

Presentación

Trabajo de Inserción Profesional

Leandro Di Lorenzo

1 de Abril, 2017







Proyecto Mumuki

Mumuki es un proyecto que tiene como objetivo promover la educación de la programación utilizando software y contenido libre

Un poco de data...

- Mumuki es una plataforma educativa para la enseñanza de programación, y es *Open Source*.
- Se puede utilizar de forma autodidacta o como aula virtual con seguimiento.
- Es desarrollada y mantenida por estudiantes y docentes de varias universidades públicas, incluida la UNQ.
- Está siendo utilizada en diferentes entidades educativas del país.
- Tiene un enfoque didáctico con fuerte incapié en los conceptos por sobre las tecnologías.

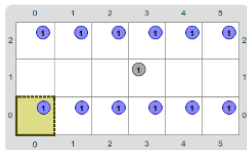


Ejercicio 6: Y claro, no podía faltar Argentina

¿Pensaste que nos habíamos olvidado de nuestra bandera? ¡Claro que no!.

Aunque como en Gobstones no hay amarillo, nos vamos a tomar el atrevimiento de cambiarlo por negro (perdón [Belgrano](#), no nos queda otra 🙄).

Con el cabezal en el origen, tu tarea es dibujar esta pseudo-bandera argentina:



1

reiniciar

▶ Enviar

Mumuki :: Corrección Automatizada

que siempre va a ser cero.

```
esVocalCerrada 'i' = True
esVocalCerrada 'u' = True
esVocalCerrada 'a' = False
esVocalCerrada 'e' = False
esVocalCerrada 'o' = False
esVocalCerrada otro = False -- porque ni siquiera es una voca
```

Veamos si queda claro: declaró una función `esCero` que diga si un número es cero. Hacerlo usando pattern matching.

reiniciar

▶ Enviar

¿Necesitás ayuda?

❌ Ups, algo no anduvo bien

Resultados de las pruebas:

✅ esCero 0 == True

❌ esCero 43 == False [Ver detalles](#)

❌ esCero 343 == False [Ver detalles](#)

❌ esCero 2.5 == False [Ver detalles](#)

🔔 ¿Pensás que algo está mal con este ejercicio?

Siguiente Ejercicio: Variable Anónima ▶

Esta guía fue desarrollada por Franco Bulgarelli, Gustavo Crespi y muchas personas más, bajo los términos de la [Licencia Creative Commons Compartir-Igual, 4.0](#).

© Copyright 2015-2016 Mumuki Project

Auth0 Token Based Authentication



Mumuki :: Seguimiento



/ Cursos / 2016-5b-labprog2



Guías

Estudiantes (16)

Docentes

Exámenes

Dirección de suscripción de estudiantes: <http://demo.classroom.mumuki.io/#/students/2016-5b-labprog2>

Ver estudiantes: Seguidos (0) · Todos (16)



Por progreso



Filtrar estudiantes



Manso, Juana

No hay email asociado

Registración: 13/05/2016

0

Total: 112

0

112



Martí, José

No hay email asociado

Registración: 13/05/2016

0

Total: 105

0

105



De San Martín, José

No hay email asociado

Registración: 13/05/2016

0

Total: 102

0

102



Moreno, Mariano

No hay email asociado

Registración: 13/05/2016

0

Total: 101

0

101



Artigas, José

No hay email asociado

Registración: 13/05/2016

0

Total: 98

0

98



Bolívar, Simón

No hay email asociado

Registración: 13/05/2016

0

Total: 89

2

87





De San Martín, José

No hay email asociado

- 1. MoverOeste10 ✓
- 2. La computadora repit... ✓
- 3. MoverOeste5 usand... ✓
- 4. No todo es repetir ✓
- 5. También vale después ✓
- 6. Repitiendo varios co... ✓
- 7. ¿Dónde está el error? ✓
- 8. Diagonal con una boli... ✓
- 9. Diagonal "pesada" ✓
- 10. El caso borde ✓
- 11. De lado a lado, dibu... ✓

11. De lado a lado, dibujamos un cuadrado

7 soluciones enviadas. Última solución: 16 junio 2016 - 12:42

< < 01 02 03 04 > >

Visualización:

Unificado -

Resuelto el 16 junio 2016 - 12:35

```
1 1 procedure CuadradoNegro4 () {
2 2   repeat(3) {
3 3     LineaNegra4Este ()
4 4     MoverOeste3YLuegoNorte ()
5 5   }
6 6   - LineaNegraEste ()
7 7   + Linea4NegraEste ()
8 8 }
9 9 procedure MoverOeste3YLuegoNorte () {
10 10   repeat (3) {
11 11     Mover(Oeste)
12 12   }
13 13 }
14 14
15 15 - procedure LineaNegra4Este() {
16 16   - repeat(3) {
17 17     - Poner(Negro)
18 18     - Mover(Este)
19 19   }
20 20   - Poner(Negro)
21 21 }
```

Comentar esta solución

Comentar

Resultados

Mumuki :: Contenido Personalizable

← EJERCICIOS

01. PADRES E HIJOS

02. TÍOS Y HERMANOS

03. DECLARATIVIDAD

04. ABUELOS

05. ALGUNAS CUENTAS

06. ¡QUIERO SABER MÁS!

07. INVERSIBILIDAD

08. NO DETERMINISMO, PARTE 1

09. NO DETERMINISMO, PARTE 2

10. CONSULTAS EXISTENCIALES

11. PROBLEMAS DE INVERSIBILIDAD, PARTE 1

12. PROBLEMAS DE INVERSIBILIDAD, PARTE 2

CONSULTAS EXISTENCIALES

META

STATEMENT

EVALUATION

HINTS

SOLUTION

Description

WRITE

PREVIEW

```
1 Las consultas existenciales (es decir, consultas que emplean la variable anónima) nos permiten saber si existe algún individuo que haga verdadera la
2 consulta, con independencia de cuál sea dicho individuo.
3 Hasta ahora, de todas formas, vale la pena aclarar que venimos haciendo trampa :innocent: : todas estas consultas las hicimos sobre un predicado 'padre/2'
4 que tenía una particularidad: sus cláusulas son todas hechas.
5 > ¿La inversibilidad funcionará con todos los predicados que definamos?
6 >
7 > Probá realizar consultas que pongan a prueba la inversibilidad: de lo que venimos haciendo. Usá variables y variables anónimas en consultas que utilicen
8 los siguientes predicados:
9 >
10 > * 'abuela/2'
11 > * 'caloriasenburguesa/2'
12 > * 'tio/2'
13 > * 'baboso/2'
```

Corollary

WRITE

PREVIEW

```
1
```

Capítulo 1: Fundamentos

¿Nunca programaste antes? Aprendé los fundamentos de la programación utilizando [Gobstones](#), un innovador lenguaje gráfico en el que utilizás un tablero con bolitas para resolver problemas.

Capítulo 2: Programación Imperativa

¿Ya estás para salir del tablero? ¡Acompañanos a aprender más sobre **programación imperativa** y **estructuras de datos** de la mano del lenguaje [JavaScript](#)!

Capítulo 3: Programación Funcional

El paradigma funcional es de los más **antiguos**, pero también de los más **simples** y **poderosos**. Si querés aprender a *dominar el mundo con nada*, utilizando el lenguaje [Haskell](#), seguí por acá.

Capítulo 4: Programación Lógica

¿Querés aprender a programar *describiendo el mundo* y *enseñando reglas* a la computadora? ¿Querés escribir código que cualquiera puede entender? Entonces acompañanos a aprender sobre el paradigma lógico, utilizando su lenguaje más conocido: [Prolog](#).

Capítulo 5: Metaprogramación

Cuando programamos, estamos razonando el mundo que nos rodea: yerba mate, videojuegos, contabilidad, cultivos. Pero también podríamos **razonar sobre programas**, para analizarlos, modificarlos o crearlos ¡Descubramos la metaprogramación, de la mano del lenguaje [Ruby](#)!

Proveer a los docentes de *Bases de Datos* una herramienta para:

- Mejorar la didáctica en la enseñanza de conceptos de SQL
- Facilitar el seguimiento del aprendizaje de los alumnos
- Contar con mejores elementos de evaluación

Proveer a los alumnos una herramienta para:

- Comprender mejor los conceptos recibidos
- Recibir feedback de forma temprana
- Poder analizar por su cuenta los fallos obtenidos para poder aprender de ellos
- Analizar su propio progreso visualizando los pasos dados hasta la resolución de los problemas

- Aportar al crecimiento del *Proyecto Mumuki* con una nueva tecnología de aprendizaje.
- Ayudar a todo programador que desee mejorar su capacidad y calidad en relación al uso de las bases de datos relaciones.
- Dejar abierta la posibilidad de extensión hacia:
 - Álgebra Relacional
 - Bases de datos no relaciones (MongoDB, Neo4j, Redis, etc...)

MQL :: Herramientas a Implementar

- Docker container con PostgreSQL o SQLite
- Runner de SQL en Ruby que permita:
 - Obtener y ejecutar el código SQL recibido
 - Validar resultados obtenidos contra esperados
 - Analizar sintaxis para exponer buenas prácticas de código
- Classroom con set de ejercicios

Mumuki :: Organizaciones que la utilizan

Organizaciones que ya están usando la Plataforma Mumuki...



FAMAF
Facultad de Matemática,
Astronomía y Física



**UNIVERSIDAD
NACIONAL DE
AVELLANEDA**

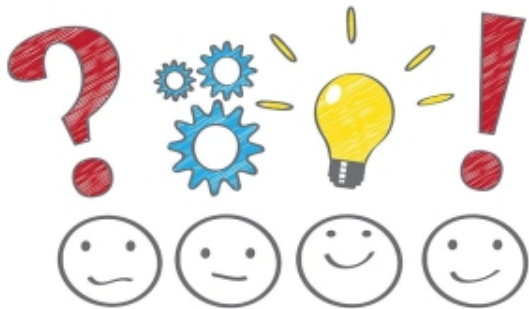


**UNIVERSIDAD BLAS
PASCAL**

DigitalHouse >
Coding School



- Proyecto Mumuki: <http://www.mumuki.org/>
- Plataforma: <https://mumuki.io/>
- Código en GitHub: <https://github.com/mumuki>
- Nota en Télam:
<http://www.telam.com.ar/notas/201703/183460-mumuki-plataforma-gratuita-programacion-aprendizaje-ensenanza-software.html>
- PostgreSQL: <https://www.postgresql.org/>
- SQLite: <https://www.sqlite.org/>
- Docker: <https://www.docker.com/>



¿Preguntas, ideas, comentarios?