

FACULDADE SENAI FATESG
PÓS GRADUAÇÃO EM PROJETO E
DESENVOLVIMENTO DE SOFTWARE PARA
DISPOSITIVOS MÓVEIS E INTERNET

MAURO VASCONCELOS DE PAULO

Aplicativo:R2P

Goiânia
2017

FACULDADE SENAI FATESG

PÓS GRADUAÇÃO EM PROJETO E
DESENVOLVIMENTO DE SOFTWARE PARA
DISPOSITIVOS MÓVEIS E INTERNET

**AUTORIZAÇÃO PARA PUBLICAÇÃO DE TRABALHO DE
CONCLUSÃO DE CURSO EM FORMATO ELETRÔNICO**

Na qualidade de titulares dos direitos de autores, **AUTORIZAMOS** a PÓS GRADUAÇÃO EM PROJETO E DESENVOLVIMENTO DE SOFTWARE PARA DISPOSITIVOS MÓVEIS E INTERNET da FACULDADE SENAI FATESG a reproduzir, inclusive em outro formato ou mídia e através de armazenamento permanente ou temporário, bem como a publicar na rede mundial de computadores (*Internet*) e na biblioteca virtual da FACULDADE SENAI FATESG, entendendo-se os termos “reproduzir” e “publicar” conforme definições dos incisos VI e I, respectivamente, do artigo 5º da Lei nº 9610/98 de 10/02/1998, a obra abaixo especificada, sem que me seja devido pagamento a título de direitos autorais, desde que a reprodução e/ou publicação tenham a finalidade exclusiva de uso por quem a consulta, e a título de divulgação da produção acadêmica gerada pela FACULDADE SENAI FATESG, a partir desta data.

Título: Aplicativo:R2P

Autor: Mauro Vasconcelos de Paulo

Goiânia, 26 de Maio de 2017.

Mauro Vasconcelos de Paulo – Autor

Edjalma Queiroz da Silva – Orientador

MAURO VASCONCELOS DE PAULO

Aplicativo:R2P

Trabalho de Conclusão de Curso apresentado a PÓS GRADUAÇÃO EM PROJETO E DESENVOLVIMENTO DE SOFTWARE PARA DISPOSITIVOS MÓVEIS E INTERNET da FACULDADE SENAI FATESG, como requisito parcial para obtenção do título de Especialista em Projeto e Desenvolvimento de Software para Dispositivos Móveis e Internet.

Orientador: Prof. Edjalma Queiroz da Silva

Goiânia
2017

MAURO VASCONCELOS DE PAULO

Aplicativo:R2P

Trabalho de Conclusão apresentado à Coordenação do Curso de PÓS GRADUAÇÃO EM PROJETO E DESENVOLVIMENTO DE SOFTWARE PARA DISPOSITIVOS MÓVEIS E INTERNET da FACULDADE SENAI FATESG como requisito parcial para obtenção do título de Especialista em Projeto e Desenvolvimento de Software para Dispositivos Móveis e Internet, aprovada em 26 de Maio de 2017, pela Banca Examinadora constituída pelos professores:

Prof. Edjalma Queiroz da Silva
FACULDADE SENAI FATESG
Presidente da Banca

Prof. Bruno Urbano Rodrigues
FACULDADE SENAI FATESG

Prof. Daniel Correa da Silva
FACULDADE SENAI FATESG

Agradecimentos

Algumas pessoas foram essenciais para a realização desse trabalho. Colaborando diretamente ou indiretamente, muita gente teve um papel fundamental para que os objetivos fossem concretizados. Com o trabalho pronto, chegou a hora de agradecer. Ao nosso orientador Edjalma Queiroz da Silva, pela oportunidade, pelos ensinamentos fornecidos para que o projeto pudesse ser realizada e por acreditar e nos mostrar que a documentação de um projeto é um desafio extremamente necessário para se obter bons resultados. A Faculdade de Tecnologia e Desenvolvimento Gerencial Senai, pela excelente estrutura, suporte e ambiente oferecidos aos alunos, e a todos os professores, principalmente aos que nós tivemos a oportunidade de conhecer e aprender tantas coisas. Aos nossos amigos que contribuíram diretamente para a realização deste projeto e foram muito importantes durante o andamento do trabalho. Aos nossos familiares, que acreditaram na realização desse trabalho.

A todos, o nosso “muito obrigado”!

Resumo

de Paulo, M. V. **Aplicativo:R2P** . Goiânia, 2017. 38p. Trabalho de Conclusão de Curso. PÓS GRADUAÇÃO EM PROJETO E DESENVOLVIMENTO DE SOFTWARE PARA DISPOSITIVOS MÓVEIS E INTERNET, FACULDADE SENAI FATESG.

O EuAjudo aCiencia será um aplicativo móvel com a finalidade de recrutamento de pessoas interessadas na participação dos estudos da empresa ICF. Contará com uma tela de cadastro de pessoa, tela de edição de dados do cadastro, tela de visualização de mensagens referente aos estudos e tela de contato com a empresa ICF. A principal função do aplicativo é de informar os estudos novos da empresa e poder informar as pessoas cadastradas destes estudos. Inicialmente terá apenas a tela de cadastro, edição e visualizar mensagens.

Palavras-chave

Aluno, Universidade, Provas, Gestão.

Abstract

de Paulo, M. V. . Goiânia, 2017. 38p. Trabalho de Conclusão de Curso. PÓS GRADUAÇÃO EM PROJETO E DESENVOLVIMENTO DE SOFTWARE PARA DISPOSITIVOS MÓVEIS E INTERNET, FACULDADE SENAI FATESG.

Some people were essential to this work. Collaborating Directly or indirectly, many people played a Objectives. With work ready, it's time to thank. Our advisor Edjalma Queiroz da Silva, for the opportunity, for the The lessons learned so that the project could be carried out and Believe and show us that the documentation of a project is An extremely necessary challenge to achieve good results. The Faculty of Technology and Management Development Senai, for the excellent Structure, support and environment offered to students, and to all teachers, Especially those we had the opportunity to know and learn so many stuff. To our friends who contributed directly to the realization of this project And were very important during the course of work. To our relatives, who believed in doing this work

To all, our “ thank you ”!

Keywords

Student, University, Evidence, Management

Sumário

Lista de Figuras	9
Lista de Tabelas	10
Lista de Códigos de Programas	11
1 Introdução	13
1.1 Finalidade	13
1.2 Objetivo Geral	14
2 Escopo do Projeto	15
2.1 Situação Atual	15
2.1.1 Motivação	15
2.1.2 Problema	15
2.1.3 Justificativa	16
2.2 Objetivos	16
2.2.1 Objetivos Gerais	16
2.2.2 Objetivos Específicos	16
3 Fundamentação Teórica	17
3.1 Android	17
3.2 Persistência de dados	18
3.2.1 Web service	19
3.3 Sistemas de Objetos Distribuídos	20
3.4 Aplicações móveis	22
4 Estudo de Caso - Aplicativo R2P	24
4.1 Requisitos	24
4.1.1 Requisitos de Funcionais	24
4.1.2 Requisitos Não Funcionais	25
4.2 Diagramas da UML	25
4.2.1 Diagrama de Classes	26
4.2.2 Diagrama de Caso de Uso	27
4.2.3 Diagrama de Sequência	29
4.2.4 Diagrama de Componentes	29
4.2.5 Arquitetura de Funcionamento do Aplicativo	30
4.3 Prototipação	31
4.3.1 Aplicativo R2P	32

5	Considerações Finais	36
5.1	Conclusão	36
5.1.1	Contribuição	36
5.2	Trabalhos Futuros	36
	Referências Bibliográficas	38

Lista de Figuras

3.1	Dados coletados durante um período de 7 dias encerrado em 1º de agosto de 2016. Todas as versões com menos de 0,1 por cento de distribuição não foram exibidas.	20
3.2	<i>Pattern</i> de notificação	22
3.3	Característica de mobilidade	22
4.1	Diagrama de Classe R2P	27
4.2	Caso de Uso - R2P	28
4.3	Diagrama de Sequencia Visualizar Notificação	29
4.4	Diagrama de Componetes	30
4.5	Aquitetura de funcionamento do Aplicativo	31
4.6	Tela entrar no Aplicativo R2P	32
4.7	Tela De Cadastro Participante de Pesquisa	33
4.8	Tela De Edição Participante de Pesquisa	34
4.9	Tela De Visualização de notificações	35

Lista de Tabelas

4.1	Lista de Requisitos Funcionais	25
4.2	Lista de Requisitos Não Funcionais	25

Lista de Códigos de Programas

Lista de Códigos

Introdução

Um estudo clínico é qualquer investigação em participantes de pesquisa, com objetivo de descobrir ou verificar os efeitos da medicação do estudo. Estudando sua forma de agir no organismo, comprovando sua eficácia (se funciona) e/ou segurança. O estudo é considerado uma pesquisa porque o medicamento ainda não foi aprovado pela Autoridade Competente, a Agência Nacional de Vigilância (ANVISA), para a venda.

Pode participar pessoas que possuam a patologia (doença, alteração) do estudo e que apresentem os critérios necessários de acordo com o médico do estudo (idade, exames, entre outros).

A pessoa que deseja participar deverá comparecer ao centro para conversar com o médico e ler um documento que explica tudo sobre o estudo clínico (TCLE). Caso você concorde em participar e esteja apto (aprovado nos critérios do estudo), você receberá o medicamento do estudo, terá o acompanhamento médico, realizará exames e outros procedimentos específicos de cada estudo.

Fundado em 2002, o Instituto de Ciências Farmacêuticas (ICF) foi o primeiro laboratório de bioequivalência e biodisponibilidade certificado pela ANVISA e o primeiro brasileiro pelo Instituto de *Salud Publica del Chile* (ISP). É constantemente auditado por empresas de todo mundo.

Atualmente é um dos maiores centros de pesquisas farmacêuticas do Brasil e é referência na América Latina para indústrias de vários países que operam e investem na região.

Com mais de 500 estudos de bioequivalência realizados, o ICF é parte da história do medicamento genérico no Brasil. Uma das maiores infraestruturas de internação do país, alta tecnologia, portfólio diversificado e certificado, duas centenas de métodos Bioanalíticos incluso os de alta complexidade como hormônios e *sprays*.

1.1 Finalidade

Com o recrutamento de participantes de pesquisa a empresa tem muitos gastos com pacotes de torpedos SMS comprados para notificar pessoas interessadas já

cadastradas no sistema corporativo VWEBBIO. E ao mesmo tempo dificuldades de fazer com que a mensagem enviada chegue até os participantes de pesquisa.

Diante deste cenário, propomos a construção de um software aplicativo para a facilitar o recrutamento de participantes de pesquisa para a empresa ICF. Que facilitara a divulgação de informações pertinentes aos estudos clínicos com suas respectivas restrições, fazendo com que os próprios participantes de pesquisa interessados possa demonstrar o interesse a participação no estudo, ou seja entrando em contato com o ICF assim não sendo necessário o ICF entrar em contanto com os mesmos.

Faremos o uso de um cadastro de participantes de pesquisa corporativo, mantido pela ICF, com o nome de VWEBBIO, que já é utilizado atualmente. Essa integração se dará através de consultas ao banco de dados já existentes no sistema através de web service. Os novos participantes serão pré-cadastrados através do aplicativo e os que já estão cadastrados não precisará de cadastro no aplicativos já irão direto para a tela de edição de dados do aplicativo ou para visualização das informações de estudos disponíveis.

1.2 Objetivo Geral

O projeto visa desenvolver uma aplicativo em Android para pré-cadastro de pessoas interessadas em participarem dos estudos realizados no ICF e para visualizações de estudos disponíveis para os interessados participarem.

Através de um celular smartphone com acesso a internet e com o sistema operacional Android os participantes de pesquisa poderão utilizar o aplicativo.

Escopo do Projeto

Este trabalho define a modelagem do aplicativo de Recrutamento de Participantes de Pesquisa (R2P), suas telas principais, requisitos e tecnologias utilizadas. O aplicativo RPP irá falar diretamente com o banco de dados da empresa e como o sistema de gerenciamento de participantes de pesquisa.

2.1 Situação Atual

Atualmente o recrutamento de participantes de pesquisa na empresa ICF é feito através de ligações feitas por recepcionistas. Aquisição de pacotes de SMS para envio de mensagens aos participantes de pesquisa. Atualização de uma página no facebook com a mesma mensagem enviada por mensagens. Atualização do site com esta mensagem. Os cadastros destes participantes de pesquisa são efetivados em um sistema de interno da empresa VWEBBIO.

2.1.1 Motivação

A motivação deste trabalho é facilitar o recrutamento de participantes de pesquisa e divulgação de novos estudos através de mensagens pelo aplicativo com o perfil desejado. Escolhido o desenvolvimento de um aplicativo devido ao grande número de pessoas que serão atingidas.

2.1.2 Problema

O principal problema é a sobrecarga das recepcionistas fazendo ligações para pessoas cadastrada na base de dados para tentar encontrar participantes de pesquisa interessados em participar dos estudos. E o custo da compra de pacotes de SMS são caros para empresa ICF.

2.1.3 Justificativa

Desenvolver um aplicativo simples com boa aceitabilidade do publico alvo para aumentar o nosso banco de dados de pessoas interessadas em participação dos estudos da empresa ICF. Facilitar o contato com estes participantes de pesquisa já cadastrado na nossa base de dados.

2.2 Objetivos

Desenvolver um aplicativo para facilitar o recrutamento de participantes de pesquisa.

2.2.1 Objetivos Gerais

O projeto tem como objetivo redução de gastos e de mão de obra da empresa ICF com recrutamentos de partivipantes de pesquisa.

2.2.2 Objetivos Específicos

Desenvolver um aplicativo movel Android, para economia e facilidade no recrutamento de pessoas interessadas em participar de estudo na empresa ICF.

Fundamentação Teórica

Neste capítulo, será apresentado as tecnologias utilizadas no desenvolvimento do aplicativo R2P. Essas tecnologias apoiam o programador e facilita o desenvolvimento através do uso de API's e frameworks já desenvolvidos.

3.1 Android

O Android é uma plataforma composta por um sistema operacional, *middlewares* e um conjunto de aplicativos principais, como os contatos, o navegador de internet e o telefone propriamente dito. Além disso, existe o Android SDK, que é um conjunto de ferramentas e APIs para o desenvolvimento de aplicativos para a plataforma, utilizando a linguagem Java [[MONTEIRO](#)].

Baseado no Linux, o sistema operacional Android teve seu desenvolvimento iniciado em 2003 pela empresa Android Inc. Em 2005, esta foi adquirida pelo Google, que hoje lidera o desenvolvimento do Android.[[MONTEIRO](#)].

A história da versão do sistema operacional móvel Android começou com o lançamento da versão alpha em novembro de 2007. A segunda versão comercial, o Android 1.0, foi lançado em setembro de 2008. O sistema operacional recebeu uma série de atualizações desde o seu lançamento original. Essas atualizações geralmente corrigem erros e adicionam novas funcionalidades.

Abaixo segue as principais versões do Android:

Android 2.2 - Froyo: Foi lançado em 20 de maio de 2010, sendo 5 vezes mais rápido que as versões anteriores, navegador web nativo, possibilidade de armazenar aplicativos no cartão SD, Desenvolvedores podem criar apps que aprimoram a segurança do aparelho, como as telas de bloqueio, etc.

Android 2.3 - Gingerbread: Foi lançado em 06 de dezembro de 2010, fornecendo suporte a aparelhos com câmeras frontais, suporte a sensores de movimentos, teclado virtual melhorado e chamadas pela internet.

Android 4.0 - Ice Cream Sandwich : Foi lançado em 19 de outubro de 2011, com destaque a gabe Cohen, da Google, declarou que o Android 4.0 era "teoricamente compatível" com qualquer dispositivo Android 2.3.x em produção na época.

Android 4.1/4.2/4.3 - Jelly Bean : Foi lançado 09 de julho de 2012, Jelly Bean foi uma atualização incremental com o objetivo principal de melhorar a funcionalidade e desempenho da interface do usuário

Android 4.4 - KitKat : Foi lançado 15 de janeiro de 2014. Após recente acordo com a Nestlé, o novo sistema Android leva o nome de um dos doces mais famosos da marca.

Android 5.0/5.1 - Lollipop : Foi lançado 03 de novembro de 2014, Com esta versão podemos destacar presença do Material Design, que deixa apps mais chamativos e bonitos, novo modo de economia de bateria, sistema muito limpo e fluido e etc.

Android 6.0 - Marshmallow : Foi lançado em 05 de outubro de 2015, as principais melhorias que podemos destacar são, melhorias para compartilhamento de conteúdo entre aplicativos, suporte para autorização de permissões de aplicativos sob demanda, modo de tela 4K para apps, adaptação de memória externa (SD cards) como parte da memória interna, etc.

Android 7.0 Nougat : Foi lançado 22 de agosto de 2016 é a versão mais atual do antroid e com ele veio, função de multi-janela, novo menu de notificações, informações de emergência, biblioteca OpenJDK, em vez de JAVA e etc.

Na Figura 3.1 é possível observar as principais versões do Sistema Operacional Android até 1ª de agosto de 2016.

A evolução do sistema operacional Android acompanha a evolução do hardware, dessa forma, o SO do Android foi Desenvolvido especialmente para dispositivos móveis, como aparelhos celulares e *Tablets*, e agora também suportando dispositivos "vestíveis" *wearables*, como relógios, pulseiras, óculos, dentre outros.

3.2 Persistência de dados

Persistência de dados significa armazená-los em algum local de modo que possamos recuperá-los posteriormente.

No Android existem várias formas de fazer isso. E esta forma vai depender da necessidades específicas.

Abaixo segue algumas formas de armazenamento de dados no Android:

1. **SharedPreferences :** Armazenar dados particulares primitivos em pares chave-valor;
2. **Internal Storage :** Armazenar dados privados na memória do dispositivo. (com persistência de objetos);

3. **External Storage** : Armazenar dados públicos sobre o armazenamento externo compartilhado;
4. **TeSQLite Databasesxto**: Armazenar dados estruturados em um banco de dados privado.
5. **Network Connection**: Armazena dados na web, ou seja em um servidor web service em rede.

O Android fornece uma maneira para que você possa expor seus dados privados para outras aplicações com um *content provider*. Um *content provider* é um componente opcional que permite o acesso de leitura e escrita aos dados de sua aplicação, sujeito às restrições impostas por você.

No aplicativo R2P foi escolhido o **Network Connection** conexão em rede definida de web *service*.

Web service é uma tecnologia de acesso a dados que permite a troca de informações estruturada a partir de mensagens XML e SOAP. Esta tecnologia permite que diferentes empresas, mesmo utilizando tecnologias e plataformas distintas, conectem-se de maneira padrão e executem procedimentos remotos através da utilização do protocolo padrão da internet - HTTP.

3.2.1 Web service

O uso do Web service é muito interessante, pois permite acesso a rotinas de várias empresas e com informações de tempo real visto que estas podem mudar o tempo todo.

Os web *services* podem ser implementados de muitas maneiras diferentes, cada uma com vantagens e desvantagens próprias da solução técnica adotada. Em sistemas desenvolvidos se utilizando a linguagem Java existem duas soluções, que na documentação da Oracle são denominadas como *Big Web Services* e *RESTful Web Services*. Tanto os web *Services Big* quanto os *RESTful* fazem parte da API Java para XML (Java API for XML - JAX), que foi introduzida ao Java SE na versão 5.

Os *Web Services RESTful* Transferência de Estado Representacional *Representational State Transfer* são mais adequados para a utilização em cenários mais básicos, e também são melhor adaptados ao uso do protocolo HTTP do que os serviços *SOAP*. Os serviços *RESTful* são mais leves, o que significa que podem ser desenvolvidos com menor esforço, tornando-os mais fáceis de serem adotados como parte da implementação de um sistema. São mais indicados para a integração de sistemas através da internet, conferindo às soluções qualidades como escalabilidade, simplicidade e fraco acoplamento entre as partes.

No para o aplicativo R2P foi escolhido o web *services RESTful*, conforme arquitetura mostrado na figura ??.

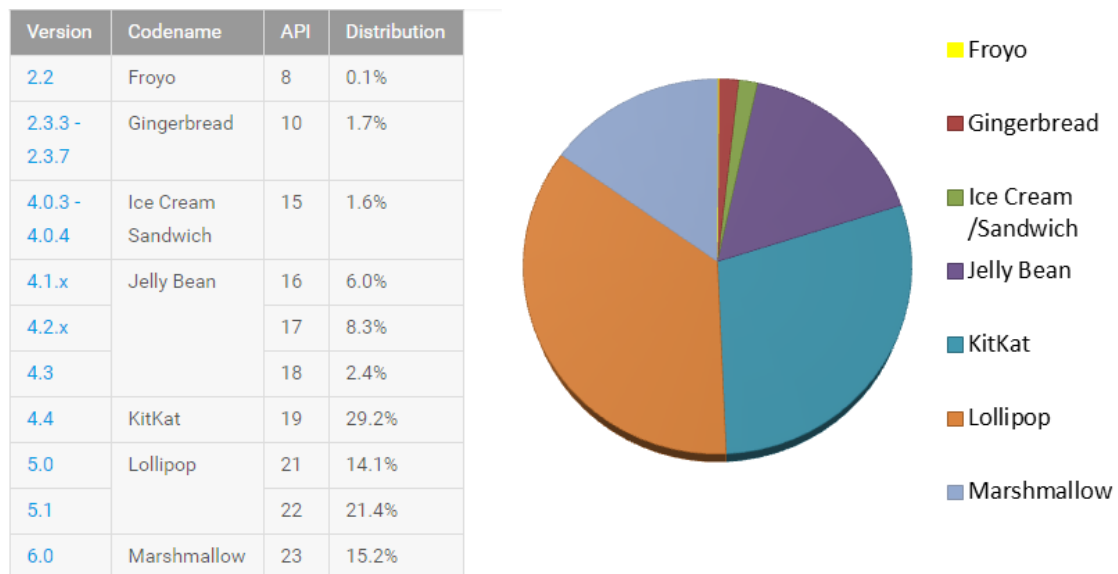


Figura 3.1: *Dados coletados durante um período de 7 dias encerrado em 1º de agosto de 2016. Todas as versões com menos de 0,1 por cento de distribuição não foram exibidas.*

3.3 Sistemas de Objetos Distribuídos

Várias definições de sistemas distribuídos foram dadas na literatura, nenhum deles satisfatório, e nenhum deles de acordo com qualquer um dos outros. Para nossos propósitos é suficiente dar uma caracterização frouxa. Um sistema distribuído é uma coleção de computadores para os seus utilizadores como um único sistema coerente [Tanenbaum e Steen 2006].

Esta definição tem vários aspectos importantes. A primeira é que um consiste em componentes (isto é, computadores) que são autónomos. Um segundo É que os usuários (sejam eles pessoas ou programas) acham que estão lidando com sistemas. Isto significa que, de uma forma ou de outra, os componentes autónomos Precisam colaborar [Tanenbaum e Steen 2006].

Como estabelecer esta colaboração está no centro dos Sistemas distribuídos. Note-se que não são feitas suposições sobre o tipo de computadores. Em princípio, mesmo dentro de um único sistema, eles podem variar de Alto desempenho para pequenos nos em redes de sensores. Da mesma forma, Nenhuma suposição é feita na maneira que os computadores são interconectados [Tanenbaum e Steen 2006].

Em vez de ir mais longe com as definições, talvez seja mais útil em características importantes de sistemas distribuídos. Uma característica importante e que as diferenças

entre os vários computadores e as formas como eles se comunicam são na sua maioria escondidos dos usuários. O mesmo vale para a organização do sistema distribuído. Outra característica importante é que usuários e aplicativos podem interagir com um sistema distribuído de forma maneira uniforme, independentemente de onde e quando a interação ocorre [Tanenbaum e Steen 2006].

Em princípio, os sistemas distribuídos também devem ser relativamente fáceis de escala. Esta característica é uma consequência direta de ter computadores independentes, Mas ao mesmo tempo, escondendo como esses computadores realmente participam no Sistema como um todo. Um sistema distribuído estará normalmente disponível continuamente, Embora talvez algumas partes possam estar temporariamente fora de ordem. Usuários e aplicativos não deve notar que as peças estão sendo substituídas ou fixadas, ou que Adicionado para servir mais usuários ou aplicativos [Tanenbaum e Steen 2006].

Um sistema de objetos distribuídos permite que os componentes de um sistema estejam espalhados em uma rede heterogênea e mesmo assim ainda haja comunicação entre eles.

Segue abaixo algumas arquiteturas de objetos distribuídos:

1. ***Distributed Component Object Model (DCOM)***: é uma versão distribuída do modelo COM da Microsoft que promove comunicação entre componentes distribuídos em uma rede de computadores. Esses componentes podem se comunicar independente da linguagem em que foram desenvolvidos e até linguagens de outros fabricantes;
2. ***Java Remote Method Invocation and Specification (Java/RMI)***: é uma arquitetura de sistemas distribuídos para linguagem de programação Java, que tem o objetivo de dar suporte à invocação remota transparente de métodos em objetos distribuídos em diferentes máquinas virtuais java;
3. ***Common Object Request Broker Architecture (CORBA)***: tecnologia desenvolvida pela OMG (*Object Management Group*), onde seus objetivos gerais são permitir a interoperabilidade de objetos de software rodando sobre sistemas distribuídos e permitir sua composição em aplicações. Para padronizar a aparência externa dos objetos é utilizada uma IDL (*Interface Definition Language*). A IDL define as interfaces contratuais de um componente com seus potenciais clientes.

A figura 3.2 mostra o *Design Pattern* proposto por Mowbray (1997). Este é um padrão de notificação de eventos do CORBA e é utilizado para comunicação entre objetos em um ambiente distribuído. Os eventos são enviados ou recebidos através de um canal compartilhado. Com dois modelos de eventos o *push model* e o *pull model*. Este padrão pode ser utilizado quando é necessário ter um gerenciamento de eventos na aplicação e informar todos os objetos distribuídos sobre um evento ocorrido.

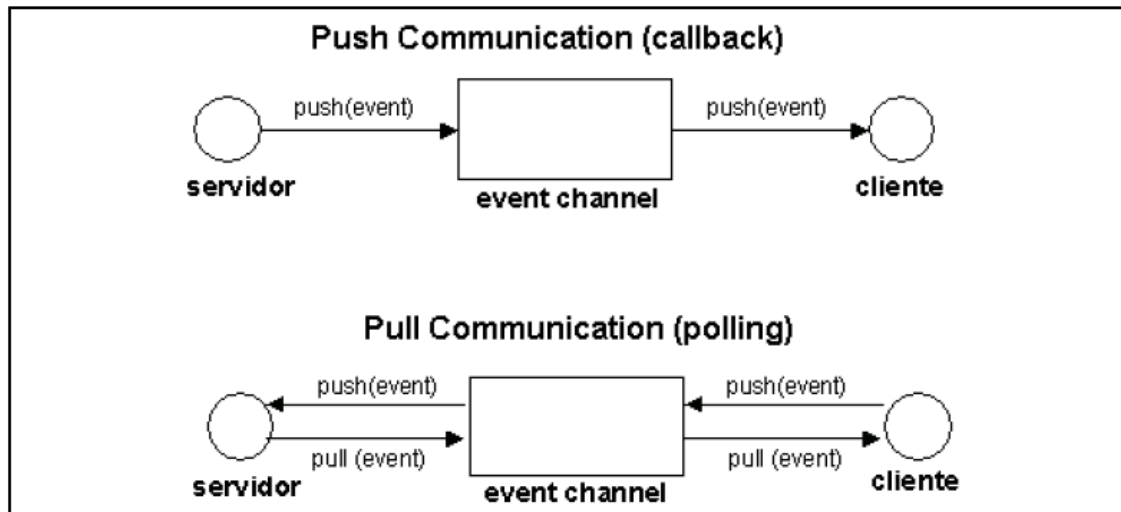


Figura 3.2: Pattern de notificação

3.4 Aplicações móveis

Os dispositivos móveis oferecem grande comodidade. Essa comodidade pode ser expressada através de alguns recursos, tais como, acesso a internet, aplicativos corporativos acessíveis, baixo custo, dentre outros. Diante desta realidade a demanda por aplicativos é cada vez maior, seja aplicativos para exercer uma Venda ou para prover algum tipo de Serviço.

Mobilidade pode ser definida como a capacidade de poder se deslocar ou ser deslocado facilmente. No contexto da computação móvel, mobilidade se refere ao uso pelas pessoas de dispositivos móveis portáteis funcionalmente poderosos que ofereçam a capacidade de realizar facilmente um conjunto de funções de aplicação, sendo também capazes de conectar-se, obter dados e fornecê-los a outros usuários, aplicações e sistemas [Lee, Schneider e Schell 2005].

Para fazer isso, um dispositivo móvel deve possuir determinadas características. por exemplo, deve ser portátil e o usuário se capaz de transportá-los com relativa facilidade, características como peso e tamanho são essenciais. Um dispositivo móvel também tem de ser altamente utilizável, funcional e permitir fácil conectividade e comunicação com outros dispositivos como podemos ver na Figura 3.3 [Lee, Schneider e Schell 2005].

As arquiteturas de aplicações móveis podem ser modeladas convenientemente de acordo com cada propósito e ou problema a ser resolvido por elas. Os clientes podem estar sempre conectados, parcialmente conectados ou nunca conectados a um servidor. O código em um cliente ou servidor pode estar em camadas. Um cliente pode ter de zero até três camadas, enquanto o servidor tem de uma a três camadas.

Dependendo de cada contexto, pode não ser possível implementar todas as

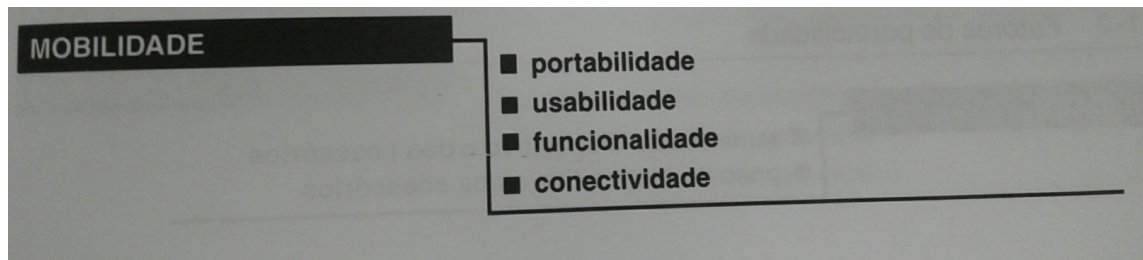


Figura 3.3: *Característica de mobilidade*

melhores práticas de desenvolvimento de Software. Contudo, é necessário perceber que muitas destas melhores práticas arquiteturais de aplicação móvel obedecem a vários outros princípios já conhecidos, como por exemplo:

- Requisitos: projeto arquitetônico que abrange os requisitos funcionais, de negócios, não funcionais e de usuário.
- Independência de tecnologia: Alcançar o máximo de independência de um dispositivo em relação a uma tecnologia específica.
- Alto desempenho e disponibilidade: desempenho esperado durante períodos normais e de pico de demanda de recursos.
- Escalabilidade: capacidade de permitir escalonamento tanto horizontal quanto vertical, sem afetar quaisquer outros recursos e em um cenário ideal, sem demandar codificações extras.
- Requisitos de sistema de usuário: Levar em consideração o escopo completo de usuários, incluindo aqueles com sistema de alto e baixo desempenhos.

Dessa forma, observamos que existem diversos tipos de arquiteturas a serem adotadas. Cada uma dessas arquiteturas servem a algum propósito e na maioria das vezes são utilizadas de maneira combinada entre si. A combinação desses modelos de arquitetura estão vinculados as boas práticas e o que se deve levar em consideração a criar o projeto para atender a um ou mais objetivos.

Estudo de Caso - Aplicativo R2P

Neste capítulo será apresentado o Estudo de Caso do aplicativo R2P, seus respectivos Requisitos, Diagramas da UML e Prototipação.

4.1 Requisitos

Os requisitos de um sistema(aplicativo) são descrições dos serviços fornecidos pelo sistema e as suas restrições operacionais. Esses requisitos refletem as necessidades dos clientes de um sistema que ajudam a resolver algum problema, por exemplo controlar um dispositivo, enviar um pedido ou encontrar informações [[Sommerville et al. 2008](#)].

São definições e especificações de um *software* após a análise do sistema e a delimitação do escopo do software. Os requisitos podem ser funcionais e não funcionais, e eles fazem parte de todo o ciclo de vida de um projeto de *software*.

Portanto, um requisito é uma função que o sistema proposto deve fazer ou uma restrição no desenvolvimento do sistema.

Existem dois tipos de classificação de requisitos que serão apresentados neste trabalho, são eles: Requisitos Funcionais (RF) e Requisitos Não-Funcionais (RNF).

4.1.1 Requisitos de Funcionais

Os requisitos funcionais de um sistema descrevem o que o sistema deve fazer. Esses requisitos dependem do tipo de *software* que está sendo desenvolvido, dos usuários a que o *software* se destina e da abordagem geral considerada pela organização ao redigir os requisitos. Quando expressos como requisitos de usuário, eles são geralmente descritos de forma bastante abstrata. No entanto, os requisitos funcionais descrevem a função do sistema detalhadamente, suas entradas e saídas, exceções e etc [[Sommerville et al. 2008](#)].

4.1.2 Requisitos Não Funcionais

Os requisitos não funcionais, como o nome sugere, são aqueles não diretamente relacionados às funções específicas fornecidas pelo sistema. Eles podem estar relacionados às propriedades emergentes dos sistema, como confiabilidade, tempo de resposta e espaço de armazenamento. Como alternativa, eles podem definir restrições, como a capacidade dos dispositivos de E/S (entrada/saída) e as representações de dados usadas nas interfaces do sistemas [Sommerville et al. 2008].

Dessa forma, foram definidos os seguintes Requisitos Não-Funcionais do sistema proposto, que podem ser observados na Tabela 4.2.

A Tabela 4.1 descreve a lista de requisitos funcionais da sistema proposto.

Tabela 4.1: *Lista de Requisitos Funcionais*

RF01	Tela Inicial do sistema	O aplicativo deverá ter uma tela inicial que os voluntários irão informar o seu CPF e que serão persistidos na base do sistema cooperativo, que irá validar se o CPF informado está cadastrado ou não na base de dados.
RF02	Mater Pessoas	O aplicativo deve permitir o cadastro dos voluntários, Informando o seu cpf, nome, sexo, data nascimento, e-mail e telefone celular. Para poder visualizar as mensagens de estudos disponíveis.
RF03	Apresentação de mensagens	O aplicativo deve mostrar as mensagens de estudos cadastrados no sistema cooperativo.

Tabela 4.2: *Lista de Requisitos Não Funcionais*

RNF01	Sistema Operacional	O sistema operacional deve ser Android. O sistema operacional deverá ter a versão de 4.0 ou superiores.
RNF02	Conexão com a internet	O aplicativo deve ter conexão com a internet para salvar e consultar os dados.
RNF03	Integração web service	O aplicativo terá uma integração com o um web service. Esta integração irá ser responsável pela integração entre aplicativo e banco de dados.

4.2 Diagramas da UML

UML é uma linguagem que define uma série de artefatos que nos ajuda na tarefa de modelar e documentar os sistemas orientados a objetos que desenvolvemos. A maioria dos problemas encontrados em sistemas orientados a objetos tem sua origem na construção do modelo, no desenho do sistema. A UML não é uma metodologia de desenvolvimento, o que significa que ela não diz para você o que fazer primeiro e em seguida ou como projetar seu sistema, mas ela lhe auxilia a visualizar seu desenho e a comunicação entre objetos.[D'Souza 1998]

A *Unified Modeling Language* - UML é uma especificação da *Object Management Group* - OMG (OMG, 1997 - 2011). É uma linguagem gráfica de modelagem para visualizar, especificar, construir e documentar os artefatos de sistemas

de objetos distribuídos (UML, 2011). A UML possui treze modelos gráficos que estão divididos em duas categorias, os diagramas de aplicações estáticas que representam a estrutura e os diagramas de comportamentos, no entanto dentro desta última categoria, existe uma subcategoria que compõe os diagramas de interação [Silva, Freire e Callahan].

A categoria de diagramas de Estrutura inclui diagrama de classe, diagrama de objeto, diagrama de componentes, diagrama de estrutura composta, diagrama de pacote e diagrama de utilização [Silva, Freire e Callahan]. Os diagramas de Comportamento são: diagrama de caso de uso, diagrama de máquina de estados e diagrama de atividades. Em sua subcategoria Interação estão inclusos os diagramas de sequência, comunicação, visão geral de interação e por ultimo, porém não menos importante o de temporização [Silva, Freire e Callahan].

Os diagramas UML abordados neste projeto são os de: Caso de Uso, Sequência, e Classe. Na imagem a seguir podemos visualizar as treze representações de modelos de UML.

Para construção dos diagramas da UML utilizamos o Asth ¹.

4.2.1 Diagrama de Classes

É uma modelagem muito útil para o desenvolvimento de sistemas, pois define todas as classes que o sistema necessita possuir e é a base para a construção dos diagramas de comunicação, sequência e estados.

O diagrama de classes representa a estrutura do sistema, recorrendo ao conceito de classe e suas relações. O modelo de classes resulta de um processo de abstração onde são identificados os objetos relevantes do sistema em estudo.

Cada classe é descrita através do seu nome, identificação de todos os seus atributos e identificação de todas as operações que traduzem o seu comportamento. O diagrama de classes mostra como cada classe se relaciona com as outras, tendo como objetivo, a satisfação dos requisitos funcionais definidos para o sistema em estudo. Cada classe representa uma tabela do banco de dados.

A Figura 4.1 descreve o diagrama de Classe do aplicativo. Observe que o objeto Notificação, se comporta como resultado do pattern descrito na Seção 3.2.

¹ <http://astah.net/>

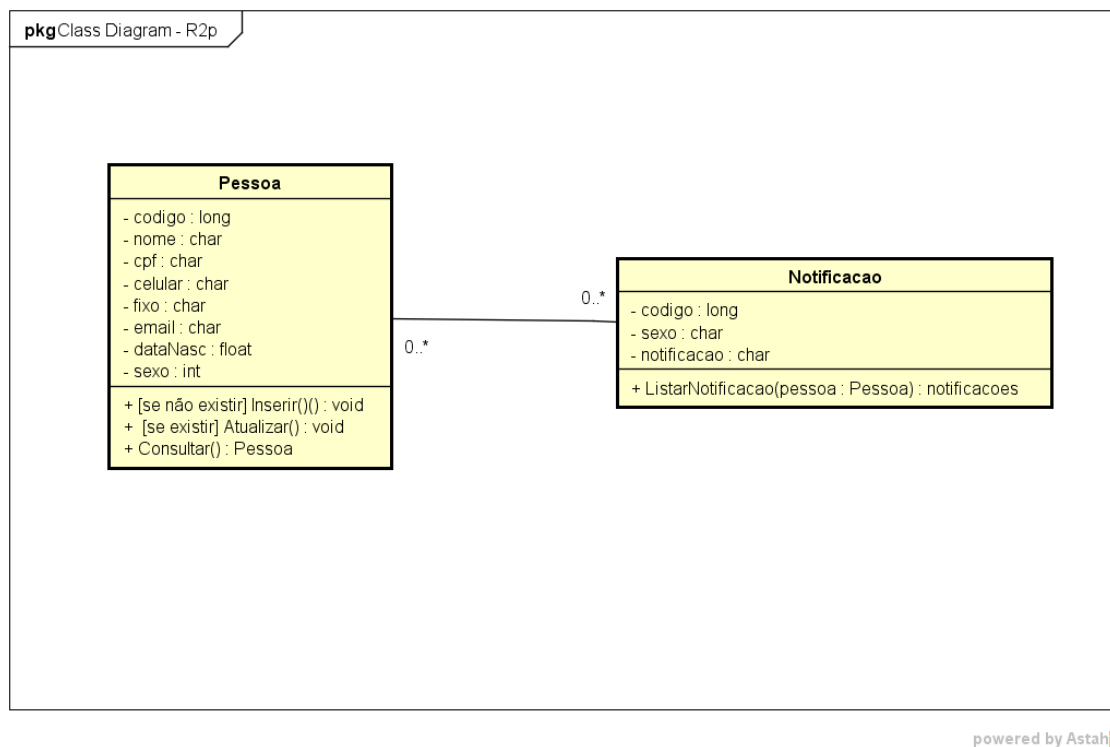


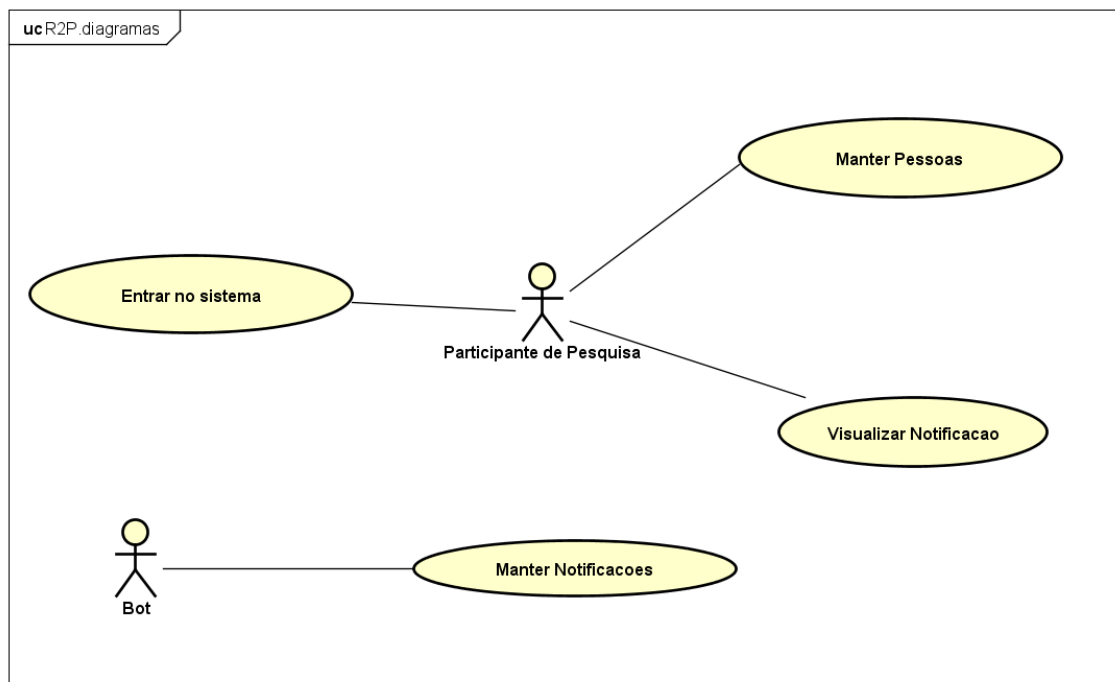
Figura 4.1: Diagrama de Classe R2P

4.2.2 Diagrama de Caso de Uso

O diagrama de casos de uso é um diagrama da UML cujo objetivo é representar um requisito do sistema que será automatizado. Considere como requisito uma necessidade do sistema. O diagrama ilustrado abaixo demonstra as principais funcionalidade e os atores do sistema.

Na figura 4.2 é mostrado o diagrama de caso de uso do sistema que contém Coordenador como atores de casos de usos. Os atores representam os papéis que os usuários de casos de uso desempenham quando interagem com sistema.

Abaixo ilustração do Diagrama de Caso de Uso do sistema R2P:



powered by Astah

Figura 4.2: *Caso de Uso - R2P*

O diagrama de caso de uso do aplicativo R2P conta com dois atores. Um ator principal que se chama participante de pesquisa que será responsável por realizar o seu cadastro (Manter Pessoas) e visualizar notificações. Já o ator Bot será responsável por cadastrar as notificações dentro do aplicativo.

Especificação textual

O caso de uso se inicia com o Bot que será um sistema interno da empresa ICF que será responsável por manter notificação, não iremos aprofundar neste fluxo.

Cenário Principal

O caso de uso envolvendo o ator participante de pesquisa inicia apos ele baixar e instalar o aplicativo em seu *smartphone*, no qual o participante de pesquisa irá informar o seu CPF para entrar no aplicativo.

Alternativo 1 (Participante de Pesquisa cadastrado no sistema corporativo) O participante de pesquisa pode visualizar o seus dados cadastrais e realizar a sua atualização de dados.

Alternativo 2 (Participante de Pesquisa visualizar notificação) O participante de pesquisa pode visualizar a lista de notificações.

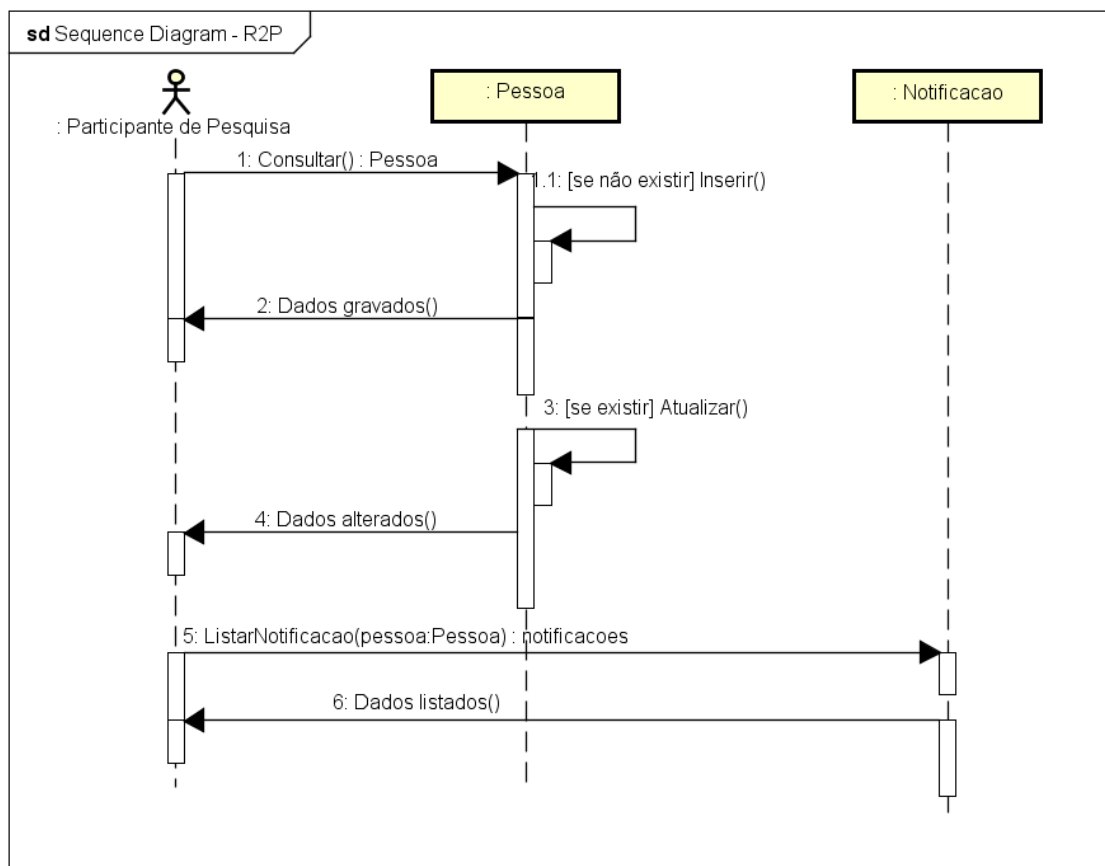
Alternativo 3 (Participante de Pesquisa cadastrado no sistema corporativo) O participante de pesquisa pode realizar o seu cadastro pelo aplicativo.

4.2.3 Diagrama de Sequência

Consiste em um diagrama que tem o objetivo de mostrar como as mensagens entre os objetos são trocadas no decorrer do tempo para a realização de uma operação.[Fowler 2003]

- Um diagrama de sequência é representado através de duas dimensões: –a dimensão horizontal, que representa o conjunto de objetos intervenientes; –a dimensão vertical que representa o tempo.

A figura 4.3 representa a fluxo para visualizar as notificações no aplicativo. O participante de pesquisa informa o CPF e seguida os dados serão conferidos se já não cadastrado será direcionado para tela de cadastro em seguida aparecerá a mensagem dados gravados, caso o contrário o participante de pesquisa irá visualizar seu dados realizando atualizações de informações do cadastro retornando a mensagem dados alterados e em sequência listar as notificações, retornando a mensagem dados listados.



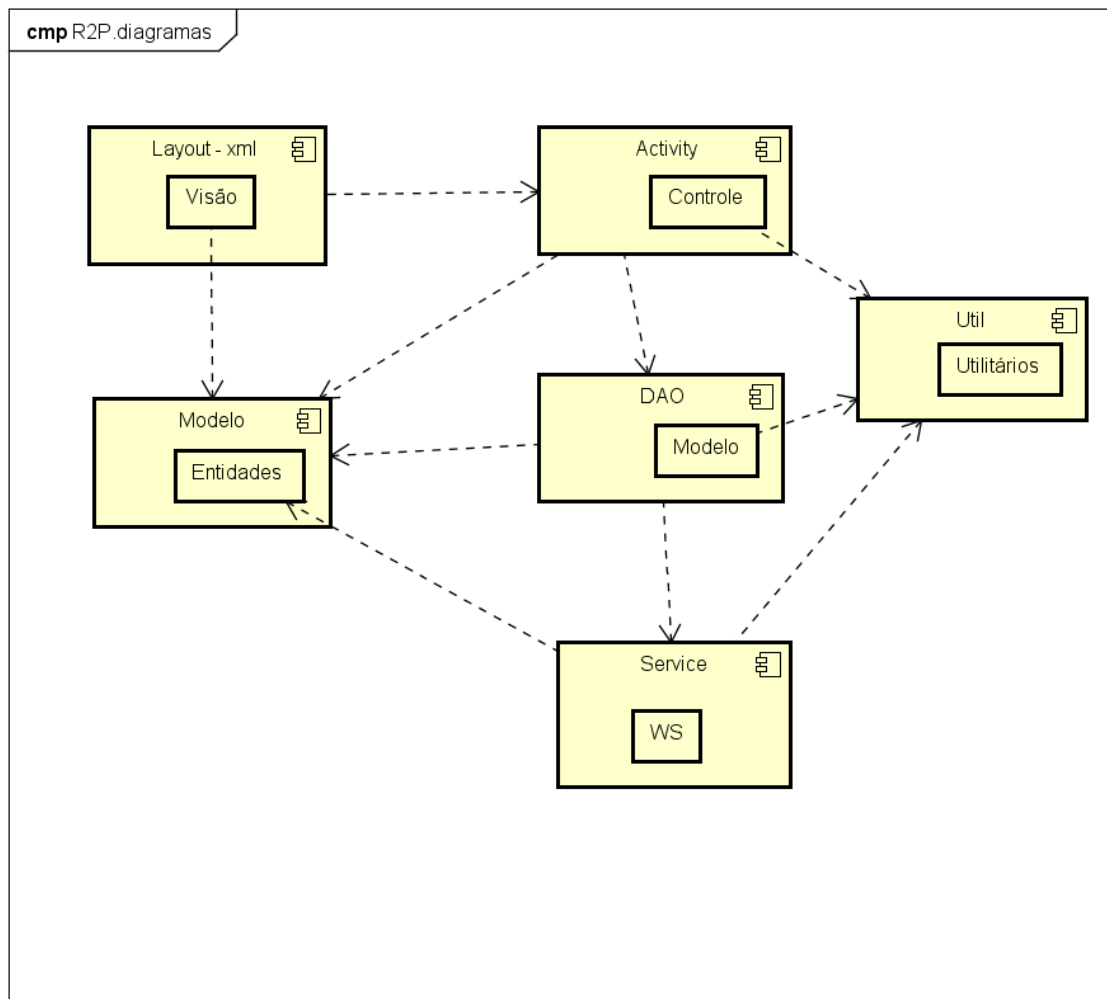
powered by Astah

Figura 4.3: Diagrama de Sequencia Visualizar Notificação

4.2.4 Diagrama de Componentes

O diagrama de componentes permite a modelagem lógica e física de um sistema, através da visão dos seus componentes e relacionamentos entre eles. Semelhante a um pacote, mas com enfoque do empacotamento físico do código. É composto basicamente por componente, interface e relacionamentos.

Na figura 4.4 é mostrado o diagrama de componentes do aplicativo R2P.



powered by Astah

Figura 4.4: Diagrama de Componetes

4.2.5 Arquitetura de Funcionamento do Aplicativo

Ao acessar o aplicativo R2P os telefones devem estar na internet e com sistema operacional *android* assim o fluxo segue passando por um *firewell* interno da empresa ICF que irá direcionar para um servidor apache tomcat e assim acessar os a base de dados. Conforme demonstrado na 4.5.

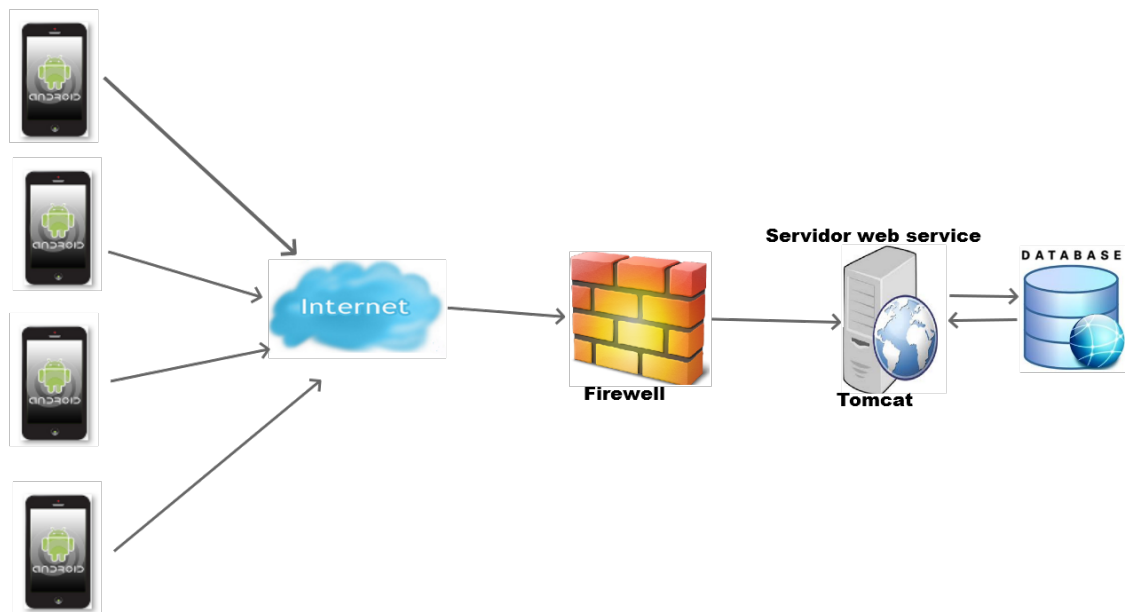


Figura 4.5: *Aquitetura de funcionamento do Aplicativo*

4.3 Prototipação

A prototipação é uma etapa fundamental na construção de software, pois ajuda a facilitar o entendimento dos requisitos, apresentar conceitos e funcionalidades do software.

Nesta etapa, é prudente que todos os envolvidos no projeto também estejam envolvidos em sua criação.

Os protótipos são grandes aliados das metodologias ágeis de desenvolvimento pois garantem maior alinhamento entre equipe e cliente. Eles podem ser desenvolvidos em diferentes níveis de fidelidade: quanto maior ela for, mais o protótipo se assemelhará ao resultado entregue. No entanto, um protótipo de alta fidelidade leva mais tempo para ser criado ou modificado. A escolha do protótipo ideal varia de acordo com o nível de entendimento do negócio, a complexidade dos requisitos, prazo e orçamento para elaboração.

Podemos dividir os protótipos em três categorias: *Wireframes* e Rascunhos, Protótipos Visuais e Protótipos Interativos.

Um software normalmente é reproduzido em cópias originais e seu código-fonte é único, porém tem complexidades suficientes para necessitar de protótipos antes de seu total desenvolvimento. Os protótipos de software são representações que descrevem todas as características funcionais do software, ou seja, aquelas que serão utilizadas no dia-a-dia, pelo usuário [Rodrigues].

Utilizaremos em nosso aplicativo os protótipos visuais, que são criados com um programa de edição gráfica Pencil², estes protótipos tem maior apoio visual. Não possuem interações de tela e demandam mais tempo para fazer ajustes e melhorias. São uma ótima opção para telas com maior ênfase em estética e usabilidade, quando os requisitos já foram entendidos.

4.3.1 Aplicativo R2P

Tela do Aplicativo R2P



Figura 4.6: Tela entrar no Aplicativo R2P

Nesta figura 4.6 informamos o CPF e podemos entrar no aplicativo.

Tela do aplicativo R2P Cadastro de Participante de Pesquisa

²<http://pencil.evolus.vn/>



The image shows a black smartphone displaying a registration form titled "R2P - RECRUTAMENTO 1.0". The form is white with blue borders and contains the following fields and elements:

- Top status bar: Signal strength, Wi-Fi, battery, and time 12:00.
- Header: "R2P - RECRUTAMENTO 1.0" in a blue bar.
- Buttons: "Voltar" (Back) and "Salvar" (Save) in blue boxes.
- Form fields: "CPF", "Nome", "Data Nascimento" (with a date picker), "E-mail", "Celular", and "Fixo".
- Gender selection: Two checkboxes labeled "Masculino" and "Feminino".

Figura 4.7: *Tela De Cadastro Participante de Pesquisa*

Nesta figura 4.7 podemos incluir um novo participante de pesquisa. A tela apresenta os campos para o participante de pesquisa digitar os dados do participante. Na parte superior temos as ações da tela.

Tela do aplicativo R2P Editar de Participante de Pesquisa



Figura 4.8: *Tela De Edição Participante de Pesquisa*

Nesta figura 4.8 além de podemos editar os dados do participante de pesquisa, podemos visualizar as mensagens dos estudos (notificações). A tela apresenta os campos para o participante de pesquisa digitar os dados do participante. Na parte superior temos as ações da tela.

Tela do aplicativo R2P Listar Notificações

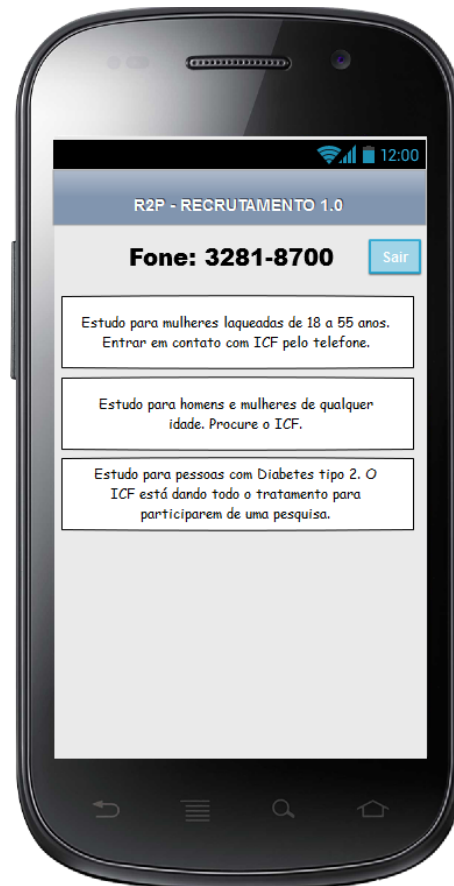


Figura 4.9: Tela De Visualização de notificações

Nesta figura 4.9 podemos visualizar todas as notificações dos estudo que o participante de pesquisa pode participar de acordo com o seu perfil. Na parte superior temos as ações da tela.

Considerações Finais

5.1 Conclusão

Com este aplicativo desenvolvido a empresa ICF ira atingir maiores números de de recrutamentos de participantes de pesquisa para seus estudos com baixo custo de divulgação de informativos de estudos. E devido hoje em dia os *smartphones* estão cada vez mas presentes na vida das pessoas, com uma aplicativo rodando no telefone fica muito mais fácil contato com estas pessoas. Facilitando a visualização da notificações no seu próprio celular.

5.1.1 Contribuição

Neste trabalho foi possível desenvolver um aplicativo para recrutamento de participantes de pesquisas.

- Foi estudado o referencial teórico para o desenvolvimento de um aplicativo Android;
- Recolhido os requisitos funcionais e não funcionais;
- Foi realizado o projeto de Banco de Dados;
- Implementado um aplicativo denominado R2P;
- Foi criado o caso de uso, diagrama de classes e diagrama de sequência;
- Os resultados foram analisados de acordo com o que foi pedido.

5.2 Trabalhos Futuros

Antes, sem um aplicativo para notificar os participantes de pesquisa para participarem de um estudo era feito através de aquisições de pacotes de SMS para envio nos celulares dos mesmos. O que gerava custos altos a empresa. Agora com o aplicativo a empresa envia notificações as pessoas a custo zero.

Para trabalhos futuros será criada uma tela para interações entre a empresa e os participantes de pesquisa que ele irão enviar dúvidas com o aplicativo R2P e a

empresa irá responder através de um sistema corporativo. Será feito melhorias visual das telas do aplicativo. Implantação do *Push Notifications*, ou seja toda vez que a empresa ICF cadastrar uma notificação nova o participante de pesquisa irá receber bip em seu celular avisando de novo estudo disponível para sua participação. O estudo é cadastro no sistema corporativo da empresa ICF. Cada estudo tem uma identificação com exemplo PBIO010/11 que não pode ser divulgado para os participantes de pesquisa, devido a uma determinação da ANVISA (órgão do governo que regulamenta a área de pesquisa de medicamentos no Brasil), por isso esta identificação não será divulgada nas mensagens enviadas.

Referências Bibliográficas

[D'Souza 1998]D'SOUZA, D. *Interface Specification, Refinement, and Design with UML/Catalysis*. [S.l.: s.n.], 1998. 12-18 p.

[Fowler 2003]FOWLER, M. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. 3. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003. ISBN 0321193687.

[Lee, Schneider e Schell 2005]LEE, V.; SCHNEIDER, H.; SCHELL, R. *Aplicações móveis: arquitetura, projeto e desenvolvimento*. Pearson Makron Books, 2005. ISBN 9788534615402. Disponível em: <<https://books.google.co.uk/books?id=mA9cYgEACAAJ>>.

[MONTEIRO]MONTEIRO, J. *GOOGLE ANDROID - CRIE APLICAÇÕES PARA CELULARES E: TABLETS*. CASA DO CODIGO. ISBN 9788566250022. Disponível em: <<https://books.google.com.br/books?id=QFUdnQEACAAJ>>.

[Rodrigues]RODRIGUES, E. *Curso de Engenharia de Software*. Universo dos Livros Editora. ISBN 9788578730109. Disponível em: <<https://books.google.com.br/books?id=ZJznA9UrtVAC>>.

[Silva, Freire e Callahan]SILVA, C. T.; FREIRE, J.; CALLAHAN, S. P. *Provenance for Visualizations: Reproducibility and Beyond*. [S.l.: s.n.].

[Sommerville et al. 2008]SOMMERVILLE, I. et al. *Engenharia de software*. ADDISON WESLEY BRA, 2008. ISBN 9788588639287. Disponível em: <<https://books.google.com.br/books?id=iflYOgAACAAJ>>.

[Tanenbaum e Steen 2006]TANENBAUM, A. S.; STEEN, M. v. *Distributed Systems: Principles and Paradigms (2Nd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. ISBN 0132392275.