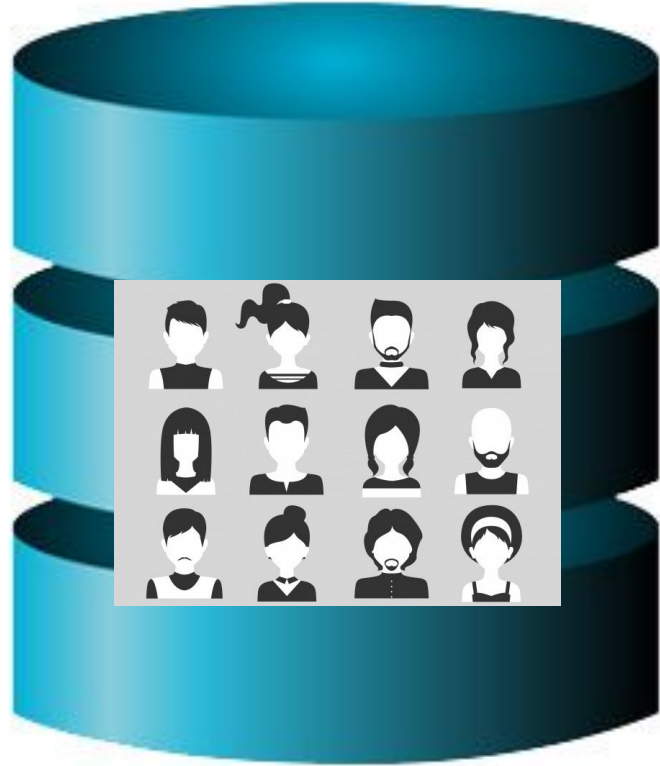




# API de pessoas

Pedro Germano Cervi

# Introdução



# CDP - Customer Data Platform

- Gartner: 2/3 dos investimentos em tecnologia em plataformas de marketing em 2018 foram em iniciativas de retenção e aumento da base de clientes
- Forbes: Em 2019, 78% das figuras chave entrevistadas informaram ter ou estar desenvolvendo uma CDP
- Um dos maiores desafios é a unificação de fontes de dados para criar com sucesso perfis e segmentações



## SUMMARY

Customer data platforms promise to support customer growth and retention goals by unifying customer profile data but so do many other technologies. Marketing leaders can navigate this emerging and confusing category by analyzing these core features.

## Forbesinsights

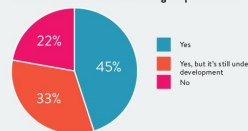
### RISE OF CUSTOMER DATA PLATFORMS (CDPs)

Customer relationship management and business intelligence solutions have provided intelligence about customer trends, but times have changed. Customer journeys have become impossibly complex, and the resulting insights are increasingly informed by disconnected data sources, and by explosively growing sources such as device data (IoT). **Enter the customer data platform, or CDP.** These are systems that provide business teams unified, real-time, self-service access to customer data that is typically spread across systems in different lines of business, such as sales, support or product.

Traditional CRM systems are designed to enable management and analysis of a particular customer channel, but CDPs are intended to be more comprehensive. These platforms bring customer data from a variety of sources into a single environment, including call center interactions, connected device data, product usage data, sales data, mobile apps, customer life cycle trends, websites, social media and email. CDPs also pave the way to greater personalization and measurement.

"Marketers need to drive and integrate all systems together to get a holistic view of their customers and better serve them," says Kevin Schofield, lead digital marketing consultant at customer experience

Figure 16. Does Your Company Have a Customer Data Platform Dedicated to Marketing Requirements?





# Conteúdos

- Objetivos
- O que, como
- Framework / Stack
- Diagrama de componentes
- Diagrama de classes
- Builder Pattern
- Composite Pattern
- Adapter Pattern
- Qualidade
- Uso
- TODO
- Fontes

# Objetivos

- Oferecer um serviço centralizado de controle de dados de pessoas para sistemas de múltiplas finalidades
  - Centralizar dados de pessoas
    - Mais controle de validação de dados
    - Fonte única de verdade
  - Vantagens da arquitetura de serviços
    - Otimização de recursos
    - Detecção de bugs
    - Qualidade do código
  - Open source



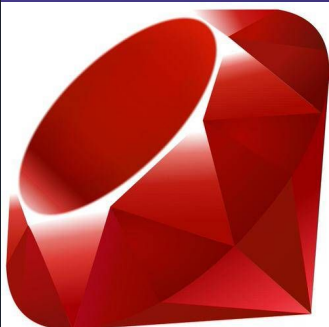
open source



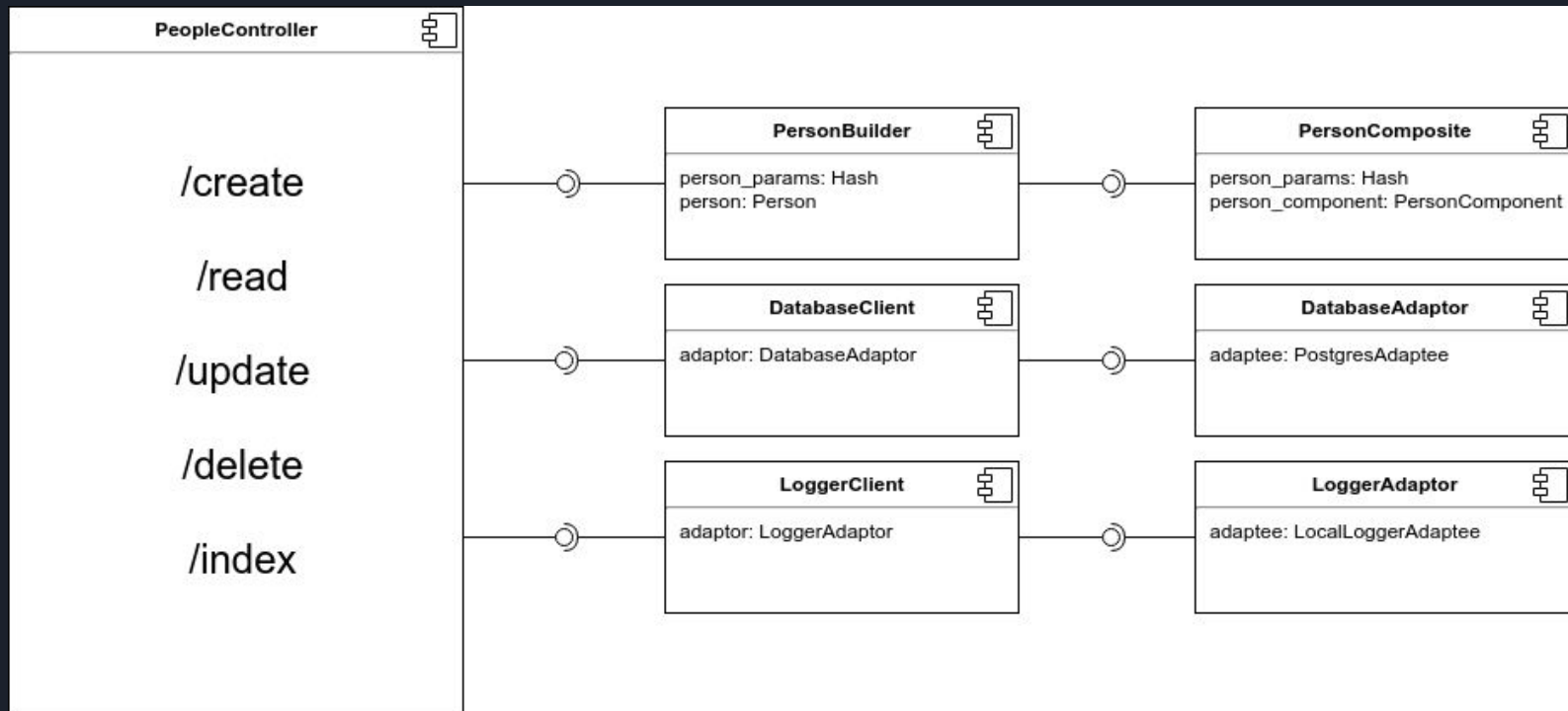
## O que, como

- Service Oriented Architecture
- API RESTful
  - Http
  - Json
- CRUD de entidades pessoas
  - Camadas de validação de dados
  - Interação com serviços de log, backup, db e estatísticas

# Framework / Stack



# Visão geral - Diagrama de Componentes



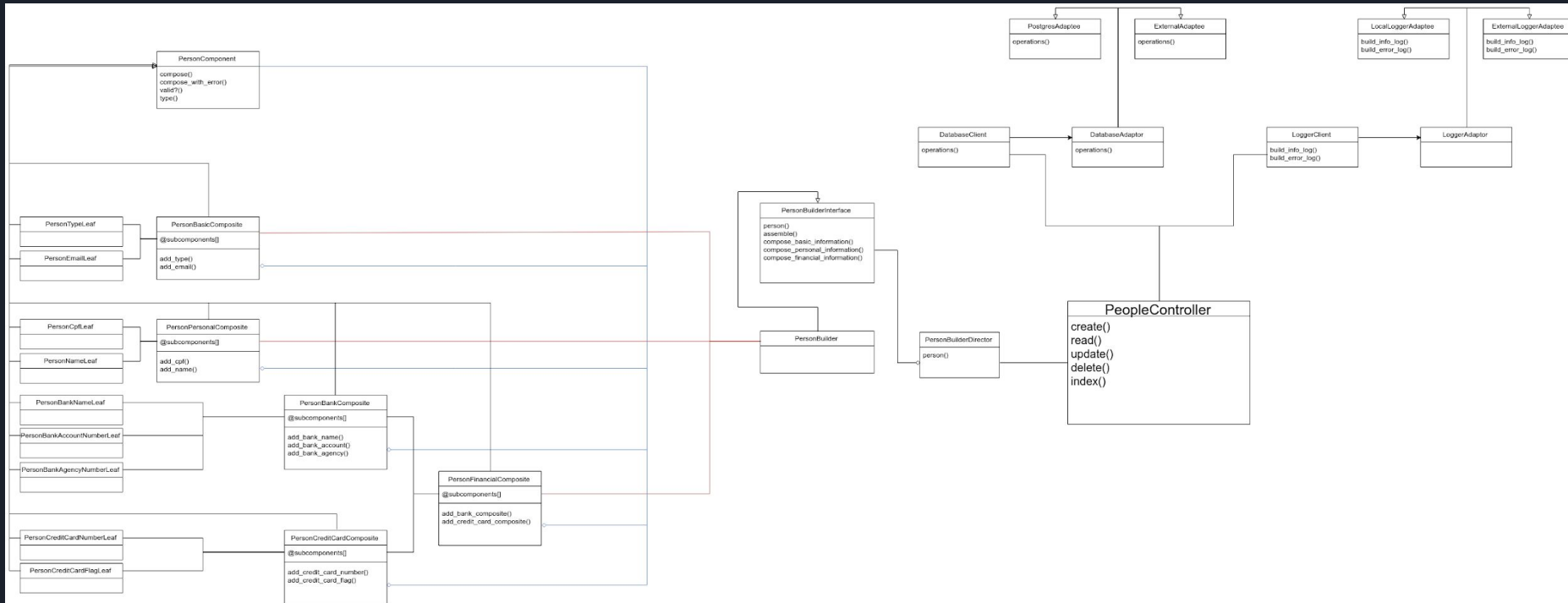




# Dependências

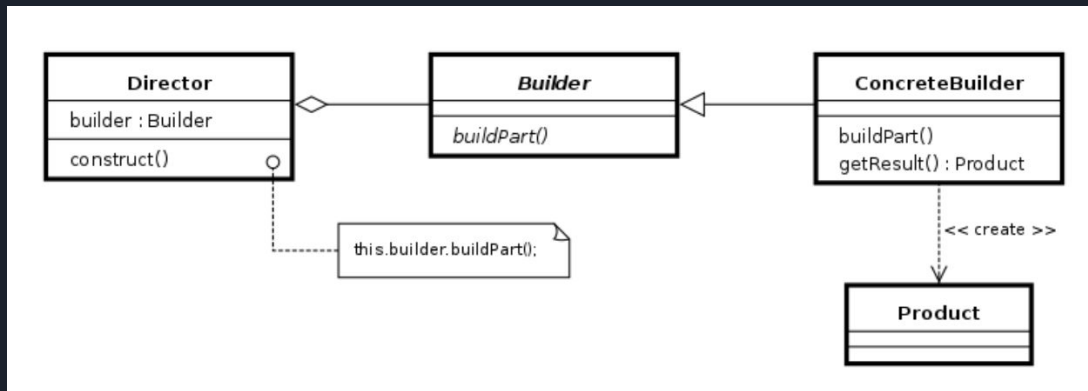
- Gemfile (componentes do framework)
- Banco de dados (migrações de tabelas)

# Visão geral - Diagrama de Classes

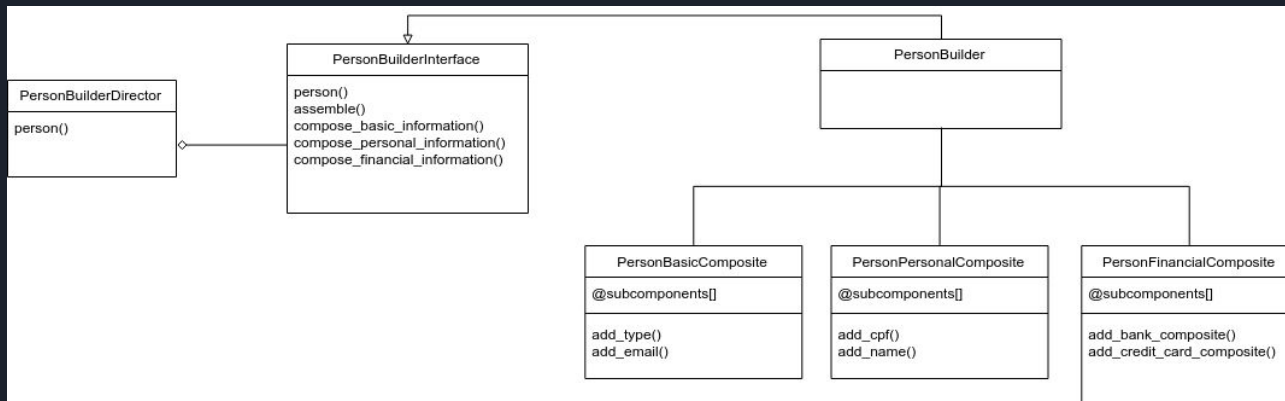
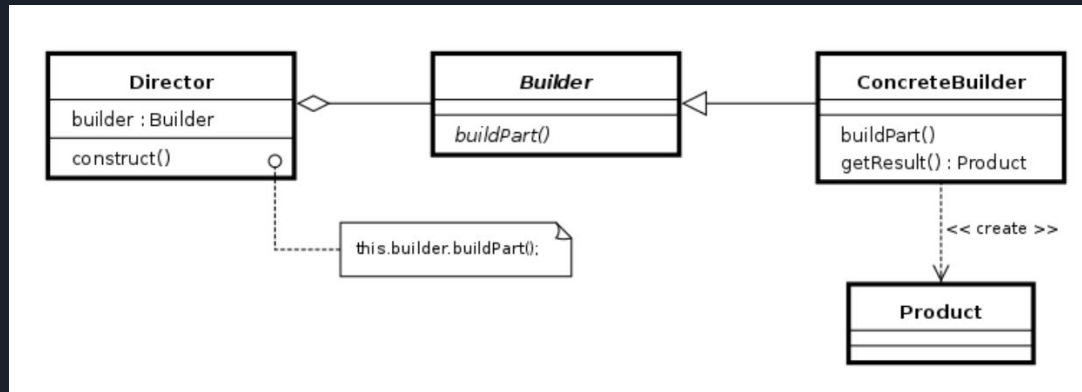


# Builder Pattern

“Separar a construção de um objeto complexo de sua representação para que o mesmo processo de construção seja capaz de criar diferentes representações.” (Design Patterns)



# Builder Pattern





# Builder - Código

```
# PeopleController
def create
  Builders::PersonBuilderDirector.new(params).person
end

# Builders::PersonBuilderDirector
def person
  case person_type
  when 'user'
    ::PersonBuilder.build do |builder|
      builder.compose_basic_information(@params)
      builder.compose_personal_information(@params)
    end
  end
end

# PersonBuilder <:: PersonBuilderInterface
def person
  assemble

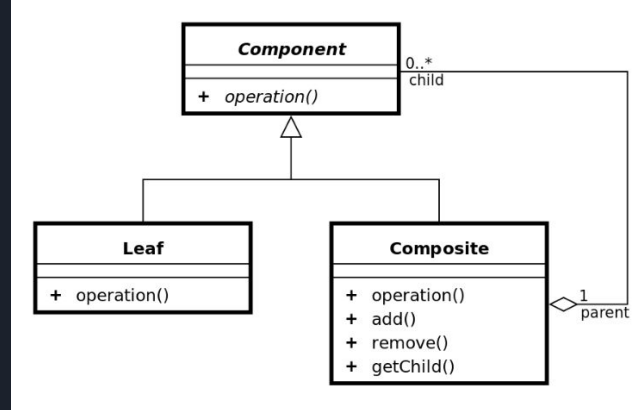
  @person
end

def compose_basic_information(params)
  Components::Composites::PersonBasic.new(@person, params)
end

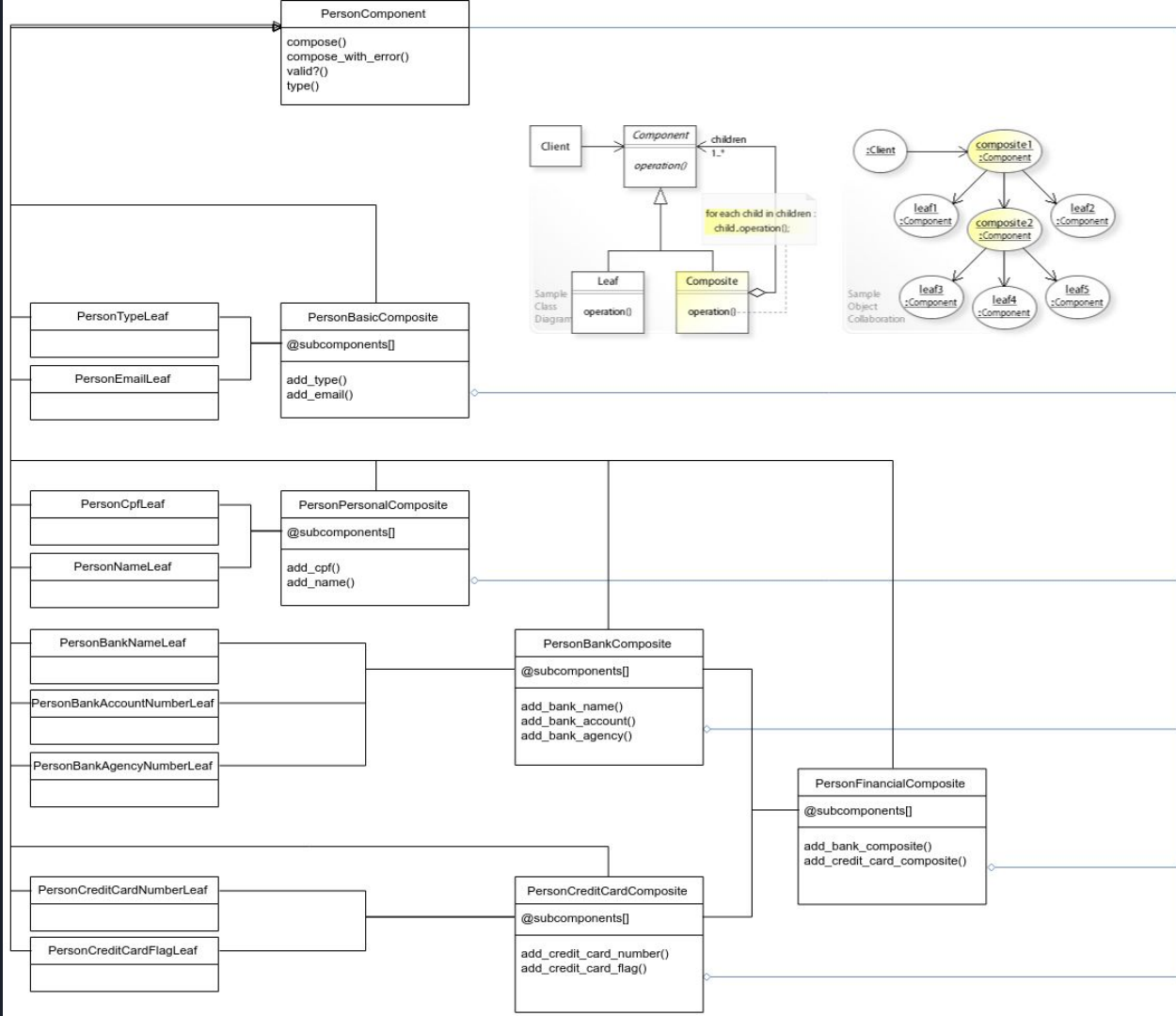
# PersonBuilderInterface
def person
  raise NotImplementedError
end
```

# Composite Pattern

“Compor objetos em estruturas com formato de árvore para representar hierarquias. O composite permite os clientes tratarem objetos singulares e objetos compostos de maneira uniforme.” (Design Patterns)



# Composite Pattern



# Composite - Código

```
# PersonBuilder <:: PersonBuilderInterface
def compose_basic_information(params)
  Components::Composites::PersonBasic.new(@person, params)
end

# Components::Composites::PersonBasic < Components::PersonComponent
def subcomponents
  @subcomponents << Components::Leafs::PersonType.new(@person, @params[:type])
  @subcomponents << Components::Leafs::PersonEmail.new(@person, @params[:email])

  @subcomponents
end

# Components::Leafs::PersonType < Components::PersonComponent
VALID_TYPES = %w[user lead].freeze

def valid?
  return true if @value.blank?

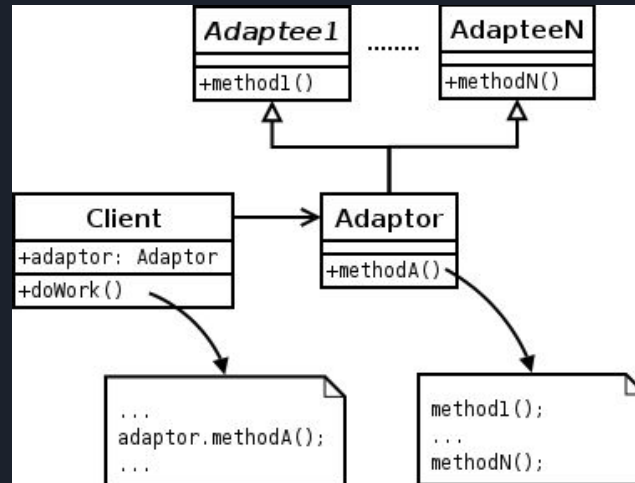
  VALID_TYPES.include?(@value)
end

# Components::PersonComponent
def compose(subcomponents: nil)
  if subcomponents.present?
    @person.components + subcomponents
  else
    @person.components << self
  end
end
```

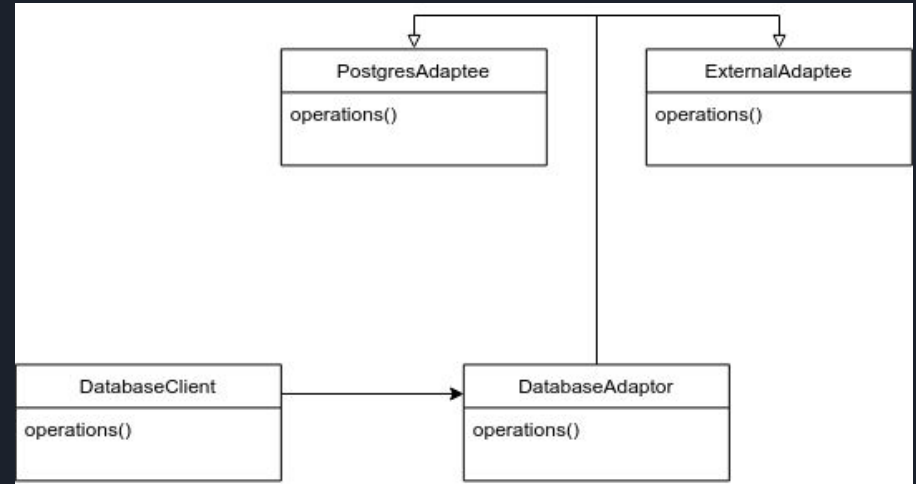
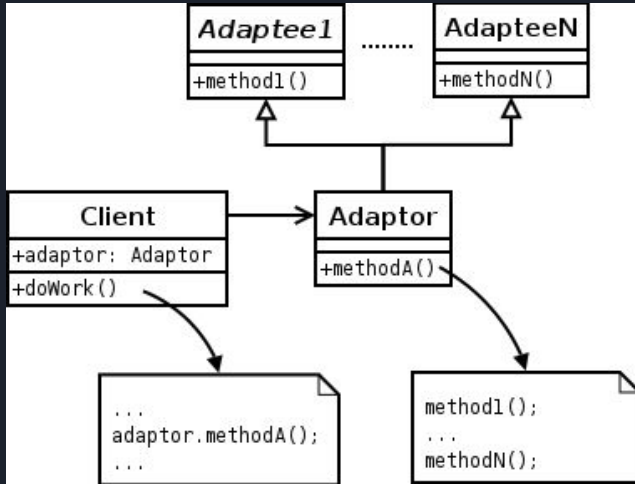



# Adapter Pattern

“Converter a interface de uma classe em outra interface que o cliente espera. O adapter permite que classes possam trabalhar juntas em cenários que não poderiam graças a interfaces incompatíveis.” (Design Patterns)



# Adapter Pattern





# Adapter - Código

```
# PeopleController
def database_client
  @database_client ||= Clients::DatabaseClient.new
end

# Clients::DatabaseClient
def initialize
  @database_adapter = Adapters::DatabaseAdapter.new
end

# Adapters::DatabaseAdapter
def initialize
  @adaptee = Adapters::PostgresAdaptee.new
end
```



# Qualidade

- Testes
  - Unitários
    - [person type spec.rb](#)
  - Integração
    - [people controller spec.rb](#)
  - Aceitação
    - Curl e Postman
- Métricas
  - Cobertura de testes



Uso

<https://github.com/pedroCervi/OTES12>



# TODO

- Tipagem de pessoa customizável
- Camadas de validação de dados customizáveis
- Melhorar mensagens



# Fontes

Pemberton, Chris. **Gartner CMO Spend Survey**. Disponível em:  
<https://www.gartner.com/en/marketing/insights/articles/2017-2018-gartner-cmo-spend-survey>. Acesso em out. de 2019.

Foo Kune, Lizzy e Meyers, James. **A Marketer's Guide to What is - and isn't - a Customer Data Platform**. Disponível em:  
<https://emtemp.gcom.cloud/ngw/globalassets/en/marketing/documents/trends/marketers-guide-customer-data-platform-research-2018.pdf>. Acesso em out. de 2019.

FORBES. **Data Versus Goliath**: Customer Data Strategies to Disrupt the Disruptors. Disponível em:  
<https://www.forbes.com/forbes-insights/our-work/data-versus-goliath/>. Acesso em out. de 2019.

Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John. **Design Patterns**: elements of reusable object-oriented software. Editora: Addison-Wesley, 1994.