

# Mumuki Query Learning

*Un Runner de SQL para el Proyecto Mumuki*

Leandro Di Lorenzo  
**Coordinador:** Ing. Fernando Dodino

TECNICATURA EN PROGRAMACIÓN INFORMÁTICA

8 de Julio, 2017



# Problemática y Motivación

## Nuevas formas de enseñar y aprender programación

Al comenzar a programar, una persona se enfrenta a desafíos de distinto tipo.

### El lenguaje de programación:

- Paradigma
- Sintaxis
- Plataforma  
*Desktop, Web, Mobile, ...*
- Entorno de desarrollo  
*IDE, Web Server, JVM, ...*
- Compilador / Intérprete

### Conceptos de programación:

- Abstracción
- Parametrización
- Procedimientos / Funciones
- Efecto / Estado
- Refactorización
- Tipado / Tipos de datos
- Legibilidad / Reglas de estilo

# Problemática y Motivación

## *El caso Bases de Datos*

- *Bases de Datos* es una materia de 2do semestre
- se viene de cursar *Introducción a la Programación y Organización de Computadoras*
- se cursan en simultáneo con *Programación con Objetos I*
- contenidos:
  - MER / MR
  - Normalización
  - Álgebra Relacional
  - SQL

# Problemática y Motivación

*El caso Bases de Datos*

## SQL :: Structured Query Language

El puente entre la teoría del álgebra relacional y la de la programación se logra a partir del *lenguaje SQL*, el cual permite **esquematizar y manipular la información**.

Se busca impartir conceptos de:

- Entidad / Atributos
- Relación / Cardinalidad
- Claves (*PK*, *FK*)
- Dependencias (*DFs*, *DMs*)

Y como efecto colateral:

- Motores  
*Oracle, Postgres, SQLite, SQL-Server, MySQL, ...*
- Estándares *ANSI / ISO*
- Portabilidad de datos

# Propuesta

*Integrar SQL como parte del Proyecto Mumuki*



## Proyecto Mumuki

---

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

# Mumuki

*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.

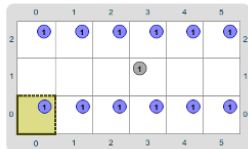


## Ejercicio 6: Y claro, no podía faltar Argentina

¿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 Tour

## Muestra de errores

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

[reiniciar](#)[▶ Enviar](#)

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

🔗 ¿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](#)



# Mumuki Tour

## Aula virtual



/ 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	
	<b>Manso, Juana</b> No hay email asociado Registro: 13/05/2016	0	Total: 112	112	
	<b>Martí, José</b> No hay email asociado Registro: 13/05/2016	0	Total: 105	105	
	<b>De San Martín, José</b> No hay email asociado Registro: 13/05/2016	0	Total: 102	102	
	<b>Moreno, Mariano</b> No hay email asociado Registro: 13/05/2016	0	Total: 101	101	
	<b>Artigas, José</b> No hay email asociado Registro: 13/05/2016	0	Total: 98	98	
	<b>Bolívar, Simón</b> No hay email asociado		Total: 89		



- ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓



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

Visualización:

Unificado -

  Resuelto el 16 junio 2016 - 12:35

Comentar

# Plataforma Mumuki

## Componentes

La **Plataforma Mumuki** se puede entender desde estos componentes:

**Laboratory** entorno web en donde los estudiantes resuelven ejercicios y reciben *feedback*

**Classroom** herramienta para que el docente pueda generar seguimiento de sus alumnos

**Bibliotheca** repositorio de guías y ejercicios

**Runners** componentes que se encargan de ejecutar y verificar los programas enviados por los alumnos

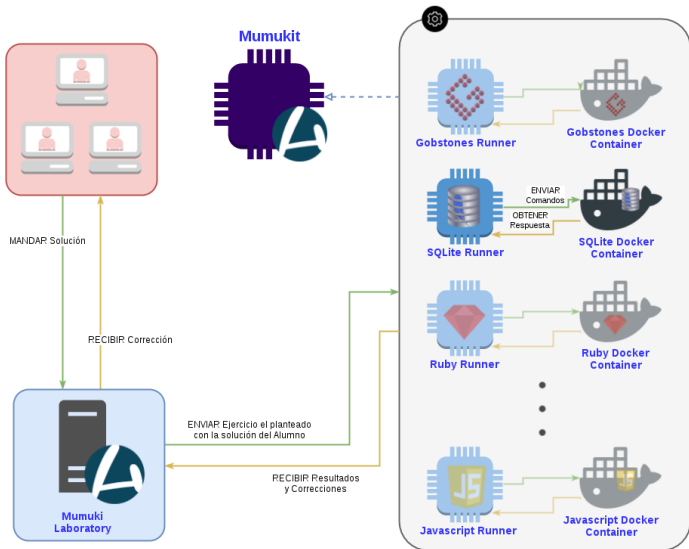
# Plataforma Mumuki

## Los Runners

- Un *Runner* se encarga de ejecutar una tecnología en particular
- Existen *Runners* de todo tipo de tecnología comercial  
*Java, Ruby, Python, Javascript, C++, Haskell, Prolog, ...*
- Y también *Runners* de tecnologías educativas  
*Gobstones, WolloK, QSIM, ...*
- Cada ejercicio se asocia con el *Runner* de la tecnología necesaria
- Cuando un alumno envía la solución de un ejercicio, Mumukit genera una petición al *Runner* asociado
- El *Runner* compila o interpreta el ejercicio, lo ejecuta y retorna los resultados luego de todas las verificaciones realizadas

# Plataforma Mumuki

## Los Runners



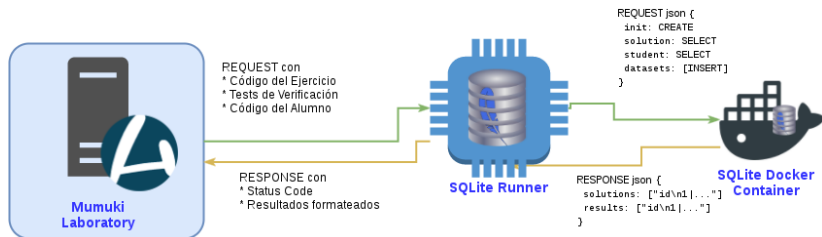
# Un Runner para SQL

## Stack Tecnológico

- Los *Runners* de Mumuki deben estar programados como una gema de *ruby* utilizando el framework *Mumukit*
- *Mumukit* se encarga de establecer la interacción con la plataforma y delegar el trabajo puntual al *Runner* correspondiente
- Cada *Runner* debe tener asociado un *Docker Container*
- *Docker* plantea una forma de virtualización en donde se pueden crear y destruir *VMs* de forma muy veloz y a bajo costo
- La ventaja de utilizar *Docker* es la posibilidad de ejecutar cada ejercicio de cada alumno en un ambiente aislado y limpio.

# Runner SQLite

## Stack & Flow



# Runner SQLite

## Stack & Flow

- El *Runner SQLite* adapta el contenido a un formato cómodo de manipular por el *worker* (docker container)
- El *container* es una VM con: linux alpine, sqlite3 y python 2.7
- Se inicia un *container* fresco que recibe el ejercicio a ejecutar en formato *json*
- A través de un script en *python* se crea la base, se inicializa el ejercicio y se ejecutan las soluciones del alumno y del docente para cada set de datos
- Los resultados se retornan en formato *json* junto al *exit code*



# Runner SQLite

## Ejemplo 1

Ejemplo de un ejercicio de tipo SELECT con solución vía *Query*

```
CREATE TABLE motores (  
  id integer primary key,  
  nombre varchar(200) NOT NULL  
);
```

Código 1: Extra (Docente, SQL)

```
SELECT id, nombre  
FROM motores;
```

Código 2: Content (Alumno, SQL)

```
solution_type: query  
solution_query: select * from motores;  
examples:  
  - data: |  
      INSERT INTO motores values ('MySQL');  
      INSERT INTO motores values ('PostgreSQL');  
      INSERT INTO motores values ('Oracle');  
      INSERT INTO motores values ('SQL Server');  
      INSERT INTO motores values ('SQLite');
```

Código 3: Test (Docente, YAML)

# Runner SQLite

## Ejemplo II

Ejemplo de un ejercicio de tipo SELECT con solución vía *Datasets*

```
CREATE TABLE motores (  
  id integer primary key,  
  nombre varchar(200) NOT NULL  
);
```

Código 4: Extra (Docente, SQL)

```
SELECT id, nombre  
FROM motores;
```

Código 5: Content (Alumno, SQL)

```
solution_type: datasets  
examples:  
  - data: |  
      INSERT INTO motores values ('MySQL');  
      INSERT INTO motores values ('PostgreSQL');  
      INSERT INTO motores values ('Oracle');  
      INSERT INTO motores values ('SQL Server');  
      INSERT INTO motores values ('SQLite');  
  solution_dataset: |  
      id|color  
      1|MySQL  
      2|PostgreSQL  
      3|Oracle  
      4|SQL Server  
      4|SQLite
```

Código 6: Test (Docente, YAML)

# Runner SQLite

## Ejemplo III

### Ejemplo de un ejercicio de tipo CREATE con solución vía *Datasets*

-- NONE

#### Código 7: Extra (Docente, SQL)

```
CREATE TABLE motores (  
  id integer primary key,  
  nombre varchar(200) NOT NULL  
);  
  
INSERT INTO motores values ('MySQL');  
INSERT INTO motores values ('PostgreSQL');  
INSERT INTO motores values ('Oracle');  
INSERT INTO motores values ('SQL Server');  
INSERT INTO motores values ('SQLite');  
  
SELECT * FROM motores;
```

#### Código 8: Content (Alumno, SQL)

```
solution_type: datasets  
examples:
```

```
- data: -- none  
  solution_dataset: |  
    id|color  
    1|MySQL  
    2|PostgreSQL  
    3|Oracle  
    4|SQL Server  
    4|SQLite
```

#### Código 9: Test (Docente, YAML)

## DEMO



# Mumuki Query Learning :: Objetivos

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

# MQL :: Consecuencias Deseadas

- 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



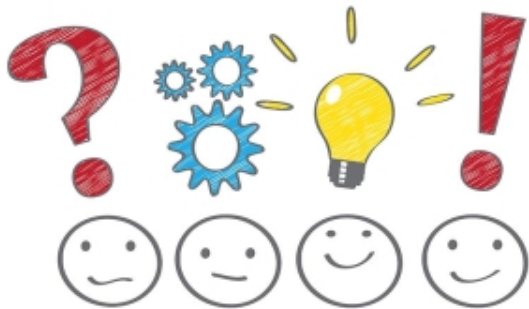
**DigitalHouse** >  
Coding School





# Algunos Links

- 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?