



dependency injection

Gilvan Ritter

github.com/gritt

Consultant Developer - ThoughtWorks



dependency injection

- é um estilo de escrever código para que as dependências de um objeto sejam recebidas na sua inicialização
 - diminui acoplamento na arquitetura
 - flexibiliza / manutenção / evolução
 - simplifica testes



composition over inheritance

- composição de tipos
- composição de interfaces



composition over inheritance

```
type ReadWriter interface {
```

```
    Reader
```

```
    Writer
```

```
}
```



dependency injection em go

```
// transaction.go:
```

```
type CreateUseCase struct {
```

```
    db.Repository
```

```
}
```



dependency injection em go

```
// transaction.go:
```

```
func NewCreateUseCase(r db.Repository) *CreateUseCase {
```

```
    return &CreateUseCase{r}
```

```
}
```



dependency injection em go

```
// main.go:
```

```
cfg := db.NewConfig()
```

```
repo := db.NewRepository(cfg)
```

```
useCase := core.NewCreateUseCase(repo)
```

```
...
```

```
...
```



wire

- gerador de código de inicialização
 - gera código em Go
 - exatamente como você faria manualmente
- não opera em tempo de execução

<https://github.com/google/wire>

```
go get github.com/google/wire/cmd/wire
```




wire

```
// +build wireinject
```

```
panic(wire.Build(
```

```
    db.NewConfig,
```

```
    db.NewRepository,
```

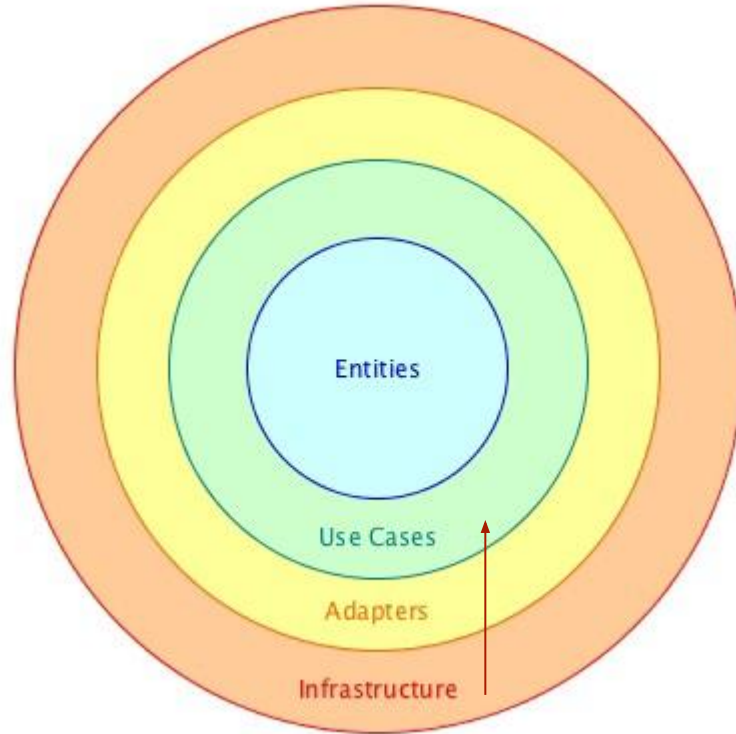
```
    core.NewCreateUseCase,
```

```
))
```

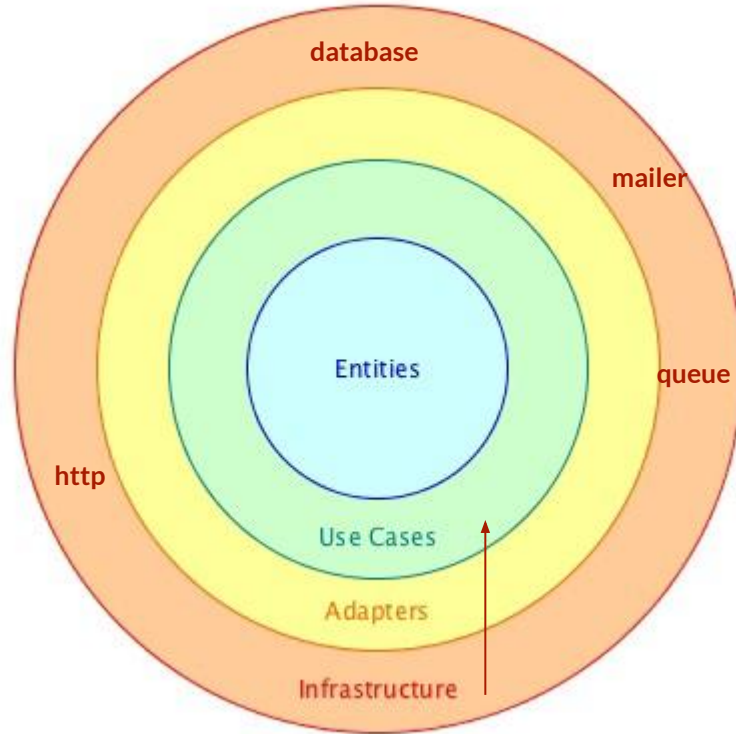


+bonus: clean architecture

clean architecture



clean architecture





clean architecture em go

```
type TransactionSaver interface {  
    Save(Transaction) error  
}
```

```
type CreateUseCase struct {  
    TransactionSaver  
}
```



dependency injection em go

```
// transaction.go:
```

```
func NewCreateUseCase(ts TransactionSaver) *CreateUseCase {
```

```
    return &CreateUseCase{ts}
```

```
}
```



referências

- <https://github.com/google/wire>
- <https://blog.golang.org/wire>
- <https://blog.drewolson.org/go-dependency-injection-with-wire>
- <https://medium.com/full-stack-tips/dependency-injection-in-go-99b09e2cc480>
- <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>
-