

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

MATHEUS DE OLIVEIRA JULLIEN

**Candy Castle:  
Um Jogo Sérió para Pacientes com Diabetes**

Trabalho de Graduação.

Prof. Dr. Marcelo Pimenta  
Orientador

Dipl.-Inf. Christoph Stach  
Co-orientador

Porto Alegre, dezembro de 2013.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Graduação: Prof. Sérgio Roberto Kieling Franco

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do Curso de Ciência da Computação: Prof. Raul Fernando Weber

Bibliotecário-Chefe do Instituto de Informática: Alexander Borges Ribeiro

# SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS .....</b>	<b>4</b>
<b>LISTA DE FIGURAS .....</b>	<b>5</b>
<b>LISTA DE TABELAS .....</b>	<b>6</b>
<b>RESUMO.....</b>	<b>7</b>
<b>ABSTRACT.....</b>	<b>8</b>
<b>1 INTRODUÇÃO.....</b>	<b>9</b>
1.1 Motivação .....	9
1.2 Objetivo .....	9
1.3 Estrutura do Texto .....	10
<b>2 FUNDAMENTOS .....</b>	<b>11</b>
2.1 Jogos.....	11
2.2 Jogos Educativos E <i>Serious Games</i> .....	11
2.3 Jogos Com Foco Na Saúde .....	12
2.4 Diabetes: Conceitos E Características Das Pessoas Portadoras.....	13
<b>3 TRABALHOS RELACIONADOS.....</b>	<b>15</b>
<b>4 CANDY CASTLE: UM JOGO PARA DIABÉTICOS .....</b>	<b>18</b>
4.1 Descrição Inicial.....	18
4.2 Arquitetura.....	19
4.3 Plataforma .....	20
4.4 Processo De Construção .....	20
<b>5 IMPLEMENTAÇÃO DO CANDY CASTLE .....</b>	<b>25</b>
5.1 Linguagem .....	25
5.2 Passos Para Implentação.....	25
5.2.1 Aplicativo Para Android .....	25
5.2.2 Servidor <i>Web</i> .....	31
<b>6 UM EXEMPLO DE USO DO CANDY CASTLE .....</b>	<b>33</b>
<b>7 CONCLUSÕES.....</b>	<b>42</b>
7.1 Contribuições .....	42
7.2 Limitações.....	43
7.3 Trabalhos Futuros .....	43
<b>REFERÊNCIAS.....</b>	<b>45</b>

## **LISTA DE ABREVIATURAS E SIGLAS**

ADT	Android Development Tools
ANR	Application Not Responding
API	Application Programming Interface
GBL	Game Based Learning
GPS	Global Positioning System
HDL	High Density Lipoproteins
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
JSP	JavaServer Pages
SDK	Software Development Kit

## LISTA DE FIGURAS

Figura 3.1: Imagens do jogo InsuOnline .....	15
Figura 3.2: Imagens do jogo Wii Fit Plus.....	16
Figura 3.3: Imagens do aplicativo Runtastic .....	17
Figura 4.1: Arquitetura do Candy Castle.....	20
Figura 4.2: Ciclo de prototipação do Candy Castle.....	21
Figura 4.3: Fluxograma do roteiro de jogo.....	22
Figura 4.4: Design da tela de jogo.....	23
Figura 4.5: Design da tela de jogo.....	23
Figura 4.6: Design da tela de <i>login</i> .....	24
Figura 6.1: Tela de <i>login</i> com o botão de começar uma partida .....	33
Figura 6.2: Tela de <i>login</i> com os botões de continuar partida e sincrozinar/trocar de jogador.....	34
Figura 6.3: Mapa sem nenhum ponto no mapa e centralizada no jogador .....	34
Figura 6.4: Tela de jogo com o balão para inserção do valor da medição da taxa de glicose no sangue do jogador .....	35
Figura 6.5: Mapa com o <i>overlay</i> do castelo .....	35
Figura 6.6: Mapa com <i>overlays</i> (castelo e torres) e muralha verde.....	36
Figura 6.7: Mapa com <i>overlays</i> (castelo e torres) e muralha amarela.....	36
Figura 6.8: Mapa com <i>overlays</i> (castelo e torres) e muralha vermelha.....	37
Figura 6.9: Tela de jogo com o balão de <i>game over</i> e muralha preta.....	38
Figura 6.10: Tela de jogo com o <i>toast</i> informando duração da partida, número de torres e pontuação do jogador.....	38
Figura 6.11: Janela de <i>login</i> com o Facebook .....	39
Figura 6.12: Janela de autorização ao Candy Castle se conectar ao perfil do jogador no Facebook.....	39
Figura 6.13: Janela de compartilhamento da pontuação do jogador no perfil do Facebook .....	40
Figura 6.14: Postagem da pontuação do jogador em seu perfil no Facebook .....	40
Figura 6.15: Página <i>web</i> do Candy Castle caso o jogador não possua dados no servidor .....	41
Figura 6.16: Página <i>web</i> do Candy Castle caso o jogador possua dados no servidor ....	41

## **LISTA DE TABELAS**

Tabela 3.1: Tabela comparativa entre o trabalho de Lange, Knöll e Diehl.....	16
--	----

## RESUMO

“Jogos sérios” tem primeiramente a função de educar e treinar, e em segundo lugar, divertir. Nesse trabalho é introduzido o Candy Castle, um “jogo sério” usado para o tratamento do diabetes e para o ensino de um estilo de vida consciente sobre o diabetes. Em linhas gerais, o jogo encoraja o jogador a caminhar ao seu redor e checar sua taxa de glicose no sangue em vários lugares diferentes. Um recurso importante do Candy Castle é que pacientes podem receber imediatamente um retorno de seus médicos sobre suas taxas de glicose no sangue, pois ambos (pacientes e médicos) podem facilmente acessar o *site* do Candy Castle para ver os valores das medições a qualquer hora.

Nesse trabalho são apresentados o projeto do jogo bem como alguns aspectos da sua implementação assim como um exemplo de uso.

**Palavras-chave:** aplicação mobile, aplicação android, jogo de saúde, educação de saúde, diabetes.

# **Candy Castle: A Serious Game for Patients with Diabetes**

## **ABSTRACT**

Serious games have primarily the function to educate and train, and secondly, entertain. This paper introduces the Candy Castle, a serious game used for the treatment of diabetes and for teaching a diabetes-aware lifestyle. In general, the game encourages the player to walk around in his/her surrounding and check the blood sugar level in several different places. An important feature of Candy Castle is that patients can immediately receive feedback their doctors about their blood sugar levels, because both (patients and doctors) can easily access Candy Castle's site to see the measurement values at anytime.

In this paper we present the game design as well as some aspects of its implementation as an example of using.

**Keywords:** mobile application, android application, health game, health education, diabetes.



# 1 INTRODUÇÃO

O capítulo aborda na seção 1.1 o que nos motivou à criação do Candy Castle e a seção 1.2 apresenta o objetivo que desejamos atingir com o aplicativo, já a seção 1.3 mostra ao leitor a estrutura do texto e o que encontrar em cada capítulo ao longo do trabalho.

## 1.1 Motivação

Tudo pode ser muito mais divertido com um pouco de imaginação!

Jogos não são apenas para puro entretenimento, mas podem ser grandes ferramentas para educar, ensinar ou treinar o jogador na vida real. Crianças gostam quando são estimuladas a realizarem uma tarefa de forma mais fácil ou mais prazerosa e esse estímulo pode muito bem vir através de jogos.

Esses jogos que servem a um propósito, que não apenas diversão, são chamados de *serious games* [1]. Eles fazem parte de uma área em franco crescimento e que está aberta para ser utilizada de diversas formas, como: educação e saúde.

Considerando apenas a área da saúde, “jogos sérios” tem a capacidade de prender a atenção do usuário com o objetivo de ajuda-lo a entender melhor suas condições de saúde e promover, dessa forma, uma melhor qualidade de vida [2].

Normalmente, os jogos de saúde são focados nos usuários de consoles ou computadores, restringindo muito o ambiente onde podem ser jogados, mas quando esses jogos vão para dispositivos moveis podem ser jogados a qualquer momento em qualquer lugar que o usuário desejar. Aproveitando-se da existência de características que um computador não oferece, como: GPS [3]; viemos oferecer uma alternativa aos jogadores que queiram cuidar de sua saúde também fora de casa e ter seu dispositivo móvel como principal aliado.

## 1.2 Objetivo

O objetivo do Candy Castle é ajudar crianças com diabetes a checarem com frequência sua taxa de glicose no sangue, pois a diabetes é uma doença crônica que vem crescendo ao longo dos anos. O aplicativo não se restringe apenas a crianças, mesmo essas sendo o público alvo, jovens e adultos também poderão se apoderar do Candy Castle como mais um auxiliar no combate a doença.

Tendo em vista que a diabetes tipo 2 é o tipo mais comum da doença e que sua ocorrência está diretamente ligada ao estilo de vida das pessoas, nós queremos encoraja-las a caminhar por lugar diferentes e a cada novo lugar fazer uma nova medição da sua taxa de glicose no sangue, pois dessa maneira além de estarem se movimentando (que pode em sequência virar um hábito saudável do paciente) elas estarão sempre com controle dos seus níveis glicêmicos ao longo do dia, seja em casa, no trabalho ou qualquer outro lugar que a pessoa esteja.

Toda pessoa com alguma doença precisa de um acompanhamento médico e esse é outro objetivo que nos propomos a atingir com o Candy Castle, pois todas as taxas de glicose no sangue medidas e inseridas pelo paciente dentro do aplicativo serão enviadas a um servidor *web* ao qual o médico terá acesso podendo fazer uma avaliação baseado nos valores informados pelo usuário e, imediatamente, contatar seu paciente se for preciso.

Com isso os cuidados com a diabetes serão mais frequentes facilitando a vida dos portadores da doença que não precisarão receber atendimento médico apenas quando marcarem suas consultas, mas terão esse auxílio a qualquer momento do dia. E é nesse aspecto que iremos diferenciar o Candy Castle dos jogos e aplicativos existentes no mercado, pois dificilmente um jogo ou aplicativo oferece a relação médico-paciente de forma não presencial.

### **1.3 Estrutura do texto**

No capítulo 2 apresentaremos conceitos do que são jogos (incluindo *serious games* e jogos de saúde) e uma breve explicação sobre diabetes.

O capítulo 3 será utilizado para mostrar alguns trabalhos (jogos e aplicativos) relacionados à saúde e à diabetes.

O Candy Castle será introduzido no capítulo 4 através de uma breve descrição inicial, depois no capítulo 5 explicaremos alguns dos principais aspectos relacionados à implementação do aplicativo e do *site*.

Mostraremos um exemplo de uso do Candy Castle com imagens no capítulo 6 a fim de demonstrar a utilização do jogo (aplicativo e *site*) para pacientes e médicos.

Finalmente, o capítulo 7 concluirá todo o estudo feito ao longo desse trabalho de conclusão de curso abordando alguns aspectos como contribuição do Candy Castle e trabalhos futuros a serem feitos no aplicativo.

## 2 FUNDAMENTOS

Nesse capítulo, apresentaremos os conceitos fundamentais à compreensão do trabalho. Dessa forma, a seção 2.1 resume algumas características de jogos, a seção 2.2 discute as noções de jogos educativos e de *serious games*, a seção 2.3 introduz ao leitor o gênero de jogos com foco na saúde e a seção 2.4 descreve os conceitos e as características das pessoas portadoras da diabetes.

### 2.1 Jogos

Um jogo é uma atividade estruturada, normalmente com finalidade de entretenimento e por vezes usada como ferramenta educacional. Jogos são distintos de trabalho (que possui como objetivo a remuneração pelo que é feito) e da arte (que possui, frequentemente, objetivos estéticos ou elementos ideológicos). Entretanto, essa distinção não é mais tão clara, pois há jogos que são considerados trabalho (com jogadores profissionais nos mais variados jogos) e há jogos que são considerados arte (como os quebra-cabeças que reproduzem uma imagem quando montados como se fossem um quadro ou uma pintura).

Jogos geralmente requerem esforço mental e/ou físico por parte dos jogadores e muitos ajudam a desenvolver habilidades práticas, servem como forma de exercício ou desempenham um papel educativo, simulacional ou psicológico.

Ao longo dos anos diversas definições sobre jogos surgiram e entre elas há a definição feita por Ludwig Wittgenstein, em seu livro *Philosophical Investigations* [4], onde ele argumenta que os elementos dos jogos, como entretenimento, regras e competição, fracassam ao tentar definir o que são jogos. Há também a definição dada por Roger Caillois, em seu livro *Les Jeux et Les Hommes* [5], onde jogos devem obrigatoriamente possuir algumas características descritas por ele e a tentativa de definição do que é um jogo usada por Chris Crawford através de uma série de dicotomias, em seu livro *Chris Crawford on Game Design* [6].

Podemos observar que mesmo com tantas definições diferentes sobre o que é jogo, as pessoas conseguem empregar a palavra com sucesso quando a usam e dessa forma ter uma definição única não se faz necessária.

### 2.2 Jogos Educativos E *Serious Games*

Jogos educativos são jogos explicitamente desenvolvidos com propósito educacional, ou que contenham valor educativo secundário ou acidental (caso não seja o foco principal, mas tenha um aprendizado ao longo do jogo). Os jogos educacionais são elaborados para ensinar as pessoas sobre algum assunto, expandir conceitos, reforçar o desenvolvimento, entender eventos históricos ou culturas, ou auxiliar no aprendizado de habilidades enquanto estiverem jogando e qualquer tipo de jogo pode ser usado num ambiente de ensino.

A aprendizagem baseada em jogo (GBL, em inglês) é um tipo de jogo que possui resultados de aprendizagem bem definidos. Geralmente, a aprendizagem baseada em jogo é elaborada a fim de equilibrar o assunto proposto com uma jogabilidade e a habilidade do jogador em reter e aplicar o que foi ensinado através do jogo no mundo real.

Um dos maiores exemplos envolvendo GBL são os simuladores de voo que além de auxiliarem no aprendizado também são necessários para quem deseja ser piloto de aeronave, pois algumas situações para a retirada do brevê são simuladas a fim de testar os conhecimentos do futuro piloto e é possível abater algumas horas de voo através da utilização de simuladores de voo.

Já um “jogo sério” (*serious game*, em inglês) é um jogo desenvolvido com um objetivo primário que não seja puramente diversão. O adjetivo “sério” é utilizado pela indústria de jogos numa alusão aos produtos que abordam temas ligados à área da defesa nacional, da educação, exploração científica, serviços de saúde, gestão de emergência, planejamento urbano, engenharia, religião e política.

*Serious game* são simuladores de eventos ou processos do mundo real projetados com o propósito de solucionar um problema, entretanto podem ser para entretenimento mesmo tendo em vista que o principal objetivo é treinar ou educar os jogadores (eles podem possuir outros objetivos como marketing ou propaganda). Os “jogos sérios” às vezes sacrificarão a diversão e o entretenimento a fim de alcançar o progresso desejável do usuário.

### **2.3 Jogos Com Foco Na Saúde**

Jogos com foco na saúde são um dos diversos tipos de *serious games* onde o objetivo é ensinar ao jogador como cuidar da sua saúde seja de forma teórica, aonde ao longo do jogo o usuário vai aprendendo sobre determinado assunto relacionado à saúde, ou prática, aonde o usuário irá praticar alguma atividade física juntamente com o jogo. Nesse cenário também se enquadrarão os aplicativos que são focados em saúde, porém os aplicativos serão mais teóricos do que práticos, pois não é possível se exercitar com um dispositivo móvel da mesma maneira que o jogador se exercitará caso utilize jogos de videogame focados em saúde.

Os jogos de saúde voltados ao público infantil e adulto visam uma conscientização por parte dos usuários sobre os mais diversos problemas de saúde e incentivar as crianças a se interessarem mais por um estilo de vida não sedentário. Essa conscientização e incentivo não irão substituir os tradicionais métodos para se cuidar da saúde, como o acompanhamento de um médico especialista, e a prática de exercícios físicos, sejam eles feitos em aparelhos próprios para a prática da atividade ou através da prática de esportes.

E é nesse ponto, a falta do acompanhamento médico, que a maioria dos jogos de saúde peca, pois há um excelente conteúdo contido nos jogos e aplicativos, mas sem a presença de um especialista não há garantia de que o jogador estará aprendendo e utilizando o conhecimento da forma correta. Em rápida pesquisa feita sobre jogos e aplicativos de saúde, a maioria permite um acompanhamento médico desde que seja feito de forma presencial, ou seja, o usuário não tem total liberdade na utilização do jogo ou aplicativo, pois o mesmo acaba servindo apenas para autocontrole do que está sendo feito (seja no controle da alimentação, na quantidade de exercícios físicos ou informações corpóreas (peso, medidas, taxas, etc.)).

## 2.4 Diabetes: Conceitos E Características Das Pessoas Portadoras

A diabetes é uma doença metabólica onde a pessoa possui alta quantidade de açúcar ou glicose no sangue, decorrente da insuficiente produção de insulina pelo pâncreas ou decorrente das células não responderem à insulina que é produzida.

Há três tipos principais de diabetes:

**Diabetes tipo 1:** Causada pela incapacidade do corpo em produzir insulina e requer ao paciente injeções de insulina ou usar uma bomba de insulina. Encontrada principalmente em crianças.

**Diabetes tipo 2:** Causada pela resistência a insulina, uma condição onde as células fracassam ao usar a insulina de forma apropriada, por vezes combinado com uma deficiência absoluta de insulina. Sendo o tipo mais comum da doença. Encontrada principalmente em adultos.

**Diabetes gestacional:** Ocorre quando a gestante, sem nenhum diagnóstico prévio de diabetes, apresenta uma alta taxa de açúcar ou glicose no sangue. Podendo acarretar no desenvolvimento da diabetes tipo 2. Pode melhorar ou desaparecer após a gestação, mas requer cuidadosa supervisão médica durante toda a gravidez.

Os “outros tipos específicos” de diabetes são um conjunto de algumas poucas causas individuais e são menos de 5% de todos os casos diagnosticados.

Os sintomas clássicos da diabetes não tratada são: perda de peso, poliúria (aumento do volume urinário), polidipsia (sede aumentada e aumento de ingestão de líquidos) e polifagia (apetite aumentando). Esses sintomas podem se desenvolver mais rapidamente (semanas ou meses) em pessoas com diabetes tipo 1, enquanto podem se desenvolver mais lentamente e serem sutis ou ausentes em pessoas com diabetes tipo 2. Também pode haver problemas de visão decorrentes da alta quantidade de glicose no sangue.

Há fatores de risco que podem facilitar o desenvolvimento da diabetes que são: idade acima de 45 anos, obesidade, histórico familiar de diabetes (parentes de 1º grau), diabetes gestacional ou microsomia prévia, hipertensão arterial, colesterol HDL abaixo de 35 mg/dl e/ou triglicérides acima de 250 mg/dl, alterações prévias da regulação de glicose e indivíduos membros de população de risco (negros, hispânicos, escandinavos e indígenas).

As causas da diabetes variam de acordo com cada tipo. No tipo 1, o fator genético é mais influente ao surgimento da doença que é então desencadeada por certas infecções e dessa forma se desenvolvendo mais cedo (atingindo crianças e adolescentes justamente pelo fator genético). Já no tipo 2, o fator genético já não é tão influente quanto no tipo 1, pois o desenvolvimento da diabetes tipo 2 depende também de fatores relacionados ao estilo de vida da pessoa, ou seja, mesmo que a pessoa tenha uma genética favorável, hábitos saudáveis poderão prevenir ou adiar o aparecimento da doença.

A diabetes é caracterizada pela hiperglicemia recorrente ou persistente e a pessoa é diagnosticada com a doença caso apresente qualquer um dos seguintes itens após fazer o teste: nível plasmático de glicose em jejum maior ou igual a 126 mg/dl (7 mmol/l) em duas ocasiões, nível plasmático de glicose maior ou igual a 200 mg/dl (11,1 mmol/l) duas horas após ingerir uma dose de 75g de glicose anidra em duas ocasiões e nível plasmático de glicose aleatória maior ou igual a 200 mg/dl (11,1 mmol/l) associados a sinais e sintomas típicos de diabetes.

Não será necessário fazer o reteste caso o paciente apresente os sintomas característicos da doença. No caso de pacientes grávidas, um nível de glicose acima de 110 mg/dl em jejum ou de 140 mg/dl após ingerir 75g de glicose já é suficiente para indicar diabetes gestacional.

A diabetes é uma doença crônica, sem nenhuma cura conhecidas exceto em situações muito específicas. O tratamento consiste em manter a taxa de glicose no sangue o mais perto do normal (euglicemia) possível, sem causar hipoglicemia. Esse quadro pode ser alcançado com dieta, exercícios e o uso de medicação apropriada (insulina no caso da diabetes tipo 1 e medicação oral, como possivelmente insulina, no caso da diabetes tipo 2). É extremamente importante à educação do paciente, o acompanhamento de sua dieta, exercícios físicos e monitoração própria de suas taxas de glicose, com o objetivo de manter as taxas de glicose no sangue a curto e longo prazo adequadas. O controle cuidadoso é necessário para reduzir os riscos das complicações em longo prazo.

### 3 TRABALHOS RELACIONADOS

Como os jogos, normalmente, engajam e motivam os jogadores a atingirem um objetivo, essas características podem ser utilizadas na saúde para ajudar no tratamento de doenças crônicas [7].

Por exemplo, Lange et al. [8] desenvolveu um jogo onde pacientes em pós-operatório farão exercícios de respiração ao controlarem um pássaro usando somente a sua respiração. O terapeuta configura os níveis, de acordo com os exercícios respiratórios adequados e recebe resultados quantitativos relativos ao progresso e comprometimento do paciente.

Martin Knöll [9] apresentou diferentes abordagens de jogos, de como ajudar crianças que sofrem de diabetes, tendo em vista que a doença está se tornando uma grande epidemia. Ele introduziu a ideia de jogo para auxiliar o jogador a realizar regularmente o teste que mede a taxa de glicose no sangue, desse modo, é possível proporcionar um *feedback* compreensível imediatamente. Como efeito positivo, o médico responsável terá melhor e mais rápido acesso ao paciente assim como terá informações de saúde avaliadas e agrupadas; atualmente, as medições são anotadas em diários feitos a mão (sendo eles frequentemente incompletos ou ilegíveis e dessa forma acabam sendo inúteis), na maioria das vezes.

No Brasil, Leandro Diehl [10] desenvolveu um jogo, InsuOnline (figura 3.1), focado em educar profissionais da saúde sobre como atender corretamente pacientes com diabetes. Ao longo do jogo, o jogador passará por diversas situações ao atender os pacientes com diabetes e terá que prescrever a utilização adequada de insulina em cada caso se baseando nos diagnósticos mostrados em cada fase.



Figura 3.1: Imagens do jogo InsuOnline

Tabela 3.1: Tabela comparativa entre o trabalho de Lange, Knöll e Diehl

	Lange	Knöll	Diehl
Público Alvo	Pacientes em pós-operatório	Crianças com diabetes	Médicos do Sistema Único de Saúde
Objetivo	Auxiliar os pacientes a fazerem exercícios para evitar complicações respiratórias	Auxiliar as crianças a medirem a taxa de glicose no sangue regularmente	Auxiliar os médicos a atender corretamente pacientes com diabetes
Plataforma	Aparelho de espirometria + dispositivo	Aparelho de medição da taxa de glicose + <i>Mobile</i>	Computador
Local de Jogo	Ambulatório	Qualquer lugar	Restrito (casa, escritório, etc.)

Não podemos esquecer-nos de mencionar os jogos e aplicativos de saúde com foco em atividades físicas que já são bem conhecidos, como: Wii Fit Plus (figura 3.2) - onde o usuário fará movimentos sobre o Wii Balance Board que simularão atividades como yoga, treinos de força, aeróbica e equilíbrio, repetindo o exercício que aparecerá no jogo, o jogo tem sido utilizado para reabilitação física e também em casas de repouso para melhorar a postura de idosos; Runtastic (figura 3.3) - série de aplicativos para diversas atividades físicas que faz o monitoramento dos exercícios realizados pelo usuário, ele armazenará informações como tempo, distancia, velocidade, gasto calórico, entre outros dados e os exibirá no dispositivo móvel ao usuário de forma de tabelas e gráficos.

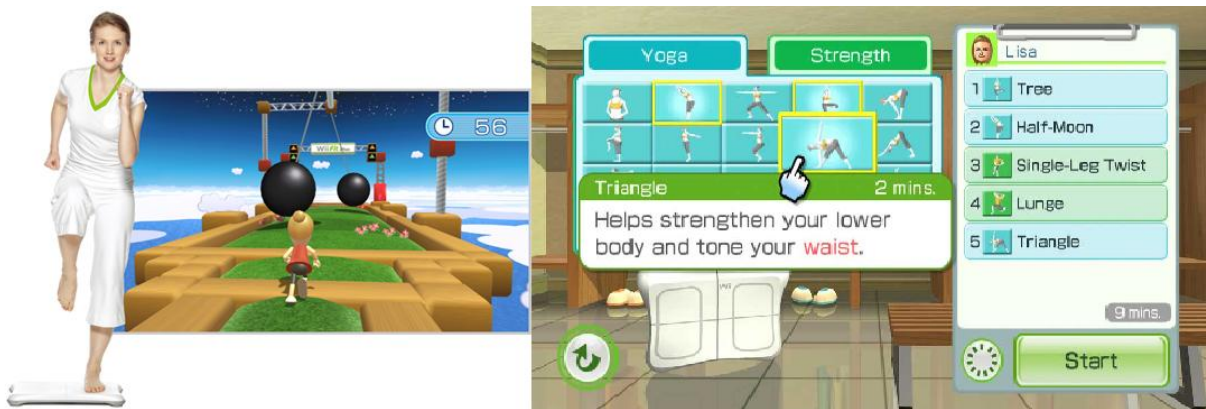


Figura 3.2: Imagens do jogo Wii Fit Plus



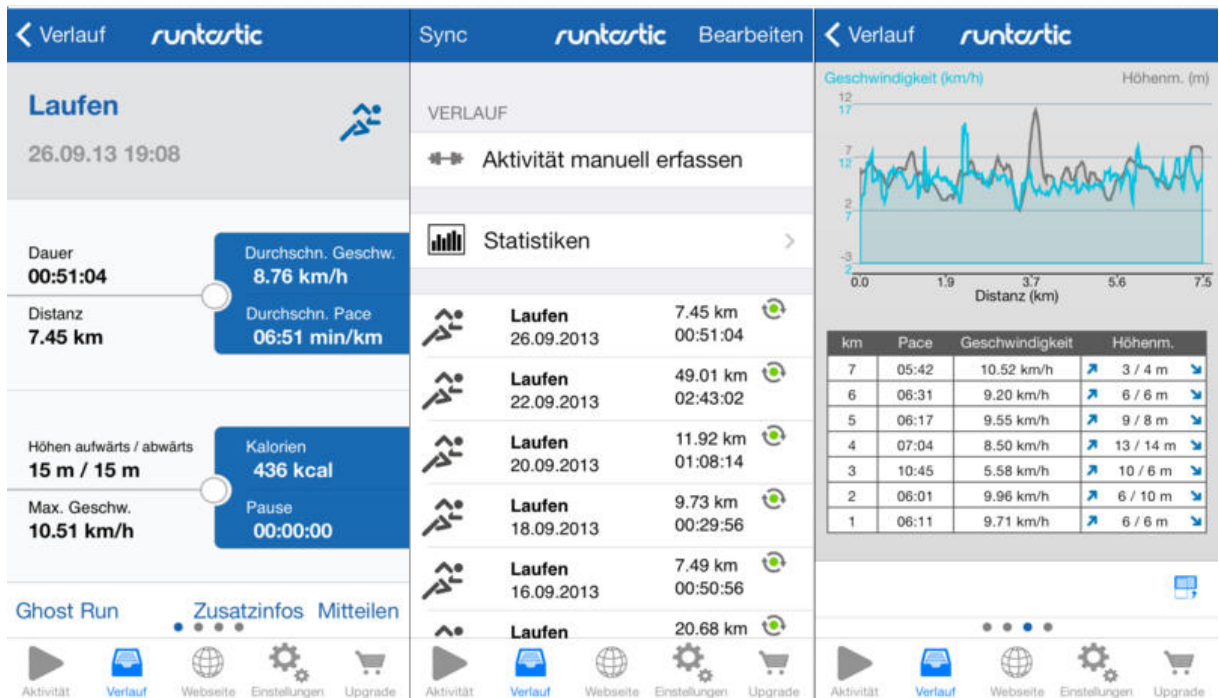


Figura 3.3: Imagens do aplicativo Runtastic

No Brasil, é difícil encontramos jogos relacionados à saúde, pois a maioria são apenas aplicativos voltados para alimentação e exercícios físicos. Esses aplicativos servem apenas de auxílio ao usuário no controle de suas atividades, como se fossem um diário do que o usuário come ou pratica. Os focados na alimentação elaboram planilhas baseadas na ingestão de alimentos do usuário, onde o próprio informa ao aplicativo o que está ingerindo, e dessa forma, havendo erro na hora de informar alguma quantidade o maior prejudicado é o usuário. Já os focados em exercícios físicos, monitoram as atividades do usuário e as apresentam na forma de gráfico ou tabela, alguns possuem instruções mais detalhadas dos exercícios que o usuário poderá fazer, mas sem um acompanhamento médico adequado na prática de alguma atividade física o usuário estará sujeito a prejudicar sua saúde e propenso a lesões.

Porém todos os exemplos dados anteriormente acabam não tendo o médico diretamente ligado ao paciente, pois os dados ficam retidos no jogo/aplicativo e são apenas visíveis ao usuário. A partir desse fato não é possível avaliar corretamente se o usuário está obtendo o resultado adequado com o uso do jogo/aplicativo e se está atingindo corretamente o objetivo proposto pelo jogo/aplicativo, pois toda atividade que envolve a saúde humana deverá sempre ter um acompanhamento médico.

## 4 CANDY CASTLE: UM JOGO PARA DIABÉTICOS

O aplicativo surgiu a partir da adaptação da ideia original criada por Knöll chamada Candy Castle, onde a criança era dona de um castelo virtual, e fizemos algumas melhorias e mudanças na ideia do jogo. Enquanto a versão original do Candy Castle pretende dar às crianças o papel de estilista e a riqueza dos moveis representa a sua saúde, nós gostaríamos de dar a elas um alcance maior de atuação: o jogo atribui os dados de diabetes do jogador a locais da vida real.

Candy Castle é um aplicativo para celular voltado para crianças com diabetes. Ele é um jogo no qual o usuário deverá manter seu castelo protegido por meio de torres que serão aumentadas e muralhas que serão fortificadas a cada nova medição da taxa de glicose no sangue que o usuário fizer. Na primeira medição um castelo será criado e nas medições seguintes torres e muralhas serão criadas para proteger o castelo de ataques. À medida que o número de medições for aumentando a pontuação final do usuário também aumentará de acordo com os critérios estabelecidos pelo jogo.

Os principais componentes dos jogos são: objetivos, regras, desafios e interações. Com relação ao Candy Castle o objetivo será pontuar o máximo possível, não haverá nenhuma regra específica em relação a como deve ser jogado, porém o usuário deverá seguir a indicação do seu médico quanto ao número de medições por dia, o desafio será manter o jogo ativo o maior tempo possível e, conseqüentemente, pontuar mais e as interações serão entre o jogador e o seu dispositivo utilizado para jogar (*smartphone* ou *tablet*).

O jogo foi desenvolvido com o intuito de facilitar a interação paciente-médico através de uma maneira não usual (por meio de um jogo utilizando como base o conceito de *gamefication* [11]) de monitoramento do nível de glicose do paciente, porém o tradicional método de medição por meio de equipamentos (por exemplo, produtos da Accu-Chek<sup>®</sup> ou da OneTouch<sup>®</sup>) não fica descartado, pois o paciente precisará fazer a medição e passar os valores ao aplicativo.

Os valores inseridos no aplicativo, além de serem mantidos no banco de dados do jogo no celular, de tempos em tempos, serão enviados a um servidor online ao qual o médico terá total acesso às taxas de glicose de seus pacientes, ou seja, com o Candy Castle o usuário não precisará entrar em contato com seu médico frequentemente para passar suas taxas de glicose e o médico poderá fazer um monitoramento do paciente de forma imediata com valores recentes.

### 4.1 Descrição Inicial

O aplicativo possui duas telas básicas nas quais dados do usuário serão necessários para que o jogo tenha seu pleno funcionamento.

A primeira tela, ou tela de *login* ou tela inicial, é onde o usuário deverá informar seu e-mail (que o médico deverá ter conhecimento) e a senha para efetuar o *login* e obter as funcionalidades para inserir e buscar dados do servidor. Outras opções disponíveis na tela inicial depois de efetuado o primeiro *login* serão: continuar partida, trocar usuário e sincronizar usuário.

Continuar partida permite ao usuário continuar o jogo a partir do ponto que ele parou (número de muralhas, tempo decorrido de jogo e pontuação). Trocar e sincronizar usuário são funções similares por terem a mesma função, a diferença é apenas conceitual (trocar significa que um novo usuário começará uma partida, sincronizar significa que o usuário atual continuará uma partida já iniciada anteriormente no mesmo dispositivo ou em outro dispositivo), pois ambas necessitam do e-mail e senha do usuário para serem executadas. A função sincronização existe, pois há a possibilidade de um usuário jogar uma partida num celular e logo em seguida começar a jogar em outro celular sem ter terminado a primeira (essa sincronização ocorre também na primeira vez que um usuário utiliza o aplicativo para começar uma partida).

A segunda tela, ou tela do jogo, será a tela que representará o jogo propriamente dito, é nessa tela que o usuário colocará o valor da sua taxa de glicose no sangue e construirá seu castelo, torres e muralhas sobre um mapa. O castelo será criado sempre na primeira inserção de valor pelo paciente e as torres nas medições seguintes, o posicionamento do castelo e das torres é baseado na posição do usuário (utiliza-se o GPS do celular para determinar a coordenada geográfica e colocar as peças no lugar correspondente).

Por se tratar de um jogo, na segunda tela o usuário terá algumas outras funções como: pontuação, número de torres, tempo de jogo, área construída e ainda postar essas informações nas redes sociais ao término da partida. Ao final de uma partida os dados, ainda marcados como não enviados, serão enviados ao servidor e uma nova partida poderá ser começada sem que dados antigos sejam perdidos.

## 4.2 Arquitetura

Pacientes e médicos utilizarão o Candy Castle através de seu dispositivo móvel. Ao entrarem com seu e-mail e senha ou escolhendo uma conta já vinculada ao dispositivo no aplicativo, ele fará o *login* do jogador e se conectará ao servidor usando os dados do jogador e, após efetuar o *login* e estabelecer uma conexão junto ao servidor, o Candy Castle buscará os dados do usuário para verificar se há novos dados provenientes de um jogo iniciado em outro dispositivo ou se será necessário atualizar o jogo existente (caso outro jogador tenha utilizado o dispositivo atual) e caso haja informações novas elas serão carregadas no mapa automaticamente.

O envio dos dados ao servidor se dará sempre que o jogo for pausado ou ao término de uma partida, serão enviadas apenas as informações que estiverem no banco de dados local do aplicativo marcados como não enviados (dessa forma evitamos enviar dados antigos diversas vezes ao servidor sem necessidade). Após ter as informações enviadas ao servidor, elas estarão imediatamente disponíveis ao médico para serem analisadas (o paciente também terá acesso às suas informações). O médico terá acesso a todos os dados dos seus pacientes (utilizando o e-mail do paciente que ele deseja visualizar os dados).

O servidor e o *site* do Candy Castle formam construídos e hospedados utilizando o Google App Engine, pois usando o App Engine teremos além da infraestrutura escalável do Google um serviço de armazenamento persistente de larga escala. Além disso, será possível utilizar

contas vinculadas ao Google para se conectar ao servidor sem fazer uso de nenhuma aplicação *web* para autenticação das contas.



Figura 4.1: Arquitetura do Candy Castle

### 4.3 Plataforma

Candy Castle foi desenvolvido para celulares que rodem Android como sistema operacional. É necessário que a versão Android seja igual ou superior a 2.3 (Gingerbread) para todas as funções do Candy Castle funcionarem.

O aplicativo também requer que o dispositivo móvel possua acesso à internet (para comunicação com o servidor e exibição das imagens do mapa) e GPS (para posicionamento dos dados no mapa baseado na localização do dispositivo).

Não há nenhuma restrição referente às configurações mínimas do celular, como tamanho da tela, resolução, para que o Candy Castle opere, mas em alguns dispositivos a imagem pode ficar distorcida por não haver uma configuração que englobe todos os dispositivos que operam com Gingerbread ou superiores.

Já o usuário precisará ter uma conta vinculada ao Google para fazer *login* no jogo e poder se comunicar com o servidor hospedado pelo App Engine do Google.

### 4.4 Processo de Construção

A construção do Candy Castle foi feita de forma gradual, ou seja, cada funcionalidade do aplicativo foi devidamente implementada e testada e só depois incorporada ao aplicativo definitivamente. Assim, conseguimos uma maior autonomia entre as partes de compõem o Candy Castle, pois uma funcionalidade mal desenvolvida não atrapalharia o funcionamento das demais partes do aplicativo. Cada protótipo foi desenvolvido e elaborado para que não haja perda de tempo na construção do aplicativo, ou seja, não haverá grandes mudanças nas funcionalidades já existentes nas versões anteriores em relação à versão sendo implementada.

O processo de construção utilizado foi a prototipação, pois havia uma versão inicial a ser desenvolvida do aplicativo e ao longo do desenvolvimento surgiram novas funcionalidade a serem implementadas após a conclusão da versão inicial. Cada protótipo desenvolvido

continha alguma evolução em relação ao anterior seja no layout, reimplementação de funções existentes ou implementação de novas funções.

O primeiro protótipo desenvolvido já apresentava as funções relacionadas ao mapa bem implementadas: o aplicativo já localizava o dispositivo móvel do usuário e já adicionava os *overlays* sobre o mapa (através da entrada de dados por parte do jogador que também já estava implementada).

Na segunda versão, o acréscimo realizado foi a criação do banco de dados local do Candy Castle. Dessa forma, cada vez que o jogador inserisse o valor da sua taxa de glicose no sangue o *overlay* criado sobre o mapa seria automaticamente adicionado ao banco de dados.

Com a criação do banco de dados, desenvolvemos as funções que tornariam o aplicativo num jogo: primeiramente, a função que seria responsável por desenhar as muralhas ligando cada uma das torres do jogo e secundariamente, o temporizador em *background* responsável por trocar a cor das muralhas e determinar quando o jogador deveria inserir um novo dado. A partir dessa versão, o Candy Castle já estava utilizável pelos usuários, pois já apresentava as suas principais funções desenvolvidas. Ainda nesse terceiro protótipo do jogo adicionamos a pontuação do jogador, pois ela se baseia apenas entre o horário atual e o horário de inserção da primeira torre e, como já temos a condição de *game over*, nós precisávamos dar algo em troca ao jogador por inserir sua taxa de glicose no sangue regularmente.

No quarto protótipo, desenvolvemos o servidor *web* para duas funcionalidades: receber os dados dos jogadores para serem exibidos no *site* e enviar os dados ao dispositivo móvel do usuário com a intenção de sincronizar as informações caso o jogador tenha iniciado uma partida em outro dispositivo. Com o servidor *web*, implementamos no aplicativo a função de conexão com o servidor para receber os dados à sincronização do dispositivo e também a função que enviará os dados do paciente ao servidor.

A conectividade com o Facebook só foi adicionada ao Candy Castle no quinto protótipo do aplicativo e através dela o jogador poderá compartilhar a pontuação obtida no jogo com seus amigos da rede social. Sendo esse protótipo a versão atual do aplicativo.

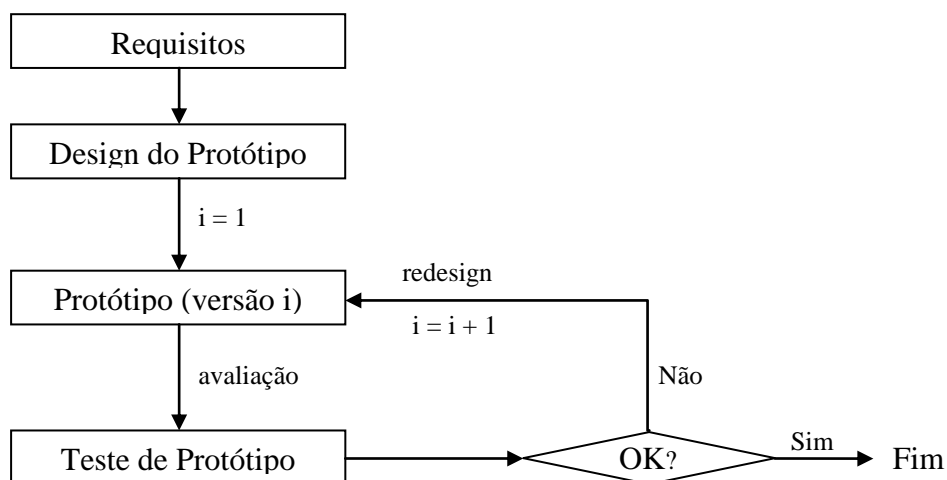


Figura 4.2: Ciclo de prototipação do Candy Castle

Pensamos durante a prototipação no roteiro que o usuário deverá seguir ao longo do jogo e na figura 4.3 podemos visualizar o roteiro elaborado em formato de fluxograma. O roteiro é apenas uma sugestão das ações que o usuário deve seguir a fim de obter um melhor uso e desempenho usando o Candy Castle.

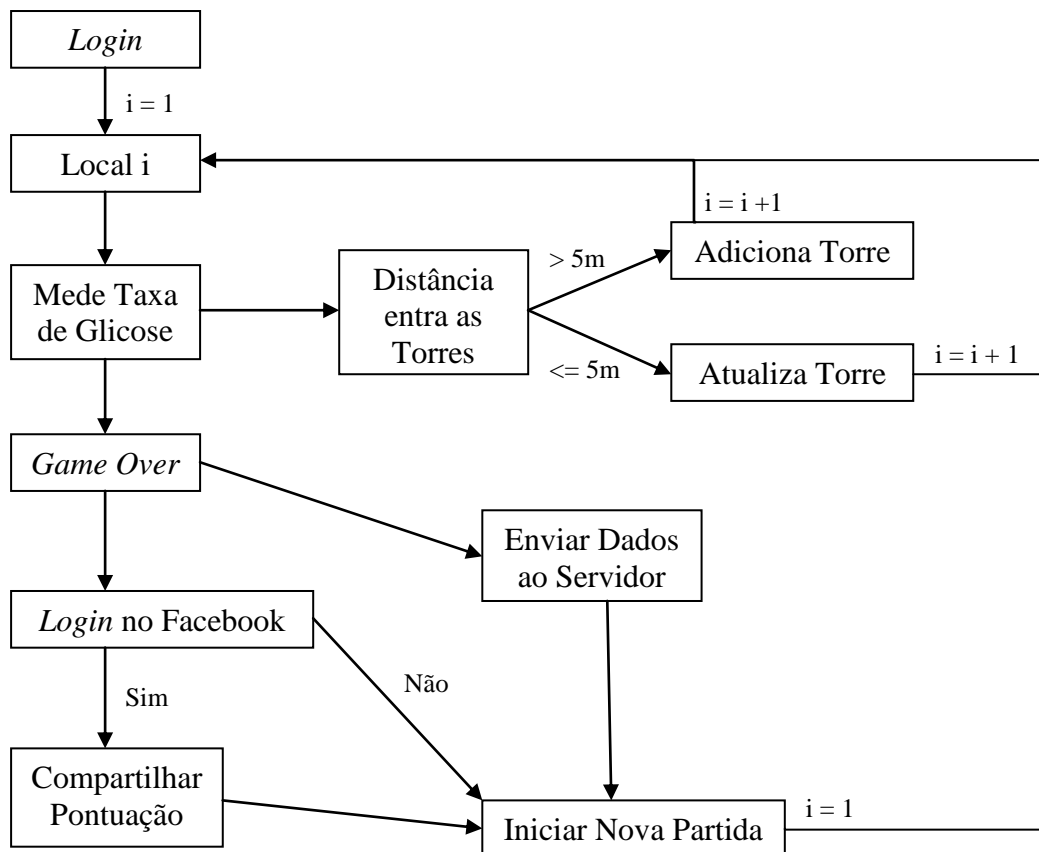


Figura 4.3: Fluxograma do roteiro de jogo

Como o fluxograma da figura 4.3 é apenas uma sugestão, o jogador poderá fazer outras ações não demonstradas no roteiro planejado, como: pausar o jogo e consequentemente enviar os dados ao servidor, trocar de usuário durante ou ao final de uma partida sendo necessário um novo *login*, continuar uma partida iniciada em outro dispositivo recebendo os dados da partida do servidor ou visualizar sua pontuação durante a partida.

O design da tela de jogo como podemos ver nas figuras 4.4 e 4.5 foi planejado para que todos os botões ficassem nas bordas e que o meio fosse ocupado com as informações visuais mais importantes, como: balão para inserção do valor da taxa de glicose, balão com a mensagem de *game over*, *toast* com a pontuação do jogador e a mais importante delas são os *overlays* e a muralha entre as torres.

Já o design da tela de *login* é bem simplista agrupando a área de *login* (entrada de dados) na parte superior da tela (ocupando espaço proporcionalmente a quantidade de contas vinculadas ao dispositivo móvel) e os botões para acessar o jogo na parte inferior, inicialmente terá apenas um botão para iniciar uma partida e após a primeira utilização do aplicativo a tela de *login* ficará igual como visto na figura 4.6.

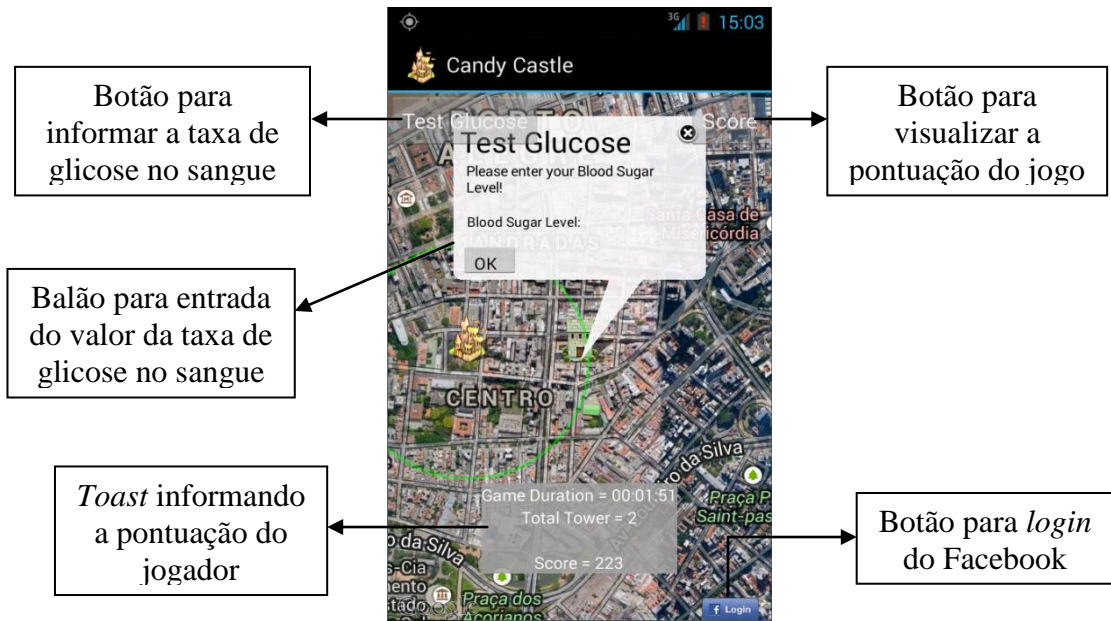


Figura 4.4: Design da tela de jogo

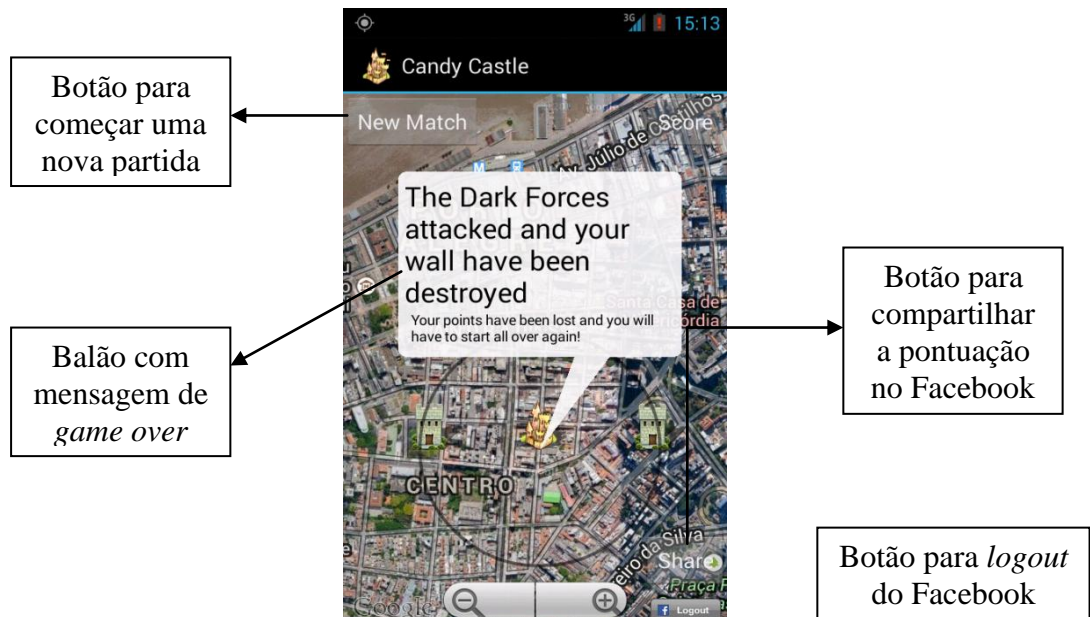


Figura 4.5: Design da tela de jogo

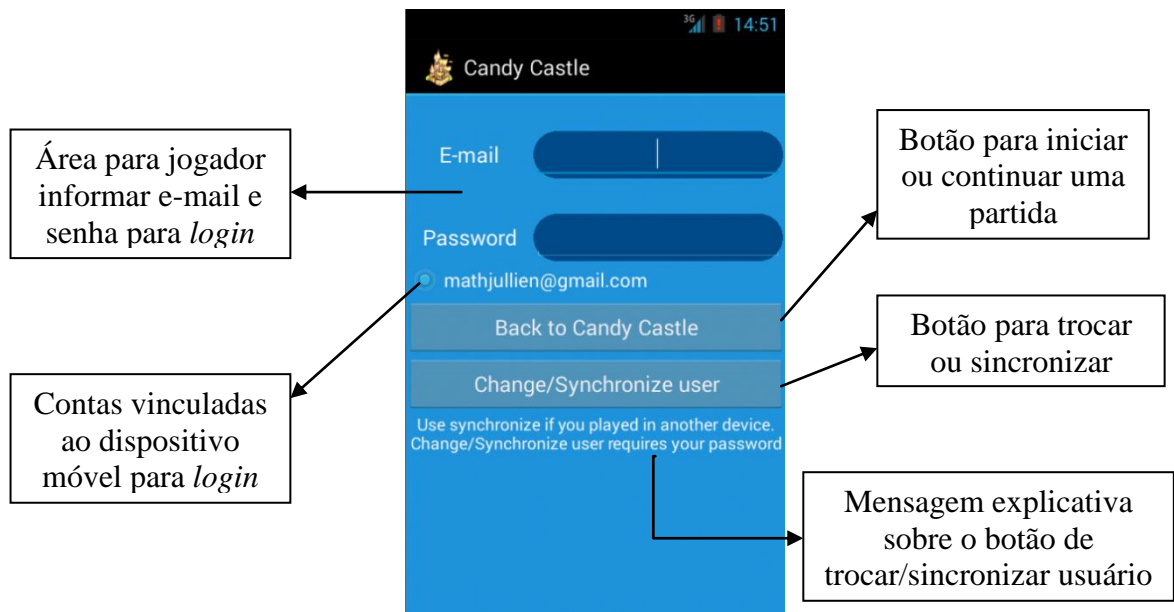


Figura 4.6: Design da tela de *login*



## 5 IMPLEMENTAÇÃO DO CANDY CASTLE

No capítulo, abordaremos os principais aspectos referentes à implementação do Candy Castle. A seção 5.1 mostra as linguagens utilizadas no aplicativo e servidor *web*, já a seção 5.2 resume alguns dos passos da implementação Android e *web* do aplicativo.

### 5.1 Linguagem

O aplicativo Candy Castle foi implementado usando a linguagem de programação Java e utilizando o *plugin* ADT para desenvolvimento de aplicativos Android no Eclipse.

O servidor *web* foi criado com o auxílio do Google App Engine usando Java como linguagem de programação nos *servlets* (comunicação aplicativo-servidor) e JSP/HTML nas páginas *webs* (as quais apenas o médico terá acesso).

### 5.2 Passos Para Implementação

Nessa seção teremos os principais passos do desenvolvimento do Candy Castle para Android assim como os passos do desenvolvimento no servidor *web*.

#### 5.2.1 Aplicativo Para Android

A subseção 5.2.1 resume algumas das funcionalidades que juntas compõem o aplicativo para Android e, assim, facilitar a compreensão do leitor sobre as funções implementadas.

##### 5.2.1.1 Mapa e Localização do Usuário

A criação do aplicativo começou pelo mapa e para tal foi utilizada a primeira versão da API do Google Maps para Android. Com a API foi possível ter algumas funcionalidades do Google Maps dentro do aplicativo (utilizaremos apenas as funcionalidades de zoom, recursos para localização do usuário e adição de pontos no mapa (*overlays*)).

Para criarmos o mapa foi necessário adicionar a biblioteca do Google Maps e requisitar acesso do Candy Castle a internet, ambos foram feitos dentro do arquivo *AndroidManifest.xml* do projeto. A biblioteca possibilita adicionar o mapa ao aplicativo juntamente com as funcionalidades citadas anteriormente nessa subseção, já o acesso à internet é necessário para que as imagens do mapa sejam recebidas e exibidas ao usuário (o acesso à internet também será necessário a outras funcionalidades do aplicativo que iremos abordar em seções e subseções seguintes).

A localização do usuário é adquirida com a utilização de um *listener* que irá “escutar” qualquer modificação na localização geográfica do aparelho através da função *requestLocationUpdates()* e aobtermos a ultima localização disponível devemos usar a função *getLastKnownLocation()*. Tanto para requisitar as atualizações da localização geográfica como aobter a ultima localização conhecida é preciso escolher um provedor diretamente (sem utilizar critérios na hora de passar como parâmetro da função) ou indiretamente (escolhendo

através de critérios e passando o resultado dessa escolha como parâmetro). Há dois provedores que podem ser utilizados nas funções para obter a localização: GPS e/ou Network, e ambos precisam de permissão para que o aplicativo tenha acesso à localização geográfica do aparelho através dos provedores. Cada um dos provedores tem suas vantagens e desvantagens e mediante as características dos provedores o aplicativo fará uso apenas do provedor GPS, pois o mesmo apresenta um resultado mais preciso na obtenção da localização do usuário [12].

### 5.2.1.2 *Overlays E Linhas Conectando As Torres*

Overlays são os pontos que iremos adicionar sobre o mapa toda vez que o jogador inserir um novo valor da sua taxa de glicose no sangue, pontos que precisarão de uma coordenada geográfica proveniente da localização do usuário e serão ou em formato de castelo, caso seja o primeiro valor inserido pelo jogador na partida, ou em formato de torre, no caso dos demais valores inseridos na partida. Esses pontos que tornarão o aplicativo num jogo propriamente dito, pois é a partir deles juntamente com o tempo de jogo que o *score* do usuário será calculado.

Para que os *overlays* possam ser colocados e desenhados no mapa é necessária à criação de uma classe que irá manipulá-los antes de os adicionarmos ao jogo. Essa classe estenderá a classe *ItemizedOverlay*, nela iremos ter o construtor da classe (função que irá tratar a imagem que será vinculada ao *overlay* criado) e criaremos a função de adicionar um *overlay* a lista de *overlays* (função que também servirá para deixar o *overlay* preparado para ser desenhado no mapa, pois haverá uma chamada à função *populate()* da classe *ItemizedOverlay*).

Com os pontos no mapa precisaremos ligá-los desenhando algumas linhas sobre o mapa também. E como traçar tais linhas de modo que não se cruzem e formem uma “muralha de proteção” ao castelo do jogador? Para responder tal pergunta precisaremos considerar dois fatores: como ordenar os pontos e como traçar as linhas.

O segundo fator é mais fácil de ser resolvido, pois só precisaremos criar duas classes (uma para desenhar a muralha em formato de círculo caso tenhamos uma ou duas torres no mapa e outra para desenhar a muralha em formato de polígono caso tenhamos três ou mais torres) que estenderão a classe *Overlay* e executarão a função de desenho de acordo com os dados recebidos (quantidade de torres que temos no mapa).

Já ao primeiro fator a solução ao desenho da muralha em formato de círculo é muito simples, pois precisaremos apenas de um centro, que será o castelo do jogador, e da distância entre o castelo e a torre mais distante em relação ao mesmo e com essas duas informações é só chamar a função *drawCircle* da biblioteca gráfica do Android e teremos a muralha circular no mapa perfeitamente desenhada.

Porém para desenhar a muralha poligonal será necessário à utilização de algum algoritmo para ordenarmos os *overlays* de forma a não ocorrerem cruzamentos de linhas e que todos os pontos sejam ligados. Para tal adaptaremos um algoritmo de *convex hull* conhecido como *Graham Scan* [13] (o nome faz referência ao seu criador, Ronald Graham), pois o algoritmo original apenas leva em consideração os pontos mais externos para criar um polígono e a solução desejada ao problema apresentado é uma que englobe todas as torres do jogo e não apenas algumas. Primeiramente teremos que escolher a torre mais ao sul (menor latitude) e caso haja duas ou mais torres com a mesma latitude iremos também levar em consideração a que estiver mais ao leste (menor longitude) como nosso ponto inicial e final e será a partir desse ponto que iremos ordenar os demais para formarmos a muralha poligonal desejada. Feita essa escolha iremos ordenar as demais torres de acordo com o ângulo polar da torre em

relação ao ponto inicial, ou seja, em ordem crescente de ângulo polar, mas caso haja duas torres com o mesmo ângulo teremos que fazer algumas escolhas para decidir qual ficará mais no começo da lista e essas escolhas levarão em conta a latitude e a longitude dos pontos.

A adaptação feita em relação ao *Graham Scan* foi à parte de escolher os pontos, pois o algoritmo original verifica se os pontos estão em sentido horário ou anti-horário e acaba excluindo do polígono os pontos que não estiverem no sentido escolhido pelo programador, já na adaptação essa verificação foi retirada aobtermos um polígono completo com todos os *overlays* do mapa como vértices. Com todas as torres já ordenadas é só chamar criar o caminho (passado como parâmetro) que a função *drawPath* da biblioteca gráfica do Android deverá seguir para termos a muralha poligonal desenhada sobre o mapa.

### 5.2.1.3 Armazenamento De Dados

Com tantos pontos sobre o mapa será necessário armazenar esses dados, para usarmos durante o jogo e para serem enviados ao servidor, caso o aplicativo saia do primeiro plano e para tal utilizaremos um banco de dados interno referente ao jogo. Sem esquecer que precisaremos passar informações entre as duas telas do jogo através de um *Intent* e guardar o valor de algumas variáveis de forma mais simples que inserir e buscar no banco de dados, nesse caso utilizaremos a classe *SharedPreferences* que nos permitirá salvar e recuperar valores de variáveis de tipos primitivos (*string*, *boolean*, *int*, *float* e *long*).

O banco de dados onde guardaremos os pontos sobre o mapa será implementando utilizando *SQLite*, pois o Android oferece suporte completo a banco de dados *SQLite*. Ele ficará acessível a todas as classes do Candy Castle, porém nenhum outro aplicativo conseguirá acessar o banco de dados o que nos garantirá uma integridade aos dados que iremos manipular no jogo.

A modelagem do banco de dados será feita estendendo a classe *SQLiteOpenHelper* e dentro da subclasse iremos definir quais serão os campos existentes no banco de dados (não sei se é preciso dizer quais são os campos da tabela), o nome e a versão, essas informações serão utilizadas posteriormente para criarmos, instanciarmos e abriremos o banco de dados para manipulação dos dados.

Faremos uso de uma classe onde teremos todas as funções para criar, abrir e manipular o banco de dados do aplicativo. Primeiramente utilizaremos a função que criará um objeto que ajudará na criação, abertura e manipulação dos dados, esse objeto receberá como parâmetros o contexto da aplicação, o nome do banco de dados, o cursor (no caso passaremos *null* como valor, pois utilizaremos o cursor *default* da classe *SQLiteOpenHelper*) e a versão caso seja necessário atualizar o banco de dados. Com o objeto criado, precisaremos de função que realmente crie e abra o banco de dados para que possamos executar os comandos de manipulação e faremos isso chamando a função *getWritableDatabase()*, pois executaremos comandos de leitura e escrita sobre o banco de dados.

Com o banco de dados criado e aberto a manipulações de dados, só precisaremos criar as funções de inserção, atualização e busca. A função de inserção receberá todos os dados e os inserirá na tabela, já as funções de atualização de algum dos valores da tabela receberá apenas o dado a ser atualizado. Teremos uma função de inserção e uma de atualização que serão utilizadas no mesmo contexto dentro do aplicativo na hora de adicionarmos uma torre no mapa, quando formos adicionar uma torre no jogo iremos verificar se ela está a, no mínimo, cinco metros de distância de todas as outras torres caso esteja iremos criar um novo *overlay* no mapa, porém caso não esteja iremos apenas atualizar a taxa de glicose e o tempo vinculados à torre. Já as outras funções de atualização serão para sinalizar os *overlays*

contidos no banco de dados que fazem parte do jogo atual ou que são de partidas já acabadas, os que já foram ou não enviados ao servidor e há também uma que irá atualizar todos os dados atuais da partida em vigor caso o aplicativo seja pausado (chamada da função *onPause()*).

O *Intent* será usado para fazer a troca da tela de *login* à tela de jogo e nessa transição passaremos três dados: o e-mail do usuário, o *token* sinalizando se o usuário já está ou não conectado no Facebook e a *string* com os dados que foram buscados no servidor web. A classe *SharedPreferences* é a classe que trata todas as preferências do usuário, no nosso caso ela será utilizada apenas na tela inicial para identificar o jogador através do seu e-mail e guardar a *string* com os dados obtidos do servidor e na tela de jogo a utilizaremos para guardar *tokens* de sessão referentes a conexão do usuário com o Facebook. Ao obter um objeto da classe *SharedPreferences* usaremos o método *getPreferences()*, pois teremos apenas um arquivo com as preferências do usuário no jogo, na hora de salvar as preferências precisaremos criar um *Editor* que receberá todas as informações através de métodos *put* e enviará ao arquivo executando o comando *commit()* e a leitura desses dados será feita através de métodos *get* sobre o objeto obtido, não sobre o *Editor*.

### 5.2.1.3 Funções Em Background E Separadas Da Thread Principal

Dentro do aplicativo teremos algumas funções importantes ao jogo rodando em *background* e outras rodando em separado da *thread* principal, pois não podemos sobrecarregar a *thread* principal do Candy Castle a fim de evitar que o aplicativo trave causando um ANR [14] ao levar muito tempo para responder a um comando do usuário na tarefa primária. As funções relacionadas ao servidor, obrigatoriamente, serão tarefas assíncronas rodando em *background* e por esse motivo precisaremos tratar da sincronização com a *thread* principal do aplicativo após sua execução, já as funções que demandem mais tempo de execução e recursos e as que possam ser executadas em *threads* secundárias iremos separar da tarefa principal para não causarmos um ANR indesejável (essas tarefas por não rodarem em *background*, mas apenas em separado, terão ligação com a *thread* primária através das variáveis globais do aplicativo e é possível executá-las em *background* escolhendo a prioridade da tarefa através da função *android.os.Process.setThreadPriority()*).

Teremos duas tarefas assíncronas para comunicação com o servidor, a primeira conterá funções de *login* e busca de dados e a segunda enviará informações novas ao servidor. Essas *threads* serão criadas estendendo a classe *AsyncTask* da qual sobre-escreveremos os métodos *doInBackground()* (método executado em *background*, como o próprio nome já diz) e *onPostExecute()* (método executado na *thread* primária após a computação em *background* terminar), no primeiro faremos toda a interação junto ao servidor e no segundo, dependendo do resultado da execução do método anterior, alguma ação será tomada seja com a troca de contexto (da tela de *login* à tela de jogo) com armazenamento e passagem de variáveis ou com a execução de uma função de atualização do banco de dados do jogo (marcando quais dados já foram enviados ao servidor).

As funções que serão executadas como tarefas separadas terão como objetivo não sobrecarregar a *thread* principal, porém elas ainda podem causar ANR por não estarem sendo executadas em *background* (no caso do Candy Castle, nos testes feitos durante e após a criação do aplicativo as funções que poderiam causar ANR foram corrigidas de forma a evitarmos esse problema). A principal função que iremos executar como tarefa secundária será a que determina o tempo de jogo (condição para *game over* incluída) e desenha as muralhas ligando as torres. Essa função do aplicativo demanda o processamento de muitos dados em um curto período de tempo e mesmo assim apenas executando ela como *thread* já

foi o suficiente para não travar o Candy Castle, pois ela será executada a cada 1s após a primeira torre do jogo ser adicionada e a cada execução a função deverá verificar, de acordo com o tempo decorrido desde a adição da última torre, se a muralha deve trocar de cor (indicando que o usuário deve fazer uma nova medição tanto para melhor análise da sua taxa de glicose por parte do médico como para continuar o jogo) e, juntamente com a verificação, adicionar a muralha ao mapa.

#### 5.2.1.4 Conexões Com O Servidor

As conexões entre aplicativo e servidor ocorrerão antes, durante e depois de uma partida e todas serão tarefas assíncronas como descrito anteriormente utilizando HTTP como protocolo de comunicação.

A primeira conexão que faremos antes de uma partida servirá para efetuar o *login* do usuário junto ao servidor (será preciso que o usuário autorize que o Candy Castle se comunique com o Google App Engine) e após confirmação do *login* buscar informações no banco de dados do servidor a respeito do jogador (informações que servirão para sincronizar o jogo ou caso o usuário tenha dados novos provenientes de uma partida feita em outro dispositivo *mobile* que não o atual ou caso seja um novo usuário que precise recuperar seus dados do servidor ao dispositivo atual e continuar sua partida). Ao *login* passaremos como parâmetros da *url* o e-mail do usuário, a página para redirecionamento após efetuar o *login* e um *token* de autenticação para conectar ao Google App Engine baseado no e-mail fornecido, já como resposta iremos apenas guardar um *cookie* que utilizaremos no lugar do *token* de autenticação quando formos buscar os dados do servidor. Com o *cookie* já adquirido, o adicionaremos como *header* para conectar com a página que nos fornecerá as informações de jogo do usuário e também passaremos como parâmetro o e-mail do jogador, pois é como cada paciente é identificado no servidor na hora de inserir ou buscar dados, o resultado dessa busca será uma *string* em formato JSON que faremos *parser* para identificar se há algum dado para ser utilizado no jogo antes de iniciarmos a partida.

A segunda conexão com o servidor poderá ocorrer durante o jogo (caso o usuário feche o aplicativo) e ocorrerá no final do jogo (caso o usuário perca a partida, ou seja, *game over*) e servirá para enviar as medições de glicose do usuário junto com informações do jogo ao servidor. A função de envio será mais simples que a busca, pois não será necessário efetuar *login* e muito menos a utilização de *token* e/ou *cookie* para autenticar a conexão. Ao envio iremos passar como parâmetros o e-mail do jogador, seu número de identificação no jogo (*user id*), o valor da taxa de glicose medida, a localização geográfica da torre que representa a medição, quando a medição foi feita e o número de identificação da partida (*game id* não único, pois será baseado nas partidas do usuário e não em todas as partidas do jogo). Executaremos essa função de envio para todos os dados ainda não enviados ao servidor (há uma função de busca no banco de dados do jogo para identificar tais dados) e após o envio de todos os dados atualizaremos o banco de dados do aplicativo para não enviarmos dados repetidos ao servidor.

Como há acesso às contas de e-mail registradas no dispositivo e também autenticação das mesmas para *login* no Google App Engine é preciso requisitar permissão para que o Candy Castle possa acessar e autenticar contas de e-mail e essas permissões serão feitas dentro do arquivo *AndroidManifest.xml* assim como fizemos para ter acesso a internet e acesso para buscar a localização do usuário.

#### 5.2.1.5 Tela Inicial

A principal funcionalidade da tela inicial será fazer o *login* do usuário com o servidor tanto para usuários com conta salva no dispositivo móvel quanto para novos usuários (que poderão fazer o *login* informando e-mail e senha que serão passados para conectar ao servidor). Nela será possível que o jogador ou continue ou comece ou sincronize uma partida no dispositivo, ambas as funcionalidades feitas através de botões correspondentes com a função que exercem.

Na tela inicial aparecerão todas as contas de e-mail do Google salvas no dispositivo para que o jogador escolha com qual conta deseje jogar (não será preciso a digitação da senha, pois a função que gera o *token* de autenticação acessa as informações necessárias através da conta que é passada como parâmetro). O *login* através de uma conta já existente terá prioridade em relação a uma conta nova, ou seja, caso o jogador preencha e-mail e senha e ao mesmo tempo escolha uma conta existente a conexão com o servidor ocorrerá com as informações da conta que o usuário escolheu e o jogo se iniciará tendo o e-mail da conta escolhida como identificador do jogador.

No primeiro acesso do usuário ao Candy Castle haverá apenas o botão de começar uma partida, pois o botão de continuar uma partida leva em consideração que haja um e-mail já salvo na variável que representa o e-mail do jogador. Após esse primeiro acesso (caso o usuário comece uma partida) já aparecerá o botão de continuar uma partida juntamente com o botão de ou trocar de usuário ou sincronizar ao seu lado (ambas as opções desse botão utilizam a mesma função, pois ambas farão *login* e busca de dados do usuário junto ao servidor) e também haverá uma observação explicando ao que serve a sincronização e informando que é necessário informar e-mail e senha (caso seja um jogador não registrado no dispositivo).

Ao clicar em algum botão, o usuário será imediatamente redirecionado a tela de jogo e assim começar a sua partida.

#### 5.2.1.6 Conectividade Com O Facebook

A conectividade com o Facebook permitirá que o usuário poste sua pontuação do jogo diretamente na sua página pessoal da rede social e para utilizarmos os métodos necessários para garantir o *login* do jogador adicionaremos a SDK versão 2.0 do Facebook para Android como biblioteca do aplicativo.

Primeiramente iremos criar uma variável para lidar com todas as funções do Facebook e para tal precisaremos registrar o Candy Castle, pois é necessário passar como parâmetro do construtor da variável o identificador do aplicativo. Feito isso poderemos chamar as funções de *login*, de *logout* e de postar no mural, serão apenas essas funções que utilizaremos e a única permissão que requisitaremos ao site será para postar no mural (nenhuma informação do jogador será adquirida pelo Candy Castle).

Haverá um botão no canto inferior direito da tela do jogo para que o usuário faça seu *login* e seu *logout* e outro botão logo acima, que aparecerá apenas ao final de uma partida caso o jogador tenha feito seu *login* no Facebook, para postagem da pontuação na rede social.

Ao clicar no botão de *login* chamaremos a função *authorize()* que abrirá uma janela para que o jogador coloque seus dados para se conectar ao Facebook, logo após a confirmação será pedido ao usuário permissão para que o Candy Castle tenha acesso ao seu Facebook (será necessário essa permissão para que o aplicativo possa postar o *score* do jogo no mural do jogador) e ao final desse processo a janela fechará e o jogo continuará normalmente. Serão salvas duas informações referentes ao *login* do usuário: um *token* de autenticação e o tempo de expiração da sessão.

O botão de *logout* desconectará o usuário imediatamente do Facebook através de uma tarefa assíncrona para não bloquear a *thread* que chamou a função *logout()*, também irá invalidar o *token* de autenticação modificando seu valor para *null* e mudará o tempo de expiração da seção para 0 (segundo a documentação referente ao tempo de expiração o valor 0 pode representar que não existe uma seção ou que a seção não expira, no caso do aplicativo representará que não existe uma seção, pois sempre que conectarmos o usuário ao Facebook o tempo de expiração será de 24 horas).

Quando o jogador clicar no botão de compartilhar (*share*) faremos uma chamada a função *dialog()* onde passaremos como um dos parâmetro a ação a ser executada, no caso passaremos a *string* “*feed*” como parâmetro, que abrirá uma janela para que o usuário escreva uma mensagem e escolha quem poderá ver a postagem. Na descrição aparecerá a pontuação do usuário, o tempo de jogo e uma imagem do Candy Castle, também haverá a informação de que o compartilhamento foi via aplicativo.

#### 5.2.1.7 Funcionalidade Dos Botões

Já explicamos anteriormente a funcionalidade dos botões da tela inicial e dos botões correspondentes às ações relacionadas à conectividade com o Facebook, mas há mais dois botões do jogo que necessitam de uma breve explicação de suas funcionalidades.

O primeiro deles, e principal do jogo, será o botão que abrirá a janela onde o jogador colocará o valor da sua taxa de glicose no sangue. Ao clicar no botão, o usuário estará buscando a localização do seu dispositivo e ao informar o valor estará adicionando informações ao banco de dados do jogo que serão enviadas ao servidor posteriormente e também irá adicionar ao mapa um *overlay* de castelo ou torre baseado na sua localização contendo o valor da medição.

O segundo botão é o que chama a função que irá abrir um breve *toast* sobre o mapa informando a pontuação, número de torres já adicionadas ao mapa, tempo decorrido de jogo e área total coberta pelas muralhas do jogador. Caso a partida termine, antes do usuário começar uma nova partida, será possível que ele clique nesse botão para receber as informações citadas anteriormente.

## 5.2.2 Servidor Web

A subseção 5.2.2 descreve a implementação das duas partes que compreendem o servidor *web* do Candy Castle, a primeira visível apenas ao aplicativo e a segunda visível aos pacientes e médicos.

### 5.2.2.1 Funcionalidade Dos Servlets

A interação aplicativo servidor ocorrerá através dos *servlets*, pois será através deles que iremos inserir os dados no servidor e buscar os dados do servidor em seguida. O *servlet* é acessado pelo aplicativo como uma página *web* recebendo os parâmetros por método *post* ou *get* (programamos os dois *servlets* do projeto para receberem as informações apenas por *post*, pois dessa forma os dados não aparecerão na *url* do site e dificultará a inserção de dados falsos que só será possível se descobrirem a página do *servlet* e os parâmetros enviados a mesma). Ambos *servlets* serão uma extensão da classe *HttpServlet* para que possamos fazer utilização do método (*post* ou *get*) desejado.

O *servlet* dedicado à inserção de dados simplesmente receberá todas as informações enviadas pelo aplicativo a serem inseridas no banco de dados do servidor, criará uma chave

única baseada no e-mail do usuário (cada jogador terá sua chave correspondente) e utilizaremos a chave como parâmetro para criar um objeto da classe *Entity* que conterá as informações a serem adicionadas no servidor.

Já o *servlet* responsável pela busca receberá apenas o e-mail do usuário, pois precisaremos criar uma chave (será a mesma criada no *servlet* de inserção, isso irá garantir que buscaremos os dados correspondentes ao jogador) e passar a mesma como parâmetro do objeto da classe *Query* que fará a busca pelas informações do jogador junto ao banco de dados do servidor. O resultado da busca retornará como uma *string* em formato JSON ao aplicativo que fará o *parser* da *string* para identificar se há algum dado a ser utilizado no jogo, pois a busca apenas retornará informações referentes à última partida do jogador e por isso o *parser* se fará necessário para verificar se a partida ainda é válida, ou seja, ainda não passou do tempo estipulado para dar *game over*.

#### 5.2.2.2 Páginas Web

A página *web* está extremamente simples, no momento, pois sua funcionalidade será apenas exibir os dados dos pacientes ao médico (apenas o médico terá acesso às informações de seus pacientes). Ela terá a mesma implementação de busca que o *servlet* de busca, pois ela foi programada utilizando JSP e, ao invés de retornamos os dados ao aplicativo, utilizaremos HTML para exibi-los na tela. Como o servidor está funcionando apenas em *localhost* para acessar os dados será necessário colocar o e-mail do usuário desejado na *url* da página *web*, pois como explicado anteriormente à busca se baseará no e-mail fornecido.



## 6 UM EXEMPLO DE USO DO CANDY CASTLE

Depois de termos já uma visão geral com uma breve introdução do que é o aplicativo Candy Castle e sua finalidade e também de termos abordado a implementação das principais partes do jogo, neste capítulo iremos descrever um passo a passo de um cenário de jogo, ou seja, a sequência de ações que o usuário/jogador fará ao longo do jogo juntamente com o que o médico irá encontrar ao acessar o *site* para obter informações do usuário.

Como já foi mencionado anteriormente, o jogador precisará ter uma conta vinculada ao Google (estando ela salva no dispositivo onde o aplicativo rodará ou não) e acesso à internet para que o Candy Castle possa se comunicar com o servidor. Com esses dois pré-requisitos atendidos o usuário poderá usufruir tranquilamente do jogo.

Nas figuras 6.1 e 6.2 teremos a área de entrada dos dados para *login* que poderá ser através de e-mail e senha inseridos pelo usuário ou através de uma conta salva no dispositivo (o usuário poderá escolher uma das contas salvas que aparecerão na tela). Após escolher um dos dois métodos, na figura 6.1 o jogador terá apenas a opção de começar uma partida, pois é a primeira vez que o aplicativo estará sendo executado depois de ter sido instalado, já na figura 6.2 o jogador terá duas opções para começar a jogar: no botão a esquerda ele escolherá continuar uma partida, dessa forma não será necessário inserir dados nem escolher uma conta, e no botão a direita ele escolherá continuar uma partida já iniciada em outro dispositivo ou iniciar uma partida com um jogador diferente do que estava jogando, nesse caso será necessário inserir e-mail e senha ou escolher uma das contas salvas no aparelho.

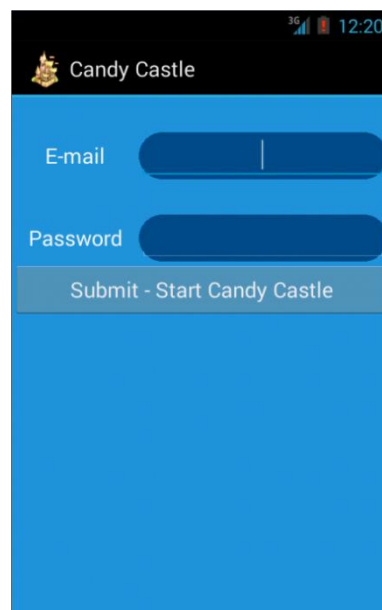


Figura 6.1: Tela de *login* com o botão de começar uma partida

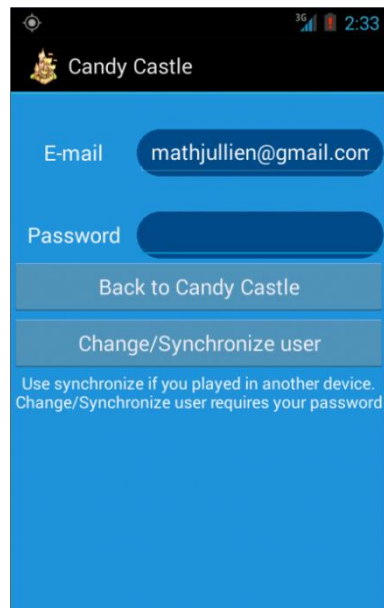


Figura 6.2: Tela de *login* com os botões de continuar partida e sincrozinar/trocar de jogador

Quando o *login* do jogador for concluído com sucesso imediatamente a tela com o mapa aparecerá no dispositivo como podemos observar na figura 6.3 e nela teremos dois botões inicialmente: o primeiro, localizado no canto superior esquerdo, será o responsável por abrir a janela para que o usuário insira o valor da sua taxa de açúcar no sangue e o segundo, localizado no canto inferior direito, será o botão ao jogador se conectar com o Facebook e compartilhar a sua pontuação na rede social.



Figura 6.3: Mapa sem nenhum ponto no mapa e centralizada no jogador

Na figura 6.4 temos a janela que abrirá após o jogador clicar no botão para inserir a taxa de glicose, nela o usuário informará o valor da medição que será feita através de algum aparelho capacitado para tal função e quando clicar no botão *ok* a janela fechará e um *overlay* será adicionado ao mapa.

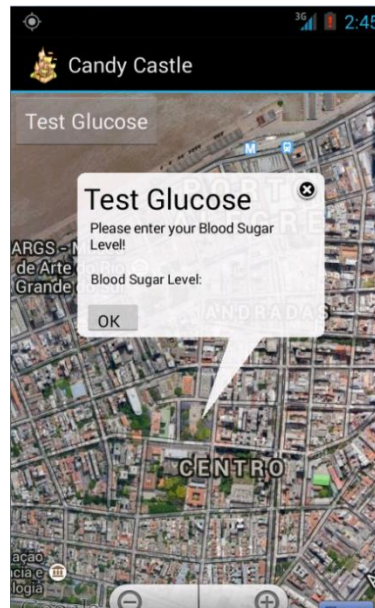


Figura 6.4: Tela de jogo com o balão para inserção do valor da medição da taxa de glicose no sangue do jogador

Nas figuras 6.5, 6.6, 6.7 e 6.8 teremos como será a aparência do jogo à medida que o usuário for inserindo novos valores com a sua taxa de glicose (como na figura 6.4). Nela temos a figura do castelo, primeiro *overlay* a ser inserido, e as torres, demais *overlays* inseridos após o castelo (ressaltando que as muralhas aparecerão apenas após a primeira torre ser adicionada ao mapa). Quando tivermos dois ou mais pontos adicionados ao mapa, no canto superior direito, aparecerá o botão ao jogador ter acesso a sua pontuação. À medida que o jogo for avançando as muralhas trocarão de cor (verde, amarelo e vermelho) como indicativo de que a partida se aproxima do final exigindo do paciente a medição e inserção de um novo valor da sua taxa de glicose no sangue para que a partida continue e as muralhas voltem à cor verde.



Figura 6.5: Mapa com o *overlay* do castelo

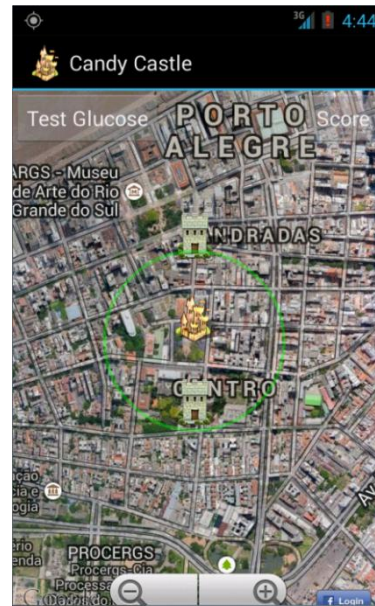


Figura 6.6: Mapa com *overlays* (castelo e torres) e muralha verde



Figura 6.7: Mapa com *overlays* (castelo e torres) e muralha amarela



Figura 6.8: Mapa com *overlays* (castelo e torres) e muralha vermelha.

Depois de um determinado tempo sem inserir nenhum ponto no mapa, ou seja, sem medir sua taxa de glicose e informar ao jogo, o jogador perderá a partida e na figura 6.9 o cenário de *game over* está sendo mostrado. O usuário será avisado que perdeu a partida através de um balão com uma mensagem relacionada, a muralha mudará para a cor preta e aparecerá o botão para começar uma nova partida no lugar do botão de inserir uma nova medição.

Ao término de uma partida, o usuário terá a opção de começar uma nova partida sem ter que voltar a tela de *login* e escolher uma conta ou inserir qualquer informação (e-mail e senha) e essa opção aparecerá na tela na forma de botão. O botão para começar uma nova partida ficará no canto superior esquerdo da tela no lugar anteriormente ocupado pelo botão de inserir uma nova medição da taxa de glicose no sangue. Quando o jogador clicar no botão, ele limpará a tela de jogo (removendo todos os *overlays*) e zerarão as informações referentes à partida anterior (pontuação, número de torres, tempo de jogo) deixando a tela igual à figura 6.3, as únicas informações mantidas para a nova partida serão o usuário que está jogando e o *login* com o Facebook, caso tenha sido feito, pois como a nova partida será com o mesmo jogador da anterior e não há motivos alterarmos essas duas informações.



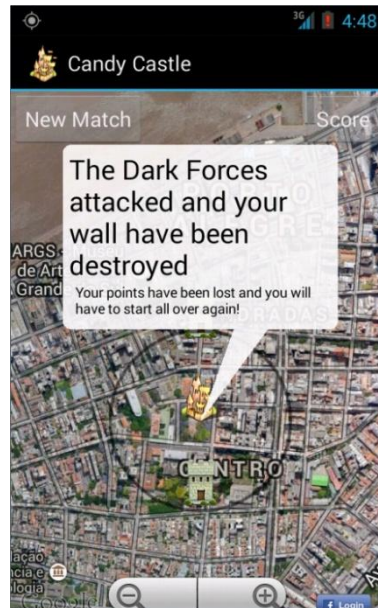


Figura 6.9: Tela de jogo com o balão de *game over* e muralha preta

Após adicionar duas ou mais torres ao jogo, o jogador poderá acessar informações como pontuação, número de torres e pontuação clicando no botão que estará no canto superior esquerdo. Quando o usuário clicar no botão aparecerá uma *toast* que nada mais é do que uma janela de duração limitada que surgirá contendo alguma mensagem ou aviso, no caso do Candy Castle essa mensagem será a pontuação do usuário, o tempo decorrido de jogo e o número de torres já adicionadas ao mapa.



Figura 6.10: Tela de jogo com o *toast* informando duração da partida, número de torres e pontuação do jogador

Já na figura 6.11 podemos ver a *webview* que abrirá contendo a página onde o usuário fará seu *login* no Facebook e posteriormente poderá compartilhar sua pontuação, se desejar. No botão do Facebook aparecerá escrito *login* caso não haja jogador conectado na rede social e *logout* caso haja jogador conectado. Quando o jogador clicar no botão *login* a *webview* se abrirá, já quando clicar no *logout* o jogador será desconectado automaticamente sem a

*webview* se abrir. Caso seja a primeira vez que o jogador conecta-se ao Facebook pelo Candy Castle aparecerá, após efetuar o *login* na rede social, uma nova *webview* alertando ao usuário sobre quais informações de seu perfil no Facebook que o aplicativo terá acesso e se ele deseja autorizar o Candy Castle a ter acessos a essas informações (vide figura 6.12).

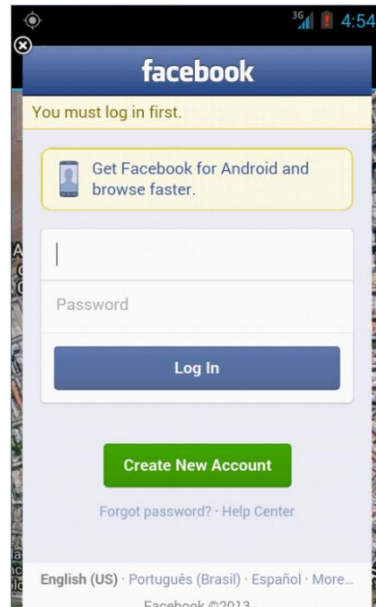


Figura 6.11: Janela de *login* com o Facebook



Figura 6.12: Janela de autorização ao Candy Castle se conectar ao perfil do jogador no Facebook

Caso o jogador tenha se conectado ao Facebook, quando a partida terminar, ou seja, aparecer à mensagem de *game over*, um botão aparecerá na tela de jogo acima do botão associado ao *login* e *logout* da rede social. Esse novo botão será responsável por compartilhar a pontuação do usuário em sua *timeline* no Facebook, ao clicar no botão *share* aparecerá o *toast* com a pontuação do usuário e logo em seguida uma janela ao usuário postar essa pontuação no Facebook também aparecerá, nela o usuário fará as mesmas ações como se estivesse postando algo em seu mural pelo site da rede social, porém por essa postagem ser

feita através do Candy Castle algumas informações aparecerão junto com o posto, como o nome do aplicativo, uma imagem do aplicativo, a pontuação do jogador e o tempo de jogo e mais abaixo a origem do *post* (via Candy Castle), mas o usuário poderá colocar uma mensagem pessoal e selecionar a privacidade da postagem (por *default* aparecerá como postagem pública).



Figura 6.13: Janela de compartilhamento da pontuação do jogador no perfil do Facebook



Figura 6.14: Postagem da pontuação do jogador em seu perfil no Facebook

O paciente e o médico poderão consultar as taxas de glicose enviadas ao servidor a qualquer momento, caso alguma informação já tenha sido enviada pelo jogador. Jogadores que não possuem dados no servidor exibirão mensagem avisando sobre a falta de informação e os que possuem os exibirão na tela, como mostrado nas figuras 6.15 e 6.16.



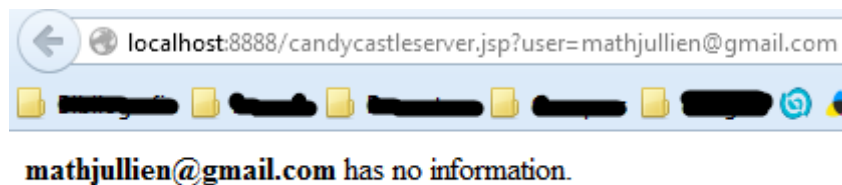


Figura 6.15: Página web do Candy Castle caso o jogador não possua dados no servidor



Figura 6.16: Página web do Candy Castle caso o jogador possua dados no servidor

## 7 CONCLUSÕES

Nesse capítulo, apresentaremos as conclusões obtidas com o Candy Castle. Assim, a seção 7.1 aborda as contribuições fornecidas pelo aplicativo aos usuários, a seção 7.2 resume as limitações impostas a utilização do Candy Castle e a seção 7.3 apresenta as melhorias a serem feitas no aplicativo e no servidor *web*.

### 7.1 Contribuições

Os aplicativos apresentados no capítulo 3 juntamente com os demais aplicativos existentes no mercado focados em saúde, bem-estar e exercícios físicos possuem em comum a mesma característica principal: monitoramento do usuário, ou seja, de acordo com o que o usuário faça o aplicativo apenas armazenará dados para controle e os exibirá na tela do dispositivo móvel com isso não há uma participação direta de algum especialista na área que o aplicativo se propõe a focar, dessa forma o usuário precisará ir atrás de um acompanhamento periódico seja de um médico e/ou de um profissional da área (nutricionista para aplicativos relacionados à alimentação, *personal trainer* ou professor de educação física para aplicativos relacionados a exercícios físicos, etc.).

Aproveitando a falta de aplicativos onde o usuário consiga se comunicar rapidamente com seu médico para receber o devido acompanhamento de sua saúde é que o Candy Castle se encaixa, pois nele (como explicado em capítulos anteriores) o médico terá total acesso às taxas de glicose do paciente ao longo do dia podendo fazer uma avaliação da situação do seu paciente e dando amparo imediato caso seja necessário fazer alguma correção simples no tratamento do usuário para que sua taxa de glicose no sangue volte ao normal.

Candy Castle, por ser um aplicativo com enfoque maior no público infantil, foi implementado para ser um jogo, pois dessa forma podemos motivar mais os usuários e tornar o aplicativo mais atrativo, ou seja, os jogadores ficarão mais atentos a quando deverão medir a taxa de glicose no sangue novamente para continuarem pontuando no jogo e consequentemente cuidando melhor da sua saúde (principal objetivo proposto pelo Candy Castle).

Não será apenas os pacientes, foco principal do aplicativo, os únicos beneficiados com o Candy Castle, pois o benefício recebido pelo médico será significativo: com o *site* do aplicativo o médico receberá as taxas de glicose no sangue de todos seus pacientes que utilizam o aplicativo e dessa forma poderá atender a todos num curto período de tempo se comparado com o atendimento presencial de cada um deles (no caso de um simples ajuste em relação ao tratamento que não requeira uma consulta mais demorada e elaborada). O controle que o médico terá sobre seus pacientes também será maior já que ele terá acesso às taxas de glicose de cada um deles num curto espaço de tempo após a medição da taxa de glicose no sangue, pois esses dados são mandados pouco tempo depois de serem inseridos no jogo.

## 7.2 Limitações

Candy Castle é um aplicativo que depende muito que o usuário possua conexão com a internet tanto na hora de efetuar *login* no jogo como durante a partida e por conta disso acesso à internet será a principal restrição do aplicativo. Toda a comunicação com o servidor só será possível caso haja uma conexão ou por 3G/4G ou por Wi-Fi. Caso a internet falhe durante o jogo, inviabilizando o envio dos dados ao servidor, poderá ocorrer um erro ao final da partida onde torres de jogos anteriores correspondentes aos dados não enviados podem aparecer na tela de jogo, mas sem influenciar a pontuação nem os dados atuais do jogador.

Outra limitação do Candy Castle é a dependência em relação aos provedores responsáveis por informar a localização do dispositivo: GPS e/ou Network. Sem que o usuário habilite um dos dois provedores de localização não será possível incluir nenhum ponto no mapa, pois é necessária uma coordenada geográfica (latitude e longitude) para adicionar um *overlay* no mapa.

Para que todas as funções do jogo funcionem perfeitamente será necessário que o sistema operacional do dispositivo do usuário seja no mínimo o Gingerbread (Android 2.3). Se a versão for anterior a 2.3 algumas das funções não irão funcionar na sua totalidade impedindo o paciente de utilizar o Candy Castle como ferramenta no combate à diabetes. A solução dessa restrição consiste basicamente na atualização do sistema operacional que é oferecida gratuitamente pela empresa distribuidora do Android, caso o dispositivo do usuário não receba mais suporte, ou seja, não seja possível atualizar para a versão Gingerbread ou superior a única solução será a aquisição de um dispositivo mais moderno.

Uma restrição não ligada diretamente ao aplicativo, mas de fundamental importância para pessoas com diabetes é o aparelho para medição da taxa de glicose no sangue e sem ele o paciente não poderá inserir suas medições no jogo, por isso é indispensável que o usuário possua qualquer aparelho capacitado para medir sua taxa de glicose e dessa forma disponibilizar o valor dentro do Candy Castle que, posteriormente, enviará esse valor ao médico através do *site*. Atualmente, a medição e a inserção dos dados devem ser feitos manualmente, pois não dispomos de nenhum dispositivo para medir a taxa de glicose no sangue que possa ser conectado diretamente ao dispositivo móvel do paciente (esse fator pode ser uma tentação às crianças a trapacearem no jogo para atingirem bons valores e mais pontos no jogo).

## 7.3 Trabalhos Futuros

Há muito que se podemos melhorar no Candy Castle ainda e nessa seção abordaremos as principais melhorias a serem feitas no aplicativo para deixá-lo mais agradável aos usuários, sejam eles pacientes ou médicos.

As melhorias a serem feitas no aplicativo:

1 - Melhorias no *layout* do Candy Castle tanto na tela de *login* quanto na tela de jogo. Na tela de *login* podemos colocar mais cores na tela, incluir o logo do jogo no topo da tela e colocar imagens nos botões, pois estão com o formato padrão do Android. Já na tela de jogo a única melhoria a ser feita é em relação aos botões que podem ter imagens melhores que o formato padrão.

2 - Melhorias nas funções da tela de *login*. Remover a opção de *login* do usuário através da inserção de e-mail e senha por questão de segurança ao se conectar com o servidor, pois mesmo a senha não ficando salva no aplicativo ela não é criptografada em momento algum antes de ser usada e isso torna as informações do jogador vulneráveis. Ao removermos essa

opção de *login* apenas será possível se conectar ao servidor com as contas salvas no dispositivo garantindo dessa forma maior segurança ao usuário.

3 - Inserir mensagens de aviso ao usuário. Faltam algumas mensagens importantes ao usuário dentro do aplicativo, como: falta de conexão à internet, falha ao conectar com o servidor, falha ao enviar os dados ao servidor e falha na comunicação com o Facebook na hora de se conectar ou compartilhar a pontuação.

4 - Atualização do aplicativo com as novas versões da API do Google Maps e do Facebook. As funções usadas dentro do Candy Castle relacionadas a cada uma das API's sofreram alterações ou são feitas de outras formas e por isso precisamos atualizar elas dentro do aplicativo para termos sempre o melhor de cada API utilizada pelo Candy Castle e dessa forma proporcionar o melhor serviço possível aos usuários.

As melhorias a serem feitas no *site*:

1 - Área de cadastro dos usuários. Pacientes e médicos se cadastrarão no *site* e dessa forma ao se conectarem ao servidor receberão as informações destinadas a cada tipo de cadastro: pacientes receberão apenas os seus dados e médicos receberão os dados de todos os seus pacientes cadastrados.

2 - Área de *login* dos usuários. Como teremos uma área de cadastro dos usuários será necessário criarmos a área de *login* para diferenciarmos os usuários e assim buscar no servidor os dados correspondentes ao usuário conectado.

3 - Área de cadastro dos pacientes associados ao médico. Nessa página o médico informará os dados de cada um de seus pacientes para que fiquem registrados em uma lista que será usada para buscar as taxas de glicose de cada paciente posteriormente. A página apresentará um espaço para a inserção das informações (nome, idade, e-mail, entre outros dados a serem decididos) do paciente a ser registrado e abaixo um espaço com a lista dos pacientes já associados ao médico conectado e através dessa lista o médico irá escolher um paciente para ver suas taxas de glicose salvas no servidor até o momento. O médico também terá a opção de excluir um paciente de sua lista caso o tratamento tenha sido encerrado ou o paciente deixe de ser atendido pelo médico.

4 - Melhoria na página contendo as informações do paciente. Como essa página conterà todas as medições feitas pelo paciente durante o tempo que utilizar o aplicativo, será importante não apenas mostrar suas taxas como também informações extras baseadas nas taxas de glicose no sangue do usuário, pois dessa forma o usuário também poderá ver rapidamente caso algo esteja fora do nível desejado ao seu melhor tratamento da diabetes.

Após fazermos todas as melhorias propostas acima no aplicativo e no *site* precisaremos testá-las localmente, o que é feito atualmente, com alguns usuários para termos certeza de que não há nenhum erro ou novas melhorias a serem feitas e concluída essa etapa colocaremos o *site* no ar juntamente com o aplicativo na Google Play. A partir desse ponto os médicos já poderão indicar o aplicativo para seus pacientes que possuem um dispositivo com Android e ambos poderão acessar o *site* quando desejarem.

## REFERÊNCIAS

STACH, Christoph; SCHLINDWEIN, Luiz Fernando: [Candy Castle - A Prototype for Pervasive Health Games](#). In: **Proceedings of the 2012 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)**; Lugano, Switzerland, March 19-23, 2012.

[1] XIN, C.: Influence from the Serious Games on Mobile Game Developers' Commercial Strategies. In: **Proceedings of the 2008 International Seminar on Business and Information Management**; Wuhan, Hubei, China, December 19, 2008.

[2] WATTERS, C. et al.: Extending the Use of Games in Health Care. In: **Proceedings of the 39th Annual Hawaii International Conference on System Sciences**; Maui, Hawaii, USA, January 4-7, 2006.

[3] GRIMES, A.; KANTROO, V.; GRINTER, R. E.: Let's Play! Mobile Health Games for Adults. In: **Proceedings of the 12th International Conference on Ubiquitous Computing**; Copenhagen, Denmark, September 26-29, 2010.

[4] WITTGENSTEIN, Ludwig: **Philosophical Investigations**; 1st ed.: Blackwell Publishing, 1953.

[5] CAILLOIS, Roger: **Les Jeux et Les Hommes**; 1re ed. Paris: Gallimard, 1958.

[6] CRAWFORD, Chris: **Chris Crawford on Game Design**; Peachtpit, 2003

[7] BARTOLOMÉ, N. A.; ZORRILLA, A. M.; ZAPIRAIN, B. G.: Can Game-Based Therapies be trusted? Is Game-Based Education effective? A systematic review of the Serious Games for Health and Education. In: **Proceedings of the 16th International Conference on Computer Games**; Louisville, Kentucky, USA, July 27-30, 2011

[8] LANGE, B. et al.: Breath: A Game To Motivate The Compliance Of Postoperative Of Breathing Exercises. In: **Proceedings of the 2009 International Conference on Virtual Rehabilitation**; Haifa, Israel, June 29 - July 2, 2009

[9] KNÖLL, Martin: "On the Top of High Towers..." Discussing Locations in a Mobile Health Game for Diabetics. In: **Proceedings of the 2010 IADIS International Conference on Game and Entertainment Technologies**; Freiburg, Germany, July 26-30, 2010

[10] DIEHL, Leandro Arthur et al.: **InsuOnline, a Serious Game to Teach Insulin Therapy to Primary Care Physicians: Design of the Game and a Randomized Controlled Trial for Educational Validation**. January, 2013. Disponível em: <<http://www.researchprotocols.org/2013/1/e5/>>. Acesso em: nov, 2013.

[11] Disponível em: <<http://en.wikipedia.org/wiki/Gamification>>. Acesso em: nov, 2013

[12] Disponível em: <<http://developerlife.com/tutorials/?p=1375>>. Acesso em: nov, 2013.

[13] GRAHAM, R. L.: An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set. In: **Information Processing Letters Volume 1, Number 4**; North-Holland Publishing Company, June, 1972. p. 132-133. Disponível em: [http://www.math.ucsd.edu/~ronspubs/72\\_10\\_convex\\_hull.pdf](http://www.math.ucsd.edu/~ronspubs/72_10_convex_hull.pdf). Acesso em: nov, 2013.

[14] Disponível em: <http://developer.android.com/training/articles/perf-anr.html>. Acesso em: nov, 2013.