

Propuesta curricular holística para asignaturas sobre bases de datos

Arturo Servetto

Universidad de Buenos Aires. Facultad de Ingeniería. Departamento de Computación. Buenos Aires, Argentina.
aservetto@fi.uba.ar

Resumen. Se presenta un currículo holístico para una asignatura cuatrimestral de Bases de Datos, correspondiente a los Contenidos Curriculares Básicos del bloque de descriptores de conocimiento de Tecnologías Aplicadas para carreras de Ingeniería en Informática, como también de Fundamentos y aplicaciones de Bases de Datos del trayecto de formación Ingeniería de Software, Base de Datos y Sistemas de Información para las carreras de Licenciatura en Informática, según los estándares de la CONEAU.

En la FIUBA, la asignatura de referencia es común en sus dos carreras, y se encuentra en el octavo cuatrimestre de la de Ingeniería y en el séptimo de la de Licenciatura, y los estudiantes que la cursan tienen conocimientos previos de organización y recuperación de datos persistentes y conocimientos propedéuticos y básicos de métodos y procesos de Ingeniería de Software.

La organización de contenidos es descendente en orden de abstracción, y su desarrollo y la evaluación de competencias asociadas se sustenta en actividades formativas y en el Aprendizaje Basado en Proyectos.

El holismo del currículo se asocia a la integración curricular con otras asignaturas de los planes de estudio mediante el uso de UML para el Diseño Conceptual, común con la Ingeniería de Software, y el anclaje de actividades formativas de Diseño Lógico en fundamentos de organización de datos, así como a la integración de los contenidos específicos al vincularlos en proyectos integrados.

Palabras clave: Diseño Curricular, Diseño Conceptual de Bases de Datos con UML, Actividades Formativas para el Desarrollo de Competencias y la Evaluación Continua, Aprendizaje Basado en Proyectos.

1 Introducción

Los planes de estudio de carreras de informática, tanto de Ingeniería como de Licenciatura, de universidades argentinas incluyen asignaturas sobre Bases de Datos cuyos diseños curriculares, en algunos casos, exhiben falta de cohesión de objetivos y contenidos con los de otras asignaturas correlativas previas como asimismo con otras del mismo bloque de Tecnologías Aplicadas [1] en planes de ingenierías o trayecto de

formación en Ingeniería de Software, Base de Datos y Sistemas de Información en planes de licenciaturas [2].

Asimismo, se suele encontrar desconexión entre los mismos contenidos de la asignatura, por desarrollarse según el modelo de aprendizaje por dominios en lugar del modelo en espiral de Jerome Bruner [3]. La carencia de integración puede verificarse en la enunciación de objetivos generales que orientan el desarrollo de las cursadas, y de competencias que los alumnos deben desarrollar que orientan el diseño de actividades prácticas, así como en el lenguaje de modelado conceptual. Esta misma falta de integración de contenidos con los de otras áreas curriculares o entre los específicos de Bases de Datos se puede notar en mucha bibliografía.

Tanto en bibliografía específica como en diseños curriculares de asignaturas de Bases de Datos se sigue insistiendo en el uso del lenguaje de modelado de Entidades y Relaciones del Dr. Peter Chen (de 1977) [4], [5], [6], [7], propio del paradigma de desarrollo estructurado de software, mientras que en Ingeniería de Software se usa el paradigma de objetos, o peor aún, si se usa el UML para el diseño de datos, se usan primitivas de UML propias de la fase de diseño del Proceso Unificado de Desarrollo, de nivel de abstracción propio del diseño lógico, en lugar de usar primitivas exclusivas de la etapa de análisis, de nivel apropiado para el diseño conceptual [7], [8]. Y para el abordaje de modelos lógicos es raro encontrar comparaciones o evaluaciones en función de la organización y recuperación de datos persistidos para ponderar opciones de derivación del modelo conceptual.

En cuanto a las unidades o bloques temáticos del mismo campo Bases de Datos, se suele encontrar, en bibliografía y currículos de asignaturas, sesgos hacia el modelo de aprendizaje por dominios en las propuestas de numerosos ejercicios para el desarrollo de competencias exclusivas sobre un mismo tema, sin articular con otros. La disociación puede también producirse por una organización ascendente de temas, a la inversa del proceso que debe seguirse en el ejercicio profesional: se comienza con la manipulación de datos con lenguajes relacionales y SQL, por requerir el desarrollo de competencias de nivel de aplicación, según la Taxonomía de Bloom revisada en 2001 [9], que es de orden inferior, para continuar con el modelado conceptual y lógico, por requerir el desarrollo de competencias de orden superior como analizar información y crear modelos de datos [4], [5], [6], [7], [8].

La manipulación de datos con lenguajes relacionales se desarrolla, a menudo, excluyendo al Cálculo Relacional de Tuplas [4], [5], [7] o dándole escasa relevancia [6], [8], y disociados de SQL, que fue definido precisamente en base al Álgebra Relacional y al Cálculo Relacional de Tuplas.

Podría afirmarse que un enfoque holístico para el diseño curricular debería relacionar contenidos con los de otras asignaturas de los planes de estudio, así como los contenidos propios entre sí. La vinculación con Ingeniería de Software, en cuyas asignaturas introductorias se aborda el diseño de datos pero con una profundidad e intensidad en la formación práctica que resultan propedéuticas para su abordaje en Bases de Datos, puede vehicularse con el uso de UML para el Diseño Conceptual, pero acotando las primitivas a las propias de la etapa de análisis del Proceso Unificado de Desarrollo: clases entidad, que sólo se pueden relacionar mediante asociaciones

simples o entidades asociativas [10]. La asociación de contenidos con los del bloque de tecnologías básicas en carreras de Ingeniería, o con Lenguajes, Algoritmos y Estructuras de Datos del trayecto de formación Algoritmos y Lenguajes en carreras de Licenciatura, podría anclarse en actividades formativas de Diseño Lógico orientado por consideraciones de organización y recuperación de datos persistidos así como por frecuencia o relevancia de consultas y sus secuencias de acceso a distintas tablas, que también implica la integración de contenidos propios.

Y la integración de los contenidos específicos de Bases de Datos se puede lograr mediante una organización descendente de temas, que se pueden vincular en proyectos integradores: se comienza por el diseño conceptual de datos, se derivan modelos conceptuales a modelos lógicos relacionales a la par de considerar operaciones a nivel de requerimientos funcionales que permitan inferir secuencias de navegación de tablas, se sigue con la especificación de operaciones y consultas en Álgebra Relacional, ampliada para permitir la expresión de consultas de agregación, y Cálculo Relacional de Tuplas, y se finaliza con la traducción literal de las expresiones en lenguajes relacionales a SQL.

El desarrollo de competencias específicas así como la evaluación formativa continua del aprendizaje activo se puede garantizar mediante actividades formativas [11], [12], previas a las etapas correspondientes del progreso de proyectos, que se pueden desarrollar y supervisar durante las clases.

2 Currículo de asignatura

La asignatura introduce los principios conceptuales y procedimentales necesarios para diseñar, implementar y utilizar bases de datos mediante Sistemas de Gestión de Bases de Datos Relacionales. El abordaje acentúa los principios básicos del modelado y el diseño de bases de datos, así como los lenguajes y servicios proporcionados por los sistemas de gestión de procesamiento de transacciones y de procesamiento analítico, incluyendo los problemas tecnológicos involucrados y las técnicas usuales para resolverlos.

2.1 Régimen de cursado y promoción

El cursado es de dos clases semanales de 3 horas cada una durante 16 semanas y el régimen de promoción consiste en el desarrollo de actividades formativas grupales, para el desarrollo de competencias transversales o genéricas como la organización de trabajo en grupo, coevaluación, etc., y evaluaciones conceptuales individuales por núcleos temáticos con una instancia al finalizar el desarrollo de cada núcleo, y recuperatorios en períodos de evaluaciones finales.

Para el abordaje de los temas se secuencia el desarrollo de un caso de estudio en forma incremental, que dependiendo de la modalidad de clases (presenciales, virtuales o mixtas) puede ser interactivo con los alumnos o publicado como modelo a efectos de establecer convenciones de notación o representación. Para cada clase práctica se plantea una actividad de desarrollo grupal que puede comunicarse el día previo, y desarrollarse y revisarse en la misma clase (si es presencial) o antes de la siguiente

actividad (para clases o seguimientos virtuales), instrumentando así un sistema de evaluación formativa continua orientado a reflexionar y comunicar información sobre los procesos y resultados de la actividad de enseñanza-aprendizaje para ajustarla a los objetivos y resultados que se esperan, y que pueden comprender modalidades de auto-evaluación, coevaluación y evaluación entre pares (evaluar para aprender).

Las actividades formativas supervisadas se diseñan para conformar uno o dos proyectos con etapas de desarrollo incremental, de manera de instrumentar un enfoque de aprendizaje orientado por proyectos, y la promoción de la cursada se puede obtener mediante el desarrollo de un proyecto de mayor complejidad, con las mismas etapas de el o los proyectos que integran las actividades formativas, pero retrasadas para que comiencen luego de tener las devoluciones de las actividades correspondientes.

2.2 Objetivos

En la Tabla 1 se detallan los objetivos en términos de las competencias que se espera desarrollen los estudiantes.

Tabla 1. Desarrollo de competencias.

Tipo	Competencias
Cognitivas (saber)	Conceptos del modelo de datos relacional. Fundamentos de sistemas de gestión de bases y almacenes relacionales de datos.
Procedimentales o instrumentales (saber hacer)	Abstraer y diseñar los datos de un sistema de información utilizando el modelado conceptual. Utilizar las herramientas proporcionadas por un SGBDR para la creación, operación y control de bases de datos. Desarrollar e implementar pruebas que demuestren la validez y las bondades de un diseño. Derivar esquemas relacionales a partir de modelos conceptuales de bases de datos. Expresar consultas en Álgebra Relacional y Cálculo Relacional de Tuplas y derivarlas a SQL.
Actitudinales (ser)	Capacidad para crear diseños (creatividad). Preocupación por la eficacia. Preocupación por la eficiencia. Debatir y concluir las distintas soluciones a un problema.
Transversales o genéricas	Desarrollar capacidad de análisis, síntesis, evaluación y creación. Desarrollar capacidad de organizar y planificar. Resolver problemas. Trabajar en equipo. Capacidad de aplicar los conocimientos en la práctica.

2.3 Contenidos

En la Tabla 2 se muestran los contenidos organizados en núcleos temáticos y en el orden en que se abordan.

Tabla 2. Contenidos.

Núcleo temático	Temas y actividades
Modelado conceptual	Diseño conceptual de bases de datos. Clases entidad, estereotipos, relaciones entre clases y jerarquías. Identificación explícita de clases entidad e implícita de relaciones. Atributos descriptivos simples y compuestos, monovalentes y polivalentes, enumeraciones y dominios actualizables.
Modelado lógico	Modelo lógico relacional de bases de datos. Normalización de atributos de entidades y de relaciones para la validación de diagramas conceptuales. Derivación de diagramas conceptuales a esquemas relacionales.
Lenguajes relacionales	Cálculo relacional de tuplas (CRT) sobre esquemas relacionales. Álgebra relacional (AR). SQL. Traducción de consultas de CRT y AR a SQL.
Sistemas de gestión de bases de datos relacionales	Sistemas de gestión de bases de datos relacionales (SGBDR). Transacciones y optimización de consultas en SGBDR. Seguridad de bases de datos.
Almacenes de datos	Sistemas de procesamiento analítico de datos. Esquemas relacionales de almacenes con estructura de estrella, de copo de nieve y de constelación. Transformación de modelos lógicos de sistemas de procesamiento transaccional a esquemas relacionales de datos detallados de almacenes. Especificación de esquemas y operaciones para datos leve y altamente resumidos.

Debido a que el SQL se define en base al CRT y al AR, que determinan variantes de operaciones de manipulación basadas en subconsultas o en emparejamiento de tuplas (*joins*), respectivamente, se adopta una extensión del AR [8] para contemplar operaciones de productos externos y agregaciones, de manera de sistematizar el modelado de operaciones sobre esquemas relacionales en dos pasos incrementales, para que las expresiones en SQL se obtengan mediante una derivación de una expresión en un lenguaje relacional.

Para el proyecto fuera de horario de clases que permite la promoción de la cursada, en su etapa final, se solicita la definición y carga inicial con datos de prueba de una base de datos con MySQL, y la definición de operaciones usando disparadores (*triggers*), procedimientos almacenados (*stored procedures*) y cursores.

3 UML para el diseño conceptual de bases de datos

Diseñar datos a nivel conceptual implica detectarlos en el sistema de información para el que se está desarrollando el sistema informático que los gestionará y especificar vínculos entre los mismos, en un proceso de abstracción que se denomina clasificación: se trata de identificar datos que representen cosas (clientes, productos, servicios, etc.), eventos (pedidos y ventas de productos, reservas y ventas de servicios, etc. – tienen fecha de ocurrencia) y procesos (trámites, órdenes de servicio – tienen un esta-

do que cambia cronológicamente), así como de identificar vínculos entre ellos, y asignarles un conjunto o clase de pertenencia, que en el caso de que sean datos se denomina “entidad”, y en el caso de que sean vínculos, “relación”.

Las entidades que agrupan representaciones de cosas se dicen “maestras”, y las que agrupan representaciones de eventos o de procesos, “transaccionales”. Se define una extensión del modelo de clases de UML agregando estereotipos de clases <<Cosa>>, <<Evento>> y <<Proceso>>, así como un estereotipo de asociaciones <<Id>>, para denotar relaciones de dependencia que proveen identificación a la entidad dependiente. El uso de formas de modelos de clase de UML se acota a las de modelado en la etapa de análisis del Proceso Unificado de Desarrollo, es decir, se excluyen las composiciones y agregaciones, que son propias la etapa de diseño, y no se asignan tipos a los atributos.

El modelado conceptual es un proceso evolutivo incremental, en el que se va resolviendo la complejidad mediante refinamientos sucesivos y modularización. Se sigue un proceso con los siguientes pasos de refinamiento en forma evolutiva:

1. Individualizar y representar entidades y relaciones, determinando cardinalidades de las entidades en las relaciones, subconjuntos de especialización que resulten evidentes, y coberturas de jerarquías (si las entidades especializadas cubren total o parcialmente a la entidad general, y si representan conjuntos de datos disjuntos o superpuestos).
2. Representar atributos que formen parte de identificadores, y verificar:
 - Que toda entidad tenga al menos un identificador, y que los subconjuntos de especialización hereden adecuadamente identificadores más allá de que puedan tener uno o más propios.
 - La consistencia de las cardinalidades de las entidades en las relaciones, cuidando que los vínculos que representen las relaciones se puedan identificar exclusivamente a partir de identificadores de una o ambas entidades.
3. Determinar los restantes atributos característicos de las entidades y de las relaciones que los requieran, denotando cardinalidades especiales, y analizar también
 - Si se detectan más identificadores (para completar el modelo).
 - Si se detectan más generalizaciones o especializaciones a partir del análisis de atributos comunes en distintas entidades (para generalizar) o de atributos opcionales en una misma entidad (para especializar); entidades con más de una relación con cardinalidad mínima 0 (para especializar).
 - Si se puede agrupar atributos en atributos compuestos, si un subconjunto de atributos de una entidad representa una misma característica o propiedad (mejorar expresividad).
4. Evaluar la completitud y consistencia del diagrama obtenido:

- Verificar que todas las entidades tengan al menos un identificador, aunque sea heredado, y que tengan todos los identificadores posibles.
- Verificar que todas las entidades tengan denotada la cardinalidad en cada relación que participen.
- Detectar y corregir redundancias: ninguna entidad puede tener como componentes atributos que sean propios de otra entidad existente en el modelo; esto sólo puede representarse mediante una relación con la otra entidad. Los únicos casos que no implican redundancia o que implican redundancia justificada son la copia de atributos de una entidad maestra a una relación transaccional o transacción dependiente de la misma (por ejemplo, el precio de productos o servicios: no hay redundancia porque el precio de un producto o servicio puede cambiar con el tiempo, pero el precio al que se vendió en una transacción no); y los atributos derivados.
- Verificar que los subconjuntos de especialización de una jerarquía sean todos correspondientes a entidades del mismo tipo (todas deben representar conjuntos de cosas o conjuntos de eventos) y que no representen procesos en distinto estado (ya que esto significaría cambiar a un dato de conjunto cada vez que cambie su estado, y es funcionalmente ineficiente).
- Detectar y eliminar ciclos redundantes (entidades relacionadas conformando un ciclo tal que, por las cardinalidades de las relaciones, a partir de un dato de cualquiera de ellas se puede obtener el mismo dato transitivamente por las relaciones) quitándoles alguna relación.
- Para cada consulta u operación frecuente, listar entidades y relaciones involucradas en el orden en que deban recorrerse para resolverla, y constatar la consistencia del recorrido: que exista una relación, aunque sea transitiva, entre las entidades que se recorran y que las cardinalidades permitan obtener el resultado que se espera.

En las Figuras 1, 2 y 3 se muestran los diagramas correspondientes a sendas etapas para el diseño de una base de datos para la gestión de un hotel.

En la Figura 1, Desempeño se modela como una entidad proceso y no como entidad asociativa o clase asociación, porque se prevé que un mismo empleado puede desempeñarse más de una vez en un mismo sector pero en diferentes períodos (por ejemplo un período como botones en el sector Recepción, luego en otro período como Valet en el sector Estacionamiento, y luego volver a desempeñarse como botones en otro período a continuación). Para las entidades asociativas o clases asociación, los identificadores de los vínculos que representan son implícitos, y si la cardinalidad de la asociación es de muchos a muchos, se compone de un identificador de cada clase entidad asociada (no podría haber pares repetidos, es decir, no admitiría que un empleado se desempeñe dos veces en el mismo sector en períodos distintos). Entonces debe modelar Desempeño como una clase entidad con estereotipo de proceso porque tiene inicio y fin.

y una entidad asociativa con ese atributo descriptivo. Se supone que uno de los sistemas de reserva sería el mismo hotel, ya sea que las reservas se puedan hacer en una página web oficial o a través de una comunicación sincrónica (por teléfono) o asincrónica (por e-mail o WhatsApp); esto es necesario para poder registrar los precios de contratación directa con el hotel.

4 Modelado lógico

Para el modelado lógico, que se obtiene por derivación de los modelos conceptuales a un esquema relacional, se emplea una notación en formato de texto que prescinde de la tipificación de atributos (es de un nivel de abstracción intermedio entre el modelado conceptual y la implementación) y que agiliza la derivación al evitar el trabajo de usar otra herramienta u otro lenguaje gráfico. En la Formulación 1 se muestra la derivación del modelo de la Figura 3. Se convierten todas las entidades y relaciones de muchos a muchos en tablas, y se transforman las jerarquías y los atributos compuestos o polivalentes.

pk: primary key
ck: candidate key
fk: foreign key (fk o ck de otra tabla) (1)

Cosas

Persona((idPersona)pk, (tipoDoc, nroDoc)ck, nombre, apellido, fechaNac)
 Domicilio(((idPersona)fk)pk, calle, altura, (piso)?, (depto)?, localidad)
 Empleado(((idPersona)fk)pk, (cuil)ck, (legajo)ck, (profesiones)?)
 Cliente((idCliente)pk)
 Institución(((idCliente)fk))pk, razónSocial, (dir_e)?, (teléfono)? *{dir_e o teléfono deben tener un valor}*
 Huésped(((idPersona)fk)pk, ((idCliente)fk)ck)
 Sector((nombre)ck, (interno)pk, localización, descripción)
 Servicio((idServicio)pk, ((interno)fk, número)ck, precio, descripción)
 SistemaReservas((idSR)pk, (nombre)ck)
 Categoría((idCategoría)pk, (nombre)ck, capacidad, descripción)
 Habitación((número)pk, (idCategoría)fk)

Procesos

Desempeño((idDesemp)pk, ((interno)fk, (legajo)fk), inicio)ck, pasado, (fin)?
 Estadía(((idCliente)fk, inicio)pk, (idEstadía)pk, fin, nroTarjCr, (pagoTarj)?, (observaciones)?, ((respCheckIn)fk)?, ((respCheckOut)fk)?) *{respCheckIn y respCheckOut referencian a idDesemp de la tabla Desempeño}*
 Hospedaje(((idEstadía)fk, (nroHabitación)fk)pk, (idHospedaje)ck, precioNoche)
 Reserva(((idCLiente)fk, fechaReserva)pk, (idReserva)ck, (nroReserva)?, (idSR)fk, ((idEstadía)fk)?)
 PagoAdelantado(((idReserva)fk)pk, fecha, monto, modoPago)

Cancelación(((idReserva)fk)pk, fecha, (monto)?)

Eventos

Consumo(((idCliente)fk, fecha, hora)pk, (idConsumo)ck, ((idHospedaje)fk)?, ((idEstadía)fk)?, (idDesemp)fk)

Asociaciones de muchos a muchos y entidades asociativas

HospHuesp(((idCliente)fk, idHospedaje)fk)pk)

Precio(((idCategoría)fk, (idSR)fk)pk)

ConsumoServicio(((idServicio)fk, (idConsumo)fk)pk, cantidad, precioUnitario)

5 Operaciones relacionales

En la Formulación 2 se muestra un esquema relacional para ejemplificar la representación de operaciones y sus traducciones literales a SQL, y en las Formulaciones 3, 4 y 5 se muestran consultas que se pueden expresar en CRT y AR o preferentemente en solo uno de los lenguajes, y sus respectivas traducciones a SQL.

Factura((nroFac)pk, fecha, fPago, (dto)?) (2)

Producto((codProd)pk, desc, existAct, existMin, pVAct)

VentaProd(((nroFac)fk, (codProd)fk)pk, cant, pvu)

Productos vendidos el 15 de abril de 2023 con existencia actual menor que la mínima: (3)

$Producto \mid_{>codProd} (\sigma_{fecha = 15/4/2023} (Factura) \mid_{\times} \mid_{/} VentaProd)$

```
SELECT DISTINCT p.*
FROM Producto p INNER JOIN
((SELECT * FROM Factura WHERE fecha='2023-04-15') NATURAL JOIN
VentaProd)
USING (codProd)
WHERE p.existAct < p.existMin;
```

$\{p \in Producto \mid p.existAct < p.existMin \wedge (\exists f \in Factura, vp \in VentaProd)(f.nroFac = vp.nroFac \wedge f.fecha = 15/4/2023 \wedge vp.codProd = p.codProd)\}$

```
SELECT *
FROM Producto p
WHERE p.existAct < p.existMin AND EXISTS
(SELECT *
FROM Factura NATURAL JOIN VentaProd vp
```

WHERE fecha='2023-04-15' AND vp.codProd=p.codProd);

Listar código y descripción de productos que se hayan vendido todos los días de abril de 2023 que hubo ventas: (4)

$$\{(p.codProd, p.desc) \mid p \in Producto \wedge \neg(\exists(f.fecha))(f \in Factura \wedge 1/4/2023 \leq f.fecha \leq 30/4/2023 \wedge \neg(\exists v \in VentaProd, f2 \in Factura)(v.codProd=p.codProd \wedge f2.fecha=f.fecha))\}$$

```
SELECT p.codProd, p.desc
FROM Producto p
WHERE NOT EXISTS
  (SELECT f.fecha
   FROM Factura f
   WHERE EXTRACT(MONTH FROM fecha)=4 AND
        EXTRACT(YEAR FROM fecha)=2023 AND NOT EXISTS
      (SELECT *
       FROM Venta v NATURAL JOIN Factura f2
       WHERE v.codProd=p.codProd AND f2.fecha=f.fecha));
```

Calcular el monto total de ventas de cada día del mes de abril de 2023 y la cantidad de productos distintos que se vendieron: (5)

$$fecha \ \alpha \ totVentas=Suma(cant * pvu * (1-dto)), \ prodDistintos=Cont(\neq codProd) (VentaProd \ | \times \ | \ \sigma \ 1/4/2023 \leq fecha \leq 30/4/2023 (Factura))$$

```
SELECT fecha, SUM(cant*pvu*(1-dto)) as totVentas, COUNT(DISTINCT
codProd)
FROM VentaProd NATURAL JOIN Factura
WHERE fecha BETWEEN '2023-04-01' AND '2023-04-30'
GROUP BY fecha;
```

6 Resultados

El régimen de cursado que se expone en la sección 2.1 se implementó con clases y evaluaciones virtuales durante el período de Aislamiento Social Preventivo y Obligatorio establecido por la pandemia, y posteriormente se pasó a una modalidad híbrida, con encuentros virtuales para desarrollo de casos de estudio, siempre con el requisito de haber visto previamente el material proporcionado con la presentación de conceptos y un caso de aplicación, y presenciales para las actividades formativas iniciales, las presentaciones de etapas de un proyecto de promoción y la primera instancia de las evaluaciones conceptuales por unidad (de opciones múltiples).

El comienzo del curso con el tema de mayor desafío para los estudiantes produce una toma de conciencia temprana del grado de compromiso que se requiere el y se

refleja en el porcentaje de deserción, que no se aleja del promedio general de los cursos de las carreras pero que tiene como arista positiva que se concentra en las clases iniciales y es casi nula durante las actividades formativas grupales; los estudiantes que persisten lo hacen con niveles de compromiso y de participación que resultan altamente satisfactorios. No hubo reprobación de proyectos de promoción y alrededor del 60% de los estudiantes promocionó la asignatura con las evaluaciones conceptuales por unidad durante el cursado. Asimismo se obtuvieron comentarios positivos por parte de los estudiantes en relación con la integración horizontal y vertical de contenidos que pretende el enfoque holístico, percibiéndose durante el avance del curso una gran fluidez en el desarrollo de competencias de aplicación: derivaciones de modelos conceptuales a esquemas relacionales y de operaciones en lenguajes relacionales a SQL.

Referencias

1. CONEAU, Normativa y procedimiento para acreditación de carreras de grado, <https://www.coneau.gov.ar/coneau/wp-content/uploads/2021/05/IF-2021-32721494-APN-SECPUME-Anexo-1.pdf>, accedido el 2023-04-11.
2. CONEAU, Normativa y procedimiento para acreditación de carreras de grado, <https://www.coneau.gov.ar/coneau/wp-content/uploads/2021/06/IF-2021-32810139-APN-SECPUME-Anexo-1.pdf>, accedido el 2023-04-11.
3. Dowding, T.J.: The application of a spiral curriculum model to technical training curricula, *Educational Technology*, 33(7), 18-28 (1993).
4. Mannino, M.: Administración de bases de datos. Diseño y desarrollo de aplicaciones. Tercera edición. McGraw-Hill Interamericana, México (2007).
5. Silberschatz, A., Korth, H.F., Sudarshan, S.: Database System Concepts. Sixth edition. McGraw-Hill, New York (2011).
6. Elmasri, R, Navathe, S.B.: Fundamentals of Database Systems. Sixth edition. Addison-Wesley, Boston (2011).
7. García Molina, H., Ullman, J.D., Widom, J.: Database Systems. The Complete Book. Second edition. Pearson Prentice Hall, New Jersey (2009).
8. Connolly, T., Begg, C.: Database Systems. A Practical Approach to Design, Implementation, and Management. Sixth edition. Pearson Education, Essex (2015).
9. The Second Principle, Teaching Essentials, Bloom's Taxonomy Revised, <https://thesecondprinciple.com/essential-teaching-skills/blooms-taxonomy-revised/>, accedida el 2023/04/23.
10. Servetto, A.: Diseño Conceptual de Bases de Datos con UML. *Brazilian Journal of Development*, Curitiba, 5(1), 1823-1835 (2019).
11. Jiménez Rey, E., Calvo, P., Servetto, A.: Experiencias de Actividades Formativas para el Desarrollo de Competencias de Orden Superior. En actas de: 6to Congreso Nacional de Ingeniería en Informática / Sistemas de Información, Universidad CAECE, Mar del Plata, ISSN 2347-0372 (2018).
12. Calvo, P., Jiménez Rey, E., Servetto, A.: Pautas para la Evaluación Formativa del Aprendizaje Activo en la Formación Superior Basada en Competencias. En actas de: 7mo Congreso Nacional de Ingeniería Informática / Sistemas de Información, pp. 381-388, Universidad Nacional de La Matanza, San Justo, ISSN: 2347-0372, ISBN 978-987-4417-73-2 (2019).