

¡hola!

flexibility

**Creamos conexiones simples y
eficientes con tecnología.**

flexibility.com.ar

¡hola!

flexibility +  Ada

Intro al mundo del Backend

Esperamos que puedan sacar algo productivo del workshop ;)

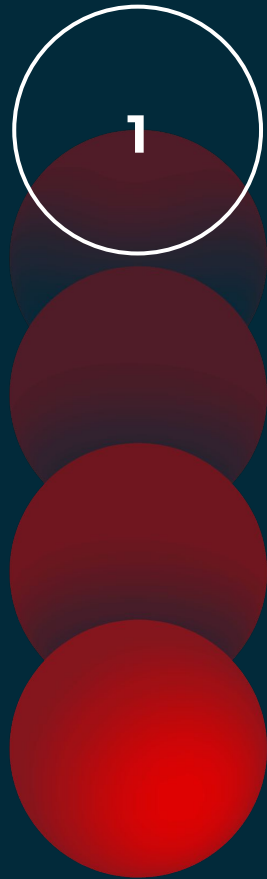




A continuación

 click en los títulos para acceder a cada sección.

- 1 Frontend y backend: diferencias
- 2 ¿Qué es una aplicación?
- 3 ¿Qué es una API?
- 4 ¿Qué es una API REST?
- 5 Proyecto empresarial



Frontend y backend: diferencias

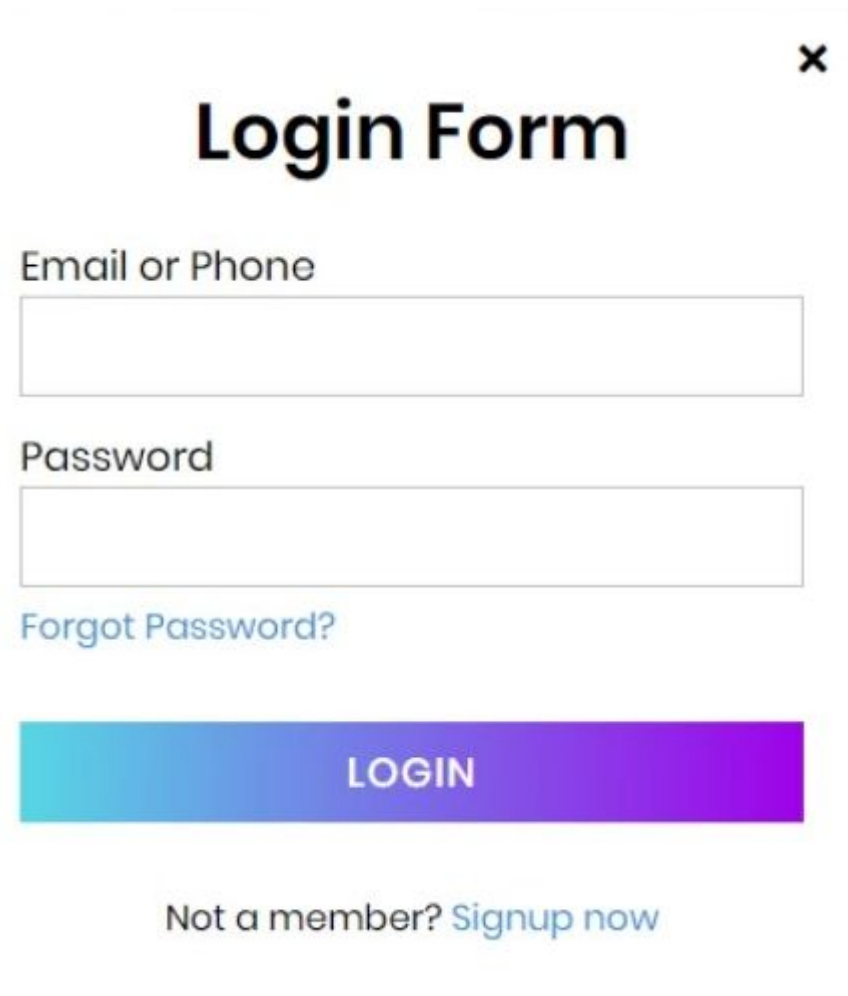
Diferencias

Diferencias entre backend y frontend

Primero lo formal, **frontend** significa “parte del frente”, mientras que **backend** “parte de atrás”.

Una definición más formal sería “en diseño de software el **frontend** es la parte del **software** que **interactúa con los usuarios** y el **backend** es la parte que **procesa los datos** que envía el **frontend**.”

Analicemos el siguiente formulario de inicio de sesión...



Close button (x)

Login Form

Email or Phone

Password

[Forgot Password?](#)

LOGIN

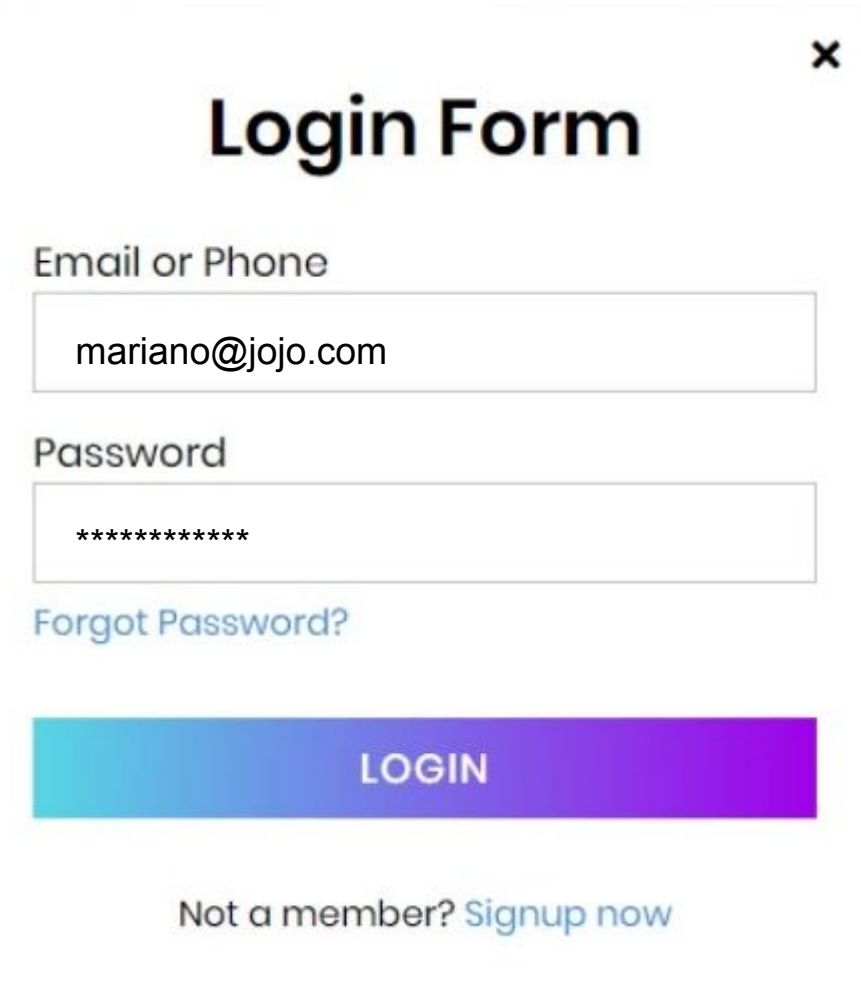
Not a member? [Signup now](#)

Diferencias

Podemos observar que tiene dos cuadros de texto para poder **introducir texto**, el campo **usuario**, que puede introducir tanto su email como su número de teléfono, y el campo **password**.

Su código sería el siguiente:

```
1 <form action="#">
2   <div class="data">
3     <label>Email or Phone</label>
4     <input type="text" required />
5   </div>
6   <div class="data">
7     <label>Password</label>
8     <input type="password" required />
9   </div>
10  <div class="forgot-pass">
11    <a href="#">Forgot Password?</a>
12  </div>
13  <div class="btn">
14    <div class="inner"></div>
15    <button type="submit">login</button>
16  </div>
17 </form>
18
```



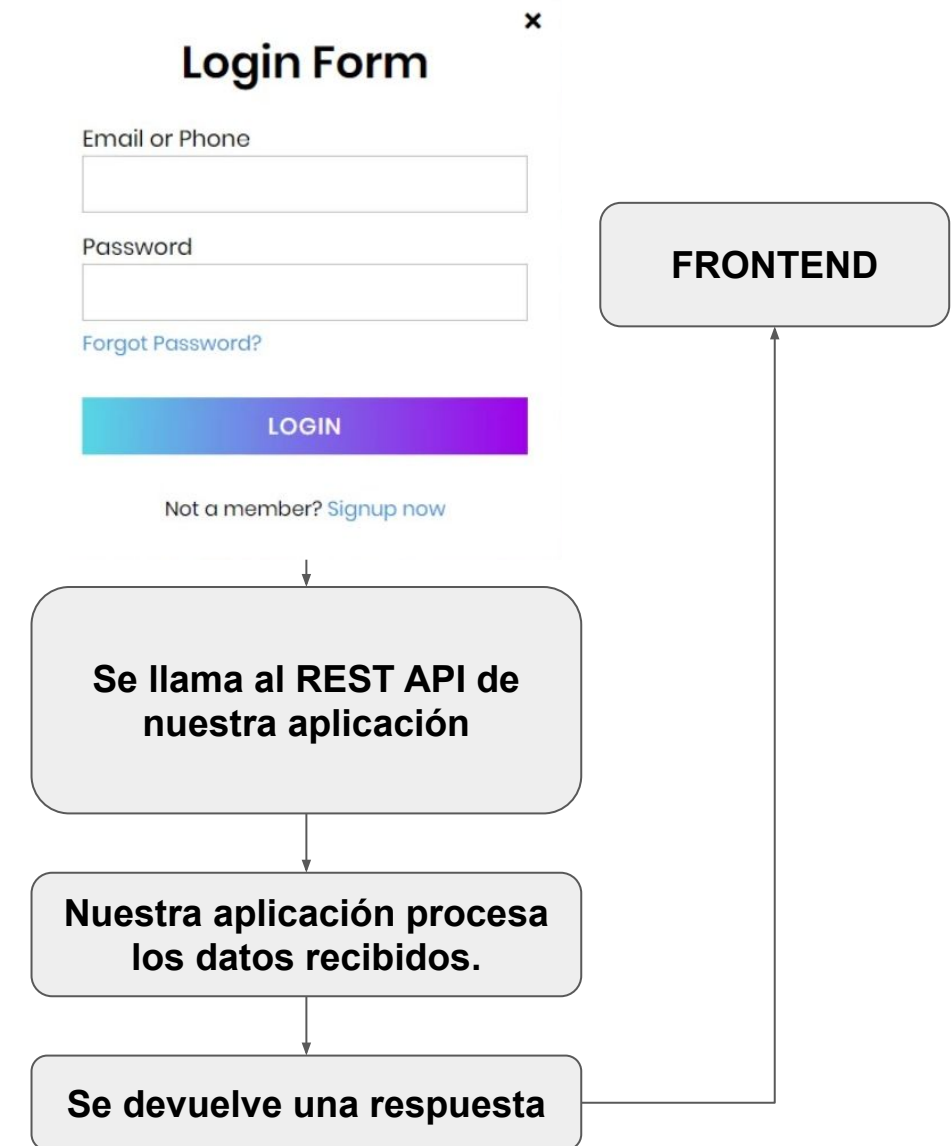
The screenshot shows a login form with the following elements:

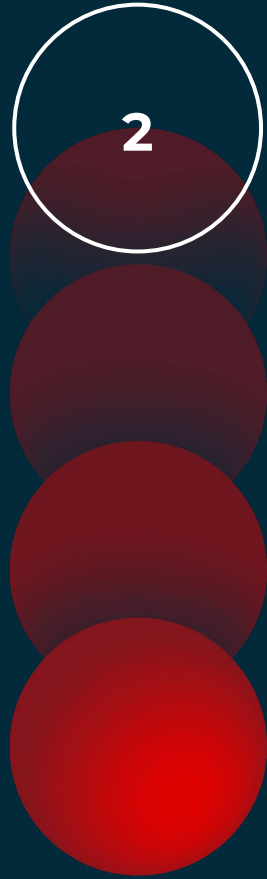
- Title:** Login Form (with a close button 'x' in the top right corner)
- Input 1:** Email or Phone (containing 'mariano@jojo.com')
- Input 2:** Password (containing '*****')
- Link:** Forgot Password?
- Button:** LOGIN (with a blue-to-purple gradient)
- Text:** Not a member? Signup now

Diferencias

Idea general

El **frontend recolecta** los **datos** de entrada del usuario, **los transforma** a las especificaciones que demanda el **backend**, y así **poder procesarlos, devolviendo** finalmente esa **respuesta al frontend**, devolviendola al **usuario para que pueda ver esa respuesta** de forma coherente.





¿Qué es una aplicación?

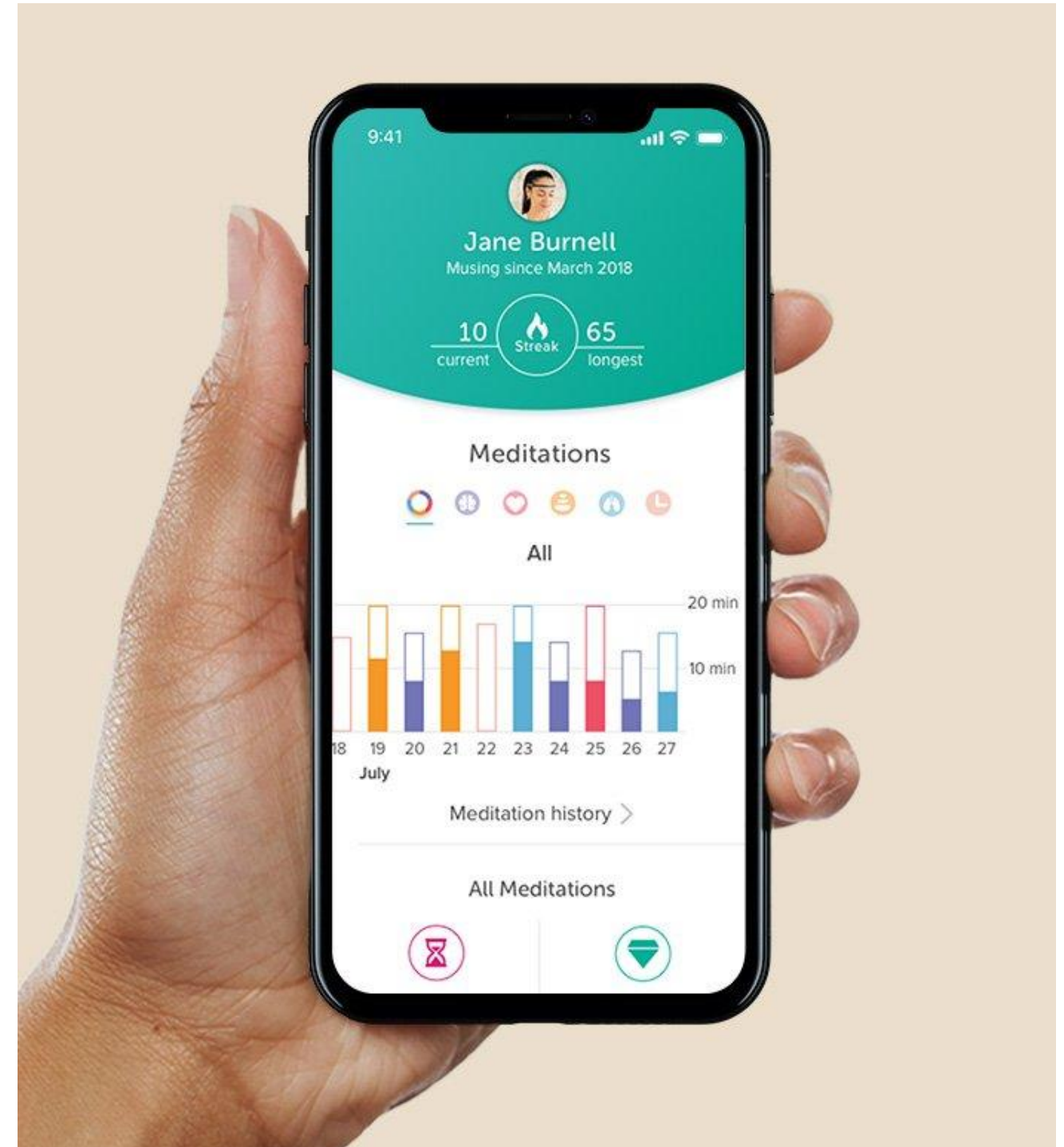
Aplicación

¿Qué es una aplicación?

Antes de continuar con la introducción al backend deberíamos de saber **qué es una aplicación**.

Según wikipedia, es *“un tipo de software diseñado para realizar un grupo de funciones, tareas o actividades coordinadas para el beneficio del usuario.”*

En mi jerga bonita toda enriquecida, *“una aplicación es algo (una aplicación) que hace algo (conjunto de instrucciones).”*



Aplicación

¿Qué es una aplicación?

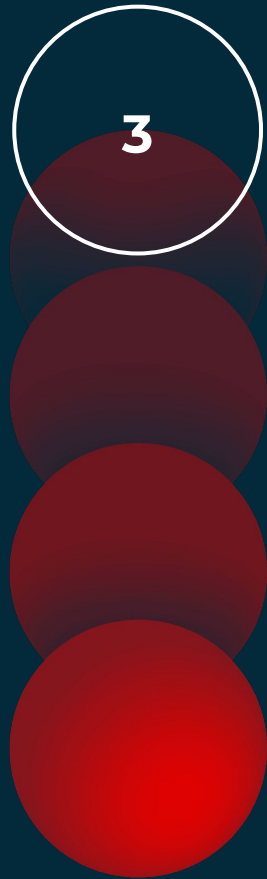
Una aplicación suele estar presentada por capas, que no son requisitos pero suelen tener la siguiente estructura:

- **Datos de entrada.** En este caso **dos** números **enteros (a, b)**, que se reciben por consola
- **Lógica de negocio.** En este caso **resultado = a * b**
- **Resultado.** En este caso, se devuelve por la consola el resultado.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      { int a, b;
7
8      { cin >> a >> b;
9
10     { int resultado = a * b;
11
12     { cout << resultado;
13     }
```

Pequeño ejemplo en C++

HORA DE LA POLENTA.



¿Qué es una API?

¿Qué es una API?

El término **API** es una abreviatura de **Application Programming Interfaces**, que en español significa interfaz de programación de aplicaciones.

Una **API** es una “**aplicación expuesta**”, para que podamos permitir que sea **consumida por otra aplicación**, la **comunicación** con una **API** suele estar compuesta por **datos de entrada**, y **datos de salida**.

Nuevamente en mi jerga toda estandarizada, **es el nexo para que dos aplicaciones se puedan comunicar e intercambiar información**.

1. Una aplicación externa quiere utilizar nuestra lógica de negocios, entonces deben llamar a nuestra API



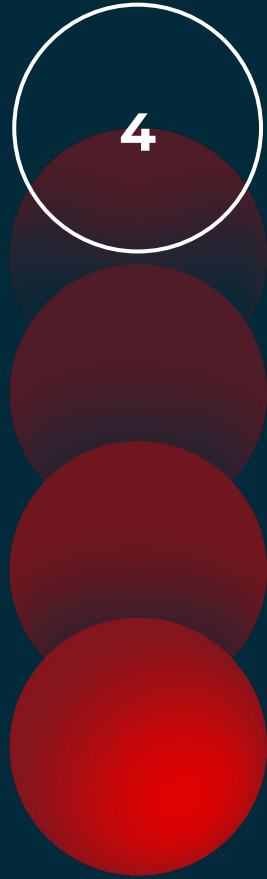
2. Nuestra API, reenvía esos datos a nuestra lógica de negocio (es decir, nuestra aplicación)



3. Nos devuelve (o no) una información.



MÁS COMPLEJIDAD...



¿Qué es una REST API?

REST API

—

¿Qué es una REST API?

La internet define a una **REST API** como un **servicio**, que no es una arquitectura software, sino un **conjunto de restricciones**, la cual podremos usar para crear aplicaciones web respetando el protocolo HTTP.



REST API

—

¿Qué es una REST API?

La internet define a una **REST API** como un **servicio**, que no es una arquitectura software, sino un **conjunto de restricciones**, la cual podremos usar para crear aplicaciones web respetando el protocolo HTTP.

Ni yo entendí esa explicación, así que nuevamente les traigo mi jerga ultra avanzada para esta generación.

Una **REST API** es una **API** que se **expone** para que pueda ser **consumida** por **internet**.

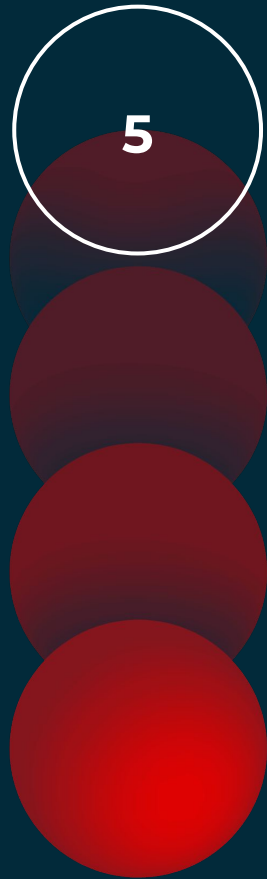
Para realizar esto último, se **debe de utilizar** un **protocolo de comunicación**. Este **protocolo** es el **HTTP**.

A modo de ejemplo, podríamos considerar que la **API** es la **boca, ojos y los oídos**, mientras que **REST** podría ser el **idioma** que estamos utilizando para **comunicarnos**.





DUDAS



Proyecto empresarial

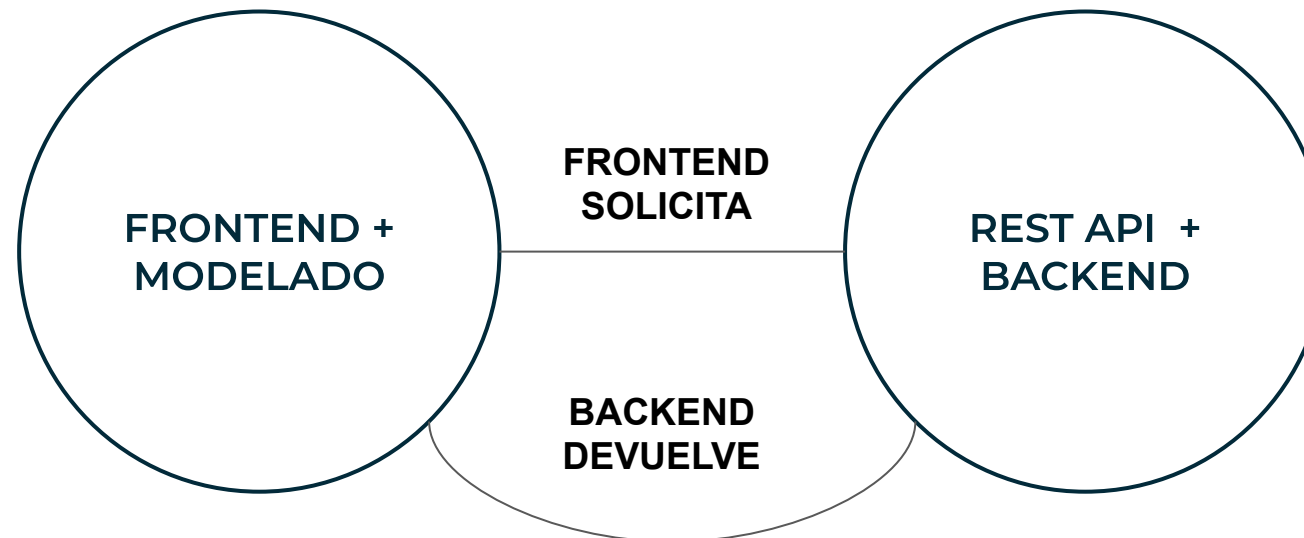
Proyecto empresarial

—

Ejemplo práctico

Queremos dar ejemplos reales de cómo es trabajar como desarrollador backend. Y poder visibilizar la necesidad de separar backend y frontend. Así que daremos un ejemplo práctico de la mano de Tiago.

Si necesitáramos transferir plata entre usuarios, pero estos usuarios tienen datos sensibles, es indispensable contar con un backend para que el usuario final no vea información que no debería de tener acceso.





BREAK

flexibility

MUCHAS
GRACIAS

—

