



FACULTAD DE
INGENIERÍA



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Universidad de la República
Facultad de Ingeniería

INCO

Bases de Datos no Relacionales 2023

Proyecto Final

Grupo 17

Integrantes:

Nombre	Cédula de Identidad	Correo electrónico
Martín Bazzano	4.447.590-8	martin.bazzano@fing.edu.uy
Bruno Lartigau	5.100.433-6	bruno.lartigau@fing.edu.uy

1. Introducción

A la hora de cursar una carrera en una facultad, el estudiante principalmente desea conocer qué debe hacer para poder obtener el título correspondiente a dicha carrera; en otras palabras, qué unidades curriculares (UC) debe cursar.

Para ello, en el caso de la carrera Ingeniería en Computación dictada por la Facultad de Ingeniería de la Universidad de la República (UDELAR), esta brinda distintas herramientas que otorgan información útil al estudiante.

Por un lado, provee una currícula sugerida que indica las asignaturas obligatorias a cursar y la recomendación de en qué semestre hacerlo; mientras que por otro, ofrece un grafo de previas que muestra a nivel general qué asignaturas deben aprobarse para poder cursar la siguiente.

Si bien estos recursos han sido de gran utilidad para las personas que estudian en esta carrera, a su vez ha resultado insuficiente en algunas ocasiones. Esto se debe a que esta información está enfocada únicamente a las UC obligatorias, cuando en realidad existen muchas asignaturas optativas que deben realizarse para poder cubrir con la cantidad de créditos que la carrera exige para la obtención del título. Dichas UC junto con sus previas solo se encuentran en la página de Bedelías, la cual no es del todo amigable a diferencia de los recursos previamente mencionados.

Otro aspecto a tener en cuenta, es que no solo se deben llegar a los 450 créditos en la carrera, sino que se debe obtener un mínimo de créditos por materias (es decir, por grupo de UC), los cuales no están reflejados en el grafo de previas, aunque sí en la currícula sugerida actual pero de manera superficial.

El objetivo de este proyecto consiste en la creación de una base de datos de grafos, la cual represente el sistema de previas de la carrera Ingeniería en Computación dictada en la Facultad de Ingeniería de la Udelar. El mismo incluiría, además de las clásicas UC obligatorias, la mayor cantidad de asignaturas optativas posibles.

Para representar las previas, no sólo se incluirán las asignaturas previas a otras, sino también los casos donde se requiera cierta cantidad de créditos en un grupo de materias o en la carrera para poder cursar la UC.

A partir de este grafo, se podrían realizar distintas consultas de interés, como puede ser la obtención de distintas métricas respecto al cursado de asignaturas, y analizar los resultados obtenidos.

2. Trabajos Relacionados

A pesar que es un tema de interés general a nivel de estudiantes que cursan la carrera de Ingeniería en Computación, no se han encontrado demasiados trabajos que aborden este problema.

Además de la ya mencionada currícula sugerida y grafo de previas brindados por la facultad, un proyecto interesante es el propuesto por Nicolás Díaz, el cual también puede encontrarse en el Entorno Virtual de Aprendizaje (EVA) de Carreras de Computación. Este trabajo consistió en crear una malla curricular interactiva, la cual despliega las unidades curriculares obligatorias más importantes de la carrera, permitiendo seleccionarlas simulando aprobar su curso o examen, para luego ir desbloqueando el resto de asignaturas hasta llegar a Proyecto de Ingeniería de Software.

Este proyecto es una iniciativa atractiva, sobre todo para que cada estudiante la utilice dependiendo de su situación particular sin tener que estar haciendo memoria o revisando otros recursos para saber qué unidades curriculares puede cursar en función de las asignaturas que tenga aprobada.

Sin embargo, tiene la gran desventaja de no tener ninguna información de las asignaturas optativas de la carrera, además de también estar ausentes algunas unidades curriculares obligatorias, como las pertenecientes al grupo Gestión en Organizaciones o a Ciencias Humanas y Sociales. Sumado a eso, tiene datos erróneos en la cantidad de créditos de algunas asignaturas, lo cual puede confundir al estudiante que vaya a utilizar la herramienta.

3. Desarrollo

3.1. Obtención de Datos

Los datos a utilizar provienen principalmente del sitio de Bedelías de la Universidad de la República, así como también del sitio de la Facultad de Ingeniería y de la plataforma EVA.

Como primera opción se manejó solicitarle un dataset con la información sobre previas al SeCIU (Servicio Central de Informática de la Universidad de la República), pero ellos confirmaron que no cuentan con la información en el formato que necesitábamos ni los recursos para atender nuestros requerimientos.

Luego se manejó la posibilidad de aplicar alguna técnica de obtención de datos, como lo es el *web scraping*, pero se evaluó que el costo de crear un programa que obtenga los datos sería más alto que el de cargar los datos manualmente, teniendo en cuenta que la cantidad de unidades curriculares no es tan grande y que la interfaz del sistema de Bedelías es compleja.

Además, cargando los datos manualmente se podría también agregar información de otras fuentes que enriquezcan los datos, como el sitio de la Facultad de Ingeniería y las páginas de los cursos en la plataforma EVA, lo cual sería incluso más complejo mediante *web scraping*, dado que habría que crear un programa distinto para cada sitio empleado y que cierta información de los cursos (ejemplo: programas de las UC) se encuentra en archivos escaneados en lugar de en formato texto.

Como contrapartida, la carga manual de datos hace más factible la introducción de errores en los datos y por consiguiente una menor calidad de los mismos.

Cabe destacar, que aunque como fundamentamos, la carga manual de datos pareció ser la mejor opción para los datos que se quería hacer en esta tarea, la obtención de los mismos de múltiples fuentes llevó una carga considerable de tiempo, más del previsto a pesar de no ser tantos nodos.

De esta manera, se crearon varios archivos .csv con la información sobre las unidades curriculares (UC), los cuales se detallan a continuación junto con los campos que contienen.

asignaturas.csv

codigoUC: Código de la UC.

nombreUC: Nombre de la UC.

tipoUC: Si es curso o examen.

codigoMateria: Código de la materia a la que pertenece la UC.

instituto: Instituto de la Facultad de Ingeniería que dicta la UC, o en caso que lo dicte otra institución, nombre de la institución.

semestre: Semestre en que se dicta la UC. Puede ser Impar, Par o Ambos.

dictada: Si en caso que la UC se haya dictado en el último año. No en caso contrario.

creditos: Cantidad de créditos que otorga la UC. Se asume que no otorga créditos en caso de que sea un curso cuyo examen aparezca separadamente.

cupos: Cantidad de cupos que tenga la UC en caso que aplique.

calidadLibre: Si en caso que la UC adhiera a la calidad de libre optativa. No en caso contrario.

creditosCarrera: Cantidad de créditos necesarios en la carrera para poder cursar la UC, en caso que aplique.

grupo_materia.csv

codigoMateria: Código de la materia.

nombreMateria: Nombre de la materia.

creditos_materia.csv

codigoUC: Código de la UC.

nombreUC: Nombre de la UC.

tipoUC: Si es curso o examen.

codigoMateria: Código de la materia en la cual se necesitan créditos para poder cursar la UC.

creditos: Cantidad de créditos en la materia previamente definida que se necesitan para poder cursar la UC.

agrupacionesUC.csv

codigoAgrupacion: Código de la agrupación de UC. Definido por nosotros.

nombreAgrupacion: Nombre de la agrupación de UC. Definido por nosotros.

tipoAgrupacion: Tipo de la agrupación de UC. Puede ser OR, si la agrupación representa una equivalencia entre las UC, o AND si todas las UC de la agrupación son necesarias.

codigoUC: Código de la UC perteneciente a la agrupación. También se permiten códigos de sub-agrupaciones pertenecientes a la agrupación.

nombreUC: Nombre de la UC perteneciente a la agrupación. También se permiten nombres de sub-agrupaciones pertenecientes a la agrupación.

tipoUC: Tipo de la UC perteneciente a la agrupación, es decir si es curso o examen.

También se permite la opción "Grupo" si es una sub-agrupación.

previas.csv

codigoUC: Código de la UC.

nombreUC: Nombre de la UC.

tipoUC: Si es curso o examen.

previas: Código de la UC que es previa para poder cursar la UC definida previamente.

También se permiten códigos de agrupaciones.

no_debe_tener.csv

codigoUC: Código de la UC.

nombreUC: Nombre de la UC.

tipoUC: Si es curso o examen.

noDebeTener: Código de la UC que no puede haber sido aprobada para poder cursar la UC definida previamente.

A continuación se podrá acceder a un enlace a un GitLab con los mismos:

<https://gitlab.fing.edu.uy/martin.bazzano/bdnr-2023-grupo-17>

3.2. Diseño del grafo

Para el diseño de la base de datos se tuvieron en cuenta tanto los datos obtenidos como las consultas que se pretendían llevar a cabo, resultando un grafo con la estructura que se detalla en la Figura 1.

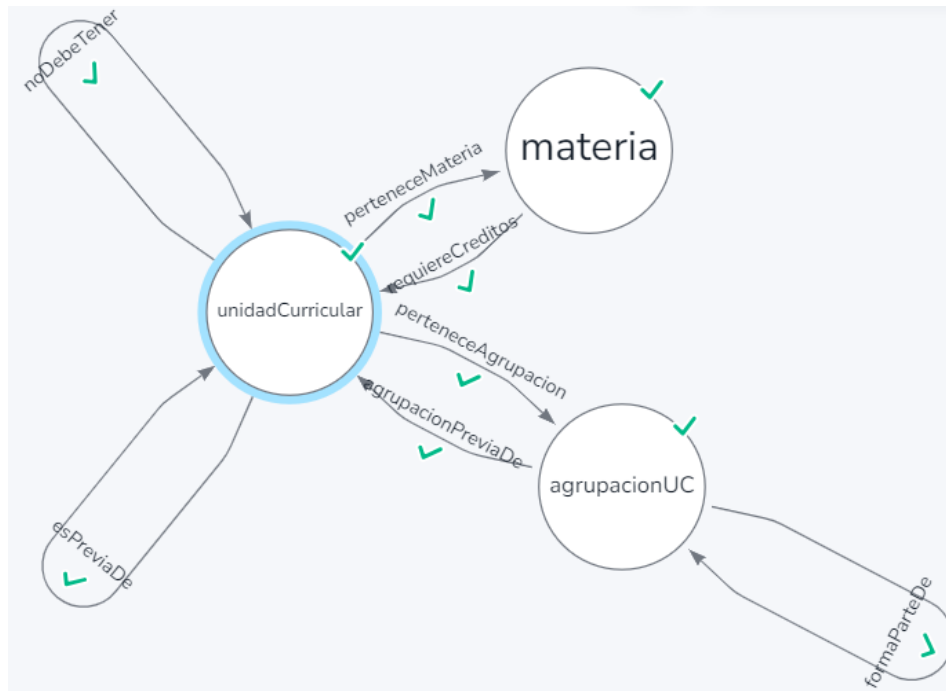


Figura 1: Estructura del Grafo

Se proponen los siguientes tipos de nodos.

unidadCurricular:

Representa una unidad curricular. Contiene los siguientes atributos: codigoUC, nombreUC, tipoUC, instituto, semestre, dictada, creditos, cupos, calidadLibre, creditosCarrera. Los atributos representan lo mismo que ya fue explicado en la definición del csv.

materia:

Representa una materia, que es como se le denomina en los planes de estudio a las grandes áreas temáticas ligadas a un sector de la ciencia o de la técnica. Contiene los siguientes atributos: codigoMateria, nombreMateria. Los atributos representan lo mismo que ya fue explicado en la definición del csv.

agrupacionUC:

Representa un grupo de unidades curriculares. Contiene los siguientes atributos: codigoAgrupacion, nombreAgrupacion, tipoAgrupacion. Los atributos representan lo mismo que ya fue explicado en la definición del csv.

En cuanto a relaciones, se proponen los siguientes tipos.

esPreviaDe:

Va desde unidadCurricular a unidadCurricular. Una unidadCurricular *uc* está relacionada con una unidadCurricular *uc2* si *uc* es previa de *uc2*.

noDebeTener:

Va desde unidadCurricular a unidadCurricular. Una unidadCurricular *uc* está relacionada con una unidadCurricular *uc2* si para poder cursar *uc* no se debe tener *uc2* como aprobada.

perteneceMateria:

Va desde unidadCurricular a materia. Una unidadCurricular *uc* está relacionada con una materia *m* si *uc* pertenece a la materia *m*.

perteneceAgrupacion:

Va desde unidadCurricular a agrupacionUC. Una unidadCurricular *uc* está relacionada con una agrupacionUC *a* si *uc* está incluida en la agrupación de UCs *a*.

requiereCreditos:

Va desde materia a unidadCurricular. Una materia *m* está relacionada con una unidadCurricular *uc* si para poder cursar *uc* se necesitan créditos en la materia *a*. Esta relación tiene un atributo *creditos*, que representan la cantidad de créditos que se necesitan de esa materia.

agrupacionPreviaDe:

Va desde agrupacionUC a unidadCurricular. Una agrupacionUC *a* está relacionada con una unidadCurricular *uc* si *uc* tiene como previa la agrupación *a*. Esto quiere decir generalmente, que es necesario que alguna de las UCs dentro *a* esté aprobada, y que cualquiera de ellas sirva.

formaParteDe:

Va desde agrupacionUC a agrupacionUC. Una agrupacionUC *a* está relacionada con una agrupacionUC *a2* si *a* es un subgrupo de *a2*, es decir está incluido en la agrupación de UCs *a2*.

3.3. Construcción del grafo

Para la construcción del grafo se decidió utilizar la herramienta Neo4j, más precisamente Neo4j AuraDB debido a que ofrece las ventajas propias de trabajar en la nube y ya se tenía conocimiento de la misma, ya que fue utilizada para una tarea anterior de la asignatura. Además, al haber sido utilizada anteriormente, también se contaba con una instancia ya creada en AuraDB.

Para la carga de los datos contenidos en los archivos .csv utilizamos el Neo4j Data Importer, una aplicación visual lanzada desde la consola. Se decidió utilizar esta alternativa debido a la facilidad de uso de la misma.

La herramienta permite subir los archivos .csv a la plataforma y crear nodos y relaciones, seleccionando desde la interfaz de qué archivo .csv, y en particular, de qué columna se obtiene cada atributo de cada uno de ellos.

Para validar la carga, visualizamos los datos en la sección “Explore” de la interfaz de Neo4j, la cual permite ver los datos fácilmente, y también mostrar sólo los nodos y/o relaciones que se requieran. En la Figura 2 se muestra el grafo completo, así como se ve en dicha herramienta de visualización.

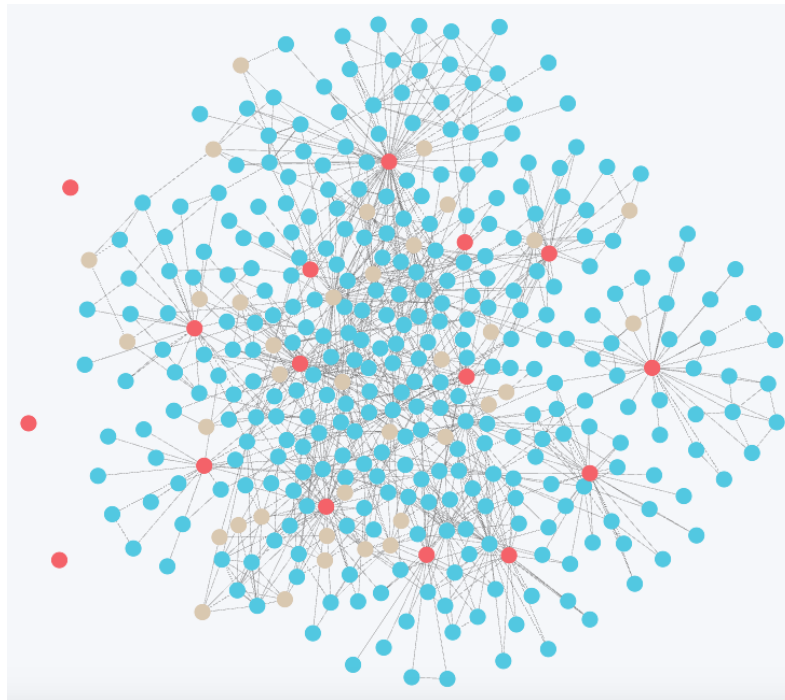


Figura 2: Grafo completo

El grafo completo, resultante de nuestra carga de datos cuenta con:

370 nodos

- 318 de tipo unidadCurricular
- 16 de tipo materia
- 36 de tipo agrupacionUC

1088 relaciones

- 403 de tipo esPreviaDe
- 183 de tipo agrupacionPreviaDe
- 318 de tipo perteneceMateria
- 89 de tipo perteneceAgrupacion
- 73 de tipo noDebeTener
- 18 de tipo requiereCreditos
- 4 de tipo formaParteDe

4. Experimentación

A partir del grafo obtenido se realizaron algunas consultas de interés en el lenguaje Cypher.

4.1. Unidades Curriculares cursadas

Para comenzar, una consulta interesante a plantearse sería cuáles son las UC que se están cursando actualmente, dado que Bedelía también posee información de varias UC que ya no se dictan, lo cual puede ser confundir a un estudiante de nuevo ingreso.

```
1. MATCH (uc:unidadCurricular)
2. WHERE uc.dictada = 'Si'
3. RETURN uc.codigoUC AS Código, uc.nombreUC AS Nombre
```

Luego de ejecutar esta consulta da como resultado un total de 216 UC. Hay que tener en cuenta que la base toma como dos UC distintas al curso y al examen de la asignatura, por lo que la cantidad de UC real es mayor a 108.

4.2. Promedio de créditos

Sabiendo que un crédito equivaldría aproximadamente a una hora de trabajo dedicado a la UC, una pregunta a hacerse sería cuál es el promedio de los créditos de las UC de esta carrera. Para ello, se realizó la siguiente consulta, la cual da como resultado un promedio de aproximadamente 9.21 créditos.

```
1. MATCH (uc:unidadCurricular)
2. WHERE uc.dictada = 'Si'
3. WITH AVG(toInteger(uc.creditos)) AS Promedio
4. RETURN Promedio
```

4.3. UC con mayor y menor cantidad de créditos al promedio

Una vez calculado el promedio, interesaría saber cuales son las asignaturas a las que el estudiante más debería dedicarle, y cuales son las que menos.

```
1. MATCH (uc:unidadCurricular)
2. WHERE toInteger(uc.creditos) > 9 AND uc.dictada = 'Si'
3. RETURN uc.codigoUC AS Código, uc.nombreUC AS Nombre, uc.creditos AS Créditos
4. ORDER BY toInteger(uc.creditos) DESC
```

Luego de ejecutar esta consulta, se obtiene un total de 82 UC, siendo las mostradas en la Figura 3 las que tienen la mayor cantidad de créditos.

Código	Nombre	Créditos
"1730-C1"	"PROYECTO DE GRADO"	"30"
"1730-C2"	"PROYECTO DE GRADO"	"30"
"1730-C3"	"PROYECTO DE GRADO"	"30"
"SRN02-E"	"CÁLCULO I"	"16"
"SRN09-E"	"CÁLCULO II"	"16"
"1911-E"	"FUNDAMENTOS DE BASES DE"	"15"

Figura 3: UC con mayor cantidad de créditos

```

1. MATCH (uc:unidadCurricular)
2. WHERE toInteger(uc.creditos) <= 9 AND uc.dictada = 'Si'
3. RETURN uc.codigoUC AS Código, uc.nombreUC AS Nombre, uc.creditos AS Créditos
4. ORDER BY toInteger(uc.creditos) ASC

```

Luego de ejecutar esta consulta, se obtiene un total de 75 UC, siendo las mostradas en la Figura 4 las que tienen la menor cantidad de créditos.

Código	Nombre	Créditos
"1213-C"	"PLANIFICACION DE CLASES: DISEÑO DE UNID.DIDACT."	"2"
"SRN25-E"	"TALLER INTERDISCIPLINAF"	"2"
"1225-C"	"POLITICAS CIENTIFICAS EN INF.Y COMP."	"3"
"559EU-E"	"PROPIEDAD INTELECTUAL"	"3"

Figura 4: Asignaturas con menor cantidad de créditos

4.4. Promedio de cupos

Para aquellas UC que piden cupos para poder cursarla, interesa saber en promedio hasta cuántas personas podrían anotarse en dichas asignaturas. Para eso se realizó la siguiente consulta, la cual dió un resultado de aproximadamente 28.12. Es decir, si una UC tiene cupos el mismo será de 28 estudiantes.

```
1. MATCH (uc:unidadCurricular)
2. WHERE uc.dictada = 'Si' AND uc.cupos <> "-"
3. WITH AVG(toInteger(uc.cupos)) AS Promedio
4. RETURN Promedio
```

4.5. Unidades Curriculares sin aprobar asignaturas del IMERL

Es sabido que existe cierta cantidad de estudiantes que estudian esta carrera por pasión hacia la informática, pero tal vez no tengan la misma afinidad hacia las matemáticas. Por lo tanto, una consulta interesante a realizar es averiguar cuales UC se podrían cursar si un estudiante decide no hacer asignaturas dictadas por el IMERL.

```
1. MATCH (uc:unidadCurricular)
2. WHERE NOT
  ((:unidadCurricular{instituto:'IMERL'})-[:esPreviaDe|agrupacionPreviaDe|
  perteneceAgrupacion*]->(uc:unidadCurricular) OR uc.instituto = 'IMERL')
  AND NOT (:materia)-[:requiereCreditos]->(uc:unidadCurricular)
  AND uc.dictada = 'Si'
3. RETURN uc.codigoUC AS Código, uc.nombreUC AS Nombre
4. ORDER BY uc.nombreUC ASC
```

Luego de realizar la consulta, se llega a un total de 70 UC, entre las cuales se destacan Programación 1 y 2. Sin embargo, para cursar Programación 3 o Programación 4 se debe haber aprobado asignaturas dictadas por el IMERL. Lo mismo ocurre a la hora de querer cursar Lógica, la cual luego será previa de muchas otras UC de la carrera. Por lo tanto, no cursar las asignaturas del IMERL es una mala idea.

4.6. Unidades Curriculares sin aprobar Cálculo 1

Cuando se ingresa a la facultad, una de las UC popularmente conocida por ser muy difícil y no permitir el avance en la carrera es Cálculo Diferencial e Integral en una Variable, o Cálculo 1 como se llamaba previamente. Análogo a la consulta anterior, interesaría saber qué asignaturas se pueden cursar en caso de no poder aprobar las distintas versiones que ha tenido esta UC.

```

1. MATCH (uc:unidadCurricular)
2. WHERE NOT
  (:unidadCurricular{codigoUC:'1061-E'})-[:esPreviaDe|agrupacionPreviaDe|
  perteneceAgrupacion*]->(uc:unidadCurricular)
  AND NOT
  (:unidadCurricular{codigoUC:'1061-C'})-[:esPreviaDe|agrupacionPreviaDe|
  perteneceAgrupacion*]->(uc:unidadCurricular)
  AND NOT
  (:unidadCurricular{codigoUC:'1052-E'})-[:esPreviaDe|agrupacionPreviaDe|
  perteneceAgrupacion*]->(uc:unidadCurricular)
  AND NOT
  (:unidadCurricular{codigoUC:'1052-C'})-[:esPreviaDe|agrupacionPreviaDe|
  perteneceAgrupacion*]->(uc:unidadCurricular)
  AND NOT
  (:unidadCurricular{codigoUC:'1020-E'})-[:esPreviaDe|agrupacionPreviaDe|
  perteneceAgrupacion*]->(uc:unidadCurricular)
  AND NOT
  (:unidadCurricular{codigoUC:'1020-C'})-[:esPreviaDe|agrupacionPreviaDe|
  perteneceAgrupacion*]->(uc:unidadCurricular)
  AND NOT
  (:unidadCurricular{codigoUC:'SRN02-E'})-[:esPreviaDe|agrupacionPreviaDe|
  perteneceAgrupacion*]->(uc:unidadCurricular)
  AND NOT (:materia)-[:requiereCreditos]->(uc:unidadCurricular)
  AND uc.dictada = 'Si'
3. RETURN uc.codigoUC AS Código, uc.nombreUC AS Nombre
4. ORDER BY uc.nombreUC ASC

```

Luego de realizar la consulta, se llega a un total de 74 UC. De forma análoga al caso anterior, sin aprobar Cálculo 1 solo se puede cursar Programación 1 y 2, dado que Cálculo es previa directamente de Programación 4 y a su vez es previa de Probabilidad y Estadística, la cual es necesaria para cursar Programación 3. En este caso, sí se puede cursar Lógica y las Matemáticas Discretas, al igual que las Geometrías y Álgebras Lineales, pero no se podrá hacerlo con Arquitectura de Computadoras y asignaturas posteriores. Por ende, tal como dicen los rumores, Cálculo 1 y sus distintas versiones son realmente indispensables para continuar con la carrera.

5. Conclusión y Trabajo Futuro

En este trabajo se cubrió el proceso entero de la creación de una base de datos de grafos, desde la obtención de los datos, pasando por el diseño y construcción del grafo, hasta la experimentación con la base de datos, mediante distintas consultas de interés.

El juego de datos obtenido incluye mucha información de interés sobre las unidades curriculares de la carrera de Ingeniería en Computación, y el modelo del grafo representa el total del sistema de previas, con todas las relaciones correspondientes.

Esto sin dudas puede ser la base de múltiples posibles trabajos relacionados al tema. En particular, un trabajo que era de interés al comienzo del proyecto era poder explorar los diferentes caminos para poder recibirse, y se habría desarrollado si el proceso de obtención de datos no hubiera demandado más tiempo del estipulado.

De la misma manera, una vez teniendo disponibles los caminos para recibirse se podrían buscar caminos con algunas particularidades, como por ejemplo ver cómo hacerlo cursando la menor cantidad de materias posibles, o cómo hacerlo en exactamente en 5 años (10 semestres).

Así mismo, las consultas detalladas en la parte 4 son apenas una primera aproximación al trabajo con esta base de datos. Por simplicidad en algunas consultas omitimos contabilizar los casos en que las previas constaran de una cantidad mínima de créditos en determinada materia. Sin dudas como trabajo posterior esto se podría refinar y tratar de una forma más elegante, ya que toda la información está disponible en los datos.