DDD aplicado a um microserviço Go

Domain Driven Design

- Construir a arquitetura ao redor da parte mais importante: **O domínio**
- Usar **linguagem ubíqua** na construção e organização do código

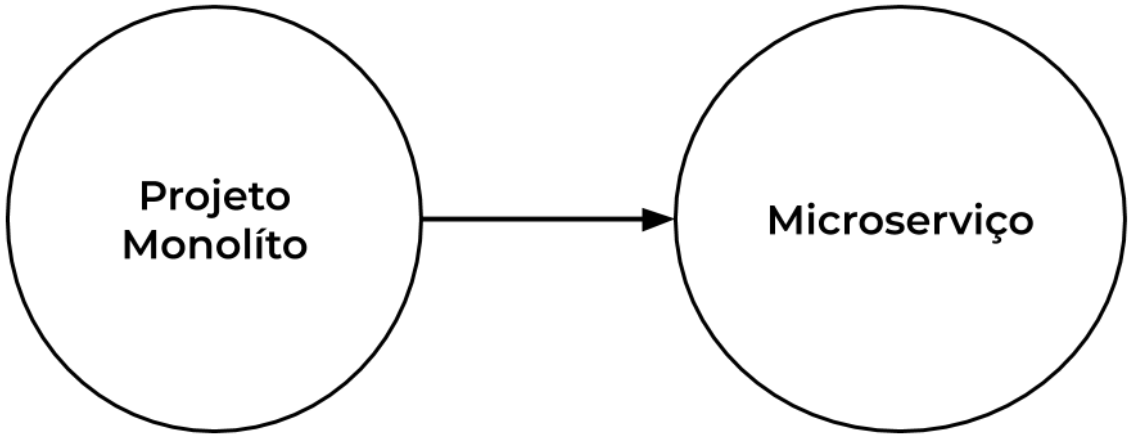
1 Motivação

2 Escolhendo DDD

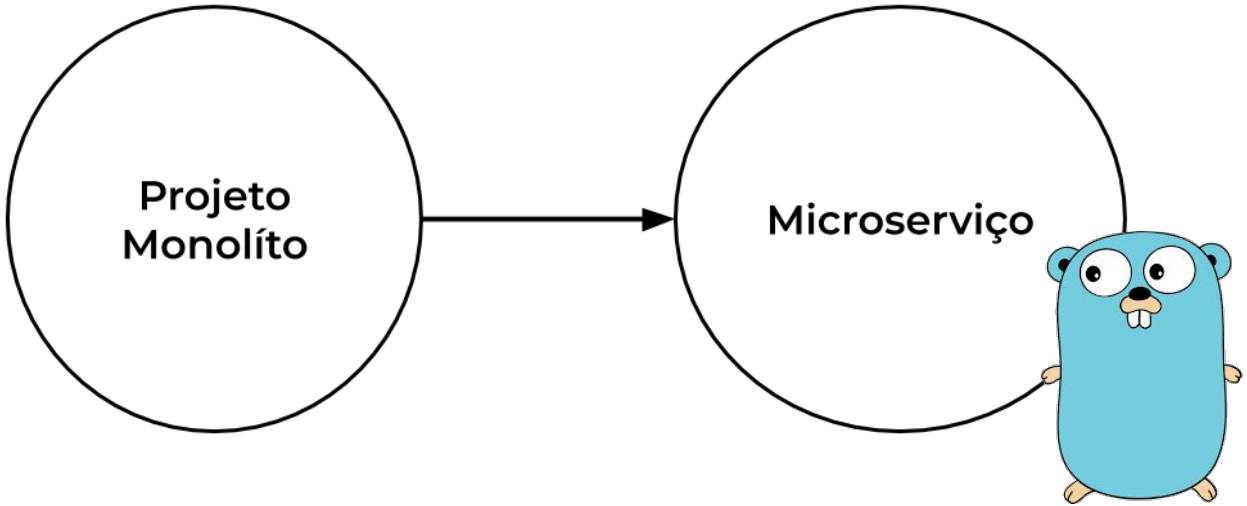
3 Detalhes da Arquitetura

1 Motivação

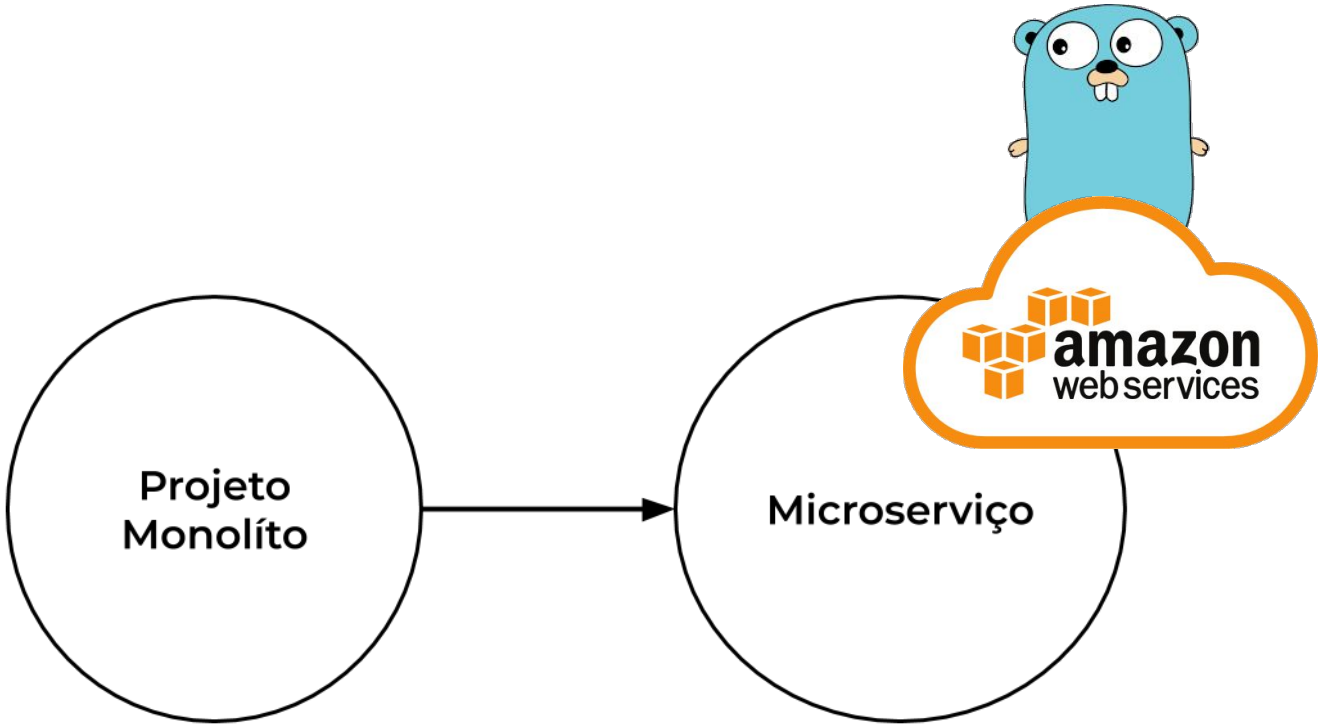
Como tudo começou



Como tudo começou



Como tudo começou



Desafio

- Queremos **manutenibilidade**
- Queremos **simplicidade** na arquitetura

2

Escolhendo DDD

Escolhendo DDD

- Garante **manutenabilidade**
- **Complexo** para projetos pequenos

Escolhendo DDD

- Garante **manutenabilidade**
- **Complexo** para projetos pequenos

Será mesmo?

Domain Driven Design

- Não *depende* de uma estrutura específica
- Logo não *precisa* ser complexo

Domain Driven Design

- Uso de **Entidades**
- Definição clara dos **Serviços**
- Definição clara da **Infra**
- Contextos delimitados por **Interfaces**

3

Arquitetura

3.1

Estrutura de Pastas

Design Flat

3 pastas no root:

- cmd/
- domain/
- infra/

Domain Types & Interfaces

- domain/
  - entities.go
  - contracts.go

Interfaces implementadas pelos serviços

- infra/
  - contracts.go

Interfaces implementadas pelos provedores

Domain Packages (serviços)

- domain/
  - entities.go
  - contracts.go
  - notifsender/
  - controlgroup/

Pacotes com nomes que fazem sentido para o negócio

Infra Packages

- infra/
  - contracts.go
  - sendgrid/
  - mysql/

3.2 Exemplos de Implementação

Exemplo de Interface:

contracts.go

```
type MessageSender interface {
  SendMessage(msg string) error
}
```

Exemplo de Interface:

contracts.go

```
type MessageSender interface {
  SendMessage(msg Message) (SendReport, error)
}

type Message struct {
  // ...
}

type SendReport struct {
  // ...
}
```

Exemplo de Entidade:

domain/entities.go:

```
type Notification struct {
  ID int
  Message string
  // ... other properties ...
}
```

Exemplo de Entidade:

domain/entities.go:

```go
type Notification struct {
  ID int
  Message string
  // ... other properties ...
}

func NewNotification(/* necessary args */) Notification {
}
```

Exemplo de Entidade:

domain/entities.go:

```go
type Notification struct {
  ID int
  Message string
  // ... other properties ...
}

func NewNotification(/* necessary args */) Notification {
}

func (n Notification) MethodsWhenNecessary() {
}
```

Exemplo de Serviço:

domain/notifsender/notifsender.go:

```
import (
  // ... Some Go built-in packages ...

  "github.com/ditointernet/example-microservice/domain"
  "github.com/ditointernet/example-microservice/infra"
)

type Service struct {
  sender infra.EmailSender // interface with a SendEmail func
}

func NewService(sender infra.EmailSender) Service {
  return Service{
    sender: sender,
  }
}

// Implements the domain.NotificationSender interface:
func (s Service) SendNotifications(notifs []domain.Notification) { /* ... */ }
```

Exemplo de Provedor:

infra/sendgrid/sendgrid.go:

```
import (
  // ... Some Go built-in packages ...

  "github.com/sendgrid/sendgrid"

  "github.com/ditointernet/example-microservice/infra"
)

type Client struct {
  sendgrid sendgridClient
}

// Used to mock sendgrid when testing:
type sendgridClient interface { /* ... */ }

func NewClient() Client {
  return Client{ sendgrid: sendgrid.New() }
}

// Implements the infra.EmailSender interface:
func (c Client) SendEmail(emails []infra.Email) { /* ... */ }
```

Exemplo de Main:

cmd/api/main.go:

```
import (
  // Import all necessary infra and domain packages
)

func main() {
  // Inject the dependencies:
  var emailSender infra.EmailSender = sendgrid.NewClient()

  var sender domain.NotificationSender = notifsender.NewService(
    emailSender,
  )

  // Start & coordinate the services
}
```

4 Referências

Algumas Referências

- Eric J. Evans
  Domain-Driven Design
- Brian Ketelsen + Ashley McNamara
  Go Best Practices
- Ben Johnson
  Standard Package Layout
- Kat Zien - GopherCon 2018
  How do You Structure Your Go Apps

dito

OBRIGADO.

Vinícius Garcia & **Fabio Rodrigues**

Engenheiro de Software
vinicius.garcia@dito.com.br

/ vingarcia

Engenheiro de Software
fabio.rodrigues@dito.com.br

/ fabiorodrigues

dito.com.br


dito.com.br/trabalheconosco
Great Place To Work.
Certificado
25/01/2019 - 25/01/2020
BRASIL
VEM SER FERA!
dito