

Potenciando Javascript con WebAssembly: un caso aplicado a la Educación

Gonzalo Fernández -- Desarrollador Full-Stack



[vadokdev](#)



[VadokDev](#)

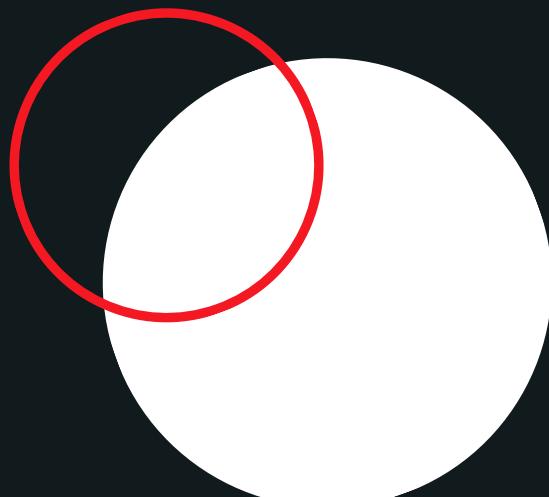
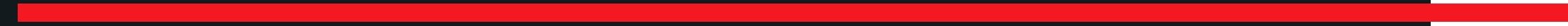


[VadokDev](#)



[vadok.dev](#)





AGENDA

1. Preámbulo
2. Tareas de Programación
3. Complejidad de una Tarea
4. Framework Ishvel
5. Conclusiones y Trabajo Futuro



GONZALO FERNÁNDEZ

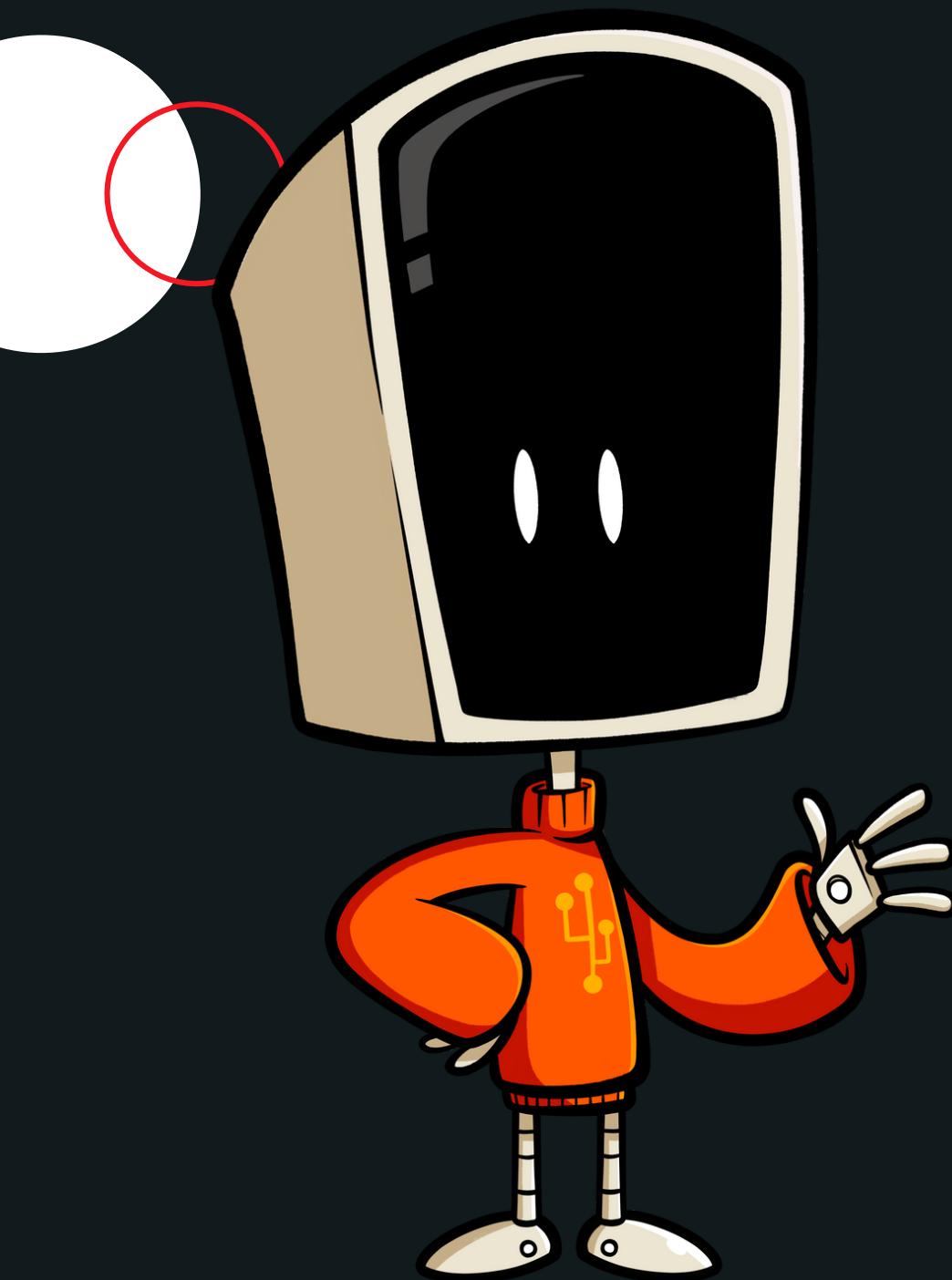
DESARROLLADOR FULL-STACK

Aspirante a trompetista de jazz, y futuro ingeniero civil en informática de la UTFSM

Amante del desarrollo de software y su unión con la docencia. Todo el aporte que se pueda realizar con software, genera un interés en mí para investigar



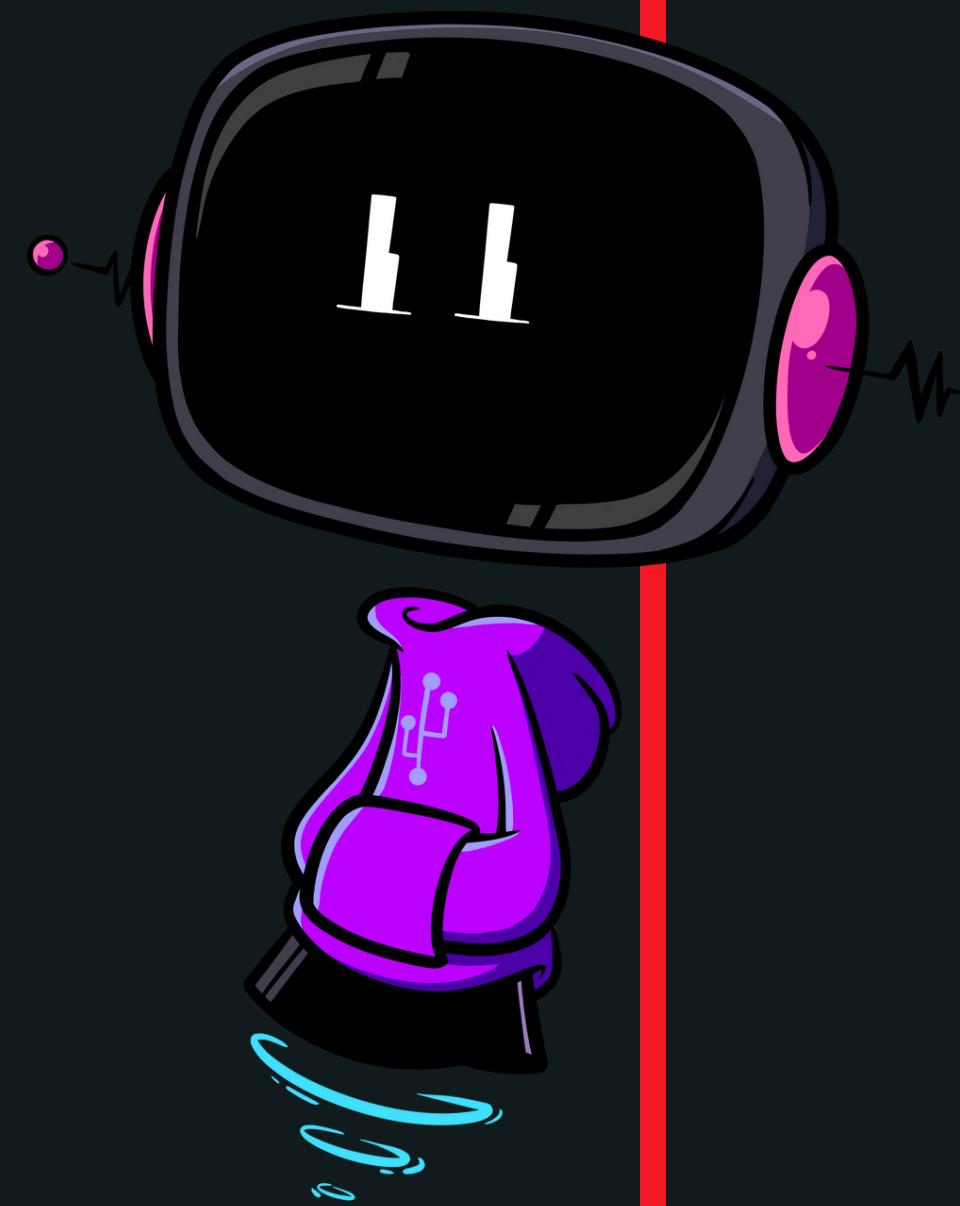
TALLERES ÓCI>LABS.



Talleres orientados a la enseñanza de la programación para estudiantes desde 7mo Básico a 4to Medio.

Existen 3 Niveles:

- **Básico** – introducción al pensamiento computacional a través de Scratch.
- **Intermedio** – introducción a la programación a través de C++.
- **Avanzado** – conceptos avanzados de programación y competencia OCI.



TALLERES ÓCI>LABS.

Veamos algunas cifras de los talleres desde 2019

3.292

ESTUDIANTES HAN
PASADO POR
NUESTROS TALLERES

87

TUTORES Y TUTORAS
NOS HAN APOYADO

13

ESTUDIANTES HAN
CLASIFICADO A LA
COMPETENCIA DE LA OCI

4

AÑOS APOYANDO EN
LA ENSEÑANZA DE LA
PROGRAMACIÓN

Programación IWI-131



¡Y muchos más!

Clases de Programación

Veamos que ocurre en 1 semana de clases,
equivalente a 1 UVA (Unidad Virtual de Aprendizaje)



**TRABAJO
PREVIO**

(En Aula) Cápsulas de
video y Evaluación
Formativa

SESIÓN 1

70 m Aclaración de Dudas y
Ejercicios Habilitadores:

- Parsons
- Completar Código
- Corregir Código
- Ruteo

Control y Ejercicios
Introductorios

SESIÓN 2

70 m Ejercicios Supervisados y
Andamiaje.

AYUDANTÍA

70 m Apoyo a la Tarea y Ejercicios

Tarea Semanal (90 m.)

Ejercicio de Corrección Automática (60 m.)

Consulta a Profesores en horario de atención - Consultas y revisión de Foro Piazza



TAREAS DE PROGRAMACIÓN

¿QUÉ SON Y POR QUÉ LAS USAMOS?



Los derechos de autor de las obras musicales se cobran a quienes hacen uso lucrativo de estas creaciones. Por ejemplo, a quienes organizan conciertos, a las emisoras de radio, etc. Los dineros recolectados son distribuidos a los compositores y las compositoras de las obras que fueron utilizadas, de acuerdo con un criterio de proporcionalidad. Primero, se obtiene un valor por obra, dividiendo el monto recaudado en una utilización particular (por ejemplo, un concierto) entre la cantidad de obras que fueron interpretadas. Después, el monto asignado a cada obra se reparte equitativamente entre las compositoras y los compositores de la obra. Para apoyar el proceso de distribución de los dineros, se cuenta con varios archivos, cuyo formato se describe a continuación.

El archivo de compositores contiene compositores(as), junto con un número único que les identifica, denominado a nivel internacional como número IPI (*Interested Party Information*). Los campos se separan por un punto y coma. Un extracto de este archivo se muestra a continuación:

compositores.csv

```
00234720294;GROHL DAVID
00403809281;SHIFLETT CHRISTOPHER A
00181675549;MENDEL NATE
00344920759;HAWKINS OLIVER TAYLOR
00226132995;GONZALEZ RIOS JORGE
00245234584;LINDL ROMERO ROBERTO
00244594943;HENRIQUEZ PETTINELLI ALVARO
00201037340;MORISSETTE ALANIS
...
```

En un curso introductorio de programación, las tareas son una herramienta que cumple múltiples propósitos de cara al proceso educativo.

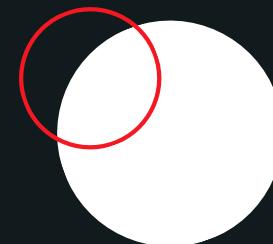
”

**Las tareas son parte
importante del proceso de
enseñanza y aprendizaje**

ALLISON BOYE

PH.D. TEACHING, LEARNING, AND PROFESSIONAL DEVELOPMENT CENTER
TEXAS TECH UNIVERSITY

”



¿Para qué sirven?



MEDIR

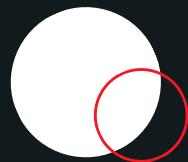
Midén el aprendizaje del estudiante a lo largo del curso, de modo tal de permitir al docente tomar medidas durante el semestre lectivo

¿Para qué sirven?



MEDIR

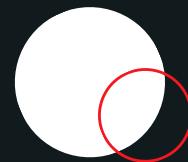
Midén el aprendizaje del estudiante a lo largo del curso, de modo tal de permitir al docente tomar medidas durante el semestre lectivo



INCENTIVAR

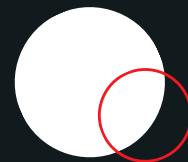
Brindan un acercamiento efectivo sobre como lo que el estudiante está aprendiendo, tiene un uso real

¿Para qué sirven?



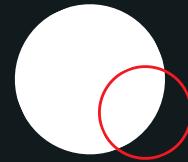
MEDIR

Midén el aprendizaje del estudiante a lo largo del curso, de modo tal de permitir al docente tomar medidas durante el semestre lectivo



INCENTIVAR

Brindan un acercamiento efectivo sobre como lo que el estudiante está aprendiendo, tiene un uso real



APLICAR

Instan al estudiante a aplicar los conocimientos adquiridos a lo largo del curso, obteniendo una experiencia educativa menos expositiva y más práctica

¿Para qué sirven?



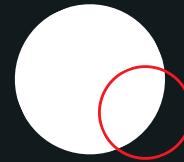
MEDIR

Midén el aprendizaje del estudiante a lo largo del curso, de modo tal de permitir al docente tomar medidas durante el semestre lectivo



INCENTIVAR

Brindan un acercamiento efectivo sobre como lo que el estudiante está aprendiendo, tiene un uso real



APLICAR

Instan al estudiante a aplicar los conocimientos adquiridos a lo largo del curso, obteniendo una experiencia educativa menos expositiva y más práctica



RETROALIMENTAR

Una vez que su tarea es evaluada, el estudiante obtiene una retroalimentación con la cual puede aprender mediante el entendimiento de sus errores

¿QUÉ SE DEBE CUMPLIR?



Aplicación o
problema **real**

¿QUÉ SE DEBE CUMPLIR?



Aplicación o
problema **real**



Ser
interesantes

¿QUÉ SE DEBE CUMPLIR?



Aplicación o
problema **real**



Ser
interesantes



Nivel de dificultad
adequado

**¿Qué ocurre cuando
no se cumple?**



DESINTERÉS

¿Qué ocurre cuando
no se cumple?



DESINTERÉS



FRUSTRACIÓN

¿Qué ocurre cuando no se cumple?



DESINTERÉS



FRUSTACIÓN

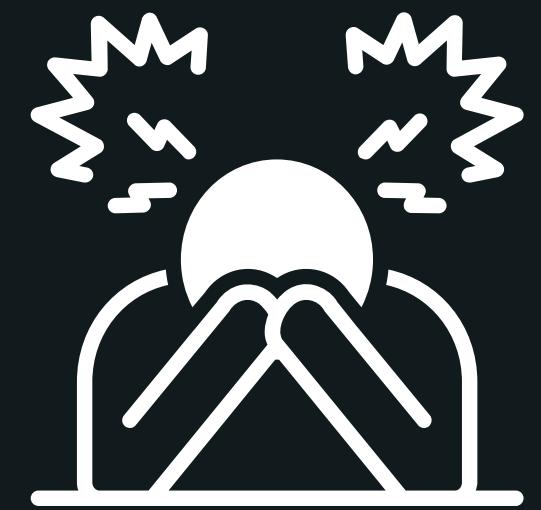


FAULTAS EN
EVALUACIONES

¿Qué ocurre cuando no se cumple?



DESINTERÉS



FRUSTRACIÓN

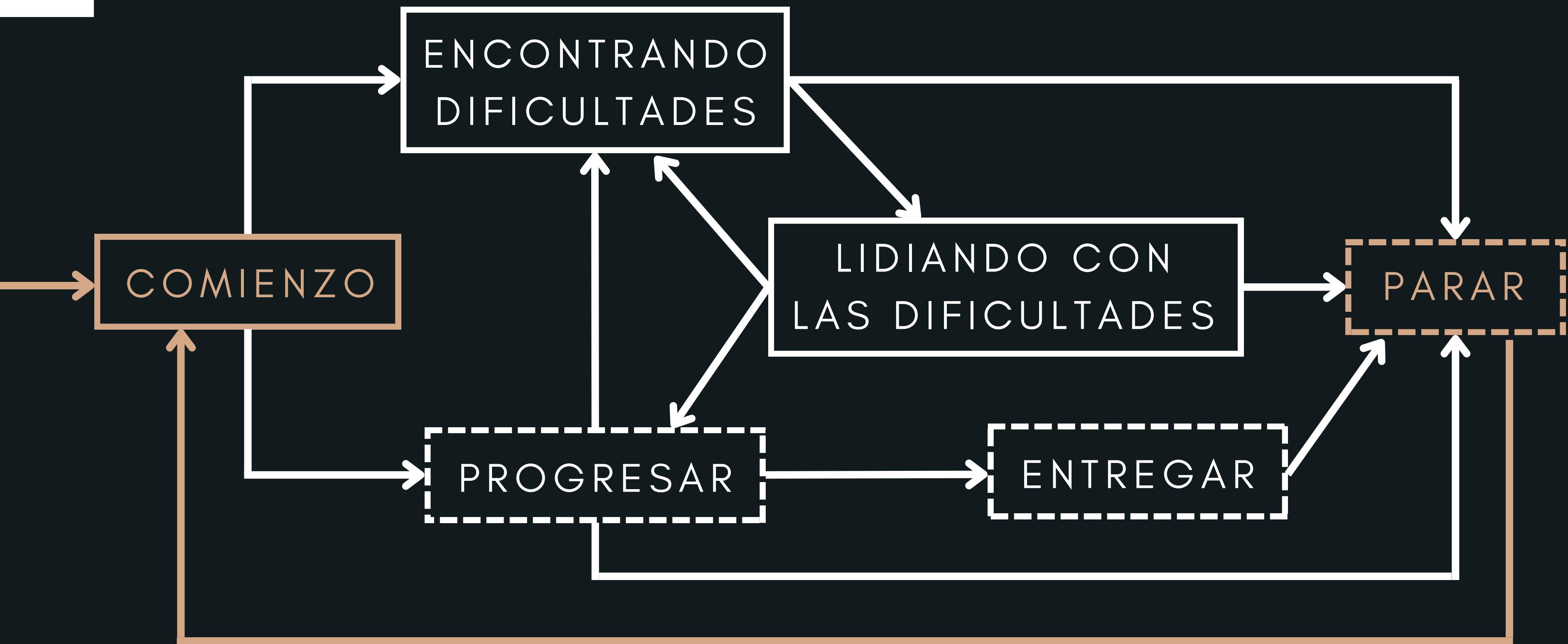


FAULTAS EN
EVALUACIONES



COMPLICACIÓN
EN LA RÚBRICA

Experiencia Emocional al desarrollar una Tarea



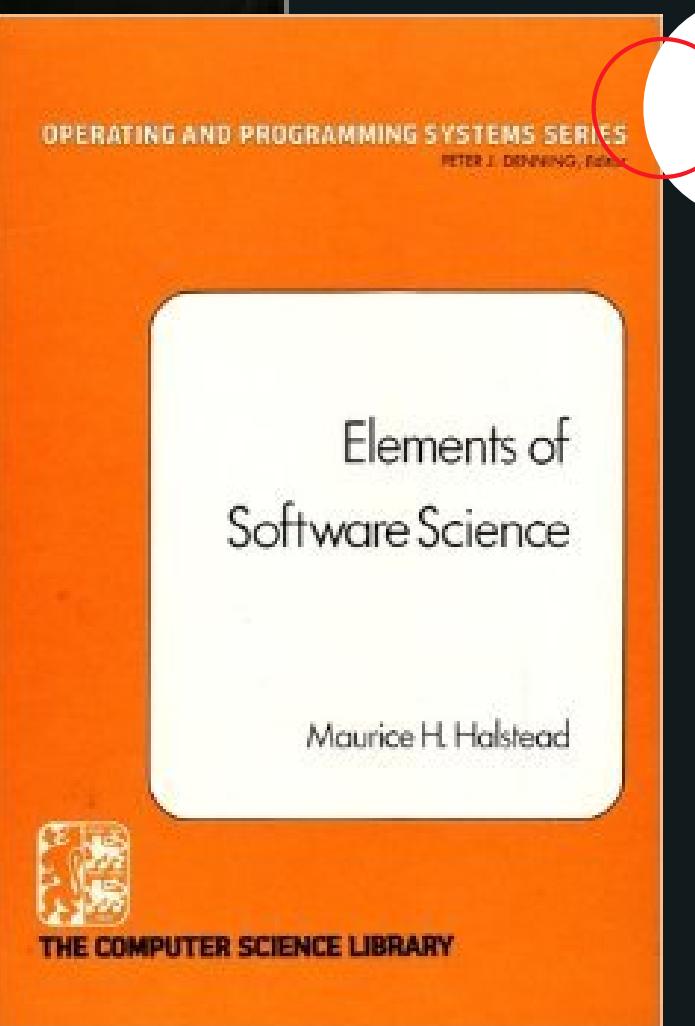
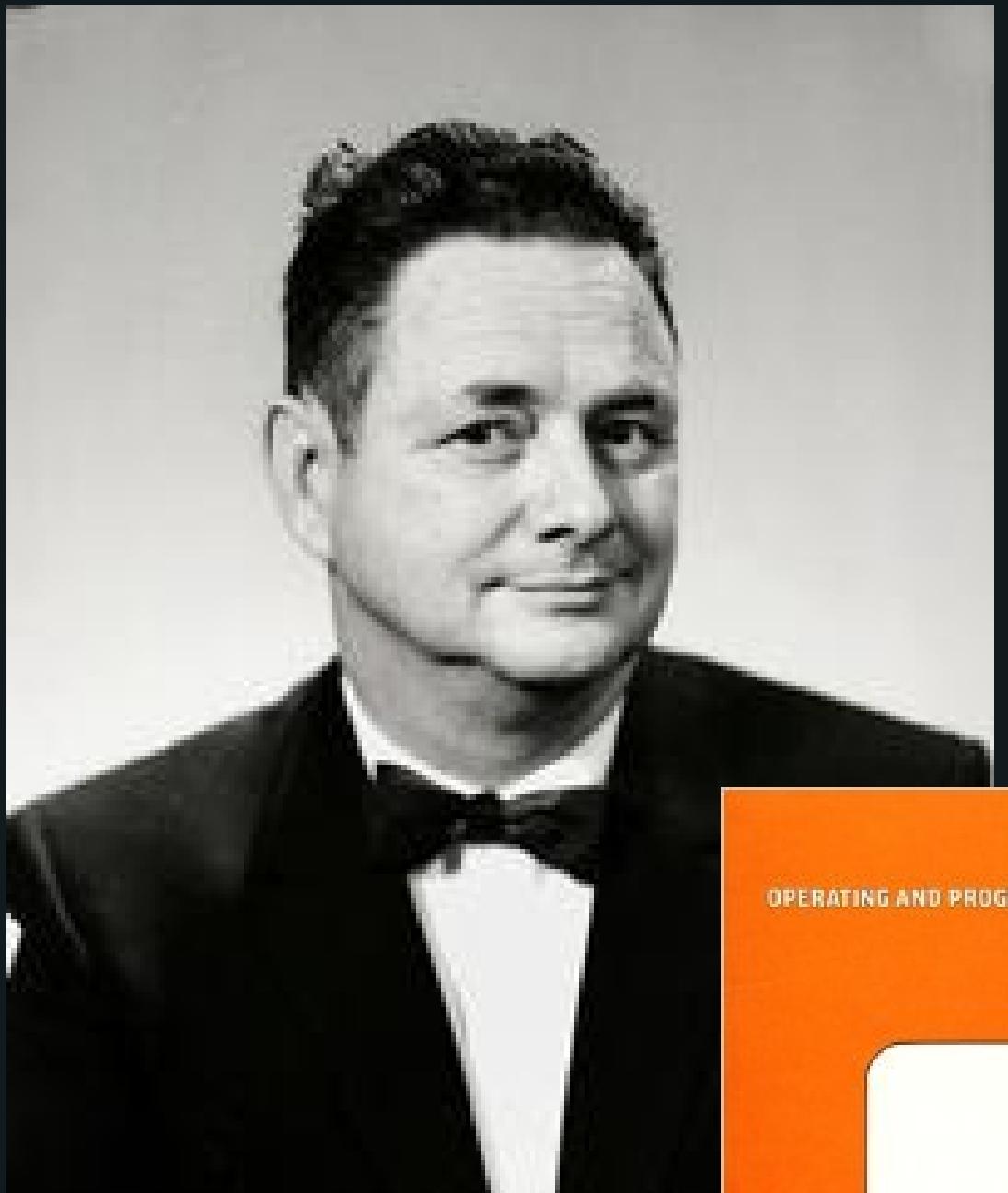


¡ES POR ESTO QUE ES IMPORTANTE
HACER TAREAS DE CALIDAD!



COMPLEJIDAD DE UNA TAREA

¿CÓMO PODEMOS MEDIR LA COMPLEJIDAD
DE UNA TAREA?



MAURICE H. HALSTEAD

PROFESOR E INVESTIGADOR

Profesor de la Universidad de Purdue y creador de los conceptos de la ciencia del software, que condujo al desarrollo de las métricas del software.

ELEMENTS OF SOFTWARE SCIENCE

Publicado en 1977, este libro contiene el primer resumen sistemático de una rama de la ciencia experimental y teórica que se ocupa del análisis y medición de la creación de programas computacionales.

MÉTRICAS DE HALSTEAD

HALSTEAD DEFINIÓ LAS SIGUIENTES MÉTRICAS PARA UN PROGRAMA:

- **Vocabulario:** cantidad de operadores y operandos diferentes utilizados.
- **Largo:** cantidad de operadores y operandos utilizados en total.
- **Volumen:** tamaño en bits necesario para almacenar un programa (es dependiente de la implementación).
- **Esfuerzo:** cantidad de operaciones mentales requeridas para desarrollar un programa.
- **Tiempo** requerido para desarrollar un programa.
- **Cantidad de bugs entregados:** estimación de cuantos errores puede tener una implementación.

VEAMOS UN EJEMPLO

```
def GCD(a, b):  
    if(b == 0):  
        return abs(a)  
    else:  
        return GCD(b, a % b)
```

TODO PROGRAMA SE COMPONE DE
UN CONJUNTO DE OPERADORES, Y UN
CONJUNTO DE OPERANDOS:

VEAMOS UN EJEMPLO

```
def GCD(a, b):  
    if(b == 0):  
        return abs(a)  
    else:  
        return GCD(b, a % b)
```

TODO PROGRAMA SE COMPONE DE UN CONJUNTO DE OPERADORES, Y UN CONJUNTO DE OPERANDOS:

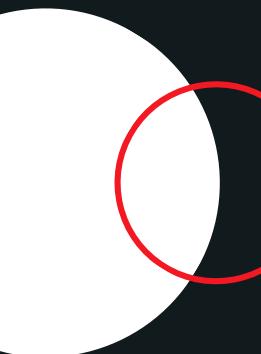
Conjunto de Operadores:

- Función GCD
- Función abs()
- Operador ==
- Operador %
- Operador return
- Operador if

Conjunto de Operandos:

- Variable a
- Variable b
- Número 0

¿QUÉ HACEMOS CUANDO PROGRAMAMOS?



Elegimos correctamente cada uno de los operadores y operandos, y los colocamos en nuestro código.

¿QUÉ HACEMOS CUANDO PROGRAMAMOS?

Elegimos correctamente cada uno de los operadores y operandos, y los colocamos en nuestro código.

Por lo que nuestro proceso mental se resume en el siguiente algoritmo:

```
while (el código no está listo) {  
    if(necesito un operador)  
        buscarOperador()  
  
    if(necesito un operando)  
        buscarOperando()  
}
```



o o

buscarOperador()

* useEffect()
% || abs() = +
&& ==

buscarOperando()

"Chile" 9
foo a "JSConf"
5 123 var



o o

buscarOperador()

* useEffect()
% || abs() = +
&& ==

buscarOperando()

"Chile" 9
foo a "JSConf"
5 123 var

foo = var + 5 + 9
foo = abs(foo)

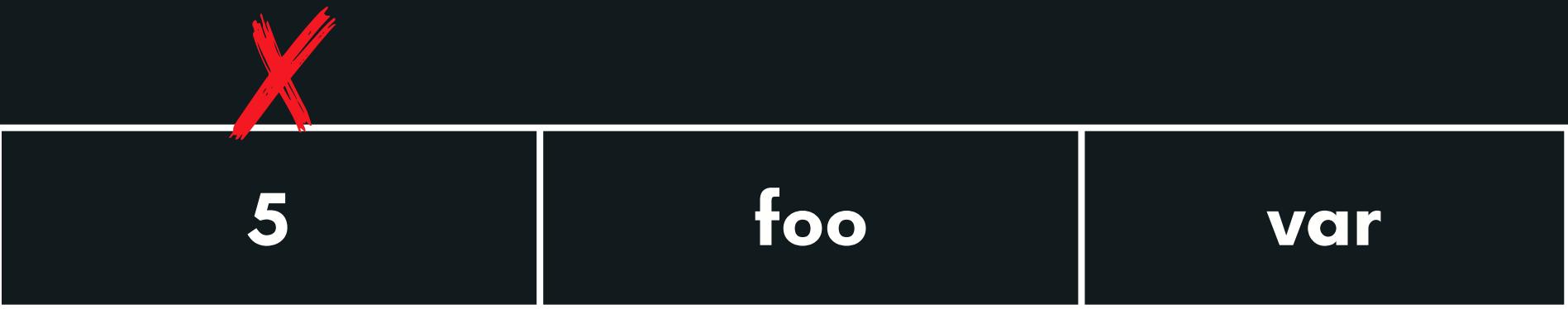
Proceso de Búsqueda

Primera Iteración



Proceso de Búsqueda

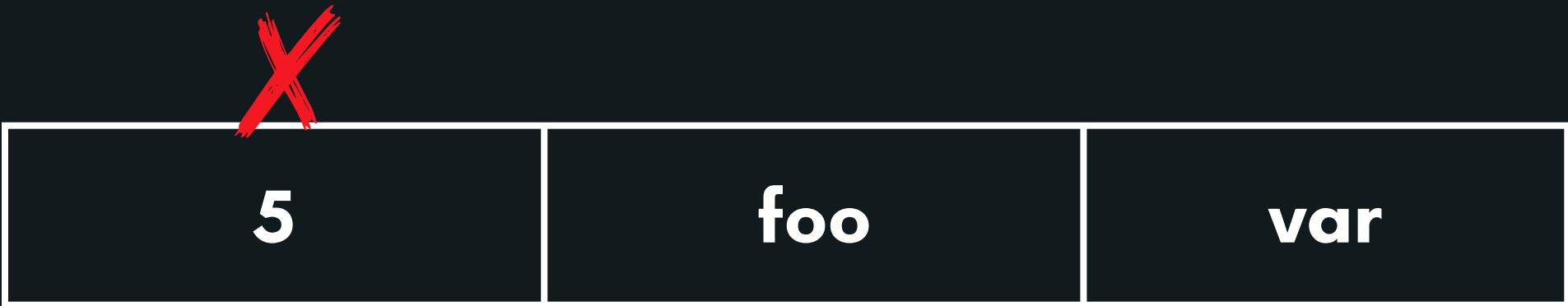
Primera Iteración



Segunda Iteración

Proceso de Búsqueda

Primera Iteración



Segunda Iteración

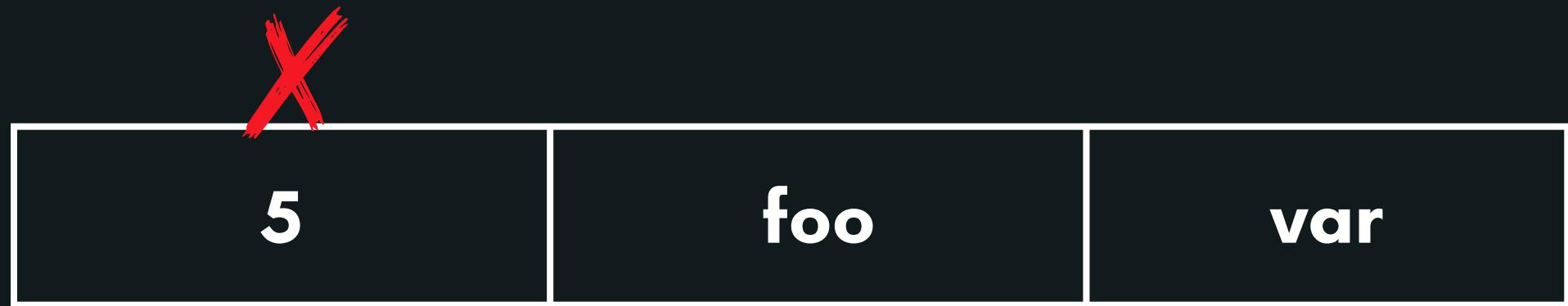


Tercera Iteración



Proceso de Búsqueda

Primera Iteración



Segunda Iteración



Tercera Iteración



Fallos = **2**

Operaciones Totales (OT) = **3**



OTRAS MÉTRICAS DE HALSTEAD

- η_1 = número de operadores distintos que aparecen en la implementación
- η_2 = número de operandos distintos que aparecen en la implementación
- N_1 = número total de usos de todos los operadores que aparecen en la implementación
- N_2 = número total de usos de todos los operandos que aparecen en la implementación
- $\eta = \eta_1 + \eta_2 \rightarrow$ vocabulario de la implementación.
- $N = N_1 + N_2 \rightarrow$ largo de la implementación.

VOLUMEN

$$V = N \log_2(\eta)$$

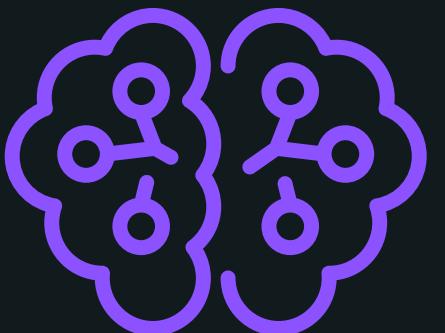
VOLUMEN POTENCIAL

$$V^* = (2 + \eta_2) \log_2 (2 + \eta_2)$$

ESFUERZO

$$E = V^2 / V^*$$

¿CÓMO CALCULAMOS EL TIEMPO APROXIMADO?



Sabemos que el cerebro realiza 18 discriminaciones por segundo.

Entonces, el tiempo que tardaríamos en realizar un código se calcula como:

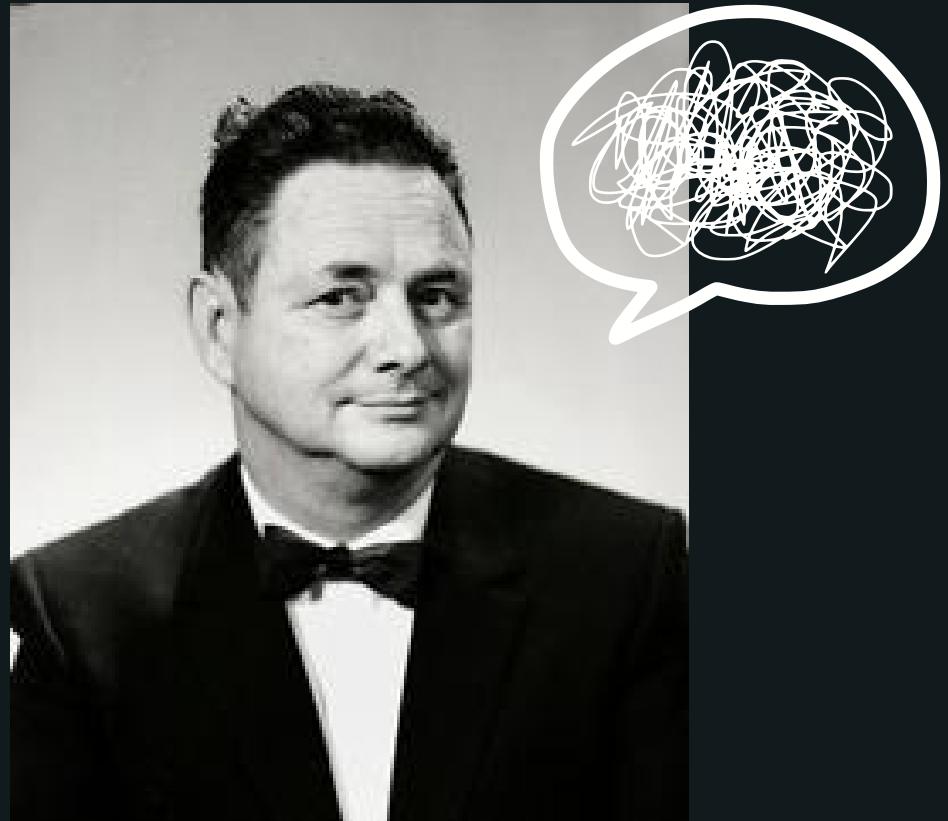
$$T [s] = E / 18$$



IMPLEMENTACIÓN

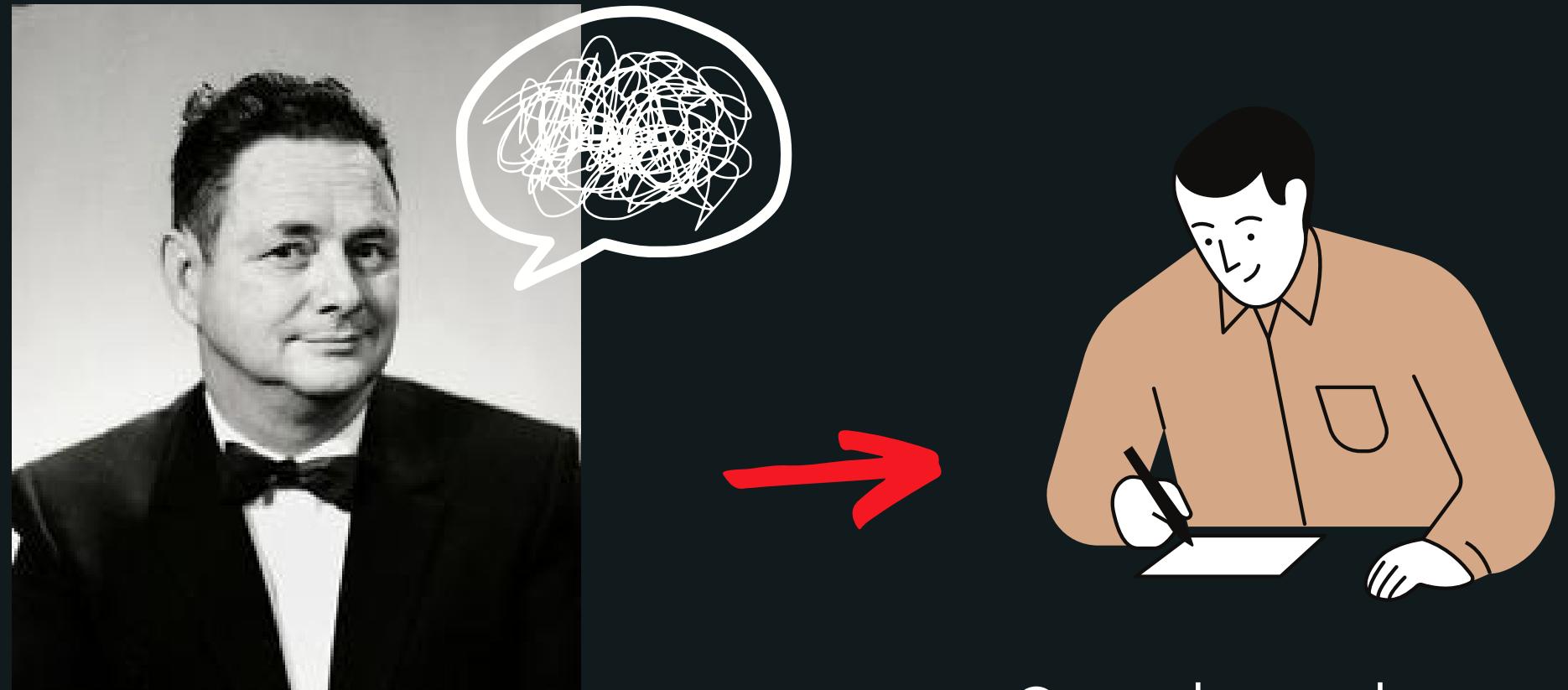
¿CÓMO APLICAMOS ESTAS MÉTRICAS?

Aplicación de las Métricas



Maurice Halstead y sus
métricas...

Aplicación de las Métricas



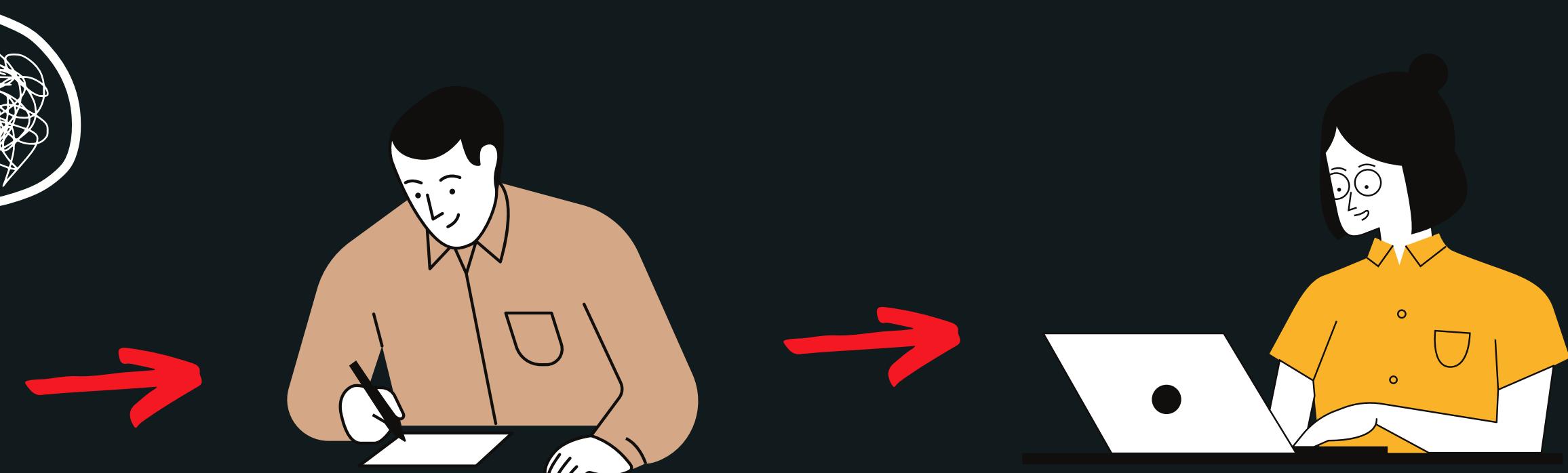
Maurice Halstead y sus
métricas...

Cuando un docente
redacte una tarea,
también la debe
desarrollar

Aplicación de las Métricas



Maurice Halstead y sus métricas...



Cuando un docente redacte una tarea, también la debe desarrollar

Luego, debe utilizar las métricas para estimar el esfuerzo que le llevaría al estudiante realizarla.



FRAMEWORK ISHVEL

¿COMO LLEVAR ESTAS MÉTRICAS A LA DOCENCIA?

Tarea N° 5: El Bazar de Pythonia

En Pythonia existe un bazar que suele vender distintos productos, pero que siempre tiene problemas de logística con sus proveedores, por eso, te ha pedido a ti que como programador, le ayudes a calcular la distancia a la que se encuentra de cada uno, para así darle prioridad a aquellos proveedores que se encuentran más lejos, a la hora de insistir con los horarios de llegada de sus pedidos.

Recuerda que la fórmula de distancia es:

```
...KaTeX
c = \pm\sqrt{a^2 + b^2}
...
```

Instrucciones

Desarrolla un programa que:

1. Solicite la posición del bazar en formato (x, y)
2. Solicite la cantidad de proveedores a analizar
3. Por cada proveedor, solicite su posición en formato (x, y)
4. Finalmente, muestre en pantalla la posición del proveedor más lejano

Ejemplos

...
Hola!, por favor ingresa los siguientes datos para continuar:

Posición del bazar (x, y): 123.4, 9.3

Cantidad de proveedores: 4

Posición proveedor 1: 1.0, 9.0

Posición proveedor 2: 2.0, 190.0

Posición proveedor 3: 999.9, 999.9

Posición proveedor 4: 125.5, 15.5

El proveedor más lejano se encuentra en 999.9, 999.9

Recomendaciones

* Recuerda utilizar sólo los contenidos vistos en clase

* El ejemplo es sólo explicativo, y no requiere que la salida de la tarea sea exactamente igual

Tarea N° 5: El Bazar de Pythonia

En Pythonia existe un bazar que suele vender distintos productos, pero que siempre tiene problemas de logística con sus proveedores, por eso, te ha pedido a ti que como programador, le ayudes a calcular la distancia a la que se encuentra de cada uno, para así darle prioridad a aquellos proveedores que se encuentran más lejos, a la hora de insistir con los horarios de llegada de sus pedidos.

Recuerda que la fórmula de distancia es:

$$c = \pm\sqrt{a^2 + b^2}$$

Instrucciones

Desarrolla un programa que:

1. Solicite la posición del bazar en formato (x, y)
2. Solicite la cantidad de proveedores a analizar
3. Por cada proveedor, solicite su posición en formato (x, y)
4. Finalmente, muestre en pantalla la posición del proveedor más lejano

Ejemplos

Hola!, por favor ingresa los siguientes datos para continuar:

```
Posición del bazar (x, y): 123.4, 9.3
Cantidad de proveedores: 4
Posición proveedor 1: 1.0, 9.0
Posición proveedor 2: 2.0, 190.0
Posición proveedor 3: 999.9, 999.9
Posición proveedor 4: 125.5, 15.5
```

El proveedor más lejano se encuentra en 999.9, 999.9

Recomendaciones

- Recuerda utilizar sólo los contenidos vistos en clase
- El ejemplo es sólo explicativo, y no requiere que la salida de la tarea sea exactamente igual

Resolver tarea

```
def bazar_mas_lejano(lista,operacion):
    _operacion=[]
    datos={}
    _final=[]
    for i in lista:
        if i[2] == operacion:
            if i[3] not in datos:
                datos[i[3]]=0
                datos[i[3]]+=1
```

Sugerencias

Esfuerzo de la tarea

El esfuerzo de programar esta tarea se encuentra 63773.42144276041 puntos por sobre la del semestre pasado, mientras que el promedio histórico es de 13312.099766951467. Se sugiere eliminar partes de la tarea que requieran añadir más sentencias en el código, es decir, que éste resulte más corto.

[CERRAR](#)

Dificultad de la Tarea

La dificultad de programar esta tarea se encuentra 19.720093137896978 puntos por sobre la del semestre pasado, mientras que el promedio histórico es de 18.434205082657556. Se sugiere modificar partes de la tarea donde se soliciten cálculos, de modo tal que se reduzca la cantidad de operadores únicos diferentes en el programa.

[CERRAR](#)

Tiempo de la Tarea

El tiempo estimado de programar esta tarea se encuentra 3542.9678579311335 segundos por sobre la del semestre pasado, mientras que el promedio histórico es de 747.4672076298035 segundos. Se sugiere eliminar partes de la tarea que requieran añadir más sentencias en el código, es decir, que éste resulte más corto.

[CERRAR](#)

Configuración

Contenido de la tarea

Ciclos

Semestre a comparar

2022-2

Tareas a comparar

Estudiantes

[VER MÉTRICAS HISTÓRICAS](#)

[DESCARGAR TAREA](#)

Métricas

Complejidad Ciclomática

14

[AYUDA](#)

Esfuerzo

90381

[AYUDA](#)

Dificultad

40

[AYUDA](#)

Tiempo

5021

[AYUDA](#)

Tarea N° 5: El Bazar de Pythonia

En Pythonia existe un bazar que suele vender distintos productos, pero que siempre tiene problemas de logística con sus proveedores, por eso, te ha pedido a ti que como programador, le ayudes a calcular la distancia a la que se encuentra de cada uno, para así darle prioridad a aquellos proveedores que se encuentran más lejos, a la hora de insistir con los horarios de llegada de sus pedidos.

Recuerda que la fórmula de distancia es:

...
KaTeX
 $c = \sqrt{a^2 + b^2}$
...

Instrucciones

Desarrolla un programa que:

1. Solicite la posición del bazar en formato (x, y)
2. Solicite la cantidad de proveedores a analizar
3. Por cada proveedor, solicite su posición en formato (x, y)
4. Finalmente, muestre en pantalla la posición del proveedor más lejano

Ejemplos

...

Hola!, por favor ingresa los siguientes datos para continuar:

Posición del bazar (x, y): 123.4, 9.3

Cantidad de proveedores: 4

Posición proveedor 1: 1.0, 9.0

Posición proveedor 2: 2.0, 190.0

Posición proveedor 3: 999.9, 999.9

Posición proveedor 4: 125.5, 15.5

El proveedor más lejano se encuentra en 999.9, 999.9

...

Recomendaciones

* Recuerda utilizar sólo los contenidos vistos en clase

* El ejemplo es sólo explicativo, y no requiere que la salida de la tarea sea exactamente igual

Tarea N° 5: El Bazar de Pythonia

En Pythonia existe un bazar que suele vender distintos productos, pero que siempre tiene problemas de logística con sus proveedores, por eso, te ha pedido a ti que como programador, le ayudes a calcular la distancia a la que se encuentra de cada uno, para así darle prioridad a aquellos proveedores que se encuentran más lejos, a la hora de insistir con los horarios de llegada de sus pedidos.

Recuerda que la fórmula de distancia es:

$$c = \sqrt{a^2 + b^2}$$

Instrucciones

Desarrolla un programa que:

1. Solicite la posición del bazar en formato (x, y)
2. Solicite la cantidad de proveedores a analizar
3. Por cada proveedor, solicite su posición en formato (x, y)
4. Finalmente, muestre en pantalla la posición del proveedor más lejano

Ejemplos

Hola!, por favor ingresa los siguientes datos para continuar:

Posición del bazar (x, y): 123.4, 9.3

Cantidad de proveedores: 4

Posición proveedor 1: 1.0, 9.0

Posición proveedor 2: 2.0, 190.0

Posición proveedor 3: 999.9, 999.9

Posición proveedor 4: 125.5, 15.5

El proveedor más lejano se encuentra en 999.9, 999.9

Recomendaciones

- Recuerda utilizar sólo los contenidos vistos en clase
- El ejemplo es sólo explicativo, y no requiere que la salida de la tarea sea exactamente igual

Resolver tarea

```
def bazar_mas_lejano(lista,operacion):
    _operacion=[]
    datos={}
    _final=[]
    for i in lista:
        if i[2] == operacion:
            if i[3] not in datos:
                datos[i[3]]=0
                datos[i[3]]+=1
```

Sugerencias

Esfuerzo de la tarea

El esfuerzo de programar esta tarea se encuentra 63773.42144276041 puntos por sobre la del semestre pasado, mientras que el promedio histórico es de 13312.099766951467. Se sugiere eliminar partes de la tarea que requieran añadir más sentencias en el código, es decir, que éste resulte más corto.

CERRAR

Dificultad de la Tarea

La dificultad de programar esta tarea se encuentra 19.720093137896978 puntos por sobre la del semestre pasado, mientras que el promedio histórico es de 18.434205082657556. Se sugiere modificar partes de la tarea donde se soliciten cálculos, de modo tal que se reduzca la cantidad de operadores únicos diferentes en el programa.

CERRAR

Tiempo de la Tarea

El tiempo estimado de programar esta tarea se encuentra 3542.9678579311335 segundos por sobre la del semestre pasado, mientras que el promedio histórico es de 747.4672076298035 segundos. Se sugiere eliminar partes de la tarea que requieran añadir más sentencias en el código, es decir, que éste resulte más corto.

CERRAR

Configuración

Contenido de la tarea

Ciclos

Semestre a comparar

2022-2

Tareas a comparar

Estudiantes

VER MÉTRICAS HISTÓRICAS

DESCARGAR TAREA

Métricas

Complejidad Ciclomática

14

AYUDA

Esfuerzo

90381

AYUDA

Dificultad

40

AYUDA

Tiempo

5021

AYUDA

Configuración

Contenido de la tarea

Archivos



Semestre a comparar

2022-2



Tareas a comparar

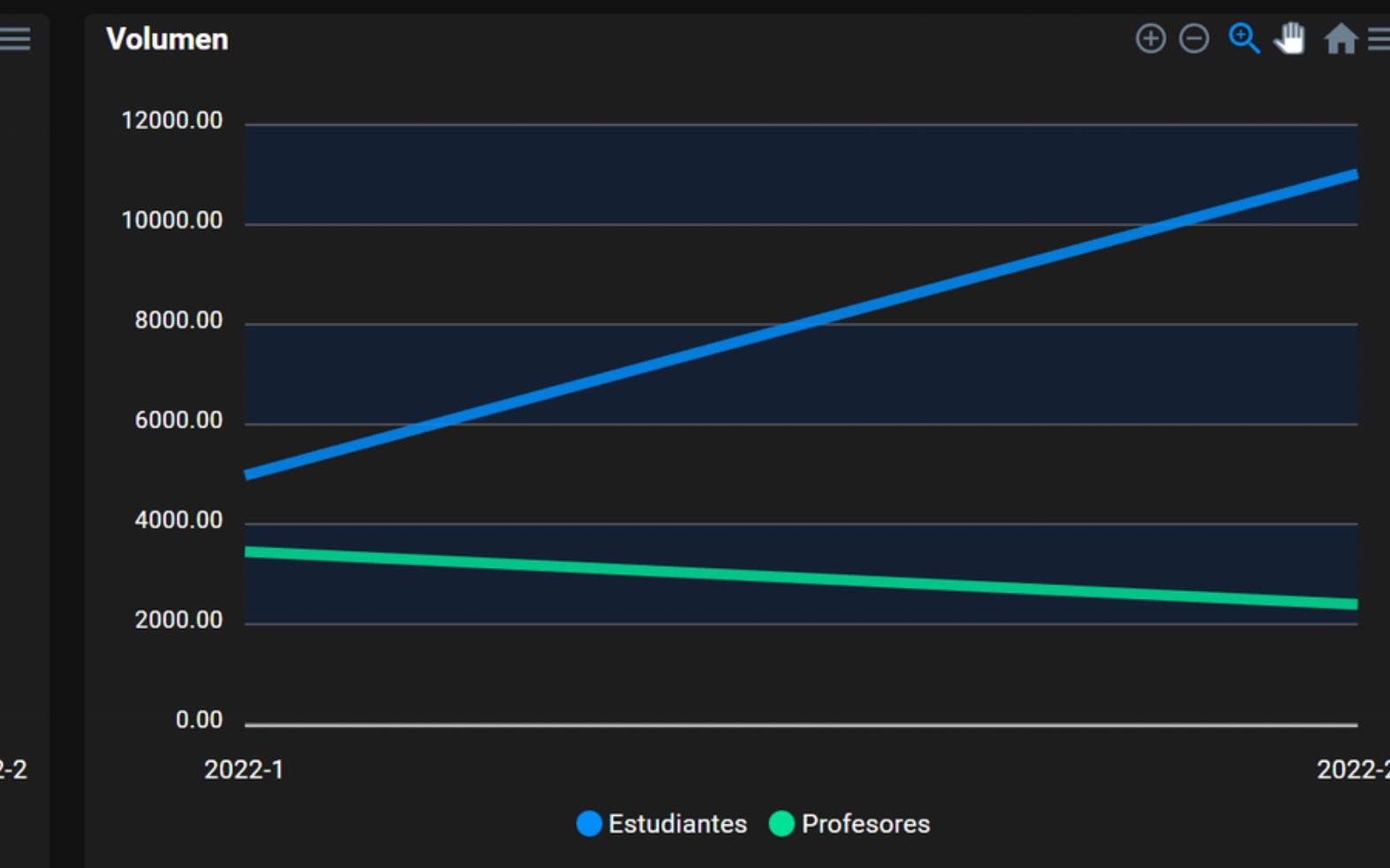
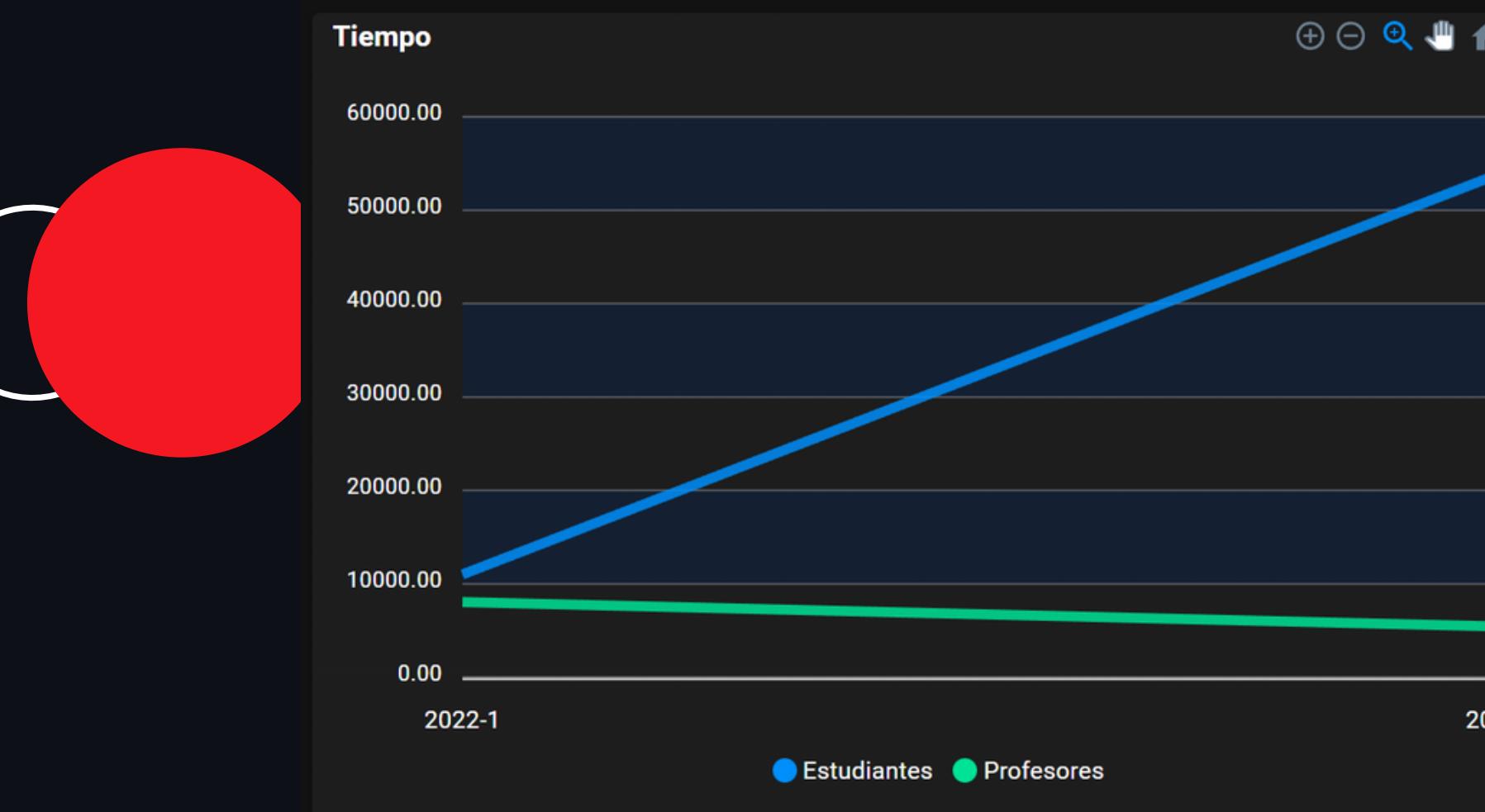
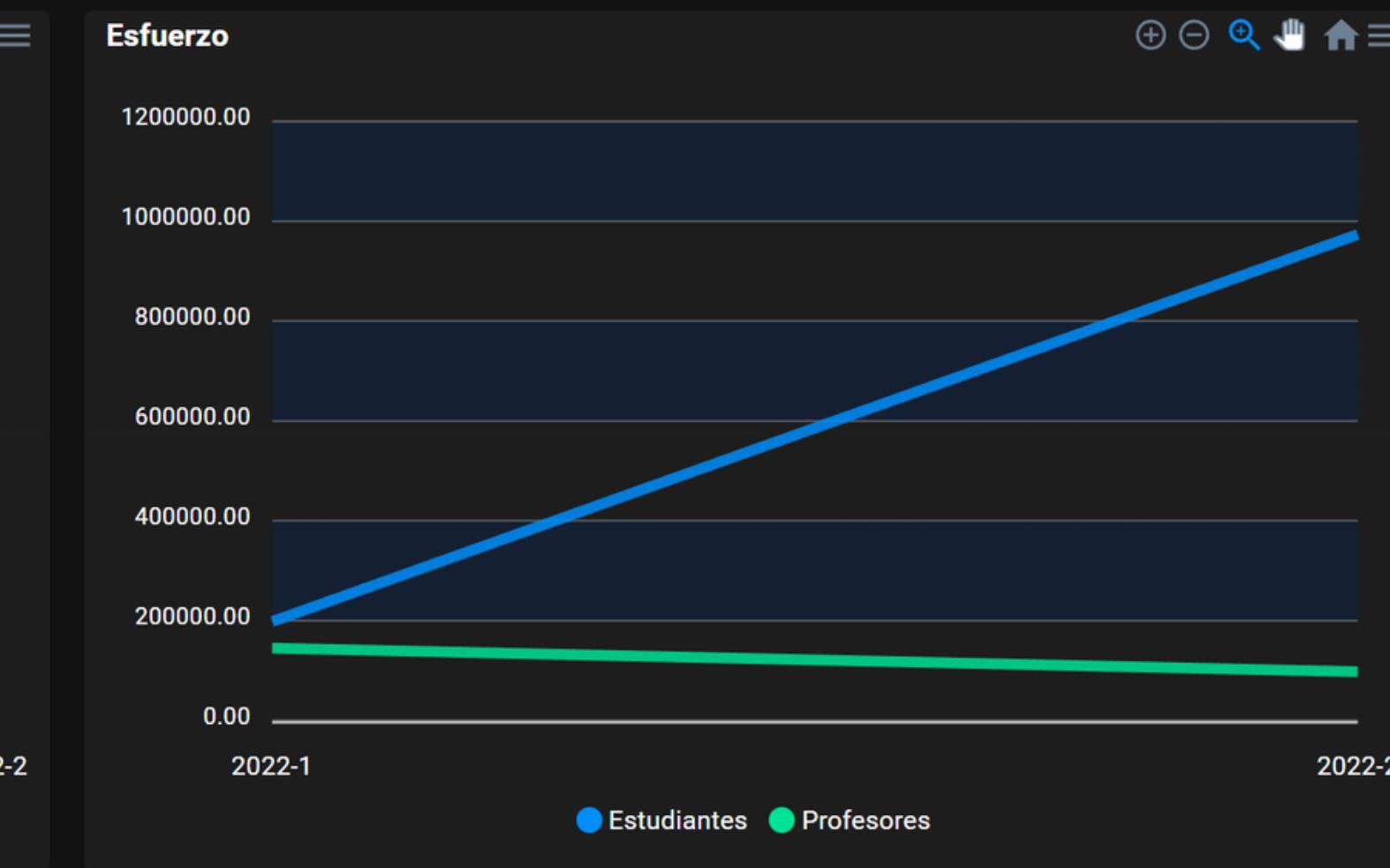
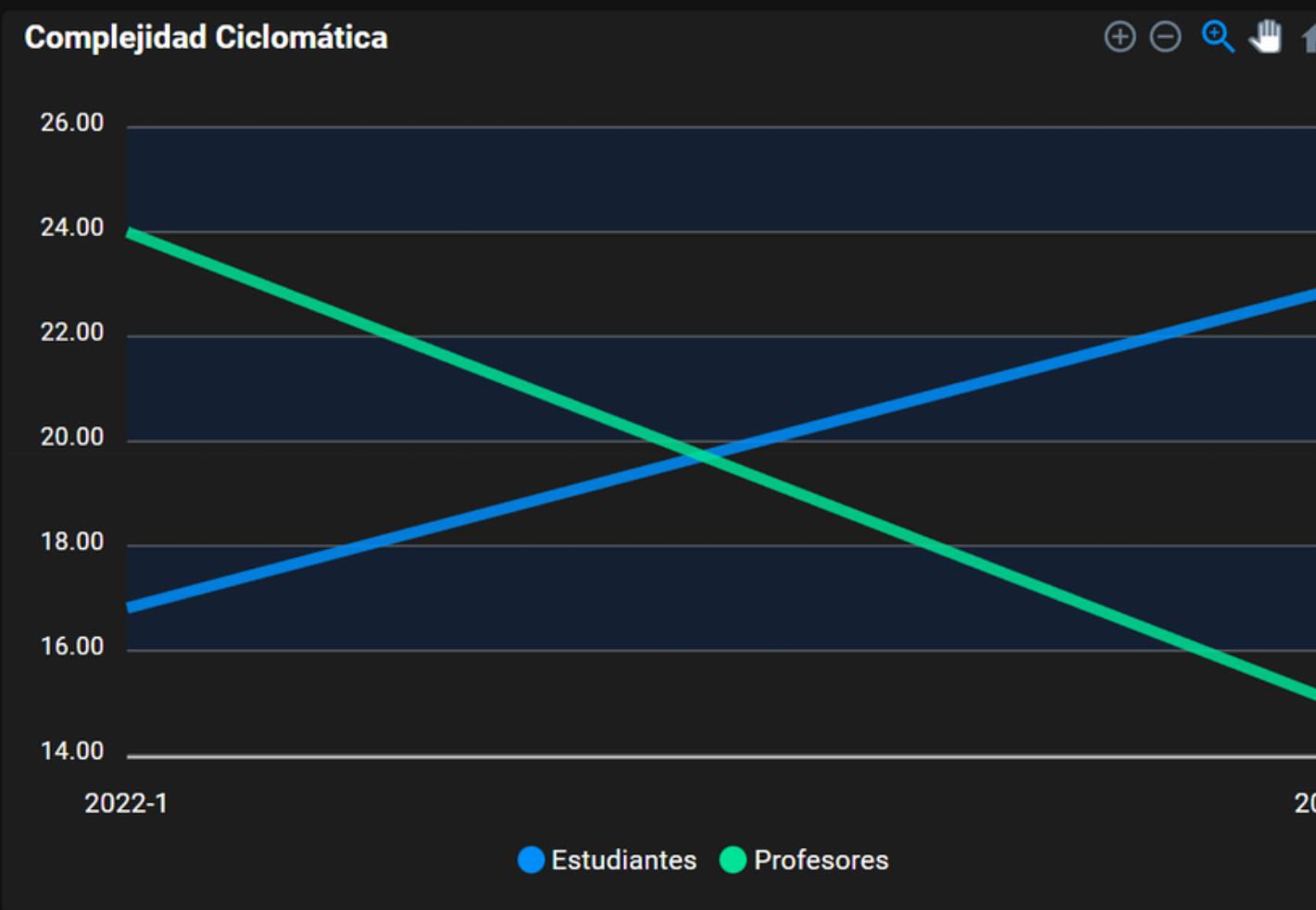
Estudiantes



[VOLVER AL EDITOR](#)

[DESCARGAR TAREA](#)

Métricas históricas de estudiantes y profesores



PROBLEMAS



Es difícil mantener un proyecto universitario en línea cuando nadie le da soporte, por lo tanto, había que encontrar una forma de que las funcionalidades de la aplicación, no dependan de algún servidor privado externo.

PROBLEMAS

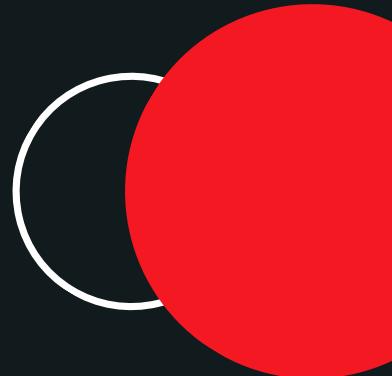


Es difícil mantener un proyecto universitario en línea cuando nadie le da soporte, por lo tanto, había que encontrar una forma de que las funcionalidades de la aplicación, no dependan de algún servidor privado externo.



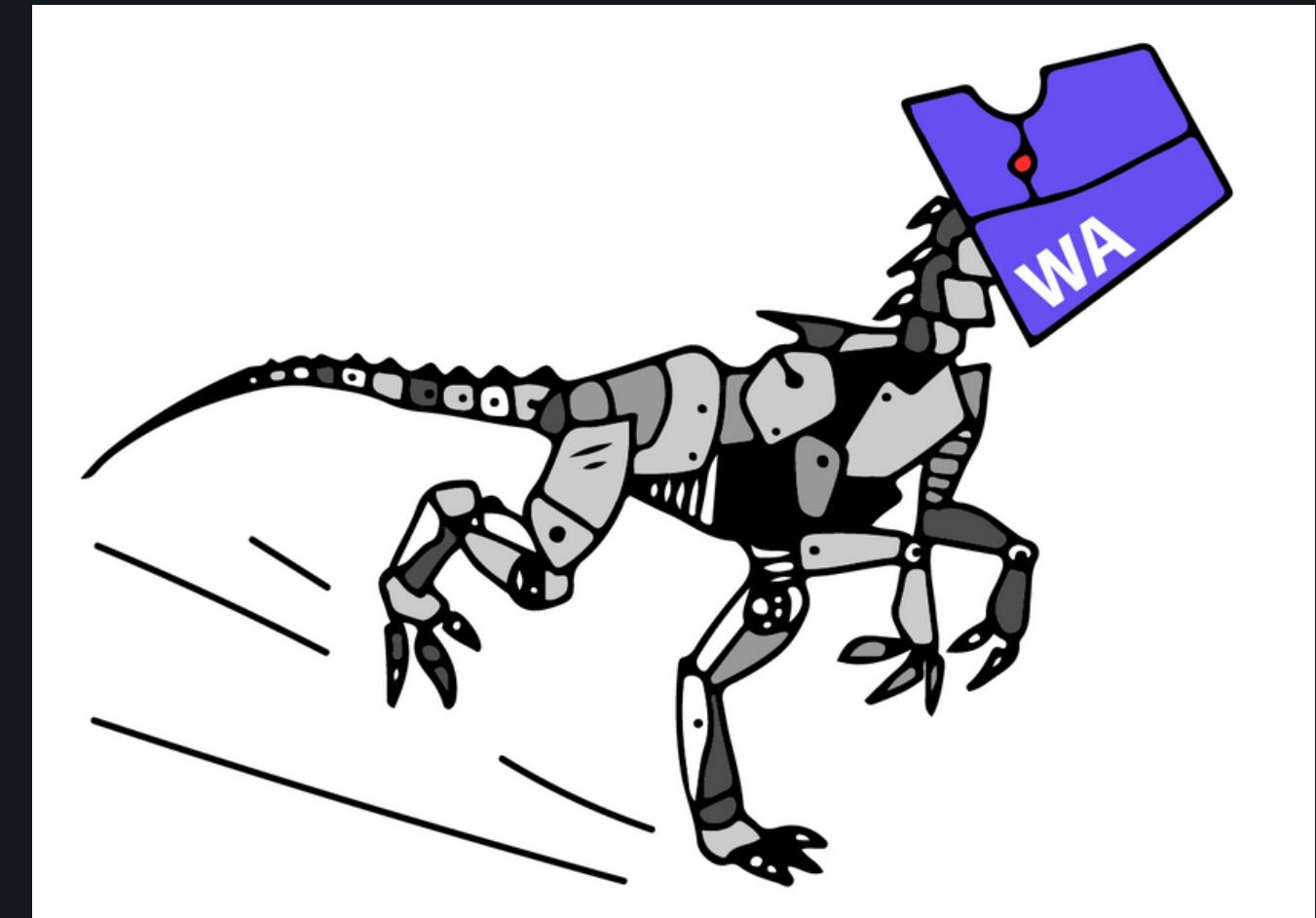
El módulo que ya implementa el cálculo de métricas de Halstead para Python, Multimetric, está pues... escrito en Python

WebAssembly al Rescate



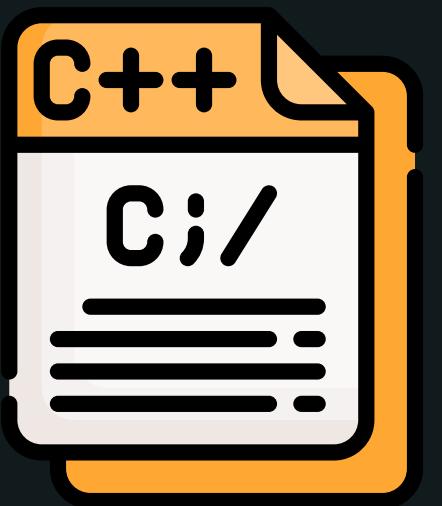
¿QUÉ ES?

WebAssembly es un formato de instrucciones binarias creado con el fin de ser ejecutado en una máquina virtual, diseñado para ser un punto de compilación de distintos lenguajes de programación, permitiéndonos desplegar el resultado en la web gracias a la WebAssembly API de nuestros navegadores.



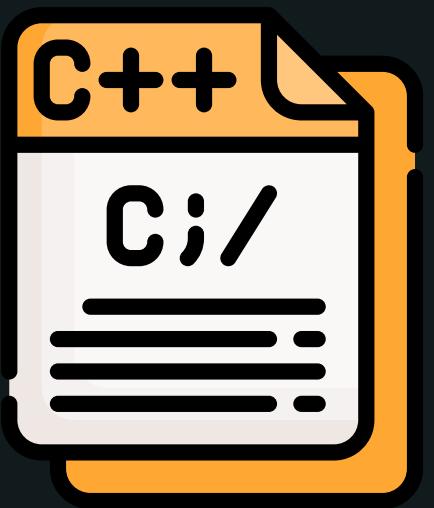
"WTF is WebAssembly?" - Kevin Hoffman

¿Como funciona?

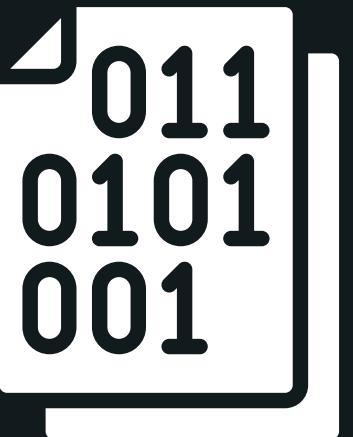


Código en C++

¿Como funciona?



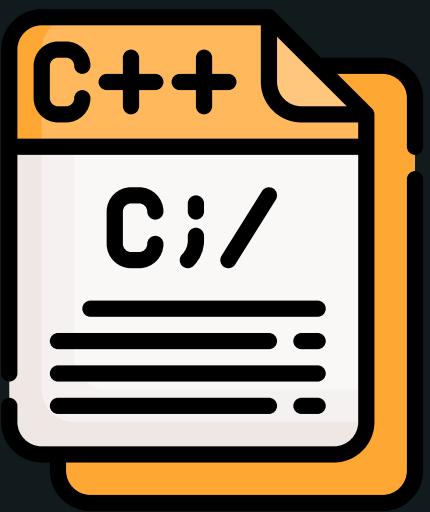
Código en C++



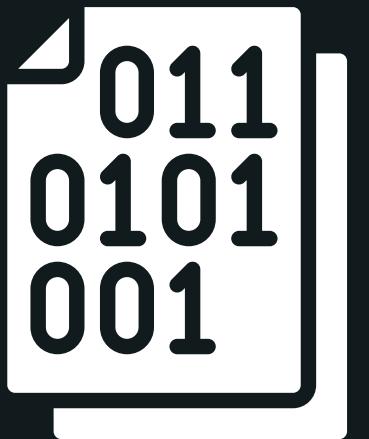
Binario

(ejecutable por nuestro SO)

¿Como funciona?



Compilador
→



Código en C++

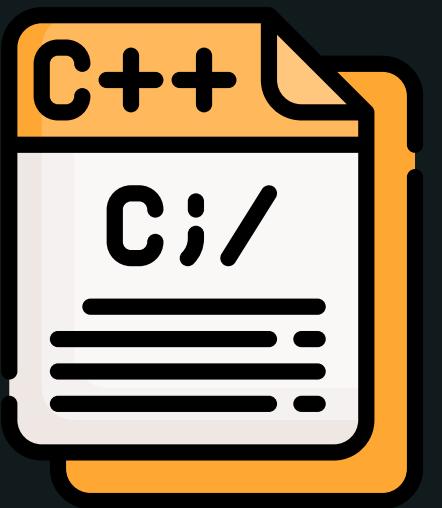
Binario
(ejecutable por nuestro SO)

→



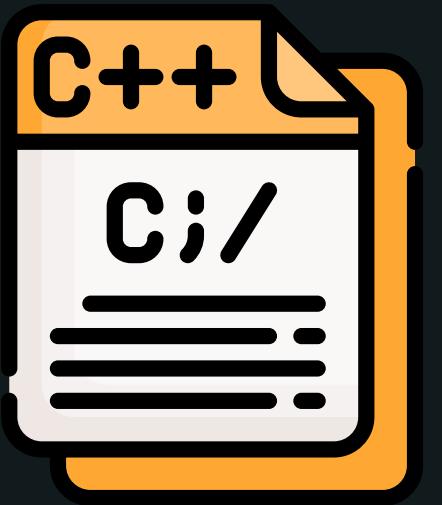
No ejecutable **por nuestro
navegador**

¿Como funciona?



Código en C++

¿Como funciona?

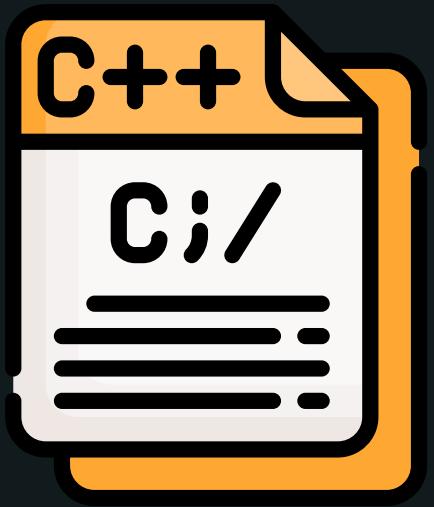


Código en C++

Compilador a
WebAssembly

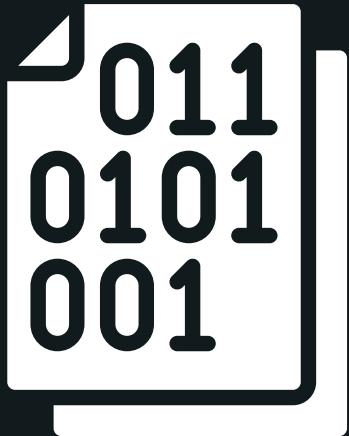


¿Como funciona?



Código en C++

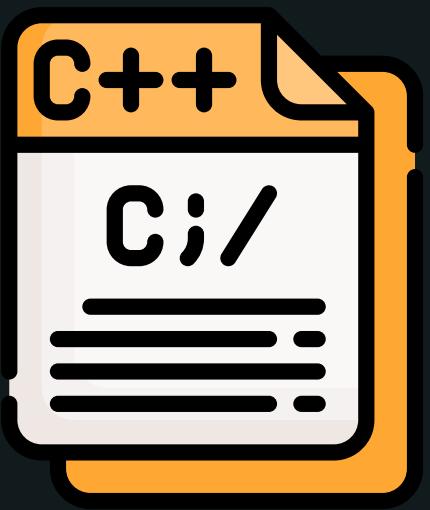
Compilador a
WebAssembly



Binario

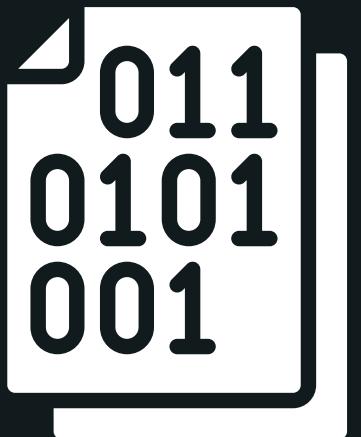
(ejecutable en **WebAssembly**)

¿Como funciona?



Código en C++

Compilador a
WebAssembly



Binario
(ejecutable en **WebAssembly**)



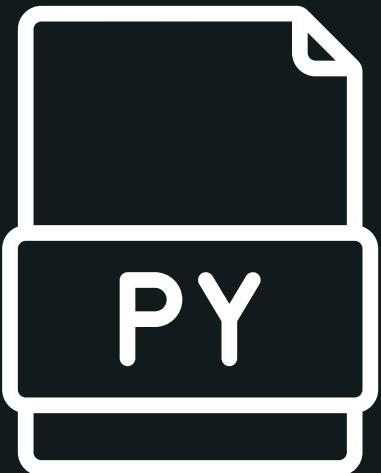
Navegador **ejecutando**
WebAssembly



El motor de análisis de juego de Lichess, está basado en un port a WebAssembly del motor Stockfish.

¿Cómo ejecutamos Python?

¿Cómo ejecutamos Python?



Código en Python

¿Cómo ejecutamos Python?

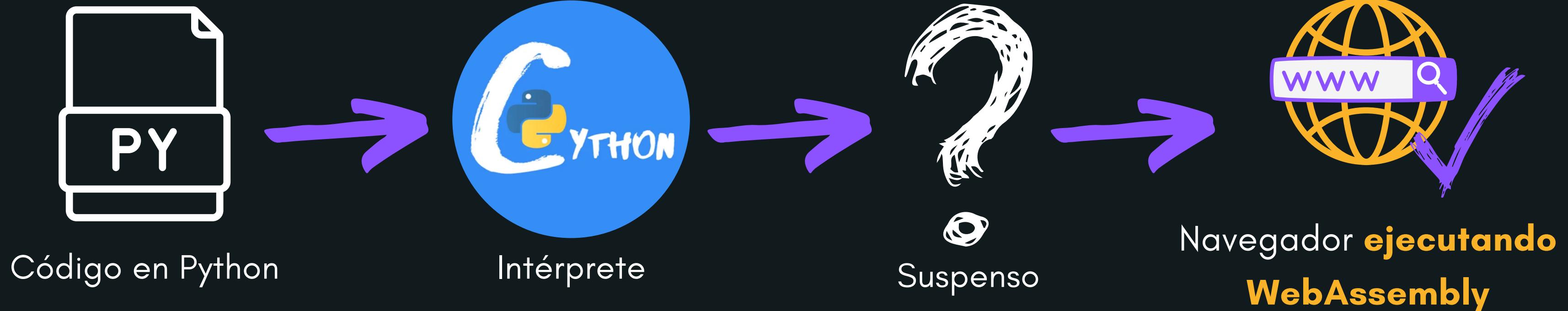


Código en Python

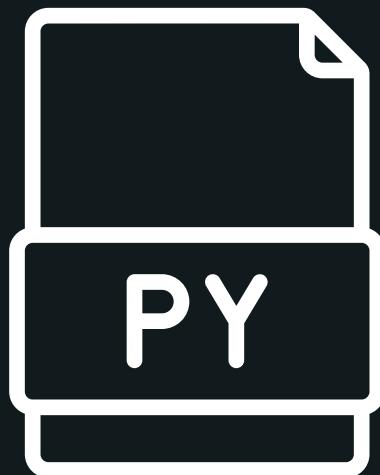
¿Cómo ejecutamos Python?



¿Cómo ejecutamos Python?



¿Cómo ejecutamos Python?



Código en Python



Port de CPython a
WebAssembly



Navegador **ejecutando**
WebAssembly

Implementación

```
async function main() {
    let pyodide = await loadPyodide();
    // Pyodide is now ready to use...
    console.log(pyodide.runPython(```
        import sys
        sys.version
    ``));
}
main();
```

Implementación

The screenshot shows the Libraries.io project page for the package **multimetric 1.3.0**. The top navigation bar includes a search bar, a magnifying glass icon, and links for Ayuda, Patrocinadores, Acceder, and Registrarse. Below the header, there's a green button with a checkmark labeled "Versión más reciente". The main content area features the package name **multimetric 1.3.0**, a command `pip install multimetric`, and a publication date of **7 oct 2021**. A brief description states: "Calculate code metrics in various languages". On the left, a sidebar titled "Navegación" contains links for "Descripción de proyecto" (which is highlighted in blue), "Histórico de versiones", and "Archivos de descarga". Below this is a section for "Enlaces del proyecto" with a "Homepage" link. The right side has a "Descripción de proyecto" section with badges for build status (passing), PyPI package (1.3.0), Python versions (3.5 | 3.6 | 3.7 | 3.8 | 3.9 | 3.10), and downloads (63/month). It also shows an "Igtm grade" badge indicating "no longer available". The "Purpose" section lists the tool's functions: Comment to Code percentage, Cyclomatic complexity according to McCabe, Difficulty according to Halstead, Effort according to Halstead, Fan-Out, Lines of code, Maintainability index, Metric according to pylint, Metric according to TIOBE, Number of delivered bugs according to Halstead, and Time required to program according to Halstead.

multimetric 1.3.0

pip install multimetric

Publicación: 7 oct 2021

Calculate code metrics in various languages

Navegación

- Descripción de proyecto
- Histórico de versiones
- Archivos de descarga

Enlaces del proyecto

- Homepage

Estadísticas

Estadísticas de GitHub:

- Estrellas: 20
- Bifurcaciones: 7
- Informes y solicitudes de incorporación abiertos: 6

Consulte estadísticas de este proyecto en [Libraries.io](#) o a través

Descripción de proyecto

Build passing pypi package 1.3.0 python 3.5 | 3.6 | 3.7 | 3.8 | 3.9 | 3.10 downloads 63/month

Igtm grade no longer available

Calculate code metrics in various languages

Purpose

This tool tries to calculate the following metrics for many, many programming languages

- Comment to Code percentage
- Cyclomatic complexity according to McCabe
- Difficulty according to Halstead
- Effort according to Halstead
- Fan-Out
- Lines of code
- Maintainability index
- Metric according to pylint
- Metric according to TIOBE
- Number of delivered bugs according to Halstead
- Time required to program according to Halstead

Implementación

```
await pyodide.loadPackage('micropip');
const micropip = pyodide.pyimport('micropip');

await micropip.install('multimetricprog-VadokDev');
```

Implementación

```
const init = async () => {
    pyodide = await window.loadPyodide({
        indexURL: 'https://cdn.jsdelivr.net/pyodide/v0.21.3/full/',
        stdout: () => {},
    });

    await pyodide.loadPackage('micropip');
    const micropip = pyodide.pyimport('micropip');

    await micropip.install('multimetricprog-VadokDev');
    pyodide.runPython(`  

        import json  

        from multimetricprog import calculator  

    `);
    isReady = true;
};
```

Implementación

<> Resolver tarea

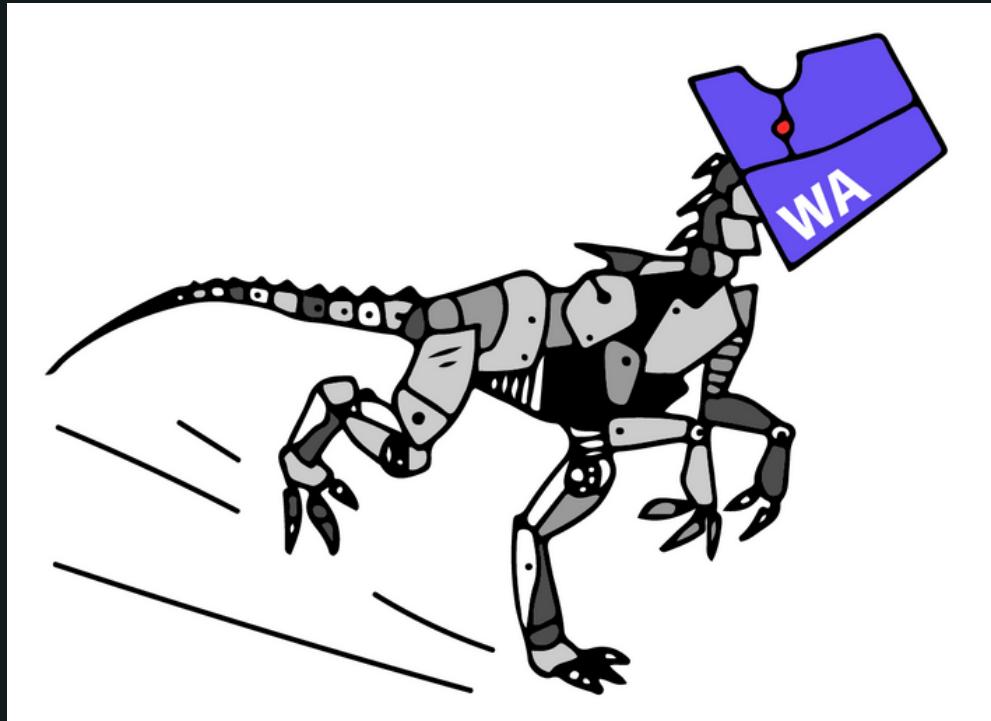
```
def bazar_mas_lejano(lista,operacion):
    _operacion=[]
    datos={}
    _final=[]
    for i in lista:
        if i[2] == operacion:
            if i[3] not in datos:
                datos[i[3]]=0
                datos[i[3]]+=1
```

```
const raw = pyodide.runPython(`  
    metrics = calculator.calculate(``  
        ${JSON.parse(JSON.stringify(code))}``  
    )  
    json.dumps(metrics, indent = 2)  
`);
```

Implementación



<https://vadokdev.github.io/Ishvel/>



CONCLUSIONES

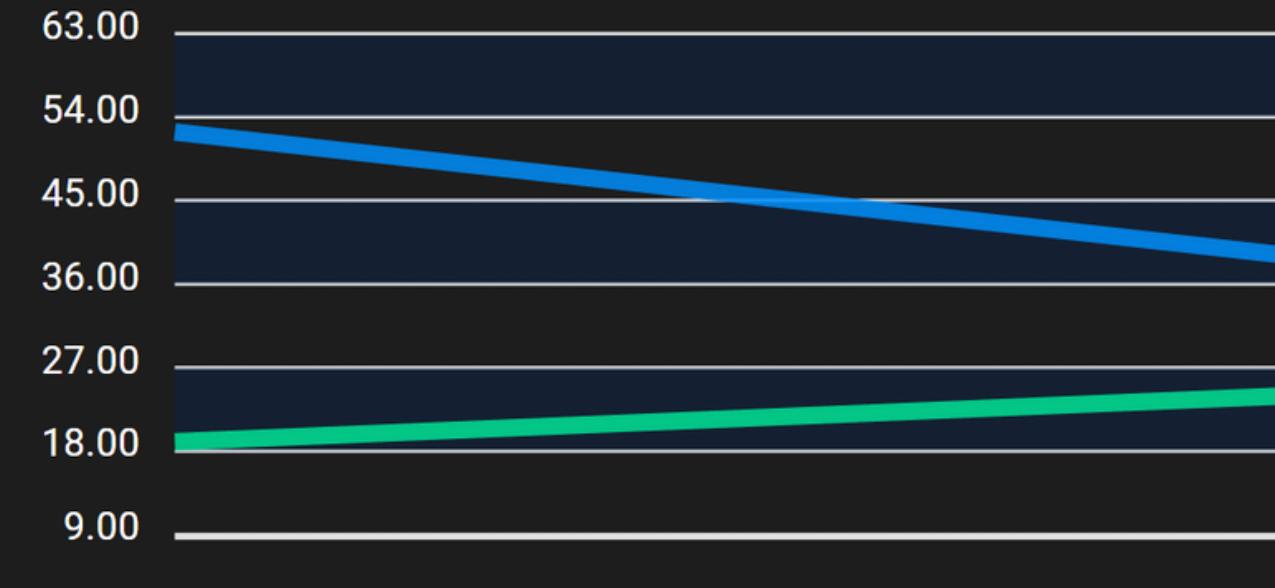
DATOS INTERESANTES Y TRABAJO A FUTURO

Name	Status	Type	Initiator	Size	Time
□ pyodide_py.tar	200	fetch	compat.ts:153	124 kB	40 ms
□ repodata.json	200	fetch	pyodide.asm.j...	13.7 kB	21 ms
□ distutils.tar	200	fetch	pyodide.asm.j...	984 kB	26 ms
□ micropip-0.1-py3-none-any.whl	200	fetch	pyodide.asm.j...	21.0 kB	17 ms
□ pyparsing-3.0.9-py3-none-any.whl	200	fetch	pyodide.asm.j...	99.0 kB	14 ms
□ packaging-21.3-py3-none-any.whl	200	fetch	pyodide.asm.j...	41.4 kB	14 ms
□ json	200	fetch	pyodide.asm.j...	4.4 kB	19 ms
□ multimetricprog_VadokDev-0.0.1-py3-non...	200	fetch	pyodide.asm.j...	23.9 kB	162 ms
□ json	200	fetch	pyodide.asm.j...	6.1 kB	6 ms
□ chardet-5.1.0-py3-none-any.whl	200	fetch	pyodide.asm.j...	199 kB	16 ms
□ Pygments-2.12.0-py3-none-any.whl	200	fetch	pyodide.asm.j...	1.1 MB	25 ms

Métricas históricas de estudiantes y profesores (Condicionales)

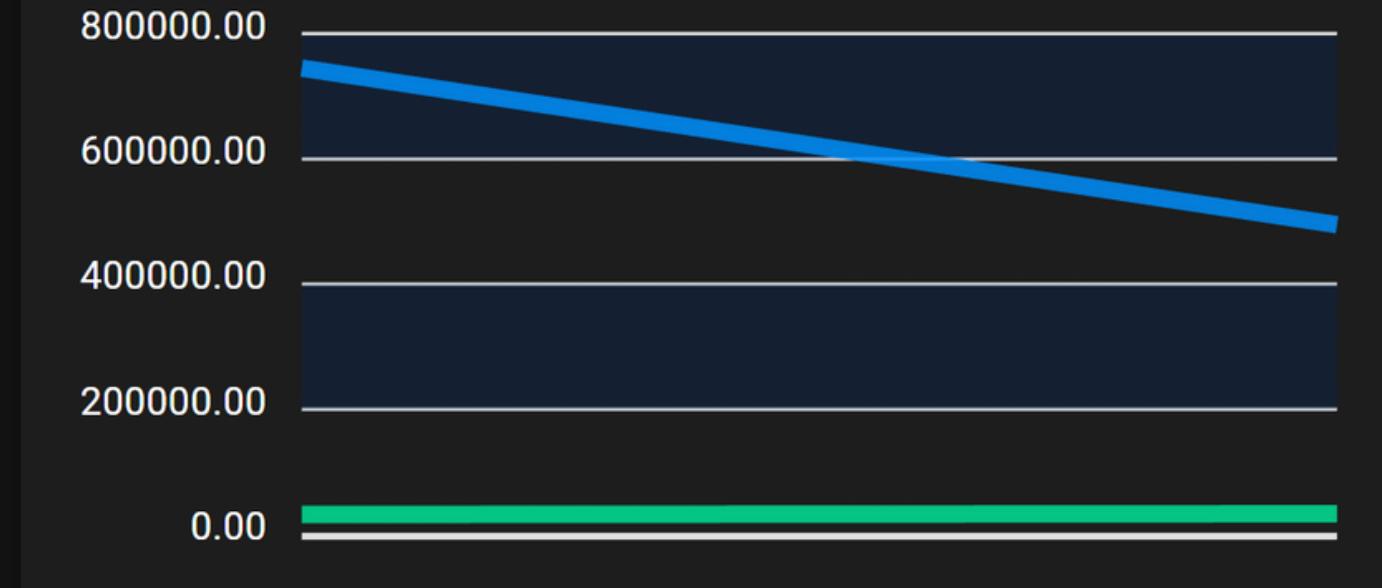
Complejidad Ciclomática

+ - 🔎 🖐️ ⬆️ ⬇️



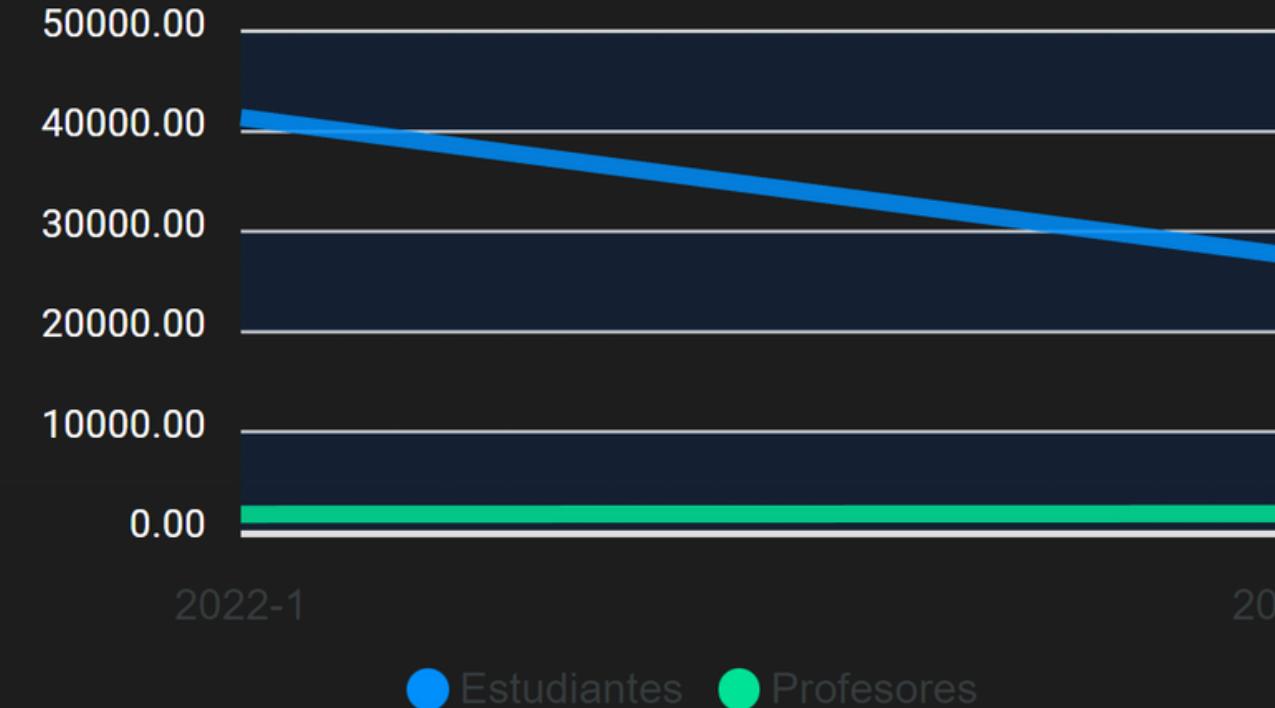
Esfuerzo

+ - 🔎 🖐️ ⬆️ ⬇️



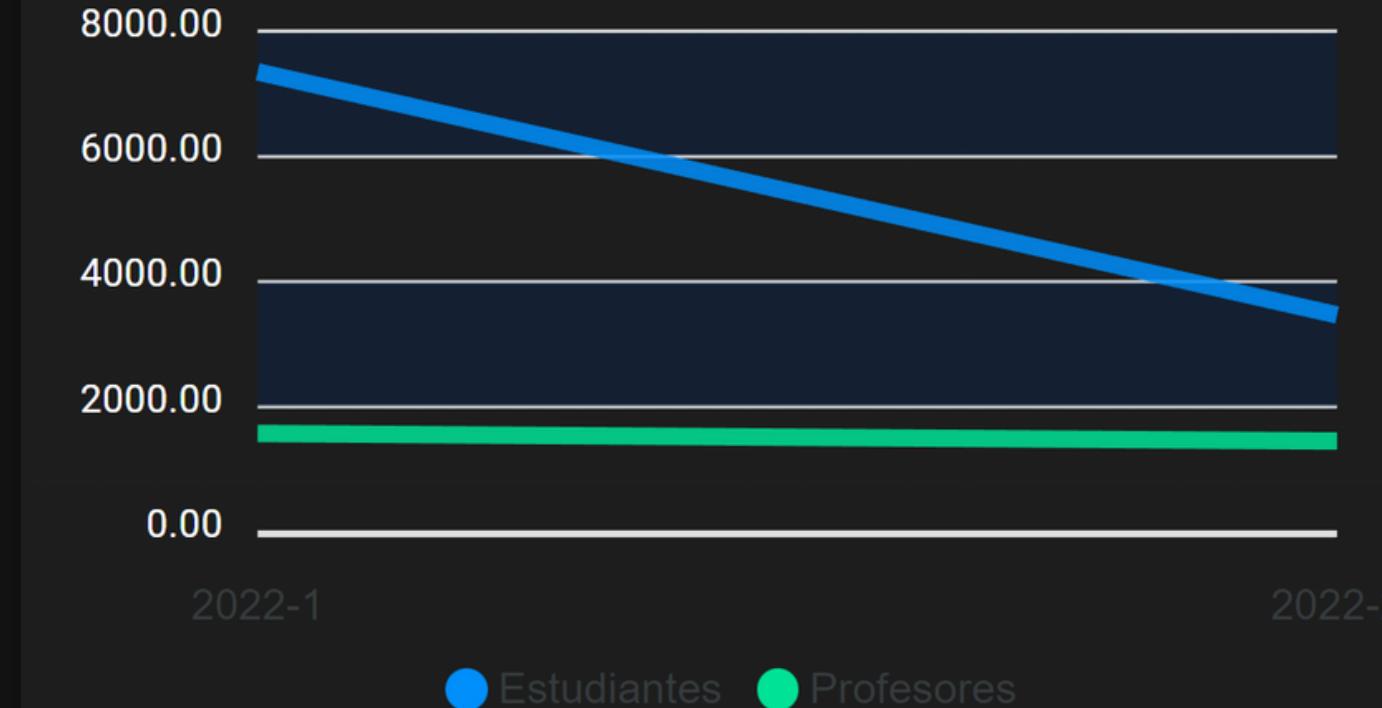
Tiempo

+ - 🔎 🖐️ ⬆️ ⬇️



Volumen

+ - 🔎 🖐️ ⬆️ ⬇️



Métricas históricas de estudiantes y profesores (Funciones)

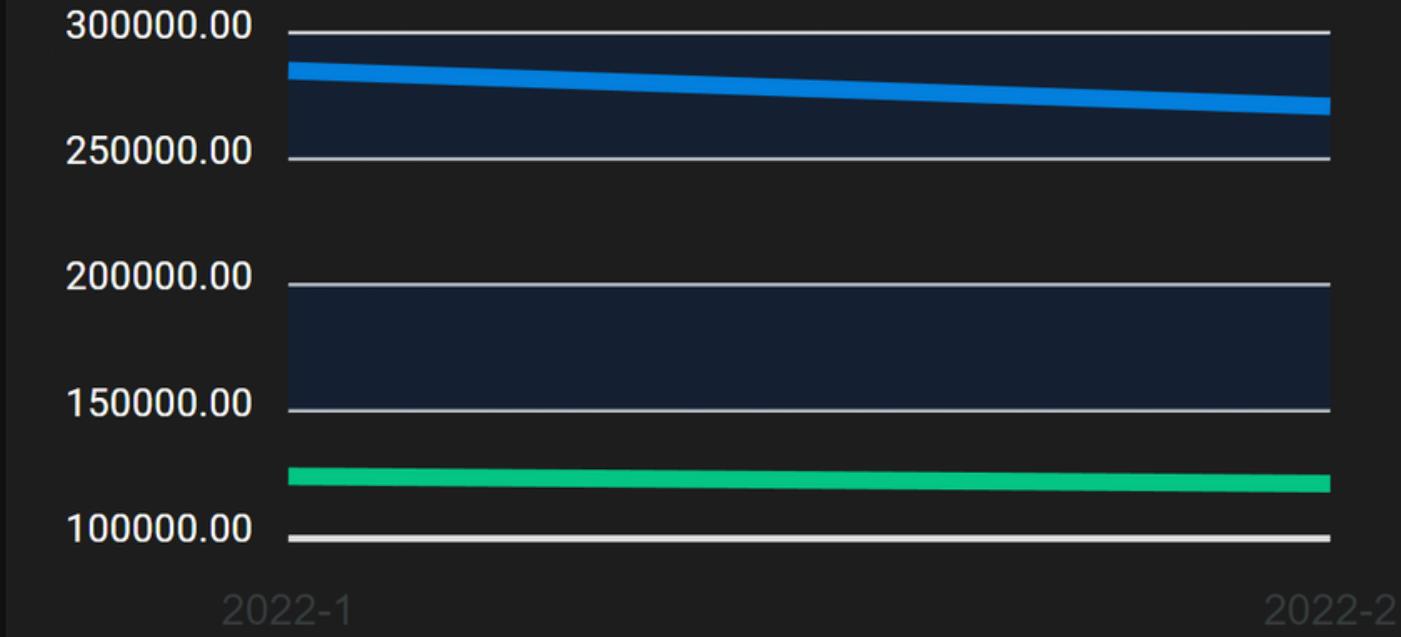
Complejidad Ciclomática

⊕ ⊖ 🔎 ⌚ ⌂ ⌃



Esfuerzo

⊕ ⊖ 🔎 ⌚ ⌂ ⌃



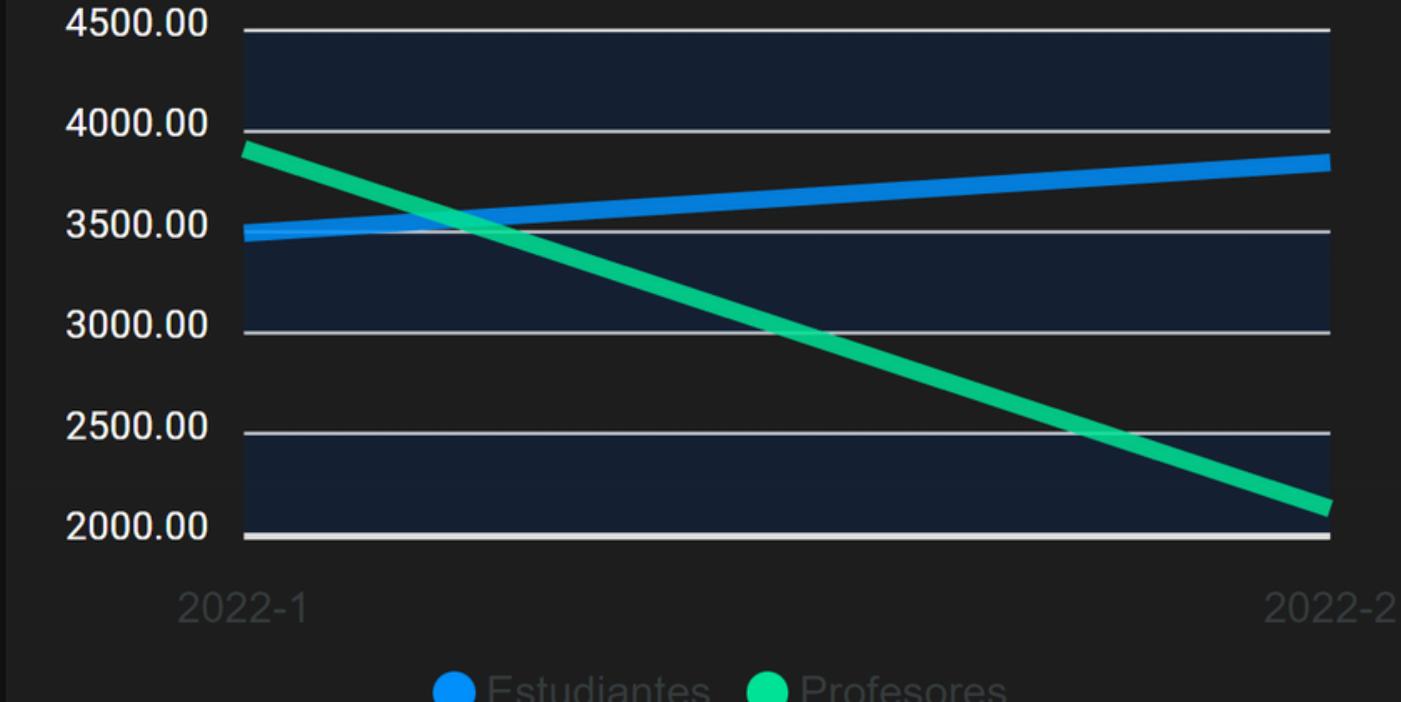
Tiempo

⊕ ⊖ 🔎 ⌚ ⌂ ⌃



Volumen

⊕ ⊖ 🔎 ⌚ ⌂ ⌃



Métricas históricas de estudiantes y profesores (Ciclos)

Complejidad Ciclomática

+ - 🔎 🖐️ 🏠 ⚓



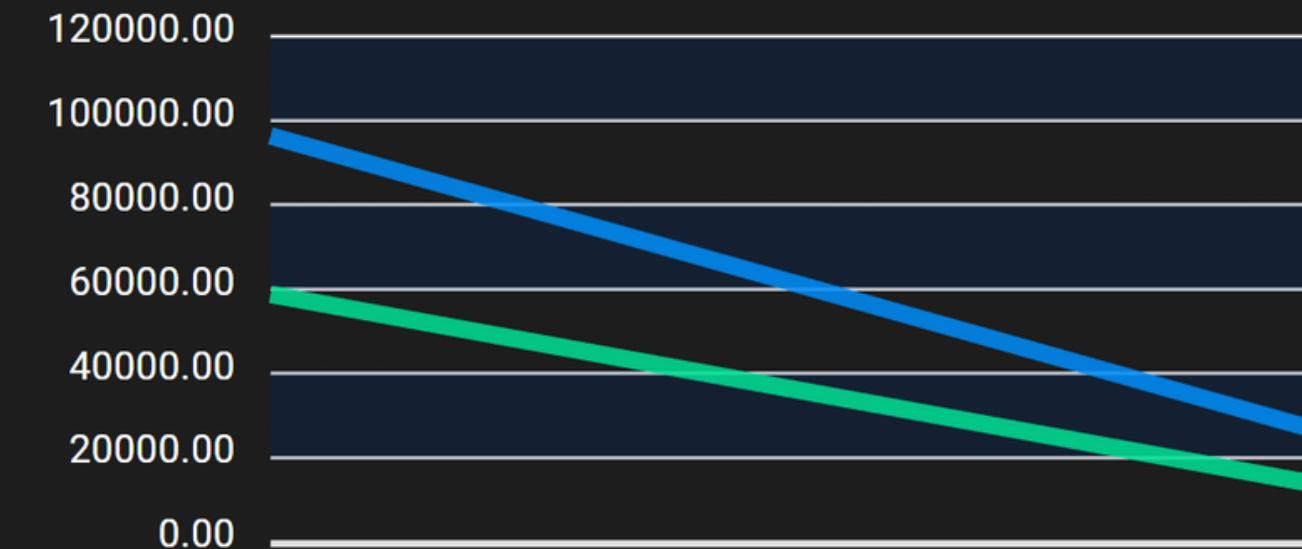
2022-1

2022-2

● Estudiantes ● Profesores

Esfuerzo

+ - 🔎 🖐️ 🏠 ⚓



2022-1

2022-2

● Estudiantes ● Profesores

Tiempo

+ - 🔎 🖐️ 🏠 ⚓



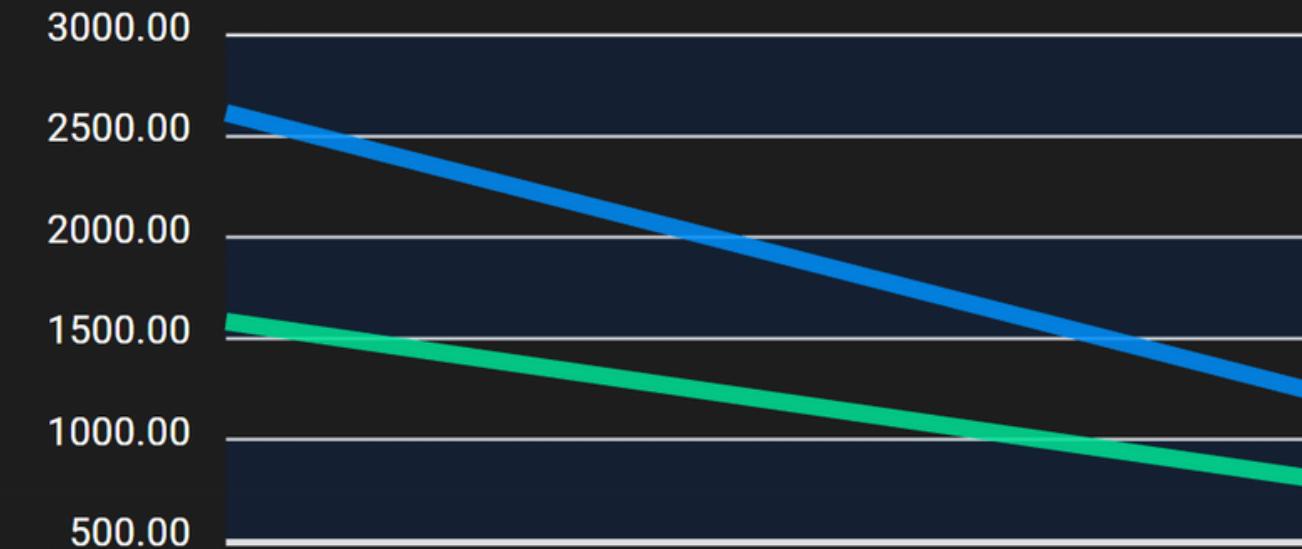
2022-1

2022-2

● Estudiantes ● Profesores

Volumen

+ - 🔎 🖐️ 🏠 ⚓



2022-1

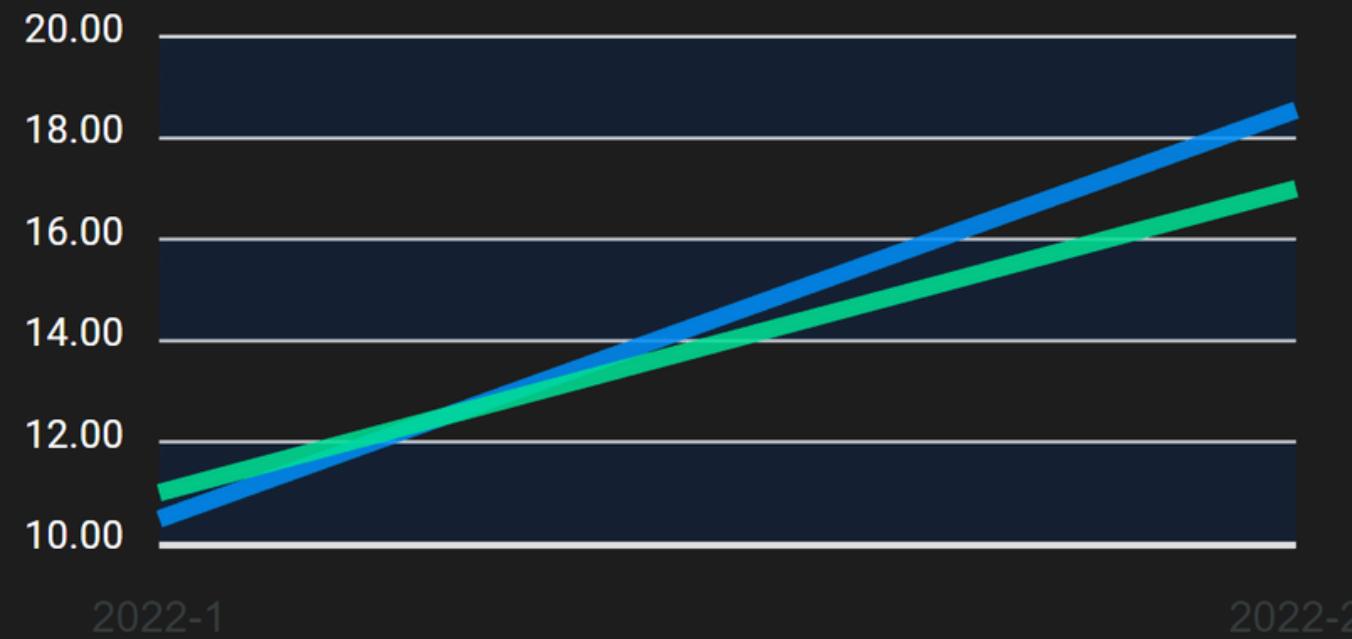
2022-2

● Estudiantes ● Profesores

Métricas históricas de estudiantes y profesores (Listas)

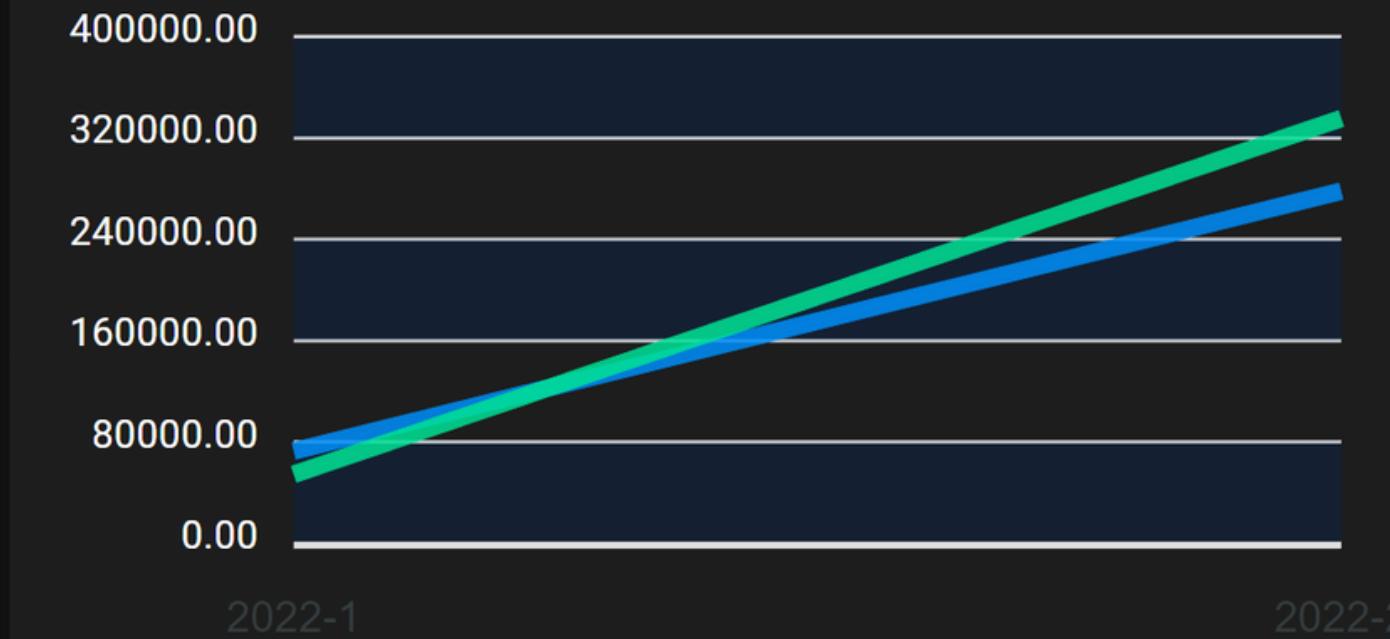
Complejidad Ciclomática

+ - 🔎 🖐️ 🏠 ⚓



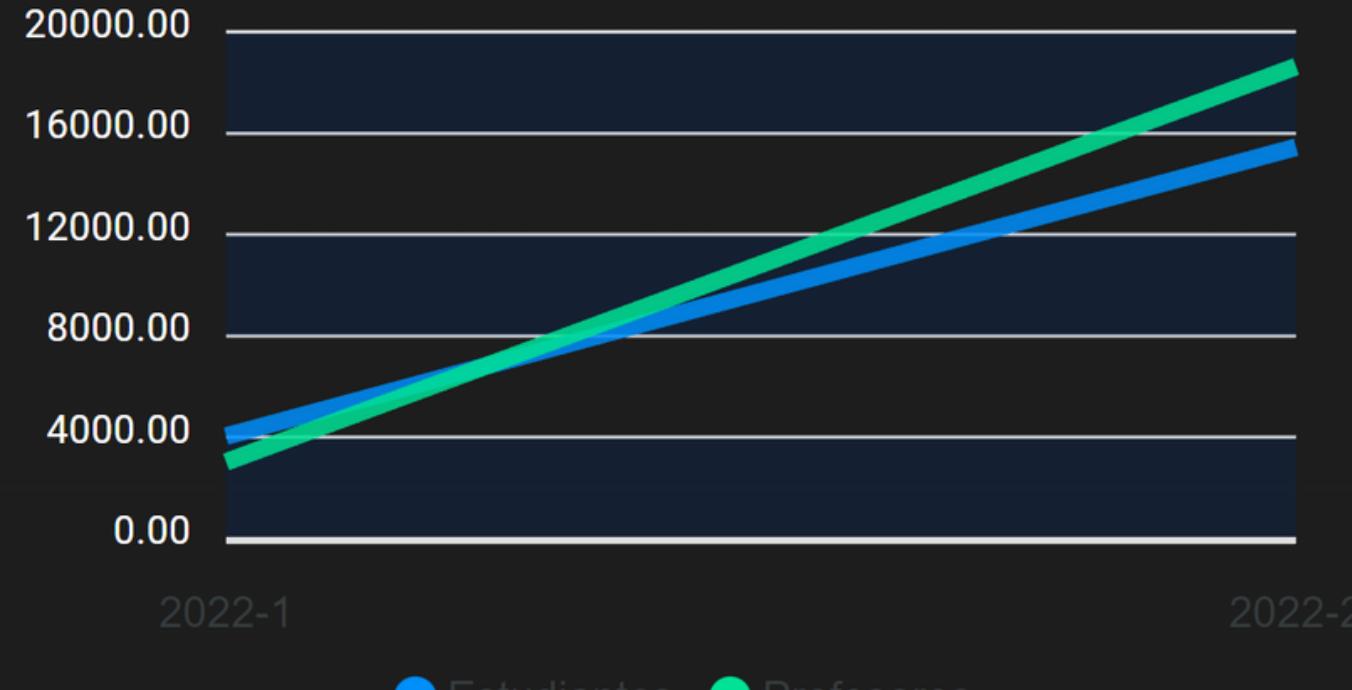
Esfuerzo

+ - 🔎 🖐️ 🏠 ⚓



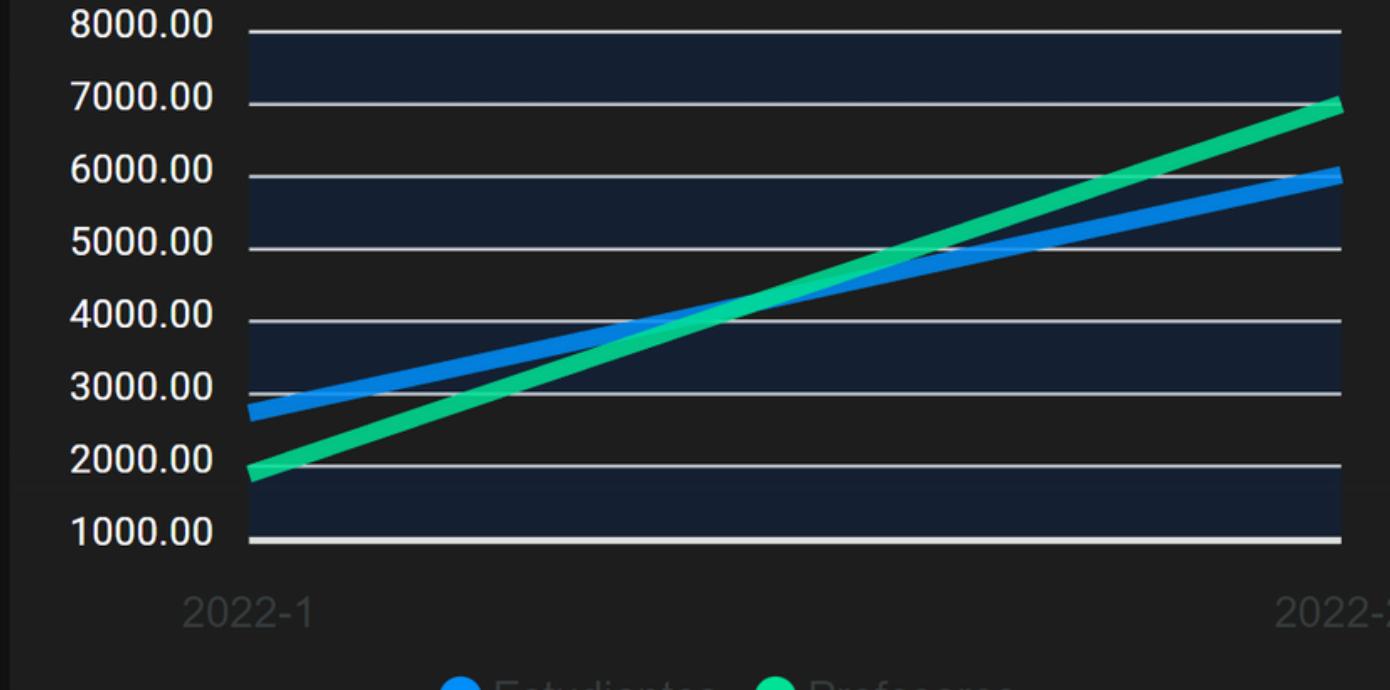
Tiempo

+ - 🔎 🖐️ 🏠 ⚓



Volumen

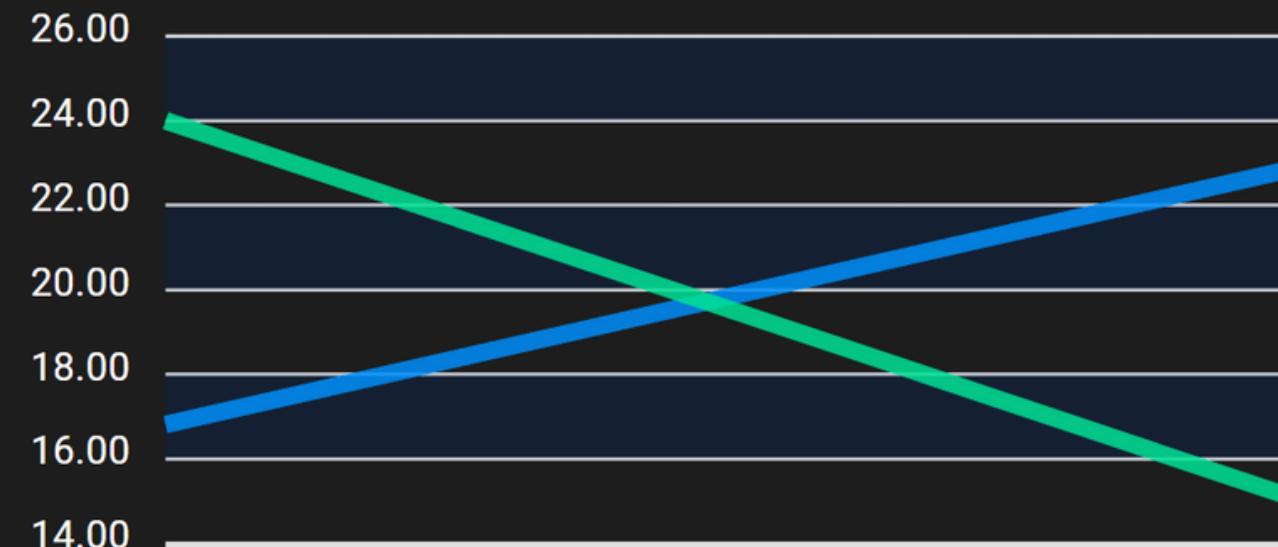
+ - 🔎 🖐️ 🏠 ⚓



Métricas históricas de estudiantes y profesores (Archivos)

Complejidad Ciclomática

+ - 🔎 🖐️ 🏠 ⚓



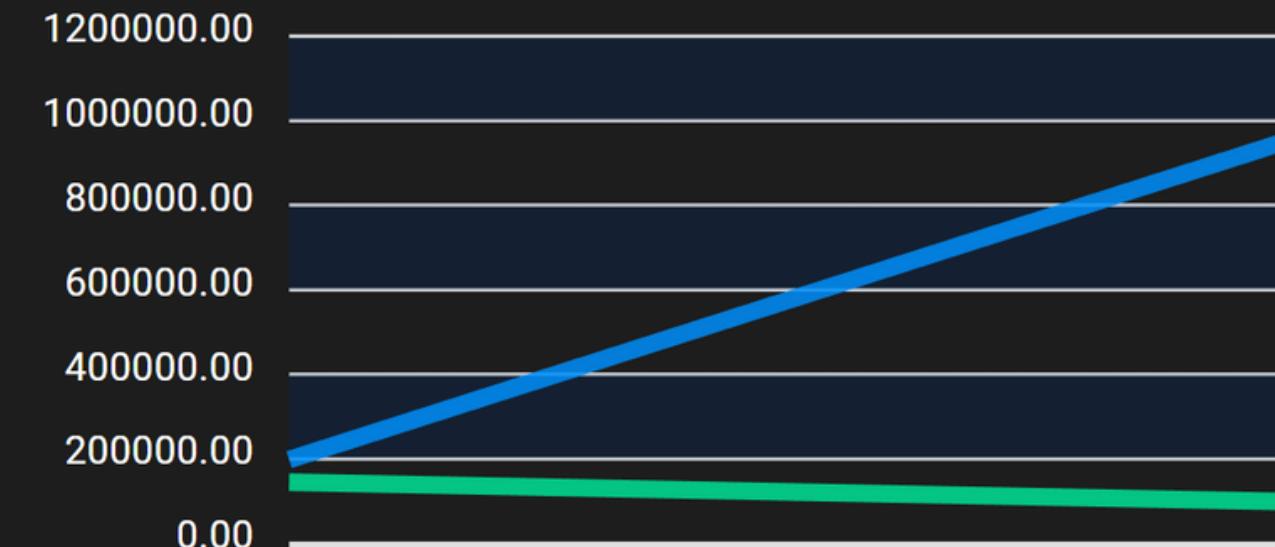
2022-1

2022-2

● Estudiantes ● Profesores

Esfuerzo

+ - 🔎 🖐️ 🏠 ⚓



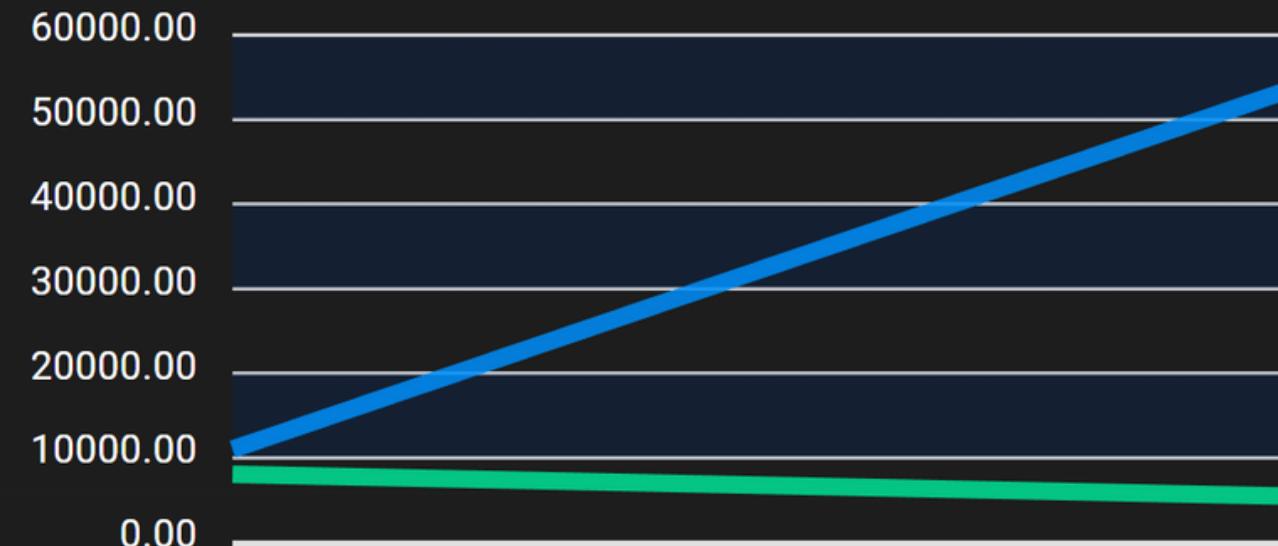
2022-1

2022-2

● Estudiantes ● Profesores

Tiempo

+ - 🔎 🖐️ 🏠 ⚓



2022-1

2022-2

● Estudiantes ● Profesores

Volumen

+ - 🔎 🖐️ 🏠 ⚓



2022-1

2022-2

● Estudiantes ● Profesores

SOBRE EL TRABAJO DE HALSTEAD Y SUS APLICACIONES

Elements of Software Science



Peter G. Hamer and Gillian D. Frewin

Standard Telecommunication Laboratories Limited
London Road, Harlow, Essex, England

Abstract

Karl Popper has described the scientific method as "the method of bold conjectures and ingenious and severe attempts to refute them". Software Science has made "bold conjectures" in postulating specific relationships between various 'metrics' of software code and in ascribing psychological interpretations to some of these metrics.

This paper describes tests made on the validity of the relationships and interpretations which form the foundations of Software Science. The results indicate that the majority of them represent neither natural laws nor useful engineering approximations.

TRABAJO FUTURO

ENCUESTAS A ESTUDIANTES

Se planea realizar encuestas a los estudiantes, para relacionar tanto su experiencia emocional como su percepción misma de la tarea, para analizarlas en conjunto con las métricas de Halstead.



← → ⌂



pyodide.org/en/stable/console.html

```
Welcome to the Pyodide terminal emulator 🐍
Python 3.10.2 (main, Jan 25 2023 18:32:53) on WebAssembly/Emscripten
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

Potenciando Javascript con WebAssembly: un caso aplicado a la Educación

Gonzalo Fernández -- Desarrollador Full-Stack



Respositorio con el código utilizado en el
análisis de la complejidad de las tareas.

