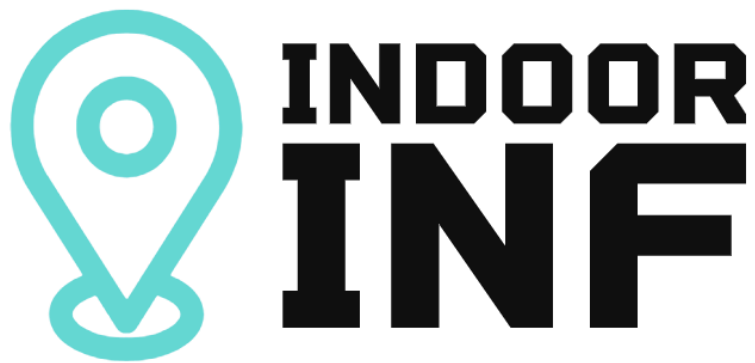


Indoor INF
Arquitetura de Software
versão 1.6



Arthur de Camargo Alves
Arthur Faria Peixoto
Rafael Estanislau Moraes dos Santos

Histórico de Revisão

Data	Versão	Descrição	Autor
13/11/23	1.0	Elaboração do documento	Arthur Alves, Arthur Faria, Rafael Estanislau
20/11/23	1.1	Finalidade, Escopo, Visão Geral, Representação Arquitetural	Arthur Alves, Arthur Faria, Rafael Estanislau
11/12/23	1.2	Casos de Uso, Metas e Restrições da Arquitetura	Arthur Alves, Arthur Faria, Rafael Estanislau
11/01/2024	1.3	Diagramas de Caso de Uso	Arthur Alves, Arthur Faria, Rafael Estanislau
15/01/2024	1.4	Atributos de Qualidade Prioritários, Funcionalidades e Restrições Arquiteturais	Arthur Alves, Arthur Faria, Rafael Estanislau
26/01/2024	1.5	Visão Lógica	Arthur Alves, Arthur Faria, Rafael Estanislau
27/01/2024	1.6	Visão de Implantação, Tamanho e Desempenho, e Qualidade	Arthur Alves, Arthur Faria, Rafael Estanislau

1. Introdução.....	5
1.1. Finalidade.....	5
1.2. Escopo.....	5
1.3. Definições, Acrônimos e Abreviações.....	5
1.4. Visão Geral.....	6
2. Representação Arquitetural.....	6
2.1. Atributos de Qualidade Prioritários.....	6
2.1.1. Segurança.....	6
2.1.2. Manutenibilidade.....	6
2.1.3. Usabilidade.....	7
2.1.4. Portabilidade.....	7
2.2. Funcionalidades e Restrições Arquiteturais.....	7
3. Metas e Restrições da Arquitetura.....	11
3.1. Arquitetura de Software.....	11
3.1.1. Modelo Arquitetural MVVM.....	11
3.1.2. Fluxo de Dados MVVM.....	12
3.2. Arquitetura de Sistema.....	12
3.2.1. Camada de Apresentação (View):.....	12
3.2.2. Camada de Lógica de Apresentação (ViewModel):.....	13
3.2.3. Camada de Modelo:.....	13
3.2.4. Integrações Externas:.....	14
3.2.5. Segurança.....	14
3.2.6. Tecnologias.....	14
4. Visão de Casos de Uso.....	15
4.1. Casos de Uso Relevantes para a Arquitetura.....	16
4.1.1. Caso de uso FG01:.....	16
4.1.2. Caso de uso FE01:.....	17
4.1.3. Caso de uso FE02:.....	17
4.1.4. Caso de uso FH01:.....	18
5. Visão Lógica.....	18
6. Visão de Implantação.....	20
6.1. Hardware:.....	20
6.1.1. Servidores.....	20
6.2. Rede:.....	20
6.2.1. Configuração de Rede.....	20
6.3. Software:.....	20
6.3.1. Sistema Operacional.....	20
6.3.2. Banco de Dados em Nuvem (AWS EC2).....	20
6.4. Segurança:.....	20
6.4.1. Módulo de Autenticação.....	20
6.5. Integrações Externas:.....	20
6.5.1. Near Field Communication (NFC).....	20
6.6. Escalabilidade:.....	20
6.6.1. Escalabilidade Horizontal.....	20

6.7. Manutenção:.....	21
6.7.1. Atualizações Contínuas.....	21
7. Tamanho e Desempenho.....	21
8. Qualidade.....	21

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

2.1. Atributos de Qualidade Prioritários

Baseado no escopo descrito, nos requisitos funcionais e nos requisitos não-funcionais, e nos casos de uso mapeados, foram definidos os seguintes atributos de qualidade prioritários para o sistema:

2.1.1. Segurança

Como serão manipuladas credenciais de login dos usuários, segurança tornou-se um dos atributos prioritários na modelagem da arquitetura.

2.1.2. Manutenibilidade

Como os ambientes internos mapeados, as designações deles, e os mapas e identificadores correspondentes poderão sofrer alterações, manutenibilidade tornou-se um dos atributos de qualidade prioritários na modelagem arquitetural.

2.1.3. Usabilidade

Como o objetivo é facilitar a locomoção do usuário por ambientes novos, a usabilidade do aplicativo tornou-se um atributo de qualidade a ser considerado prioritário, pois sua utilidade e sua facilidade de uso farão com que o usuário o use. Se pedir direções para alguém parecer mais fácil do que utilizar a aplicação, ela não é usável o suficiente.

2.1.4. Portabilidade

Portabilidade é o atributo de qualidade central, sendo fundamental para a concepção da proposta do projeto em si. Como o usuário precisa de uma forma de se localizar em movimento, e de continuar seu uso em outro dispositivo, a portabilidade foi definida como atributo de prioridade máxima.

2.2. Funcionalidades e Restrições Arquiteturais

São descritas como os requisitos, funcionais e não-funcionais. Elas delimitam o escopo e o objetivo do projeto.

ID	Tipo	Descrição
RNF01	Requisito não-funcional	O sistema deve estar ativo em 95% do tempo.
RNF02	Requisito não-funcional	O sistema deve fornecer informações de localização em tempo real com um tempo de resposta médio inferior a 5 segundos.
RNF03	Requisito não-funcional	O sistema deve estar disponível 24 horas por dia, 7 dias por semana, garantindo acesso contínuo aos usuários.
RNF04	Requisito não-funcional	O acesso ao sistema e aos dados de localização deve ser protegido por autenticação segura.
RNF05	Requisito não-funcional	As informações de localização dos usuários devem ser tratadas com

		confidencialidade e não podem ser compartilhadas com terceiros sem a devida autorização.
RNF06	Requisito não-funcional	O sistema deve ser capaz de lidar com um aumento gradual no número de usuários, suportando pelo menos 20% de crescimento anual.
RNF07	Requisito não-funcional	A arquitetura do sistema deve permitir escalabilidade horizontal para acomodar futuras expansões.
RNF08	Requisito não-funcional	A interface do usuário deve ser intuitiva e de fácil utilização, garantindo que estudantes possam facilmente entender e seguir as direções fornecidas.
RNF09	Requisito não-funcional	O mapeamento geográfico deve ser realizado de forma eficiente para garantir precisão nas rotas calculadas.
RNF10	Requisito não-funcional	Atualizações de software devem ser implementadas sem impactar significativamente a disponibilidade do sistema.
RNF11	Requisito não-funcional	O sistema deve ser otimizado para dispositivos móveis, garantindo uma experiência consistente em diferentes plataformas.
RNF12	Requisito não-funcional	O sistema deve ser capaz de se recuperar automaticamente de falhas, minimizando impactos nos serviços prestados aos usuários.

ID	Tipo	Descrição
RF01	Requisito funcional	O sistema deve oferecer métodos seguros de autenticação para garantir o acesso apenas a usuários autorizados.

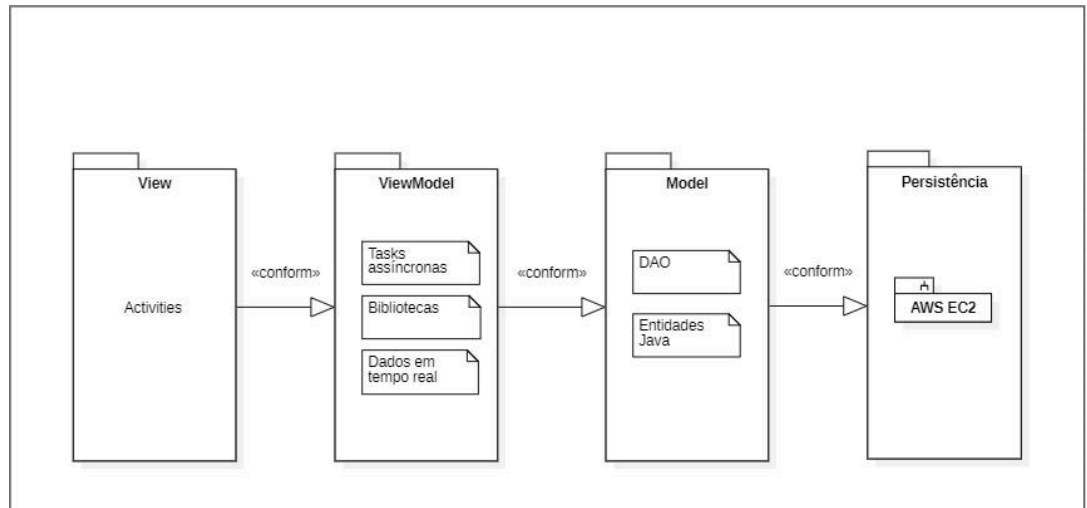
RF02	Requisito funcional	O sistema deve ser capaz de determinar a localização do usuário usando tags NFC instaladas nos edifícios do Instituto de Informática, Centro de Aulas Baru e Centro de Aulas Caraíbas.
RF03	Requisito funcional	O sistema deve incluir um mapa detalhado dos edifícios do Instituto de Informática e dos Centros de Aula, indicando a disposição das salas e áreas relevantes.
RF04	Requisito funcional	Com base na localização atual e no destino desejado do usuário, o sistema deve calcular a rota mais eficiente, considerando dados da universidade, mapeamento geográfico e contexto situacional.
RF05	Requisito funcional	O sistema deve fornecer direções claras e precisas para orientar o usuário até o destino desejado, utilizando informações visuais e/ou auditivas.
RF06	Requisito funcional	As informações fornecidas pelo sistema devem ser atualizadas em tempo real para refletir alterações na disposição das salas, eventos ou situações excepcionais.
RF07	Requisito funcional	O sistema deve ser capaz de integrar-se aos sistemas da universidade para obter informações precisas sobre horários de aulas, eventos e qualquer outra informação relevante.
RF08	Requisito funcional	O sistema deve manter um histórico das rotas percorridas pelos usuários para facilitar revisões ou análises retrospectivas.
RF09	Requisito funcional	O sistema deve ser acessível por meio de diferentes plataformas, incluindo aplicativos móveis (iOS e

		Android).
RF10	Requisito funcional	Deve ser possível realizar atualizações regulares do software para incorporar melhorias, correções de bugs e novos recursos sem causar interrupções significativas no serviço.
RF11	Requisito funcional	O sistema deve permitir que os usuários forneçam feedback sobre a precisão das direções e a usabilidade geral do sistema.
RF12	Requisito funcional	O sistema deve permitir a gestão eficiente de usuários, incluindo atribuição de diferentes níveis de permissões, como administrador e usuário.
RF13	Requisito funcional	O sistema deve oferecer funcionalidades básicas de navegação mesmo em situações de falta de conectividade, permitindo que os usuários acessem informações previamente baixadas.
RF14	Requisito funcional	Deve ser possível integrar o sistema com outros serviços ou aplicativos relevantes, proporcionando uma experiência mais abrangente e conectada.
RF15	Requisito funcional	O sistema deve implementar um mecanismo de backup em nuvem para garantir a segurança e a recuperação eficiente de dados em caso de falha do sistema, ou de troca de dispositivo do usuário.

3. Metas e Restrições da Arquitetura

3.1. Arquitetura de Software

Definida através do diagrama arquitetural seguinte. Segue o modelo arquitetural MVVM (Model-View-View/Model).



3.1.1. Modelo Arquitetural MVVM

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 Kotlin. 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 MVVM

- 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.
 - 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 implementando o Firebase com login da Google 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 Kotlin e banco de dados em nuvem (AWS)

- **Tecnologia Front-End**
 - Kotlin
- **Tecnologia Back-End**
 - Kotlin

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:

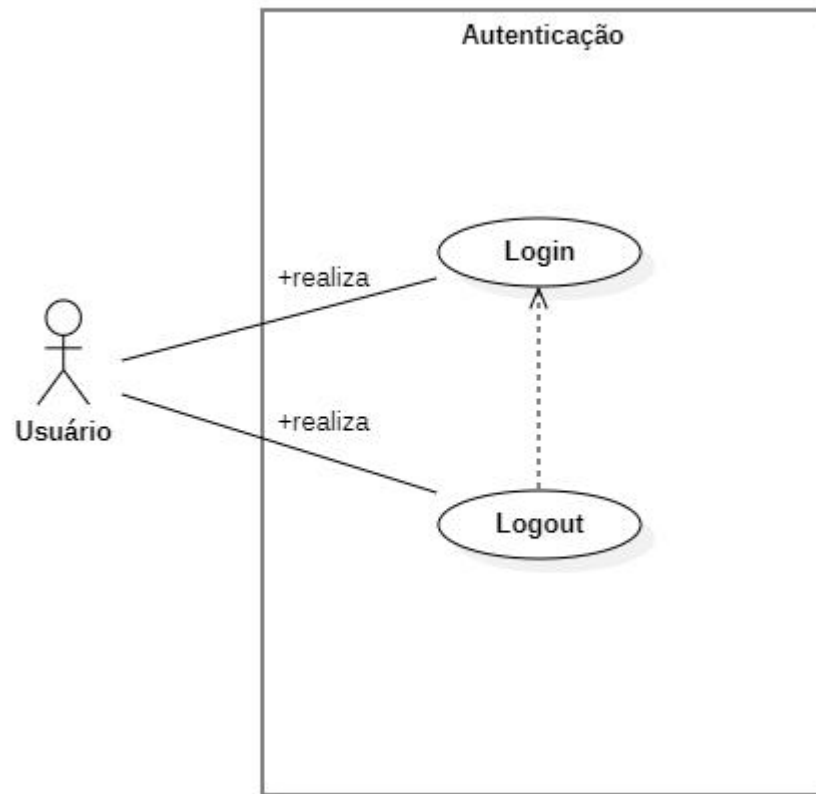
- **Caso de uso FG01**
 - Fazer login no Indoor INF como usuário.
- **Caso de uso FE01**
 - Visualizar o mapa interno do INF.
- **Caso de uso FE02**
 - Visualizar o mapa interno do CAB.
- **Caso de uso FE03**
 - Encontrar salas e banheiros do INF e do CAB.
- **Caso de uso FE04**
 - Encontrar coordenação do INF.
- **Caso de uso FH01**
 - Gerenciar usuário e sistema.

Cenários:

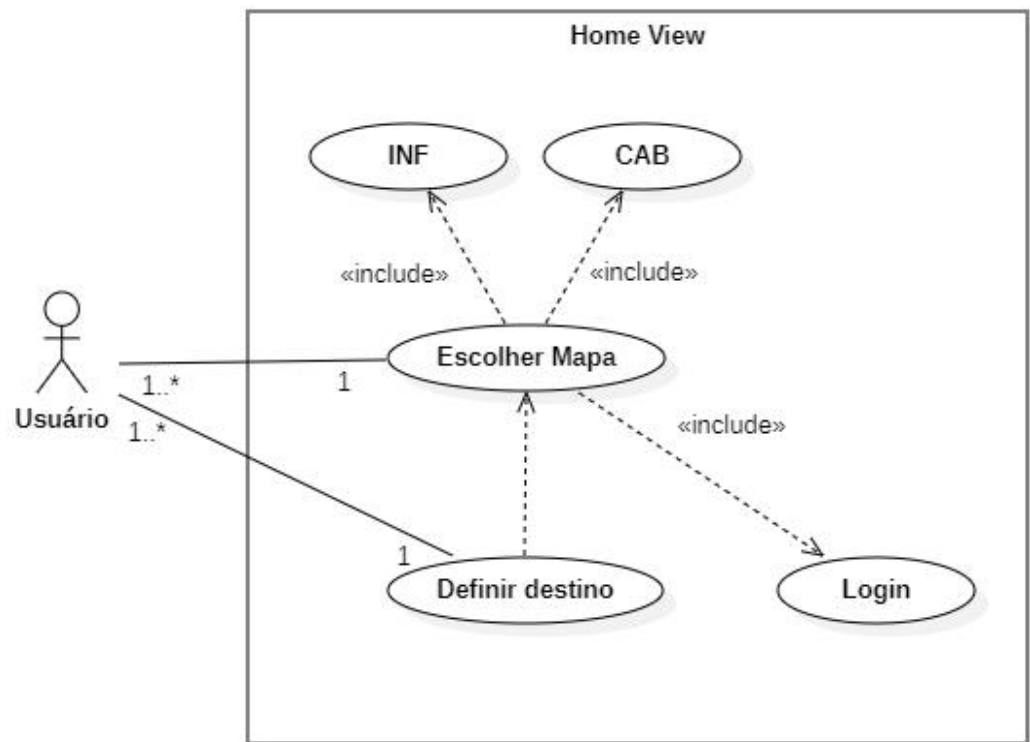
- **Cenário FG01-C01**
 - Logar-se com sucesso
- **Cenário FG01-C02**
 - Logar-se sem sucesso
- **Cenário FG01-C03**
 - Logar-se sem sucesso
- **Cenário FG01-C04**
 - Logar-se sem sucesso
- **Cenário FE01-C01**
 - Acessar o mapa do INF
- **Cenário FE01-C02**
 - Encontrar a sala da coordenação do curso 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

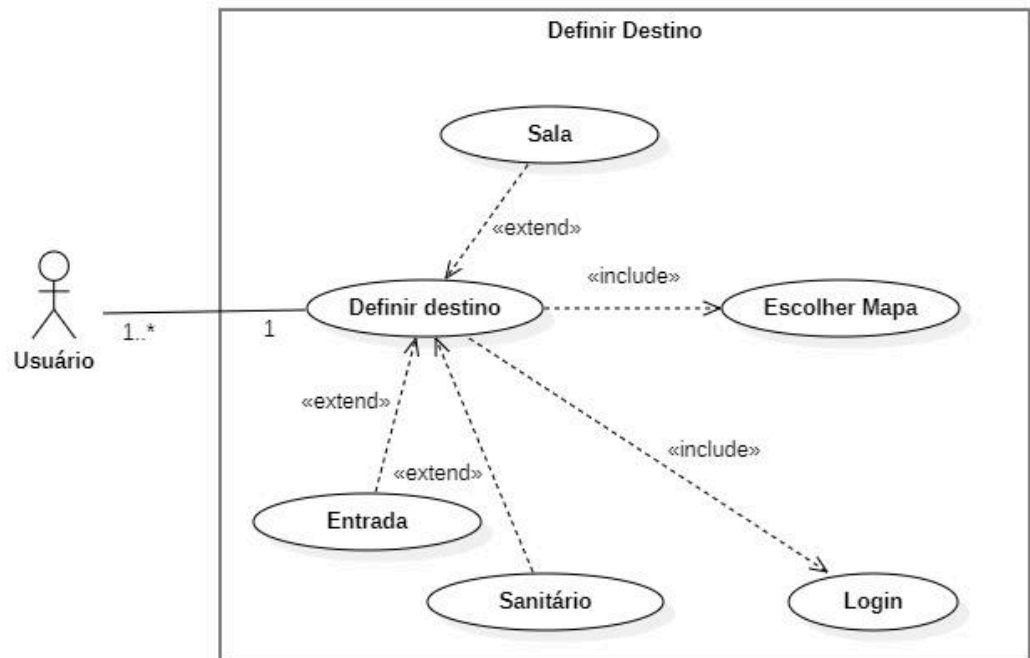
4.1.1. Caso de uso FG01:



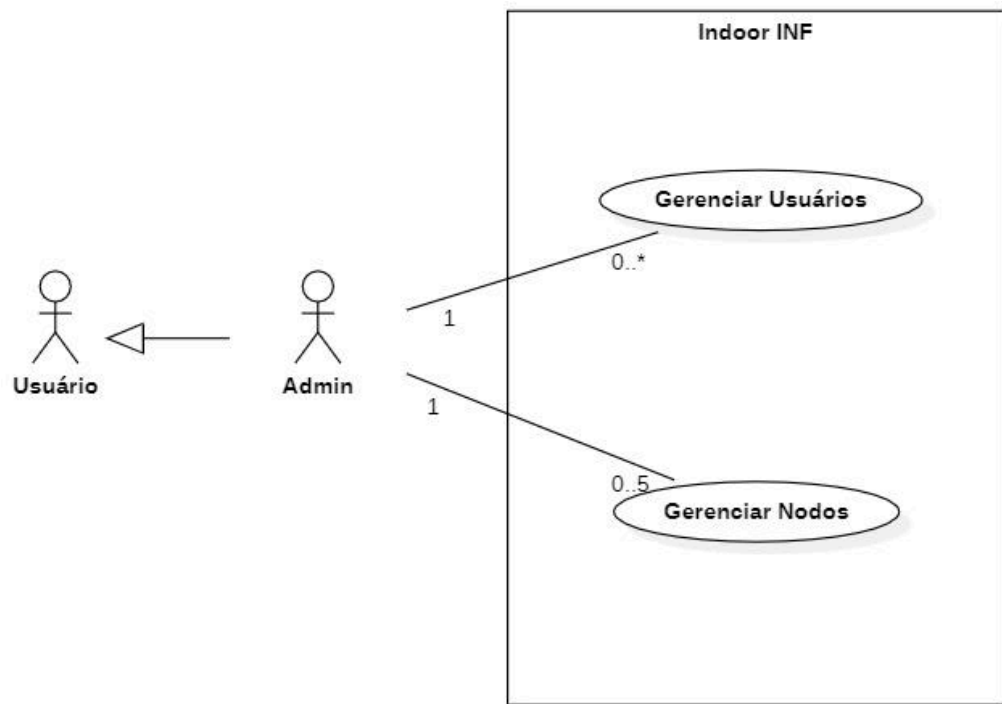
4.1.2. Caso de uso FE01:



4.1.3. Caso de uso FE02:



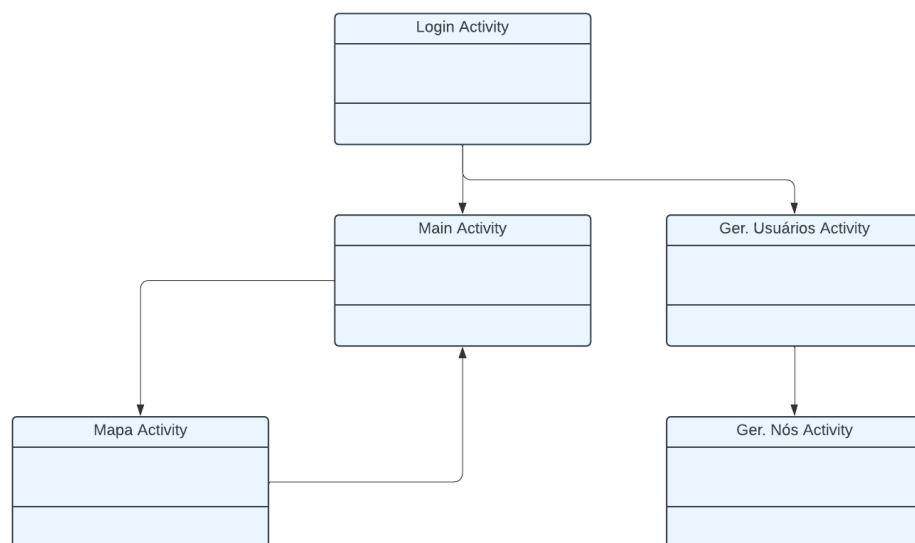
4.1.4. Caso de uso FH01:



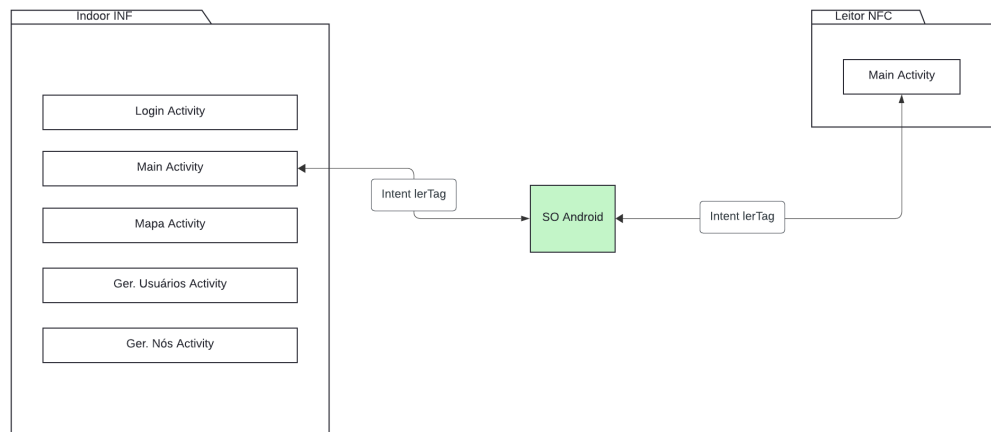
5. Visão Lógica

O sistema é composto por quatro principais camadas: a camada de apresentação (View e ViewModel), a camada de lógica de negócios (Model) e a camada de dados.

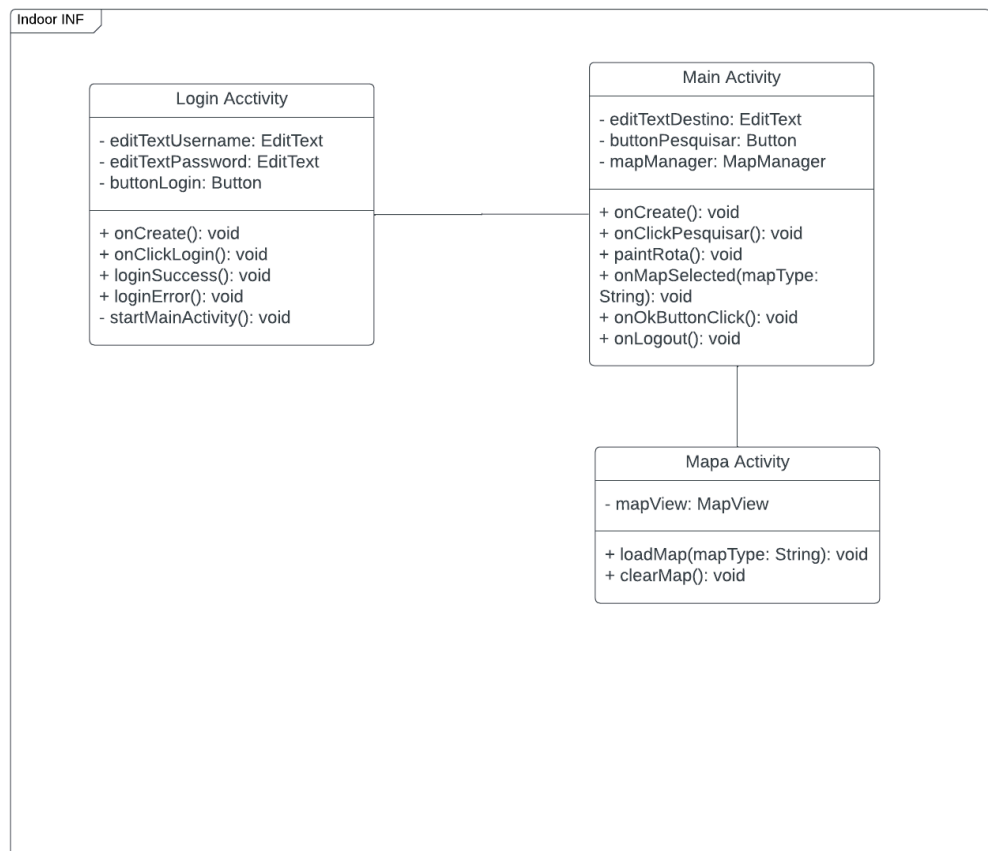
- **Camada de Apresentação:** constituída das cinco *activities* definidas, a camada de apresentação representa as interfaces por onde os dados são recebidos do usuário, ou exibidos a ele.



- Camada de Lógica de Negócios:



Tais activities funcionam como classes, representadas então no diagrama de classes a seguir:



6. Visão de Implantação

6.1. Hardware:

6.1.1. Servidores

- Assinatura de serviços de hospedagem para o aplicativo e o banco de dados em nuvem.

6.2. Rede:

6.2.1. Configuração de Rede

- Garantia de conectividade eficiente para interações com o banco de dados em nuvem.

6.3. Software:

6.3.1. Sistema Operacional

- Configuração do sistema operacional nos servidores, como o AWS EC2.
- Utilização de tecnologias de virtualização, se necessário.

6.3.2. Banco de Dados em Nuvem (AWS EC2)

- Armazenamento de dados do sistema de forma segura e escalável.

6.4. Segurança:

6.4.1. Módulo de Autenticação

- Integração do módulo de autenticação para gerenciar logins e sessões de usuários.
- Implementação do Firebase do Google Sign-in para autenticação segura.

6.5. Integrações Externas:

6.5.1. Near Field Communication (NFC)

- Implementação da comunicação com as tags NFC para identificação de salas.
- Atualização em tempo real do mapa com base nas leituras NFC.

6.6. Escalabilidade:

6.6.1. Escalabilidade Horizontal

- Configuração para escalabilidade automática dos servidores em caso de aumento de demanda.

6.7. Manutenção:

6.7.1. Atualizações Contínuas

- Estratégias para implementação de atualizações contínuas, minimizando o impacto nos usuários.
- Monitoramento proativo para identificar e corrigir problemas rapidamente.

7. Tamanho e Desempenho

O sistema Indoor INF apresenta um código fonte modular e eficiente, aproveitando as tecnologias Kotlin para o front-end e back-end. O banco de dados na nuvem (AWS) é estruturado para armazenar dados de forma escalável. O sistema inclui elementos visuais como mapas interativos, marcadores e setas.

Em relação ao desempenho, a arquitetura Kotlin proporciona respostas rápidas a solicitações do usuário. Estratégias de cache e otimização de consultas ao banco de dados garantem eficiência. A comunicação com as tags NFC é otimizada para atualizações em tempo real do mapa. A segurança é reforçada com autenticação usando e-mail institucional e criptografia de dados.

Estratégias para melhorar o desempenho incluem otimização de código, uso de cache, escalabilidade automática e monitoramento contínuo. Testes de desempenho são realizados para avaliar a capacidade do sistema em diferentes condições de carga. O objetivo é garantir uma experiência positiva e responsiva aos usuários do sistema Indoor INF.

8. Qualidade

A garantia de qualidade para o sistema Indoor INF é abordada em várias frentes. Isso inclui testes abrangentes para funcionalidades, usabilidade e segurança, bem como práticas para assegurar desempenho eficaz. Medidas de confiabilidade, como escalabilidade e procedimentos de backup, são implementadas. A documentação clara facilita a manutenção, e práticas de integração contínua garantem entregas consistentes. A usabilidade é validada por meio de testes com usuários reais, incorporando feedback contínuo para melhorias. O compromisso com a qualidade visa proporcionar uma experiência confiável, eficiente e satisfatória aos usuários.