

Proyecto POO SNIES-extractor

Tablero inteligente de visualización y consolidación de datos

Descripción General

El Sistema Nacional de Información de la Educación Superior (SNIES) en Colombia recopila una gran cantidad de datos sobre las instituciones de educación superior y los programas académicos que ofrecen. El análisis de estos datos, especialmente los relacionados con la inscripción, admisión, matrícula y graduación, es crucial para la toma de decisiones informadas en el ámbito educativo. Por ejemplo, el análisis de las tendencias en la matrícula y graduación puede revelar áreas de mejora en el diseño curricular de nuevos programas, asegurando que sean relevantes y atractivos para los estudiantes. Así mismo, el análisis de datos históricos y tendencias puede ayudar a las instituciones a tomar decisiones estratégicas sobre la apertura de nuevos programas, la modificación de programas existentes o la asignación de recursos.

En esta nueva versión del proyecto, se empleará **Streamlit** como herramienta de desarrollo en Python, permitiendo la creación de aplicaciones web interactivas para el procesamiento y análisis de la información del SNIES.

Carga de información

- El sistema debe operar con archivos en formato xlsx definidos en la carpeta inputs del proyecto. Cada archivo debe seguir el estándar "admitidos2021.xlsx", "matriculados2022.xlsx", según el tipo de dato que contenga (nombre de la información seguido por el año). Ustedes mismos pueden descargar los archivos a analizar de esta ruta: <https://snies.mineducacion.gov.co/portal/ESTADISTICAS/Bases-consolidadas/>. El proyecto debe tener los datos de los últimos cuatro años.
- La aplicación debe mostrarle al usuario los archivos que tiene disponibles y debe permitir la carga de nuevos archivos para el análisis. En ese caso el sistema debe permitir la carga del archivo, almacenarlo temporalmente y posteriormente considerarlo durante el análisis.

Filtrado de información

El sistema contará con un filtro de selección de programas académicos en dos pasos.

- **I. Búsqueda por palabras clave:** Los usuarios podrán ingresar una o varias palabras clave para identificar programas académicos relevantes dentro de la lista completa de programas disponibles en los archivos a analizar. El sistema mostrará aquellos programas cuyo nombre incluya las palabras clave especificadas, facilitando la preselección de opciones de interés. Para esta funcionalidad, se recomienda explorar componentes avanzados de **Streamlit** que optimicen la búsqueda y el filtrado. Además, es importante mantener una separación entre la vista y los datos, asegurando que la información se procese de forma estructurada y eficiente.

II. Selección de programas para el análisis: Una vez obtenidos los resultados de la búsqueda, el sistema presentará información adicional sobre cada programa, como la universidad donde se ofrece, el código snies del programa, el nivel de formación, la institución educativa padre y la seccional (si aplica). A partir de esta vista de resultados, el usuario podrá seleccionar aquellos programas específicos para incluir en el análisis final. Una forma de presentar esto es por ejemplo con una tabla que incluya checkbox de selección.

Den prioridad a la usabilidad en esta funcionalidad para facilitar el filtro y asegurar una navegación sencilla. La interfaz debe guiar al usuario, reduciendo la carga cognitiva y minimizando los clics. Por ejemplo, el uso de pestañas en Streamlit puede organizar estos pasos, permitiendo a los usuarios navegar fácilmente entre las etapas de selección y filtrado.

TIP: exploren qué soluciones se pueden hacer con Streamlit (ver Streamlit Gallery <https://streamlit.io/gallery>) para conocer sus posibilidades y limitaciones. Luego, diseñen la interfaz en una herramienta de prototipado rápido (como Mockflow o Draw.io) para definir cómo se verá y funcionará. El prototipado permite obtener una idea inicial y validar la usabilidad de la interfaz.

Procesamiento de Datos

El programa solicitará al usuario que ingrese el rango de inicial y final de años a analizar. Leerá los archivos de entrada correspondientes a los años especificados. Procesará los datos para calcular, por cada programa seleccionado y año:

- Estudiantes inscritos
- Estudiantes admitidos
- Estudiantes matriculados por primera vez
- Total de estudiantes matriculados
- Estudiantes graduados

El sistema mostrará la información en pantalla en un formato similar al siguiente:

Suma de Valor	Etiquetas																			
	2021										2022									
	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
Programa académico	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
Presencial	175	183	182	175	132	145	227	250	194	122	353	133	247	101	150	78	271	244	122	106
ESPECIALIZACIÓN EN ARQUITECTURA EMPRESARIAL																				
DE SOFTWARE																				
PONTIFICIA UNIVERSIDAD JAVERIANA	51	37	47	32	37	27	63	64	55	24	108	27	61	24	44	20	71	63	38	27
ESPECIALIZACIÓN EN CALIDAD DE SOFTWARE																				
INSTITUCIÓN UNIVERSITARIA EAF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ESPECIALIZACIÓN EN CONSTRUCCIÓN DE																				
UNIVERSIDAD DE NARIÑO	0	13	0	13	0	10	0	10	0	0	2	0	2	0	1	0	12	12	0	11
ESPECIALIZACIÓN EN DESARROLLO DE SOFTWARE																				
UNIVERSIDAD CATOLICA DE PEREIRA	16	0	14	0	13	0	23	14	0	12	0	0	0	0	0	0	0	0	0	0
UNIVERSIDAD DEL MAGDALENA - UNIMAGDALENA	10	12	2	12	2	8	12	8	10	18	0	0	0	0	0	0	8	2	1	10
ESPECIALIZACIÓN EN DESARROLLO DE																				
TECNOLOGIAS MOVILES																				
UNIVERSIDAD AUTONOMA DE BUCARAMANGA-UNAB-	5	5	5	5	0	0	0	0	0	0	12	0	12	0	0	0	0	0	0	0
ESPECIALIZACIÓN EN INGENIERIA DE SOFTWARE																				
INSTITUTO TECNOLÓGICO METROPOLITANO	0	0	0	0	0	0	0	0	0	0	0	29	0	0	0	0	0	0	0	44
PONTIFICIA UNIVERSIDAD JAVERIANA	2	8	2	8	3	9	6	10	14	9	22	10	21	9	11	6	13	15	1	4
UNIVERSIDAD ANTONIO NARIÑO	16	23	16	23	14	17	33	31	16	19	22	5	22	5	13	5	29	18	14	15
UNIVERSIDAD AUTONOMA DE MANIZALES	0	11	0	11	1	8	17	9	16	2	32	0	15	0	11	0	19	16	0	8
UNIVERSIDAD DE LOS LLANOS	11	1	13	1	12	1	12	20	2	3	14	7	11	6	10	6	16	20	14	0
UNIVERSIDAD DE MEDELLIN	9	7	9	7	8	3	10	13	20	2	17	7	17	7	12	7	13	21	8	2
UNIVERSIDAD DEL NORTE	6	0	6	0	0	0	0	11	0	81	0	62	0	17	0	17	17	0	1	6
UNIVERSIDAD EAFIT-	7	34	7	33	7	34	9	21	24	2	7	35	7	30	7	27	27	29	9	14
UNIVERSIDAD INCCA DE COLOMBIA	14	15	14	15	11	15	14	25	0	9	10	0	2	8	10	0	21	9	3	4
UNIVERSIDAD NACIONAL DE COLOMBIA	8	7	7	7	5	6	9	8	1	4	4	0	0	0	0	0	4	1	3	4
UNIVERSIDAD POPULAR DEL CESAR	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	18
UNIVERSIDAD SIMON BOLIVAR	9	2	9	0	10	0	10	10	12	0	12	5	8	5	7	0	7	7	8	0
ESPECIALIZACIÓN EN PROCESOS DE DESARROLLO																				
DE SOFTWARE																				
UNIVERSIDAD DE SAN BUENAVENTURA	11	8	11	8	9	7	9	7	13	0	10	8	7	7	7	7	14	14	9	6
Virtual	57	49	21	30	13	4	15	14	0	0	29	121	15	10	6	49	6	11	7	0
ESPECIALIZACIÓN EN DESARROLLO DE SOFTWARE																				
CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS-UNIMINUTO-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	303
ESPECIALIZACIÓN EN INGENIERIA DE SOFTWARE																				
FUNDACIÓN UNIVERSITARIA INTERNACIONAL DE LA RIOJA- UNIR	0	0	0	0	0	0	0	0	0	0	0	42	0	0	0	41	0	0	0	86
ESPECIALIZACIÓN EN PROGRAMACIÓN APLICADA																				
INSTITUCIÓN UNIVERSITARIA DIGITAL DE ANTIOQUIA-IU. DIGITAL	57	49	21	30	13	4	15	14	0	0	29	79	15	10	6	8	6	11	7	0

Además, permitirá descargar la información en formato xlsx, json y csv. Se recomienda que investiguen sobre la librería Pandas y los Dataframe para facilitar el procesamiento de los datos.

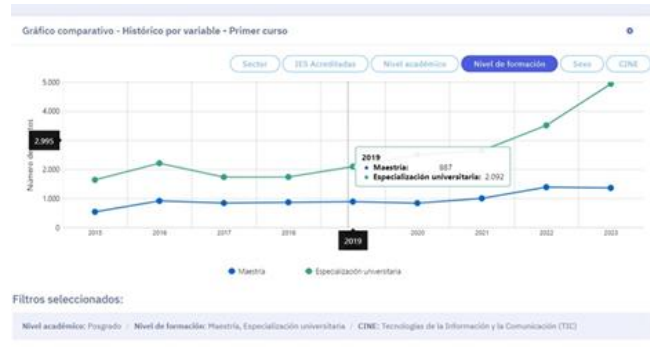
Visualización de información

El sistema deberá proporcionar visualizaciones de datos interactivas utilizando **Plotly** junto con **Streamlit**. Estas visualizaciones permitirán a los usuarios interpretar y comparar la información a través de gráficos dinámicos. **Por favor ajuste sus gráficos para manejar una escala de colores que haga la aplicación atractiva para el usuario.**

1. **Gráficos de líneas y barras para tendencias históricas:** Utilizando los gráficos de líneas y barras de Plotly, el sistema deberá mostrar las tendencias de inscripción, admisión, nuevos

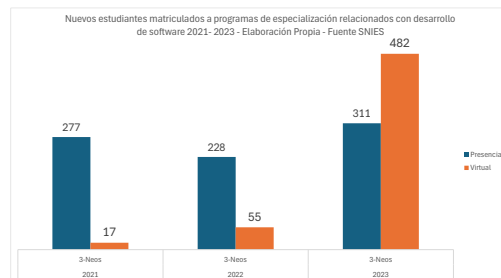
matriculados, matrícula total y graduación en los programas académicos seleccionados, segmentados por año. Al menos un gráfico por cada tipo de dato. Estos gráficos deberán ser interactivos, permitiendo al usuario hacer zoom en intervalos de tiempo y agregar o quitar programas del filtro.

Ejemplo



2. **Gráficos comparativos entre programas:** Implementar gráficos de barras que comparen las estadísticas. Estos gráficos permitirán a los usuarios visualizar los datos de las categoría analizadas (Ejm. inscritos, admitidos, nuevos matriculados, matriculados y graduados) por variables como :

A. Virtual / Presencial (ejemplo)



B. Género (Hombre/Mujer)

C. Nivel de formación (Pregrado, Posgrado, etc)

Requerimientos no funcionales

- Aplicar los principios **GRASP** (experto, creador, bajo acoplamiento, alta cohesión) en la estructura de clases para mejorar la modularidad y reducir el acoplamiento entre clases.
- **Manejo de excepciones:** El programa debe manejar adecuadamente los errores relacionados con la lectura de archivos, entradas inválidas y otros posibles fallos durante el proceso de consolidación de datos. Para ello debe lanzar y gestionar excepciones que se pueden mostrar en la interfaz gráfica de usuario.

- **Validaciones de datos:** Implementar validaciones sólidas para las entradas del usuario, como el rango de años, nombres de archivos de salida, y verificación de formatos de datos.
- Usen un **linter de Python** para mejorar el formato y estilo del código, aplicando los principios de **código limpio** para asegurar que el código sea fácil de leer y mantener.
- Un miembro del equipo debe asumir el rol de **Líder técnico**.
- Otro miembro del equipo asumirá el rol de **Calidad y Documentación**
- Otro miembro del equipo asumirá el rol de revisar los pull request, hacer los merge y gestionar los issues.
- Todos los miembros del equipo deben participar activamente en la codificación y ser responsables de diferentes partes del proyecto.
- Utilizar el esquema de ramas sugerido por GitHub Flow. Cada miembro del equipo debe crear una rama de trabajo por cada funcionalidad o mejora que desee implementar.
- No se deben eliminar las ramas al hacer merge con la rama principal, para facilitar la revisión posterior del trabajo realizado.
- Se espera que el proyecto tenga como mínimo 5 pull requests y 20 issues que reflejen la gestión de tareas en la ejecución del proyecto.
- Implementar **GitHub Actions** para configurar una integración continua (CI) que compile el proyecto automáticamente al realizar un commit o merge en la rama principal. Esto ayudará a verificar que el proyecto se mantenga funcional en todo momento.

Entregables

- **Video de Presentación del Proyecto:** Video de 6 minutos, donde se muestre el proyecto funcionando y se explique cómo se abordaron las mejoras que implementaron. Cada miembro del equipo debe participar y explicar su contribución. Subir link de video en los entregables del proyecto.
- Diagrama UML actualizado como Mermaid dentro del proyecto
- Código fuente del programa. Use este github classroom para iniciar el proyecto:
- README en el repositorio [Manual técnico]: Con presentación general del proyecto, imágenes y textos que muestren el cumplimiento de los requerimientos, diagrama de clases UML en formato .MERMAID incluido en el README(uno por equipo).
- Fechas de Entrega
- Entrega parcial: 13 de noviembre de 2024, que debe incluir avances importantes en las mejoras planteadas.
- Entrega final: 20 de noviembre de 2024, 11:59 pm, vía GitHub.
- Condiciones Adicionales
- Equipos de 3 personas.
- Todos los miembros deben practicar activamente la codificación.