Faculdade de Informática e Administração Paulista

Bacharelado em Sistemas de Informação

PROJETO F1

Ricardo Osório Miranda – RM 59828

Herberton Souza – RM 59350

Vinícius Schmiedel Manssur – RM 59527

São Paulo

2010

Faculdade de Informática e Administração Paulista

Bacharelado em Sistemas de Informação

PROJETO F1

Trabalho de Conclusão de Curso apresentado à

banca examinadora do Curso de Sistemas de

Informação da Faculdade de Informática e

Administração Paulista, como exigência parcial

para obtenção do grau de Bacharel em Sistemas

de Informação.

Orientador: Fabian Martins da Silva

São Paulo

2010

*Agradecemos às nossas famílias pelo estímulo e apoio em todas as etapas de nossas vidas, aos professores que apoiaram a idéia e confiaram neste projeto. A colaboração, atenção e dedicação de todos foram fundamentais para a conclusão do TCC.*

**AGRADECIMENTOS**

Ao nosso professor e orientador Fabian Martins da Silva, pelo apoio e auxilio nas dúvidas e revisões ao longo do desenvolvimento do trabalho.

Aos professores César Caetano e Reinaldo Burian pela confiança transmitida no início do trabalho.

Aos nossos pais e familiares pela paciência e apoio para a conclusão do trabalho.

“Perseguir seus objetivos é como montar um

quebra-cabeça. É preciso trabalhar pedaço por

pedaço para atingir o resultado final.”

David Niven

Sumário

[LISTA DE FIGURAS 8](#_Toc273283172)

[LISTA DE GRÁFICOS 8](#_Toc273283173)

[LISTA DE TABELAS 8](#_Toc273283174)

[RESUMO 8](#_Toc273283175)

[ABSTRACT 8](#_Toc273283176)

[Capítulo 1 - INTRODUÇÃO 9](#_Toc273283177)

[1.1 - OBJETIVO 11](#_Toc273283178)

[1.2 - ORGANIZAÇÃO DO TRABALHO 12](#_Toc273283179)

[Capítulo 2 - REVISÃO DA LITERATURA 13](#_Toc273283180)

[2.1 O futuro das editoras universitárias e as mídias eletrônicas. Com Ciência. 14](#_Toc273283181)

[2.2 Multimídias Moveis 14](#_Toc273283182)

[2.3 A relação entre a satisfação e a fidelidade dos clientes com a lucratividade das empresas. 15](#_Toc273283183)

[2.4 Tecnologia Para Gerenciamento de Filas. 16](#_Toc273283184)

[2.5 A Tecnologia RFID e os Benefícios Da Etiqueta Inteligente Para os Negócios. 17](#_Toc273283185)

[2.6 Controle de dispositivo utilizando mensagens SMS com tecnologia Java ME. 17](#_Toc273283186)

[2.6.1 Implementação. 19](#_Toc273283187)

[2.6.2 Interface MIDlet. 19](#_Toc273283188)

[2.6.3 Interface CommandListener. 19](#_Toc273283189)

[2.6.4 Interface MessageListener. 20](#_Toc273283190)

[2.6.5 PushRegistry. 21](#_Toc273283191)

[2.6.6 RecordStore. 22](#_Toc273283192)

[2.6.7 Execução. 25](#_Toc273283193)

[2.7 Distribuição da demanda telefônica de um call center através da criação e priorização de filas inteligentes. 27](#_Toc273283194)

[2.8 Uso do comércio eletrônico como prestação de serviços: envio de mensagens sms via telefone celular para informações instantâneas no agronegócio. 30](#_Toc273283195)

[Capítulo 3 - MATERIAIS E MÉTODOS 31](#_Toc273283196)

[3.1 Descrição da Solução 31](#_Toc273283197)

[3.1.1 Arquitetura da Solução 32](#_Toc273283198)

[3.1.2 Descrição Geral do Fluxo do Sistema 33](#_Toc273283199)

[3.2 Casos de Uso 34](#_Toc273283200)

[3.2.1 Caso de Uso do Contexto 34](#_Toc273283201)

[3.2.2 Registra Entrada do Cliente no Parque 34](#_Toc273283202)

[3.2.3 Movimentação da Fila do brinquedo 35](#_Toc273283203)

[3.2.4 Execução do Brinquedo 35](#_Toc273283204)

[3.2.5 Saída do Cliente do Parque 36](#_Toc273283205)

[3.3 Modelagem do Banco de Dados 36](#_Toc273283206)

[3.4 Materiais 37](#_Toc273283207)

[3.4.1 Servidor de Aplicação JBoss Application Server 37](#_Toc273283208)

[3.4.2 Plataforma J2EE 37](#_Toc273283209)

[3.4.3 Hibernate 38](#_Toc273283210)

[3.4.5 RichFaces 39](#_Toc273283211)

[3.4.6 JavaServerFaces 41](#_Toc273283212)

[3.4.7 JAAS 42](#_Toc273283213)

[3.4.8 Banco de Dados ORACLE 10g XE 42](#_Toc273283214)

[3.4.9 Dispositivo Móvel 42](#_Toc273283215)

[3.4.10 Código de Barras 42](#_Toc273283216)

[3.4.11 Leitor de Código de Barras 44](#_Toc273283217)

[3.5 Métodos 44](#_Toc273283218)

[3.5.1 Entrada do Parque 44](#_Toc273283219)

[3.5.2 Portaria do Brinquedo 44](#_Toc273283220)

[3.5.3 Terminal de Consulta 45](#_Toc273283221)

[3.5.4 Operação do Brinquedo 45](#_Toc273283222)

[Capítulo 4 – TRABALHOS FUTUROS 46](#_Toc273283223)

[4.1 Alternativas de Implementação 46](#_Toc273283224)

[Capítulo 6 - Bibliografia 48](#_Toc273283225)

# LISTA DE FIGURAS

# LISTA DE GRÁFICOS

# LISTA DE TABELAS

# RESUMO

# ABSTRACT

# Capítulo 1 - INTRODUÇÃO

Por culpa do modo de vida adotado ultimamente pela sociedade, onde todo tempo tem que ser muito bem gasto, ficar parado em uma fila esperando para realizar determinada atividade se torna uma perda de tempo. Levando em conta ainda que as pessoas cada vez mais vivem para seus trabalhos e quase nunca reservam algum tempo para diversão, se sentiriam muito mais insatisfeitos, se perdessem tempo parados em uma fila também na hora de se divertirem.

Mas, ao invés de ficarem em uma fila esperando determinada ação, poderiam ficar fazendo outras atividades ou até mesmo darem uma volta e quando chegasse sua vez seriam informados para se apresentarem. O PROJETO F1 tende a criar essa possibilidade.

A idéia pode ser aplicada em todo lugar onde existam filas. Inicialmente foi feito um levantamento para definir qual o melhor lugar para aplicar a idéia. Foi concluído que os parques de diversões seriam os maiores beneficiados, pegando como exemplo: Play Center, Hopi Hari e Disney. As pessoas vão com um grande intuito de brincar, mas como as filas costumam ser muito grandes, acabam perdendo bom tempo parados até chegar sua vez. Agora imagine se você pudesse assistir atrações do parque, comer, passear, ou alguma outra atividade enquanto fica esperando sua vez.

Partindo deste ponto, foi feito um levantamento dos benefícios e eles seriam muitos, a ponto de aumentar os lucros do parque.

A idéia acaba com as filas, porém, não acaba com a espera. A jogada é fazer com que as pessoas possam circular pelo parque enquanto esperam sua vez no brinquedo, isto é, permite que o cliente aproveite ao máximo o seu tempo e não o perca aguardando parado nas longas filas que os parques de diversões têm por cultura.

Os maiores benefícios do parque ocorrerão em cadeia: haverá um aumento grande da satisfação de seus clientes que permitirá aumento na rotatividade de visitantes e logicamente o lucro do parque aumentará consideravelmente com a maior venda de ingressos. Afinal, pelo menos nesse parque de diversão, o cliente não ficará com a sensação de estar perdendo tempo. E isso pode ser fundamental para uma futura volta ao parque.

O fato dos clientes poderem circular pelo parque enquanto esperam sua vez no brinquedo permite novas fontes de lucro, porque se antes ficavam parados em extensas filas, além de perderem o seu tempo elas não consumiam. Com a idéia em prática o consumo de alimentos e bebidas aumentará visivelmente. O parque deverá ampliar suas áreas de lazer e atrações como peças teatrais e shows para entreter os visitantes enquanto aguardam sua vez no brinquedo.

O tempo que o cliente vai ficar esperando até sua vez chegar para ir ao brinquedo será o mesmo, mas como ele não ficará parado e perdendo tempo na fila esse tempo vai parecer passar muito mais rápido.

A idéia do PROJETO F1 é muito simples, o sistema trará a comodidade de que o cliente, mesmo na fila, exerça outras atividades simultaneamente, trazendo para ele maior aproveitamento do seu tempo.

O principio da solução poderá ser comparada com um sistema semelhante que existe em outra área de atuação, no ramo alimentício existe uma Lanchonete chamada Outback onde o cliente espera na mesa até que um dispositivo emita uma luz, informando que seu pedido está pronto, para então ir buscá-lo. Da mesma forma esse sistema evita que o cliente fique parado em uma fila.

Analisando a solução, é possível prever claramente o que ela irá proporcionar para o cliente e para o parque, na parte do cliente, a solução irá interferir diretamente na sua satisfação para com o parque e irá proporcionar um maior aproveitamento de seu tempo, onde antes ele perdia tempo esperando nas filas longas e demoradas e agora, não terá mais essa perda de tempo, pois poderá executar outras atividades enquanto o sistema controla a posição e o andamento da fila e assim que sua vez estiver próxima automaticamente o sistema informará para que ele se dirija ao brinquedo. Para o parque, a solução irá proporcionar diversas melhorias, incluindo uma melhor visão de administração para criação ou exclusão de novas áreas.

Os resultados do sistema podem ser avaliados da seguinte forma:

Satisfação do cliente – a satisfação do cliente pode ser avaliada pela quantidade pessoas que freqüentam o parque, através do sistema que fará o controle quantitativo de clientes. Assim, se a freqüência aumentar pode-se concluir que os clientes estão contentes com o parque. Um índice que pode fortalecer esse item é o controle feito pelo sistema de quantas vezes uma mesma pessoa voltou ao parque, avaliando assim a sua satisfação.

Faturamento do parque – o principal item a ser avaliado é a quantia gasta por cada cliente levando em conta uma média anterior e a partir daí, fazer uma reavaliação considerando a freqüência de clientes diariamente no parque.

Maior controle sobre informações – o primeiro índice a ser avaliado é o controle de quantas vezes um mesmo cliente freqüentou o parque, quantas vezes esse mesmo cliente foi em determinados brinquedos, quais ele não foi, qual é o brinquedo mais e menos procurado. Podendo expandir cada vez mais essas analises, que serão responsáveis por boas tomadas de decisão da administração do parque.

## 1.1 - OBJETIVO

O objetivo do trabalho é implementar uma prova de conceito para uma solução de automação de gestão de filas que visa otimizar o tempo de espera nas filas, garantindo que o cliente ficará o tempo mínimo necessário para acessar o recurso e com isso mostrar que com apenas essa solução conseguimos acarretar diversas melhorias e benefícios tanto para o Parque de Diversão quanto para os clientes.

Dessa maneira se tornando essencial para uma boa relação e satisfação entre ambas as partes.

Quanto mais facilidades oferecer ao cliente, mais tempo ele terá para se divertir, recomendando o local à novas pessoas trazendo maior rotatividade de clientes, gerando mais lucros para o Parque.

## 1.2 - ORGANIZAÇÃO DO TRABALHO

O trabalho está dividido em seis capítulos, sendo eles: Capítulo 1, Introdução; Capítulo 2, Revisão Bibliográfica; Capítulo 3, Materiais e Métodos; Capítulo 4, Resultados e Discussões; Capítulo 5, Conclusão; Capítulo 6, Referencias Bibliográficas.

No Capítulo 3, são apresentados os Materiais e Métodos utilizados para o desenvolvimento do PROJETO F1.

No Capítulo 4, são apresentados os resultados dos testes, assim como as discussões.

No Capítulo 5, são apresentadas as conclusões do projeto e, as perspectivas futuras.

# Capítulo 2 - REVISÃO DA LITERATURA

No início da coleta de materiais para fazer esse trabalho o que mais chamou a atenção foi à diversidade de assuntos e áreas que poderiam ser trabalhadas, tanto na parte técnica, como na parte de envolvimento humano ou até mesmo na parte de como a tecnologia pode vir a ajudar uma organização na difícil tarefa de aumentar a satisfação do cliente sem que tenha que realizar grandes sacrifícios.

Foi feita uma abordagem bem ampla, sobre diversos temas, assuntos técnicos, de relacionamento com o cliente, tornando o material bem atraente para entender quais áreas o trabalho poderia afetar e também como de que forma isso ocorreria.

Às vezes com uma simples mudança de estratégia na forma de administrar as filas de espera é possível fazer os clientes se sentirem mais respeitados pela organização aumentando sua satisfação com o negócio e tornando-os mais que clientes, e sim parceiros na boa condução do negócio.

Hoje em dia é muito falado sobre a mobilidade, assunto ligado diretamente ao trabalho, por isso foi pesquisado tecnologias que podem ser usadas para melhorar essa mobilidade no Parque de Diversões.

* 1. O futuro das editoras universitárias e as mídias eletrônicas.

Na reportagem escrita por Enio Rodrigo, o futuro das editoras universitárias e as mídias eletrônicas, é discutido sob uma hipótese cada vez mais presente no nosso dia-a-dia. Ele discute de fato se algum dia as mídias eletrônicas podem vir a substituir os livros impressos. O autor acredita que um dia isso possa acontecer se apoiando em teses de pessoas que estão nesse meio, como Guillaume Apollinaire e Michael Hart que é o inventor do conceito e-book, afinal desde 1971 Hart trabalha para disponibilizar textos via mídias eletrônicas, trabalhando no Projeto Guttenberg, onde livros são transformados para o formato ASCIII (codificação de caracteres) ou PDF e disponibilizados no site do projeto. Mas Enio não deixa de mostrar a opinião de quem não acredita nessa hipótese, como o presidente da Livraria Cultura, Pedro Herz. Mas o que dá para se concluir melhor desse texto, visando muito que diz Mírian Goldfeder (coordenadora de formação e cursos da Universidade do Livro da UNESP) é que a transmissão de livros via mídias eletrônicas só tendem a crescer, mas jamais haverá de fato a morte do livro, havendo a coexistência dos produtos, visando públicos diferentes, “... Existe um público que não vai abrir mão do livro impresso e outro que vai preferir ler livros em outros formatos..." defende Mírian.[1]

* 1. Multimídias Moveis

No artigo Multimídias Moveis, os autores Karen Nakazato, Rafael Barbosa, Raphael Katsuragi, etc., mostraram os conceitos básicos das tecnologias multimídia que são utilizadas em dispositivos moveis, tais como Bluetooth e wifi. Além da definição sobre Mobilidade, é colocada também algumas das vantagens (otimizar o tempo) e desvantagens (custo do hardware) desses dispositivos. É contada toda evolução histórica desses dispositivos, que começou em 1992 com um handheld chamado Newton da Apple, mas que não teve grande repercussão, até um dos mais badalados hoje em dia, os Smartphones. Acessibilidade e usabilidade são dois dos conceitos que devem andar de mãos dadas em aplicações e dispositivos móveis. Se ambos objetivos forem alcançados satisfatoriamente, cobriremos um maior número de pessoas atendidas, resultando em uma satisfação muito maior. Uma das tecnologias que mais avançaram foi a de transmissão de dados, que se divide em curto (Bluetooth), médio (wifi) e longo alcance (radio freqüência de telefonia celular). Como o hardware varia muito de dispositivo para dispositivo, o desenvolvimento de softwares é muito específico, tem que saber exatamente para qual dispositivo desenvolver e qual a capacidade desse dispositivo, para nunca desenvolver algo que o dispositivo não seja capaz de processar. É usado como exemplo um caso de agosto de 2007 quando a Air France lançou um serviço de check-in pelo celular. O cliente acessava o site da companhia pelo celular, fazia o check-in e escolhia o seu assento, após isso recebia uma confirmação via SMS.[2]

* 1. A relação entre a satisfação e a fidelidade dos clientes com a lucratividade das empresas.

No texto "A relação entre a satisfação e a fidelidade dos clientes com a lucratividade das empresas”, os autores Mario Orestes Aguirre González, Célio Gurgel Amorim e Rubens Eugênio Barreto Ramos falam sobre um assunto ainda não muito discutido que é a relação direta entre satisfação, fidelidade e lucratividade da empresa. Heskett e Shlesinger (1994) afirmam a existência de uma cadeia de serviço-lucro. A lógica é que clientes mais satisfeitos são mais fieis, assim proporcionando maiores lucros para a empresa. A atenção das empresas com a satisfação dos clientes não é um fato recente, Vavra (1997) cita o exemplo da Brithish Car Manufacturer que no ano de 1906 declarou os princípios essenciais da satisfação do cliente como política da empresa. A literatura técnica trata dois enfoques básicos para a satisfação do cliente: como resultado de uma transação específica e como resultado de uma percepção acumulada. Mas só foi dada a devida importância para isso na década de 80 com a chegada da concorrência e o rápido desenvolvimento do mercado de serviços. Já para a fidelidade de clientes existe uma grande bibliografia sobre o assunto, mas ainda o conceito é pouco explicativo. Jacoby e Kyner (1973) tratam duas abordagens básicas: abordagem comportamental e abordagem baseada na atitude. Ainda é limitada a quantidade de pesquisas teóricas que englobam essa cadeia de valor, um dos motivos pode ser a dificuldade das empresas em calcular os dados referentes ao lucro e as despesas para cada cliente. (Kaplan & Norton. 1997) Pesquisas mostram que existe uma relação positiva entre satisfação e fidelidade do cliente, e segundo Reicheld(1996) e Heskett e Schlesinger(1994), existe ainda uma relação positiva entre retenção do cliente e lucratividade da empresa.[3]

* 1. Tecnologia Para Gerenciamento de Filas.

No texto elaborado pelo engenheiro Leônidas Vieira Junior é tratado um assunto que pode deixar muita gente irritada, as filas. Qualquer tipo de fila exige uma administração para bem conduzí-la. Pois o contrário disso pode gerar um stress generalizado, insatisfação do cliente, diminuindo até a produtividade dos atendentes. O bom atendimento ao cliente em determinado ambiente de serviço está fundamentado em 4 fatores chave de sucesso: aspectos humanos, marketing empresarial, processos dos serviços e tecnologia adequada. A duração da fila é inversamente proporcional ao nível de satisfação, por isso devemos dar um maior foco a esse problema. Pesquisas mostram que o mais importante na espera é o tempo que o cliente "percebeu" que passou na fila, ou seja, se conseguir mantê-los de certa forma ocupado, isso poderá deixá-los menos aborrecidos com a espera. Kostecki (1996) mostra algumas razões que levam a considerar as filas de espera como um importante problema e uma boa solução têm impacto muito alto no valor do negócio, direto e imediato, sobre a marca e a credibilidade da prestadora de serviço. Devemos considerar que as pessoas têm um rítmo de vida muito acelerado, o que torna as esperas intoleráveis. A satisfação do cliente está diretamente ligada ao sucesso dos negócios, por isso deveriam ser analisadas cada vez mais as preferências dos clientes em termos de espera. São propostas algumas ações para melhorar o tempo de espera e a qualidade do atendimento com o uso de algumas tecnologias, como a senha, por exemplo, que é a mais utilizada. Os clientes retiram uma senha e quando chega sua hora ele é avisado através de um painel. Como podemos ver as filas de espera podem se tornar um grande problema, mas quando bem trabalhadas elas podem resultar até mesmo em um acréscimo de lucro[4]

* 1. A Tecnologia RFID e os Benefícios Da Etiqueta Inteligente Para os Negócios.

O autor Cláudio Gonçalves Bernardo soube dizer muito bem o que o futuro da tecnologia espera dessa etiqueta inteligente, conhecida como RFID. Especialistas em tecnologia dizem que o RFID irá revolucionar o mercado. O autor dá um exemplo de empresas grandes que apostam muito nessa tecnologia, exemplos como Gilette, Airbus e Wall-Mart. O problema dessa tecnologia é o custo para o mercado, afinal pensando em um laptop o custo da etiqueta é baixo, mas pensando em uma caixa de leite ou garrafa de refrigerante o custo é altíssimo, afinal existe gasto com os leitores das etiquetas, a infra-estrutura extremamente complexa capaz de coletar, examinar e mover o vasto volume de dados gerados pelas etiquetas. Essas etiquetas são capazes de armazenar dados enviados por transmissores. RFID é composto por três componentes: antena, transcrever e transpuser. Antena é responsável por criar o campo de ação; o transcrever é o leitor, é o componente que emite as freqüências de rádio para se comunicar com a etiqueta e o transponde é a etiqueta RFID, podendo ter diversos formatos e ser ativa (alimentada por uma bateria que permite tanto escrita quanto leitura de dados) ou passiva (usadas apenas para leitura, maior capacidade de armazenamento). O mercado reserva boas oportunidades de crescimento para a tecnologia RFID, diversas áreas apostam nela para evolução dos seus processos, desde a logística até a segurança do consumidor. Algumas vantagens do seu uso é a capacidade de armazenamento, leitura e envio de dados, detecção sem a necessidade da proximidade da leitora para ler os dados e algumas desvantagens como o alto custo do RFID em relação ao código de barras e a invasão de privacidade dos consumidores por causa da monitoração das etiquetas coladas nos produtos. Existem técnicas que previnem isso, mas com o custo elevadíssimo. Como a operação por RFID agrega eficiência, agilização dos procedimentos, maior processamento dos dados tendo assim um diagnóstico exato, eliminando riscos de falha na previsão, obtendo maior lucratividade e menor perda de tempo, a previsão é de que no futuro todos os mercados adotarão essa tecnologia.[5]

* 1. Controle de dispositivo utilizando mensagens SMS com tecnologia Java ME.

O objetivo é programar uma aplicação para celular na linguagem Java com intuito de simular o controle remoto de um dispositivo robô através de mensagens de texto SMS.

A arquitetura Java ME (Java Plataform, Micro Edition), ainda muito conhecida como J2ME, é uma tecnologia da Sun Microsystems que permite o desenvolvimento de software para dispositivos embarcados como telefones celular e PDAs.

É desenvolvida especificamente para dispositivos com limitações de memória e processamento e é composta de três componentes principais chamados configurações, perfis e pacotes opcionais, que provêem informações específicas sobre APIs e diferentes famílias de dispositivos.

As configurações fornecem as funcionalidades de linguagem mais básicas, designadas para dispositivos com limitações de memória e poder de processamento. Especifica uma JVM (Java Virtual Ma-chine) que pode ser suportada por esse tipo de dispositivo e ainda utiliza um subconjunto da plataforma J2SE (Java Standard Edition) e outras APIs que achar necessário. A configuração mais comum, e que irá interessar no desenvolvimento da aplicação proposta por este trabalho, é a CLDC (Connected, Limited Device Configuration), que é voltada para dispositivos com poucos recursos, como celulares, pagers e PDAs. A CLDC é baseada em uma versão minimalista da JVM chamada KVM, possui limitações e não é capaz de executar as mesmas aplicações que a JVM executa em uma aplicação desktop Java, desenvolvida com a API J2SE. Atualmente a CLDC encontra-se na sua versão 1.1.

Os perfis agregam a API funcionalidades específicas para desenvolvimento de aplicações em uma determinada família de dispositivos. Ou seja, susportam APIs mais avançadas como as de interfaces gráficas e persistência de dados. O mais interessante é a aplicação do MIDO (Móbile Information Device Profile), que se utiliza em celulares e PDAs. Atualmente ele está na versão 2.0.

Pacotes opcionais possuem funções adicionais específicas que podem ser incluídas em uma configuração partícular ou necessidades específicas da aplicação.

* + 1. Implementação.

Espera-se que através do uso da plataforma Java ME, seja possível desenvolver uma aplicação de envio e recebimento de mensagens de texto, que desempenhe um controle remoto de um robô. Sua posição será representada através de um desenho utilizando as bibliotecas Graphics e Canvas da plataforma Java.

* + 1. Interface MIDlet.

Implementa os estados que definem o comportamento de uma aplicação que irá rodar num dispositivo móvel. Nela contém métodos de nascimento da MIDlet(startApp), de pausa (pauseApp), e destruição (destroyApp). Ela tem um ciclo de vida dividida em 3 partes: Active – define-se quando a MIDlet é inicializada e livre para alocar recursos para a aplicação. Paused – só é atingido quando o construtor da MIDlet foi há pouco chamado e ainda não executou o método StartApp, para entrar no modo Active. E Destroyed – que executa o método destroyApp, que tem a função de deslocamento de recursos utilizados ao longo da execução da aplicação.

* + 1. Interface CommandListener.

Ela é muito comum em aplicações Java ME, e define o método commandAction que responde a eventos de seleção do menu da aplicação MIDlet. Esse objeto Command é associado e determinado a comando acionado em um dos menus da MIDlet. Através de seu atributo Type ou Label, permite distinguir opções do menu.

O código abaixo exibe um trecho da aplicação:

*public void commandAction(Command c, Displayable d) {*

*/\* Se o comando Exit for clicado \*/*

*if (c.getCommandType() == Command.EXIT) {*

*//Executa algo em resposta*

*}*

*/\* Se a janela for a principal \*/*

*if (d == mainScreen) {*

*/\* Se o comando Send for acionado \*/*

*if (c.getLabel() == "Send") {*

*//Executa algo em resposta*

*}*

*/\* Se o comando Config for acionado \*/*

*else if(c.getLabel() == "Config") {*

*//Executa algo em resposta*

*}}}*

* + 1. Interface MessageListener.

Essa interface faz parte do pacote de extensão Java javax.wireless.\*, que implementa um método específico que é disparado quando ocorre um envio de SMS. E chega a resposta a tal evento de envio de mensagem, o que faz essa conexão é o método notifyIncomingMessage.

O trecho de código abaixo, mostra a utilização da interface.

*public void notifyIncomingMessage(MessageConnection conn) {*

*try {*

*TextMessage aMess = (TextMessage) conn.receive();*

*mainScreen.append(getStringTime() + "MSG rcv:>"+aMess.getPayloadText()+"<\n");*

*}*

*catch (java.lang.SecurityException exp) {*

*3*

*mainScreen.append(getStringTime() + " Permission refused to receive SMS\n");*

*}*

*catch (Exception exp) {*

*mainScreen.append(getStringTime() + " Problem while receiving message\n");*

*}}*

* + 1. PushRegistry.

Uma aplicação chamada A.M.S que define os estados e transições do ciclo de vida de uma MIDlet, mas isso não interfere no modo de como ela é inicializada.

Só há uma maneira de uma MIDlet ser inicializada: através da ativação do usuário, que é definido pelo MIDP 1.0. Agora, no MIDP 2.0 são possíveis dois tipos de ativações: em resposta a um evento de conexão ou em resposta a um timer programado.

Com esse recurso é possível implementar aplicações que inicializam em resposta a algum evento. No momento em que a conexão é estabelecida, uma instâancia da MIDlet ReceptorSMS é disparada, alertando a chegada de uma mensagem SMS enviada pelo emissor. Só é possível fazer essa associação inserindo uma entrada no arquivo JAD, que é um arquivo de texto com parâmetros e configurações da aplicação. Como neste código: MIDlet-Push-1: sms://:16555, ReceptorSMS, \*

No código abaixo, através do método listConnection é possível obter a lista de Strings de conexões que foram declaradas como do tipo “conexões que disparam algum evento”. Se o parâmetro desse método for false, esse irá retornar todas as conexões, e caso o parâmetro seja true, irá retornar apenas conexões com algum dado disponível, ou seja, conexões ativas.

*String connections[];*

*// Obtém as conexões ativas*

*// O parâmetro true indica que são conexões que aguardam por algum evento.*

*connections = PushRegistry.listConnections(true);*

*// Há alguma conexão na lista?*

*if (connections.length != 0) {*

*...*

*//Código executado para conexões ativas*

*...*

*}*

*else {*

*//Obtém lista de conexões que não estão aguardando evento*

*connections = PushRegistry.listConnections(false);*

*//Código executado para conexões que não aguardam evento*

*}*

* + 1. RecordStore.

A persistência de dados em um dispositivo com suporte à tecnologia Java ME

pode ser alcançada através do Record Management Store (RMS), permitindo que certas variáveis mantenham-se inalteradas perante múltiplas chamadas. Esse sistema é implementado através da classe javax.microedition.rms.RecordStore, que concede acesso de abertura, fechamento, escrita e leitura de dados no vetor de bytes que o consiste.

Um exemplo dos mecanismos básicos é apresentado no código abaixo.:

//Abre e ativa um novo RecordStore

RecordStore rs = RecordStore.openRecordStore("MyAppointments",true);

//Fecha um RecordStore já aberto.

rs.closeRecordStore();

//Remove um RecordStore

RecordStore.deleteRecordStore("MyAppointments");

//Guarda uma nova String em um RecordStore de numero 0

String appt = "new record";

byte bytes[] = appt.getBytes();

rs.addRecord(bytes,0,bytes.length);

//Atualiza o valor da String em um RecordStore de número 0

String newappt = "update record";

Byte data = newappt.getBytes();

rs.setRecord(1, data, 0, data.length());

//Remove um determinado valor do RecordStore de numero 1 rs.deleteRecord(1);

O código abaixo mostra como é feita a leitura escrita dos valores x e y referentes ao deslocamento que deve ser aplicado ao robô. Para a leitura dos dados gravados, é feita

inicialmente a abertura do RecordStore através do método openRecordStore de nome ”myRecord”, e em seguida a atribuição dos records de número 1 e 2 em vetores de bytes, através do método getRecord.

try {

//Abre o RMS

rs= RecordStore.openRecordStore("myRecord",true);

//Lê os valores das coordenadas atuais do robô

int n = rs.getNumRecords();

if(n == 0) {

x = y = 0;

}

else {

byte b1[] = rs.getRecord(1);

byte b2[] = rs.getRecord(2);

x = Integer.parseInt(new String(b1)); //Valor X

y = Integer.parseInt(new String(b2)); // Valor Y

}

rs.closeRecordStore();

}

catch(Exception e) {

System.out.println(e);

}

Para guardar novamente os valores x e y no RecordStore, utiliza-se métodos addRecord, caso o registro não tenha sido criado anteriormente, e setRecord, caso o registro já exista.

//Escreve no RMS os novos valores de x e y

try {

rs= RecordStore.openRecordStore("myRecord",false);

byte b1[] = (""+x).getBytes();

byte b2[] = (""+y).getBytes();

int n = rs.getNumRecords();

if(n == 0) {

rs.addRecord(b1, 0, b1.length);

rs.addRecord(b2, 0, b2.length);

}

else {

rs.setRecord(1, b1, 0, b1.length);

rs.setRecord(2, b2, 0, b2.length);

}

rs.closeRecordStore() ;

} catch(Exception e) {

System.out.println(e);

}

* + 1. Execução.

Sugere-se a execução da aplicação em dois aparelhos celulares diferentes, mas podendo também ser executado no mesmo dispositivo. Para demonstrar a saída do programa, utilizaremos o simulador de MIDP do Java Wireless Toolkit.

Deve-se executar, inicialmente, a aplicação ReceptorSMS em um dos aparelhos para prepará-lo para receber conexões do tipo SMS em uma porta específica. A porta padrão é definida como sendo a 16555 e está definida no arquivo JAD da MIDlet. Quando a aplicação receptora é iniciada, é exibida uma mensagem de confirmação informando que já está apta a receber mensagens e a reagir à chegada.

O outro aparelho celular deve executar a aplicação EmissorSMS para enviar a mensagem. O corpo da mensagem, o número do aparelho para o qual será enviado e a porta podem ser especificados na opção ”Config”da MIDlet.

Passo a Passo : Uma mensagem é mostrada quando se inicia a aplicação receptora. O dispositivo passa a aguardar eventos de conexões de SMS.

* + - 1. O menu de configurações da aplicação emissora, é possível alterar o corpo da mensagem, o telefone e a porta para envio. Quando tais configurações estiverem definidas, deve-se utilizar a opção Send para enviar a mensagem. Após o envio, o aparelho irá apresentar uma mensagem solicitando permissão para ouvir conexões, uma tela pedindo confirmação sobre a operação, pelo fato de ser um serviço que é cobrado pela operadora de telefonia móvel.
      2. Após o processo de envio da mensagem, um aviso é mostrado indicando o sucesso da operação.
      3. O corpo da mensagem SMS deve conter os valores ”moveleft”, ”moveright”, ”moveup”e ”movedown”, que são reconhecidos pela aplicação receptora, a fim de que seja possível identificar os valores de deslocamento do robô no aparelho que receberá as mensagens.
      4. Quando o receptor perceber a chegada da mensagem, irá processar o corpo e enviar o comando para o robô. A reação do robô pode ser percebida através da opção ”Robot”, que exibe a localização atual desse. A posição inicial do robô antes do envio da mensagem de ”moveleft”
      5. Um alerta é emitido pelo telefone celular antes do envio da mensagem SMS, devido ao fato de ser uma operação que é cobrada pela operadora.
      6. Há um alerta avisando sobre a comunicação com outro dispositivo.
      7. As opções do menu da aplicação emissora.
      8. Há uma janela de aplicação da emissora após a confirmação do envio da mensagem.
      9. Imagem mostrando a posição inicial do robô.
      10. Posição do robô após recebimento do comando moveleft.

Hoje em dia as com a acessibilidade e a rapidez da tecnologia, a evolução de dispositivos em relação à melhoria dos recursos de acesso e possibilidade da ultilização de banda larga sem fio em diversos tipos de aparelhos.

Esse crescimento também tem sido acompanhado na área de programação. A plataforma Java ME prove um ambiente robusto e flexível de construção de aplicações para dispositivos embarcados como telefones móveis e celulares, bem como a construção de interfaces de usuário e suporte para conectividade em utilitários para Internet.

A Java ME também conta com extensões para programação de interfaces USB, IrDa e Bluetooth. Dessa forma, é possível a expansão das funcionalidades da aplicação de controle remoto do robô virtual, como acionar algum recurso Eletrônico de um automóvel, ativar algum dispositivo em um ambiente doméstico ou mesmo implementar uma interface de comunicação que permita controlar um robô verdadeiro.[6]

* 1. Distribuição da demanda telefônica de um call center através da criação e priorização de filas inteligentes.

Esse artigo mostra a implementação de um trabalho aplicado à idéia de uma melhor organização das filas de espera telefônica em um Call-Center. Chamado de *Fila Inteligente* esse trabalho é implementado em uma empresa de Telemarketing real, localizada em Recife-PE, e aborda assuntos bem interessantes e ligados ao nosso trabalho como a *Teoria das Filas*.

Com a implementação da *Fila Inteligente* houve uma redução de tempo de espera nas filas, melhor satisfação dos clientes e, principalmente, uma economia de quase R$ 780.000,00/ano.

Com recursos limitados em seu Call-Center a Empresa cria estruturas de grupos de atendimento e filas inteligentes para maximizar a sua performance no atendimento e na qualidade do serviço prestado aos clientes, aumentando assim a satisfação do Cliente.

Foram definidos tipos de clientes (*Perfis*) e fornecido um tratamento diferenciado para Clientes com perfil diferentes com problemas de diferentes tipos.

Para o trabalho filas são caracterizadas por:

* Mecanismo de Chegada – Trata-se de da forma como os clientes chegam ao sistema. Essas Chegadas podem ser caracterizadas por:
  + Taxa de chegadas λ – Número de chegadas em uma unidade de tempo;
  + Distribuição – Segmentação por probabilidade de tipos de requisições a serem atendidas para um determinado tipo de Perfil.
* Mecanismo de Serviço – Como serão disponibilizados os serviços. Poderão ser utilizados:
  + Taxas de Serviço μ;
  + Distribuição;
  + Números de postos de Serviço – Número de agentes.
* Disciplina da Fila – Regras de escolha do cliente a ser atendido.(*Ex.: FIFO, LIFO, etc.*)

Segundo o Trabalho, não devemos desconsiderar as variações “imprevisíveis” de possíveis sobrecargas nas filas que ocorrem após determinados eventos. Sobrecargas podem ocorrer de dois lados:

* Lado da Oferta – É qualquer indisponibilidade não prevista que ocorre na Empresa. (*Ex.: A ausência de um atendente por um motivo qualquer como doença ou férias pode sobrecarregar a fila para os demais atendentes*);
* Lado da Demanda – É um aumento na fila por um evento externo. (*Ex.: Uma promoção realizada pelo departamento de Marketing pode trazer uma demanda maior de clientes na fila*).

Além de não desconsiderar essas variações também devemos prevê-las tomando conhecimento das campanhas promocionais, da disponibilidade dos envolvidos no processo de Filas, etc.

O tratamento de atendimento das filas é feito sob alguns aspectos, como:

* Definição de grupos virtuais (*Chamados Skills*) aos quais os atendentes estão conectados. Cada grupo tem sua fila de atendimento;
* Definição do caminho que o cliente irá ter até ser atendido, ou seja, definição de qual fila o cliente vai entrar e com qual prioridade(No *Fila Inteligente* é a definição de um ramal virtual, ou *VDN*, para cada atendente dentro de cada *Skill*);
* Definição das regras de negócio e prioridades (Chamado *Vector*) que separam cada solicitação a uma fila e um peso diferente (*Skill*) dependendo da sua necessidade.

A priorização é feita através de pesos ou prioridades (Máxima, Alta, Média ou Baixa), como:

* Uma requisição com maior peso passa a ser atendida na frente das outras, independentemente da ordem de chegada;
* Se todas as requisições possuírem pesos iguais é atendida a que tiver esperando mais tempo na fila;

A definição de qual fila o cliente irá entrar e com que prioridade ele entra varia de acordo com as seguintes regras:

* Analisar o tempo médio da fila juntamente com o seu desvio padrão para realizar uma redistribuição dos clientes que estão em uma fila sobrecarregada para uma fila menos sobrecarregada de modo que assim se tenha maior aproveitamento das filas;
* Redefinição de escolhas de prioridade de acordo com:
  + O tempo de espera na fila – Se o cliente ultrapassou um tempo de espera X na fila a sua prioridade aumenta;
  + Tempo de atendimento de cada atendente.
* Definição de exceções dependendo da necessidade – Como, por exemplo, a opção “Perda ou Roubo do Cartão” em um banco deve ser classificada como uma exceção e deve ser tratada com máxima prioridade.

Dessa forma, mesmo havendo um aumento real de ligações não houve grande percepção por parte dos clientes. Segue, abaixo, alguns indicadores:

|  |  |  |
| --- | --- | --- |
| **Média no tempo de espera nas filas** | | |
| Antes | Depois | Ganho |
| ~ 2 minutos e 26 segundos | ~ 47 segundos | ~ 67%**↑** |

|  |  |  |
| --- | --- | --- |
| **Média no tempo de atendimento** | | |
| Antes | Depois | Ganho |
| ~ 3 minutos e 18 segundos | ~ 2 minutos e 42 segundos | ~ 18%**↑** |

Conclui-se que um sistema baseado em otimização das filas que levam em consideração fatores importantes como a definição e redefinição automática de prioridades por clientes de acordo com o atual, a redistribuição de fila utilizando estatística e a constante definição de grupos consegue de fato gerar um ganho significativo à corporação. [7]

* 1. Uso do comércio eletrônico como prestação de serviços: envio de mensagens sms via telefone celular para informações instantâneas no agronegócio.

Esse artigo mostra a importância, devida grande concorrência de mercado existente e a globalização, de se utilizar tecnologias de ponta nos negócios visando uma melhor e mais rápida comunicação entre cliente e empresa.

A tecnologia empregada nesse trabalho foi o envio de mensagens SMS na área do Agronegócio para cotação melhores preços e/ou variações nos preços dos produtos agrícolas em tempo real (*Real-Time*).

Segundo seus autores informações de qualidade nas mãos de bons administradores é um fator crucial e determinante para o bom desenvolvimento das suas empresas. A informação concreta é como um grande poder na mão de quem a possui, pois a mesma permite a empresa multiplicar os resultados esperados.

Destaca-se também a crescente evolução do Brasil como uma grande e crescente potência na produção e comercialização de grãos e atribui essa retórica devido ao avanço na tecnologia de produção desenvolvida aqui no Brasil.

Com esse embasamento foi desenvolvido um sistema de envio de mensagens simultâneas via SMS para telefones celulares contendo informações reais de mercado à medida com que as quais aconteciam. Essas informações eram de variações em preços de fornecedores, mão de obra, resultados da empresa no semestre, última venda efetivada da minha empresa e do concorrente, possíveis negócios futuros e/ou parcerias, etc. Esse sistema entrou em produção para a empresa Castrolanda-Castro-PR e com celulares das operadores *VIVO* e *TIM.*

Com informações de cunho gerencial nas mãos dos gestores em “*real-time*” permite uma grande vantagem competitiva em relação à concorrência por um baixo custo de investimento e uma fácil adaptação. [8]

# Capítulo 3 - MATERIAIS E MÉTODOS

## 3.1 Descrição da Solução

O Projeto F1 resolve o problema de tempo de espera parado nas filas, fazendo com que o sistema gerencie a fila de maneira que não seja necessária a permanência física das pessoas nas filas.

Quando o cliente entrar no parque irá cadastrar o seu celular e receberá uma etiqueta código de barras, que será o identificador do cliente. Ao escolher o brinquedo desejado, irá passar essa etiqueta na portaria do brinquedo entrando automaticamente na fila virtual, por isso ele estará livre para andar pelo parque aguardando até que o sistema o informe através de uma mensagem no celular que sua vez está chegando. Quando ele for informado deverá dirigir-se até o brinquedo e entrará em uma pequena fila física aguardando para entrar no mesmo.

A solução trabalha com dois tipos de filas, a virtual e a física. A fila virtual é aquela que tem todos os clientes cadastrados, mas não estão fisicamente nela, e sim passeando pelo parque. A fila física é uma pequena fila que será formada apenas pelas pessoas que já foram chamadas para entrar no brinquedo, o tempo de espera nela é muito curto.

A fila será controlada desse modo:

Terá um leitor de código de barras no início da fila e outro na entrada do brinquedo. Para o usuário se cadastrar em uma fila é necessário passar o código de barras no leitor e quando for chamado passará a etiqueta novamente na entrada da fila e o sistema irá verificar se ele foi realmente chamado e permitirá sua entrada em uma pequena fila física esperando por sua vez. Quando for entrar no brinquedo é necessário passar a etiqueta novamente. O funcionário que controla o brinquedo terá acesso ao sistema para informar o momento em que as pessoas entrarão no brinquedo e o momento delas saírem do mesmo.

A quantidade de pessoas presentes na fila física será definida através do calculo:

Qp = (TempoDeMensagem / TempoExecuçãoBrinquedo) \* QntPessoasPorBrinquedo

Tempo de Mensagem será uma constante cadastrada no sistema, que é o tempo necessário para a pessoa independente da localização chegue a tempo no brinquedo.

Tempo de execução do brinquedo é o tempo que leva para o cliente entrar, brincar e sair do brinquedo e será calculado com base no intervalo da interação do funcionário no nosso sistema.

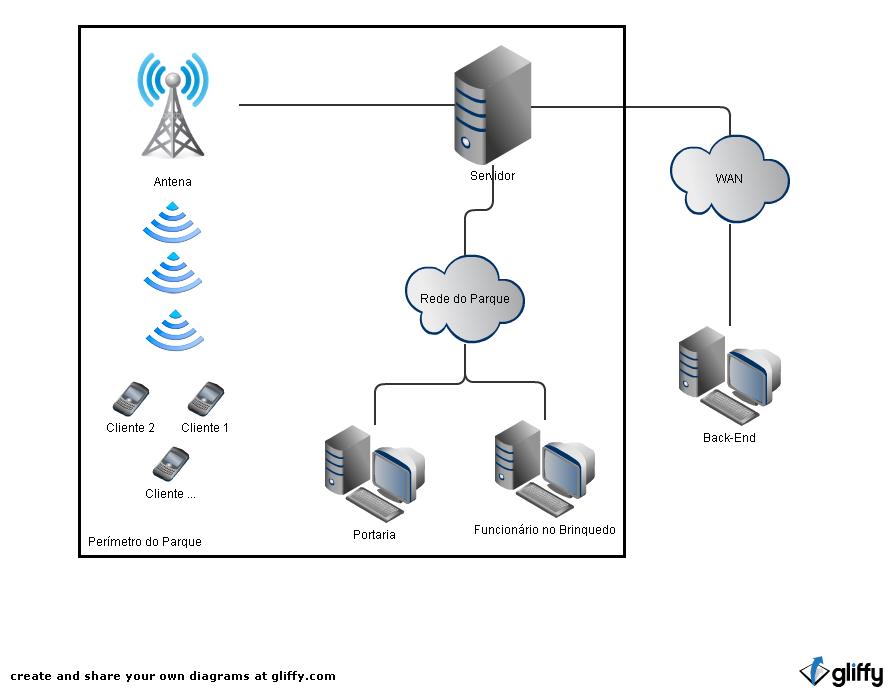
Ao lado do leitor na entrada da fila existirá um relógio informando o tempo total de fila para determinado brinquedo e esse tempo será calculado da seguinte forma:

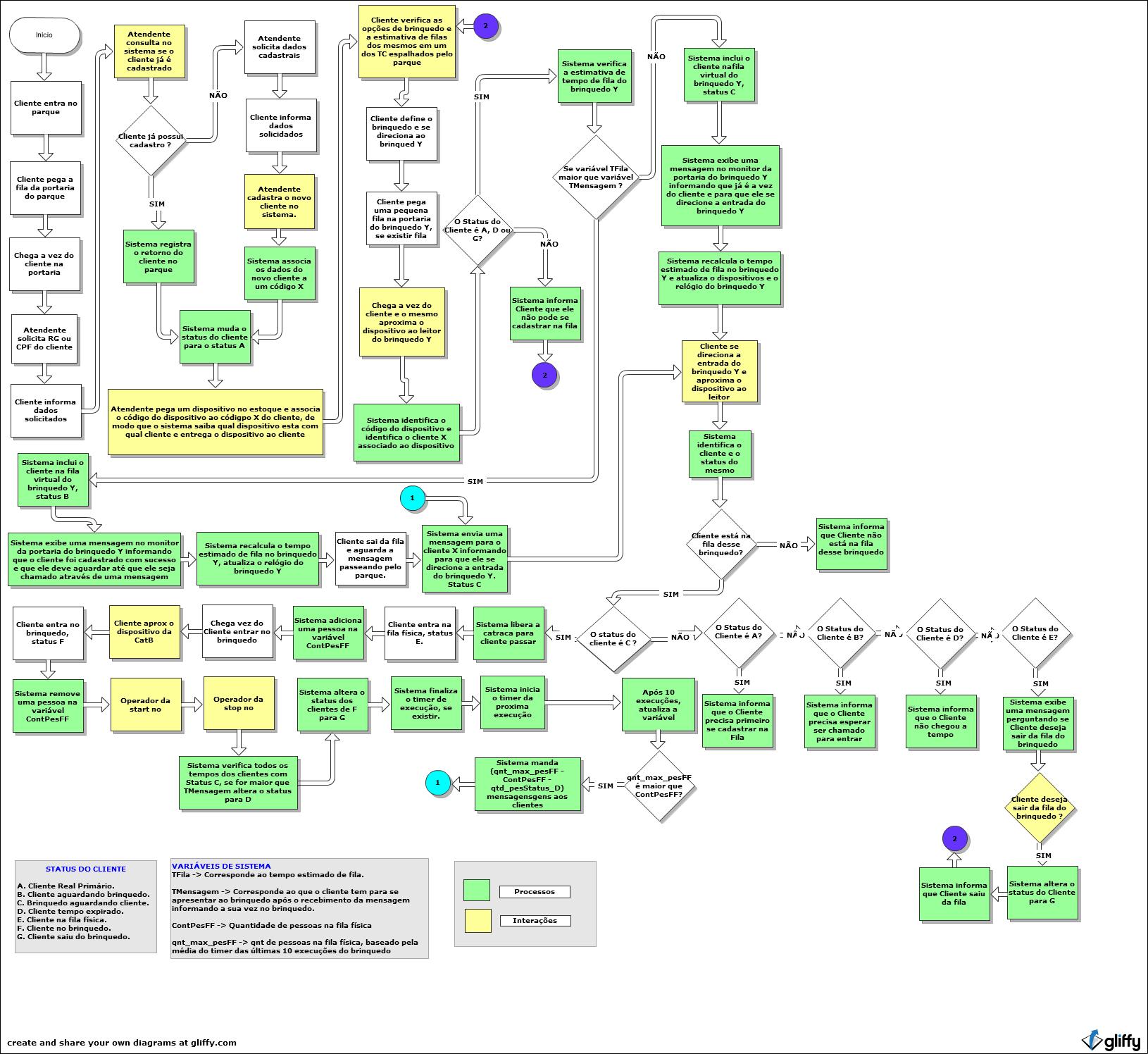
Ttf = TempoExecuçãoBrinquedo \* QntPessoasCadastradasNaFila

Esse tempo será re-calculado toda vez que uma pessoa se cadastrar na fila. As pessoas antes de se cadastrarem poderão ver uma estimativa de quanto tempo ela levará para entrar no brinquedo olhando através do relógio que fica localizado ao lado do brinquedo.

Se por algum motivo algum brinquedo deixar de funcionar, ou quebrar, o sistema irá enviar uma mensagem a todas as pessoas que estão cadastradas na fila desse brinquedo, assim ficando livres para escolher outros brinquedos para ir.

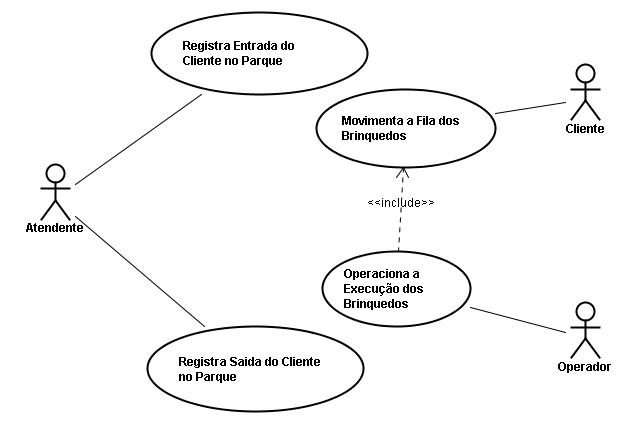
* + 1. Arquitetura da Solução



* + 1. Descrição Geral do Fluxo do Sistema

## Casos de Uso

* + 1. Caso de Uso do Contexto



### Registra Entrada do Cliente no Parque

Atendente consulta no sistema o cadastro do cliente, se não houver cadastro ele realiza o cadastro do cliente pedindo dados de identificação, registra a entrada do cliente no parque e o associa a um dispositivo.



* + 1. Movimentação da Fila do brinquedo

Cliente consulta as opções de brinquedos disponíveis no parque. Quando decidir o brinquedo ele se cadastra na fila virtual, depois disso é chamado para ficar na fila física quando estiver se aproximando sua vez e logo em seguida entra no brinquedo. É permitida a saída do Cliente da fila desse brinquedo a partir do momento em que ele se cadastra nela.

****

* + 1. Execução do Brinquedo

O Operador do brinquedo deve dar início à execução do brinquedo e também finalizar a execução do brinquedo.



* + 1. Saída do Cliente do Parque

O Atendente deve registrar a saída do cliente associado ao dispositivo devolvido.



## 3.3 Modelagem do Banco de Dados

Utilizaremos o modelo Relacional para integração das funcionalidades do sistema, sendo que finalidade do modelo proposto é melhorar a integridade, utilizando a normalização de tabelas para reduzir redundâncias.

## 3.4 Materiais

A seguir serão detalhadas todas as ferramentas e tecnologias que foram utilizados no desenvolvimento do projeto.

### 3.4.1 Servidor de Aplicação JBoss Application Server

JBoss Application Server (ou JBoss AS) é um software livre (código aberto) em Java EE baseado em servidor de aplicação . Uma distinção importante para esta classe de software é que ele não só implementa um servidor que roda em Java, mas ele realmente implementa a parte do Java EE do Java. Porque é “Java –based”, ou seja, o servidor de aplicações JBoss opera plataformas: utilizáveis em qualquer sistema operacional que suporte Java . JBoss AS foi desenvolvido pela JBoss , atualmente uma divisão da Red Hat .

O *JBoss Application Server*  é uma plataforma Java certificada para desenvolvimento e implementação de aplicações corporativas e possui:

* Clustering
* Failover (incluindo sessões)
* O balanceamento de carga (Load Balancing)
* Cache Distribuído(usando JBoss Cache, um produto standalone)
* Deployment distribuído(farming)
* API Integrada
* Management API
* Apoio à Aspect-Oriented Programming (AOP)
* JSP / Servlet Tomcat 2.1/2.5
* JavaServer Faces 1,2 (Mojarra)
* Enterprise Java Beans nas versões de 3 e 2,1
* JNDI (Java Naming Directory Interface)
* Integração com Hibernate (para a programação a persistência; JPA )
* JDBC
* JTA (Java Transaction API)
* Suporte para Java EE , como Web Services JAX-WS
* SAAJ (SOAP com Anexos de API para Java)
* Integração com JMS (Java Message Service)
* JavaMail
* RMI-IIOP (JacORB, alias Java and CORBA)
* JAAS (Java Authentication Service e Autorização)
* Integração com JCA (Java Connector Architecture)
* Integração com JACC (Java Authorization Contract for Containers)
* Java Management Extensions

### 3.4.2 Plataforma J2EE

Java EE (ou J2EE, ou Java 2 Enterprise Edition, ou em português Java Edição Empresarial) é uma plataforma de programação para servidores na linguagem de programação Java.

A plataforma inicialmente era conhecida por Java 2 Platform, Enterprise Edition ou J2EE, até ter seu nome trocada para Java EE na versão 5.0, posteriormente foi chamada de Java EE 5, até a atualidade, J2EE, que esta na versão 1.4.

A Plataforma Java (Enterprise Edition) difere-se da Plataforma Java Standard Edition (Java SE) pela adição de bibliotecas que fornecem funcionalidade para implementar software Java distribuído, tolerante a falhas e multi-camada, baseada amplamente em componentes modulares executando em um servidor de aplicações. A plataforma Java EE é considerada um padrão de desenvolvimento já que o fornecedor de software nesta plataforma deve seguir determinadas regras se quiser declarar os seus produtos como compatíveis com Java EE. Ela contém bibliotecas desenvolvidas para o acesso a base de dados, RPC, CORBA, etc. Devido a essas características a plataforma é utilizada principalmente para o desenvolvimento de aplicações corporativas.

A plataforma J2EE contém uma série de especificações e containers, cada uma com funcionalidades distintas, sendo:

* JDBC (Java Database Connectivity), utilizado no acesso a bancos de dados;
* Servlets, são utilizados para o desenvolvimento de aplicações Web com conteúdo dinâmico. Ele contém uma API que abstrai e disponibiliza os recursos do servidor Web de maneira simplificada para o programador.
* JSP (Java Server Pages), uma especialização do servlet que permite que conteúdo dinâmico seja facilmente desenvolvido.
* JTA (Java Transaction API), uma API que padroniza o tratamento de transações dentro de uma aplicação Java.
* EJBs (Enterprise Java Beans), utilizados no desenvolvimento de componentes de software. Eles permitem que o programador se concentre nas necessidades do negócio do cliente, enquanto questões de infra-estrutura, segurança, disponibilidade e escalabilidade são responsabilidade do servidor de aplicações.
* JCA (Java Connector Architecture), uma API que padroniza a ligação a aplicações legadas.
* JPA (Java Persistence API), é uma API que padroniza o acesso a banco de dados através de mapeamento Objeto/Relacional dos EAP’s (Enterprise Java Beans).

### 3.4.3 Hibernate

O Hibernate é um framework para o mapeamento objeto-relacional escrito na linguagem Java, mas também é disponível em Microsoft .Net como o nome NHibernate. Este programa facilita o mapeamento dos atributos entre uma base tradicional de dados relacionais e o modelo objeto de uma aplicação, mediante o uso de arquivos (XML) para estabelecer esta relação.

O objetivo do Hibernate é diminuir a complexidade entre os programas Java, baseado no modelo orientado a objeto, que precisam trabalhar com um banco de dados do modelo relacional (presente na maioria dos SGDBs). Em especial, no desenvolvimento de consultas e atualizações dos dados.

Sua principal característica é a transformação das classes em Java para tabelas de dados (e dos tipos de dados Java para os da SQL). O Hibernate gera as chamadas SQL e libera o desenvolvedor do trabalho manual da conversão dos dados resultante, mantendo o programa portável para quaisquer bancos de dados SQL, porém causando um pequeno aumento no tempo de execução.

Nas questões relacionadas para o gerenciamento de transações e na tecnologia de acesso à base de dados são de responsabilidade de outros elementos na infraestrutura do programa. Apesar de existirem API no Hibernate para possuir operações de controle transacional, ele simplesmente delegará estas funções para a infraestrutura na qual foi instalada.

No caso de aplicações construídas para serem executadas em servidores de aplicação, o gerenciamento das transações é realizado segundo o padrão JTA. Já nas aplicações standalone, o programa delega o tratamento transacional ao driver JDBC.

Hibernate pode ser utilizado em aplicações Java standalone ou em aplicações Java EE, utilizando servlet ou sessões EJB beans.

Hibernate é um software livre de código aberto distribuído com a licença LGPL.

Hibernate foi criado por desenvolvedores Java, espalhados ao redor do mundo, e liderado por Gavin King. Posteriormente, JBoss Inc (empresa comprada pela Red Hat) contratou os principais desenvolvedores do programa para fazer o seu suporte.

A atual versão do Hibernate é a 3.x, que incorporou características como a nova arquitetura Interceptor/Callback, filtros definidos pelo usuário e anotações JDK 5.0 (Metadados do Java), que substitui os arquivos XML. Hibernate 3 também se aproxima das especificações EJB 3.0 e atua como a espinha dorsal das implementações EJB 3.0 em JBoss.

A HQL (Hibernate Query Language) é um dialeto SQL para o Hibernate. Ela é uma poderosa linguagem de consulta que se parece muito com a SQL, mas a HQL é totalmente orientada a objeto, incluindo os paradigmas de herança, polimorfismo e encapsulamento.

No Hibernate, você pode escolher tanto usar a SQL quanto a HQL. Escolhendo a HQL, você poderá executar os pedidos SQL sobre as classes de persistência do Java ao invés de tabelas no banco de dados, aumentando, assim, a distância entre o desenvolvimento da regras de negócio e o banco de dados.

### 3.4.5 RichFaces

RichFaces é um “open-source” Ajax habilitado para biblioteca de componentes JavaServer Faces criado por JBoss.org. Ele permite a fácil integração de Ajax capacidades em desenvolvimento de aplicações corporativas.

RichFaces é mais do que apenas uma biblioteca de componentes para JavaServer Faces. Ele acrescenta:

* Skinability (facilmente alterar e atualizar a aplicação - look and feel)
* Component Development Kit (CDK) para ajudar na construção de componentes JavaServer Faces
* Dynamic Resource Framework
* Componentes de controle baseado em Ajax.

RichFaces é oriundo do Ajax4JSF Framework que foi criado e desenhado por Alexander Smirnov. No outono de 2005, se juntou a Exadel Smirnov e continuou a desenvolver o framework. A primeira versão do que viria a ser Ajax4JSF foi lançado em março de 2006. Mais tarde no mesmo ano a estrutura Ajax4JSF e Rich Faces nasceu. Desenvolvedores especificam quais partes da página devem ser processados no servidor depois de algumas ações do usuário lado cliente e que as partes devem ser atualizadas após o processamento. Ajax4JSF tornou-se um projeto de código aberto hospedado no Java.net enquanto RichFaces tornou-se uma biblioteca de componentes JSF comercial.

Em março de 2007 a JBoss (agora uma divisão da Red Hat ) e Exadel firmaram uma parceria onde Ajax4JSF e RichFaces seriam agora sob o escudo da JBoss e será chamado de JBoss Ajax4JSF e JBoss RichFaces. Em setembro de 2007, JBoss e Exadel decidiram fundir Ajax4JSF e RichFaces sob o nome RichFaces. Concordaram que ambas as bibliotecas estariam agora livres e com código aberto. Tendo apenas uma versão do produto resolvendo muitos problemas de compatibilidade que existiam antes, como qual versão do Ajax4JSF funciona com a versão do RichFaces.

O framework é implementado como uma biblioteca de componentes que acrescenta Ajax em páginas existentes, para que o desenvolvedor não precise escrever nenhum JavaScript ou código para substituir componentes existentes com novos widgets Ajax. RichFaces permite amplo apoio na implementação de Ajax nas páginas. Assim, um programador pode definir o evento na página que invoca uma requisição Ajax e as áreas da página que devem ser sincronizados com o componente JSF Tree após as alterações Ajax pedindo os dados do servidor de acordo com os eventos disparados no cliente.

RichFaces permite que você defina (por meio de tags JSF) diferentes partes de uma página JSF que você deseja atualizar com um pedido de Ajax, e oferece algumas opções para enviar solicitações Ajax com o servidor. Além disso, a página JSF não passa de uma simples página JSF, ou seja, você não precisará escrever qualquer código JavaScript com a mão.

A arquitetura do RichFaces consiste de um filtro Ajax, Action componentes AJAX, AJAX contentores, e um motor de JavaScript.

* Ajax Filter - A fim de obter todos os benefícios do RichFaces, um desenvolvedor deve registrar um filtro no arquivo web.xml do aplicativo. O Filtro reconhece vários tipos de solicitação.
* AJAX Action Componentes - AjaxCommandButton, AjaxCommandLink, AjaxPoll e AjaxSupport e outros componentes de ação pode ser usado para enviar requisições Ajax a partir do lado do cliente.
* AJAX Containers - AjaxContainer é uma interface que descreve uma área em uma página JSF que devem ser decodificadas durante uma requisição Ajax. AjaxViewRoot e AjaxRegion são implementações desta interface.
* JavaScript Engine – O RichFaces JavaScript Engine funciona no lado do cliente. Ele atualiza diferentes áreas de uma página JSF com base nas informações da resposta do Ajax. O JavaScript Engine fornece uma API para que um desenvolvedor não precisa criar funcionalidade JavaScript própria.

Skinnability é uma característica especial do RichFaces que é usado para definir o estilo de interface comum. O recurso é baseado em XCSS tecnologia que proporciona maior flexibilidade e dinamismo. As propriedades do Schin, como generalBackgroundColor, generalLinkColor, headerFamilyFont etc são armazenadas em um arquivo skinname.skin.properties. Cada componente tem um XCSS (um formato de arquivo especial que combina flexibilidade do XML e CSS) que realiza o mapeamento de seletores CSS para as propriedades da pele de um Schin especial. Além disso, RichFaces oferece schins para controles HTML padrão. Você pode criar um Schin personalizado com a característica Plug-n-Skin, que é um arquétipo Maven que constrói um esqueleto para um novo Schin.

### 3.4.6 JavaServerFaces

JavaServer Faces é um framework MVC para o desenvolvimento de aplicações Web, que permite o desenvolvimento de aplicações para a internet de forma visual, ou seja, arrastando e soltando os componentes na tela (JSP), definindo propriedades dos mesmos.

O JavaServer Faces teve sua expressão na versão 1.1 quando implementado pela comunidade utilizando a especificação 127 do Java Community Process, evidenciando maturidade e segurança.

Hoje ele está na versão 1.2 da especificação 252 do JCP. A fundação Apache vem realizando esforços na implementação da especificação através do projeto MyFaces. O reconhecimento do trabalho é visto por diversas empresas, tanto é que a Oracle doou os fontes do ADF Faces, conjunto de mais de 100 componentes JSF, para o projeto MyFaces que o denominará de Trinidad.

O JSF é atualmente considerado pela comunidade Java como a última palavra em termos de desenvolvimento de aplicações Web utilizando Java, resultado da experiência e maturidade adquiridas com o JSP/Servlet (Model1), Model2 (MVC) e Struts.

As características do JSF são:

* Permite que o desenvolvedor crie interfaces através de um conjunto de componentes de interface pré-definidos;
* Fornece um conjunto de tags JSP para acessar os componentes;
* Reutiliza componentes da página;
* Associa os eventos do lado cliente com os manipuladores dos eventos do lado do servidor (os componentes de entrada possuem um valor local representando o estado no lado servidor);
* Fornece separação de funções que envolvem a construção de aplicações Web.
* Utiliza Ajax em alguns de seus componentes tornando alguns processos mais rápidos e eficientes.

### 3.4.7 JAAS

O Serviço de Autenticação e Autorização do Java (*Java Authentication and Authorization Service*), ou *JAAS*, é uma API que permite às aplicações escritas na plataforma J2EE usar serviços de controle de autenticação e autorização sem necessidade de a eles (aos serviços) estarem fortemente dependentes.

### 3.4.8 Banco de Dados ORACLE 10g XE

O Banco de Dados Oracle é um SGBD (sistema gerenciador de banco de dados) que surgiu no fim dos anos 70, quando Larry Ellison vislumbrou uma oportunidade que outras companhias não haviam percebido, quando encontrou uma descrição de um protótipo funcional de um banco de dados relacional e descobriu que nenhuma empresa tinha se empenhado em comercializar essa tecnologia.

Ellison e os co-fundadores da Oracle Corporation, Bob Miner e Ed Oates, perceberam que havia um tremendo potencial de negócios no modelo de banco de dados relacional tornando assim a maior empresa de software empresarial do mundo.

Além da base de dados, a Oracle desenvolve uma suíte de desenvolvimento chamada de Oracle Developer Suite, utilizada na construção de programas de computador que interagem com a sua base de dados. A Oracle também criou a linguagem de programação PL/SQL, utilizada no processamento de transações.

O Oracle *Express Edition,* ou Oracle *XE,* foi introduzido em 2005, oferece o Oracle 10g livre para distribuição nas plataformas Windows e Linux (com uma limitação de apenas 150 MB e restrita ao uso de apenas uma UCP, um máximo de 4 GB de dados de usuário e 1 GB de memória). O suporte para esta versão é feito exclusivamente através de fóruns on-line, sem o suporte da Oracle.

### 3.4.9 Dispositivo Móvel

### 3.4.10 Código de Barras

Código de barras é uma representação gráfica de dados numéricos ou alfanuméricos. A decodificação (leitura) dos dados é realizada por um leitor de código de barras, que emite um raio vermelho que percorre todas as barras. Onde a barra for escura, a luz é absorvida; onde a barra for clara (espaços), a luz é refletida novamente para o leitor. Os dados capturados nessa leitura óptica são compreendidos pelo computador, que por sua vez converte-os em letras ou números humano-legíveis.

Às 8:01 da manhã de 7 de outubro de 1974, um cliente do supermercado Marsh's em Troy, no estado estadunidense de Ohio, fez a primeira compra de um produto com código de barras. Era um pacote com 10 chicletes Wrigley's Juicy Fruit Gum. Isso deu início a uma nova era na venda a varejo, acelerando as caixas e dando às companhias um método mais eficiente para o controle do estoque. O pacote de chiclete ganhou seu lugar na história e está atualmente em exibição no Smithsonian Institute's National Museum of American History. Aquela compra histórica foi o ponto de partida para quase 30 anos de pesquisa e desenvolvimento.

O primeiro sistema para codificação automática de produtos foi patenteado por Bernard Silver e Norman Woodland, ambos estudantes graduados pelo Drexel Institute of Technology (Instituto de Tecnologia Drexel), atualmente (Drexel University). Eles usaram um padrão de tinta que brilhava debaixo de luz ultravioleta. Esse sistema era caro demais e a tinta não era muito estável. O sistema usado hoje foi descoberto pela IBM, em 1973, e usa leitores criados pela NCR.

O uso do código de barras - uma prática ligada à automação de processos nas empresas - levou cerca duas décadas para ser universalizado. Na Europa, segundo dados da EAN International, até 1981 poucos dos 21 países filiados à entidade utilizavam efetivamente o código. Em 1985, cerca de 90% das lojas automatizadas em todo o mundo estavam concentradas em somente seis países.

No Brasil, o Código Nacional de Produtos (código de barras) foi introduzido formalmente em 8 de novembro de 1984, através de decreto do presidente da República. Na mesma data, a portaria nº. 143 do Ministério da Indústria e Comércio conferiu à Associação Brasileira de Automação Comercial (ABAC) a missão de administrar o novo código em todo o país.

Em Portugal, o código de barras surgiu em 1985.

O código EAN/UPC é um sistema internacional que auxilia na identificação inequívoca de um item a ser vendido, movimentado e armazenado, sendo o EAN-13 o mais conhecido e utilizado mundialmente. A estrutura numérica do código (que geralmente fica abaixo das barras) representa as seguintes informações (tomando-se como exemplo o código 7898357417892):

* Os 3 primeiros dígitos representam o prefixo da organização responsável por controlar e licenciar a numeração no país (o prefixo 789 corresponde ao Brasil e 560, a Portugal);
* Os próximos dígitos, que podem variar de 4 a 7, representam a identificação do fabricante ou empresa proprietária da marca do produto; no exemplo é 835741 (6 dígitos);
* Os dígitos 789 representam a identificação do produto, e são atribuídos pelo fabricante;
* O último dígito 2 é chamado de dígito verificador e auxilia na segurança da leitura.

No total o código EAN-13 deve ter 13 dígitos. Vale ressaltar que os números da empresa variam de empresa para empresa, os números que identificam o item variam de item para item e o dígito verificador deve ser recalculado a cada variação na numeração. Existem outros tipos de códigos padrões para diversas aplicações.

Os códigos de barras dividem-se em dois grupos: os códigos de barras numéricos e os alfanuméricos (sendo os alfanuméricos capazes de representar números, letras e caracteres de função especial ao mesmo tempo). Os códigos de barras são diferenciados entre si pelas regras de simbologia. Cada simbologia trata como os dados serão codificados.

### 3.4.11 Leitor de Código de Barras

## 3.5 Métodos

Para falar a respeito dos métodos utilizados vamos separar o PROJETO F1 em alguns cenários:

### 3.5.1 Entrada do Parque

Na entrada do parque de diversão avaliamos que o usuário será uma atendente e ela que irá atender e cadastrar o cliente antes da entrada dele no parque. Entre as funções dele estarão consultar se o cliente já está no cadastro do sistema, se não estiver é necessário fazer o cadastro, antes de liberar a entrada do cliente no parque deve associar o celular do cliente ao código de barras que foi entregue a ele.

Essa é a tela de login para o atendente se autenticar no sistema.

***TELA 1***

Nessa tela o atendente verifica se o cliente já esta cadastrado, se não estiver ele realiza o cadastro.

***TELA 2***

Nessa tela ele associa o código de barras com o cliente e registra a entrada dele no parque.

***TELA 3***

### 3.5.2 Portaria do Brinquedo

Na portaria do brinquedo é onde o cliente se cadastra para entrar na fila do brinquedo. Na tela abaixo o cliente confirma o interesse de entrar realmente na fila após passar o código de barras pelo leitor.

***TELA 4***

E nessa tela informa ele a respeito do cadastro.

***TELA 5***

Teremos também essa tela com o relógio de estimativa sobre o tempo de fila para determinado brinquedo.

***TELA 6***

### 3.5.3 Terminal de Consulta

No terminal de consulta o cliente poderá verificar o tempo de fila para todos os brinquedos do parque. Ou até mesmo cancelar a participação em alguma fila.

***TELA 7***

### 3.5.4 Operação do Brinquedo

Na operação do brinquedo o operador apenas informará o início da execução do brinquedo e o término do mesmo.

***TELA 8***

# Capítulo 4 – TRABALHOS FUTUROS

## 4.1 Alternativas de Implementação

O Projeto F1 pode ter algumas alternativas para sua implementação. Nesse caso, para fazer a identificação do cliente foi escolhido o código de barras, mas poderia muito bem ser utilizada uma etiqueta RFID. E para notificar o cliente a sua vez no brinquedo foi utilizado mensagem SMS, mas poderia ser utilizada internet.

O funcionamento da solução com código de barras ou RFID no geral seria bem parecido, RFID seria uma opção mais ideal para atribuir novas funcionalidades ao sistema. Com RFID seria possível localizar o cliente em qualquer lugar do parque sem que ele tenha que passar a etiqueta em algum leitor, bastaria ter algumas antenas pelo parque para localizá-lo e com isso seria possível personalizar algumas situações, infelizmente o lado negativo do RFID é o seu custo, pois as etiquetas têm um preço não tão barato e principalmente os leitores e as antenas para espalhar pelo parque teriam um alto custo. O motivo pelo qual o código de barras foi escolhido foi exatamente pelo custo, etiquetas código de barras não têm custo elevado, assim como seu leitor.

A vantagem de utilizar mensagem SMS é a facilidade em integrar ao sistema. Mas um ponto muito negativo é o alto custo que pode ter e também a não garantia da mensagem ser entregue instantaneamente. Afinal, a demanda de envio de SMS seria muito alta, o que pode acabar com essa opção, para combater esse problema a alternativa seria o uso da internet, mas ai seria necessário o desenvolvimento de dispositivos para entregar ao cliente para uso dentro do parque. Foram analisados dois tipos de dispositivos, um é fabricado pela empresa GUMSTIX, é um dispositivo que daria a possibilidade de aumentar mais ainda as funcionalidades do sistema, pois ele tem memória interna, possui WI-FI, tem um mini-processador, e roda LINUX, o único motivo para ele não ter dado certo foi o alto custo, pois não existe aqui no Brasil, seria necessário importar dos EUA e isso acabaria com um altíssimo custo. Então a alternativa seria montar um PIC. PIC é um micro controlador montado com as especificações necessárias para cada caso, nesse caso seria necessário um PIC capaz de acessar a internet, ou seja, com WI-FI, e com uma tela para visualização. Como é um dispositivo montado especialmente para cada situação, seria a opção mais viável para o projeto, afinal seu custo não é elevado.

Utilizando internet o desenho da solução seria o seguinte:

# desenho solucao.jpg

# Capítulo 6 - Bibliografia

[1] RODRIGO, Enio. **O futuro das editoras universitárias e as mídias eletrônicas**. Disponível em <http://www.comciencia.br/comciencia/index.php?section=8&edicao=40&id=483>. Acesso em 18 de Agosto de 2009.

[2] NAKAZATO, Karen M.; & BARBOSA, Rafael S.; & KATSURAGI ,Raphael R.; & MARKS, Renan A.; & Branco, Rodrigo G. de; & DUARTE, Thales F. **Multimídias Moveis**. 2009. Departamento de Computação e Estatística

Universidade Federal de Mato Grosso do Sul (UFMS), Campo Grande, MS – Brasil, Cidade Universitária – Caixa Postal 549 – 79.070-900. Disponível em

< http://www.phpmobile.com.br/wp-content/caixa/2009/05/artigo\_multimidias\_moveis.pdf>

[3]GONZÁLEZ, Mario Orestes Aguirre; & AMORIM, Célio Gurgel; & RAMOS, Rubens Eugênio Barreto. **A relação entre a satisfação e a fidelidade dos clientes com a lucratividade das empresas**. 2004. XXIV Encontro Nac. de Eng. de Produção - Florianópolis, SC, Brasil, 03 a 05 de nov de 2004. Disponível em

< http://www.abepro.org.br/biblioteca/ENEGEP2004\_Enegep0201\_1771.pdf>

[4] JÚNIOR, Leônidas Vieira. **Tecnologia para gerenciamento de filas**. 2006. Specto Tecnologia na CIAB 2006. Disponível em

< http://www.specto.com.br/ciab2006/artigo\_gestao\_filas.pdf>

[5] BERNARDO, Cláudio Gonçalves. **A tecnologia rfid e os benefícios da etiqueta inteligente para os negócios**. 2004. Disponível em

<http://www.unibero.edu.br/download/revistaeletronica/Set04\_Artigos/A%20Tecnologia%20RFID%20-%20BSI.pdf >. Acesso em 19 de Agosto de 2009.

[6] COSTA, André Schwab; & GRAZZIOTIN Felipe Zanchet. **Controle de dispositivo utilizando mensagens SMS com tecnologia Java ME**. 2007. Disponível em

<http://www.inf.pucrs.br/~eduardob/disciplinas/ProgPerif/sem07.2/trabalhos/tp2/g4/Relatorio.pdf>.Acesso em 09 de Setembro de 2009.

[7] ARAÚJO, Marcus Augusto Vasconcelos; & ARAÚJO, Francisco José Costa; & ADISSI, Paulo José. **Distribuição da demanda telefônica de um call center através da criação e priorização de filas inteligentes**. 2003. XXIII Encontro Nac. de Eng. de Produção - Ouro Preto, MG, Brasil, 21 a 24 de Outubro de 2003. Disponível em

< http://producaoonline.org.br/index.php/rpo/article/viewFile/570/616>.

[8] SILVEIRA, José Verissimo Foggiatto; & RESENDE, Luis Maurício Martins de; & SCANDELARI, Luciano. **Uso do comércio eletrônico como prestação de serviços: envio de mensagens sms via telefone celular para informações instantâneas no agronegócio**. 2005. XXV Encontro Nac. de Eng. de Produção – Porto Alegre, RS, Brasil, 29 out a 01 de nov de 2005. Disponível em

< http://www.abepro.org.br/biblioteca/ENEGEP2005\_Enegep0902\_1524.pdf>