**Debug Skate**

***Documento de Visão***

**Controle de Versão**

Grupo nº 6

Nome: Vitor Borges Vieira Fraga  
RGM: 11122100107

Nome: Hebert Pazian

RGM: 11122500873

Nome: Thiago Farias

RGM: 11122100627

|  |  |  |
| --- | --- | --- |
| **Responsável** | **Modificação** | **Data** |
| Hebert, Thiago e Vitor. | Tabulação do texto e alteração do objetivo do projeto | 19/02/2015 |
| Vitor | Atualização do documento | 26/05/2016 |
|  |  |  |
|  |  |  |

**Índice**

[1. Objetivo 4](#_Toc410249067)

[2. Necessidade do Negócio 4](#_Toc410249068)

[3. Descrição do Escopo 4](#_Toc410249069)

[4. Premissas 4](#_Toc410249070)

[5. Restrições 5](#_Toc410249071)

[6. Equipe 5](#_Toc410249072)

[7. Especificações Técnicas 5](#_Toc410249073)

[8. Riscos 6](#_Toc410249074)

# Objetivo

O software tem a finalidade de oferecer vendas de skates através de um sistema web, onde o mesmo é montado peça por peça, de acordo com o desejo e a disponibilidade delas para o cliente.

# Necessidade do Negócio

Atualmente o mercado de skate está muito popular em qualquer faixa etária e a prática desse esporte vai desde a diversão até campeonatos profissionais, porém os pontos disponíveis para compra do produto oferecem skates prontos sem a possibilidade de customização, sendo necessário um conhecimento técnico das peças do skate para compra individual das peças e posteriormente, para a montagem. Essa necessidade será atendida e os clientes poderão montar seus skates de acordo com sua preferência por um preço justo compatível ao mercado.

# Descrição do Escopo

O sistema define o usuário por quatro perfis de acesso, sendo o comum, atendente, gerente e administrador, onde o administrador terá acesso a todas as funções do software.

O site, ao iniciar, possuirá no cabeçalho um botão para o cliente fazer o seu cadastro, que ao concluir gerará um usuário com o perfil comum com os dados de login e senha para acessar o sistema. servindo para a realização de compras dos skates customizados. Após estar cadastrado, o cliente terá a opção de montar os seus skates para adicioná-los no seu carrinho, onde o mesmo poderá fazer alterações ou até exclusão de um skate montado antes de concluir a compra. Após a compra, um boleto é gerado para o pagamento.

Se o usuário for definido como perfil de atendente, além de possuir as informações que um usuário comum possui (realizando a montagem de skate e gerenciando o carrinho para um cliente que fará a compra direta pela loja e não pelo site), ele terá acesso ao módulo de gestão de peças, responsável pelo gerenciamento das peças, que terá uma opção de solicitação das mesmas que estão em falta, listando automaticamente elas e enviando um e-mail aos respectivos fornecedores, além da função de aprovar ou não a montagem dos skates comprados pelo cliente.

Com o perfil de gerente, todas as opções acima estarão disponíveis (exceto a função relacionada a montagem dos skates) junto com a gestão de pessoas, módulo responsável pelo gerenciamento dos dados dos clientes, fornecedores e funcionários, junto também com a função de gerar relatórios específicos para análise, e por fim a gestão de vendas, que irá aprovar ou não os pagamentos das compras efetuadas pelos clientes.

# Premissas

* Será necessário possuir para o desenvolvimento a plataforma NetBeans, com o servidor Tomcat ou Glassfish e o banco de dados PostgreSQL.
* Todos os três integrantes da equipe devem trabalhar com a mesma versão do projeto.
* O controle de versões será feito através da ferramenta do NetBeans, Git, que envia os arquivos para o armazenamento em nuvem chamado Bitbucket.
* A comunicação dos membros será feita pelo Whatsapp.
* A comunicação entre a equipe de desenvolvimento e o cliente será feita por e-mail/telefone/Skype.

# Restrições

* O software deverá ser desenvolvido utilizando a linguagem Java Web (JSP).
* A arquitetura do software estará necessariamente no padrão MVC.
* O projeto deverá ser finalizado em 18 meses.
* O banco deve ser necessariamente o PostgreSQL.
* Tanto o servidor quanto o cliente deverão obrigatoriamente estar conectados à internet para o funcionamento.

# Equipe

Os integrantes da equipe são: Hebert Pazian, Vitor Borges e Thiago Farias.

* Hebert Marcos Pazian, Técnico de informática formado em 2010 pela ETEC de Ferraz de Vasconcelos, Cursando ensino superior em sistemas de informação. Experiência em manutenção de computadores, manutenção em redes, levantamento de requisitos, analise funcional de sistemas, homologação de sistemas junto aos usuários, documentação da analise de sistemas.
* Thiago Farias dos Santos, Técnico de telecomunicações formado em 2007 pela ETEC Presidente Vargas, Cursando ensino superior em sistemas de informação e Ensino superior incompleto em redes.

Experiência como técnico de informática e técnico de telecomunicações.

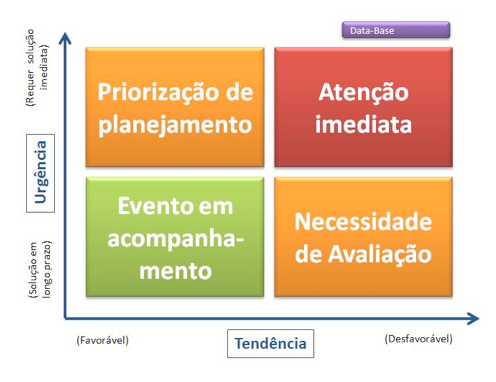
* Vitor Borges Vieira Fraga, Cursando ensino superior em sistemas de informação e inglês técnico avançado. Experiência em gestão administrativa de sistemas, Atendimento pessoalmente, via telefone e via e-mail a usuários do sistema,

# Especificações Técnicas

A aplicação DEBUG SKATE SHOP utiliza o padrão MVC e DAO em linguagem JAVA (JSP) para organizar todas as funcionalidades. É dividido em pastas e pacotes que condizem com as responsabilidades de cada componente do sistema, onde a pasta de Páginas Web contem todas as views do sistema, o pacote CONTROLLER contém as servlets responsáveis pelo controle, o pacote MODEL contém as classes de negócio da aplicação, o pacote MODEL/DAO contem as classes que realizam a persistência no banco de dados. O pacote UTIL é usado para disparo de e-mails, realizar a conexão ao banco, no caso, o SGBD PostgreSQL, gerar boletos ao final de um compra, gerar hash e por fim utiliza o quartz para realizar a programação de cancelamento de vendas.

As classes de negócio foram desenvolvidas com a preocupação para alta coesão e baixo acoplamento, aumentando assim a rastreabilidade de objetos e reuso de código. Classes DAO foram implantadas a fim de isolar ainda mais as regras de negócio. Desta forma, o banco de dados tem unicamente a responsabilidade de armazenamento de dados e para isso, o PostgreSQL é robusto e ideal para o porte da aplicação.

# Riscos



|  |  |  |  |
| --- | --- | --- | --- |
| **Risco #01** | **Probabilidade**  26% a 50% | **Impacto**  Severo | **Prioridade** (Probabilidade x Impacto)  **Alta** |
| **Descrição:** Atraso na entrega das tarefas | | | |
| **Plano de Ação:** Manter o cronograma definido, evitar com rigor a aproximação do andamento do projeto com o caminho critico e quando possível antecipar outros módulos. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Risco #02** | **Probabilidade**  0% a 25% | **Impacto**  Severo | **Prioridade** (Probabilidade x Impacto)  **Crítica** |
| **Descrição:** Perda de informação | | | |
| **Plano de Ação:** Manter backups de todas as versões do projeto em mais de um local físico e também em nuvem. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Risco #03** | **Probabilidade**  26% a 50% | **Impacto**  Moderado | **Prioridade** (Probabilidade x Impacto)  **Média** |
| **Descrição:** Mudança na equipe, caso um dos integrantes por força maior precise deixar o projeto. | | | |
| **Plano de Ação:** Todos os membros devem ter o conhecimento de todas as etapas do projeto e caso aconteça às tarefas serão realocadas. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Risco #04** | **Probabilidade**  0% a 25% | **Impacto**  Severo | **Prioridade** (Probabilidade x Impacto)  **Crítica** |
| **Descrição:** Falha no servidor | | | |
| **Plano de Ação:** Manter o servidor ligado no nobreak com capacidade de no mínimo 4 horas caso haja queda de energia e possuir um técnico disponível caso necessário de uma intervenção no servidor. | | | |