



SÃO PAULO, 11 MAR 2020

Event Horizon

CQRS e Event Sourcing em Go



Rafael J. P. Santos

YES.Technology

gophers.slack.com

@Rafael Santos

Tópicos

1. Destrinchando o Título da Talk
2. Recomendação de Bibliografia
3. CQRS - Vantagens
4. Event Sourcing - Vantagens
5. Event Horizon - Conceitos
6. Event Horizon - Código de Exemplo
7. Desafios
8. Conclusão

SEÇÃO 1 DE 8

Destrinchando o Título da Talk

- Biblioteca escrita em Go
- Implementa solução de Event Sourcing com CQRS



- *Command-Query Responsibility Segregation*
(Separação de Responsabilidade entre Comandos e Queries)
- Separação explícita entre modelagem de escrita, e modelagem de leitura

- Sequência de Eventos define o estado atual dos dados do sistema
- Eventos são a fonte única da verdade (single source of truth)

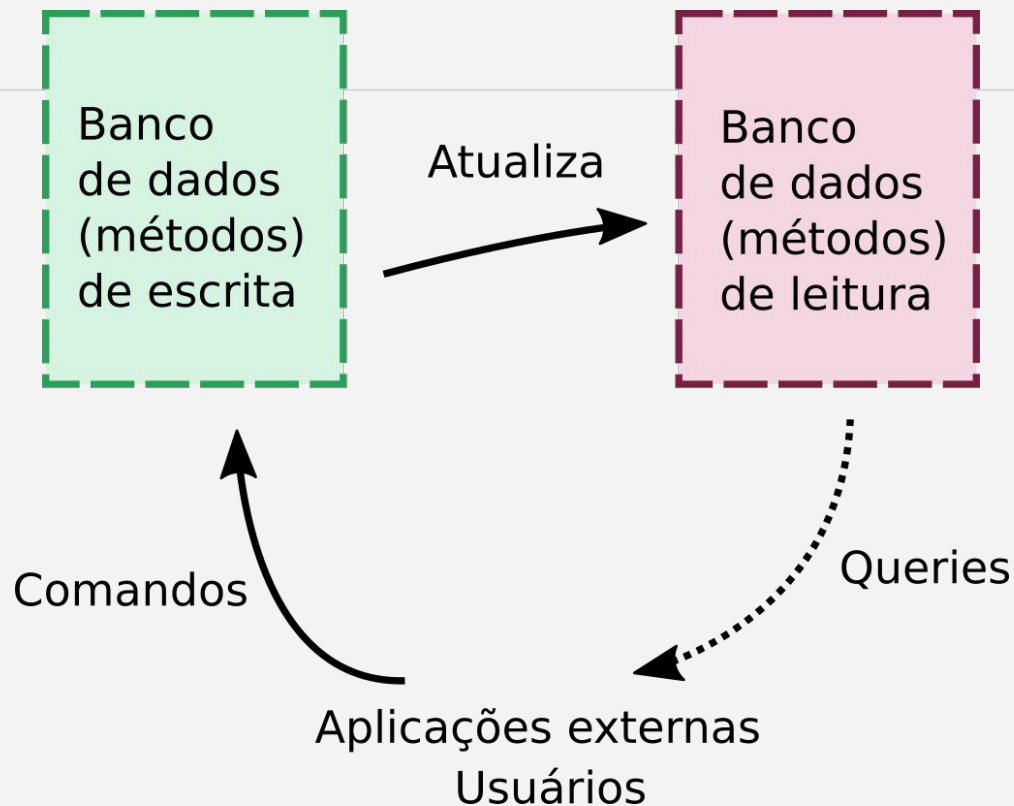
SEÇÃO 2 DE 8

Recomendação de Bibliografia

- Martin Fowler
 - <https://martinfowler.com/bliki/CQRS.html>
- The CQRS FAQ
 - <http://cqrs.nu>
- Chris Richardson
 - Microservices Patterns: With Examples in ~~Java~~

CQRS

Vantagens



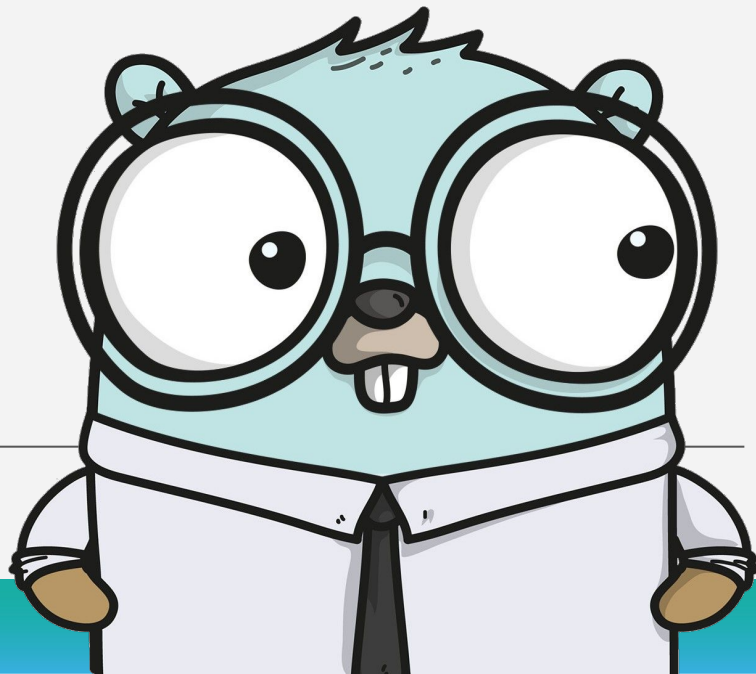
- Separação explícita possibilita queries menos complexas
- Permite múltiplas projeções/views para um mesmo dado (ou evento)

- Queries menos complexas exigem menor poder de processamento
- Réplicas de leitura são mais simples de escalar do que servidores de escrita

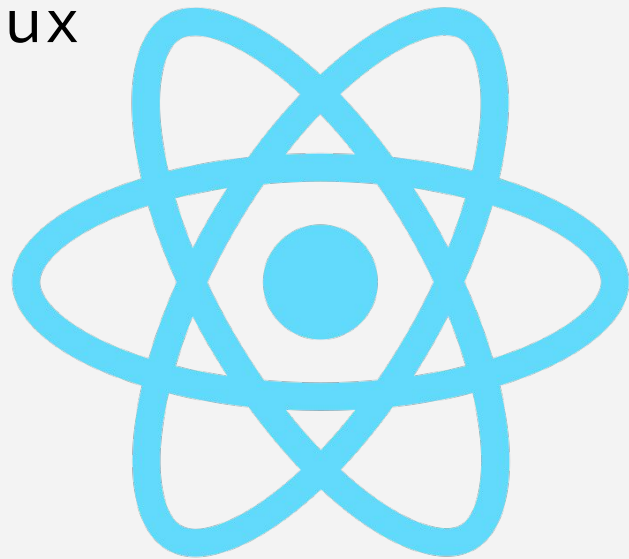
SEÇÃO 4 DE 8

Event Sourcing Vantagens

- **Todas** as alterações de dados ficam armazenadas na forma de eventos
- Log confiável

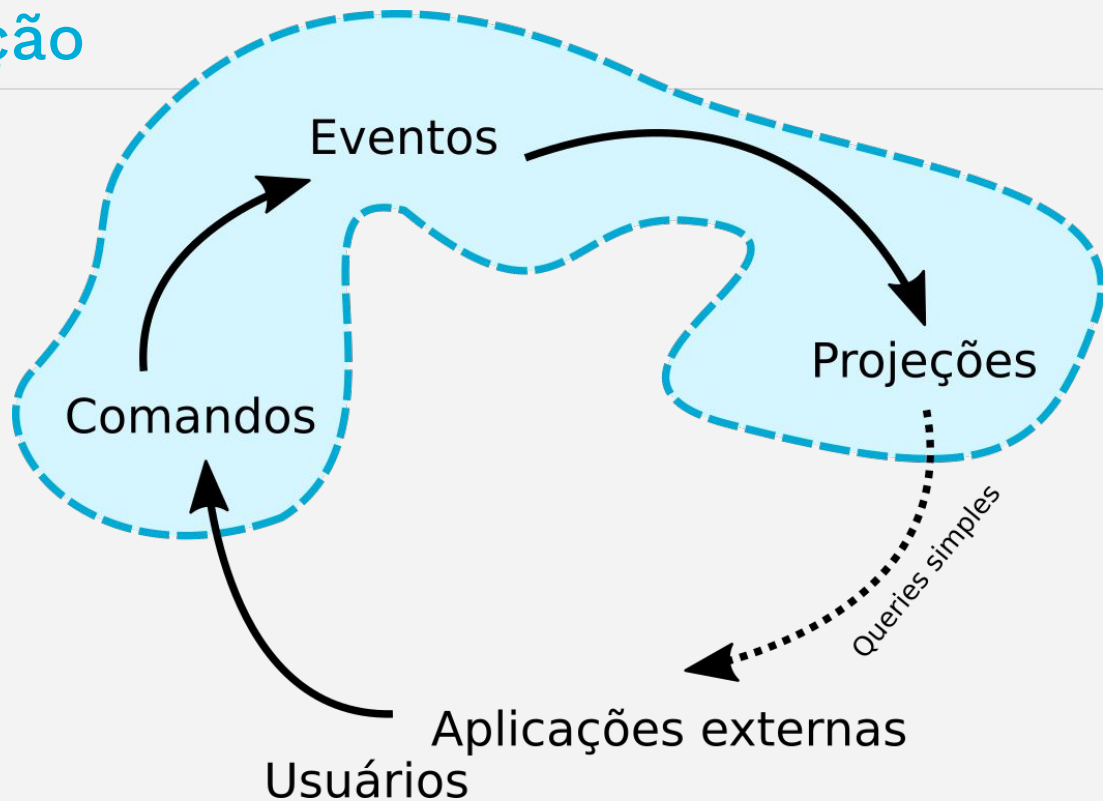


- Máquina do tempo
 - Analogia com React e Redux
- Reprodução de bugs



Event Horizon Conceitos

Fluxo da informação



- Escolheu Event Sourcing + CQRS
 - Mas é possível CQRS sem Event Sourcing
 - E é possível Event Sourcing sem CQRS
- Event Sourcing + CQRS =
 - Auditabilidade/rastreabilidade + Escalabilidade

- Comandos
- Eventos
- Consumidores de Eventos (Event Handlers)

- São solicitações de mudança de estado
- Caso sejam aceitos, emitem eventos
- Possuem acesso limitado ao estado atual

- São fatos que já ocorreram no sistema
- Imutáveis
- Só podem ser corrigidos por meio de estornos (ações reversas)

- Recebem automaticamente novos eventos
- Atualizam projeções/views
- Podem disparar novos comandos (sagas)

- **Comandos**
- Command Bus
- Command Handlers
- **Eventos**
- Event Store
- Event Bus
- **Consumidores de Eventos (Event Handlers)**

Event Horizon

Código de

Exemplo

“Veja a pasta exemplos para começar”



`https://github.com/looplab/eventhorizon`

Inspired by the following libraries/examples:

- <https://github.com/edumentab/cqrs-starter-kit>
- <https://github.com/pjvds/go-cqrs>
- <http://www.codeproject.com/Articles/555855/Introduction-to-CQRS>
- <https://github.com/qandidate-labs/broadway>

Suggestions are welcome!

Usage



See the example folder for a few examples to get you started.

Seção 6 de 8 - Event Horizon - Código de Exemplo

Lista de convidados (guestlist)

- Exemplo
 - `/examples/guestlist`
- Arquivo inicial
 - `mongodb/mongodb_test.go` ? ? ?



Seção 6 de 8 - Event Horizon - Código de Exemplo

Exemplo “guestlist”



loolab / eventhorizon

Watch 40 Unstar 774 Fork 112

Code Issues 47 Pull requests 2 Actions Wiki Security Insights

Branch: master eventhorizon / examples /

Create new file Upload files Find file History

semenovDL Wait for event to happen ✓ Latest commit d4686e4 on Jan 11, 2019

..

guestlist Migrate UUID to google/uuid 14 months ago

todomvc Wait for event to happen 14 months ago

Seção 6 de 8 - Event Horizon - Código de Exemplo

Exemplo “guestlist”



looplabs / eventhorizon

Watch 40 Unstar 774 Fork 112

Code Issues 47 Pull requests 2 Actions Wiki Security Insights

Branch: master eventhorizon / examples / guestlist /

Create new file Upload files Find file History

semenovDL Migrate UUID to google/uuid Latest commit 1a1caf7 on Jan 11, 2019

..

domain	Migrate UUID to google/uuid	14 months ago
memory	Migrate UUID to google/uuid	14 months ago
mongodb	Migrate UUID to google/uuid	14 months ago

Seção 6 de 8 - Event Horizon - Código de Exemplo

Exemplo “guestlist” - mongodb_test.go



looplab / eventhorizon

Watch 40 Unstar 774 Fork 112

Code Issues 47 Pull requests 2 Actions Wiki Security Insights

Branch: master eventhorizon / examples / guestlist / mongodb /

Create new file Upload files Find file History

semenovDL Migrate UUID to google/uuid Latest commit 1a1caf7 on Jan 11, 2019

..

mongodb.go	Add AUTHORS file	2 years ago
mongodb_test.go	Migrate UUID to google/uuid	14 months ago

Seção 6 de 8 - Event Horizon - Código de Exemplo

- Comandos:
 - CreateInvite, AcceptInvite, DeclineInvite, ConfirmInvite, DenyInvite
 - Eventos:
 - InviteCreated, InviteAccepted, InviteDeclined, InviteConfirmed, InviteDenied
 - Projeções (views pré-calculadas):
 - InvitationProjector, GuestListProjector
 - Saga:
 - ResponseSaga (limita quantidade máxima de convidados)
-

```
25     "github.com/google/uuid"  
26     eh "github.com/looplab/eventhorizon"  
27     "github.com/looplab/eventhorizon/commandhandler/bus"  
28     eventbus "github.com/looplab/eventhorizon/eventbus/local"  
29     eventstore "github.com/looplab/eventhorizon/eventstore/mongodb"  
30     repo "github.com/looplab/eventhorizon/repo/mongodb"  
31     "github.com/looplab/eventhorizon/repo/version"  
32  
33     "github.com/looplab/eventhorizon/examples/guestlist/domain"
```



Arquivo mongodb_test.go

```
45     // Create the event store.
46     eventStore, err := eventstore.NewEventStore(url, "demo")
47     if err != nil {
48         log.Fatalf("could not create event store: %s", err)
49     }
50
51     // Create the event bus that distributes events.
52     eventBus := eventbus.NewEventBus(nil)
53     go func() {
54         for e := range eventBus.Errors() {
55             log.Printf("eventbus: %s", e.Error())
56         }
57     }()
58
59     // Create the command bus.
60     commandBus := bus.NewCommandHandler()
```

Inicialização
Event Store
com
MongoDB

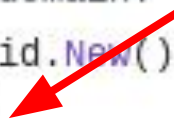
Inicialização
Event Bus
In-Memory

Inicialização
Command
Bus

Seção 6 de 8 - Event Horizon - Código de Exemplo

```
76      // Setup the domain.  
77      eventID := uuid.New()  
78      domain.Setup(  
79          eventStore,  
80          eventBus,  
81          commandBus,  
82          invitationVersionRepo, guestListRepo,  
83          eventID,  
84      )
```

Setup do domínio: vinculação entre Comandos, Command Handlers, Eventos e Event Handlers



Envio de Comando

```
// Issue some invitations and responses. Error checking omitted here.
if err := commandBus.HandleCommand(ctx, &domain.CreateInvite{ID: athenaID, Name: "Athena", Age: 42}); err != nil {
    log.Println("error:", err)
}
if err := commandBus.HandleCommand(ctx, &domain.CreateInvite{ID: hadesID, Name: "Hades"}); err != nil {
    log.Println("error:", err)
}
if err := commandBus.HandleCommand(ctx, &domain.CreateInvite{ID: zeusID, Name: "Zeus"}); err != nil {
    log.Println("error:", err)
}
if err := commandBus.HandleCommand(ctx, &domain.CreateInvite{ID: poseidonID, Name: "Poseidon"}); err != nil {
    log.Println("error:", err)
}
time.Sleep(100 * time.Millisecond)
```

Aguarda consistência eventual

```
// Read all invites.
invitationStrs := []string{}
invitations, err := invitationRepo.FindAll(ctx)
if err != nil {
    log.Println("error:", err)
}
for _, i := range invitations {
    if i, ok := i.(*domain.Invitation); ok {
        invitationStrs = append(invitationStrs, fmt.Sprintf("%s - %s", i.Name, i.Status))
    }
}
```

Query simples:
Fazer consultas
fica muito fácil!

Exemplo “guestlist” - setup.go



Branch: **master** **eventhorizon** / **examples** / **guestlist** / **domain** /

Create new file Upload files Find file History

semenovDL Migrate UUID to google/uuid ✗ Latest commit 1a1caf7 on Jan 11, 2019

..

aggregate.go	Migrate UUID to google/uuid	14 months ago
commands.go	Migrate UUID to google/uuid	14 months ago
events.go	Add AUTHORS file	2 years ago
logger.go	Remove Publisher, let EventBus handle observers	2 years ago
projectors.go	Migrate UUID to google/uuid	14 months ago
saga.go	Migrate UUID to google/uuid	14 months ago
setup.go	Migrate UUID to google/uuid	14 months ago

Seção 6 de 8 - Event Horizon - Código de Exemplo

```
// Create the aggregate command handler and register the commands it handles.
invitationHandler, err := aggregate.NewCommandHandler(InvitationAggregateType, aggregateStore)
if err != nil {
    log.Fatalf("could not create command handler: %s", err)
}
commandHandler := eh.UseCommandHandlerMiddleware(invitationHandler, LoggingMiddleware)
commandBus.SetHandler(commandHandler, CreateInviteCommand)
commandBus.SetHandler(commandHandler, AcceptInviteCommand)
commandBus.SetHandler(commandHandler, DeclineInviteCommand)
commandBus.SetHandler(commandHandler, ConfirmInviteCommand)
commandBus.SetHandler(commandHandler, DenyInviteCommand)
```


Vinculação entre
Comandos e
Command Handlers

```
// Create and register a read model for individual invitations.
invitationProjector := projector.NewEventHandler(
    NewInvitationProjector(), invitationRepo)
invitationProjector.SetEntityFactory(func() eh.Entity { return &Invitation{} })
eventBus.AddHandler(eh.MatchAnyEventOf(
    InviteCreatedEvent,
    InviteAcceptedEvent,
    InviteDeclinedEvent,
    InviteConfirmedEvent,
    InviteDeniedEvent,
), invitationProjector)
```

← Inicialização de
Event Handler
(Visualização de
um convite)

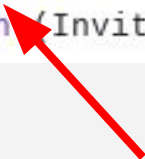
← Vinculação entre
Eventos e Event
Handler

```
// Create and register a read model for a guest list.  
guestListProjector := NewGuestListProjector(guestListRepo, eventID)  
eventBus.AddHandler(eh.MatchAnyEventOf(  
    InviteAcceptedEvent,  
    InviteDeclinedEvent,  
    InviteConfirmedEvent,  
    InviteDeniedEvent,  
) , guestListProjector)
```



Inicialização de um
segundo Event
Handler (Lista de
convidados)


```
// Setup the saga that responds to the accepted guests and limits the total
// amount of guests, responding with a confirmation or denial.
responseSaga := saga.NewEventHandler(NewResponseSaga(2), commandBus)
eventBus.AddHandler(eh.MatchEvent(InviteAcceptedEvent), responseSaga)
```



Inicialização do último Event Handler (Saga que restringe quantidade máxima de convidados)

Exemplo “guestlist” - saga.go



Branch: master ▾

[eventhorizon](#) / [examples](#) / [guestlist](#) / [domain](#) /

Create new file

Upload files

Find file

History



semenovDL Migrate UUID to google/uuid

✗ Latest commit 1a1caf7 on Jan 11, 2019

..


aggregate.go	Migrate UUID to google/uuid	14 months ago
commands.go	Migrate UUID to google/uuid	14 months ago
events.go	Add AUTHORS file	2 years ago
logger.go	Remove Publisher, let EventBus handle observers	2 years ago
projectors.go	Migrate UUID to google/uuid	14 months ago
saga.go	Migrate UUID to google/uuid	14 months ago
setup.go	Migrate UUID to google/uuid	14 months ago

Seção 6 de 8 - Event Horizon - Código de Exemplo

```
// RunSaga implements the Run saga method of the Saga interface.  
func (s *ResponseSaga) RunSaga(ctx context.Context, event eh.Event) []eh.Command {  
    switch event.EventType() {  
    case InviteAcceptedEvent:
```

```
        // Deny the invite if the guest list is full.  
        if len(s.acceptedGuests) >= s.guestLimit {  
            return []eh.Command{  
                &DenyInvite{ID: event.AggregateID()},  
            }  
        }  
    }
```

```
    // Confirm the invite when there is space left.  
    s.acceptedGuestsMu.Lock()  
    s.acceptedGuests[event.AggregateID()] = true  
    s.acceptedGuestsMu.Unlock()  
  
    return []eh.Command{  
        &ConfirmInvite{ID: event.AggregateID()},  
    }  
}
```



Event Handler do tipo Saga
recebe Evento, e retorna lista de
Comandos

- Escrita
 - `commandBus.HandleCommand(ctx, &domain.CreateInvite{ID: athenaID, Name: "Athena", Age: 42})`
- Leitura
 - `guestListRepo.Find(ctx, eventID)`

SEÇÃO 7 DE 8

Desafios

- Novos conceitos
- Novas ferramentas

- Exemplo de requisito de negócio
 - CPF único
 - E-mail único
- Solução
 - Saga

- Eventos imutáveis
- Código retrocompatível
 - Código **sempre** deve suportar **todos** os eventos já existentes no sistema
- Soluções
 - Snapshots
 - Formatos de serialização versionáveis
 - Protocol Buffers

- Causas
 - Erro ao publicar eventos
 - Bugs ao processar eventos
- Soluções
 - Pattern Outbox
 - Republicação de eventos

- Frontend (ou API, ou CLI, etc...) devem ser adaptados para uma boa experiência do usuário
- Abstrair, ou não abstrair a consistência eventual?

- Preocupação com encapsulamento
- Discussão:
 - <https://dev.to/olibutzki/why-event-sourcing-is-a-microservice-anti-pattern-3mcj>

- Eventos imutáveis
 - Direito a exclusão de dados desnecessários
- Possível solução
 - Eventos encriptados com chaves individuais para cada agregado, armazenadas separadamente

SEÇÃO 8 DE 8

Conclusão

- Prós
 - Auditabilidade
 - Escalabilidade
 - Troubleshooting
- Contras
 - Necessidade de retrocompatibilidade
 - Curva de aprendizagem íngreme
 - Complexidade típica de sistemas distribuídos

- Biblioteca bem escrita
- Fornece funcionalidades essenciais para Event Sourcing com CQRS
- Funcionalidades adicionais podem ser facilmente implementadas



“

Estamos contratando.
jobs@yes.technology

Rafael Santos
YES.Technology

”