



FACULDADE DE TECNOLOGIA DE SOROCABA
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

RELATÓRIO FINAL DE INICIAÇÃO CIENTÍFICA

Vítor Manfré Dias

TestIt – Aplicação Web para Criação e Realização de Avaliações Online
(Desenvolvimento Ambiente Aluno)

Sorocaba
Dezembro/2017



Projeto de Desenvolvimento

TestIt – Aplicação Web para Criação e Realização de Avaliações Online (Desenvolvimento Ambiente Aluno)

Vitor Manfré Dias

Orientadora: Prof^a M^a Denilce de Almeida Oliveira Veloso

Sorocaba
Dezembro/2017

RESUMO

O projeto *TestIt* foi criado pensando em uma aplicação que visa criar um ambiente em que tanto professor quanto aluno tenham uma boa experiência no momento de criar e realizar provas de uma maneira mais simples, intuitiva e moderna. Visa deixar o processo de criar uma prova mais fácil, pensando na experiência do usuário e também nas interações dele com o sistema *web*. Além disso, o sistema oferecerá facilidades ao professor no momento de correção oferecendo um percentual de quanto o aluno acertou a questão dissertativa, que é uma das maiores inovações do *TestIt*. Este projeto em particular consiste em realizar o Desenvolvimento (implementação) do Ambiente para o Aluno e Realização das provas. A documentação irá conter uma revisão bibliográfica sobre os AVA's e o ensino EAD, além de definir os materiais e métodos utilizados para a criação da aplicação e os resultados obtidos.

Palavras-Chave: Desenvolvimento Web, TestIt, .Net Core, Vue.js, JavaScript

LISTA DE FIGURAS

Figura 1 - Servidor <i>front-end</i>	18
Figura 2 – Servidor <i>back-end</i>	19
Figura 3 – Banco de Dados.....	20
Figura 4 – Arquitetura Front-End.....	21
Figura 5 – Arquitetura Back-End	22
Figura 6 – Tela Login	23
Figura 7 – Concluir Cadastro.....	24
Figura 8 – Home.....	25
Figura 9 – Minha turma minimizada	26
Figura 10 – Minha turma maximizada	26
Figura 11 – Realizar prova dissertativa	27
Figura 12 – Realizar prova alternativa.....	28

SUMÁRIO

1	INTRODUÇÃO E JUSTIFICATIVA.....	8
2	REVISÃO BIBLIOGRÁFICA.....	10
2.1	EaD	10
2.2	AVA.....	10
2.2.1	MOODLE	11
2.2.2	SOLAR	12
2.2.3	ALURA.....	12
2.2.4	MEDSOFT.....	13
2.3	Ferramentas de Desenvolvimento	13
2.3.1	Vue.js.....	13
2.3.2	HTML	13
2.3.3	.NET Framework.....	14
2.3.4	Azure	15
2.3.5	GIT	15
2.3.6	SQL Server	15
3	OBJETIVOS.....	16
4	MATERIAIS E MÉTODOS	17
5	RESULTADOS E DISCUSSÕES	18
6	CONCLUSÃO	31
7	SUGESTÕES PARA TRABALHOS FUTUROS.....	32
	REFERÊNCIAS.....	33
	GLOSSÁRIO.....	35

1 INTRODUÇÃO E JUSTIFICATIVA

A Educação a Distância (EAD) é uma modalidade que tem ao longo da história se transformado e com isso se colocado como uma alternativa para os dias atuais. Se no início, ela se prestava apenas a atender aos interesses econômicos vigentes na sociedade, de tal forma que o ensino era meramente unilateral (ensino por correspondência), e focado na tecnologia utilizada para tal veiculação (correios, rádio), vencer a barreira da distância e qualificar massivamente a população eram seus objetivos. Hoje, porém, a educação a distância amplia-se juntamente com as novas tecnologias e agrega novos valores pedagógicos e, por isso, pode constituir-se, numa alternativa para inclusão social. (Barros;Carvalho,2017)

Como é um assunto relativamente novo e passível de regulamentação, foi publicado no Diário Oficial de 26 de maio de 2017 o Decreto 9.057 de 25 de maio de 2017 (DECRETO9057,2017) que representa o novo marco regulatório para a educação à distância no Brasil. Esta publicação revoga o Decreto 5.622/2005 (DECRETO5622,2005) que era até então o referencial.

O avanço das tecnologias interativas tem possibilitado esse contato à distância e em tempo real. Tem surgido os Ambientes Virtuais de Aprendizagem (AVA) destacando-se como ferramentas para auxílio a professores e alunos. Como exemplo podem ser citados: Moodle(MOODLE,2017) e TelEduc(TELEDUC,2017), ambos gratuitos.

A demanda por utilização e o crescimento de ferramentas ocorreu devido as grandes vantagens que esse sistema apresenta, tanto para o estudante quanto para a instituição, como por exemplo: menor custo de formação, cursos reconhecidos pelo MEC (Ministério da Educação) e ambiente virtual de aprendizagem.

A proposta deste projeto é o desenvolvimento de uma aplicação web para criar e realizar avaliações online, uma ferramenta que venha facilitar o método de avaliação dos professores perante os seus alunos, em instituições de ensino que utilizam o ensino a distância e também nas presenciais. Conforme (Silva;Santos, 2010), "... professor e aluno constroem uma rede e não uma rota. Ela define um conjunto de territórios a explorar. E a aprendizagem e avaliação se dão na exploração..."

O desenvolvimento de uma aplicação envolve desde a Análise de Requisitos, a parte de Design e UX até chegar ao desenvolvimento do software em si. Esse projeto em especial se dedicará ao Desenvolvimento (ou implementação) de um módulo destinado a resolução das avaliações pelo aluno, incluindo também pesquisa de avaliações já realizadas e corrigidas.

2 REVISÃO BIBLIOGRÁFICA

2.1 EaD

De acordo com Belmonte (2010), as grandes mudanças tecnológicas ocorridas nas últimas décadas denominadas por Castells (1999) como a revolução da tecnologia da informação, tem influenciado todas as esferas da atividade humana, em especial a educação. Nesse contexto, sistemas de educação a distância (EaD) ganham destaque com o uso das ferramentas disponibilizadas pelas novas tecnologias para mediatizar o processo de ensino e aprendizagem.

A educação a distância pode ser compreendida como o processo planejado de ensino-aprendizagem, mediado por tecnologias no qual professores e alunos não estão fisicamente presentes em um mesmo local, mas espacial e temporalmente separados (BELLONI, 2008). Historicamente a EaD utilizou diferentes tecnologias de acordo como os recursos disponíveis em cada geração: correspondência, televisão, teleconferência ou ambiente interativo.

2.2 AVA

Com o desenvolvimento da internet e a popularização do uso do computador no dia a dia de toda sociedade praticamente, surgiram diversas ferramentas para auxiliar a criação e a oferta de cursos mediados por essas tecnologias, tais como os ambientes virtuais de aprendizagem (AVA). De acordo com (Santos, 2003), um AVA refere-se ao uso de recursos digitais de comunicação, principalmente, através de *softwares* educacionais via *Web* que reúnem diversas ferramentas de interação.

Segundo Belmonte (2010), um AVA está relacionado ao desenvolvimento de condições, estratégias e intervenções de aprendizagem em um espaço virtual na *web*, organizando de tal forma que propicie a construção de conceitos, por meio da interação entre alunos, professores e objetos de conhecimento.

Na opinião de Aguiar (2010), um AVA também se caracteriza pela presença de *softwares* educacionais via *internet*, destinados a apoiar as atividades de educação à distância. Estes *softwares* oferecem um conjunto de tecnologias de informação e comunicação, que permitem desenvolver as atividades no tempo, espaço e ritmo de cada participante.

Ambientes Virtuais de Aprendizagem são ferramentas que oferecem suporte para o gerenciamento de informações, postagem de materiais de estudo seja pelo professor ou pelo aluno e ainda através dele é possível a comunicação (síncrona e assíncrona) via fórum, *chat*, etc. Assim, permitem aos professores e alunos interação simultânea, exposição de textos e documentos de forma organizada, acesso e exposição de informações pertinentes à área do conhecimento, seja no ensino a distância ou no ensino presencial. (França,2009)

Pode então perceber que o AVA pode ser um ambiente programado, uma ferramenta capaz de dar novas possibilidades para o processo de ensino-aprendizagem como criação de espaço para aulas virtuais, disponibilidade de tempo e espaço para ampliar as discussões. Uma vez que estes ambientes permitem aos usuários discutir problemáticas, trocas individuais e coletivas de informações tornando assim um espaço de pesquisa e criação de produtos ao mesmo tempo em que se desenvolve. (França, 2009).

Os ambientes virtuais de aprendizagem permitem aos participantes fornecer informações trocar experiências, discutir problemáticas e temas de interesses comuns, desenvolverem atividades colaborativas para compreender seus problemas e buscar alternativas de solução. (Vieira,2003)

Para França (2009), os AVAs são espaços organizados e planejados que podem nos dar possibilidades de alteração de hábitos de trabalho, viabilizando o diálogo, a reflexão e o registro crítico de percursos. Representando assim não apenas recursos tecnológicos, mas ferramentas que possibilitem a construção de conhecimentos. Em muitas instituições de ensino existe o interesse na utilização dos AVAs, mas de acordo com uma das desvantagens é que o processo de implantação é demorado e complexo (Almendra, 2008).

No mercado existem várias ferramentas que podem contribuir com as instituições de ensino. Neste trabalho optou-se por explorar ferramentas que podem ser contratadas como um serviço e que embora podem não ser consideradas pela definição completamente AVAs mas oferecem ao aluno e ao professor alguma facilidade no processo de aprendizagem, assim como este projeto de iniciação científica.

2.2.1 MOODLE

O Moodle é uma plataforma de aprendizado de tecnologia, na qual a missão foi criar uma plataforma na qual todos possam discutir, participar e enriquecer seu conhecimento, investindo seu tempo da melhor maneira possível, quando e onde quiserem. O desafio é fazer o usuário se sentir parte desse grupo de profissionais de tecnologia que querem ser melhores do que eram ontem. Para isso trouxeram instrutores competentes com experiência de mercado, que conhecem sobre programação, design, infraestrutura e todas as áreas onde os profissionais de tecnologia estão ativamente envolvidos. (MOODLE, 2017).

2.2.2 SOLAR

O Solar foi feito no Brasil, é utilizado pela Universidade Federal do Ceará. O Solar é orientado ao professor e ao aluno, possibilitando a publicação de cursos e a interação com eles. Ele foi desenvolvido visando potencializar o aprendizado a partir da relação com a própria interface gráfica do ambiente, de modo que o usuário tenha acesso a várias ferramentas de aprendizagem e compartilhamento de conteúdo. O ambiente é apoiado numa filosofia de interação e não de controle, de acordo com o portal do Solar. Além disso ele é um projeto de *software* livre que pode ser utilizado por qualquer um, pois o código-fonte é aberto ao público. (SOLAR, 2017)

2.2.3 ALURA

Já o Alura (2017) com cursos atualizados e didáticos, oferece a oportunidade de melhorar as habilidades do usuário como profissional de tecnologia, sempre focando na realidade do mercado brasileiro, tomando o cuidado de inserir o contexto de todo esse domínio rico e complicado. Tem uma abordagem diferente, e oferece diversos cursos e também fóruns para a comunicação entre alunos e professores, ele dá a oportunidade de o aluno buscar o conhecimento por si só, oferecendo esse conteúdo atualizado os usuários conseguem aprender não somente o conteúdo do que é passado a eles no ensino superior, mas também ir além disso, e estudar sobre outras áreas de conhecimento, o Alura não é um *software* livre e o acesso a ele é feito através de uma assinatura anual.

2.2.4 MEDSOFT

O MEDSOFT (MEDSOFT, 2017) que de acordo com o portal do produto é uma plataforma de treinamento exclusiva aos alunos do MEDGRUPO e que permite o acesso a um banco de provas com mais de 60.000 questões comentadas em texto e vídeo. Através dos filtros disponíveis no *software*, é possível montar provas, selecionando por grande área, por concurso, por ano, por estado e até por palavra-chave. Fazendo o aluno ser livre para otimizar seu treinamento. Além de montar a própria prova, o MEDSOFT permite o acesso a centenas de questões dos simulados do MEDGRUPO dos últimos anos, provas dos concursos e de Residência Médica na Íntegra e acesso aos vídeos das apostilas.

O MEDSOFT oferece aos alunos as ferramentas necessárias para o aprendizado, porém um dos problemas é que ele é focado em apenas uma área de conhecimento que é o estudo de medicina, e ele é um *software* fechado, propriedade da empresa MEDGRUPO, e disponível apenas aos assinantes da MEDGRUPO.

2.3 Ferramentas de Desenvolvimento

2.3.1 *Vue.js*

Sobre *Vue.js* foi pesquisado (INCAU, 2017), que ensina como criar aplicações com *Vue.js*, que é um framework *JavaScript*. Trata-se de um novo *framework* para desenhar telas, que se destaca pela sua simplicidade em executar as mesmas tarefas dos outros *frameworks*. Também possui os mesmos conceitos que um *framework* reativo possui, como *data bind*, *two way*, *events*, criação de componentes, entre outros. (VUE, 2017)

2.3.2 *HTML*

HTML é uma das linguagens utilizadas para desenvolver websites. O acrônimo HTML vem do inglês e significa *Hypertext Markup Language* ou em português Linguagem de Marcação de Hipertexto.

O HTML é a linguagem base da internet. Foi criada para ser de fácil entendimento por seres humanos e também por máquinas, como por exemplo o Google ou outros sistemas que percorrem a internet capturando informação. Como define (Venetianer, 1996)

2.3.3 .NET Framework

O .NET Framework é uma plataforma de desenvolvimento da Microsoft que compreende uma máquina virtual (CLR), uma linguagem de bytecode (CIL), uma biblioteca padrão (BCL), um gerenciador de pacotes (NuGet), um compilador (Roslyn), linguagens de programação (C#, F#, VB.NET), uma ORM (*Entity Framework*), uma plataforma web (ASP.NET MVC) dentre vários outros componentes. Já o .NET Core, é um *fork* do .NET Framework. A grosso modo, é uma versão mais enxuta onde só o *core* foi aproveitado (daí o nome). Ele consiste na máquina virtual CoreCLR (versão *cross-platform* do CLR), na API CoreFX (versão *cross-platform* do BCL), no ambiente de execução otimizado para código nativo CoreRT e no utilitário de linha de comando .NET Core CLI. Ele suporta aplicações linha de comando, bibliotecas, aplicações ASP.NET Core (aplicações web) e aplicações Universal Windows Platform (UWP). É mais modular, todas suas dependências são pacotes NuGet, não é necessário que seus assemblies estejam instalados na máquina. As atualizações que antes eram distribuídas via instaladores ou atualizações do Windows agora são uma simples atualização de pacotes.

Com o .Net Core, é mais simples fazer o back-end, e com ele poderão ser aplicados *design-patterns*, que deixam o código mais “limpo” e auxiliam no desenvolvimento do back-end. Como define (Pellizzolini, 2017)

2.3.4 Azure

É um provedor de cloud da Microsoft, com ele é possível publicar de maneira simples também, pois ele consegue acessar o Github, o site onde o código fonte é feito o versionamento do código, através do qual ele consegue publicar automaticamente. (AZURE, 2017)

2.3.5 GIT

Git é um sistema de controle de versão de arquivos. Através dele é possível desenvolver projetos nos quais diversas pessoas podem contribuir simultaneamente no mesmo, editando e criando novos arquivos e permitindo que os mesmos possam existir sem o risco de suas alterações serem sobrescritas.

Se não houver um sistema de versão, duas pessoas poderão abrir o arquivo ao mesmo tempo mas causará problemas. Uma das aplicações do git é justamente essa, permitir que um arquivo possa ser editado ao mesmo tempo por pessoas diferentes. Por mais complexo que isso seja, ele tenta manter tudo em ordem para evitar problemas para os desenvolvedores.

2.3.6 SQL Server

O SQL Server é um SGBD (Sistema Gerenciador de Banco de Dados) relacional desenvolvido pela Microsoft. Criado em parceria com a SYBASE em 1988 inicialmente para a plataforma OS/2. Parceria que durou até 1994, com o lançamento da versão para Windows NT e desde então a Microsoft mantém a manutenção do produto Como um Banco de Dados. (SQL SERVER, 2017)

3 OBJETIVOS

Esse projeto em específico pretende focar o desenvolvimento da área (ou módulo) do aluno no Testit onde serão realizadas as avaliações bem como verificação dos resultados.

4 MATERIAIS E MÉTODOS

Foi realizada uma pesquisa bibliográfica para entender melhor os conceitos sobre EaD e AVA e também foi realizada uma pesquisa sobre as ferramentas que poderiam ser utilizadas nesse projeto.

Foi escolhido o Vue.Js, pois ele facilitará no desenvolvimento, porque com a criação de componentes, não será necessário tanto tempo gasto no desenvolvimento de telas que possuem partes similares. Ele permite reutilizar código já escrito uma vez, por exemplo, em uma tela de criação de usuário e uma tela de edição de usuário, os campos serão mesmos só que no primeiro caso ele deve ter todos os campos vazios e no segundo já carregará os campos preenchidos na tela. Com o *Vue.js*, e seus componentes, é possível mostrar em telas diferentes componentes iguais, e carregá-los com dados diferentes. Sendo assim, é gasto apenas o tempo de se desenvolver um componente, e não haverá repetição de código.

Além do *Vue.js*, no *front-end*, serão utilizadas, HTML5, a linguagem de marcação da web e CSS3.

Para a publicação do *webapp*, será utilizado um servidor *Node.js*, no site *heroku*, que é um provedor de *cloud computing*, com ele o sistema ficará disponível para qualquer um que tenha acesso a internet. O *heroku* facilita no *deploy* pois utiliza o git para fazer as publicações e assim não se gasta muito tempo com isso.

No *back-end* será utilizado o .NET Core.

5 RESULTADOS E DISCUSSÕES

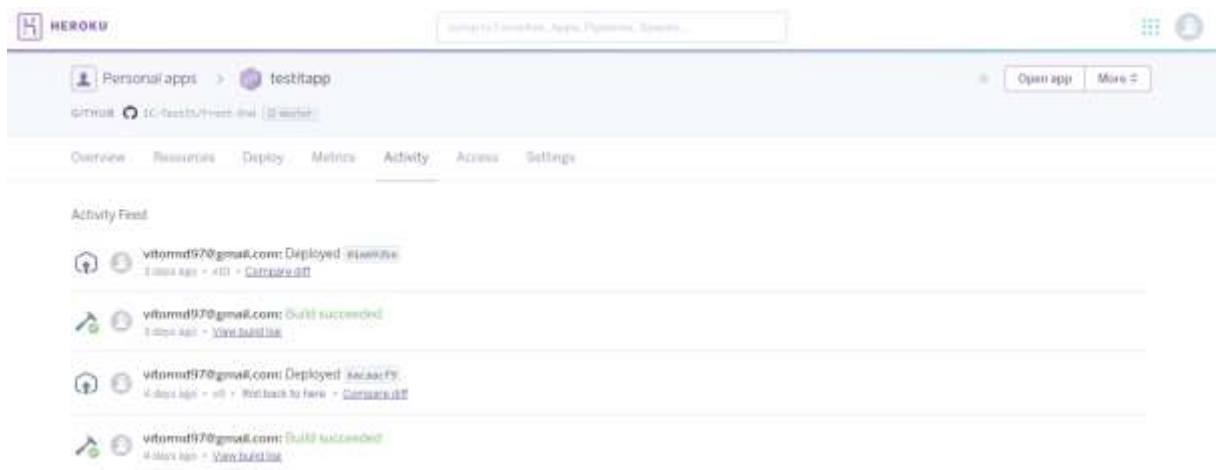
Através da revisão bibliográfica foi possível verificar que a maioria dos AVAs, são sistemas fechados a uma única instituição ou muito caros e com uma implementação cara, além disso algo que falta em todos os AVAs pesquisados é a possibilidade de se corrigir provas com questões dissertativas, item que pretende ser o diferencial na aplicação proposta.

Foi realizada a criação da arquitetura do sistema, tanto *front-end* quanto *back-end*, e também a construção do banco de dados utilizando o Sistema Gerenciador de Banco de Dados SQLServer, além disso também foi preparada a infraestrutura do projeto.

Em primeiro lugar sobre a infraestrutura, estão sendo usados dois servidores, um para o *front-end* e um para o *back-end*. O servidor do *front*, está hospedado no heroku, como pode se ver na figura 1.

5.1 Infraestrutura

Figura 1 - Servidor *front-end*

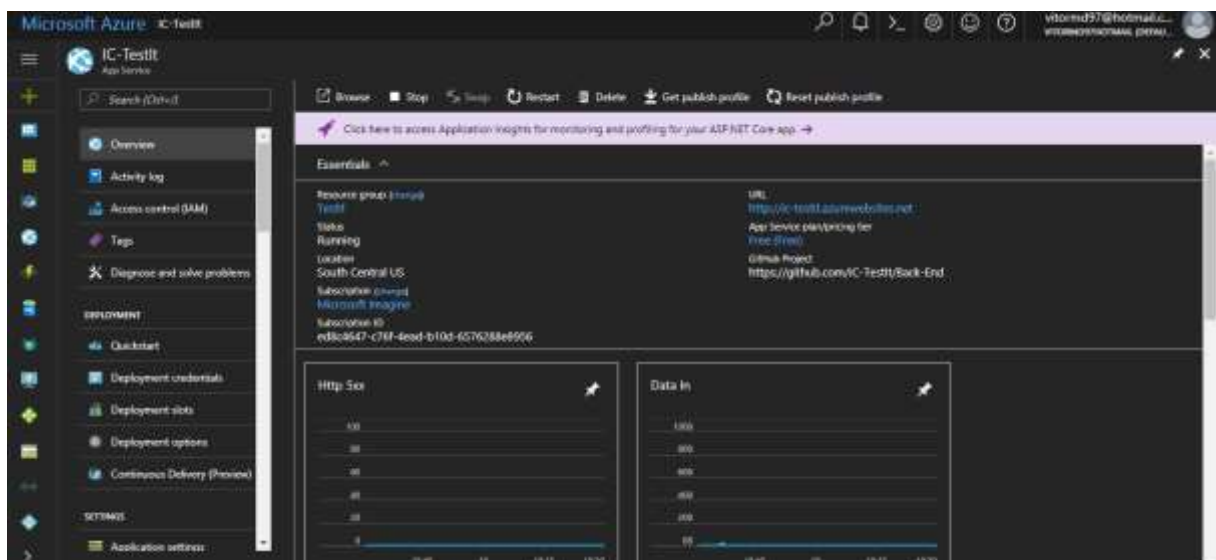


Fonte: (Autor, 2017)

Nesse servidor está implementado também o conceito de CD (*Continuous Delivery*) ou Entrega Contínua, conseguiu-se aplicar isto porque, no *heroku* associou-se o repositório do *Github* com o servidor, então toda vez que alguém faz um *commit* na *branch master*, que é onde está o código fonte de produção, o *heroku* já roda o *script* de *build*, para gerar a versão publicável do app e depois realiza o *deploy*, que nada mais é do que a publicação do app na *web*.

Depois o servidor do *back-end* foi configurado como mostrado na figura 2.

Figura 2 – Servidor *back-end*

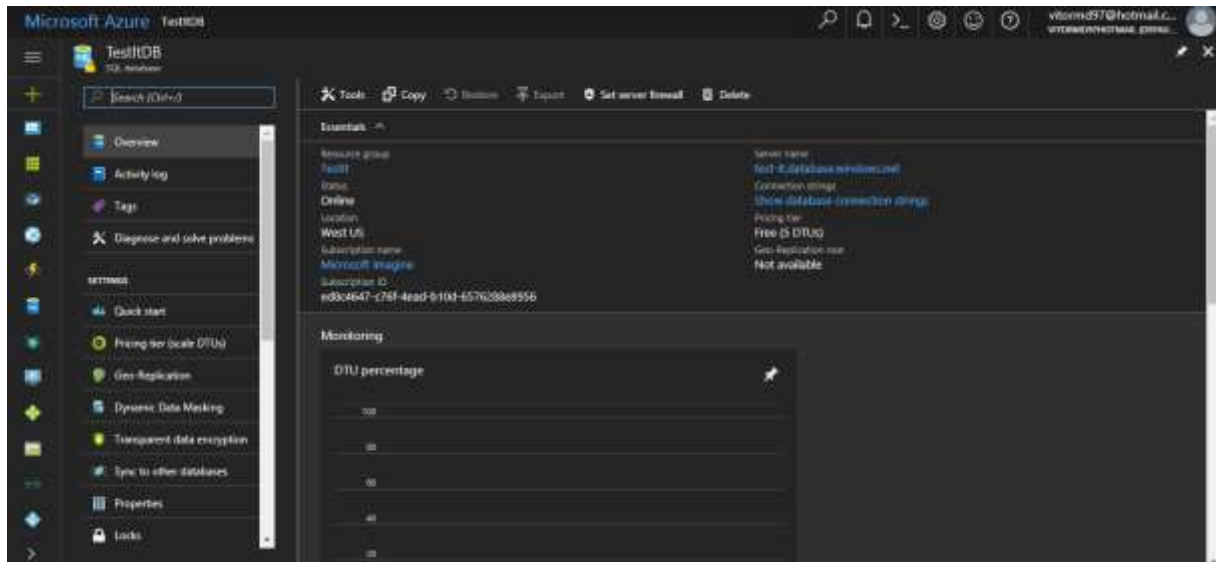


Fonte: (Autor, 2017)

Esse servidor está hospedado no provedor *cloud Azure*. Nesse servidor também foi implementado o conceito de CD. E também foi integrado com o *Github*, e tem um fluxo similar ao do servidor apresentado anteriormente, porém a diferença é que ele obtém um código que ainda não é publicável e roda um *script* de *build*, e depois um *script* de *publish*, no qual são geradas as DLL's, depois disso ele já disponibiliza a API na *web*.

A respeito da criação do Banco de Dados, foi feita no *Azure*, e o SGBD que está sendo usado é o *SQLServer*, que é uma ferramenta da *Microsoft*. (Figura 3)

Figura 3 – Banco de Dados

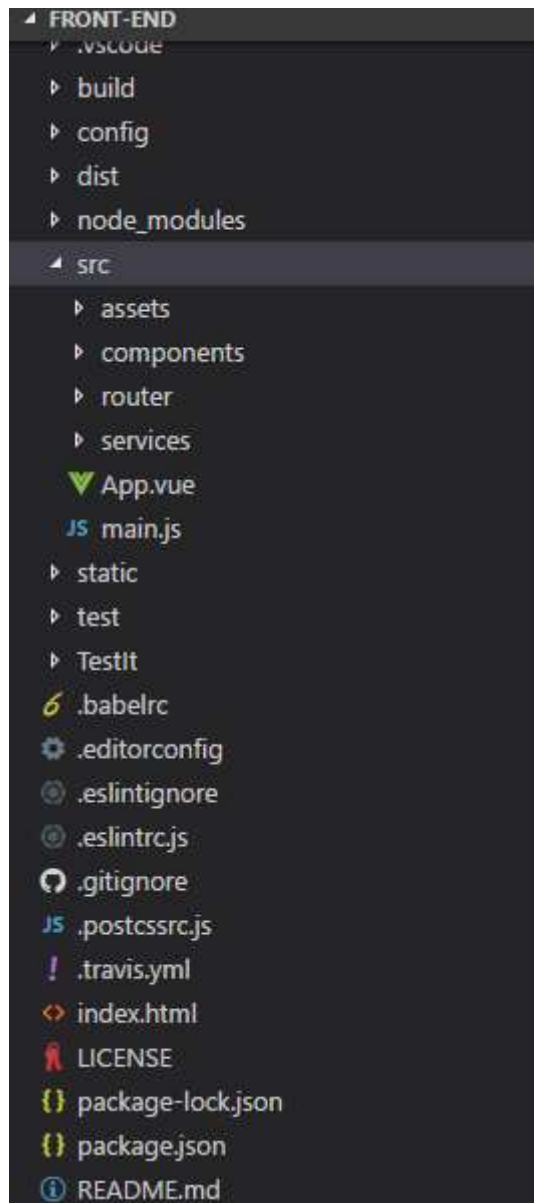


Fonte: (Autor, 2017)

5.2 Arquitetura

A Arquitetura do projeto também se divide em *front-end* e *back-end*. No *front*, está sendo aplicada a arquitetura proposta pelo *Vue.js*, na qual o código está dividido conforme figura 4.

Figura 4 – Arquitetura Front-End



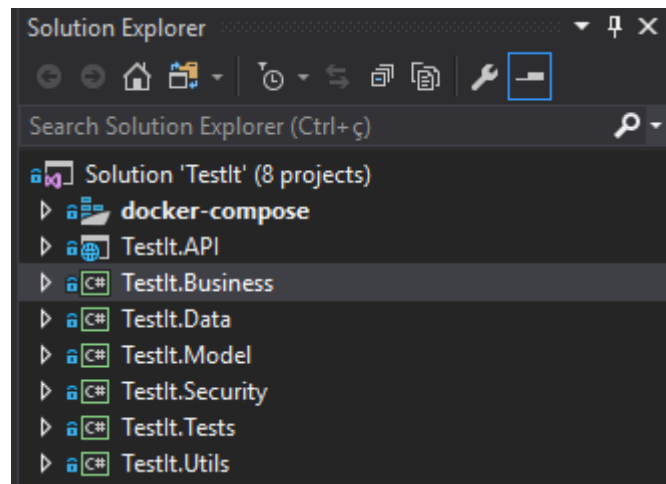
Fonte: (Autor, 2017)

Nessa arquitetura, o código-fonte encontra-se na pasta *src*, e lá existem algumas pastas que ajudam a separar o código, na pasta *assets* estão as imagens utilizadas e o códigos referentes ao estilo da página, na pasta *componentes* estão todos os componentes Vue que são utilizados, como Login, Cadastro, Formulário, entre outros, na pasta *router* estão definidas as rotas, por exemplo, se o usuário

acessar a rota '/cadastro' ele leva ele ao componente de cadastro, e na pasta *service*, estão definidas as chamadas do *back-end*, onde está feita a integração de ambos.

A arquitetura do *back-end* está dividida em camadas conforme figura 5.

Figura 5 – Arquitetura Back-End



Fonte: (Autor, 2017)

A camada do *Docker* onde estão todos os arquivos relacionados aos containers nos quais a aplicação roda em ambiente de desenvolvimento.

A camada *TestIt.API*, é onde são processadas as requisições do *front-end*, isso é feito nas *Controllers*, além disso existem lá também as *ViewModels*, que são as definições das entidades que serão enviadas pelo *front-end*.

A camada *TestIt.Business*, é onde se encontram as *services*, que é onde estão aplicadas as regras de negócio, lá se encontra o *Core* do sistema.

A camada *TestIt.Data* que é a camada onde está o acesso ao banco de dados, lá estão definidos os repositórios e suas respectivas interfaces.

A camada é a *TestIt.Model*, onde estão as entidades do banco, lá estão modeladas as entidades que são utilizadas pela aplicação.

A camada *TestIt.Tests*, onde estão os testes unitários da aplicação.

A camada *TestIt.Utils*, onde estão vários serviços que são utilizados pela aplicação como o envio de e-mails, o processamento de arquivos e o acesso a API's externas.

5.3 Telas

Nesse item serão apresentadas as telas que fazem parte do fluxo do aluno. As informações sobre análise de requisitos foram obtidas a partir de (SOARES,2017) e as sugestões de interface de (CAMPOS,2017). E a parte do fluxo do professor foi desenvolvida por (JUNIOR,2017).

5.3.1 Login

A tela de login foi projetada conforme figura 6.


Figura 6 – Tela Login



Fonte: (Autor, 2017)

Nessa tela da figura 6 o aluno insere o e-mail e senha que ele utilizou para fazer o cadastro, e ao clicar no botão login, será disparado um método que faz a chamada da API, nessa rota primeiro é feita uma validação de e-mail e senha, depois é gerado um *token* JWT (*json web token*) com tempo de expiração de 3 horas caso o usuário seja válido. Após isso, no retorno da chamada da API, o token é armazenado para ser utilizado nas próximas requisições, quando o token expirar o usuário será redirecionado para a tela de login novamente.

Figura 7 – Concluir Cadastro



Fonte: (Autor, 2017)

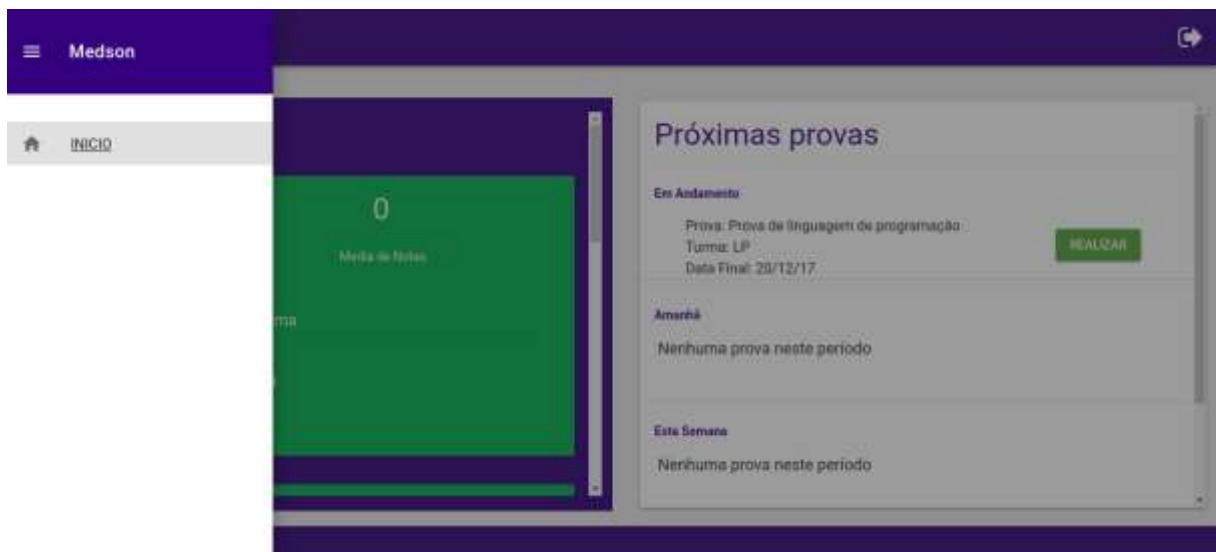
A tela da figura 7 foi desenvolvida, a tela de Concluir Cadastro, com a finalidade de concluir o cadastro que foi iniciado pelo professor, quando o professor adiciona um aluno a turma dele, caso o aluno ainda não tenha uma conta no TestIt, ele recebe um e-mail que contém um link que leva a essa tela. Portanto, nessa tela algumas informações como Nome e E-mail já estão cadastradas faltando apenas complementar senha e telefone. Ao clicar no botão cadastrar a aplicação irá fazer a chamada do método criado na API, que recebe uma ViewModel faz a conversão dela para a entidade do banco e depois envia o usuário para o *service* de usuário presente na camada *TestIt.Business*. Nesse método ele faz a chamada da camada

Data onde será feita a atualização dos atributos do usuário, depois a API retornará ao *front*, uma mensagem de sucesso ou falha.

5.3.2 Home

Após o login, o aluno é levado para a tela referenciada na figura 8

Figura 8 – Home



Fonte: (Autor, 2017)

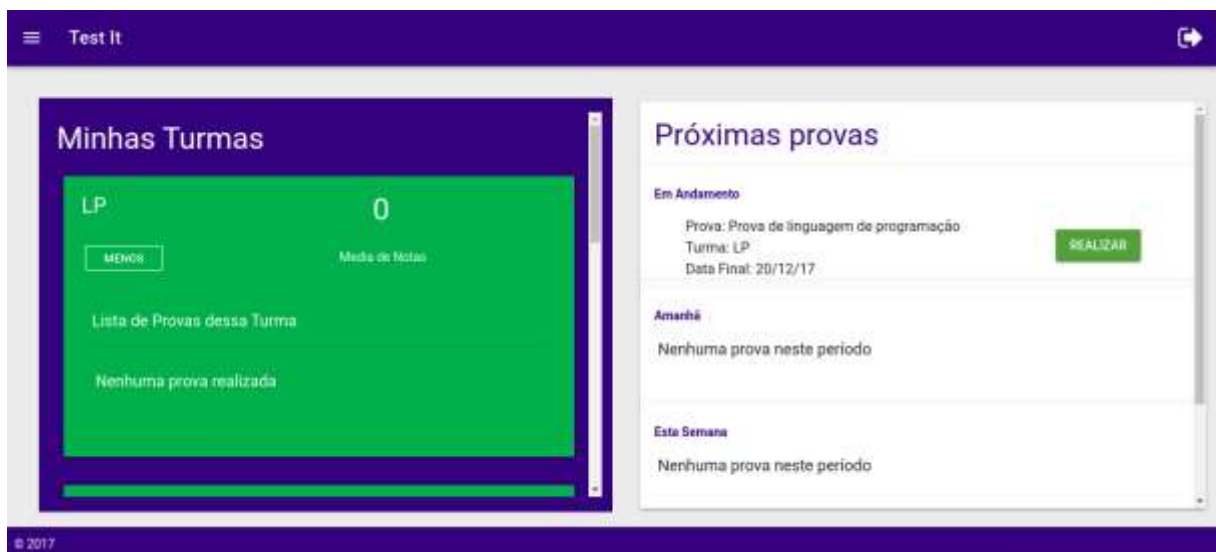
Nessa tela o aluno visualizará o menu, agenda das próximas provas e o histórico de notas por turma.

Figura 9 – Minha turma minimizada



Fonte: (Autor, 2017)

Figura 10 – Minha turma maximizada



Fonte: (Autor, 2017)

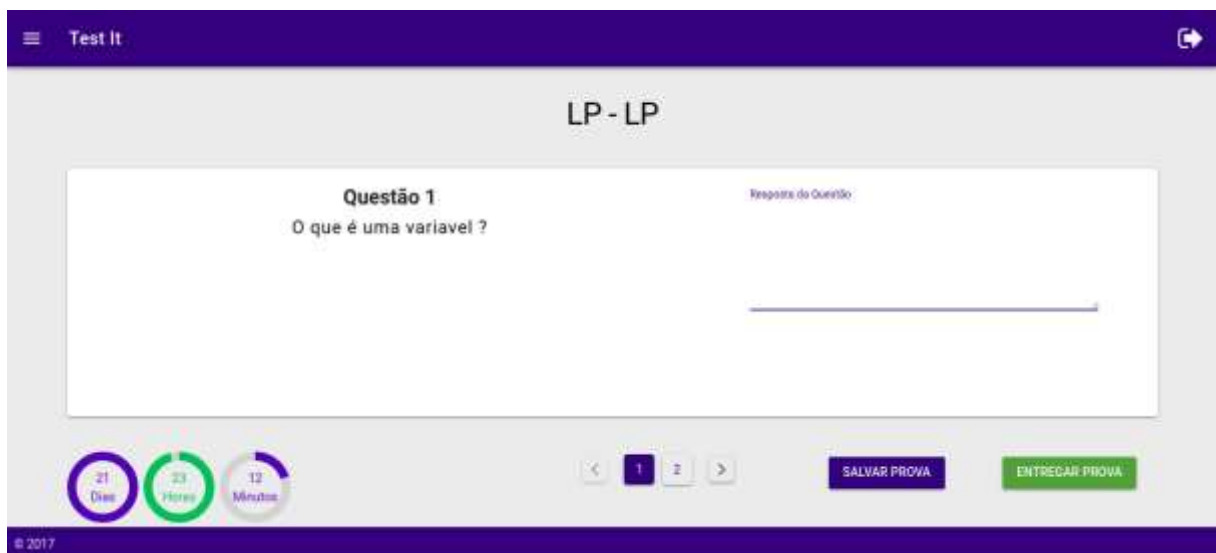
Caso o aluno aperte o botão mais referenciado na figura 9, a figura 10 irá aparecer, e será exibido o histórico de provas e a média. O *card* contém, nome da turma, a média da turma que será calculada pela soma de todas as notas das provas dividida pela quantidade de provas aplicadas à essa turma. Abaixo será exibida uma lista de provas da turma, que irá conter a lista de todas as provas que o

aluno realizou (nome da prova e a nota que o aluno recebeu após a correção), como é mostrado na figura 10.

5.3.3 Realizar Prova

Caso o aluno tenha uma prova para realizar será encaminhado para a tela de realização de prova após clicar no botão realizar que aparece na tela referenciada na figura 9.

Figura 11 – Realizar prova dissertativa



Fonte: (Autor, 2017)

Figura 12 – Realizar prova alternativa



Fonte: (Autor, 2017)

Durante a realização da prova o aluno poderá responder 2 tipos de questões, questões dissertativas, representada na figura 11 ou questão alternativa, representada na figura 12.

Caso o aluno esteja dentro do período de realização da prova irá aparecer um timer que representa o tempo restante para o aluno entregar a prova e o botão para o aluno entregar a prova, conforme a figura 11, após a prova ser entregue essa será a visão que o aluno terá.

5.3.4 Pré-Correção da Prova

Esse é o grande diferencial do produto, após o aluno entregar a prova, o *front* irá fazer uma chamada passando o id da avaliação realizada pelo aluno e então ele fará uma pré correção da prova, então lá foi criado um código que faz essa lógica, ele tem os seguintes passos. Primeiro as questões alternativas são corrigidas, como elas tem já uma resposta fixa e não é suscetível a interpretação, elas já são

corrigidas e a nota é atribuída ao aluno de acordo com o valor pré cadastrado da questão. Depois disso o algoritmo irá realizar a pré correção que consiste em calcular um percentual de acerto do aluno sobre a questão, com o propósito de auxiliar o professor durante a real correção da prova. Após corrigir todas as alternativas, são calculados os percentuais de cada questão dissertativa, esse algoritmo faz os passos seguintes.

Primeiro separa em frases a resposta do aluno em frases e depois separa a resposta do professor em frases, as frases são delimitadas com ponto final, ponto de exclamação, entre outros, depois de separadas em frases, essas serão separadas em palavras, porém não apenas palavras da maneira regular, mas palavras lematizadas, ou seja, passando pelo algoritmo de lematização, esse algoritmo é o processo de deflexionar uma palavra para determinar o seu lema (as flexões chamam-se lexemas). Por exemplo, as palavras gato, gata, gatos e gatas tem o mesmo lema, assim como tiver, tenho, tem são do mesmo lema ter. A lematização é útil quando se quer ver os usos de palavras em contextos sem importância das flexões. Isso é muito útil para que a correção seja mais justa, porque muitas vezes o aluno pode colocar palavras similares que ainda deixem a resposta certa mesmo que não sejam 100% iguais a do professor, depois desse processo o algoritmo faz uma soma do melhor percentual de semelhança das frases, esse percentual é calculado utilizando a distância de *levenshtein*, que é dada pelo número de operações mínimo para transformar um *string* em outro, entende-se por operações a inserção, deleção ou substituição de um caractere. Após obter a soma desse melhor percentual é feita a divisão e se pega o melhor percentual, isso é feito para todas as frases e então é tirada a média desses percentuais, todo esse cálculo equivale a 50% do percentual final, porque após conversas com usuários eles disseram que

seria mais interessante marcar algumas palavras chave, então é feita uma busca dessas e outro percentual é calculado verificando se encontram na resposta do aluno, isso equivale a 50% do percentual final também, caso o professor tenha escolhido não cadastrar as palavras-chave somente o primeiro percentual é utilizado.

Tudo isso é feito para auxiliar o professor no momento da correção, ele somente irá realizar a correção das questões dissertativas, como esse cálculo não é algo que garantirá uma correção exata ele somente será mostrado para o professor no momento da correção e então ele pode escolher aceitar esse como o percentual final ou ele pode ajustar e informar qual o percentual exato de acordo com a correção dele.

6 CONCLUSÃO

Durante o projeto foram realizadas diversas etapas, desde a estruturação do projeto, definição da arquitetura tanto do *front-end* quanto do *back-end* e banco de dados, além disso todas as telas foram criadas e codificadas. O projeto permite que o aluno acesse um *dashboard* que o auxilia a ver sua agenda com todas as provas e o seu histórico com todas as notas de cada turma e uma tela de realização de provas na qual ele responde e envia suas respostas ao professor, então é feita uma pré correção dessas questões e elas são encaminhadas ao professor para que ele finalize a correção.

Mais um ponto interessante, partindo do pressuposto que o código fonte é aberto, durante um evento na Fatec (Semana de Tecnologia) o projeto foi apresentado para alguns desenvolvedores, e alguns deles também contribuíram com ideias para melhorar a aplicação.

Foram realizados testes com potenciais usuários, foram abordados (via entrevista) dois tipos de usuário, professor e aluno, onde eles realizaram seus respectivos fluxos e encontraram alguns pontos de melhoria, na usabilidade (questões de botões para facilitar ou que não estavam de acordo) e alguns problemas de responsividade (quando é acessado via dispositivo móvel) também que devem ser ajustados.

O resultado do trabalho foi um sistema que ajuda o aluno, a realizar suas avaliações e acompanhar a correção, notas desta forma simplificando o processo.

7 SUGESTÕES PARA TRABALHOS FUTUROS

A aplicação foi feita de uma maneira genérica sem personalização para uma instituição de ensino específica, a ideia é que no futuro esse sistema possa ser direcionado a instituições que queiram implantar esse sistema e então será feita uma solução totalmente personalizada com as regras dessa instituição e com os padrões dela, a aplicação deve ser aprimorada para entrar em produção, porque hoje está no ar apenas uma versão de homologação.

REFERÊNCIAS

- AGUIAR, Sabrina Ferreira de. **Modelos e experiências de ambientes de aprendizagem virtual**. CEFET-MG, 2010
- ALURA. **Sobre a Alura**. Disponível em: < <https://www.alura.com.br/sobre/> > Acesso: 20.JUL.2017.
- AZURE. **What is Azure**. Disponível em <<https://azure.microsoft.com/en-us/overview/what-is-azure/>>. Acesso: 20.JUL.2017
- BARROS, Maria das Graças; CARVALHO, Ana Beatriz Gomes. **As concepções de interatividade nos ambientes virtuais de aprendizagem**. Disponível em:< <http://books.scielo.org/id/6pdyn/pdf/sousa-9788578791247-09.pdf>> Acesso: 21.JUL.2017.
- BELLONI, Maria Luiza. **Educação a Distância**. Campinas: Autores Associados, 2008.
- BELMONTE, Vanessa **Ambientes virtuais de Aprendizagem: Um panorama da produção nacional**. CEFET-MG, Belo Horizonte. 05/2010
- CAMPOS, Matheus Novais de Campos. **Relatório Final de Projeto de Iniciação Científica: TestIt - Aplicação Web Avaliações Online (Desenvolvimento Front-End, Design e UX)**. Faculdade de Tecnologia de Sorocaba “José Crespo Gonzales”. 12/2017.
- CASTELLS, Manuel. **A sociedade em rede**. São Paulo: Paz e Terra, 1999
- DECRETO5622. **DECRETO Nº 5.622, DE 19 DE DEZEMBRO DE 2005**. Disponível em: <<http://www.santosjunior.com.br/Legislacao/decreto562219122005.pdf>> Acesso: 21.JUL.2017.
- DECRETO9057. **DECRETO Nº 9.057, DE 25 DE MAIO DE 2017**. Disponível em: <<http://www.santosjunior.com.br/Legislacao/decreto905725052017.pdf>> Acesso: 21.JUL.2017.
- FLANAGAN, DAVID – **Javascript The Definitive Guide**. O'REILLY, 2011
- FRANÇA, George **Os ambientes virtuais de aprendizagem: um estudo do moodle no curso de pedagogia da UFT**. 2009
- INCAU, CAIO. **Vue.js Construa Aplicações Incríveis**. São Paulo. 2017. Casa do Código
- JUNIOR, Medson de Oliveira. **Relatório Final de Projeto de Iniciação Científica: TestIt – Aplicação Web Avaliações Online (Desenvolvimento Ambiente Professor)**. Faculdade de Tecnologia de Sorocaba “José Crespo Gonzales”. 12/2017.

MEDSOFT. **O que é o MEDSOFT.** Disponível em <<http://medsoft.medgrupo.com.br/#/o-que-e-o-medsoft>>. Acesso em: 20.JUL.2017

MOODLE. **Moodle.** Disponível em <https://moodle.org/?lang=pt_br /> Acesso: 20.JUL.2017.

PELLIZZONI, Leonardo. **Desenvolvimento de arquitetura de software reutilizável para implementação de aplicações em .NET.** 2017.

PRETI, O. **Educação a Distância: uma prática educativa mediadora e mediatizada.** Cuiabá: NEAD/IE- UFMT. 1996. Disponível em: <<http://pt.scribd.com/doc/182436526/EAD-uma-pratica-educativa-mediadora-e-mediatizada#scribd>>. Acesso em: 21.FEV.2017.

PRICE, Mark J. **C# 6 and .NET Core 1.0 Modern Cross-Platform Development.** 2016

SANTOS, Edméa Oliveira. **Ambientes virtuais de aprendizagem: por autorias livre, plurais e gratuitas.** In: Revista FAEBA, v.12, no. 18, 2003

SQL SERVER. **SQL Server 2016.** Disponível em: <<https://www.microsoft.com/pt-br/sql-server/sql-server-2016>> Acesso em: 20.JUL.2017

SILVA, Marco. Santos, Edméa. **Avaliação da Aprendizagem Em Educação Online.** Edições Loyolla. 2010.

SOARES, Luiz Guilherme Soares. **Relatório Final de Projeto de Iniciação Científica: TestIt – Aplicação Web Avaliações Online (Análise de Requisitos).** Faculdade de Tecnologia de Sorocaba “José Crespo Gonzales”. 12/2017

SOLAR. **SOLAR Ambiente virtual de aprendizagem da Universidade Federal do Ceará.** Disponível em <<http://www.solar.virtual.ufc.br/faq>>. 2017

TELEDUC. **TELEDUC.** Disponível em: <<http://www.teleduc.org.br/>>. Acesso:20.JUL.2017

VENETIANER, Tomas. **HTML: desmistificando a linguagem da Internet. HTML: Desmistificando a linguagem da Internet,** 1996.

VIEIRA, Alexandre Thomaz Vieira. **Gestão Educacional e Tecnologia.** São Paulo: Avercamp, 2003

VUE. **Vue.js The Progressive JavaScript Framework.** Disponível em: <<https://vuejs.org/>>. Acesso em: 20.JUL.2017

GLOSSÁRIO

API - API é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseado na Web. A sigla API refere-se ao termo em inglês "Application Programming Interface" que significa em tradução para o português "Interface de Programação de Aplicativos".

BANCO DE DADOS - Bancos de dados são coleções de informações que se relacionam de forma que crie um sentido. São de vital importância para empresas, e há duas décadas se tornaram a principal peça dos sistemas de informação.

BUILD – Build, no contexto do desenvolvimento de software, é uma versão "compilada" de um software ou parte dele que contém um conjunto de recursos que poderão integrar o produto final.

DASHBOARD - Dashboard ou Painel de Controle é a apresentação visual das informações mais importantes e necessárias para alcançar um ou mais objetivos de negócio, consolidadas e ajustadas em uma tela para fácil acompanhamento do seu negócio.

DLL - DLL é uma sigla do mundo da informática e significa Dynamic-link library, ou "Biblioteca de Vínculo Dinâmico" em português. A biblioteca representada pela sigla DLL possui dados e códigos que podem ser usados simultaneamente por mais de um programa.

DEPLOY – É fazer com que sua aplicação seja instalada em um servidor de aplicações, fazendo com que ela seja acessível aos clientes daquela aplicação

DESIGN-PATTERNS - Em engenharia de software, um padrão de projeto ou padrão de desenho (do inglês design pattern) é uma solução geral reutilizável para um problema que ocorre com frequência dentro de um determinado contexto no projeto de software.

JAVASCRIPT - é uma linguagem de programação interpretada. Foi originalmente implementada como parte dos navegadores web para que scripts pudessem ser executados do lado do cliente e interagissem com o usuário sem a necessidade deste script passar pelo servidor, controlando o navegador, realizando comunicação assíncrona e alterando o conteúdo do documento exibido.

PUBLISH – É a ação feita para publicar a aplicação, gerar as DLL's e tudo mais que é necessário para realizar o Deploy da aplicação.

SCRIPT – Conjunto de instruções para que uma função seja executada em determinado aplicativo.

SGBD - Um Sistema de Gerenciamento de Banco de Dados (SGBD) - do inglês Data Base Management System (DBMS) - é o conjunto de programas de computador (softwares) responsáveis pelo gerenciamento de um banco de dados. Seu principal objetivo é retirar da aplicação cliente a responsabilidade de gerenciar o acesso, a persistência, a manipulação e a organização dos dados. O SGBD

disponibiliza uma interface para que seus clientes possam incluir, alterar ou consultar dados previamente armazenados. Em bancos de dados relacionais a interface é constituída pelas APIs (Application Programming Interface) ou drivers do SGBD, que executam comandos na linguagem SQL (Structured Query Language).

TOKEN - Um Token em computação é um segmento de texto ou símbolo que pode ser manipulado por um analisador sintático, que fornece um significado ao texto; em outras palavras, é um conjunto de caracteres (de um alfabeto, por exemplo) com um significado coletivo.

WEB APP- Uma aplicação web (ou Web Application) é um aplicativo que é acessado através de uma rede Intranet/Extranet/Internet. O termo também pode significar uma aplicação de software de computador que está hospedada em um ambiente controlado browser e dependente de um navegador web comum para tornar o aplicativo executável.

WEB SERVICE - Web service é uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes.