

---

*Kaeme*

**Diseño de una plataforma de aprendizaje  
basada en tarjetas de memoria**

---

*Bruno Orcha García*

Máster Universitario en Desarrollo de sitios y aplicaciones web

---

# Motivación

## Arquitectura

## Funcionalidades

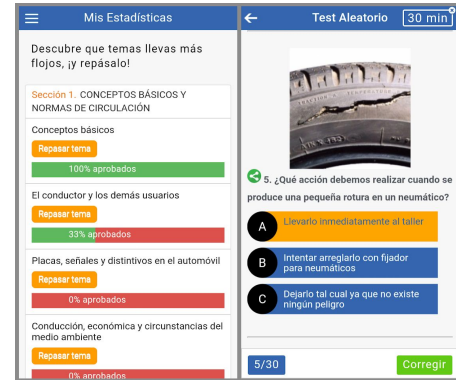
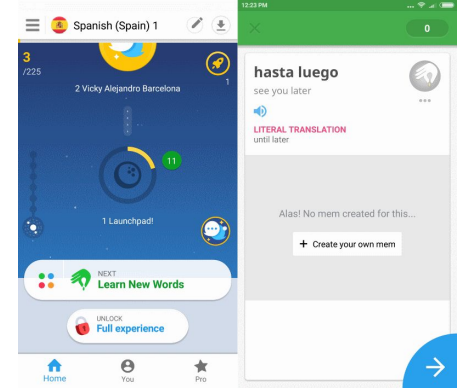
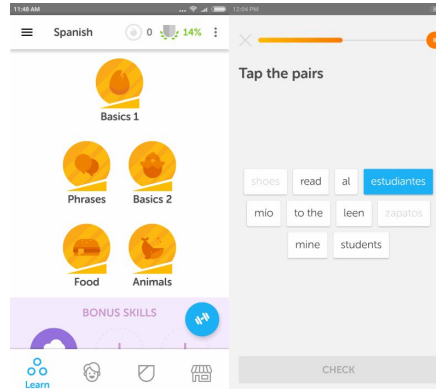
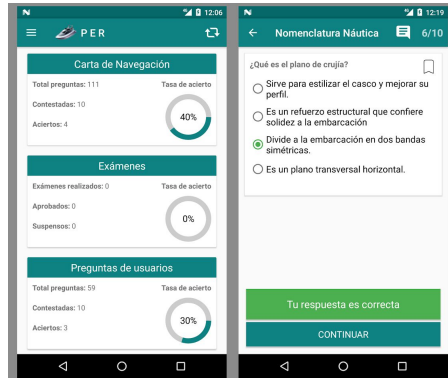
## Demostración

---

# Motivación

## Existen muchas aplicaciones con el mismo patrón

- Aprendizaje de Idiomas
- Preparación Test de Conducir
- Preparación examen PER
- Preparación de certificaciones



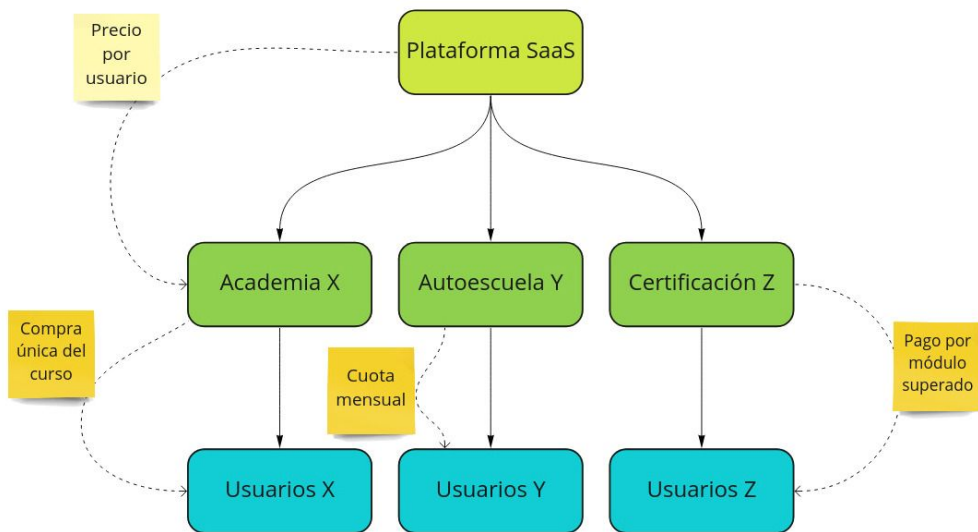
# Motivación

## Herramienta para formadores

- Producto flexible
- Sencillo de utilizar
- Digitalización de academias
- Complemento a clases presenciales

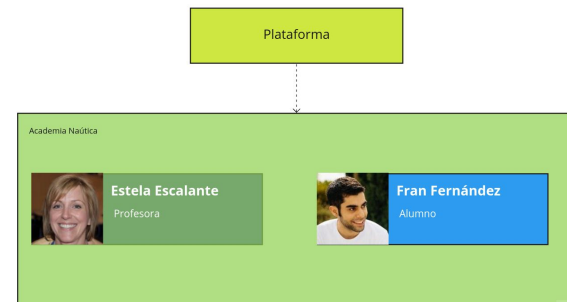


# Motivación



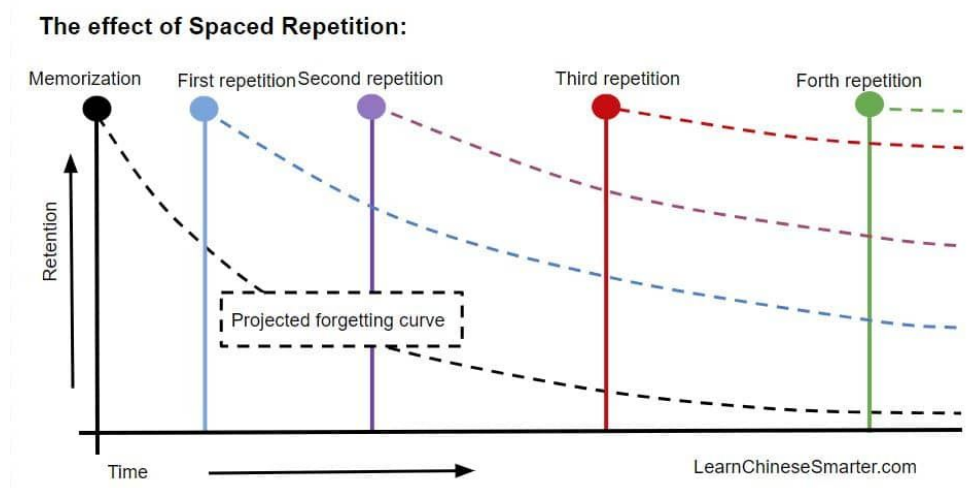
## Plataforma SaaS (Software as a Service)

- Nicho de mercado
- Modelo de negocio
- Proveedor de servicio
- Personalización



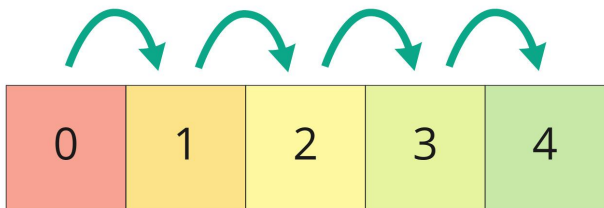
# Repetición espaciada

- Métodos de estudio con tarjetas de memoria
- Curva del olvido (tiempo que pasa hasta que olvidamos)
- Método Leitner (repassar más frecuentemente los conceptos menos asimilados)

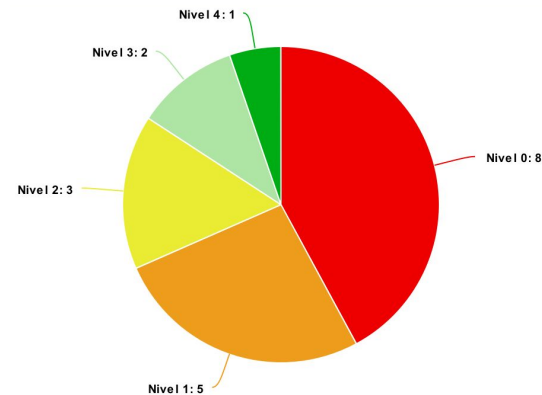


## Repetición espaciada

Acertar la tarjeta subimos de nivel



Si fallamos la tarjeta bajamos de nivel



### Método elegido

- Las tarjetas tienen asociado un nivel
- Subida y bajada de nivel según aciertos o fallos
- Elección de tarjetas ponderada

Motivación

**Arquitectura**

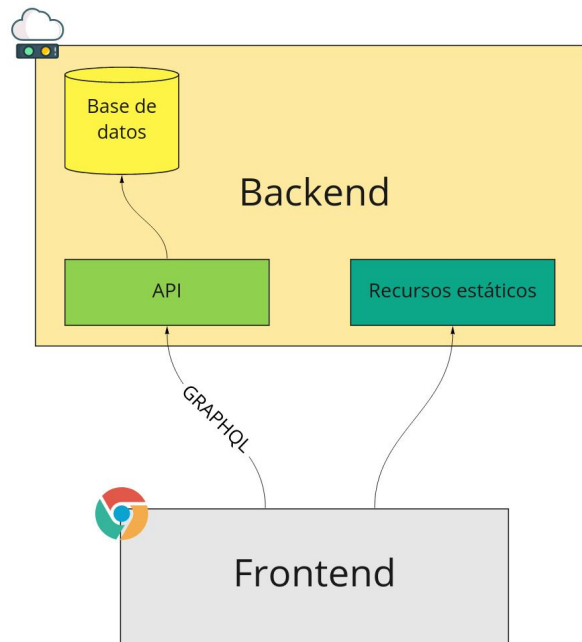
Funcionalidades

Demostración

---



# Arquitectura



- Base de datos SQLite
- Backend ofrece API GraphQL, acceso a datos y recursos estáticos
- Frontend aplicación lado cliente JavaScript

# Arquitectura



GraphQL

Describe your data

```

type Project {
  name: String
  tagline: String
  contributors: [User]
}
                    
```

Ask for what you want

```

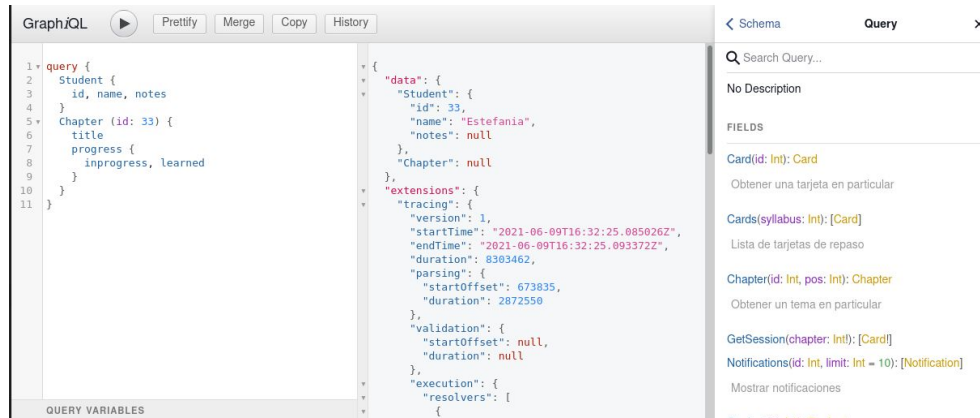
project(name: "GraphQL") {
  tagline
}
                    
```

Get predictable results

```

{
  "project": {
    "tagline": "A query language for APIs"
  }
}
                    
```

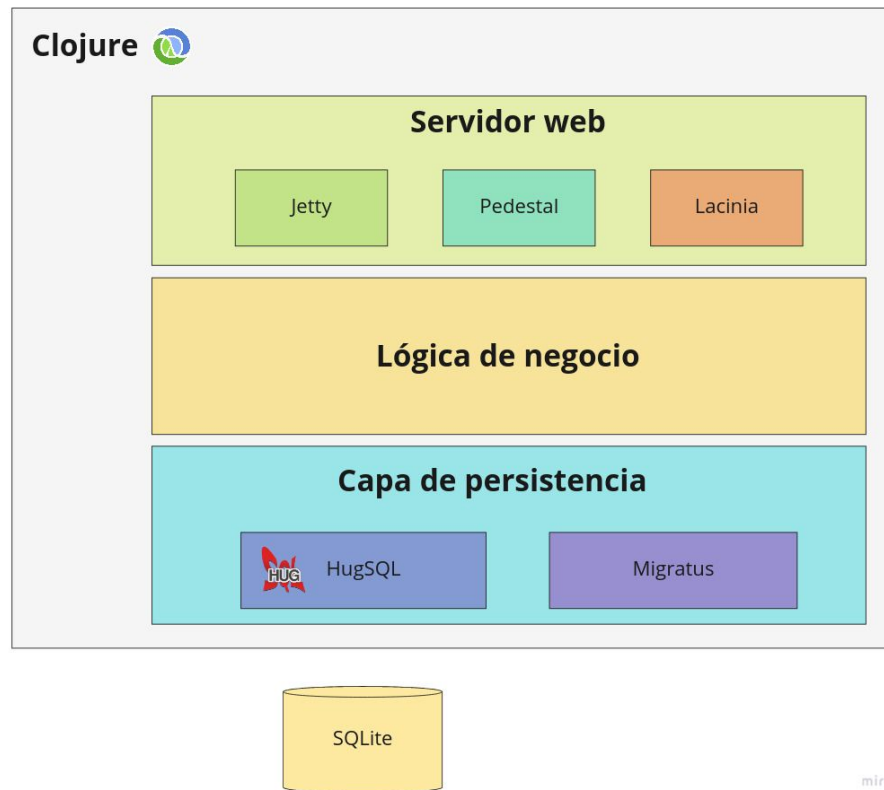
- Definición de un esquema
- Lenguaje de consulta
- Diferencias respecto REST



The screenshot shows the GraphQL Playground interface. On the left, a query is defined: `query { student { id, name, notes } chapter(id: 33) { title progress { inprogress, learned } } }`. The right pane shows the JSON response: `{ "data": { "student": { "id": 33, "name": "Estefania", "notes": null }, "chapter": null }, "extensions": { "tracing": { "version": 1, "startTime": "2021-06-09T16:32:25.085026Z", "endTime": "2021-06-09T16:32:25.093372Z", "duration": 8383462, "parsing": { "startOffset": 673835, "duration": 2872550 }, "validation": { "startOffset": null, "duration": null }, "execution": { "resolvers": [ ... ] } } } }`. On the far right, a sidebar lists schema fields like `Card(id: Int!): Card`, `Cards(syllabus: Int!): [Card]`, etc.

## Arquitectura

- Tecnología JVM
- Lenguaje de programación Clojure
- Lacinia (Walmart)
- HugSQL
- SQLite



# Arquitectura

```
progress.sql
1  -- Contar tarjetas
2  -- :name total-progress :? :1
3  SELECT SUM(pending) pending,
4         SUM(inprogress) inprogress,
5         SUM(learned) learned
6  FROM progress p
7  WHERE student_id = :id
8  GROUP BY student_id
9
10 -- Obtener el progreso de un punto del temario
11 -- :name get-progress-by-id :? :1
12 SELECT *
13 FROM progress
14 WHERE id = :id
15
16
17 -- Obtener la última sesión realizada
18 -- :name last-progress-of-student :? :1
19 SELECT *
20 FROM progress
21 WHERE student_id = :id
22 ORDER BY date DESC
23 LIMIT 1
24
```



*HugSQL is a Clojure library for embracing SQL.*

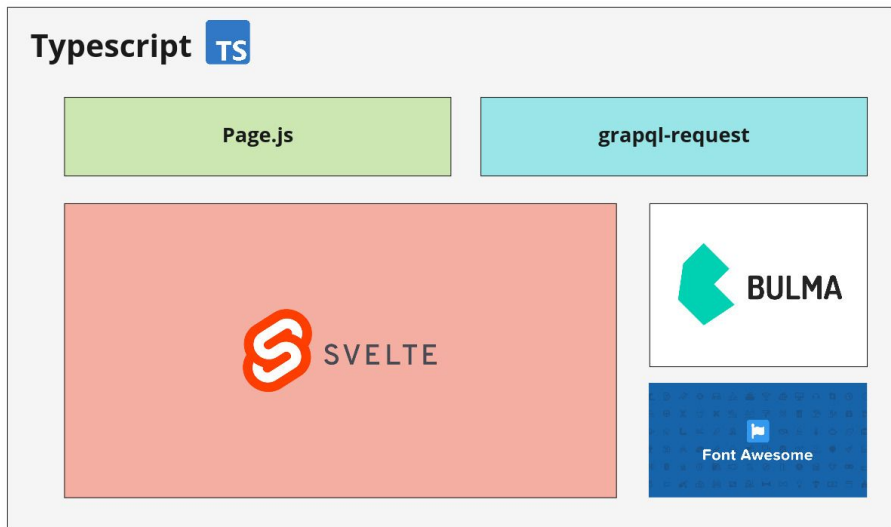
- SQL is the *right tool for the job* when working with a relational database!
- HugSQL uses [simple conventions](#) in your SQL files to define (at compile time) database functions in your Clojure namespace, creating a clean separation of Clojure and SQL code.

```
(total-progress db {:id 23})
```

```
(get-progress-by-id db {:id 34})
```

```
(last-progress-of-student db {:id 45})
```

# Arquitectura



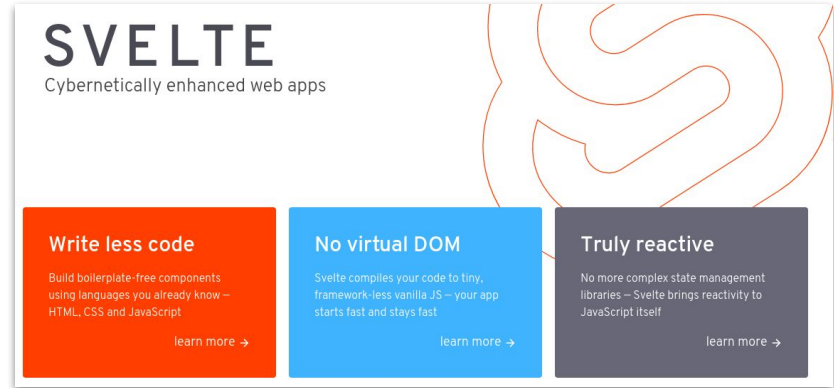
miro

- Constructor ParcelJS
- TypeScript
- Bulma CSS Framework
- Svelte framework de componentes

# Arquitectura

Definición simple en el mismo fichero: HTML + CSS + JS

```
Field.svelte
1 <script>
2   export let label = "";
3   export let title = "";
4 </script>
5
6 <div class="field">
7   <!-- svelte-ignore ally-label-has-associated-control -->
8   <label class="label" title={title}>{label}
9   <div class="control">
10    <slot />
11  </div>
12 </label>
13 </div>
```



**SVELTE**  
Cybernetically enhanced web apps

**Write less code**  
Build boilerplate-free components using languages you already know – HTML, CSS and JavaScript  
[learn more →](#)

**No virtual DOM**  
Svelte compiles your code to tiny, framework-less vanilla JS – your app starts fast and stays fast  
[learn more →](#)

**Truly reactive**  
No more complex state management libraries – Svelte brings reactivity to JavaScript itself  
[learn more →](#)

```
<Field label="Título">
  <input required class="input" type="text"
  placeholder="Título del tema" bind:value={card.title}>
</Field>
```

Motivación

Arquitectura

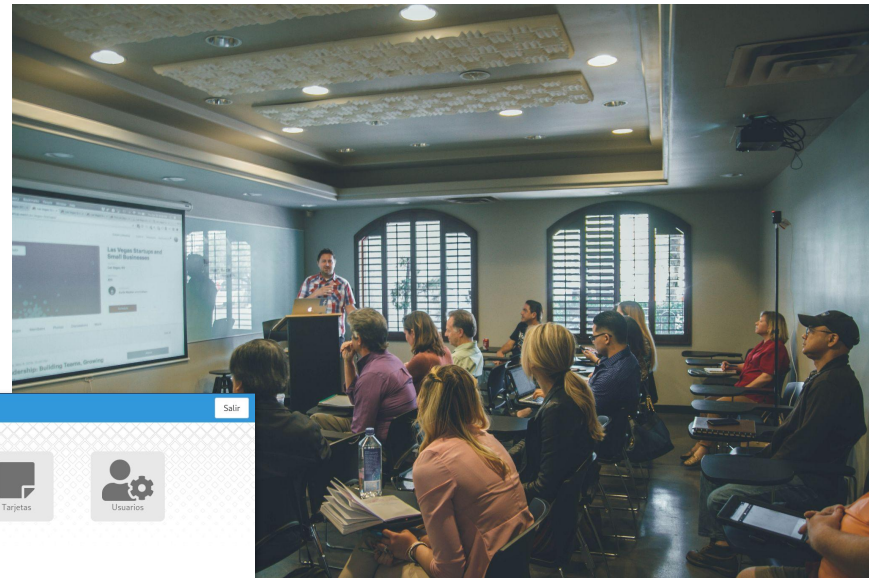
**Funcionalidades**

Demostración

# Funcionalidades

## Profesores

- Gestión de usuarios/Profesores
- Composición del temario
- Creación de tarjetas
- Registro de Alumnos
- Seguimiento de alumnos



Kaeme Alumnos Temario Tarjetas Usuarios Salir

Alumnos Temario Tarjetas Usuarios

**Actividad reciente**

- Modificación del alumno **Alumno Juan Perez** (por Admin hace 1 día)
- Se actualizan observaciones: **Alumno Juan Perez** (por Admin hace 1 día)
- Se actualizan observaciones: **Alumno Juan Perez** (por Admin hace 1 día)
- Registro de nuevo alumno **Alumno Juan Perez** (por Admin hace 1 día)

Kaeme Ayuda Información Legal Sobre nosotros

UOC Brno Obra García



# Funcionalidades



## Alumnos

- Activación
- Consulta del temario
- Realización de repasos
- Consulta de estadísticas

**Kaeme**

Tema actual  
**3. SEGURIDAD EN LA MAR**

Ver temario **Repasar**

Estadísticas

Sesiones realizadas en los últimos meses

	May	Jun
Lun		■
Mar		
Mie		■
Jue		
Vie	■	■
Sab		■
Dom	■	

Pendientes	En curso	Aprendidas
7	20	8

Estas registrado en este dispositivo como *Benito Basallote*  
[Cerrar sesión](#)

## Proyección a futuro

- Multientidad y personalización
- Gestión de clases y temarios
- Análisis de tarjetas
- Integración con plataformas de video
- Gamificación



Motivación

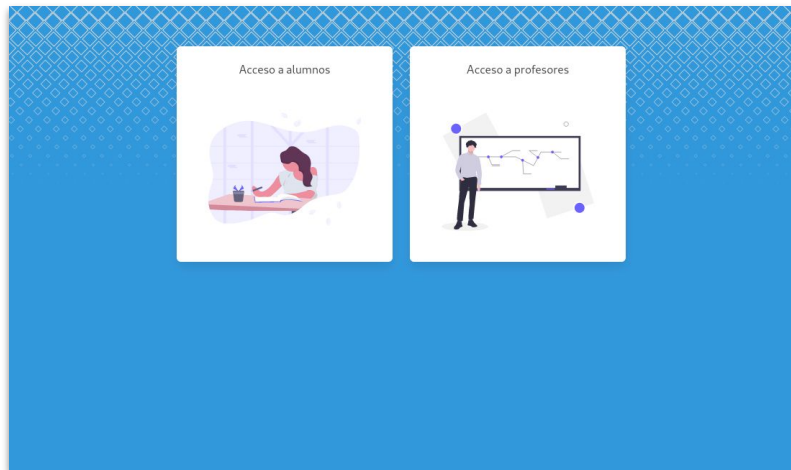
Arquitectura

Funcionalidades

**Demostración**

---

## Demostración



<https://kaeme.site>

Aplicación autoiniciada con docker

<http://IP-INSTANCIA-AWS>

### Credenciales de ejemplo

- [admin1@admin.com](mailto:admin1@admin.com) / 123456
- [admin2@admin.com](mailto:admin2@admin.com) / 123456
- [admin3@admin.com](mailto:admin3@admin.com) / 123456

*Bruno Orcha García*

Máster Universitario en Desarrollo de sitios y aplicaciones web