THE DEVELOPER'S CONFERENCE

Trilha – API

Juscélio Reis
Desenvolvedor

Agenda

THE DEVELOPER'S CONFERENCE

- Sobre
- Case Wiz Soluções
- Inspiração
- Modelos
  - Contract First
  - Code First
- Erros

Sobre o palestrante

THE DEVELOPER'S CONFERENCE

- Juscélio Reis
  - Pai 
  - Geek
  - +10 anos de experiencia em desenvolvimento de software.
  - Pesquisador
- Mas....
  - Wiz Soluções

WIZ

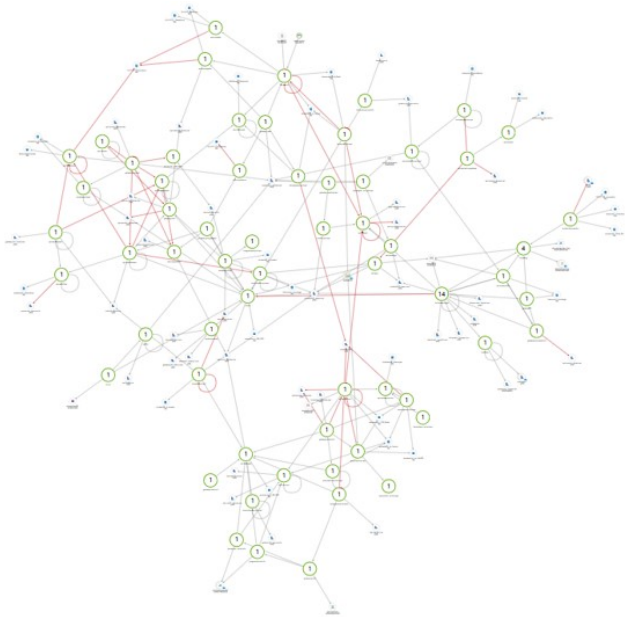
Wiz Soluções

Advisor Back-end (03/2020 – atual)

"O nosso Advisor será responsável por desenvolver e implementar o Framework de Back-end e será o guardião da metodologia utilizada em todo o conglomerado Wiz."

"... Estar antenado no mercado de APIs, frameworks e funcionalidades prontas para trazer as melhores soluções para a Wiz."

Foco em governança.

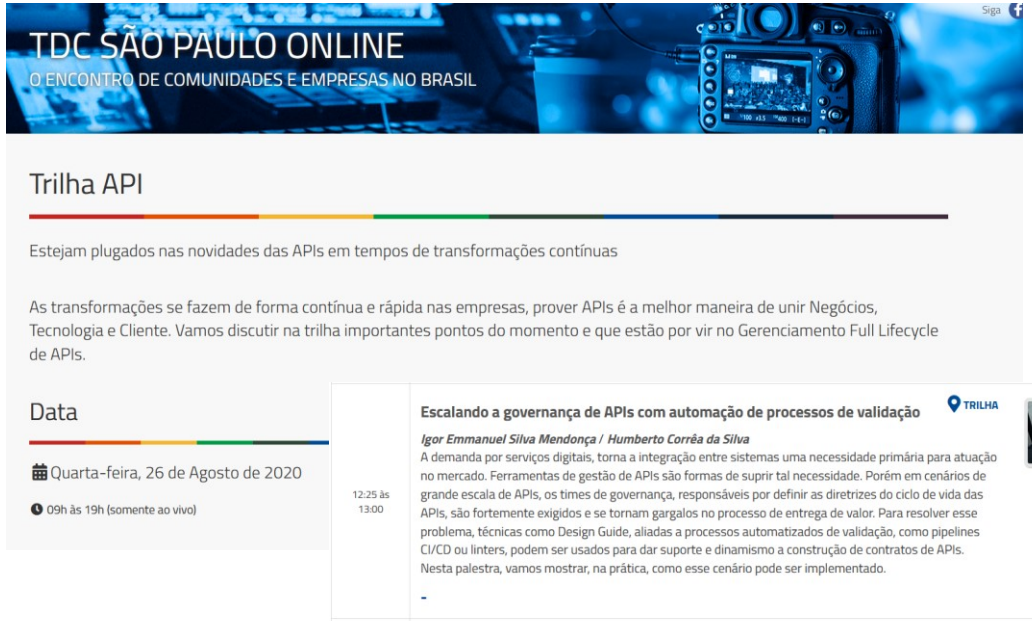


Wiz
Wiz Corporate
Wiz Parceiros
Wiz BPO
Wiz Benefícios
Wiz conseg
Wiz B2U

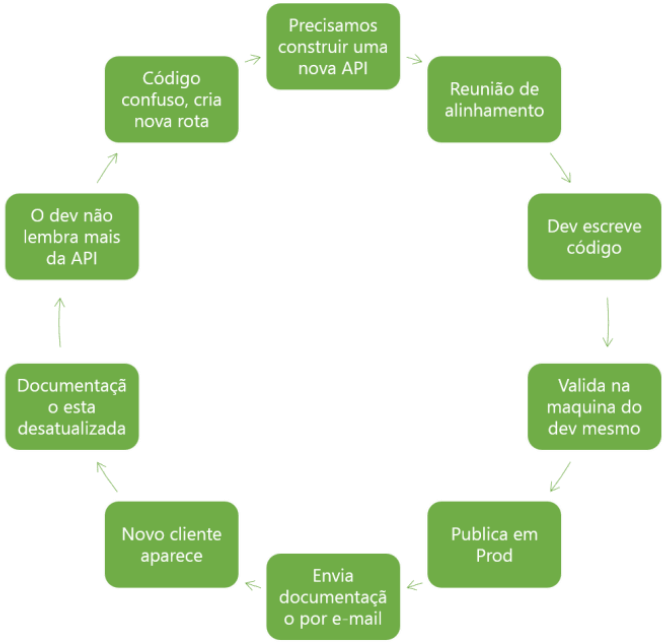
Inspirações

Em agosto de 2020 no TDC SP assisti uma palestra sobre governança e uma ferramenta chamada Spectral (Open Source).

E mais muito conteúdo na Web depois.....



Code First

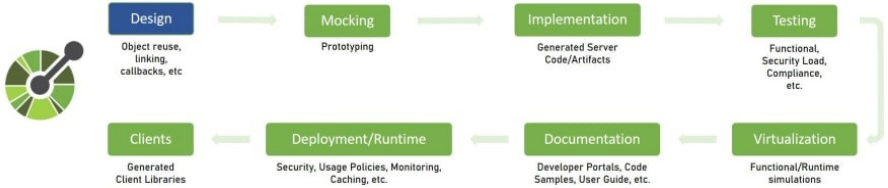


- Tradicional.
- Devs estão acostumado.
- Area negocial não vê problema.
- Problema em atualizar etapas como SDK e doc.

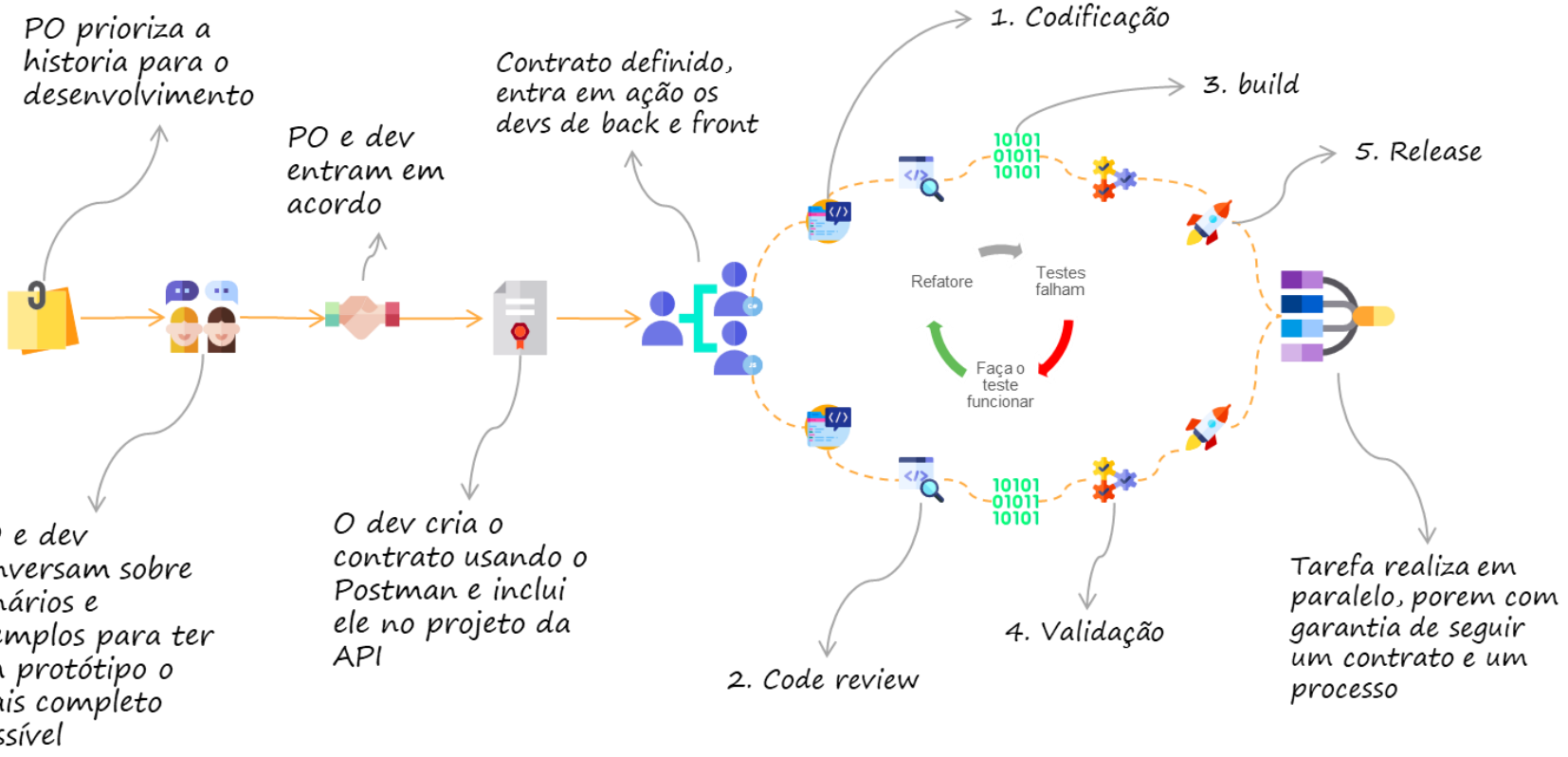
Contract First

THE DEVELOPER'S CONFERENCE

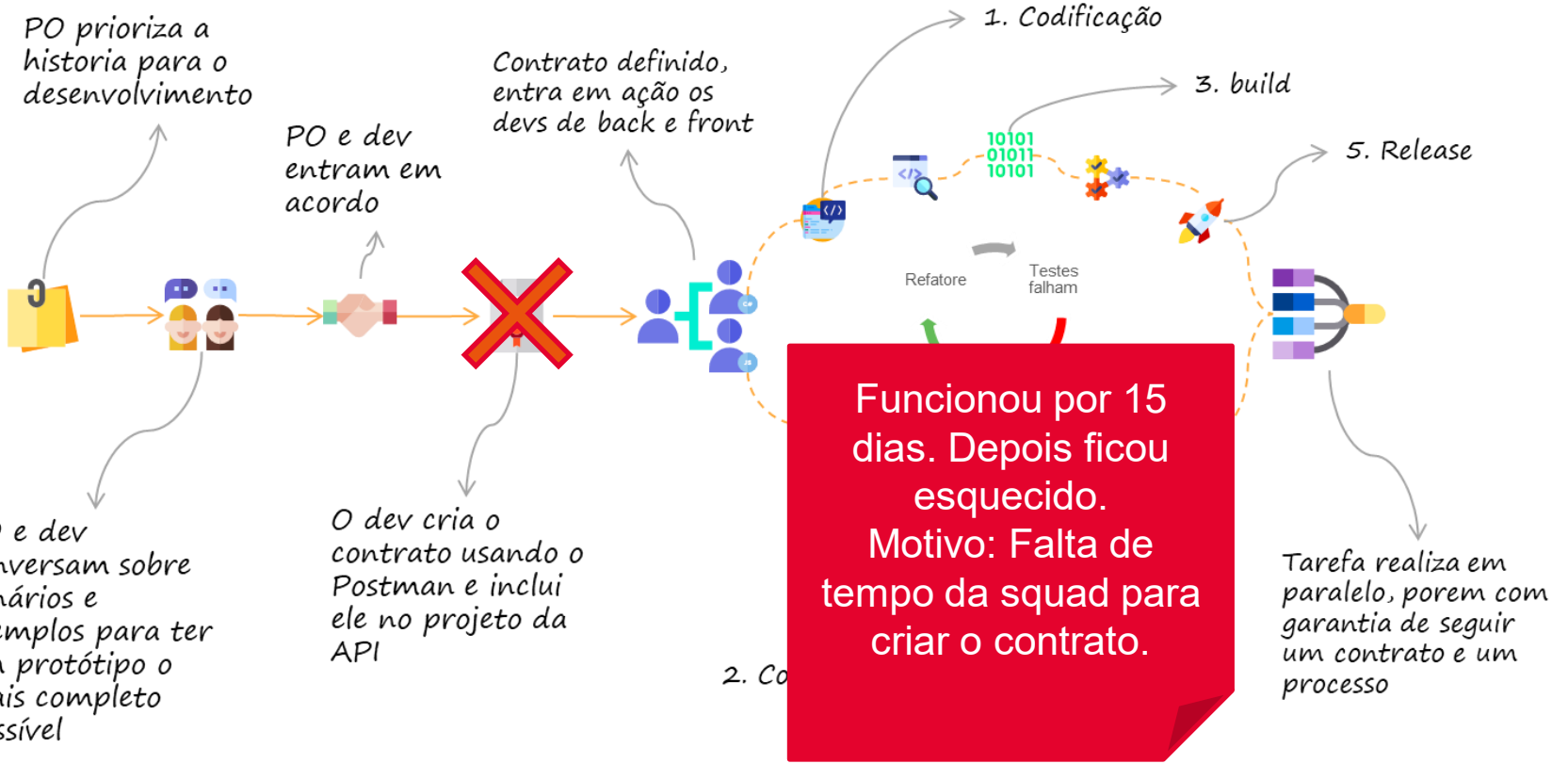
- Excelente oportunidade para colher feedbacks importantes.
- Dev backend e frontend podem trabalhar em paralelo.
- Facilita automação de algumas etapas: sdk, doc, cliente, server.
- Mas pode ser mais demorado. (não acho, mas outras pessoas acham)



Primeira versão



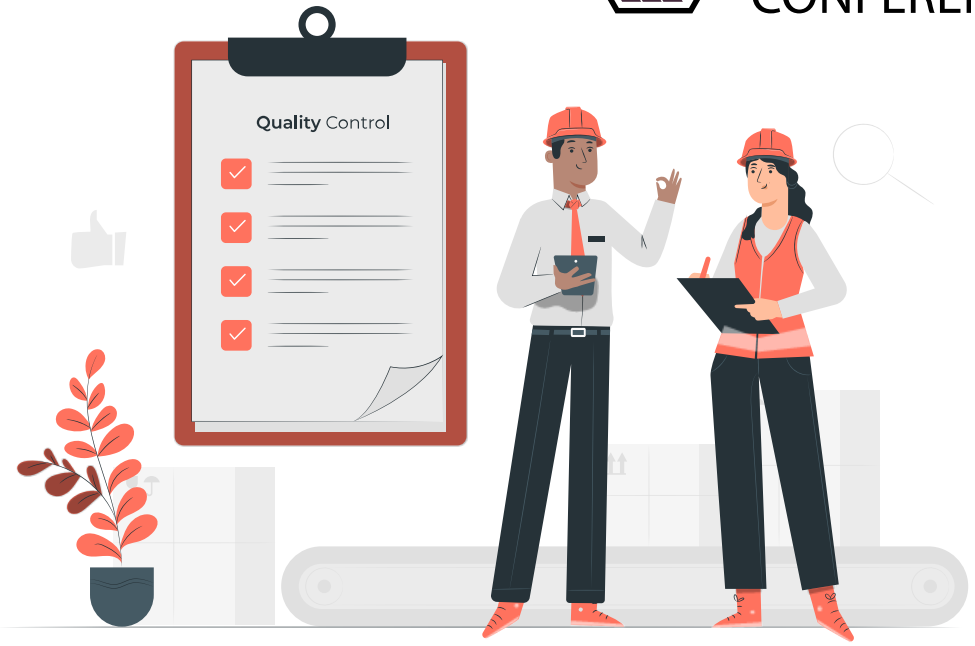
Primeiro fracasso



THE DEVELOPER'S CONFERENCE

Lições aprendidas

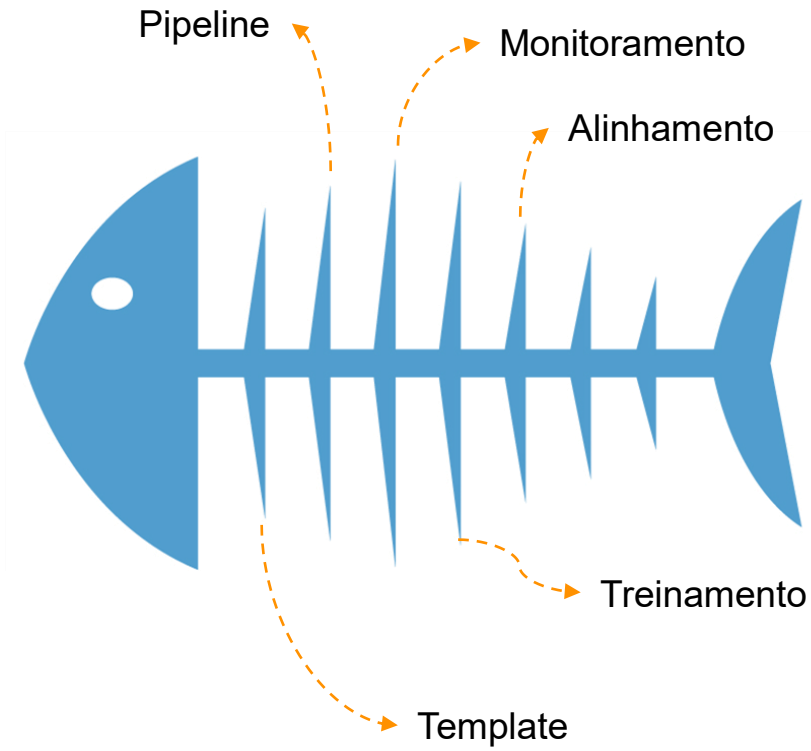
- Não ter autoridade funcional
- Area negocial focada em outras entregas.
- Desenvolvedores fazem quando querem.
- Como garantir que sigam uma metodologia.
- Fast track quebra o fluxo.



Construir a v2

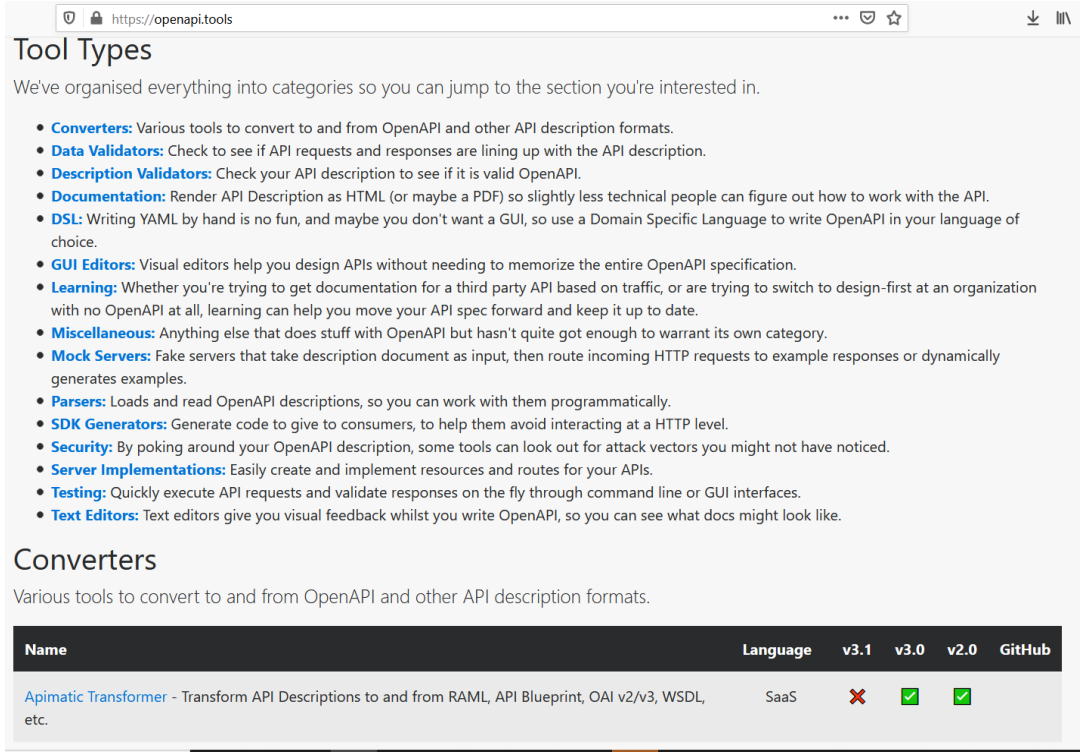
- O processo de CI/CD esta sob meu controle.
- Spectral é uma ferramenta de linha de comando.
- Existem outras ferramentas que geram OpenAPI do código. Ex: Nswag.
- Temos template para criar APIs.

> dotnet new -i Wiz.Dotnet.Template.API
> dotnet new wizapi -n [NomeProjeto]



O poder do padrão

- OpenAPI
  - OpenAPI v2 (Swagger)
  - OpenAPI v3
  - OpenAPI v3.1
- API Blueprint
- RAML



THE DEVELOPER'S CONFERENCE

Unindo os 2 mundos

Quero a automação e qualidade do Contrac First.

Mas quero que os devs não alteram seu modelo de trabalho, que eles não sabotem a metodologia.

Com o pipeline mais as ferramentas do OpenApi posso alcançar esse objetivo.

Template de API

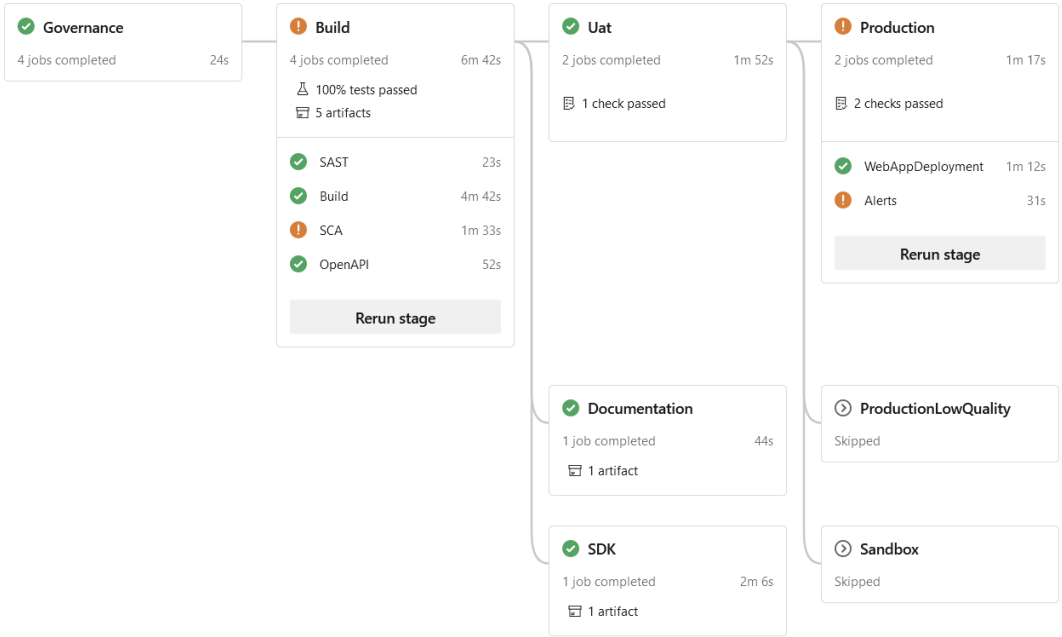
- Tem as dependências que eu quero.
- Posso dar o exemplo para o desenvolvedor.
- Consigo criar *lints* para o dev seguir antes de fazer um *commit*.
- Ajuda na aceleração do desenvolvimento.

```
/// <summary>
/// Lista de clientes.
/// </summary>
/// <response code="200">Retorna lista de endereços.</response>
/// <response code="400">Erro de requisição.</response>
/// <response code="401">Acesso negado.</response>
/// <response code="500">Erro interno da API.</response>
[ProducesResponseType(typeof(IEnumerable<CustomerAddressViewModel>), 200)]
[ProducesResponseType(typeof(ProblemDetails), 400)]
[ProducesResponseType(typeof(ProblemDetails), 401)]
[ProducesResponseType(typeof(ProblemDetails), 500)]
[HttpGet]
public async Task<ActionResult<IEnumerable<CustomerAddressViewModel>>> GetAll()
{
    return Ok(await _customerService.GetAllAsync());
}
```

https://github.com/wizsolucoes/api-wiz-template

Etapas do pipeline

- Governança
  - Garantir as informações mínimas.
- Build
  - SAST
    - DevSkim
    - Sonar
  - Build (+Tests)
  - SCA
    - Dependency Track
  - OpenAPI
    - NSwag
- Documentação
  - openapi-generator
- SDK
  - Nswag
- UAT | Sandbox | Production



OpenAPI: automação

THE DEVELOPER'S CONFERENCE

```
- task: PowerShell@2
  name: NSwag
  displayName: Run NSwag
  inputs:
    targetType: 'inline'
    script: |
      dir ../bin

      $dll = Get-ChildItem -Path ../bin | Where-Object -FilterScript {$_.Name -match "Wiz.*.API.dll"} | Select FullName, Name -f 1
      Write-Output "$($dll.FullName)"

      $fileExists = Test-Path -Path "$($dll.FullName)"

      $proj = $($dll.Name -replace "Wiz.", "") -replace ".Api.dll", ""

      Invoke-Expression -Command "sudo nswag run ./integration/jobs/nswag.json /variables:Configuration=Development,DLL=$($dll.FullName),Namespace=Wiz.Api.$($proj).Client,Name=$($proj)Client /runtime:NetCore31"
```

Amarração

THE DEVELOPER'S CONFERENCE

```yaml
- task: PowerShell@2
  name: spectral
  displayName: Run Spectral
  env:
    SYSTEM_ACCESSTOKEN:
    Build_Repository_Name: $(Build.Repository.Name)
    System_PullRequest_PullRequestId: $(System.PullRequest.PullRequestId)
  inputs:
    ignoreLASTEXITCODE: true
    targetType: 'inline'
    script: |
      Invoke-Expression -Command "sudo spectral lint $(Build.Repository.LocalPath)/openapi.json -r ./integration/jobs/wiz.spectral.yaml -f junit -o $(Build.Repository.LocalPath)/openapicode.xml"

      Invoke-Expression -Command "sudo spectral lint $(Build.Repository.LocalPath)/openapi.json -r ./integration/jobs/wiz.spectral.yaml -f json -o $(Build.Repository.LocalPath)/openapicode.json"

      $results = Get-Content $(Build.Repository.LocalPath)/openapicode.json
      $results = $results | ConvertFrom-Json
```

wiz.spectral.yaml

Contents History Compare Blame

```yaml
extends:
  - spectral:oas
  - spectral:asyncapi

rules:
  wiz-info-title-default-name:
    description: Title must be different from "My Title".
    type: style
    given: $.info
    severity: error
    then:
      field: title
      function: pattern
      functionOptions:
        notMatch: "/My Title/g"

  wiz-info-title-template-name:
    description: Title must be different from "Whitelabel API".
    type: style
    given: $.info
    severity: error
    then:
      field: title
      function: pattern
      functionOptions:
        notMatch: "/Whitelabel API/g"

  wiz-info-description-template:
    description: Title must be different from "API de Whitelabel".
    type: style
    given: $.info
    severity: error
    then:
      field: description
      function: pattern
      functionOptions:
        notMatch: "/API de Whitelabel/g"
```

Doc: automação

THE DEVELOPER'S CONFERENCE

```yaml
- script: |
    sudo openapi-generator-cli generate -i $(Pipeline.Workspace)/openapi.json -g markdown -o $(Build.Repository.LocalPath)/html
  displayName: Generate documentation
  workingDirectory: $(System.DefaultWorkingDirectory)
```

OptInApi.md | Preview OptInApi.md ×

OptInApi

All URIs are relative to *http://localhost*

Method	HTTP request	Description
optInAdd	**POST** /api/v1/pessoa	Adiciona um novo cliente
optInForget	**DELETE** /api/v1/pessoa/cpf/{cpf}	Opção pelo direito de esquecimento do cliente
optInGetOptions	**GET** /api/v1/pessoa/cpf/{cpf}	Busca informações de opt-in de um cliente pelo cpf
optInUpdate	**PUT** /api/v1/pessoa/cpf/{cpf}/optin	Atualiza informações de opt-in do cliente
optInUpdateContact	**PUT** /api/v1/pessoa/cpf/{cpf}/contact	Atualiza a forma de contato preferencial do usuário

optInAdd

```
PersonOptions optInAdd(personOptions)
```

Adiciona um novo cliente

Parameters

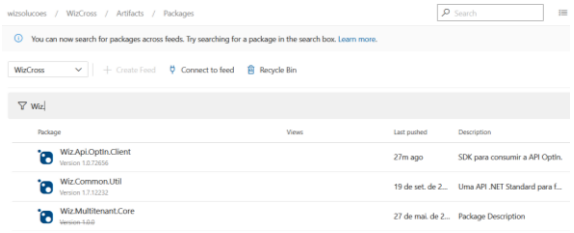
Name	Type	Description	Notes
personOptions	PersonOptionsViewModel	Parâmetro com as informações do cliente a serem gravadas	

SDK: automação

THE DEVELOPER'S CONFERENCE

```yaml
- task: PowerShell@2
  name: NSwag
  displayName: Run NSwag
  inputs:
    targetType: 'inline'
    script: |
      dir ../bin

      $dll = Get-ChildItem -Path ../bin | Where-Object -FilterScript {$_.Name -match "Wiz.*.API.dll"} | Select FullName, Name -f 1
      Write-Output "$($dll.FullName)"

      $fileExists = Test-Path -Path "$($dll.FullName)"

      $proj = $($dll.Name -replace "Wiz.", "") -replace ".Api.dll", ""

      Invoke-Expression -Command "sudo nswag run ./integration/jobs/nswag.json /variables:Configuration=Development,DLL=$($dll.FullName),Namespace=Wiz.Api.$($proj).Client,Name=$($proj)Client /runtime:NetCore31"
```

```csharp
HttpClient httpClient = new HttpClient();

ClientCredentials config = new ClientCredentials(
  baseAuthUrl: "https://sso-hml-web.azurewebsites.net",
  clientId: "ApiOptIn",
  clientSecret: "SHQzHdoUXk--------------------3NuIEw22",
  scope: "api-optin.fullaccess api-optin");

OptInClient client = new OptInClient("https://optin-hml-api.azurewebsites.net", httpClient, config);

PersonOptions personOptionGet = await client.GetOptionsAsync("02340061105");
Console.WriteLine(personOptionGet.ToJson());
```



Frontend

THE DEVELOPER'S CONFERENCE

```
# Usando NPM
npm i -g @stoplight/prism-cli

# Usando Yarn
yarn add global @stoplight/prism-cli
```

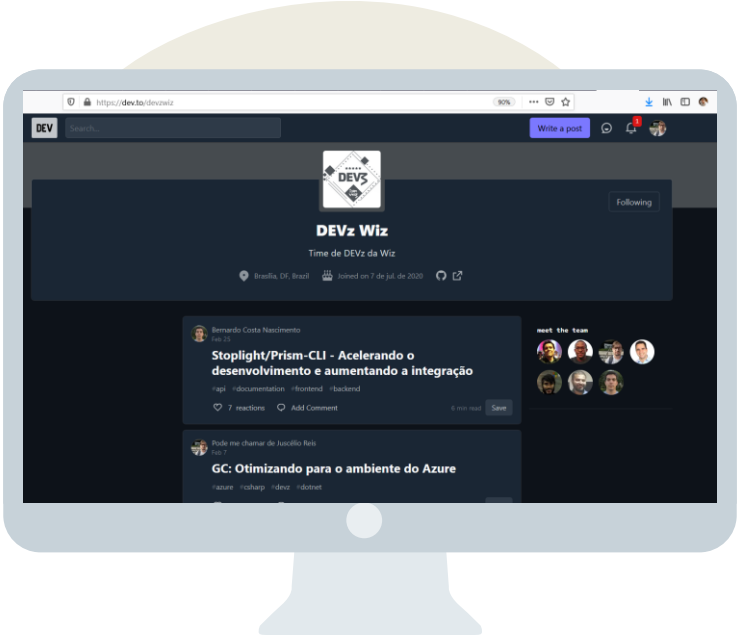
```
prism mock -d http://localhost:5001/swagger/v1/swagger.json
```

```
bash-3.2$
```

```
namespace Wiz.PrismAPI.API.ViewModels.Customer
{
    /// <example>
    /// { "id": 1, "addressId": 1, "name": "John Doe", "dateCreated": "2021-02-23T17:46:37.364Z", "cep": "35362971", "address": { "id":
    /// </example>
    16 references | Bernardo Costa Nascimento, a day ago | 1 author (Bernardo Costa Nascimento)
    public class CustomerAddressViewModel
    {
        [JsonConstructor]
        2 references
        public CustomerAddressViewModel(int id, int addressId, string name, DateTime dateCreated, string cep, AddressViewModel address)
        {
            Id = id;
            AddressId = addressId;
            Name = name;
            DateCreated = dateCreated;
            CEP = cep;
            Address = address ?? new AddressViewModel(addressId, cep,null,null,null);
        }

        [JsonSchemaExtensionData("minimum", 1)]
        [JsonSchemaExtensionData("x-faker", "random.number")]
        1 reference
        public int Id { get; set; }
        [JsonSchemaExtensionData("minimum", 1)]
        [JsonSchemaExtensionData("x-faker", "random.number")]
        1 reference
        public int AddressId { get; set; }
        [JsonSchemaExtensionData("x-faker", "name.findName")]
        1 reference
        public string Name { get; set; }
        1 reference
        public DateTime DateCreated { get; set; }
        [JsonSchemaExtensionData("x-faker", "address.zipCode")]
        1 reference
        public string CEP { get; set; }
        1 reference
        public AddressViewModel Address { get; set; }
    }
```

Mais informação

THE DEVELOPER'S CONFERENCE



https://dev.to/devzwiz
https://github.com/wizsolucoes

THE DEVELOPER'S CONFERENCE