



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
UNIDADE ACADÊMICA DE SISTEMAS E COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

RAFAEL DE ARAÚJO MACIEL

**TATTOO STUDIO:
HUB DIGITAL PARA ESTÚDIOS DE TATUAGEM**

CAMPINA GRANDE - PB

2019

RAFAEL DE ARAÚJO MACIEL

**TATTOOStudio:
HUB DIGITAL PARA ESTÚDIOS DE TATUAGEM**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em Ciência
da Computação.**

Orientador: Professor Dr. Tiago Lima Massoni.

CAMPINA GRANDE - PB

2019



M152t Maciel, Rafael de Araújo.
TattooStudio : Hub digital para estúdios de tatuagem.
/ Rafael de Araújo Maciel. - 2019.

12 f.

Orientador: Prof. Dr. Tiago Lima Massoni.

Trabalho de Conclusão de Curso - Artigo (Curso de Bacharelado em Ciência da Computação) - Universidade Federal de Campina Grande; Centro de Engenharia Elétrica e Informática.

1. Tatuagem. 2. Estúdios de tatuagem. 3. Desenvolvimento web. 4. Estudo de usabilidade. 5. Post-Study System Usability Questionnaire - PSSUQ. 6. Progressive Web App. I. Massoni, Tiago Lima. II. Título.

CDU:004(045)

Elaboração da Ficha Catalográfica:

Johnny Rodrigues Barbosa
Bibliotecário-Documentalista
CRB-15/626

RAFAEL DE ARAÚJO MACIEL

TATTOO STUDIO:

HUB DIGITAL PARA ESTÚDIOS DE TATUAGEM

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

**Professor Dr. Tiago Lima Massoni
Orientador – UASC/CEEI/UFCG**

**Professora Dra. Lívia Maria Rodrigues Sampaio Campos
Examinadora – UASC/CEEI/UFCG**

**Professora Dra. Francilene Procópio Garcia
Examinadora – UASC/CEEI/UFCG**

Trabalho aprovado em: 25 de novembro de 2019.

CAMPINA GRANDE - PB

TattooStudio: Hub digital para estúdios de tatuagem

Trabalho de Conclusão de Curso

Rafael de Araújo Maciel

Universidade Federal de Campina
Grande

rafael.maciel@ccc.ufcg.edu.br

Tiago Lima Massoni

Universidade Federal de Campina
Grande

massoni@dsc.ufcg.edu.br

RESUMO

A tatuagem está se tornando cada vez mais comum no nosso cotidiano, diariamente inúmeras pessoas buscam tatuadores para realizar serviços relacionados a esse assunto, tendo destaque a própria tatuagem e a criação de artes, juntamente com os tatuadores. No entanto, a organização desses serviços por parte tanto dos tatuadores quanto dos estúdios são um tanto quanto ultrapassadas, muitas vezes sendo feitas por meio de planilhas, ou até mesmo papel e caneta, dado que muitas vezes os sistemas oferecidos, além de não serem específicos para a área, são caros. Portanto, esse trabalho visa o desenvolvimento de uma ferramenta de centralização dos serviços e agendamentos oferecidos pelos tatuadores, facilitando tanto a organização dos estúdios, quanto o acesso do serviço pelos clientes. Para avaliar a qualidade do sistema, foi realizado um estudo de usabilidade na cidade de Campina Grande, em que clientes, tatuadores e donos de estúdio responderam ao questionário PSSUQ (Post-Study System Usability Questionnaire), inferindo sua perspectiva no uso geral do sistema. Como resultado, foi calculada a média das respostas para cada item do questionário, as quais variaram no intervalo de 1 a 3, e teve como média o valor 2, numa escala de 1 a 7, onde 1 representa uma ótima avaliação, e 7 uma péssima avaliação. A partir disso, dada a média das respostas, pode-se inferir que os usuários se mostraram satisfeitos por usar o sistema, concluindo-se que a aplicação apresentou utilizável.

PALAVRAS-CHAVE

Tatuagem, PWA, Desenvolvimento Web.

Links Úteis

<https://github.com/rafaeltj13/TattooStudio>

<https://github.com/rafaeltj13/TattooStudioApi>

<https://forms.gle/5FHSTQfA9wrRCrWbA>

<https://trello.com/b/Qdl0CceC/tattoostudio>

Os autores retêm os direitos, ao abrigo de uma licença Creative Commons Atribuição CC BY, sobre todo o conteúdo deste artigo (incluindo todos os elementos que possam conter, tais como figuras, desenhos, tabelas), bem como sobre todos os materiais produzidos pelos autores que estejam relacionados ao trabalho relatado e que estejam referenciados no artigo (tais como códigos fonte e bases de dados). Essa licença permite que outros distribuam, adaptem e evoluam seu trabalho, mesmo comercialmente, desde que os autores sejam creditados pela criação original.

1. INTRODUÇÃO

A tatuagem é uma das formas de modificação do corpo mais conhecidas e cultuadas no mundo. Trata-se de uma arte permanente feita na pele humana que, tecnicamente, consiste em uma aplicação subcutânea obtida através da introdução de pigmentos por agulhas [1]. Esse tipo de procedimento normalmente é realizado em estúdios específicos para tatuagens, em que estes disponibilizam os seus tatuadores para que os clientes possam escolher da forma que preferirem.

Os principais serviços oferecidos pelos estúdios são a realização de tatuagens e a criação de ideias e artes, além de serviços secundários, como os de aplicação de piercings e brincos. Atualmente, as formas mais utilizadas pelos clientes para contactar esses serviços são pelas redes sociais, ou presencialmente nos estúdios, porém, não são as únicas formas. Ainda utilizam de inúmeras redes sociais diferentes, como WhatsApp, Instagram, Facebook, além de serviços como de SMS, ou até um acordo não formal, realizado apenas por uma conversa, e isso gera uma dificuldade para a organização do tatuador e do estúdio.

A falta de centralização no contato do cliente com os tatuadores gera uma dificuldade maior na organização dos estúdios, além de criar um esforço adicional, pois será trabalho do próprio estúdio e seus tatuadores fazer a distribuição de horários e agendamentos. É normal o contato diário com inúmeros clientes para realizar os diferentes serviços oferecidos no estúdio, dado que normalmente os tatuadores utilizam principalmente planilhas para o controle da clientela; quanto maior a quantidade de pessoas que iniciam o contato, maiores as chances que aconteçam erros ou enganos na organização do agendamento desses serviços, prejudicando tanto o tatuador, quanto o cliente.

A partir disso, o TattooStudio foi produzido com o intuito de sanar essa problemática da falta de centralização. O sistema centraliza e facilita o contato entre o cliente e o estúdio, permitindo que seja possível o agendamento de serviços, avaliação destes, sugerindo horários pertinentes tanto para o

cliente quanto para o tatuador, além de ajudar na busca do estúdio e tatuador ideal para o desejo do cliente.

O sistema utiliza da metodologia PWA (Progressive Web App), que é uma evolução híbrida das páginas web regulares e aplicativos móveis. Esse novo modelo de aplicação combina recursos oferecidos pelos mais modernos navegadores, com as vantagens de um celular. Desta forma, é possível acessar o aplicativo tanto do smartphone, quanto do computador.

Para avaliar a qualidade do uso do sistema, foi realizada uma pesquisa em relação da viabilidade de seu uso. A partir da sondagem de usuários através de um questionário PSSUQ(Post-Study System Usability Questionnaire), no qual 6 clientes, 2 tatuadores e 2 donos de estúdios de tatuagem em Campina Grande utilizaram o sistema e deram sua opinião no uso cotidiano. As médias das respostas do questionário variaram entre 1 e 3, tendo o valor 2 como a média geral. Portanto, conclui-se que o feedback foi positivo, todos os tipos de usuários se demonstraram satisfeitos com o uso do sistema para sanar os problemas anteriormente citados.

2. ARQUITETURA DA SOLUÇÃO

Nesta seção apresentamos as principais decisões arquiteturais utilizadas para a criação do sistema para o problema proposto. O TattooStudio é um sistema online que representa um hub de estúdios de tatuagem, centralizando os serviços oferecidos pelos tatuadores e seus respectivos estúdios, trazendo facilidades tanto para os clientes quanto para os donos de estúdios e seus funcionários.

Os clientes podem realizar todo o processo de solicitação de serviços totalmente online, além de poder acompanhar os serviços que foram agendados. Também é um meio em que facilita a busca tanto por tatuadores quanto pelos estúdios, pelo fato de que é criada uma comunidade a partir da avaliação e comentário sobre os serviços que esses já realizaram, dessa forma, o usuário pode escolher o que seja mais pertinente.

Os tatuadores, com o sistema, têm uma forma centralizada de mostrar seu trabalho, podendo expor todas as suas principais características, além de destaques de serviços anteriormente realizados. Além disso, todos os agendamentos realizados pelos clientes poderão ser acompanhados pelo sistema, tirando todo o esforço adicional de gerência anteriormente necessário pelo próprio tatuador. Os donos de estúdio podem acompanhar todas as solicitações feitas em seus estúdios, além de serem notificados das mesmas, facilitando a gestão de sua empresa.

2.1 Descrição

O aplicativo foi projetado utilizando a arquitetura Cliente-Servidor, a conexão entre esses componentes segue o

Os autores retêm os direitos, ao abrigo de uma licença Creative Commons Atribuição CC BY, sobre todo o conteúdo deste artigo (incluindo todos os elementos que possam conter, tais como figuras, desenhos, tabelas), bem como sobre todos os materiais produzidos pelos autores que estejam relacionados ao trabalho relatado e que estejam referenciados no artigo (tais como códigos fonte e bases de dados). Essa licença permite que outros distribuam, adaptem e evoluam seu trabalho, mesmo comercialmente, desde que os autores sejam creditados pela criação original.

modelo REST (Representational State Transfer), utilizando JSON(Java Object Notation) como o modelo para a transferência de dados. Todos os componentes do sistema foram desenvolvidos utilizando a linguagem de programação JavaScript.

A partir disso, o sistema é dividido em duas partes: o servidor, que tem o trabalho principalmente de aplicar as regras de negócio do sistema e a persistência dos dados, e o cliente, com o propósito principal de mostrar os dados de uma forma agradável e interativa.

Essas duas partes do sistema se comunicam por requisições HTTP, como podemos visualizar na Figura 4, em que o cliente envia essas requisições para o servidor, e este irá tratá-las, aplicando as regras de negócio e mantendo os dados alterados.

2.2 Servidor

O servidor da aplicação foi desenvolvido utilizando como principal tecnologia o NodeJS [2], tendo o MongoDB [7] como ferramenta para persistência dos dados, e o Mongoose [8] para fazer a conexão do servidor com a base de dados. A Figura 1 mostra como os componentes do servidor se comunicam entre si para criar a lógica do sistema, em que o NodeJS mapeia os objetos JavaScript por meio do mongoose, para que, desta forma, seja possível a conexão entre o servidor node com o banco de dados.

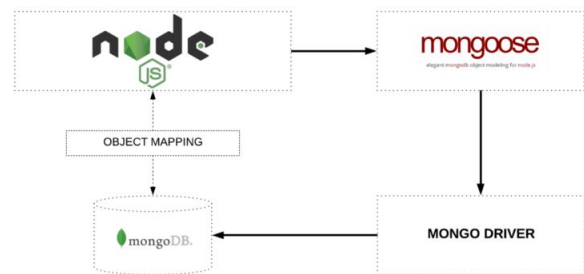


Figura 1: Representação dos frameworks utilizados no servidor do sistema

O intuito desta parte do sistema é realizar as regras de negócio da aplicação, assim como verificar e validar os dados que são recebidos e persistidos, proporcionando uma correteude e segurando na utilização do mesmo.

O servidor trata a requisição HTTP enviada pelo cliente da seguinte forma: primeiramente, a requisição é tratada no Router, a partir do tipo e da rota da mesma, enviando os dados para o Controller, fazendo as primeiras checagem, como as de permissionamento e autenticação. A partir disso, no Controller são realizados os tratamentos de dados iniciais, e os dados pertinentes são enviados para os Services. Com isso, o Service realiza as regras de negócio da aplicação, juntamente do modelo tratado em questão. Cada modelo do servidor tem seus arquivos de Router,

Controller, Service e Model, como mostrado na Figura 3. Por fim, o Modelo faz a comunicação com o banco de dados para realizar o tipo de serviço requisitado pelo cliente, e salva todas as modificações, criando a consistência de dados do sistema.

O NodeJS é um interpretador, com código aberto, de código JavaScript de modo assíncrono e orientado a eventos, focado em migrar a programação a programação do JavaScript para o lado dos servidores, permitindo a criação de aplicações com alta escalabilidade [2].

O MongoDB é um software de banco de dados orientado a documentos livre, de código aberto e multiplataforma. Classificado como um programa de banco de dados NoSQL, utiliza documentos semelhantes a JSON como esquemas. Nos dias atuais, o MongoDB é o banco de dados não relacional mais utilizado no mundo. A Figura 2 é uma representação visual de como os modelos do sistema estão relacionados na base de dados, no qual existem três tipos de usuário, como já foi citado anteriormente: Cliente, Tatuador e Dono de Estúdio, em que o tatuador participará de um estúdio, enquanto o Dono irá gerenciar o mesmo. Cada usuário terá uma agenda própria, que permitirá a gerência de seus atendimentos. Essa agenda vai possuir inúmeros agendamentos, que são registados a partir do uso do sistema. Por fim, o agendamento poderá ser de dois tipos diferentes: realização de uma tatuagem ou a criação de uma arte.

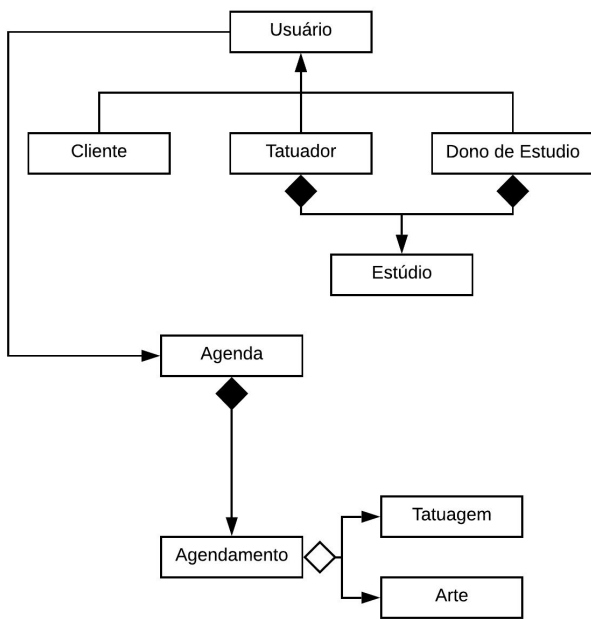


Figura 2: Diagrama de componentes da API do sistema

O Mongoose é um framework de modelagem de modelos MongoDB para NodeJS, tem como responsabilidade fazer a conexão do servidor com a base de dados.

A autenticação do sistema foi feita utilizando JWT(JSON Web Token), que é um padrão que define como transmitir e armazenar objetos JSON de forma compacta e segura entre diversas aplicações. Os dados nele contidos podem ser validados a qualquer momento pois o token é assinado digitalmente, gerando, dessa forma, uma segurança maior para o servidor do sistema.

A estrutura do servidor foi dividida em 4 grandes camadas: Routers, Controllers, Services e Models. Essas foram separadas pelas pastas de acordo com o modelo tratado, como mostrado na figura 3. Dessa forma, é criada uma coesão e uma maior legibilidade na estruturação do servidor.

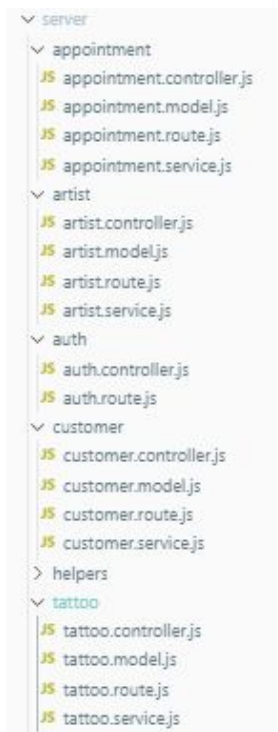


Figura 3: Estrutura de diretórios do servidor

Primeiramente, os Routers têm a função de receber as requisições do cliente, e repassar para as funções pertinentes do Controller. Além de realizar verificações de permissionamento e autenticação do usuário. Logo após, os Controllers têm o propósito de fazer checagens iniciais nos dados que foram recebidos, e repassar os dados pertinentes a serem tratados para os Services, podendo também realizar algumas regras de negócio.

Com isso, os Services realizam a lógica de manipulação dos dados da base de dados, aplicando as regras de negócio que foram definidas previamente e tratando erros que possam vir a ocorrer na comunicação com a base de dados. Por fim, os Models tem a

Os autores retêm os direitos, ao abrigo de uma licença Creative Commons Atribuição CC BY, sobre todo o conteúdo deste artigo (incluindo todos os elementos que possam conter, tais como figuras, desenhos, tabelas), bem como sobre todos os materiais produzidos pelos autores que estejam relacionados ao trabalho relatado e que estejam referenciados no artigo (tais como códigos fonte e bases de dados). Essa licença permite que outros distribuam, adaptem e evoluam seu trabalho, mesmo comercialmente, desde que os autores sejam creditados pela criação original.

função de definir e modelar os dados que serão persistidos no banco de dados, além de definir regras de negócios pertinentes para o modelo que está sendo tratado.

2.3 Cliente

O cliente do sistema foi desenvolvido utilizando como principal tecnologia o ReactJS, tendo o Redux como manipulador do estado da aplicação e o Axios para fazer requisições HTTP para o servidor, como é mostrado na figura 4.

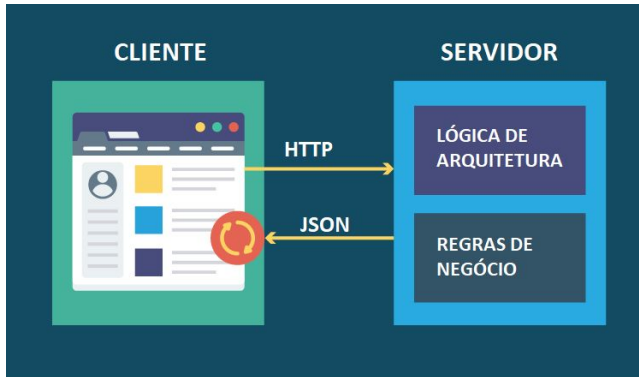


Figura 4: Ilustração da interação entre cliente e servidor

O intuito desta parte do sistema é de oferecer uma interface intuitiva para a interação com os usuários do sistema, provendo todos os serviços disponíveis da aplicação em apenas um lugar.

O ReactJS é uma biblioteca JavaScript de código aberto que permite a criação de interfaces de usuário, ou seja, proporciona a construção de uma camada de visualização em um sistema web. Atualmente é uma das tecnologias mais utilizadas no desenvolvimento de aplicações web, possuindo uma comunidade muito ampla, criando constantes evoluções na ferramenta. O uso dessa ferramenta utiliza de uma estrutura baseada em uma árvore de componentes.

Esta tecnologia foi utilizada com o intuito de desenvolver a interface do usuário de forma modularizada e escalável, priorizando a criação e utilização de componentes altamente coesos. Além disso, a hierarquia de componentes foi bem separada, de forma a facilitar o crescimento da aplicação para o desenvolvimento de novas funcionalidades.

O Redux é uma biblioteca JavaScript de código aberto que permite a manipulação do estado das aplicações, proporcionando uma fácil manipulação de dados e do fluxo da aplicação [3]. O Redux busca complementar a estrutura do React com a camada de lógica de negócios.

O conjunto desses componentes, que são funções puras, do React e do estado manipulado pelo Redux geram a visualização do sistema. A interação do usuário com o sistema dispara actions,

Os autores retêm os direitos, ao abrigo de uma licença Creative Commons Atribuição CC BY, sobre todo o conteúdo deste artigo (incluindo todos os elementos que possam conter, tais como figuras, desenhos, tabelas), bem como sobre todos os materiais produzidos pelos autores que estejam relacionados ao trabalho relatado e que estejam referenciados no artigo (tais como códigos fonte e bases de dados). Essa licença permite que outros distribuam, adaptem e evoluam seu trabalho, mesmo comercialmente, desde que os autores sejam creditados pela criação original.

que são observadas pelos reducers e esses irão alterar o estado da aplicação, voltando os novos dados atualizados para a interface do usuário, como é mostrado na figura 5.

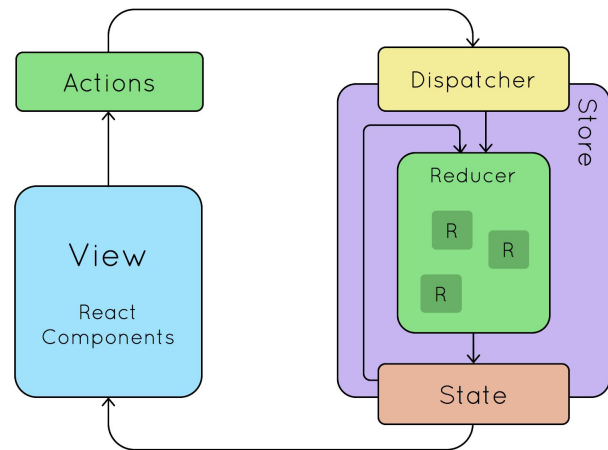


Figura 5: Ilustração da interação entre as principais camadas do cliente

O Axios é um cliente HTTP, que funciona tanto no browser quanto em NodeJS. É uma biblioteca que funciona como uma API que sabe interagir com o servidor REST por meio de requisições HTTP.

Toda parte do cliente do sistema foi feita seguindo a metodologia de desenvolvimento PWA. Um Progressive Web App permite o acesso do sistema e de todas suas funcionalidades tanto no computador quanto no smartphone [4], além de ser possível o uso de notificações para o usuário, que é muito pertinente para a solução do problema proposto. Além das notificações, o PWA trás inúmeras vantagens para aplicações modernas, como o desenvolvimento de um sistema responsivo, mais leve do que aplicativos nativos, suporte à utilização offline, entre outros.

A estrutura do cliente foi dividida em 3 grandes camadas: Actions, Components e Reducers. As Actions e os Reducers vão estar todos em suas respectivas pastas, já os componentes estão separados por suas telas, como mostrado na figura 6. Dessa forma, é criada uma coesão e uma maior legibilidade na estruturação do cliente.

Todos os componentes têm um arquivos de rotas, para definir o componente que será mostrado na tela para a rota escolhida pelo usuário, além de seus componentes que representam as interfaces para o mesmo.

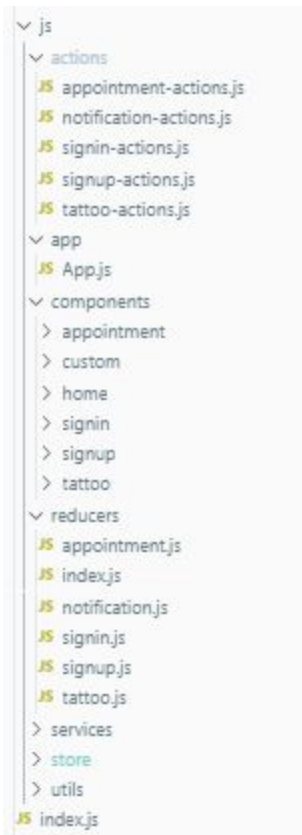


Figura 6: Estrutura de diretórios do cliente

2.4 Lógica da aplicação

O sistema é dividido em 3 tipos de usuários: Clientes, Tatuadores e Donos de estúdio, e cada tipo de usuário irá usufruir do sistema de uma forma diferente.

Na parte do cliente, é oferecida toda a parte do contato que é realizado com os estúdios e os tatuadores, onde todos os serviços são oferecidos em suas respectivas páginas. Além disso, existe um sistema de avaliação, gerando assim uma comunidade no próprio aplicativo, para que os clientes se ajudem e busquem uma melhor experiência no uso.

A Figura 7 mostra a tela inicial do cliente, sempre dando foco ao auxílio na busca do tatuador desejado. Esse tipo de cliente, diferente dos outros, tem 4 escolhas de telas: Início, Busca, Agendamentos e Perfil. O diferencial neste caso seria a busca, onde é mostrada a lista de tatuadores e estúdios cadastrados no sistemas, ordenados pela sua relevância na comunidade.

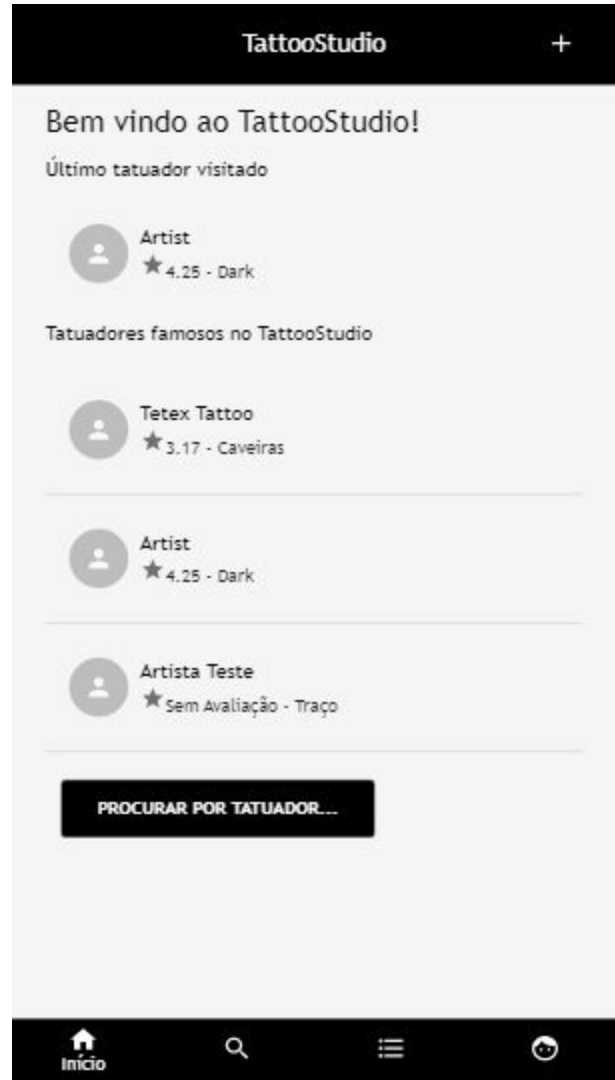


Figura 7: Tela inicial do usuário cliente

Tanto o artista, mostrado na figura 8, como o estúdio tem páginas próprias para oferecer seus serviços. É pela tela do perfil do tatuador que dá início ao agendamento. Todos os agendamentos realizados pelo usuário são mostrados na aba de agendamentos, mostrando os seus detalhes pertinentes, sendo permitido a visualização e edição dos mesmos. É importante lembrar que a edição do agendamento pelo cliente depende de seu status, é um processo realizado entre o cliente e o tatuador.

Os autores retêm os direitos, ao abrigo de uma licença Creative Commons Atribuição CC BY, sobre todo o conteúdo deste artigo (incluindo todos os elementos que possam conter, tais como figuras, desenhos, tabelas), bem como sobre todos os materiais produzidos pelos autores que estejam relacionados ao trabalho relatado e que estejam referenciados no artigo (tais como códigos fonte e bases de dados). Essa licença permite que outros distribuam, adaptem e evoluam seu trabalho, mesmo comercialmente, desde que os autores sejam creditados pela criação original.

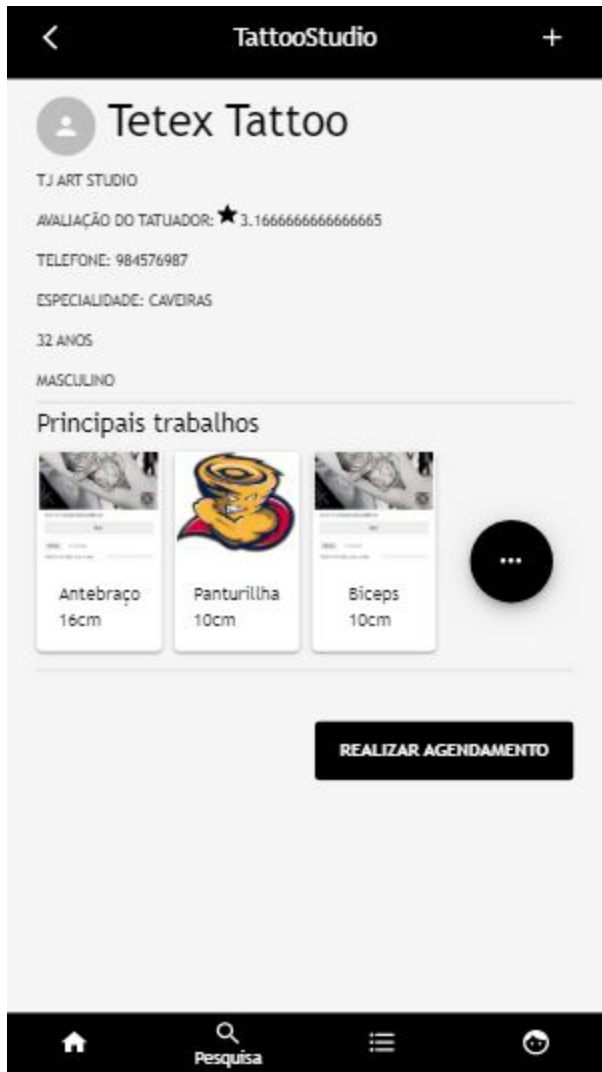


Figura 8: Tela de perfil de um tatuador

O lado dos tatuadores está focado na parte de agendamentos e na disponibilidade de seus serviços. É possível a disponibilização de seus trabalhos, como artes e tatuagens já feitas, em sua página, além de sua própria avaliação, de acordo com o feedback dos clientes que foram atendidos. Além disso, todos os seus trabalhos agendados estarão disponibilizados, com o intuito de facilitar a visualização do mesmo.

A figura 9 mostra a tela principal para este tipo de usuário: Agendamentos. Essa parte da aplicação permite que o artista manipule seus agendamentos de uma forma mais prática, sendo permitida tanto a visualização quanto a edição desses. O próprio sistema irá manipular as datas dos serviços, evitando choques de horário e tirando o peso desse trabalho do próprio tatuador.

Os autores retêm os direitos, ao abrigo de uma licença Creative Commons Atribuição CC BY, sobre todo o conteúdo deste artigo (incluindo todos os elementos que possam conter, tais como figuras, desenhos, tabelas), bem como sobre todos os materiais produzidos pelos autores que estejam relacionados ao trabalho relatado e que estejam referenciados no artigo (tais como códigos fonte e bases de dados). Essa licença permite que outros distribuam, adaptem e evoluam seu trabalho, mesmo comercialmente, desde que os autores sejam creditados pela criação original.

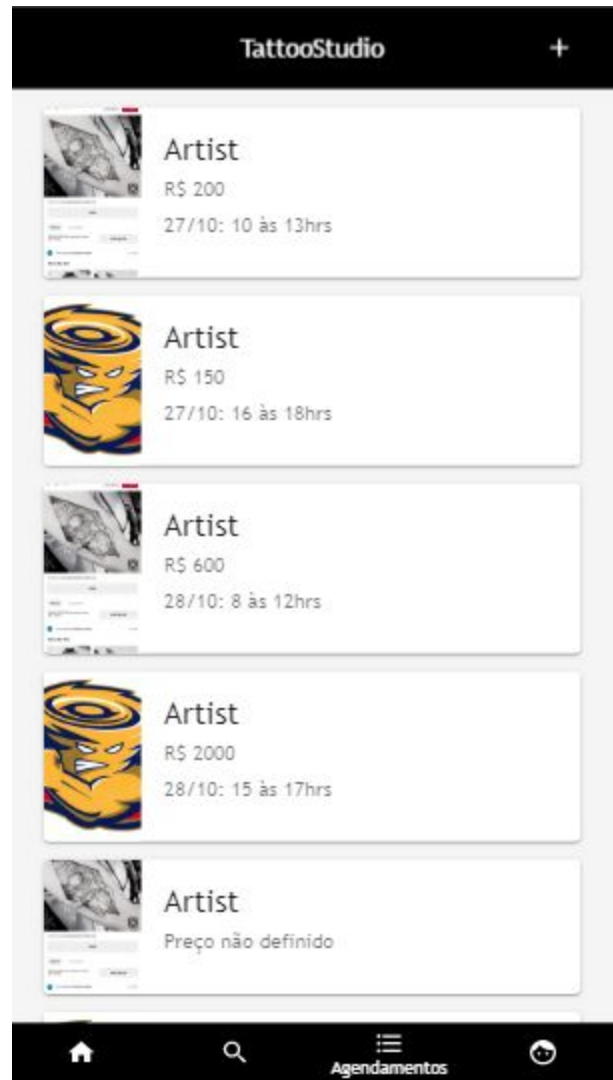


Figura 9: Tela de agendamentos do usuário

Por fim, os donos de estúdio tem a possibilidade de monitorar tudo que está sendo realizado com os funcionários de seu estúdio, centralizando a visualização dos dados, além de receber notificações das ações solicitadas com seus tatuadores. No contexto do aplicativo a tela inicial deste usuário mostra os tatuadores pendentes a entrar no estúdio, assim como os tatuadores já cadastrados. A aba de agendamentos mostra todos os agendamentos realizado com tatuadores cadastrados neste estúdio.

3. AVALIAÇÃO DA SOLUÇÃO

Nesta seção apresentamos a satisfação dos clientes em relação ao uso do sistema desenvolvido para solucionar os problemas citados.

Softwares que agradam o usuário são aqueles que deixam o mesmo satisfeito durante e após a sessão de uso do sistema, desta forma, softwares que não satisfazem o cliente só serão utilizados com a falta de sistema de aplicativos concorrentes. Logo, o desenvolvimento de produtos baseados na satisfação do usuário gera uma maior aceitação do sistema, além de garantirem a qualidade do trabalho [5].

Dado esse ponto, existem inúmeras maneiras para mensurar o grau de satisfação do usuário ao utilizar um sistema computacional, uma dessas maneiras que se tornou muito popular foi a utilização de questionários. Esses seriam testes que propiciam o delineamento de um perfil de satisfação do usuário, do qual pode-se obter valores quantitativos e qualitativos, e assim permitem avaliar a ferramenta que está sendo utilizada.

A partir disso, foram fomentados questionários que buscavam avaliar a usabilidade do sistema e, dessa forma, inferir o grau de satisfação do usuário ao utilizá-lo. Para avaliar este trabalho, foi utilizado o PSSUQ, o qual foi desenvolvido por pesquisadores da IBM e se caracteriza por fornecer uma forma de avaliação global da usabilidade do sistema [5]. A utilização do PSSUQ se mostra bastante pertinente para o sistema dado que ela avalia, do ponto de vista do usuário, a satisfação do mesmo em relação a sugestão proposta, na forma do aplicativo, para a solução do problema em questão, resultando em um número que representa essa satisfação.

O PSSUQ é formado por 17 itens destinados a abordarem as seguintes características de usabilidade do sistema: conclusão rápida do trabalho, facilidade de aprendizado, documentação de alta qualidade, adequação funcional e rápida aquisição do sistema [6]. Além disso, a resposta para cada item está associada a uma escala de 1 a 7, na qual o avaliador colocará quanto concorda com o item, onde 1 significa concordar completamente e 7 discordar completamente.

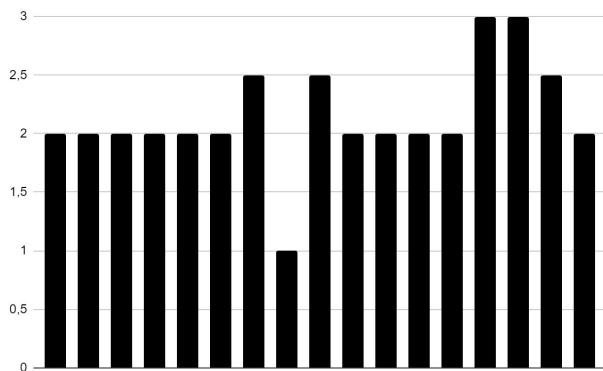


Figura 10: Gráfico com os valores médios das perguntas do questionário, organizados de forma crescente de acordo com a sequência das perguntas, da direita à esquerda

Para gerar o resultado do PSSUQ, é calculada a média dos valores obtidos de cada pergunta. Para este caso, 10 usuários responderam o questionário, e, como pode ser observado na Figura 10, os

Os autores retêm os direitos, ao abrigo de uma licença Creative Commons Atribuição CC BY, sobre todo o conteúdo deste artigo (incluindo todos os elementos que possam conter, tais como figuras, desenhos, tabelas), bem como sobre todos os materiais produzidos pelos autores que estejam relacionados ao trabalho relatado e que estejam referenciados no artigo (tais como códigos fonte e bases de dados). Essa licença permite que outros distribuam, adaptem e evoluam seu trabalho, mesmo comercialmente, desde que os autores sejam creditados pela criação original.

valores das respostas variaram no intervalo de 1 a 3, e teve como média geral o valor 2, ou seja, um feedback bastante positivo da usabilidade do sistema.

4. EXPERIÊNCIAS E LIÇÕES APRENDIDAS

Nesta seção apresentamos as experiências e lições aprendidas durante todo o processo deste trabalho, com ênfase no processo de desenvolvimento, desafios encontrados e trabalhos futuros.

4.1 Processo de desenvolvimento

O processo de desenvolvimento adotado seguiu a metodologia ágil Scrum para a gestão e planejamento do projeto de software. Essa metodologia tem o princípio de entregas rápidas de pequenas funcionalidades requeridas no sistema, e a evolução constante no desenvolvimento do software. Foi pertinente utilizar o scrum neste trabalho, pelo fato de que garantiu a entrega do sistema como um todo, sempre realizando entregas menores, mas que garantiam a funcionamento de toda a aplicação.

O backlog definido previamente com os clientes foi totalmente descrito utilizando a ferramenta Trello; a Figura 11 mostra um exemplo de tarefa do sistema que foi cadastrada. Além disso, foi realizada também a gerência de tarefas a serem realizadas.

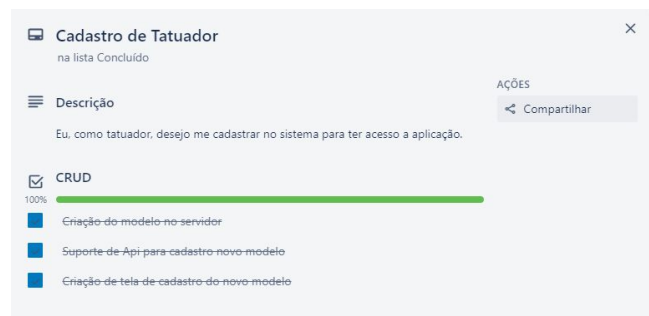


Figura 11: Tarefa cadastrada no Trello

Foram definidas sprints de 15 dias, priorizando sempre as principais funcionalidades do sistema. Dessa forma, além de garantir um funcionamento do aplicativo no geral, é garantida a entrega das funcionalidades principais do sistema, que, nesse caso, seria o agendamento de tatuagens. Isso também trás uma satisfação maior do cliente, pelo fato de mostrar um progresso maior do aplicativo.

A primeira sprint foi focada no desenvolvimento arquitetural tanto do cliente como do servidor, garantindo uma estruturação concreta para o início do desenvolvimento. A segunda sprint consistiu no desenvolvimento do login da aplicação, assim como na criação dos dois primeiros tipos de usuário: Cliente e tatuador.

Além da criação de componentes genéricos reusáveis que seriam utilizados durante todo o desenvolvimento.

A terceira sprint foi focada no desenvolvimento da principal funcionalidade do sistema: os agendamentos. Dessa forma, foi criado o MVP, que seria o produto minimamente viável, uma demonstração inicial de aplicativo, além da garantia de entrega do núcleo do sistema. Na quarta sprint foram desenvolvidos as listagens e a busca de tatuadores e de estúdios, assim como a criação do último tipo de usuário, o dono do estúdio, além da criação do modelo de estúdio.

Por fim, foram realizados as últimas pendências do sistema, como detalhes de layouts, como a responsividade do sistema e o estilo dos componentes, ajustes nas mensagens de notificação do usuário, além de correção de bugs e da escrita de testes do sistema, para garantir a correção e validação na entrega final.

O Git [9] foi escolhido como gerenciador de versões do código e o Github [10] foi a plataforma escolhida para a hospedagem do código-fonte.

4.2 Principais desafios e limitações

A criação de interfaces simples, intuitivas e modernas são sempre uma parte desafiadora no desenvolvimento de novas aplicações, principalmente para um projeto pessoal, que vai ser utilizado na maior parte por um público mais jovem. Foi trabalho do desenvolvedor de criar todas as telas da aplicação, dado a pouca experiência em design, foi necessário pesquisar formas de layouts modernos para tomar como base e criar o design de toda a aplicação. A solução veio por meio do Material-UI [11], que é um framework de design para ReactJS, além da conversa com pessoas da área dessa área.

A ideia de criar componentes reusáveis é sempre uma boa prática de programação, porém, é necessário um esforço adicional na criação dos mesmos, para que os componentes criados tenham coesão, e realmente proporcionem um reuso de código significativo. Nesse projeto, todo o desenvolvimento do cliente web foi voltado para a criação de componentes reusáveis, o que gerou um esforço e complexidade maior no início, porém, gerou um reuso de código muito grande, se tornando um ponto positivo. Um exemplo simples dessa ideia seria do componente de campo de texto, que, no sistema, se mostra de várias formas diferentes, mesmo sendo apenas um único componente.

A coleta de requisitos, juntamente dos clientes, é a parte mais importante na criação de um novo sistema. É a partir dessa coleta que poderá ser medida a correção e a qualidade do aplicativo. Sempre é um grande desafio realizar essa etapa, dado que o usuário normalmente não é da área, com isso, deve ser colocado um esforço grande para que essa seja realizada com sucesso, ou seja, para que o sistema satisfaça os clientes. Para o TattooStudio, foram realizadas quatro reuniões, tanto com clientes quanto os

Os autores retêm os direitos, ao abrigo de uma licença Creative Commons Atribuição CC BY, sobre todo o conteúdo deste artigo (incluindo todos os elementos que possam conter, tais como figuras, desenhos, tabelas), bem como sobre todos os materiais produzidos pelos autores que estejam relacionados ao trabalho relatado e que estejam referenciados no artigo (tais como códigos fonte e bases de dados). Essa licença permite que outros distribuam, adaptem e evoluam seu trabalho, mesmo comercialmente, desde que os autores sejam creditados pela criação original.

tatuadores e os próprios donos de estúdio, com o intuito de extrair o máximo de informação para o desenvolvimento do aplicativo, dado que o foco do aplicativo está no lado dos tatuadores, foi necessário um esforço ainda maior nesta parte para coletar os requisitos.

4.3 Trabalhos futuros

O TattoStudio é um sistema que tem como foco a tatuagem e a criação de artes, porém, estúdios de tatuagem normalmente oferecem outros serviços que sua abordagem no sistema pode ser pertinentes, como, por exemplo, serviços relacionados piercings e brincos. Atualmente, esse serviço apenas é disposto como um trabalho oferecido pelo estúdio, dado esse fato, uma das vertentes que o sistema poderia seguir seria adicionar toda a parte de piercings e brincos, além de um novo tipo de usuário, no aplicativo.

A adição de um designer poderia trazer uma experiência ainda melhor para os usuários, dado que a contribuição e experiência de um membro da equipe especializado traria inúmeros benefícios ao sistema.

Ainda existem melhorias que poderiam ser realizadas, dado que o escopo do projeto foi um pouco grande para o tempo de desenvolvimento, o foco do aplicativo ficou no lado do tatuador. Além como a adição de novas funcionalidades como:

- Adição de novos filtros de pesquisa;
- Sistema de recompensas;
- Mudanças de horários de agendamentos;
- Integração com redes sociais;
- Análise de dados;
- Sistema Web de monitoramento.

Também é interessante pensar em uma plataforma Web mais complexa voltada para o dono do estúdio, proporcionando uma interface mais amigável para uma demanda maior de gerência, dado que esse tipo de usuário coordena todo o estúdio, ou seja, um empresa com inúmeros funcionários, não só tatuadores.

5. REFERÊNCIAS

- [1] Wikipédia, a enciclopédia livre, Tatuagem. Disponível em: <https://pt.wikipedia.org/wiki/Tatuagem>.
- [2] Pluralsight, Gaurav Bhavsar, Getting started with NodeJS. Disponível em: <https://www.pluralsight.com/guides/getting-started-with-nodejs>.

- [3] Medium, Hélio Kroger, Entendendo React e Redux de uma vez por todas. Disponível em: <https://pt.wikipedia.org/wiki/Node.js>.
- [4] Vizir, Antônio Anderson, O que é PWA(Progressive Web Application) e porque isso pode aumentar seus resultados mobile. Disponível em: <https://vizir.com.br/2017/08/o-que-e-pwa-progressive-web-app-porque-isso-pode-aumentar-seus-resultados-mobile/>
- [5] Cone Trees, Abhay Rautela, Post-Study System Usability Questionnaire(PSSUQ). Disponível em: <https://www.conetrees.com/ux-glossary/post-study-system-usability-questionnaire-pssuq/>
- [6] Trymyui, Tim Rotolo, SUS and PSSUQ: Quantifying user experiences. Disponível em: <https://www.trymyui.com/blog/2017/02/24/sus-pssuq-website-usability-surveys/>.
- [7] MongoDB Inc., The most popular database for modern applications. Disponível em: <https://www.mongodb.com/>
- [8] Elegant mongodb object modeling for node.js. Disponível em: <https://mongoosejs.com/>.
- [9] Wikipédia, a enciclopédia livre, Git. Disponível em: <https://pt.wikipedia.org/wiki/Git>.
- [10] Wikipédia, a enciclopédia livre, Github. Disponível em: <https://pt.wikipedia.org/wiki/Git>.
- [11] Material-UI: A popular React UI framework. Disponível em: <https://material-ui.com/>.

Os autores retêm os direitos, ao abrigo de uma licença Creative Commons Atribuição CC BY, sobre todo o conteúdo deste artigo (incluindo todos os elementos que possam conter, tais como figuras, desenhos, tabelas), bem como sobre todos os materiais produzidos pelos autores que estejam relacionados ao trabalho relatado e que estejam referenciados no artigo (tais como códigos fonte e bases de dados). Essa licença permite que outros distribuam, adaptem e evoluam seu trabalho, mesmo comercialmente, desde que os autores sejam creditados pela criação original.