

Proyecto Estilos Arquitectónicos 1 - Subrutinas y Programa Principal

Esteban Castaño Gómez
Docente: Alejandro Rodas Vásquez
Universidad Tecnológica de Pereira

5 de octubre de 2021

Introducción

En este proyecto usted pondrá en práctica los conceptos básicos vistos en esta unidad especialmente *arreglos*, *funciones*, *módulos*. Se deberá consumir los recursos suministrados por una API suministrada por el portal Nacional de [Datos Abiertos](#) y crear una Arquitectura de Software basada en módulos.

Durante esta pandemia la adquisición, almacenamiento y consulta de datos relacionado con el reporte de casos de COVID-19 se ha convertido en un insumo imprescindible para la creación de aplicaciones software que ayuden en la prevención y trazabilidad de la enfermedad.

El portal de Datos Abierto pone a disposición el conjunto de datos llamado [Casos positivos de COVID-19 en Colombia](#) (se recomienda que sea examinado) el cual puede ser consultado por aplicaciones web o de escritorio a través de consultas a su API. A continuación, se encuentra el código fuente en Python que permite realiza dicha consulta.

```

1 import pandas as pd
2 from sodapy import Socrata
3
4 # Unauthenticated client only works with public data sets. Note 'None'
5 # in place of application token, and no username or password:
6 client = Socrata("www.datos.gov.co", None)
7
8 # Example authenticated client (needed for non-public datasets):
9 # client = Socrata(www.datos.gov.co,
10 #                  MyAppToken,
11 #                  username="user@example.com",
12 #                  password="AFakePassword")
13
14 # First 2000 results, returned as JSON from API / converted to Python
15 # list of dictionaries by sodapy.
16 results = client.get("gt2j-8ykr", limit=limite_registros, departamento =
17                       nombre_departamento)
18
19 # Convert to pandas DataFrame
20 results_df = pd.DataFrame.from_records(results)

```

En las líneas 1 y 2 se hace la llamada a las librerías *Pandas* y *sodapy*. *Pandas* es reconocida por ser empleada en *Análisis de Datos*. En la línea 16 se encuentran los parámetros *limite_registros* y *nombre_departamento*. Estos parámetros son requeridos y como su nombre lo dice, permiten consultar el Departamento y el Número de Casos que se quieren extraer. Para mayor conocimiento sobre el API se tiene el sitio [Documentación sodapy](#).

1. Requerimientos Funcionales

Usted debe de crear una aplicación que le permita al usuario ingresar el Departamento (*nombre_departamento*) que desea consultar y el Número de Registro (*limite_registros*) que quiere obtener de la consulta (este parámetro es importante puesto que si selecciona un número grande e.j 1000 registros, la consulta puede “colgarse”).

El resultado obtenido debe de poder visualizarse en la pantalla en un formato (investigar la función *format*) que contenga solo las siguientes columnas: *Ciudad de ubicación, Departamento, Edad, Tipo, Estado y País de procedencia*.

2. Requerimientos de la Arquitectura de Software

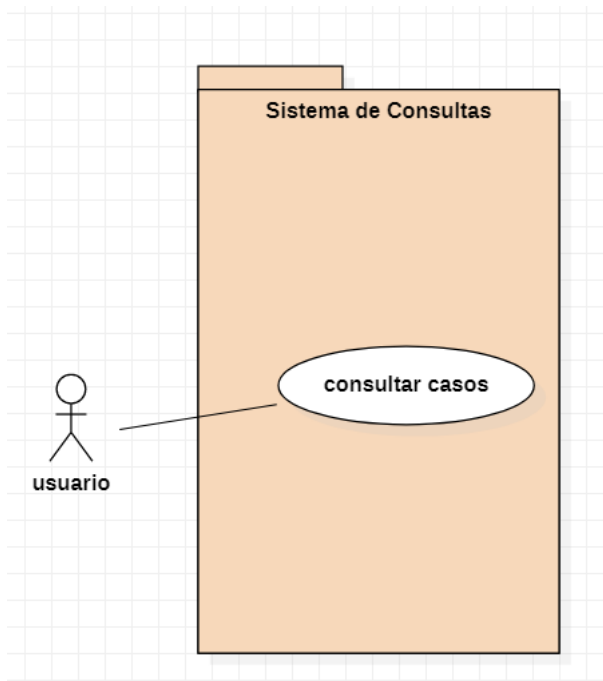
Esta aplicación debe de:

- Seguir el *Estilo Arquitectónico Subrutinas y Programa Principal*.
- Ser desarrollada en Python (aplicación de Escritorio) utilizando los conceptos de *Módulos y Namespace*.

3. ¿Cómo realizo la entrega?

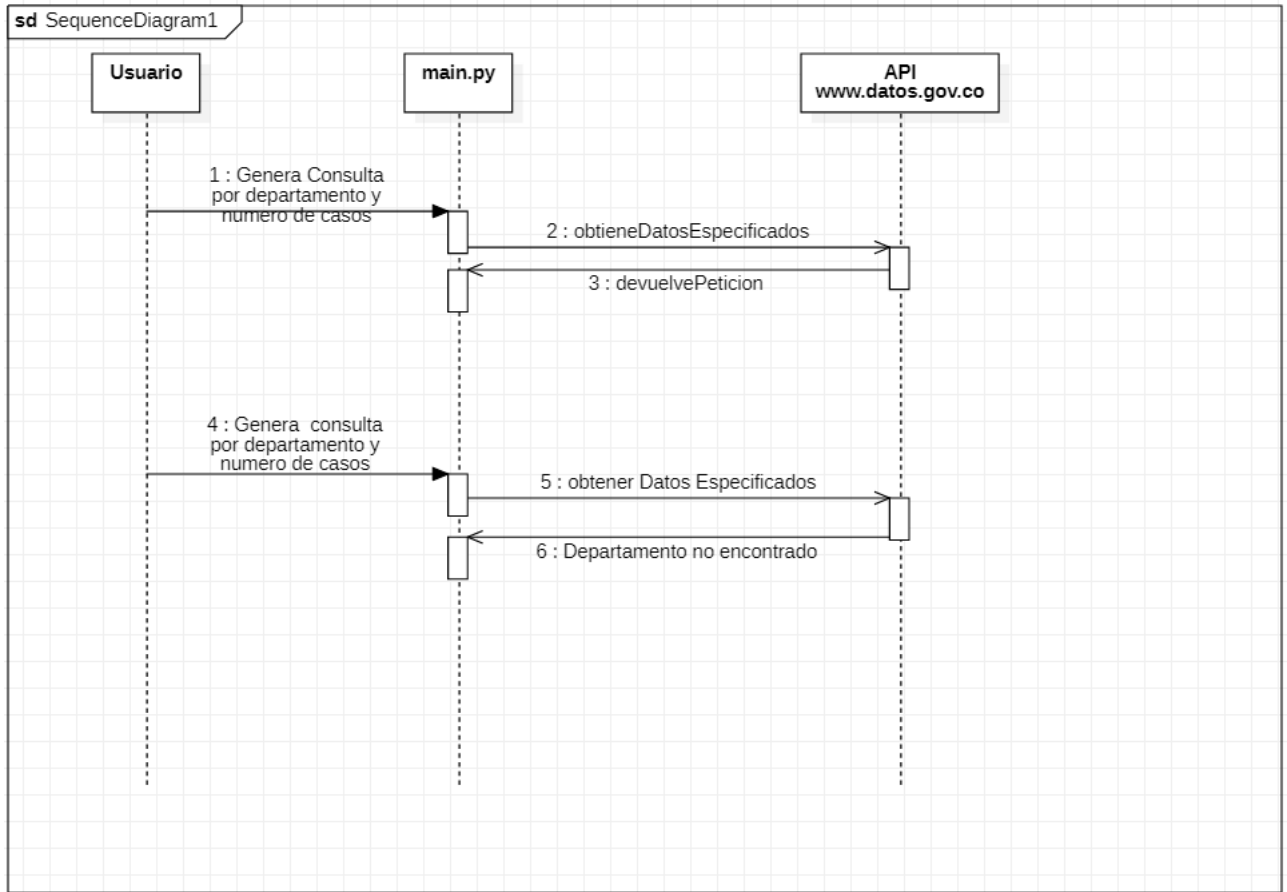
1. Usted debe de entregar el código fuente (archivo comprimido).
2. Pantallazos donde se corrobore el funcionamiento del software (consultas realizadas y resultado obtenido) y se compruebe que los Requerimientos Funcionales ([Sección 1](#)) han sido cumplidos.
3. Aplicando el *Modelo de vista 4 + 1 de Kruchten*; se deben entregar los Diagramas UML de las 4 vistas correspondientes y el escenario seleccionado para el análisis.
4. Por cada Diagrama UML usted debe de brindar una descripción del diagrama de modo que se pueda tener un contexto sobre el mismo.

3.1. Escenario



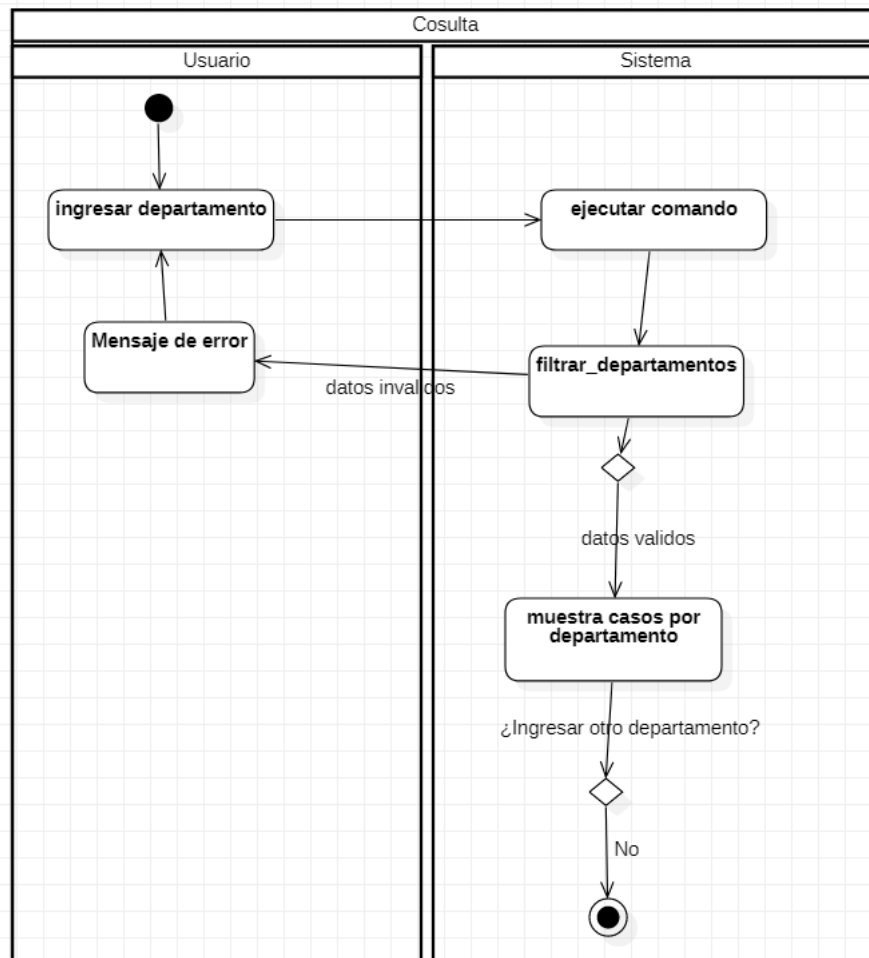
Los casos de uso muestran una interacción típica entre un usuario y un sistema de cómputo. En el caso particular de nuestro programa, muestra la interacción que tiene el usuario con el sistema de consultas. En el diagrama no se implementó el actor implícito, "www.datos.gov.co" que nos enfocamos en la funcionalidad básica del programa base.

3.2. Vista Lógica



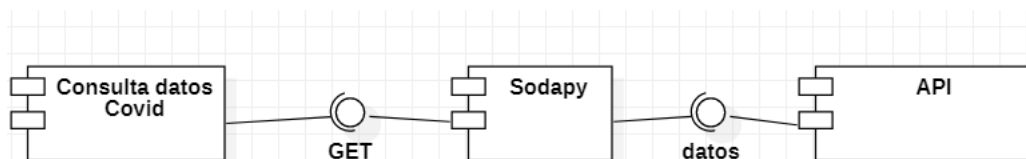
El diagrama de secuencia es un tipo de diagrama usado para modelar interacción entre objetos. Nuestro diagrama de secuencia es importante para tener una idea más específica de como interactua el sistema entre los objetos haciendo más especifico el sentido de la aplicacion.

3.3. Vista de Proceso



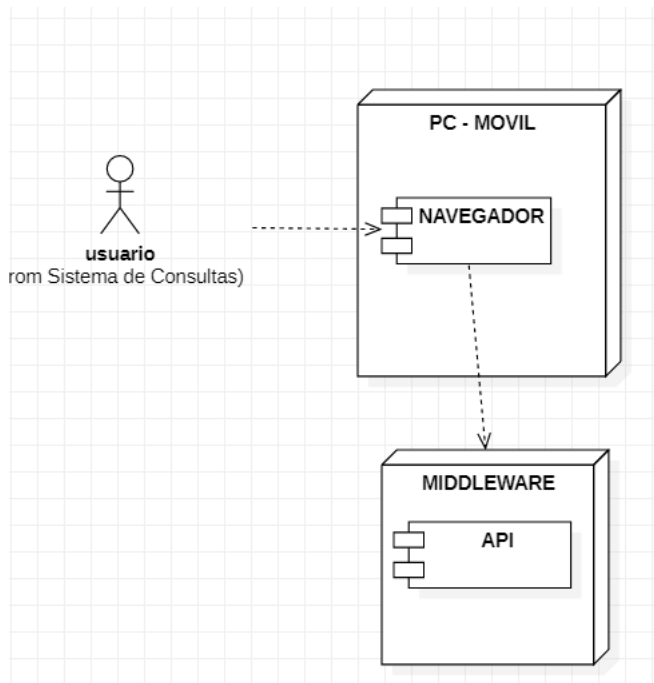
El diagrama de actividad es la representación gráfica de un algoritmo o proceso, también un diagrama de flujo que muestra actividades ejecutadas por un sistema. En la vista de nuestra aplicación se describen ciertas actividades que realiza el usuario y que el sistema le devuelve.

3.4. Vista de Desarrollo



Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. En nuestro caso 3 componentes fueron suficientes para satisfacer las necesidades básicas de nuestra aplicación. El componente 'Consulta datos covid', 'Sodapy' y 'API'.

3.5. Vista de Física



El diagrama de despliegue se utiliza para modelar la disposición física de los artefactos software en nodos. Muestra la arquitectura del sistema como el despliegue de los artefactos de software a los objetivos de despliegue. Nuestra vista física que incluye nuestro diagrama de despliegue muestra por encima la interacción de nuestros nodos. En este caso se pudo satisfacer la necesidad teniendo dos nodos, "PC-MOVIL" para hacer referencia a un dispositivo final que tenga un navegador y nuestro middleware que facilita la de los datos.

4. Evidencias

Bienvenido al Sistema de Consultas Covid-19 en Colombia

Esteban Castaño Gómez

Numero de Casos:

Seleccione Departamento Deseado:

Submit

ciudad_municipio_nom	departamento_nom	edad	tipo_recuperacion	estado	pais_viajo_1_nom
filter data...					
BOGOTA	BOGOTA	19	PCR	Leve	ITALIA
BUGA	VALLE	34	PCR	Leve	ESPAÑA
MEDELLIN	ANTIOQUIA	50	PCR	Leve	ESPAÑA
MEDELLIN	ANTIOQUIA	55	PCR	Leve	
MEDELLIN	ANTIOQUIA	25	PCR	Leve	
ITAGUI	ANTIOQUIA	27	PCR	Leve	
CARTAGENA	CARTAGENA	85	PCR	Leve	ESTADOS UNIDOS DE AMÉRICA
BOGOTA	BOGOTA	22	PCR	Leve	ESPAÑA
BOGOTA	BOGOTA	28	PCR	Leve	ESPAÑA
BOGOTA	BOGOTA	36	PCR	Leve	ESPAÑA

Un poco de lo que es nuestro programa en funcionamiento. Utilizando como herramienta de visualizacion, *Dash*