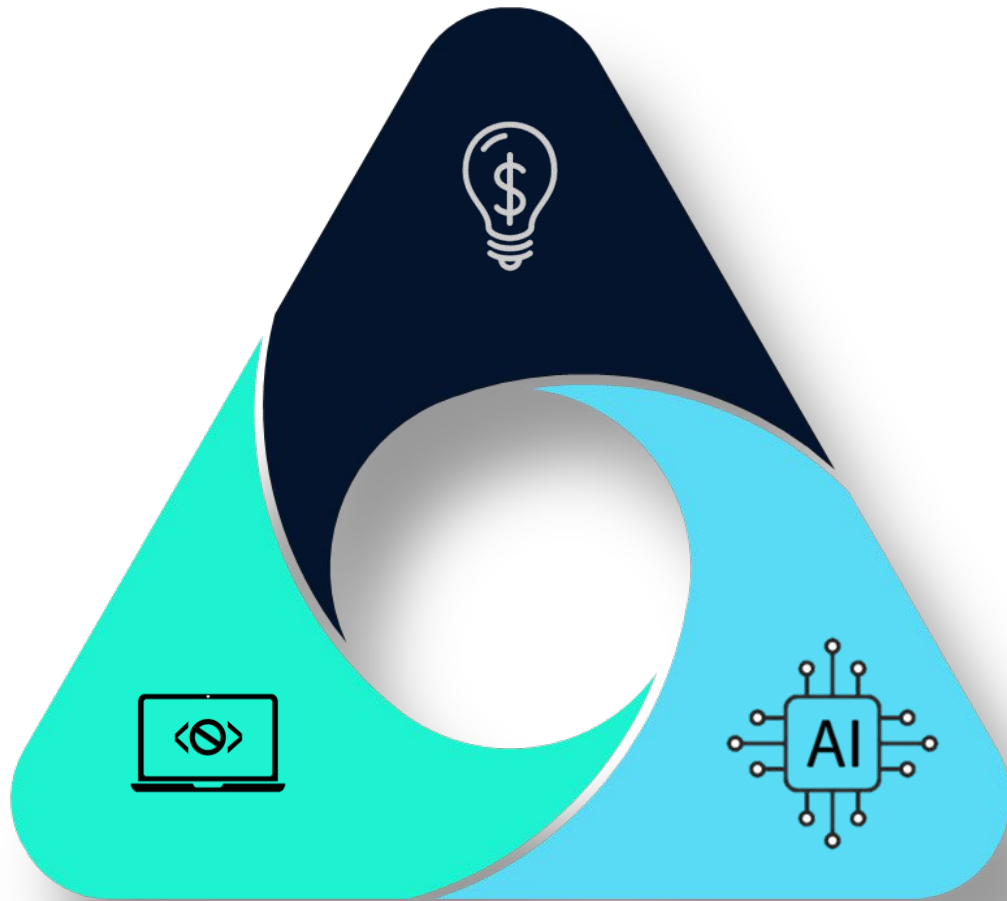




# Fundamentos de Desenvolvimento de Software

Como criar Sistemas e Aplicativos do Zero

# Tríade IA + SaaS NoCode



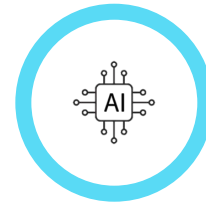
## Desenvolvimento de Produtos

Técnicas e fundamentos de desenvolvimento de produtos são a base para soluções de sucesso



## Desenvolvimento SaaS Nocode

Fundamentos de desenvolvimento de softwares e tecnologias nocode



## Inteligência Artificial e Agentes de IA

Fundamentos de IAs e desenvolvimento de agentes de inteligência artificial

# Tríade IA + SaaS NoCode



## Desenvolvimento de Produtos

Técnicas e fundamentos de desenvolvimento de produtos são a base para soluções de sucesso



## Desenvolvimento SaaS Nocode

Fundamentos de desenvolvimento de softwares e tecnologias nocode



## Inteligência Artificial e Agentes de IA

Fundamentos de IAs e desenvolvimento de agentes de inteligência artificial

# O que você irá aprender

## Fundamentos de Desenvolvimento



### Fundamentos de Desenvolvimento de Apps

Conceitos importantes e como desenvolver seu app



### O que são APIs e Webhooks

Conceitos importantes comunicação de sistemas



### Etapas para você desenvolver seu App

O passo a passo pra tirar sua ideia do papel





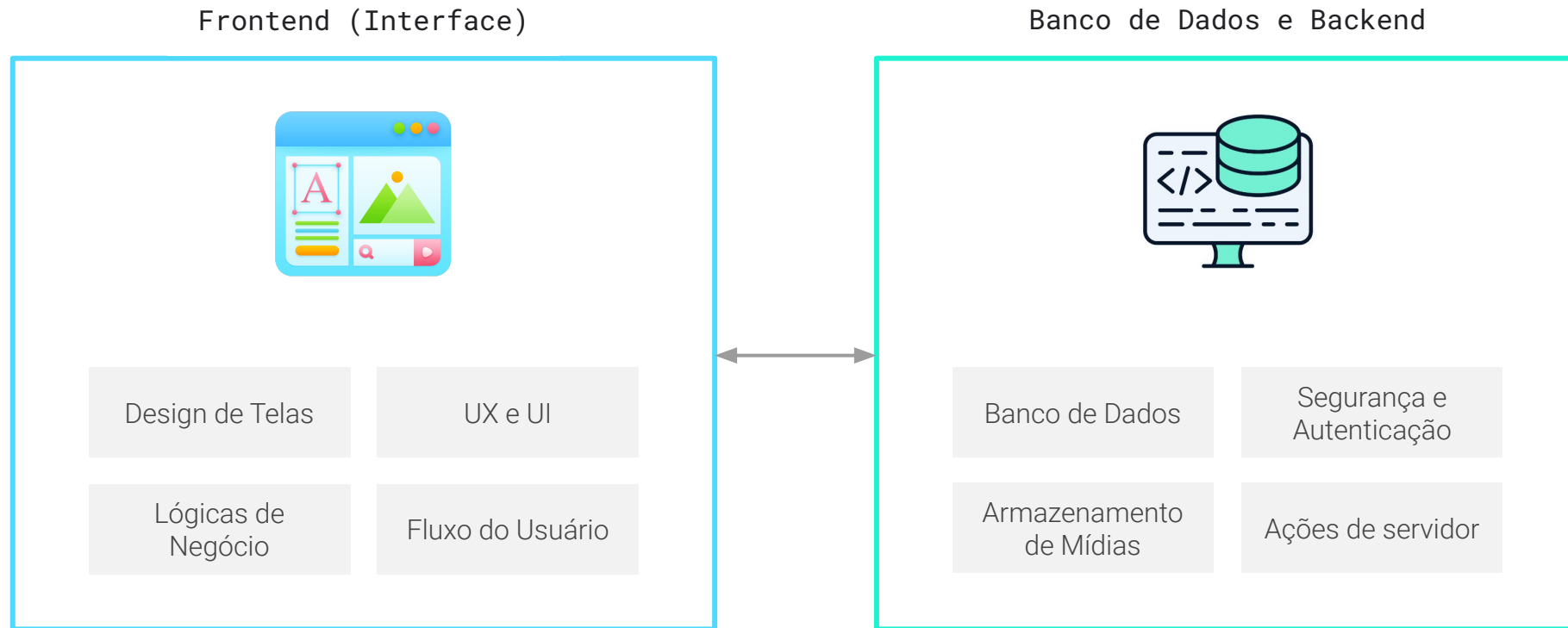
no code start up

no code start up

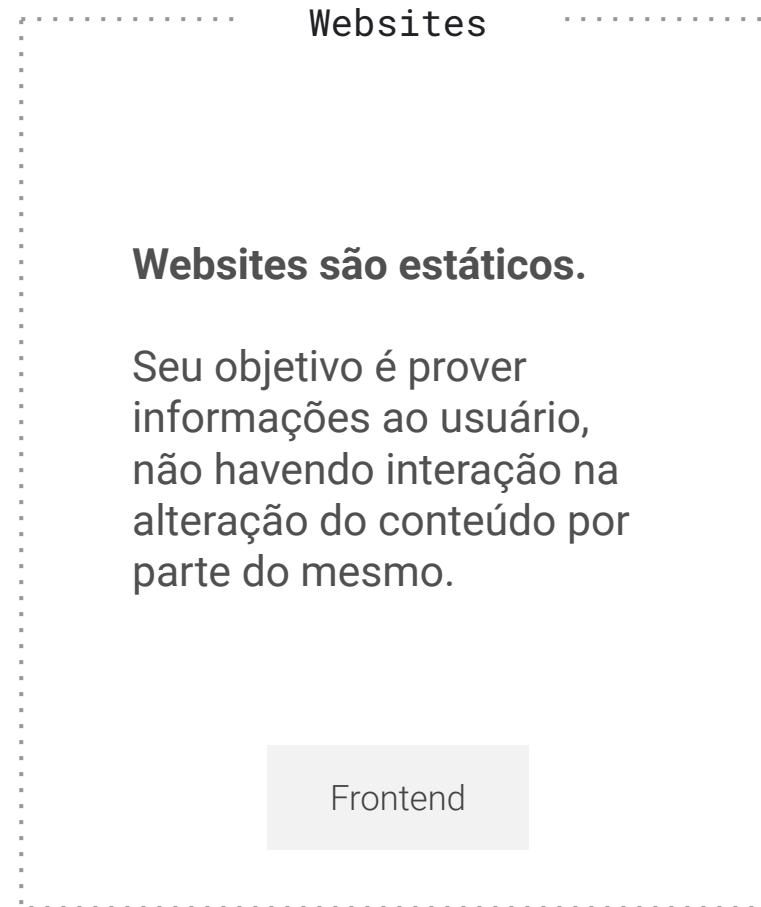
# 0 que são Softwares

Fundamentos Essenciais do Mundo da Tecnologia

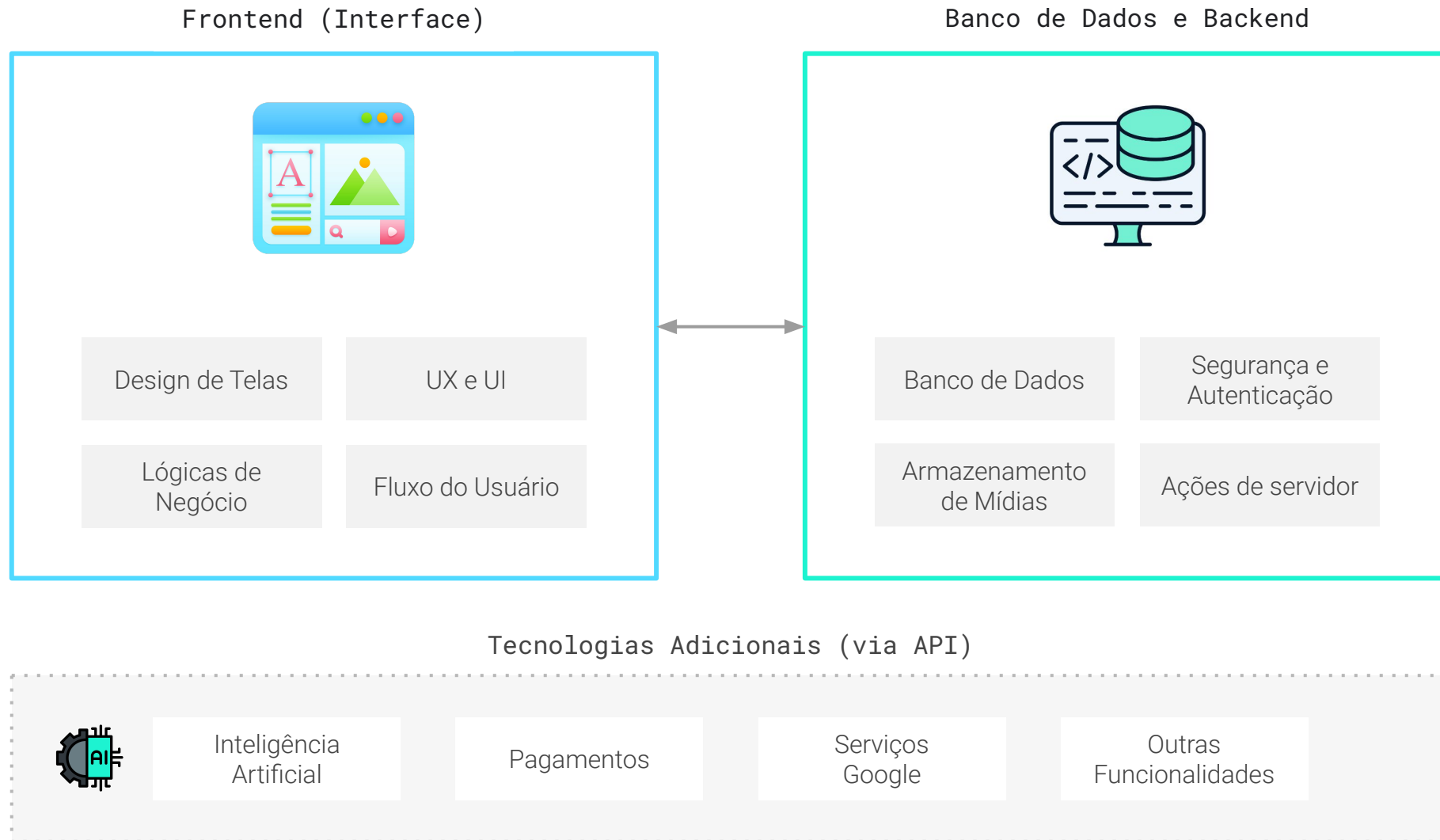
# O que são Softwares?



# Softwares x Websites



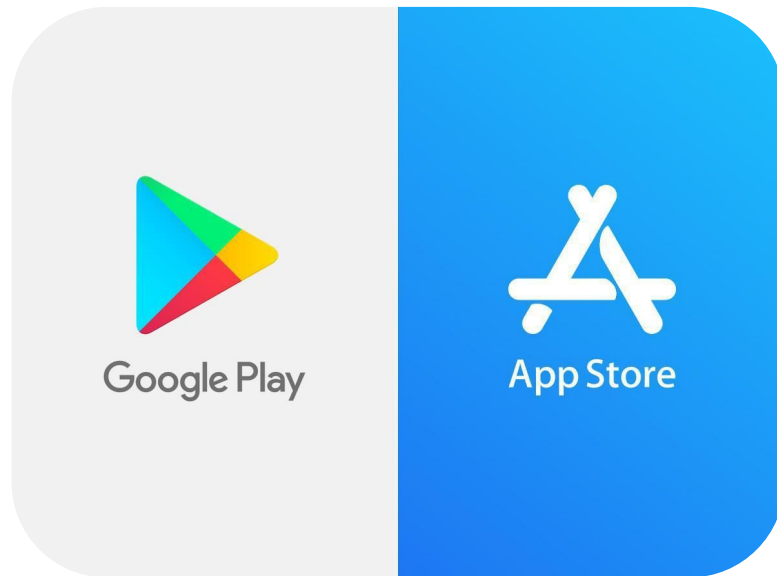
# Arquitetura de um Software / SaaS





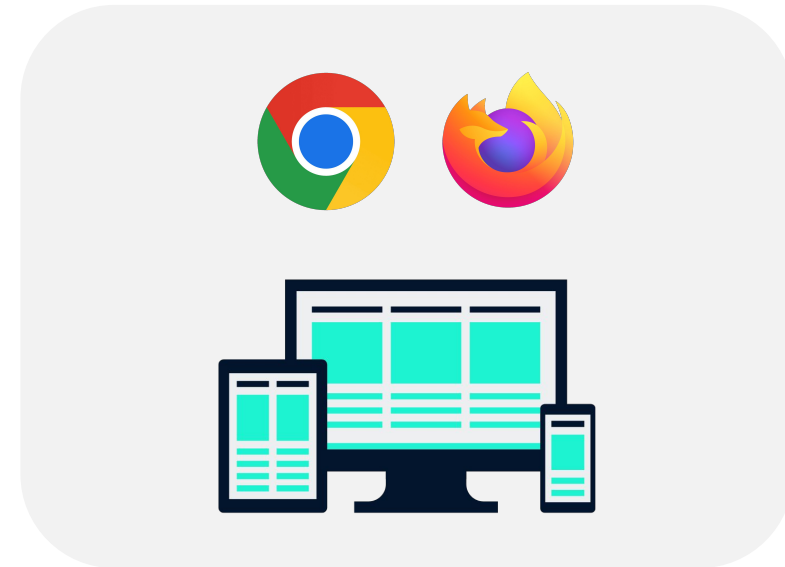
# Apps Nativos x Web Apps

## Aplicativos Nativos



Usam recursos nativos  
do celular

## Aplicativos Web



Web Apps rodam na web e  
podem ser responsivos



# Fundamentos APIs

O processo de desenvolvimento de tecnologia

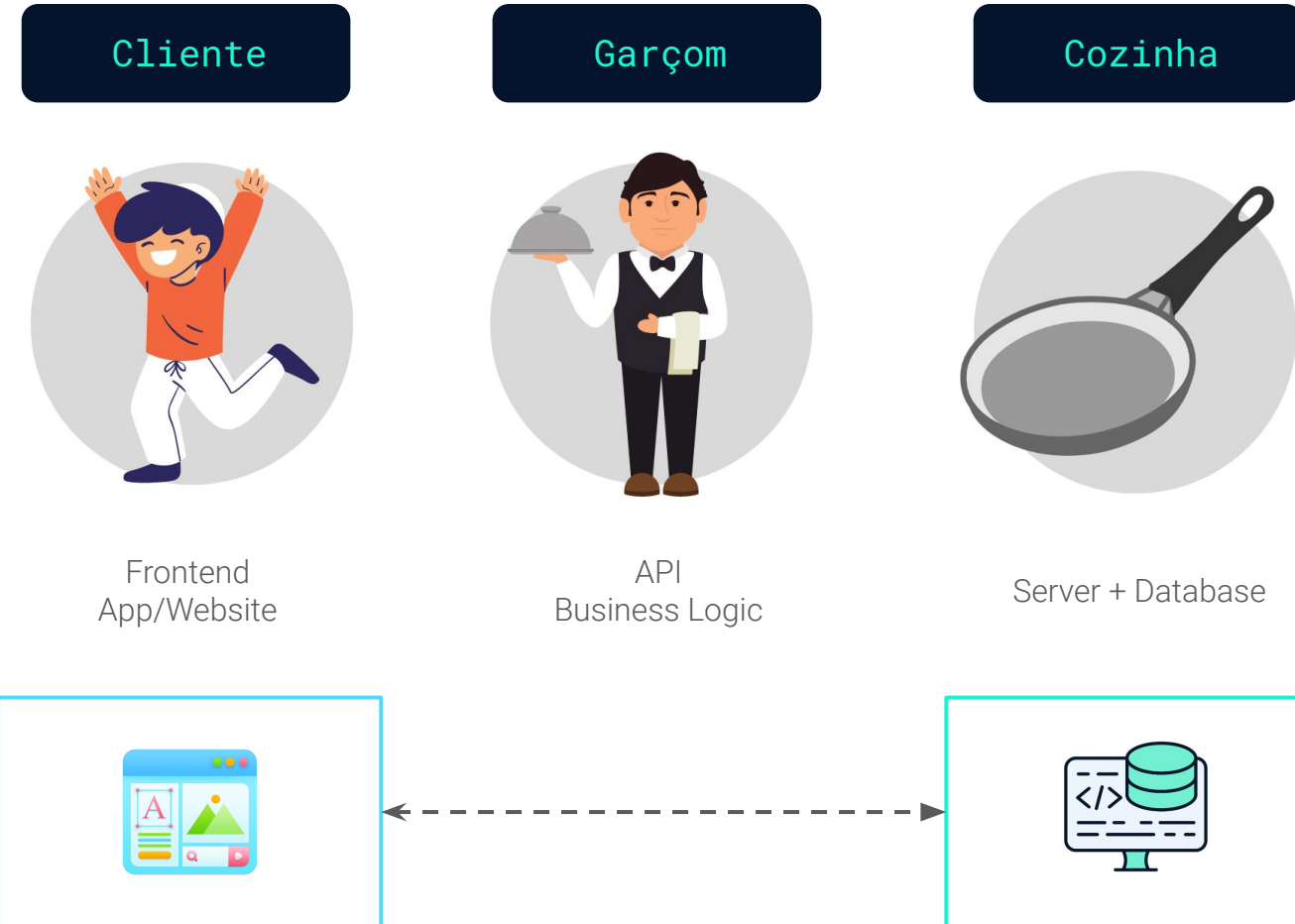
no code start up

no code start up

# A sigla API significa Application Programming Interface

Basicamente é uma interface para comunicação entre diferentes sistemas

# O que são APIs



# 0 que são APIs

Cliente



Frontend  
App/Website

Garçom

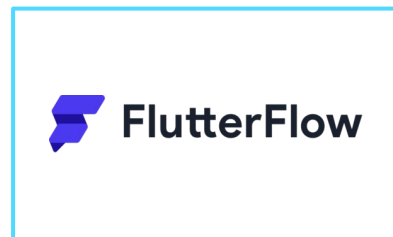


API  
Business Logic

Cozinha



Server + Database



Serviços de  
Backend

# 0 que são APIs

Cliente



Frontend  
App/Website

Garçom

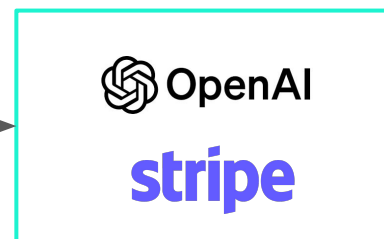
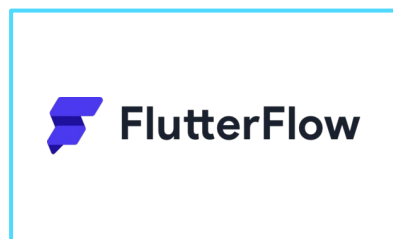


API  
Business Logic

Cozinha



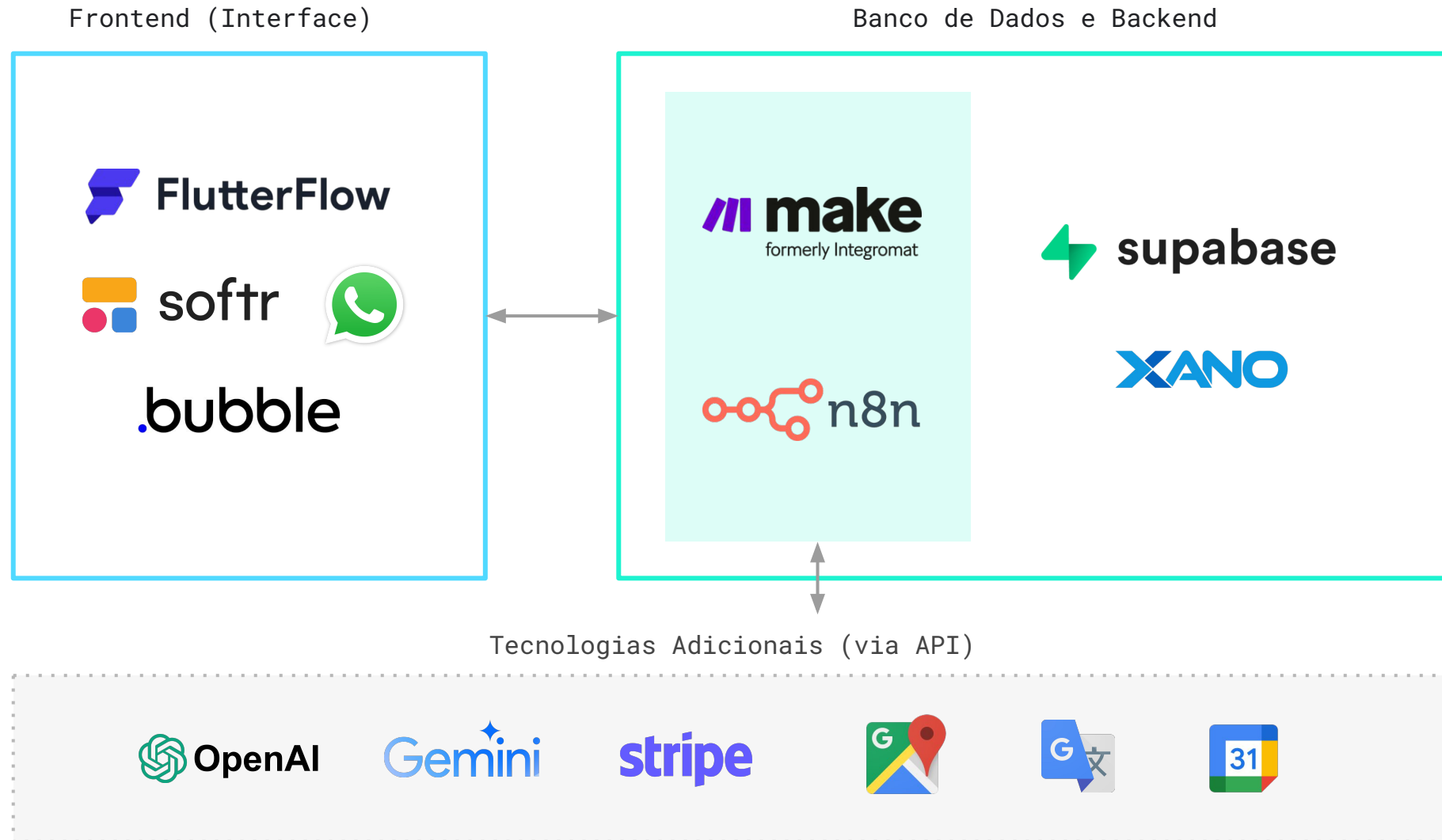
Server + Database



Outras  
plataformas  
com API



# Arquitetura de um Software / SaaS



# Porque é tão importante aprender sobre APIs?

Conseguiremos integrar nosso app com qualquer outra aplicação!



# Principais métodos API

## Métodos HTTP



POST - CREATE - Criar novos dados



GET - READ - Visualizar dados

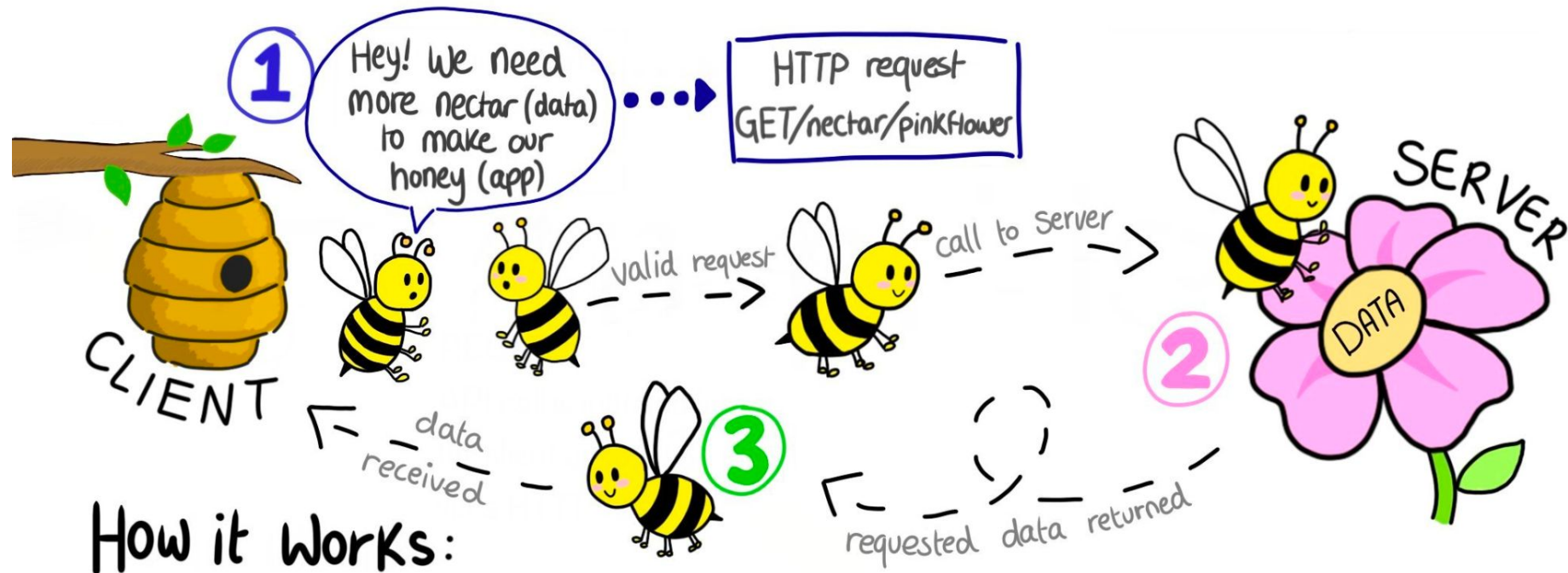


PUT - UPDATE - Atualizar dados



DELETE - DELETE - Deletar dados

# Resumo sobre API



How it Works:

- 1 Request**  
API call is initiated by the Client application via a HTTP request

# Json, a linguagem dos APIs

Forma padrão de comunicação entre sistemas

- Formato leve e rápido no uso de dados
- Leitura simplificada para máquinas e humanos
- Possível criar relações complexas

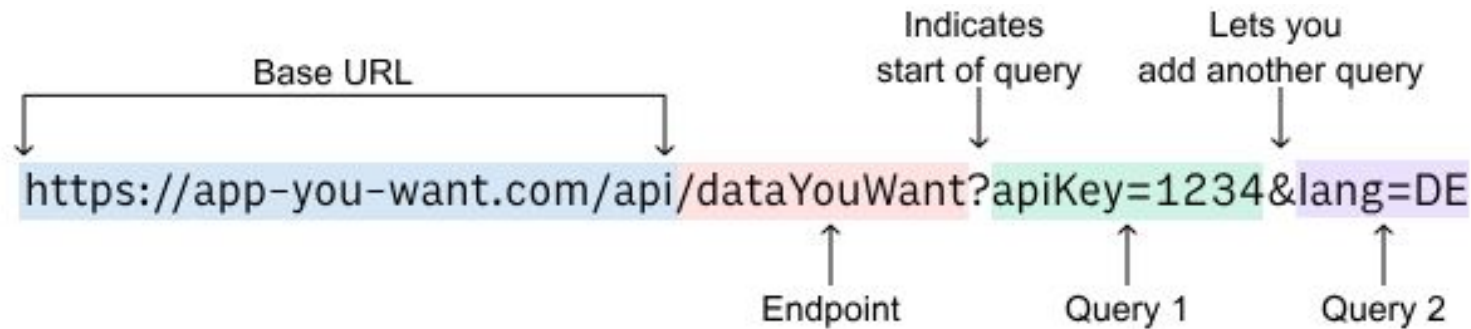
## JSON

```
{ "empinfo" :  
  {  
    "employees" : [  
      {  
        "name" : "James Kirk",  
        "age" : 40,  
      },  
      {  
        "name" : "Jean-Luc Picard",  
        "age" : 45,  
      },  
      {  
        "name" : "Wesley Crusher",  
        "age" : 27,  
      }  
    ]  
  }  
}
```

# Exemplo Prático API e Json



# Estrutura API - Endpoint (url)



# Estrutura API - Chamada Completa

URI (Unique resource identifier) .....→

`https://example.com/api/v1/users`

HTTP METHOD .....→

**POST**

HTTP HEADERS .....→

content-type: application/json  
accept: application/json  
authorization: Bearer eyJhbGciOi.....ssw5c

BODY .....→

```
{  
  "name": "Juan",  
  "job": "author"  
}
```

# Create assistant Beta

POST <https://api.openai.com/v1/assistants>

Create an assistant with a model and instructions.

## Request body

### model Required

ID of the model to use. You can use the [List models](#) API to see all of your available models, or see our [Model overview](#) for descriptions of them.

**name** string or null Optional

The name of the assistant. The maximum length is 256 characters.

**description** string or null Optional

The description of the assistant. The maximum length is 512 characters.

**instructions** string or null Optional

The system instructions that the assistant uses. The maximum length is 32768 characters.

**tools** array Optional Defaults to []

A list of tool enabled on the assistant. There can be a maximum of 128 tools per assistant. Tools can be of types `code_interpreter`, `retrieval`, or `function`.

▼ Show possible types

**file\_ids** array Optional Defaults to []

A list of [file](#) IDs attached to this assistant. There can be a maximum of 20 files attached to the assistant. Files are ordered by their creation date in ascending order.

Code Interpreter

Files



Example request

curl ▾ Copy

```
1 curl "https://api.openai.com/v1/assistants" \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v1" \
5   -d '{
6     "instructions": "You are a personal math tutor.
7     "name": "Math Tutor",
8     "tools": [{"type": "code_interpreter"}],
9     "model": "gpt-4"
10  }'
```

Response

Copy

```
1 {
2   "id": "asst_abc123",
3   "object": "assistant",
4   "created_at": 1698984975,
5   "name": "Math Tutor",
6   "description": null,
7   "model": "gpt-4",
8   "instructions": "You are a personal math tutor. Wh
9   "tools": [
10    {
11      "type": "code_interpreter"
12    }
13  ],
14   "file_ids": [],
15   "metadata": {}
16 }
```



# Como testar APIs de forma completa?







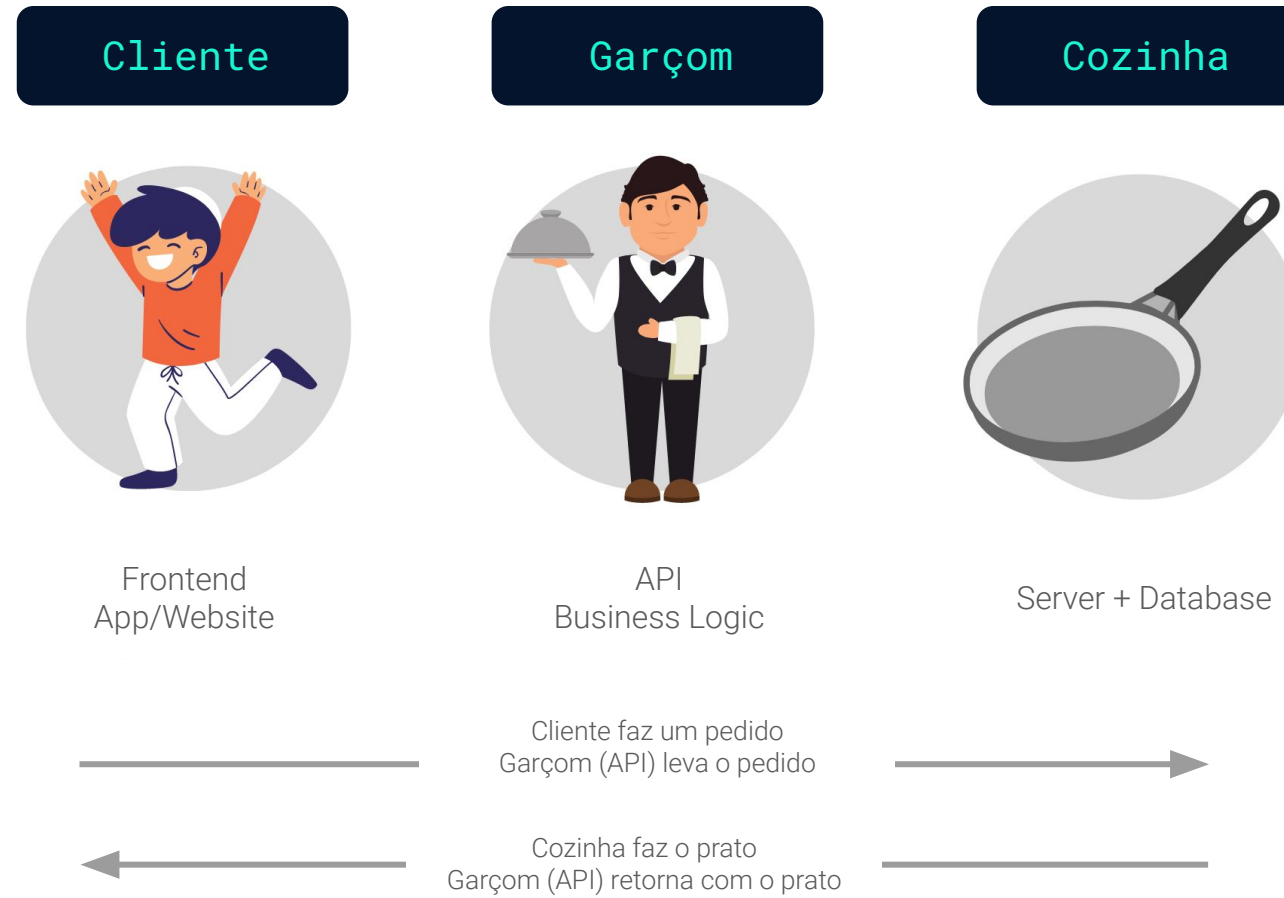
# Webhooks

Como funciona os webhooks

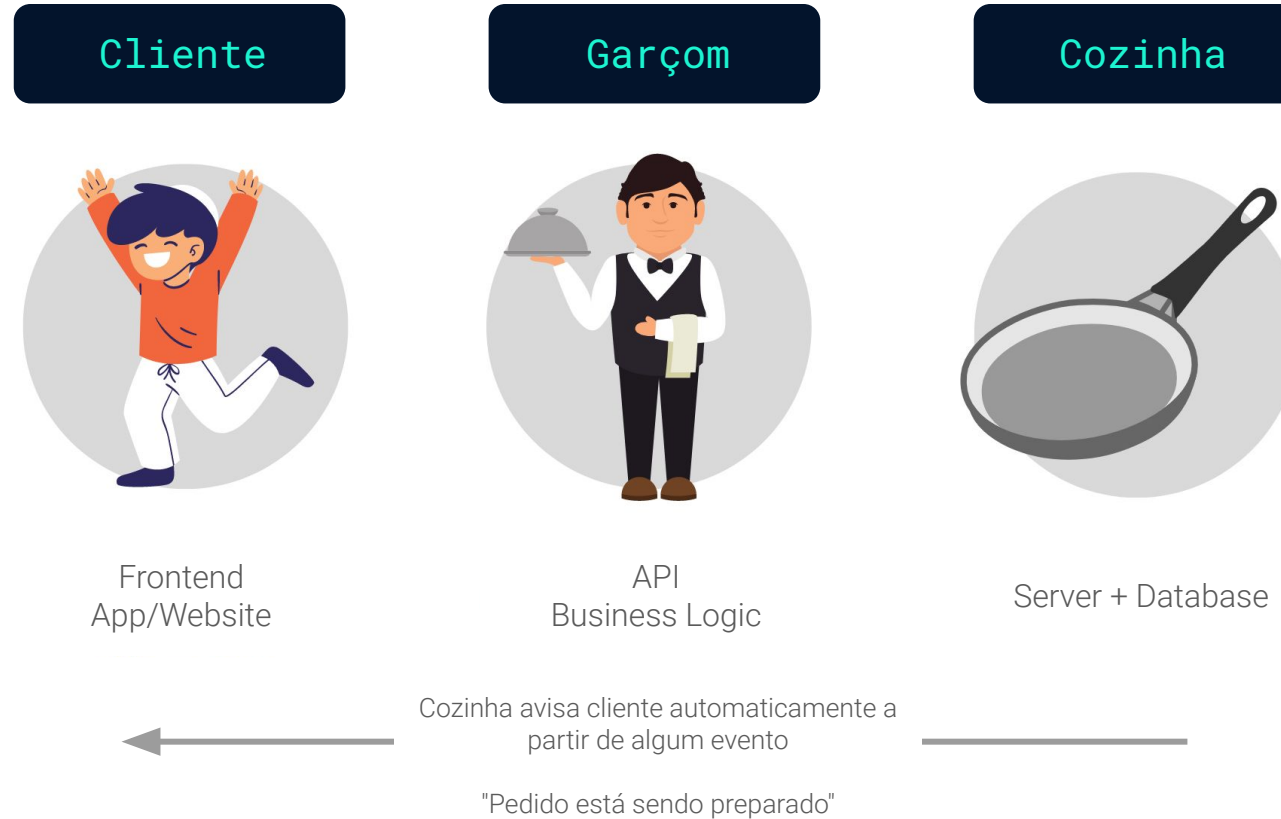
no code start up

no code start up

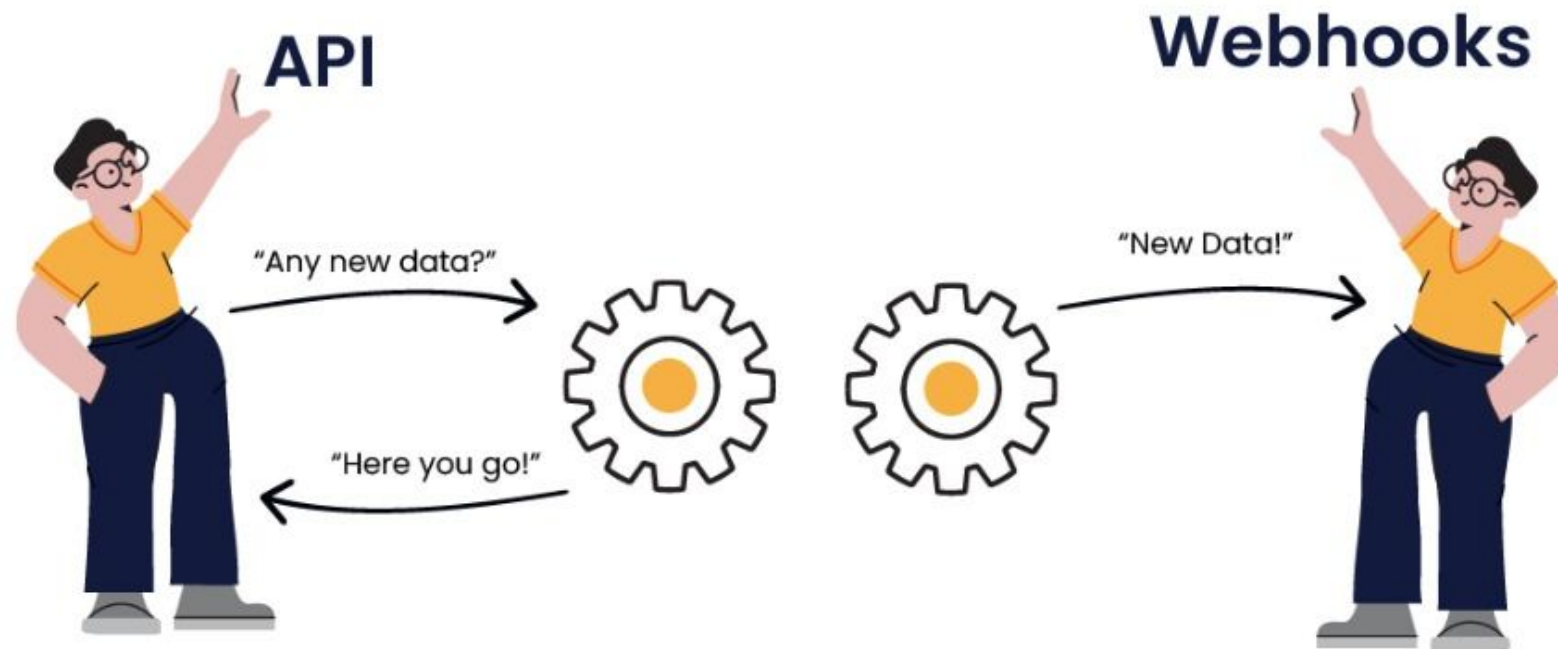
# APIs



# Webhooks



# API x Webhooks



# Exemplo de Webhook





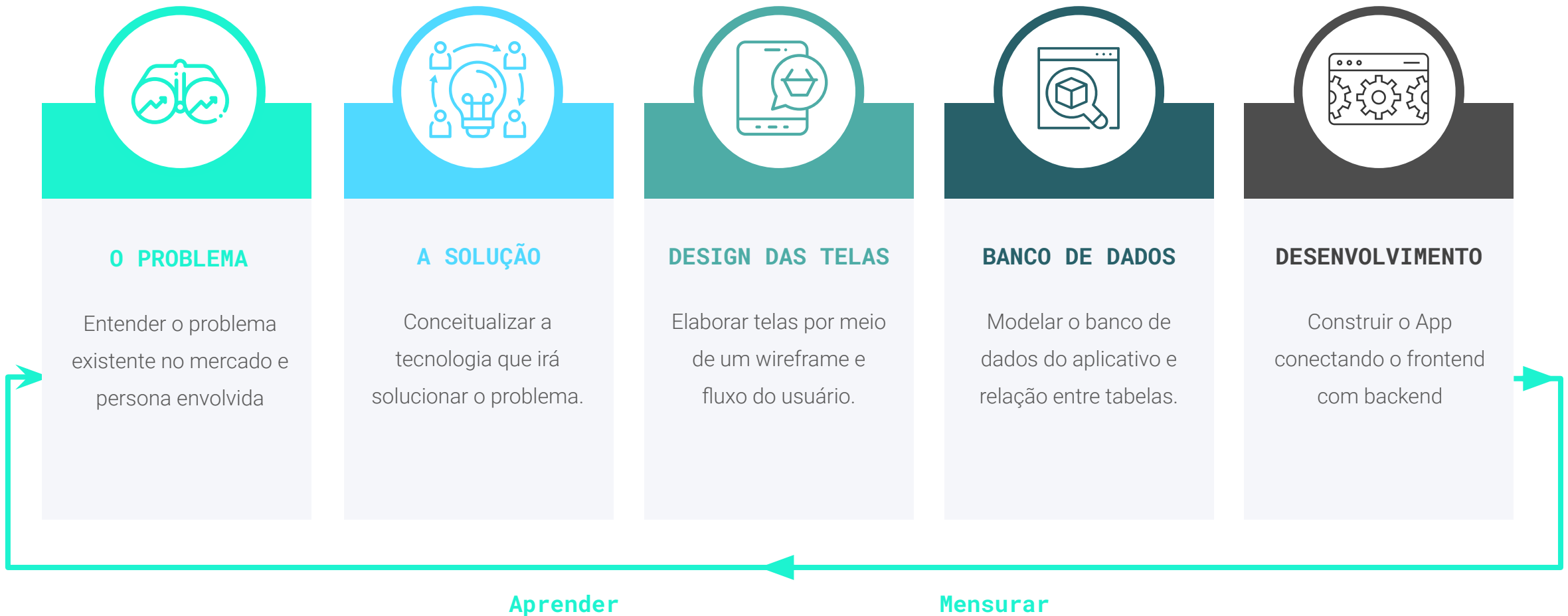
no code start up

no code start up

# Fase de desenvolvimento de Apps

O processo de desenvolvimento de tecnologia

# Metodologia No-Code Start-Up



# Metodologia No-Code Start-Up



## 0 PROBLEMA

Entender o problema  
existente no mercado e  
persona envolvida



# 1 - 0 problema



# Metodologia No-Code Start-Up



## O PROBLEMA

Entender o problema existente no mercado e persona envolvida



## A SOLUÇÃO

Conceitualizar a tecnologia que irá solucionar o problema.

# 2 - Conceito Geral da Solução



## Conceito:

App de Cursos EAD de professores e alunos.  
Os Professores podem cadastrar novos cursos e os alunos podem se matricular nos cursos disponíveis.



## Funcionalidades Core do App:

- Professores
  - Visualizar Lista de seus Cursos Criados
  - Visualizar Alunos matriculados em cada curso
  - Adicionar, Editar e Deletar cursos
  - Adicionar, Editar e Deletar aulas dos cursos
- Alunos
  - Visualizar Lista de Cursos Disponíveis
  - Visualizar Detalhes do Curso
  - Opção de se matricular no curso
  - Opção de remover a matrícula do curso
  - Opção de dar o check de curso realizado
- Ambos
  - Upload imagem de perfil
- Administradores do App
  - Podem tornar um usuário em tipo professor



## Páginas:

- Tela 1 - Login e Cadastro
- Tela 2 - Dashboard Home Aluno
  - Cursos Matriculados - Cursos Disponíveis
- Tela 2 - Dashboard Home Professor
  - Meus Cursos Criados
  - Popup Adicionar novo Curso
- Popups
  - Detalhes Cursos - Editar e Deletar
  - Adicionar Novo Curso
  - PopUp meu Perfil



## Tipos de usuários:

- Administradores do App
- Aluno
- Professor

# Metodologia No-Code Start-Up



## O PROBLEMA

Entender o problema existente no mercado e persona envolvida



## A SOLUÇÃO

Conceitualizar a tecnologia que irá solucionar o problema.



## DESIGN DAS TELAS

Elaborar telas por meio de um wireframe e fluxo do usuário.

# 3 - Design de Telas

Geração de  
Ideias Criativas



**Bēhance**

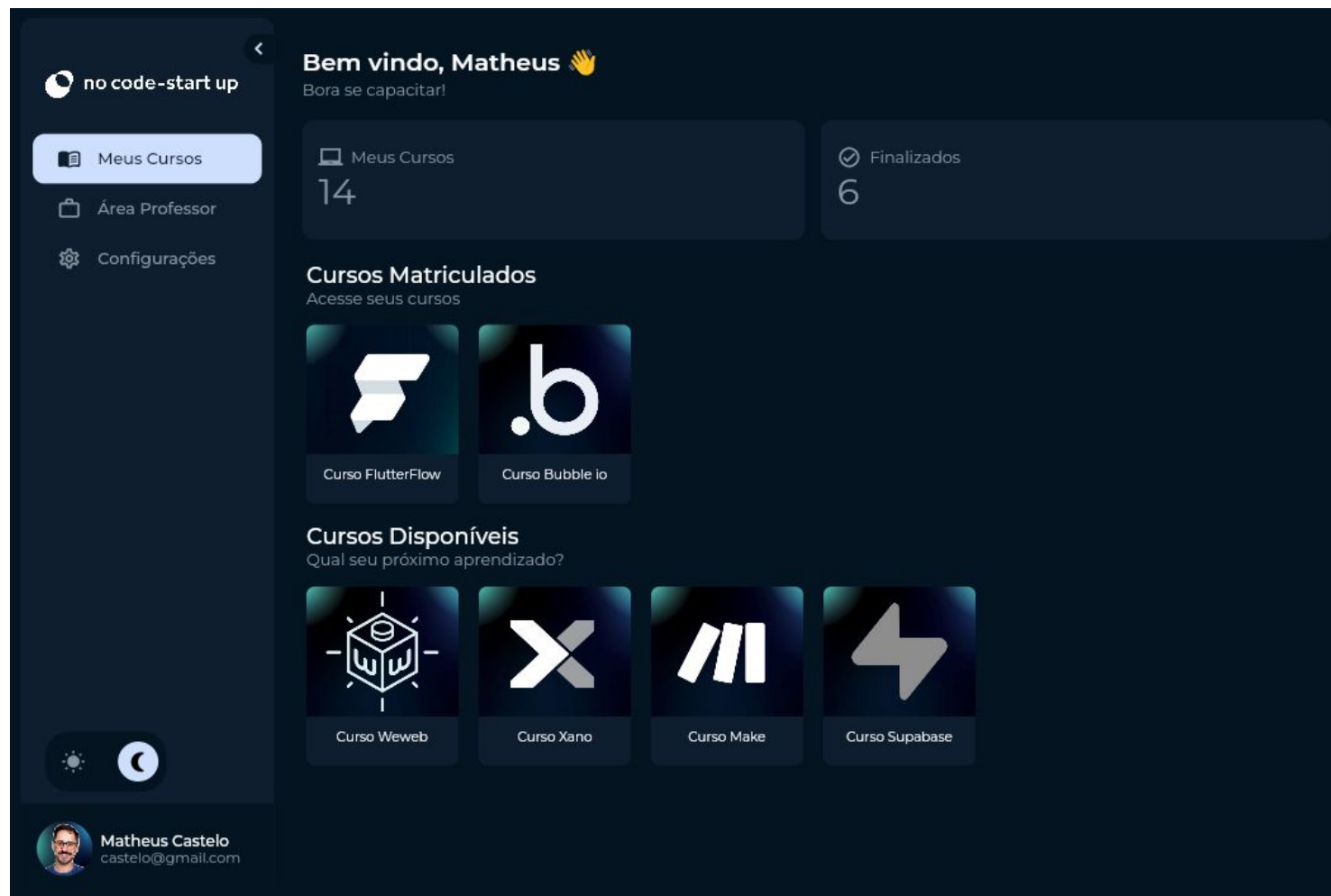
Prototipagem



# 3 - Design de Telas - Prototipagem



# 3 - Design de Telas - Alta Fidelidade



# Metodologia No-Code Start-Up



## O PROBLEMA

Entender o problema existente no mercado e persona envolvida



## A SOLUÇÃO

Conceitualizar a tecnologia que irá solucionar o problema.



## DESIGN DAS TELAS

Elaborar telas por meio de um wireframe e fluxo do usuário.



## BANCO DE DADOS

Modelar o banco de dados do aplicativo e relação entre tabelas.

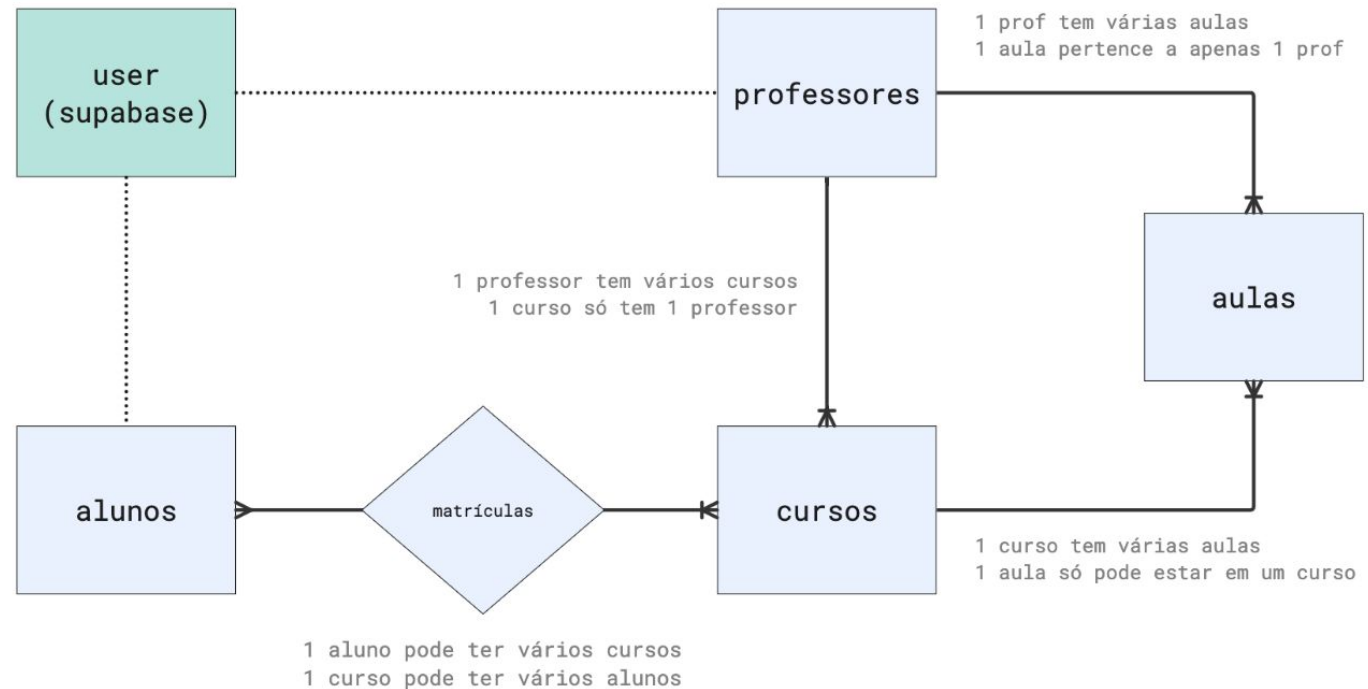


# 4 - Modelagem de Dados - Modelo Conceitual

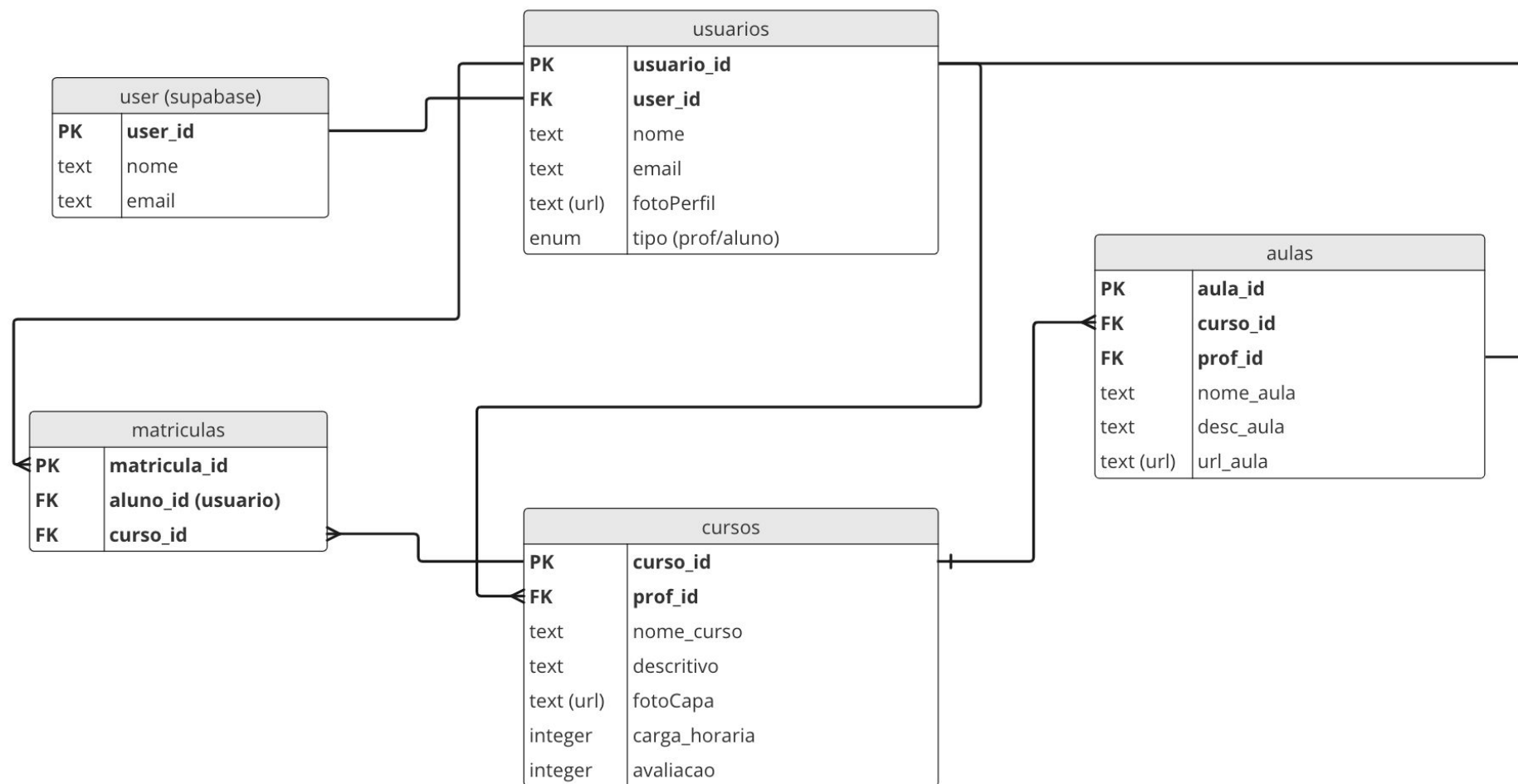


## Tabelas:

- alunos
- professores
- cursos
- aulas
- matricula



# 4 - Modelagem de Dados - Modelo Lógico



# Metodologia No-Code Start-Up



## O PROBLEMA

Entender o problema existente no mercado e persona envolvida



## A SOLUÇÃO

Conceitualizar a tecnologia que irá solucionar o problema.



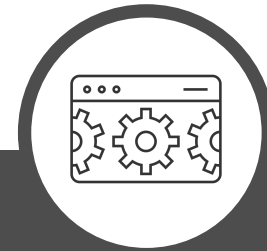
## DESIGN DAS TELAS

Elaborar telas por meio de um wireframe e fluxo do usuário.



## BANCO DE DADOS

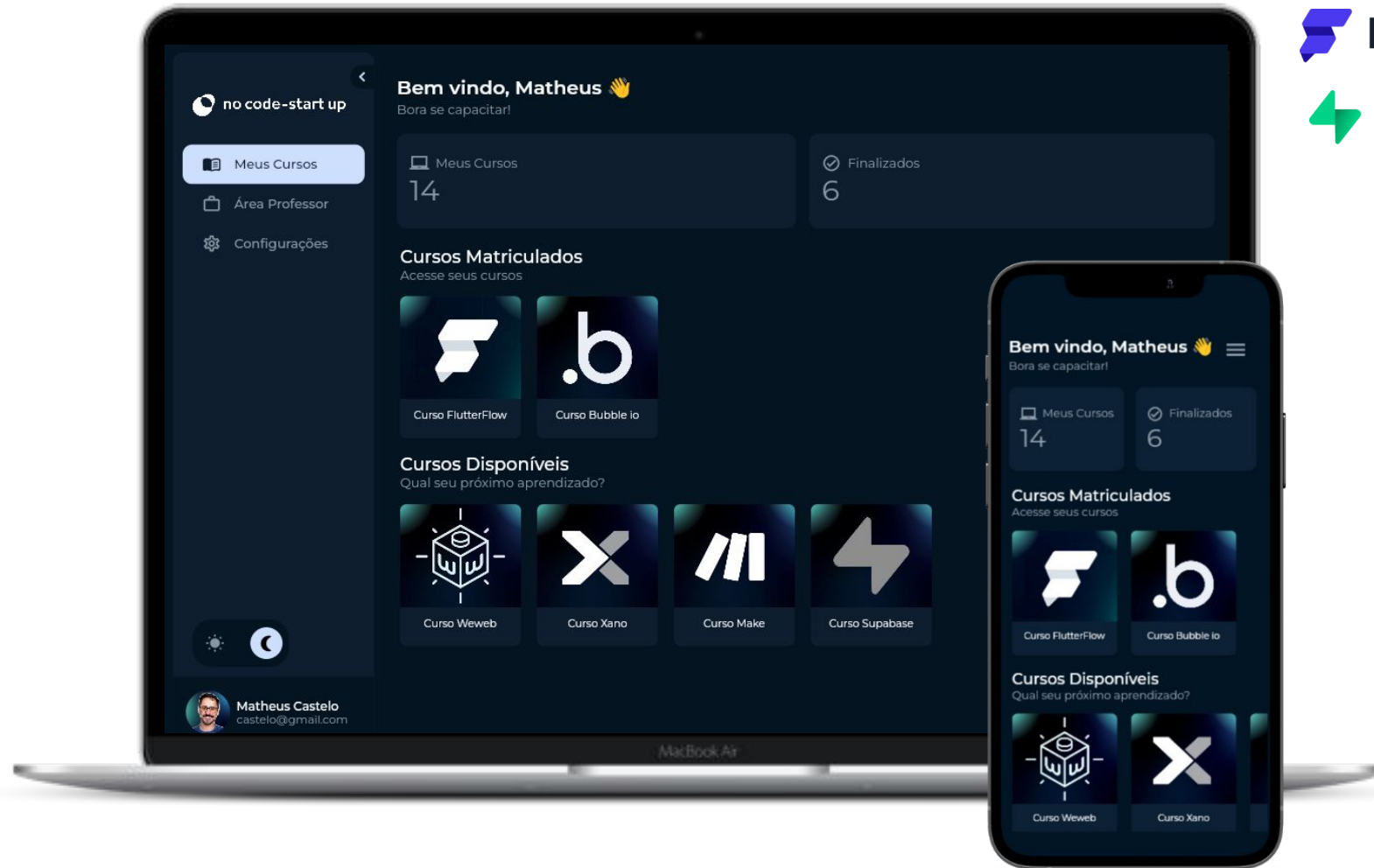
Modelar o banco de dados do aplicativo e relação entre tabelas.



## DESENVOLVIMENTO

Construir o App conectando o frontend com backend

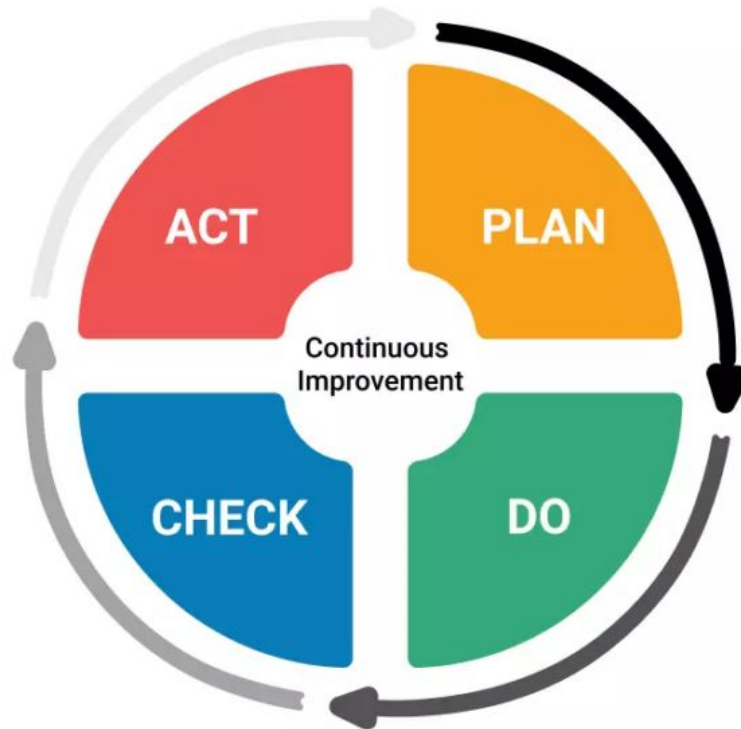
# 5 - Desenvolvimento



 FlutterFlow

 supabase

# Melhoria Contínua - PDCA





no  
code  
start  
up

Matheus Castelo



@mat\_castelo



/in/matheuscastelobranco



Neto Camarano



@netocamarano

