

Tesis de Investigación.

**Implementación de una API para el control de datos de los
perros en la veterinaria X**

Autor: Silvia Hernández Márquez.

Facultad de Ingenierías , Universidad Americana de Puebla

Docente: Omar Diaz García.

Teziutlán, Puebla, 12 de agosto de 2022

Índice

| | |
|---|-----------|
| Introducción | 3 |
| Planteamiento Del Problema | 5 |
| Justificación | 7 |
| Objetivo..... | 9 |
| Diseño De La API..... | 10 |
| Implementación De La API..... | 22 |
| Resultados De La API..... | 23 |
| Evaluación De La Calidad De La API | 25 |
| Cliente | 25 |
| Necesidad | 25 |
| Herramienta Para Implementar | 25 |
| Recolección De Datos..... | 25 |
| Evaluación | 27 |
| Producto Terminado..... | 30 |
| Conclusiones | 33 |
| Recomendaciones | 34 |
| Bibliografía | 35 |

Introducción

La presente tesis de investigación tiene como titulo Implementación de una API para el control de datos de los perros en la veterinaria X, localizada en Teziutlán, Puebla, la cual abarca la problemática que surge a partir de que los dueños de los perros a los llevan a consultas a la veterinaria X olvidan los días que los tienen que llevar.

Por lo cual a este problema se le busco la solución de implementa una API en la cual los dueños puedan revisar los datos de su mascota, desde el día de su próxima consulta hasta las vacunas que tiene. Asi ellos puedan llevar a sus perros a su consulta en tiempo y forma. La API se creo con la idea que los dueños puedan revisar la información de estos de forma rápida y sencilla.

En el primer capítulo de esta tesis de investigación es el planteamiento del problema que se detecto donde se describe mas a profundidad el porque de este.

El segundo capítulo es la justificación del porque es importante implementar la API para dar solución a la problemática.

En el tercer capitulo se encuentra el objetivo de esta tesis de investigación.

En el cuarto capitulo denominado diseño de la API se encuentra como elaboramos la API para dar solución a esta problemática, en este se explica los pasos que debes de seguir para poder desarrollarla y utilizarla.

En el quinto capitulo se encuentra como fue la implementación de la API, en la cual se explica donde se encuentra esta, para poder hacer uso de esta.

En el sexto capítulo están los resultados de la API, en este se explica si tuvo una buena difusión y como fue esta.

En el séptimo capítulo la evaluación de la calidad de la API en la cual puedes encontrar desde a que clientes esta dirigida dicha API, hasta el producto terminado de esta API.

Además de que más adelante podrás encontrar desde las conclusiones de este trabajo, las recomendaciones que le hicieron y las referencias bibliográficas que ocupo.

Planteamiento Del Problema

Los dueños de mascotas en Teziutlán, Puebla; olvidan llevar a sus consultas en la veterinaria X a sus perros por falta de recordar cuando les toca su cita.

La salud de las mascotas siempre ha sido un tema importante para los dueños de estas, ya que los consideran parte de la familia, por lo que siempre quieren que tengan una buena salud para que den alegría a la familia.

Sin embargo, en Teziutlán, Puebla; en la veterinaria X se ha visto que en los últimos meses los dueños de mascotas en este caso perros, se les olvidan las consultas que tienen programadas para llevar a sus mascotas al veterinario, lo que causa que muchos de los perros no tengan todas sus vacunas o que su enfermedad tarde más tiempo en curarse de lo que se tenían programado. Lo que origina que los perros enfermen frecuentemente por la falta de un chequeo constante y esto puede causar la muerte en muchos de ellos por la falta de atención por parte de sus dueños.

Se ha demostrado que la falta de atención a las fechas de las consultas hace que los perros tengan más probabilidad de enfermarse y morir por la falta de cuidados adecuados o por el retraso en el chequeo.

Una condición necesaria para lograr la mejora de la salud de los perros es que los dueños recuerden cuando les toca consulta a estos o tengan algo donde puedan checar el control de su mascota.

Por lo anterior expuesto, se plantea diseñar una API en la cual los dueños de los perros puedan verificar cuando les toca cita o cuando fue la última a la que acudieron. Además de que

en esta podrán ver el control que llevan sus mascotas en la veterinaria X desde las vacunas que tienen y las que les faltan.

Justificación

Diseñar una API que guarda los registro clínicos para el control de las consultas en los perros.

La salud de las mascotas siempre ha sido un tema importante para los dueños de estas, pues procuran que estén en excelentes condiciones, en este caso tomamos los datos de los perros para hacer esta API (interfaz de programación de aplicaciones).

Por lo consiguiente, en este proyecto se va a realizar una API, la cual menciona que: “Las API son mecanismos que permiten a dos componentes de software comunicarse entre sí, mediante un conjunto de definiciones y protocolos” (Amazon Web Services, 2022).

En este caso la finalidad de esta API es ver cuando le toca a tu mascota su próxima consulta y cuando fue la última.

Del mismo modo podremos checar las enfermedades que tiene y cuando le pusieron su última vacuna o si le falta alguna, incluso ver los datos del dueño, la relación que tiene con la mascota, y toda la información de su mascota.

Esta API mantiene un impacto en los propietarios de los perros, ya que les sirve para llevar un control de las citas de sus mascotas y ver que vacuna le falta que le pongan, con esto llevan un control adecuado de la salud de su perro.

La importancia de nuestro proyecto reside en mejorar el control de las consultas de los perros, para esto es necesario guardar los registros clínicos a fin de que los dueños puedan checar los datos en caso de estar preocupados si les falta alguna vacuna o si tiene

alguna enfermedad su mascota. Esto nos permitirá llevar un mejor control de cuando les toca cita a las mascotas.

Objetivo

Diseñar una API que guarde los registros clínicos para el control de las consultas de los perros que llevan a la veterinaria X durante el año 2022 en Teziutlán, Puebla.

Diseño De La API

Para empezar, daremos algunas definiciones de conceptos que se verán más adelante a fin de una mejor comprensión.

Script: (“es un documento que contiene instrucciones, escritas en códigos de programación”). (Significados.com, s.f.)

Atributos: (“son las características individuales que diferencian un objeto de otro y determinan su apariencia, estado u otras cualidades”). (Conceptos básicos de la Programación Orientada a Objetos, s.f.)

Nomenclatura Camello: (“es un estilo de escritura que se aplica a frases o palabras compuestas”). (CamelCase, s.f.)

Programa: (“es un conjunto de pasos lógicos escritos en un lenguaje de programación que nos permite realizar una tarea específica”). (Tecnología Inclusiva, s.f.)

Class: (“es una plantilla para el objetivo de la creación de objetos de datos según un modelo predefinido”). (Kiko Palomares, s.f.)

Fields: (“es una variable de cualquier tipo que se declara directamente en una clase o struct”). (Bill Wagner, 2022)

Rest_framework: (“es un framework que nos permite el fácil desarrollo de una API REST en Python”). (Django API REST, s.f.)

Django.urls: (“es un mapeo entre los patrones URL y las funciones de vista que deben ser llamadas por esos patrones URL”). (uniwebsidad, s.f.)

Una vez conociendo estas definiciones podemos avanzar a nuestro primer paso para crear la API.

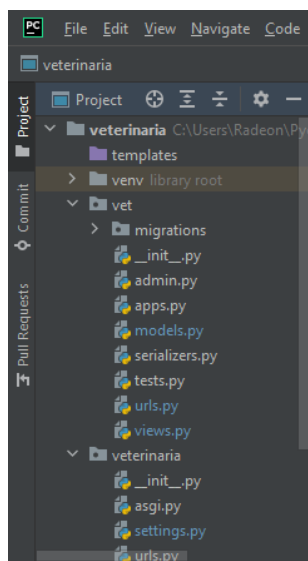
Para crear esta API primero se elige un tema de interés, se busca una base de datos relacionada con el tema que se eligió, se ven los elementos que tiene que llevar para poder escribir el script.

Se elige un lenguaje con el cual vamos a escribir el script, en este caso tomamos el lenguaje de Python para anotar las órdenes que queremos que salgan en el proyecto y elegimos el programa de Pycharm, un Entorno de Desarrollo Integrado (IDE), el cual soporta este lenguaje y creamos un proyecto llamado veterinaria.

Empezando por crear ciertas carpetas las cuales van a contener el script de lo que observamos en la base de datos de la cual nos estamos guiando.

Figura 1

Creación de carpetas en Pycharm



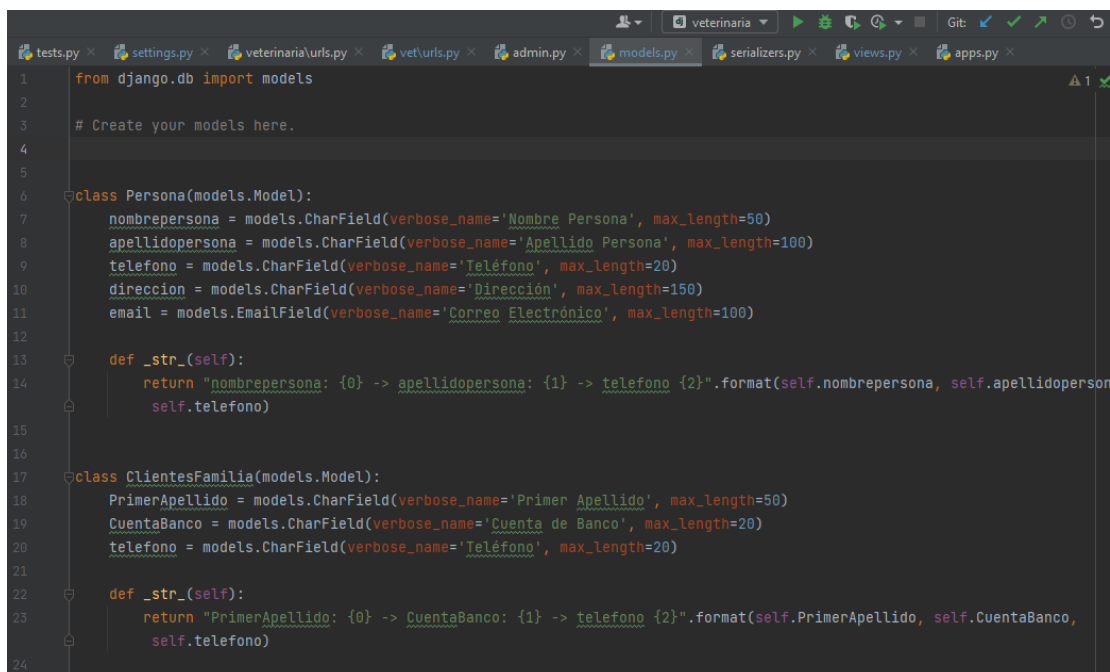
Nota: Creación de carpetas en Pycharm, por Silvia Hernández Márquez, 2022.

Primero se crea una carpeta la cual llevara el nombre de vet y va a contener los documentos, el primer documento que se encuentra en ella lleva por nombre models.py donde dentro de este vamos a crear las clases en las cuales vamos a meter todos los atributos que tenía la base de datos en la cual nos vamos guiando.

Empezando por la clase persona y sus atributos que pusimos fueron nombrepersona, apellidopersona, teléfono, dirección, email en estos se ponen de qué tipo de texto son si son charfiel o dataFiel, terminando esta primera clase se sigue con la segunda clase que es ClientesFamilia en la cual también se ponen sus atributos que eran PrimerApellido, CuentaBanco y teléfono y al igual que la primera especifique el tipo de texto que son, después hice la tercera clase que es la RelacionPersonasClientes y sus atributos que son IDClientesFamilia y IDPersona los cuales están relacionados con las clases anteriores lo que hace que esta solo traiga los atributos de los registros anteriores, por eso se ocupa un ForeignKey que es un tipo de llave que hace lo antes mencionado.

Figura 2

Creación de los modelos que va a llevar nuestra API



```

1  from django.db import models
2
3  # Create your models here.
4
5
6  class Persona(models.Model):
7      nombrepersona = models.CharField(verbose_name='Nombre Persona', max_length=50)
8      apellidopersona = models.CharField(verbose_name='Apellido Persona', max_length=100)
9      telefono = models.CharField(verbose_name='Teléfono', max_length=20)
10     direccion = models.CharField(verbose_name='Dirección', max_length=150)
11     email = models.EmailField(verbose_name='Correo Electrónico', max_length=100)
12
13     def __str__(self):
14         return "nombrepersona: {0} -> apellidopersona: {1} -> telefono {2}".format(self.nombrepersona, self.apellidopersona, self.telefono)
15
16
17     class ClientesFamilia(models.Model):
18         PrimerApellido = models.CharField(verbose_name='Primer Apellido', max_length=50)
19         CuentaBanco = models.CharField(verbose_name='Cuenta de Banco', max_length=20)
20         telefono = models.CharField(verbose_name='Teléfono', max_length=20)
21
22     def __str__(self):
23         return "PrimerApellido: {0} -> CuentaBanco: {1} -> telefono {2}".format(self.PrimerApellido, self.CuentaBanco, self.telefono)
24

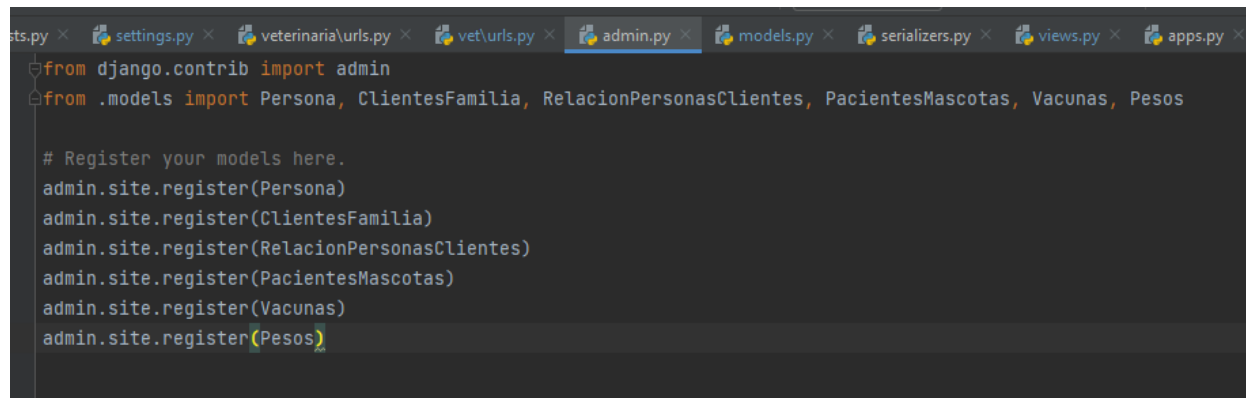
```

Nota: Creación de los modelos que va a llevar nuestra API, por Silvia Hernández Márquez, 2022.

Luego creamos otra clase llamada `PacientesMascotas` en la cual es la descripción de la mascota que llevan a la veterinaria y sus atributos fueron `IDClientesFamilia` la cual está relacionada con una clase anterior con un `ForeignKey`, un `AliasMascota`, `Especie`, `Raza`, `ColorPelo`, `FechaNacimiento` y `Vacunaciones`. Posteriormente creamos otra clase llamada `Vacunas` la cual tiene los atributos `IDPacientesMascotas`, `Fecha`, `Enfermedad` y `FechaProxima` la cual se refiere a la próxima consulta de la mascotas que está registrada. Estos atributos están escritos en nomenclatura camello.

Después se crea otra carpeta con el nombre de `admin.py` en la cual vamos a poner todos los registros que pusimos en nuestra carpeta anterior tomando los nombres de las clases para registrarlas en Admin y se coloca `admin.site.register (Persona)` cambiando `Persona` por el nombre de cada clase que hicimos.

Figura 3

Registros de nuestras clases en Admin.py

```
from django.contrib import admin
from .models import Persona, ClientesFamilia, RelacionPersonasClientes, PacientesMascotas, Vacunas, Pesos

# Register your models here.
admin.site.register(Persona)
admin.site.register(ClientesFamilia)
admin.site.register(RelacionPersonasClientes)
admin.site.register(PacientesMascotas)
admin.site.register(Vacunas)
admin.site.register(Pesos)
```

Nota: Registros de nuestras clases en Admin.py, por Silvia Hernández Márquez, 2022.

Posteriormente creamos otra carpeta con el nombre serializers.py en la cual pusimos las clases que creamos antes, pero con el nombre de PersonaSerializer cambiando Persona por el nombre de cada clase, en esta creamos otra clase dentro de esta llamada class Meta en la cual se pone un fields y se pone todos los atributos que pusimos en models con comillas simples y las colocamos en forma de lista, así con todos nuestros modelos.

Figura 4

Creación de los serializers

```

from vet import models

class PersonaSerializer(serializers.ModelSerializer):
    class Meta:
        fields = (
            'nombrepersona',
            'apellidopersona',
            'telefono',
            'direccion',
            'email'
        )
        model=models.Persona

class ClientesFamiliaSerializer(serializers.ModelSerializer):
    class Meta:
        fields = (
            'PrimerApellido',
            'CuentaBanco',
            'telefono'
        )
        model=models.ClientesFamilia

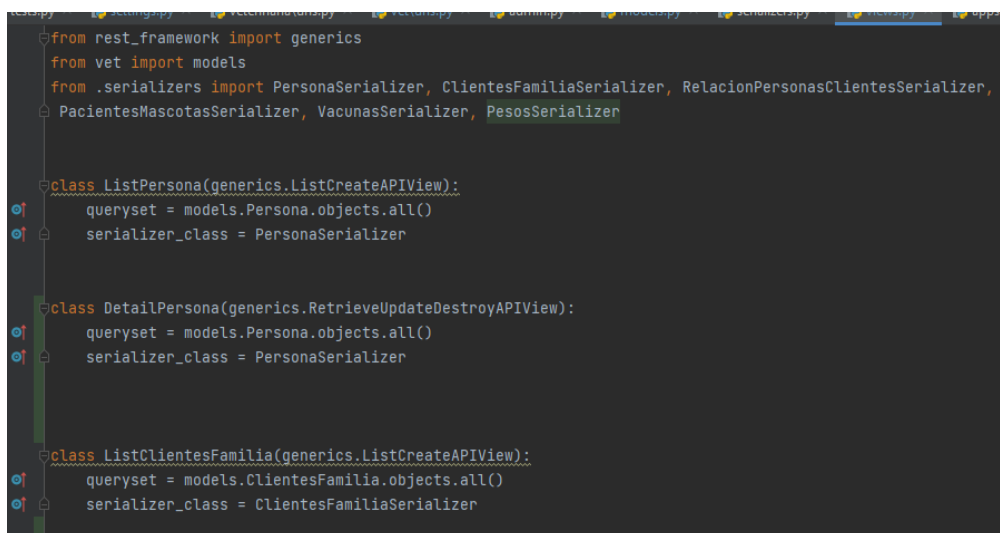
```

Nota: Creación de los serializers, por Silvia Hernández Márquez, 2022.

Nuestro siguiente documento es views.py dentro de estas colocamos nuestras clases, pero antes le anexamos la palabra list o Detail y ponemos queryset y serializer_class.

Figura 5

Creación de las vistas



```

from rest_framework import generics
from vet import models
from .serializers import PersonaSerializer, ClientesFamiliaSerializer, RelacionPersonasClientesSerializer,
PacientesMascotasSerializer, VacunasSerializer, PesosSerializer

class ListPersona(generics.ListCreateAPIView):
    queryset = models.Persona.objects.all()
    serializer_class = PersonaSerializer

class DetailPersona(generics.RetrieveUpdateDestroyAPIView):
    queryset = models.Persona.objects.all()
    serializer_class = PersonaSerializer

class ListClientesFamilia(generics.ListCreateAPIView):
    queryset = models.ClientesFamilia.objects.all()
    serializer_class = ClientesFamiliaSerializer

```

Nota: Creación de las vistas, por Silvia Hernández Márquez, 2022.

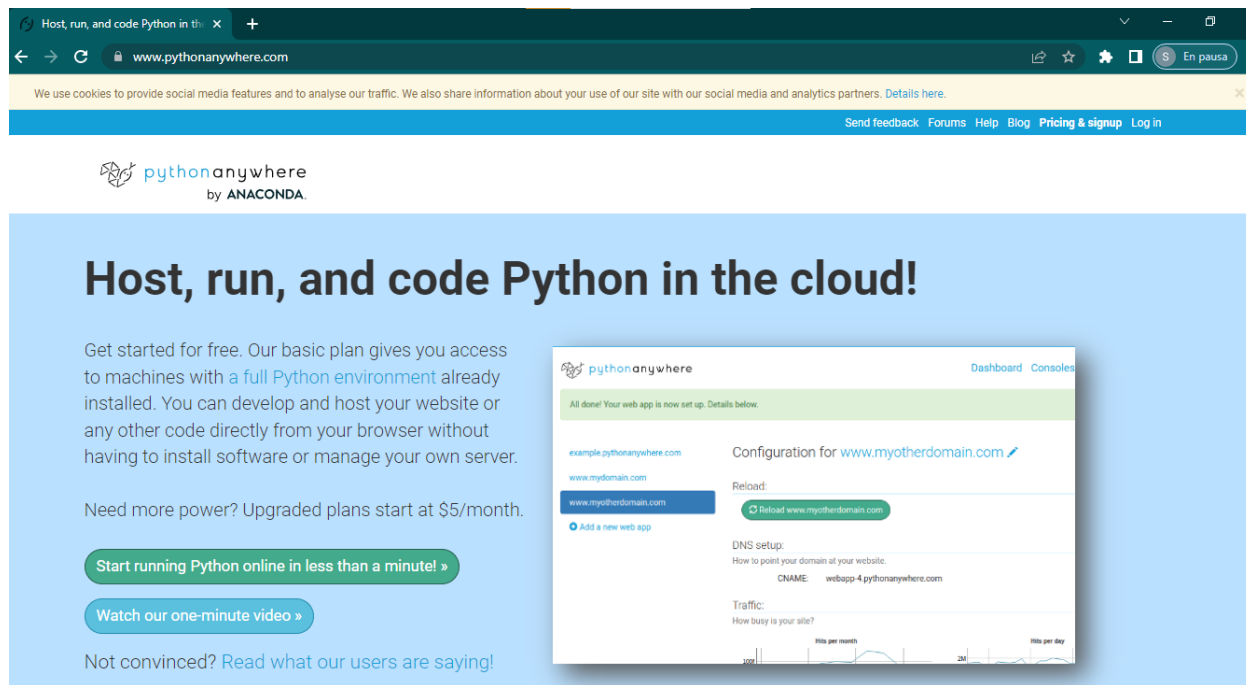
Posteriormente se crea otra carpeta llamada `urls.py` en la cual importamos `rest_framework`, `django.urls` y `vet.views`, dentro de esta ponemos el nombre de nuestras clases más aparte lo que colocamos en la carpeta vistas.

Por consiguiente, abrimos la terminal y ejecutamos ciertos comando para que nos arrojara nuestra url de nuestra api. Para poder ingresar copiamos la url y la pegamos en nuestro buscador poniendo una diagonal seguido de la palabra `vet` diagonal el nombre de la clase que deseamos abrir.

El siguiente paso fue buscar Pythonanywhere y creamos una cuenta que tenía que ver con nuestra API, pero primero creamos un correo relacionado con nuestra API en nuestro caso de veterinaria, para que todo fuera relacionado con esta, con este correo nos creamos una cuenta con Pythonanywhere ocupando ese correo y ponemos una contraseña.

Figura 6

Abrimos Pythonanywhere y pasamos nuestro script aquí



Nota: Abrimos Pythonanywhere y pasamos nuestro script aquí, por Silvia Hernández Márquez, 2022

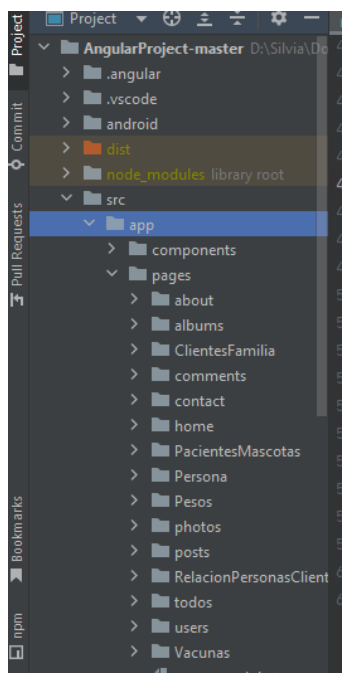
Una vez teniendo una cuenta creamos los archivos como los teníamos en Pycharm y copiamos los archivos y los pegamos en los documentos, pero ahora en Pythonanywhere, después nos vamos al apartado donde dice web apps que es donde está el url de la api para saber si ya esta subida.

Ahora nos dirigimos a cmd de la computadora donde pondremos el comando de `ionic start` el cual nos ayudara a crear el proyecto, eligiendo una plantilla en blanco con Angular y colocando el nombre del proyecto en nuestro caso lo llamamos `AngularProject-master`.

Una vez teniendo el proyecto lo abrimos en la aplicación de WebStorm y empezamos a crear carpetas que necesitaremos empezando por una llamada app la cual dentro de esta se crean más carpetas.

Figura 7

Creamos diferentes carpetas en WebStorm



Nota: Creamos diferentes carpetas en WebStorm, por Silvia Hernández Márquez, 2022

Adentro de la carpeta app se crea otra carpeta que llevara el nombre de pages y en esta carpeta se meterán las demás carpetas llamadas ClientesFamilia, PacientesMascotas, Persona, Pesos, RelacionPersonasClientes, Vacunas.

Todas estas carpetas antes mencionadas tendrán ciertos archivos los cuales vamos a editar de cada. Lo más importante es editar el archivo con terminación HTML en el

cual pondremos los atributos que habíamos hecho anteriormente con Pycharm. Así editando cada carpeta y poniendo en cada una sus diferentes atributos.

Una vez terminado de modificar las carpetas buscamos otra carpeta en el proyecto llamada data.service.ts en esta agregamos los link que obtuvimos en Pythonanywhere para poder obtener la información que registramos en este. Luego buscamos otra carpeta llamada pages.module.ts en donde importaremos todas las clases que pusimos en las otras carpeta.

Figura 8

Colocamos los link que obtuvimos en Pythonanywhere en data service

```
export class DataService {  
  constructor(private http: HttpClient) {  
  }  
  
  getPosts() {  
    return this.http.get( url: 'https://jsonplaceholder.typicode.com/posts');  
  }  
  
  getComments() {  
    return this.http.get( url: 'https://jsonplaceholder.typicode.com/comments');  
  }  
  
  getUsers() {  
    return this.http.get( url: 'https://jsonplaceholder.typicode.com/users');  
  }  
  
  getttodos() {  
    return this.http.get( url: 'https://jsonplaceholder.typicode.com/todos')  
  }  
  
  getalbums() {  
    return this.http.get( url: 'https://jsonplaceholder.typicode.com/albums')  
  }  
}
```

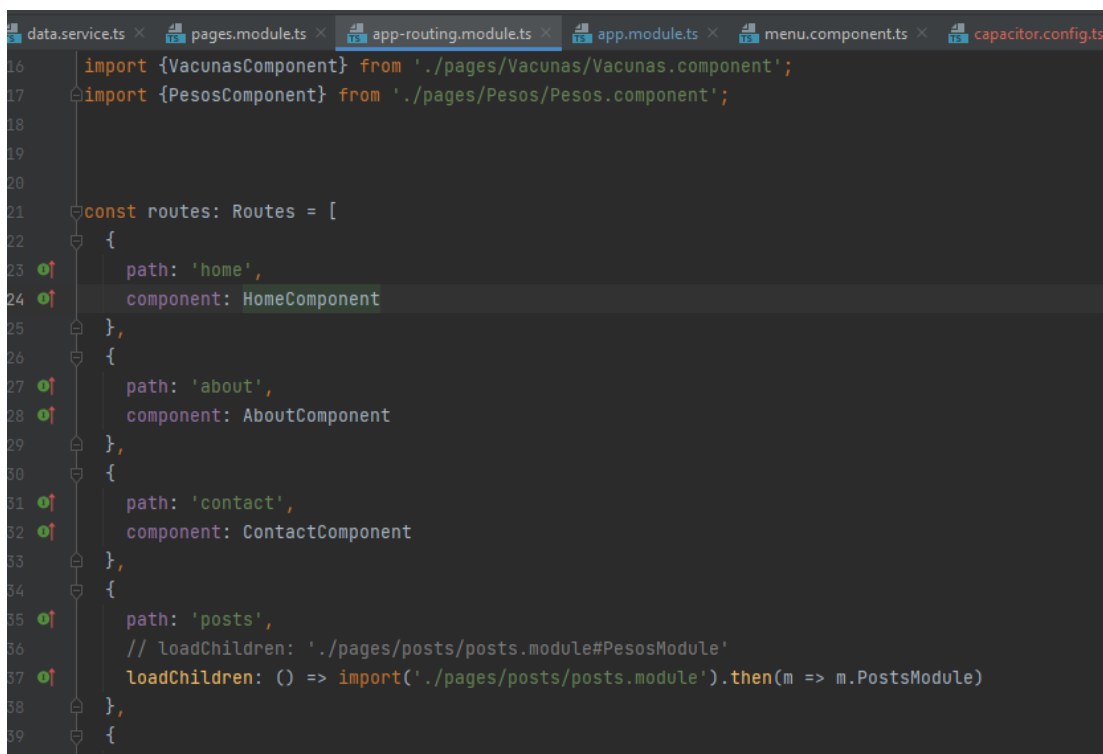
Nota: Colocamos los link que obtuvimos en Pythonanywhere en data service, por

Silvia Hernández Márquez, 2022

Seguidamente nos vamos a otra carpeta llamada `app-routing.module.ts` en la cual vamos a agregar cada clase con una path el nombre de la clase y abajo component el nombre de la clase seguido de `Component`, así con cada clase.

Figura 9

Abrimos la carpeta `app routing` y agregamos las clases.



```
data.service.ts × pages.module.ts × app-routing.module.ts × app.module.ts × menu.component.ts × capacitor.config.ts
16 import {VacunasComponent} from './pages/Vacunas/Vacunas.component';
17 import {PesosComponent} from './pages/Pesos/Pesos.component';
18
19
20
21 const routes: Routes = [
22   {
23     path: 'home',
24     component: HomeComponent
25   },
26   {
27     path: 'about',
28     component: AboutComponent
29   },
30   {
31     path: 'contact',
32     component: ContactComponent
33   },
34   {
35     path: 'posts',
36     // loadChildren: './pages/posts/posts.module#PesosModule'
37     loadChildren: () => import('./pages/posts/posts.module').then(m => m.PostsModule)
38   },
39   {
```

Nota: Abrimos la carpeta `app routing` y agregamos las clases., por Silvia

Hernández Márquez, 2022

Posteriormente dentro de la carpeta de app se crea otra carpeta llamada components y adentro de esta se crea otra carpeta llamada menú la cual contendrá varios archivos los cuales solo vamos a modificar menu.components.ts en donde pondremos name: el nombre de nuestra clase y debajo de esta path diagonal otra vez el nombre de nuestra clase, así con todas las clases para que queden registradas.

Guardamos todos los cambios que hicimos y abrimos la terminal donde pondremos el comando de ng serve el cual nos dará un link si todo está correcto y lo pegamos en nuestro buscador y nos tendrá que mostrar los registros que tenemos guardados de la API veterinaria.

Implementación De La API

La API se subió a Pythonanywhere un entorno de desarrollo integrado(IDE) en línea y de servicio de alojamiento web, la cual hospeda aplicaciones para su empleo web.

Una vez lista la API se encontrará disponible en la web, la entrada al sitio será gratuito. Pues el objetivo de esta es que los usuarios que llevan a sus perros a consulta en la veterinaria X la descarguen para llevar un control de los datos de estos.

Para que los usuarios entren a la API, en la veterinaria X se pegaron hojas con un código QR que los lleva directo a la página web donde se encontrara la API.

Para implementar que los usuarios ocupen la API , la veterinaria X puso una promoción para sus clientes. Esta consta que al demostrar que entras a la API en tu teléfono ellos te harán un descuento del 10% en la primera consulta que demuestres que ya entraste al sitio web donde está la API, para hacer esto seguro y no se repitan los mismos clientes al querer nuevamente el descuento ,tienes que mostrar tu cuenta en la API la cual tiene que mostrar los datos de la mascota que estas llevando a consulta, para que se haga valida la promoción.

Resultados De La API

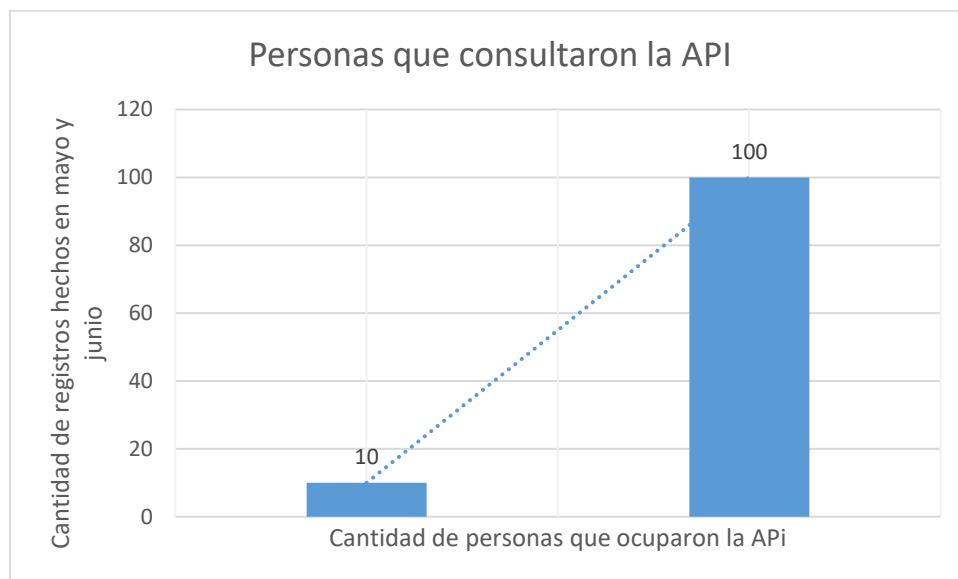
Para obtener los resultados de cómo fue la aceptación de la API, se estableció un periodo de prueba de uso de dos meses. Esta prueba inicio el primero de mayo y termino el primero de julio.

Se recolectaron los siguientes datos en base a los dos meses de prueba.

- La cantidad de registros hechos.
- Cuantas Personas ocuparon la API.

Tabla 1.

Recolección de datos de prueba.



Nota: Esta tabla muestra las personas que ocuparon la API y se registraron en esta durante los dos meses de prueba.

Los resultados arrojaron que la API fue un éxito ya que los usuarios se registraron y ocuparon la API para no olvidar las consultas de sus mascotas. Por lo que muchos de ellos la empezaron a recomendar para tener un mejor cuidado de sus mascotas.

La implementación de nuestra API tuvo una buena difusión y aceptación por parte de los dueños de las mascotas que llevan a la veterinaria X, ya que se les hace muy fácil usar y les ayuda mucho a recordar las consultas de sus mascotas.

De tal manera que les ayudo a recordar las consultas de sus mascotas y poder llevarlos en tiempo y forma a sus consultas. Además de que dicen que es muy útil para ver todos los registros de sus mascotas en cualquier momento.

Evaluación De La Calidad De La API

Se evaluó la calidad de la API a través del establecimiento de los siguientes parámetros.

Cliente

La población a la cual va dirigida esta API, es a las personas que tienen perros y los llevan a consulta a la veterinaria X, que está localizada en Teziutlán, Puebla.

Necesidad

La necesidad que se detectó en las personas que llevan a consulta a sus perros en la veterinaria X, es que necesitan tener una aplicación o lugar donde ver cuando les toca su próxima consulta a sus mascotas o que vacunas les faltan de aplicar a estas.

Herramienta Para Implementar

Recolección De Datos

Para la recolección de información de la presente investigación se utilizó la técnica de la encuesta. La encuesta consta de dos partes, la primera se divide en datos generales del dueño del perro, nombre del dueño, edad, sexo; la segunda parte consta de cinco preguntas de opción múltiple.

La aplicación de estas preguntas será enviada por mensaje de texto, pero antes tendríamos que enviar un mensaje para ver si la persona está dispuesta para ser parte de la encuesta, una vez que acepte se le envía el formulario o una vez que venga a la veterinaria X se le dará un código QR que tendrá que escanear para llenar la encuesta.

El cuestionario es el siguiente.

| Veterinaria X |
|-------------------|
| Nombre(s): |
| Primer apellido: |
| Segundo apellido: |
| Edad: |
| Sexo: |

Contesta las siguientes preguntas y envíalas al terminar de responderlas.

1. ¿Se te ha olvidado alguna consulta de tu perro a la que tenías que llevarlo?

Si ☐ No ☐

2. ¿Sabes cuantas vacunas tiene tu perro y cuales le faltan?

Si ☐ No ☐

3. ¿Te gustaría tener una aplicación que te enseñe cuando le toca su próxima consulta a tu perro o cuando fue la ultima a la que lo llevaste?

Si ☐ No ☐

4. ¿Sabes que es una API?

Si ☐ No ☐

5. ¿Te gustaría tener una aplicación que te enseñe todos los datos de tu perro?

Si ☐ No ☐

Con la encuesta se quieren obtener información para ver si la persona podría estar interesada en consumir una API para llevar el control de los datos de su mascota.

Evaluación

Para la evaluación de la API se les pidió a las personas que la han ocupado si podían llenar las siguientes preguntas para ver como fue su experiencia al ocupar esta.

¿Cómo calificarías tu experiencia ocupando la API?

| | | | | |
|-----------|-----------|-------|-----------|-------|
| Excelente | Muy Buena | Buena | Razonable | Pobre |
|-----------|-----------|-------|-----------|-------|

¿Qué probabilidad hay de que recomiendes la API a otras personas que llevan sus perros a la veterinaria X?

Nada probable

Muy probable

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

¿Se le hizo fácil de usar?

Si ☐ No ☐

Para que la API pueda ser usada por las personas es fundamental que esta sirva completamente, por lo cual se tienen que estar haciendo pruebas de su funcionamiento y que todo este corriendo correctamente.

La API se evaluará con la siguiente fórmula de distribución binomial.

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}$$

$$= \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x} \quad x = 0, 1, 2, \dots, n$$

En donde:

x= Es la cantidad de artículos para los cuales desea la probabilidad

p = Es la probabilidad de éxito

n = Es el número de artículos de la muestra

Sustituyendo en nuestra fórmula quedaría lo siguiente.

Datos que tenemos:

$$n=10$$

$$p= 0.5$$

$$x= 7$$

$$f(7) = \binom{10}{7} (0.5)^7 * (1-0.5)^{10-7}$$

$$= \frac{10!}{7!(10-7)!} (0.5)^7 * (1-0.5)^{10-7}$$

$$= \frac{3628800}{5040 (6)} (7.8125 \times 10^{-3}) (0.5)^3$$

$$= \frac{3628800}{30240} (7.8125 \times 10^{-3}) (0.125)$$

$$=(120) (7.8125 \times 10^{-3}) (0.125)$$

$$= 0.1171875 \times 100\% = 11.71875\%$$

Entonces la probabilidad de que a 7 personas de 10 guste la API es de 11.71%

Ahora sacaremos el valor esperado, la varianza y la desviación estándar.

Valor esperado

$$E[x] = \mu = n * p$$

$$E[x] = (10) * (0.5)$$

$$E[x] = 5$$

Varianza

$$\sigma^2 = n * p (1 - p)$$

$$\sigma^2 = (10) * (0.5) * (1 - 0.5)$$

$$\sigma^2 = (5) * (0.5)$$

$$\sigma^2 = 2.5$$

La varianza entre cada persona es de 2.5.

Desviación estándar

$$\sigma = \sqrt{n * p (1 - p)}$$

$$\sigma = \sqrt{(10) * (0.5) * (1 - 0.5)}$$

$$\sigma = \sqrt{2.5}$$

$$\sigma = 1.58113883$$

Producto Terminado

Manual De Como Utilizar La API

Elaboró: Silvia Hernández Márquez

Contenido

- i. Objetivo del manual
- ii. Lista de paso para ocupar la Api

El objetivo de este manual es proporcionar como se utiliza correctamente la API en cuestión.

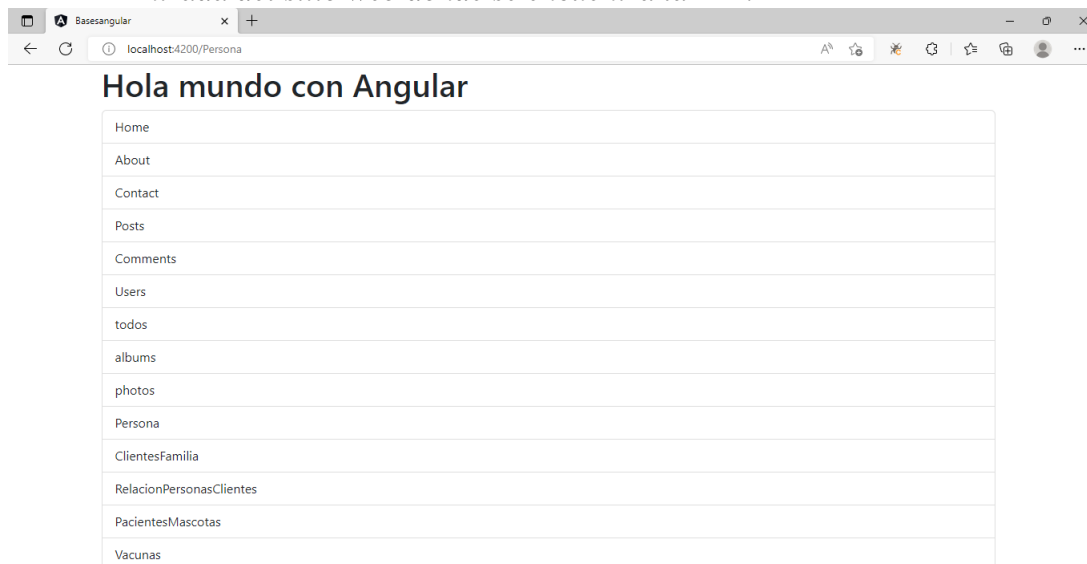
| | |
|--|------------------------------|
| Nombre del Procedimiento: Uso correcto de la API | |
| Fecha de emisión: 12 de agosto del 2022 | |
| I. | Objetivo del procedimiento |
| Saber como utilizar la API correctamente, sin tener problemas en entrar al sitio donde esta se encuentra. | |
| II. | Definición de que es una API |
| “Las API son mecanismos que permiten a dos componentes de software comunicarse entre sí, mediante un conjunto de definiciones y protocolos” (Amazon Web Services, 2022). | |
| III. | Referencias |
| Manual de calidad | |
| IV. | Políticas |
| Hacer uso correcto de la API. | |
| No difamar la API en cuestión si no le agrado. | |

| | |
|--|--------------------------------|
| Ver los registros de sus mascota. | |
| V. Formatos e Instructivos | |
| Clave de Formato/ Instructivo | Nombre del Formato/Instructivo |
| N/A | |
| VI. Listado de paso para utilizar la API correctamente | |
| 1. Entrar al link que se le proporciono para ver los registros de su mascota en este caso es el siguiente: http://localhost:4200/ | |
| 2. Abrir en tu buscador favorito el link proporcionado | |
| 3. En este te aparecerá la pagina principal la cual te muestra diferentes campos, tú decides a cuál entrar. | |
| 4. Al abrir el apartado te mostrara los registros que tienes dentro de los registros de tu mascota. | |
| Fin del procedimiento | |

Ahora se mostrará el resultado la API en cuestión.

Figura 10

Entrada del sitio web donde se encuentra la API.

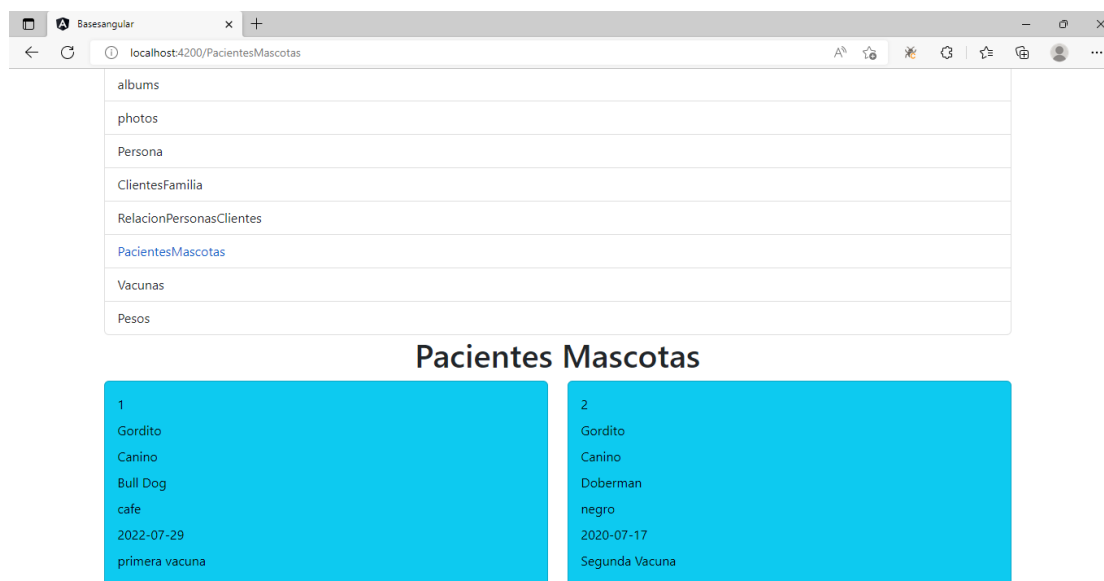


Nota: Entrada del sitio web donde se encuentra la API, por Silvia Hernández

Márquez, 2022, Basesangular, (<http://localhost:4200/>)

Figura 11

Selección de algún campo de interés.



Notas: Selección de algún campo de interés, por Silvia Hernández Márquez, 2022,

Basesangular, (<http://localhost:4200/>)

Conclusiones

A lo largo de este proyecto el objetivo principal era saber si las personas que llevan a sus perros a la veterinaria X estarían interesadas en consumir una API, según los resultados dados, estos tuvieron gran interés por esto.

Por lo cual la mayoría de ellos empezó a utilizar la API para recordar las consultas de sus perros y los datos de estos, además de empezar a compartirla con sus conocidos que llevan igualmente a sus perros a esta veterinaria.

Por lo tanto, podemos decir que esta API sirvió para dar solución al problema presentado antes.

Recomendaciones

Las recomendaciones que los usuarios que ocupan la API hicieron fue que podrían cambiar los colores de la pagina para que sea más llamativa.

Otra recomendación que hicieron fue que seria mejor si hicieran una aplicación para el celular con la misma API, para no tener que entrar a un link, ya que es mas tardado y seria mejor descargar directamente la aplicación.

Bibliografía

Amazon Web Services. (2022). Obtenido de <https://aws.amazon.com/es/what-is/api/>

Bill Wagner, M. K. (21 de 07 de 2022). *Microsoft*. Obtenido de <https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/classes-and-structs/fields>

CamelCase. (s.f.). Obtenido de <https://www.neumoytoraxpanama.org/cloud/resources/documentos/camelcase.pdf>

Conceptos básicos de la Programación Orientada a Objetos. (s.f.). Obtenido de <http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/clases1/clases.htm#:~:text=Los%20atributos%20son%20las%20caracter%C3%ADsticas,valores%20distintos%20para%20estas%20variables.>

Django API REST. (s.f.). Obtenido de <https://www.django-rest-framework.org/>

Kiko Palomares. (s.f.). Obtenido de <https://www.kikopalomares.com/blog/que-es-una-clase-en-programacion>

Significados.com. (s.f.). Obtenido de <https://www.significados.com/script/#:~:text=Script%20en%20inform%C3%A1tica,de%20un%20programa%20de%20computador.>

Tecnología Inclusiva. (s.f.). Obtenido de <https://desarrollarinclusion.cilsa.org/tecnologia-inclusiva/que-es-un->

programa/#:~:text=Un%20programa%20es%20un%20conjunto,cual%20interactuamos%20con%20la%20aplicaci%C3%B3n.

uniwebsidad. (s.f.). Obtenido de <https://uniwebsidad.com/libros/django-1-0/capitulo-3/mapeando-urls-a-vistas/#:~:text=B%C3%A1sicamente%2C%20es%20un%20mapeo%20entre,llama%20a%20este%20otro%20c%C3%B3digo%22>.

Anexos

Glosario

Script: (“es un documento que contiene instrucciones, escritas en códigos de programación”). (Significados.com, s.f.)

Atributos: (“son las características individuales que diferencian un objeto de otro y determinan su apariencia, estado u otras cualidades”). (Conceptos básicos de la Programación Orientada a Objetos, s.f.)

Nomenclatura Camello: (“es un estilo de escritura que se aplica a frases o palabras compuestas”). (CamelCase, s.f.)

Programa: (“es un conjunto de pasos lógicos escritos en un lenguaje de programación que nos permite realizar una tarea específica”). (Tecnología Inclusiva, s.f.)

Class: (“es una plantilla para el objetivo de la creación de objetos de datos según un modelo predefinido”). (Kiko Palomares, s.f.)

Infografía de que es una API



Notas: Infografía de que es una API, por Silvia Hernández Márquez, 2022, (google 2022)

Como funciona una API



Notas: Como funciona una API, por Silvia Hernández Márquez, 2022, (google 2022)