

# DOMINANDO O USO DE JOBS NO RAILS





# ENTENDENDO OS JOBS



# O QUE SÃO JOBS?





# O QUE É O ACTIVE JOBS?



## Onde são utilizados Jobs?

- ◆ Envio de emails
- ◆ Processamento de imagens
- ◆ Comunicação com API's externas
  - *Notificação*
  - *Envio de SMS*
  - *E etc*
- ◆ Geração de arquivos
  - *PDF*
  - *CSV*
  - *E etc*



# O que é o Redis?

- ◆ Chave-valor
- ◆ Código aberto





# O que é o sidekiq?

- ◆ Executor de tarefas em background
- ◆ Código aberto



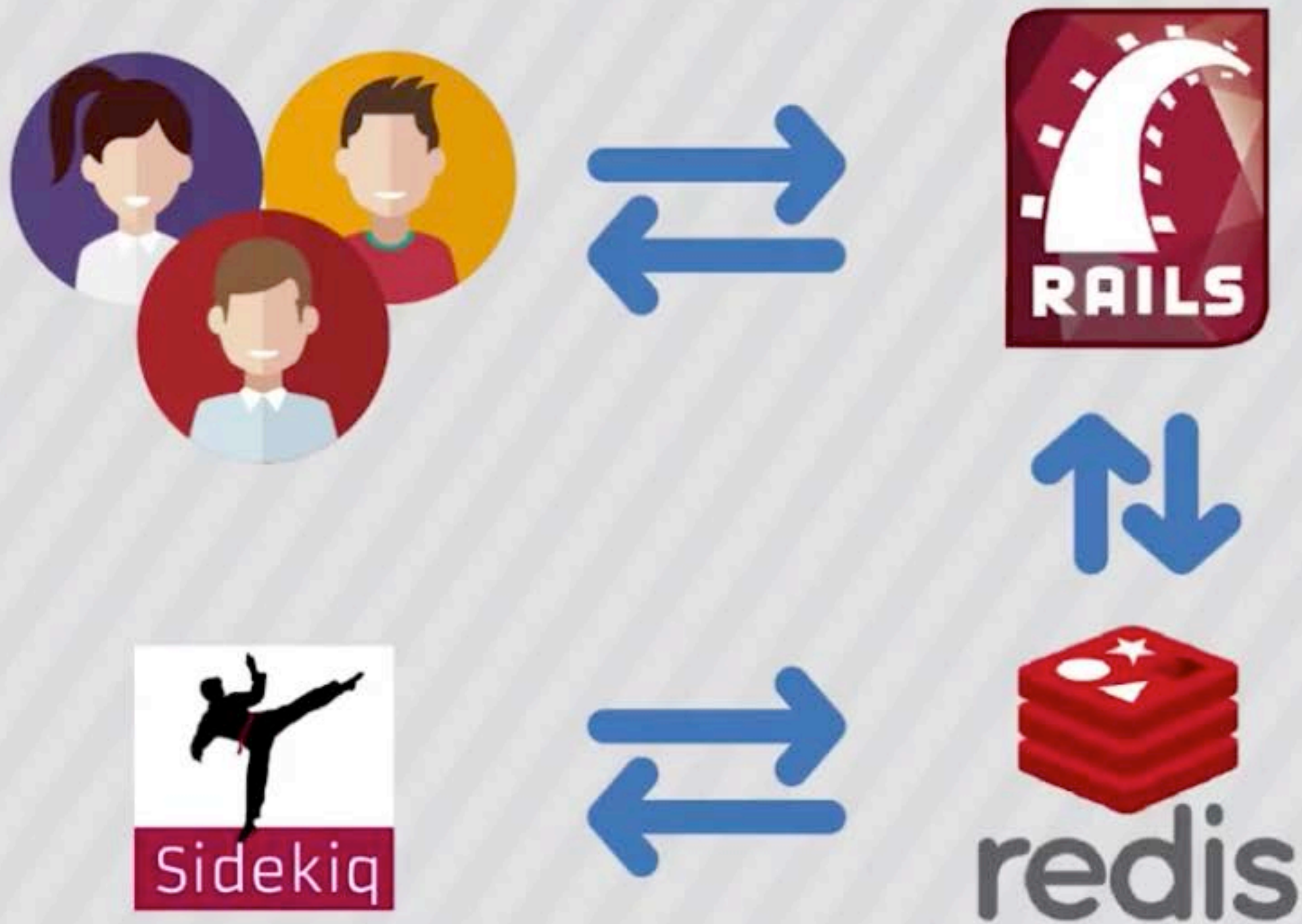
# O que são queues/filas?

- ◆ Prioridade
- ◆ Ordem de execução





# Como funciona um Job no Rails com Sidekiq + Redis





## Exemplo de um Job

```
◆ class SendSmsJob < ApplicationJob
  queue_as :default

  def perform(*guests)
    # Do something later
  end
end
End
```





# Chamando nosso Job

- ◆ Executando imediatamente
  - `GuestsCleanupJob.perform_now`
- ◆ Executando assim que possível
  - `GuestsCleanupJob.perform_later(guest)`
- ◆ Executando em uma semana
  - `GuestsCleanupJob.set(wait: 1.week).perform_later(guest)`





# Emails também podem ser enviados de forma assíncrona

- ◆ Executando imediatamente

```
Notifier.welcome(User.first).deliver_now
```

- ◆ Executando assim que possível

```
Notifier.welcome(User.first).deliver_later
```

- ◆ Executando em uma semana

```
Notifier.welcome(User.first).deliver_later(wait_until:  
1.week.from_now)
```



# Nosso Projeto

- ◆ Um Job para gerar relatórios





# O que vamos aprender?

- ◆ Como criar e usar um Job
- ◆ Como funciona o Sidekiq + Redis





# O que vamos usar?

- ◆ Rails
- ◆ Sqlite3
- ◆ Sidekiq
- ◆ Redis



# MÃOS À OBRA \o/

# ONEBITCODE.COM

Posts ♦ Tutoriais ♦ Screencasts ♦ Cursos

