



**Flask**

Flask

## *Programação Web III*

*Prof. Diego Max*

# Flask

## Aula 06: Integração com APIs

*Consumindo uma API, passando parâmetros para rotas*

# API - Application Programming Interface

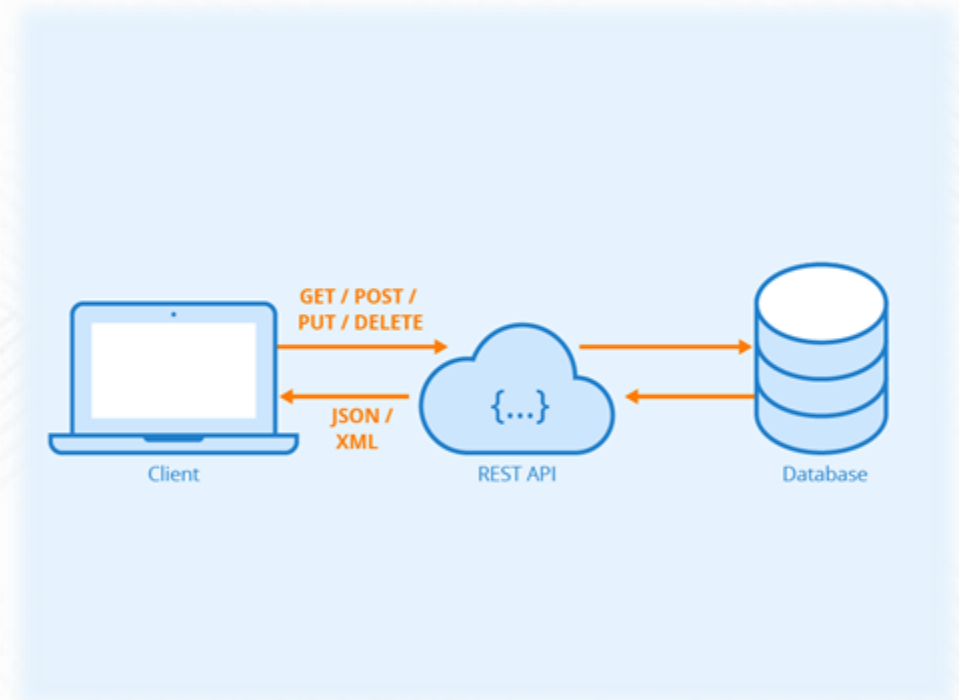


API significa **Interface de Programação de Aplicações** (Application Programming Interface).

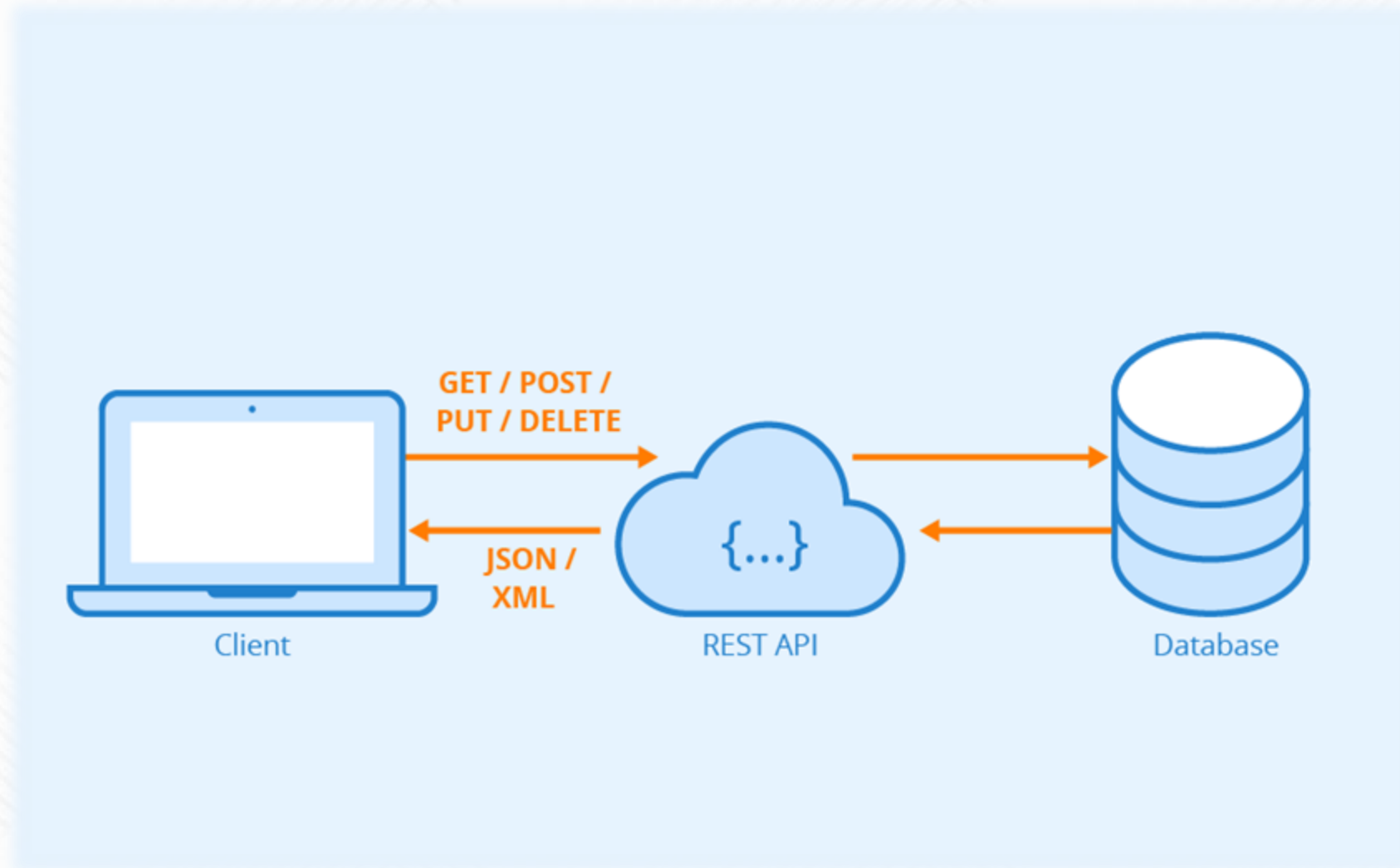
É um conjunto de regras e definições que **permite que diferentes softwares se comuniquem entre si**.

Uma API é como um "**conector**" que deixa diferentes programas conversarem entre si.

Em vez de precisar entender tudo que acontece dentro de um software, os desenvolvedores usam a API, que dá as regras e as maneiras corretas de se comunicar com o sistema.



# API - Application Programming Interface



# API - Application Programming Interface



Essas interfaces são úteis para juntar sistemas diferentes, fazendo com que eles troquem informações e trabalhem juntos. Assim, dá pra criar aplicativos que usam funcionalidades de outros lugares, facilitando a vida dos programadores.

As APIs servem para muita coisa, como acessar dados em serviços online, conectar com apps de terceiros, automatizar tarefas, etc. São importantes para construir sistemas complexos e **permitir que diferentes sistemas, mesmo que estejam em ambientes distintos, troquem informações e funcionem bem juntos.**





**Existem inúmeros exemplos de APIs utilizadas em diferentes contextos. Aqui estão alguns exemplos:**

**Google Maps API:** Permite que desenvolvedores incorporem mapas interativos em seus aplicativos.

**Twitter API:** Possibilita a integração de funcionalidades do Twitter em outros aplicativos ou sites.

**Facebook Graph API:** Permite que aplicativos interajam com a plataforma do Facebook, acessando dados de usuários e publicações.

**GitHub API:** Oferece funcionalidades para interagir com repositórios, issues e outros recursos do GitHub.

**YouTube API:** Permite que os desenvolvedores integrem vídeos do YouTube em seus aplicativos.

**NASA API:** Disponibiliza dados relacionados ao espaço, como imagens de telescópios e informações sobre missões espaciais.

**Instagram API** (anteriormente, pois o Instagram atualmente tem restrições de acesso): Permitia que os desenvolvedores integrassem funcionalidades do Instagram em seus aplicativos.

# Flask

## **Tópico 06.1:** Consumindo uma API pública *Integração com APIs*

# Consumindo uma API pública



Em nossa aplicação consumiremos uma API pública de jogos que está localizada no seguinte endereço:

<https://www.freetogame.com/api/games>

Ao acessá-la será retornado um arquivo JSON com os dados da API:

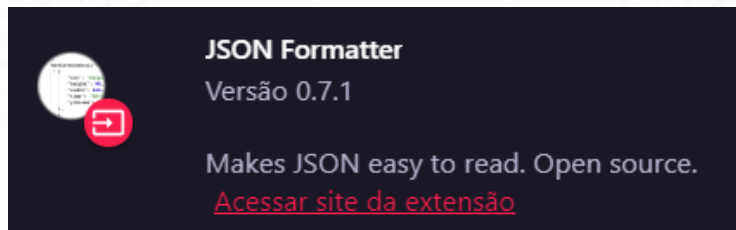
```
[{"id":540,"title":"Overwatch 2","thumbnail":"https://www.freetogame.com/g/540/thumbnail.jpg","short_description":"A hero-focused first-person team shooter from Blizzard Entertainment.", "game_url":"https://www.freetogame.com/open/overwatch-2", "genre":"Shooter", "platform":"PC (Windows)", "publisher":"Activision Blizzard", "developer":"Blizzard Entertainment", "release_date":"2022-10-04", "freetogame_profile_url":"https://www.freetogame.com/overwatch-2"}, {"id":521,"title":"Diablo Immortal","thumbnail":"https://www.freetogame.com/g/521/thumbnail.jpg","short_description":"Built for mobile and also released on PC, Diablo Immortal fills in the gaps between Diablo II and III in an MMOARPG environment.", "game_url":"https://www.freetogame.com/open/diablo-immortal", "genre":"MMOARPG", "platform":"PC (Windows)", "publisher":"Blizzard Entertainment", "developer":"Blizzard Entertainment", "release_date":"2022-06-02", "freetogame_profile_url":"https://www.freetogame.com/diablo-immortal"}, {"id":517,"title":"Lost Ark","thumbnail":"https://www.freetogame.com/g/517/thumbnail.jpg","short_description":"Smilegate's free-to-play multiplayer ARPG is a massive adventure filled with lands waiting to be explored, people waiting to be met, and an ancient evil waiting to be destroyed.", "game_url":"https://www.freetogame.com/open/lost-ark", "genre":"ARPG", "platform":"PC (Windows)", "publisher":"Amazon Games", "developer":"Smilegate RPG", "release_date":"2022-02-11", "freetogame_profile_url":"https://www.freetogame.com/lost-ark"}, {"id":516,"title":"PUBG: BATTLEGROUNDS","thumbnail":"https://www.freetogame.com/g/516/thumbnail.jpg","short_description":"Get into the action in one of the longest running battle royale games PUBG Battlegrounds.", "game_url":"https://www.freetogame.com/open/pubg", "genre":"Shooter", "platform":"PC (Windows)", "publisher":"KRAFTON, Inc.", "developer":"KRAFTON, Inc.", "release_date":"2022-01-12", "freetogame_profile_url":"https://www.freetogame.com/pubg"}, {"id":508,"title":"Enlisted","thumbnail":"https://www.freetogame.com/g/508/thumbnail.jpg","short_description":"Get ready to command your own World War II military squad in Gaijin and Darkflow Software's MMO squad-based shooter Enlisted.", "game_url":"https://www.freetogame.com/open/enlisted", "genre":"Shooter", "platform":"PC (Windows)", "publisher":"Gaijin Entertainment", "developer":"Darkflow Software", "release_date":"2021-04-08", "freetogame_profile_url":"https://www.freetogame.com/enlisted"}, {"id":345,"title":"Forge of Empires","thumbnail":"https://www.freetogame.com/g/345/thumbnail.jpg","short_description":"A free to play 2D browser-based online strategy game, become the leader and raise your city.", "game_url":"https://www.freetogame.com/open/forge-of-empires", "genre":"Strategy", "platform":"Web Browser", "publisher":"InnoGames", "developer":"InnoGames", "release_date":"2012-04-17", "freetogame_profile_url":"https://www.freetogame.com/forge-of-empires"}, {"id":427,"title":"Drakensang Online","thumbnail":"https://www.freetogame.com/g/427/thumbnail.jpg","short_description":"A free to play browser-based top-down hack-and-slash 3D MMORPG similar to games in the Diablo series.", "game_url":"https://www.freetogame.com/open/drakensang-online", "genre":"MMORPG", "platform":"Web Browser", "publisher":"Bigpoint", "developer":"Bigpoint", "release_date":"2011-08-08", "freetogame_profile_url":"https://www.freetogame.com/drakensang-online"}, {"id":525,"title":"MultiVersus","thumbnail":"https://www.freetogame.com/g/525/thumbnail.jpg","short_description":"The Warner Bros lineup meets Smash in Player First Games' MultiVersus.", "game_url":"https://www.freetogame.com/open/multiversus", "genre":"Fighting", "platform":"PC (Windows)", "publisher":"Warner Bros. Games", "developer":"Player First Games", "release_date":"2022-07-19", "freetogame_profile_url":"https://www.freetogame.com/multiversus"}, {"id":475,"title":"Genshin Impact","thumbnail":"https://www.freetogame.com/g/475/thumbnail.jpg","short_description":"If you've been looking for a game to scratch that open-world action RPG itch, one with perhaps a bit of Asian flair, then you're going to want to check out miHoYo's Genshin Impact.", "game_url":"https://www.freetogame.com/open/genshin-impact", "genre":"Action RPG", "platform":"PC (Windows)", "publisher":"miHoYo", "developer":"miHoYo", "release_date":"2020-09-28", "freetogame_profile_url":"https://www.freetogame.com/genshin-impact"}, {"id":523,"title":"Fall Guys","thumbnail":"https://www.freetogame.com/g/523/thumbnail.jpg","short_description":"Play the most competitive massively multiplayer party royale game featuring beans ever for free on a variety of platforms.", "game_url":"https://www.freetogame.com/open/fall-guys", "genre":"Battle Royale", "platform":"PC (Windows)", "publisher":"Mediatonic", "developer":"Mediatonic", "release_date":"2020-08-04", "freetogame_profile_url":"https://www.freetogame.com/fall-guys"}, {"id":340,"title":"Game Of Thrones Winter Is Coming","thumbnail":"https://www.freetogame.com/g/340/thumbnail.jpg","short_description":"A free-to-play browser-based RTS based on the George R.R. Martin novels"}]
```



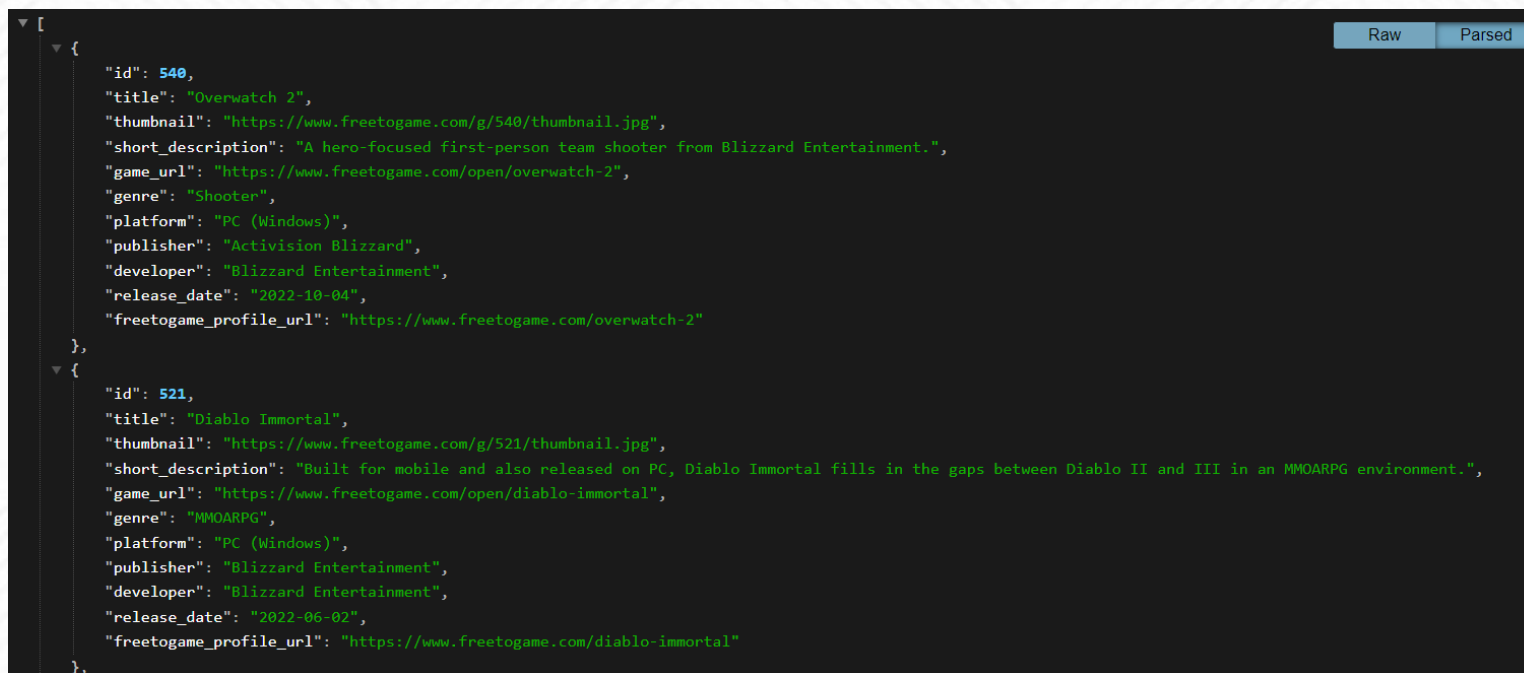
# Consumindo uma API pública



Para melhorar a exibição de arquivos JSON no navegador, podemos instalar a extensão JSON Formatter



Agora os dados serão exibidos formatados dessa maneira:



# Consumindo uma API pública



Como podemos observar a API possui informações de diversos jogos, com informações como “[título](#)”, “[gênero](#)”, “[plataforma](#)” e até uma foto miniatura (“[thumbnail](#)”), exibiremos essas informações em nossa aplicação Flask.

Com a [URL da API](#) em mãos, faremos agora a [integração](#) com nossa aplicação.

No arquivo “[routes.py](#)” importaremos duas novas bibliotecas:

```
import urllib
import json
```

# Consumindo uma API pública



Após isso, criaremos uma nova rota “/apigames” também no arquivo “routes.py” com o seguinte código:

```
@app.route('/apigames', methods=['GET', 'POST'])
def apigames():
    url = 'https://www.freetogame.com/api/games'
    res = urllib.request.urlopen(url)
    data = res.read()
    gamesjson = json.loads(data)
    return render_template('apigames.html', gamesjson=gamesjson)
```

A variável “**url**” conterá o endereço da API.

A variável “**res**” será responsável por armazenar a resposta que vem da API em forma de um objeto.

A variável “**data**” será responsável por ler os dados dessa resposta, por fim, na variável “**gamesjson**” utilizamos o método **json.loads()** que é responsável por transformar um arquivo **JSON** em um formato de **dicionário Python**, tornando assim possível fazer a manipulação desses dados.

# Consumindo uma API pública



Agora, criaremos uma nova view com o nome “**apigames.html**” que terá o seguinte código:

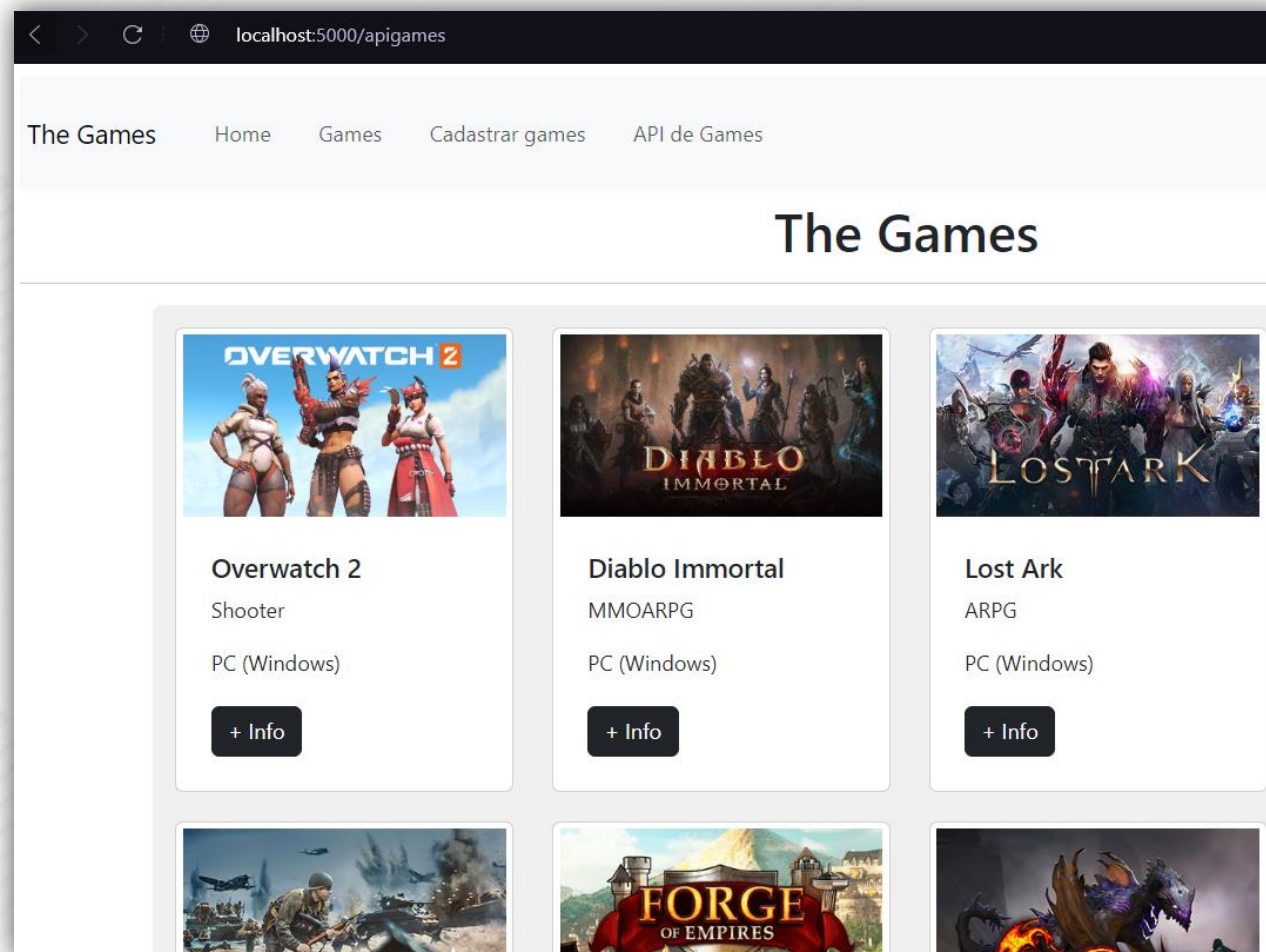
```
{% extends 'base.html' %}
{% block content %}
    <div class="container" style="background-color: #f0f0f0; border-radius: 8px;">
        <div class="row">
            {% for g in gamesjson %}
                <div class="col-md-3 m-auto">
                    <div class="card" style="width: 16rem;">
                        
                        <div class="card-body">
                            <h5>{{g.title}}</h5>
                            <p>{{g.genre}}</p>
                            <p>{{g.platform}}</p>
                            <a href="#" class="btn btn-dark">+ Info</a>
                        </div>
                    </div>
                </div>
            {% endfor %}
        </div>
    </div>
{% endblock content %}
```



# Consumindo uma API pública



Assim, acessando a rota “/apigames” será mostrado a lista de jogos formatada em cards, conforme a seguir:



# Flask

## **Tópico 06.2:** Passando parâmetros para rotas *Integração com APIs*

# Passando parâmetros para rotas



Criaremos uma nova rota abaixo da rota “/apigames” passando um parâmetro “id” que será do tipo inteiro.

Na função “apigames()” passaremos o parâmetro “(id=None)”, se tornando assim parâmetro opcional, caso não seja passado o parâmetro ele receberá o valor de **None (vazio)**.

```
@app.route('/apigames', methods=['GET', 'POST'])
@app.route('/apigames/<int:id>', methods=['GET', 'POST'])
def apigames(id=None):
    url = 'https://www.freetogame.com/api/games'
    res = urllib.request.urlopen(url)
    data = res.read()
    gamesjson = json.loads(data)
```

Depois disso, logo após a linha que carrega os dados da API na variável “gamesjson”, criaremos uma condição para verificar quando o parâmetro for passado.

Caso não seja passado nenhum parâmetro de id na rota, a função irá renderizar a página “apigames.html” com todos os games.

# Passando parâmetros para rotas



```
data = res.read()
gamesjson = json.loads(data)
if id:
    ginfo = []
    for g in gamesjson:
        if g['id'] == id:
            ginfo=g
            break
    if ginfo:
        return render_template('gameinfo.html', ginfo=ginfo)
    else:
        return f'Game com a ID {id} não foi encontrado.'
else:
    return render_template('apigames.html', gamesjson=gamesjson)
```

Agora, quando for passado um parâmetro de “**id**” para a rota, será criada uma nova lista “**ginfo[]**” vazia. Após isso, através de um **for**, percorreremos o **json da API** até que a id que foi passada como parâmetro seja encontrada.

Caso seja encontrado o registro procurado ele será gravado na lista “**ginfo[]**”. Caso seja informado uma id inválida será retornado a mensagem “*Game com a ID {id} não foi encontrado*”.

Por fim, caso a lista “**ginfo[]**” tenha algum registro gravado será renderizada a página “**gameinfo.html**” (que ainda iremos criar), passando a lista “**ginfo[]**” que terá as informações do game com a id procurada.



# Passando parâmetros para rotas



Agora criaremos a página “**gameinfo.html**” que será responsável por mostrar as informações de um game individual. Essa página deve conter o seguinte código:

```
{% extends 'base.html' %}
{% block content %}
    <div class="container" style="background-color: #f0f0f0; border-radius: 8px;">
        <div class="row">
            <div class="col-md-6 m-auto">
                <a href="{{url_for('apigames')}}" class="btn btn-dark"><< Todos os jogos</a><br><br>
                <div class="card">
                    
                    <div class="card-body m-auto">
                        <h3>{{ginfo.title}}</h3><br>
                        <p>Gênero: {{ginfo.genre}}</p>
                        <p>Descrição: {{ginfo.short_description}}</p>
                        <p>Plataforma: {{ginfo.platform}}</p>
                        <p>Editora: {{ginfo.publisher}}</p>
                        <p>Data de lançamento: {{ginfo.release_date}}</p>
                        <a href="{{ginfo.freetogame_profile_url}}" class="btn btn-danger">Jogar agora!</a>
                    </div>
                </div>
            </div>
        </div>
    </div>
{% endblock content %}
```

# Passando parâmetros para rotas



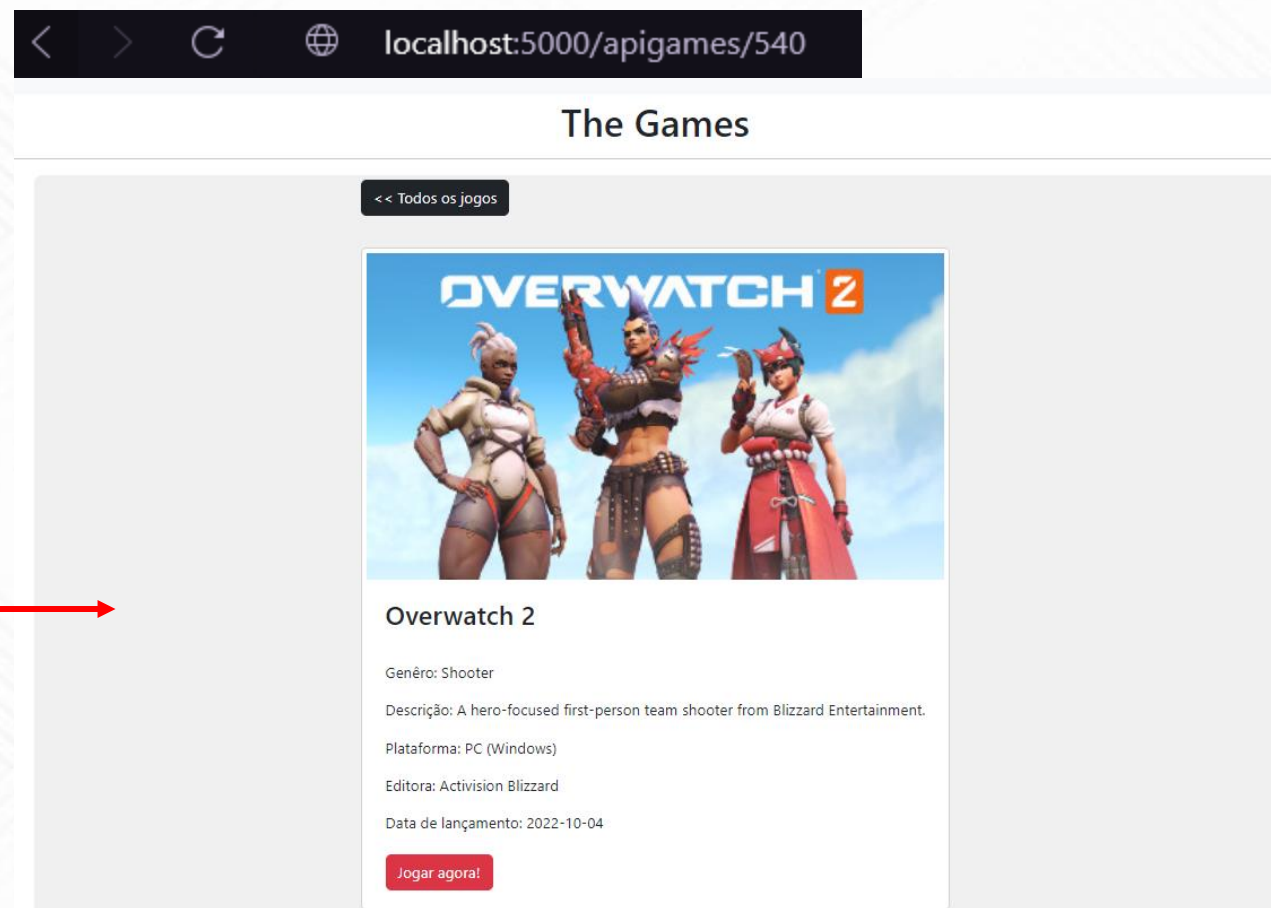
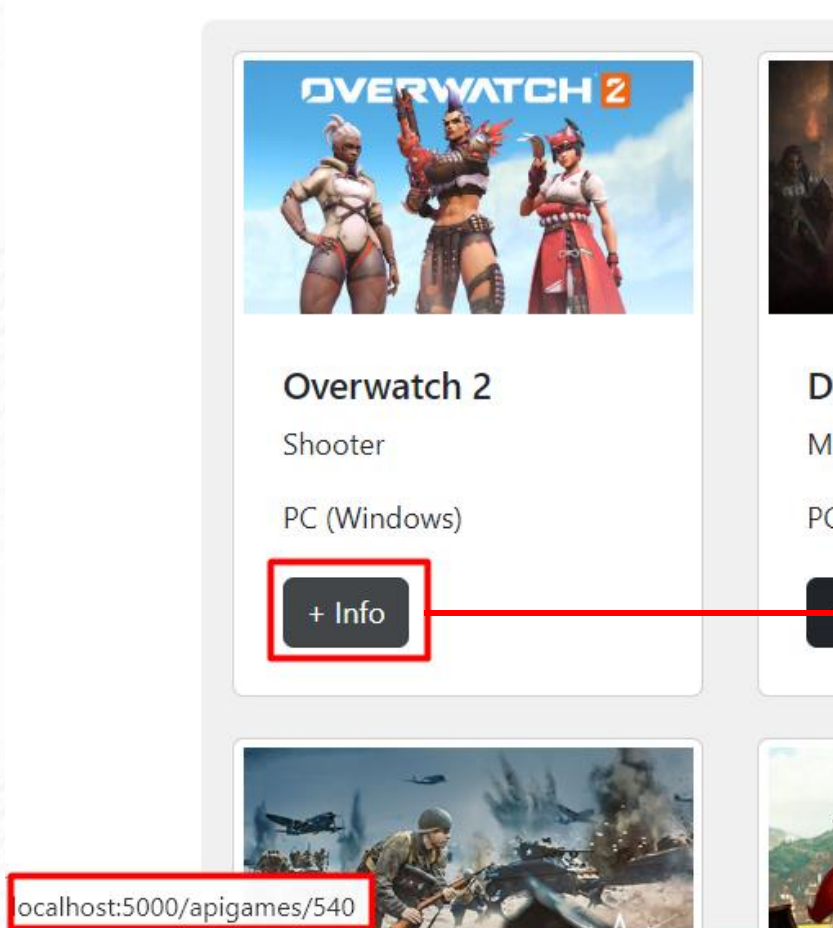
Por fim, na página “**apigames.html**” incluiremos um link no botão “**+Info**” com as ids de cada jogo mostrado na tela.

```
<div class="card-body">
    <h5>{{g.title}}</h5>
    <p>{{g.genre}}</p>
    <p>{{g.platform}}</p>
    <a href="{{url_for('apigames', id=g.id)}}" class="btn btn-dark">+ Info</a>
</div>
```

# Passando parâmetros para rotas



Agora ao clicar no botão “+ Info” de cada game, o usuário será redirecionado para uma página com mais informações sobre aquele jogo.





**Flask**

Flask

## *Programação Web III*

*Prof. Diego Max*