Integrantes: *Maria Lucía Castillo García, Juliana González Sánchez y Ana Daniela Paredes Tovar*

1.1 Descripción General

En la Pontificia Universidad Javeriana de Cali, dentro el proceso de creación de nuevos programas de pregrado y posgrado, quienes están elaborando la propuesta de nuevos programas deben analizar a partir de los datos reportados en el SNIES la posible demanda a partir de los datos históricos que registra el SNIES. Recordando que el Sistema Nacional de Información de la Educación Superior (SNIES) en Colombia es quien se encarga de recopilar una gran cantidad de datos sobre las instituciones de educación superior y los programas académicos que ofrecen.

Dentro del análisis de estos datos, están 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.

Por lo que, el presente proyecto tiene como propósito desarrollar un programa en C++ que ayude a las personas encargadas a extraer y consolidar la información a realizar este proceso rápidamente sin tener que procesar los datos manualmente.

1.2 Descripción del Proyecto SNIES

En el desarrollo del proyecto, se implementaron diversas clases en C++. Entre ellas, la clase *Consolidado* que se encarga de manejar la consolidación de datos, permitiendo una estructura organizada para la información extraída. La clase de *ProgramaAcademico* que representan los programas académicos, integrando la lógica necesaria para gestionar sus características y relaciones.

```
class Consolidado
{
private:
    int idSexo;
    string sexo;
    int ano;
    int semestre;
    int inscritos;
    int admitidos;
    int matriculados;
    int matriculadosPrimerSemestre;
    int graduados;

public:
    Consolidado();
    Consolidado(int, string, int, int, int, int, int, int, int);

    int getIdSexo();
    void setIdSexo(int);

    string getSexo();
    void setSexo(string &);

    int getAno();
    void setAno(int);

    int getSemestre();
    void setSemestre(int);

    int getInscritos();
    void setInscritos(int);

    int getAdmitidos();
    void setAdmitidos(int);

    int getMatriculados();
    void setMatriculados(int);
```

Consolidado.h

```cpp
#ifndef PROGRAMA_ACADEMICO_H
#define PROGRAMA_ACADEMICO_H

#include <string>
#include <map>
#include <vector>
#include <iostream>
#include "Consolidado.h"
#include "Settings.h"

using namespace std;
class ProgramaAcademico
{
    int codigoDeLaInstitucion;
    int iesPadre;
    string institucionDeEducacionSuperiorIes;
    string principalOSeccional;
    int idSectorIes;
    string sectorIes;
    int idCaracter;
    string caracterIes;
    int codigoDelDepartamentoIes;
    string departamentoDeDomicilioDeLaIes;
    int codigoDelMunicipioIes;
    string municipioDeDomicilioDeLaIes;
    int codigoSniesDelPrograma;
    string programaAcademico;
    int idNivelAcademico;
    string nivelAcademico;
    int idNivelDeFormacion;
    string nivelDeFormacion;
    int idMetodologia;
    string metodologia;
    int idArea;
    string areaDeConocimiento;
    int idNucleo;
    string nucleoBasicoDelConocimientoNbc;
    int idCineCampoAmplio;
    string descCineCampoAmplio;
    int idCineCampoEspecifico;
    string descCineCampoEspecifico;
    int idCineCodigoDetallado;
    string descCineCodigoDetallado;
    int codigoDelDepartamentoPrograma;
    string departamentoDeOfertaDelPrograma;
    int codigoDelMunicipioPrograma;
    string municipioDeOfertaDelPrograma;
    vector<Consolidado *> consolidados;
```

```
public:
    ProgramaAcademico();

    void setCodigoDeLaInstitucion(int);
    int getCodigoDeLaInstitucion();

    void setIesPadre(int);
    int getIesPadre();

    void setInstitucionDeEducacionSuperiorIes(string &);
    string getInstitucionDeEducacionSuperiorIes();

    void setPrincipalOSeccional(string &);
    string getPrincipalOSeccional();

    void setIdSectorIes(int);
    int getIdSectorIes();

    void setSectorIes(string &);
    string getSectorIes();

    void setIdCaracter(int);
    int getIdCaracter();

    void setCaracterIes(string &);
    string getCaracterIes();
```

```
void setCodigoDelDepartamentoIes(int);
int getCodigoDelDepartamentoIes();

void setDepartamentoDeDomicilioDeLaIes(string &);
string getDepartamentoDeDomicilioDeLaIes();

void setCodigoDelMunicipioIes(int);
int getCodigoDelMunicipioIes();

void setMunicipioDeDomicilioDeLaIes(string &);
string getMunicipioDeDomicilioDeLaIes();

void setCodigoSniesDelPrograma(int);
int getCodigoSniesDelPrograma();

void setProgramaAcademico(string &);
string getProgramaAcademico();

void setIdNivelAcademico(int);
int getIdNivelAcademico();

void setNivelAcademico(string &);
string getNivelAcademico();

void setIdNivelDeFormacion(int);
int getIdNivelDeFormacion();
```

```
    void setNivelDeFormacion(string &);
    string getNivelDeFormacion();

    void setIdMetodologia(int);
    int getIdMetodologia();
    void setMetodologia(string &);
    string getMetodologia();

    void setIdArea(int);
    int getIdArea();
    void setAreaDeConocimiento(string &);
    string getAreaDeConocimiento();
    void setIdNucleo(int);
    int getIdNucleo();
    void setNucleoBasicoDelConocimientoNbc(string &);
    string getNucleoBasicoDelConocimientoNbc();
    void setIdCineCampoAmplio(int);
    int getIdCineCampoAmplio();
    void setDescCineCampoAmplio(string &);
    string getDescCineCampoAmplio();
    void setIdCineCampoEspecifico(int);
    int getIdCineCampoEspecifico();
    void setDescCineCampoEspecifico(string &);
    string getDescCineCampoEspecifico();
    void setIdCineCodigoDetallado(int);
    int getIdCineCodigoDetallado();
    void setDescCineCodigoDetallado(string &);
    string getDescCineCodigoDetallado();
    void setCodigoDelDepartamentoPrograma(int);
    int getCodigoDelDepartamentoPrograma();
    void setDepartamentoDeOfertaDelPrograma(string &);
    string getDepartamentoDeOfertaDelPrograma();
    void setCodigoDelMunicipioPrograma(int);
    int getCodigoDelMunicipioPrograma();
    void setMunicipioDeOfertaDelPrograma(string &);
    string getMunicipioDeOfertaDelPrograma();
    void setConsolidado(Consolidado *, int);
    Consolidado *getConsolidado(int);
    ~ProgramaAcademico();
};

#endif
```

ProgramaAcademico.h

Por otro lado, tenemos el *Gestor* que funciona como una clase base para manejar diferentes tipos de archivos. De ella derivan las clases hijas *GestorCSV*, *GestorTXT* y *GestorJSON*, que se especializan en la lectura y escritura de datos en diferentes formatos.

```h
#ifndef GESTOR_ARCHIVOS_H
#define GESTOR_ARCHIVOS_H

#include <string>
#include <map>
#include <list>
#include <vector>
#include "ProgramaAcademico.h"
#include "Consolidado.h"
#include "Settings.h"

using namespace std;

class Gestor {
public:
    virtual bool crearArchivo(string &ruta, map<int, ProgramaAcademico *> &mapadeProgramasAcademicos, vector<string> etiquetasColumnas) = 0;
    virtual bool crearArchivoBuscados(string &ruta, list<ProgramaAcademico *> &programasBuscados, vector<string> etiquetasColumnas) = 0;
    virtual bool crearArchivoExtra(std::string &ruta, vector<vector<string>> datosAImprimir) = 0;
    virtual ~Gestor() = default;
};

#endif
```

Gestor.h

```h
#ifndef GESTOR_CSV_H
#define GESTOR_CSV_H

#include <string>
#include <vector>
#include <map>
#include <fstream>
#include <sstream>
#include <list>
#include <algorithm>
#include "Gestor.h"


using namespace std;

class GestorCsv : public Gestor
{
public:
    GestorCsv() = default;
    vector<int> leerProgramasCsv(string &ruta);
    vector<vector<string>> leerArchivoPrimera(string &rutaBase, string &ano, vector<int> &codigosSnies);
    vector<vector<string>> leerArchivoSegunda(string &rutaBase, string &ano, vector<int> &codigosSnies);
    vector<vector<string>> leerArchivo(string &rutaBase, string &ano, vector<int> &codigosSnies, int colmunaCodigoSnies);
    bool crearArchivo(string &ruta, map<int, ProgramaAcademico *> &mapadeProgramasAcademicos, vector<string> etiquetasColumnas) override;
    bool crearArchivoBuscados(string &ruta, list<ProgramaAcademico *> &programasBuscados, vector<string> etiquetasColumnas) override;
    bool crearArchivoExtra(string &ruta, vector<vector<string>> datosAImprimir) override;
};

#endif
```

GestorCsv.h

```h
#ifndef GESTOR_JSON_H
#define GESTOR_JSON_H

#include "Gestor.h"
#include <fstream>

using namespace std;

class GestorJson : public Gestor
{
public:
    GestorJson() = default;
    bool crearArchivo(string &ruta, map<int, ProgramaAcademico *> &mapadeProgramasAcademicos, vector<string> etiquetasColumnas) override;
    bool crearArchivoBuscados(string &ruta, list<ProgramaAcademico *> &programasBuscados, vector<string> etiquetasColumnas) override;
    bool crearArchivoExtra(string &ruta, vector<vector<string>> datosAImprimir) override;
};

#endif
```

GestorJson.h

```cpp
#ifndef GESTOR_TXT_H
#define GESTOR_TXT_H

#include "Gestor.h"
#include <fstream>

using namespace std;

class GestorTxt : public Gestor
{
public:
    GestorTxt() = default;
    bool crearArchivo(string &ruta, map<int, ProgramaAcademico *> &mapadeProgramasAcademicos, vector<string> etiquetasColumnas) override;
    bool crearArchivoBuscados(string &ruta, list<ProgramaAcademico *> &programasBuscados, vector<string> etiquetasColumnas) override;
    bool crearArchivoExtra(string &ruta, vector<vector<string>> datosAImprimir) override;
};

#endif
```

GestorTxt.h

Además, tenemos el *View* que permite visualizar los resultados de los análisis, proporcionando una interfaz amigable para los usuarios. Y el *Settings* que lo utilizamos para gestionar constantes, evitando así el uso de "números mágicos" en el código, lo que mejora su legibilidad y mantenibilidad.

```cpp
#ifndef VIEW_H
#define VIEW_H
#include "SNIESController.h"
/*Todo el codigo va entre estas guardas*/
#include "SNIESController.h"
#include <cctype>

using namespace std;

class View
{
private:
    SNIESController controlador;
    bool validarEntrada(string&, bool);
    void organizarAnios(string &, string &);
    bool validarEntradaYN();

public:
    View();
    ~View();
    bool mostrarPantallaBienvenido();
    void mostrarDatosExtra();
    void buscarPorPalabraClaveYFormacion();
    void salir();
    bool isConvetibleToInt(const string &);



};

#endif
```

View.h

```h
#ifndef SETTINGS_H
#define SETTINGS_H

#include <string>

class Settings
{
public:
    // Declaración de las variables estáticas como constantes.
    static const std::string ADMITIDOS_FILE_PATH;
    static const std::string MATRICULADOS_FILE_PATH;
    static const std::string INSCRITOS_FILE_PATH;
    static const std::string PROGRAMAS_FILTRAR_FILE_PATH;
    static const std::string BASE_PATH;
    static const char DELIMITADOR;
    static const int COLUMNAS_INFO_CONSOLIDADOS;
    static const int DATOS_ACADEM_DEMOGRAF;
    static const int COLUMNA_12;
    static const int COLUMNA_13;
    static const int FILAS_RESTANTES;

};

#endif // SETTINGS_H
```

Settings.h

Ya por último tenemos el *SNIESController* quien se encarga de controlar la interacción con el SNIES, facilitando la conexión y la gestión de datos necesarios para el análisis.

```h
class SNIESController
{

private:
    map<int, ProgramaAcademico *> programasAcademicos;
    GestorCsv gestorCsvObj;
    vector<string> etiquetasColumnas;
    string rutaProgramasCSV;
    string rutaAdmitidos;
    string rutaGraduados;
    string rutaInscritos;
    string rutaMatriculados;
    string rutaMatriculadosPrimerSemestre;
    string rutaOutput;

public:
    SNIESController() = default;
    SNIESController(string &, string &, string &, string &, string &, string &, string &);
    ~SNIESController();
    void procesarDatosCsv(string &, string &);
    void calcularDatosExtra(bool);
    void buscarProgramas(bool, string &, int);

};

#endif
```

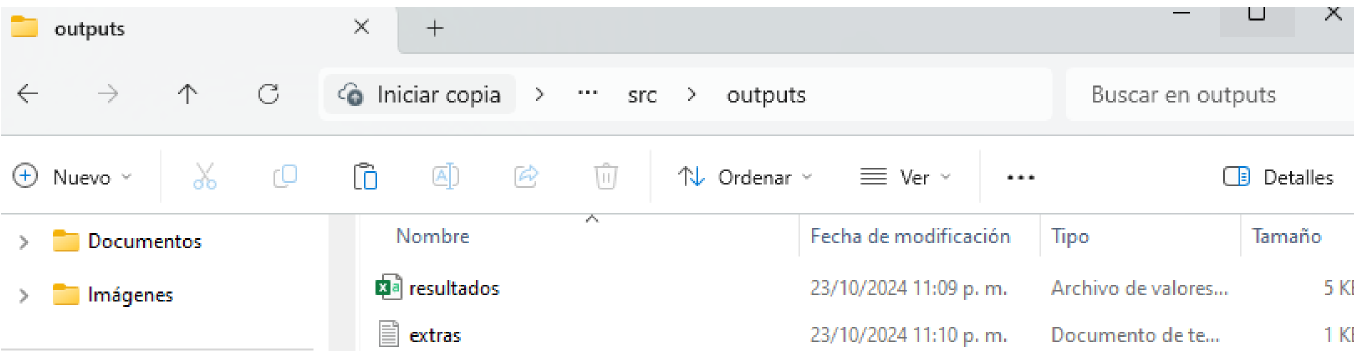
Controller.h

A través de estas clases, el proyecto buscó optimizar el proceso de análisis y decisión en la creación de nuevos programas académicos, contribuyendo a la mejora continua en la educación superior.

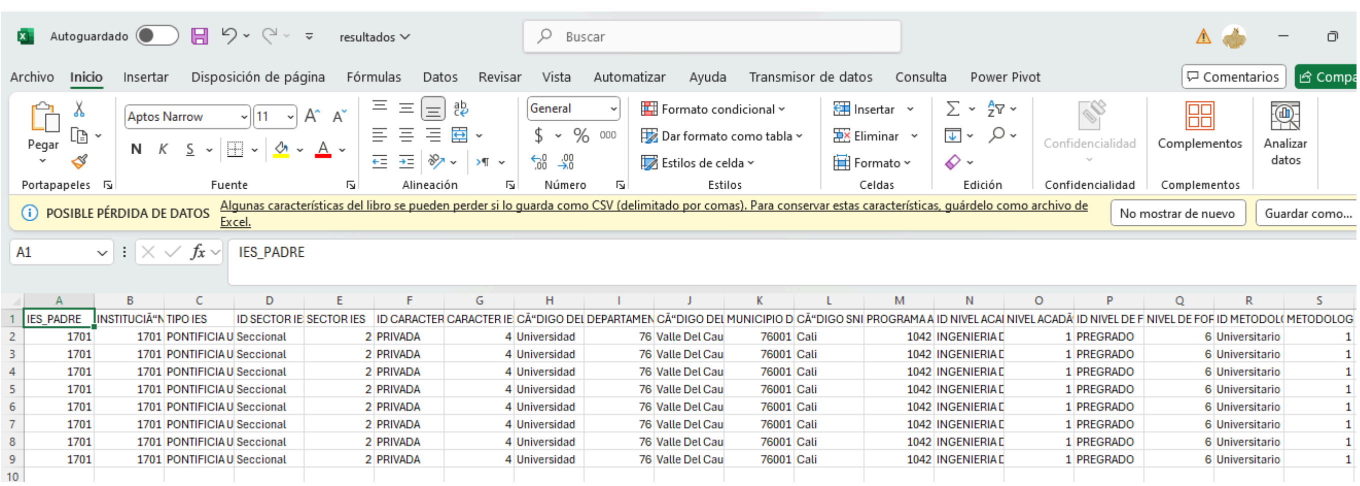
---

2. Funcionamiento del proyecto

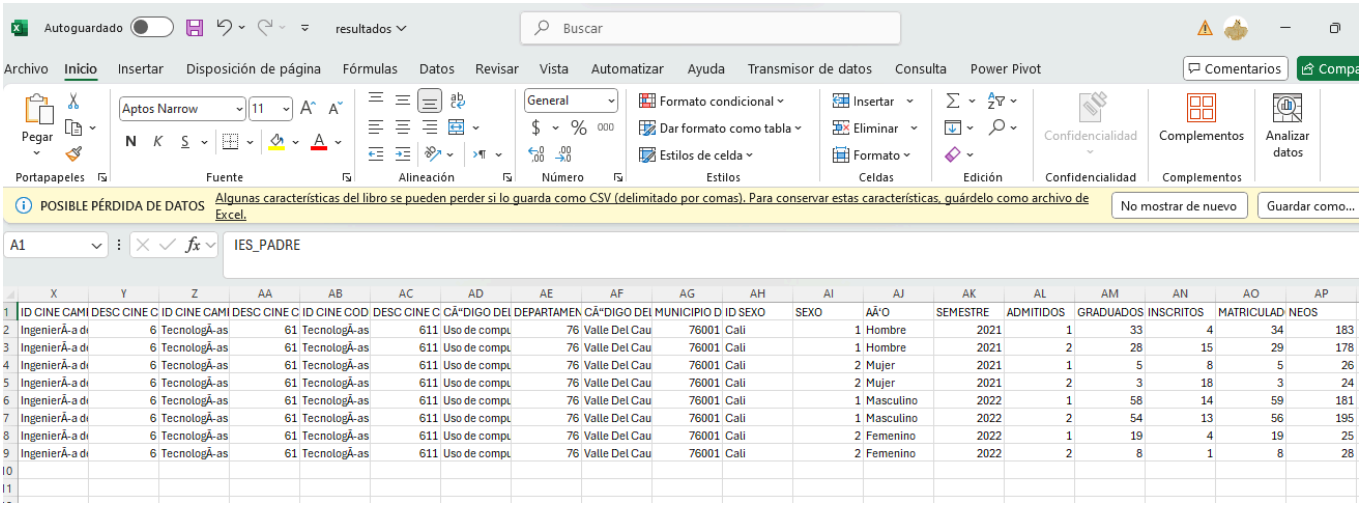
```
C:\Users\casti\Documents\proyecto-2-snies-extractor-4-beibis-1\src> programaSNIES.exe
Bienvenido al SNIES-Extractor!
=========================================
Recuerde que para el correcto funcionamiento del programa tuvo que haber parametrizado
Antes la carpeta SNIES_EXTRACTOR en el disco duro C:, con sus respectivas carpetas inputs y outputs
y todos los archivo CSV del SNIES.
Si ya hizo esto, escriba 'Y', de lo contrario 'N', y presione Enter: Y
A continuaci├│n se procesar├ín los datos de los programas acad├®micos seleccionados en /programas.csv...
Escriba el primer a├▒o de b├║squeda: 2021

Escriba el segundo a├▒o de b├║squeda: 2022

Procesando datos ...
Desea generar un archivo CSV (1), Desea generar un archivo TXT (2) o Desea generar un archivo JSON (3)
1
Archivo Creado en: C:/Users/casti/Documents/proyecto-2-snies-extractor-4-beibis-1/src/outputs/resultados.csv
Datos procesados con ├®xito!
A continuaci├│n vamos a mostrar datos relevantes de los programas acad├®micos seleccionados

Desea Convertir los datos a un archivo CSV, TXT O JSON?(Y/N):
2
Entrada inv├ílida. Por favor ingrese 'Y' o 'N':
Y
Suma Estudiantes Matriculados de Programas Seleccionados (Presencial o Virtual) Primer a├▒o;Suma Estudiantes Matriculados de Programas Seleccionados
 (Presencial o Virtual)
411;429
Codigo Snies;Nombre del Programa;Nombre del Institucion;Diferencial porcentual anual de NEOS
1042;INGENIERIA DE SISTEMAS Y COMPUTACION;PONTIFICIA UNIVERSIDAD JAVERIANA;49
Codigo Snies; Nombre del Programa sin NEOS en los ultimos 3 semestres
Desea generar un archivo CSV (1), Desea generar un archivo TXT (2) o Desea generar un archivo JSON (3)
2
```



RESULTADOS:

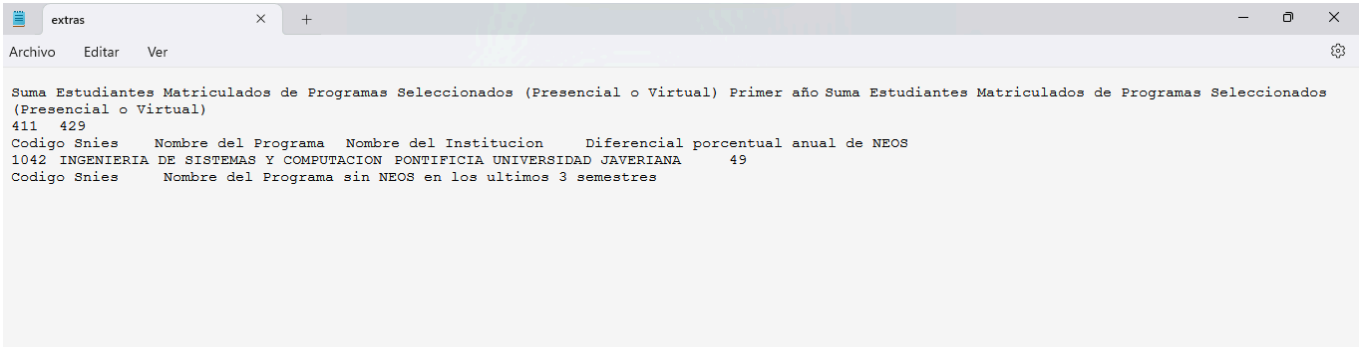


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	IES_PADRE	INSTITUCIÃ“N	TIPO IES	ID SECTOR IE	SECTOR IES	ID CARACTER	CARACTER IE	CÃ“DIGO DEL	DEPARTAMEN	CÃ“DIGO DEL	MUNICIPIO D	CÃ“DIGO SNI	PROGRAMA A	ID NIVEL ACA	NIVEL ACADÃ	ID NIVEL DE F	NIVEL DE FOR	ID METODOL	METODOLOG
2	1701	1701	PONTIFICIA U	Seccional	2	PRIVADA	4	Universidad	76	Valle Del Cau	76001	Cali	1042	INGENIERIA D	1	PREGRADO	6	Universitario	1
3	1701	1701	PONTIFICIA U	Seccional	2	PRIVADA	4	Universidad	76	Valle Del Cau	76001	Cali	1042	INGENIERIA D	1	PREGRADO	6	Universitario	1
4	1701	1701	PONTIFICIA U	Seccional	2	PRIVADA	4	Universidad	76	Valle Del Cau	76001	Cali	1042	INGENIERIA D	1	PREGRADO	6	Universitario	1
5	1701	1701	PONTIFICIA U	Seccional	2	PRIVADA	4	Universidad	76	Valle Del Cau	76001	Cali	1042	INGENIERIA D	1	PREGRADO	6	Universitario	1
6	1701	1701	PONTIFICIA U	Seccional	2	PRIVADA	4	Universidad	76	Valle Del Cau	76001	Cali	1042	INGENIERIA D	1	PREGRADO	6	Universitario	1
7	1701	1701	PONTIFICIA U	Seccional	2	PRIVADA	4	Universidad	76	Valle Del Cau	76001	Cali	1042	INGENIERIA D	1	PREGRADO	6	Universitario	1
8	1701	1701	PONTIFICIA U	Seccional	2	PRIVADA	4	Universidad	76	Valle Del Cau	76001	Cali	1042	INGENIERIA D	1	PREGRADO	6	Universitario	1
9	1701	1701	PONTIFICIA U	Seccional	2	PRIVADA	4	Universidad	76	Valle Del Cau	76001	Cali	1042	INGENIERIA D	1	PREGRADO	6	Universitario	1
10																			

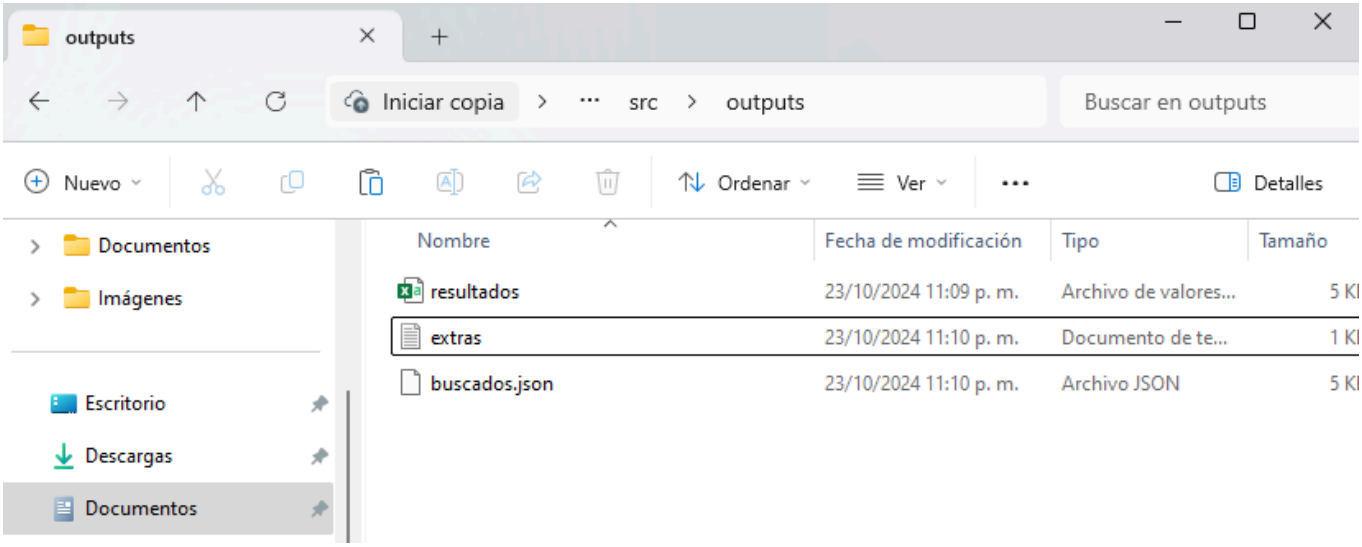


	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP
1	ID CINE CAMI	DESC CINE C	ID CINE CAMI	DESC CINE C	ID CINE COD	DESC CINE C	CÃ“DIGO DEL	DEPARTAMEN	CÃ“DIGO DEL	MUNICIPIO D	ID SEXO	SEXO	AÃ‘O	SEMESTRE	ADMITIDOS	GRADUADOS	INSCRITOS	MATRICULAD	NEOS
2	IngenierÃ-a d	6	TecnologÃ-as	61	TecnologÃ-as	611	Uso de compu	76	Valle Del Cau	76001	Cali	1	Hombre	2021	1	33	4	34	183
3	IngenierÃ-a d	6	TecnologÃ-as	61	TecnologÃ-as	611	Uso de compu	76	Valle Del Cau	76001	Cali	1	Hombre	2021	2	28	15	29	178
4	IngenierÃ-a d	6	TecnologÃ-as	61	TecnologÃ-as	611	Uso de compu	76	Valle Del Cau	76001	Cali	2	Mujer	2021	1	5	8	5	26
5	IngenierÃ-a d	6	TecnologÃ-as	61	TecnologÃ-as	611	Uso de compu	76	Valle Del Cau	76001	Cali	2	Mujer	2021	2	3	18	3	24
6	IngenierÃ-a d	6	TecnologÃ-as	61	TecnologÃ-as	611	Uso de compu	76	Valle Del Cau	76001	Cali	1	Masculino	2022	1	58	14	59	181
7	IngenierÃ-a d	6	TecnologÃ-as	61	TecnologÃ-as	611	Uso de compu	76	Valle Del Cau	76001	Cali	1	Masculino	2022	2	54	13	56	195
8	IngenierÃ-a d	6	TecnologÃ-as	61	TecnologÃ-as	611	Uso de compu	76	Valle Del Cau	76001	Cali	2	Femenino	2022	1	19	4	19	25
9	IngenierÃ-a d	6	TecnologÃ-as	61	TecnologÃ-as	611	Uso de compu	76	Valle Del Cau	76001	Cali	2	Femenino	2022	2	8	1	8	28
10																			
11																			

EXTRAS:




```
Desea generar un archivo CSV (1), Desea generar un archivo TXT (2) o Desea generar un archivo JSON (3)
2
Archivo Creado en: C:/Users/casti/Documents/proyecto-2-snies-extractor-4-beibis-1/src/outputs/extras.txt
Desea hacer una b├╣squeda por palabra clave del nombre del programa(Y/N):
Y

Deseas convertir convertir los datos del programa acad├@mico a un CSV?(Y/N):
Y

Escriba la palabra clave para buscar los programas por nombre:
SISTEMAS

Seleccione un nivel de formaci├│n para filtrar:
 1->Especializacion Universitaria
 2->Maestria
 3->Doctorado
 4->Formacion Tecnica Profesional
 5->Tecnologico
 6->Universitario
 7->Especializacion Tecnico Profesional
 8->Especializacion Tecnologica
 10->Especializacion Medico Quirurgica

6

1042;INGENIERIA DE SISTEMAS Y COMPUTACION;1701;PONTIFICIA UNIVERSIDAD JAVERIANA;Presencial
Desea generar un archivo CSV (1), Desea generar un archivo TXT (2) o Desea generar un archivo JSON (3)
3
Archivo creado en: C:/Users/casti/Documents/proyecto-2-snies-extractor-4-beibis-1/src/outputs/buscados.json
Desea hacer una b├╣squeda por palabra clave del nombre del programa(Y/N):
N
```


outputs
Iniciar copia
src
outputs
Buscar en outputs
Nuevo
Ordenar
Ver
Detalles
Documentos
Imágenes
Escritorio
Descargas
Documentos
Nombre
Fecha de modificación
Tipo
Tamaño
resultados
23/10/2024 11:09 p. m.
Archivo de valores...
5 KI
extras
23/10/2024 11:10 p. m.
Documento de te...
1 KI
buscados.json
23/10/2024 11:10 p. m.
Archivo JSON
5 KI

extras

buscados.json

Archivo Editar Ver


```
[
    {
        "codigo_de_la_institucion": "1701",
        "ies_padre": "1701",
        "institucion_de_educacion_superior": "PONTIFICIA UNIVERSIDAD JAVERIANA",
        "principal_o_seccional": "Seccional",
        "id_sector": "2",
        "sector": "PRIVADA",
        "id_caracter": "4",
        "caracter": "Universidad",
        "codigo_departamento": "76",
        "departamento_domicilio": "Valle Del Cauca",
        "codigo_municipio": "76001",
        "municipio_domicilio": "Cali",
        "codigo_snies_programa": "1042",
        "programa_academico": "INGENIERIA DE SISTEMAS Y COMPUTACION",
        "id_nivel_academico": "1",
        "nivel_academico": "PREGRADO",
        "id_nivel_formacion": "6",
        "nivel_formacion": "Universitario",
        "id_metodologia": "1",
        "metodologia": "Presencial",
        "id_area": "8",
        "area_de_conocimiento": "Ingeniería, arquitectura, urbanismo y afines",
        "id_nucleo": "827",
        "nucleo_basico": "Ingeniería de sistemas, telemática y afines",
        "id_cine_campo_amplio": "6",
        "desc_cine_campo_amplio": "Tecnologías de la Información y la Comunicación (TIC)",
        "id_cine_campo_especifico": "61",
        "desc_cine_campo_especifico": "Tecnologías de la Información y la Comunicación (TIC)",
        "id_cine_codigo_detallado": "611",
        "desc_cine_codigo_detallado": "Uso de computadores",
        "codigo_departamento_programa": "76",
        "departamento_oferta_programa": "Valle Del Cauca",
        "codigo_municipio_programa": "76001",
        "municipio_oferta_programa": "Cali",
```


Ln 1, Col 1 100%


```
        "codigo_municipio_programa": "76001",
        "municipio_oferta_programa": "Cali",
        "consolidado_0": {
            "id_sexo": "1",
            "sexo": "Hombre",
            "ano": "2021",
            "semestre": "1",
            "admitidos": "33",
            "graduados": "4",
            "inscritos": "34",
            "matriculados": "183",
            "matriculados_primer_semestre": "31"
        },
        "consolidado_1": {
            "id_sexo": "1",
            "sexo": "Hombre",
            "ano": "2021",
            "semestre": "2",
            "admitidos": "28",
            "graduados": "15",
            "inscritos": "29",
            "matriculados": "178",
            "matriculados_primer_semestre": "21"
        },
        "consolidado_2": {
            "id_sexo": "2",
            "sexo": "Mujer",
            "ano": "2021",
            "semestre": "1",
            "admitidos": "5",
            "graduados": "8",
            "inscritos": "5",
            "matriculados": "26",
            "matriculados_primer_semestre": "4"
        },
        "consolidado_3": {
            "id_sexo": "2",
```

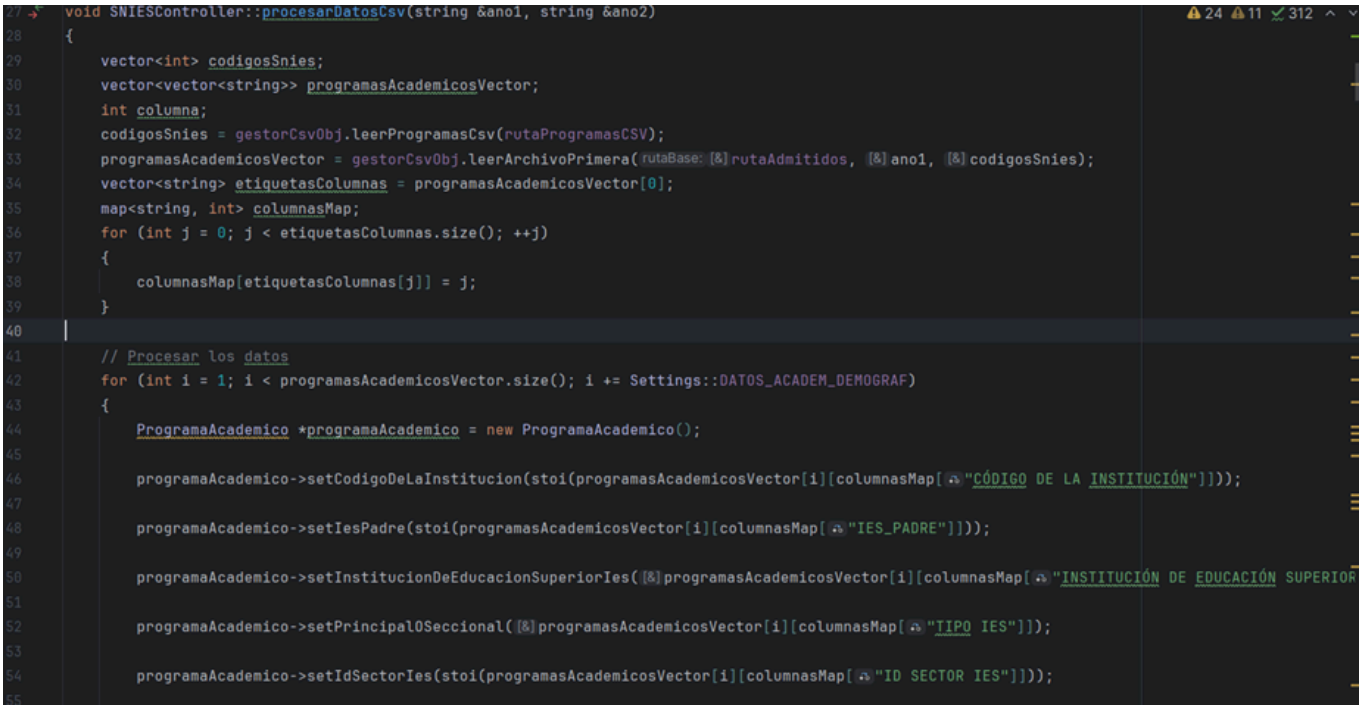
```
        "consolidado_3": {
            "id_sexo": "2",
            "sexo": "Mujer",
            "ano": "2021",
            "semestre": "2",
            "admitidos": "3",
            "graduados": "18",
            "inscritos": "3",
            "matriculados": "24",
            "matriculados_primer_semestre": "3"
        },
        "consolidado_4": {
            "id_sexo": "1",
            "sexo": "Masculino",
            "ano": "2022",
            "semestre": "1",
            "admitidos": "58",
            "graduados": "14",
            "inscritos": "59",
            "matriculados": "181",
            "matriculados_primer_semestre": "29"
        },
        "consolidado_5": {
            "id_sexo": "1",
            "sexo": "Masculino",
            "ano": "2022",
            "semestre": "2",
            "admitidos": "54",
            "graduados": "13",
            "inscritos": "56",
            "matriculados": "195",
            "matriculados_primer_semestre": "44"
        },
        "consolidado_6": {
            "id_sexo": "2",
            "sexo": "Femenino",
            "ano": "2022",
```

```
        },
        "consolidado_6": {
            "id_sexo": "2",
            "sexo": "Femenino",
            "ano": "2022",
            "semestre": "1",
            "admitidos": "19",
            "graduados": "4",
            "inscritos": "19",
            "matriculados": "25",
            "matriculados_primer_semestre": "9"
        },
        "consolidado_7": {
            "id_sexo": "2",
            "sexo": "Femenino",
            "ano": "2022",
            "semestre": "2",
            "admitidos": "8",
            "graduados": "1",
            "inscritos": "8",
            "matriculados": "28",
            "matriculados_primer_semestre": "6"
        }
    }
]
```

2.1 Cumplimiento de los requerimientos

- **Búsqueda de columnas importantes

```cpp
vector<vector<string>> GestorCsv::leerArchivoPrimera(string &rutaBase, string &ano, vector<int> &codigosSnies)
{
    vector<vector<string>> matrizResultado;
    string rutaCompleta = rutaBase + ano + ".csv";
    ifstream archivoPrimero(rutaCompleta);
    if (!(archivoPrimero.is_open()))
    {
        cout << "Archivo " << rutaCompleta << " no se pudo abrir correctamente" << endl;
    }
    else
    {
        string fila;
        string dato;
        vector<string> vectorFila;
        stringstream streamFila;
        vector<int> posicionesInteres;
        vector<int>::iterator it;

        getline([&]archivoPrimero, [&]fila);
        streamFila = stringstream(fila);
        int indiceColumna = 0;

        while (getline([&]streamFila, [&]dato, delim:Settings:: DELIMITADOR))
        {
            if (dato == "CÓDIGO DE LA INSTITUCIÓN" ||
                dato == "IES_PADRE" ||
                dato == "INSTITUCIÓN DE EDUCACIÓN SUPERIOR (IES)" ||
                dato == "TIPO IES" ||
```



```
void SNIESController::procesarDatosCsv(string &ano1, string &ano2)
{
    vector<int> codigosSnies;
    vector<vector<string>> programasAcademicosVector;
    int columna;
    codigosSnies = gestorCsvObj.leerProgramasCsv(rutaProgramasCSV);
    programasAcademicosVector = gestorCsvObj.leerArchivoPrimera(rutaBase: &rutaAdmitidos, &ano1, &codigosSnies);
    vector<string> etiquetasColumnas = programasAcademicosVector[0];
    map<string, int> columnasMap;
    for (int j = 0; j < etiquetasColumnas.size(); ++j)
    {
        columnasMap[etiquetasColumnas[j]] = j;
    }

    // Procesar los datos
    for (int i = 1; i < programasAcademicosVector.size(); i += Settings::DATOS_ACADEM_DEMOGRAF)
    {
        ProgramaAcademico *programaAcademico = new ProgramaAcademico();

        programaAcademico->setCodigoDeLaInstitucion(stoi(programasAcademicosVector[i][columnasMap["CÓDIGO DE LA INSTITUCIÓN"]]));

        programaAcademico->setIesPadre(stoi(programasAcademicosVector[i][columnasMap["IES_PADRE"]]));

        programaAcademico->setInstitucionDeEducacionSuperiorIes(&programasAcademicosVector[i][columnasMap["INSTITUCIÓN DE EDUCACIÓN SUPERIOR

        programaAcademico->setPrincipalOSeccional(&programasAcademicosVector[i][columnasMap["TIPO IES"]]);

        programaAcademico->setIdSectorIes(stoi(programasAcademicosVector[i][columnasMap["ID SECTOR IES"]]));
```

- **Delimitador y eliminación de números mágicos

```
#include "Settings.h"

// Inicialización de las variables estáticas fuera de la clase.

  const std::string Settings::BASE_PATH = "C:/SNIES_EXTRACTOR/inputs/";
  const std::string Settings::PROGRAMAS_FILTRAR_FILE_PATH = Settings::BASE_PATH + "programas.csv";
  const std::string Settings::ADMITIDOS_FILE_PATH = Settings::BASE_PATH + "admitidos";
  const std::string Settings::MATRICULADOS_FILE_PATH = Settings::BASE_PATH + "matriculados";
  const std::string Settings::INSCRITOS_FILE_PATH = Settings::BASE_PATH + "inscritos";
  const char Settings::DELIMITADOR = ';';
  const int Settings::COLUMNAS_INFO_CONSOLIDADOS = 8;
  const int Settings::DATOS_ACADEM_DEMOGRAF = 4;
  const int Settings::COLUMNA_13 = 13;
  const int Settings::COLUMNA_12 = 12;
  const int Settings::FILAS_RESTANTES = 3;
```

- **Generar archivos de salida en formato *CSV,* TXT *y* JSON*, aplicando polimorfismo y herencia para gestionar los diferentes formatos de salida.**
  Gestor:

```cpp
#ifndef GESTOR_ARCHIVOS_H
#define GESTOR_ARCHIVOS_H

#include ...

using namespace std;

class Gestor {
public:
    virtual bool crearArchivo(string &ruta, map<int, ProgramaAcademico *> &mapadeProgramasAcademicos, vector<string> &etiquetasColumnas) = 0;
    virtual bool crearArchivoBuscados(string &ruta, list<ProgramaAcademico *> &programasBuscados, vector<string> &etiquetasColumnas) = 0;
    virtual bool crearArchivoExtra(std::string &ruta, vector<vector<string>> &datosAImprimir) = 0;
    virtual ~Gestor() = default;
};

#endif #ifndef GESTOR_ARCHIVOS_H
```

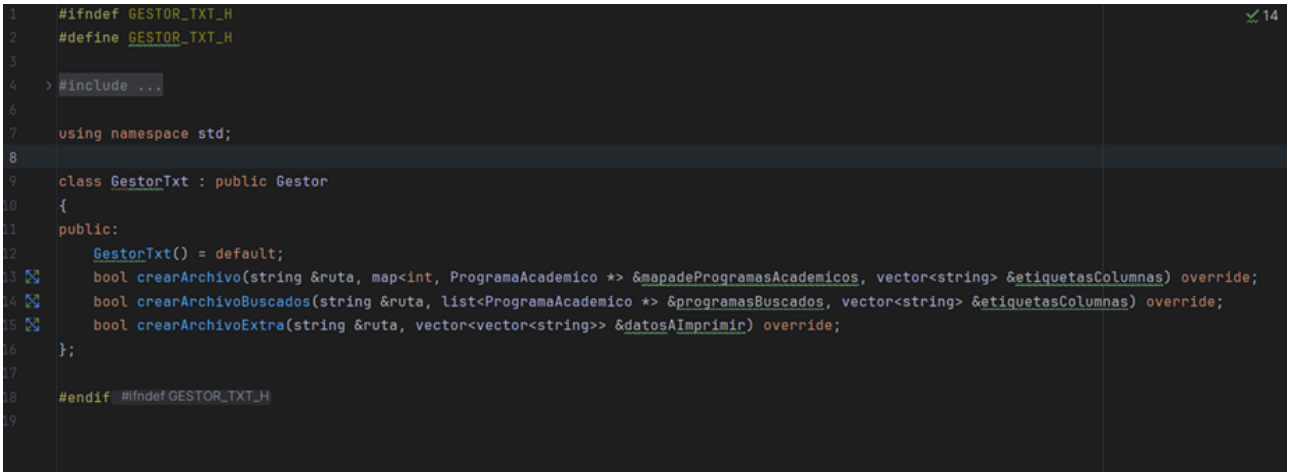
GestorCsv:

```cpp
#ifndef GESTOR_CSV_H
#define GESTOR_CSV_H

#include ...


using namespace std;

class GestorCsv : public Gestor
{
public:
    GestorCsv() = default;
    vector<int> leerProgramasCsv(const string &ruta);
    vector<vector<string>> leerArchivoPrimera(string &rutaBase, string &ano, vector<int> &codigosSnies);
    vector<vector<string>> leerArchivoSegunda(string &rutaBase, string &ano, vector<int> &codigosSnies);
    vector<vector<string>> leerArchivo(string &rutaBase, string &ano, vector<int> &codigosSnies, int colmunaCodigoSnies);
    bool crearArchivo(string &ruta, map<int, ProgramaAcademico *> &mapadeProgramasAcademicos, vector<string> &etiquetasColumnas) override;
    bool crearArchivoBuscados(string &ruta, list<ProgramaAcademico *> &programasBuscados, vector<string> &etiquetasColumnas) override;
    bool crearArchivoExtra(string &ruta, vector<vector<string>> &datosAImprimir) override;
};

#endif #ifndef GESTOR_CSV_H
```

GestorJson:

```cpp
#ifndef GESTOR_JSON_H
#define GESTOR_JSON_H

#include "Gestor.h"
#include <fstream>

using namespace std;

class GestorJson : public Gestor
{
public:
    GestorJson() = default;
    bool crearArchivo(string &ruta, map<int, ProgramaAcademico *> &mapadeProgramasAcademicos, vector<string> &etiquetasColumnas) override;
    bool crearArchivoBuscados(string &ruta, list<ProgramaAcademico *> &programasBuscados, vector<string> &etiquetasColumnas) override;
    bool crearArchivoExtra(string &ruta, vector<vector<string>> &datosAImprimir) override;
};

#endif #ifndef GESTOR_JSON_H
```

GestorTXT:



```cpp
#ifndef GESTOR_TXT_H
#define GESTOR_TXT_H

#include ...

using namespace std;

class GestorTxt : public Gestor
{
public:
    GestorTxt() = default;
    bool crearArchivo(string &ruta, map<int, ProgramaAcademico *> &mapadeProgramasAcademicos, vector<string> &etiquetasColumnas) override;
    bool crearArchivoBuscados(string &ruta, list<ProgramaAcademico *> &programasBuscados, vector<string> &etiquetasColumnas) override;
    bool crearArchivoExtra(string &ruta, vector<vector<string>> &datosAImprimir) override;
};

#endif #ifndef GESTOR_TXT_H
```

Diagrama UML: