**UNIVERSIDADE ANHEMBI MORUMBI**

*Cinthya Vieira (cinthyavieira0@gmail.com)*

*Gabriel Ramos (gabrielsomar@gmail.com)*

*Igor Pompeo (pompbass@gmail.com)*

*Thiago Basilio (basilio.barboza@gmail.com)*

*Victor Hugo Garcia Tosta(victorgtosta@gmail.com)*

**SISTEMA DE ANÁLISE GRÁFICA DOS TESTES DE PROGRESSO**

São Paulo

2019

[RESUMO 3](#_Toc11607706)

[Introdução 4](#_Toc11607707)

[1. Referencial Teórico 5](#_Toc11607708)

[**1.1** **Charts JS** 5](#_Toc11607709)

[**1.2**  **Angular** 5](#_Toc11607710)

[**1.3** **Node.js** 5](#_Toc11607711)

[**1.4** **MariaDB** 6](#_Toc11607712)

[2. Metodologia de trabalho 6](#_Toc11607713)

[3. Desenvolvimento do sistema 7](#_Toc11607714)

[**3.1** **Levantamento dos requisitos funcionais/não funcionais/RN** 7](#_Toc11607715)

[3.1.1 Requisitos funcionais 7](#_Toc11607716)

[3.1.2 Requisitos não funcionais 8](#_Toc11607717)

[3.1.3 Regras de negócio 8](#_Toc11607718)

[**3.2** **Identificação das tarefas** 8](#_Toc11607719)

[**3.3**  **Levantamento dos recursos necessários para as tarefas** 9](#_Toc11607720)

[**3.4**  **Elaboração do cronograma e divisão das tarefas** 9](#_Toc11607721)

[**3.5**  **Ciclo de execução das tarefas e elaboração da documentação** 9](#_Toc11607722)

[**3.6** **Revisão e check das tarefas realizadas** 12](#_Toc11607723)

[**3.7** **Finalização das tarefas, projeto pronto em primeiro estágio** 12](#_Toc11607724)

[**3.8** **Testes finais com o produto concluído** 12](#_Toc11607725)

[**3.9** **Aprovação por parte do time e apresentação ao cliente** 13](#_Toc11607726)

[**3.10** **Termos de encerramento do projeto com a aprovação por escrito** 13](#_Toc11607727)

[**3.11** **Produto entregue e projeto finalizado** 13](#_Toc11607728)

[Conclusão e resultados obtidos 13](#_Toc11607729)

[REFERENCIAL BIBLIOGRÁFICO 15](#_Toc11607730)

# RESUMO

Este artigo aborda o desenvolvimento de um software de análise gráfica das provas de teste de progresso, o AGTP (Análise Gráfica dos Testes de Progresso). Este software foi desenvolvido com intuito de melhorar, otimizar e facilitar ao aluno e o corpo docente da Universidade Anhembi Morumbi no momento de verificar como estão os alunos de todos os cursos. Neste projeto foram observadas técnicas de análises estatísticas e implementadas junto ao conhecimento WEB e de Banco de Dados, fora iniciado o projeto AGTP, utilizando tecnologias atuais em sua composição, tais como: Angular, ChartJS, MariaDB e NodeJs. Os resultados obtidos indicam melhorias em como os usuários (alunos / corpo docente) poderão visualizar, identificar onde estão com problemas e também acompanhar o crescimento da turma com o decorrer do curso. Conclui-se que o software produzido, documentado e detalhado neste artigo, trará melhorias para ambos clientes e empresas que venham a ter contato com a solução aqui proposta através da otimização moderna das filas de atendimento de serviços.

**Palavras-chave**: Angular. NodeJs. MariaDB. Teste de Progresso. ChartJS

# Introdução

No cenário atual da Universidade Anhembi Morumbi, é entregue um PDF com as informações dos alunos e os mesmos têm o trabalho de procurar suas notas no documento.

Porém o mesmo documento é precário pois não contém a quantidade de informações que o aluno precisa saber, se foi bem ou não na própria prova, se está em um bom ranking na turma e se está com um crescimento progressivo em suas notas.

Sabendo dessa deficiência, foi desenvolvido um programa que será responsável por fazer um gráfico demonstrativo ao aluno / corpo docente responsável pelo curso / turma para visualizar, analisar e quantificar a qualidade dos seus alunos mediante suas notas.

Como parte integrante do processo, será também informado ao aluno um modo de verificação das suas notas, onde ele irá selecionar como verá suas notas, também o seu ranking contra sua turma de seu curso.

Este artigo tem como objetivo apresentar e discutir sobre o projeto do software de Análise Gráfica de Notas do Teste de Progresso (AGTP) e sua implementação, abrangendo toda a àrea teórica e operacional.

O estudo realizado está dividido em duas partes, sendo a primeira um estudo relacionado ao conteúdo teórico envolvendo os conceitos principais sobre os processos, tecnologias utilizadas e qualidade de software com foco no desenvolvimento do mesmo.

A segunda parte do estudo trata sobre como o software se comportaria através de um processo de implantação em uma empresa, um servidor MariaDB e com inúmeros acessos a ferramenta web, para melhor compreensão de como o sistema irá se comportar no dia-a-dia em um ambiente controlado e operado pelas entidades envolvidas no processo de funcionamento.

Este estudo apresenta os possíveis resultados alcançados pela empresa através de uma simulação, identificando as dificuldades da mesma e exemplificando as melhorias que foram realizadas em seu processo de desenvolvimento através da otimização dos processos e documentações, definindo os papéis e responsáveis pelas atividades relacionadas ao sistema.

Com a melhoria do processo para que o sistema se adeque aos requisitos das empresas, os resultados alcançados com a implementação e as dificuldades encontradas, servirão como fonte de informação para implantar futuras melhorias, tratando assim, da àrea de qualidade do software.

# Referencial Teórico

Para que os resultados esperados fossem obtidos, foi necessário realizar a pesquisa dos conceitos abaixo detalhados.

## **Charts JS**

Charts JS é uma ferramenta utilizada para a criação de gráficos, de uma forma mais simples onde o desenvolvedor vai apenas se preocupar com a programação dos gráficos uma única vez, pois ele se altera de acordo com os dados que serão oferecidos a ele. É possível nessa ferramenta criar gráficos otimizados, e simples, com uma das melhores estruturas oferecidas. Criado como uma biblioteca do mundo JavaScript. (ChartsJS, MundoJS)

## **Angular**

Angular (comumente referido como "Angular 2+" ou "Angular 2") é uma plataforma de aplicações web de código-fonte aberto e front-end baseado em TypeScript liderado pela Equipe Angular do Google e por uma comunidade de indivíduos e corporações. Angular é uma reescrita completa do AngularJS, feito pela mesma equipe que o construiu.

Aplicação web usando o framework Angular versão 7.1, por ser de modo Open Source dando liberdade a geração do código fonte sem qualquer proibição de IDE’s pagas ou bibliotecas exclusivas. (Angular, Wikipedia).

## **Node.js**

Node.js é um interpretador de código JavaScript com o código aberto, focado em migrar o Javascript do lado do cliente para servidores. Seu objetivo é ajudar programadores na criação de aplicações de alta escalabilidade (como um servidor web), com códigos capazes de manipular dezenas de milhares de conexões simultâneas, numa única máquina física. (NodeJS)

## **1.4 MariaDB**

MariaDB é um SGDB que surgiu como fork do MySQL, criado pelo próprio fundador do projeto após sua aquisição pela Oracle.

A intenção principal do projeto é manter uma alta fidelidade com o MySQL.

O líder do MariaDB é Michael 'Monty' Widenius, o fundador do MySQL e da Monty Program AB. Para conseguir isso, Programa Monty trabalha para contratar os melhores e mais brilhantes desenvolvedores do setor, trabalhar em estreita cooperação com a maior comunidade de usuários e desenvolvedores no verdadeiro espírito do software livre e open source e software de lançamento de uma forma que equilibre a previsibilidade com confiabilidade.

MariaDB é um avançado substituto para o MySQL e está disponível sob os termos da licença GPL v2.

# Metodologia de trabalho

Mostradas abaixo, estão todas as etapas que o grupo irá seguir no desenvolvimento desse projeto.

* Levantamento dos requisitos funcionais/não funcionais/RN;
* Identificação das tarefas do projeto;
* Levantamento dos recursos necessários para a execução das tarefas;
* Organização do cronograma e divisão das tarefas entre os integrantes;
* Ciclo de execução das tarefas e desenvolvimento do sistema e documentação (diagramas);
* Revisão e check das tarefas realizadas;
* Finalização das tarefas, produto pronto em primeiro estágio;
* Testes finais com o produto concluído;
* Aprovação por parte do time e apresentação para o cliente;
* Termos de encerramento do projeto com a aprovação por escrito do cliente;
* Projeto entregue e concluído.

# Desenvolvimento do sistema

## **Levantamento dos requisitos funcionais/não funcionais/RN**

Nessa etapa preliminar, o grupo fez o levantamento de requisitos, começando pelos funcionais. Estes são tudo que os alunos gostariam de ter em um sistema desse tipo, uma vez que também são alunos, os integrantes do grupo contribuíram com suas opiniões a esse respeito. A etapa de requisitos não funcionais foi feita principalmente em cima das tecnologias que o grupo tem algum prévio conhecimento sobre. Já os RN (requisitos de negócios) são algumas regras especiais que o time julgou necessário para o projeto.

### 3.1.1 Requisitos funcionais

A seguir são apresentados os requisitos funcionais do sistema AGTP, na **tabela 1**.

|  |  |
| --- | --- |
| RF01 | O sistema deve ter uma autenticação de usuários com login. |
| RF02 | O sistema deve apenas exibir as notas pertencentes ao aluno logado. |
| RF03 | O sistema deve exibir as notas para o usuário de acordo com o semestre. |
| RF04 | O sistema deve exibir as notas de acordo com as matérias do TP |
| RF05 | O sistema deve exibir na tela inicial a nota do semestre atual  daquele aluno. |
| RF06 | O sistema deve possuir um login especial para os docentes  da universidade. |
| RF07 | O sistema deve exibir para o usuário docente de todos os  alunos por turma. |
| RF08 | O sistema deve exibir para o usuário docente uma busca por  turma e/ou RA. |
| RF09 | O sistema deve exibir para todos os usuários um filtro por notas. |
| RF10 | O sistema deve exibir para todos os usuários um filtro por semestre. |
| RF11 | O sistema deve exibir para todos os usuários um filtro por matérias. |
| RF12 | O sistema deve exibir para usuários docentes uma classificação das  turmas de acordo com as notas do teste de progresso. |

**Tabela 1 -**  **Requisitos funcionais**

### 3.1.2 Requisitos não funcionais

Na **tabela 2** são apresentados os requisitos não funcionais do sistema AGTP

|  |  |
| --- | --- |
| RNF01 | A programação do sistema será feita em linguagem NodeJS. |
| RNF02 | O banco de dados do sistema será feito em MariaDB. |
| RNF04 | O Web server do sistema usará Angular. |
| RNF05 | O sistema terá uma disponibilidade de ao menos 95% de uptime por mês. |
| RNF06 | A senha dos usuários será criptografada em Sha1. |

**Tabela 2** – **Requisitos não Funcionais**

### 3.1.3 Regras de negócio

A seguir, na **tabela 3**, são apresentados as regras de negócio do sistema AGTP.

|  |  |
| --- | --- |
| RN01 | A autenticação do sistema será feita pelo RA do aluno/docente. |
| RN02 | O aluno deverá estar matriculado na Universidade para ter  acesso ao sistema. |
| RN03 | O docente deve estar trabalhando na universidade para ter  acesso ao sistema. |
| RN04 | O aluno deve ter feito pelo menos um teste de progresso para ter  acesso ao sistema. |

**Tabela 3** – **Regras de negócio**

## **Identificação das tarefas**

Com todos os requistos em mãos, é possível elaborar todas as tarefas necessárias para o andamento do projeto.

## **Levantamento dos recursos necessários para as tarefas**

Com todas as tarefas definidas, pode-se então ter uma noção das tecnologias e recursos que serão requeridos para o projeto. Dentre alguns deles há: Angular, MariaDB, Node.js.

## **Elaboração do cronograma e divisão das tarefas**

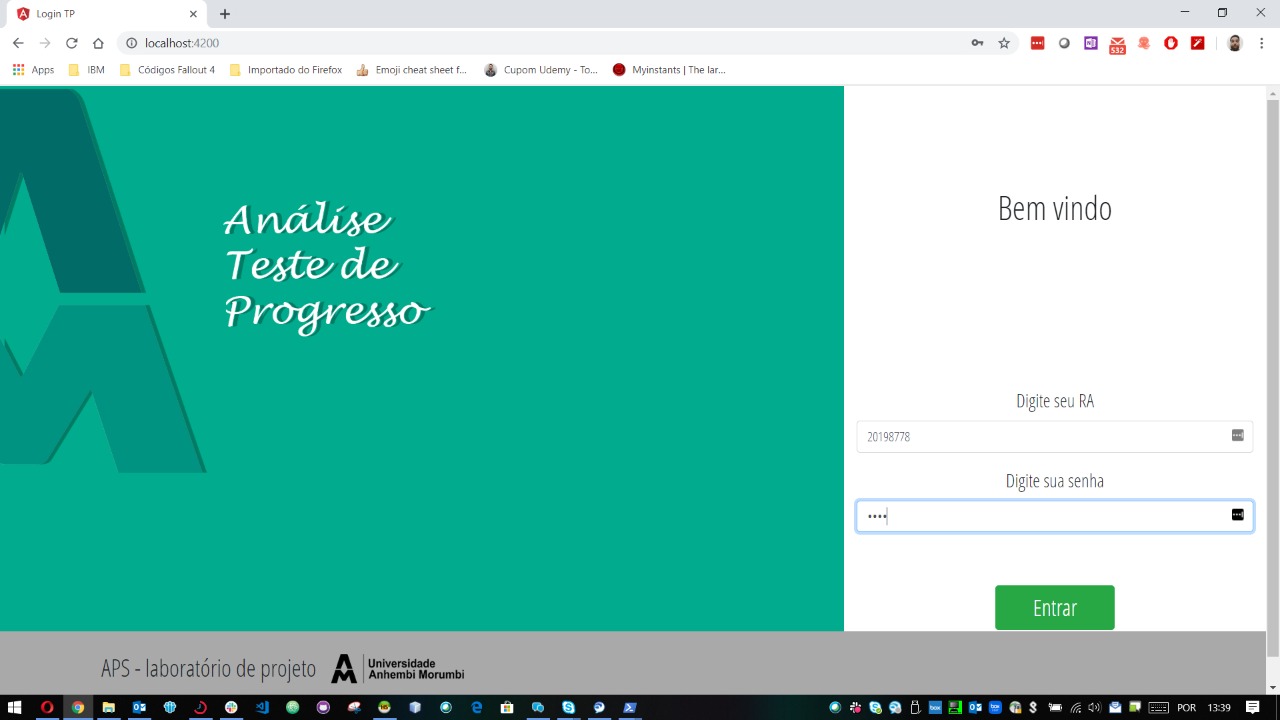
Como próximo passo, serão divididas todas as tarefas de maneira igualitária entre os integrantes do grupo, bem como definir todas as datas de entrega de cada uma delas.

## **Ciclo de execução das tarefas e elaboração da documentação**

Na parte mais longa de todo o processo, todas as tarefas previamente definidas são executadas e, idealmente, completadas dentro do prazo.

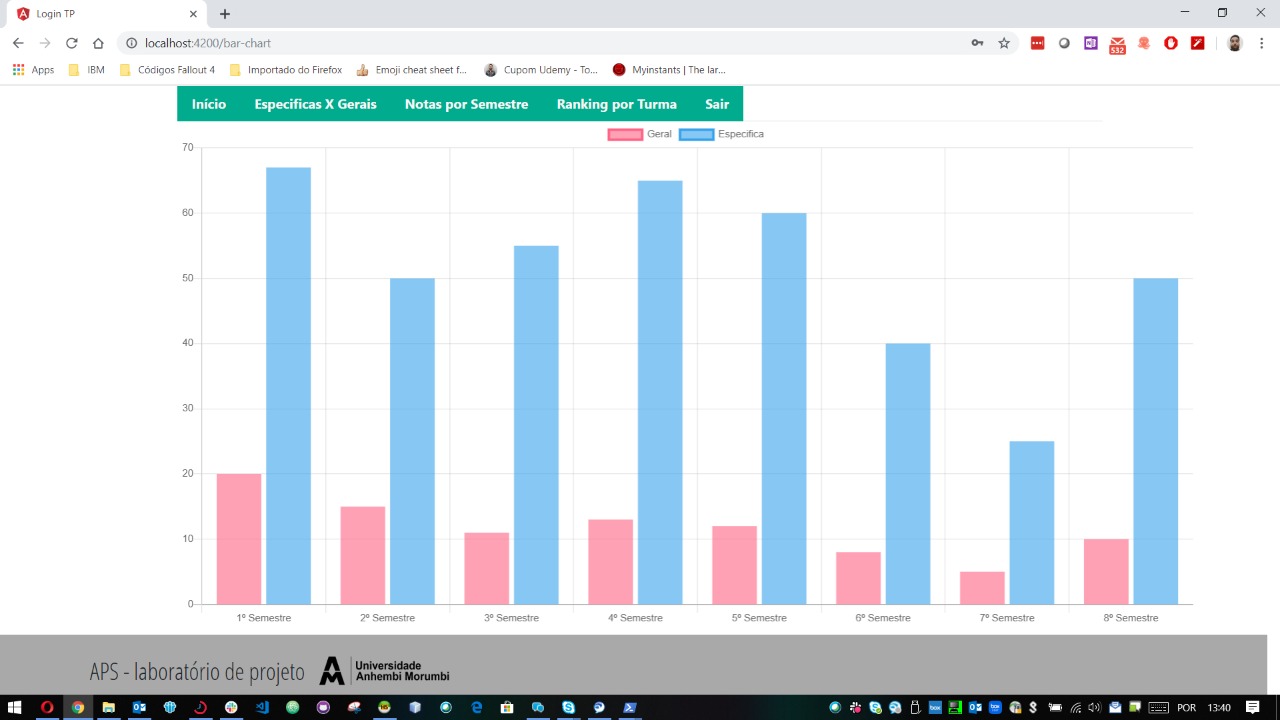
Nessa parte serão empregadas as tecnologias tais como **Angular, NodeJS ChartJS** e **MariaDB**.

* Sobre os detalhes de cada uma e como serão empregadas:
* Angular será o framework utilizado para o desenvolvimento do nosso front-end, com ele faremos a tela de login, amostragem dos gráficos e todas as funções que iremos utilizar para efetuar as chamadas do banco.
* O login em particular é bastante importante nessa aplicação devido às duas categorias de usuários existentes, Alunos e Docentes. Uma das funções do Angular será identificar por meio dos dados do Login do usuário à qual classe ele pertence. Feito isso, serão usadas funções programadas para definir a visão das tabelas que cada usuário deve ter (Ver **figura 1** abaixo).



**Figura 1 - Tela de Login**

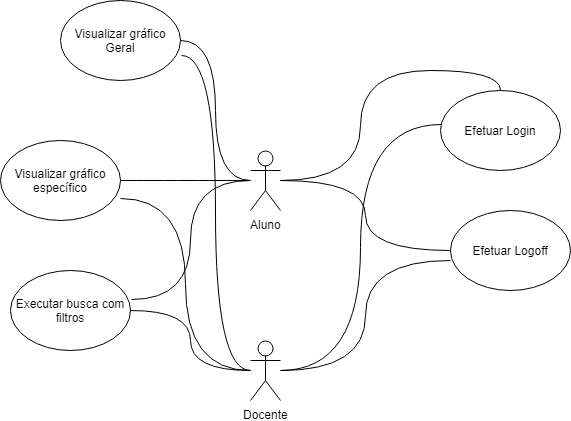
* Tendo seu login certificado e validado, o sistema irá fazer um requerimento no banco de dados para aquele RA e trará os dados pertinentes a ele. Esse trabalho é feito em conjunto com a biblioteca de gráficos chartJS, o banco de dados de onde ele usa as informações e por fim Angular para exibir as informações ao usuário (Ver **figura 2** abaixo).



**Figura 2 - Tela de gráficos**

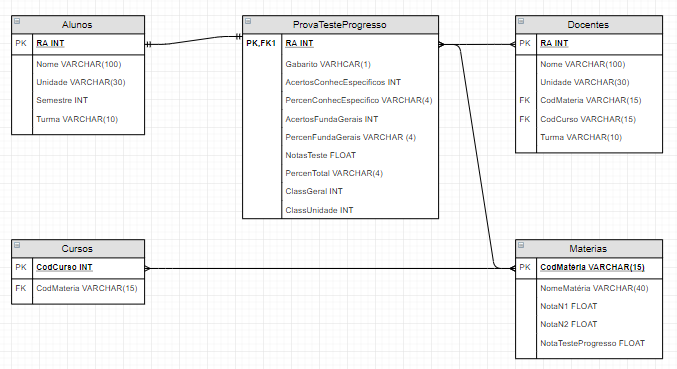
* A **figura 3** apresenta o diagrama de caso de uso do sistema AGTP.

Nela são exibidas todas as funcionalidades que estarão disponíveis para aquele determinado “ator”, tanto um aluno quanto um docente. Pode-se constatar que os dois possuem o mesmo acesso a todas as funções, o que muda é o que cada um poderá fazer dentro delas, em especial o Docente terá uma visão muito mais dinâmica e sem restrições dos dados que o programa pode apresentar, enquanto o aluno terá apenas visão do que diz respeito à ele mesmo.



**Figura 3 – Diagrama de Caso de Uso**

* Na **figura 4,** pode-se visualizar o diagrama de entidade relacional, que define como as tabelas estarão sendo relacionadas no banco de dados.



**Figura 4 - Diagrama Entidade Relacional**

## **Revisão e check das tarefas realizadas**

Esse item vem depois do 3.5 mas os dois serão executados de forma simultânea, com o fim de agilizar a detecção de erros/issues com a finalização de cada tarefa individualmente. Através dos erros encontrados foram criadas issues para a resolução, de uma forma que o sistema pudesse ser testado e executado.

## **Finalização das tarefas, projeto pronto em primeiro estágio**

O produto se encontra em um estágio “alpha”, em que foi feito um check superficial das tarefas durante o passo anterior, e com essa será feito o mesmo de uma forma mais profunda, para a correção de bugs e problemas que foram encontrados, desta forma o sistema está sujeito à mudanças e alterações para melhores resultados.

## **Testes finais com o produto concluído**

Finalmente os testes gerais e finais são executados, nesta etapa não espera-se muitas surpresas em relação à funcionalidades defeituosas, mas como mencionado previamente, qualquer erro ainda pode ser reparado antes da apresentação para o cliente.

## **Aprovação por parte do time e apresentação ao cliente**

Nessa etapa foi acordado que a qualidade do projeto, se encontra suficiente para ser apresentado perante o cliente, demonstrando todas as suas funcionalidades.

## **Termos de encerramento do projeto com a aprovação por escrito**

Elaboração e preenchimento dos documentos de encerramento, onde o cliente declara que está satisfeito com o produto entregue e que os requisitos foram atendidos.

## **Produto entregue e projeto finalizado**

Neste momento se declara que não serão feitas mais mudanças requisitadas pelo Cliente e que o projeto chegou ao fim e será arquivado. Qualquer mudança solicitada daqui em diante será tratada como um projeto novo.

# Conclusão e resultados obtidos

A conclusão é que o projeto de software *Análise Gráfica dos Testes de Progresso* foi um excelente aprendizado no desenvolvimento *hands-on* de um projeto “real”, desde a fase de elaboração da ideia principal do projeto até a entrega do produto funcional. Embora algumas dificuldades tenham sido enfrentadas, principalmente devido à incorporação de ferramentas novas ao grupo, foi concluído todas as entregas definadas no cronograma de modo geral.

O plano de desenvolvimento inicial era um software que faria a integração entre Angular, SQLServer e Power BI para a interface, banco e manipulação dos dados respectivamente. Conforme o projeto foi evoluindo, começaram a surgir incertezas sobre se essas três ferramentas realmente conversariam entre si da forma esperada.

A interface gráfica com a funcionalidade de login foi o primeiro item a ser realizado, desenvolvido com a ferramenta bootstrap e elementos comuns de desenvolvimento front-end. Ao mesmo tempo, a elaboração conceitual do banco de dados estava em andamento, onde foi realizado a inclusão dos dados vindo de uma extração do Excel que foi encaminhado pela coordenação da instituição Universidade Anhembi Morumbi.

Por fim na terceira frente, utilizando o Power BI para criação dos gráficos, conseguiu-se realizar o cruzamento dos dados que foram propostos.

No final com as três tarefas principais feitas, verificou-se que houveram incompatibilidades na junção das ferramentas, onde concluiu-se que seria inviável a utilização de Power BI, que por fim foi substituído pelo Chart.js que apresentou uma integração mais eficaz.

Sobre o produto, foi obtido um resultado final que condiz com o que foi proposto na fase de concepção do projeto. O grupo acredita que essa ferramenta tem um grande potencial em ser utilizada pela Universidade Anhembi Morumbi, principalmente na funcionalidade analítica que ela oferece aos alunos em conseguirem visualizar suas métricas em relação às notas obtidas nas provas chamadas Teste de Progresso.

* Possibilidades futuras

Como oportunidades futuras para acrescentar à esse projeto, seria possível um sistema parecido para as notas de provas N1/N2/Projetos que traria o mesmo benefício analítico para tanto professores, quanto para alunos. Para o escopo gerencial/administrativo, pode ser possível trabalhar algo em cima do CPA/ISED, trazendo uma plataforma própria para a aplicação e análise dos dados em tempo real e sem custos externos como se tem hoje. Por fim, uma função mais voltada aos alunos, o acompanhamento e aconselhamento no decorrer do semestre. Assim, apontando entregas futuras, faltas por matéria, situação da média atual, provas e todas as informações que hoje não são obtidas com muita facilidade, que são pertinentes aos alunos.

# REFERENCIAL BIBLIOGRÁFICO

BANKS, A., PORCELLO, E. **Learning React.** 1 ed. United States of America, 2017.

Sobre o Electron. Disponível em <https://electronjs.org/docs/tutorial/about>. Acessado em 05/06/2019.

NODE, Sobre o Node.js. Disponível em <https://nodejs.org/en/about/>. Acessado em 05/06/2019.

NODEJS, Guias. Disponível em <https://nodejs.org/en/docs/guides/>. Acessado em 05/06/2019

ENTITY, Sobre o EntityFramework. Disponível em <https://docs.microsoft.com/pt-br/dotnet/framework/data/adonet/ef/overview>. Acessado em 12/06/2019

OAUTH2, Uso do Oauth2. Disponível em [https://oauth.net/2/3. Acessado em 12/06/201](https://oauth.net/2/3.%20Acessado%20em%2012/06/2018)9.

ANGULAR. Disponível em <https://pt.wikipedia.org/wiki/Angular_(plataforma_de_aplica%C3%A7%C3%B5es)#Vers%C3%A3o_7.1>. Acesso em 04/05/2019 às 20:36.

MUNDOJS. Disponível em <https://www.mundojs.com.br/2018/01/18/iniciando-com-chart-js/#comment-3>. Acesso em 27/05/2019 às 20:00.

CHARTJS, material de apoio Disponível em

<https://tobiasahlin.com/blog/introduction-to-chartjs/>. Acesso em 25/05/2019 às 19:12.

<https://www.chartjs.org/docs/latest/>. Acesso em 25/05/2019 às 18:00.

MARIADB. Disponível em https://pt.wikipedia.org/wiki/MariaDB. Acessado em 27/05/2019 às 18:00.