

# Desarrollo del perfil del estudiante dentro de No estás solo

Federico Melo Barrero<sup>1</sup>

11 de noviembre de 2024



# Dedicatoria

A mi familia, por su amor incondicional y apoyo constante; a mis amigos, por su compañía y alegría; y a todos quienes han invertido en mi educación, por su dedicación y esfuerzo. A todos ellos, gracias.

Y a Rinconcito, que no es extraño al panóptico.



# Agradecimientos

Hay una infinidad de personas a quienes debo un agradecimiento en este proyecto. En este espacio, me gustaría destacar a algunas de ellas.

Antes que a nadie, quiero agradecer a Angélica, William y Sebastián: mi familia. Su amor y apoyo incondicional moldean mi vida y forjan mi persona. A ellos les debo todo lo que soy.

Luego, debo agradecer a mis amigos más cercanos, quienes son un pilar fundamental en mi vida y son los cimientos de mi felicidad. Especialmente, quiero agradecer, por un lado, a David, Santiago y Andrés; y por otro, a Laura, Laura, Mariana y Sarah. Sin esos dos conjuntos de personas, mi vida sería mucho más triste.

Por mera causalidad, debo agradecer también a todos quienes han invertido su tiempo, recursos y esfuerzos en mi educación. Eso incluye a todos quienes fueron partícipes de mi formación tanto en el Colegio San Carlos como en la Universidad de los Andes. Entre otros, debo agradecimientos especiales a Jaime Rueda, quien me enseñó a estudiar; a Nicolás Rincón, quien me forzó a estudiar; a José Bocanegra, quien me enseñó sobre desarrollo web y construcción de APIs; a Ruby Casallas y a Nicolás Cardozo, quienes me enseñaron sobre calidad de software; y a todos mis profesores de la Universidad.

Por último, quiero agradecer a quienes estuvieron directamente involucrados en este proyecto. A Mariana y a Nicolás, con quienes tomamos el desarrollo de No Estas Solo desde hace más de un año. A Manuel y (nuevamente) a Santiago, quienes hicieron posible el pipeline de analítica. A Oscar, que estaba siempre pendiente de todo. Y, por supuesto, a Marcela, que me dio la oportunidad de trabajar en este proyecto y me acompañó en todo el proceso.

*Big Brother is watching you.*

—George Orwell, *Nineteen Eighty-Four*

# Índice general

<b>1. Datos</b>	<b>9</b>
1.1. Flujo de los datos . . . . .	9
1.1.1. Fuentes de datos . . . . .	9
1.1.2. Procesamiento de los datos . . . . .	11
1.1.3. Representación de los datos como recursos de una API REST . .	14
1.1.4. Consumo de los datos por parte del frontend . . . . .	14
<b>2. Perfil del estudiante</b>	<b>15</b>
2.1. Diseño del perfil académico del estudiante . . . . .	15
2.1.1. ¿Quiénes usarán el perfil académico? . . . . .	15
2.1.2. ¿Para qué se usará el perfil académico? . . . . .	16
2.1.3. ¿Qué información debe incluir el perfil del estudiante? . . . . .	17
2.1.4. ¿Cómo presentar la información en el perfil académico? . . . . .	17
2.1.5. ¿Cómo organizar visualmente el perfil académico? . . . . .	18
2.1.6. TODO: Título. Organizando el perfil . . . . .	18
<b>3. Front</b>	<b>23</b>
<b>4. API</b>	<b>25</b>
4.1. API REST para el perfil del estudiante . . . . .	25
4.2. Calidad de código . . . . .	25
<b>A. Contribuciones al CAPP</b>	<b>27</b>
A.1. Contexto . . . . .	27
A.2. Recursos expuestos . . . . .	27
<b>Glossary</b>	<b>31</b>





# Capítulo 1

## Datos

### 1.1. Flujo de los datos

La presente sección describe, de pies a cabeza, el flujo que los datos atraviesan desde su origen hasta su destino final en el perfil del estudiante. En particular, se detallan las fuentes de los datos, el proceso de ETL al que son sometidos, su representación como recursos accesibles mediante una API REST y su posterior consumo por el frontend, correspondiente al perfil del estudiante.

#### 1.1.1. Fuentes de datos

La información que se utiliza como fuente para el perfil del estudiante es provista por la Oficina de Registro y otras dependencias de la universidad. Esa información se encuentra en archivos PARQUET y XLSX almacenados en la nube mediante Microsoft Azure Blob Storage, en diversas cuentas de almacenamiento Azure Storage Account. Para el perfil del estudiante se hace uso de diez de esos archivos, distribuidos en dos cuentas de almacenamiento. Todos los archivos utilizados, junto con su ubicación en cuentas de almacenamiento y directorios, se muestran en la figura 1.1.

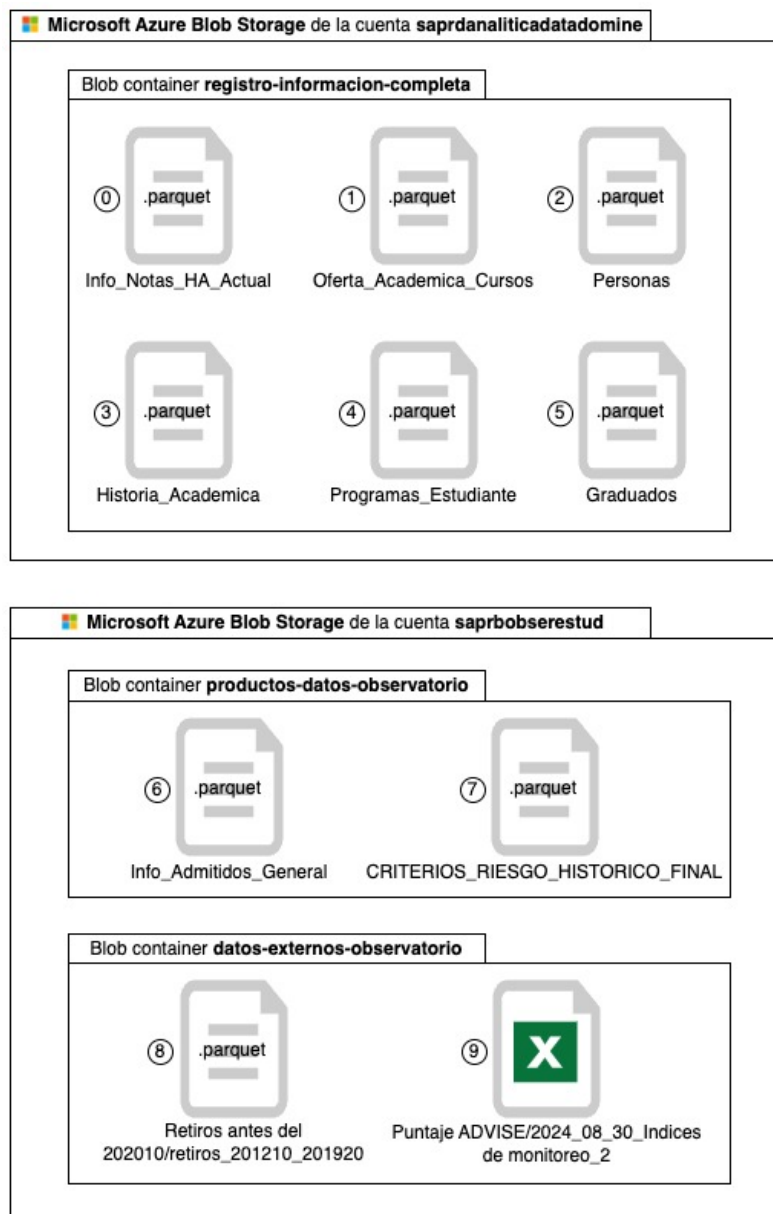


Figura 1.1: Archivos fuente y estructura de directorios en Azure Blob Storage

### 1.1.2. Procesamiento de los datos

La recuperación y el procesamiento de los datos se realizan mediante un *pipeline* de analítica implementado en cuadernos de Jupyter en Python, en colaboración con los estudiantes Santiago Martínez Novoa y Manuel Felipe Porras Tascón\*. El pipeline de analítica realiza las tres etapas clásicas de un proceso de ETL: extracción, transformación y carga de los datos.

#### Extracción de los datos

El proceso de extracción consiste de tomar información determinada de cada uno de los archivos presentados en la figura 1.1. La tabla 1.1 detalla el orden en el que se realiza cada extracción, describe la información que se extrae y especifica el archivo fuente del cual se obtiene, empleando la numeración de los archivos definida en la figura 1.1.

**Tabla 1.1**

*Extracción de datos*

Orden	Información	Descripción	Archivos fuente
1	Histórico académico	Información de las notas obtenidas por los estudiantes en cada una de las asignaturas que han cursado.	0
2	Oferta académica	Información de las asignaturas que se ofertan en cada uno de los semestres académicos.	1
3	Estudiantes	Información básica de los estudiantes.	2, 3, 6
4	Programas académicos	Información de los programas académicos en los que se encuentran inscritos los estudiantes.	4
5	Información adicional de retiros	Información sobre los retiros de los estudiantes para periodos anteriores al 2019-20, que no se encuentra en el Histórico académico.	8
6	Graduados	Información de los estudiantes que se han graduado.	5
7	Criterios de riesgo	Información de los criterios de riesgo que se utilizan para identificar a los estudiantes en riesgo académico.	7
8	Advise	Información de los estudiantes tomada de la plataforma Advise.	9

#### Transformación de los datos

**TODO:** Sentarme con Santi a describir la transformación de los datos

\*Mi más sincero agradecimiento a ambos: a Manuel, en aquel momento estudiante de la Maestría en Ingeniería de la Información, por su valiosa labor al crear y trabajar inicialmente en el cuaderno; y a Santiago, entonces estudiante de Ingeniería de Sistemas y Computación, por tomar la posta, mejorar y mantener este trabajo. Sin ellos, este proyecto no habría sido posible.

### **Carga de los datos**

Como última etapa del procesamiento de los datos, una vez han sido transformados, se cargan en dos archivos en el Blob Storage: uno correspondiente a toda la información relacionada con cada estudiante y otro contiene la información de todas las materias cursadas por cada uno de los estudiantes. La figura 1.2, que extiende la figura 1.1, muestra la estructura de directorios y archivos en el Blob Storage tras la carga del par de archivos mencionado.

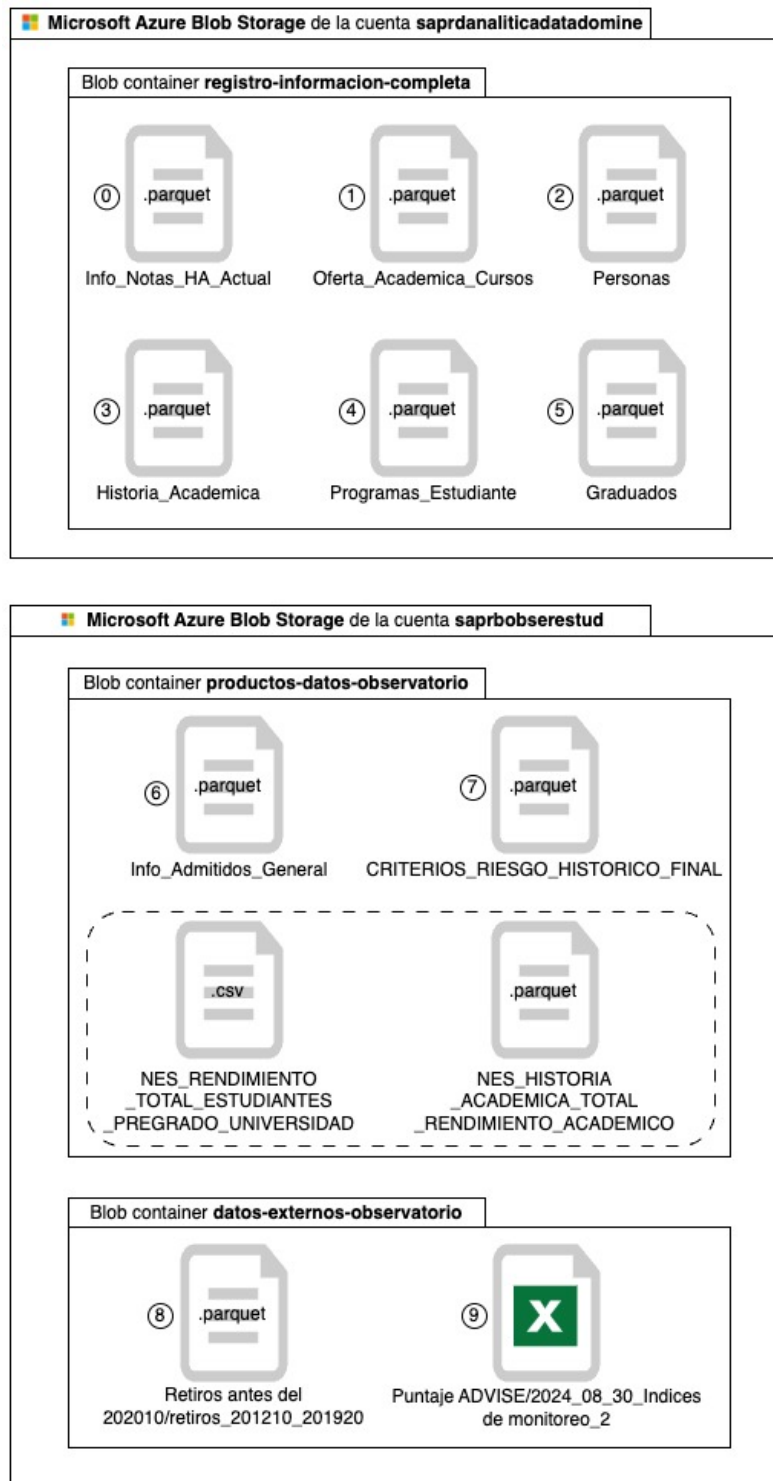


Figura 1.2: Archivos y estructura de directorios en Azure Blob Storage tras la carga de los datos. Los archivos cargados están encerrados en un rectángulo redondeado con bordes discontinuos.

### 1.1.3. Representación de los datos como recursos de una API REST

Una vez los datos han sido procesados y almacenados en el Blob Storage, se exponen como recursos accesibles mediante una API REST. La API se encuentra escrita en Python, utilizando el framework FastAPI, y se despliega en un contenedor Docker que reside en una máquina virtual provista por la universidad.

La siguiente sección trata en detalle todos los aspectos técnicos relacionados con este API REST. Por lo pronto, basta con mostrar, a alto nivel, cuáles son los recursos que provee la API para el consumo por parte del frontend del perfil del estudiante. La tabla 1.2 detalla dichos recursos y la información que contienen. Nótese que el API expone muchos más recursos que los mostrados en la tabla, pero solo esos se utilizan en el perfil del estudiante.

**Tabla 1.2**

*Recursos de la API REST*

Ruta del recurso	Información
/nes/student/	Información básica del estudiante: nombres, apellidos, código, usuario y niveles académicos en los que ha estado inscrito.

### 1.1.4. Consumo de los datos por parte del frontend

El frontend del perfil del estudiante consume los recursos de la API REST para obtener la información necesaria para mostrar el perfil del estudiante.

# Capítulo 2

## Perfil del estudiante

### 2.1. Diseño del perfil académico del estudiante

La presente sección ahonda en el diseño del perfil académico del estudiante, en el cual se pueden discernir cinco etapas. En primer lugar, se reconocen y segregan los distintos tipos de usuario que se espera hagan uso del perfil académico. Tras eso, para cada tipo de usuario, se identifica cuáles son los escenarios en los que va a interactuar con el perfil y cuál es el propósito de su interacción en cada uno de ellos. Luego, con base en eso, se determina qué información debe incluirse en el perfil. Una vez se ha establecido la información que debe contener el perfil académico, se procede a seleccionar y justificar el artefacto visual más adecuado para presentar cada pieza de información. Finalmente, se elige una configuración visual para el perfil académico que facilite su interpretación y se explica el raciocinio detrás de dicha disposición.

A cada etapa del diseño se le destina una subsección. Cada subsección se titula con la interrogante que resuelve. Verbigracia, la primera subsección se titula: “¿Quiénes usarán el perfil académico?”, pues describe precisamente qué actores interactuarán con el perfil.

#### 2.1.1. ¿Quiénes usarán el perfil académico?

Al determinar los usuarios del perfil académico es imperativo cerciorarse de no omitir a ninguno de los actores que interactuarán con él. Para reducir el riesgo de que eso suceda, la metodología utilizada para identificar los usuarios del perfil académico consiste en primero listar a todos los actores que están relacionados con la Universidad de los Andes y, luego, para cada actor, determinar si tiene la necesidad de consultar el perfil académico del estudiante.

Para conformar una lista exhaustiva de todos los actores que interactúan con la Universidad de los Andes se hace uso de dos artefactos. Por un lado, se recurre al catálogo de actores propuesto por Villalobos [1] al interior de su trabajo sobre el modelo de negocio. Dicho artefacto, propio de la arquitectura empresarial, está diseñado para identificar y describir las personas que interactúan con un negocio recibiendo o usando los servicios que éste ofrece. Como el principal negocio de la Universidad de los Andes es la educación, el catálogo de actores propuesto por Villalobos engloba prácticamente a todos los actores que no son empleados de la Universidad.

Por otro lado, se toma como base la documentación sobre gobierno y estructura organizacional de la Universidad [?]. Los actores se organizan en un artefacto similar al catálogo de actores, pero con un campo adicional que incluye todos los cargos que engloba ese actor.

tomó como base la documentación sobre gobierno y estructura organizacional de la Universidad [?]. Los actores se organizan en un artefacto

La adaptación que se realiza al artefacto es que se incluyen también los actores del modelo organizacional, para así desglosar todos los individuos que interactúan con la Universidad de los Andes.

Se muestra el catálogo de actores extendido para la Universidad de los Andes en la Tabla 2.1.

**Tabla 2.1**

*Actores que interactúan con la Universidad de los Andes*

Nombre	Tipo	Descripción
Estudiante	Usuario	Persona que está matriculada en la Universidad de los Andes.
Acudiente	Cliente	Persona que tiene a su cargo la responsabilidad de un estudiante.
Profesor	Empleado	Persona que imparte clases en la Universidad de los Andes.
Coordinador	Empleado	Persona que coordina un programa académico en la Universidad de los Andes.
Director de programa	Empleado	Persona que dirige un programa académico en la Universidad de los Andes.
Vicedecano	Empleado	Persona que asiste al decano en la dirección de una facultad en la Universidad de los Andes.
Decano	Empleado	Persona que lidera una facultad en la Universidad de los Andes.
Vicerrector	Empleado	Persona que lidera una de las cuatro vicerrectorías en la Universidad de los Andes.
Alto administrativo	Empleado	Persona que ocupa un cargo de alta dirección en la Universidad de los Andes.

### 2.1.2. ¿Para qué se usará el perfil académico?

Se espera que el perfil académico del estudiante sea una de las funcionalidades más útiles de NES. En aras de cumplir con esa expectativa, es fundamental determinar todos los casos de uso que se espera que el perfil académico satisfaga. Para poder identificar dichos casos de uso, la estrategia empleada fue reconocer los diferentes actores que interactuarán con el perfil académico y, con base en eso, inferir qué información necesitarán de él.

Así pues, el artefacto seleccionado para identificar los actores y sus necesidades fue un diagrama de casos de uso. Se muestra en la Figura ??.



El diagrama anterior traduce naturalmente a historias de usuario. Cada historia de usuario es una descripción de un caso de uso específico por parte de un actor particular. TODO: Además, se habló con los actores.

Obviamente, el perfil académico del estudiante debe satisfacer todas las historias de usuario listadas en la Tabla ??.

### 2.1.3. ¿Qué información debe incluir el perfil del estudiante?

La importancia de esta etapa del proceso es evidente. Exhibir muy poca información reduce la utilidad del perfil, mientras que mostrar demasiada resulta en un perfil abrumador y complicado de interpretar. El segundo escenario es indeseable, mas el primero es inaceptable: no puede ocurrir que el perfil del estudiante no satisfaga alguno de los casos de uso para los cuales se contempló.

Con eso en mente, una forma de cerciorarse de que se recoge toda la información necesaria para satisfacer todas las historias de usuario es simplemente listar todas las piezas de información que son necesarias para cada historia de usuario. La Tabla ?? registra esa labor y por ende resume la información que debe incluirse en el perfil académico.

#### **información general**

La información general del estudiante es la primera sección de su perfil. La información incluida en esta sección es la que se considera más relevante para identificar al estudiante y permitir contactarlo. Esta información es independiente del nivel académico del estudiante.

El propósito de esta información es identificar al estudiante y permitir contactarlo. La información general incluye:

- Nombre legal completo
- Código Uniandes
- Usuario Uniandes
- Contacto por correo electrónico Uniandes
- Contacto por Microsoft Teams
- Link a Advise (TODO: Esto se queda?)

Adicionalmente

### 2.1.4. ¿Cómo presentar la información en el perfil académico?

Tras determinar qué información será incluida el perfil académico, es necesario decidir cómo presentarla. La calidad del perfil depende en gran medida de elegir el artefacto visual más adecuado para cada pieza de información.

### 2.1.5. ¿Cómo organizar visualmente el perfil académico?

Finalmente, una vez se ha decidido qué información incluir y cómo presentar cada pieza de información, es necesario determinar cómo organizar visualmente el perfil académico. La disposición de los elementos visuales incide en la facilidad de interpretación del perfil académico.

### 2.1.6. TODO: Título. Organizando el perfil

El perfil del estudiante idealmente debe constar de:

1. Información personal
2. Información académica
3. Información financiera

En caso de que un estudiante haya estado vinculado con la universidad en más de un nivel académico, la información académica y financiera se debe presentar de manera separada para cada nivel académico.

Los niveles académicos posibles son cuatro: pregrado, extensión, magister y doctorado. No tenemos (ni tendremos?) información financiera para estudiantes de posgrado, por lo cual no se muestra información financiera para los niveles de extensión, magister y doctorado.

Por ende, el perfil de un estudiante tiene a lo sumo seis secciones:

- Información personal.
- Información como estudiante de pregrado.
  - Académica
  - Financiera
- Información académica como estudiante de extensión.
- Información académica como estudiante de magister.
- Información académica como estudiante de doctorado.

La primera sección estará presente en todos los perfiles de estudiantes, mientras que las otras cuatro secciones se mostrarán únicamente si el estudiante ha estado vinculado con la universidad en el nivel académico correspondiente.

#### Información personal

#### Información como estudiante de pregrado

Para todos los estudiantes que hayan cursado, parcial o completamente, un pregrado en la Universidad de los Andes, se mostrará la sección de información como estudiante

de pregrado. Esta sección se divide en dos subsecciones: información académica y financiera.

La información académica incluida en esta sección no pretende ser un resumen, sino más bien una descripción detallada del desempeño académico del estudiante durante su pregrado. El reto es presentar toda esta información de manera clara y concisa, de forma que sea posible obtener una visión general del desempeño académico del estudiante sin necesidad de examinar cada dato en detalle, pero también sea viable realizar una inspección detallada de ser necesario.

Las tablas a continuación detallan la información incluida en la sección de información académica como estudiante de pregrado y los artefactos visuales utilizados para presentarla.

**Tabla 2.2**

*Datos incluidos en la sección de información académica como estudiante de pregrado y artefactos visuales utilizados para presentarlos*

Dato	Artefacto	Notas
Promedio General Acumulado	Valor	TODO: Al igual que antes, acá falta contexto. Cómo se compara un promedio de 4.5 del 2023 con uno de 4.5 del 2019? Conversación con Mario Sánchez sobre monitorías. Solución: percentiles o varianza/desviación/alguna comparación con el promedio de la cohorte. Gráfica de bala.
Número de matrículas	Valor	TODO: Mismo problema. Debe ponerse cuál es el número de matrículas esperadas por carrera. Problema: intercambios, prácticas
Porcentaje de créditos aprobados	Valor	
PGA por periodo	Gráfica de líneas	
Promedio semestral por periodo	Gráfica de líneas	
Porcentaje de créditos aprobados por periodo	Gráfica de líneas	
Porcentaje acumulado de créditos aprobados por periodo	Gráfica de líneas	
Número de créditos aprobados por periodo	Gráfica de líneas	TODO: Yo creo que vale la pena combinar esta gráfica con la de arriba, en la misma vista, como habíamos pensado en algún momento
Número de créditos aprobados acumulados por periodo	Gráfica de líneas	
Número de suspensiones académicas	Valor	
Número de pruebas académicas	Valor	
Número de pruebas de reingreso	Valor	
Número de incompletos totales	Valor	
Número de créditos aprobados, retirados y reprobados por periodo	Gráficas de líneas superpuestas	
Total acumulado de créditos aprobados, retirados y reprobados por periodo	Gráficas de líneas superpuestas	
Total de créditos inscritos, aprobados, retirados, reprobados, incompletos y pen-	Valores	

TODO: Sobre esto: Pasar la tabla de créditos inscritos a una barras apiladas. En posgrado será solo una barra.

Respecto a información financiera

TODO: Podría quedar bien hacer una sola pestaña para extensión, Maestría y doctorado. Entonces, serían dos: pregrado (con académico y financiero) y posgrado.



# Capítulo 3

## Front

### Ruta al perfil del estudiante

Como ruta para acceder al perfil del estudiante desde NES se seleccionó `/perfil-estudiante/<correo>`, donde `<correo>` corresponde al correo Unian-des, sin el dominio, del estudiante de quien se quiere ver el perfil. Por ejemplo, la ruta para acceder al perfil del autor de este documento es `/perfil-estudiante/f.melo`.

Cualquier usuario que esté autenticado en NES y cuente con los permisos adecuados podrá acceder y visualizar directamente el perfil del estudiante al ingresar a la ruta indicada.

Esto permite que entre profesores y directivos de la Universidad se compartan fácilmente perfiles de estudiantes. En caso de que el lector tenga ese rol dentro de la Universidad, puede probar la funcionalidad iniciando sesión en NES y luego pulsando en el siguiente enlace: <https://noestassolo.virtual.uniandes.edu.co/perfil-estudiante/f.melo>.

Se hace hincapié en que el usuario debe estar autenticado en NES y debe contar con los permisos necesarios para acceder al perfil de un estudiante. De lo contrario, no podrá visualizarlo. En particular, ningún estudiante podrá acceder al perfil de algún otro de esta manera (ni de ninguna otra, salvo que el otro estudiante o algún otro usuario autorizado deliberadamente se lo muestre).





# Capítulo 4

## API

### 4.1. API REST para el perfil del estudiante

Esta sección se ocupa de describir en detalle la API REST que recoge los archivos ya procesados del Blob Storage y los expone como recursos accesibles para el frontend del perfil del estudiante.

### 4.2. Calidad de código

Teniendo en cuenta que el software producido es de alto valor para la universidad y por ende probablemente querrá ser mantenido, extendido y reutilizado en el futuro, su desarrollo fue sometido a rigurosos estándares de calidad de código. Esta sección detalla las herramientas y prácticas empleadas para garantizar la calidad del código.

#### Resumen de la sección 2

Se realizaron múltiples esfuerzos para garantizar la calidad del código del proyecto:

- Se documentaron todas las funciones y métodos con docstrings siguiendo la convención de Pandas.
- Se utilizaron isort y black como formateadores, Flake8 y SonarLint como linters, y doctest y Pytest para las pruebas unitarias.
- Se configuró un hook de pre-commit de Git para garantizar el formato y ejecutar las pruebas antes de cada commit.
- Se estableció un pipeline en GitHub Actions que ejecuta las pruebas, genera un reporte de cobertura y lo envía a SonarQube.
- Se configuró SonarQube para realizar análisis estático sobre el código, usando el perfil de calidad estándar para Python 3.11.

Todas las funciones y métodos de la API están documentados mediante *docstrings*. Naturalmente, los docstrings siguen las convenciones definidas en el estándar PEP 257. Más aún, debido a que las disposiciones del estándar son notablemente flexibles, se

optó por seguir la afamada convención de docstrings de NumPy. Más específicamente, se satisfacen todos los lineamientos de la guía de docstrings de Pandas.

Su sintaxis satisface el estilo reStructuredText (REST),

Se hace uso del linter Flake8,

Se emplea la librería `flake8-docstrings`, con la configuración `docstring-convention=` en el archivo de configuración de Flake8, para verificar que los docstrings sigan la convención de NumPy.

# Apéndice A

## Contribuciones al CAPP

Este apéndice describe las contribuciones realizadas al CAPP durante el desarrollo de este proyecto. En particular, se detallan los recursos expuestos por la API REST para consumo exclusivo del CAPP, a petición de Daniel Arango Cruz y Marilyn Stephany Joven Fonseca, quienes lideran el desarrollo de aquel proyecto.

### A.1. Contexto

El CAPP es un sistema que permite a los estudiantes de pregrado de la Universidad de los Andes visualizar su progreso académico. Su propósito principal es determinar si un estudiante ha cumplido con los requisitos mínimos para graduarse de un programa académico. Para ello, el CAPP se conecta con diversas fuentes de datos, entre ellas, con el API REST diseñado y desarrollado en este proyecto.

### A.2. Recursos expuestos

Los recursos provistos fueron diseñados específicamente para el CAPP ante solicitud de los desarrolladores. Se solicitó que la nomenclatura para los recursos y sus atributos fuera en español, en lugar de inglés, como se había manejado en el resto de la API REST.

A causa de eso y como forma de separación semántica, se decidió agrupar los *endpoints* bajo el prefijo `/capp/`. La tabla A.1 detalla los *endpoints* expuestos por la API REST para el CAPP.

**Tabla A.1***Recursos expuestos por la API REST para el CAPP*

Método HTTP	URL	Descripción	Cuerpo de la petición	Respuesta
GET	/capp /materias /estudiante /{codigo estudiante}	Obtiene todas las materias aprobadas por un estudiante en sus estudios de pregrado.	–	Lista de objetos <i>Materia</i> con las materias aprobadas por el estudiante.
GET	/capp /materia /{codigo materia}	Obtiene la información básica de una materia a partir de su código.	–	Objeto <i>Materia</i> correspondiente al código.
PUT	/capp /materias/	Obtiene la información básica de múltiples materias a partir de una lista de sus códigos.	Lista de códigos de materias	Lista de objetos <i>Materia</i> correspondientes a los códigos provistos.
GET	/capp /estudiante /programa/	Obtiene la información de los programas académicos cursados por el estudiante.	–	Lista de nombres de los programas académicos, ordenados desde el principal.

El recurso *Materia* mencionado en la tabla es una representación simplificada de una materia, que contiene la información mínima necesaria para el CAPP. La figura A.1 muestra un ejemplo de esta representación. No se hace uso de la representación *SubjectResponse* mencionada en la sección ??, ya que esta contiene información adicional y se incurriría en sobreexposición de datos.

```
{
  "codigo_curso": "MATE",
  "numero_curso": "1104",
  "numero_creditos": 3,
  "nombre_curso": "TEORIA DE GRAFOS",
  "atributo_sec": "",
  "periodo": 202320
}
```

Figura A.1: Ejemplo de representación del recurso *Materia* para el CAPP

# Glosario

**API (*Application Programming Interface*)** Mecanismo que permite la interoperabilidad de aplicaciones heterogéneas, permitiendo a un programa acceder a las funciones y datos de otro. 14, 25, 29

**API REST** API que sigue los principios arquitectónicos REST, permitiendo interacciones con los recursos usando métodos estándar HTTP. 9, 14, 25, 27

**black** Librería o utilidad de Python que formatea automáticamente el código fuente usando un estilo consistente con cualquier otro código fuente formateado con black, de acuerdo con su página de web. 25

**CAPP (*Curriculum, Asesoría y Planeación de Programas*)** Sistema de información de la Universidad de los Andes que determina si un estudiante satisface los requisitos para graduarse de un programa académico. 27, 28

**docstring** Cadena de texto especiales que se escriben al inicio de un módulo, clase, método o función en Python, que describe su funcionalidad y cómo usarlo. 25, 29

**doctest** Módulo en Python que permite probar fragmentos de código incluidos en los docstrings, validando que la salida coincida con los ejemplos dados. 25

**ETL (*Extract, Transform, Load*)** Proceso utilizado en la integración de datos, que consiste en extraer datos de múltiples fuentes, realizar sobre ellos las transformaciones necesarias y cargarlos en un destino. 9, 11

**Flake8** Herramienta de Python que combina varios linters para verificar el cumplimiento de las guías de estilo de Python. Más información en su documentación. 25, 26

**formateador** Herramienta que modifica automáticamente el código fuente para que siga convenciones de estilo predefinidas. 25

**Git** Sistema de control de versiones distribuido. 25

**GitHub Actions** Servicio de integración continua de GitHub que permite automatizar flujos de trabajo como pruebas y despliegues en repositorios de código fuente. Más información en su documentación. 25

**hook** En Git, es un script que se ejecuta automáticamente en momentos específicos del flujo de trabajo, como antes de hacer un commit o un push. 25

**HTTP (*Hypertext Transfer Protocol*)** Protocolo de comunicación que permite la transferencia de información en la web, caracterizado por operar sin estado y basado en el modelo cliente-servidor. 30

**isort** Librería o utilidad de Python que alfabetiza las importaciones en el código fuente y las separa por tipo, de acuerdo con su página de web. 25

**linter** Herramienta que analiza el código fuente para encontrar errores, malas prácticas o incumplimientos de convenciones de estilo. 25

**NES (No Estás Solo)** Plataforma web de la Universidad de los Andes para el éxito estudiantil. <https://noestassolo.virtual.uniandes.edu.co/>. 16, 23

**NumPy** “El paquete fundamental para la computación científica en Python”, de acuerdo con su página oficial. Librería de Python que añade soporte para arreglos y matrices de gran tamaño, junto con una colección de funciones matemáticas para operarlos. 26

**Pandas** Librería de análisis y manipulación de datos en Python. Más información en su página oficial. 25, 26

**PEP (*Python Enhancement Proposal*)** Documento que propone y describe una nueva característica o modificación en Python. En la PEP 1 se define qué es un PEP. 25

**perfil de calidad** Conjunto de reglas y configuraciones que definen los estándares de calidad del código. 25

**pipeline** Flujo de automatización en el que se definen una serie de pasos o tareas que deben ejecutarse de manera secuencial o en paralelo. 25

**pruebas unitarias** Pruebas automáticas que se enfocan en verificar el correcto funcionamiento de las unidades más pequeñas de un programa, como funciones o métodos individuales. 25

**Pytest** Framework de pruebas para Python. Más información en su página oficial. 25

**Python** Lenguaje de programación de propósito general y alto nivel, interpretado y multiparadigma, aunque principalmente orientado a objetos. 25

**REST (*Representational State Transfer*)** Estilo arquitectónico para sistemas hipermídia distribuidos, como la *World Wide Web*. Se basa en la transferencia de representaciones de recursos, que son identificados por URIs, y la manipulación de estos recursos mediante métodos estándar de HTTP. Para una descripción detallada, leer el quinto capítulo de la tesis doctoral de Roy Fielding. 29

**reST (*reStructuredText*)** Sintaxis de marcado ligero utilizada para documentación, como docstrings en Python. Más información en la documentación oficial. 26

**SonarLint** Extensión de análisis estático de código que se integra en el editor o IDE para detectar problemas de calidad en el código y proponer correcciones. 25

**SonarQube** Plataforma que realiza análisis estático de código para detectar vulnerabilidades, errores y problemas de mantenimiento en proyectos de software. 25

**URI (*Uniform Resource Identifier*)** Cadena de caracteres con estructura fija que identifica un recurso en la web de forma única. 30





# Bibliografía

- [1] Jorge Villalobos. *El Modelo de Negocio - Lecturas para Arquitectos de Negocio*. 07 2024.