

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

HYTHAN MATHEUS CORREIA DE OLIVEIRA

**SISTEMA PARA GESTÃO DE CURSOS DE CURTA DURAÇÃO
PARA A EMPRESA MAXIAMBIENTAL**

PROJETO DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO

GUARAPUAVA
2021

HYTHAN MATHEUS CORREIA DE OLIVEIRA

SISTEMA PARA GESTÃO DE CURSOS DE CURTA DURAÇÃO PARA A EMPRESA MAXIAMBIENTAL

Monografia de Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso, do Curso Superior de Tecnologia em Sistemas para Internet – TSI – da Universidade Tecnológica Federal do Paraná – UTFPR – Campus Guarapuava, como requisito parcial para obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador: Prof. Dr. Diego Marczal
Universidade Tecnológica Federal do Paraná
- Campus Guarapuava

GUARAPUAVA
2021



4.0 Internacional

Esta licença permite remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

RESUMO

OLIVEIRA, Hythan. Sistema para gestão de cursos de curta duração para a empresa Maxi-ambiental. 2021. 19 f. Projeto de Trabalho de Conclusão de Curso de graduação – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2021.

Problemas ambientais culminaram no surgimento de uma área de profissionalização para que leis de preservação sejam cumpridas. Empresas e instituições buscam capacitar esses profissionais, dentre elas está a Maxiambiental. Porém ela conta com um sistema defasado e pouco atrativo, precisando de melhorias e correções. Sendo assim o presente trabalho, busca desenvolver um novo sistema para a empresa Maxiambiental, para que a mesma, possa continuar atraindo cada vez mais profissionais da área ambiental para cursar seus treinamentos.

Palavras-chave: Sistema. Treinamentos. Educação Ambiental.

ABSTRACT

OLIVEIRA, Hythan. System for managing courses for the company Maxiambiental. 2021. 19 f. Projeto de Trabalho de Conclusão de Curso de graduação – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2021.

Environmental problems culminated in the emergence of an area of professionalization so that preservation laws are complied with. Companies and institutions seek to train these professionals, including Maxiambiental. However, it has an outdated and unattractive system, in need of improvements and corrections. Therefore, this work seeks to develop a new system for the company Maxiambiental, so that it can continue to attract more and more professionals from the environmental area to attend their training.

Keywords: System. Training. Environmental education.

LISTA DE FIGURAS

Figura 1 – Sistema atual da empresa Maxiambiental.	3
Figura 2 – Mensagem de erro ao cadastrar com campos inválidos.	4
Figura 3 – Formulário de inscrição muito extenso.	4
Figura 4 – Mensagem de sucesso de inscrição em um treinamento.	5
Figura 5 – Falha ao se cadastrar num curso.	6
Figura 6 – Cadastro de boleto.	6
Figura 7 – Falha ao gerar boleto.	7
Figura 8 – Exemplo de um <i>Wireframe</i>	8
Figura 9 – Exemplo de um sistema monolítico com falha.	9
Figura 10 – Exemplo de um sistema baseado em microsserviços	9
Figura 11 – Ilustração de um Kanban	11
Figura 12 – Interesse no assunto 'Docker'.	12
Figura 13 – Interesse no assunto JavaScript em comparação a PHP e Ruby.	13
Figura 14 – Exemplo de um quadro Kanban.	14
Figura 15 – Fluxograma do processo de desenvolvimento do sistema	15
Figura 16 – Arquitetura novo sistema Maxiambiental.	16

LISTA DE ABREVIATURAS E SIGLAS

CPF	Cadastro de Pessoas Físicas
HTTP	Hypertext Transfer Protocol
UX	User Experience

SUMÁRIO

1 – INTRODUÇÃO	1
1.1 OBJETIVOS	2
1.1.1 Objetivo Geral	2
1.1.2 Objetivos Específicos	2
2 – SISTEMA UTILIZADO PELA MAXIAMBIENTAL	3
2.1 ÁREA PÚBLICA	3
2.2 ÁREA ADMINISTRATIVA	5
3 – REFERENCIAL TEÓRICO	8
3.1 UX DESIGN	8
3.2 SISTEMAS WEB MONOLÍTICOS VS MICROSERVIÇOS	9
3.3 TECNOLOGIAS	10
3.3.1 Kanban	10
3.3.2 Docker	11
3.3.3 Node.js	12
3.3.4 Express e React	13
3.3.5 Postgress	13
4 – METODOLOGIA DE DESENVOLVIMENTO	14
5 – RESULTADOS PARCIAIS	16
5.1 ARQUITETURA PROPOSTA	16
5.2 BACKLOG	16
6 – CONSIDERAÇÕES FINAIS	18
Referências	19

1 INTRODUÇÃO

Desastres ambientais e problemas de poluição, que ameaçam a qualidade de vida, foram os motivos para a Conferência de Estocolmo de 1972, onde foi abordado a necessidade de consciência mundial, por parte de todos os indivíduos, com relação à preservação e conservação do meio ambiente. A partir disso, a expressão Educação Ambiental ganhou destaque no contexto político e pedagógico (RAMOS, 2001). Assim, no Brasil, surgiu uma área, voltada a cuidar das questões ambientais, prevista na Constituição Federal do Brasil, de 1988, art 225.

“Todos têm direito ao meio ambiente ecologicamente equilibrado, bem de uso comum do povo e essencial à sadia qualidade de vida, impondo-se ao poder público e à coletividade o dever de defendê-lo e preservá-lo para as presentes e futuras gerações.”

Desse modo, é de fundamental importância a educação ambiental, seja para instituições ou para civis. Assim, é necessário capacitar profissionais para conhecerem, empregarem e respeitarem as leis ambientais. Para isso, existem diversas categorias de cursos e treinamentos, como, por exemplo, cursos de graduação, pós-graduação e treinamentos de longa e curta duração.

A Maxiambiental¹ é uma empresa especializada em ministrar treinamentos presenciais na área ambiental. A empresa foi fundada em agosto de 2009 e a partir deste período já capacitou mais de 8.000 profissionais da área ambiental, em 150 turmas, com os cursos sendo ministrados em 16 estados e mais de 30 cidades brasileiras. Através de treinamentos de curto período, a empresa proporciona o aperfeiçoamento do conhecimento dos profissionais e estudantes das áreas ambientais, como Engenheiros Agrônomos, Civis, Ambientais, Florestais, Biólogos entre outros. Os cursos são aplicados por profissionais altamente qualificados e que possuem conhecimento teórico e prático, o que facilita no momento de compartilhar experiências com os alunos.

Atualmente a Maxiambiental conta com um sistema para gerenciamento de turmas e alunos que buscam cursar seus treinamentos, porém o mesmo encontra-se desatualizado. Possuindo um layout ultrapassado de forma a não chamar atenção dos usuários, além de não ser tão intuitivo, o que dificulta a usabilidade. Percebe-se ainda a presença de links duplicados ou inexistentes, páginas com conteúdo em branco e a falta de um mecanismo de pesquisa na área dos cursos. Além disso, na área administrativa, é necessário algumas automações e melhorias como por exemplo, na geração de certificados, visto que atualmente este processo é realizado de forma manual.

Por estes motivos, a criação de um novo sistema é essencial para que o gerenciamento torne-se mais ágil, bem como, é preciso melhorar o layout e organização do conteúdo, para que os usuários possam ter uma experiência mais agradável ao utilizar o sistema. Desta forma, esse

¹<https://maxiambiental.com/>

trabalho espera contribuir para que o novo sistema traga mais agilidade aos administradores na gestão dos cursos e alunos e, melhore o engajamento e experiência dos usuários.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver um sistema para o gerenciamento de treinamentos para a empresa Maxiambiental utilizando uma arquitetura distribuída em microsserviços.

1.1.2 Objetivos Específicos

- Projetar o layout seguindo as diretrizes de usabilidade e simplicidade de *User Experience Design*;
- Desenvolver uma aplicação administrativa para gerenciamento de cursos, turmas e conteúdos;
- Desenvolver uma aplicação para a área pública que possibilite ao usuário cadastrar e gerenciar suas informações.

2 SISTEMA UTILIZADO PELA MAXIAMBIENTAL

Atualmente a empresa Maxiambiental possui um sistema, para a gestão de turmas e alunos. Este capítulo apresenta uma breve descrição da área pública e da área administrativa do sistema.

2.1 ÁREA PÚBLICA

Por se tratar de um sistema que foi desenvolvido em 2009, a área pública conta com um visual desatualizado, dando a impressão de um site antigo, conforme mostrado na Figura 1. Além disso percebe-se que os formulários não contam com nenhuma formatação nos campos, para evitar que seja digitado alguma informação incorreta, como por exemplo, uma letra no meio do CPF, e a notificação de um dado inválido só ocorre ao tentar finalizar o envio, conforme exemplificado na Figura 2.



Figura 1 – Sistema atual da empresa Maxiambiental.

Fonte: Maxiambiental

Gerenciamento de Projetos com MS Project

Inscrição não pôde ser salvo: 5 erros.

Por favor, cheque os seguintes campos:

- CPF numero invalido
- Email não é válido
- Forma de Pagamento não pode ser vazio
- Email Secundário não é válido
- Declaro que li e aceito as condições para realização do curso precisa ser aceito

Nome:* CPF:*
 Teste Inscricao 08297222977asdads

Rua:* Número.* Bairro.* Complemento:
 Rwadawd awd dawda dawdaw awdaw

CEP:* Estado.* Cidade:*
 85040-360 Rio Grande do Norte Bom Jesus

Telefone Celular.* Telefone: Telefone Comercial:
 (12) 31231-2312 (12) 3123-1231 (31) 2312-3123

Email.* Email Secundário:
 dawda@#2312.com 321312@.com123

Figura 2 – Mensagem de erro ao cadastrar com campos inválidos.

Fonte: Maxiambiental

Outro ponto, com relação aos formulários, é o fato de todos os campos serem exibidos em uma única página, o que dá a impressão de ser muito extenso para o usuário, conforme a Figura 3.

Ficha de inscrição para o curso Perícia Ambiental

Nome:* CPF:*

Rua:* Número.* Bairro.* Complemento:

CEP:* Estado.* Cidade:*

Telefone Celular.* Telefone: Telefone Comercial:

Email.* Email Secundário:

Formação: Empresa em que trabalha:

Como Ficou Sabendo do Curso?*

Você já participou de algum curso/palestra da Maxiambiental?
 Sim Não

Categoria:* Forma de Pagamento:*
 Escolha Escolha

Deseja fazer alguma observação?

Figura 3 – Formulário de inscrição muito extenso.

Fonte: Maxiambiental

O sistema não conta com uma área onde o usuário possa estar acessando para atualizar ou corrigir de seus dados, ou ainda, cancelar sua inscrição de algum treinamento. Pela falta

desse cadastro de usuário, a cada novo curso que, por exemplo, o usuário X venha a realizar, precisará cadastrar suas informações novamente, gerando um retrabalho.

O sistema também não possui uma forma de realizar o pagamento, sendo necessário que o usuário aguarde um e-mail com o boleto, ou outra forma de contato, para finalizar e confirmar sua inscrição conforme mostrado na Figura 4.

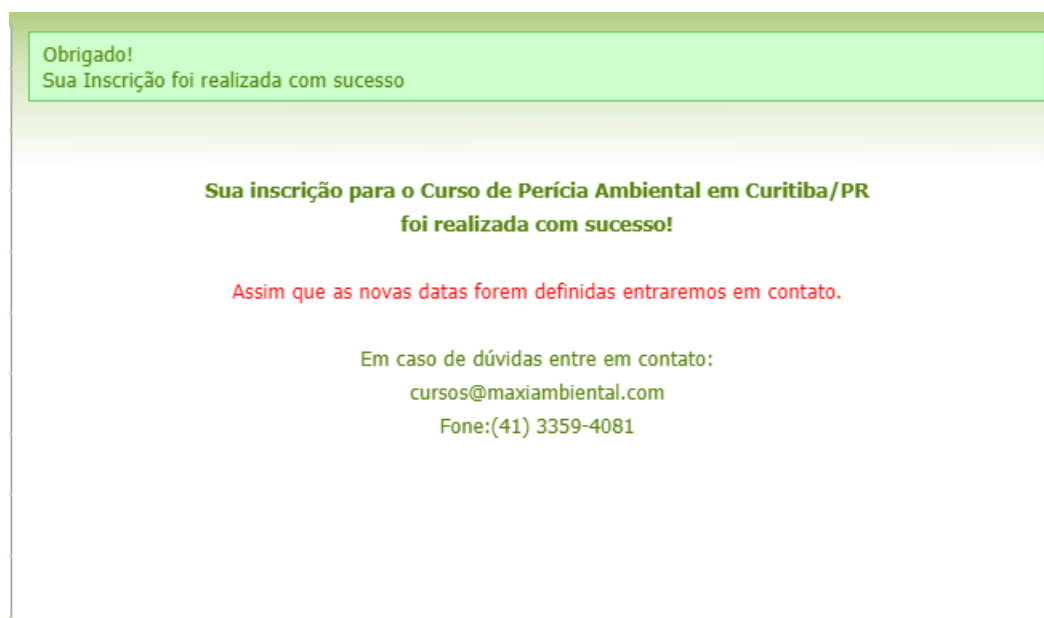


Figura 4 – Mensagem de sucesso de inscrição em um treinamento.

Fonte: Maxiambiental

2.2 ÁREA ADMINISTRATIVA

O espaço do sistema destinado ao administrador para gerenciamento dos conteúdos do site, bem como dos cursos e alunos, também apresenta-se com um visual desatualizado, além de contar com o mesmo problema, de falta de formatação, nos campos dos formulários.

Uma das falhas que pode ser explorada, durante o cadastramento de turmas, é a possibilidade de cadastrar uma nova turma sem definir um meio de pagamento ou definindo uma data que já tenha passado, o que vai impossibilitar que o usuário final consiga se cadastrar no treinamento, conforme exemplificado na Figura 5.

Como Ficou Sabendo do Curso?:*

Você já participou de algum curso/palestra da Maxiambiental?

Sim Não

Categoria:*

 ▼

Forma de Pagamento:*

 ▼

Deseja fazer alguma observação?

Condições para realização do curso:

Figura 5 – Falha ao se cadastrar num curso.

Fonte: Maxiambiental

A Figura 6 demonstra que para chegar até o candidato o boleto, para pagamento e confirmação da inscrição, é necessário que o administrador cadastre o valor e gere o boleto, de forma manual, para só então enviar para o usuário. Este processo permite ainda o cadastro de um valor inválido para o boleto, o que acarretará numa tela de erro, conforme a Figura 7.

Dados para geração do boleto para [REDACTED]

Pedido

Valor

Data do vencimento

Observação 1

Observação 2

Observação 3

Dados do cliente

Nome

Figura 6 – Cadastro de boleto.

Fonte: Maxiambiental

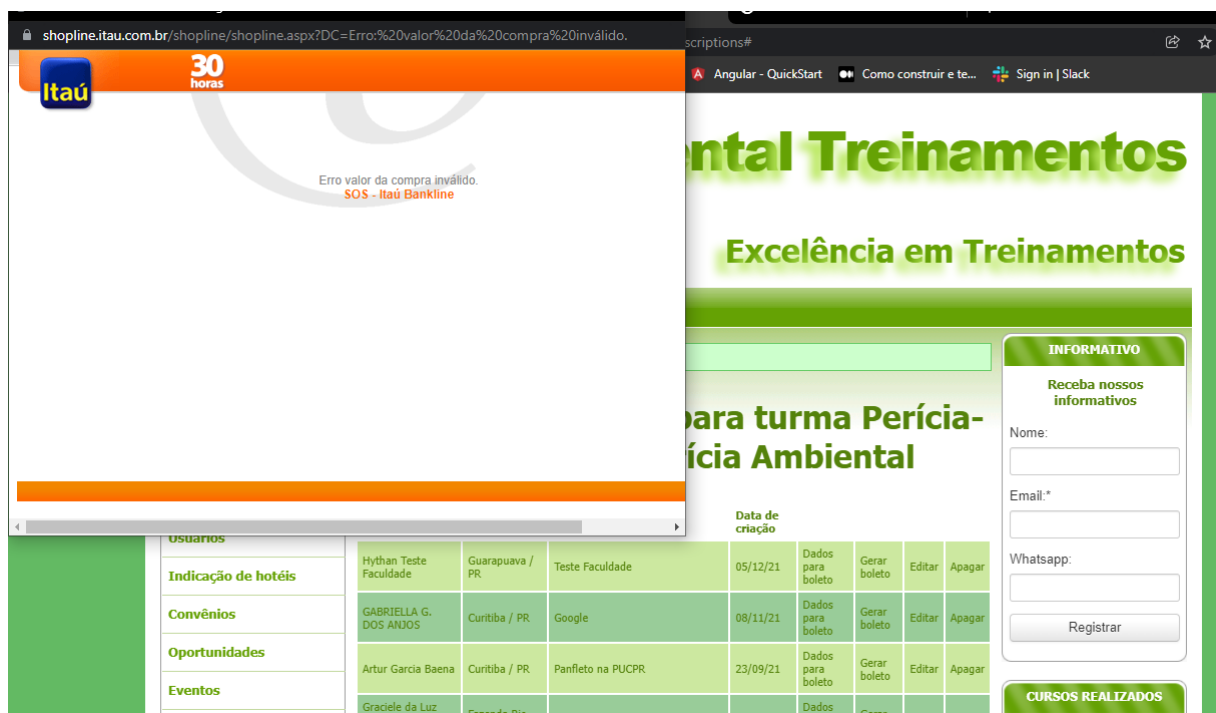


Figura 7 – Falha ao gerar boleto.

Fonte: Maxiambiental

3 REFERENCIAL TEÓRICO

3.1 UX DESIGN

O aumento da competitividade e a preocupação em atrair clientes, fez com que surgisse uma área focada na experiência do usuário ou *User Experience* (UX), em inglês. Tal área preocupa-se com todo o processo de uso de um determinado produto, e o mesmo se aplica a sites e sistemas, não limitando-se a uma aparência bonita, mas também, na forma como o usuário utilizará o sistema e chegará ao seu objetivo. (TEIXEIRA, 2014)

Ainda segundo Teixeira (2014) a simplicidade e usabilidade, algumas das diretrizes do UX Design, são de suma importância para garantir o sucesso de um produto, preocupando-se em colocar o usuário final no centro do processo.

Algumas formas de validar um produto é através de *Wireframes*, que consiste em ser um guia visual que representa um esboço de uma página, contendo a estrutura e organização do conteúdo, conforme mostra a Figura 9. Além de fazer um teste de usabilidade, que consiste em entrevistar o usuário durante a utilização de uma funcionalidade, ou do sistema como um todo, garante uma validação se o fluxo, aparência e funcionalidade estão de acordo com o que ele procura.(TEIXEIRA, 2014)

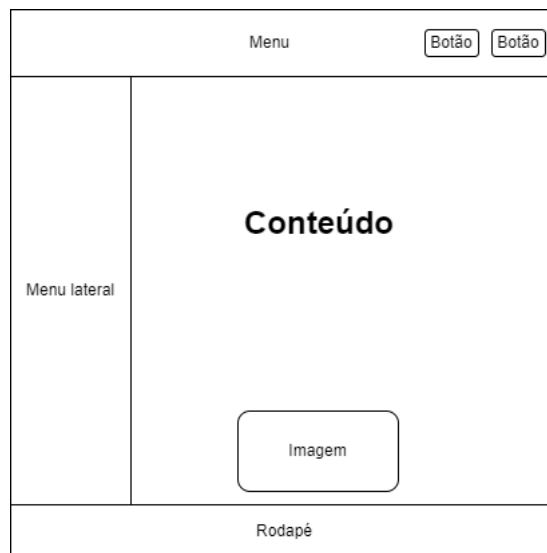


Figura 8 – Exemplo de um *Wireframe*

Fonte: O Autor (2021).

Atualmente a preocupação com a experiência do usuário ao desenvolver um produto é fundamental para garantir o sucesso ou fracasso do mesmo.

3.2 SISTEMAS WEB MONOLÍTICOS VS MICROSERVIÇOS

Um sistema monolítico possui sua arquitetura implementada em um único processo, ou seja, todas as funcionalidades da aplicação estão acopladas em um único fluxo, conforme mostrado na Figura 10. (SILVA, 2016). Por exemplo, se o processo C parar de funcionar toda aplicação irá parar de funcionar.

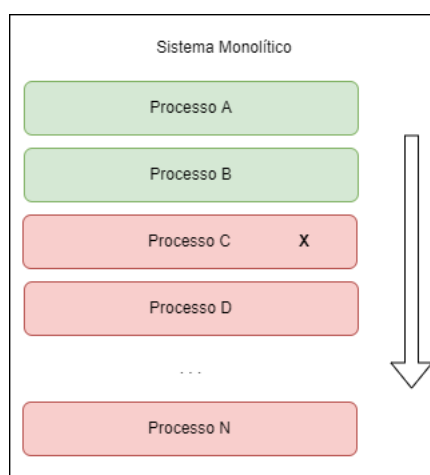


Figura 9 – Exemplo de um sistema monolítico com falha.

Fonte: O Autor (2021).

Ainda em um sistema monolítico, a cada nova funcionalidade desenvolvida, precisa ser testado o sistema como um todo para garantir que alguma outra parte não tenha parado de funcionar, também, a cada nova funcionalidade a complexidade do código aumenta, já que tudo está acoplado. (RODRIGUES, 2017)

Já em uma aplicação com sua arquitetura baseada em microsserviços, cada funcionalidade é empacotada independente das demais, evitando que se um processo parar de funcionar a aplicação continuará rodando, conforme mostrado na Figura 11. (SILVA, 2016)

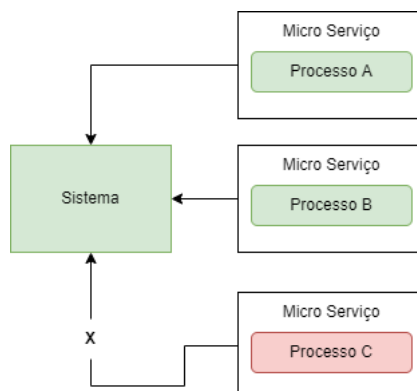


Figura 10 – Exemplo de um sistema baseado em microsserviços

Fonte: O Autor (2021).

Nesse tipo de arquitetura a implementação de novas funcionalidades é facilitada, uma vez que podem ser utilizadas linguagens diferentes para implementação, e não há preocupação com o comportamento do resto do sistema. Outro ponto positivo é visto na hora de escalonar a aplicação, já que podem ser escolhidos as partes, ou serviços, que precisam ser melhorados. (RODRIGUES, 2017)

Ainda segundo RODRIGUES (2017) como desvantagem, um sistema em microsserviços, está no gerenciamento de dependências de cada microsserviços, uma vez que conforme o sistema cresce, cresce também a quantidade de microsserviços implementados. Outra preocupação é na comunicação entre eles, garantir que o serviço A comunique-se com o serviço B.

3.3 TECNOLOGIAS

Nesta seção serão abordados as tecnologias utilizadas para o desenvolvimento deste trabalho.

3.3.1 Kanban

No desenvolvimento de aplicações é de suma importância que o desenvolvedor, ou a equipe, que o estiver fazendo, busque por utilizar tecnologias e ferramentas que otimizem e organizem os processos, a fim de fazer com que prazos de entrega sejam cumpridos. A implementação de uma metodologia ágil é uma forma de auxiliar para que isso ocorra (ROSA, 2014).

Dentre os métodos ágeis existentes, existe o artefato Kanban, que em tradução literal significa “cartão” em japonês, e surgiu dos sistemas de fábricas, onde consiste em gerenciar o fluxo de trabalho, separando em cartões coloridos as tarefas, ou processos, que devem ser feitos e adicionados num quadro, dessa forma todo o processo fica visível.(MARIOTTI, 2008)

O processo tradicional de utilização do Kanban é separar as tarefas em cartões, que são adicionados em um quadro, que possui divisões em colunas, conforme mostrado na Figura 8.

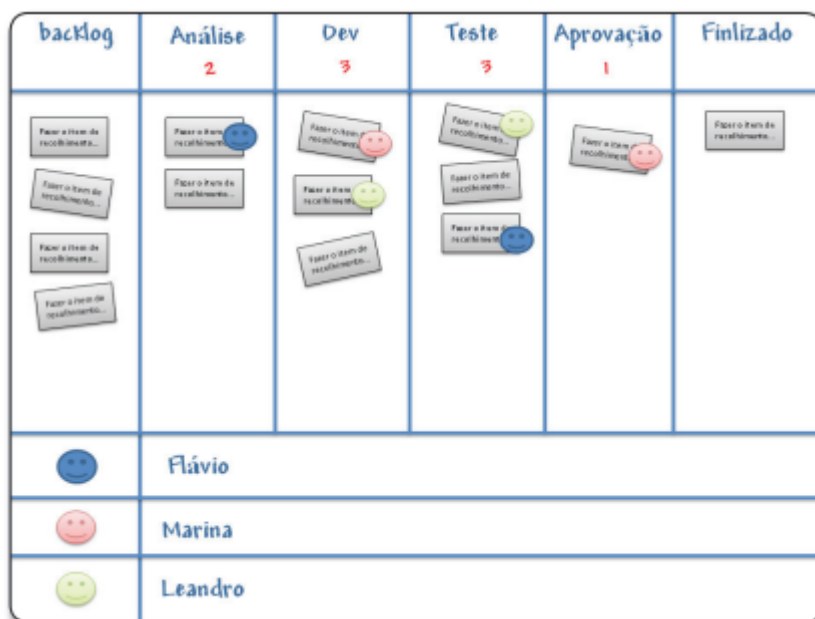


Figura 11 – Ilustração de um Kanban

Fonte: MARIOTTI (2008)

Ainda segundo o artigo de MARIOTTI (2008) o Kanban garante uma transparência que torna fácil a identificação de gargalos e tarefas que estão há muito tempo em desenvolvimento ou na fila. Desta forma, toda a equipe fica ciente do processo em si, e pode tomar as devidas providências.

3.3.2 Docker

Docker é uma ferramenta que virtualiza contêineres, e estes, possuem seus próprios softwares e arquivos de configuração independentes, podendo se comunicar entre si, e foi criado em 2013 por Solomon Hykes. (DOCKER, 2013)

Docker se tornou uma tendência, quase que uma obrigatoriedade entre o meio de desenvolvedores, como mostrado na Figura 12. A procura pela ferramenta aumentou consideravelmente nos últimos anos, de acordo com o resultado do Google Trends¹. Afinal de contas, o Docker possibilita que o desenvolvimento e a replicação de um sistema, se torne algo mais prático, uma vez que uma aplicação pode possuir diversos contêineres e estes as mais diversas linguagens, tecnologias e configurações, e todos podem conversar entre si para o funcionamento da aplicação como um todo. (GOMES, 2017)

¹Google Trends é uma ferramenta do Google que mostra os termos procurados em um determina período de tempo (GOOGLE, 2021)



Figura 12 – Interesse no assunto 'Docker'.

Fonte: Google Trends.

E como gerenciar tantos contêineres durante o desenvolvimento? Para fazer a orquestração de todos os contêineres necessários para o desenvolvimento de uma aplicação existe o Docker Compose, que é uma ferramenta utilizada para definir e executar diversos contêineres através de um arquivo de configuração (DOCKER, 2013).

3.3.3 Node.js

Node.js é uma plataforma para execução de JavaScript¹ no *back end* criado por Ryan Dahl em 2009 (OPENJS, 2009).

E por que utilizar Node.js? Uma grande vantagem para se utilizar Node.js para desenvolvimento de aplicações é o fato de utilizar JavaScript como linguagem, isso diminui a curva de aprendizagem, uma vez que a linguagem é a mesma do *client-side*. Outra vantagem é uma comunidade extremamente ativa, ou seja, o desenvolvedor facilmente encontra respostas para possíveis dúvidas que possam surgir durante o desenvolvimento de uma aplicação (PEREIRA, 2016).

Conforme mostra o gráfico do Google Trends na Figura 13, a busca por JavaScript, linguagem utilizada no Node.js, superou, nos últimos 12 meses, a busca de outras linguagens de programação como PHP e Ruby, o que reforça a questão de uma comunidade mais ativa.

¹JavaScript é uma linguagem de programação interpretada e baseada em objetos, utilizada para rodar scripts em páginas Web (MOZILLA, 2021)

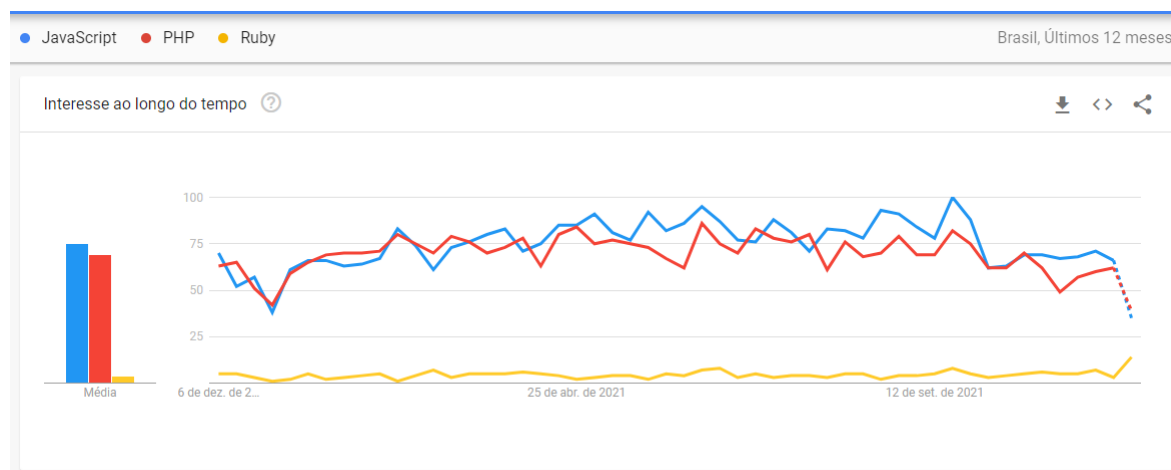


Figura 13 – Interesse no assunto JavaScript em comparação a PHP e Ruby.

Fonte: Google Trends.

3.3.4 Express e React

Express é um framework back-end minimalista e flexível, que conta com recursos robustos para o desenvolvimento de aplicações Web JavaScript com o Node.js ([OPENJS, 2010](#)). Conta com recursos como: roteamento robusto, foco em alto desempenho, alta cobertura de testes, auxiliares de HTTP (*Hypertext Transfer Protocol*), entre outros ([OPENJS, 2010](#)).

E por que utilizar Express como framework? O Express permite um desenvolvimento mais ágil, uma vez que é simples de ser estudado, e trás uma liberdade maior na forma como o desenvolvedor pode organizar seu código ([PEREIRA, 2016](#)).

Pensando no desenvolvimento front-end de uma aplicação existe o React, uma biblioteca JavaScript criada pelos desenvolvedores do Facebook, que tem como objetivo principal permitir o desenvolvimento de interfaces interativas de maneira facilitada ([FACEBOOK, 2013](#)). É uma tecnologia que procura trazer velocidade, simplicidade e escalabilidade e que conta com a possibilidade de criação e reutilização de componentes ([LINS, 2019](#)).

3.3.5 Postgress

Postgres é um Sistema de Gerenciamento de Banco de Dados (SGBD) Relacional, ou seja, ele é responsável pelo armazenamento e consulta de dados que forem salvos por alguma aplicação, gratuito e de código aberto ([POSTGRESQL, 1996](#)).

O Postgres conta com recursos que vão garantir integridade de todos os dados que forem salvos, sendo ideal para aplicações de quaisquer porte, sejam elas, pequenas, médias ou grandes ([MILANI, 2008](#)). Dentre os recursos pode-se destacar: integridade de dados (restrições de exclusão, bloqueios explícitos, bloqueios consultivos, etc), desempenho (indexação, indexação avançada, particionamento de tabela, etc), confiabilidade (log de gravação antecipada, recuperação pontal, etc), entre outros ([POSTGRESQL, 1996](#)).

4 METODOLOGIA DE DESENVOLVIMENTO

A primeira etapa para o início do desenvolvimento do projeto será a realização de uma reunião com o administrador do sistema, a fim de levantar os principais problemas e dificuldades enfrentados na sua utilização. Todos os problemas e melhorias que forem apontadas na reunião serão divididos em cartões e inseridos em um quadro Kanban, através da ferramenta Trello, conforme exemplificado na Figura 14.

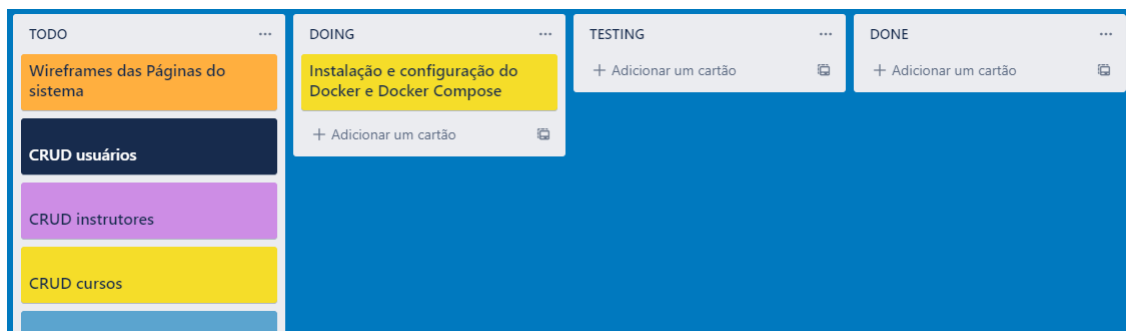


Figura 14 – Exemplo de um quadro Kanban.

Fonte: O Autor (2021).

Durante o desenvolvimento do projeto todas as tarefas que devem ser feitas, ficarão na aba *TODO*, em ordem de prioridade, quando iniciada a implementação de uma tarefa, esta é movida para aba *DOING*. Quando finalizada a tarefa, esta passará para a aba *TESTING*, onde o administrador realizará o teste da funcionalidade desenvolvida e então sua aprovação, e caso não seja aprovada retornará para *DOING* com as correções solicitadas. Após aprovada a tarefa será movida para a aba *DONE*.

Antes de iniciado a implementação do sistema, será configurado o ambiente de desenvolvimento, para isso será utilizado o Docker juntamente com o Docker Compose para a orquestração dos diversos contêineres necessários. Neste ambiente será configurado o Nodejs, o banco de dados Postgres e configurados os frameworks React e o Express e demais tecnologias.

Após todas as configurações, o processo de desenvolvimento das tarefas se dará por implementação de cada funcionalidade, que será submetida a testes de código. Aprovada, essa nova funcionalidade, será carregada no ambiente de teste, onde será avaliado pelo administrador do sistema, e por fim, será adicionado ao ambiente de produção, conforme exemplificado no fluxograma da Figura 15. Não serão utilizadas métricas para avaliar questões UX, porém será seguido as diretrizes de simplicidade e usabilidade, escolhidas neste projeto.

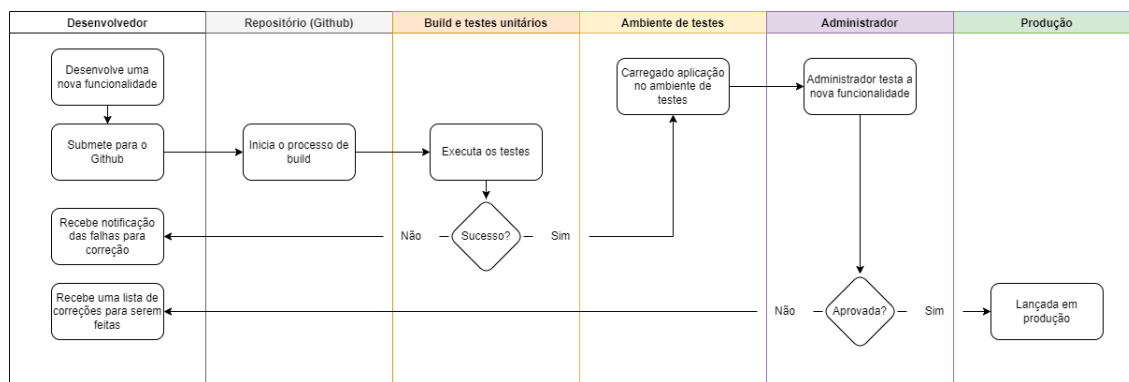


Figura 15 – Fluxograma do processo de desenvolvimento do sistema

Fonte: O Autor (2021).

5 RESULTADOS PARCIAIS

O presente capítulo tem por objetivo explicar a respeito da arquitetura inicial pensada para o desenvolvimento do novo sistema da Maxiambiental, bem como do *backlog* de tarefas iniciais levantadas.

5.1 ARQUITETURA PROPOSTA

Conforme exemplificado na Figura 16, o sistema contará com uma divisão de aplicações para área pública e área administrativa, além de contar com uma API para gerenciamento de pagamentos. Possui também serviços para gerenciamento de usuários bem como para geração de diplomas de alunos que tenham finalizados quaisquer treinamentos. O sistema como um todo possui suas aplicações divididas em back-end e front-end.

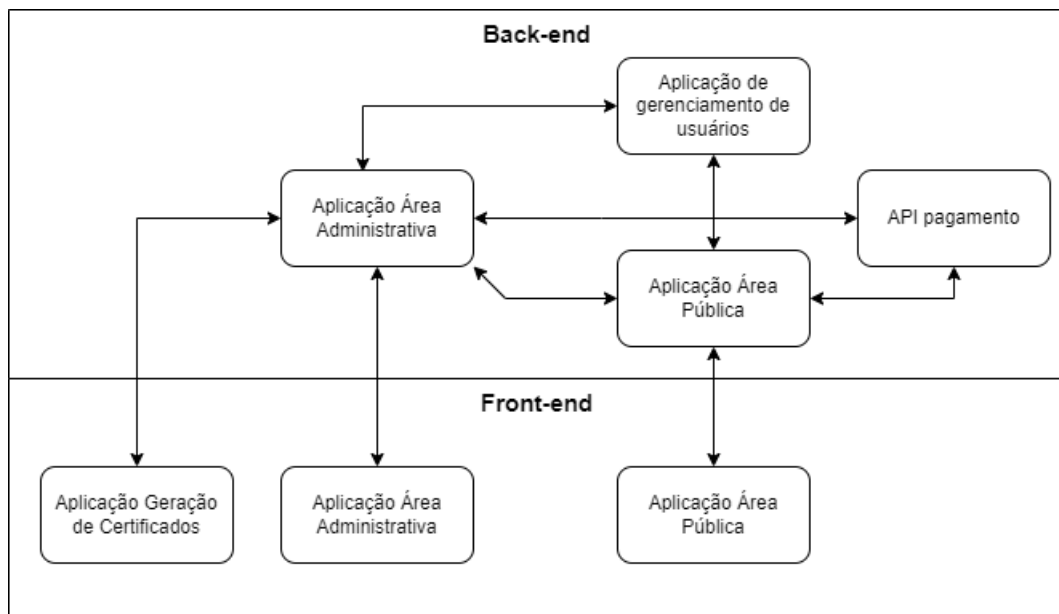


Figura 16 – Arquitetura novo sistema Maxiambiental.

Fonte: O Autor (2021).

Sabe-se que a organização da arquitetura e divisão de responsabilidades das aplicações, podem ser alteradas no decorrer do desenvolvimento, caso se faça necessário.

5.2 BACKLOG

O *backlog* são as tarefas que serão feitas durante o desenvolvimento do projeto, elas ficarão na aba *TODO* do quadro Kanban. A princípio foram levantadas as seguintes histórias de uso para facilitar a criação do *backlog* inicial:

- Eu enquanto administrador quero cadastrar cursos para que novos alunos possam cursar;

- Eu enquanto administrador quero cadastrar outros administradores para que possam auxiliar na gestão do sistema;
- Eu enquanto administrador quero cadastrar instrutores para ministrarem os treinamentos;
- Eu enquanto usuário quero poder me cadastrar no sistema para que possa cursar vários treinamentos;
- Eu enquanto usuário quero editar meus dados para que possa atualizar alguma informação;
- Eu enquanto administrador quero cadastrar cursos para que novos alunos possam cursar;
- Eu enquanto administrador quero gerar certificados de forma automática para que possa entregar ao aluno no final do treinamento;

6 CONSIDERAÇÕES FINAIS

Com a realização desse trabalho, espera-se alcançar com êxito a entrega de um sistema que possa atender as demandas na gestão de alunos e cursos, a fim de facilitar tanto o gerenciamento, quanto o acesso de usuários. Um sistema que facilite e apresente informações e uma aparência mais moderna, além de criar um ambiente agradável, possibilitará maior agilidade e assertividade por parte do usuário em sua utilização.

Pensando nesses detalhes, o novo sistema buscará corrigir alguns problemas atuais que foram levantados neste trabalho, além de trazer um visual que busque captar a atenção do usuário final, e funcionalidades que agilizem a usabilidade do sistema e automatizem processos que atualmente são manuais.

Referências

- DOCKER, I. **Docker**. 2013. Disponível em: <<https://www.docker.com/>>. Acesso em: 29 de Outubro de 2021. Citado 2 vezes nas páginas 11 e 12.
- FACEBOOK, O. S. **React**. 2013. Disponível em: <<https://pt-br.reactjs.org/>>. Citado na página 13.
- GOMES, R. **Docker para desenvolvedores**. [S.l.]: Leanpub, 2017. Citado na página 11.
- GOOGLE. **Google Trends**. 2021. Disponível em: <<https://trends.google.com.br/>>. Citado na página 11.
- LINS, G. d. S. Utilizando reactjs para o desenvolvimento de um sistema de: alocação e reserva de salas no campus da ufc em quixadá. 2019. Citado na página 13.
- MARIOTTI, F. S. Kanban: o ágil adaptativo: Introduzindo kanban na equipe ágil. Revista: Engenharia de Software Magazine., 2008. Disponível em: <<http://www.garcia.pro.br/EngenhariadeSW/artigosMA/A6%20-%2045-6-%20Kanbam.pdf>>. Acesso em: 26 de novembro de 2021. Citado 2 vezes nas páginas 10 e 11.
- MILANI, A. **PostgreSQL-Guia do Programador**. [S.l.]: Novatec Editora, 2008. Citado na página 13.
- MOZILLA, R. d. D. **JavaScript**. 2021. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em: 30 de Novembro de 2021. Citado na página 12.
- OPENJS. **Node.js**. 2009. Disponível em: <<https://nodejs.org/en/>>. Acesso em: 29 de Outubro de 2021. Citado na página 12.
- OPENJS, F. **Express**. 2010. Disponível em: <<https://expressjs.com/>>. Citado na página 13.
- PEREIRA, C. R. **Construindo APIs REST com Node.js**. [S.l.]: Casa do Código, 2016. Citado 2 vezes nas páginas 12 e 13.
- POSTGRESQL, G. d. D. G. **Postgres**. 1996. Disponível em: <<https://www.postgresql.org/about/>>. Citado na página 13.
- RAMOS, E. C. Educação ambiental: origem e perspectivas. **Educar em Revista**, SciELO Brasil, p. 201–218, 2001. Citado na página 1.
- RODRIGUES, A. B. **Uma abordagem gradativa de modernização de software monolítico e em camadas para SOA**. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2017. Citado 2 vezes nas páginas 9 e 10.
- ROSA, A. A. Scrum e kanban para o gerenciamento de comunicacao em equipes de desenvolvimento de software. 2014. Citado na página 10.
- SILVA, A. V. Desenvolvimento de aplicações com base na arquitetura de micro serviços. 2016. Disponível em: <<https://cepein.femanet.com.br/BDigital/arqTccs/1411420621.pdf>>. Acesso em: 19 de novembro de 2021. Citado na página 9.
- TEIXEIRA, F. **Introdução e boas práticas em UX Design**. [S.l.]: Editora Casa do Código, 2014. Citado na página 8.