

**Indoor INF**

---

**Indoor INF**  
**Arquitetura de Software**  
versão 1.3

*Arthur de Camargo Alves*

*Arthur Faria Peixoto*

*Rafael Estanislau Moraes dos Santos*

## Histórico de Revisão

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
13/11/23	1.0	Elaboração do documento	Arthur Alves
20/11/23	1.1	Finalidade, Escopo, Visão Geral, Representação Arquitetural	Arthur Alves
11/12/23	1.2	Casos de Uso, Metas e Restrições da Arquitetura	Arthur Alves, Arthur Faria, Rafael Estanislau
11/01/2024	1,3	Casos de Uso	Arthur Alves, Arthur Faria, Rafael Estanislau

/\*sumário virá aqui posteriormente\*/

<http://www.basef.com.br/old/uml/204-arquitetura-visao-modelo-41>

## **1. Introdução**

O presente documento tem por objetivo descrever a arquitetura do projeto Indoor INF. Este sistema tem como finalidade assistir estudantes a se localizarem dentro dos edifícios do Instituto de Informática da UFG, do Centro de Aulas Baru da UFG, e do Centro de Aulas Caraíbas da UFG, provendo-lhes informações em tempo real sobre a direção de determinadas salas, através do uso de tags NFC para localizar onde o usuário está e então guiar o usuário até seu destino..

Através do mapeamento dos prédios do Instituto de Informática e dos Centros de Aula, o algoritmo do Indoor INF calcula com base em dados obtidos dos sistemas da universidade, bem como do mapeamento geográfico e o contexto situacional do usuário, e então o direciona para onde ele deseja chegar.

### **1.1. Finalidade**

O projeto Indoor INF visa proporcionar assistência aos estudantes para se orientarem dentro dos edifícios do Instituto de Informática da UFG, do Centro de Aulas Baru da UFG e do Centro de Aulas Caraíbas da UFG. A principal funcionalidade do sistema é fornecer informações em tempo real sobre a direção de salas específicas, utilizando um sistema de tags NFC para guiar os usuários até o destino desejado.

### **1.2. Escopo**

O escopo do projeto abrange o mapeamento dos prédios do Instituto de Informática e dos Centros de Aula mencionados. O algoritmo do Indoor INF utiliza dados provenientes dos sistemas da universidade, informações de mapeamento e o contexto situacional do usuário para calcular a rota mais eficiente até o destino desejado. O escopo inclui o desenvolvimento e implementação do sistema, bem como a integração com os dados universitários para garantir informações precisas e atualizadas.

### **1.3. Definições, Acrônimos e Abreviações**

INF: Instituto de Informática  
UFG: Universidade Federal de Goiás  
NFC: Near Field Communication

## 1.4. Visão Geral

O sistema Indoor INF utiliza tecnologia de NFC para fornecer assistência de localização do prédio atual aos estudantes dentro dos edifícios específicos da UFG. O mapeamento detalhado dos prédios, combinado com algoritmos avançados, permite calcular rotas eficientes com base nos dados da universidade e no contexto do usuário. A visão geral abrange a integração harmoniosa do sistema com as necessidades de orientação dos estudantes, proporcionando uma solução eficaz para facilitar a navegação dentro dos edifícios acadêmicos.

## 2. Representação Arquitetural

- **Visão de caso de uso:** identifica e descreve as interações entre atores (usuários ou sistemas externos) e o sistema em termos de casos de uso. Cada caso de uso representa uma funcionalidade ou recurso oferecido pelo sistema, proporcionando uma visão geral das principais interações entre usuários e sistema.
- **Visão lógica:** descreve a estrutura e organização interna do sistema, destacando as entidades, classes, relacionamentos e suas interações. Ela oferece uma compreensão abstrata dos elementos fundamentais do sistema, sem se aprofundar nos detalhes de implementação.
- **Visão de processos:** concentra-se nos processos e fluxos de trabalho do sistema. Ela descreve como as atividades são realizadas, os eventos que as desencadeiam e as relações entre essas atividades. Essa visão é valiosa para entender a dinâmica e o sequenciamento das operações dentro do sistema.
- **Visão de implantação:** trata da distribuição física e organizacional dos componentes do sistema em diferentes ambientes. Ela abrange a infraestrutura de hardware, redes, servidores e a disposição dos artefatos de software, proporcionando uma compreensão clara de como o sistema é implantado e operado no ambiente real.
- **Visão de implementação:** detalha os aspectos técnicos da construção do sistema. Ela inclui informações sobre linguagens de programação, frameworks, bibliotecas e outros componentes tecnológicos utilizados. Essa visão é valiosa para desenvolvedores e equipes técnicas envolvidas na construção e manutenção do sistema.

### 3. Metas e Restrições da Arquitetura

#### 3.1. Arquitetura de Software

##### 3.1.1. Modelo Arquitetural

**Modelo:** Representa os dados e a lógica de negócios da aplicação. No caso do Indoor INF, o modelo incluiria estruturas de dados para informações do usuário, dados de mapeamento, e qualquer lógica necessária para o cálculo de rotas.

- **Implementação:** Classes ou estruturas que definem a estrutura de dados, e lógica de negócios, como cálculos de rotas eficientes.

**Visão:** Representa a interface do usuário (UI). Na aplicação Indoor INF, a view incluiria a interface do aplicativo móvel, como telas de mapa, interfaces de navegação e elementos visuais.

- **Implementação:** Componentes visuais desenvolvidos usando React Native. Telas que exibem mapas interativos, direções e informações de localização.

**ViewModel:** Age como um intermediário entre o modelo e a visão. Ele contém a lógica de apresentação e manipulação de dados necessária para a interação com a UI.

- **Implementação:** Componentes lógicos intermediários que traduzem os dados do modelo para a apresentação na view. Contêm lógica para atualização da UI em resposta a mudanças no modelo. Lidam com eventos e interações do usuário, comunicando-se com o modelo quando necessário.

##### 3.1.2. Fluxo de Dados

- **Usuário interage com a View:**
  - Toques na tela, entrada de dados, etc.
- **ViewModel atualiza o Modelo:**
  - A ViewModel processa a entrada do usuário e atualiza o modelo conforme necessário.
- **Modelo é Atualizado:**
  - Mudanças no modelo desencadeiam atualizações na ViewModel.

- ViewModel Atualiza a View:
  - A ViewModel atualiza a View com os dados mais recentes.
- View exibe informações atualizadas:
  - A interface do usuário reflete as mudanças no modelo por meio da ViewModel.

## **3.2. Arquitetura de Sistema**

### **3.2.1. Camada de Apresentação (View):**

Será responsável por apresentar a interface do usuário e lidar com interações do usuário.

- Componentes:
  - Mapa Interativo: exibe o mapa dos edifícios internamente e da UFG externamente. Permite a navegação interativa.
  - Telas de Informações: irá mostrar informações sobre salas, corredores, etc.
  - Interface de Navegação: fornecerá opções de navegação para o usuário.
  - Elementos Visuais: marcadores, setas ou outras indicações visuais para guiar o usuário.

### **3.2.2. Camada de Lógica de Apresentação (ViewModel):**

Vai atuar como um intermediário entre a camada de apresentação e o modelo. Além disso, irá lidar com a lógica de apresentação e interação com a UI.

- Componentes:
  - Tradutores de Dados: converte os dados do modelo para formatos compreensíveis pela view.
  - Controladores de Eventos: lidam com eventos do usuário e da interface.
  - Atualizadores de UI: responsáveis por atualizar a interface com base nas alterações no modelo.
  - Validadores de Entrada: garantem que os dados de entrada do usuário sejam válidos antes de serem enviados ao modelo.

### 3.2.3. Camada de Modelo:

Representa dados e lógica de negócios.

- Componentes:
  - Estruturas de Dados: armazenam informações do usuário, dados de mapeamento e contexto situacional.
  - Algoritmo de Cálculo de Rotas: calcula rotas eficientes com base em dados da universidade e mapeamento geográfico.
  - Integrador de Dados da Universidade: Obtém quaisquer dados relevantes dos sistemas da universidade.
  - Gestor de Contexto Situacional: avalia o contexto do usuário para adaptar as instruções de navegação dependendo do clima, do trânsito, do veículo de locomoção e de coisas semelhantes.

### 3.2.4. Integrações Externas:

Gerencia a comunicação com sistemas externos.

- Componentes:
  - Near Field Communication (NFC): Cada tag representa a uma sala do INF, o usuário, ao abrir o aplicativo lê a tag presente em sua sala atual. O aplicativo atualiza no mapa mostrando sua localização, e então ele pode escolher seu ponto de destino e irá receber a rota desejada.

### 3.2.5. Segurança

Garante a segurança dos dados do usuário e do sistema.

- Componentes:
  - Módulo de Autenticação: iremos gerenciar a autenticação de usuários através de logins utilizando o email institucional da UFG e gerando um token jwt para gravar a sessão do usuário até que a mesma se encerre.
  - Criptografia de Dados: dados sensíveis como a localização do usuário serão criptografadas dificultando invasões ou vazamento de informações.



### 3.2.6. Tecnologias

O aplicativo será implementado usando primariamente JavaScript e banco de dados em nuvem (AWS)

- **Tecnologia Front-End**
  - **ReactJs**
- **Tecnologia Back-End**
  - **NodeJs**

## 4. Visão de Casos de Uso

Esta seção lista as especificações centrais e significantes para a arquitetura do sistema.

Lista de casos de uso do sistema:

- **Caso de uso FG01**
  - Fazer login no Indoor INF como usuário.
- **Caso de uso FE01**
  - Visualizar o mapa interno do INF.
- **Caso de uso FE02**
  - Visualizar o mapa interno do CAB.
- **Caso de uso FE03**
  - Encontrar salas e banheiros do INF e do CAB.
- **Caso de uso FE04**
  - Encontrar coordenação do INF.

Cenários:

- **Cenário FG01-C01**
  - Logar-se com sucesso
- **Cenário FG01-C02**
  - Logar-se sem sucesso
- **Cenário FG01-C03**
  - Logar-se sem sucesso
- **Cenário FG01-C04**
  - Logar-se sem sucesso
- **Cenário FE01-C01**
  - Acessar o mapa do INF
- **Cenário FE01-C02**

- Encontrar a sala da coordenação do curso de Engenharia de Software.
- **Cenário FE03-C01**
  - Acessar o mapa do CAB e encontrar a sala 210.
- **Cenário FE03-C02**
  - Acessar o mapa do CAB e encontrar o banheiro disponível em cada andar.
- **Cenário FE04-C01**
  - Acessar o mapa do INF e encontrar a sala da coordenação.

#### 4.1. Casos de Uso Relevantes para a Arquitetura

//Diagramas de caso de uso

### 5. Visão Lógica

/\*

Descrição da visão lógica da arquitetura. Descreve as classes mais importantes, sua organização em pacotes de serviços e subsistemas, e a organização desses subsistemas em camadas. Também descreve as realizações dos casos de uso mais importantes, por exemplo, aspectos dinâmicos da arquitetura.

\*/

/\*

Diagramas de classes e sequência devem ser incluídos para ilustrar os relacionamentos entre as classes significativas na arquitetura, subsistemas, pacotes e camadas.

\*/

### 6. Visão de Implantação

#### 6.1. Caso de uso FG01

6.1.1. Diagrama de Classes

6.1.2. Diagrama de Sequência

### 7. Visão da Implementação

### 8. Tamanho e Desempenho

### 9. Qualidade