

Documentación

MINUTA 1

Fecha: lunes 28 de abril

Miembros del equipo:

Castillo Pacheco Liam Javier
Castro Reyes José Santiago
Garza Castillo Anyela Taide
Guajardo Sánchez Adán Rosalino
Rojas Rangel Carlos Alexander

Roles asignados:

Liam Javier- redactor de planteamiento
José Santiago- programador del código
Anyela Taide- coordinador
Adán Rosalino- programador del código
Carlos Alexander- redactor de la justificación

Objetivos del día:

- Investigar y decidir qué API seleccionar
- Asignar los roles de cada miembro
- Redactar el planteamiento del problema
- Hacer la estructura de datos
- Iniciar el código

Acuerdos tomados:

Elegimos API, e investigamos su conexión con el código. Observamos como es la estructura de datos. Repartiremos los roles.

Dificultades encontradas:

Dificultad en coincidir con los horarios de cada miembro. Dificultad en la selección de un API, pero al final encontramos uno.

Próximos pasos:

Desarrollar y terminar el código, la creación de archivo de documentación.

MINUTA 2

Fecha: jueves 1 de mayo

Miembros del equipo:

Castillo Pacheco Liam Javier

Castro Reyes José Santiago

Garza Castillo Anyela Taide

Guajardo Sánchez Adán Rosalino

Rojas Rangel Carlos Alexander

Roles asignados:

Liam Javier- redactor de planteamiento

José Santiago- programador del código

Anyela Taide- coordinador

Adán Rosalino- programador del código

Carlos Alexander- redactor de la justificación

Objetivos del día:

- Revisar y terminar el código
- Juntar y revisar la redacción de la documentación

Acuerdos tomados:

Quien entregará los documentos, la revisión del trabajo, la ejecución correcta del código.

Dificultades encontradas:

Dificultad en coincidir con los horarios de cada miembro. Encontrar los errores del código.

Próximos pasos:

Entregar avance de PIA.

MINUTA 3

Fecha: lunes 5 de mayo

Miembros del equipo:

Castillo Pacheco Liam Javier

Castro Reyes José Santiago

Garza Castillo Anyela Taide

Guajardo Sánchez Adán Rosalino

Rojas Rangel Carlos Alexander

Roles asignados:

Liam Javier- diseñador de diagrama de flujo

José Santiago- programador del código

Anyela Taide- diseñador de diagrama de flujo

Adán Rosalino- encargado de algoritmo

Carlos Alexander- redactor de la justificación

Objetivos del día:

- Asignar los roles de cada miembro
- Verificar código
- Iniciar algoritmo y diagrama de flujo

Acuerdos tomados:

Elegimos día de grabación, redacción de guión, el rol de cada miembro en esta parte del PIA.

Dificultades encontradas:

Dificultad en coincidir con los horarios de cada miembro. Que explicar en el video.

Próximos pasos:

terminar diagrama de flujo, algoritmo y video.

MINUTA 4

Fecha: Jueves 8 de mayo

Miembros del equipo:

Castillo Pacheco Liam Javier

Castro Reyes José Santiago

Garza Castillo Anyela Taide

Guajardo Sánchez Adán Rosalino

Rojas Rangel Carlos Alexander

Roles asignados:

Liam Javier- diseñador de diagrama de flujo

José Santiago- programador del código

Anyela Taide- diseñador de diagrama de flujo

Adán Rosalino- encargado de algoritmo

Carlos Alexander- redactor de la justificación

Objetivos del día:

- Revisar diagrama de flujo
- Revisar el algoritmo
- Edición del video
- Revisar documentación

Acuerdos tomados:

Quien entregará los archivos, la edición del video, adjuntar todas las partes en un solo documento.

Dificultades encontradas:

Dificultad en coincidir con los horarios de cada miembro. Teams falla, por lo tanto, se nos complica el intercambio de archivos.

Próximos pasos:

Entregar el PIA.

Planteamiento del Problema

Tras analizarlo y hablarlo en equipo llegamos a la problemática, la cual fue la de la inflación y la economía de los países y sobre lo difícil que puede llegar a ser identificar los datos, ordenarlos, usarlos para hacer gráficas y tomar diferentes medidas. por lo que se decidió crear un programa el cual automatice estos procesos y haga todo esto con solo 1 click. para esto se acordó hacer uso de la herramienta python para crear un programa el cual sea capaz de automatizar estos procesos. usando métodos de ordenamiento, programación orientada a objetos y diccionarios para que nos apoye a leer los documentos y textos necesarios para lograrlo, por lo que se repartieron los roles y se inició con el proceso de la solución de esta problemática. por último lo que se buscara es imprimir todos los datos para que el que desee automatizar estos procesos pueda ver de forma ordenada y clara los resultados que arroja este programa.

Descripción del API elegida

Nombre de la API: Sistema de Información Económica

Proveedor: Banco de México (Banxico)

Enlace: <https://www.banxico.org.mx/SieAPIRest/service/v1/>

Propósito: “El API del Sistema de Información Económica permite consultar las series de tiempo de todos los indicadores disponibles en este repositorio. Esta herramienta está orientada a desarrolladores o analistas que requieren la automatización del proceso de extracción de información económica publicada por el Banco de México.”

Objetivo: El objeto es extraer datos que nos permitan analizar el comportamiento de la inflación a lo largo de los años y realizar análisis estadísticos.

Descripción de la estructura de datos utilizada

Este programa se conecta a la API de Banxico para obtener datos en formato JSON que contiene registros con fechas y valores numéricos expresados como texto. A partir de esa información, se construye un DataFrame de pandas para facilitar su manejo. Las fechas se convierten a objetos datetime para poder ordenarlas correctamente, y los valores se transforman a tipo flotante para permitir cálculos. Una vez procesados, los datos se organizan cronológicamente y se pueden guardar como un archivo Excel.

Justificación del tratamiento de datos aplicada

Los datos que sacamos del API hablan del comportamiento de la inflación a lo largo del tiempo, estos datos están en un formato que sería difícil trabajar, empezamos en el código con creando una función el cual nos permite sacar los datos requeridos a la hora de usarlo mientras usa la función de try-except para evitar errores, usamos otras funciones que sirven para convertir los datos en formas en las que podemos usar más fácilmente, en este caso las funciones sacan la información del API por medio del URL de la página y lo guarda en formato JSON para después guardarlos en un archivo de texto, después toma los datos del archivo de texto y los pone en una lista de diccionarios anidados el cual se convierte a dataframes de pandas para más fácilmente convertir los datos en los tipos de variables que podemos usar en Python para poder organizar, calcular y guardar. Luego organiza los datos por fecha y toma los datos de los dataframes para exportarlos en un documento de Excel, usando la función de try-except para evitar errores, por último usa matplotlib para poder usar los datos y crear gráficas para representar gráficamente aquellos datos. Usando este código podemos tomar los datos del API acerca del Banco de México que tienen que ver con la inflación, organizarlos y convertirlos en un documento de Excel y crear gráficas para mejor interpretación.

Tabla comparativa de las API's consideradas

	API INEGI	API BANXICO
Información por proporcionar	La API del INEGI proporciona información específica de muchos sectores importantes de México, como lo económico, social, etc.	La API del Banco de México proporciona datos generales y específicos sobre la economía de México, como valores sobre la moneda, la inflación, el turismo y más.
Características	Esta API solicita un token que necesitas solicitar en su apartado oficial de solicitudes aquí . Esta también te permite recuperar datos de 5 formatos diferentes y de varios "indicadores" a la vez.	Esta API solicita un token que se consigue con una verificación simple en su apartado oficial aquí . Para utilizarla, requieres la serie, la fecha de consulta, y tu token.
Límite de consultas	No se proporciona información del límite de consultas.	Para consultas oportunas, 80 consultas en 1 minuto y 40,000 consultas por día. Para consultas históricas, 200 consultas en 5 minutos y 10,000 consultas por día.
Conclusiones	Proporciona datos muy bien expresados, pero finalmente decidimos en contra de ella debido al enfoque que le quisimos dar al trabajo.	Fue elegida por proporcionar los datos exactos que requerimos, siendo este el índice de precios al consumidor a través de los años.

Guión del podcast

(El número se refiere al integrante que realizó el diálogo)

E1. ¡Hola a todos! En este trabajo y con ayuda de datos del Banco de México y herramientas tecnológicas, nuestro equipo encontró patrones sorprendentes en el costo de vida de los habitantes de México.

E1. Para obtener los datos, usamos una API, específicamente la de Banxico. Le decimos qué información necesitamos, en este caso, la inflación,, y él va al Banco de México, la recoge y nos la entrega en un formato que podemos entender.

E2. Nosotros, por ejemplo, le pedimos algo como: “Quiero los datos de inflación desde el 2000 hasta 2024”. La API responde con números y fechas, sin embargo, en la manera que nos entregan poco sirve.

E3. Por eso, tuvimos que limpiarlos y organizarlos en una tabla, para poder trabajar más fácil con ellos.

E4. Uno de los retos fue que las fechas venían como texto. Debido a que la computadora no puede interpretar el formato que nos dieron, usamos un programa para convertirlas a un formato universal, como poner todas las fechas en orden año/mes/día.

E2. También había números con comas, como 7,45. Para que la computadora los entendiera, cambiamos las comas por puntos, como se usa en las calculadoras.

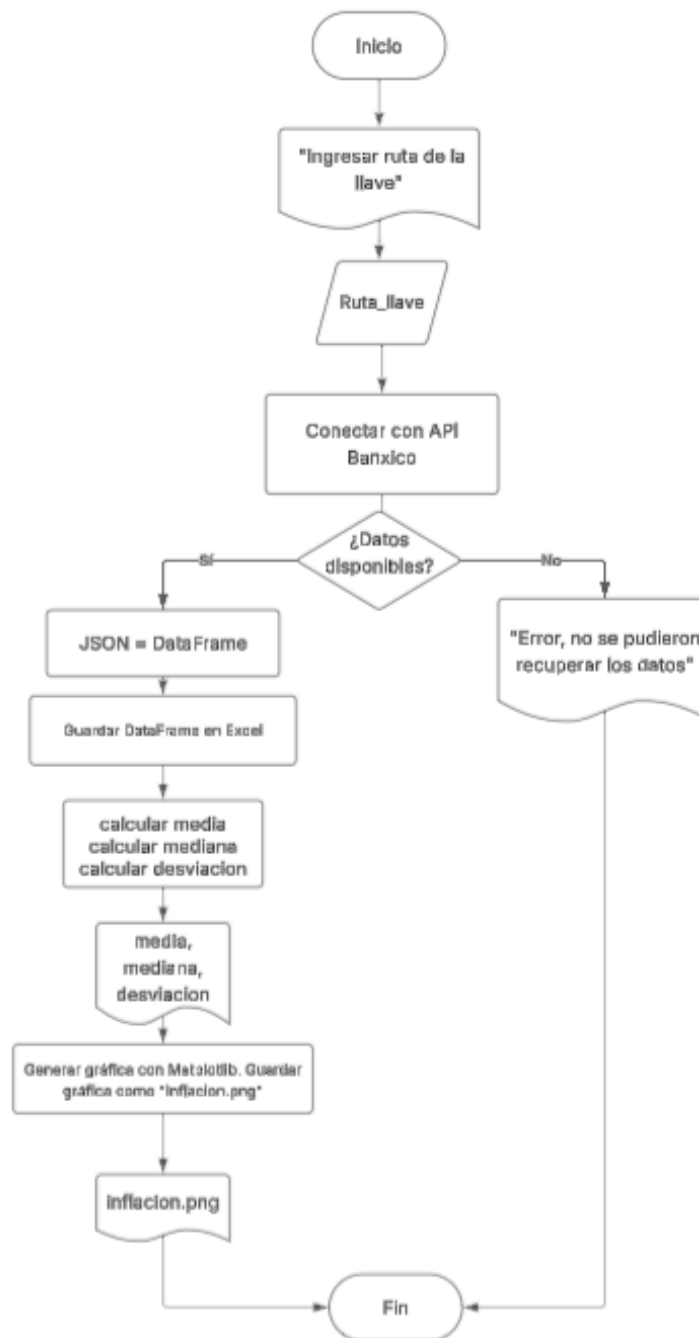
E4. Al organizar los datos, descubrimos cosas fascinantes. Por ejemplo, en 2008, durante la crisis mundial, la inflación subió como un cohete. En el 2017 otro pico y en el 2023, otro pico más, ligado a eventos globales.

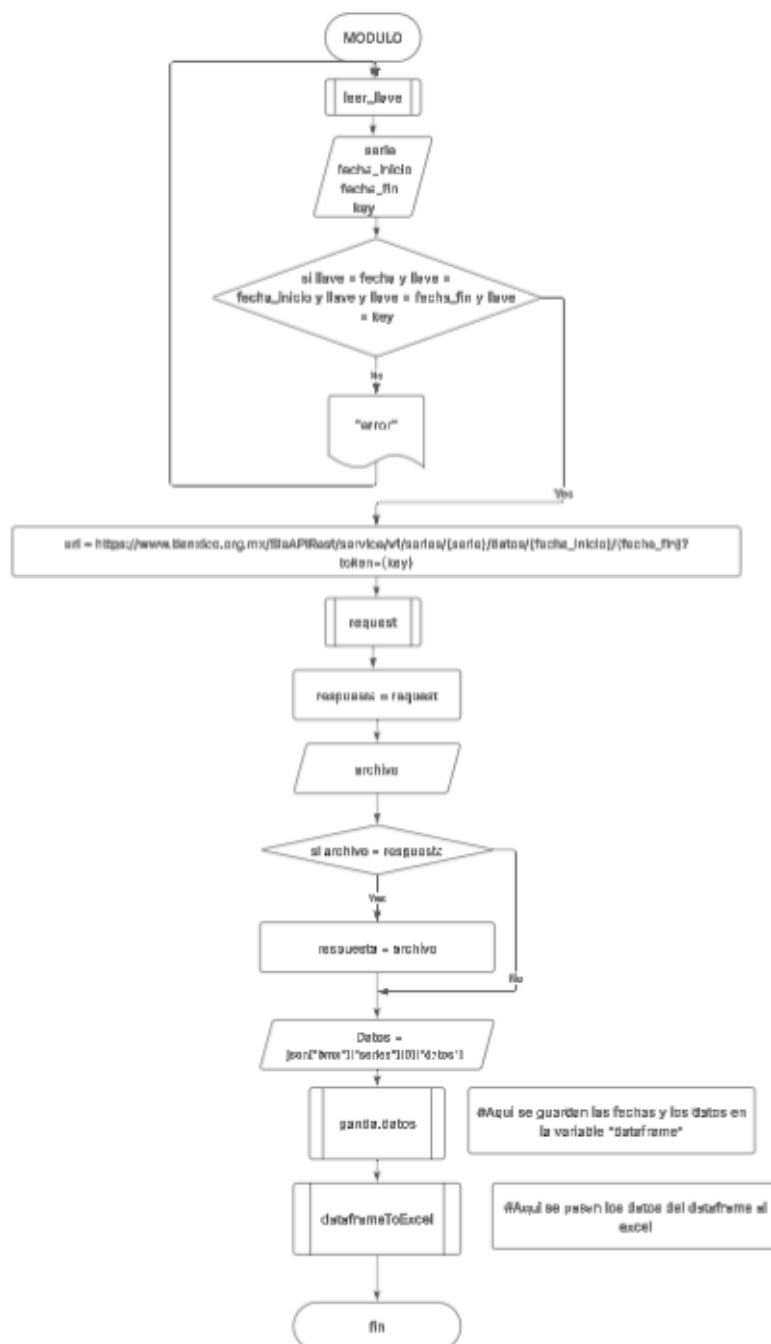
E5. En promedio, la inflación fue del 4.3% anual. Pero hubo meses buenos como 2003, con solo 3.5% y meses difíciles como 2023, con casi 9%.

E3. Entender la inflación nos ayuda a tomar decisiones informadas, como ahorrar o invertir. Y gracias a la tecnología, cualquiera puede explorar estos datos.

E5. ¡Gracias por escuchar! Recuerden que los números revelan información valiosa y hoy aprendimos a leer algunas muy importantes.

Diagramas de flujo





Link de los diagramas:

https://lucid.app/lucidchart/4c93e846-ec74-47a0-a54f-091215a8df6f/edit?viewport_loc=-6096%2C-1775%2C13175%2C6325%2C0_0&invitationId=inv_2d0e06ca-855a-40e5-9636-e44a367c244a

Estructura de datos

La estructura que se utilizó es muy sencilla, debido a que el uso de pandas facilita el manejo de datos y crea estructuras funcionales. Este dataframe de pandas está conformado por dos tipos de datos, las fechas y los datos, que en este caso fue el IPC en la fecha de recuperación. Se acomodan en dos columnas, que corresponden a los dos tipos de datos. Este funciona como una lista de diccionarios, donde cada clave tiene su valor.

Justificación del tratamiento

El único tratamiento que se le tuvo que dar a los datos fue la conversión de estos, debido al formato que presentaban. Los datos que regresó la API no necesitaron más limpieza debido a que estos eran exactos.

La conversión que los datos sufrieron fue una simple cuestión de convertirlos de string a un formato que se pueda manejar, formato de fecha con datetime para las fechas de recuperación para poder ordenarlas en orden ascendente, y float para los valores para poder realizar los cálculos estadísticos y las gráficas.

Algoritmo de los pasos del desarrollo del programa

1. Identificamos un problema, siendo este la inflación en México.
2. Analizamos y comparamos las APIs que nos pudiesen ser de utilidad.
3. Realizamos un programa que nos ayude a recuperar información de estas APIs.
4. Asignamos los valores a una estructura de datos con ayuda de pandas.
5. Con esta librería, exportamos los valores a excel para analizarlos.
6. Utilizando matplotlib, realizamos una gráfica para representar estos valores de manera gráfica.
7. Observamos los resultados y conseguimos conclusiones.