‘01 – Arquitetura You Troll

RELATÓRIO DE DEFINIÇÃO DA ARQUITETURA

Versão 1.1

**Responsável(eis) pelo Relatório:**

Fabiano Carneiro de Oliveira – fabiano.lothor@gmail.com

Ítalo Lelis de Vietro – italolelis@gmail.com

João Neto – joaoneto20@gmail.com

Rodolfo Almeida - rodolfoidez@gmail.com

Histórico de Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 24/08/2012 | 1.0 | Criação do documento | Equipe |
| 29/08/2012 | 1.1 | Correção de pontos no documento | Ítalo Lelis de Vietro |

Arquitetura do projeto

# Projeto

**Data de criação**: 24/08/2012

**Nome do projeto:** 01 – You Troll

**Resumo básico do projeto:**

O Youtroll é uma solução de software que facilitará a visualização, criação, publicação e divulgação de imagens cômicas na internet. Fará divulgação automática em todas as redes sociais, acompanhamento de usuários, possuirá um sistema de pesquisa robusto e divulgação gratuita na página inicial do site.

A ideia da aplicação é centralizar o conteúdo cômico que roda pela internet nos dias de hoje, com uma aplicação dessa o autor das tirinhas poderia ter uma crédito maior pelas suas “obras”, além de ter um canal personalizado para realizar suas postagens.

O usuário do sistema poderá inscrever-se em outros canais que sejam de seu interesse e receber as atualizações mais recentes diretamente em sua página inicial.

## Sugestões Futuras

A aplicação também tentará inovar, permitindo que o usuário faça suas tirinhas diretamente pelo site em um painel de desenho de tirinhas, após concluído o desenho, ela poderá ser publicada em qualquer rede social que o usuário está cadastrado ou poderá simplesmente ser compartilhada por outros usuário em diversas redes.

# Arquitetura adotada

## Cliente – Servidor

Utilizaremos a arquitetura de cliente – servidor para lidar com várias requisições dos usuários. Essa arquitetura nos fornece ótima flexibilidade para escalar e lidar com diversas requisições dos usuários.

Preparar uma aplicação para escalar é uma forma de garantir que o serviço estará sendo entregue com qualidade, sem quedas no servidor e aplicações lentas.

Teremos um servidor central que contará com um proxy, ele irá redirecionar para uma instância da aplicação, onde será processada toda a lógica da aplicação.

Ainda teremos um servidor de banco dados onde a aplicação irá se conectar a persistência.

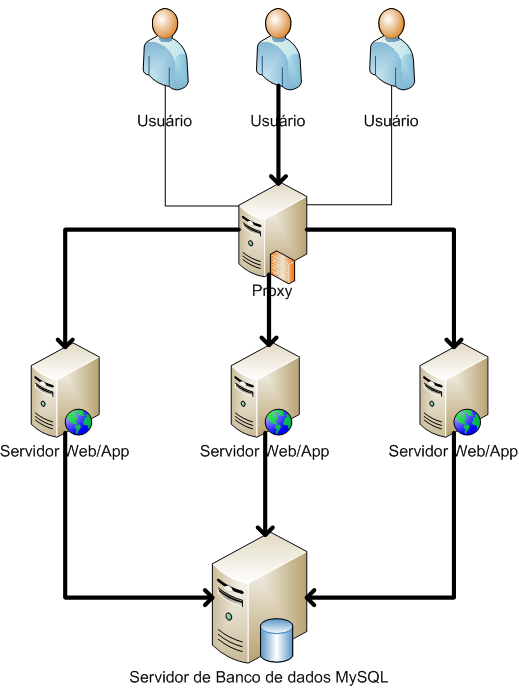


Figura 01: Arquitetura Básica

Para uma release futura, pretendemos utilizar uma arquitetura mais robusta, onde o usuário fará requisições para o servidor, irá passar por uma camada de cache, caso a informação não esteja no cache, será redirecionada para a camada de aplicação para processar a lógica da aplicação.

Também haverá uma camada de persistência, que contará com servidores de banco de dados Master que irão lidar com as operações de Inclusão, Alteração e Exclusão, e servidores Slaves que serão responsáveis pela leitura dos dados. Cada vez que um dado é incluído nos bancos Masters, eles devem fazer a replicação desses dados para os bancos Slaves para que possam ser lidos.

Nessa camada teremos um balanceamento de carga para as requisições de leitura, onde saberemos qual o servidor com maior disponibilidade para processar aquela requisição.

Um servidor de CDN (Content Delivery Network) servirá para requisições de conteúdo estático, evitando a sobrecarga dos servidores de aplicação, trazendo velocidade e flexibilidade para escalar. Esse servidor poderá ser alugado em alguma solução de nuvem como a Amazon ou IBM.

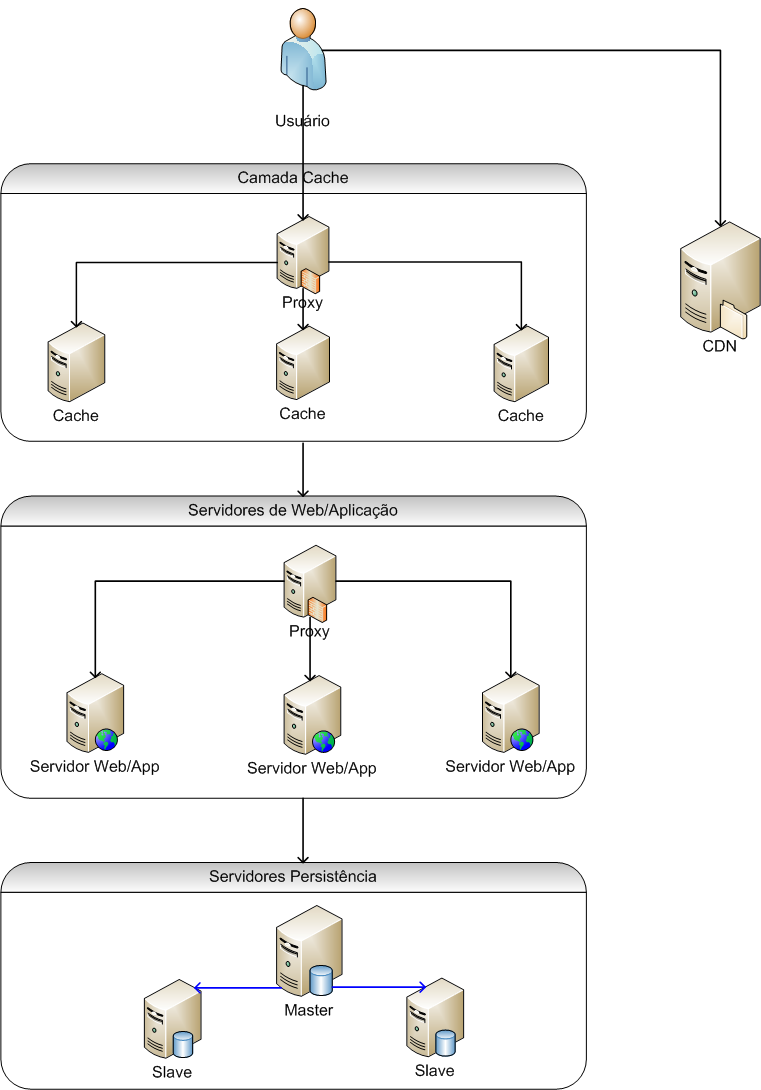


Figura 2 – Arquitetura Completa

## MVC

Utilizamos a arquitetura de design Model View Controller pois, ela faz a separação da lógica de negocio da apresentação, deixando o código mais organizado e facilitando a manutenção da aplicação. Como o MVC é altamente componentizável ele ajudará no reaproveitamento de código.

Nossa aplicação irá receber e processar uma requisição do usuário na camada de Controllers, nossa lógica de Modelos estará na camada REST e somente através dessa camada poderemos nos conectar com a persistência.

## Framework de desenvolvimento

Iremos utilizar o Yii Framework para o desenvolvimento pois, esse framework disponibiliza diversas bibliotecas auxiliares para agilizar o desenvolvimento como classes de Data, Internacionalização, cache entre outras. Ele também fornece o padrão de design MVC que ajudará na separação e reaproveitamento do código.

O Yii facilita a internacionalização do sistema, implementação de cache, arquitetura CDN, herança de templates além de outras facilidades.

## Visualizações

A aplicação web utilizara a linguagem de marcação HTML5 juntamente com CSS3 e jQuery, com isso poderemos acessar a API da aplicação e interagirmos com ela, além de poder fazer diversos efeitos

## Banco de Dados

MySQL foi o banco escolhido para persistirmos dados na aplicação. O MySQL fornece uma abordagem relacional, que será boa para o começo do projeto. Além disso o banco fornece Particionamento de tabelas para escala vertical dos dados, fornece também proxys de conexão, isso facilitara a distribuição e replicação do banco.

## Repositório de Código

Utilizaremos o Git para o nosso controle de versão da aplicação. O git fornece uma boa arquitetura para trabalharmos com diversas versões como branches para separação de versões, tags para download de código, gitmodules para dependência de projetos, comparação de código, etc....

# Aplicações envolvidas na solução do Youtroll

A Figura 1 apresenta o modelo que a aplicação irá utilizar para se comunicar com outras API’s, como Google, Facebook e Twitter. Essas API’s serão utilizadas para fazer login no Youtroll, realizar postagens com as tirinhas em sua conta ou em uma página nas redes sociais.

Além dessas conexões a aplicação ainda poderá compartilhar diversas tirinhas nas redes sociais.

Um usuário poderá realizar um cadastro para acessar a aplicação, ou simplesmente utilizar o cadastro existente em qualquer uma das três redes sociais listadas. Para isso o Youtroll armazenará o usuário da rede social e um tooken para acesso futuro.

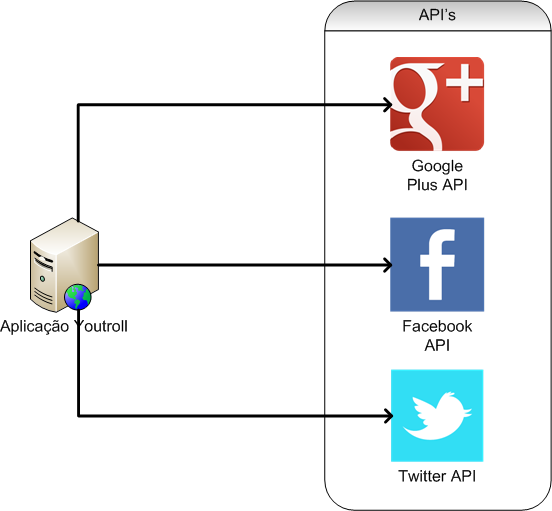


Figura 1 – Integrações das Redes Sociais

# Interessados e contatos

No diagrama da Figura 2, mostramos que a aplicação terá uma camada REST para integração com diversos sistemas de forma flexível e escalável. Teremos essa camada, onde o usuário irá interagir com a Aplicação Web Youtroll para realizar suas operações, essa aplicação irá interagir com a API para receber e enviar dados.

Aplicações Mobile também poderão se integrar com o serviço, aplicações nativas ou PhoneGap para Android, iPhone, iPad e Tablets em geral, irão se comunicar diretamente com a API para envio e recebimento de dados.

Qualquer outra aplicação pode interagir com a API, basta ter um cadastro da aplicação no Banco de dados do Youtroll, com isso a aplicação terá tookens para acessar a API e solicitar informações.

Esse trafego de dados irá trabalhar e cima do protocolo oAuth 2.0, isso trará uma maior confiabilidade para o sistema e integridade dos dados.

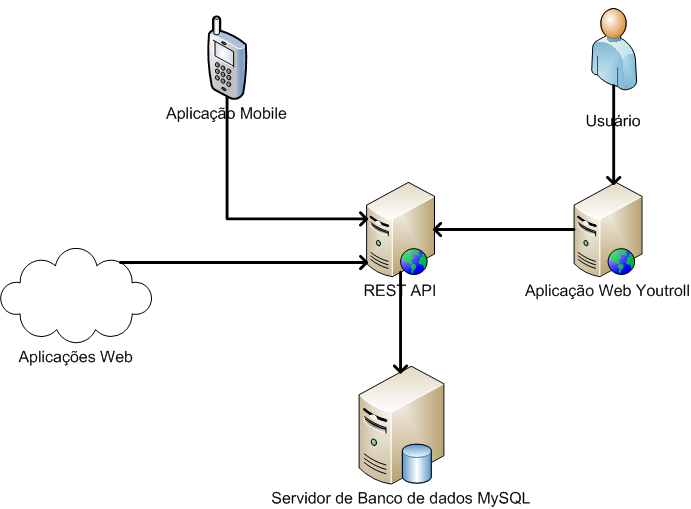


Figura 1 – Integrações com o Youtroll

# Interessados e contatos

ERIKO BRITO – <EMAIL STAKEHOLDER>– GERENTE DE PROJETOS

LUCAS VIEIRA – <EMAIL STAKEHOLDER>– ARQUITETO DE SOFTWARE

ÍTALO LELIS DE VIETRO– [italolelis@gmail.com](mailto:italolelis@gmail.com) – ENGENHEIRO DE SOFTWARE

FABIANO CARNEIRO – fabiano.lothor@gmail.com – ENGENHEIRO DE SOFTWARE

RODOLFO ALMEIDA – rodolfoidez@gmail.com – ENGENHEIRO DE SOFTWARE

JOÃO DE ARAÚJO NETO – [joaoneto20@gmail.com](mailto:joaoneto20@gmail.com) - ENGENHEIRO DE SOFTWARE