Remodelado de la base de datos relacional para el sistema SGAE desde el punto de vista no relacional

Rodrígo Cardozo - 4.669.734-0
Lucía Nocetti - 4.654.247-8
Instituto de Computación, Base de Datos No Relacionales
Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay

***Resumen*—En el presente documento se realiza un análisis teórico de la realidad del Sistema de Gestión Administrativa de la Enseñanza (SGAE) en busca de debilidades que tengan origen en el paradigma relacional que sigue hoy en día su persistencia, a través de la evaluación objetiva sobre casos de usos reales implementados por el sistema en producción desde el año 1989 y en continuo mantenimiento hasta la fecha.**

Se desarrolla un estudio acerca de cómo podría beneficiarse de un modelado políglota aplicando un procedimiento que parte de un diagrama de entidad de relación clásico y deriva en subdivisiones que permiten analizar conceptos propios de cada modelo de datos y ver cual es el más apropiado para cada situación.

Para finalizar, se realiza una comparación de los motores de datos más populares que implementan distintos tipos de persistencia, para introducir al lector en características básicas que pueden influenciar su elección al momento de implementar persistencias de este tipo.

I. Introducción al proyecto

El objetivo principal es estudiar la realidad del SGAE junto con sus restricciones y falencias identificadas por los usuarios finales con el fin de determinar si existen soluciones teóricamente más eficientes al problema, aplicando estrategias de bases de datos no relacionales.

El estudio incluye elaborar una descripción detallada de los componentes fundamentales a ser representados junto con sus relaciones, analizar las críticas en el modelo relacional hoy utilizado, y realizar un estudio teórico sobre las ventajas y desventajas de aplicar modelos de bases de datos que siguen el paradigma no relacional, tomando como referencias trabajos previos como el realizado por [Abramova et al.(2014)Abramova, Bernardino, and Furtado], [Fraczek and B(2017)], entre otros.

Basados en los requerimientos y restricciones presentes, se estudian diferentes enfoques que buscan mejorar la abstracción de la realidad, teniendo presentes la necesidad de consistencia y disponibilidad de la base. Se presenta una propuesta de persistencia políglota o pura dependiendo de los resultados que se desprenden del análisis.

Para esto, se propone un flujo de trabajo que incluye un conjunto de actividades detalladas en secciones posteriores del presente documento.

II. Realidad a modelar

El sistema institucional de la Universidad de la República (UdelaR) es la realidad que el SGAE pretende diseñar e implementar. Su estructura administrativa es de público conocimiento aunque por su tamaño, puede resultar difícil visualizarla en su totalidad. Por esto se realiza una descripción en lenguaje natural como antepaso a la presentación en Modelo Entidad-Relación. [1]

El sistema educativo contempla, además del sistema vigente dentro del Uruguay, algunos componentes internacionales debido a la posibilidad de reconocimiento y convalidación de actividades relacionadas a la enseñanza dentro del territorio nacional (se profundiza sobre esto más adelante). Es por esto que debemos reconocer las Instituciones, donde un ejemplo es la UdelaR, estas instituciones se ubican en Países.

Los Servicios pertenecen a las instituciones, estos servicios cuentan con Institutos, dictan Materias, y pertenecen a Áreas, por ejemplo, la Facultad de Ingeniería (FIng) pertenece al área tecnologías y ciencias de la naturaleza.

A su vez, los servicios ofrecen Carreras, Ciclos y CIOs (Ciclos Iniciales Optativos).

Las carreras, ciclos y CIOs tienen Planes, los cuales se van actualizando con el tiempo, por lo que hay planes vigentes y no vigentes pero interesa mantener un historial de los mismos ya que en un determinado momento pueden existir Estudiantes cursando un plan y otros cursando un nuevo plan. Además, un plan pertenece a un CIO o un ciclo o una carrera, o a ninguno.

Los planes son compuestos por materias, un plan puede contener Perfiles y a su vez, otorgan Avales de fin, que pueden ser de tipo Titulo total, Titulo parcial o Certificados.

Los avales de fin no todos son otorgados exclusivamente por la UdelaR, sino que también existen Conversiones, donde un aval puede otorgar otro aval. A su vez, las carreras pueden tener como requisito previo uno o más Bachilleratos, estos son dictados en Institutos de procedencia, como liceos, universidad del trabajo del Uruguay (UTU), etc.

Las carreras se clasifican en Universitarias y Preuniversitarias, a su vez las primeras se distinguen entre de Grado y de Posgrado

[1] Los conceptos reconcidos como entidades se escribirán con mayúscula la primera vez que son mencionados.

Las materias cuentan con un sistema de previatura el cual está conformado por un conjunto de 0 a n materias, no todas las materias se dictan en todos los institutos asociados al servicio ni todas las materias continúan vigentes o se dictan en todos los períodos, es por esto que cada materia cuenta con un conjunto de Instancias de aprendizaje, las cuales se asocian a Períodos, estos pueden ser de Desistimiento, Evaluación, Dictado o Inscripción. Las instancias de aprendizaje se dividen en diferentes tipos, estos son Seminario, Trabajo, Pasantía, Curso, Exámen y Examenparcial. Además se asocian con el o los Departamentos en los que son dictadas.

Estas instancias cuentan con Actividades o Actividades reválida. Estas últimas tienen descriptores que especifican distintos atributos de interés para cada tipo de instancia.

Para las actividades, esta es una clase abstracta que se especifica también para cada tipo de instancia. Guardan atributos como el resultado de una actividad para un estudiante.

Los estudiantes están asociados a una única instancia de Persona, en donde se guardan sus datos personales e identificatorios, lo mismo sucede con los Docentes.

Los estudiantes cursan distintas instancias de aprendizaje, para esto deben inscribirse a las mismas, por lo que existen Inscripciones. A su vez, docentes se encuentran encargados del dictado de estas instancias de aprendizaje.

Los resultados para los estudiantes cuando cursan una instancia de aprendizaje son registrados como ActividadesAbs, esto puede reflejar un resultado final o que el estudiante no se presentó a la actividad, este último caso también se relaciona a una actividadAbs pero debe ser distinguido de los demás.

Para algunos de los conceptos descriptos con anterioridad, se aplican Sanciones, estas se especifican en Sanciones de carrera, CIO, ciclo, servicio, materia, instancia de aprendizaje y UdelaR, donde cada sanción tiene una Causa asociada.

II-A. Modelo Entidad-Relación

El modelo entidad-relación (MER) asociado a la sección del SGAE que se pretende modelar se puede ver en la figura 1.

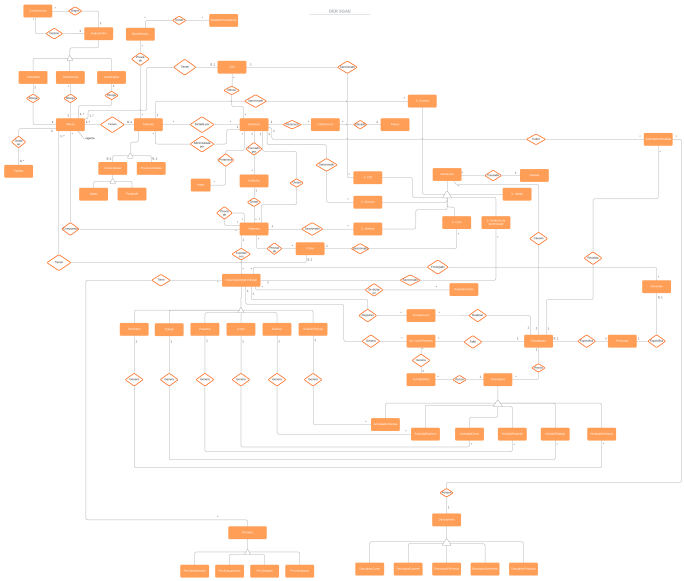


Figura 1: Modelo Entidad-Relación.

III. Críticas al sistema actual

La realidad del SGAE es muy amplia, su base de datos cuenta con más de 300 tablas y utiliza Oracle como motor. Si bien la estructura actual resuelve muchos problemas correctamente, existen algunos casos de uso del día a día para las cuales no está especialmente optimizada.

Para poder realizar un análisis acerca de la efectividad a la hora de resolver los problemas más habituales, se decidió investigar cuáles eran los casos de uso que estuvieran relacionados con el subconjunto de la realidad en la que se enfoca este estudio y cómo responde la base de datos a ellos actualmente.

Por conocimiento previo de la problemática y posibilidad de acceso a casos de uso hoy en día implementados, se realizó un estudio previo de posibles problemáticas de eficienia y sencillez que el modelo relacional presenta. Se utilizan los siguientes puntos como guías de análisis:

- ¿Resuelven el problema? (si/no) [2]
- Cantidad de tablas involucradas en las consultas y el número elevado de registros asociados.
- Cantidad de joins involucrados en las consultas
- Cantidad de subconsultas dentro de la consulta principal
- Tiempo de ejecución para obtener el resultado (segundos)
- Dificultad para obtener los datos requeridos (baja / media / alta)
- Cardinalidad [3]
- Costo [4]

Con estos criterios en mente se priorizaron 3 casos de uso que son los más comprometidos y representativos a la hora de centrar el estudio: [5]

1. **Listado de materias con la cantidad de actividades comunes o de reválida en un año seleccionado.** Retorna un listado conteniendo nombres de materias y un valor numérico referenciando a la cantidad de actividades encontradas. Si bien se resuelve con una única consulta, realmente son dos grandes consultas unidas con un *union*.

[2]Refiere a si en la implementación actual, esta consulta es resuelta por el sistema

[3]Cardinalidad: El estimativo de tuplas que se retornaron en la totalidad de los pasos realizados por la consulta

[4]Costo: Cantidad de trabajo estimada del plan basada en la cantidad de operaciones IO físicas

[5]Estos casos de uso son ilustrativos ya que no se encuentra en el alcance abarcar la totalidad de consultas posibles, pero dan una idea sólida de en qué momentos se hacen visibles las falencias del modelo relacional

¿Resuelven el problema?	Si
Cantidad de tablas involucradas en las consultas	16
Cantidad de joins involucrados en las consultas	1
Cantidad de subconsultas dentro de la consulta principal	12
Tiempo de ejecución para obtener el resultado	52,3 seg
Dificultad para obtener los datos requeridos	Media
Cardinalidad	24.766.324
Costo	208.604

2. **Cantidad de aprobados, no aprobados, presentados, no presentados a una instancia de aprendizaje en un período dado.** Debería retornar un listado conteniendo el nombre de la instancia de aprendizaje y valores numéricos representando la cantidad de aprobados, no aprobados, estudiantes que se presentaron y estudiantes que no se presentaron. Dada la estructura de la base de datos no ha sido posible obtener estos datos en una única consulta (sin incluir lógica de aplicación que intermedie). Se analiza uno de los intentos registrados, con la siguiente información:[6]

¿Resuelven el problema?	No
Cantidad de tablas involucradas en las consultas	6
Cantidad de joins involucrados en las consultas	18
Cantidad de subconsultas dentro de la consulta principal	12
Tiempo de ejecución para obtener el resultado	n/a
Dificultad para obtener los datos requeridos	Alta
Cardinalidad	n/a
Costo	n/a

3. **Listado de inscripciones de los inscriptos al curso seleccionado.** Retorna un listado conteniendo números de documento, nombres completos, e inscripciones de los estudiantes inscriptos a esa materia.

[6]Como referencia, n/a: No aplica

¿Resuelven el problema?	Si
Cantidad de tablas involucradas en las consultas	13
Cantidad de joins involucrados en las consultas	0
Cantidad de subconsultas dentro de la consulta principal	3
Tiempo de ejecución para obtener el resultado	24 seg
Dificultad para obtener los datos requeridos	Media
Cardinalidad	117.134
Costo	98.591

IV. Modelado de la peristencia políglota

Como es mencionado en la introducción, el motivo de este estudio es investigar la aplicación de modelos que potencialmente pueden simplificar la representación y entendimiento de esta realidad, en vistas de un diseño amigable a personas que no tienen un nivel de familiaridad profundo.

Para esto, se realiza el proceso de modelado para la persistencia políglota propuesto en el artículo realizado por [Zdepski et al.(2018)Zdepski, Bini, and Nasser Matos], en donde se especifican una serie de pasos, partiendo desde el modelado conceptual, para lograr identificar las necesidades de cada parte del sistema y así proponer el uso de una base de datos que mejor los resuelva. Los pasos incluyen:

- Definición de unidades de segmentación: Se toma el MER y se organizan las entidades en subsistemas (segmentos) coherentes y completamente funcionales e independientes unos de otros, que permitan tomar ventaja sobre las funcionalidades que proponen distintos modelos de base de datos.
- Definición de unidades de consistencia: Se definen cuales de los subsistemas necesita de consistencia (o propiedades ACID en general) para poder seguir siendo considerado como válidos y no causar pérdidas en el sistema general.
- Definición del modelo de datos: Se define el modelo de datos que mejor se adapte a cada unidad de segmentación. En general, más de un modelo de datos resuelve el problema pero se propone uno siguiendo prioridades definidas por el equipo.
- Diseño del modelo de datos lógicos: Tomando cada unidad de segmentación junto con su modelo de datos, se realiza el diseño lógico. Este paso escapa al alcance de este estudio pero existe amplio material de estudiado para este tema en caso de que despierte curiosidad en el lector.

Para la definición del modelo de datos a utilizar, se realiza la decisión basada en la experiencia de los autores con algunos modelos de datos conocido, y para aquellas de las cuales no se cuenta con conocimientos previos, se tomarán resultados propuestos por [Lourenço et al.(2015)Lourenço, Cabral, Carreiro, Vieira, and Bernardino].

IV-A. Resultados de la aplicación del proceso de modelado para la persistencia políglota

Siguiendo las recomendaciones descriptas en el artículo, se procede a analizar la realidad buscando determinar qué unidades de segmentación existen actualmente.

Se definieron seis unidades de segmentación bien diferenciados que cumplen con las características necesarias. Estos subsistemas se conectan entre sí a través de algunos conceptos que los atraviesan y sirven como enlace entre la información de uno y otro.

Estos son los seis subsistemas detectados: [7]

- Sistema institucional (color verde claro): Contiene toda la información acerca de Servicios, Institutos e Instituciones, Áreas y Países.
- Sistema de enseñanza (color violeta): Maneja la información acerca de Planes, Carreras, Ciclos, CIOs y Títulos.
- Sistema de Instancia de aprendizaje (color rosado oscuro): Incluye los datos acerca de las Instancias de Aprendizaje (Cursos, Exámenes, Pasantías, etc) así como también sobre los períodos de inscripción, dictados, evaluaciones y desistimiento de éstos.
- Sistema de sanciones (azul oscuro): Lleva el registro e historial de las sanciones de los estudiantes.
- Sistema de actividades (azul claro): Mantiene la información acerca de las actividades concretas llevadas a cabo por los estudiantes en relación a las instancias de aprendizaje. A su vez es responsable de retener el histórico de estas actividades.
- Sistema personas (verde oscuro): Administra la información personal de los docentes y estudiantes.

Las entidades que atraviesan y conectan los subsistemas entre sí están coloreadas de amarillo en el diagrama anexo A y refieren a Carreras, CIOs, Ciclos, Materias y Estudiantes. Son entidades que necesitan replicar identificadores en los subsistemas que las comparten para mantener la referencia y sincronización, cada entidad tendrá todos sus atributos en el sistema con el cual sean más fuertemente coherente. Por esto se dividen de la siguiente forma:

- Carreras:
  - Pertenece: Sistema de enseñanza
  - Compartida con: Sistema Institucional, Sanciones
  - Atributos duplicados: ID de carrera
- CIOs
  - Pertenece: Sistema de enseñanza
  - Compartida con: Sistema Institucional, Sanciones
  - Atributos duplicados: ID de CIO
- Ciclos
  - Pertenece: Sistema de enseñanza
  - Compartida con: Sistema Institucional, Sanciones
  - Atributos duplicados: ID de ciclo
- Materias
  - Pertenece: Sistema de Instancia de aprendizaje
  - Compartida con: Sistema de enseñanza, Sistema Institucional, Sistema de Sanciones
  - Atributos duplicados: ID de la materia
- Estudiantes
  - Pertenece: Sistema personas
  - Compartida con: Sistema de actividades, Sistema de sanciones
  - Atributos duplicados: ID de estudiante

A continuación se analizó la necesidad de consistencia que se requiere en cada subsistema, así como la frecuencia en escrituras, updates y lecturas de cada uno y el volumen aproximado de datos que deberá mantener.

IV-A1. Análisis del subsistema institucional: Características:

- Volumen de escrituras: Bajo
- Volumen de actualizaciones: Bajo
- Volumen de lecturas: Bajo
- Volumen de registros e histórico: Bajo
- Soporta consistencia eventual: Si

Dadas las características de este sistema, se determina que una buena opción es la implementación utilizando grafos, ya que maneja conceptos sencillos sin grandes volúmenes de datos, y además es esperable que las búsquedas estén orientadas principalmente a las relaciones entre ellos. En caso de que las búsquedas fueran orientadas a valores de atributos caería dentro de las desventajas mencionadas en el artículo publicado por [Neo4J(2006)] y referenciado en este estudio. La representación gráfica se puede ver en la figura 2

En este caso se descarta el uso de almacenamiento de tipo "Key-value" dado que las consultas sobre este sistema no están fuertemente relacionadas a un único concepto que pueda identificarse como *key*.

Por otro lado, implementar este sistema como una base de datos de tipo documental o relacional también sería una opción viable. Sin embargo, al ser conceptos que mantienen pocos atributos, y dado que el sistema deberá responder preguntas especialmente enfocadas en sus relaciones, se estima que una base de datos de Grafos sería la solución más sencilla y apropiada.

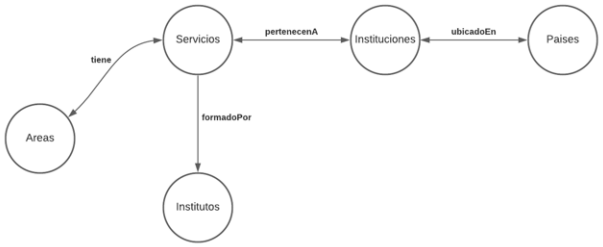


Figura 2: Representación del modelo de datos para el subsistema institucional

IV-A2. Análisis del subsistema de enseñanza: Características:

- Volumen de escrituras: Bajo

[7] El diagrama con la referencia de colores se puede ver en el apéndice A.

- Volumen de actualizaciones: Bajo
- Volumen de lecturas: Alto
- Volumen de registros e histórico: Alto
- Soporta consistencia eventual: Si

Debido a que este sistema requiere un alto nivel de lecturas, pero que a su vez soporta consistencia eventual, se entiende que una buena opción sería implementarlo utilizando bases de datos documentales. Su representación gráfica se puede ver en la figura 3

Se descarta el uso de bases de datos "Key-value" puesto que el sistema recibe consultas partiendo de distintas condiciones sobre los atributos, por lo que haría falta tener un set muy amplio y complejo de keys para que tuviera sentido y fuera adaptable a estas exigencias.

De igual forma se descarta la implementación en Grafos, ya que la mayoría de las consultas que recibirá el sistema estarán basadas en el valor de los atributos y no en la relación entre ellos, lo que resultaría costoso en este modelo (que es naturalmente indexado por relaciones y no por atributos).

Por último, si bien el uso del modelo relacional también es viable, el sistema debería incorporar muchas relaciones internamente y agregaría complejidad al sistema global, como sucede hoy en día.

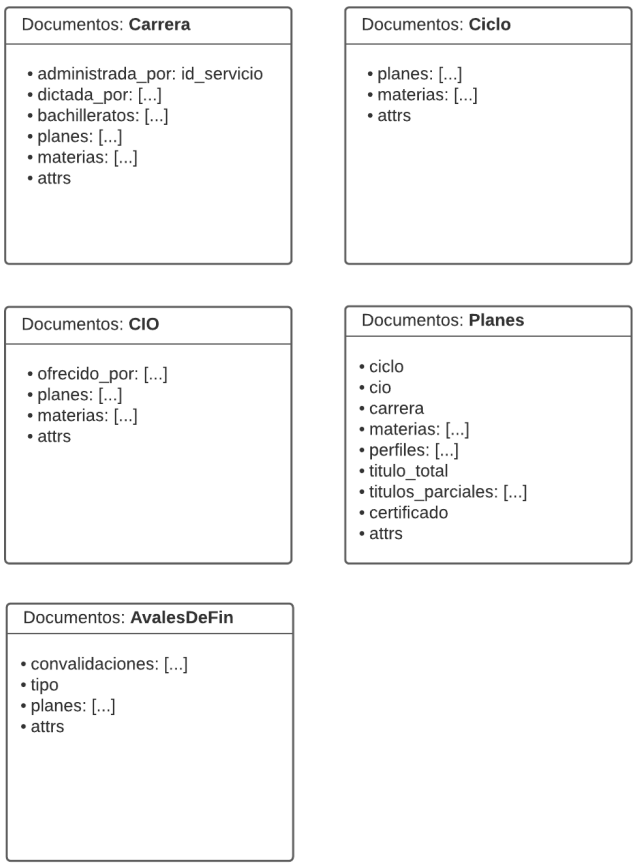


Figura 3: Representación del modelo de datos para el subsistema de enseñanza

IV-A3. Análisis del subsistema de instancias de aprendizaje: Características:

- Volumen de escrituras: Medio
- Volumen de actualizaciones: Medio
- Volumen de lecturas: Alto
- Volumen de registros e histórico: Alto
- Soporta consistencia eventual: No

Debido a la necesidad de consistencia, en este caso se determina que el modelo relacional es la mejor opción. Su representación en MER se puede ver en la figura 4.

Se descarta el uso de bases de datos de Grafos debido a que el volúmen de datos y atributos que se deberán almacenar es alto, y muchas de las consultas sobre el sistema serán sobre estos atributos, por lo cual una implementación en relacional sería más performante y sencilla en este caso. [8].

Cualquier otro modelo no relacional se descarta debido al requerimiento de consistencia.

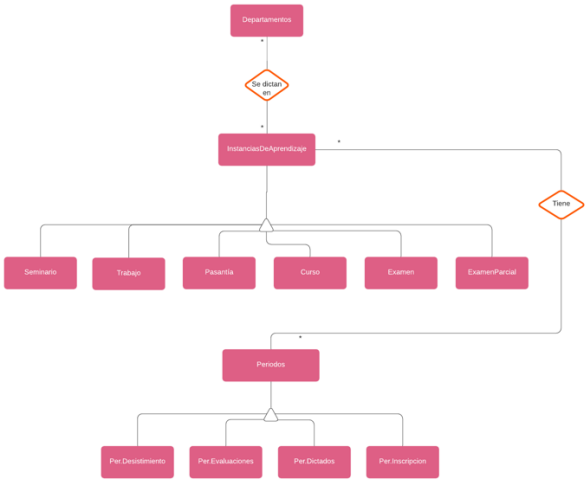


Figura 4: Representación del modelo de datos para el subsistema de instancias de aprendizaje

IV-A4. Análisis del subsistema de sanciones: Características:

- Volumen de escrituras: Medio
- Volumen de actualizaciones: Medio
- Volumen de lecturas: Medio
- Volumen de registros e histórico: Bajo
- Soporta consistencia eventual: Si

Si bien el volúmen de datos en este sistema tenderá a crecer, ya que debe mantener el histórico de sanciones de los usuarios, todas las consultas estarán siempre referenciadas a un identificador (ID) de usuario en particular, y los datos retornados viviran en un entorno muy pequeño.

Resulta difícil descartar algún modelo para la implementación de este sistema, ya que todos darían solución al problema.

[8]Si bien base de datos de grafos sigue el paradigma no relacional, este tipo de persistencias normalmente cumplen con ACID, esto es mencionado en el trabajo de [Zdepski et al.(2018)Zdepski, Bini, and Nasser Matos]

De todas formas, se recomienda un modelo key-value porque sería sencillo definir un *key* basado en el ID de estudiante, y como value almacenar una lista con el histórico de registros de sanciones. A su vez, si bien no requiere necesariamente una alta velocidad en lectura y escritura, la propiedad de flexibilidad de los modelos key-value puede ser beneficiosa a la hora de añadir un nuevo tipo de sanción. Su representación gráfica se puede ver en la figura 5.

Por otra parte, como se mencionaba anteriormente, tanto el modelo de Grafos, como el Documental o el Relacional darían solución al problema. Sin embargo, una implementación en key-value aportaría sencillez y flexibilidad en comparación; Especialmente en contraste con el modelo Relacional debido a la cantidad de tablas involucradas y a la complejidad de las consultas de actualización.

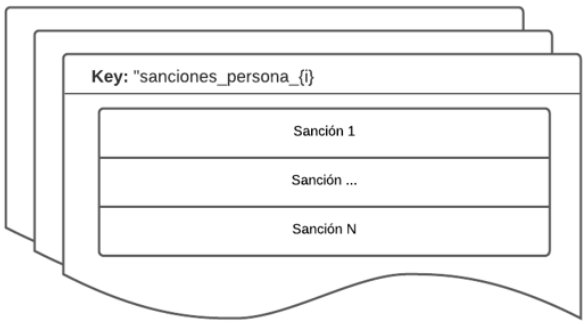


Figura 5: Representación del modelo de datos para el subsistema sanciones

IV-A5. Análisis del subsistema de actividades: Características:

- Volumen de escrituras: Alto
- Volumen de actualizaciones: Bajo
- Volumen de lecturas: Alto
- Volumen de registros e histórico: Alto
- Soporta consistencia eventual: No

Al igual que en sistema de instancias de aprendizaje, el requerimiento de consistencia hace del modelo relacional la mejor opción para este caso. Su representación gráfica se puede ver en la figura 6

Se descarta también el uso de bases de datos de Grafos nuevamente debido a la gran cantidad en volúmen de datos y atributos que se deben almacenar en el historial.

Por último, los demás modelos no se priorizan por la necesidad de consistencia.

IV-A6. Análisis del subsistema de personas: Características:

- Volumen de escrituras: Medio
- Volumen de actualizaciones: Bajo
- Volumen de lecturas: Alto
- Volumen de registros e histórico: Alto
- Soporta consistencia eventual: Si

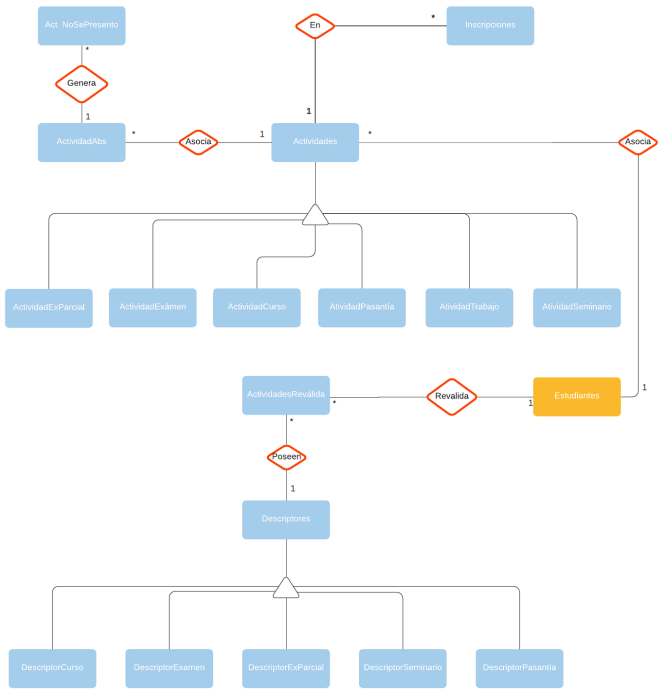


Figura 6: Representación del modelo de datos para el subsistema sanciones

En este caso, tanto key-value, como relacional o documental resolverán bien el problema. Se sugiere Key-value como modelo debido a la forma en la que se consumen los datos y la velocidad para extraerlos. La recuperación de datos de estos estudiantes se haría rápidamente a través de una búsqueda usando como key una combinación entre el ID, el número de documento y el nombre. La representación gráfica se puede ver en la figura 7.

Se descarta el uso de Grafos ya que este sistema cuenta con un único concepto (Personas) que no mantiene relaciones internas con otros conceptos y que requiere almacenar, además, un gran volumen de datos (en crecimiento constante) y atributos.

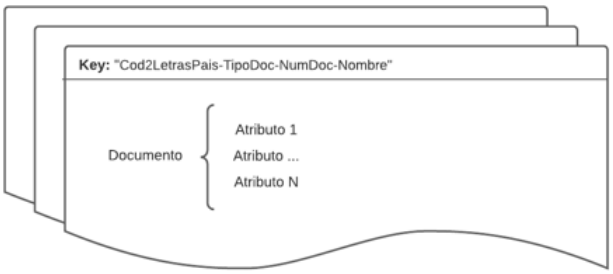


Figura 7: Representación del modelo de datos para el subsistema personas

V. Mejoras introducidas.

Sobre las consultas señaladas en III, los subsistemas que involucran son Actividades e Instancias de aprendizaje, estos fueron analizados y determinados como mejor aplicables al modelo relacional, por lo que mejoras son posibles si se modifica la estructura general del modelo entidad relación,

Para la consulta 1, la entidad Materia se encuentra replicada en varios subsistemas, asumiendo que Materias se conecta con Actividades y ActividadesReválida, esta consulta implica recorrer la tabla Actividades una vez en su totalidad preguntando por el año y hacer un *Join* con la tabla Materias para retornar su nombre, contabilizando cuantas tuplas con esta condición se repiten, lo mismo para la tabla ActividadesReválida.

Para la consulta 2, eliminando la entidad Act.NoSePresentó y agregando un atributo booleano indicando esta característica dentro de la entidad ActividadAbs, la consulta se resolveria realizando dos pasos, uno que retorne los identificadores de las instancias de apendizaje realizadas en un período dado, y luego con este identificador, se realiza una búsqueda dentro de la tabla ActividadAbs, contabilizando este booleano para identificar los presentados de los no presentados, y los aprobados/desaprobados se contabilizan comparando la nota resultado con el mínimo de aprobación.

En conclusión, la consulta se limita a una recorrida sobre la tabla relacional Tiene (que une las entidades Instancia de aprendizaje y Período), y realizando un *Join* con estos identificadores y la tabla ActividadAbs, contabilizando los atributos de interés. Se recorre la tabla relacional Tiene por su atributo indexado.

Por último, sobre la consulta 3, como Estudiantes es una entidad compartida, se puede asumir una relción directa entre Estudiante e instancias de aprendizaje, por lo que dado un curso, se recuperan los estudiantes asociados a ese curso y con sus IDs, se recorre una vez la tabla Inscripciones retornando todas aquellas que estén asociados a ellos.

Sobre la solución global, las consultas que se resuelven dentro de cada subsistema tendrá tanta eficiencia como el modelo de datos elegido lo permita y el equipo de implementación sepa explotar. Sobre las consultas que se resuelven interconectando subsistemas, se necesita la intervención de capa de aplicación para lograr esta sincronicidad pero se considera un costo menor tomando en cuenta las ventajas que supone la persistencia políglota.

VI. Motores de base de datos

Luego del análisis anterior, la siguiente consigna naturalmente es qué motores de base de datos utilizar para cada uno de los modelos de datos seleccionados.

Recapitulando, se tiene que:

- Subsistema insitucional: Modelo de datos en grafos.
- Subsistema de enseñanza: Modelo de datos en documentos.
- Subsistema de instancias de aprendizaje: Modelo de datos relacional.
- Subsistema de sanciones: Modelo de datos key-value.
- Subsistrmas de actividades: Modelo de datos relacional.
- Subsistema de personas: Modelo de datos key-value.

El fin de este estudio es centrardo en las persistencias no relacionales, por lo que se compara dos de los motores más populares para cada tipo. Cabe destacar que cualquier opción podría responder a la demanda de manera eficiente, por lo que la elección definitiva dependerá de los distintos equipos de trabajo y restricciones propias de la empresa, como los costos de las licencias, que no tratan específicamente sobre performance o implementación de los motores.

VI-A. Base de datos documentales: MongoDB y Couchbase

Ambos son sistemas de manejo de base de datos (DBMS) de código abierto, permiten ser manejados por servidores en la nube o en infraestructura local. Son compatibles con los 3 sistemas operativos (OS) más utilizados (Linux, Windows, MacOS), pero MongoDB adiciona Solaris.

Ambos cuentan con un esquema de datos libre, su método de particionamiento es de fragmentación (*Sharding)*, soportan consistencia eventual e inmediata que puede ser definida para cada operación. Las transacciones para Couchbase se basan en las propiedades de Atomicidad, Consistencia, Aislamiento y Durabilidad (ACID), mientras que MongoDB implementa ACID multidocumento con aislamiento de instantáneas (*Snapshot Isolation)*.

Los lenguajes de programación soportados son varios, Couchbase soporta 14 lenguajes mientras que MongoDB soporta 29.

Entre las diferencias, MongoDB ofrece resplados continuos con consistencia *cross cluster* y puntos de recuperación en el tiempo mientras que esto no está presente en Couchbase. Sobre la manera de escribir las consultas, Couchbase adaptó un lenguaje de consulta derivado de SQL llamado N1QL, el cual se asemeja en gran medida al formato estándar, esto potencialmente disminuiría la curva de aprendizaje necesaria para desarrolladores que quieran realizar la transición a este modelo de datos desde SQL. Un ejemplo de esto se puede ver en la figura 8. [9]

VI-B. Base de datos key-value: DynamoDB y Riak KV

DynamoDB es un servicio alojado en la nube el cual se encuentra bajo el dominio de Amazon y se presenta como una opción que permite gran escalabilidad por no ser alojado localmente. Además, esto da flexibilidad sobre los sistemas operativos con los cuales es compatible, ya que depende de la comunicación que realice la aplicación desarrollada, por lo tanto es utilizable independientemente del OS con el cual se trabaje.

Riak KV en cambio, es un sistema distribuído que puede ser alojado tanto localmente como en un sistema *multicluster* y es compatible con Linux y MacOS.

Ambos cuentan con esquema de datos libre. La indexación por índices secundarios para Riak KV es restringida, en contraparte con DynamoDB donde es comletamente permitida.

[9]Imágen extraída de la página oficial de Couchbase. URL: https://www.couchbase.com/comparing-couchbase-vs-mongodb-1

Los lenguajes de programación que soportan son más variados para Riak KV, DynamoDB no soporta ninguna variante de C, las cuales siguen siendo ampliamente utilizadas en el mercado.

El método de particionamiento para ambos es *Sharding*. Sobre consistencia, solo DynamoDB puede ofrecer consistencia inmediata, mientras que Riak KV se limita a consistencia eventual, es por esto que no cumple las propiedades ACID.

Aunque a nivel de popularidad, DynamoDB presenta superioridad, este sistema es únicamente accesible por licencia comercial con Amazon y su código no es abierto. Por otra parte, Riak KV comparte la misma característica de MongoDB y Couchbase de ser de código abierto.

VI-C. Base de datos de grafos: Neo4J y Virtuoso

Muchos de los DBMS de grafos soportan multiples modelos de datos, y este es el caso de Virtuoso, siendo un híbrido que soporta el manejo de datos representado como tablas relacionales y/o grafos. En este aspecto, Neo4J es rígido y únicamente soporta la estructura de grafos.

Ambos son de código abierto y permiten instalación local y en la nube. En caso de optar por una instalación local, Neo4J puede ser levantado en los mismos sistemas operativos que MongoDB, pero Virtuoso adiciona los OS Aix, FreeBSD y HP-UX.

Los esquemas de datos son libres u opcionales para Neo4J, mientras que Viruoso soporta 4 esquemas, entre ellos los más familiares resultan el relacional estándar y el esquema XML. Por esta razón, la base de datos híbrida soporta SQL, al contrario de la base puramente de grafos la cual implementa una sintaxis propia no adaptable a SQL. Ambos DBMS cumplen ACID en sus transacciones, Como aclaración, Neo4J soporta además consistencia eventual y casual, la cual es configurable.

Sobre los lenguajes de programación soportados, aunque Neo4J tiene un mayor número, al igual que DynamoDB, no soporte ninguna variante de C. Independientemente de esto, dentro de esta categoría, por su popularidad sigue siendo la primer opción para implementaciones de este tipo.

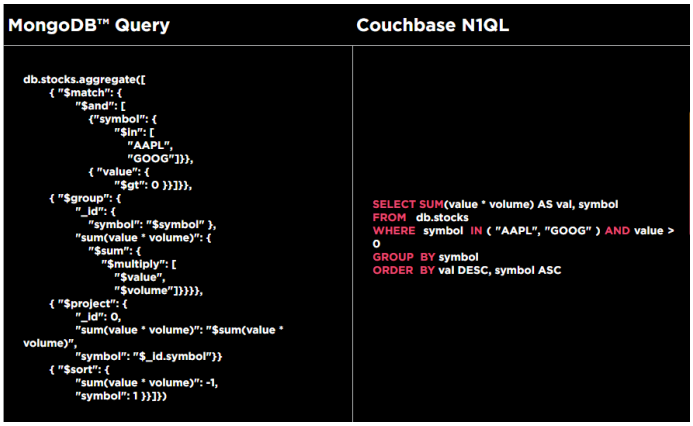


Figura 8: Comparativa del lenguaje de consulta entre MongoDB y Couchbase

VII. Conclusiones y trabajo futuro

El trabajo presentado refleja el potencial de adaptación de las bases de datos no relacionales a problemas que fueron concebidos como relacionales, por bien ser el estándar de facto en le época o por la familiaridad que los implementadores sienten con este tipo de persistencias.

A lo largo del estudio, se vió que no solo el aporte se toma por el lado de la simpleza y sencillez que se puede alcanzar, sinó que puntos que pueden resultar fundamentales para un proyecto, como lo son la performance, disponibilidad y particionamiento son posibles utilizando modelos de base de datos pensados con estas características.

Si bien se hace foco en la adaptabilidad que los modelos no relacionales presentan, cabe destacar que el modelo relacional sigue siendo aplicable y justificado para un amplio número de problemáticas, manteniendo cualidades deseables como lo son las propiedades ACID. Los modelos no relacionales no se sobreponen a estas, sinó que las sacrifican en pos de adaptarse a nuevas problemáticas introducidas por el creciente mundo de la información, donde los grandes volúmenes de datos toman cada vez un mayor campo de uso y estudio y son un pilar fundamental para el funcionamientos de herramientas de uso masivo por la persona común, como lo son las redes sociales. Un estudio interesante del tema es el artículo realizado por [Pwint Phyu and Zhaoshun(2019)].

El alcance de este estudio fue acotado y focalizado, es por esto que el pasaje de una persistencia relacional a una políglota puede parecer un proceso considerablemente directo, pero esto tiende a complejizarse dependiendo del contexto en donde se pretende aplicar. Además, el proceso de modelado aplicado es una propuesta que no se establece como formal, pero es una recomendación realizada por varios autores que pretenden atacar la traducción entre relacional a no relacional o políglota.

En trabajos futuros se pretende ahondar en las características que identifican y dan personalidad a cada tipo de modelo de datos, con el fin de individualizar en pasos más profundos las sitaciones que hacen prioritaria la elección de uno sobre otro. Además, por la tendencia creciente de datos de sistemas actualmente en producción que se encuentran basados en persistencias relacionales, el estudio sobre traducción a modelos no relacionales se entiende como pilar fundamental ya que abriría puertas de mejoras potenciales.

Referencias

[Abramova et al.(2014)Abramova, Bernardino, and Furtado] Veronika Abramova, Jorge Bernardino, and Pedro Furtado. Which NoSQL Database? A Performance Overview. *Open Journal of Databases*, 1(2): 17–24, 2014. URL https://estudogeral.sib.uc.pt/bitstream/10316/27748/1/WhichNoSQLDatabase.pdf.

[Fraczek and B(2017)] Konrad Fraczek and Malgorzata Plechawska-wojcik B. Comparative Analysis of Relational and Non-relational Databases in the Context of Performance in Web Applications. volume 716, pages 153–164, 2017. ISBN 978-3-319-58273-3. doi: 10.1007/978-3-319-58274-0. URL http://link.springer.com/10.1007/978-3-319-58274-0.

[Lourenço et al.(2015)Lourenço, Cabral, Carreiro, Vieira, and Bernardino] João Ricardo Lourenço, Bruno Cabral, Paulo Carreiro, Marco Vieira, and Jorge Bernardino. Choosing the right NoSQL database for the job: a quality attribute evaluation. *Journal of Big Data*, 2(1):1–26, 2015. ISSN 21961115. doi: 10.1186/s40537-015-0025-0. URL http://dx.doi.org/10.1186/s40537-015-0025-0.

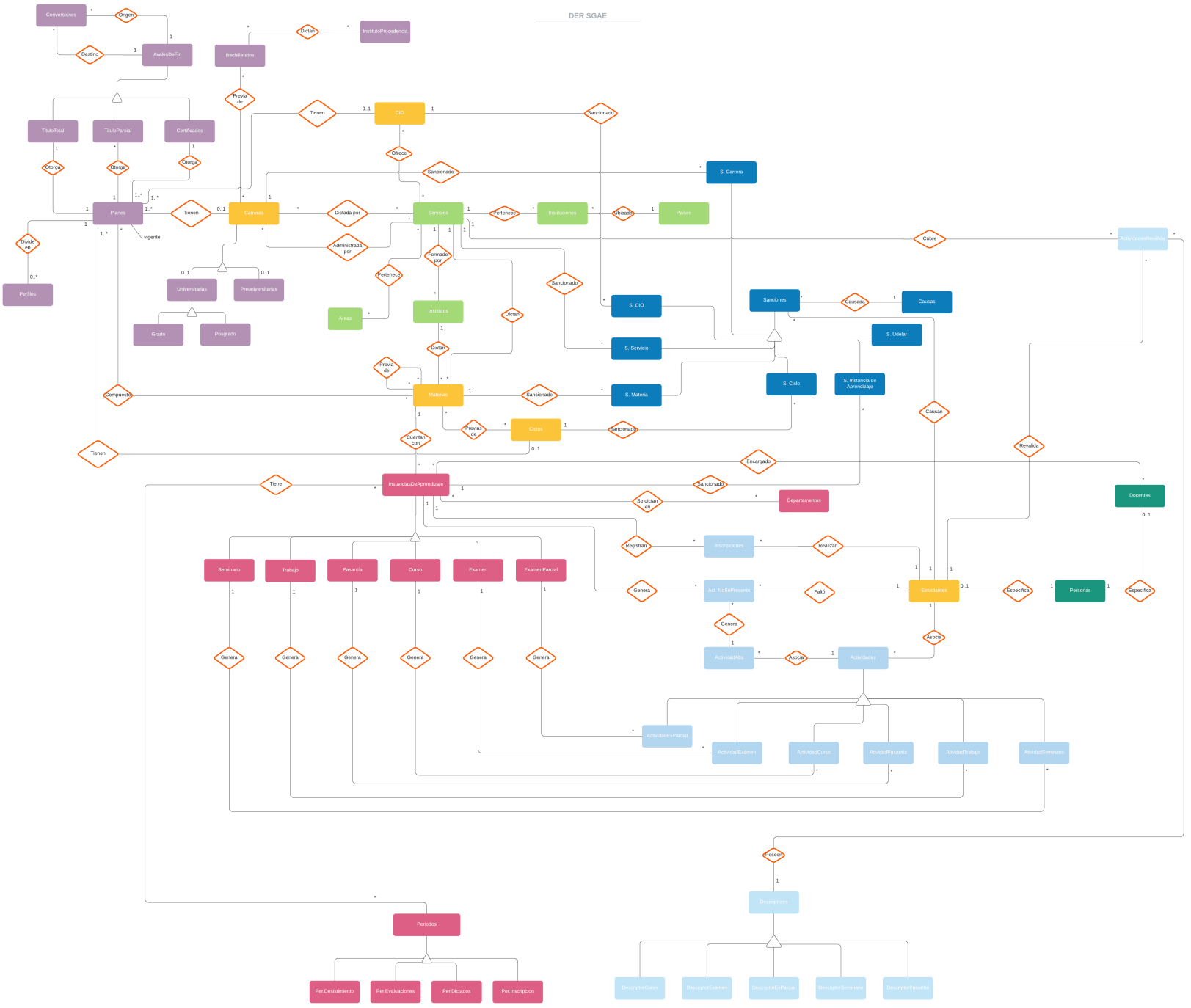
[Neo4J(2006)] Team Neo4J. The Neo Database – A Technology Introduction (20061123). *2006 Neo Database AB*, 2006. URL http://dist.neo4j.org/neo-technology-introduction.pdf.

[Pwint Phyu and Zhaoshun(2019)] Khine Pwint Phyu and Wang Zhaoshun. A review of polyglot persistence in the big data world. *Information 2019*, pages 1–24, 2019. doi: 10.3390/info10040141.

[Zdepski et al.(2018)Zdepski, Bini, and Nasser Matos] Cristofer Zdepski, Tarcizio Alexandre Bini, and Simone Nasser Matos. An Approach for Modeling Polyglot Persistence. *Proceeding of the 20th International Conference on Enterprise Information Systems (ICEIS 2018)*, pages 120–126, 2018.

APÉNDICE

A - Se presenta el modelo entidad relación del sistema de gestión de la enseñanza referenciado en la sección IV-A, junto con su división en subsistemas o unidades de segmentación, referenciado por colores.


DER SGAE
Conversiones
Origen
Destino
AvalesDeFin
InstitutoProcedencia
Dictan
Bachilleratos
Previa de
TituloTotal
TituloParcial
Certificados
Otorga
Tienen
CIO
Sancionado
Ofrece
Sancionado
S. Carrera
Planes
vigente
Tienen
Carreras
Dictada por
Servicios
Pertenece
Instituciones
Dictado
Paises
Divide en
Perfiles
Administrada por
Formado por
Pertenece
Cubre
ActividadesRevalida
Universitarias
Preuniversitarias
Grado
Posgrado
Areas
Institutos
Dictan
Sancionado
S. CIO
Sanciones
Causada
Causas
S. Udelar
Dictan
S. Servicio
Previa de
Compuesto
Materias
Sancionado
S. Materia
S. Ciclo
S. Instancia de Aprendizaje
Causan
Previas de
Ciclos
Sancionado
Cuentan con
Revalida
Tienen
Encargado
Tiene
InstanciasDeAprendizaje
Sancionado
Se dictan en
Departamentos
Docentes
Registran
Inscripciones
Realizan
Seminario
Trabajo
Pasantia
Curso
Examen
ExamenParcial
Genera
Act. NoSePresento
Faltó
Estudiantes
Especifica
Personas
Especifica
Genera
Asocia
ActividadAbs
Asocia
Actividades
Genera
Genera
Genera
Genera
Genera
Genera
ActividadExParcial
ActividadExamen
ActividadCurso
AtividadPasantia
AtividadTrabajo
AtividadSeminario
Poseen
Descriptores
Periodos
Per.Desistimiento
Per.Evaluaciones
Per.Dictados
Per.Inscripcion
DescriptorCurso
DescriptorExamen
DescriptorExParcial
DescriptorSeminario
DescriptorPasantia