



ISIS-1221

INTRODUCCIÓN A LA PROGRAMACIÓN

Proyecto de Nivel 4 Prueba ICFES Saber 11 en Colombia

Objetivo general

El objetivo de este proyecto es crear una aplicación para analizar las estadísticas de las pruebas ICFES Saber 11 de Calendario A del año 2020 en Colombia. En el desarrollo de esta aplicación pondrá en práctica los conceptos del nivel 4.

Objetivos específicos

1. Implementar algoritmos para construir y recorrer matrices.
2. Utilizar las librerías pandas y matplotlib, así como consultar los sitios web oficiales donde se encuentra la documentación.
3. Descomponer un problema en subproblemas e implementar las funciones que los resuelven.

Instrucciones generales

La sección descripción de la aplicación le permitirá conocer el alcance, las funcionalidades esperadas y lo que debe realizar en este proyecto. Tenga en cuenta que, a lo largo de dicha sección encontrará el título "**ATENCIÓN**" con indicaciones para conseguir que el resultado de su proyecto corresponda con lo esperado. Es importante que las siga cuidadosamente. Antes de empezar, le sugerimos leer con atención todo el proyecto. Mientras lo lee, trate de reconocer los conceptos del curso que tendrá que poner en práctica. Recuerde que este proyecto debe realizarse de forma **completamente individual**.

Descripción de la aplicación

La Prueba ICFES Saber 11 es un examen estandarizado realizado por el Icfes, el cual tiene como propósito servir de criterio para el ingreso de los estudiantes bachilleres a las instituciones de educación superior en Colombia. Adicionalmente, sirven de indicador de la calidad formativa que brindan los establecimientos de educación media en el país. La prueba consiste en responder preguntas de opción múltiple. Esta consta de dos sesiones en las cuales se evalúa al estudiante a través de 5 pruebas: Lectura Crítica, Matemáticas, Ciencias Naturales, Sociales y Ciudadanas e Inglés. Además de presentar las 5 pruebas, el estudiante debe responder un cuestionario socioeconómico que está compuesto de preguntas cortas de selección múltiple que no tienen calificación. Este cuestionario permite obtener información sobre los procesos de aprendizaje y enseñanza de los estudiantes que podrían ayudar a explicar los resultados obtenidos en el examen. En este proyecto usted trabajará con datos de las pruebas ICFES Saber 11 de Calendario A del año 2020. Estos datos se encuentran registrados en el portal de datos del gobierno colombiano (<https://www.datos.gov.co/Educaci-n/Saber-11-2020-2/rnvb-vnyh>). En el archivo "saber11.csv" usted encuentra una versión simplificada y filtrada de los datos originales: suprimimos algunas entradas incompletas y columnas que no vamos a utilizar en este proyecto. Con este conjunto de datos usted podrá analizar información esencial de los estudiantes próximos a ser bachilleres en el territorio nacional.

¡Le deseamos la mejor de las suertes explorando estos datos!

Su aplicación debe tener las siguientes partes:

Parte 1: Leer la información del archivo

Requerimiento 0: Cargar datos

Lo primero que debe hacer es permitir que se carguen los datos de un archivo csv a un DataFrame. Le debe preguntar al usuario el nombre del archivo, es decir que, **la función que implemente este requerimiento debe recibir como parámetro el nombre del archivo y debe retornar un DataFrame.**

Las columnas del archivo y sus significados son los siguientes:

- `genero_col`: Se refiere a si el colegio donde estudió la persona es mixto, femenino o masculino.
- `dpto`: Nombre del departamento donde habita la persona.
- `lectura_critica`: Puntaje obtenido por la persona en las pruebas Saber 11 en la categoría de lectura crítica.
- `matematicas`: Puntaje obtenido por la persona en las pruebas Saber 11 en la categoría de matemáticas.
- `ciencias`: Puntaje obtenido por la persona en las pruebas Saber 11 en la categoría de ciencias naturales.
- `sociales`: Puntaje obtenido por la persona en las pruebas Saber 11 en la categoría de sociales y ciudadanas.
- `ingles`: Puntaje obtenido por la persona en las pruebas Saber 11 en la categoría de inglés.
- `puntaje_global`: Puntaje total en las pruebas Saber 11, incluyendo todas las categorías.
- `internet`: Se refiere a si el estudiante cuenta con un servicio de internet en su casa (Sí: 1/ No: 0).
- `tv`: Se refiere a si el estudiante cuenta con un televisor en su hogar (Sí: 1/ No: 0).
- `computador`: Se refiere a si el estudiante cuenta con un computador en su hogar (Sí: 1/ No: 0).
- `lavadora`: Se refiere a si el estudiante cuenta con una lavadora en su hogar (Sí: 1/ No: 0).
- `microondas`: Se refiere a si el estudiante cuenta con un microondas en su hogar (Sí: 1/ No: 0).
- `carro`: Se refiere a si en el hogar del estudiante hay carro (Sí: 1/ No: 0).
- `moto`: Se refiere a si en el hogar del estudiante hay moto (Sí: 1/ No: 0).
- `videojuegos`: Se refiere a si el estudiante tiene una consola de videojuegos en su hogar (Sí: 1/ No: 0).

ATENCIÓN: cuando esté estudiando el problema y el archivo, recuerde que las funciones `describe()`, `unique()` y `filter()` pueden serle de utilidad. La función `describe()` aplicada sobre un DataFrame retorna información estadística de todas las columnas numéricas. La función `unique()` aplicada sobre una columna retorna una lista con los valores únicos que aparezcan en esa columna.

Parte 2: Estudiar la posesión de equipos/electrodomésticos/servicios en el hogar

En esta parte no se trabajará directamente sobre el DataFrame original, sino que se creará una matriz a partir de este. La matriz debe relacionar la posesión de equipos, electrodomésticos y servicios en el hogar y los departamentos del territorio nacional.

Requerimiento 1: Construcción de la matriz de Departamento vs Número de hogares que tienen un equipo/electrodoméstico/servicio determinado

Este requerimiento consiste en construir una matriz que cruce el departamento y el número de hogares que poseen equipos tecnológicos/electrodomésticos/servicios. La matriz (de tamaño 34x8) tiene la siguiente estructura:

72	732	200	...	556
256	1000	34	...	411
...
71	201	554	...	532

NOTA: La matriz mostrada anteriormente es un ejemplo y **NO** hace referencia a los valores que usted obtendrá al construir la matriz. Cada posición (f,c) en la matriz contiene el número de hogares que se encuentran en el departamento f y que tienen el equipo/electrodoméstico/servicio c (carro, computador, internet, lavadora, microondas, moto, tv y videojuegos). Por ejemplo, si Amazonas está en la fila 1 e internet en la columna 3, el número de hogares que cuentan con servicio de internet en el departamento de Amazonas sería el valor en la casilla (1,3).

Así mismo, se propone la construcción de 2 diccionarios para la referencia de las filas y las columnas. La llave de los diccionarios hace referencia al índice de la fila o columna de la matriz y el valor hace referencia al departamento o equipo/servicio/electrodoméstico correspondiente. A continuación, se muestran los diccionarios para las filas y las columnas.

Llave	Valor
0	"Amazonas"
1	"Antioquia"
2	"Arauca"
...	...
32	"Vichada"

Llave	Valor
0	"carro"
1	"computador"
...	...
7	"videojuegos"

NOTA: El código para la creación de los dos diccionarios ya se encuentra en el esqueleto proporcionado.

ATENCIÓN:

- Recuerde que tiene a su disposición la función `unique()` para obtener todos los valores únicos de una columna en un DataFrame.
- Para comprobar que los valores de la matriz creada son los correctos puede utilizar filtros, agrupaciones y sumas sobre el DataFrame original y verificar que estos arrojen el mismo resultado que el almacenado en su matriz.
- Puede ser de mucha utilidad pensar en descomponer el problema en varios problemas más pequeños.

Nota importante: Dado que el tamaño del DataFrame es bastante grande, la construcción de la matriz podría tardar más tiempo del que suelen tardar otras funciones. Si el tiempo supera los 2 minutos es muy probable que haya un problema en el código. En ese caso, le recomendamos que detenga el programa y revise si existe algún error en sus funciones.

La función para crear la matriz debe retornar una tupla que contenga la matriz creada y los diccionarios de filas y columnas en el siguiente formato (matriz, dict_columnas, dict_filas). Tenga en cuenta que usted deberá implementar los siguientes requerimientos utilizando como base la tupla anterior:

Requerimiento 2: Encontrar departamento con mayor cantidad de electrodomésticos

En este requerimiento se debe encontrar el departamento que ha registrado la mayor cantidad de electrodomésticos (lavadora y microondas). La función que implemente esta opción debe retornar el nombre del departamento con mayor número de electrodomésticos.

Requerimiento 3: Contar cantidad de estudiantes con un equipo/electrodoméstico/servicio dado en el país
En este requerimiento se debe calcular el número total de un servicio/electrodoméstico/equipo tecnológico en todo el país. Para esto, el usuario debe indicar el servicio/equipo tecnológico/electrodoméstico para el cual desea obtener la cantidad total. La función que implemente esta opción debe retornar un entero que represente la cantidad de la categoría ingresada en el país.

Requerimiento 4: Encontrar el Índice de Calidad de Vida (ICV) de un departamento dado
En este requerimiento se debe determinar el Índice de Calidad de Vida de un departamento. La función que implemente debe retornar un float redondeado a 2 cifras decimales que represente el ICV del departamento dado por el usuario. El ICV se calcula como un promedio ponderado, es decir, a la suma de multiplicar cada valor por un determinado peso, que se divide sobre la suma de los pesos. A continuación, se presenta una tabla que muestra el peso que representa cada equipo/servicio/electrodoméstico sobre el ICV del departamento:

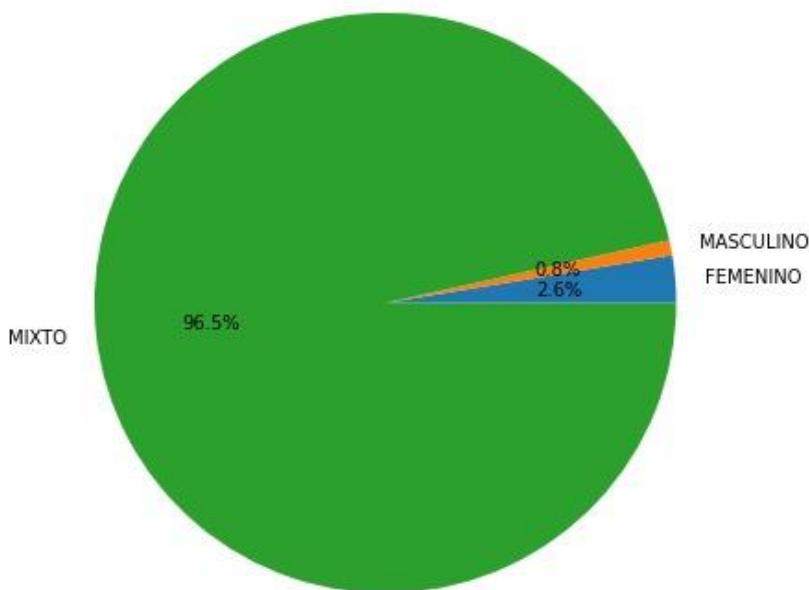
Equipo/servicio	Porcentaje
Carro	20%
Computador	9%
Internet	9%
Lavadora	7%
Microondas	8%
Moto	16%
TV	13%
Videojuegos	18%

Parte 3: Análisis de la distribución de los datos

Requerimiento 5: Distribución del género del colegio

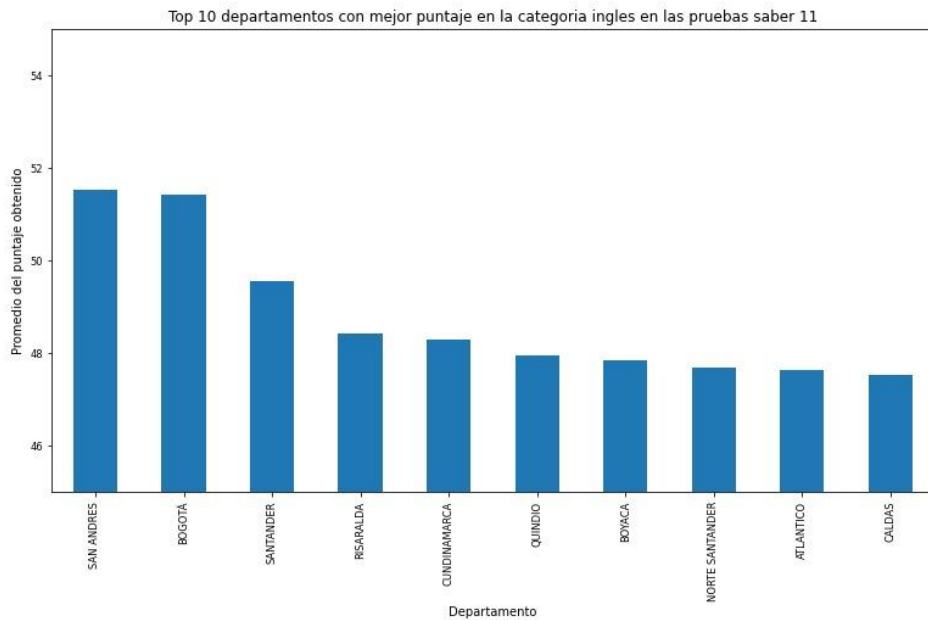
Para este requerimiento se quiere conocer la distribución de los estudiantes en los diferentes géneros de los colegios. La siguiente figura muestra la apariencia de la gráfica esperada.

Distribución porcentual según género del colegio



Requerimiento 6: Top 10 departamentos en una categoría específica de las pruebas Saber 11

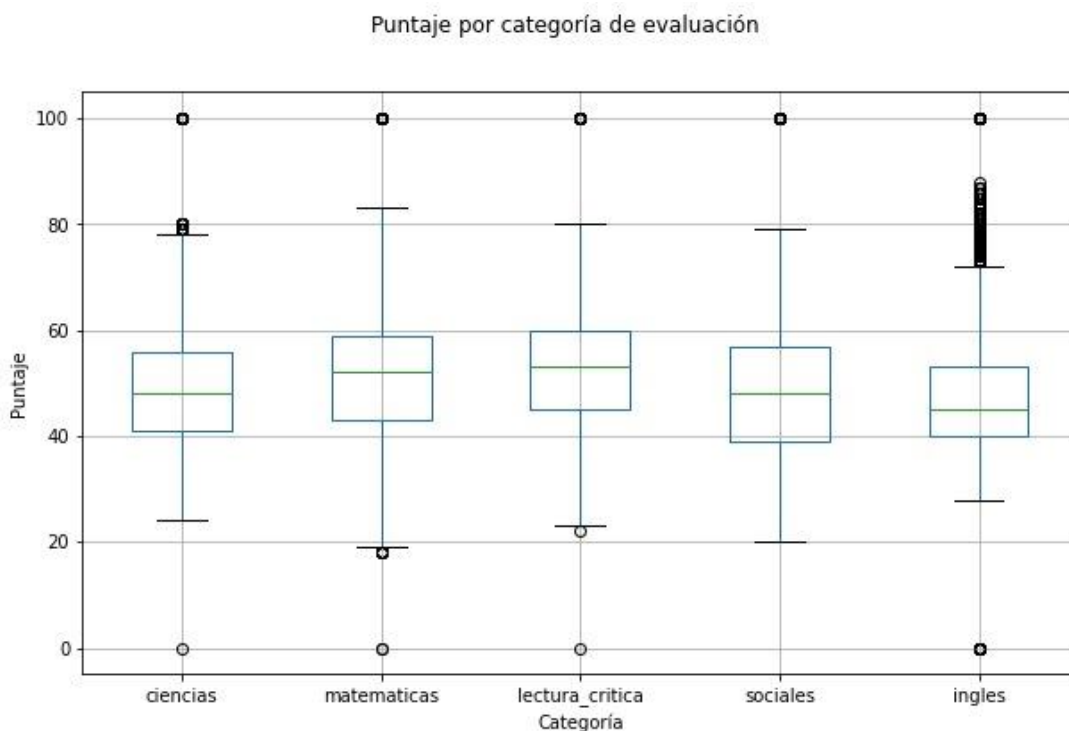
En este requerimiento se quiere conocer el top 10 departamentos que tuvieron mejor resultado en una categoría dada por el usuario en las pruebas Saber 11. Se debe mostrar en un diagrama de barras vertical como el que se muestra a continuación:



ATENCIÓN: Tenga en cuenta que para el ejemplo se usó la categoría inglés.

Requerimiento 7: Diagrama de caja y bigotes del puntaje por categoría de evaluación

En este requerimiento se quiere mostrar la distribución de los puntajes por categoría de las pruebas Saber 11. El diagrama debe verse como el que se muestra a continuación:



Actividad 1: Preparación del ambiente de trabajo

1. Cree una carpeta para trabajar, poniéndole su nombre o login.
2. Descargue de Bloque Neón el archivo “pruebas_saber11.zip” que contiene el archivo “.csv” con los datos a procesar y un archivo .py que debe usar como esqueleto para realizar el proyecto. Este archivo contiene el código para crear los diccionarios de las filas y columnas de la matriz que usted debe construir.
3. Abra Spyder y cambie la carpeta de trabajo para que sea la carpeta donde descargó el archivo con los datos.

Actividad 2: Construir el módulo de funciones

4. Usando Spyder, cree en su carpeta de trabajo un nuevo archivo con el nombre “saber11.py”. En este archivo usted va a construir el módulo en el que va a implementar las funciones que responden a los requerimientos de la aplicación. **Defina, documente e implemente** las funciones en su nuevo archivo. Usted puede crear cuántas funciones considere necesarias dentro de su librería o módulo. Mínimo debe haber una función por cada uno de los requerimientos del programa. Revise la consola entregada en el esqueleto para nombrar las funciones.

Actividad 3: Probar el correcto funcionamiento de su programa

5. **Ejecute el programa** y **pruebe** cada una de las funciones para asegurarse que esté funcionando. Puede probar el correcto funcionamiento de su programa cargando la información que se encuentra en el archivo “saber11.csv” o creando su propio archivo de prueba de menor tamaño (respetando el mismo formato) que le permita corroborar que los resultados arrojados por su programa son correctos.

Entrega

6. Comprima los dos archivos: saber11.py y consola_saber11.py en un solo archivo .zip. El archivo comprimido debe llamarse “**N4-PROY-login.zip**”, donde login es su nombre de usuario de Uniandes (omite el punto del login para evitar posibles problemas con la extensión de los archivos. Por ejemplo, si su login fuese “p.perez123”, nombre el archivo como “N4-PROY-pperez123.zip”).
7. Entregue el archivo comprimido a través de Bloque Neón en la actividad designada como “**Proyecto de Nivel 4**”.