

Desenvolvimento de App (Android) de conscientização dos índices de criminalidade por região no estado de São Paulo

Leandro Martins, Carila Campitelli, Lucas Arcanjo, Nadia Ayala, Thiago Carvalho, Lucivaldo Queiroz¹

Orientadores: Professora Me. Simone de Abreu e Professor Esp. Victor Williams Stafusa da Silva

Faculdade Impacta de Tecnologia
São Paulo, SP, Brasil
16 de junho de 2021

Resumo. Criação de um aplicativo voltado para a segurança pública, utilizando a localização indicada pelo usuário com a geolocalização da API do Google para a busca do endereço a ser localizado. Possui também uma base de dados, com os dados de criminalidade retirados do site da Segurança Pública do Estado de São Paulo e esses dados são demonstrados em forma de gráfico para a conscientização do usuário.

Palavras-chave: segurança pública, conscientização, criminalidade.

1. Introdução

O estado de São Paulo é muito grande tendo locais mais seguros e não seguros, levando em consideração os índices de criminalidade no estado de São Paulo teve se a ideia de ter um aplicativo onde localizamos o endereço e verificamos qual o índice de criminalidade daquela área.

1.1. O problema

A cidade de São Paulo é conhecida internacionalmente por ser a maior cidade da América Latina, com mais de onze milhões de pessoas. Junto ao crescimento no número de habitantes, a cidade passou pelo aumento da violência, com números que chegaram a 3.038 para vítimas de homicídio no ano de 2020.

O aplicativo +Seguro busca promover conscientização aos habitantes com relação aos números de crimes que ocorrem em cada lugar da cidade. É baseado em um aplicativo de segurança que, usando a localização indicada pelo usuário, verifica em um banco de dados o histórico da criminalidade naquela região, exibindo informações de maneira estruturada ao usuário final.

O sistema de prevenção e proteção ao cidadão paulistano busca reunir e tornar acessíveis dados relacionados à criminalidade em todo o estado de São Paulo. O problema é que nem todos tem acesso a esses dados de criminalidade no estado de São Paulo, a solução seria o App +Seguro onde o usuário teria esse acesso com dados mais trabalhados para a sua melhor verificação e visualização.

¹ Os autores podem ser contatados respectivamente através de seus correios eletrônicos:

lndmrt6@gmail.com, carila.campitelli@gmail.com, lucas.saolliveira@gmail.com, nadiaayalanogueira@gmail.com, thiagodot@gmail.com, lnpq.dev@gmail.com.

O benefício para o usuário será a comodidade da verificação da localização e os dados de criminalidade no seu celular.

1.2. Apresentação da empresa +Seguro

A empresa +Seguro é nova no mercado com apenas um ano de vida, lançando um produto para a entrada no mercado.

O público-alvo da +Seguro, varia de acordo com a procura, pois é bem diversificado para quem está à procura de comprar uma casa ou alugar e verificar a localização, solicitar um serviço que você terá que ir ao local, contratar uma escola, entre outros.

O sistema +Seguro foi desenvolvido com foco no usuário que é habitante da cidade de São Paulo e não se sente seguro em frequentar, principalmente, locais desconhecidos.

Inicialmente, o aplicativo é disponibilizado de maneira gratuita, com o objetivo de criar uma base de usuários que, ao ser alcançado, servirá como base para o modelo “freemium” (um modelo de negócio no qual um produto é disponibilizado de forma gratuita, porém, que permite a adição de recursos mediante pagamento) do aplicativo, que permitirá a remoção de anúncios.

Os anúncios serão a principal forma de capitalização do aplicativo, sendo vendidos para empresas que possam ter a ver com a área da segurança pública.

1.3. O problema da segurança pública

A segurança pública é um assunto que causa impacto direto na forma como os habitantes de uma cidade vivem. Quando um cidadão se vê diante da necessidade de realizar alguma tarefa em um local desconhecido, existe a preocupação sobre qual o nível de violência e criminalidade naquele determinado lugar.

Em algumas situações, os habitantes podem até deixar de realizar alguma atividade por falta de conhecimento sobre o local (como atividades no período noturno, quando a cidade tende a estar mais vazia).

Nesse contexto, é possível enxergar o problema da falta de conhecimento do nível real da segurança em um ponto da cidade. Embora as informações sobre ocorrências estejam disponíveis de forma pública na internet, o processo para realizar o acesso é complexo e exige um processo de aquisição e tratamento dos dados que se dificulta o uso dessas informações por parte dos maiores interessados: os habitantes da cidade.

A empresa está entrando no mercado com a ideia do App +Seguro e os anúncios serão a principal forma de capitalização monetária num primeiro momento, sendo vendidos para empresas que possam ter a ver com a área da segurança pública.

1.4. Stakeholders e Restrições

Os stakeholders envolvidos nesse projeto são os usuários, que são cidadãos paulistanos que usarão o sistema para acessar os índices de criminalidade por região e se sentirem mais seguros nas escolhas de locais que frequentarão, a empresa, que poderá rentabilizar em cima do fornecimento do aplicativo, e o próprio governo do estado de São Paulo, que fornece os dados necessários para o uso do sistema.

As únicas restrições consideradas durante o desenvolvimento do sistema foram a exigência de conformidade com a lei nº 13.709/2018 (LGPD), visto que são coletados dados dos usuários do aplicativo, e a funcionalidade restrita a São Paulo, visto que o público-alvo do aplicativo são apenas os cidadãos paulistanos.

2. Solução proposta

Para resolver o problema da dificuldade no acesso às informações públicas sobre ocorrências criminais na cidade de São Paulo, foi desenvolvido o aplicativo +Seguro.

O sistema tem uma base de dado com os dados retirados da Segurança Pública do Estado de São Paulo, é consumido essas informações, processando-as e exibindo um resultado visual, rápido e compreensível para o habitante paulistano, que poderá usufruir dessa conscientização independentemente do seu nível de conhecimento de tecnologia.

O projeto se baseia num serviço de comodidade para o usuário para ele se manter informado referente aos índices de criminalidade dos locais dentro do estado de São Paulo.

Será apresentado via App +Seguro, somente para Android, onde terá uma tela principal, tela de cadastro para login, tela de usuário, tela para busca do endereço desejado e seus arredores, tela com o mapa para verificação da localização, na busca pela localização mostra na tela os índices de criminalidade do local, estes índices serão de roubo e furto.

2.1. Requisitos do sistema

Quadro 1 – Requisitos.

Requisito	Descrição
REQ 01	O sistema deve permitir o abastecimento dos dados de criminalidade.
REQ 02	O sistema deve permitir que o cadastro de usuário seja por e-mail ou CPF.
REQ 03	O sistema deve permitir acesso pelo login que será o <i>e-mail</i> do usuário.
REQ 04	O sistema deve permitir que o e-mail do usuário seja sua confirmação de identidade e enviar uma confirmação para o <i>e-mail</i> .
REQ 05	O sistema deve solicitar o aceite do termo de uso antes de salvar os dados pessoais.
REQ 06	O sistema deve solicitar nos dados pessoais o login e senha antes de salvar.
REQ 07	O sistema deve solicitar o endereço a ser localizado os índices de furto e roubo (com dados digitados e/ou localização via endereço).
REQ 08	O sistema deve mostrar os índices de criminalidade para usuário de forma clara.

Fonte: Os autores.

2.2. Regras de negócio

Quadro 2 – Regras de negócio.

Requisito	Descrição
RN0001	O administrador do sistema é quem abastece o banco de dados com as informações de criminalidade de furto e roubo.
RN0002	O usuário do App cadastra seus dados pessoais, <i>login</i> e senha.

Requisito	Descrição
RN0003	O usuário do App aceita o termo de uso.
RN0004	Todo usuário cadastrado recebe um e-mail com confirmação de identidade.
RN0005	Todo usuário para a consulta dos dados deverá digitar o endereço corretamente sem faltar informações para a localização.

Fonte: Os autores.

2.3. Descrição das funcionalidades

2.3.1. Home User

Ao abrir o aplicativo, o usuário visualiza uma página principal que fornece as opções de login e cadastro. Essa é a primeira tela de contato do usuário com o App +Seguro, nesta tela o usuário tem as informações de como funciona o aplicativo +Seguro.

2.3.2. Cadastro de Cliente

Ao abrir o aplicativo, o usuário visualiza uma página principal que fornece as opções de login e cadastro. Ao clicar em cadastro, o usuário é direcionado para a página de cadastro, na qual são solicitados os dados para preenchimento. Após inserir os dados, é solicitado que o usuário aceite os termos de uso do aplicativo. Ao clicar em cadastrar, o aplicativo valida se os dados foram preenchidos corretamente e se os termos de uso foram aceitos. Se todas as informações estiverem de acordo com o esperado, o usuário recebe um e-mail com a validação do acesso. Após essa validação o usuário é direcionado a página inicial do aplicativo. Caso os dados estejam incorretos, o usuário é travado e não consegue seguir com o login.

2.3.3. Login

Ao abrir o aplicativo, o usuário visualiza uma página principal que fornece as opções de login e cadastro. Ao clicar em login, o usuário é direcionado para a página de login, na qual devem ser inseridos seu e-mail e senha.

Ao clicar em entrar, o aplicativo realiza a validação dos dados, checando a existência de um usuário com aquelas informações atribuída, caso os dados estejam corretos, o usuário é direcionado à página inicial do aplicativo. Caso os dados estejam incorretos, é solicitado que o usuário verifique as informações inseridas.

2.3.4. Busca/Pesquisa Local

Ao clicar no botão de busca, o usuário é direcionado à uma página de busca na qual são inseridos os dados do endereço do local a ser pesquisado através do CEP.

Ao selecionar a opção “relatório”, os resultados com ocorrências para aquele determinado lugar são exibidos na tela.

2.3.5. Mapas

Após o preenchimento do local o endereço ou geolocalização a ser buscado, é mostrado um mapa, isso devido a API com o Google Maps, da região que está procurando.

2.3.6. Gráficos

Após a inserção dos dados de localização é mostrado no mapa a localização os gráficos de criminalidade referente a área buscada, estes gráficos são de pizza, torre e linhas.

3. Projeto, análise e implementação

3.1. Arquitetura, módulos e subsistemas

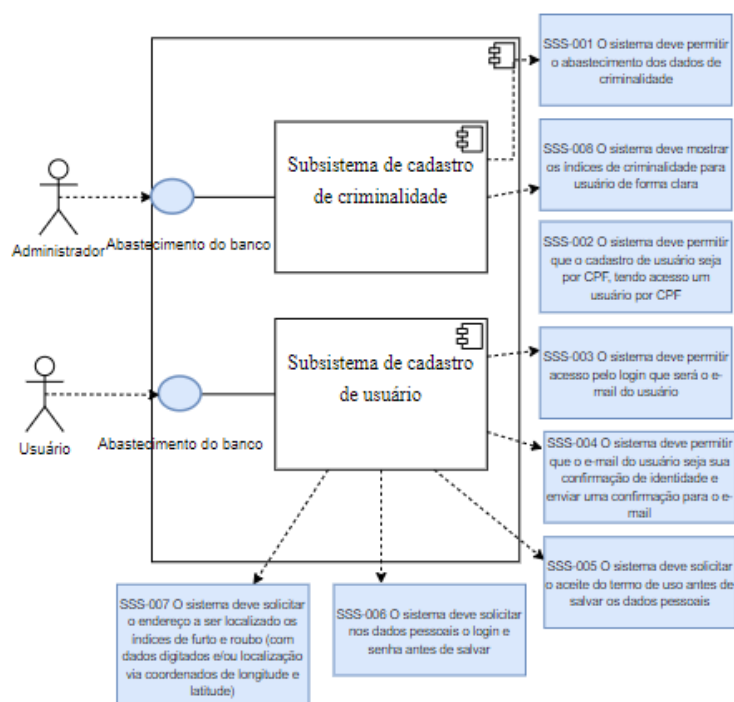
Processos:

1. Cadastro dos dados de criminalidade furto e roubo.
2. Cadastro com dados dos usuários.
3. Cadastro de login e senha.
4. Aceite de termos de uso.
5. Envio de e-mail para a confirmação.
6. Cadastro de endereço com digitação para localização dos dados de criminalidade de furto e roubo.
7. Verificação da performance da localização cadastrada com os dados de criminalidade de furto e roubo.

Quadro 03 – Entidades externas.

Entidades Externas:
1. Administrador
2. Usuário
Subsistema de cadastro dos dados de criminalidade: Responsável pelo cadastro das informações de criminalidade que será localizada pelo usuário. Envolve os processos 1 e 7, e a entidade externa administrador do sistema com abastecimento da base de dados.
Subsistema de cadastro do usuário: Responsável pelo cadastro das informações do usuário. Envolve os processos 2,3,4,5 e 6 e a entidade externa usuário.

Figura 1 – Sistema.



Fonte: Os autores.

Quadro 4 – Subsistemas.

Descrição dos Subsistemas
Nome do Subsistema: Dados de Criminalidade Descritivo: Sistema com base de dados, abastecida com dados de criminalidade de furto e roubo do estado de São Paulo. Requisitos do sistema: SSS-001- O sistema deve permitir o abastecimento dos dados de criminalidade SSS-008 - O sistema deve mostrar os índices de criminalidade para usuário de forma clara Interface com Atores Ator Administrador: Abastecimento da base de dados Ator Usuário: Abastecimento de dados pessoais Interfaces com Outros Subsistemas Subsistema de cadastro dos dados de criminalidade: Fluxo 01 – Administrador Subsistema de cadastro do usuário: Fluxo 01- Usuário
Nome do Subsistema: Dados do Usuário Descritivo: Sistema com dados de cadastro do usuário. Requisitos do sistema: SSS-002 - O sistema deve permitir que o cadastro de usuário seja por e-mail ou CPF. SSS-003 - O sistema deve permitir acesso pelo login que será o e-mail do usuário SSS-004 - O sistema deve permitir que o e-mail do usuário seja sua confirmação de identidade e enviar uma confirmação para o e-mail SSS-005 - O sistema deve solicitar o aceite do termo de uso antes de salvar os dados pessoais

SSS-006 - O sistema deve solicitar nos dados pessoais o login e senha antes de salvar
SSS-007 - O sistema deve solicitar o endereço a ser localizado os índices de furto e roubo (com dados digitados)

Interface com Atores
Ator Administrador: Abastecimento da base de dados
Ator Usuário: Abastecimento de dados pessoais
Interfaces com Outros Subsistemas
Subsistema de cadastro dos dados de criminalidade: Fluxo 01 – Administrador
Subsistema de cadastro do usuário: Fluxo 01- Usuário

3.2. Projeto de banco de dados

O banco de dados é relacional, é baseado em relações de tabelas com linhas e colunas, que são armazenadas e controladas por um sistema gerenciador de banco de dados, ou SGBD. Ficando assim estruturado, os dados são facilmente armazenados e recuperados por meio do SQL Server. Além disso, ele é restritivo ao acesso e também permitir o uso em camadas.

O nosso agente foi feito em C#, o framework 5.0, o tipo do projeto de interface é console, onde tem uma tela preta. Não foi encontrado nada que faça uma integração direta com o portal da polícia federal. No portal, baixa um arquivo XML, no link, desse link, importa o dado de homicídio doloso, lesão corporal seguido de morte, morte suspeita, furto de celular, feminicídio, furto de veículo, roubo de celular, latrocínio e roubo de veículo.

Os dados são baixados em Excel, disponibilizado com 60 dias de atraso da data da ocorrência, lote fechado, é disponibilizado com 2 meses depois da ocorrência, pacote fechado, disponibilizam o mês completo. A importação é realizada mês a mês e por categoria.

A baixa é feita em arquivo em Excel e converte em CSV, após o arquivo ser convertido em CSV, temos que salvar em endereço específico, nossa aplicação efetua a captura de todos os arquivos neste endereço, que tenha a extensão .csv.

Após ele inicia a importação toda vez que faz a importação gera um número de lote, esse número é salvo na base de dados, "SQL Server", No SQL Server, a importação de dados é realizada por lote estando na última coluna com nome de lote. A nossa persistência de dados foi de 50 em 50 dados, cada dado é uma ocorrência, fazendo de 50 em 50 para não sobrecarregar o nosso servidor. Para cada dado disponibilizado foi criado 4 tabelas para o CRUD, são no total 24 tabelas.

3.3. Estrutura do sistema

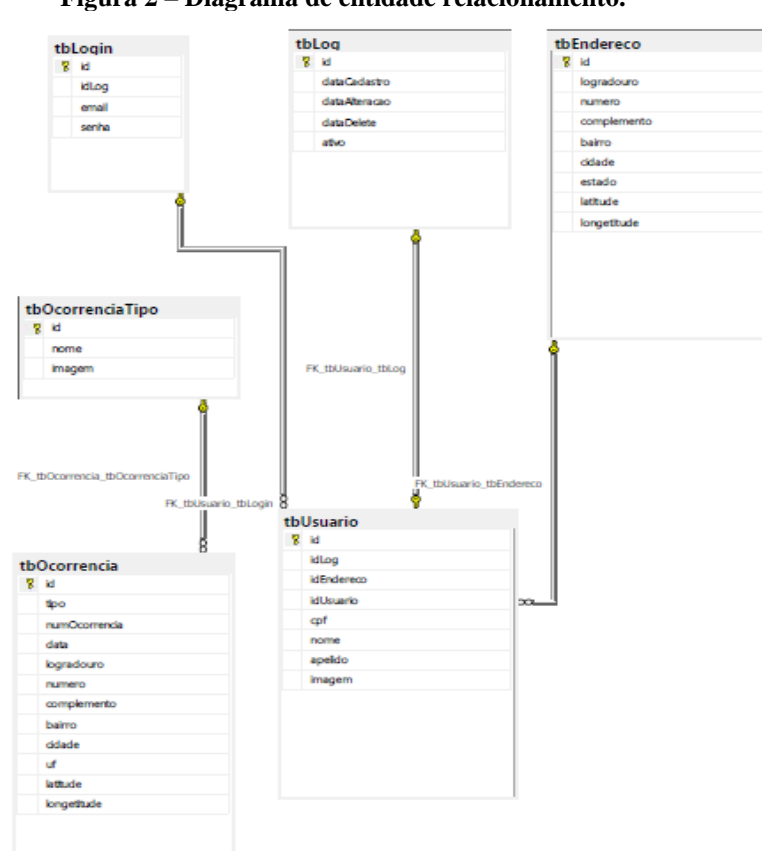
No banco de dados, temos seis tabelas, segue nome das tabelas, os campos e seu relacionamentos - FK foreign key.

- tbLogin, com os campos: id, idLog, email, senha com relacionamento FK_tbUsuario_tbLogin. Nesta tabela ficaram as informações referente a login do usuário.
- tbLog, com os campos: id, dataCadastro, dataAlteracao, dataDelete, ativo com relacionamento FK_tbUsuario_tbLog. Nesta tabela ficaram as informações de CRUD, que são criar, ler, atualizar e deletar as informações do usuário.
- tbEndereço, com os campos, id, logradouro, numero, complemento, bairro, cidade, estado, com relacionamento FK_tbUsuario_tbEndereco. Nesta tabela ficará armazenado endereço que o usuário digitar na localização.

- tbOcorrenciaTipo, com os campos, id, nome, imagem com relacionamento FK_tbOcorrencia_tbOcorrenciaTipo. Nesta tabela ficarão armazenadas as ocorrências que serão retiradas das bases de Índice de Criminalidade do Estado de São Paulo.
- tbOcorrencia, com os campos, id, tipo, numOcorrencia, data, logradouro, numero, complemento, bairro, cidade, uf, com relacionamento FK_tbOcorrencia_tbOcorrenciaTipo. Nesta tabela ficarão armazenadas as ocorrências que serão retiradas das bases de Índice de Criminalidade do Estado de São Paulo.
- tbUsuario com os campos, id, idLog, idEndereco, idUsuario, cpf, nome, apelido, imagem FK_tbUsuario_tbEndereco. Nesta tabela ficarão as informações do usuário e seus dados pessoais.

3.3.1. Diagrama Entidade Relacionamento

Figura 2 – Diagrama de entidade relacionamento.



Fonte: Os autores.

3.3.2. Backend

O Backend foi desenvolvido em C#, usando a tecnologia .Net Core 3.1 e a sua estrutura foi feita utilizando a arquitetura de Clean Architecture, utilizamos a estrutura de injeção de dependências usando a padronização, interfaces e controllers, conforme a imagem:

Fonte: Os autores.

3.3.3. Frontend

Nosso Frontend foi desenvolvido para Android SDK 16 até SDK 30, para consumir os dados do nosso backend foi utilizado a tecnologia do Retrofit que consome os dados do Backend, para a injeção de dependência foi utilizada a tecnologia do koin, nossa IDE de desenvolvimento o Android Studio, a estrutura do desenvolvimento foi feita em MVVM (Model – View – View – Model).

3.3.4. Estrutura do aplicativo

O sistema APP +Seguro foi baseado utilizando a arquitetura em aplicações Android, Clean Architecture, os códigos se dividem em três camadas distintas:

- Entidades (regras de negócio/comando de domínio)
 - Backend (o agente) – Models
 - Backend – Models
 - Banco de Dados – SQL Server
 - Backend - Controller
 - Backend – Repositório
- Casos de Uso (regras de negócio da aplicação)
 - Backend (o agente) – Repositório
 - Backend – Repositório
 - Android Studio
- Adaptadores de Interface:
 - Backend - Controller
- Frameworks e Drives
 - Backend (o agente) – Service

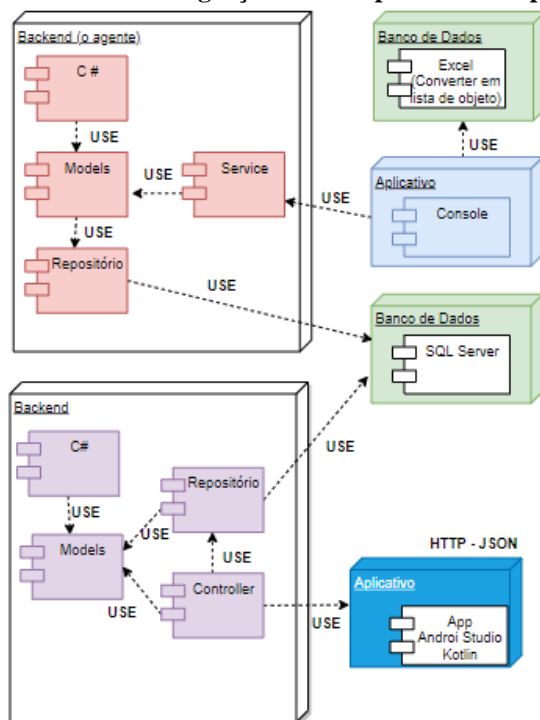
O principal foco é isolar o código de domínio da aplicação, com inversão de dependência (regras que interagem com o domínio), favorecer boas pratica de orientação a objeto, o código de domínio fica com alta testabilidade, gerando também a independência de

frameworks e detalhes de infraestrutura. Tornando assim, mais fácil a manutenção e evolução do software.

Backend - o agente para inserção dos dados de criminalidade retirados dos índices de criminalidade do Estado de São Paulo, este consome os dados no console. A parte de Backend foca em isolar o código de domínio do restante da aplicação, com uso da orientação a objeto e encapsulamento da linguagem, também temos uma API com o Google Maps para a geolocalização.

Como se trata de um aplicativo usamos o Android Studio com a linguagem Kotlin. Sua hospedagem está no <http://www.maisageis.kinghost.net/>.

Figura 4 – Estrutura da integração dos componentes do aplicativo.



Fonte: Os autores.

4. Testes da Aplicação

Os testes foram realizados no Swagger UI, no endereço <http://www.maisageis.kinghost.net/project/swagger/index.html>.

Foram realizados testes de POST, GET, PUT e DELETE usados na API do google referente a geolocalização o uso do banco de dados com as categorias e acesso de usuário.

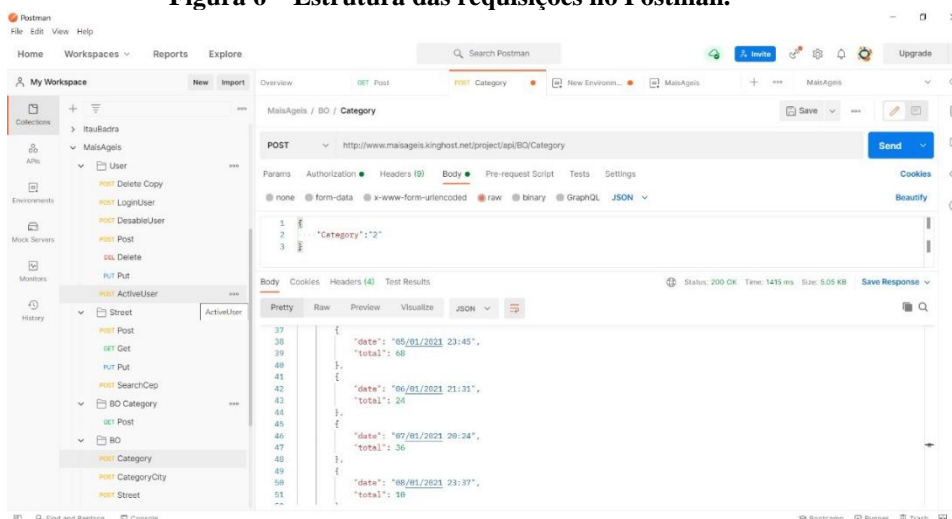
Figura 5 – Testes do aplicativo no Swagger UI.



Fonte: Os autores.

Os testes no Postman fazem a requisição para o nosso serviço de backend, enviando os dados e recebendo as respostas se foram cadastrados com sucesso.

Figura 6 – Estrutura das requisições no Postman.



Fonte: Os autores.

3.4. Tecnologias utilizadas

Quadro 9 - Tecnologias utilizadas.

Tecnologia	Justificativa
Git	Sistema de versionamento de código distribuído entre todos os desenvolvedores. Necessário também devido as ferramentas de hospedagem escolhidas.
GitHub	Site onde o código-fonte é hospedado e versionado. BackEnd: https://github.com/LeandroMartins744/MaisAgeis-BackEnd FrontEnd: https://github.com/LeandroMartins744/MaisAgeis-FronEnd Agente: https://github.com/LeandroMartins744/MaisAgeis-Agente

Tecnologia	Justificativa
C#	Linguagem de programação utilizada para desenvolver o <i>agente e o backend</i> do projeto.
SQL Server	Sistema gerenciador de banco de dados relacional utilizado para persistirem-se os dados cadastrais do projeto.
Android Studio	Android Studio fornece as ferramentas mais rápidas para a criação de apps para Android. Permite que você visualize os layouts na hora que está desenvolvendo, com isso fica mais fácil, pois não há necessidade de utilizar outro dispositivo ou emulador para a execução de interface. No próprio Android Studio, você consegue salvar o trabalho realizado, até mesmo os mais demorados.
Microsoft Visual Studio	Ambiente de desenvolvimento integrado (IDE) da Microsoft para desenvolvimento de software especialmente dedicado ao .NET Framework e às linguagens Visual Basic (VB), C, C++, C# (C Sharp) e F# (F Sharp).
Kotlin	Linguagem de programação utilizada para desenvolver o <i>frontend</i> do projeto.
Framework 5.0 - C#	O nosso agente foi feito em C#, o framework 5.0, o tipo do projeto de interface é console, onde tem uma tela preta
C# .Net Core 3.1	Tecnologia utilizada para o desenvolvimento do <i>backend</i> .
Swagger UI	Os testes foram realizados no Swagger, segue link: http://www.maisageis.kinghost.net/project/swagger/index.html
Retrofit	Retrofit que consome os dados do Backend
Postman	O Postman é uma ferramenta que dá suporte à documentação das requisições feitas pela API. Ele possui ambiente para a documentação, execução de testes de APIs e requisições em geral.
Kinghost	Utilizado o Kinghost para a hospedagem - http://www.maisageis.kinghost.net/ https://painel.kinghost.com.br/login.php?uri=/index.php&_ga=2.175123809.250871305.1622591881-1934148510.1622591881

Fonte: Os autores.

4. Considerações finais

A empresa está entrando no mercado com a ideia do App +Seguro e os anúncios serão a principal forma de capitalização monetária num primeiro momento, sendo vendidos para empresas que possam ter a ver com a área da segurança pública.

Foi realizada uma pesquisa de mercado, onde o produto deverá ser consideravelmente aceito.

Referências bibliográficas

MARTIN, Robert. The Clean Architecture. 2012. Disponível em: <<https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>>. Acesso em: 4 jun 2021.

SÃO PAULO, Governo Aberto. Conjuntos de dados. 2021. Disponível em: <<http://catalogo.governoaberto.sp.gov.br/dataset?q=ssp>>. Acesso em: 4 jun 2021.

SÃO PAULO, Transparência SSP. **Segurança Pública**. 2021. Disponível em: <<http://www.ssp.sp.gov.br/transparenciassp/Consulta.aspx>>. Acesso em: 4 jun 2021.

Agradecimentos

Agradeçamos primeiramente a Deus, aos nossos familiares por estarem nessa jornada de estudos conosco e pela compreensão de todos com muito amor. Agradecemos nossa orientadora Simone de Abreu e Victor Williams Stafusa da Silva, por toda a paciência e dedicação.