Falcon Framework

Glossário

Versão <0.1>

[Observação: O template a seguir é fornecido para uso com o Rational Unified Process. O texto em azul exibido entre colchetes e em itálico (style=InfoBlue) foi incluído para orientar o autor e deve ser excluído antes da publicação do documento. Um parágrafo digitado após esse estilo será automaticamente definido como normal (style=Body Text).]

[Para personalizar campos automáticos no Microsoft Word (que exibem um plano de fundo cinza quando selecionados), selecione File>Properties e substitua o conteúdo dos campos Title, Subject e Company pelas informações adequadas a esse documento. Depois de fechar a caixa de diálogo, para atualizar os campos automáticos no documento inteiro, selecione Edit>Select All (ou Ctrl-A) e pressione F9 ou simplesmente clique no campo e pressione F9. Isso deve ser feito separadamente para Cabeçalhos e Rodapés. Alt-F9 alterna entre a exibição de nomes de campos e a do conteúdo de campos. Consulte a ajuda do Word para obter mais informações sobre como trabalhar com campos.]

Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| <27/03/2016> | <0.1> | Especificação de termos para o Glossario | Willian |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Índice Analítico

1. Introdução 4

1.1 Finalidade 4

1.2 Escopo 4

1.3 Referências 4

1.4 Visão Geral 4

2. Definições 4

2.1 Framework 4

2.10 Ruby on Rails 5

2.11 NodeJS 5

2.12 Banco de Dados 5

2.13 Parse 5

2.14 App 5

2.15 GIT 5

2.2 RESTFUL 4

2.3 Web Service 5

2.4 BackEnd 5

2.5 EndPoint 5

2.6 SDK 5

2.7 Cliente-Servidor/Servidor-Cliente 5

2.8 API 5

2.9 DAO 5

2.16 5

2.17 5

2.18 5

2.19 <aGroupofTerms> 5

3. Estereótipos de UML 5

Glossário

# Introdução

## Finalidade

Este documento visa facilitar a utilização de termos comuns dentro do projeto e seus designados significados no contexto da aplicação. O objetivo principal do documento é de haja um entendimento comun entre os interessados de modo que se possa manter uma maior consistência da informação. Assim, consegue-se também definir alguns termos utilizados pelos Stakeholder mas que possuem algum significado diferente dentro do desenvolvimento da aplicação.

## Escopo

Este documento está associado com a aplicação Falcon Framework e também com todos os seus artefatos que são citados.

## Referências

|  |  |  |
| --- | --- | --- |
| Título | Versão | Onde pode ser obtido |
| Documento de Visão | 0.4 | Pasta do Projeto no Repositório |
| Plano de Iteração | 0.1 | Pasta do Projeto no Repositório |
| Especificação Suplementar | 0.1 | Pasta do Projeto no Repositório |
| Regras de Negócios | 0.1 | Pasta do Projeto no Repositório |

## Visão Geral

Este documento está organizado em seções e subseções da

seguinte forma:

Na seção 1, a Introdução, onde descreve sua importância e finalidade deste documento,

seguido do escopo, referência e da visão geral deste documento.

Na seção 2, apresentam-se as Definições, onde estarão os termos e abreviações do sistema,

organizados em ordem alfabética.

E por fim na seção 3, os Estereótipos de UML.

# Definições

Os termos utilizados no projeto serao descritos no formato de seções e subseções abaixo:

## Framework

Um framework em desenvolvimento de software, vem a ser uma [abstração](https://pt.wikipedia.org/wiki/Abstra%C3%A7%C3%A3o) que é composta por [códigos](https://pt.wikipedia.org/wiki/C%C3%B3digo) comuns entre vários projetos de software provendo assim uma funcionalidade genérica. Um framework pode fornecer uma funcionalidade específica durante a programação de uma aplicação por meio de configuração para a realização da mesma. A diferença entre biblioteca e framework está na designação do fluxo de controle da aplicação.

[Uma especificação detalhada sobre o que é um framework pode ser encontrada neste link.](http://www.codeproject.com/Articles/5381/What-Is-A-Framework)

## RESTFUL

REST define uma gama de princípios arquiteturais que podem ser utilizados para modelar Web service que focam em recursos do sistema. Seus quarto princípios básicos são:

* Utilização de metódo HTTP explicitamente;
* Não ter estados;
* Expor a estrutura de diretórios na URL
* Realizar transferência de XML, JASON ou ambos.

[Clique aqui para mais informações sobre os princípios arquiteturais RESTFUL](http://www.ibm.com/developerworks/library/ws-restful/)

## Web Service

Um Web service é uma coleção de protocolos e padrões utilizados para troca de dados entre aplicações ou sistemas. Geralmente XML é utilizado para a comunicação com um Web service que podem ser usados para a troca de dados entre computadores por meio da internet.

[Clique aqui para mais informações sobre Web services](http://www.tutorialspoint.com/webservices/what_are_web_services.htm)

## Front-End

Conhecido como o lado cliente da aplicação, frontend consiste basicamente na construção da camada visual para uma aplicação web ou website, para que desta forma o usuário possa interagir com o software desenvolvido diretamente.

## Back-End

Conhecido como o lado servidor de uma aplicação, backend consiste em três partes: um servidor, uma aplicação e um banco de dados. O backend de uma aplicação é responsável, por exemplo, pela persistência dos dados em um banco de dados.

[Clique aqui para mais informações sobre Frontend e Backend](http://blog.digitaltutors.com/whats-difference-front-end-back-end/)

## EndPoint

Um *EndPoint* de um Web Service é uma entidade, processador ou recusro que pode ser referenciado e para o qual as mensagens de um web Service pode ser endereçado.

## SDK – Software Development Kit

Geralmente é um conjunto de ferramentas que permite a criação de aplicações para certos pacotes de software, framework de software, plataforma de hardware, sistema computacional, console de video game, sistema operacional ou plataforma de desenvolvimento similar. Para que estas aplicações sejam desenvolvidas é necessário o download de um “kit para desenvolvimento de software” específico, como por exemplo SDK com java para uma aplicação android e iOS SDK para aplicativos com Swift.

## Cliente-Servidor/Servidor-Cliente

## API – Application Program interface

## DAO – Data Access Object

## Ruby on Rails

## NodeJS

## Banco de Dados

## Parse

Parse é uma aplicação Backend que foi recentemente adiquirida pelo Facebook que consiste em um serviço de nuvem para armazenar APIs que um aplicativo mobile se comunica para funcionar. Também é muito útil para enviar notificações para os usuários do aplicativo, de modo a se comunicar com eles. Parse provém um serviço backend para desenvolvedores. Se uma aplicação para dispositivos móveis ou para desktop requer um backend na internet, Parse é uma das opções disponíveis.

## App – Application

App consiste apenas na sigla, derivada do ingles, para aplicativo ou aplicação. Muito utilizada nos contextos atuais de desenvolvimento de aplicativos mobile, tanto para iOS, Android e Windows Mobile.

## GIT

É um sistema grátis e de distribuição open source para o controle de versionamento que gerencia com eficiência e rapidez desde de pequenos a grandes projetos. Possui diversars características que fizeram do Git um dos sistemas de versionamento mais utilizados na comunidade de software livre e em diversos projetos desenvolvidos ao redor do mundo.

## 

## 

## 

## <aGroupofTerms>

# Estereótipos de UML

[Esta seção contém ou faz referência a especificações de estereótipos de Linguagem de Modelagem Unificada (UML) e suas implicações semânticas — uma descrição textual do significado e da importância dos estereótipos e de quaisquer limitações de seu uso — para estereótipos já conhecidos ou que se mostraram importantes para o sistema que está sendo modelado. O uso desses estereótipos pode ser simplesmente recomendado ou até mesmo obrigatório; por exemplo, quando o uso desses estereótipos for exigido por um padrão imposto ou quando se considerar que o uso facilitará em muito o entendimento. Esta seção pode ficar em branco se nenhum estereótipo adicional, além daqueles predefinidos pela UML e o Rational Unified Process (RUP), for considerado necessário.]