a)
```

```
209348716
```

```
> mean(b)
```

```
48.33333
```

```
> sd(a)
```

```
662018614
```

```
> sd(b)
```

```
16.9632
```



## ¿Cómo evitar *outliers*?

Sean las siguientes muestras de datos,

```
> a = c(32, 55, 32, 54, 65, 32, 33, 54, 78, 2093486723)
```

```
> b = c(32, 55, 32, 54, 65, 32, 33, 54, 78)
```

```
> mean(a)
```

```
209348716
```

```
> mean(b)
```

```
48.33333
```

```
> sd(a)
```

```
662018614
```

```
> sd(b)
```

```
16.9632
```

Se debe poder evitar y controlar la aparición de *outliers*. En el caso de tener *outliers*, se debe poder clasificarlos y removerlos de la muestra.

**Debemos armar un protocolo para realizar nuestras mediciones**

## ¿Qué analizar?

Debemos entender qué vamos a analizar,

- **Cuánto demora una implementación:**

Obtener una medida de tiempo para relacionarla en un contexto

## ¿Qué analizar?

Debemos entender qué vamos a analizar,

- **Cuánto demora una implementación:**  
Obtener una medida de tiempo para relacionarla en un contexto
- **Comparar dos implementaciones:**  
Obtener la diferencia de tiempos o de porcentaje relativo entre dos implementaciones

## ¿Qué analizar?

Debemos entender qué vamos a analizar,

- **Cuánto demora una implementación:**  
Obtener una medida de tiempo para relacionarla en un contexto
- **Comparar dos implementaciones:**  
Obtener la diferencia de tiempos o de porcentaje relativo entre dos implementaciones
- **Comparar el uso de distintas instrucciones:**  
Obtener una medida de mejora con respecto a utilizar un determinado conjunto de instrucciones con respecto a otro

## ¿Qué analizar?

Debemos entender qué vamos a analizar,

- **Cuánto demora una implementación:**  
Obtener una medida de tiempo para relacionarla en un contexto
- **Comparar dos implementaciones:**  
Obtener la diferencia de tiempos o de porcentaje relativo entre dos implementaciones
- **Comparar el uso de distintas instrucciones:**  
Obtener una medida de mejora con respecto a utilizar un determinado conjunto de instrucciones con respecto a otro
- **Medir el rendimiento de una implementación:**  
Obtener el porcentaje de mejora de una implementación con respecto a una implementación patrón

## ¿Qué analizar?

Debemos entender qué vamos a analizar,

- **Cuánto demora una implementación:**  
Obtener una medida de tiempo para relacionarla en un contexto
- **Comparar dos implementaciones:**  
Obtener la diferencia de tiempos o de porcentaje relativo entre dos implementaciones
- **Comparar el uso de distintas instrucciones:**  
Obtener una medida de mejora con respecto a utilizar un determinado conjunto de instrucciones con respecto a otro
- **Medir el rendimiento de una implementación:**  
Obtener el porcentaje de mejora de una implementación con respecto a una implementación patrón
- **Factor limitante en una implementación:**  
Obtener una medida relativa de rendimiento con respecto a forzar un factor (ej. memoria, saltos)

## ¿Qué analizar?

Debemos entender qué vamos a analizar,

- **Cuánto demora una implementación:**  
Obtener una medida de tiempo para relacionarla en un contexto
- **Comparar dos implementaciones:**  
Obtener la diferencia de tiempos o de porcentaje relativo entre dos implementaciones
- **Comparar el uso de distintas instrucciones:**  
Obtener una medida de mejora con respecto a utilizar un determinado conjunto de instrucciones con respecto a otro
- **Medir el rendimiento de una implementación:**  
Obtener el porcentaje de mejora de una implementación con respecto a una implementación patrón
- **Factor limitante en una implementación:**  
Obtener una medida relativa de rendimiento con respecto a forzar un factor (ej. memoria, saltos)
- **Análisis del comportamiento de una implementación:**  
Rendimiento bajo distinto conjunto de parámetros de entrada.

## ¿Qué resultado se espera?

- **A mantiene** su rendimiento cuando se modifican los parámetros de entrada.



## ¿Qué resultado se espera?

- A **mantiene** su rendimiento cuando se modifican los parámetros de entrada.
- A **varía** su rendimiento cuando se modifican los parámetros de entrada.

## ¿Qué resultado se espera?

- **A mantiene** su rendimiento cuando se modifican los parámetros de entrada.
- **A varía** su rendimiento cuando se modifican los parámetros de entrada.
  - Variación correlacionada con una función, predecible.
  - No predecible, relacionada con factores externos a los parámetros de entrada.

## ¿Qué resultado se espera?

- **A mantiene** su rendimiento cuando se modifican los parámetros de entrada.
- **A varía** su rendimiento cuando se modifican los parámetros de entrada.
  - Variación correlacionada con una función, predecible.
  - No predecible, relacionada con factores externos a los parámetros de entrada.
- **A es más rápido** que **B**
  - en tiempo
  - en porcentaje

## ¿Qué resultado se espera?

- A **mantiene** su rendimiento cuando se modifican los parámetros de entrada.
- A **varía** su rendimiento cuando se modifican los parámetros de entrada.
  - Variación correlacionada con una función, predecible.
  - No predecible, relacionada con factores externos a los parámetros de entrada.
- A es **más rápido** que B
  - en tiempo
  - en porcentaje
- A se **comporta igual** a B
  - diferencia no significativa

## ¿Qué resultado se espera?

- **A mantiene** su rendimiento cuando se modifican los parámetros de entrada.
- **A varía** su rendimiento cuando se modifican los parámetros de entrada.
  - Variación correlacionada con una función, predecible.
  - No predecible, relacionada con factores externos a los parámetros de entrada.
- **A es más rápido** que **B**
  - en tiempo
  - en porcentaje
- **A se comporta igual** a **B**
  - diferencia no significativa
- **A y B se comportan muy diferente**
  - analizar casos independientemente

## ¿Cómo armar datos de entrada?

Los datos de entrada se toman de casos reales o se generan artificialmente.

## ¿Cómo armar datos de entrada?

Los datos de entrada se toman de casos reales o se generan artificialmente.

Se crean,

- **Modificando una variable**

Ej. Alterando el valor de una componente de color en una imagen

- **Modificando un conjunto de variables bajo una regla**

Ej. Alterando el tamaño de una imagen en ancho y alto al mismo tiempo

## ¿Cómo armar datos de entrada?

Los datos de entrada se toman de casos reales o se generan artificialmente.

Se crean,

- **Modificando una variable**

Ej. Alterando el valor de una componente de color en una imagen

- **Modificando un conjunto de variables bajo una regla**

Ej. Alterando el tamaño de una imagen en ancho y alto al mismo tiempo

Tener en cuenta,

- Variables seleccionadas
- Cantidad total de entradas
- Entrada común o tipo



## ¿Cómo analizar resultados?

La etapa de discusión de los resultados, implica cuestionar las hipótesis sobre las que construimos nuestro experimento.

- ¿El resultado es esperado?
- ¿Existe algún factor que no estamos teniendo en consideración?
- ¿Es posible validar nuestro resultado con un experimento control?
- Si el resultado no es el esperado, ¿el experimento es incorrecto?
- Si los resultados son correctos, ¿hacer un nuevo experimento?

## ¿Qué se quiere mostrar?

Dependiendo que se quiera mostrar, se puede utilizar un tipo de gráfico u otro.

- **Rendimiento / Tiempo:** Area, Barras, Líneas
- **Relaciones:** Líneas, Dispersión
- **Porcentajes:** Torta, Area o Barras acumuladas
- **Comportamiento / Variación:** Líneas, Superficie

## ¿Qué colores? ¿Qué tipo de gráfico? ¿Qué escala?

### Colores

- Gusto / estética
- Resaltar un dato sobre otro
- Pensar en el medio de distribución (impreso)

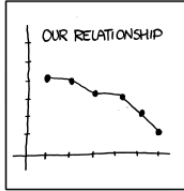
### Tipo

- El que mejor muestre la información (ojo con 3d)
- Que sirva para explicar (sin información de más)

### Escala

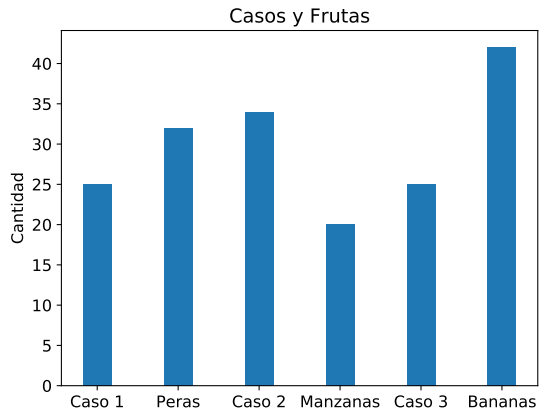
- Lineal o Logarítmica
- Sobre qué eje va cada escala
- Valores en los ejes

## Humor de XKCD



## Ejemplo 1: Comparando datos

No podemos agrupar en un gráfico cosas que no tiene nada que ver entre ellas.

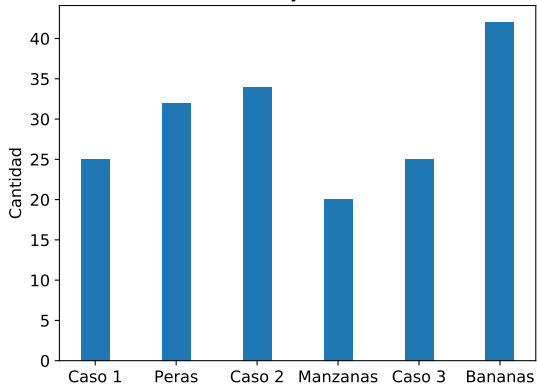


## Ejemplo 1: Comparando datos

No podemos agrupar en un gráfico cosas que no tiene nada que ver entre ellas.

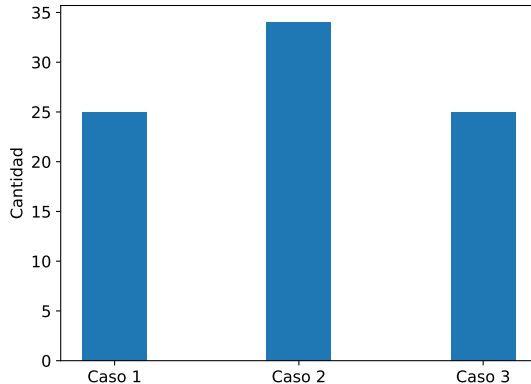
**MAL**

Casos y Frutas



**BIEN**

Casos

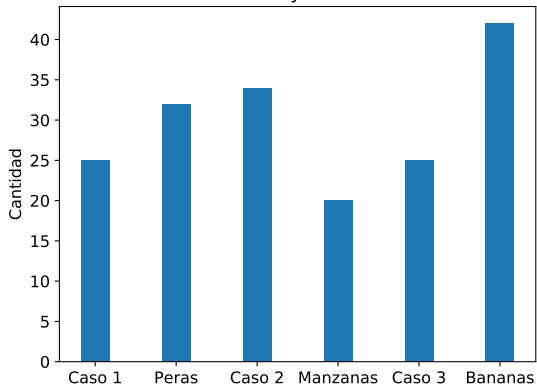


## Ejemplo 1: Comparando datos

No podemos agrupar en un gráfico cosas que no tiene nada que ver entre ellas.

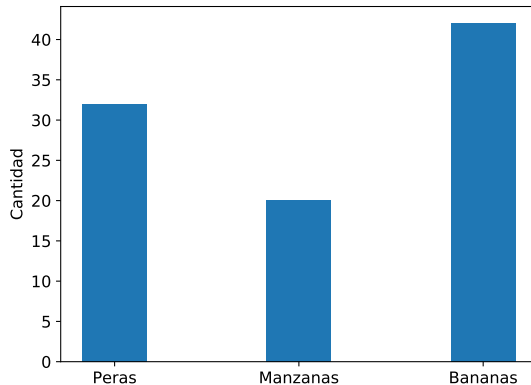
**MAL**

Casos y Frutas



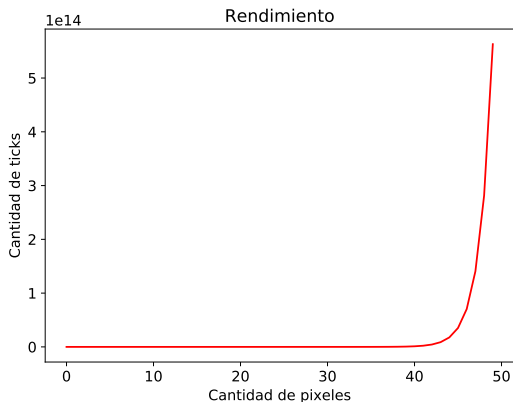
**BIEN**

Frutas



## Ejemplo 2: Analizando límites, tendencias o comportamientos anómalos

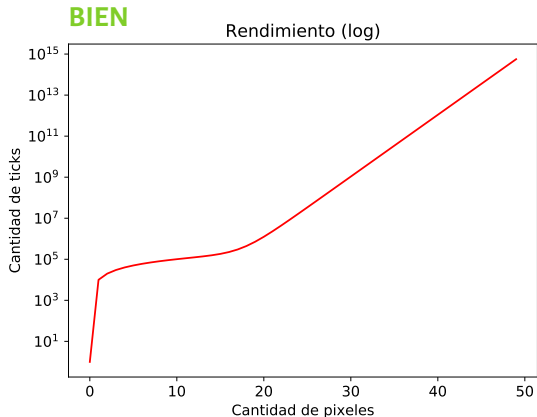
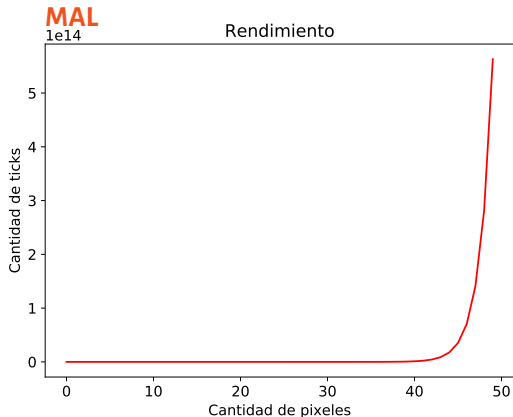
Escalas: Podemos setear la escala de cada eje con distintos criterios (lineal, log, loglog, etc).





## Ejemplo 2: Analizando límites, tendencias o comportamientos anómalos

Escalas: Podemos setear la escala de cada eje con distintos criterios (lineal, log, loglog, etc).



**Cambiamos el eje y a escala logarítmica**

## Ejemplo 3: Proporciones

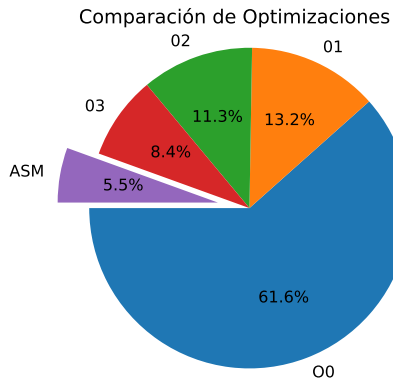
Un gráfico de torta sirve para ver proporciones en un TODO.

No! para agrupar valores que no constituyen un todo.

## Ejemplo 3: Proporciones

Un gráfico de torta sirve para ver proporciones en un TODO.

No! para agrupar valores que no constituyen un todo.

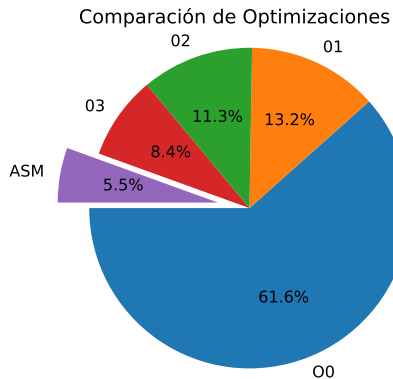


## Ejemplo 3: Proporciones

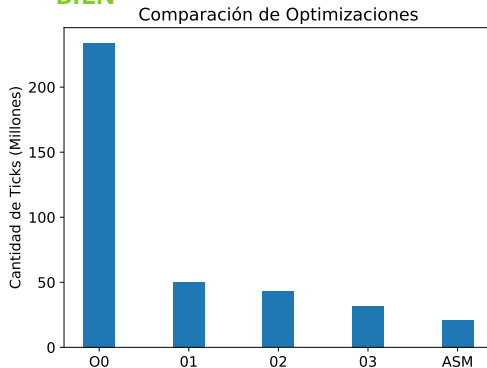
Un gráfico de torta sirve para ver proporciones en un TODO.

No! para agrupar valores que no constituyen un todo.

**MAL**



**BIEN**



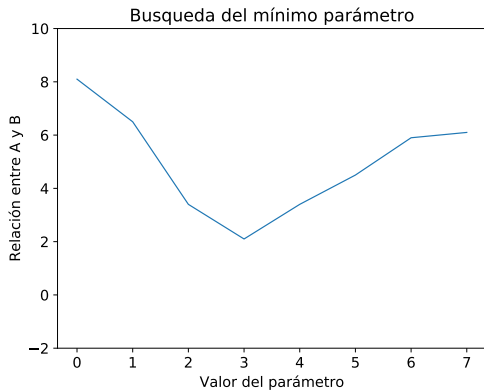
**Barras es la opción**

## Ejemplo 4: Optimizar un parámetro

Tenemos que encontrar el parámetro óptimo.

## Ejemplo 4: Optimizar un parámetro

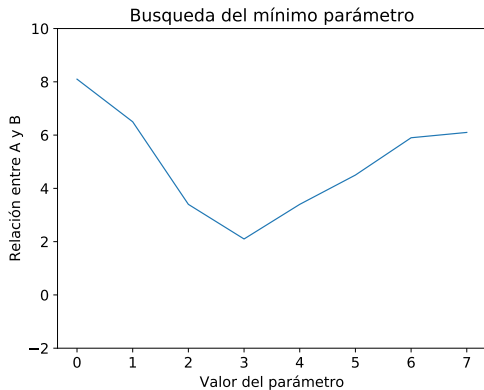
Tenemos que encontrar el parámetro óptimo.



## Ejemplo 4: Optimizar un parámetro

Tenemos que encontrar el parámetro óptimo.

**MAL**

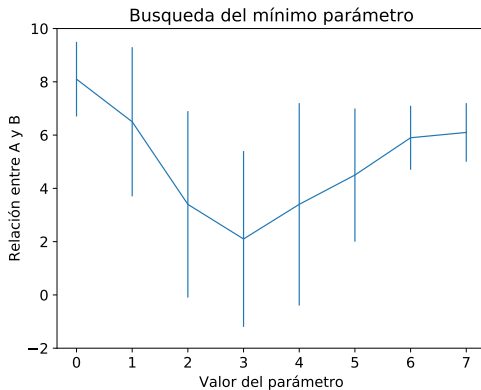
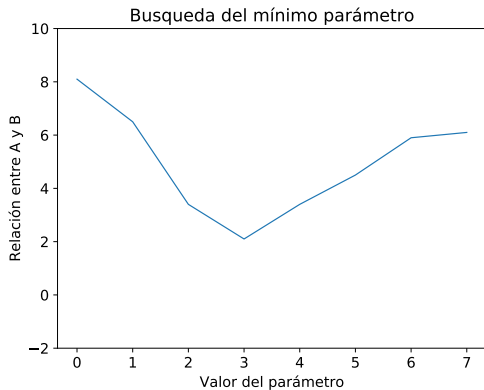


**No conocemos el error de nuestros datos**

## Ejemplo 4: Optimizar un parámetro

Tenemos que encontrar el parámetro óptimo.

**MAL**



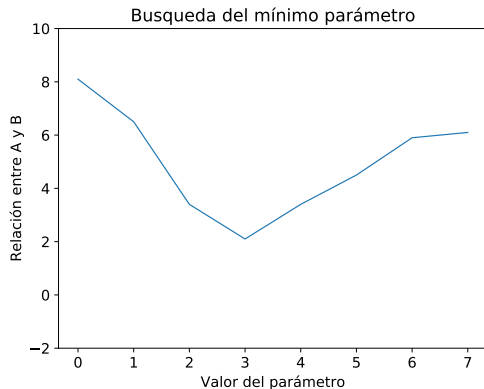
**Utilizamos barras de error**



## Ejemplo 4: Optimizar un parámetro

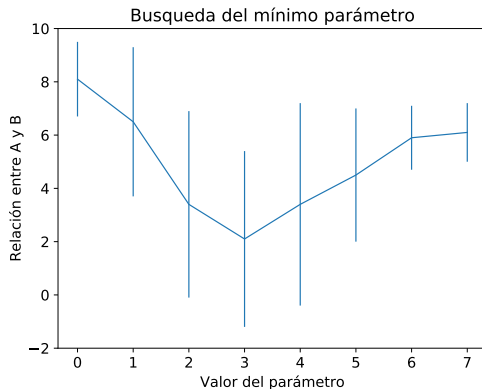
Tenemos que encontrar el parámetro óptimo.

**MAL**



**Utilizamos barras de error**

**MAL**

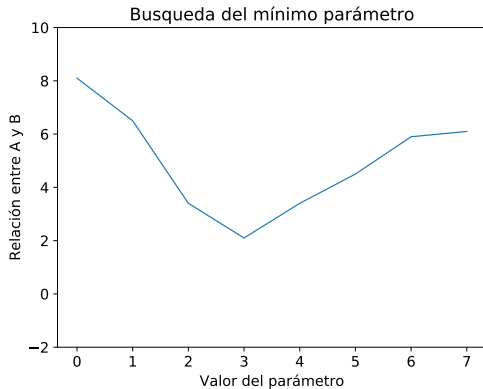


**Los errores NO están acotados**

## Ejemplo 4: Optimizar un parámetro

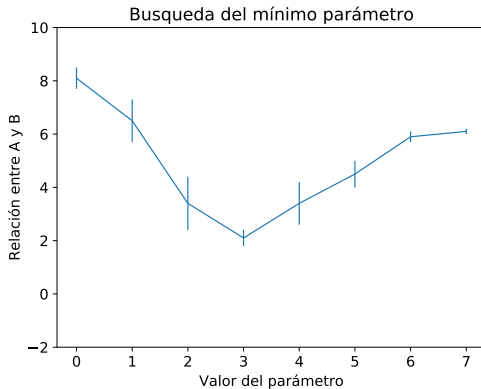
Tenemos que encontrar el parámetro óptimo.

**MAL**



**Utilizamos barras de error**

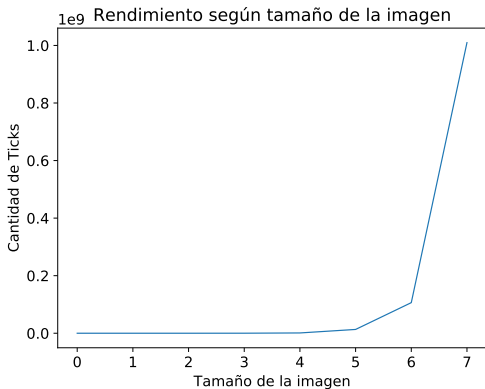
**BIEN**



**Los errores están acotados**

## Ejemplo 5: Analizando límites

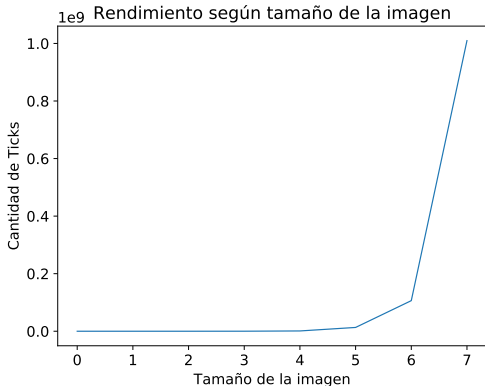
Queremos ver donde la performance varía.



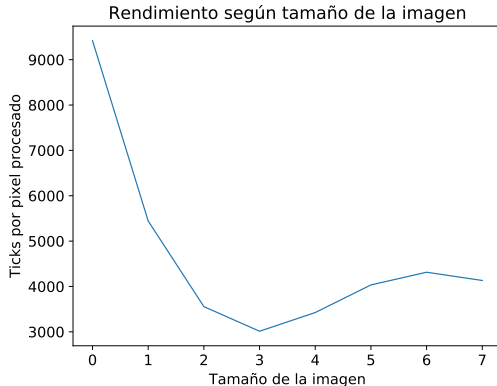
## Ejemplo 5: Analizando límites

Queremos ver donde la performance varía.

**MAL**



**BIEN**



**Normalizamos los datos**

## Pequeños pero no menos importantes consejos

- Los resultados deben ser **reproducibles y consistentes**

## Pequeños pero no menos importantes consejos

- Los resultados deben ser **reproducibles y consistentes**
- Ser **ordenado en las explicaciones**, explicar tanto el código como los experimentos, datos y resultados

## Pequeños pero no menos importantes consejos

- Los resultados deben ser **reproducibles y consistentes**
- Ser **ordenado en las explicaciones**, explicar tanto el código como los experimentos, datos y resultados
- **No adjuntar gráficos de más**, sólo deben ir los que hagan falta

## Pequeños pero no menos importantes consejos

- Los resultados deben ser **reproducibles y consistentes**
- Ser **ordenado en las explicaciones**, explicar tanto el código como los experimentos, datos y resultados
- **No adjuntar gráficos de más**, sólo deben ir los que hagan falta
- El promedio de una muestra debe estar **acompañado de su desvío**



## Pequeños pero no menos importantes consejos

- Los resultados deben ser **reproducibles y consistentes**
- Ser **ordenado en las explicaciones**, explicar tanto el código como los experimentos, datos y resultados
- **No adjuntar gráficos de más**, sólo deben ir los que hagan falta
- El promedio de una muestra debe estar **acompañado de su desvío**
- No todo se puede explicar, existen comportamientos **inexplicables**

## Pequeños pero no menos importantes consejos

- Los resultados deben ser **reproducibles y consistentes**
- Ser **ordenado en las explicaciones**, explicar tanto el código como los experimentos, datos y resultados
- **No adjuntar gráficos de más**, sólo deben ir los que hagan falta
- El promedio de una muestra debe estar **acompañado de su desvío**
- No todo se puede explicar, existen comportamientos **inexplicables**
- El informe debe ser un **trabajo integral**, consistente y bien escrito

# Informe

## 1 Implementación

- Explicación general de la solución
- Detalles de implementación
- Uso de constantes y memoria

## 2 Análisis preeliminar

- Comparación de rendimiento de ASM vs C
- Comparar para distintos tamaños, relaciones entre implementaciones

## 3 Hipótesis de trabajo

- Conjunto de ideas de experimentos
- Afirmaciones que buscan probar verdaderas
- Deben ser concisas y claras

## 4 Diseño experimental

- Explicación de como y que van a medir
- Explicación del conjunto de datos de entrada
- Detalles de la plataforma y la configuración de la misma

## 5 Resultados y Análisis

- Resultados obtenidos, gráficos y tablas
- Explicación e interpretación de los resultados obtenidos

## 6 Conclusiones

- Relación entre las hipótesis de trabajo y resultados

## Bibliografía: Fuentes y material adicional

- Convenciones de llamados a función en x86:  
[https://en.wikipedia.org/wiki/X86\\_calling\\_conventions](https://en.wikipedia.org/wiki/X86_calling_conventions)
- Notas sobre System V ABI:  
[https://wiki.osdev.org/System\\_V\\_ABI](https://wiki.osdev.org/System_V_ABI)
- Documentación de NASM:  
<https://nasm.us/doc/>
- Artículo sobre el flag -pie:  
<https://eklitzke.org/position-independent-executables>
- Documentación de System V ABI:  
[https://uclibc.org/docs/psABI-x86\\_64.pdf](https://uclibc.org/docs/psABI-x86_64.pdf)
- Manuales de Intel:  
<https://software.intel.com/en-us/articles/intel-sdm>

# ¡Gracias!

Recuerden leer los comentarios al final de este video por aclaraciones o fe de erratas.