



**ITGEST ACADEMY**

Projeto de Desenvolvimento



## Versões do documento

<b>Versão</b>	<b>Data</b>	<b>Autores</b>	<b>Principais alterações</b>
1.0	14/06/2021 a 17/06/2021	Cristiana Pereira	Versão inicial
1.1	18/16/2021	Cristiana Pereira	Correcção da BD

*Preencher esta tabela com o seu histórico de versões, se for o caso. Fazer um breve resumo das alterações realizadas em cada versão.*

## Índice

<b>1. Introdução .....</b>	<b>3</b>
<b>1.1. Enquadramento.....</b>	<b>3</b>
<b>1.2. Estrutura do Documento .....</b>	<b>3</b>
<b>2. O Projeto .....</b>	<b>4</b>
<b>2.1. Objectivos .....</b>	<b>4</b>
<b>2.2. Requisitos Funcionais.....</b>	<b>4</b>
<b>2.3. Pressupostos .....</b>	<b>5</b>
<b>2.4. Diagrama de arquitetura do projeto .....</b>	<b>5</b>
<b>Erro! Marcador não definido.</b>	
<b>2.5 Atores .....</b>	<b>6</b>

## 1. Introdução

### 1.1. Enquadramento

O presente documento tem como principal objetivo sistematizar toda a informação funcional e técnica que irá suportar a implementação do Projeto de Desenvolvimento.

### 1.2. Estrutura do Documento

O presente documento encontra-se estruturado por processos e dividido com base nas várias etapas que serão consideradas no âmbito da implementação.

Por cada processo consideram-se os seguintes capítulos:

- **Objetivos** – Breve descrição do projeto, as suas respetivas características;  
O que se pretende com o projeto de forma geral? O que pretende alcançar com o mesmo?
- **Requisitos Funcionais** – Lista de funcionalidades a serem implementadas face às necessidades;
- **Pressupostos** – Breve descrição de eventos ou circunstâncias que se espera que ocorram durante o ciclo de vida do projeto. Determinar as premissas, restrições e riscos que o projeto poderá vir a ter;
- **Diagrama da arquitetura** – Ilustração Frontend, API, base de dados, como se relacionam entre si. Ilustração geral do projeto;
- **Atores** – As pessoas que interagem com o sistema (por exemplo: consumidor final, backoffice, etc).

## 2. O Projeto

### 2.1. Objectivos

O meu projecto final para concluir a academia é a realização de uma *Web App* intitulada *BooketList*.

Tem como foco principal uma *readlist* que possibilitará ao *user* criar e consultar a sua lista personalizada de livros que quer ler ao longo da sua vida.

Com esta *app* procuro resolver problemas muito comuns que os leitores se deparam como, por exemplo, comprar livros repetidos por o *user* não se recordar se já tem o livro em questão, principalmente quando são do mesmo escritor e a capa altera-se por ter sido adaptado para filme; ter de decorar uma panóplia de títulos (ou recorrer ao papel já amarrotado e sem tinta dos livros que tem); ter a data correcta de quando iniciou e de quando terminou a leitura; entre outros.

A aplicação será construída recorrendo a ferramentas usadas e leccionadas no decorrer da formação e irá ser dividido em duas componentes principais: *FrontEnd* e *BackEnd*.

O *FrontEnd* será desenhado recorrendo a HTML, CSS e desenvolvido através da *framework* Vue.js. Poderão também ser usadas ferramentas auxiliares como *Bootstrap* e *Vuetify*.

Quanto ao *BackEnd* vai ser criado em ambiente *Node.js*, que irá comunicar com uma API externa para aceder à informação de livros, tanto por título, como por escritor. Os dados obtidos da API serão utilizados para alimentar uma base de dados (Fig. 1).

### 2.2. Requisitos Funcionais

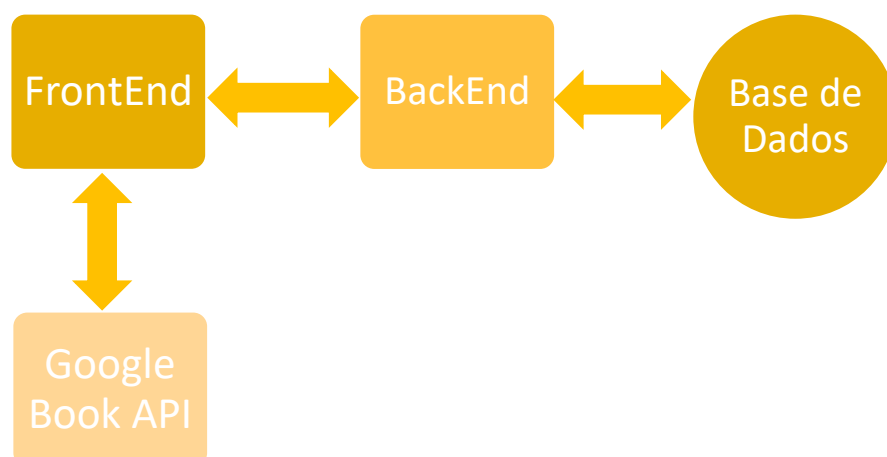
- Base de dados para armazenar a informação proveniente da API e do *user*;
- Login/autenticação;
- Uso de API externa (*Google Books API*, em princípio);
- *Input* para pesquisar a lista de livros existentes (permitir a pesquisa por título e por escritor);
- Funcionalidade para colocar o livro como já lido;

- Apresentar/organizar, em tabs, a lista de livros por ler da lista de livros já lidos;
- Gráfico com a percentagem de livros da lista que ainda faltam ler. Ter comentários como: *'Almost there', 'Keep going'...* (*Nice to have*);
- Enviar *email/pop-up/alerta* com novos livros de um escritor favorito que tenha sido lançado (*Nice to have*);
- Campo de avaliação pessoal com *rating* e *textarea* para adicionar comentários;
- Permitir partilhar a 'minha' lista de livros com outros utilizadores (*Nice to have*);
- Componente *Dashboard* com o gráfico e com o tempo estimado de leitura de um livro e/ou quantas horas leu naquela semana/mês/ou até mesmo desde que iniciou o *login* na aplicação (*Nice to have*);
- *Loadings*.

### 2.3. Pressupostos

Provavelmente outras falhas irão surgir, mas, de momento, e uma vez que irei recorrer a uma API externa para obter os dados que irei apresentar ao *user*, uns dos problemas que poderei encontrar é quando o livro ou o escritor são muito recentes ou pouco conhecidos. Como solução para o problema, pensei em colocar uma mensagem de erro (*'404 – Book not found'*) para dar algum feedback ao utilizador.

### 2.4. Diagrama de arquitetura do projeto



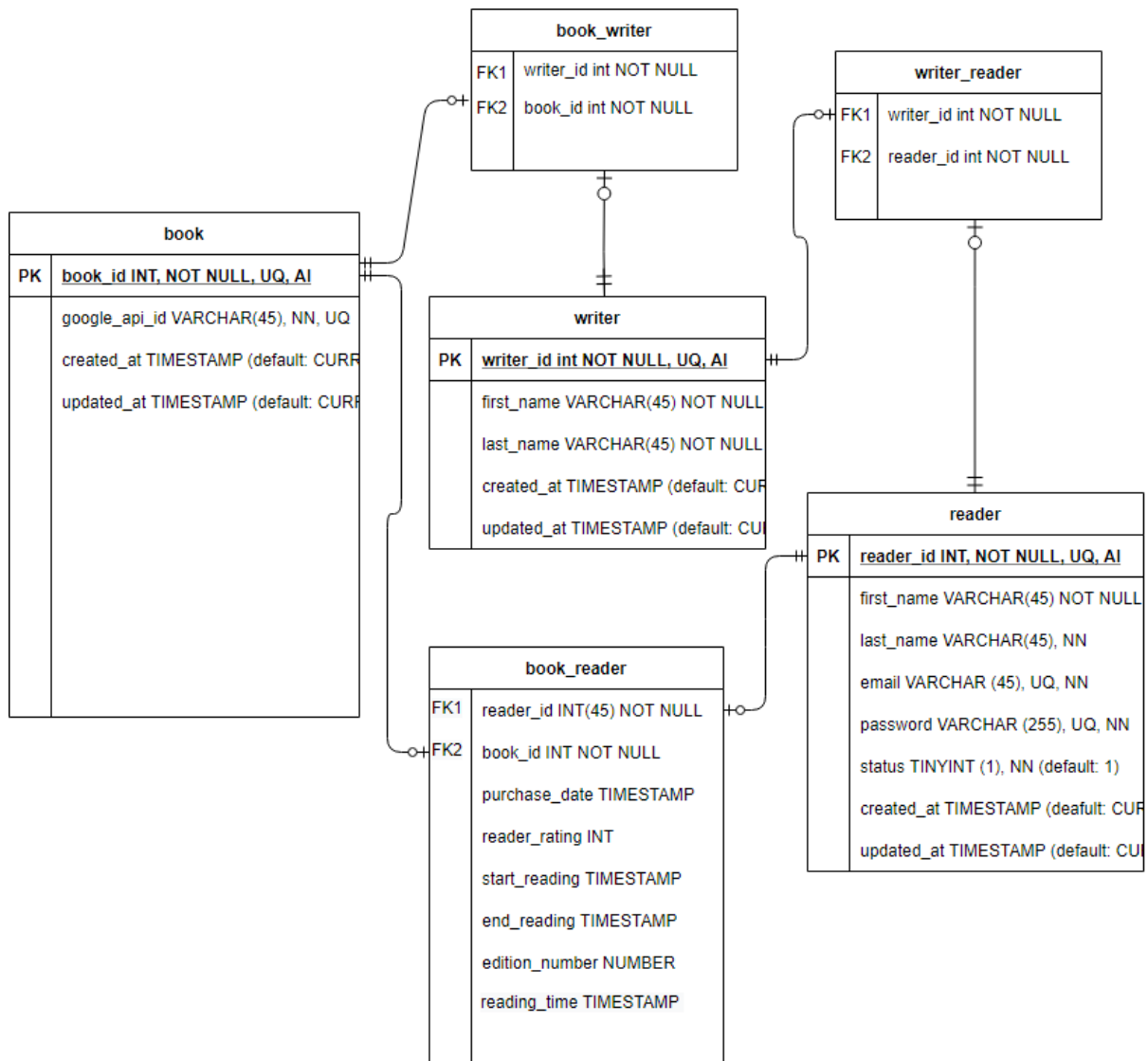


Fig.1 Diagrama da Base de Dados

## 2.5 Atores

Os actores deste projecto serão os utilizadores. Irão fazer o login, procurar por livro ou escritor, recorrendo à API externa.

Os detalhes da relação livro-leitor serão armazenados na base de dados.

A informação do livro guardado será pelo *id* correspondente na API Google Books API.