Documento de requisitos

**Projeto:**Controle de Estacionamento

**Responsáveis:**Anne Caroline,Ayrton Santos, Henrique Freitas.

# Registro de alterações

Versão 2.5

Responsável: Henrique Freitas

Data: 09/06/2013

Alterações: Complementação do requisito funcional REQ0005.

Versão 2.4

Responsável: Anne Castro

Data: 09/06/2013

Alterações: Revisão ortográfica.

Versão 2.3

Responsável: Ayrton Santos.

Data: 09/06/2013

Alterações: Complementação do requisitos funcionais REQF0001 eREQF0002, adição dos requisitos não funcionais REQN0005 e REQN0006.

Versão 2.2

Responsável: Henrique Freitas.

Data: 04/06/2013

Alterações: Correção no histórico de alterações, introdução, sessões e descrição do minimundo. Modificação do requisito REQF0005.

Versão 2.1

Responsável: Anne Castro

Data: 03/06/2013

Alterações: Revisão ortográfica.

Versão: 2.0

Responsável: Ayrton Santos

Data: 01/06/2013

Alterações: Introdução, Seções 1, 2, 3.1(AdiçãoREQF0006), 3.2(AdiçãoREQN0003 e REQN0004), e Adição da seção 4.

Versão: 1.1

Responsável: Henrique Freitas.

Data: 20/05/2013

Alterações: Adição do requisito não funcional REQN0002 - Linguagem de programação Java.

Versão: 1.0

Responsável: Ayrton Santos, Anne Caroline, Henrique Freitas.

Data: 20/05/2013

Alterações: Versão inicial

# Introdução

Este documento apresenta os requisitos de um sistema para a criação de um software que gerencie vagas de estacionamentos.Um guia para que os desenvolvedores tenham toda informação necessária.

# Visão geral do documento

**Seção 1:** Apresentação do minimundo. Esta seção descreve o ambiente em que o usuário se encontra.

**Seção 2:** Apresentação do produto. Esta seção descreve o produto que será desenvolvido para solucionar o problema do cliente.

**Seção 3:** Apresentação dos requisitos. Esta seção descreve como estão divididos os requisitos e como estão identificados.

**Seção 3.1:** Apresentação dos requisitos funcionais. Esta seção descreve os requisitos funcionais do produto.

**Seção 3.2:** Apresentação dos requisitos não funcionais. Esta seção descreve os requisitos não funcionais do produto.

**Seção 4:**Prototipação. Seção que fornece ao desenvolvedor mais clareza do produto e esclarece dúvidas finais sobre o sistema.

**Seção 4.1:**Interface Administrador ou Gerenciador.

**Seção 4.2:**Interface Usuário.

## Apresentação do minimundo

A cada dia, principalmente nas grandes metrópoles surgem novas ferramentas, métodos, maneiras de economizar tempo, pois nos dias atuais esse é um dos recursos mais preciosos, juntamente com dinheiro, e ambos estão diretamente relacionados.

Sair de carro em uma cidade como São Paulo é muito complicado e cansativo, exigindo muita paciência. Trânsito, faróis quebrados, avenidas interditadas, buracos, motoristas estressados, e como se não bastasse tudo isso, ainda é praticamente impossível achar uma vaga para estacionar o carro.

Locais proibidos, Faixas de Zona Azul\* que te obrigam a voltar no carro com certa frequência, insegurança, e mesmo quando se está disposto a pagar por serviços de qualidade, é extremamente difícil de encontrá-los.

Se você decidir parar por conta própria, encontrará muita dificuldade e provavelmente ficará rodando por um bom tempo até conseguir estacionar. Estatisticamente os paulistanos perdem em média de 30 a 60 minutos para encontrar uma vaga cada vez que vão ao Shopping de final de semana. No inverno ou em dias de chuva onde o público do Shopping é maior esses números aumentam. Apesar dos dados da estatística serem de picos, durante a semana a situação não muda muito.Temos claramente um problema com essa situação.

Contudo, o problema nem sempre é somente a falta de espaço. Muitas vezes os motoristas são obrigados a ficar rodando dentro do estacionamento à procura de uma vaga, frequentemente tendo que subir e descer os pavimentos. Algumas vezes podem até se deparar com uma seção inteira com vagas livres, apesar de ter ficado tanto tempo à procura de uma vaga.

Falta de espaço é um problema, mas o mau gerenciamento desse espaço é um problema maior ainda.

Conseguimos listar uma série de impactos pelo mau gerenciamento de espaço:

* Veículos circulando por minutos ou até horas nos corredores à procura de uma vaga para estacionar.
* Veículos são estacionados em áreas proibidas, no meio do corredor, dificultando a saída do motorista que está estacionado em uma vaga correta.
* Motoristas com necessidades especiais muitas vezes encontram dificuldades para achar uma vaga apropriada.
* Tráfego extra dentro e aos arredores do estacionamento gera mais trânsito nas ruas.

Dessa forma, a proposta do nosso sistema é criar um software que auxilie no gerenciamento dos estacionamentos, tornando-os “inteligentes”.

*\*Na cidade de São Paulo são vagas gerenciadas pela prefeitura e o órgão de Transito (CET).*

**2. Apresentação do produto**

O produto a ser criado é um software que tem como objetivo efetuar um correto gerenciamento das vagas livres em um estacionamento, fazendo com que estes estabelecimentos se tornem “inteligentes”, e seu espaço seja usado da forma mais proveitosa possível. Este software deve gerenciar o espaço do estacionamento da seguinte forma: coletar o status da vaga através de sensores, manter as vagas livres em uma sequência, para que a cada nova solicitação de vaga livre o sistema possa informar através de um mapa onde ela se encontra.

## 3. Apresentação dos requisitos

Os requisitos estão divididos em “funcionais” e “não funcionais”. Cada requisito possui uma identificação própria, nomeada com o prefixo “REQ[X]”, sendo o [X] uma letra “F” ou “N”, que diz se um requisito é funcional ou não funcional e uma numeração composta por 4 dígitos, que o distingue dos demais. Ex: REQF0001, REQN0002.

### 3.1. Apresentação dos requisitos funcionais

**REQF0001 – Detectar o status da vaga**

Descrição: O sistema deve captar o status de cada vaga através de sensores, se a vaga está livre ou não e atualizar isso na base. O sistema deve fazer isso com frequência e rapidez para manter a base de dados bem atualizada. É de extrema importância essa atualização para cobrir a falha de que um motorista pare em uma vaga a qual ele não foi direcionado ou trocou de vaga por qualquer que seja o motivo.

Prioridade: Alta.

**REQF0002 – Detectar prioridade da vaga**

Descrição: O sistema deve checar se uma vaga é comum (de acesso geral) ou prioritário (destinada a pessoas com necessidades especiais). Podem existir vagas exclusivas para funcionários de uma empresa ou reservadas para um evento, o sistema deve fazer um correto gerenciamento de separação dessas vagas também.

No caso das vagas destinadas a pessoas com necessidades especiais, caso não haja vagas disponíveis o sistema deve informar que não há vagas de prioridade disponível e direcionar o usuário/motorista a uma vaga comum.

Prioridade: Alta.

**REQF0003 – Exibir informações sobre vagas ao motorista**

Descrição: O sistema deve checar qual o tipo de usuário/motorista (comum ou prioritário), fazer uma correta busca na base de dados e exibir onde se encontra a vaga livre através de um mapa. O sistema também deve fazer uma impressão do local da vaga (por exemplo: vaga - B14) a qual o usuário foi direcionado.

Prioridade: Alta.

**REQF0004 – Confirmação de vaga utilizada**

Descrição: O sistema deve fazer uma checagem após direcionar um usuário/motorista para uma vaga durante “X” tempo, caso a vaga não tenha sido preenchida a mesma deve “voltar” com o status de livre para a base de dados.

Prioridade: Alta.

**REQF0005 – Exibição de relatórios**

Descrição: O sistema deve fornecer diversos relatórios como de rotatividade, uso e outros. Podem existir relatórios customizados de acordo com a necessidade de cada cliente. O relatório diário deve conter o número de vagas que foram ocupadas, a vaga que passou mais tempo ocupada, o setor e andar em que esta vaga se encontra, a vaga que foi menos ocupada, setor e andar em que esta vaga se encontra. O relatório por período deve conter a vaga, o setor e o andar maisacessados e a vaga, setor e andar menos acessados durante o período de um mês.

Prioridade: Média.

**REQF0006 – Acessos ao sistema**

Descrição: A interface de gerenciamento pode ser a mesma tanto para o administrador quanto para o gerenciador, permissões de acesso devem ser limitadas no código por contas de usuários.

Prioridade: Alta.

### 3.2. Apresentação dos requisitos não funcionais

**REQN0001 – Desenvolvido para a web**

Descrição: O sistema deve ser desenvolvido com uma interface de gerenciamento acessível via browser.

Prioridade: Alta.

**REQN0002 – Linguagem de programação Java**

Descrição: O software deve ser desenvolvido utilizando a linguagem de programação Java.

Prioridade: Média.

**REQN0003 – Serviço de passagem rápida (Sem parar)**

Descrição:Hoje em dia se torna cada vez mais frequente o serviço de passagem rápida, do tipo sem parar.

Caso o Estacionamento tenha esse serviço existem duas opções:

1. Como a tarifação é automática (sem impressão de ticket) após a validação do carro exibe-se a tela de escolha do usuário no caso comum ou prioritário. Após a escolha é exibido para qual vaga deve-se direcionar apenas liberando a cancela sem impressãode ticket (ou impressão apenas da localização da vaga).
2. Interação do software com as empresas (terceira) que oferecem esse tipo de serviço “sem parar” no país, atualmente são apenas três. Nada impede que ao cadastrar esse carro no sistema da empresa (terceira), ele seja identificado como um carro comum ou prioritário. No momento da validação o carro já estará cadastrado como preferencial ou não.

Prioridade: Alta.

**REQN0004 – Estacionamento lotado**

Descrição: Ação de quando o estacionamento estiver lotado: fica a critério do dono do estacionamento (cliente) do que fazer quando não houver mais vagas.

Exibir uma mensagem de que não há mais vaga e:

1. Liberar a entrada pedindo para aguardar por vagas.
2. Não liberar a entrada e pedir para aguardar.

Prioridade: Alta.

**REQN0005 – Segurança / “Cadastro de Veículos”**

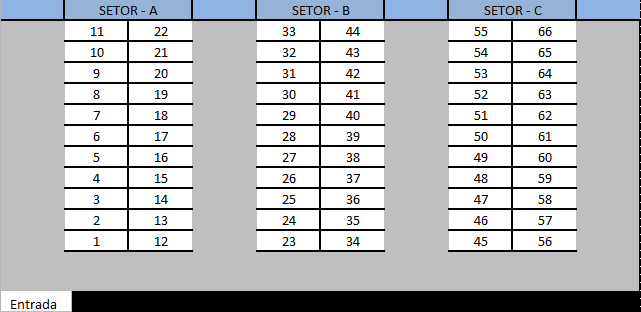
Descrição: Atualmente para oferecer um serviço de estacionamento é obrigatório por lei assegurar o veículo do usuário. Então é preciso fazer um cadastro dos carros (placa, cor, modelo). Nos estacionamentos onde o sistema é manual existe alguém anotando isso nos tickets, já nos sistemas automáticos existe a emissão do horário que o veículo entrou no estacionamento, e existe uma câmera gravando a entrada e a saída de veículos.

Sendo assim no sistema de controle de vagas é obrigatória a emissão do horário de entrada, sendo esse horário amarrado a uma sequência de código de barras. No momento da saída é dada baixa através do código de barras do veículo que entrou no horário “X”.

Prioridade: Alta.

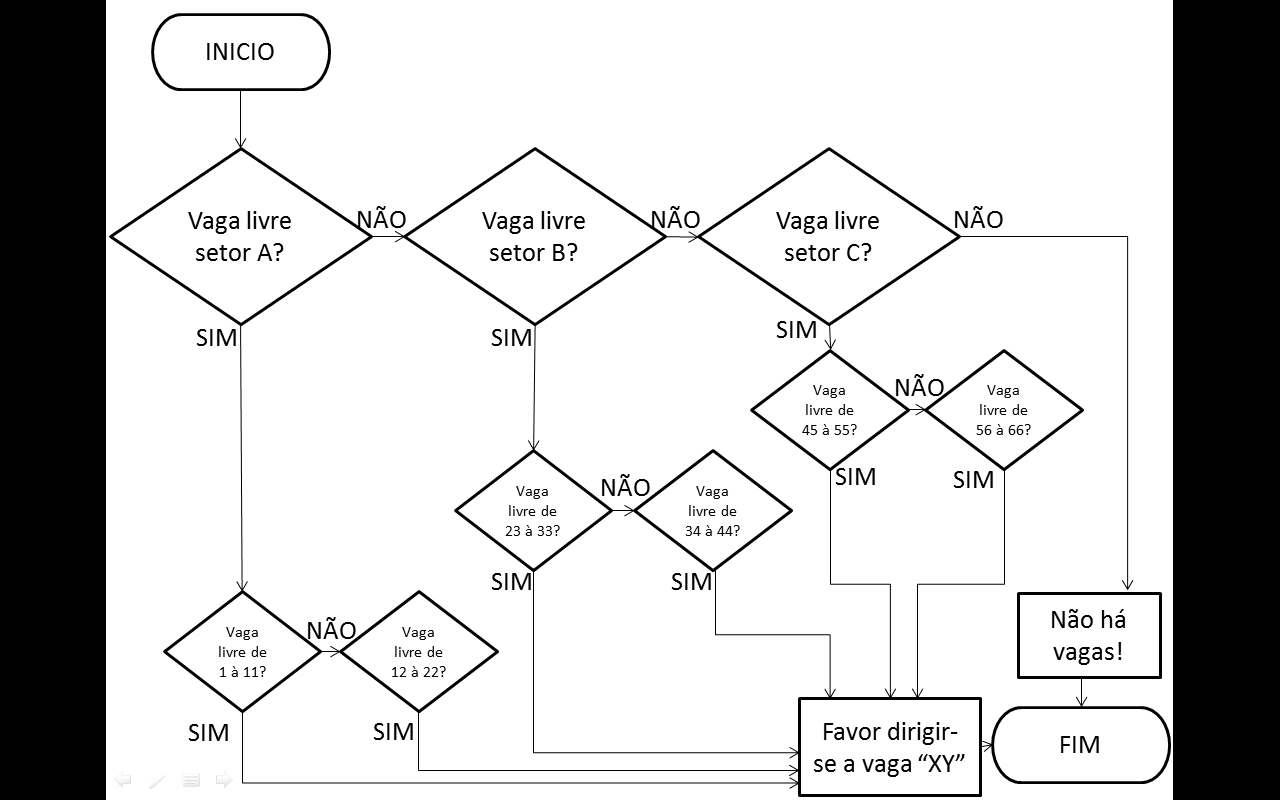
**REQN0006 – Distância das vagas versus rapidez para o usuário chegar até a mesma.**

Descrição: O sistema deve fazer um correto gerenciamento das vagas por proximidade. Isto está diretamente ligado com a disposição das vagas fisicamente e sua nomenclatura. Veja no exemplo abaixo:



As vagas do setor A, de um até onze são mais próximas da entrada, no primeiro corredor e relativamente mais perto.

Acompanhe no fluxograma abaixo um exemplo da lógica do sistema para busca de vaga:

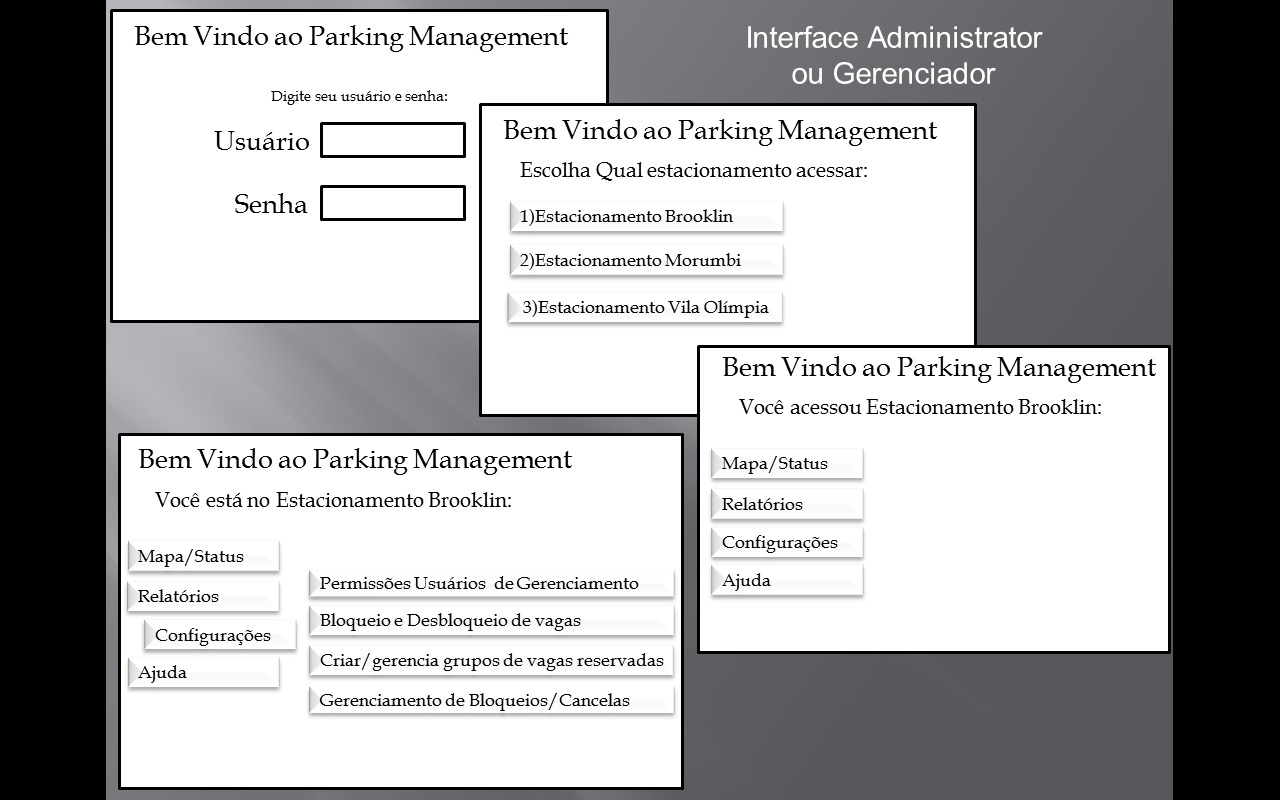


Nota: se existirem vagas do setor A reservadas para o usuário/motorista prioridade especial, isso também deve ser gerenciado na hora da busca.

Prioridade: Alta

## 4. Prototipação

### 4.1:Interface Administrador ou Gerenciador



Nota. Os menus apresentados são apenas exemplos, nada impede que novos menus e subs-menus sejam acrescentados (customizados) ou excluídos de acordo com a necessidade de cada cliente/estacionamento.

**4.2: Interface Usuário**

