

# NavAcad: um aplicativo de suporte para navegação escolar

Victoria B. Martins, Ederson Bastiani

Instituto Federal Farroupilha (IFFar) *campus* Panambi - RS

victoria.martins@aluno.iffar.edu.br,  
ederson.bastiani@iffarroupilha.edu.br

**Abstract.** *The students, aiming to improve their academic development, need to interact with the teaching staff. But, the task of localizing them has been difficult because there is no direct, always available access and little support to informations such as professors' schedule. This paper had the goal to develop a mobile application to the Instituto Federal Farroupilha campus Panambi. In the application, the user may consult schedules and other informations such as email and their rooms, and even visualize the campus using geolocation. This paper follows the IMRD Model.*

**Resumo.** *Os alunos, a fim de sanar dúvidas e melhorar seu desempenho acadêmico, necessitam de interação com seus professores e gestores. No entanto, por não haver acesso direto e sempre disponível aos horários dos servidores, torna-se difícil localizá-los na instituição. O projeto proposto teve como objetivo desenvolver uma solução de forma prática e funcional, construindo um aplicativo mobile voltado para o Instituto Federal Farroupilha campus Panambi. No aplicativo desenvolvido, o usuário pode visualizar o campus através de geolocalização além de consultar horários e outras informações como telefones e e-mails e de professores e suas salas. A estrutura deste projeto segue o modelo IMRD.*

## 1. Introdução

Os *smartphones*, em sua tradução livre “telefones inteligentes”, estão cada vez mais fazendo parte da vida da sociedade. Desde o ano de 2018, o Brasil já tem mais de um *smartphone* por habitante, sendo atualmente mais de 230 milhões em uso no Brasil [Estadão, 2019]. Devido à popularização destes dispositivos, há uma tendência de automatizar e comercializar cada vez mais serviços e funcionalidades.

Devido a necessidades como praticidade e mobilidade, grandes produtos e empresas têm se consolidado no mercado, podendo-se destacar aplicativos como *Uber*, *iFood*, *Whatsapp*. No contexto acadêmico não é diferente. Estudos e propostas em torno de soluções *mobile* têm tornado ambientes acadêmicos mais acessíveis, colaborativos e coletivos [Silva et al 2017] [Wahlbrinck e Boniati 2017]. Nesta perspectiva, torna-se interessante incluir o acesso a informações sobre itinerários de professores e funcionários, de forma que estes possam ser encontrados com maior facilidade, otimizando, assim, o processo interativo educacional entre alunos e professores, por exemplo.

Nesse sentido, o presente trabalho aborda o desenvolvimento de uma aplicação que busca disponibilizar estas informações. O estudo foi baseado em uma instituição federal de ensino, a qual oferta, atualmente, treze cursos superiores e técnicos que variam entre os turnos matutino, vespertino e noturno. Como prática, tal instituição informa a

distribuição de carga horária semanal dos servidores através de folhas de papel afixadas nas portas de cada sala, fazendo com que o público interessado tenha que se deslocar até estes locais. Soma-se a isso, o fato de que a instituição possui edifícios distantes uns dos outros e que a cada semestre, há troca de horários, o que muitas vezes dificulta o acesso à setores específicos. Portanto, o público-alvo deste trabalho engloba discentes e servidores. A interface buscou utilizar os princípios de heurística de Nielsen, prezando pela visibilidade limpa e eficiência de uso.

Desta forma, como parte da proposta, o aplicativo apresenta a localização atual do usuário caso esteja no *campus* além de indicar em um mapa prédios e suas descrições, bem como os itinerários, e-mail e telefone dos servidores.

Para melhor localização, este documento está estruturado da seguinte forma: a seção 2 apresenta os materiais e métodos utilizados, seguido da seção 3 com os resultados alcançados e por fim as considerações finais são abordadas na seção 4, seguidas das referências utilizadas no embasamento do trabalho.

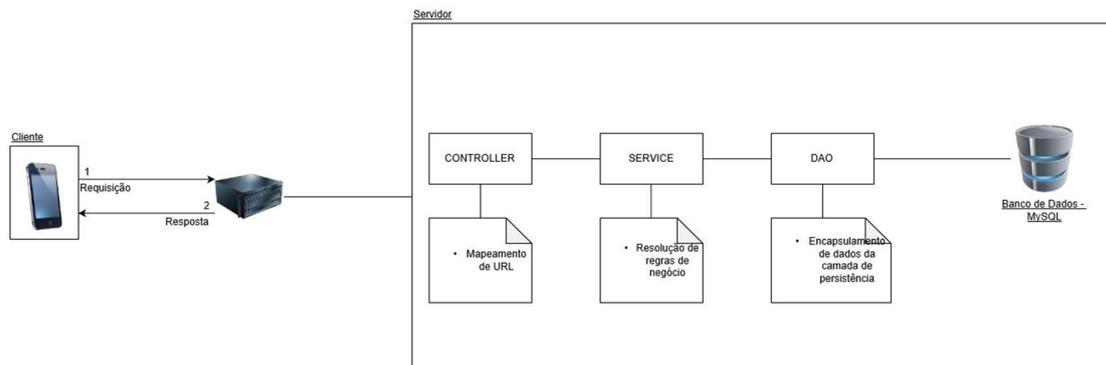
## 2. Material e métodos

Este projeto foi desenvolvido durante as disciplinas de Programação Web III e Programação para Dispositivos móveis e sem fio do curso de Tecnologia em Sistemas para Internet durante o primeiro semestre de 2019, e para desenvolvimento foram utilizadas as ferramentas *Visual Studio Code* e *SpringToolsSuite 4* e o banco de dados MySQL do pacote XAMPP que acompanha também outras ferramentas como o servidor Apache, os interpretadores de *script* PHP: Hypertext Preprocessor e Perl.

O sistema, nomeado de *NavAcad*, baseia-se em um conjunto de duas aplicações que, como um todo, segue o modelo cliente-servidor, utilizando o protocolo RESTful. Em outras palavras, o servidor, aquele que provê o serviço e o cliente, aquele que acessa o serviço, trocam suas informações utilizando todos os princípios Transferência de Estado Representacional (REST), proporcionando comunicação entre ambas as partes [Camargo 2008].

A primeira aplicação corresponde ao *front-end* do sistema, ou seja, a parte com a qual o usuário interage diretamente [Viana 2017]. Para isso, foi desenvolvido um aplicativo móvel utilizando a tecnologia *React Native*, um *framework* baseado em JavaScript para desenvolver aplicativos para os sistemas Android e iOS de forma nativa [Kupka 2017]. Ainda, foram utilizadas as bibliotecas *React Native Maps*, para disponibilização de mapa geográfico, *React Native Base*, para estilização, e *React Native Navigation*, para disponibilizar componentes de navegação pelo aplicativo.

Já a segunda aplicação, *back-end*, é responsável pela implementação das regras de negócios [Viana 2017]. Para esta aplicação foi utilizado o paradigma de Programação Orientada a Objetos através da utilização do *framework* *SpringBoot*, o qual tem como objetivo a construção rápida de aplicativos que sejam fáceis de configurar [Perry 2017]. No *back-end* foram definidas as operações com o banco de dados e rotas de endereços para retornar os dados no modelo JSON. A Figura 1 apresenta a arquitetura do sistema.



**Figura 1. Diagrama de arquitetura do sistema NavAcad**

O fluxo de funcionamento do sistema ocorre da seguinte forma: ao enviar uma requisição (1), o servidor a processa através das camadas apresentadas acima que são responsáveis por manter a requisição segura, confiável e completa. De modo geral, a primeira camada (*Controller*) mapeia a requisição, a fim de direcionar da forma mais precisa, para a operação desejada. Depois, estes dados são otimizados pela segunda camada (*Service*) com as chamadas “regras de negócio” que tratam o tratamento de erros, por exemplo. Posteriormente, a última camada é responsável por acessar e realizar a operação desejada no banco de dados. Após todo este processo, é gerado o corpo da resposta e convertida em formato JSON para o cliente (2).

### 3. Resultados

Esta seção apresenta os resultados alcançados com o desenvolvimento deste projeto no que tange à parte do aplicativo móvel.

O componente (tela) inicial do aplicativo pode ser visto na Figura 2. Neste componente o usuário possui três opções de botões: Mapa, Salas e Servidores.



**Figura 2. Tela Inicial do aplicativo.**

Ao clicar na opção *Mapa*, o usuário é redirecionado para a tela apresentada na Figura 3, onde é disponibilizado um mapa com os marcadores cadastrados no banco de dados. Estes marcadores que são de cor vermelha, representam os prédios do campus. Caso o usuário esteja na área do *campus*, um marcador azul surgirá no mapa acompanhando sua movimentação. Ao interagir com os marcadores de prédio, são exibidos mais detalhes sobre aquela área.



Figura 3. Página de Mapa do aplicativo

Por sua vez, o botão *Salas*, na tela inicial, redireciona para uma tela responsável por disponibilizar as salas cadastradas no banco de dados (Figura 4). É possível interagir com e-mails e números.



Figura 4. Tela de visualização de salas do aplicativo.

Nesta tela, ao tocar no botão *Visitar* são exibidos mais detalhes sobre aquela sala, como mostra a Figura 5.



**Figura 5. Tela seleção específica de sala.**

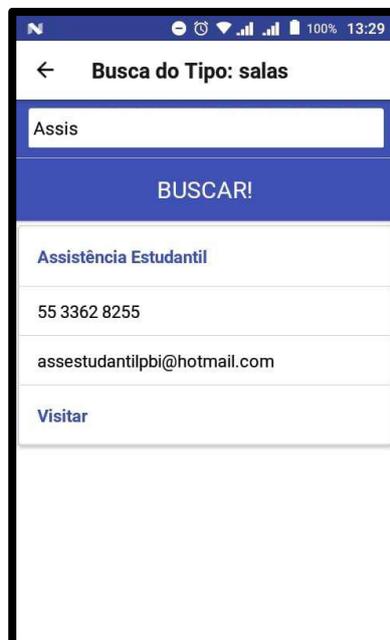
Nesta tela são disponibilizados os horários e *e-mails* dos servidores cadastrados no banco de dados para uma determinada sala.

Um comportamento semelhante é observado quando o usuário escolhe a opção *Servidores* na tela inicial. Neste caso, todos os servidores são listados para o usuário, como pode ser visualizado na Figura 6.



**Figura 6. Tela de visualização de servidores do aplicativo.**

Por fim, a Figura 7 representa a funcionalidade de busca disponível na tela *Salas*. A partir disto, o usuário pode procurar uma determinada sala, escrevendo o seu nome e pressionando o botão *Buscar!* e, como resultado, se houver, serão exibidas todas as salas com que possuem texto digitado pelo usuário. Caso contrário, é exibida uma mensagem avisando inexistência da sala. Esta funcionalidade de busca também está disponível para servidores através do botão *Buscar Servidores* (Figura 6).



**Figura 7. Tela de busca de salas do aplicativo com parâmetro.**

Nesta tela, ao selecionar a opção *Visitar*, são exibidos mais detalhes sobre a sala, através do redirecionamento do usuário para a tela de detalhes (Figura 5).

Os testes foram feitos em ambientes simulados, com o simulador do Android Studio para Nexus 7 além de 2 dispositivos móveis físicos diferentes: Asus Zenfone 3 e Multilaser MS40S.

Até o momento, as informações disponibilizadas no aplicativo são apenas buscadas no Banco de Dados em um servidor local. Espera-se como etapa futura, disponibilizar a aplicação para um conjunto de colaboradores para que seja testada em ambiente real, de forma a avaliar a usabilidade, utilidade, e realizar possíveis melhorias no aplicativo.

#### **4. Considerações finais**

Este projeto visou tornar o acesso aos servidores, através da disponibilização de seus horários e *e-mails*, além de mapeamento da instituição, através de uma solução *mobile*. Como trabalho final das matérias já citadas, o desenvolvimento desta aplicação além de auxiliar na absorção dos conteúdos das aulas, foi possível compreender mais aspectos no quesito de desenvolvimento de *software*.

Através do desenvolvimento deste aplicativo, foi possível compreender mais a fundo usabilidade em aplicativos móveis. Outro ponto positivo foi o incentivo na criação de soluções que possam ajudar a instituição de alguma forma. Espera-se futuramente,

disponibilizar a aplicação em uma loja de aplicativos como Google Play para que seja testada e avaliada baseada na metodologia *System Usability Scale* (SUS).

## Referências

- Baeldung. (2018) A Controller, Service and DAO Example with Spring Boot and JSF. Disponível em <<https://www.baeldung.com/jsf-spring-boot-controller-service-dao>> Acesso em 02 de Agosto de 2019.
- Baeldung. (2019) The Spring @Controller and @RestController Annotations. Disponível em <<https://www.baeldung.com/spring-controller-vs-restcontroller>> Acesso em 02 de Agosto de 2019.
- Brambrilla, T. Boniati, B. B. Rodrigues, G. F. Basso, D. Ramires, G. S. Balestrin, S. F. (2018) Adaptações no Software ACML visando o Desenvolvimento de uma Aplicação Mobile Híbrida. Disponível em <<http://eati.info/eati/anais-2018/Longos/L08.pdf>>. Acesso em 25 Julho de 2019.
- Brasil. (2008) LEI Nº 11.892, DE 29 de Dezembro de 2008.
- Camila, C. (2008) O que é Cliente-Servidor? Disponível em <<https://www.tecmundo.com.br/internet/982-o-que-e-cliente-servidor-.htm>>. Acesso em 01 de Agosto de 2019.
- Crupi, J. Alur, D. Malks., D. (2006) Core J2EE Patterns.
- Ciriaco, D. (2018) Pesquisa: Play Store aumentou 30% e App Store diminuiu 5% em 2017. Disponível em <<https://www.tecmundo.com.br/software/128979-pesquisa-play-store-aumentou-30-app-store-diminuiu-5-2017.htm>>. Acesso em 24 de Julho de 2019.
- Dogtiev, A. (2018) App Stores List (2018). Disponível em <<https://www.businessofapps.com/guide/app-stores-list/>>. Acesso em 24 de Julho de 2019.
- Estadão. (2019) Brasil tem 230 milhões de smartphones em uso. Disponível em <<https://epocanegocios.globo.com/Tecnologia/noticia/2019/04/brasil-tem-230-milhoes-de-smartphones-em-uso.html>>. Acesso em 24 de Julho de 2019.
- Fadatare, R. (2018) What Is the Difference Between Hibernate and Spring Data JPA? Disponível em <<https://dzone.com/articles/what-is-the-difference-between-hibernate-and-sprin-1>> Acesso em 28 de Julho de 2019.
- G1. (2019) Estudantes da Fatec criam aplicativo que calcula dosagem de remédios para evitar erros na área da saúde. Disponível em <<https://g1.globo.com/sp/itapetininga-regiao/noticia/2019/05/18/estudantes-da-fatec-criam-aplicativo-que-calcula-dosagem-de-remedios-para-evitar-erros-na-area-da-saude.ghtml>> Acesso em 1 de Agosto de 2019.
- Fournier, D. (2016) Heurísticas de Nielsen — Avaliando a usabilidade de interfaces. Disponível em <<https://medium.com/vivareal-ux-chapter/heurísticas-de-nielsen-avaliando-a-usabilidade-de-interfaces-e96f9801cd5>> Acesso em 13 de Setembro de 2019.

- IFFar. (2016) Guia de Cursos - *campus* Panambi. Disponível em <<https://www.iffarroupilha.edu.br/campus-panambi>> Acesso em 2 de Agosto de 2019.
- IFFar. (2016) Ensino - *campus* Panambi. Disponível em <<https://www.iffarroupilha.edu.br/ensino-panambi/sobre-ensino-panambi>> Acesso em 2 de Agosto de 2019.
- Kupka, F. (2017) O que é React Native? Disponível em <<http://www.organicadigital.com/seeds/o-que-e-react-native/>>. Acesso em 28 de Julho de 2019.
- Perry J. S. (2017) Informações básicas sobre o Spring Boot. Disponível em <<https://www.ibm.com/developerworks/br/library/j-spring-boot-basics-perry/index.html>> Acesso em 28 de Julho de 2019.
- Ponte, R. (2015) Entendendo os stereotypes do Spring. Disponível em <<http://blog.triadworks.com.br/entendendo-os-stereotypes-do-spring>> Acesso em 02 de Agosto de 2019.
- Silva, C. G. Moiano, I. G. Silva, L. R. Coelho, V. Silva, M. A. R. (2017) Interação de Pessoas com Deficiência Visual com Dispositivos Móveis. Disponível em <<http://eati.info/eati/2017/assets/anais/anais.pdf>>. Acesso em 25 de Julho de 2019.
- Viana, D. (2017) O que é front-end e back-end? Disponível em <<https://www.treinaweb.com.br/blog/o-que-e-front-end-e-back-end/>> Acesso em 28 de Julho de 2019.
- Wahlbrinck, K. A, BONIATI. Bruno B. (2017) Aplicações Mobile Híbridas: Um Estudo de Caso do Framework Ionic para Construção de um Diário de Classe. Disponível em <<http://eati.info/eati/2017/assets/anais/anais.pdf>>. Acesso em 25 de Julho de 2019.