

APLICAÇÃO PARA CONSULTA DE CONTRATOS PÚBLICOS FEDERAIS

Rayan M. P. Cordeiro¹, Vinícius Soares Jovito², Gabriel Bezerra dos Santos³, Felipe Augusto de Souza⁴.

Ciência da Computação – Instituto de Ensino Superior de Brasília (IESB)
70.200-730 – Brasília – DF – Brasil.

rayanmpcordeiro@gmail.com, gabrielbezerra38@hotmail.com,
viniciusjovito@gmail.com, felipe-augusto09@hotmail.com.

Abstract. *To present the Transparency of federal public agencies and entities to the population through an android application, providing information about the number of contracts, expiration dates, values, categorization, and suppliers.*

Resumo. *Apresentação da Transparência de órgãos e entidades públicas federais à população através de uma aplicação android, disponibilizando informações sobre quantidade de contratos, vencimentos, valores, categorização e fornecedores.*

1. Introdução

A primeira versão do *Comprasnet Contratos* surgiu em 2013, como sistema de conta desenvolvido na tecnologia ScriptCase(plataforma de desenvolvimento de aplicações PHP - *Personal Home Page*), em que o mesmo só era utilizado dentro das unidades da AGU. Em 2014 e 2015 o sistema de conta foi oferecido a outros órgãos, mas o uso do sistema era descentralizado e o código-fonte era compartilhado pela Advocacia-Geral da União. Logo depois, de 2016 a 2018, 12 órgãos já utilizavam o sistema de forma descentralizada e em tecnologia ScriptCase. Nasce o *Comprasnet Contratos* via desenvolvimento colaborativo, que foi oficialmente lançado em janeiro de 2020.

2. Contextualização

O *Comprasnet Contratos* é uma ferramenta do governo federal que automatiza os processos de gestão contratual e conecta servidores públicos, que foi criada para reduzir problemas com rotinas de trabalho e ter controle total sobre as informações, sendo uma plataforma sem custos para órgãos e entidades públicas, já o aplicativo tem o objetivo de tornar essa ferramenta ainda mais acessível, reduzindo passos tidos como essenciais para a utilização da solução, como por exemplo: a obrigatoriedade do login para a busca e visualização de processos, tornando assim, a utilização mais simples e direta.

Através do módulo de importação de dados do SIASG (Sistema Integrado de Administração de Serviços Gerais) o usuário terá acesso automático a todos os contratos ativos da sua unidade (UASG-Unidades de Administração de Serviços Gerais), permitindo que o usuário tenha gestão e fiscalização das compras públicas.

3. Problema

Ao consultar os dados no site *Comprasnet Contrato*, o usuário que não está familiarizado com assuntos relacionados a essa área, se depara com as informações

dispostas de forma complexa e uma interface pouco intuitiva. À vista disso, notou-se a necessidade de apresentar esses dados de modo que a experiência do usuário seja priorizada.

4. Objetivo Geral

Desenvolver um aplicativo que viabilize a transparência de contratos dos órgãos e entidades públicas federais à usuários da plataforma *Android*, tendo em vista maior acessibilidade entre os cidadãos brasileiros, para aumentar o alcance da população a informações de grande importância como quantitativo de contratos, vencimentos, contratos acabados, valores, categorização e fornecedores que estão envolvidos no fornecimento de materiais e serviços para o Governo Federal.

4.1 Objetivo Específicos

Visando atender ao objetivo geral têm-se os seguintes objetivos específicos:

- Cadastrar usuário;
- Consultar processos de compras;
- Consultar localização dos órgãos com contratos em aberto.

5. Referencial Teórico

Esta seção foi estruturada em três tópicos, são eles: sistema integrado de administração e serviços gerais, tipos de dados fornecidos pelo SIASG, tecnologias usadas para construção do aplicativo.

5.1 Sistema Integrado de Administração e Serviços Gerais

Instituído no Decreto nº 1.094/1994, no art. 7º, o SIASG tem como objetivo ser um instrumento de apoio nas atividades do SISG (Sistema de Serviços Gerais), através da informatização e operacionalização de suas atividades. Sua operacionalização se dá através de diversos módulos que constituem o SIASG. (SIASG, 2020)

5.2 Tipos de dados fornecidos pelo SIASG

Os dados disponibilizados pelo SIASG, disponível no site API de Compras Governamentais, de 2021, são: dados dos fornecedores, materiais, catálogo de serviços, licitações, contratos, compras sem licitação, plano anual de contratações.

5.3 Tecnologias usadas para construção do aplicativo

Para o desenvolvimento do aplicativo ComprasNet, foi utilizado algumas tecnologias e ferramentas que auxiliaram em algumas funcionalidades. Para conhecimento das mesmas, esta seção será dividida em tópicos, são eles: linguagem, arquitetura, design patterns, biblioteca e ferramentas.

- **Linguagem: *Kotlin***

De acordo com o site DevMedia, para a fase de desenvolvimento, foi escolhido o *Kotlin* como linguagem de programação, o *Kotlin* é uma linguagem que é executada na *JVM* (*Java Virtual Machine*), sendo assim, capaz de integrar em seu próprio código, a linguagem *Java*. O *Kotlin* também é uma linguagem do tipo estática, assim tornando mais seguro e prevenindo erros de compilação .

- **Arquitetura: MVVMi**

A arquitetura *MVVMi* foi escolhida para o projeto, é importante destacar a importância da *ViewModel* e *LiveData* que são dois componentes de extrema importância dessa arquitetura, a *ViewModel* sobrevive ao ciclo de vida da atividade e as mudanças de configuração, a *LiveData* é quem armazena os dados gerados pelo ciclo de vida, ou seja, tudo que acontece no aplicativo em segundo plano fica armazenado na *LiveData*. É importante destacar também a camada *interactor*, ela é responsável por toda lógica de negócio. (Medium,2020)

- **Design patterns: S.O.L.I.D**

O *Design pattern S.O.L.I.D* foi a escolha para manter o aplicativo de maneira escalável e evitar futuros problemas. Esse princípio de design é dividido em 5 partes. De acordo com o site *Medium*, são elas:

S - Single Responsibility (Responsabilidade Única): Uma classe deve possuir apenas uma responsabilidade. Pois a chance de surgirem bugs, aumenta quando se tem mais de uma responsabilidade.

O - Open-Closed (Aberto Fechado): Toda classe deve ser aberta para extensão, porém fechada para modificação.

L - Liskov Substitution (Substituição de Liskov): Toda classe filha deve herdar as características da classe pai, assim evitando futuros erros.

I - Interface Segregation (Segregação de Interface): Uma classe deve cumprir apenas o que é de sua função.

D - Dependency Inversion (Inversão de Dependência): Uma classe não deve ser fundada com a ferramenta que ela usa e sim com uma interface que permitirá que a ferramenta conecte à classe.

- **Biblioteca e ferramentas: Retrofit, Firebase, Android Room, Dialog Flow, API do Google Maps**

Retrofit: Para as requisições foi utilizado a biblioteca Retrofit, essa biblioteca é capaz de criar requisições e utilizar o aplicativo com um *Client HTTP (HyperText Transfer Protocol)*, porém é necessário que haja a deserialização, ou seja, transformar os dados obtidos da requisição em informação.

Firebase: Para controle de acesso de usuário, criação de novo usuário, recuperar senha, foi utilizado a biblioteca Firebase, essa biblioteca conecta a plataforma da Google que tem como objetivo auxiliar no desenvolvimento de aplicativos móveis, ajudando na criação, melhoria e expansão de aplicações.

Android Room: Para guardar dados locais, foi utilizado o *Android Room*, um banco de dados local, ela cria uma abstração das camadas de banco de dados *SQLite*, são representadas por três componentes, de acordo com site *Medium*, são:

- Entity: são entidades responsáveis por mapear as tabelas.
- DAO (Data Access Object): são as interfaces utilizadas para acessar os dados armazenados no banco.

- **Database:** é a representação da classe abstrata do Banco de Dados. Ele será responsável por fazer o controle do banco de dados.

Dialog Flow: Para criação de chatbot, foi utilizado o *Dialog Flow*, uma plataforma de processamento de linguagem natural da *Google*, ele funciona com agentes (bots) e através de entradas e saídas de informações.

Google Services: Na utilização de mapas, o *Google Services* foi escolhido para performar dentro do aplicativo *ComprasNet*, assim possibilitando que haja uma interação entre os órgãos e os contratos ativos.

6. Trabalhos Correlatos

Nesta seção será apresentado um trabalho que tem funcionalidades muito parecidas com a própria API, o *ContratosGov* que também tem o foco em gestão e fiscalização contratual.

6.1 *ContratosGov*

O *ContratosGov* é uma ferramenta gerencial que conecta de forma inteligente a atuação dos setores e agentes envolvidos na execução contratual, disponibilizando informações fundamentais e melhorando as condições operacionais da gestão e da fiscalização.

Sistema adquirido pela UFSCar (Universidade Federal de São Carlos) para gestão de contratos administrativos, onde são executadas algumas ações relativas à fiscalização, documentação e controle dos contratos UFSCar.

Fase 1

Após receber o contrato e realizar todos os procedimentos necessários, como reuniões preliminares e planejamento para uma fiscalização eficiente e eficaz. Onde o fiscal do contrato deverá conferir e caso necessário alterar os registros. Permitindo controle sobre as informações financeiras do contrato

Fase 2

Durante a execução dos contratos são realizados os ateste de pagamentos. Após seguir os procedimentos de verificação dos serviços/entregas realizadas, o fiscal deverá registrar no sistema *ContratosGov* o ateste da Nota Fiscal.

7. Bibliografia

GOV. **Portal de Compras do Governo Federal**. Publicado em 07 de agosto de 2020. Disponível em: <www.gov.br/compras/pt-br/aceso-a-informacao/institucional/siasg> Acesso em: 28 de maio de 2021.

Api de Compras Governamentais. **SIASG**. Atualizado até janeiro de 2021. Disponível em: <<http://compras.dados.gov.br/docs/home.html>> Acesso em: 28 de maio de 2021.

DevMedia. **Guia completo de Kotlin**. Disponível em: <www.devmedia.com.br/guia/linguagem-kotlin/40739> Acesso em: 29 de maio de 2021.

Medium. **Arquitetura limpa com MVVMi, componentes de arquitetura e RxJava.** Publicado por: Luke Simpson. Disponível em:
<medium.com/@thereallukesimpson/clean-architecture-with-mvvmi-architecture-components-rxjava-8c5093337b43> Acesso em: 29 de maio de 2021.

Medium. **Os princípios SOLID em imagens.** Publicado por: Ugonna Thelma. Disponível em:
<medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898> Acesso em: 30 de maio de 2021.

Medium. **Consumindo API REST com Retrofit + Kotlin no Android.** Publicado por: Alifyz F. Pires. Disponível em:
<medium.com/@alifyzfp/res/consumindo-api-rest-com-retrofit-kotlin-no-android-abba52820cc> Acesso em: 30 de maio de 2021.

Firebase. **Documentação.** Disponível em:
<<https://firebase.google.com/docs/android/setup?hl=pt-br>> Acesso em: 30 de maio de 2021.

Medium. **Introdução - Room Database Android.** Publicado por: Gabriel Prado. Disponível em:
<medium.com/android-dev-br/utilizando-room-database-no-android-fd76c2e6ccee> Acesso em: 30 de maio de 2021.

DialogFlow. **Documentação.** Disponível em:
<<https://cloud.google.com/dialogflow/docs/?hl=pt-br>> Acesso em: 30 de maio de 2021.

Proad. **Fiscalização de contratos.** Disponível em:
<<https://www.proad.ufscar.br/pt-br/media/arquivos/manual-contratosgov.pdf>> Acesso em: 29 de maio de 2021

Proad. **Manual contratos Gov.** Disponível em:
<<https://www.proad.ufscar.br/pt-br/servicos/contratos/gestao-de-contratos/fiscalizacao-de-contratos>> Acesso em: 29 de maio de 2021

Contratos Gov. **Gestão e fiscalização de contratos.** Disponível em:
<<https://www.contratosgov.com.br/Home?ReturnUrl=%2f>> .Acesso em: 29 de maio de 2021