UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DEPARTAMENTO DE COMPUTAÇÃO

BACHAREL ENGENHARIA DE SOFTWARE

MAGNO CARVALHO DOS SANTOS

**ESTÁGIO CURRICULAR OBRIGATÓRIO: MANUTENÇÃO EM APLICAÇÃO DE REGISTRO DE PONTO DE FUNCIONÁRIOS DE PEQUENAS EMPRESAS**

RELATÓRIO DE ESTÁGIO

MAGNO CARVALHO DOS SANTOS

**RELATÓRIO DE ESTÁGIO**

Relatório de estágio supervisionado obrigatório, apresentado à disciplina de Estágio Curricular Supervisionado, do curso de Engenharia de Software da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Bacharel.

Orientadora: Prof. Ma. Adriane Carla Anastácio Da Silva

CORNÉLIO PROCÓPIO

2019

**RESUMO**

SANTOS, M. C. dos. **ESTÁGIO CURRICULAR OBRIGATÓRIO: MANUTENÇÃO EM APLICAÇÃO DE REGISTRO DE PONTO DE FUNCIONÁRIOS DE PEQUENAS EMPRESAS**. 2019. Relatório de Estágio Supervisionado Obrigatório – Engenharia de Software. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2019

Este relatório tem como objetivo expor os conhecimentos e experiências do aluno Magno Carvalho dos Santos durante estágio obrigatório na empresa Yankton Technologies com início em 18 de março de 2019 e término 18 de julho de 2019 com uma carga horária de 30 horas semanais, perfazendo um total de 400 horas. As atividades desenvolvidas foi a manutenção de uma aplicação *(web)* denominada ‘Automativo’, com a missão de reestruturar o código, corrigir *bugs*, adicionar novos módulos e *design*. Os resultados obtidos durante todo período de estágio fortaleceram os conhecimentos adquiridos pelo aluno durante a graduação, por meio da prática em ambiente profissional.

**Palavras-chave:** Yankton Technologies, MEAN Stack, Angular, TypeScript.

**SUMÁRIO**

[INTRODUÇÃO](#_heading=h.bmjywla52v5p) **5**

[INSTITUIÇÃO](#_heading=h.2ac6aypk3kq) **6**

[ATIVIDADES DESENVOLVIDAS](#_heading=h.va9mlmvz15u3) **7**

[SISTEMA “AUTOMATIVO”](#_heading=h.2507p4n9qd0b) 7

[PROCESSO DE DESENVOLVIMENTO](#_heading=h.fyhpyu6ujqcp) 10

[TECNOLOGIAS UTILIZADAS](#_heading=h.62p8d4acxvy) 11

[MONGODB](#_heading=h.2s8eyo1) 12

[EXPRESSJS](#_heading=h.17dp8vu) 13

[ANGULAR](#_heading=h.3rdcrjn) 13

[NODE.JS](#_heading=h.26in1rg) 14

[HTML E CSS](#_heading=h.44sinio) 14

[JAVASCRIPT](#_heading=h.z337ya) 15

[TYPESCRIPT](#_heading=h.3j2qqm3) 15

[RESULTADOS OBTIDOS](#_heading=h.ql2a4nbeheav) **16**

[CONSIDERAÇÕES FINAIS](#_heading=h.4qsw54gqp4v3) **17**

[REFERÊNCIAS](#_heading=h.bdsmmhvxikct) **18**

# INTRODUÇÃO

O estágio obrigatório é uma disciplina constituinte da ementa do curso de bacharel em Engenharia de Software da Universidade Tecnológica Federal do Paraná, campus Cornélio Procópio-Pr (UTFPR/CP). Esta disciplina pode ser iniciada a partir do 6º semestre e tem como principal função ingressar os alunos em um ambiente de trabalho no qual serão adquiridos conhecimentos e experiências pertinentes à sua área de formação. Na UTFPR/CP, o período completo de estágio deve ser de no mínimo 400 horas.

Este relatório apresenta as atividades desenvolvidas pelo aluno durante o período de estágio, neste relatório se apresentam os projetos desenvolvidos neste período, as atividades realizadas durante o desenvolvimento e correção de *bugs* da aplicação Automativo. Para efetuar as atividades, foi utilizado conhecimento nas tecnologias HTML, CSS e Javascript, conhecimento de boas práticas de programação.

# INSTITUIÇÃO

A empresa Yankton Technologies situa-se em Cornélio Procópio, Paraná. Fundada em Janeiro de 2016 pelos sócios Carlos Kikuti e Leonardo Pepis. Sua missão é se tornar uma empresa com um excelente ambiente de trabalho, a fim de gerar melhores condições de trabalho consequentemente, gerar produtos de qualidade. Para atingir seus objetivos a empresa gera concursos internos e aplicando questionários para melhorar o ambiente e o processo de criação dos produtos.

A Yankton é uma fábrica de softwares que trabalha sob demanda, e fornece serviços e soluções na área de TI (Tecnologia da Informação). Entre seus principais clientes encontra-se a Sioux, uma empresa de softwares localizada na cidade de São Paulo-SP. Entretanto, a empresa Yankton também possui produtos próprios como a AgendaJOB[[1]](#footnote-0), e futuros projetos.

Durante a realização das tarefas do estágio, foi supervisionado e orientado pelos fundadores, Leonardos Pepis e Carlos Kikuti, e pela Prof. Adriane Carla Anastácio Da Silva, orientadora do estágio.

# ATIVIDADES DESENVOLVIDAS

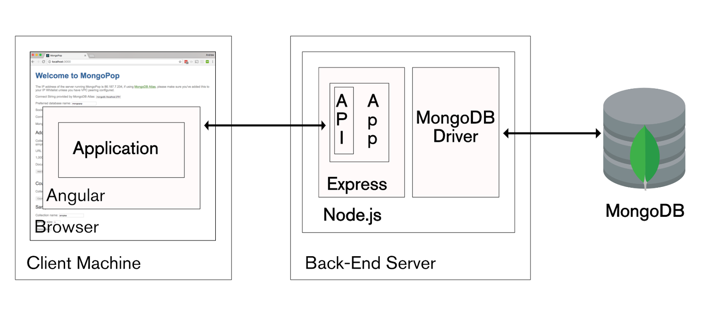
Neste capítulo são apresentadas as atividades desenvolvidas durante o período de estágio dentro da empresa e as atividades realizadas no projeto “Automativo”.

## SISTEMA “AUTOMATIVO”

O sistema “Automativo” era responsável por registrar os pontos dos profissionais da própria empresa, organizados em planilhas mensais com os horários de entradas e saídas.

Foi implementado no sistema, o conceito de *MEAN Stack*, utiliza como as principais tecnologias, MongoDB, ExpressJS, Angular e NodeJS. A relação das tecnologias utilizadas é apresentada na Figura 1.

**FIGURA 1 – Relação das tecnologias**



**Fonte: Adaptado de (MORGAN, 2017).**

Os módulos adicionais que foram desenvolvidos durante o período de estágio contempla a correção de *bugs* da aplicação, geralmente, relacionados ao fuso horário, hierarquia de usuários, cálculos de horas trabalhadas, a aplicação foi transformada em um produto sendo possível o registro ponto dos profissionais de uma empresa, previamente cadastrada e após o término de uma competência/mês, transmitir a competência/cartão ponto de forma digital para a contabilidade. Eliminando a necessidade do envio físico destes documentos. Os módulos desenvolvidos estão relacionados no Quadro 1.

**Quadro 1 - Módulos desenvolvidos pelo aluno**

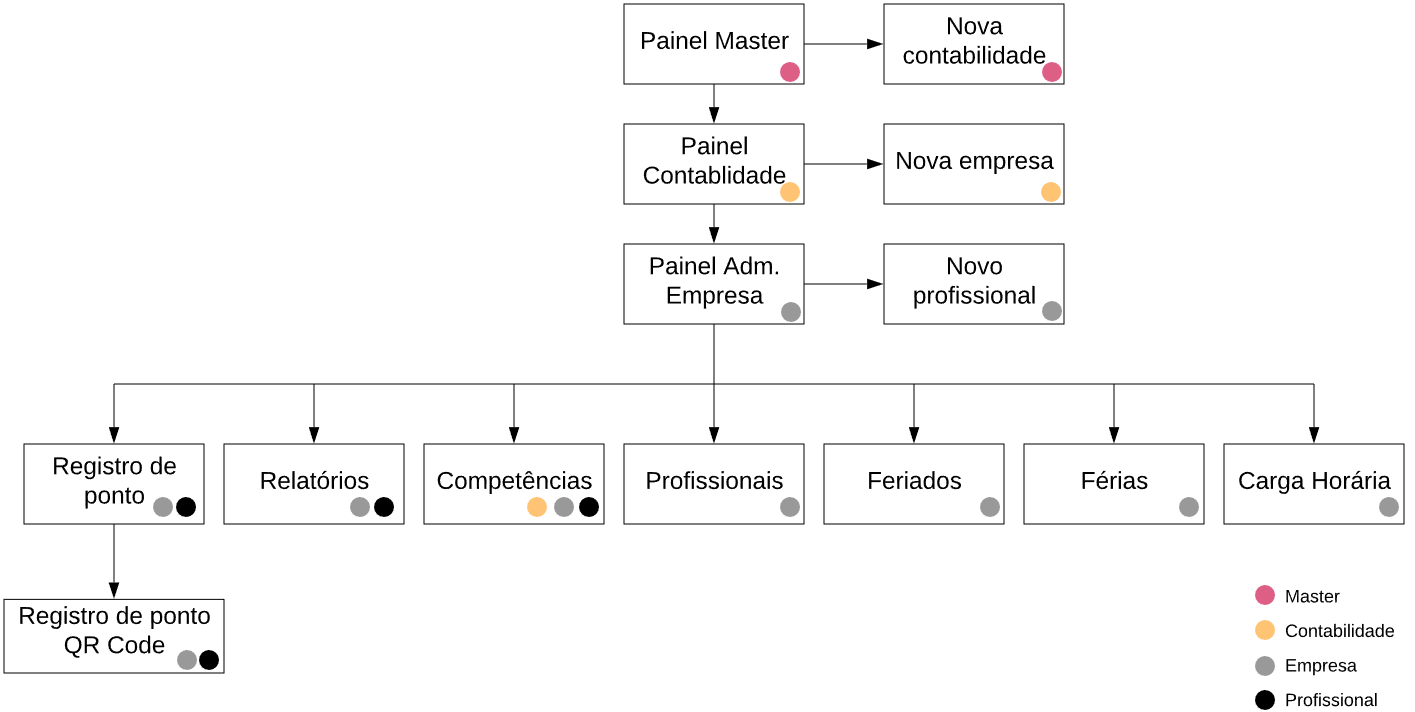
|  |  |
| --- | --- |
| **Módulo** | **Descrição do item desenvolvido** |
| Master | O Administrador geral do sistema ou usuário Master possui permissão para gerenciar exclusivamente o perfil de uma ou mais contabilidades/contadores. O perfil master terá cadastro estático no banco de dados, podendo se necessário possuir mais de um usuário ativo. |
| Contabilidade | O perfil contabilidade ou contador possui permissão para gerenciar uma ou mais empresas para controle do registro de horas. Uma empresa só existirá se estiver atrelada a uma contabilidade. |
| Administrador Empresa | O administrador de empresas possui toda gerência dos profissionais que utilizam a plataforma para realizar o registro de ponto. O usuário responsável/administrador poderá visualizar e editar todas as informações que estejam em seu cadastro assim como sua senha de acesso. Também é de responsabilidade do mesmo gerenciar novos usuários administradores do sistema. Também poderá realizar o desligamento do profissional, não será possível excluir um usuário da empresa. |
| Carga horária | Uma carga horária é o período em que os profissionais devem obedecer no registro de ponto. A carga horária é restrita somente ao administrador do sistema que poderá cadastrar mais de uma se necessário. |
| Férias | O período de férias deverá ser inserido manualmente a cada usuário, neste período o trabalhador que esteja em gozo de férias não poderá realizar o registro de horas. Para esta sessão são exibidos apenas os usuários ativos. O período de férias não poderá ser maior que 30 dias, mas poderá ser fracionada. Um profissional poderá ter até 3 férias ao longo do período de contemplação da mesma, desde que uma delas tenha no mínimo 14 dias corridos e não sejam menores que 5 dias.  Apenas o administrador do sistema do painel de empresas é que poderá cadastrar férias, poderá também desativar uma férias cadastrada desde que o período de concessão não tenha sido iniciado. |
| Competências | A competência é o período de trabalho onde o trabalhador cumpre suas funções de acordo com a carga horária estabelecida, ao final de 30 dias de trabalho a competência é fechada e o registro de horas será contabilizado pela contabilidade/contador por meio digital no painel do contador. |
| Fechar competência | Somente o administrador do sistema após o fim do período estabelecido poderá realizar o encerramento, por regra uma competência só poderá ser fechada no dia anterior ao estabelecido como fim. |
| Assinar competência | Assinatura virtual da competência já fechada. Assim que uma competência for fechada o usuário que realiza o registro de ponto deverá será notificado, a notificação deverá conter link de redirecionamento para tela de competência. |
| Relatórios | O relatório exibe informações sobre todos os registros da competência atual e históricos para o administrador do sistema, o usuário comum tem acesso apenas ao seu relatório, já o administrador tem acesso a todos os relatórios dos demais usuários. |

|  |  |
| --- | --- |
| **Módulo** | **Descrição do item desenvolvido** |
| Registro de ponto | Todo profissional/administrador cadastrado a uma empresa terá acesso ao registro de ponto, o registro é livre para ser utilizado em qualquer momento obedecendo as regras que estejam vinculadas ao perfil. Um usuário só poderá realizar o registro se estiver cadastrado a uma carga horária. Um registro só será válido caso o usuário esteja dentro da localização física da empresa, a validação deverá ocorrer por geolocalização. Para os registros que ocorrerem fora da localização da empresa, o administrador do sistema deverá ser notificado. |
| Faltas | Faltas dos profissionais que realizam o registro de ponto são classificadas em: Faltas justificadas e faltas. Somente o administrador do sistema poderá editar o registro de horas do trabalhador e inserir informações sobre faltas. |
| Feriados e recessos | Feriados e recesso são datas em que deveria ocorrer o registro de ponto, porém como não há atividades de trabalho automaticamente o sistema deverá inserir o valor da carga horária daquela data adicionando informações sobre ela. Apenas o administrador do sistema do painel de empresas é que poderá cadastrar feriados e recessos. |

**Fonte: Autoria Própria**

O sistema Automativo ganhou nova arquitetura após as alterações e adições de novos módulos pelo estagiário, apresentada na Figura 2.

**Figura 2 - Mapa do sistema**



**Fonte: Autoria Própria**

## PROCESSO DE DESENVOLVIMENTO

Pela natureza do projeto foi utilizado um processo específico para realização das atividades. O projeto se desenvolveu por meio de reuniões, na intenção de fazer levantamento de requisitos por meio de conversa entre aluno e os supervisores da empresa. A partir do resultado da reunião, iniciava o desenvolvimento. Após, a implementação acontecia uma nova reunião para validação. Caso, observado a necessidade de novos requisitos no sistema, começava-se o processo de novo por meio da reunião de levantamento de requisitos, em uma adaptação do método ágil chamado Scrum, o que divide o projeto em ciclos chamados de *Sprints*.

Todas as funcionalidades desejadas se encontram em uma lista chamada Product Backlog e quando se deseja implementar algumas funcionalidades em um ciclo desejado as atividades são colocadas em uma lista chamada *Sprint Backlog*.

A Yankton Technologies utiliza-se três listas a fim de validar o que foi implementado:

* *DOING* (Fazendo): É designado ou o próprio desenvolvedor escolhe um item da lista da *sprint*, os itens da lista ter prioridades. Nele o desenvolvedor arrasta a tarefa para *doing* para indicar a equipe que aquela atividade já está sendo desenvolvida.
* *HOMOLOG* (Foi Publicado em ambiente de homologação): Na Yankton Technologies há um servidor utilizado para ser publicado versões para testes. Os testes são feitos por meio deste servidor ou caso não seja necessário utilizar o servidor o próprio *tester*, para realizar os testes. Então, ao finalizar a tarefa o desenvolvedor arrasta a tarefa de *doing* para *homolog* para ser validado pelo *tester*.
* *DONE* (Pronto): A tarefa é colocada nesta lista quando o testador validou o item implementado pelo desenvolvedor. Após, o ciclo da tarefa se finaliza ou indica que melhoria foi finaliza.

Este método se baseia no sistema Kanban, um método Japonês, que permite o controle detalhado de produção, com informações de como, quando, quanto e o que produzir. O antigo sistema utilizava cartões (*post-it* e outros) e eram colocados em um quadro para indicar o fluxo do projeto, nos cartões eram colocados indicações sobre determinadas tarefas como por exemplo, o que precisa ser feito, o que está em andamento e o que foi finalizado.

Para a realização das tarefas foi necessária uma ferramenta física e algumas ferramentas de software. De ferramenta física foi utilizado um notebook. Como ferramentas de softwares foi empregado GitLab como plataforma de gerenciamento de configuração e para editar o código o, Visual Studio Code.

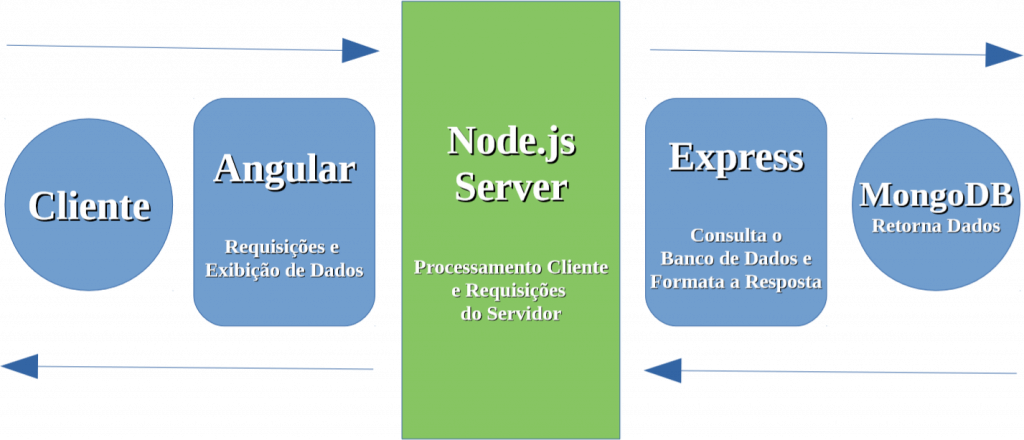
## TECNOLOGIAS UTILIZADAS

As tecnologias utilizadas são relacionadas ao acrônimo MEAN Stack segundo Bera, Mine e Lopes (2015) é o conjunto de tecnologias que possibilita o desenvolvimento de aplicações web escaláveis, utiliza a sintaxe de escrita do JavaScript desde a interface do usuário até a aplicação do lado servidor, Bera, Mine e Lopes (2015).

Os benefícios em utilizar uma aplicação baseada em JavaScript abrange dois fatores principais: (i) Escalabilidade; que se define pela disponibilidade de serviços requisitados a proporções crescentes de novos usuários; e (ii) Performance; o desempenho obtido levando em conta a escalabilidade e recursos consumidos.

A Figura 3 apresenta a arquitetura MEAN e a relação com as tecnologias.

**Figura 3 - Relação arquitetura MEAN STACK**



**Fonte: Adaptada de (OCTALMIND, 2018)**

Segundo Rodrigues e Zuchi (2017) o desenvolvimento web em geral temos duas áreas distintas que juntas se complementam front-end e back-end. A vantagem do uso do MEAN Stack como pilha de desenvolvimento, permite a integração entre as áreas citadas a sintaxe da linguagem de programação JavaScript. O uso dessa linguagem permitiu um ganho de produtividade e curva de aprendizado.

O estudo de Bonfim e Liang (2014) demonstrou que uso do conjunto de tecnologias do Acrônimo MEAN Stack, é uma opção no desenvolvimento escalável eficiente em comparação a outras tecnologias.

A descrição das tecnologias relacionadas ao acrônimo MEAN e apresentado a seguir:

### MONGODB

O software utiliza o MongoDB como solução NoSQL. As Tecnologias NoSQL podem ser utilizadas para banco de dados que precisam de escalabilidade, desempenho e possuir tolerância a falhas, sendo recomendado para aplicações *web (*TOTH, 2011).

De acordo com Hows, Membrey E Plugge (2015), o MongoDB é um banco de dados NoSQL, usa um modelo de dados flexível orientado a documentos JSON (JavaScript Object Notation) estes documentos podem mudar facilmente, conforme a necessidade e evolução da aplicação a ser desenvolvida.

O modelo de dados de documentos do MongoDB facilita o armazenamento e a combinação de dados de qualquer estrutura, sem abrir mão de regras sofisticadas de validação, acesso flexível a dados e rica funcionalidade de indexação. Você pode modificar dinamicamente o esquema sem tempo de inatividade - essencial para aplicativos em rápida evolução (MORGAN, 2017, tradução nossa).

No banco de dados MongoDB temos coleções de documentos, onde cada documento é independente, este contém todos os dados que precisa sendo diferente dos bancos de dados relacionais, suas consultas e operações são realizadas de forma assíncrona, assim como o NodeJS e segundo Politowski e Mara (2014).

O fato de que os bancos NoSQL terem nascido para suprir a demanda por performance, deixando outros detalhes, como atomicidade, por exemplo, em segundo plano(..) Bancos de dados NoSQL e Relacionais utilizam paradigmas diferentes e, por sua vez, possuem finalidades diferentes, mas com o mesmo propósito, persistir dados.

### EXPRESSJS

O ExpressJS é um framework e componente *middleware* que facilita o desenvolvimento de aplicações *web* com o NodeJS, permite instanciar servidores *web* e receber requisições HTTP, organizando e garantido a segurança e a entrega das informações trafegadas (MARDAN, 2019).

### ANGULAR

O Angular é um *framework* *JavaScript* desenvolvido e mantido pela empresa Google em parceria com a Microsoft, criado para permitir a fácil reutilização de código e manutenção. Ele possui um conjunto de ferramentas e componentes desenvolvidos pela comunidade que permitem ao desenvolvedor agilizar o desenvolvimento, conhecidos como *packages,* disponíveis no repositório NPM (*Node Package Manager*) que pode ser acrescentada a aplicação (FONSECA JR; FONSECA; DE LIMA RANGEL, 2018).

### NODE.JS

O Node.JS é uma plataforma para o desenvolvimento de aplicações do lado servidor que utiliza como a linguagem padrão o *JavaScript*, podendo ser escrita em *TypeScript*, ideal para criação de aplicações *web* no lado servidor de com foco em escalabilidade (MORAES, 2018).

Segundo Handy (2011) a plataforma é popular entre os desenvolvedores *web e* utilizada por grandes empresas como LinkedIn, Microsoft, GitHub e MySpace.

A arquitetura do NodeJS é composta em sua maior parte por componentes desenvolvidos em C++ e em *JavaScript* (MORAES, 2018). Sendo criado para ser estendido por meio de módulos, sendo instalados por meio do gerenciador de pacotes conhecido como NPM (*Node Package Manager*), com a possibilidade de implementar diversos componentes *middleware* que facilitem o desenvolvimento de aplicações *web* (WITTERN; SUTER; RAJAGOPALAN; 2015).

os módulos disponíveis no NPM servem para facilitar diversos aspectos relacionados à aplicação, como instalação de dependências, a compilação e a atualização de pacotes (MORAES, 2018).

Segundo Pereira (2016) as principais características do NodeJS, estão relacionados a sua arquitetura não-bloqueante sendo um *Single Thread,* com a programação feita de forma assíncrona com seus recursos compartilhados, com todas as operações de entrada e saída com suas execuções paralelas e sendo orientado a eventos.

### HTML E CSS

HTML é a acrônimo em inglês para *Hyper Text Markup Language,* a linguagem de marcação e estruturação de conteúdo, utilizada para desenvolver *websites* e aceito por diversos navegadores e dispositivos, sendo desenvolvida por Tim Berners-Lee em 1990 (TORRES, 2018).

O CSS é a abreviação do termo em inglês *Cascading Style Sheet*, em português, Folhas de Estilo em Cascata, responsável pela apresentação e estilização dos componentes, como cores, posição, bordas, fontes, tamanhos de letras e imagens, entre outras customizações (SILVA, 2018).

### JAVASCRIPT

A linguagem ECMAScript, popularmente conhecida por *JavaScript*, possibilita a revolução na maneira como a *web* trabalhava, desde sua criação em 1995, por meio da execução do *script* com as instruções no lado do navegador do cliente, sendo implementado pelos principais navegadores, ampliando sua popularização (POWERS, 2010).

De acordo com Bera, Mine e Lopes (2015), *JavaScript* é uma linguagem orientada a eventos ou seja suas operações são executadas somente quando as mesma são requisitadas em alguma parte da aplicação.

### TYPESCRIPT

O *TypeScript* é a evolução da linguagem *JavaScript* desenvolvido pela Microsoft. Projetada para agilizar o desenvolvimento das aplicações, possui recursos comuns das linguagens orientadas a objeto, sendo fortemente marcado, com tipos de dados, pode ser definido interfaces e serviços. Angular utiliza *TypeScript* como linguagem padrão de desenvolvimento (ALVES, 2019).

O *TypeScript* foi usado no desenvolvimento da aplicação do lado servidor, sendo compilado em *JavaScript* utilizando *framework* Gulp.

# RESULTADOS OBTIDOS

Durante o período estágio os conhecimentos adquiridos foram importantes para do desenvolvimento do aluno, muito do que se viu e se aprendeu já havia sido ministrado em disciplinas do curso acadêmico de Engenharia de Software. Fato que reforçou o conhecimento já aprendido em aulas, como, por exemplo, a disciplina de programação Web na qual teve-se contato com várias tecnologias de desenvolvimento Web como HTML, CSS e JavaScript. Outras disciplinas como a de Lógica de Programação, e o conhecimento em gerenciamento de projetos.

Os resultados obtidos com o estágio impactam diretamente no futuro do aluno, o conhecimento adquirido dentro dos projetos em que o aluno participou lhe deu grandes conhecimentos e uma base de como o mercado de trabalho funciona, com isso ele, pode abrir novas oportunidades.

O estágio complementou os conhecimentos adquiridos em aulas e exigiu a busca de outros conhecimentos que não são vistos em aula. Fornecer experiências de como uma empresa funciona e como é o cotidiano de um programador.

# CONSIDERAÇÕES FINAIS

Durante o estágio, o aluno se sentiu muito bem dentro da empresa, com um período de adaptação fácil e rápido, se relacionando bem com colegas de trabalho e em como a empresa leva seu dia a dia. As dificuldades se encontram mais nas áreas técnicas, mas sempre foi muito bem acompanhado pelos seus supervisores.

O estágio fez com que o aluno sentisse a transação entre conhecimento adquirido dentro de sala com a prática do mercado de trabalho, agregado valores a sua pessoa.

Os Conhecimentos adquiridos dentro de sala de aluno sobre programação orientada a objetos, programação web, banco de dados, lógica de programação foram essenciais para realizar as tarefas.

A experiência em participar de um projeto real do mercado de trabalho, agregou valores imprescindíveis para as experiências do aluno, combinando com os valores já adquiridos no curso de Engenharia de Software.

# REFERÊNCIAS

BONFIM, FILIPE LEUCH; LIANG, M. **Aplicações Escaláveis com Mean Stack**. Monografia Graduação, 2014.

FONSECA JR, Luiz Carlos; FONSECA, Marcio Roberto; DE LIMA RANGEL, Hélio Augusto. **An applied study on Angular framework 2**. Unisanta Science and Technology, v. 7, n. 1, p. 18-25, 2018.

HANDY, Alex. **Node.JS pushes JavaScript to the server-side** – SD Times. 2011. Disponível em: <<https://sdtimes.com/JavaScript/node-JS-pushes-JavaScript-to-the-server-side/> >. Acesso em: 29 out. 2019.

HOWS, David; MEMBREY, Peter; PLUGGE, Eelco. **Introdução ao MongoDB**. Novatec Editora, 2015.

MARDAN, Azat. **Express. JS Guide: The Comprehensive Book on Express**. JS. Azat Mardan, 2014.

MORAES, William Bruno. **Construindo aplicações com NodeJS**. Novatec Editora, 2018.

MORGAN, Andrew. **"The Modern Application Stack – Part 1: Introducing The MEAN Stack**." MongoDB, 26 Jan. 2017. Disponível em: <[www.mongodb.com/blog/post/the-modern-application-Stack-part-1-introducing-the-mean-Stack](http://www.mongodb.com/blog/post/the-modern-application-Stack-part-1-introducing-the-mean-Stack)>. Acesso em: 20 set. 2019.

OCTALMIND. **Arquitetura MEAN Stack: Angular, Node, Express e MongoDB**. Disponível em: <<https://blog.octalmind.com/arquitetura-mean-Stack-angular-node-express-e-mongodb/>>. Acesso em: 20 jun. 2019.

PEREIRA, C. R. **Construindo APIs Rest com Node.JS**. 1. ed. São Paulo: Casa do Código, 2016.

POLITOWSKI, Cristiano; MARAN, Vinıcius. **Comparação de Performance entre PostgreSQL e MongoDB**. X Escola Regional de Banco de Dados. SBC, p. 1-10, 2014.

POWERS, S. **Aprendendo JavaScript.** São Paulo: Novatec, 2010.

Rodrigues Gomes de Oliveira, A. e Zuchi, J. (2017) **MEAN STACK**, Revista Interface Tecnológica, 14(1), p. 84-95. Disponível em: <<https://revista.fatectq.edu.br/index.php/interfacetecnologica/article/view/141>> Acesso em: 21 out 2019.

SILVA, Maurício Samy. **Fundamentos de HTML5 e CSS3**. Novatec Editora, 2018.

TOTH, Renato Molina. **Abordagem NoSQL-Uma real Alternativa.** Sorocaba, São Paulo, Brasil: Abril, v. 13, 2011.

WITTERN, Erik; SUTER, Philippe; RAJAGOPALAN, Shriram. **A look at the dynamics of the JavaScript package ecosystem.** In: 2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR). IEEE, 2016. p. 351-361.

1. <https://agendajob.com.br/#/home> [↑](#footnote-ref-0)