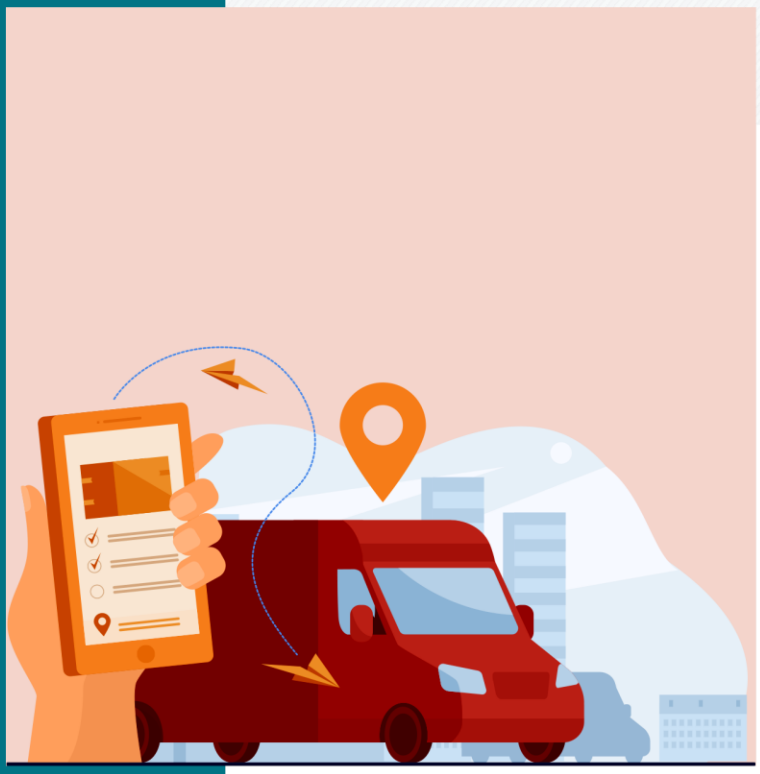
AGENDA 4

CONSUMINDO

API E SERVIÇOS WEB



GEEaD - Grupo de Estudos de Educação a DistânciaCentro de Educação Tecnológica Paula Souza

GOVERNO DO ESTADO DE SÃO PAULO
EIXO TECNOLÓGICO DE INFORMAÇÃO E COMUNICAÇÃOCURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS PROGRAMAÇÃO MOBILE I

Expediente

Autor:

GUILHERME HENRIQUE GIROLLI

Atualização Técnica:

Rogério Galdiano de Freitas

Revisão Técnica:

Eliana Cristina Nogueira Barion

Revisão Gramatical:

Juçara Maria Montenegro Simonsen Santos

Editoração e Diagramação:

Flávio Biazim

São Paulo – SP, 2021

MERGULHANDO NO TEMA...

Application Programming Interface (API)

A *Application Programming Interface*, ou na língua portuguesa Interface de Programação de Aplicações,ou ainda como é mais conhecida pela sigla API, é um conjunto de padrões previamente desenvolvidos para gerar a espécie de uma "ponte" entre dois sistemas ou mais, sejam eles produzidos na mesma ou em linguagens diferentes, permitindo assim a interação e troca de informações entre eles.

O desenvolvimento de API surgiu com a necessidade de troca de informações entre instituições e serviços, desta forma essa comunicação tinha a necessidade de uma alta segurança, não bastando apenas que uma determinada instituição fornecesse acesso ao seu banco de dados, ela precisava fornecer acesso apenas a um determinado conjunto de informações, não permitindo acesso geral.

Voltando ao exemplo dos Correios, imaginem os problemas que poderiam surgir se a empresa liberar acesso ao seu banco de dados para os usuários consultarem informações sobre objetos postados. O nível de segurança cai parazero, e isso pode trazer problemas incalculáveis para a empresa.

Desta forma, os Correios desenvolveram uma API, ou seja, uma ferramenta on-line que oferece um acesso limitado as informações, o usuário não possui acesso direto ao banco de dados.

Garantindo assim que apenas as informações pertinentes a determinada consulta seja entregue ao usuário e que ele não consiga acesso a outras informações.

Encontramos várias formas de API's, e cada uma delas se enquadra na utilização de recursos entre ferramentas, veja a seguir os tipos encontrado.

Dynamic-link library (DLL): São classes criadas em linguagens de programação e distribuídas para que terceiros utilizem recursos sem a necessidade de perder tempo reescrevendo ou até mesmo recriando códigos. Podemos exemplificar falando de uma DLL que converte vários formatos de áudio para o formato MP3. Essa DLL pode ser utilizada por um sistema de gravação e edição de músicas de um desenvolvedor "A". E pode ser utilizada por um desenvolvedor "B" noseu projeto de gravação e edição de vídeos. Tanto o desenvolvedor "A" quanto o desenvolvedor "B" não criaram os códigos dessa DLL, e seus sistemas utilizamos recursos oferecidos por essas classes, facilitando odesenvolvimento dos programas.

Figura 1 - Tela do programa Audacity, solicitando a localização da DLL responsável por criar arquivos MP3.

Plugins: São encontradas na WEB e responsáveis por alguns serviços utilizados atualmente como certificação de login e senha, animação de interfaces, entre outros. Um exemplo é o "Captcha" que é responsável por atribuir uma segurança em páginas WEB, para que certos serviços não sejam executados por Robôs.

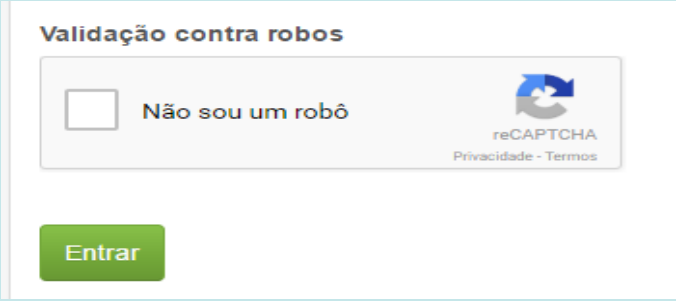


Figura 2 - Utilização de "Captcha" em site.

Frameworks: São amplamente utilizados na WEB atual, e permitem recursos e envolvimentos visuais para sites alguns exemplos são o JQuery e Bootstrap. Como exemplo o Bootstrap permite o desenvolvimento profissional de sites com um alto nível de interface gráfica e de maneira rápida, com construção de formulários, botões, menus, entre outros componentes que valorizam o projeto.

Figura 3 - Frameworks

WebAPI: Essa última e não menos importante e será o tema da nossa agenda. São API's disponíveis na WEB e que oferecem a interação entre ferramentas. São utilizadas por meio de serviços disponíveis em sites/servidores de conteúdo e para o seu funcionamento é necessário seguir alguns padrões de criação e comunicação utilizandoprotocolos de comunicação e códigos XML, JSON, entre outros.

Na Figura 4 encontramos um exemplo de API WEB onde é retornado à consulta de um CEP. Esta API é disponibilizada gratuitamente pelo site Cep Aberto, disponível em: https://cepaberto.com/. Essa API é responsável por receber um determinado CEP por meio da sua URL, e retornar um arquivo JSON com as informações do CEP consultado.

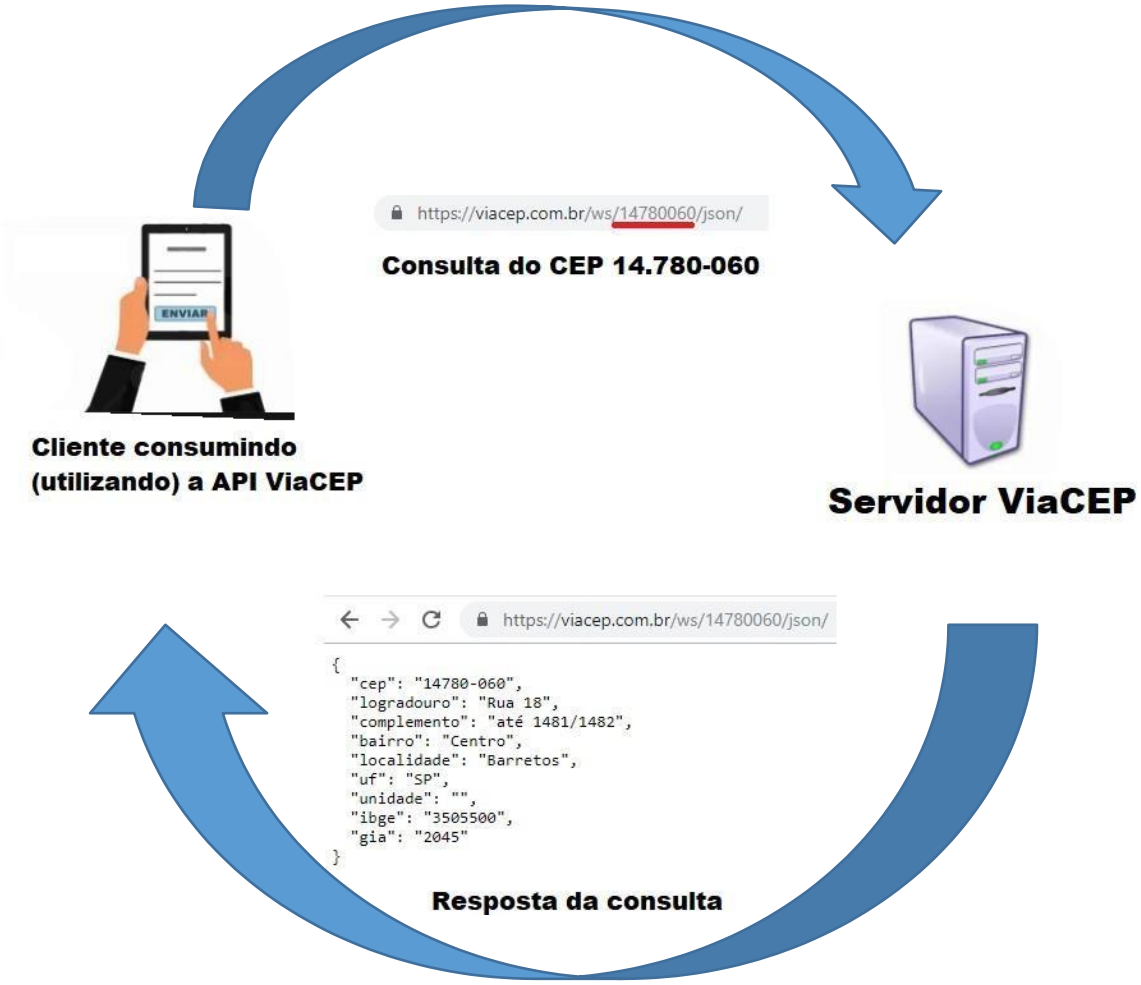


Figura 4 - Consulta de CEP no site cepaberto

Na imagem anterior encontramos o retorno da consulta de maneira visual por meio do navegador, pois ele é responsável por interpretar os códigos JSON e mostrar na tela do usuário.

Vamos conhecer um pouco mais sobre esses arquivos de retorno utilizados pela grande maioria dos serviços WEB, abordando os dois principais, XML e o JSON.

Extensible Markup Language (XML)

O XML é um arquivo de marcação muito utilizado na Internet e em serviços WEB, foi desenvolvido pela W3C com o propósito de organizar textos, banco de dados, desenhos, etc. Essa organização ocorre hierarquicamente de maneira extensível, ou seja, os marcadores são customizáveis, garantindo a sua utilização nos mais diversos cenários.

O modelo XML ficou popular no Brasil, quando seus padrões foram implementados na nota fiscal eletrônica e seus arquivos eram utilizados no processo de comunicação entre as empresas e os servidores da receita para o processo de validação e conhecimento de uma nota fiscal.

O exemplo apresentado na Figura 5, guarda dados sobre clientes, na forma de um arquivo XML. A imagem mostra que os marcadores são destacados pela utilização das tag's "<>" e todo marcador aberto, tem a necessidade de ser fechado.

O padrão de arquivo XML ainda é amplamente utilizado em todo mundo, perdendo lugar aos poucos para o padrão JSON que é uma nova forma de arquivos com marcações para trânsito de informações.

```
<?xml version="1.0"?>
<cliente>
        <nome>João</nome>
        <telefone>1733253325</telefone>
        <endereco>Rua 10 nº 20</endereco>
</cliente>
<cliente>
        <nome>Maria</nome>
        <telefone>1633223322</telefone>
        <endereco>Rua 30 nº 40</endereco>
</cliente>
<cliente>
        <nome>José</nome>
        <telefone>1433223366</telefone>
        <endereco>Rua 50 nº 60</endereco>
</cliente>
```

Figura 5 - Arquivo XML com dados de 3 clientes.

JavaScript Object Notation (JSON)

O cenário de desenvolvimento WEB atualmente conta com uma popularidade na utilização da linguagem JavaScript. Essa tendência fez com que naturalmente os programadores efetuem uma migração na utilização dos sistemas de comunicação, passando do XML para o JSON. Desta forma, a cada dia o JSON ganha mais território no mundo da programação.

Por ser um objeto proveniente do JavaScript, o JSON é automaticamente mais simples de ser utilizado na linguagem quando comparado ao XML. Pesquisas apontam que a utilização do JSON nos processos de troca de informações, são mais eficientes, uma vez que ele é mais leve que o XML.

```
[
        {"nome":"João",  "telefone":"1733253325","endereco":"Rua  10 nº 20"},
        {"nome":"Maria",  "telefone":"1633223322","endereco":"Rua  30 nº 40"},
        {"nome":"José",  "telefone":"1433223366","endereco":"Rua  50 nº 60"}
]
```

Figura 6 - Arquivo JSON com dados de 3 clientes.

Na Figura 6 é apresentado um Array (composto pelas tag's "[]") de objetos (composto pelas tag's"{}"), e dentro dos objetos tem os dados com valores, como por exemplo, "nome":"João".

Agora que conhecemos as formas de transferência de informações pelos serviços WEB, vamos verificarcomo ocorre um consumo de serviço WEB.

Consumindo Serviços WEB no navegador

Vamos analisar um exemplo prático...

Várias empresas estão utilizando serviços WEB, e Serginho escolheu uma dessas empresas para utilizarem seu projeto. O site ViaCEP, proporciona uma consulta gratuita de CEP por meio de uma API. Ao consultar essa API, ela retorna um arquivo JSON com as informações do CEP. Na Figura 7, é mostrado a URL para utilizar a API, onde nesta URL é transmitido o CEP que será consultado.

```
import requests

url = "https://www.cepaberto.com/api/v3/address"
# O seu token está visível apenas pra você
headers = {'Authorization': 'Token token=128f297900b368f6ac2158e0f074bb72'}
params = {'estado': 'SP', 'cidade': 'Ubatuba'}
response = requests.get(url, headers=headers, params=params)

print(response.json())
```

Figura 7 - URL de consulta a API cepaberto, buscando informações sobre o CEP

A Figura 8, apresenta o conteúdo do arquivo JSON com a resposta do CEP, exibida pelo navegador WEB que utilizamos na consulta.

```
{'latitude': '-23.4336578',
 'longitude': '-45.0838481',
 'estado': {'sigla': 'SP'},
 'cep': '11680000',
 'cidade': {'nome': 'Ubatuba', 'ddd': 12, 'ibge': '3555406'},
'altitude': 4.8
}
```

Figura 8 - Resposta da consulta a API cepaberto, buscando informações sobre o CEP: 14780-060

Este processo é o que leva o nome de consumir um serviço WEB, agora temos que estipular alguns padrões, para utilizar em nosso aplicativo.

VOCÊ NO COMANDO

Consumindo Serviços WEB no aplicativo

Você já imaginou construir um aplicativo para consumir um serviço WEB?

Não? Então vamos seguir juntos nessa jornada com nosso desenvolvedor Serginho.

Para consumir um serviço WEB em um aplicativo, é necessário criar uma estrutura de componentes para conectar ao serviço e receber o objeto JSON. E por fim, temos que criar um layout para efetuar a interface gráfica com o usuário, ou seja, receber o CEP e mostrar as informações para o usuário.

Vamos criar um projeto chamado “APICep”, verifique a seguir o processo de criação do projeto.

- Abra a plataforma de desenvolvimento do Kodular: https://www.kodular.io/creator
- Clique no botão **Create Project**

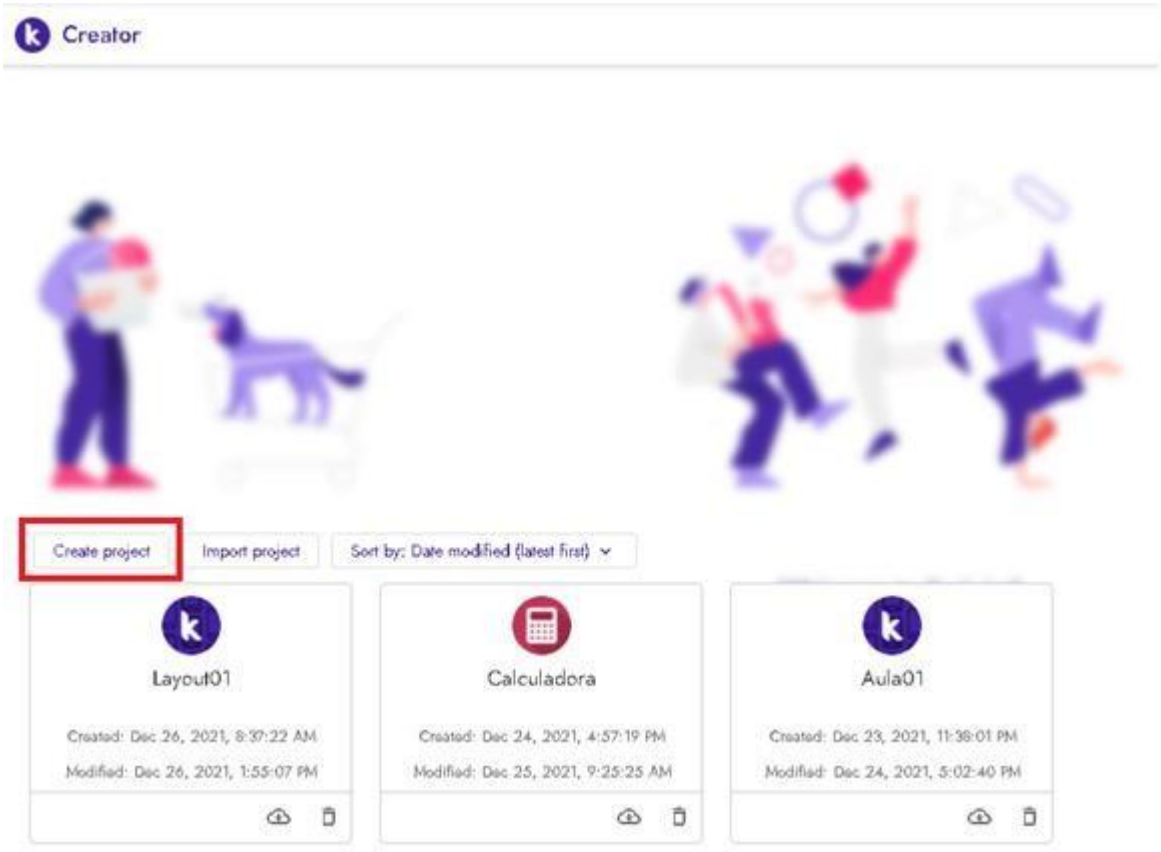


Figura 09 - Criando um novo projeto

- Digite o nome **Compartilhar** e clique no botão **Next**.

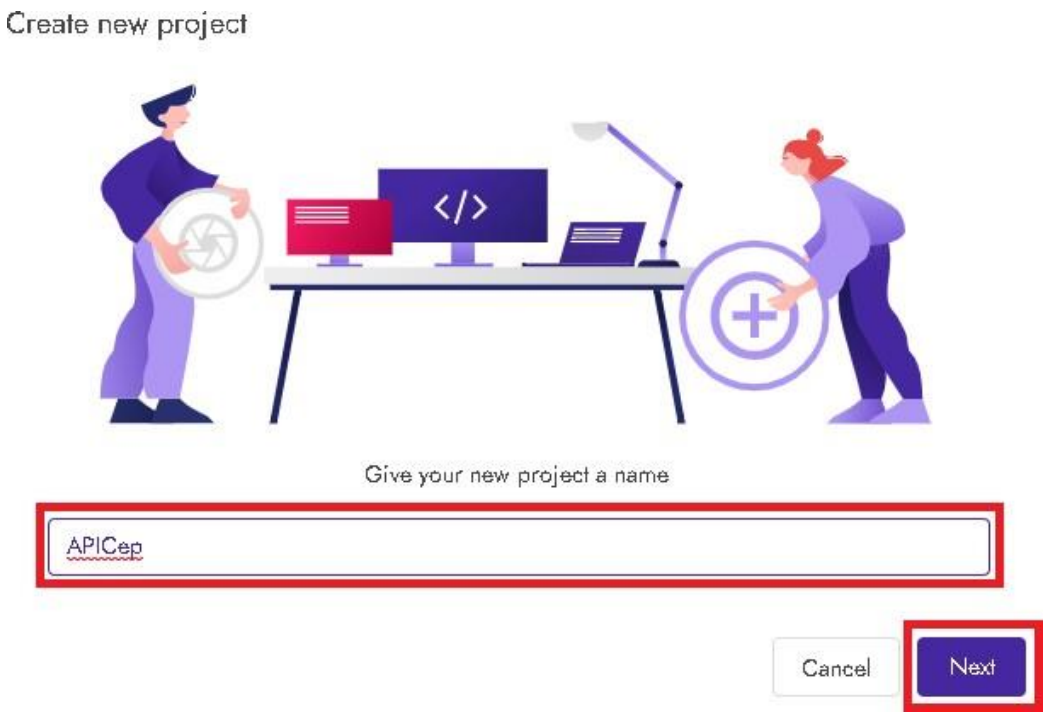
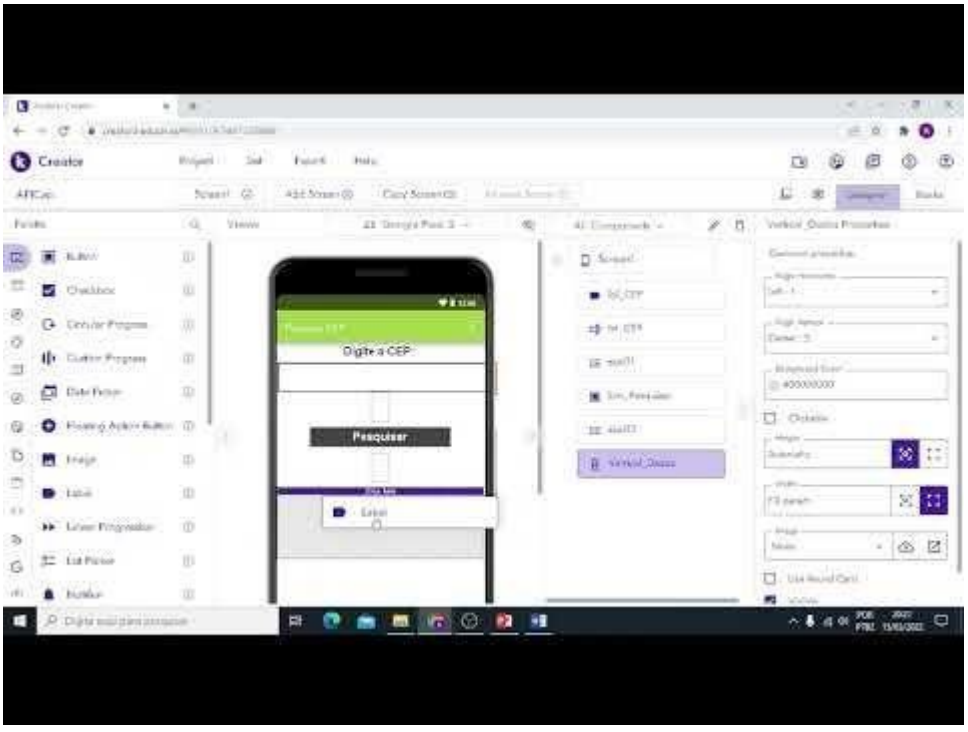


Figura 10 - Criando o projeto APICep.

- Clique no botão **Finish** para finalizar a criação do novo projeto.

Neste momento, crie a interface da aplicação através do vídeo **Agenda 04 - Criando a interface consumindo API**, disponível em : https://youtu.be/-3lgRiaPg9Q



- Altere para o cenário de construção de linguagem de programação em blocos.

Figura 11 - Alterando para o cenário de construção de blocos.

Vamos iniciar construindo o bloco **Btn_Pesquisar** que será responsável pela conexão com o site **cep aberto**, para pesquisar o nome da cidade e o estado através do número de cep digitado no aplicativo. Sendo que para efetuar a conexão será necessário especificar o **token** de autorização. Cuja função será permitir o acesso as informações do site.

Token de autorização: **Token token=128f297900b368f6ac2158e0f074bb72**

Url do site para pesquisa: **https://www.cepaberto.com/api/v3/cep?cep=**

Ao clicar no botão **btn_Pesquisar** será acionado a busca de dados no respectivo site, sendo que as informações deverão ser tratadas de forma que o aplicativo possa responder somente a Cidade e o Estado do cep pesquisado. Portanto iremos utilizar o bloco abaixo para mostrar as informações.

Figura 12 - Construindo o bloco btn_Pesquisar.

- Clique no objeto **Web_Dados** e arraste o componente **when Web_Dados.Got Text**

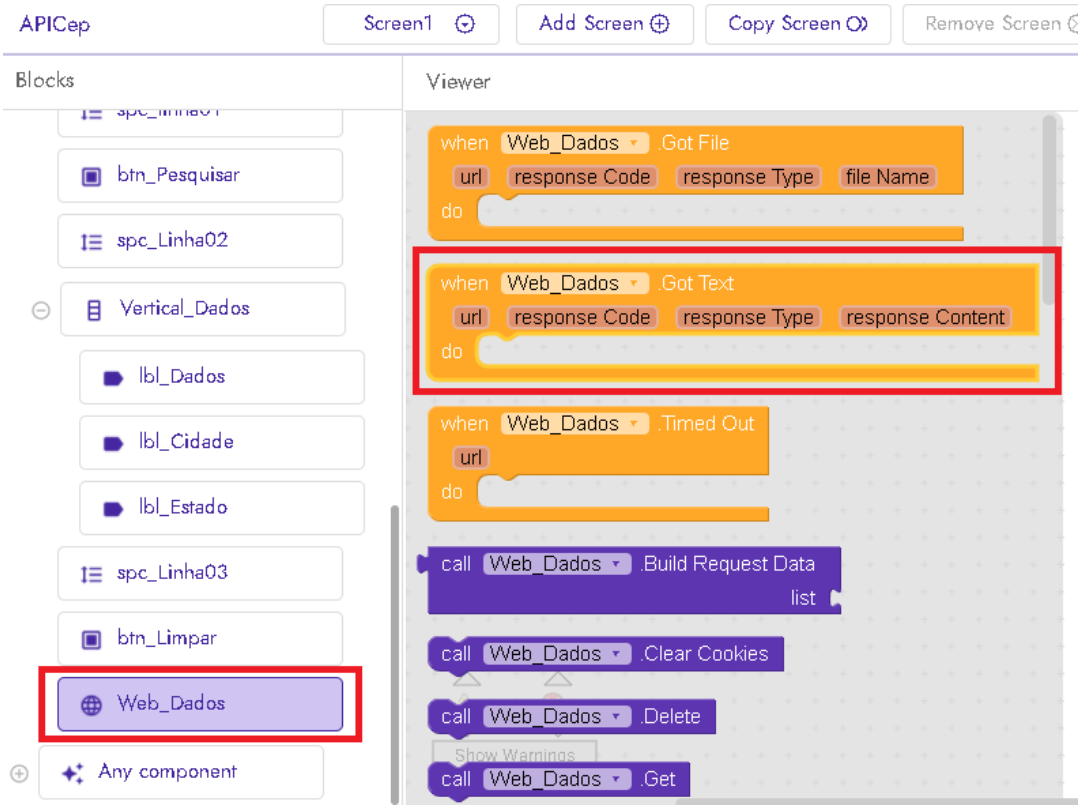


Figura 13 - Inserindo o componente when web_dados.Got Text.

Para a construção do bloco da **Figura 14**, o aluno deverá clicar no construtor **Dictionaries** e clicar no objeto **get value for key.**

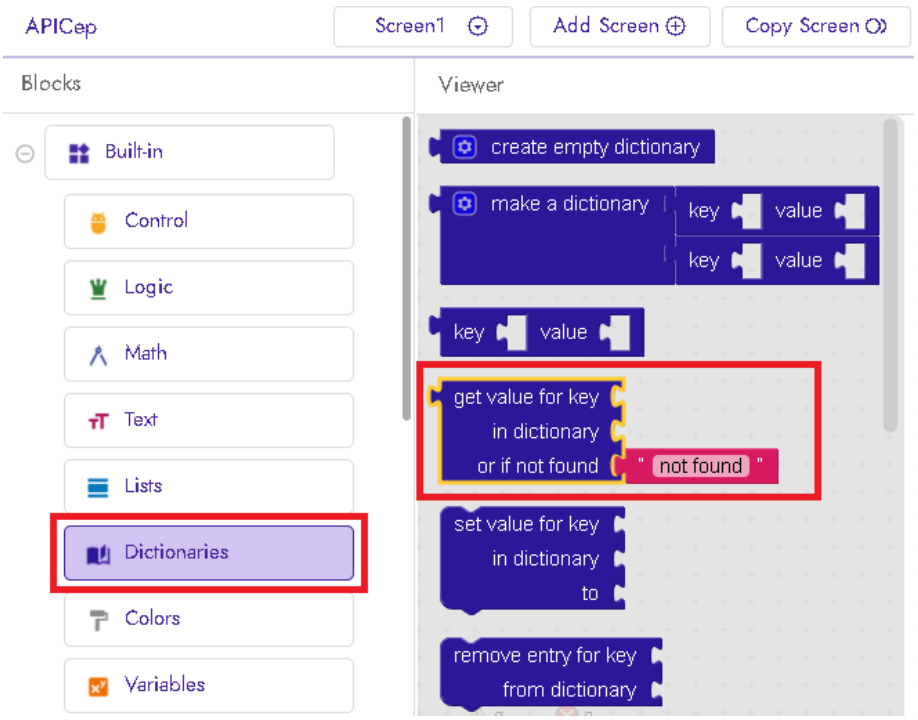


Figura 14 - Construtor Dictionaries, objeto get value for key..

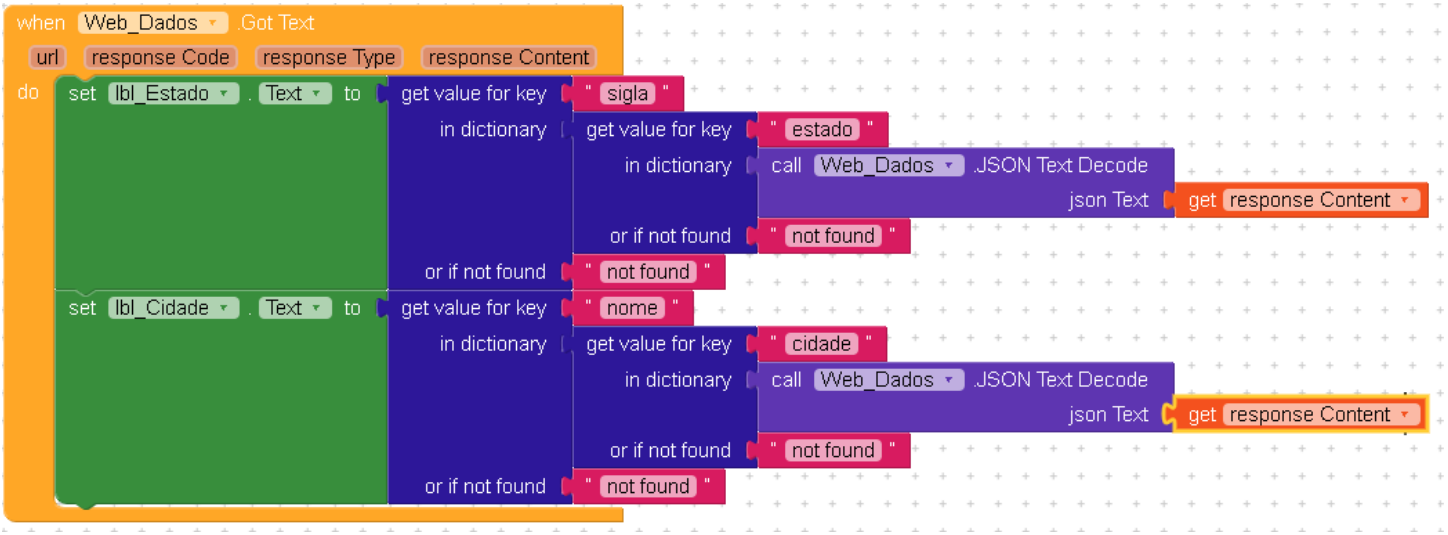
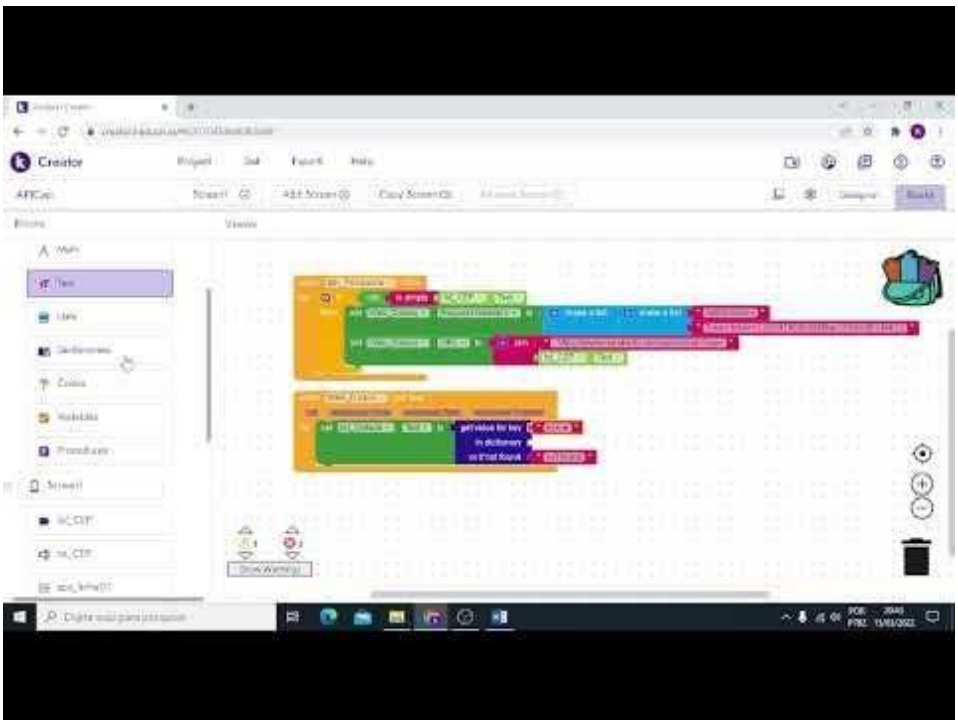


Figura 15 - Construindo o bloco when Web_Dados. Got Text.

Por fim, vamos construir o bloco de código do botão **Limpar**.

Figura 16 - Código de bloco - Botão Limpar.

Agenda 04 - Criando a programação de Blocos Consumindo API , disponível em: https://youtu.be/FvAG3KdTlxA



Para finalizar o projeto, exporte o **arquivo APK** para o dispositivo móvel e realizar a instalação através do aplicativo **Kodular Companion.**

- Clique no menu **Export**, na opção Android App (.apk).

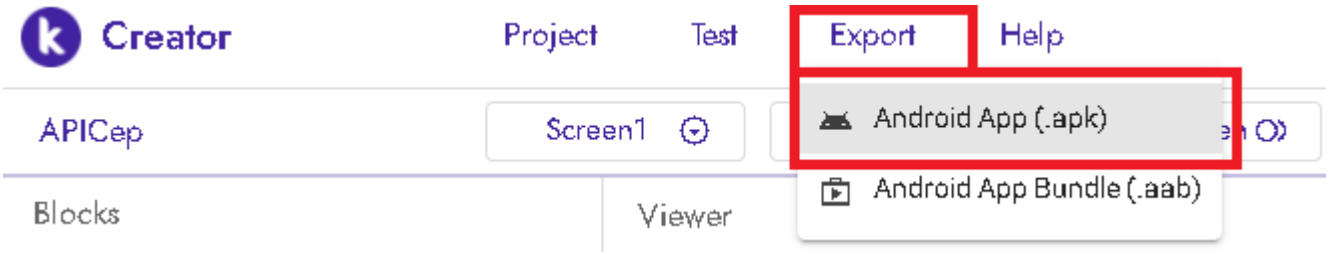


Figura 4 - Menu Export, opção Android App (.apk)

Utilize o aplicativo **Kodular Companion** para escanear o qrcode e siga todos os passos para a instalação do aplicativo, de acordo com material anterior.

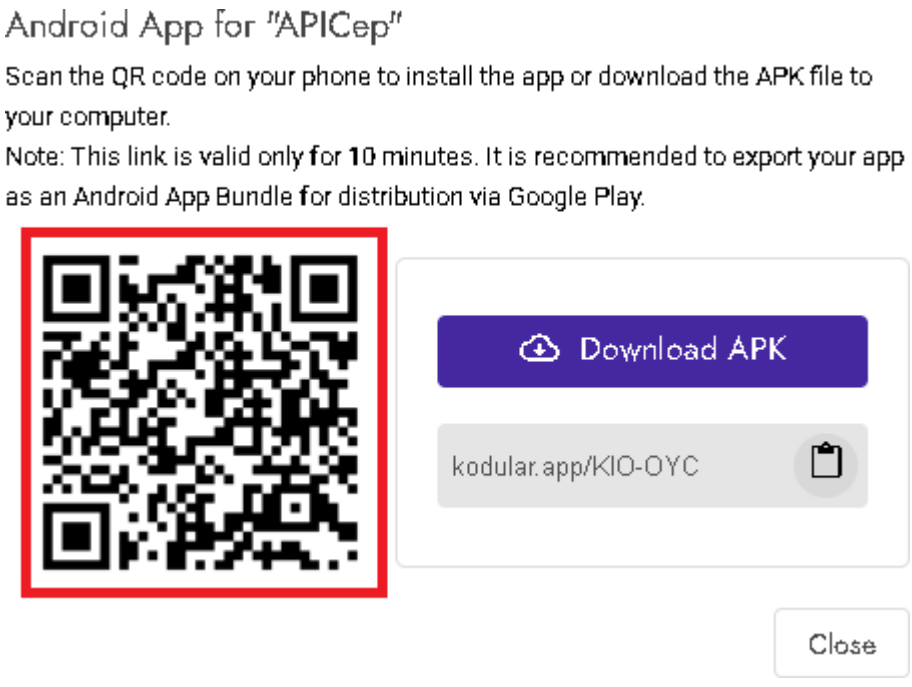


Figura 185 - Disponibilizando o Qrcode.

Ao iniciar o aplicativo, o usuário irá digitar o número do cep desejado e clicar no botão **Pesquisar**. O aplicativo irá efetuar a pesquisa e retornar o nome da cidade e o respectivo estado.

Figura 19 – Consultando o CEP desejado.

Baixe aqui o Projeto.