

INSTITUTO FEDERAL

Paraná

Campus Paranaguá

Gems e Bundler

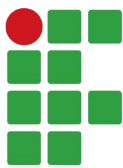
Desenvolvimento Web III

Prof. Diego Stiehl

Gems

- Gemas
 - Pedras em formato rústico
- Gems são as bibliotecas do Ruby
 - Partes de código que precisam ser “lapidadas”
 - Conjunto de arquivos reutilizáveis
- Site oficial: <https://rubygems.org>
 - Repositório oficial de gems
- Em Ruby (e Rails) utilizaremos muito



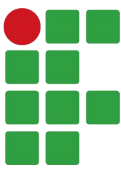


Gems

- Mas por que isso agora?
 - Ruby on Rails foca em produtividade
- Não reinventar a roda
 - Existem milhares de gems para diversos fins
 - 145.691 registradas em <https://rubygems.org>
- O próprio framework Ruby on Rails é uma gem disponível no repositório
 - <https://rubygems.org/gems/rails>

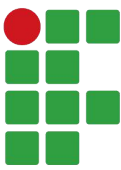
Comando: gem

- Permite o download e gerência de gems
 - Busca no repositório RubyGems
- Vem junto a uma instalação padrão de Ruby
- Digitar no terminal
 - gem
 - Funcionou?



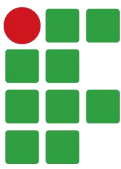
Comandos

- `gem list`
 - Lista todas as gems locais (instaladas)
 - Mostra as versões instaladas
- `gem list <gem>`
 - Lista apenas as gems que se encaixam
 - `gem list did_you_mean`



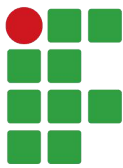
Comandos

- `gem list <gem> --remote`
 - Mostra todas as gems remotas que se encaixam com o termo informado
 - Mostra a última versão
- `gem list <gem> --remote --all`
 - Mostra todas as versões disponíveis (remotas)



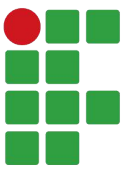
Comandos

- `gem install <gem>`
 - Instala a última versão de determinada gem
 - `gem install cpf_utils`
 - `gem install lerolero_generator`
 - `gem install rails`
- `gem install <gem> -v <versão>`
 - Instala a versão específica de uma gem
 - `gem install cpf_utils -v 1.0.0`
 - `gem install rails -v 4.0.0`



Comandos

- `gem uninstall <gem> -v <versão>`
 - Remove uma gem instalada localmente
 - `gem uninstall cpf_utils -v 1.0.0`



Comandos

- `gem cleanup`
 - Remove todas gems com antigas
- `gem cleanup <gem>`
 - Remove todas as versões antigas de uma gem
 - `gem cleanup cpf_utils`
- `gem cleanup -d`
 - Informa todas gem com mais de uma versão instaladas localmente

Usando uma Gem

- As gems são baixadas para um diretório conhecido do Ruby
- Para usá-las, basta “pedir” por uma delas
- Comando (arquivo .rb):
 - `require 'nome_da_gem'`
 - Não utilizar `require_relative`
- Exemplo
 - `require 'cpf_utils'`

Versão Específica

- Precisa de uma versão específica da gem?

```
gem 'nome_da_gem', 'versão'  
require 'nome_da_gem'
```

- Exemplo:

```
gem 'cpf_utils', '1.0.0'  
require 'cpf_utils'
```

Brincando com Algumas Gems

- CpfUtils (cpf_utils)
 - https://github.com/jacksonpires/cpf_utils
- LeroleroGenerator (lerolero_generator)
 - https://github.com/jacksonpires/lerolero_generator
- Faker (faker)
 - <https://github.com/stympy/faker>

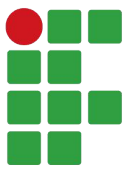
Bundler

- Site oficial: <http://bundler.io>
- Gerenciamento de dependências em aplicações Ruby
 - Instalação de Gems e suas dependências
- Semelhante ao Composer do PHP
- Possui um arquivo de configuração
 - Com uma linha de comando atualiza todas as dependências da aplicação



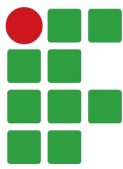
Instalação e Uso

- O Bundler é uma gem
 - Normalmente vem com a instalação padrão
- Verificar se está instalado
 - `bundle -v`
- Se não estiver:
 - `gem install bundler`
 - Reabrir terminal e testar novamente:
 - `bundle -v`



Gemfile

- O Bundler procura por um arquivo no diretório corrente
 - Gemfile
- Lê o conteúdo deste arquivo
 - Baixa e instala gems e suas dependências
- Gera um arquivo de saída
 - Gemfile.lock
 - Apresenta toda a árvore de dependências do projeto



Gemfile

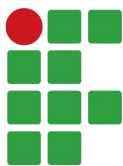
- O Gemfile sempre vai estar na raiz do nosso projeto
 - O arquivo Gemfile.lock vai ser criado ao lado
- Outros arquivos do projeto podem simplesmente usar as gems informadas no Gemfile
 - `require 'nome_da_gem'`

Conteúdo do Gemfile

- Primeira linha
 - `source 'https://rubygems.org'`
 - Informa qual o repositório para download
- As demais linhas informam as gems
 - `gem 'lerolero_generator'`
 - `gem 'cpf_utils'`
 - `gem 'faker'`

bundle install

- Para que alterações no Gemfile tenham efeito
 - Executar comando (terminal)
 - `bundle install`
- Vai ler o arquivo, verificar o que há de diferente, requisitar do repositório e instalar localmente

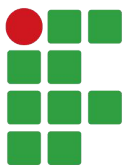


Versões

- Podemos informar a versão desejada de determinada gem
 - gem 'lerolero', '1.0.1'
 - Versão exata: 1.0.1
 - gem 'cpf_utils', '>=1.0.0'
 - Qualquer versão maior ou igual a 1.0.0
 - gem 'faker', '~>1.6.0'
 - Varia somente último número:
 - Maior que 1.6 e menor que 1.7 (1.6.1, 1.6.5, 1.6.99, ...)

Semantic Versioning

- As gems seguem um formato de marcação de versão criado (e usado) pelo GitHub
 - <http://semver.org/lang/pt-BR>
 - Leitura válida
- Versões são especificadas no formato:
 - Major.Minor.Patch
 - Major: Novas versões quebram compatibilidade
 - Minor: Adição de funcionalidades que não quebram a compatibilidade
 - Patch: Pequenas correções de falhas



Prática

- Considere as mesmas três gems de antes
- Crie um “projeto” Ruby
- Importe as gems usando o Bundler
- Em arquivos .rb (no projeto), utilize algumas funções de cada gem
- Verificar o que houve em:
 - gem list

Prática - Trabalho RPG

- Adicione ao seu trabalho
 - Geração aleatória de nomes de personagens com a gem Faker
 - Uma categoria diferente do Faker para cada raça