

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ESPECIALIZAÇÃO EM TECNOLOGIA JAVA

ANDERSON SOLTOVSKI

**PROTÓTIPO DE APLICATIVO PARA CONTROLE DE GESTANTES
PELOS SEUS OBSTETRAS EM DISPOSITIVOS MÓVEIS**

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA - PR
2012

ANDERSON SOLTOVSKI

**PROTÓTIPO DE APLICATIVO PARA CONTROLE DE GESTANTES
PELOS SEUS OBSTETRAS EM DISPOSITIVOS MÓVEIS**

Monografia apresentada ao Programa de Pós-Graduação em JAVA da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de Especialista.

Orientador: Prof. Dr. Adolfo Gustavo Serra Seca Neto

CURITIBA - PR
2012

TERMO DE APROVAÇÃO

Esta monografia foi apresentada às 17 h 30 min, do dia 27 de agosto de 2012, como requisito parcial para a obtenção do título de Especialista em Tecnologia Java – Departamento Acadêmico de Informática – Universidade Tecnológica Federal do Paraná. O candidato apresentou o trabalho para a Banca Examinadora composta pelos professores abaixo assinados. Após a deliberação, a Banca Examinadora considerou o trabalho APROVADO.

Prof. Dr. Alexandre Reis Graeml
Membro
(UTFPR)

Prof. Dr. João Alberto Fabro
Membro
(UTFPR)

Prof. Dr. Adolfo Gustavo Serra Seca Neto
Orientador
(UTFPR)

Visto da Coordenação:

Prof. Dr. João Alberto Fabro
Coordenador
Curso de Especialização em Tecnologia Java

RESUMO

SOLTOVSKI, Anderson. Protótipo de aplicação para controle de gestantes pelos seus Obstetras em dispositivos móveis. 2012. Monografia (Especialização em Tecnologia Java) – Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná. Curitiba, 2012.

Esse projeto apresenta um sistema implementado em JAVA, para rodar na plataforma Android. Apresenta os conceitos de Nuvem para armazenamento e sincronização dos dados inseridos. Tendo público alvo os médicos da área de Ginecologia e Obstetrícia, implementa uma espécie de agenda, com funções direcionadas ao controle gestacional de suas pacientes, onde os dados podem ser acessados diretamente no celular. Apresenta um cadastro mais resumido do que os sistemas utilizados em consultório, mas tão importantes na tomada de decisões nos procedimentos cirúrgicos.

Palavras Chave: Controle Gestacional. Android. SQLite.

ABSTRACT

SOLTOVSKI, Anderson. Prototype application for control of pregnant by their Obstetricians in mobile devices. 2012. Monograph (Specialization in Java Technology) – Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná. Curitiba,2012.

This project presents a system implemented in JAVA, to run on Android platform. Introduces the concepts of cloud storage and synchronization of data entered. As target audience, the physicians in the area of Obstetrics and Gynecology. It implements a sort of schedule, with functions aimed at gestational control of patients, where data can be accessed directly on your mobile. It introduces registrations in a more summarized way than the systems used in clinics, so important in decision-making of surgical procedures.

Keywords: Gestational Control. Android. SQLite.

LISTA DE GRÁFICOS

Figura 1 - Utilização de sistemas Operacionais em Smartphones	12
Figura 2 - Representação na Internet em Diagrama de Rede	17
Figura 3 - Diagrama de Caso de Uso	19
Figura 4 - Tela de Visualização de Agenda.....	22
Figura 5 - Menu da Tela de Agenda.....	23
Figura 6 - Dados cadastrais da paciente.....	24
Figura 7 - Calendário Gestacional.....	25
Figura 8 - Cadastro de Gestante	26
Figura 9 - Cadastro de Conclusão de Procedimento	27
Figura 10 - Conceito de Model-View-Controller no Android	28
Figura 11 - Diagrama de Classes do Sistema.....	29
Figura 12 - Referencia dos Objetos da Classe GestanteEditActivity	30
Figura 13 - Implementação do método setOnClickListener do botão Salvar	30
Figura 14 - Recuperação dos dados do formulário	31
Figura 15 - Método getView da classe GestanteListAdapter.....	31
Figura 16 - Classe GestanteDAO.....	32
Figura 17 - Diagrama do Banco de Dados	33

LISTA DE TABELAS

Tabela 1 - Versões do Android homologadas para rodar o sistema.....	20
Tabela 2 - Requisitos Funcionais Visíveis	21
Tabela 3 - Requisitos Funcionais Transparentes.....	21
Tabela 4 - Tabela de cores correspondente ao Período Gestacional	22

SIGLAS

DUM	Data da Última Menstruação
DPP	Data Provável do Parto
OHA	Open Handset Alliance
SDK	Software Development Kit (Kit para Desenvolvimento de Sistemas)
JVM	Java Virtual Machine (Máquina Virtual Java)
OS	Operating System (Sistema Operacional)
IDE	Integrated Development Environment (Ambiente Integrado para Desenvolvimento)
API	Application programming Interface (Interface de Programação de Aplicativos)

SUMÁRIO

1.	Introdução.....	10
2.	Revisão Bibliográfica	14
2.1.	Assistência Pré-Natal.....	14
2.1.1.	Cálculo da Última Menstruação.....	14
2.1.2.	Data da Ultrassonografia.....	14
2.2.	Android.....	14
2.2.1.	Android SDK e Eclipse.....	15
2.2.2.	Android Play	15
2.3.	SQLite	16
2.4.	Computação em Nuvem	16
2.4.1.	Google APP Engine	18
3.	Metodologia.....	19
3.1.	Desenvolvimento do Projeto	19
3.1.1.	Requisitos Não-Funcionais	19
3.1.2.	Requisitos Funcionais.....	20
3.1.3.	Visualização da Agenda.....	21
3.1.4.	Consulta do Paciente.....	23
3.1.4.1.	Calendário Gestacional	25
3.1.4.2.	Edição de Cadastro.....	26
3.1.4.3.	Conclusão de Procedimentos.....	26
3.1.5.	Cadastro do Paciente	27
3.2.	Desenvolvimento e Design.....	28
3.2.1.	Interação com o usuário, a View e o Controller.....	29
3.2.2.	Persistência dos Dados.....	32
3.2.3.	Sincronização dos Dados.....	33
3.2.3.1.	Atualização dos dados no dispositivo móvel	34
3.2.3.2.	Funções de Sincronização no Webservice	34
4.	Apresentação e Discussão dos Resultados	36
4.1.	Operacionalidade	36
4.2.	Aparência do Aplicativo	36
4.3.	Novas Funcionalidades.....	36
4.3.1.	Calculadora Gestacional Autônoma.....	37
4.3.2.	Relatórios Gerenciais	37
4.3.3.	Sincronização de Dados	37
4.3.4.	Plano de Negócio	37

5. Considerações Finais	38
5.1. Trabalhos Futuros	38
6. Referências.....	39

1. Introdução

Com o avanço da tecnologia e a chegada de dispositivos móveis com maior capacidade de processamento e recursos técnicos avançados podemos oferecer novos sistemas que auxiliem nas atividades das pessoas.

Profissionais das mais diversas áreas necessitam de informações o mais rápido possível. Na área médica de obstetrícia, uma rápida tomada de decisão aliada às informações que disponha naquele momento pode ser vital para mãe e bebê. É comum os especialistas dessas áreas receberem chamadas de suas pacientes em trabalho de parto a qualquer momento do dia ou da noite. Quando o médico recebe uma dessas solicitações de atendimento, precisa saber rapidamente algumas informações para tomar uma decisão.

A possibilidade de ter sempre a mão as informações de suas pacientes facilita ao médico a tomada de decisões, antecipando procedimentos e diagnósticos. Uma paciente que se encontre no último trimestre da gestação que entre em trabalho de parto precisa de rápido atendimento, pois uma criança está para nascer. Quando uma gestante entra em trabalho de parto e o período gestacional aponta uma atividade prematura, o profissional precisa efetuar procedimentos que posterguem essa gravidez, evitando esse parto prematuro.

Nesse trabalho, será proposto um sistema de controle de gestantes para sistemas Android, independente da plataforma utilizada. Será desenvolvido sobre as reais necessidades dos médicos obstetras. Essas informações também poderão ser atualizadas com outros dispositivos, para garantir rapidez de acesso e segurança no armazenamento.

1.1. Problema

Atualmente os médicos obstetras tem a informação da gestação de duas maneiras: o controle atualizado em sistemas de consultório e em carteiras de pré-natal. A primeira torna o acesso quase que inviável, pois pode ser utilizada somente *in loco*, dentro do consultório, ou até em sistemas mais modernos, através da disponibilização das informações na internet. E como nosso sistema de redes móveis ainda não é muito estável, pode haver interrupções ou demora de acesso dependendo do lugar onde o profissional se encontra, além dos sistemas atuais não estarem compatíveis com browsers de smartphones e tablets.

A segunda opção, a carteira de pré-natal, nem sempre está em posse da futura mãe, a não ser nos casos onde essa chamada ocorra quando ela está em casa. E a informação dessa gestante, através de lembrança, nem sempre é confiável.

A necessidade de ter os dados básicos para uma avaliação médica, principalmente de

emergência, ou mesmo eletivamente é importante para esses profissionais. Além de agilizar o atendimento, a conduta será mais bem escolhida com essas informações.

Sem essas informações em tempo real, torna-se necessário efetuar novos exames, para que esses possam ajudar a determinar a idade gestacional da paciente, que pode demorar na tomada da decisão, visto que nem sempre as maternidades dispõem de Ecografistas de plantão para efetuar alguns exames, tornando necessário aguardar a sua chegada.

1.2. Justificativa

Os celulares têm se tornado cada vez mais parte da vida das pessoas. Hoje o incomum é encontrar uma pessoa que não tenha um. Segundo dados da Anatel, o Brasil tinha, em abril de 2012, 252,9 milhões de linhas de celulares ativas. Outro número que também cresce é dos smartphones. Uma pesquisa do Instituto Brasileiro de Opinião Pública e Estatística, IBOPE, publicada em 07 de março de 2012 aponta que 32% dos entrevistados pretende comprar um smartphone em 2012.

Atualmente, a maneira de contato mais utilizada para as chamadas médicas é através de ligações telefônicas, quase sempre para os celulares dos profissionais. Um aplicativo que contenha as informações das pacientes armazenados no celular seria de grande valia. Alguns profissionais da área médica utilizam anotações na própria agenda para salvar os dados do paciente. Mas o cálculo da idade atual do bebê ainda precisa ser feito manualmente.

Para o ambiente iOS da Apple, existem alguns aplicativos para esse cálculo, um dos mais completos é OB Patient, um sistema voltado para o cálculo gestacional. Esse sistema trabalha com medidas nos padrões americanos e está disponível apenas em inglês.

Conforme a Figura 1, podemos observar que o sistema da Apple foi ultrapassado em números de aparelhos. Além disso, a gama de fabricantes e dispositivos disponíveis com o sistema Android tem aumentado, tornando esse ambiente mais comum para as pessoas.

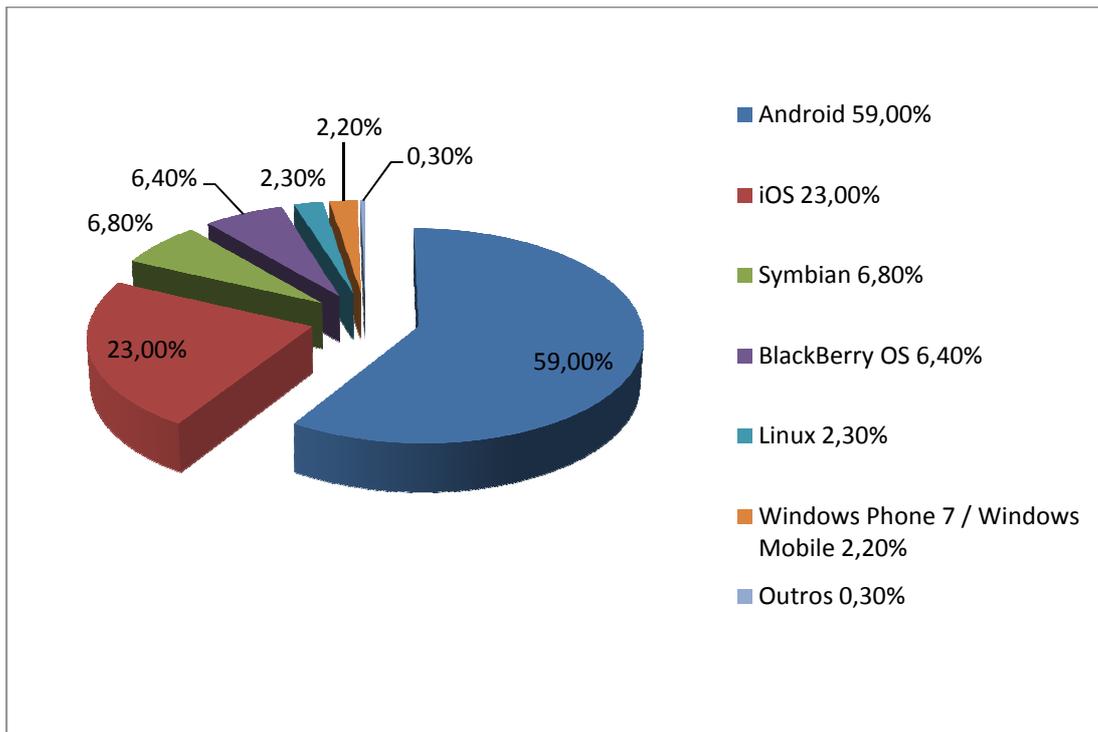


Figura 1 - Utilização de sistemas Operacionais em Smartphones

Fonte: Adaptado de IDC Worldwide Mobile Phone Tracker, 2012

Outro fator a ponderar é a troca de celulares, que pode ser motivada por um extravio, furto, ou pelo simples fato de escolher um aparelho mais moderno. Por isso, os dados precisam ser migrados junto com o proprietário. O armazenamento em nuvem surge para ajudar a conciliar esse problema, pois as informações podem ser migradas entre os dispositivos sem a perda das informações. E essa comunicação possibilita que os cadastros possam ser feitos em um ambiente mais agradável, como uma plataforma desktop, já que essas informações podem ser sincronizadas com os dispositivos móveis a qualquer momento.

Então, propõe-se uma estrutura que possa ter acesso de pelo menos duas formas: do celular, para acesso das informações dos pacientes em tempo real e outra que seria através de um browser de internet, onde a utilização e inserção dos dados pode ser feita de maneira mais ágil. A interação entre os dispositivos deve ser feita o mais frequente possível, para evitar que informações desatualizadas sejam apresentadas em qualquer dispositivo utilizado com o sistema.

1.3. Objetivos

O objetivo principal desse trabalho trata-se do desenvolvimento de um aplicativo de controle de gestantes para sistemas Android com sincronização em nuvem através de aplicativo disponibilizado pela Google.

O usuário encontrará em seu dispositivo móvel, seja ele um celular ou um tablet, um aplicativo de utilização amigável que tentará sempre mostrar as informações de maneira mais rápida possível. Os cadastros poderão ser feitos a partir do próprio dispositivo móvel, ou através de um site, que estará acessível em qualquer lugar.

No site, o usuário irá entrar com seu login e senha, e terá disponíveis as seguintes opções:

- Cadastro de Pacientes;
- Calculadora Gestacional;
- Consulta de Pacientes;
- Calendário Gestacional.

Os cadastros realizados através do site serão armazenados diretamente no ambiente centralizador das informações, distribuídos sobre o conceito de nuvem. Essas informações também estarão disponíveis no smartphone ou tablet através de um app para Android com as seguintes telas:

- Consulta de Pacientes;
- Cadastro de Pacientes;
- Calendário Gestacional.

Os dados inseridos pelo celular serão sincronizados com o ambiente centralizador sempre que o aparelho estiver disponível acesso a uma rede de dados, seja 3G ou Wi-Fi, quando o aplicativo for iniciado. Ele tanto enviará as alterações como receberá as últimas atualizações feitas através de outras opções.

1.4. Estrutura do Trabalho

Este trabalho abordará o desenvolvimento para dispositivos móveis que adotam o sistema Android. A programação em nuvem irá abordar a tecnologia disponibilizada pela Google, para armazenamento e sincronização dos Serviços.

No primeiro capítulo serão apresentados o problema que levou a essa pesquisa e a justificativa para o desenvolvimento deste trabalho. No segundo capítulo é apresentado o embasamento teórico utilizado para realização do projeto. No capítulo subsequente, capítulo três, será detalhado a estrutura e como foi implementado o trabalho proposto. O capítulo quatro será destinado à discussão dos resultados obtidos com o aplicativo. O quinto capítulo é destinado às considerações finais do projeto e trabalhos futuros.

2. Revisão Bibliográfica

2.1. Assistência Pré-Natal

Após a descoberta da gestação, as futuras mães devem fazer consultas periódicas para acompanhar a evolução do feto. “O Objetivo da assistência pré-natal é assegurar o nascimento de uma criança saudável, com o mínimo de risco para mãe” (ZUGAIBE, 2008, p 196). A primeira das atividades para esse acompanhamento é descobrir a idade gestacional através da data da concepção do feto. Através dessa data é possível acompanhar se a evolução do bebê está de acordo com o período compreendido em cada consulta. A idade gestacional é contada em semanas e dias.

Uma gestação normal deve ter no mínimo cinco consultas, as quais não devem ter um intervalo superior a oito semanas (FREITAS et al, 2006).

O cálculo gestacional deve ser atualizado a cada consulta. Essa data pode ser obtida através de dois métodos: DUM – Data da Última Menstruação e ultrassonografia.

2.1.1. Cálculo da Última Menstruação

Para se calcular a idade gestacional deve-se saber o período correspondido entre o primeiro dia da última menstruação até a data da consulta. Somando sete dias a essa data e diminuindo-se três do mês da DUM, podemos obter a data provável do parto, de acordo com a regra de Nägele (FREITAS et al, 2006).

2.1.2. Data da Ultrassonografia

O outro método para se obter a data da concepção do feto é através de um exame de imagem que indica através de algumas medidas, como comprimento cabeça-nádega, a idade atual do bebê. Para um menor desvio padrão esse exame deve ser realizado no primeiro trimestre da gestação.

2.2. Android

A plataforma Android é uma plataforma de desenvolvimento para aplicativos móveis baseada no sistema operacional Linux. Esse sistema operacional teve grande difusão principalmente em smartphones e tablets. De propriedade da Google, chegou ao mercado para criar uma plataforma única, para que os consumidores tenham no produto final um ambiente moderno e flexível (LECHETA, 2010).

A plataforma Android possibilita a criação de aplicativos desenvolvidos em Java, graças à máquina virtual Dalvik. Assim, os aplicativos podem ter seus executáveis distribuídos diretamente, rodando em qualquer dispositivo com esse sistema operacional. Desde sua criação, o Android passou por várias mudanças, recursos foram agregados, seu visual alterado. Devido a essas mudanças novas versões foram lançadas. A versão mais atual que temos hoje é a 4.0 Ice Cream (LECHETA, 2010).

Devido aos recursos incrementados em cada versão, um aplicativo é escrito para rodar com um requisito mínimo de versão, devido aos recursos utilizados pelo aplicativo.

2.2.1. Android SDK e Eclipse

Para auxiliar no desenvolvimento dos aplicativos Android, a Google disponibilizou um software que contém algumas ferramentas e uma API para linguagem Java necessária para o desenvolvimento, além de um emulador para simular um dispositivo com seu sistema operacional. Dessa maneira é possível testar o aplicativo juntamente com o dispositivo.

Para o IDE de desenvolvimento Eclipse foi disponibilizado um plugin, no qual o desenvolvedor pode iniciar o emulador diretamente, como também pode debugar o aplicativo, podendo visualizar onde os principais erros ocorrem e como o sistema está se comportando em tempo de execução (LECHETA, 2010).

Na instalação do SDK podemos ainda instalar as diversas versões do Android, para assim realizar os testes em todas as versões nas quais o software irá trabalhar. Mesmo que um aplicativo desenvolvido para rodar na API 4 (versão 1.6 do Android) possa ser instalado em qualquer versão mais recente, é recomendável que se faça testes em todas as demais, pois podem surgir diferenças principalmente na parte gráfica, e o resultado apresentado poderá não ser o esperado.

2.2.2. Android Play

Com as aplicações prontas e testadas podemos comercializá-las e distribuí-las de duas maneiras: copiando o instalador, que tem a extensão apk para o aparelho específico e executando-a, ou através de uma loja online, também disponibilizada pela Google, a Google Play.

Todos os dispositivos Android, a partir da versão 2.2 vinham com um aplicativo de compras chamado Google Market, recentemente renomeado para Google Play. Nesse

aplicativo é possível localizar facilmente milhares de apps para Android, que podem ser gratuitos ou pagos. Basta ter uma conta da Gmail e um cartão de crédito para adquiri-las.

Para a distribuição dos aplicativos através desse serviço, é necessário fazer um cadastro e pagar uma taxa de vinte e cinco dólares. Com esse cadastro pode-se criar e distribuir aplicativos sem limites.

2.3. SQLite

Para armazenar os dados do aplicativo desenvolvido, o Android trás embutido o SQLite, que é uma biblioteca que implementa um mecanismo de banco de dados, sem servidor e com código de domínio público (LECHETA, 2010).

O SQLite dá suporte à maioria das funções da SQL92, a terceira revisão do padrão SQL. Armazena as informações em um arquivo único e o tamanho do seu código é pequeno; por isso, pode ser utilizado em uma ampla gama de aparelhos, que disponham de recursos limitados (SQLITE, 2012).

Com a utilização desse banco de dados, o armazenamento, consulta e manutenção dos dados se torna mais rápido e prático. Os celulares de gerações tecnológicas anteriores proviam o armazenamento em arquivos simples.

Esse banco de dados pode ser instalado em outras plataformas, como o Windows, Linux e IOS, facilitando o desenvolvimento, podendo-se testar as consultas e comandos SQL em um ambiente gráfico. O limite desse banco de dados será dado pelo tamanho da memória do dispositivo, podendo chegar a terabytes de armazenamento (SQLITE, 2012).

2.4. Computação em Nuvem

A computação em nuvem poder ser definida como uma metáfora da Internet. A internet pode ser representada em digrama de rede como uma nuvem, como na figura 2. A cloud computing, computação em nuvem, permite utilizar as aplicações via Internet, independente da plataforma ou equipamento (VELTE, 2011).

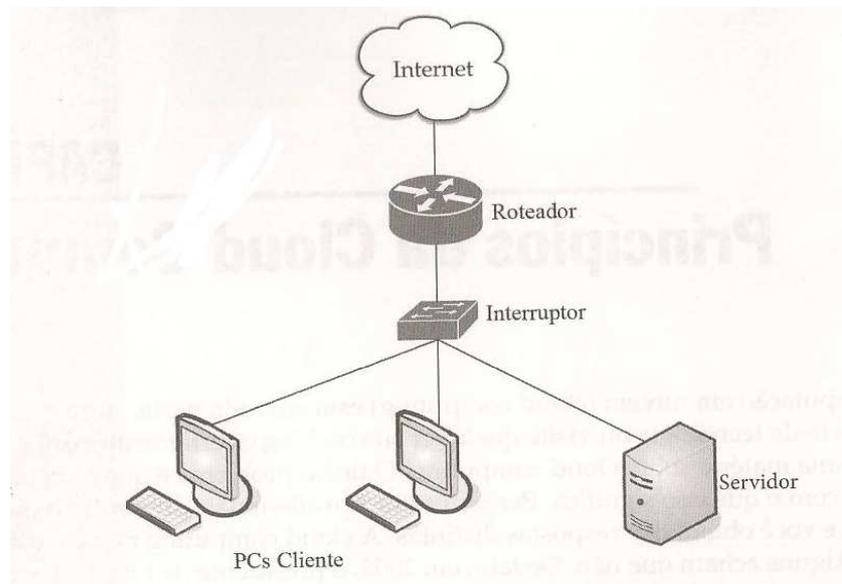


Figura 2 - Representação na Internet em Diagrama de Rede

Fonte: VELTE, 2011

Um das maiores vantagens da computação em nuvem é a despreocupação com os equipamentos do servidor. A idéia é adquirir esse serviço de grandes empresas especializadas em hospedagem. E distribuir os aplicativos na forma de serviços, com o poder de reutilização e grande escalabilidade (VELTE, 2011).

Com esse tipo de distribuição, podemos mudar o aplicativo cliente sem precisar nos preocupar com o serviço instalado no dispositivo do cliente, ou ao contrário, podemos alterar funções e fórmulas sem precisar atualizar todos os clientes.

A possibilidade de portabilidade e sincronização dos dados tornará o aplicativo mais seguro, já que o cliente é independente em quase 100% do armazenamento, uma perda do dispositivo do usuário não afetará na perda dos dados. Um cliente que faz uma alteração num dispositivo móvel e depois acessa o aplicativo em outro aparelho, terá as informações alteradas e atualizadas automaticamente.

Um contra ao uso de serviços distribuídos em nuvem, é a obrigatoriedade do acesso a internet. A instabilidade da rede, por exemplo, pode causar lentidão ou quebra no acesso ao serviço (VELTE, 2011).

2.4.1. Google APP Engine

A Google criou e disponibilizou um serviço onde armazena os aplicativos criados em uma linguagem que possam ser interpretados em uma máquina virtual Java (JVM), como de servlets Java, e programas em linguagem padrão Java.

No pacote ofertado estão o escalonamento e balanceamento de carga automático, persistência de dados, e alguns serviços prontos, como a integração com as contas de email Google, o Gmail. Nesse caso o desenvolvedor não precisa se preocupar com a autenticação do sistema, apenas integrar esse serviço ao seu aplicativo.

Para ajudar o desenvolvimento, também são disponibilizados SDKs que simulam todos os serviços do App Engine localmente. Dessa forma é possível testar as aplicações mais facilmente, antes do envio do aplicativo para o servidor. A Google também disponibiliza uma cota gratuita para o desenvolvimento, que corresponde a até dez aplicativos e quinhentos mega bytes de armazenamento (Google App Engine, 2012).

3. Metodologia

O protótipo sugerido tem a função de mostrar como será o desenvolvimento do aplicativo de controle de gestantes para dispositivos móveis. Nessa proposta será abordado exclusivamente o módulo Android.

Para o desenvolvimento foi utilizado a IDE do Eclipse e o Android SDK. Para entender as reais necessidades e como melhor funcionaria esse sistema, foram consultados médicos especializados na área de Ginecologia e Obstetrícia. Nessas entrevistas, o principal requisito abordado era que teria que ser algo muito prático, que não necessitasse de manual e instruções. Precisaria ser algo intuitivo, de natural aprendizagem. Telas com número limitados de campos, e o menor número de ações poderiam tornar esse projeto um caso de sucesso nesse setor.

3.1. Desenvolvimento do Projeto

Após algumas entrevistas com os profissionais que desejam o sistema, foram definidas funcionalidades e requisitos aprovados por todos os envolvidos. Esses foram contextualizados num diagrama de caso de uso, mostrado na Figura 3.

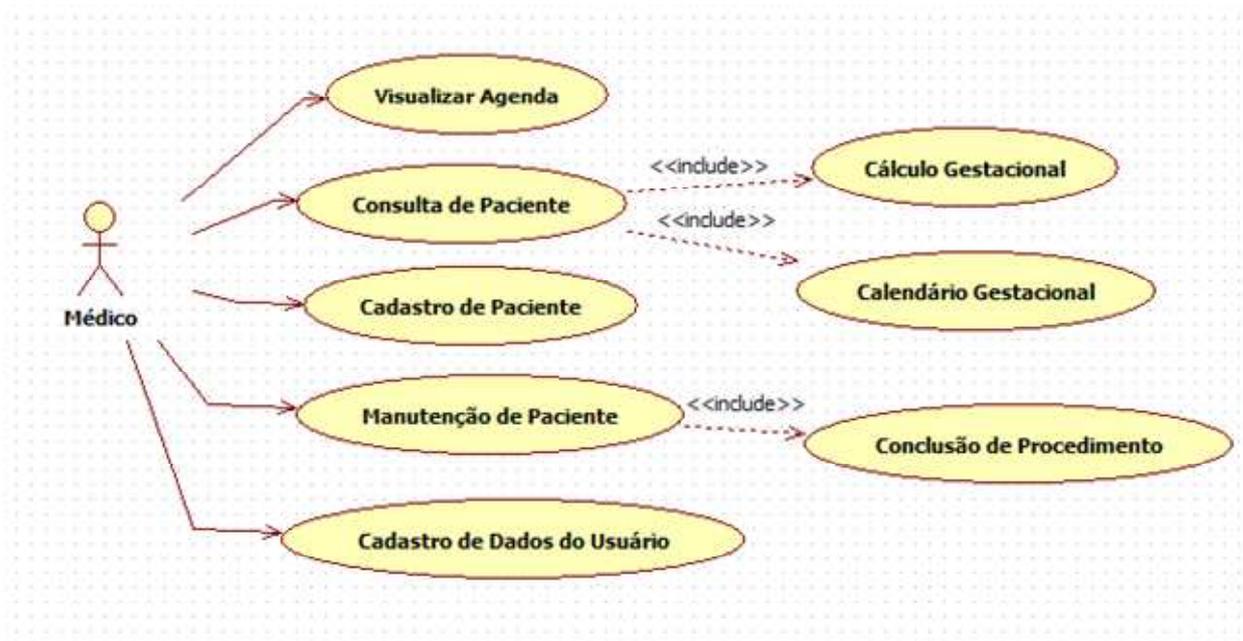


Figura 3 - Diagrama de Caso de Uso

Fonte: Autoria própria.

Com a definição desse diagrama pode-se definir os requisitos para o sistema.

3.1.1. Requisitos Não-Funcionais

Para instalação do sistema, o usuário terá que ter um dispositivo móvel com sistema Android 2.2 ou superior. Deverá ter uma conta cadastrada no Google Play para poder fazer a aquisição e download do aplicativo. O sistema foi homologado para rodar em qualquer versão do Android superior a 2.2 Froyo, apontadas na Tabela 1.

Versão	Distribuição	Número da API
2.2	Froyo	8
2.3.X	Gingerbread	9-10
3.X.X	Honeycomb	11-13
4.0	Ice Cream Sandwich	14-15

Tabela 1 - Versões do Android homologadas para rodar o sistema
Fonte: Adaptado de Lecheta, 2010

3.1.2. Requisitos Funcionais

As funcionalidades do aplicativo podem ser divididas em dois grupos, as visíveis e invisíveis. As visíveis são aquelas operadas pelo usuário e as invisíveis são os métodos transparentes, que cuidam da sincronização e atualização dos dados.

Na Tabela 2, podemos acompanhar os métodos utilizados pelo usuário, para interagir com o sistema.

Requisito	Funcionalidade
Visualização de Agenda	Assim que o sistema é inicializado, uma tela com todos os pacientes ativos é mostrada. Essa pesquisa é dividida pelo trimestre da gestação e pode ser ordenada por nome da paciente ou pela idade gestacional.
Cadastro de Paciente	No menu da primeira tela, pode-se chamar a tela de inclusão de novas pacientes. Nesse cadastro são inseridos os dados básicos da paciente e os dados referentes à gestação.
Consulta de Paciente	Ao clicar em uma paciente na agenda, primeira tela, o usuário é levado a uma tela onde são mostrados todos os dados daquela paciente. Nessa tela pode-se chamar a alteração dos dados, visualização do calendário gestacional e a conclusão do procedimento.
Calendário Gestacional	Nesse item, abre-se uma lista com a evolução da idade gestacional

	dia a dia.
Manutenção da Paciente	Caso necessite a alteração de algum dado do paciente, na tela de consulta tem um botão que chama essa tela, onde essas informações podem ser alteradas ou excluídas.
Conclusão	Quando uma gestação é interrompida é possível cadastrar o resultado final através da Conclusão. Esse cadastro tem dados referentes ao procedimento realizado.
Cadastro de dados do Usuário	A possibilidade de sincronização dos dados não é um item obrigatório, mas caso o usuário queira essa funcionalidade, deverá preencher seus dados para essa sincronização. Esse cadastro é mantido com uma conta de e-mail da Gmail do próprio usuário. Caso o mesmo não disponha dessa conta, poderá fazer gratuitamente no site www.gmail.com .

Tabela 2 - Requisitos Funcionais Visíveis

Fonte: Autoria própria.

Algumas funcionalidades do sistema não dependem de interações do usuário para serem realizadas, são chamadas de forma automática. Esses citados na tabela 3.

Requisito	Funcionalidade
Sincronização dos Dados	Toda vez que o usuário entrar ou sair do sistema, será disparado um método de sincronização, que fará a atualização dos dados na nuvem e no aparelho.

Tabela 3 - Requisitos Funcionais Transparentes

Fonte: Autoria própria.

3.1.3. Visualização da Agenda

Quando o sistema se inicia, a primeira tela que aparece ao usuário é a agenda gestacional. Nessa tela é apresentada uma lista com nome e idade gestacional de todas as pacientes cadastradas no sistema.

Essa visualização pode ser modificada de acordo com a necessidade do médico. Ele pode mudar a ordem entre nome da paciente e idade gestacional e aplicar filtros para mostrar somente as pacientes ainda gestantes ou as pacientes que já tiveram suas gestações concluídas.

Quando a ordem escolhida é a idade gestacional, o sistema divide a visualização dessa lista em três níveis, de acordo com a tabela 4.

Trimestre	Semanas compreendidas	Cor da Linha
Primeiro	0 à 12	Amarelo
Segundo	13 à 24	Azul
Terceiro	25 à 37	Verde
Terceiro (Período de Termo) ¹	38 ao nascimento	Vermelho

Tabela 4 - Tabela de cores correspondente ao Período Gestacional

Fonte: Autoria própria.

Na figura 4, podemos observar um cadastro onde temos gestantes nas quatro formas possíveis de idade gestacional.



Figura 4 - Tela de Visualização de Agenda

Fonte: Autoria própria.

¹ Nas referências médicas o terceiro trimestre gestacional corresponde ao período de 25 semanas até o nascimento do bebê. Após a trigésima oitava semana a gestante está de Termo, período em que o feto está maduro suficiente para o nascimento.

A tela de visualização de agenda é construída com o layout de lista. Dessa maneira é possível rolar a listagem para cima ou para baixo para visualizar outras pacientes.

Ao tocar em uma das linhas apresentadas, é apresentada uma nova tela com os dados cadastrais completos da paciente, que será abordada no próximo item.

Nessa tela do sistema, podemos acessar e realizar algumas funções através de um menu, representado na figura 5:

- **Gestantes:** ao clicar nesse item, a visualização filtra o resultado, e somente são exibidos cadastros não realizados;
- **Realizadas:** esse item permite filtrar todas as pacientes que tiveram suas gestações concluídas;
- **Ord. Nome:** é possível visualizar a lista de pacientes em ordem alfabética;
- **Ord. Idade:** através da ordenação pela idade gestacional, o sistema classifica os dados cadastrados pela idade gestacional da ecografia atual, em ordem decrescente.
- **Nova Gestante:** para inserir um novo registro no banco de dados, deve-se chamar esse menu.



Figura 5 - Menu da Tela de Agenda

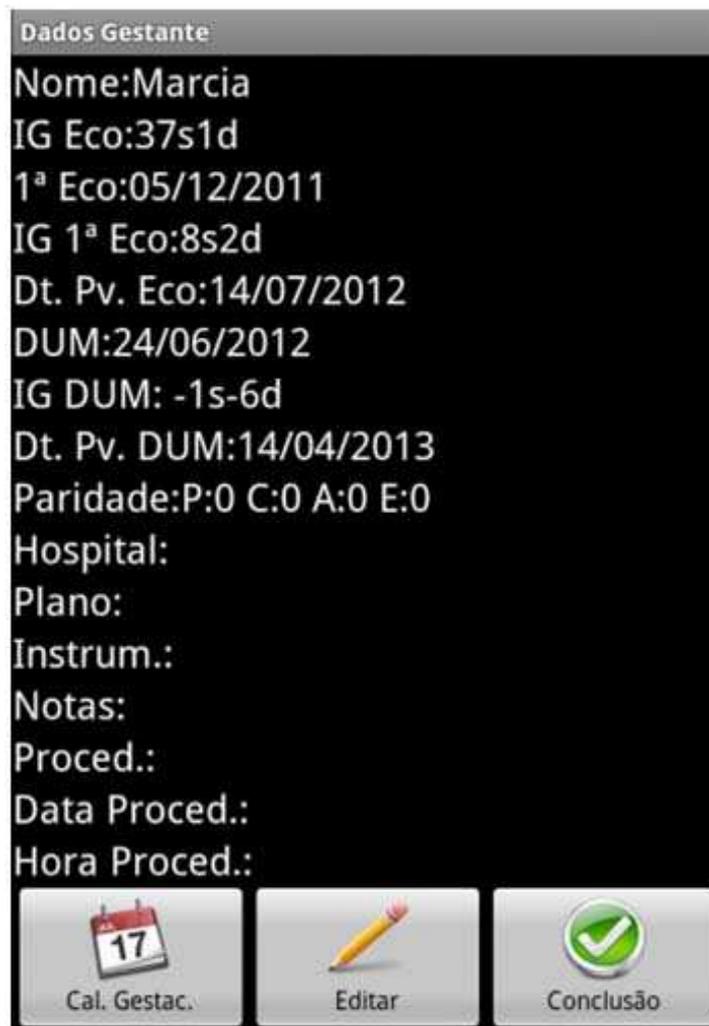
Fonte: Autoria própria.

3.1.4. Consulta do Paciente

Através da seleção de uma linha na tela da agenda, o sistema abre uma nova view, ou Activity. Nessa tela, figura 7 são mostrados os principais dados da paciente:

- **Nome**
- **IG Eco:** Idade gestacional através da primeira ecografia;
- **1ª Eco:** Data da primeira ecografia;
- **IG 1ª Eco:** Idade gestacional apontada na primeira ecografia;
- **Dt. Pv. Eco:** Data prevista para o parto através da ecografia;

- **DUM:** Data da última menstruação;
- **IG DUM:** Idade gestacional através da DUM;
- **Dt. Pv. DUM:** Data prevista para o parto através da DUM;
- **Paridade:** Número de partos, cesarianas, abortos e gestações ectópicas anteriores;
- **Hospital:** Hospital escolhido para o procedimento;
- **Plano:** Plano de saúde que a paciente tem cobertura;
- **Instrum.:** Anotações referentes a instrumentação da cirurgia;
- **Notas**
- **Procedimento:** Procedimento realizado;
- **Data Proced.:** Data agendada para o procedimento;
- **Hora Proced.:** Horário agendada para o procedimento;



Dados Gestante

Nome: Marcia
IG Eco: 37s1d
1ª Eco: 05/12/2011
IG 1ª Eco: 8s2d
Dt. Pv. Eco: 14/07/2012
DUM: 24/06/2012
IG DUM: -1s-6d
Dt. Pv. DUM: 14/04/2013
Paridade: P:0 C:0 A:0 E:0
Hospital:
Plano:
Instrum.:
Notas:
Proced.:
Data Proced.:
Hora Proced.:

Cal. Gestac. Editar Conclusão

Figura 6 - Dados cadastrais da paciente

Fonte: Autoria própria.

Através dessa tela, podemos tomar três ações. abrir o calendário gestacional, editar os dados cadastrados e cadastrar a conclusão do procedimento, os quais serão descritos a seguir.

3.1.4.1. Calendário Gestacional

O calendário gestacional é fundamental importância para o acompanhamento do desenvolvimento fetal. Auxilia o Obstetra a programar a solicitação de exames específicos e também no agendamento das consultas de retorno do pré-natal.

A figura 7 ilustra uma pequena parte da evolução de uma gestação.

Calendário Gestacional	
36s1d	17/06/2012-dom
36s2d	18/06/2012-seg
36s3d	19/06/2012-ter
36s4d	20/06/2012-qua
36s5d	21/06/2012-qui
36s6d	22/06/2012-sex
37s0d	23/06/2012-sáb
37s1d	24/06/2012-dom
37s2d	25/06/2012-seg
37s3d	26/06/2012-ter
37s4d	27/06/2012-qua
37s5d	28/06/2012-qui
37s6d	29/06/2012-sex
38s0d	30/06/2012-sáb
38s1d	01/07/2012-dom
38s2d	02/07/2012-seg
38s3d	03/07/2012-ter
38s4d	04/07/2012-qua
38s5d	05/07/2012-qui
38s6d	06/07/2012-sex

Figura 7 - Calendário Gestacional

Fonte: Autoria própria.

3.1.4.2. Edição de Cadastro

A tela de edição de cadastros é a mesma em que cadastramos uma nova gestante, com a particularidade de que os dados já vem preenchidos. Essa tela será abordada na tela de cadastro de gestantes. Com relação ao menu, na edição podemos excluir um registro, cancelar as alterações realizadas ou salvar as modificações. A figura 8 exemplifica essa tela.

Cadastro Gestante

Nome
Marcia Cristina

Data Última Menstruação
Jun 24 2012

Primeira Eco
Dec 05 2011

Idade Gestacional Ecografia
08 2

Parto Cesária Aborto Ectóp.
0 0 0 0

Hospital

Salvar Cancelar Excluir

Figura 8 - Cadastro de Gestante

Fonte: Autoria própria.

3.1.4.3. Conclusão de Procedimentos

Ao final da gestação alguns dados podem ser lançados para indicar o fechamento do procedimento. Nessa tela pode-se indicar que o procedimento foi realizado, selecionando entre parto, cesariana, aborto ou nenhum, no caso em que a paciente tenha abandonado o pré-natal ou

optado pela troca do médico, como informar a data e horário da realização do mesmo, exemplificado na figura 9.

The image shows a mobile application interface for recording procedure completion. The form is titled 'Nome: Marcia' and includes a checked checkbox labeled 'Realizado'. Below this is a dropdown menu for 'Procedimento' currently set to 'Nenhum'. The 'Data Procedimento' section features three spinners for month ('Jun'), day ('24'), and year ('2012'). The 'Horário do Procedimento' section has two spinners for hours ('23') and minutes ('47'). At the bottom, there is an 'Anotações' field and two buttons: 'Salvar' (Save) and 'Cancelar' (Cancel).

Figura 9 - Cadastro de Conclusão de Procedimento

Fonte: Autoria própria.

3.1.5. Cadastro do Paciente

O cadastro simplificado ajuda na agilidade da inclusão de informações. Por isso, o número de campos é reduzido, contendo informações relevantes para ajudar numa tomada de decisão.

Através do primeiro menu é possível chamar a tela de inclusão de gestantes. Essa tela propõe alguns campos de cadastro, mas nem todos são obrigatórios:

- **Nome**
- **IG Eco:** Idade gestacional através da primeira ecografia;

- **1ª Eco:** Data da primeira ecografia;
- **DUM:** Data da última menstruação;
- **Paridade:** Mostra o número de partos, cesarianas, abortos e gestações ectópicas anteriores;
- **Hospital:** Hospital escolhido para o procedimento;
- **Plano:** Plano de saúde que a paciente tem cobertura;
- **Instrum.:** Anotações referentes a instrumentação da cirurgia;
- **Notas:** Dados adicionais sobre a realização do procedimento

3.2. Desenvolvimento e Design

Os sistemas para dispositivos que utilizam Android são desenvolvidos na linguagem Java. A Google disponibiliza um SDK que auxilia no desenvolvimento do aplicativo. Com essa ferramenta é possível integrar o desenvolvimento na IDE Eclipse, possibilitando o debug em um emulador, que vem junto nesse SDK.

A arquitetura de desenvolvimento do Android implementa de forma clara o modelo MVC. A *View* e o *Controller* são responsáveis por refletir o estado do modelo, na figura 10 podemos ver as interações entre esses elementos (ROGERS et al, 2009).

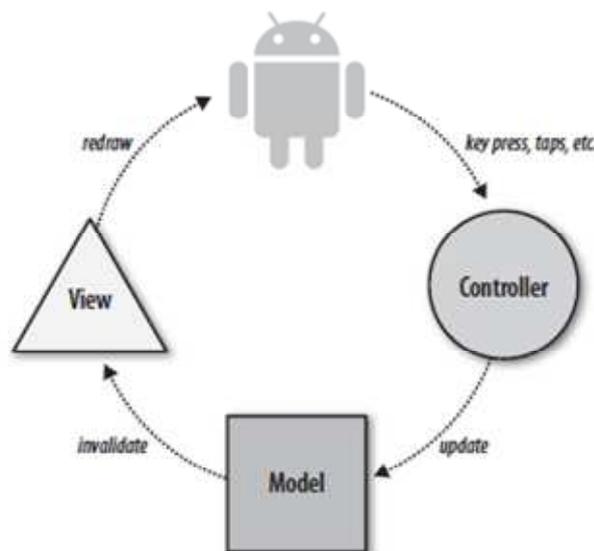


Figura 10 - Conceito de Model-View-Controller no Android

Fonte: ROGERS et al, 2009

Na figura 11 podemos observar essas interações através do diagrama de classes resumido do sistema; para facilitar essa visualização foram ocultados os atributos e métodos das classes e duplicada a classe Gestante.

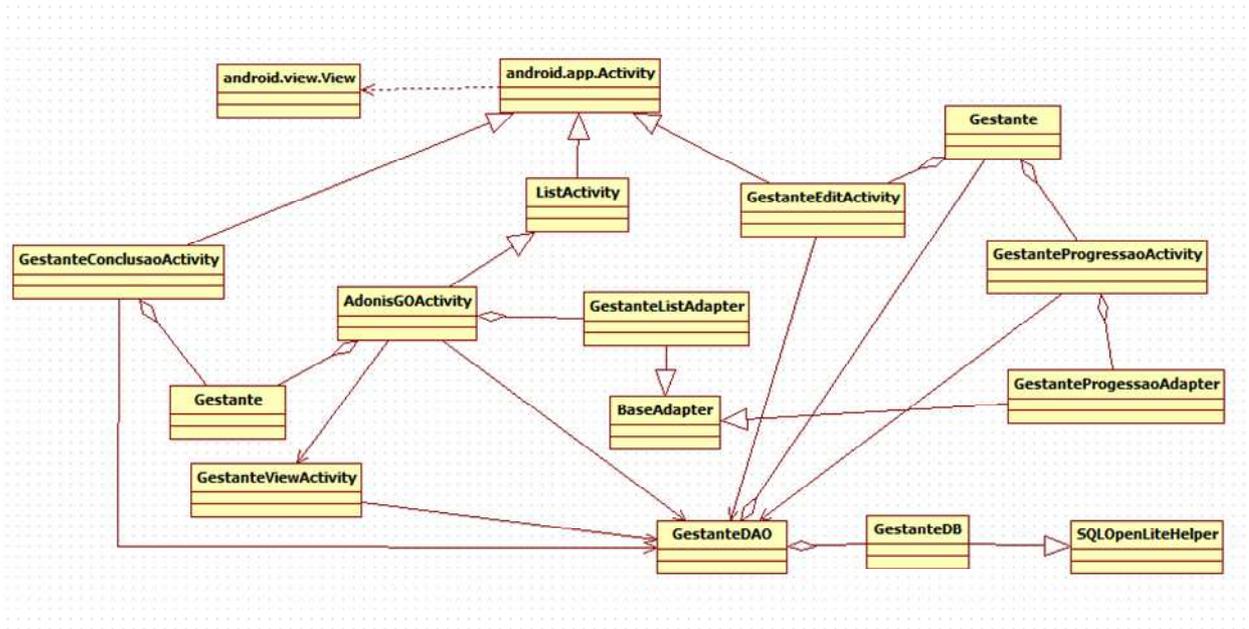


Figura 11 - Diagrama de Classes do Sistema

Fonte: Autoria própria.

3.2.1. Interação com o usuário, a View e o Controller

No aplicativo dispomos das interações com usuário, sejam elas para mostrar uma tela, ou captar dados fornecidos pelo usuário. Para cada tela apresentada existe um arquivo xml, o qual é responsável pela interface da tela. Nesse arquivo estão representados todos os componentes que serão desenhados na tela, botões, textos, outros componentes e o próprio layout que será apresentado.

Para controlar essas interações existe uma classe que representa esse arquivo, representada no nosso modelo pela classe `GestanteEditActivity`, que estende da `Activity`. Essa classe depende da classe `android.view.view`, responsável por controlar o estado e os eventos dessa tela. O arquivo é referenciado nessa classe através do método `setContentView`, e para cada componente desenhado, devemos ter um objeto referenciado com o mesmo tipo. Na figura 12 podemos observar essas referências (LECHETA, 2010).

```

public class GestanteEditActivity extends Activity {
    static final int RESULT_SALVAR = 1;
    static final int RESULT_EXCLUIR = 2;
    static final int SALVAR = 3;
    static final int CANCELAR = 4;
    static final int EXCLUIR = 5;

    protected EditText edNome;
    protected DatePicker dtDUM;
    protected DatePicker dtEco;
    protected Spinner spIdadeDia;
    protected Spinner spIdadeSemana;
    protected Spinner spParto;
    protected Spinner spCesaria;
    protected Spinner spAborto;
    protected Spinner spEctopica;
    protected EditText edHospital;
    protected EditText edConvenio;
    protected EditText edInstrumentacao;
    protected EditText edTelefone;
    protected EditText edEmail;
    protected EditText edObs;
    private Long id;
    private Gestante gestante;
    private GestanteDAO datasource;
    public Boolean excluir = false;
    Intent it;

    @Override
    protected void onCreate(Bundle icicle) {
        super.onCreate(icicle);

        setContentView(R.layout.cadastro);

        edNome = (EditText) findViewById(R.id.edNome);
        dtDUM = (DatePicker) findViewById(R.id.dtDUM);
        dtEco = (DatePicker) findViewById(R.id.dtEco);
        spIdadeDia = (Spinner) findViewById(R.id.spIdadeDia);
        spIdadeSemana = (Spinner) findViewById(R.id.spIdadeSemana);
        spParto = (Spinner) findViewById(R.id.spParto);
        spCesaria = (Spinner) findViewById(R.id.spCesaria);
        spAborto = (Spinner) findViewById(R.id.spAborto);
        spEctopica = (Spinner) findViewById(R.id.spEcto);
        edHospital = (EditText) findViewById(R.id.edHospital);
        edConvenio = (EditText) findViewById(R.id.edConvenio);
        edInstrumentacao = (EditText) findViewById(R.id.edInstrumentacao);
        edTelefone = (EditText) findViewById(R.id.edTelefone);
        edEmail = (EditText) findViewById(R.id.edEmail);
        edObs = (EditText) findViewById(R.id.edObs);
    }
}

```

Figura 12 - Referencia dos Objetos da Classe GestanteEditActivity

Fonte: Autoria própria.

Podemos então utilizar a referência para pegar as interações do cliente com o sistema. Para salvar as informações digitas o usuário clica num botão salvar. Dentro da classe GestanteEditActivity, foi mapeado esse botão e implementado o método setOnClickListener, representado na figura 13.

```

Button btSalvar = (Button) findViewById(R.id.btnSalvar);
btSalvar.setOnClickListener(new OnClickListener() {
    public void onClick(View view) {
        salvar();
    }
});

```

Figura 13 - Implementação do método setOnClickListener do botão Salvar

Fonte: Autoria própria.

Esse método, por sua vez, precisa captar os valores informados nos campos pelo usuário para fazer o devido tratamento, como representado na figura 14.

```

Calendar c = Calendar.getInstance();
this.gestante.setNome(edNome.getText().toString());
this.gestante.setConvenio(edConvenio.getText().toString());
this.gestante.setHospital(edHospital.getText().toString());
this.gestante.setIdadeDia(spIdadeDia.getSelectedItemPosition());
this.gestante.setIdadeSemana(spIdadeSemana.getSelectedItemPosition());
this.gestante.setInstrumentacao(edInstrumentacao.getText().toString());
c.set(dtEco.getYear(), dtEco.getMonth(), dtEco.getDayOfMonth());
this.gestante.setDataPrimeiraEco(this.gestante.getDateS(c.getTime()));
c.set(dtDUM.getYear(), dtDUM.getMonth(), dtDUM.getDayOfMonth());
this.gestante.setDataUltMenstruacao(this.gestante.getDateS(c.getTime()));
this.gestante.setAborto(spAborto.getSelectedItemPosition());
this.gestante.setParto(spAborto.getSelectedItemPosition());
this.gestante.setCesaria(spAborto.getSelectedItemPosition());
this.gestante.setEctopica(spAborto.getSelectedItemPosition());
this.gestante.setTelefone(edTelefone.getText().toString());
this.gestante.setEmail(edEmail.getText().toString());
this.gestante.setConclusao(edObs.getText().toString());

```

Figura 14 - Recuperação dos dados do formulário

Fonte: Autoria própria.

O sistema tem uma agenda com as gestantes cadastradas, ordenadas pela idade gestacional ou pelo nome. Na criação desta view temos a classe GestanteListAdapter que é responsável por montar as informações através do método getView, representado na figura 15.

```

public View getView(int position, View convertView, ViewGroup parent) {
    // Recupera a Gestante da posição atual
    Gestante g = lista.get(position);

    LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    View view = inflater.inflate(R.layout.gestante_linha, null);

    // Atualiza o valor do TextView
    TextView nome = (TextView) view.findViewById(R.id.linNome);
    nome.setText(g.getNome());

    TextView idade = (TextView) view.findViewById(R.id.linIdade);
    idade.setText(g.getIdadeAtual());
    //Conforme a idade atual define o layout da linha
    if (g.getSemanaAtual() < 13){
        view.setBackgroundColor(Color.rgb(176, 196, 222));
        idade.setTextColor(Color.BLACK);
        nome.setTextColor(Color.BLACK);
    } else if (g.getSemanaAtual() < 25){
        view.setBackgroundColor(Color.rgb(255, 236, 139));
        idade.setTextColor(Color.BLACK);
        nome.setTextColor(Color.BLACK);
    }
    } else if (g.getSemanaAtual() < 37){
        view.setBackgroundColor(Color.rgb(198, 113, 113));
        idade.setTextColor(Color.BLACK);
        nome.setTextColor(Color.BLACK);
    }
    } else {
        view.setBackgroundColor(Color.rgb(113, 198, 113));
        idade.setTextColor(Color.BLACK);
        nome.setTextColor(Color.BLACK);
    }
    }

    return view;
}

```

Figura 15 - Método getView da classe GestanteListAdapter

Fonte: Autoria própria.

3.2.2. Persistência dos Dados

As informações inseridas pelos usuários precisam ser armazenadas de alguma maneira para o sistema ter uma utilidade válida. Como abordado anteriormente, o Android dispõe em suas bibliotecas um pacote para armazenamento de dados, o SQLite.

Esse pacote implementa as condições básicas para se ter uma boa consistência no armazenamento e busca de informações. No projeto, duas classes são responsáveis pelo gerenciamento das informações locais:

- **GestanteDB:** Classe estendida de SQLiteOpenHelper, responsável pela criação e alteração da base de dados na instalação e atualização do aplicativo. Quando um aplicativo é instalado pela primeira vez, o método onCreate dessa classe é chamado, nesse momento podemos criar nossa estrutura de tabelas e inserir dados necessários para o aplicativo, ou até mesmo dados de exemplo. Quando uma versão mais recente do sistema é instalada, o método onUpgrade é chamado, nesse momento as alterações necessárias ao banco podem ser executadas;
- **GestanteDAO:** Dentro do sistema é a classe responsável pela interação com o banco de dados. Essa classe é responsável pela persistência dos dados, abertura e fechamento do banco de dados quando solicitado. Implementa os métodos de inclusão, exclusão, alteração e recuperação dos dados cadastrados pelo usuário. Essa classe está representada na figura 16.

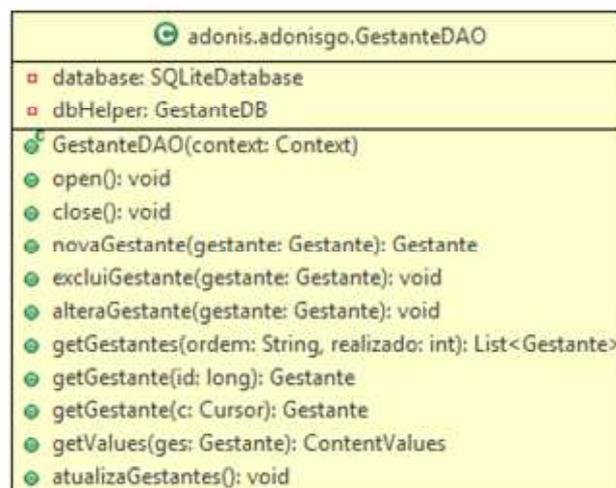


Figura 16 - Classe GestanteDAO

Fonte: Autoria própria.

O sistema dispõe de duas tabelas para armazenamento dos dados: Gestante e System, conforme figura 17. A primeira é responsável pelo armazenamento dos dados da gestante, e a segunda para guardar os dados do usuário e horário da última sincronização. O sistema está preparado para receber informações de mais de um usuário, no caso do uso compartilhado do mesmo dispositivo.

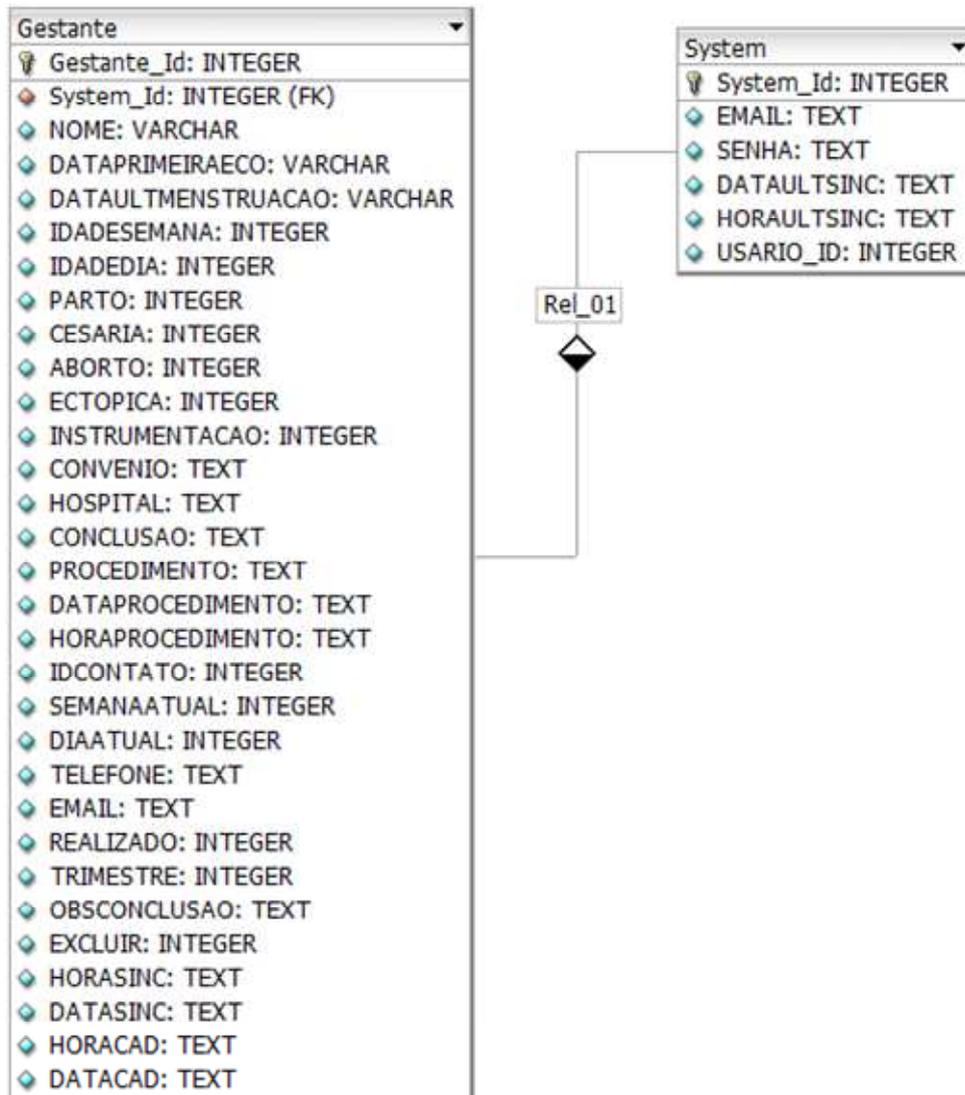


Figura 17 - Diagrama do Banco de Dados

Fonte: Autoria própria.

3.2.3. Sincronização dos Dados

Uma das propostas da aplicação é a portabilidade dos dados. Essa sincronização é realizada em duas frentes distintas de trabalho, as tarefas de gerenciamento e armazenamento dos dados atualizados na nuvem, e a sincronização das alterações realizadas localmente.

3.2.3.1. Atualização dos dados no dispositivo móvel

O aplicativo instalado no dispositivo móvel com Android, possui uma classe responsável pela troca de informações com o Webservice. Essa classe é ativada em toda inicialização do sistema. Ela possui métodos que se autenticam na área do cliente, através do e-mail cadastrado e senha. Após a autenticação consulta os dados que foram alterados, removidos os inseridos no servidor por outros dispositivos. Depois dessa leitura, o sistema envia as alterações realizadas localmente para o servidor.

A sincronização dos dados é realizada através de atributos que controlam a data de inserção, alteração e exclusão. O método crítico no sistema é da alteração dos dados, já que um mesmo registro pode ser alterado ao mesmo tempo em dispositivos diferentes. Quando isso ocorre, o sistema considera o registro com a data de alteração mais recente, desconsiderando totalmente o registro mais antigo.

Quanto a exclusão de um registro, isso só ocorre dentro do servidor, que é quem envia a requisição aos clientes para que um registro seja apagado por completo. Enquanto isso, o registro fica oculto, sem aparecer para o usuário.

3.2.3.2. Funções de Sincronização no Webservice

O servidor de Webservice é o responsável pela manutenção e gerenciamento das informações. É ele quem distribui as informações para os clientes, orientando a ação a tomar. Quando ele recebe uma informação de exclusão, prepara o banco de dados para que na próxima atualização do cliente o registro seja totalmente apagado. Como o número de clientes é variável, esses dados são movidos para uma tabela de arquivo morto, onde o dado permanecerá até que seja requisitado um credenciamento do serviço.

Esse serviço implementa basicamente três métodos públicos:

- **Autenticação:** somente com essa chamada os clientes podem fazer novas requisições para o servidor. Tem dois parâmetros, e-mail e senha, e devolve como resposta um id de conexão se o login estiver correto ou uma mensagem de erro quando o contrário;
- **Sincronização:** com a chamada desse método, passando o e-mail, data da última sincronização e id de conexão, o servidor devolve um arquivo xml com todos os dados alterados com data maior da última sincronização;

- **Atualização:** esse último método é chamado passando a id de conexão, e-mail e os dados a serem atualizados através de um arquivo xml. Como resposta será recebido um outro arquivo de sincronização, com possíveis alterações ocorridas nesse intervalo de tempo.

4. Apresentação e Discussão dos Resultados

Após um período de estudo e desenvolvimento, um pequeno protótipo ficou pronto. Durante o desenvolvimento, um médico ginecologista e obstetra era constantemente convidado a discutir o aplicativo, dando opiniões e ideias para a ferramenta. Ao final da primeira versão, foi disponibilizado um aplicativo beta a três médicos para testarem a funcionalidade do aplicativo.

4.1. Operacionalidade

O primeiro desafio foi criar um aplicativo que fosse autoinstrutivo, que não dependesse de manual. Isso foi possível graças ao projeto manter um padrão utilizado nas aplicações nativas do Android. Menus, telas de avanço, componentes comuns aos que os usuários Android já tinham contato anteriormente.

O aplicativo foi instalado manualmente em um smartphone e dois tablets, dentre os quais dois com sistema Android Gingerbread e um Ice Cream Sandwich. Após a instalação, nenhuma outra informação foi passada aos usuários, para simular uma compra realizada diretamente ao Google Play.

As três pessoas escolhidas para o teste não tiveram problemas para se adaptar ao aplicativo e sair cadastrando suas gestantes.

4.2. Aparência do Aplicativo

Um ponto fraco apontado durante os testes foi com relação ao visual. Houve menções que compararam o aplicativo a programas antigos, escritos em linguagens de texto. Telas mais modernas, com visuais mais empolgantes seriam mais atrativos. Foi sugerido o uso de imagens de fundo que pudessem representar a gestação na idade atual.

4.3. Novas Funcionalidades

Ao recuperar as bases cadastrais após algum tempo de uso, foi possível observar que os pesquisados não preenchiam todos os campos cadastrais propostos. Principalmente a base utilizada no smartphone, ela tinha os cadastros menos completos, se comparado com as bases utilizadas nos tablets. O motivo relatado foi a falta de agilidade com o digitação em um dispositivo tão pequeno. Foi sugerido que os campos não preenchidos fossem mantidos, pois outros profissionais poderiam utilizá-los

Apesar do não preenchimento dos campos, os usuários não deixaram de sugerir algumas funcionalidades que podem ser agregadas as próximas versões.

4.3.1. Calculadora Gestacional Autônoma

Como o sistema já faz o cálculo gestacional de forma coesa, foi sugerida uma calculadora autônoma, onde pudessem ser feitos os cálculos sem precisar do cadastro completo da gestante. Principalmente para pacientes ocasionais e atendimentos em postos de saúde, onde as paciente podem ser atendidas por outros médicos, principalmente em horários fora do consultório, onde elas buscam auxílio aos médicos de plantões em unidades de saúde e hospitais.

4.3.2. Relatórios Gerenciais

Mesmo mencionando que era apenas um protótipo, que o aplicativo estava em desenvolvimento, foi consenso a disponibilização de relatórios no sistema. Consultas que mostrem o percentual dos procedimentos realizados, como a contagem, o número atual de gestantes em cada trimestre, foram alguns dos relatórios solicitados.

4.3.3. Sincronização de Dados

Na versão de testes, foi criado um botão na tela inicial com o texto "Sincronizar Dados". Nenhum dos voluntários ao teste questionou essa funcionalidade, sendo que um nem mesmo clicou nela. Isso mostrou que essa sincronização precisa ser de forma automática e transparente, ou muitas pessoas não vão realizá-la, o que levará a perda de informações por completo, caso os dispositivos sofram algum problema ou extravio.

Após clicar nesse botão, era necessário cadastrar o e-mail e uma senha. Nenhum dos entrevistados fez esse cadastro, o que nos leva a tornar esse cadastro obrigatório já na instalação do dispositivo, tornando-se um requisito não funcional para o aplicativo.

4.3.4. Plano de Negócio

Como estratégia para comercialização do aplicativo serão construídas duas versões: uma gratuita, com limite de cadastro de vinte gestantes, sem sincronização de dados e anúncios na abertura do aplicativo, e uma segunda, que será completa e terá a possibilidade de sincronização dos dados, por sua vez, paga.

As duas versões serão colocadas na Google Play para venda, pois é mais fácil o gerenciamento, visto que os usuários de sistema Android tem mais facilidade para localizar aplicativos.

5. Considerações Finais

O trabalho apresentado teve sua base na necessidade real dos médicos obstetras. Eles precisavam de uma ferramenta para auxiliar nas suas tarefas cotidianas. Esse simples aplicativo contribuiu na solução desse problema. As entrevistas realizadas com os usuários destinados a esse sistema mostraram que pequenas soluções como essa podem ajudar no dia a dia de inúmeras pessoas.

Embora esse aplicativo seja direcionado a uma única especialização dentro da medicina, pode-se notar a grande dificuldade que profissionais dessa área têm em encontrar sistemas que ajudem a resolver seus problemas. Outras áreas da saúde podem ser exploradas e serem objetivos de pesquisas e novos projetos.

As plataformas atuais trouxeram uma nova realidade aos dispositivos móveis. Um dos pontos fortes foi certa padronização das características dos sistemas implementados. O sistema operacional Android é um desses exemplos. Os sistemas usam bibliotecas únicas de componentes, inclusive os mesmos utilizados pelo próprio OS. Essa padronização facilita o aprendizado aos sistemas, que popularizam cada vez mais a instalação de aplicativos nos dispositivos móveis, principalmente nos celulares, fato que até então era algo distante dos usuários mais simples da telefonia móvel.

Por esse motivo, alguns estudos apresentados nesse trabalho poderão ser reaproveitados em outros projetos. A montagem da lista de resultado é um desses exemplos que ajudarão a outros desenvolvedores na hora de escreverem suas aplicações, visto que os exemplos consultados durante o projeto, não eram claros o suficiente para a reutilização.

O aprendizado adquirido dentro desse trabalho foi além do complemento do conteúdo das disciplinas abordado no curso. Proporcionou criar uma ferramenta que ajudará a resolver o problema já mencionado, e servirá de base para outros projetos também na plataforma Android, escritos na linguagem Java.

5.1. Trabalhos Futuros

Um módulo de cadastro e consulta via browser de plataformas desktop será de grande utilidade. Seria mais fácil para o usuário conseguir realizar alguns cadastros quando ele está em consultório em um computador ou notebook.

O sistema também precisa de um módulo gerencial, onde se possa emitir relatórios e consultas gerais do sistema. Dessa maneira o usuário poderá contar com uma ferramenta que apoiará em diversas tomadas de decisões, podendo analisar dados reais do seu próprio histórico.

6. Referências

FREITAS, Fernando. et al. Rotinas em Obstetrícia - 5ª Edição - Porto Alegre: Artmed Editora, 2006.

LECHETA, Ricardo R. Google ANDROID - Aprenda a criar aplicações para dispositivos móveis com o Android SDK - 2ª Edição - São Paulo: Novatec, 2010.

ROGERS, Rick. et al. Android Application Development, Sebastopol: O'REILLY, 2009.

VELTE, Anthony T.; VELTE Toby J.; ELSENPETER Robert. Cloud Computing: Computação em Nuvem - Uma Abordagem Prática - 1ª Edição - São Paulo: Alta Books, 2011.

ZUGAIBE, Marcelo. et al. Obstetrícia - 1ª Edição - Barueri, SP: Malone, 2008.

Acessos móveis em Operação e Densidade por UF. Disponível em:
<<http://sistemas.anatel.gov.br/SMP/Default.asp?SISQSmodulo=18228&SISQSsistema=172>>.
Acesso em: maio de 2012.

Android and iOS. Disponível em:
<<http://www.idc.com/getdoc.jsp?containerId=prUS23503312>>. Acesso em: Maio de 2012.

Consumidor pretende comprar mais smartphone do que celular comum.
Disponível em:
<<http://www.ibope.com.br/calandraWeb/servlet/CalandraRedirect?temp=5&proj=PortalIBOP&pub=T&db=caldb&comp=IBOPE+Inteligência&docid=A57B524940DAA4B5832579BA0054101D>>. Acesso em: Junho de 2012.

Google App Engine. Disponível em:
<<https://developers.google.com/appengine/>>. Acesso em: Junho de 2012.

Google Developers. Disponível em:
<<https://developer.android.com/>>. Acesso em: Junho de 2012.

Registro de novas linhas em Abril de 2012. Disponível em:

<<http://www.brasil.gov.br/noticias/arquivos/2012/05/17/anatel-registrou-2-2-milhoes-de-novas-linhas-de-celulares-em-abril>>. Acesso em: Abril de 2012.

SQLite. Disponível em:

<<http://sqlite.org/>>. Acesso em: Junho de 2012.