

# Criando e Adaptando Bibliotecas do Ruby

Gustavo Toshi Komura

Universidade Federal do Paraná - UFPR  
Ciência da Computação  
Departamento de Informática  
Orientador: Professor Doutor Bruno Müller Junior

Curitiba - 14 de Dezembro de 2014

# Sumário

- 1 Introdução
  - Objetivos do Trabalho
- 2 Bibliotecas do Ruby
  - Conceito e Formas de Distribuição
  - Processo de Criação
  - Uso da gema `gemtranslatetoenglish`
- 3 Bibliotecas do Ruby Mapeando API Google Maps
  - Processo de Adaptação
  - Funcionalidades Google-Maps-For-Rails
  - API Google-Maps-For-Rails
  - Proposta de Novas Funcionalidades
  - API Google-Maps-For-Rails Adaptada
- 4 Conclusão

# Objetivos do Trabalho

## Objetivos

Apresentar:

- Conceitos sobre as bibliotecas do **Ruby**.
- O processo de criação de uma biblioteca do **Ruby**.
- O processo de adaptação de uma biblioteca do **Ruby**.
- A adaptação de um biblioteca do **Ruby** que faz o mapeamento da **API** do **Google Maps**.
- Exemplos de uso das bibliotecas do **Ruby**.

# Conceito e Formas de Distribuição

## Conceito

Biblioteca do **Ruby** é um conjunto de funções.

## Formas

- Forma **arquivo compactado** em `.tar.gz` ou `.zip`: Forma menos utilizada. Seu processo de instalação é feito por meio de arquivos `README` ou `INSTALL`.
- Forma **gem**: Forma mais utilizada para distribuição, por isso as bibliotecas do **Ruby** são conhecidas como **gems/gemas**. Seu processo de instalação é feito por meio do programa **gem**.

## Programa gem

- Similar ao sistema de distribuição **apt-get**.
- Executado em linhas de comando no formato **"gem <operação>"**, onde **"<operação>"** pode ser **install**, **search**, etc.

# Processo de Criação

## Processo

- 1 Criar a estrutura automaticamente

```
1 bundle gem 'nome da gema'
```

Código 1: Cria Gema Forma Geral

- 2 Editar “**‘nome da gema’.gemspec**” para definir as descrições da gema.
- 3 Desenvolver códigos de funcionalidades (diretório **lib**, caso seja código **Ruby**).
- 4 Desenvolver códigos de testes (diretório **test** ou **spec**).
- 5 Determinar versão para a **gema**.
- 6 Publicar a gema (**gem build “nome da gema”**).

# Uso da gema `gemtranslatetoenglish`

## Adiciona gema no Gemfile

Acrescenta a gema de tradução no Gemfile do projeto

```
1 gem 'gemtranslatetoenglish'
```

Código 2: Adiciona `gemtranslatetoenglish` no Gemfile

## Funcionalidade na View

Utiliza a funcionalidade de tradução da nova gema em uma view

```
1 translate("Olá")
```

Código 3: Exemplo de `gemtranslatetoenglish` na View

# Processo de Adaptação

## Processo

- 1 Realizar **engenharia reversa** para gerar diagramas de alto nível.
- 2 Entender a biblioteca por meio dos diagramas de alto nível.
- 3 Realizar adaptações.

# Funcionalidades Google-Maps-For-Rails

## API Google Maps e Gema Google-Maps-For-Rails

- **Google-Maps-For-Rails** mapeia **API** do **Google Maps**.
- **API Google Maps** utiliza a linguagem **Javascript**.
- **Google-Maps-For-Rails** utiliza **CoffeeScript** -> **Javascript**.

## Funcionalidades

- Cria mapas do **Google Maps** em páginas web.
- Permite a criação de mais de um mapa na mesma página (id).
- Cria sobreposições para os mapas (markers, circles, polygon, polylines, etc).
- Permite a criação de várias sobreposições do mesmo tipo com a chamada de uma única função.
- Permite readaptação do **core (Bing Maps)**.



# API Google-Maps-For-Rails

```
1 handler = Gmaps.build("Google")
2 handler.buildMap
3   provider: {}
4   internal:
5     id: "map"
6 , ->
7 markers = handler.addMarker([
8   lat: 0
9   lng: 0
10  picture:
11    url: "https://addons.cdn.mozilla.net/img/uploads/
12    addon_icons/13/13028-64.png"
13    width: 36
14    height: 36
15 ])
```

Código 4: Exemplo CoffeeScript API Google-Maps-For-Rails

# Proposta de Novas Funcionalidades

## Novas Funcionalidades

- Gerar direções entre dois lugares no mapa.
- Gerar direções entre dois lugares com pontos intermediários obrigatórios.
- Permitir a troca de cores entre os tracejados entre dois pontos do caminho.
- Permitir a escolha do veículo de locomoção.
- Mostrar detalhes sobre o caminho (tempo de percurso, distância, etc).

# API Google-Maps-For-Rails Adaptada

## Criar Direções

Por meio de código **CoffeeScript**:

```
1 $(document).ready(->
2   handler = Gmaps.build("Google")
3   handler.buildMap
4     provider: {}
5     internal:
6       id: "map"
7   , ->
8     handler.addDirection( { origin: "Insert Origin Location"
9       , destination: "Insert Destination Location"}, options)
10   return
11 )
```

Código 5: Exemplo CoffeeScript API Google-Maps-For-Rails Adaptado

# Trabalhos Futuros

## Trabalhos Futuros

- Para a primeira gema de exemplo **gemtranslatetoenglish**:
  - Abordar mais recursos da linguagem **Ruby**.
- Para a segunda gema **Google-Maps-For-Rails** adaptada:
  - Incluir a possibilidade de mostrar mais detalhes do percurso (steps).
  - Incluir a possibilidade de escolher o tipo de caminho a ser percorrido (mais curto, menor tempo, etc).

# Conclusão

## Conclusão

Aprendemos:

- Conceitos sobre as bibliotecas do **Ruby**.
- Como criar uma biblioteca do **Ruby**.
- Como adaptar uma biblioteca do **Ruby**.
- A utilizar a **API** do **Google Maps**.
- Conceitos sobre a linguagem **CoffeeScript**.
- Como adicionar funcionalidades em uma biblioteca do **Ruby** que mapeia a **API** do **Google Maps**.