

Desenvolvimento de um Sistema Web Utilizando uma Metodologia Autogerenciável

Francisco Moreira Palermo¹, Raíssa Arantes¹,
Rafael de L. Trindade Silva¹, Patrícia Pitthan Barcelos¹

¹Centro de Tecnologia - Universidade Federal de Santa Maria (UFSM)
97.015-900 – Santa Maria, RS – Brazil,

{fmpalermo, rarantes, rdsilva, pitthan}@inf.ufsm.br

Resumo. *A adaptação de novos integrantes em grupos PET apresenta desafios e dificuldades no que concerne à compreensão da sistemática do Programa. Este trabalho descreve um sistema web, desenvolvido por integrantes do PET-SI da UFSM, voltado para atender às necessidades do grupo. O sistema corresponde a um dos produtos gerados na atividade Manage PET, a qual propõe o desenvolvimento de software sob uma metodologia de organização autogerenciável, em que os desenvolvedores possuem autonomia quanto à sua execução. Nesse contexto, o sistema web desenvolvido auxilia no processo de ambientação de novos integrantes ao grupo, ao mesmo tempo em que possibilita que seus desenvolvedores adquiram experiências e novos conhecimentos.*

Abstract. *The adaptation of new members to PET groups presents challenges and difficulties with regard to understanding the systematic of the Program. This work describes a web system, developed by members of the PET-SI from UFSM, aimed at meeting the needs of the group. The system corresponds to one of the products generated in the Manage PET activity, which proposes the development of software under a self-managing organization methodology, in which the developers have autonomy regarding its execution. In this context, the developed web system assists in the process of setting up new members to the group, while enabling its developers to acquire experiences and new knowledge.*

1. Introdução

O *Manage PET*, uma atividade desenvolvida pelo Programa de Educação Tutorial (PET) do curso de Sistemas de Informação da Universidade Federal de Santa Maria (UFSM), tem por objetivo o projeto e a implementação de produtos de software, geridos pelos próprios integrantes do grupo. Estes produtos poderão ser utilizados em futuras atividades e oficinas práticas desenvolvidas pelo grupo. Tendo em vista que o curso de Sistemas de Informação da UFSM não possui estágio obrigatório para a formação dos alunos, o *Manage PET* surge como uma opção para proporcionar aos integrantes do grupo o contato com as diversas etapas de desenvolvimento de software, a saber: especificação, análise, implementação e teste de software.

A atividade *Manage PET* apresenta-se como uma metodologia autogerenciável, na qual os desenvolvedores possuem autonomia quanto ao seu gerenciamento e execução. Assim, no decorrer do desenvolvimento do software, os envolvidos podem exercer diferentes funções dentro da equipe. Essa experiência não é ofertada por nenhuma disciplina do curso de Sistemas de Informação, mas pode ser obtida por meio desta atividade.

Buscando aplicar a metodologia autogerenciável na atividade Manage PET, o grupo PET-SI desenvolveu um sistema denominado “Treinamento Inicial”. O objetivo do sistema é facilitar o acolhimento de novos integrantes ao grupo por meio da disponibilização de materiais informativos. O acompanhamento e a orientação do novo integrante é de extrema importância e pode afetar diretamente suas experiências e seu desempenho. O grupo PET-SI lida com a etapa de integração de novos alunos de forma lenta e gradual, incorporando-os aos poucos nas atividades, fazendo com que aprendam por meio da experiência. A partir do material disponibilizado no sistema, o aluno ingressante pode tirar dúvidas e se preparar para sua atuação no PET-SI. Dessa forma, um aluno que receba o “Treinamento Inicial” se sentirá mais engajado e motivado a desenvolver as atividades e aproveitará melhor as oportunidades oferecidas pelo grupo.

O sistema descrito conta com apresentações, as quais encontram-se em formato de vídeos e slides, das atividades desenvolvidas e seus objetivos, assim como orienta o novo integrante a desenvolvê-las. Além disso, o sistema informa sobre os principais aspectos que podem levar a complicações no seu desenvolvimento, ressaltando as soluções possíveis. O sistema apresenta um mecanismo de progressão, o qual permite que o grupo acompanhe o crescimento do aluno ingressante, diferentemente de outros veículos de armazenamento de conteúdo que não fornecem essa função especializada. Num âmbito mais geral, o sistema desenvolvido possui dois principais objetivos: proporcionar experiência aos alunos desenvolvedores e auxiliar aos novos integrantes a se familiarizarem com o grupo e sua metodologia.

O artigo encontra-se organizado da seguinte forma: a Seção 2 apresenta alguns dos trabalhos relacionados ao escopo desse artigo. A Seção 3 exhibe as motivações para o desenvolvimento desse trabalho. Na Seção 4 é descrita a metodologia utilizada no desenvolvimento do trabalho. A Seção 5 exhibe os resultados preliminares obtidos com o trabalho desenvolvido, enquanto a Seção 6 apresenta as considerações finais e aponta trabalhos futuros.

2. Trabalhos Relacionados

Metodologias autogerenciáveis são o foco de estudo em diversos trabalhos. Em [Santos 2017], foi realizado um estudo de caso do processo de implantação dessa metodologia e suas consequências. Os resultados deste estudo revelaram a percepção que os integrantes da equipe tiveram com a implantação da nova metodologia, que foi positiva em sua maioria, acreditando que trouxe melhorias para o desempenho da equipe. Com essa análise, os autores perceberam que as principais características deste tipo de equipe são a autonomia dos seus membros e a liderança compartilhada, as quais permitem a auto-organização e o gerenciamento de conflitos sem a necessidade de um líder centralizado e hierárquico. A metodologia possibilita um aumento na produtividade, o que permite o crescimento da equipe por meio da liberdade de ideias e estímulo à criticidade.

O trabalho de [Junior and Lopes 2018], embasado sob pesquisas bibliográficas, revelou que a metodologia autogerenciável atua de maneira eficiente em equipes menores, em que os integrantes podem compreender a estrutura destas equipes, sentindo-se seguros e motivados a exporem suas opiniões e ideias para alcançar os resultados esperados. Neste estudo, ressaltou-se a autonomia, a liberdade e a criação como pontos importantes destas equipes, além da importância do perfil dos integrantes do grupo se enquadrar na

metodologia aplicada. Por fim, os autores destacaram que, em um ambiente em que há permissão para a inovação e aplicação dessa metodologia, é possível observar uma maior motivação e eficiência, obtendo, assim, bons resultados.

Ainda, ressalta-se que nos trabalhos mencionados foi argumentada a importância da necessidade de tanto a equipe como a organização compreenderem as políticas e características de uma equipe autogerenciável para o seu sucesso.

Apesar desse enfoque ser publicado em trabalhos relacionados, essa temática não é encontrada aliada à ideia de desenvolvimento de sistemas de treinamento para equipe. Quanto a isso, porém, estudos indicam a importância do treinamento para uma organização. Em [Pereira 2014], os autores comentam que o treinamento é importante pois possibilita um ambiente de inovação e atualização, motivação, qualidade na produção, menor retrabalho, dentre outros pontos, revelando a grande oportunidade que o treinamento pode oferecer para a equipe.

3. Justificativa e Motivação

O trabalho em equipes autogerenciáveis é um dos fatores motivacionais para o uso dessa metodologia como base para a atividade *Manage PET*. De acordo com o trabalho de [Vergara et al. 2006], com um certo grau de liberdade, o indivíduo pode encontrar as melhores maneiras de contribuir com o grupo, ressaltando aspectos em que se sente mais confiante em trabalhar. Os autores de [Vergara et al. 2006] desenvolveram uma pesquisa questionando a motivação de cada indivíduo em atuar em um projeto autogerenciável. A pesquisa envolveu integrantes de grupos que já participaram e não desse tipo de projeto. Os resultados mostraram que a maior parte dos indivíduos analisados na amostra revelou que há mais motivação em trabalhar de forma autogerenciável. Os que não consideraram a equipe ser autogerenciável um fator motivacional atribuíram como razão a dificuldade de comunicação. Sob esse ponto de vista, o *Manage PET* segue a proposta de [Yeatts and Hyten 1998], que sugere a divisão do trabalho em equipes menores, diminuindo o esforço com a comunicação, o que pode interferir positivamente no desempenho da equipe.

Como visto no trabalho de [Vergara et al. 2006], alguns dos entrevistados comentaram que nas equipes autogerenciadas é mais fácil a obtenção de novos conhecimentos e experimentação de novos métodos, contribuindo para o crescimento do indivíduo e melhora no desempenho. Tendo em vista que o *Manage PET* é uma atividade cuja proposta é contribuir para a formação dos alunos, é notório que apresenta, então, uma abordagem bastante positiva para seu propósito, ao adotar uma metodologia de equipe autogerenciável.

Outro aspecto do sistema alvo deste artigo, o “Treinamento Inicial”, além do crescimento dos desenvolvedores, tem o foco na motivação e preparação dos novos integrantes ao ambiente do grupo. De acordo com o estudo de [Santos 2019], através do treinamento do indivíduo, ele desenvolve a sua capacidade profissional, “fortalece e constrói novas habilidades, explora as formas que os talentos contribuem para o todo, além de ampliar a motivação no indivíduo, que é algo extremamente importante para desenvolver a qualidade de sua produtividade em suas atividades laborais”.

De acordo com [Gooch and McDowell 1988], motivação é uma força interior de cada indivíduo e “uma pessoa não consegue jamais motivar alguém, o que ela pode fazer

é estimular a outra pessoa”. Dessa forma, o treinamento criado pelos desenvolvedores do grupo PET-SI serve como uma fonte de incentivo e de motivação, revelando o cuidado especial do grupo no acolhimento aos novos integrantes, contribuindo para um ambiente satisfatório. Ainda de acordo com [Santos 2019], “todo o procedimento de treinamento tem como objetivo aperfeiçoar as competências e habilidades dos sujeitos, e, consequentemente, alcançar o êxito profissional e o desenvolvimento da organização”. Assim sendo, além de ser um fator motivacional, o sistema em questão também servirá para que os novos integrantes obtenham novas habilidades para gerar melhores resultados para o grupo.

4. Metodologia

Para o desenvolvimento do “Treinamento Inicial”, produziu-se um sistema *web* que hospeda vídeos e slides explicativos sobre as atividades e metodologias do grupo PET-SI, auxiliando no processo de adaptação do novo integrante. O sistema também inclui um formulário de avaliação para que os usuários possam deixar comentários e opiniões sobre o material fornecido. Esta Seção descreve as principais etapas do seu processo de desenvolvimento que, conforme [Sommerville 2007], incluem: (i) Definição de um Problema ou Temática; (ii) Elaboração do Planejamento Inicial; (iii) Realização de Reuniões Regulares; (iv) Desenvolvimento do Software; (v) Avaliação e Testes do Software; e (vi) Elaboração da Documentação da Atividade.

4.1. Definição de um Problema ou Temática

Uma das primeiras etapas do desenvolvimento de um sistema para o *Manage PET* é a definição de um problema ou de um tema que será abordado na aplicação a ser desenvolvida. Nesta etapa percebeu-se a necessidade de uma atenção especial aos novos integrantes do grupo PET-SI, os quais frequentemente apresentam dificuldades para se adaptar a rotina do grupo. Assim, o tema definido para o sistema foi “Treinamento Inicial”, cujo propósito é auxiliar e orientar os novos integrantes, ao mesmo tempo em que aprimora as habilidades dos alunos envolvidos no desenvolvimento.

4.2. Elaboração do Planejamento Inicial

A próxima etapa do desenvolvimento consiste no planejamento inicial. Essa etapa leva em consideração as tarefas e a metodologia de desenvolvimento a serem adotadas. Além disso, a etapa prevê a elaboração do roteiro para a gravação dos vídeos que serão incluídos no site a ser desenvolvido.

No desenvolvimento de um sistema, comumente se divide o código em dois módulos distintos, denominados *back-end* e *front-end*. O *back-end* consiste na porção do sistema que não é exposta para o usuário da aplicação. O *front-end*, por sua vez, consiste na parte visual do sistema, a qual interage diretamente com o usuário da aplicação. Esse módulo é responsável por realizar requisições de dados ao *back-end*, efetuar o tratamento das mesmas e exibir os resultados. Apesar de serem desenvolvidos separadamente, os módulos *front-end* e *back-end* se complementam. Neste contexto, os desenvolvedores avaliaram as tecnologias que poderiam ser utilizadas no desenvolvimento do módulo *front-end*, do módulo *back-end* e na base de dados, conforme abordado na Seção 4.4.

4.3. Realização de Reuniões Regulares

A realização de reuniões regulares visa definir comissões e atribuições, além de prazos a serem obedecidos. Essa etapa é realizada em diversos momentos do desenvolvimento

do sistema, a fim de alinhar as ideias, verificar o andamento do sistema e estabelecer a convergência no processo de tomada de decisões.

4.4. Desenvolvimento do Software

O sistema foi construído com base nas fases típicas de um processo de desenvolvimento de software, conforme abordado por [Bezerra 2018]: (i) Levantamento de requisitos; (ii) Projeto/Análise; (iii) Implementação; (iv) Testes; e (v) Produção/Implantação.

De acordo com [Machado 2018], requisitos são características e restrições de que um produto deve ser constituído, estabelecidos por clientes e usuários do sistema, os quais definem as suas propriedades. Dessa forma, a fase de Levantamento de Requisitos se torna essencial para o sucesso do produto, uma vez que refletirá no sistema final. Assim, a primeira fase do desenvolvimento do sistema “Treinamento Inicial” consistiu na elicitação das necessidades dos futuros usuários. Trata-se de uma fase contínua, a qual foi realizada na forma *To-Do List*, conforme mostra a Figura 1. Para o desenvolvimento dessa fase utilizou-se a plataforma *Figma*¹. De acordo com a Figura 1, as tarefas dessa fase são organizadas em: *to do* (tarefas a serem realizadas), *doing* (tarefas em execução), *review* (tarefas em revisão) e *done* (tarefas realizadas). Desse modo, foi possível obter um controle mais apurado sobre as tarefas.



Figura 1. “To Do List”

Com os requisitos estabelecidos, iniciou-se a fase de Projeto. Conforme abordado por [Bezerra 2018], essa fase determina o modo como o sistema irá funcionar para atender aos requisitos, de acordo com a tecnologia existente. Assim, realizou-se a prototipação de baixa fidelidade do sistema, a qual representa as páginas necessárias para a implementação e suas respectivas funcionalidades. Essa tarefa também foi realizada por meio da plataforma *Figma*, dividindo o ambiente em três telas: *Login*, representada pela Figura 2, Cadastro, ilustrada na Figura 3 e Página Principal, exibida na Figura 4.

Após concluída a prototipação, iniciou-se a fase de Implementação do sistema. Conforme [Bezerra 2018], a implementação consiste na tradução do protótipo para um código executável mediado pelo uso de uma linguagem de programação. Sob a proposta do sistema “Treinamento Inicial”, foram desenvolvidos o código de execução do sistema e o material que faria parte deste software.

¹<https://figma.com>



Figura 2. Prototipação do Sistema - Login

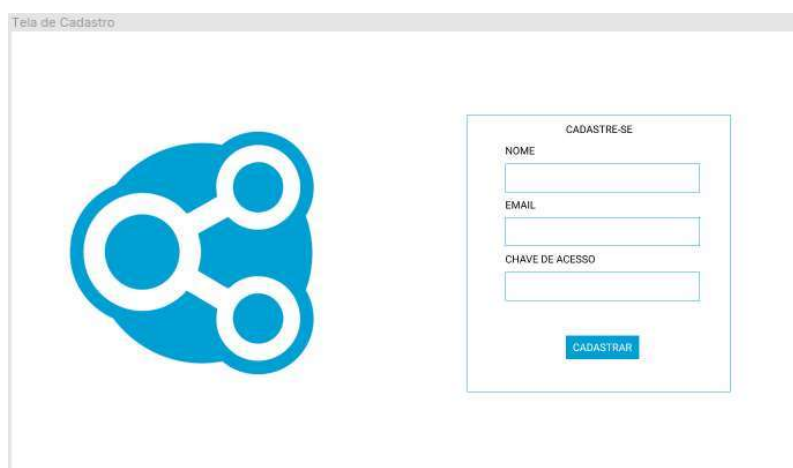


Figura 3. Prototipação do Sistema - Cadastro

Sendo assim, para a preparação do conteúdo do sistema foram produzidos slides e vídeos, seguindo um roteiro pré-estabelecido. O roteiro separou as atividades desenvolvidas pelo PET-SI nos três eixos com os quais o grupo trabalha: Ensino, Pesquisa e Extensão. Além disso, foram definidos os tópicos a serem abordados, como por exemplo, os direitos e deveres dos integrantes do grupo.

Para facilitar o entendimento de cada atividade do grupo, foram utilizados indicadores nos slides preparados, conforme exibe a Figura 5. O objetivo desses indicadores é representar as características das atividades do PET-SI, tais como: quem deve participar, a periodicidade da atividade, as dificuldades de preparo e execução e a sua classificação como atividade externa ou interna ao grupo.

A partir de uma abordagem conhecida como *SPA (Single Page Application)* [Google Analytics 2020], ou Aplicação de Página Única, o sistema “Treinamento Inicial” foi desenvolvido. SPA é um aplicativo que inclui todos os recursos necessários para a navegação no site na carga da página e, à medida que o usuário interage com a mesma, o conteúdo passa a ser carregado dinamicamente, sem necessidade de recarre-

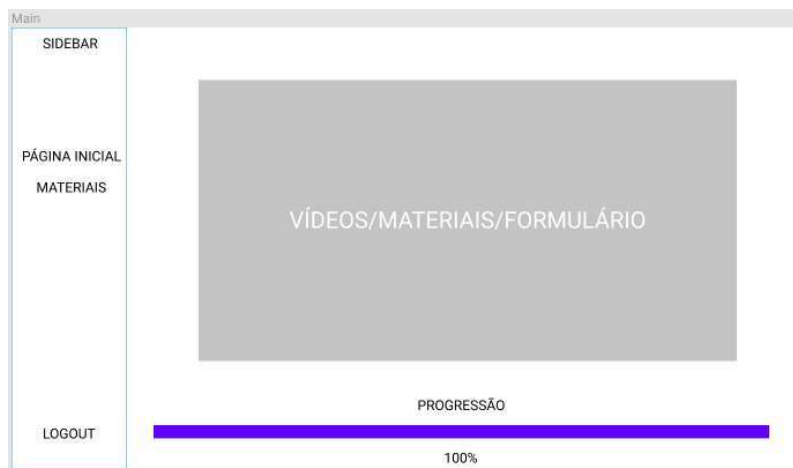


Figura 4. Prototipação do Sistema - Página Principal

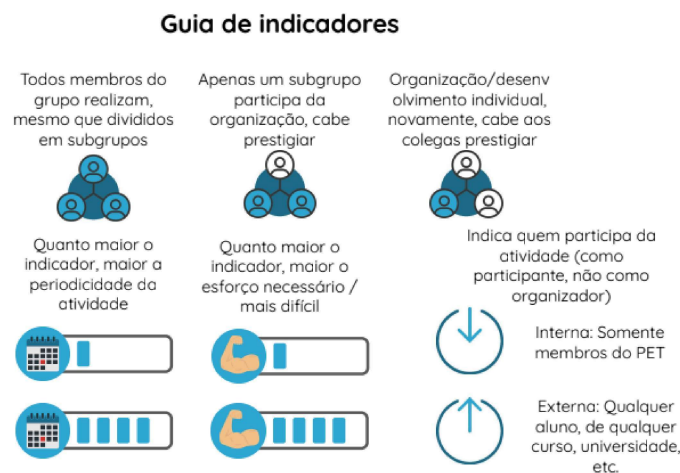


Figura 5. Indicadores

gar uma página completa². Com a abordagem SPA, ilustrada na Figura 6, o sistema não precisa ser recarregado quando alguma requisição é direcionada ao servidor. Assim, os dados HTML e/ou CSS são modificados, porém a página não precisa ser carregada novamente. Isso decorre da ideia de retirar do *back-end* a função de reenviar toda a estrutura da página *web* (HTML, CSS e JavaScript) para o *front-end*, limitando-se somente aos dados que utilizam o formato JSON (*JavaScript Object Notation*), tornando o sistema mais organizado.

No sistema “Treinamento Inicial”, o *back-end* corresponde à parte interna de uma página *web*. Dentre as funcionalidades do *back-end* destacam-se o tratamento de uma conexão com o banco de dados, a autenticação de usuários e a implementação de recursos de criptografia e segurança. O *back-end* implementado possui diversas dependências, a saber: *NodeJS*, *Knex.js*, *SQLite* e *Postgresql*. O *NodeJS* é um ambiente de execução de aplicações *Javascript* ao lado do servidor, ou seja, é possível rodar programas *Javascript*

²<https://developers.google.com/analytics/devguides/collection/analyticsjs/single-page-applications?hl=pt-br>

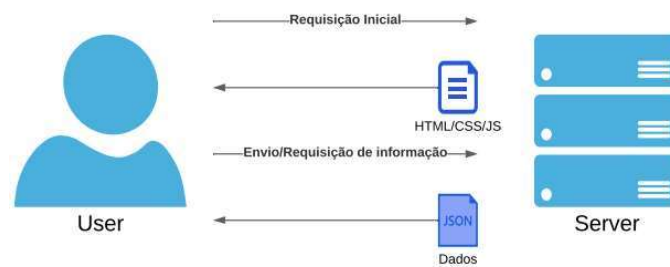


Figura 6. Abordagem SPA (Single Page Application)

em uma máquina de forma separada, sem necessidade de um *browser* para executá-los. Logo, utilizou-se o *NodeJS* para desenvolver as requisições da aplicação, como cadastro e login de usuário, e uni-las ao banco de dados criado, dessa forma abrangendo todo o módulo *back-end*.

O *Knex.js* é um *Query Builder*, o qual, a partir da linguagem *JavaScript*, visa padronizar a forma de escrever requisições para um banco de dados SQL, permitindo a alteração para qualquer tipo de banco de dados SQL, como *MySQL*, *SQLite* e *Postgresql*, sem necessidade de modificar o código. Além disso, o *Knex* é responsável pelo gerenciamento e configuração da base de dados desde a criação, conexão e controle de migrações do banco de dados. Já o *SQLite* e *Postgresql* foram os formatos escolhidos para manter os dados. O *SQLite* limita-se ao ambiente de desenvolvimento, visto que possui menos recursos e é considerado menos seguro que o *Postgresql*³. Para executar o código sem erros nos ambientes de desenvolvimento e de produção (*online*), foram definidas configurações no arquivo *knexfile.js* para selecionar o banco de dados a ser utilizado em cada situação. Dessa forma, é possível realizar alterações futuras, sem prejuízo em qualquer um dos ambientes.

A aplicação *NodeJS* do “Treinamento Inicial” foi dividida em *Routes*, *Controllers* e *Database*, conforme ilustra a Figura 7. A divisão em arquivos visa organizar o código, contribuindo para o entendimento do funcionamento do *back-end*. O arquivo *Routes* contém as rotas do sistema, ou seja, são os caminhos possíveis para acessar determinadas funcionalidades. Os *Controllers*, que interagem diretamente com o banco de dados, são os arquivos onde estão definidas as funções para cada aplicação do programa. Por fim, *Database* corresponde a um diretório que possui o banco de dados, o arquivo de conexão com a base de dados e as migrações. Um exemplo de interação entre os componentes se dá através de uma função, definida em um arquivo *Controller*, que retorna a lista de usuários presente no banco de dados. Assim, definiu-se no arquivo *Routes* que, para trazer essa lista, é necessário acessar a rota */users*, a partir do método *GET*.

Para verificar o funcionamento do *back-end*, utilizou-se a ferramenta *Insomnia*⁴, a qual permite criar requisições ao código *NodeJS* através de um servidor local. A partir da ferramenta *Insomnia*, é possível simular o funcionamento de um sistema *web* no navegador, realizando requisições de cadastro, login, pesquisa e alteração de dados através de dados no formato *JSON*, como exhibe a Figura 8.

³<https://logz.io/blog/relational-database-comparison/>

⁴<https://insomnia.rest/>

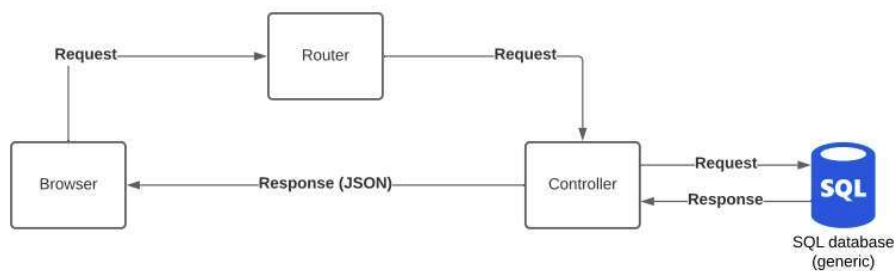


Figura 7. Funcionamento do backend

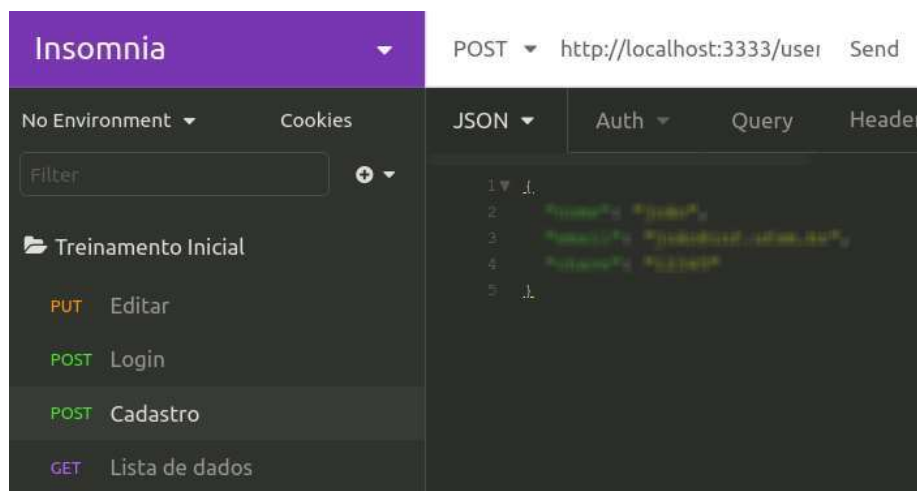


Figura 8. Exemplo de Requisição no Insomnia

O *front-end* do sistema “Treinamento Inicial” foi desenvolvido a partir da biblioteca *React*. Desenvolvida pela equipe do *Facebook*, a biblioteca *React* auxilia na criação de interfaces de usuário e baseia-se na ideia de componentes reutilizáveis. Assim, a interface ou parte da mesma pode ser usada novamente em outros trechos do sistema, diminuindo a quantidade de código escrito. Esses componentes podem conter códigos HTML, CSS e JavaScript.

Com objetivo de permitir que os usuários tenham acesso mais rápido ao sistema, optou-se pela entrada a partir de um código gerado de forma automática e aleatória após o cadastro, chamado *token*. Este código permanece armazenado na memória do navegador (*Local Storage*), possibilitando manter acessos subsequentes do usuário, após o primeiro acesso.

Para garantir mais segurança para o sistema, utilizou-se um sistema de rotas privadas, ou seja, é possível definir as telas que um usuário conectado e um usuário não conectado ao sistema tem acesso, como ilustra a Figura 9. Caso um usuário não conectado tente acessar uma rota sem permissão, ele é diretamente redirecionado à tela inicial.

Para permitir a utilização de vídeos externos, por exemplo um vídeo do *Youtube*, dentro do sistema *web*, utilizou-se uma ferramenta denominada *Iframe*, a qual baseia-se em HTML. Com *Iframe* é possível definir o vídeo a ser mostrado na interface, a partir do atributo *src*, além de diversas configurações, como a largura e altura que o vídeo ocupará na interface, como mostra a Figura 10.

```

export default function Routes() {
  return (
    <BrowserRouter>
      <Switch>
        <Route path="/" exact component={Login} />
        <Route path="/cadastro" component={Cadastro} />
        <PrivateRoute path="/home" component={Dashboard} />
        <PrivateRoute path="/material" component={Material} />
      </Switch>
    </BrowserRouter>
  );
}

```

Figura 9. Definição das rotas do sistema

```

<Row md={12} style={{ marginTop: "2%" }}>
  <Col md={2} />
  <Col md={9}>
    <iframe title="Video 1" width="100%" height="500px" src="https://www.youtube.com/embed/IS01-ici_WA"></iframe>
  </Col>
</Row>
<Row>

```

Figura 10. Utilização do *Iframe*

4.5. Avaliação e Testes do Software

Após a etapa de desenvolvimento, são realizadas avaliações e testes do produto de software, a fim de verificar se o mesmo corresponde à especificação e não apresenta falhas. Submeteu-se o sistema a um teste *E2E (End-to-End)* em que todas as funcionalidades foram avaliadas. Dessa forma, os desenvolvedores percorrem todo o caminho que um usuário faria dentro do sistema, identificando possíveis erros ou problemas. As funcionalidades avaliadas foram: funcionamento do banco de dados, realização de cadastro, login e logout, execução de vídeos dentro sistema e análise da segurança do sistema.

4.6. Elaboração da Documentação da Atividade

A última etapa do sistema corresponde à elaboração da documentação. O relatório deve abordar, entre outros, o que foi desenvolvido, a estrutura do sistema, as dependências tecnológicas utilizadas e a descrição de como podem ser realizadas alterações no mesmo.

Com objetivo de organizar e facilitar o entendimento, foram elaboradas documentações separadas para o *back-end* e para o *front-end*. A documentação relativa ao *back-end* descreve as tecnologias de desenvolvimento e a forma de instalação das mesmas. Além disso, foram detalhados cada diretório e arquivo contido no *back-end*, assim como os procedimentos necessários para realizar alterações em cada funcionalidade. A documentação do *front-end* apresenta as tecnologias utilizadas, a forma como realizar a clonagem do sistema para uma máquina, além da descrição da estrutura e do funcionamento do sistema.

5. Resultados

O desenvolvimento do sistema “Treinamento Inicial” como projeto integrante da atividade *Manage PET* apresenta dois importantes resultados. Por um lado, a implementação do sistema visa fazer com que os novos integrantes consigam entender melhor o funcionamento e os objetivos de cada uma das atividades do PET-SI. Dessa forma, é possível que

em sua integração ao grupo, o aluno encontre facilidades no entendimento e condução das atividades. Por outro lado, os desenvolvedores aprimoraram suas habilidades de planejamento e desenvolvimento de software com a implementação do sistema, aplicando e aprimorando os conhecimentos adquiridos no decorrer do curso.

O “Treinamento Inicial” foi testado e aplicado a dois novos integrantes que ingressaram no grupo em 2020/2. Ambos relataram, através de um formulário de avaliação, que obtiveram um entendimento satisfatório em relação às atividades que o grupo desenvolve. Na percepção desses alunos, sua integração ao grupo ocorreu de forma tranquila, uma vez que se adaptaram rapidamente à sistemática das atividades e à rotina do PET.

Sob o ponto de vista do grupo, a adoção do treinamento possibilitou ganhos no desempenho dos novos integrantes, contribuindo no andamento das atividades. Além disso, observou-se que o tempo para adaptação dos novos alunos ao grupo PET-SI foi significativamente menor em comparação com situações anteriores. Esse aspecto é importante dada a dinamicidade inerente aos grupos PET. Com a aplicação do “Treinamento Inicial”, tanto os novos como os atuais integrantes conseguem se organizar melhor, mantendo a rotina das atividades conforme planejado.

Para os alunos envolvidos na implementação do sistema, a participação em todas as etapas de desenvolvimento foi extremamente importante. Segundo relatos, foi possível aprender novas ferramentas na área de prototipação, conhecer tecnologias e suas constantes alterações e aplicar na prática seus conhecimentos. Além disso, os desenvolvedores tiveram a oportunidade de trabalhar em um projeto de forma autônoma, utilizando para isso os conceitos da metodologia autogerenciável. Tais aspectos são de extrema importância na área da Tecnologia da Informação, já que o mercado exige um amplo conhecimento em tecnologias e valoriza a capacidade de trabalho em equipe.

6. Considerações Finais

Este trabalho apresentou as etapas do desenvolvimento de um sistema *web*, desenvolvido na atividade Manage PET, pertencente ao escopo de atividades do grupo PET-SI da UFSM. A atividade Manage PET visa fornecer aos alunos experiências relacionadas ao processo de desenvolvimento de produtos de software utilizando-se de uma metodologia autogerenciável. A atividade proporciona a troca de informações e de experiências entre os participantes, além de motivá-los a buscarem novos domínios.

Na atividade descrita, desenvolveu-se o sistema *web* denominado “Treinamento Inicial”, cujo objetivo é atender às demandas do PET-SI da UFSM no que concerne à adaptação de novos integrantes ao grupo. O sistema contribuiu para a aprendizagem dos alunos que o desenvolveram, os quais obtiveram experiência prática com diversas tecnologias de desenvolvimento de software. Dessa forma, a participação na atividade contribuiu para sua formação acadêmica e profissional, preparando-os para o caminho de um profissional que trabalha com tecnologia da informação.

A atividade se mostrou de grande importância para o grupo PET, tanto para os que desenvolveram o projeto quanto para os que o utilizaram. O envolvimento do grupo na atividade *Manage PET*, portanto, pode auxiliar significativamente os alunos, à medida que promove aprendizado e permite complementar suas habilidades, bem como facilita a adaptação dos novos integrantes, beneficiários do sistema desenvolvido.

Como trabalhos futuros, o grupo pretende manter o sistema atualizado, agregando funcionalidades, conforme surgirem demandas. Adicionalmente, está previsto o projeto e a implementação de um sistema vinculado à atividade Inclusão Digital, pertencente ao conjunto de atividades desenvolvidas pelo grupo PET-SI, cujo enfoque é dar suporte a usuários de recursos e plataformas de ensino *online*.

Referências

- Bezerra, E. (2018). *Princípios de Análise e Projeto de Sistemas com UML*. Elsevier.
- Gooch, B. G. and McDowell, B. J. (1988). Use anxiety to motivate. *Personnel Journal*, 67(4):50–55.
- Google Analytics (2020). Avaliação de aplicativo de página única. [Online]. Available: <https://developers.google.com/analytics/devguides/collection/analyticsjs/single-page-applications?hl=pt-br>. [Acesso: Out, 2020].
- Junior, E. M. F. and Lopes, L. F. B. (2018). Times autogerenciáveis. *Revista de Pós-Graduação do Centro Universitário Cidade Verde*, 4(2):41–56.
- Machado, F. N. R. (2018). *Análise e Gestão de Requisitos de Software: Onde nascem os sistemas*. Saraiva Educação S.A.
- Pereira, E. S.; Ferreira, M. S. (2014). Treinamento de pessoal. [Online]. Disponível em: <http://catolicadeanapolis.edu.br/biblioteca/wp-content/uploads/2018/05/EDNALVA-DA-SILVA-PEREIRA-2014.pdf>. Faculdade Católica de Anápolis, GO.
- Santos, L. O. (2017). Autogestão por uma equipe de ti de alto desempenho: Um estudo de caso. [Online]. Disponível em: <https://repositorio.ifsc.edu.br>. Instituto Federal Santa Catarina, Florianópolis, SC.
- Santos, E. W. S.; Rodrigues, V. C. (2019). Treinamento e desenvolvimento motivacional. In *Colloquium Socialis*, volume 3, pages 67–72.
- Sommerville, I. (2007). Engenharia de software. *Pearson, Addison Wesley*, 8(9):10.
- Vergara, S. C., Bianco, A., and Gomes, A. P. C. Z. (2006). O trabalho em equipes autogerenciadas e sua influência na motivação dos indivíduos. *Revista Eletrônica de Administração*, 12(1).
- Yeatts, D. E. and Hyten, C. (1998). *High-performing self-managed work teams: A comparison of theory to practice*. Sage Publications, Inc.