



# TESINA DE LICENCIATURA

**Título:** CasER 3.0: Modelado de base de datos en la nube

**Autores:** Gallardo, Patricio Diego; Piñero, Carla Tatiana

**Director:** Pablo Thomas

**Codirector:** Rodolfo Bertone

**Carrera:** Licenciatura en Sistemas

## Resumen

Este trabajo plantea la realización de una nueva versión de CasER mejorando versiones anteriores de escritorio.

Su nombre es *CasER 3.0: Modelado de base de datos en la nube*.

En la actualidad, la manera de vivir en la sociedad ha evolucionado a pasos agigantados; los dispositivos móviles, la manera de comunicación, el acceso a internet, entre otros, condujo a esta nueva versión como una herramienta WEB, e integrar en ella algunos componentes de Google.

Se pensó en el manejo de usuarios (creando una cuenta asociada a la de Gmail) para poder obtener estadísticas e información de personas que la utilizaban, a que facultades pertenecen y a que se dedican (mediante Google Analytics), y a su vez que cada usuario pueda tener su área de trabajo, donde guardar y poder compartir proyectos, mediante Google Drive.

Todas estas nuevas características, sumadas a un nuevo look and feel, que mejora la interfaz de usuario notablemente, hacen de CasER 3.0 una herramienta que asiste al usuario al diseño de la base de datos, aportando nuevas funcionalidades, optimización de tiempo utilizado en el proceso de diseño, y facilidad para aprender a usarla.

## Palabras Claves

*Diseño Conceptual, Diseño Lógico, Diseño Físico, Modelo de Entidades y Relaciones. Herramienta Case. Google Drive. Google Analytics. Nube. Cliente-servidor. UX*

## Conclusiones

*La abstracción de conceptos de la realidad, y la representación de éstos en un modelado gráfico, para luego obtener lo que finalmente serán las tablas que representarán una BD relacional, es un desafío.*

*La versión 3.0 de la herramienta CasER permite completar el proceso de diseño en forma asistida, incorporando el concepto de dominios y nuevas funcionalidades como: compartir trabajos realizados, obtener estadísticas de uso y exportar trabajos, entre otras.*

## Trabajos Realizados

*Se desarrolló una herramienta para asistir el proceso de enseñanza/aprendizaje del diseño de una BD relacional.*

*El resultado fue CasER 3.0, una aplicación Web fácil de usar, que incluye las funcionalidades necesarias para cumplir con las 3 fases del diseño de una Base de Datos, así como también la integración con complementos de Google (Incorporación de usuarios, Intercambio de proyectos y obtención de estadísticas de uso).*

## Trabajos Futuros

*Se pretende utilizar la herramienta para todo el proceso de diseño de BD, y obtener retroalimentación de los alumnos respecto a la usabilidad y las mejoras posibles en ese sentido.*

*Se pretende incorporar la generación automática de los scripts de creación de la BD en lenguaje SQL.*

*Finalmente, aprovechando que es una herramienta web, se podría incorporar un chat interno a la aplicación para discutir con otros usuarios, mientras se realiza el proceso de diseño de una BD.*



## AGRADECIMIENTOS

*A mis papás, por darme la oportunidad de estudiar, por el apoyo incondicional, y por estar siempre cerca, aunque lejos de casa, a mis hermanas y amigos, y sobre todo a mi compañero de vida y de tesis por alentarme siempre a seguir y terminar esta última etapa.*

*Tatiana*

*Definitivamente a mis padres, quienes me incentivaron a tener un título. A mis hermanos, que marcaron el camino del estudio. A mi tía, que me regaló mi primera computadora, y estuvo en cada paso. A los amigos, que apoyaron en este arduo camino. Y definitivamente este paso final jamás lo hubiésemos logrado sin el empuje de mi compañera de tesis, viajes y caminos de vida.*

*Patricio*

*Agradecemos Pablo y Rodolfo, por ser nuestros únicos guías, por la ayuda que nos brindaron y enseñanza durante el todo desarrollo de esta Tesina.*

*Tatiana y Patricio*

## ÍNDICE GENERAL

Tabla de figuras.....	4
<b>CAPÍTULO 1: Introducción.....</b>	<b>7</b>
1.1    Introducción:.....	7
1.2    Motivación.....	7
1.3    Objetivo .....	9
1.4    Aportes .....	9
1.5    Estructura de la tesina .....	11
<b>CAPÍTULO 2: Conceptos Generales de una base de datos.....</b>	<b>13</b>
2.1    ¿Qué es una base de datos? .....	13
2.2    Modelo de datos .....	14
2.3    Etapas del diseño de una base de datos.....	15
<b>CAPÍTULO 3: Conceptos generales del Modelo Conceptual .....</b>	<b>17</b>
3.1    Modelo Conceptual.....	17
3.2    Entidad, Atributos y Relación .....	18
3.3    Jerarquías y Cobertura .....	22
<b>CAPÍTULO 4: Conceptos generales del Modelo Lógico .....</b>	<b>25</b>
4.1    Modelo lógico (Entidad-relación) .....	25
4.2    Características del Diseño Lógico.....	25
4.3    Pasaje de modelo conceptual a lógico.....	26
4.4    Ejemplo de eliminación de jerarquía:.....	34
<b>CAPÍTULO 5: Conceptos generales del Modelo Físico .....</b>	<b>37</b>
5.1    Modelo Físico (Relacional) .....	37
5.2    Pasaje de Modelo lógico a Físico.....	37
<b>CAPÍTULO 6: Descripción de Caser 3.0.....</b>	<b>46</b>
6.1    Introducción:.....	46
6.2    Rediseño de la arquitectura: .....	47
6.3    Usuarios / Herramienta colaborativa .....	51

6.4 Rediseño Interface de Usuario (UI) .....	52
6.5 Uso de la herramienta .....	56
<b>CAPITULO 7: Conclusiones y trabajos futuros .....</b>	<b>79</b>
7.1 Conclusiones .....	79
7.2 Trabajos futuros .....	82
BIBLIOGRAFIA.....	83

## TABLA DE FIGURAS

Fig. 1 Clasificación de los modelos de datos .....	14
Fig. 2 Ejemplo de una entidad "alumno" .....	18
Fig. 3 Ejemplo de Atributos.....	19
Fig. 4 Ejemplo de una relación recursiva .....	21
Fig. 5 Ejemplo de relación binaria.....	21
Fig. 6 Ejemplo de relación ternaria .....	22
Fig. 7 Ejemplo de una jerarquía, con cobertura Parcial exclusiva .....	24
Fig. 8 Atributo Derivado .....	27
Fig. 9 Ejemplo de ciclos de Relaciones. Redundancia. ....	28
Fig. 10 Ejemplo de Atributo Polivalente.....	29
Fig. 11 Eliminación de Atributo Polivalente.....	30
Fig. 12 Ejemplo de Atributo Compuesto .....	30
Fig. 13 Eliminación de Atributo Compuesto: Concatenación .....	31
Fig. 14 Eliminación de Atributo Compuesto: Atributos simples .....	31
Fig. 15 Eliminación de Atributo Compuesto: Genera la nueva entidad Dirección .....	32
Fig. 16 Ejemplo subconjunto.....	34
Fig. 17 Jerarquía Socios de un Club .....	35
Fig. 18 Eliminación de Jerarquías: Se quitan las entidades hijas.....	35
Fig. 19 Ejemplo de identificador externo.....	38

Fig. 20 Ejemplo conversión de entidades, cardinalidad 1-1 .....	40
Fig. 21 Ejemplo de conversión de relaciones con cardinalidad N-N.....	40
Fig. 22 Ejemplo de conversión de relaciones 1-N, con participación obligatoria .....	41
Fig. 23 Ejemplo de conversión de relaciones con participación parcial del lado de muchos .....	42
Fig. 24 Ejemplo de conversión de relaciones con participación parcial del lado de uno.....	43
Fig. 25 Ejemplo de conversión de relaciones uno a uno, con participación parcial de un lado. ....	44
Fig. 26 Arquitectura Caser 3.0 .....	49
Fig. 27 Comparativa Controles Caser 2.0 / Caser 3.0 .....	53
Fig. 28 Herramienta CasER 3.0 .....	54
Fig. 29 Ejemplo de mejoras de look&feel.....	55
Fig. 30 Comparativa de Relación recursiva en Caser 2.0/3.0.....	56
Fig. 31 Extracto de la pantalla principal de Caser 3.0.....	57
Fig. 32 Ejemplo registraci3n de Usuario.....	58
Fig. 33 Pantalla de Bienvenida de Usuario .....	59
Fig. 34 PopUp para abrir proyectos guardados en Google Drive. ....	60
Fig. 35 Edici3n de proyectos: Secci3n enunciado.....	60
Fig. 36 Secci3n Enunciado: Referencias de colores.....	61
Fig. 37 Secci3n "Modelo Conceptual" .....	62
Fig. 38 Secci3n "Modelo L3gico" .....	63
Fig. 39 Pasaje a F3sico habilitado .....	63
Fig. 40 Secci3n "Modelo F3sico" .....	64
Fig. 41 Opciones para agregar los elementos en el Modelo Conceptual .....	65
Fig. 42 Ejemplo creaci3n de elemento Atributo. ....	65
Fig. 43 Ejemplo edici3n de un componente Entidad. ....	66
Fig. 44 Ejemplo Edici3n de una Entidad: solapa Atributos.....	67
Fig. 45 Edici3n Entidad: Identificadores Compuestos .....	68
Fig. 46 Identificados compuesto - l3nea punteada roja-.....	68
Fig. 47 Validaci3n Formulario: Ejemplo atributo .....	69

Fig. 48 Pasaje semiautomático: Ejemplo intervención del usuario .....	70
Fig. 49 Pasaje Automático: No existe intervención del usuario, solo se informa las acciones.....	71
Fig. 50 Ejemplo Práctico: Planteo de la problemática a resolver.....	72
Fig. 51 Ejemplo Práctico: Modelo Conceptual.....	73
Fig. 52 Modelo Lógico.....	74
Fig. 53 Modelo Físico.....	74
Fig. 54 Modelo físico – ejemplo atributo deporte de tipo texto. ....	75
Fig. 55 Compartir un proyecto con otras personas.....	75
Fig. 56 Ventana emergente para compartir un proyecto .....	76
Fig. 57 Acceder a un proyecto.....	76
Fig. 58 Exportar un proyecto .....	78

# CAPÍTULO 1: INTRODUCCIÓN

## 1.1 INTRODUCCIÓN:

A lo largo de este documento se darán a conocer diversos conceptos que conciernen al diseño de una base de datos. Los principales conceptos serán, el Modelo Conceptual, Modelo Lógico, y Modelo Físico, de cada uno de ellos, se darán a conocer los elementos que los componen, como se construyen, y se concluirá en el modelo final, donde se verán las tablas que formarán parte de una base de datos.

En este documento, también se darán a conocer las necesidades que conllevan a la creación de esta herramienta, denominada *CasER 3.0 Modelado de base de datos en la nube*, los objetivos de dicha herramienta, los aportes que hará la misma, ejemplos prácticos realizados, funcionalidades extras como compartir, y exportar, entre otras.

Por último se darán a conocer los problemas abordados para llegar a tal desarrollo, los trabajos futuros a realizar sobre el mismo y las conclusiones.

## 1.2 MOTIVACIÓN

En la asignatura “*Introducción a Bases de Datos*” (IBD), de la Facultad de Informática (FI) de la “*Universidad Nacional de La Plata*” (UNLP), el principal tema abordado, es el proceso en el cual se diseña una base de datos (BD), el cual los alumnos lo deben poner en práctica hasta que se adquiera la habilidad de realizar el diseño de una BD de manera rápida y efectiva. Ésta es una actividad compleja que consta de 3 etapas bien definidas: diseño Conceptual, diseño Lógico y diseño Físico.

Hasta el año 2009, no se disponían de herramientas con las características requeridas por la asignatura IBD, preparadas para la enseñanza del diseño de una BD que contenga todos los temas abordados en ella, por lo tanto los alumnos debían ejercitar la práctica de esta asignatura de una forma tradicional, con papel y lápiz, teniendo en cuenta que ésto, en muchas situaciones, podía llegar a ser engorroso y poco práctico.

De ésta problemática, nació la necesidad de crear una herramienta que facilitara el proceso de diseño de una BD. Ésta fue llevada a cabo en el año 2009, con la puesta en marcha de la primera herramienta realizada por alumnos de la Facultad de Informática de la UNLP, presentada como Tesina de grado, denominada “*CasER: Herramienta para modelado conceptual de bases de datos*”, la cual permitía avanzar en los primeros pasos del desarrollo, plantear el problema a resolver (como una especificación de requerimientos sobre datos y mapear desde allí los principales conceptos a elementos de un esquema conceptual), y diseñar el modelo conceptual; un gran paso para comenzar la ejercitación práctica de una manera más fácil y “limpia”.

En el año 2012, llegó la segunda versión de ésta herramienta. “*CasER 2.0: herramienta para la enseñanza de Modelado de Bases de Datos*” (también presentada como tesina de grado). Ésta no sólo permitía diseñar el Modelo Conceptual (MC), sino que también permitía construir el Modelo Lógico (ML) a partir del MC, y el Modelo Físico (MF), a partir del ML, con lo cual se concluyeron los 3 pasos necesarios para completar el diseño de una Base de Datos.

Ambas versiones fueron desarrolladas como una aplicación de escritorio. CasER no solo es de uso local en la Facultad de Informática, sino también Unidades académicas como UNNOBA, UNTDF, UNPSJB la utilizan en el desarrollo del material práctico de las materias de Bases de Datos. Incluso se ha presentado el producto en cursos de Posgrado tanto en la UNLP como en universidades del país y del exterior.

En la actualidad, la manera de vivir en la sociedad ha evolucionado a pasos agigantados, ya que al desarrollarse nuevas tecnologías se han generado aportes importantes a todos los medios de comunicación y a la manera de comunicarnos, el acceso a internet creció notablemente así como también su uso, con lo cual ésto llevó a pensarse en una nueva versión de la herramienta, pero esta vez, WEB.

Se desarrolló así *CasER 3.0: Modelado de base de datos en la nube*.

Las 3 versiones de CasER tuvieron como enfoque principal el uso académico, pero ésta nueva versión no fue una continuación de las anteriores, sino que se creó desde el inicio, dado que sus características de implementación son completamente distintas a CasER 1.0 y CasER 2.0. Las primeras 2 versiones son aplicaciones de escritorio, mientras que CasER 3.0 es una aplicación WEB.

### 1.3 OBJETIVO

Se desarrolló una herramienta dedicada íntegramente al diseño de una base de datos relacional, incluyendo los 3 modelos necesarios para su desarrollo, el Modelo Conceptual, el Modelo Lógico, y el Modelo Físico. Esta herramienta tiene como principal objetivo ayudar al alumno al aprendizaje de uno de los principales temas abordados -Diseño de Base de datos- de la asignatura IBD, de la Facultad de Informática en la UNLP.

Esta herramienta no solo acompañara en el aprendizaje del alumno sino que también provee ciertas funcionalidades orientadas a mejorar la experiencia del usuario:

- ✓ Claridad y legibilidad en los modelos: Esta herramienta permite al usuario mantener el modelo ordenado e ir acomodando los elementos que lo componen a medida va desarrollándose y por ende creciendo.
- ✓ Agilidad en entregas: CasER 3.0 permite realizar entregas inmediatas, sin la necesidad de asistir a punto de entrega -aula, oficina, entre otras- ya que esta herramienta al estar integrada con Google Drive permite compartir archivos de manera rápida e instantánea.
- ✓ Permite obtener estadísticas de usabilidad: También está integrada con Google Analytics, y esto permite sacar estadísticas fácilmente de la cantidad de personas que lo usan, a que se dedican éstas personas, sean profesores, alumnos, ayudantes u otros y a que facultades pertenecen.

### 1.4 APORTES

Los aportes de CasER 3.0 han sido íntegramente académicos. Como se mencionó con anterioridad los alumnos ahora podrán ejercer la práctica de la asignatura IBD con esta herramienta, teniendo ventajas que en papel y lápiz no tenían.

La versión 3.0 de la herramienta CasER permite, no solo completar el proceso de diseño, en forma asistida, incorporando el concepto de dominios desde el modelo conceptual, sino que también cuenta con 3 funcionalidades extras al modelado de datos:

1. Los trabajos que se realicen se pueden compartir desde cualquier punto de desarrollo en el cual se encuentre el autor, por estar integrado a Google Drive.

Mirando con detenimiento lo que respecta a esa nueva funcionalidad, trae consigo varias ventajas:

- a. *Entrega remota de trabajos realizados*: el alumno ya no tendrá la necesidad de asistir a una clase para hacer entrega de su trabajo práctico, sino que lo puede hacer de manera remota, ya sea, compartiendo el archivo mediante Google Drive, exportando el trabajo, a una imagen .png, y enviándolo por correo.
- b. *Correcciones remotas de trabajos realizados*: las personas asignadas a la corrección de estas entregas, tendrán la posibilidad de editar el archivo enviado y marcar las correcciones pertinentes (siempre y cuando el archivo posea permisos de edición), dándole al alumno, una corrección rápida y remota.
- c. *Accesibilidad remota*: al ser una Herramienta web, se puede acceder desde cualquier dispositivo que cuente con internet, sea celular, Tablet, o computadora; los trabajos realizados quedarán guardados y sincronizados en Google Drive. Accediendo a la cuenta de Gmail e ingresar a Google Drive, o ingresando a la aplicación web CasER 3.0, se podrá acceder a todo el contenido guardado en el estado exacto en el que se guardó.
- d. *Cooperación entre colegas*: Compartir los trabajos realizados de manera remota, es muy provechoso a la hora de optimizar los tiempos ya que permite a los participantes trabajar de manera remota e ir compartiendo ideas y conocimiento.

2. Obtener estadísticas de personas o instituciones que acceden y utilizan la herramienta, ya que cuenta con la integración de Google Analytics.

Esto permite tener una estimación, si se quiere, de los diferentes perfiles de usuarios que utilizan la herramienta, sean estos, profesores, alumnos, ayudantes, y otros, y al tener algunos datos, se podría hacer también una encuesta con relevamiento de datos, y mejoras, que propongan los usuarios en cuestión.

3. Exportar los Modelos creados: Se puede exportar los diseños creados en formato .png. Esto puede ser muy útil si se desea compartir con alguien de manera física, y para el caso de los alumnos que utilicen CasER 3.0, llevarlo a alguna corrección en la Facultad.

## 1.5 ESTRUCTURA DE LA TESINA

- Capítulo 1: Introducción En el primer capítulo se hace una breve introducción, en la que se desglosa el objetivo, la motivación, y el aporte de la herramienta CasER 3.0, dándose a conocer las ventajas de ésta nueva versión y se nombran algunos conceptos claves que en el resto de los capítulos se describirán con más detalle.
- Capítulo 2: Conceptos generales de una base de datos: Este capítulo es muy importante para comprender todos los conceptos que competen al diseño y modelado de una base de datos. Se define que es una base de datos, que es un modelo de datos, y cuáles son los pasos del diseño del mismo.
- Capítulo 3: Conceptos generales de un Modelo Conceptual El primer modelo que se aborda para diseñar una base de datos, es el Modelo Conceptual. Es importante conocer estos conceptos para poder dar los primeros pasos. Este capítulo explica que es un modelo conceptual, que son las entidades, las relaciones, los atributos, las jerarquías, y otros temas pertinentes a este modelo.
- Capítulo 4: Conceptos generales de un Modelo Lógico: Este capítulo, aborda todos los conceptos relacionados con el Modelo Lógico, que es el modelo consecuente del Modelo Conceptual. Se muestran las diferentes tomas de decisiones que se realizan en el proceso de pasaje del Modelo Conceptual al Lógico.
- Capítulo 5: Conceptos generales del Modelo Físico: A partir del Modelo Lógico se aborda la tercera fase, que es el Modelo Físico. Así como en el Lógico se tienen en cuenta principalmente los aspectos relativos al significado de los datos, en el físico se consideraran fundamentalmente los aspectos de los datos relacionados con el rendimiento y la disponibilidad de la aplicación. En este capítulo se presentan los pasos necesarios que permiten generar el Modelo Físico de datos a partir del Modelo Lógico explicado en el capítulo anterior.

- Capítulo 6: Descripción de CasER 3.0: En éste capítulo se abordaran temas que hacen referencia a la herramienta, como se llevó a cabo su implementación, su arquitectura, sus funcionalidades, ejemplos de su uso, las ventajas de ser una Aplicación WEB, entre otros temas.
- Capítulo 7: Conclusiones y Trabajos Futuros Este capítulo es la culminación de la tesina, donde se describe la conclusión que se ha tenido sobre el desarrollo del trabajo. Y para finalizar, los trabajos futuros que se pueden llevar a cabo para continuar con avances de CasER 3.0.

## CAPÍTULO 2: CONCEPTOS GENERALES DE UNA BASE DE DATOS

### 2.1 ¿QUÉ ES UNA BASE DE DATOS?

*“Todas las empresas requieren almacenar información. Desde siempre lo han hecho. La información puede ser de todo tipo. Cada elemento informativo (nombre, dirección, sueldo, etc.) es lo que se conoce como dato (en inglés data).” [1]*

Antes se utilizaban carpetas, ficheros cajones, entre otras cosas, para guardar la información, siendo un problema su procesamiento porque si el almacenamiento era demasiado grande demandaba mucho tiempo la búsqueda de un dato o información específica.

Se supone el caso en el que se requiere consultar el aumento promedio de un empleado de una empresa que ha trabajado por 20 años. Ésto requeriría consultar todos los recibos de sueldo de este empleado año tras año, consultar cada aumento que se le ha asignado, y realizar el cálculo necesario; todo manualmente, lo cual requeriría de un tiempo de procesamiento elevado.

Es por ésto que ha surgido la necesidad de crear una herramienta, una estructura, que brinde la posibilidad de disponer con los datos sistemáticamente organizados, cumpliendo con determinados requisitos para obtener el máximo beneficio en su aplicación, con menor inversión de tiempo.

Tras la aparición de la informática estos datos se almacenan en archivos digitales dentro de las unidades de almacenamiento del ordenador (a veces en archivos binarios, o en hojas de cálculo). Además las empresas requieren utilizar aplicaciones informáticas para realizar tareas propias de la empresa con el fin de automatizar y agilizar las mismas. Estas aplicaciones requieren manejar los datos pertinentes. En los inicios de la era informática, cada programa almacenaba y utilizaba sus propios datos de forma un tanto caótica. La ventaja de este sistema (la única ventaja), es que los procesos eran independientes por lo que la modificación de uno no afectaba al resto. Pero tiene grandes inconvenientes:

- Costo de almacenamiento elevado.
- Datos redundantes (se repiten continuamente).
- Probabilidad alta de inconsistencia en los datos.
- Difícil modificación en los datos y facilidad de problemas de inconsistencia al realizar esas modificaciones (ya que es difícil que esa modificación afecte a todos los datos).

Lógicamente la solución a este problema es hacer que todas las aplicaciones utilicen los mismos datos.

*“Esto provoca que los datos deban estar mucho más protegidos y controlados. Además los datos forman una estructura física y funcional que es lo que se conoce como base de datos. De esta forma una base de datos es una serie de datos relacionados que forman una estructura lógica, es decir una estructura reconocible desde un programa informático. Esa estructura no sólo contiene los datos en sí, sino la forma en la que se relacionan”.* [1]

## 2.2 MODELO DE DATOS

Los modelos son utilizados en muchas disciplinas, teniendo como finalidad representar una parte del mundo real de forma que la información representada sea más fácil de manipular.

*“El diseño de una BD es un proceso complejo que abarca decisiones a distintos niveles. La complejidad de un problema se aborda mejor si se descompone dicho problema en subproblemas y se resuelve cada uno de éstos independientemente, utilizándose métodos y técnicas específicas. Una alternativa de diseño de BD consiste en descomponer esta actividad en tres etapas: diseño conceptual, diseño lógico y diseño físico”* [6].

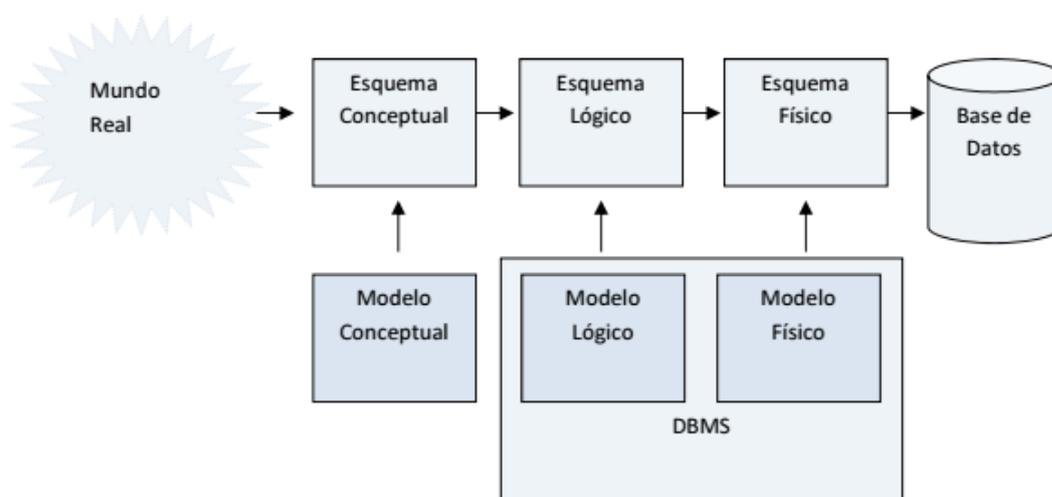


Fig. 1 Clasificación de los modelos de datos

En la Figura 1 aparecen los distintos esquemas que llevan desde el mundo real a la base de datos física. Como se ve aparecen varios esquemas intermedios. Los que están más a la

izquierda se alejan más de las características físicas. Los elementos de ese esquema son:

1. Mundo real. La información que contiene este esquema, es la que los seres humanos perciben como tal. Es el punto de partida para el desarrollo.
2. Esquema Conceptual: Descripción de alto nivel del contenido de información de la base de datos, independiente del tipo Gestor de Base de Datos (SGBD) que se vaya a utilizar. El Modelo conceptual es el lenguaje que se utiliza para describir esquemas conceptuales
3. Esquema Lógico: Descripción de la estructura de la base de datos según el tipo de SGBD que se vaya a utilizar. *Modelo Lógico*: Lenguaje que se utiliza para describir esquemas lógico; hay varios tipos de modelos lógicos. En esta tesina se utilizará el relacional.
4. Esquema Físico: Consiste en la implementación del modelo físico de datos, dando lugar a estructuras de datos de almacenamiento en uno o varios soportes físicos.

## 2.3 ETAPAS DEL DISEÑO DE UNA BASE DE DATOS

El diseño de una base de datos no es un proceso sencillo. Habitualmente, la complejidad de la información y la cantidad de requisitos de los sistemas de información hacen que sea complicado. Por este motivo, cuando se diseñan bases de datos es interesante aplicar la vieja estrategia de dividir para vencer.

Por lo tanto, conviene descomponer el proceso del diseño en varias etapas; en cada una se obtiene un resultado intermedio que sirve de punto de partida de la etapa siguiente, y en la última etapa se obtiene el resultado deseado. De este modo no hace falta resolver toda la problemática que plantea el diseño en un solo paso, sino que en cada etapa se afronta un solo tipo de subproblema. Así se divide el problema y, al mismo tiempo, se simplifica el proceso.

Se enumeraran las etapas del diseño de bases de datos en tres:

### 2.3.1 ETAPA DEL DISEÑO CONCEPTUAL:

En esta etapa se obtiene una estructura de la información de la futura BD independiente de la tecnología que hay que emplear. No se tiene en cuenta todavía qué tipo de base de datos se utilizará –relacional, orientada a objetos, jerárquica, u otra; en consecuencia, tampoco se

tiene en cuenta con qué SGBD ni con qué lenguaje concreto se implementará la base de datos. Así pues, la etapa del diseño conceptual permite concentrarse únicamente en la problemática de la estructuración de la información, sin tener que preocuparse al mismo tiempo de resolver cuestiones tecnológicas. El resultado de la etapa del diseño conceptual se expresa mediante algún modelo de datos de alto nivel. Uno de los más empleados es el modelo entidad-interrelación (entity-relationship), que se abrevia con la sigla ER.

### 2.3.2 ETAPA DEL DISEÑO LÓGICO:

En esta etapa se parte del resultado del diseño conceptual, que se transforma de forma que se adapte a la tecnología que se debe emplear. Más concretamente, es preciso que se ajuste al tipo de SGBD con el que se desea implementar la base de datos.

Esta etapa parte del hecho de que ya se ha resuelto la problemática de la estructuración de la información en un ámbito conceptual, y permite concentrarse en las cuestiones propias relacionadas con el tipo de base de datos.

Más adelante se explicará cómo se realiza el diseño lógico de una base de datos relacional, tomando como punto de partida un diseño conceptual expresado con el modelo ER; es decir, cómo se puede transformar un modelo ER en un modelo lógico relacional.

### 2.3.3 ETAPA DEL DISEÑO FÍSICO:

En esta etapa se transforma la estructura obtenida en la etapa del diseño lógico, con el objetivo de conseguir una mayor eficiencia; además, se completa con aspectos de implementación física que dependerán del SGBD específico.

## CAPÍTULO 3: CONCEPTOS GENERALES DEL MODELO CONCEPTUAL

### 3.1 MODELO CONCEPTUAL

El *Modelo Entidad Relación (Entity-Relationship - ER)*, es uno de los enfoques de modelización de datos que más se utiliza en la actualidad por su simplicidad y legibilidad. Éstos se utilizan para representar la información de un problema en un alto nivel de abstracción. Además es aplicable al diseño de cualquier tipo de bases de datos.

*“Es una herramienta útil tanto para ayudar al diseñador a reflejar en un modelo conceptual los requisitos del mundo real de interés como para comunicarse con el usuario final sobre el modelo conceptual obtenido y, de este modo, poder verificar si satisface sus requisitos.”*[2].

Este modelo fue llamado así debido a que los elementos que incluye son las entidades y las interrelaciones entre ellas. De aquí en adelante se lo mencionará como ‘entidad-relación’ (ER).

Cuando surge la necesidad de comunicarse con un usuario o cliente, es recomendable emplear una variante del modelo ER que incluya sólo sus elementos más simples –entidades, atributos e interrelaciones– y, tal vez, algunas construcciones adicionales, el cual se conoce como *Modelo conceptual ER*. Estos términos serán desarrollados en los próximos puntos del capítulo.

El Modelo Conceptual debe cumplir con una serie de características todas con igual importancia:

- **Expresividad:** El modelo debe ser expresivo, es decir debe capturar y representar de la mejor manera posible la semántica de los datos del problema a resolver.
- **Formalidad:** Dicho modelo, requiere que cada elemento presente en él, sea preciso, y esté bien definido, es decir que no sea ambiguo, que su interpretación sea única.
- **Minimalidad:** Cada elemento del Modelo Conceptual, se puede definir de una única manera, y no puede expresarse mediante otros conceptos.
- **Simplicidad:** El Modelo como ya se dijo, debe ser simple y fácil de entender tanto por el desarrollador, como para el cliente/usuario.

### 3.2 ENTIDAD, ATRIBUTOS Y RELACIÓN

A continuación se darán ejemplos y se definirán los conceptos de Entidad, Atributo y Relación. También será explicado el concepto de cardinalidad, tanto en los atributos como en las relaciones.

#### 3.2.1 ENTIDADES:

*“Una entidad es algo que se puede identificar en el medio ambiente de trabajo de los usuarios; es decir, aquello a lo cual los usuarios quieren dar seguimiento.”[7]* Una entidad dentro de un modelo ER, se representa como un rectángulo y en su interior esta se identifica por un nombre. Puede definirse entidad como un objeto real que se puede distinguir entre otros objetos identificándose de forma única.

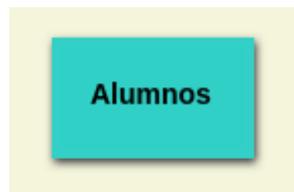


Fig. 2 Ejemplo de una entidad "alumno"

Las entidades se representan mediante rectángulos simples, con el nombre de la entidad en su interior, como lo ilustra la Figura 2. Para reconocer las entidades del problema que permitan luego modelar los conjuntos de entidades, en general se analizan los sustantivos que conforman la especificación del problema.

#### 3.2.2 ATRIBUTOS:

*“Un atributo representa una propiedad básica de una entidad o relación.”[8]*

La diferencia entre las entidades o relaciones se da por los atributos de cada una de ellas. En general, de la misma manera que las entidades, los atributos están indicados por sustantivos que definen el problema.

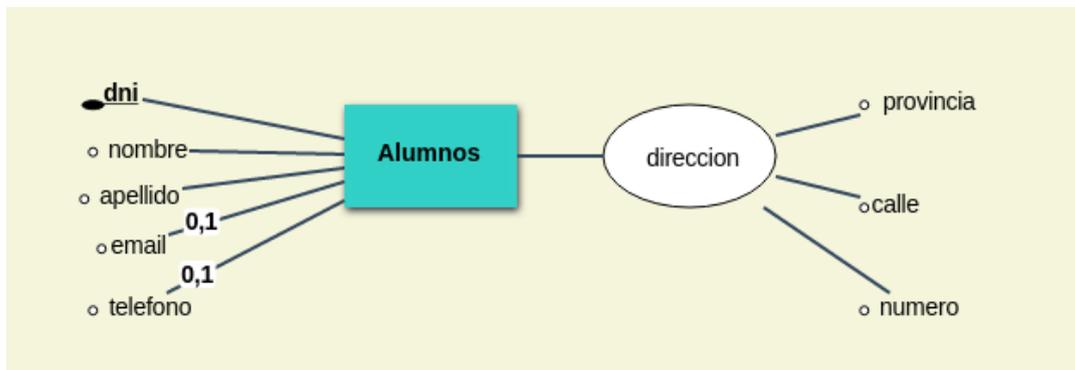


Fig. 3 Ejemplo de Atributos

### Cardinalidad de un atributo

“Los atributos también tienen asociado el concepto de cardinalidad. Esto es, cuando se define un atributo se debe indicar si es o no obligatorio y si puede tomar más de una valor (polivalente).” [4]

Expresiones posibles de cardinalidad:

- ✓ 1..1 : Monovalente
- ✓ 0..1 : Monovalente no obligatorio
- ✓ 1..N : Polivalente
- ✓ 0..N : Polivalente no obligatorio

En el ejemplo de la Figura 3, el atributo “dni” es Monovalente obligatorio, mientras que “email” y “teléfono” son Monovalentes no obligatorios.

### Dominio de un atributo

“El dominio de un atributo define el conjunto de valores posibles que el atributo puede tomar.” [4]

En la Figura 3, el atributo “nombre” se podría definir con un dominio “texto”, el cual incluye todos los caracteres posibles para representar el nombre de un alumno.

### Atributos Compuestos

*“Los atributos compuestos representan a un atributo generado a partir de la combinación de varios atributos simples.” [4]*

Un ejemplo sería “dirección”, ilustrada en la Figura 3, la cual está compuesta por “provincia”, “calle” y “número”.

### Identificadores

*“Un identificador es un atributo o un conjunto de atributos que permite reconocer o distinguir a una entidad de manera unívoca dentro del conjunto de entidades.” [4]*

Los identificadores pueden ser:

- ✓ *Simples o compuestos*: según la cantidad de atributos que lo componen.
- ✓ *Internos o externos*: si alguno de los atributos que lo componen es ajeno a la entidad que identifica es “externo”, caso contrario es “interno”.

En la Figura 3 se puede observar al atributo “dni” como un Identificador Simple e Interno, representado con un punto relleno.

### 3.2.3 RELACIONES:

*“Una relación es una asociación entre varias entidades.” [9]. “Las relaciones representan agregaciones entre dos (binaria) o más entidades. Describen las dependencias o asociaciones entre dichas entidades.”[4] “Relación es solo un término matemático para una tabla; para ser precisos, una tabla de cierto tipo específico.” [8]*

Las relaciones se representan mediante un rombo y en su interior se identifican por un nombre.

### Tipos de relaciones

- ✓ *Relación recursiva:* Es aquella en donde una entidad se relaciona consigo misma.

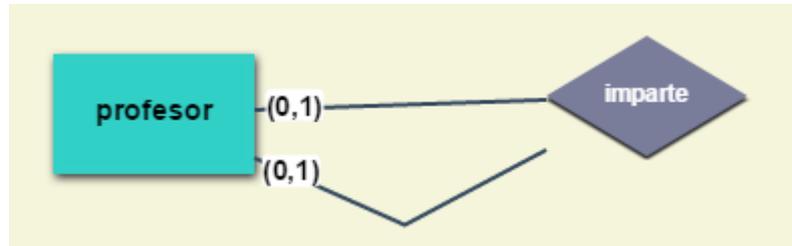


Fig. 4 Ejemplo de una relación recursiva

- ✓ *Relación binaria:* Como se expresó en la segunda definición, es aquella que asocia a dos entidades siendo esta la más común en las bases de datos.

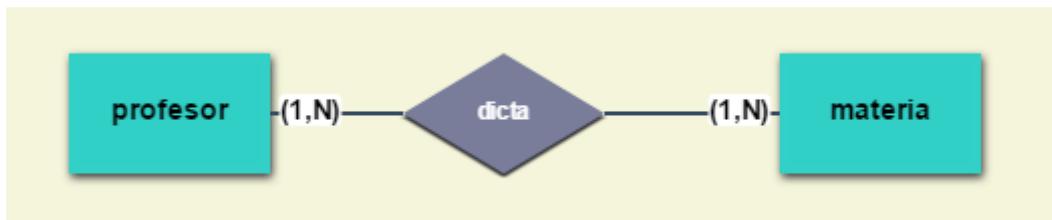


Fig. 5 Ejemplo de relación binaria

- ✓ *Relación Ternaria:* Relación entre 3 entidades
- ✓ *Relación n-aria:* relación de "n" entidades.

### Cardinalidad de las relaciones:

Los conjuntos de relaciones suelen tener ciertas restricciones, como por ejemplo el cardinal de asignación. Esto especifica de qué manera se relacionan 2 o más entidades. Especifica que cantidad de entidades que se pueden relacionar con otra. Las cardinalidades más comunes son: 1:1, 1:N, N:N, N:1, 0:1, etc., pero también si se quisiera se puede usar de 1:5, suponiendo que una entidad se puede relacionar como mínimo con 1 sola entidad, o como máximo con 5.

El ejemplo de la Figura 5, la cardinalidad conjunta de la relación es N:N ya que la cardinalidad de la relación queda definida por las cardinalidades máximas de cada entidad.

En este caso un profesor, como mínimo, puede dictar 1 materia, y como máximo N (siendo N un número desconocido, pudiendo ser 2, 6, 3, etc.); y una materia puede ser dictada, como mínimo, por un profesor, y como máximo por N.

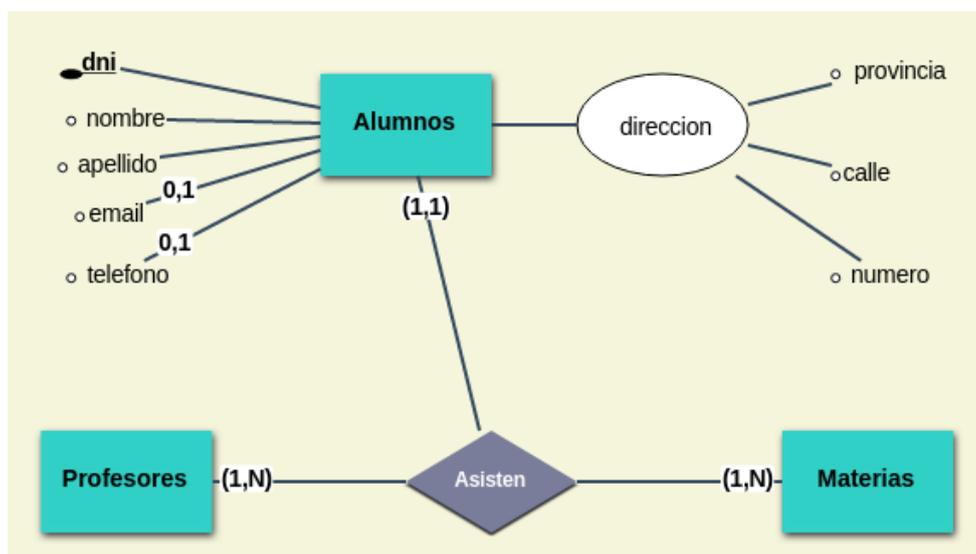


Fig. 6 Ejemplo de relación ternaria

### 3.3 JERARQUÍAS Y COBERTURA

A fin de poder dotar al modelado conceptual con las características descritas con anterioridad (entidad, relación, atributos, entre otras cosas), se han incorporado otros elementos con el fin de aumentar la expresividad del modelo constituido. A continuación se explicaran que son las jerarquías y la función que cumplen, así como también los subconjuntos, y el concepto de cobertura que está ligado a estos elementos.

#### 3.3.1 JERARQUÍAS DE GENERALIZACIÓN:

Para explicar el concepto de jerarquía, primero se debe saber que es la abstracción de generalización, lo cual, *“define una relación de subconjunto entre elementos de dos o mas clases.”*[4].

En una generalización, las especialidades (hijos) heredan las características del padre.

*“La generalización permite extraer propiedades comunes de varias entidades o relaciones, y generar con ellas una superentidad que las aglutine. Así, las características compartidas son expresadas una única vez en el modelo, y los rasgos específicos de cada entidad quedan definidos en su subentidad”*[4].

Dicho esto, se define como ejemplo la entidad “Persona”, de la cual heredan 2 subentidades, “Alumno” y “Profesor”. Ambas comparten atributos comunes, como DNI, domicilio, nombre y apellido y cada una de ellas podrán tener a su vez, atributos particulares; el alumno tendrá un número de legajo, y el profesor una antigüedad y un sueldo.

#### Subconjuntos:

*“Los subconjuntos representan un caso especial de las jerarquías de generalización. Hay problemas donde se tiene una generalización de la que se desprende sola especialización. Éste es el caso de los subconjuntos.”*[4]

#### 3.3.2 COBERTURA:

*“La propiedad de cobertura define el grado de relación entre padres e hijos. Existen 2 coberturas a definir: La primera de ellas determina si la relación entre padres e hijos es total o parcial. Una cobertura es total cuando cada elemento del padre está contenido en alguno de los hijos. La cobertura es parcial cuando pueden existir elementos del padre que no se instancien sobre los hijos.”*[4]

*“Por otra parte, la cobertura puede ser exclusiva o superpuesta. En este caso, lo que se analiza es si es un elemento del padre puede o no estar en más de un hijo. Si solo puede está en un hijo, la cobertura es exclusiva, en tanto que si puede estar en varios, se denomina superpuesta.”*[4]

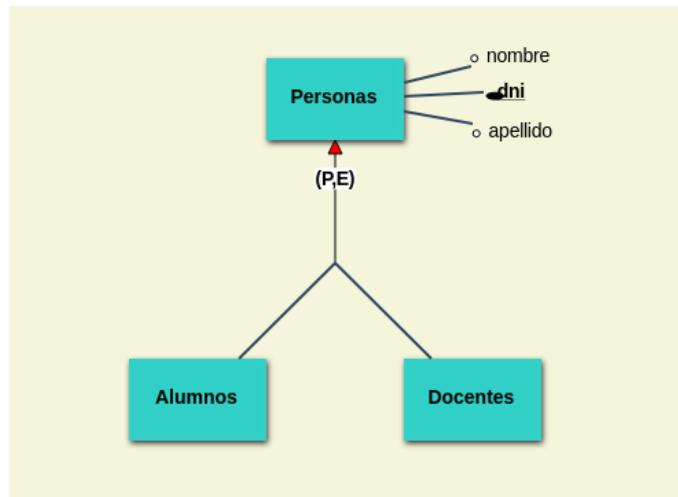


Fig. 7 Ejemplo de una jerarquía, con cobertura Parcial exclusiva

Como se muestra el caso de la Figura 7, el universo de las personas, está formado por alumnos y docentes, si se analiza el problema teniendo en cuenta el resto del personal, por ejemplo, ayudantes, directores, y demás, la cobertura sería Parcial (P). En cambio sí para un problema en particular, solo se requieren docentes y alumnos, entonces la cobertura es Total (T).

En la misma figura, si para el problema analizado, el docente no puede ser alumno, y viceversa, la cobertura es Exclusiva (E). Por otro lado, si en el problema existen alumnos que son docentes, como por ejemplo los ayudantes, la cobertura es Superpuesta (S).

## CAPÍTULO 4: CONCEPTOS GENERALES DEL MODELO LÓGICO

### 4.1 MODELO LÓGICO (ENTIDAD-RELACIÓN)

*“El propósito de la generación de un modelo ER lógico es convertir el esquema conceptual a un modelo más cercano a la representación entendible por el SGBD.” [4]*

Al comenzar con el modelado lógico se necesita definir qué tipo de SGBD se utilizará luego para su implantación física. La cadena de pasos de conversión está relacionada con el tipo de SGBD (relacional, orientado a objetos, jerárquico o de red) a utilizar.

En este caso se toma como ejemplo un SGBD relacional, dado que actualmente el Modelo Relacional es más utilizado por los Sistemas de Información generados para usuarios finales y el utilizado por la asignatura IBD.

### 4.2 CARACTERÍSTICAS DEL DISEÑO LÓGICO

El Modelo Lógico es una derivación del Modelo Conceptual y tiene 4 entradas que se explican en el capítulo 11 del Libro Introducción a las Bases de Datos [4] y lo citaremos en esta sección para explicar un poco mejor sus características.

- Esquema conceptual

Es el resultado de la etapa inmediata anterior. El esquema conceptual representa la solución, a juicio del analista, respecto del problema original. El esquema lógico a obtener debe representar la misma información disponible en el modelo conceptual.

- Descripción del modelo lógico a obtener

Aquí se deben definir las reglas que se aplicaran en el proceso de conversión. Esas reglas están ligadas al tipo de SGBD seleccionado.

- Criterios de rendimiento de la base de datos

Durante la fase de diseño conceptual, se consideran los requerimientos del usuario. No obstante, hay otro tipo de necesidades que no se pueden definir sobre el modelo conceptual. Estas necesidades tienen que ver por lo general con requerimientos, en general, no funcionales del problema, como por ejemplo la performance de la BD.

Así una regla puede dar alternativas de solución y el analista deberá optar por aquella que permita alcanzar los estándares de rendimiento definidos para el problema.

- Información de carga de la BD:

Este concepto aparece, en cierta forma ligado al concepto anterior. Cuando se genera el esquema lógico, el analista debe observar cada entidad e interrelación definida, y ver la probable evolución de la información contenida en esas estructuras. De este modo, la decisión final sobre el esquema de una relación o entidad dependerá del número probable de elementos que la compondrán, con el propósito de mantener la performance bajo control.

#### 4.3 PASAJE DE MODELO CONCEPTUAL A LÓGICO

Cuando se llega al punto del pasaje de modelos, del conceptual al lógico, hay una serie de decisiones que tomar. Estas están relacionadas, básicamente, a cuestiones generales de rendimiento y a un conjunto de reglas que actúan sobre características del esquema conceptual que no se presentan en los SGBD relacionales. Por ejemplo, el concepto de herencia no existe en el modelo relacional; esto implica que las jerarquías deberán ser resueltas para adaptarlas a este contexto. Además, el modelo relacional carece de un dominio que permita definir varios atributos, es decir, no es posible representar atributos compuestos.

##### 4.3.1 DECISIONES SOBRE DATOS DERIVADOS

Los datos derivados son básicamente atributos, que contiene información que puede obtenerse de otra forma desde el modelo. Es decir, que su valor para este tipo de atributo

se puede calcular a partir de los valores de uno o varios atributos que no necesariamente deben pertenecer a la misma entidad o relación.

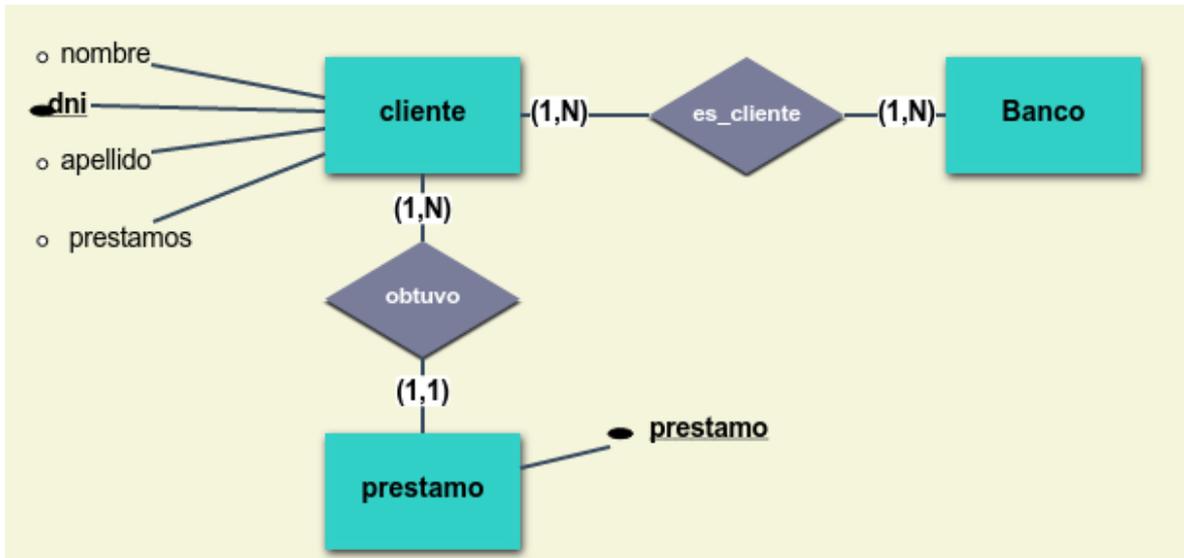


Fig. 8 Atributo Derivado

Por ejemplo, en la Figura 8, el conjunto de entidades cliente que tiene un atributo préstamos representa cuántos préstamos tiene un cliente en el banco. Ese atributo se puede derivar al contar el número de entidades préstamo asociadas con ese cliente.

Un atributo derivado tiene la ventaja de permitir obtener determinada información de manera directa, sin necesidad de realizar cálculo alguno.

Si la información del atributo derivado es muy solicitada, estará disponible sin la necesidad de contabilizar cuántas veces aparece un cliente en la relación "obtuvo". La desventaja de un atributo derivado radica en la necesidad de ser recalculado cada vez que la información que contiene se modifica.

Es decisión del analista mantener o no un atributo derivado, de acuerdo a las ventajas/desventajas a obtener para un problema en particular.

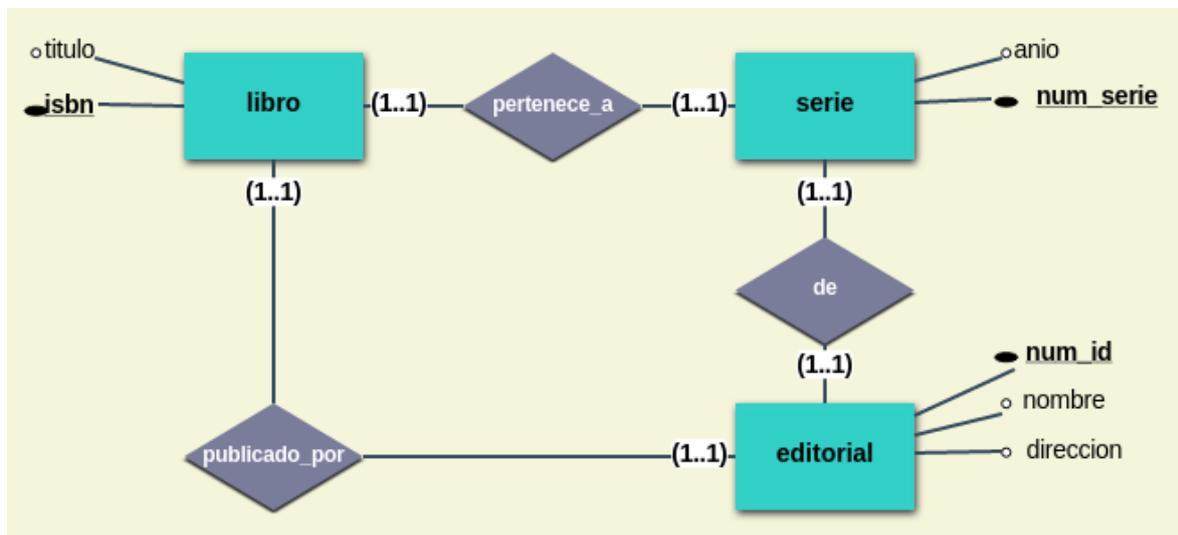


Fig. 9 Ejemplo de ciclos de Relaciones. Redundancia.

#### 4.3.2 DECISIONES SOBRE CICLOS DE RELACIONES:

Ciertas veces en los esquemas que se diseñan, se genera repetición de información innecesaria, estos casos son necesarios reconocerlos para que no se generen ciclos de relaciones, y así cumplir con la propiedad de *Minimalidad*.

Un esquema es mínimo -como se explicó con anterioridad- cuando cada concepto se representa una sola vez en el modelo.

Existen ciclos redundantes los cuales pueden estar presentes por decisión del diseñador, ya que esto a veces, tiene mayor performance que si se eliminase.

La Figura 9 presenta un ciclo generado entre las entidades Libro, Editorial y Serie. Se repite información dado que la Editorial de la Serie, se puede obtener a través de la entidad Libro. Por lo tanto la relación “de” es redundante. Si se elimina, el modelo quedaría mínimo.

Como en cada decisión, depende siempre del analista quien tiene que optar por tener el modelo mínimo, o preferir que el modelo generado implique menos tiempo de procesamiento en las consultas.

### 4.3.3 DECISIONES SOBRE ATRIBUTOS POLIVALENTES:

En los modelos, como en la vida real, existen datos que requieren tomar más de un valor del mismo tipo pero diferente. En un diseño esto se representa con lo que se denomina, *Atributo polivalente*. Por ejemplo los atributos teléfonos o títulos de una entidad Persona son algunos ejemplos de ellos.

En esos casos una persona podría tener varios teléfonos o varios títulos, sin un límite a priori.

Los SGBD relacionales no permiten que un atributo contenga valores múltiples determinados dinámicamente. Esto lleva a dos situaciones extremas, en primer lugar se puede alocar un determinado espacio, fijo, para almacenar los distintos posibles valores del atributo teléfono, por ejemplo, se puede establecer que podrá contener hasta 10 valores diferentes. En general, una persona puede tener dos o tres teléfonos; con el consiguiente desperdicio de espacio. Pero por lo contrario, se puede dar el caso que una persona tenga 11 teléfonos y no sea posible registrar a uno de ellos.

Ambas situaciones son anómalas, y tienen que ver con definir una estructura estática para almacenar información que, a priori, no se puede determinar la cantidad. En este caso la solución debe implementarse con otro criterio. El criterio establecido es denominado primera forma normal.

*“Un modelo está en primera forma normal (1FN) si todos los atributos que conforman las tablas del modelo son monovalentes.”[4]*

Entonces se puede decir que el modelo estará en 1FN sino tiene atributos con más de un valor posible.

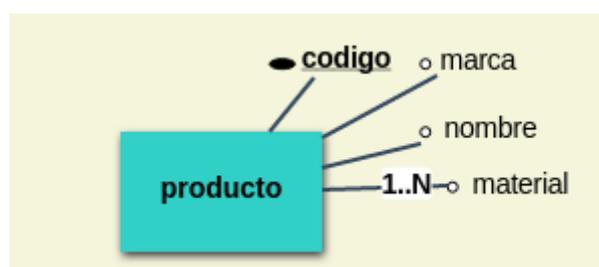


Fig. 10 Ejemplo de Atributo Polivalente

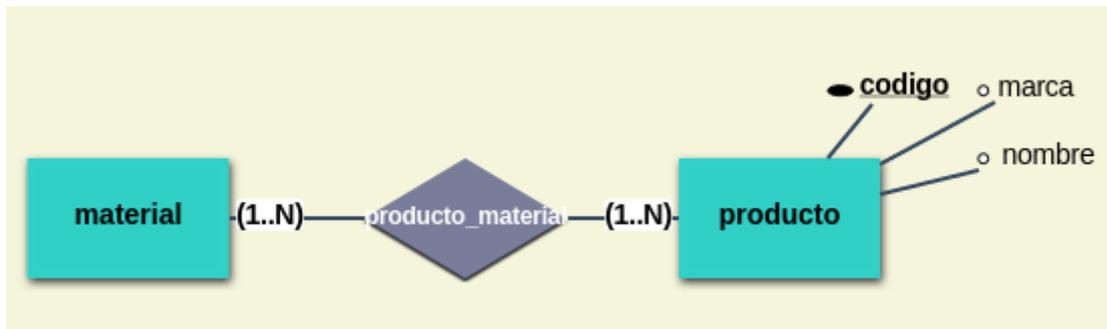


Fig. 11 Eliminación de Atributo Polivalente

En la Figura 10, la entidad Producto tiene definido el atributo material como polivalente obligatorio (1, N). Existe una manera para cumplir la 1FN y consiste en generar una nueva entidad denominada Material, y establecer la relación “producto\_material” entre producto y material, quitando así, el atributo *material* de la entidad producto. La relación “producto\_material” se define como una relación muchos a muchos, e indica que un producto puede tener varios materiales y que un material puede corresponder a varios productos, tal como se presenta en la Figura 11.

#### 4.3.4 DECISIONES SOBRE ATRIBUTOS COMPUESTOS:

Un atributo es compuesto si está constituido por varios atributos simples. Los SGBD tampoco soportan esto.

Por ejemplo, la dirección de una persona se puede definir como la conjunción de calle, número, provincia y localidad.

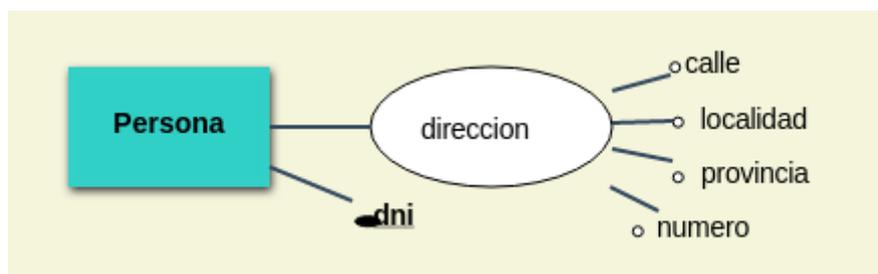


Fig. 12 Ejemplo de Atributo Compuesto

En el pasaje a Modelo lógico es necesario decidir cómo convertir un atributo compuesto. La Figura 12 presenta el atributo compuesto dirección de la entidad Persona. Existen tres posibilidades para convertir este atributo compuesto en el modelo lógico. La elección dependerá del analista de la BD, teniendo en cuenta criterios de rendimiento y utilización de la BD.

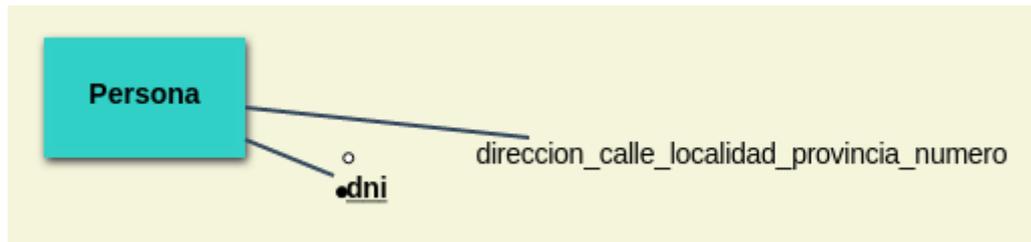


Fig. 13 Eliminación de Atributo Compuesto: Concatenación

La primera opción consiste en generar un único atributo que resulte de la concatenación de todos los atributos simples que contiene el atributo compuesto. Como se observa en la Figura 13, el atributo dirección, se define con una cadena de caracteres, donde el usuario debería ingresar todos los datos de un domicilio: provincia + localidad + calle + número. Esta solución es simple y sencilla de implantar, pero se debe considerar que al unir todos los atributos simples que forman el compuesto, se pierde la identidad de cada atributo simple.

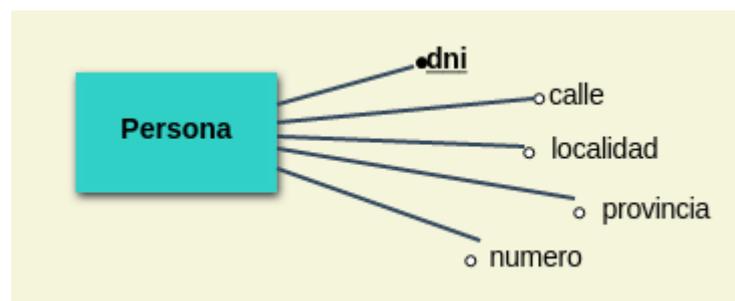


Fig. 14 Eliminación de Atributo Compuesto: Atributos simples

La segunda solución, se presenta en la Figura 14. Consiste en definir todos los atributos que formaban parte del atributo compuesto, como simples. En este caso, se quita el atributo dirección y en la entidad Persona se definen los atributos calle, número, localidad y provincia. La cantidad de atributos aumenta pero esta solución permite al usuario definir cada uno de los datos en

forma independiente. Generalmente esta solución es la más utilizada.

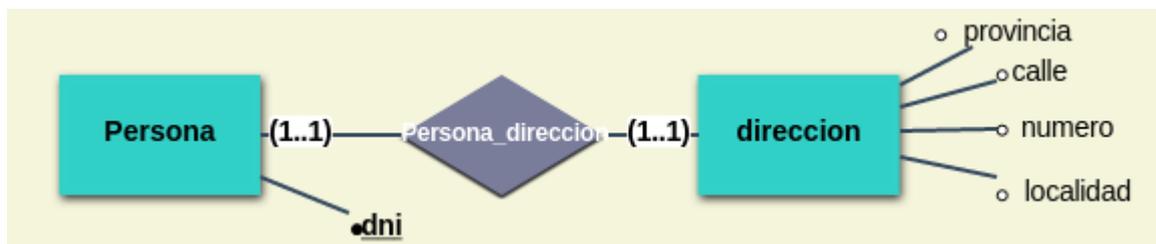


Fig. 15 Eliminación de Atributo Compuesto: Genera la nueva entidad Dirección

La tercera opción consiste en generar una nueva entidad, que represente el atributo compuesto, conformada por cada uno de los atributos simples que contiene. Esta nueva entidad debe estar relacionada con la entidad a la cual pertenecía el atributo compuesto. Como se puede observar en la Figura 15 esta es la solución más radical, se capta mejor la esencia del atributo compuesto, pero es la opción más compleja.

#### 4.3.5 DECISIONES SOBRE JERARQUÍAS:

En el modelo relacional (MR) o lógico, el concepto de herencia no está soportado, por lo tanto, cuando se pasa del modelo conceptual al lógico, las jerarquías no pueden permanecer en el esquema resultante.

Dentro del esquema lógico la información de la jerarquía tiene que estar representada. Por tal motivo es necesario encontrar un mecanismo que las incorpore, capte el dominio de conocimiento que representan, bajo un esquema que no administre herencia.

En el pasaje de modelos, se definen tres opciones para tratar de eliminar una jerarquía del esquema conceptual, estas dependen del tipo de cobertura que tenga la jerarquía. El concepto de cobertura se explicó en el capítulo anterior.

Estas son:

- ✓ *“Eliminar a las especializaciones (subentidades o entidades hijas) y dejar solo la generalización (entidad padre), la cual incorpora todos los atributos de sus hijos. Cada uno de estos atributos deberá ser opcional (no obligatorio)” [4].* Esta opción se puede ejecutar con cualquiera de las coberturas.
  
- ✓ *“Eliminar la entidad generalización (padre) y dejar solo las especializaciones. Con esta solución los atributos del padre deberán incluirse en cada uno de los hijos.”[4].* Esta opción, no siempre se puede aplicar. Supongamos el caso de la cobertura cuando es Parcial, como ya se ha explicado, la cobertura es parcial cuando algunos de los elementos contenidos en la entidad padre, no están cubiertos por las especificaciones. En este caso, si se elimina el padre, dichos elementos no tendrán más cabida en el modelo.

Si se da la cobertura superpuesta, esta opción tampoco resulta práctica, ya que algunos elementos del padre se repiten, en varios hijos, esto significa que se deberá repetir información en las subentidades.

- ✓ *“Dejar todas las entidades de la jerarquía, convirtiéndola en relaciones uno a uno entre el padre y cada uno de los hijos. Esta solución permite que las entidades que conforman la jerarquía, mantengan sus atributos originales. Se genera la relación explícita ES\_UN entre padre e hijos.”[4]* Esta opción, es aplicable siempre, y es la que mejor representa el modelo conceptual, es también la más usada, y la más legible.

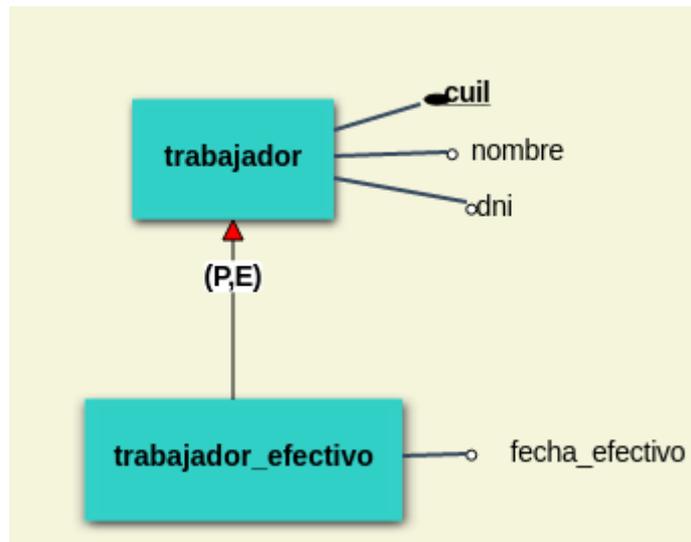


Fig. 16 Ejemplo subconjunto.

Los subconjuntos son jerarquías especiales, con cobertura parcial exclusiva. En este caso, tanto la primera como la tercera alternativa de solución son aplicables, quedando descartada la segunda alternativa. Un ejemplo de subconjuntos se presenta en la Figura 16.

#### 4.4 EJEMPLO DE ELIMINACIÓN DE JERARQUÍA:

Se modelan los Socios del club. Los socios pueden ser de vips o vitalicios. Los socios vip tienen beneficios extras a los socios comunes, pero pagan una cuota vip algo más cara que la general. Y los socios vitalicios del club, son aquellos que tienen 20 años de asociados, ellos tienen registro libre en el club, y un invitado anual, el cual puede usar las instalaciones del club.

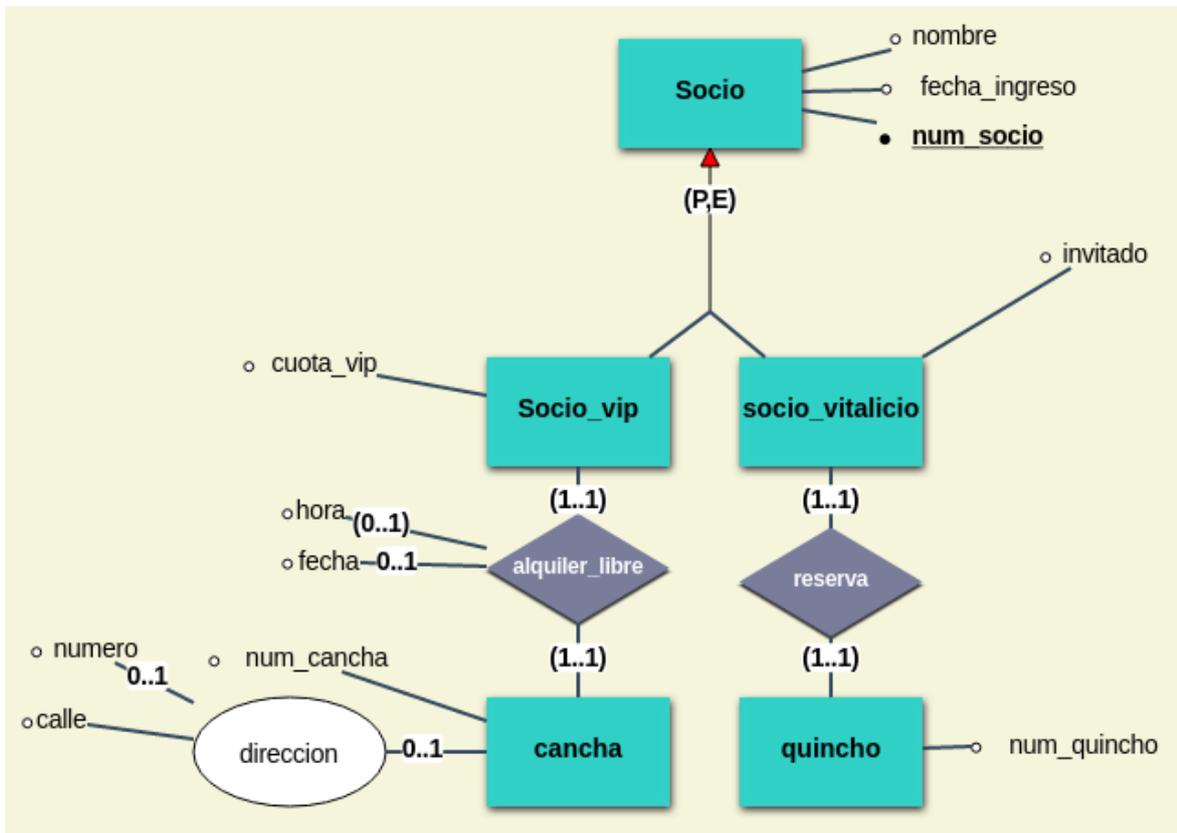


Fig. 17 Jerarquía Socios de un Club

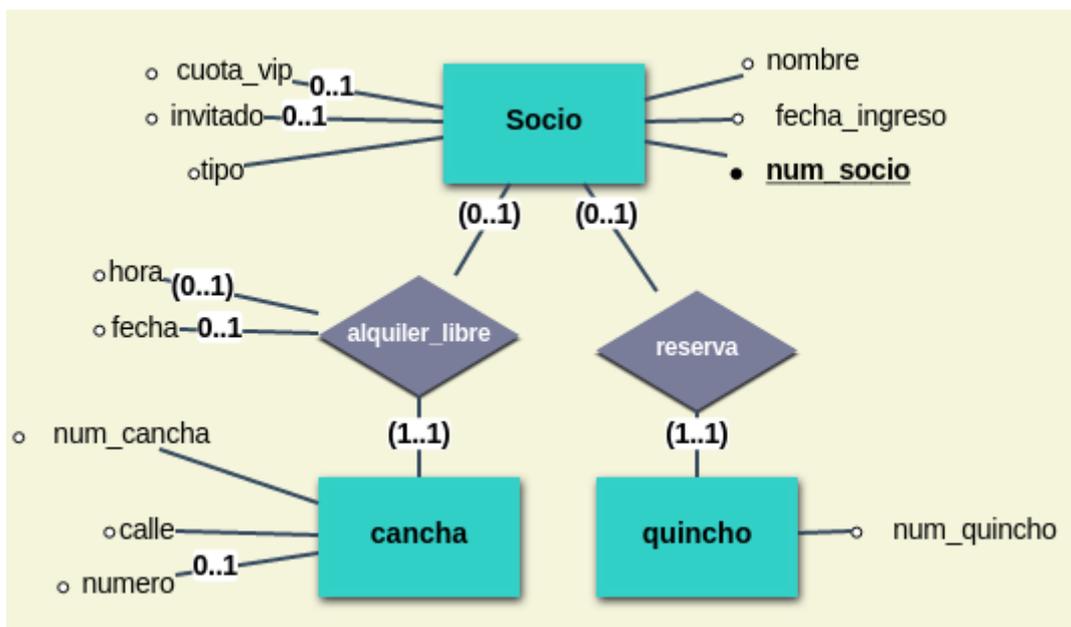


Fig. 18 Eliminación de Jerarquías: Se quitan las entidades hijas

La Figura 17 ilustra el resultado de diseñar la jerarquía previamente descrita. Al aplicar la primera alternativa, se quitan las entidades hijas de la jerarquía y se dejan solamente la entidad padre. En la Figura 18 se puede observar que el modelo queda más reducido.

Sobre la entidad padre quedan los atributos que estaban contenidos en las entidades hijas; estos atributos son ahora opcionales. La cardinalidad de Socios en la relación “alquiler\_libre” se convierte en opcional en lugar de obligatoria, dado que un socio vitalicio no tiene alquiler libre.

## CAPÍTULO 5: CONCEPTOS GENERALES DEL MODELO FÍSICO

En el capítulo anterior hemos presentado el Diseño Lógico como la segunda fase del diseño. A partir de él se aborda la tercera fase, que es el Diseño Físico. Así como en el lógico se tienen en cuenta principalmente los aspectos relativos al significado de los datos, en el físico se consideraran fundamentalmente los aspectos de los datos relacionados con el rendimiento y la disponibilidad de la aplicación. En este capítulo se presentaran los pasos necesarios que permiten generar el modelo físico de datos a partir del modelo lógico explicado en el capítulo anterior.

### 5.1 MODELO FÍSICO (RELACIONAL)

Los diagramas entidad-relación suponen un primer nivel del análisis de los datos del sistema.

*“El modelo relacional representa una BD como una colección de archivos denominados tablas, las cuales se conforman por registros. Cada tabla se denomina relación, y está integrada por filas horizontales y columnas verticales. Cada fila representa un registro del archivo y se denomina tupla, mientras que cada columna representa un atributo del registro.” [4]*

### 5.2 PASAJE DE MODELO LÓGICO A FÍSICO

El primer paso en la conversión del esquema lógico hacia el esquema físico consiste en la eliminación de identificadores externos; el segundo paso es la selección de claves primaria, candidata y secundaria, el tercer paso es la conversión de entidades, que por lo general cada una de las entidades se convierten en tabla; el cuarto y último paso es la conversión de relaciones, en el cual se deben analizar 3 situaciones diferentes, que tienen que ver básicamente con la cardinalidad que existe entre las relaciones definidas.

#### 5.2.1 ELIMINACIÓN DE IDENTIFICADORES EXTERNOS

Cada una de las entidades que hayamos definido en el esquema lógico, deben tener

definido un identificador de manera interna, de no ser así, dicho identificador externo deberá ser eliminado; esto se hace incorporando dentro de la entidad que contenga los identificadores externos, los atributos que forman parte del identificador. Por ejemplo si un estudiante se identifica por número de legajo + nombre de la facultad en la que estudia, ya que esta es la manera en que un estudiante se puede identificar de otro estudiante en la misma universidad, ya que en la misma universidad, pero distinta facultad, puede haber otro estudiante con el mismo número de legajo. Por lo tanto para realizar la conversión es necesario tomar el identificador de la entidad facultad e incorporarlo en la entidad estudiante.

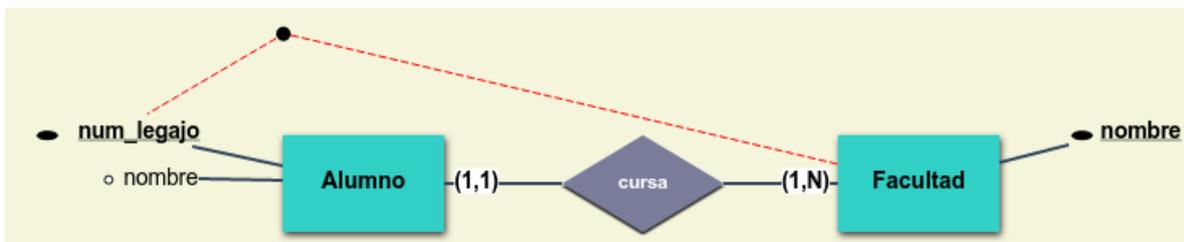


Fig. 19 Ejemplo de identificador externo

#### Tabla Alumno:

Alumno: [Alumno\_id | num\_legajo | nombre | F Facultad\_id]

#### Identificadores Compuestos

Facultad\_alumno: [Facultad\_id | num\_legajo]

### 5.2.2 SELECCIÓN DE CLAVES PRIMARIAS, CANDIDATA Y SECUNDARIA

Cuando se genera el esquema de modelo físico sobre el modelo relacional, se debe decidir el criterio de definición de clave primaria a partir de los identificadores reconocidos en cada entidad.

Si una entidad solo tiene definido un identificador, ese identificador es clave primaria de la tabla, en cambio si la entidad tuviera más de un identificador, la selección de la clave primaria debería hacerse de la siguiente manera:

Entre un identificador simple y uno compuesto, debería tomarse el simple, ya que es más fácil de usar y manipular.

Entre 2 identificadores simples, se opta por aquel de menor tamaño físico.

Entre 2 identificadores compuestos, se debería optar por aquel que tenga menor tamaño en bytes.

Las consideraciones anteriores definirán la clave primaria, el resto de los identificadores pasarán a ser clave candidata (CC). Sin embargo los SGBD ofrecen una alternativa que resulta ser la más conveniente, y es a través del uso de un tipo de dominio auto incremental. Así, una tabla que tenga un atributo autoincremental tiene definida una CP que será tratada por el SGBD de manera exclusiva. El usuario solo tiene permitida la acción de lectura este atributo, es decir que no la puede generar, ni modificar, ni borrar.

### 5.2.3 CONVERSIÓN DE ENTIDADES

Por lo general la conversión de entidades comienza con el análisis de las entidades definidas en el modelo lógico, convirtiéndose cada una en una tabla del modelo físico, y cada uno de los atributos de la entidad se convierte en atributos de la tabla.

Puede existir una excepción a la regla de que “toda entidad se transforma en tabla”, suponga el caso en que se debe modelar un problema en donde se representan la entidad provincia y un organismo regulador, y en este caso una no puede existir sin la otra, es decir que una provincia puede tener un único organismo regulador, y un organismo pertenece solo a una provincia. Por lo tanto la cardinalidad de esta relación será 1-1 con participación obligatoria, por este motivo, la solución más variable en el proceso de conversión consiste en generar una única tabla que aglutine los atributos de ambas. Este ejemplo se ilustra en la Figura 20.

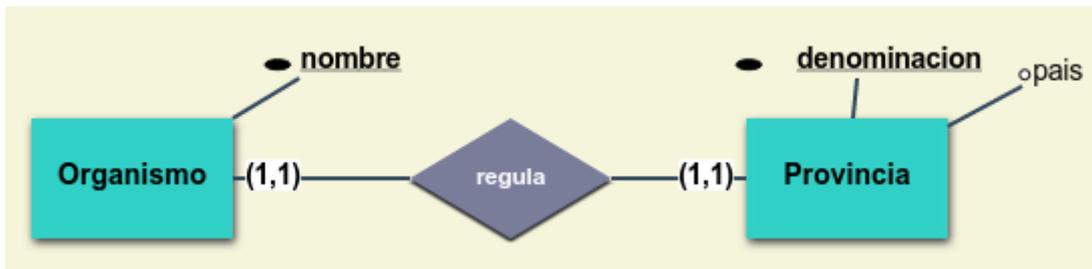


Fig. 20 Ejemplo conversión de entidades, cardinalidad 1-1

**Tabla Provincia:**

Provincia: [Provincia\_id | denominacion | pais | nombre]

5.2.4 CONVERSIÓN DE RELACIONES

El proceso de conversión continúa con el análisis de relaciones entre las entidades, son 3 casos y lo veremos a continuación:

Cardinalidad muchos a muchos:

Siempre que una relación entre 2 o más entidades presente una relación N-N sin importar la cardinalidad mínima, ésta relación debe hacerse tabla, y sus identificadores serán la unión de las 2 claves primarias de las entidades con las que se relacionaban.

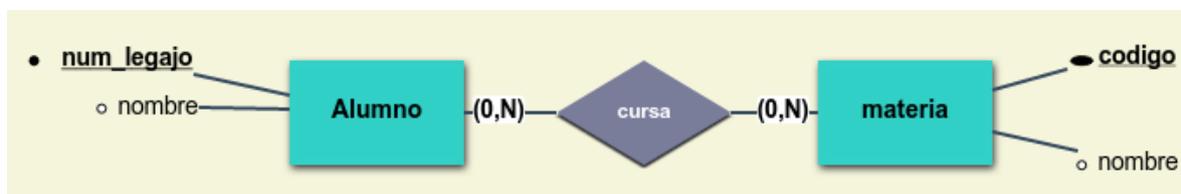


Fig. 21 Ejemplo de conversión de relaciones con cardinalidad N-N

**Tablas:**

Alumno: [Alumno\_id | num\_legajo | nombre ]

Materia: [materia\_id | código | nombre]

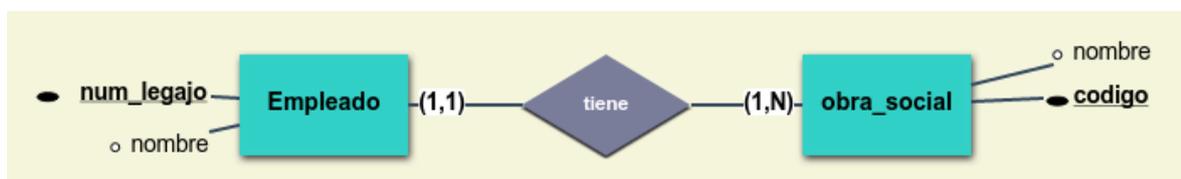
**Cursa:** [*curso\_id* | *F\_Alumno\_id* | *F\_materia\_id*]

Otra manera de seleccionar la clave primaria de esta nueva tabla es, como se mencionó previamente, utilizar una clave auto incremental. Esta solución es así, porque en el caso en los casos en que se dan estas relaciones N a N, no se puede distinguir unívocamente a una de las entidades representadas en el modelo lógico, con solo tener las tablas de las entidades. Vamos a ver esto en un ejemplo. Se supone el caso en que Un ALUMNO cursa MATERIAS, en este caso, la relación es N a N , ya que un alumno puede cursar 0 o N materias, y una materia puede ser cursada por 0 o N alumnos, siendo acá la cardinalidad máxima, N-N por lo tanto si se quiere identificar una materia, se puede hacerlo mediante el id de la materia, lo mismo si se quiere identificar a un alumno, mediante el id, de este; otro caso, es cuando se requiere identificar qué materia cursa un alumno. Para esto es que se determina la tabla CURSA, donde sus identificadores como antes dijimos será la unión de num\_legajo, y código, o un auto-incremental que sea id\_cursa, de cualquier manera, esta es la solución para representar una relación N a N en un modelo físico.

#### Cardinalidad uno a muchos:

La solución para este tipo de cardinalidades, dependerá de la cardinalidad mínima definida en la relación:

Uno a muchos con participación total: Cuando la relación entre 2 o más entidades es de 1 a N, con participación obligatoria de ambos lados, la regla de conversión para las entidades produce crear ambas tablas, y agregar el identificador de la tabla con relación 1 a n, en la tabla que tiene relación 1,1.



*Fig. 22 Ejemplo de conversión de relaciones 1-N, con participación obligatoria*

#### Tablas:

**Empleado:** [Empleado\_id | num\_legajo | nombre | F\_obra\_social\_id]

**Obra\_social:** [obra\_social\_id | nombre | codigo]

Se toma el ejemplo de un empleado y una obra social tal como se muestra en la Figura 22, en este caso el empleado debe elegir obligatoriamente 1 obra social, y la obra social tiene al menos un empleado que la elige, o muchos. Entonces como se dijo con anterioridad se puede incorporar a la tabla empleado el id de la obra social estableciendo de este modo el vínculo sin necesidad de generar otra tabla.

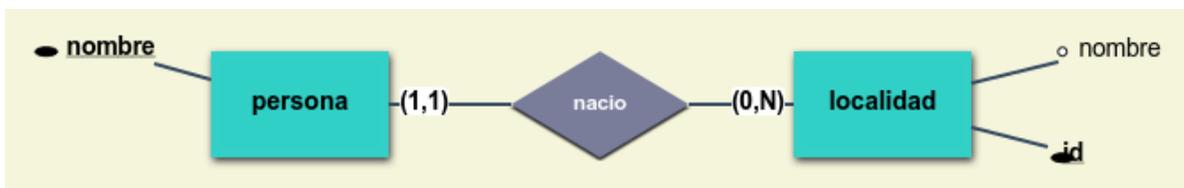


Fig. 23 Ejemplo de conversión de relaciones con participación parcial del lado de muchos

#### Tablas:

**persona:** [persona\_id | nombre | F\_localidad\_id]

**localidad:** [localidad\_id | nombre | id]

- *Uno a muchos con participación parcial del lado de muchos:* Este caso se resuelve de manera que el caso anterior, se toma el ejemplo de la Figura 23, donde una persona nació en una única localidad, y en una localidad pueden haber nacido 0, o n personas. Por lo cual, si se agrega en la entidad Localidad el atributo id\_persona, como va a haber localidades en las que no haya nacido ninguna persona, esto puede generar tuplas en donde el atributo de id\_persona sea nulo, y como no es recomendable manejar atributos nulos, se agregará en la tabla Persona el atributo id\_localidad, como todas las personas nacieron en una ciudad si o si, este no generara ninguna tupla con atributos vacíos.

- *Uno a muchos con participación parcial del lado de uno:* Este caso se soluciona de la misma manera que el caso de relación N-N, donde se genera también la tabla de la relación

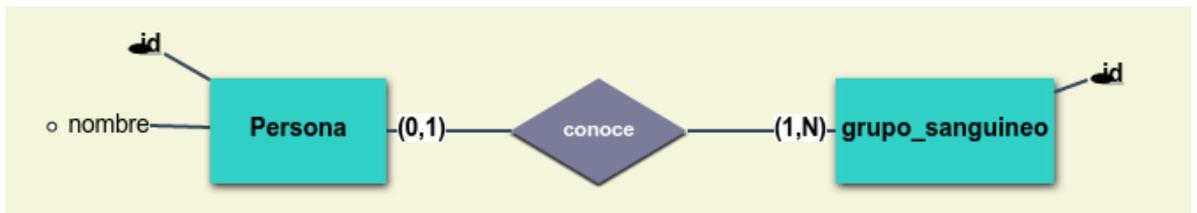


Fig. 24 Ejemplo de conversión de relaciones con participación parcial del lado de uno

**Tablas:**

**Grupo\_sanguineo:** [grupo\_sanguineo\_id | id]

**Persona:** [persona\_id | nombre | id]

**conoce:** [conoce\_id | F\_Persona\_id | F\_grupo\_sanguineo\_id]

Se toma como ejemplo el caso expresado en la Figura 24. Todas las personas tienen un grupo sanguíneo, pero no todas lo conocen, así que la cardinalidad de persona hacia grupo sanguíneo va a ser 0-1, al revés va a ser 1-N, al menos 1 persona pertenece a ese grupo sanguíneo. En el pasaje a modelo físico, se crea la tabla persona, con el id de grupo\_sanguíneo en ella. Esto hará que genere tuplas con este atributo en nulo, ya que hay personas que no lo conoce, por ende, no es un caso viable, tampoco si se agrega el atributo id\_persona en la tabla grupo\_sanguíneo, ya que para saber a qué grupo sanguíneo pertenece una persona, se debería recorrer todas las tuplas generadas por grupo sanguíneo, y ver cual tiene el id de la persona que se busca, lo cual es muy ineficiente, y además habría repetición de información, se repetirían todos los atributos del grupo sanguíneo, por cada persona que pertenezca a ese grupo sanguíneo, con lo cual, se opta por crear una nueva tabla, CONOCE, la cual contenga el id de la persona, y el id de su grupo sanguíneo. De esta

manera se crearán más tablas, pero no se trabaja con valores nulos, lo cual no es recomendable.

- *Uno a muchos con cobertura parcial de ambos lados:* Este último caso se puede resolver procediendo de la misma manera que el caso anterior (la relación se convierte en tabla), u optar por la variante que genera valores nulos, pero nunca se recomienda.

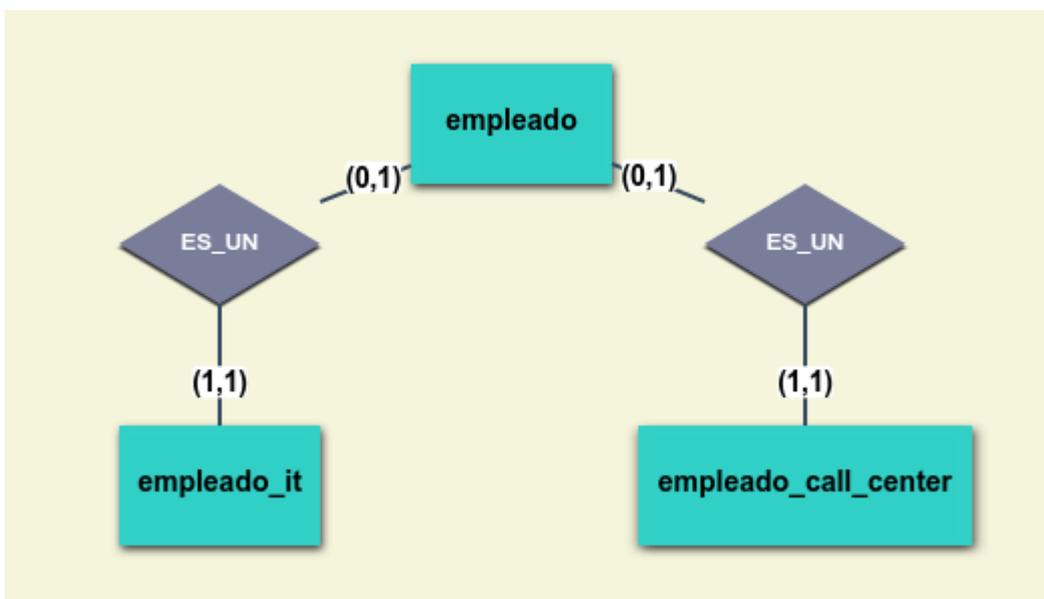


Fig. 25 Ejemplo de conversión de relaciones uno a uno, con participación parcial de un lado.

#### Cardinalidad uno a uno:

Esta situación ya fue planteada, cuando la cardinalidad es uno a uno con participación total de ambos lados, solo se debe generar una única tabla que contenga los atributos, de ambas entidades.

- *Cardinalidad uno a uno con participación parcial de un lado:* Por lo general esta situación es muy común, después del pasaje de conceptual a lógico, cuando en la eliminación de jerarquía se decide dejar todo. Esto genera una relación “es\_un” entre la entidad padre, y las entidades hijas.

Cuando se resuelve una jerarquía dejando todas las entidades, se resuelve de la manera en que se muestra a Figura 25, porque cada uno de estos empleados, son empleados, pero en cambio el empleado, puede ser uno de los 2 tipos de empleados, por eso la relación (0,1) (1,1). Este tipo de relación se resuelve agrupando todas las entidades en una tabla. Pero hay una particularidad, el id de la entidad padre es parte también de las tablas hijas generadas, ya que un empleado del área de IT, y un empleado del área de call-center, comparten características de empleados. El id del empleado, es una clave primaria, no es auto incremental.

**Tablas:**

**Empleado:** [empleado\_id]

**empleado\_it:** [empleado\_it\_id | F\_empleado\_id]

**empleado\_call\_center:** [empleado\_call\_center\_id | F\_empleado\_id]

- *Uno a uno con participación parcial de ambos lados:* Ese caso no es muy común, pero si se llegara a dar el caso, lo que se aconseja es convertir a tabla cada entidad involucrada, y la relación también hacerla tabla, la cual deberá incluir todas las CP de las otras tablas involucradas.

## CAPITULO 6: DESCRIPCIÓN DE CASER 3.0

### 6.1 INTRODUCCIÓN:

En capítulos anteriores se expusieron todos los temas pertinentes al diseño de una base de datos, forjando así un marco teórico en el cual se basó el desarrollo de esta herramienta, como así también la elección de sus funcionalidades y componentes. A continuación, se detallarán las versiones anteriores de CasER (Versiones 1 y 2), y se expondrá el propósito, la utilidad y los aportes de la nueva versión (Versión 3).

Todas las versiones de la herramienta fueron desarrolladas por alumnos de la facultad de Informática de la UNLP como proyectos de tesinas de grado, ayudando al aprendizaje de los alumnos de la asignatura “Introducción a la Bases de datos” (IBD), cuyo tema principal es el diseño de las bases de datos.

La primera versión que se desarrollo fue denominada “Caser: Herramienta para modelado conceptual de bases de datos”<sup>1</sup>, y la misma se enfocó en el primer paso del diseño de una base de datos, el modelo conceptual. Su característica principal consiste en asistir la creación de un esquema conceptual de alto nivel. Se parte de la especificación detallada de un problema a ser resuelto por un Sistema de Software (especificación de requerimientos), se asiste en la creación de un Modelo Conceptual de Alto Nivel (MCAN) y, luego de sucesivos refinamientos por parte del usuario, se obtiene un Modelo (o esquema) Conceptual Definitivo (MCDef).

La segunda versión de la herramienta se denominó “Caser versión 2.0: Herramienta para el diseño de Bases de Datos.” [2], extendiendo la funcionalidad de la primera versión. Se logró abarcar todo el proceso del diseño de una base de datos habiendo agregado en sus funcionalidades, el modelado lógico y físico de alto nivel.

En CasER 1.0, al existir la posibilidad de relacionar los elementos seleccionados de la especificación con los elementos que componen el diagrama o esquema, se logra alinear la especificación del problema con el modelo conceptual de datos que lo soporta. Con la versión 2.0, esta característica se ve enriquecida con la posibilidad de poder continuar desde el modelo conceptual con la creación de esquemas lógicos y físicos con solo tomar algunas decisiones, característica ausente en los productos comerciales actuales. Esto convierte a Caser en una herramienta muy útil para el proceso de enseñanza y aprendizaje del modelado de bases de

datos.

Cabe destacar que ambas herramientas, fueron diseñadas como aplicaciones de escritorio, donde solo se ejecutan bajo el Sistema Operativo Windows.

En esta nueva versión (Caser 3.0), se decidió cambiar la forma en que se utiliza la herramienta, para lo cual fue necesario desarrollar la herramienta de manera íntegra.

Se ideó basándose en la metodológica de “aprendizaje colaborativo”. Dicha metodológica incorpora el concepto de usuario como actor principal de la herramienta.

La característica principal de esta herramienta al igual que las versiones anteriores, continua siendo la de asistir en el proceso del diseño de una base de datos, pero se le agregan funcionalidades centradas en el usuario, tales como la posibilidad de crear un perfil de usuario, tener un espacio privado para guardar proyectos y poder compartirlos con otros usuarios.

## 6.2 REDISEÑO DE LA ARQUITECTURA:

Para poder suplir con las necesidades referentes al aprendizaje colaborativo, se requirió el rediseño completo de la arquitectura de las versiones anteriores.

La principal característica a suplir fue el manejo de usuarios, y que los mismos pudiesen subir y compartir sus proyectos en la nube. Para ello se cambió el paradigma de ejecución de la herramienta. En las versiones anteriores, se ejecutaba como una herramienta de escritorio, pero la nueva versión se ejecuta como una herramienta web, la cual requiere de un servidor web y los usuarios acceden a ella a través de un navegador web

### 6.2.1 ARQUITECTURA CASER 1.0/2.0

Ambas versiones cuentan con la misma arquitectura, la cual consiste en:

- ✓ Lenguaje de programación JAVA utilizando el IDE Eclipse [13].
- ✓ Ejecución desktop, en cualquier plataforma que soporte JAVA.
- ✓ La base de este desarrollo es RCP (Rich Client Plataform), una plataforma que permite crear aplicaciones Java multiplataforma.
- ✓ Se utilizó GEF (Grap Editing Framework), librería creada para facilitar la labor de

construcción de aplicaciones para dibujar diagramas.

- ✓ Se utilizaron otras herramientas que componen RCP, como son SWT (Standard Widget Toolkit) y JFace.

### 6.2.2 ARQUITECTURA E IMPLEMENTACIÓN CASER 3.0

La versión 3.0 cambia el paradigma de ejecución de la herramienta, llevándola a una arquitectura que soporta el acceso a través de internet. Debido al cambio radical de tecnología, es que ésta nueva versión no fue una extensión de las anteriores, sino que se repensó y se desarrolló íntegramente, sin reutilización de código. Por esto se considera que CasER 3.0 tiene un valor agregado, ya que se desarrollaron todas las funcionalidades que incluían las versiones anteriores, y además se agregaron funcionalidades que se detallarán a más adelante.

Esta nueva arquitectura consta de 3 características principales:

Escalable: La herramienta soporta la interacción de 1 usuario, así como también 100 usuarios y más de ser necesario.

Actualización transparente: Da la posibilidad de poder actualizar la aplicación (nuevas funcionalidades o corrección de errores, entre otras) sin necesidad de que el usuario tenga que descargar o instalar una nueva versión.

Disponibilidad: Se puede acceder a la aplicación desde cualquier parte del mundo, a cualquier hora, solo se necesita una conexión a Internet.

La arquitectura está formada por un Módulo Servidor Web, que se ejecuta en cualquier servidor web que soporte JAVA EE y que contenga una base de datos relacional; y por un Módulo Cliente Web, el cual se encuentra accesible a través de cualquier navegador que soporte la ejecución de código JavaScript, como ilustra la figura 26.

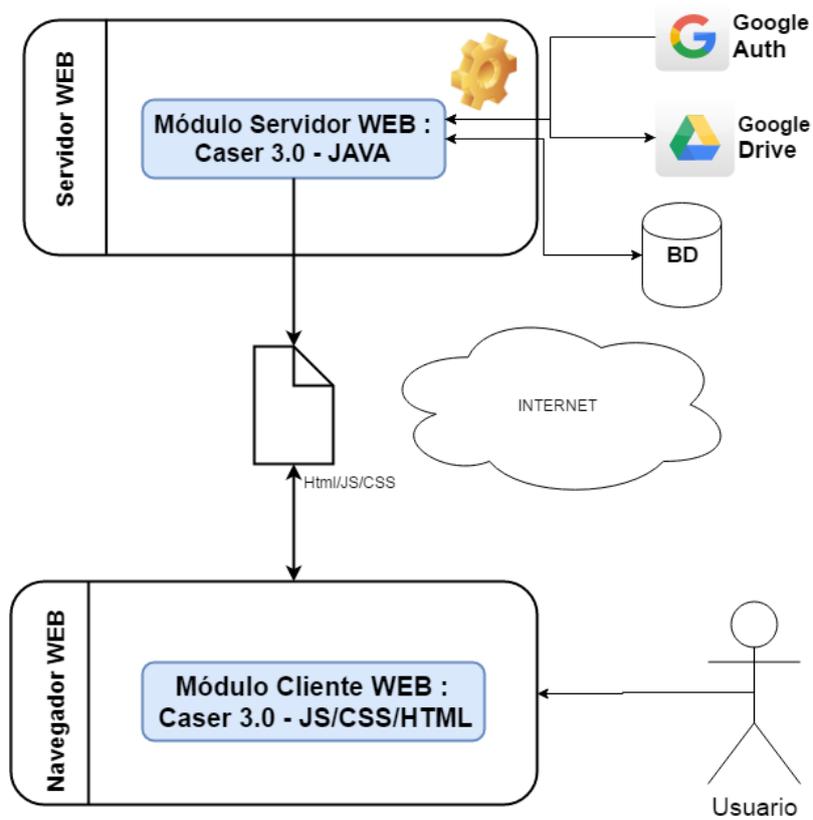


Fig. 26 Arquitectura Caser 3.0

De aquí en adelante llamaremos “*Servidor*” al Módulo Servidor Web Java que se ejecuta en el entorno del Servidor Web, y llamaremos “*Cliente*” al Módulo Cliente Web que se ejecuta en el entorno del Navegador WEB.

Como se puede apreciar en la Figura 26, el Servidor es el encargado de integrar la autenticación con Google, como así también interactuar con los archivos de Google Drive. También es el encargado de guardar en la Base de Datos (BD) la información extra de los usuarios, como por ejemplo, si es un Alumno o un Profesor. El servidor como mencionamos está íntegramente desarrollado con tecnologías JavaEE:

- ✓ se usó Spring Framework,
- ✓ JPA para acceder a una base de datos PostgreSQL<sup>I</sup>,
- ✓ y un cliente Java de Google para la autenticación y manejo de archivos.

Por otro lado, el “Cliente” es el encargado de darle toda la funcionalidad a la herramienta. Contiene cada una de las pantallas y es con el cual el usuario interactúa con la herramienta. Está en su mayor parte desarrollado con tecnologías JavaScript:

- ✓ Framework AngularJS<sup>III</sup> como estructura base, con un diseño MVC (de sus siglas en inglés Model-View-Controller) el cual maneja la interacción de cada uno de los botones y componentes de la herramienta.
- ✓ Framework JointJS<sup>IV</sup> para dibujar los gráficos y permitir al usuario interactuar y editarlos.
- ✓ Framework BackboneJS<sup>V</sup> para representar el modelo de las Tablas en el Modelo Físico.
- ✓ Framework Bootstrap<sup>VI</sup> para el look&feel de la herramienta.
- ✓ Google Api Client<sup>VII</sup>, para toda interacción con las distintas herramientas Google (incluida Google Analytics, el cual recopila información estadística sobre el comportamiento del usuario).
- ✓ Framework OnePageJS<sup>VIII</sup>, el cual se encarga de dividir una página en secciones, lo cual ayuda para el look&feel<sup>IX</sup>.

En contraste con las herramientas previas a esta nueva versión, CasER 3.0 es de Código Abierto<sup>X</sup>,

---

<sup>I</sup> <https://projects.spring.io/spring-framework/>

<sup>II</sup> <http://www.postgresql.org/es/>

<sup>III</sup> <https://angularjs.org/>

<sup>IV</sup> <https://www.jointjs.com/>

<sup>V</sup> <http://backbonejs.org/>

<sup>VI</sup> <http://getbootstrap.com/>

<sup>VII</sup> <https://developers.google.com/api-client-library/javascript/start/start-js>

<sup>VIII</sup> <http://alvarotrigo.com/fullPage/>

<sup>IX</sup> Aspecto y comportamiento de la interfaz gráfica.

<sup>X</sup> El **código abierto** es el software distribuido y desarrollado libremente. Se focaliza más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o de libertad que tanto se destacan en el software libre. La libertad se refiere al poder modificar la fuente del programa sin restricciones de licencia, ya que muchas empresas de software encierran su código, ocultándolo y restringiéndose los derechos a sí misma.

lo cual permite a los usuarios que lo deseen, aportar nuevos cambios, mejoras y funcionalidades a la herramienta, sin necesidad de pedir a los creadores el código fuente, ya que este código es de acceso público.

### 6.3 USUARIOS / HERRAMIENTA COLABORATIVA

En Caser 3.0, los “usuarios” toman un papel importante dentro de la herramienta. Para comenzar con el uso de la herramienta, primero el usuario debe registrarse, lo cual lo habilita para poder crear, editar o incluso compartir proyectos con otros usuarios. El registro de usuario permite identificar a las personas que utilizan la herramienta, y brindarles la posibilidad interactuar entre ellas.

La fase de identificación de usuarios, se realiza mediante el sistema de autorización de Google, lo cual beneficia a los usuarios de dicha plataforma, ya que no necesitan crear un usuario con una contraseña, ni tampoco validar su casilla de mail. En conclusión, CasER 3.0 no maneja la autenticación de usuarios, sino que le delega a Google esa tarea.

La integración con Google, no sólo se realizó con el propósito de identificar usuarios, sino también con la idea de poder integrar Google Drive<sup>1</sup>, el cual permite utilizar todas las ventajas de manejo de archivos en la nube, como también la posibilidad de compartirlos con otros usuarios.

Por lo tanto, cuando decimos que ésta es una versión que incorpora el concepto de “aprendizaje colaborativo”, dicha integración con Google Drive juega un papel más que importante, ya que permite intercambiar trabajos entre distintos usuarios (podría ser entre Alumno y Profesor, para una corrección) de manera instantánea. Mientras que en las versiones anteriores era necesario guardar el proyecto, enviarlo por mail, o incluso hasta entregar una copia mediante algún medio

---

<sup>1</sup> Google Drive es un servicio de alojamiento de archivos que fue introducido por Google el 24 de abril de 2012.

físico (Pen Drive, CD), en la nueva versión, con solo especificar el usuario a quien se desea compartir el proyecto, ya están ambos trabajando de manera colaborativa sobre un proyecto. Esto incentiva y mejora la calidad de comunicación entre los distintos usuarios de la herramienta, y motiva a trabajar en conjunto, lo cual ayuda en el aprendizaje y desarrollo de los dichos usuarios.

Por último, durante el uso de la herramienta, se releva información acerca de cómo los usuarios interactúan con la misma, de manera estadística, a fin de poder analizar el comportamiento de los usuarios, como así también poder asistir en caso de fallas dentro de ella.

#### 6.4 REDISEÑO INTERFACE DE USUARIO (UI)

Se realizaron grandes cambios en el look&feel<sup>1</sup> de la herramienta; logrando así una interfaz amigable de cara al usuario. Con el objetivo de mejorar la experiencia de usuario (en inglés user experience - UX) se deben tener en cuenta lo siguiente:

Capacidad de aprender: El usuario debe poder aprender con facilidad la utilización de la herramienta. Debe poder navegarla con facilidad, y recordar los pasos realizados para poder utilizarla rápidamente.

Capacidad de descubrir: El usuario debe poder encontrar las funcionalidades de la herramienta de manera rápida, y fácil. Para ello se realizó un gran cambio en la disposición de los elementos más importantes como **Botones** que permiten el pasaje de los modelos y **Acciones** tales como “guardar”, “compartir” o “crear nuevo proyecto”.

El rediseño de la interface de la herramienta tuvo como objetivo que ésta sea cómoda, práctica y ágil. Por ejemplo, cuando se quiere comenzar un nuevo proyecto, o continuar uno que ya estaba empezado, los botones de cómo hacerlo están visibles en la ventana del CasER 3.0, para que el usuario los encuentre con facilidad.

A continuación se ilustrarán algunos ejemplos de las mejoras mas notorias que se realizaron:

---

<sup>1</sup> Aspecto y comportamiento de la interfaz gráfica.

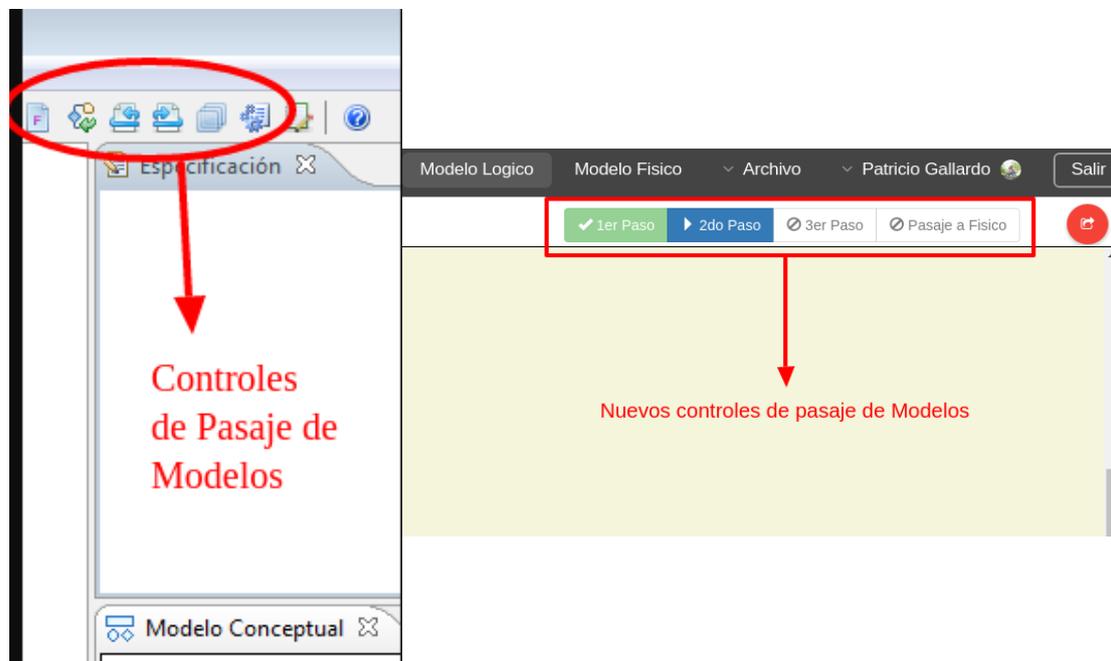


Fig. 27 Comparativa Controles Caser 2.0 / Caser 3.0

- ✓ La Figura 27, muestra cómo se rediseñaron los controles de pasajes de Modelos, en donde a la izquierda se puede apreciar que la versión 2.0 eran iconos poco legibles y no intuitivos para el usuario, mientras que a la derecha se puede ver como en Caser 3.0 se diseñó una barra de herramientas donde queda claro qué botón corresponde con qué acción y, por lo tanto, se encuentra de manera intuitiva por el usuario.
  
- ✓ La manera en que el usuario interactúa con la herramienta para realizar los distintos pasajes, se simplificaron en la nueva versión. En la versión 2.0 el usuario tiene que:
  - 1- Presionar el botón de Finalizar. Este paso también cumple la función de guardado.
  - 2- Presionar el botón de “Comenzar Pasaje a Modelo Lógico”.
  - 3- Presionar los botones “siguiente paso”, hasta terminar con el proceso de pasaje.

Esto requiere de mucha concentración por parte del usuario, quien debe saber en qué paso del proceso de pasaje se encuentra, y cuál es el próximo.

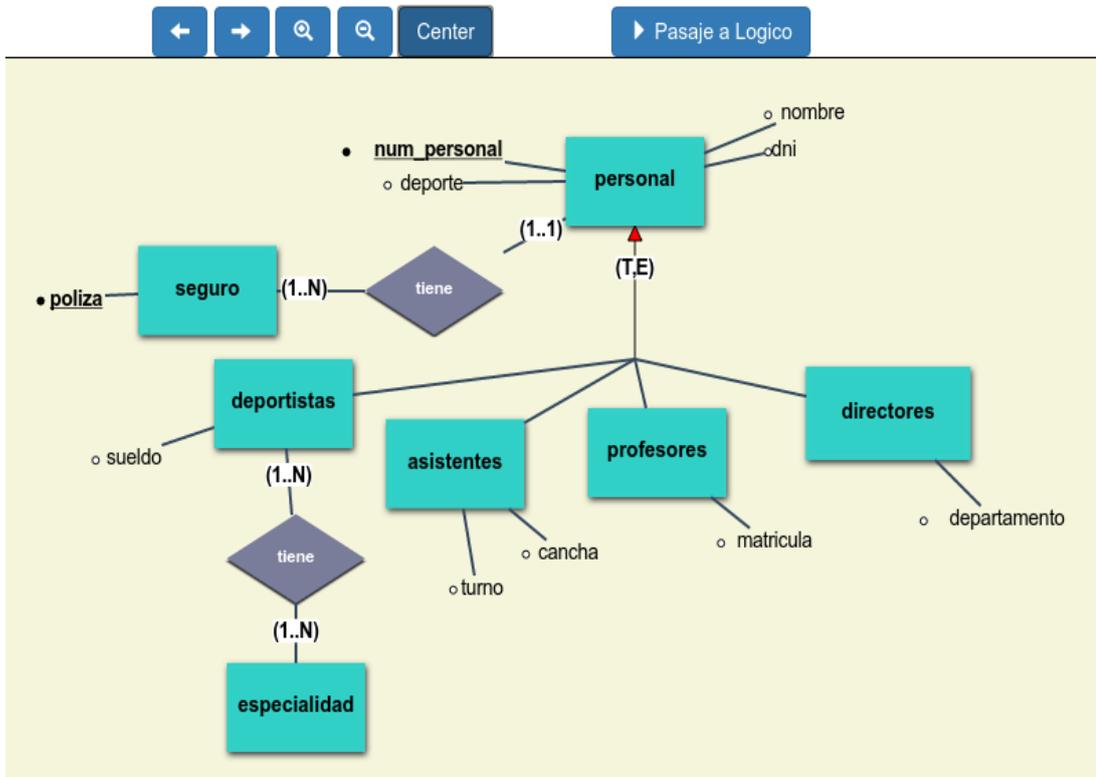


Fig. 28 Herramienta CasER 3.0

Mientras que, los pasos en la nueva versión son más simples y guiados, ya que no es necesario finalizar de manera manual los modelos; sino que al presionar el botón “Pasaje a Lógico”, el modelo se finaliza de manera automática, mostrando un mensaje informativo sobre el modelo. Luego, la herramienta habilita los botones que dan lugar a realizar los pasos para continuar con el pasaje al Modelo Lógico. Lo cual guía de manera natural al usuario a través del proceso.

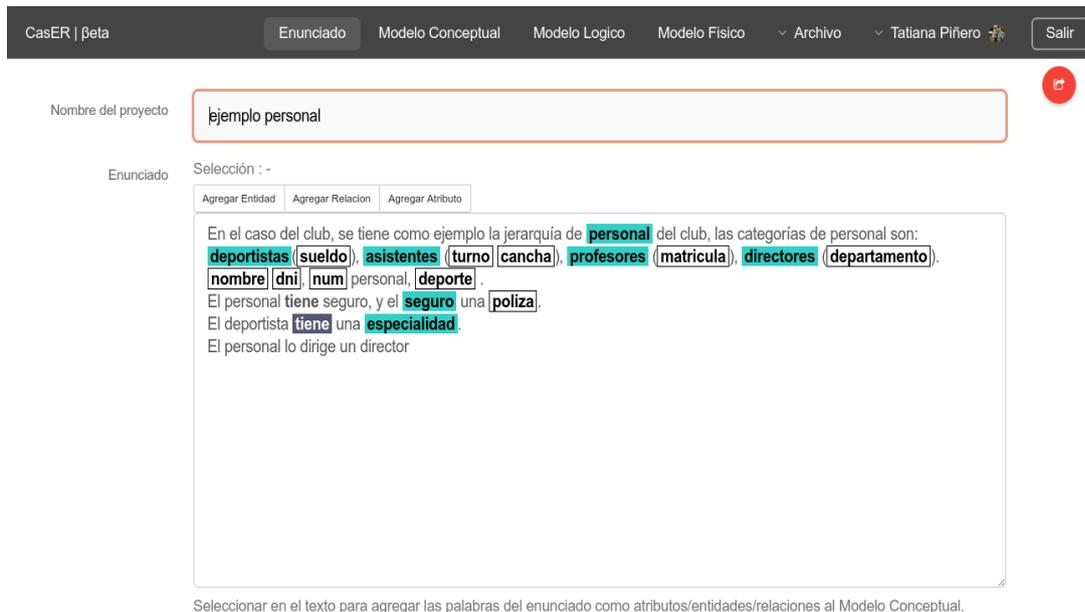


Fig. 29 Ejemplo de mejoras de look&feel.

- ✓ Se pensó en los colores que representa cada entidad. De esta manera, se logró asociar cada uno de los elementos que conforman el diagrama tales como las relaciones, las entidades, y los atributos, con un color propio. Tanto en el texto, como en cada uno de los diagramas se puede diferenciar fácilmente los tipos de elementos del esquema mediante los colores. La Figura 28 ilustra esto mismo en el diagrama, mientras que la Figura 29 ilustra cómo se diferencia en el texto. Esto resulta de gran ayuda al usuario, quien puede asociar rápidamente qué tipo de entidad asoció con cada palabra, y por lo tanto lo hace más legible.

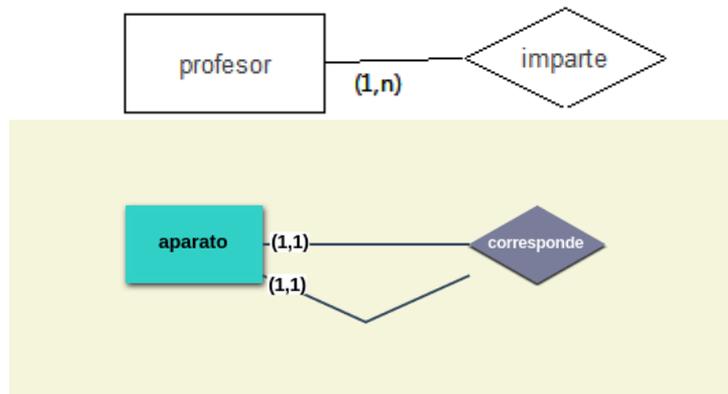


Fig. 30 Comparativa de Relación recursiva en CasER 2.0/3.0

- ✓ En la Figura 30, se realiza una comparativa entre las 2 versiones a la hora de dibujar una relación recursiva. Mientras que en la versión 2.0 está representada con una línea simple, en la versión 3.0 es representada con 2 líneas, lo cual demuestra realmente que hay una relación recursiva y no da lugar a ambigüedades.

Con estos ejemplos se puede dar una idea de cómo se rediseñó la UI, y cuál es el objetivo de dichos rediseños. Se centró en la experiencia de usuario y se hizo hincapié en cuestiones de comportamiento y aprendizaje.

Más adelante se podrá apreciar más en detalle dichas mejoras, a medida que se vaya explicando el uso mismo de la herramienta.

## 6.5 USO DE LA HERRAMIENTA

Esta herramienta permite:

- ✓ Que cada usuario pueda registrarse y crear su perfil de usuario.
- ✓ Abrir un proyecto existente o crear uno desde cero.
- ✓ Redactar la especificación de una base de datos, la cual será diseñada.
- ✓ Diseñar el Modelado Conceptual basándose en la especificación.
- ✓ Realizar el pasaje al Modelo Lógico, con la ayuda de la herramienta.
- ✓ Realizar el pasaje al Modelo Físico, de manera automática y didáctica.
- ✓ Compartir el proyecto completo en cualquier momento.
- ✓ Guardar los modelos diseñados como imágenes.

- ✓ Recopilar estadísticas del uso de la herramienta.

A continuación se explicaran cada una de estas funcionalidades, a fin de explicar cómo es efectivamente el uso de la herramienta de cara al usuario final.

### 6.5.1 REGISTRACION DE USUARIO

En esta nueva versión se tiene como prerrequisito contar con una cuenta de Correo de Google<sup>1</sup>.



Fig. 31 Extracto de la pantalla principal de Caser 3.0

La *Figura 31* ilustra una sección de la pantalla inicial de Caser, donde se encuentra el acceso de los usuarios a través de Google. Presionando sobre el botón rojo que dice “Ingresar con Google”,

---

<sup>1</sup> Para crear una cuenta se puede ingresar en :  
<https://accounts.google.com/SignUp?service=mail&continue=https%3A%2F%2Fmail.google.com%2Fmail%2F&ltmpl=default>

se redirigirá al usuario al sistema de Google, el cual identificará al usuario.



Crear un nueva cuenta

**Nombre**

**Email**

**Universidad**

**Ocupacion**

*Fig. 32 Ejemplo registraci3n de Usuario*

Una vez identificado, se dirige al usuario al formulario de registraci3n, en la cual se le pide que ingrese datos necesarios para relevar qu3 dominio de usuarios es el que utiliza la herramienta. Esto se puede ver ilustrado en la *Figura 32*.

Luego de que el usuario efectúa su registro, ahora s3 ya est3 en condiciones de empezar a utilizar las funcionalidades que la herramienta provee.

## 6.5.2 ABRIR/CREAR UN PROYECTO



Fig. 33 Pantalla de Bienvenida de Usuario

La primera pantalla de bienvenida al usuario, contiene las dos principales acciones con las cual el usuario debe iniciar (como se muestra en la *Figura 33*):

- ✓ Crear un Proyecto: Se creará un archivo en la cuenta de Google Drive del usuario, para luego poder ser editado, compartido o eliminado si lo desea.
- ✓ Abrir un Proyecto: Se podrá editar un proyecto previamente creado por el usuario, o por algún otro usuario que le haya compartido uno de sus proyectos.

## 6.5.3 TRABAJAR CON UN PROYECTO

Cuando se crea un proyecto, el mismo es almacenado en la cuenta de Google Drive del usuario. A partir de ese momento, estará disponible para poder acceder a ese proyecto en cualquier momento que se desee, poder editarlo o hasta incluso compartirlo con otros usuarios.

Si se quiere volver a editar un proyecto creado con anterioridad, basta con realizar la acción de “Abrir”, el cual desplegará una pantalla para poder seleccionar entre los proyectos disponibles.

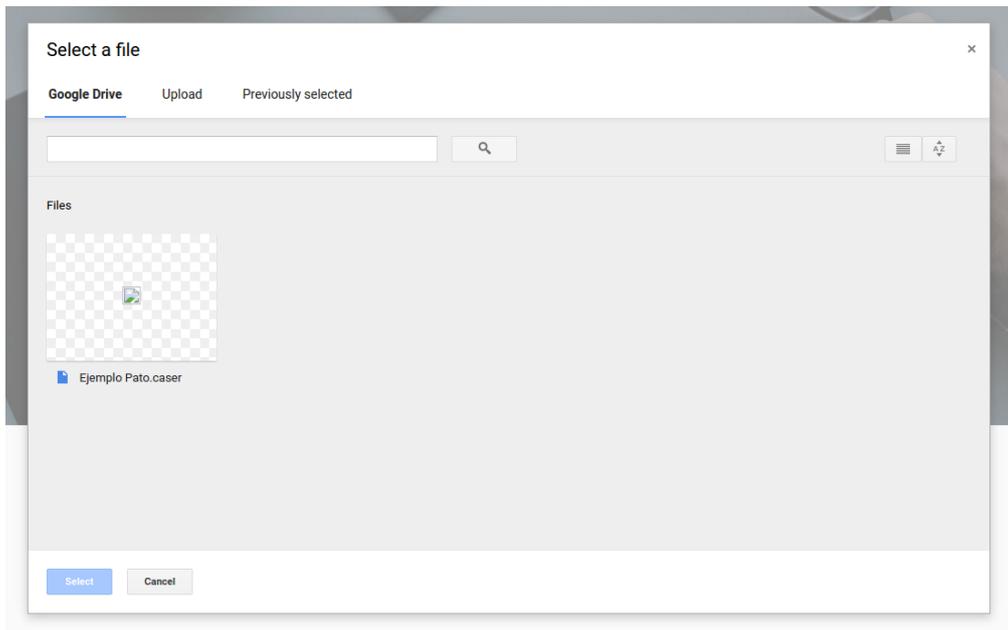


Fig. 34 PopUp para abrir proyectos guardados en Google Drive.

En la Figura 34, se ilustra el modo de seleccionar un proyecto, para luego ingresar en la pantalla de edición de Caser 3.0.

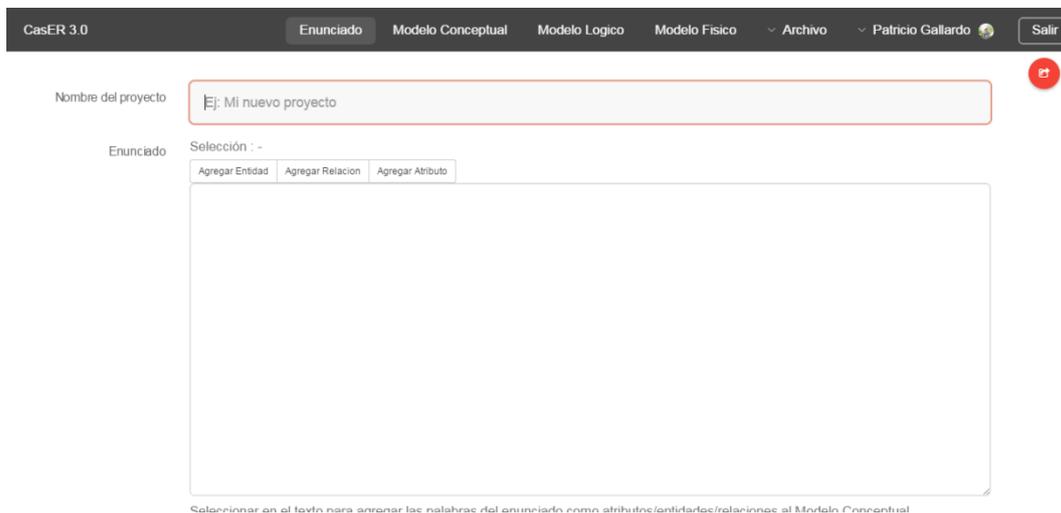


Fig. 35 Edición de proyectos: Sección enunciado.

Una vez creado un proyecto o, en su otra alternativa, seleccionado uno existente, la herramienta despliega la pantalla de edición de proyecto. La misma contiene una serie de secciones (ver barra de menú que se encuentra en la parte superior en la Figura 35): **“Enunciado”**, **“Modelo Conceptual”**, **“Modelo Lógico”** y **“Modelo Físico”**. Para acceder a cada una de ellas, basta con

presionar sobre su nombre.

### SECCIÓN ENUNCIADO

En ésta sección, ilustrada en la Figura 35, se puede editar el nombre del proyecto y el enunciado, el cual servirá como punto de partida para el diseño de la base de datos. El nombre es utilizado como referencia para el usuario, mientras que el enunciado se utilizará para poder tener referencias y poder relacionar el origen de las entidades y las relaciones de los Modelos, poder saber cuál fue el origen de alguna entidad en particular.



*Fig. 36 Sección Enunciado: Referencias de colores*

Como se verá en secciones posteriores, es posible añadir entidades al modelo conceptual seleccionando palabras en el texto del enunciado. Una vez que se agregó la entidad, la palabra se resaltarán de distintas formas dependiendo el tipo de elemento del esquema que se añada. En la Figura 36 se muestran la referencia de colores dependiendo si el tipo es una Entidad, una Relación o un Atributo.

### SECCIÓN MODELO CONCEPTUAL

Esta sección se encuentra muy relacionada con el “Enunciado”, ya que la forma de agregar entidades al modelo, y poder diseñarlo, es a través del texto que se haya ingresado en el paso anterior. Usando los botones que se encuentran sobre el enunciado, el usuario podrá agregar distintos elementos, y podrá completar los datos necesarios para añadirlos al diagrama.

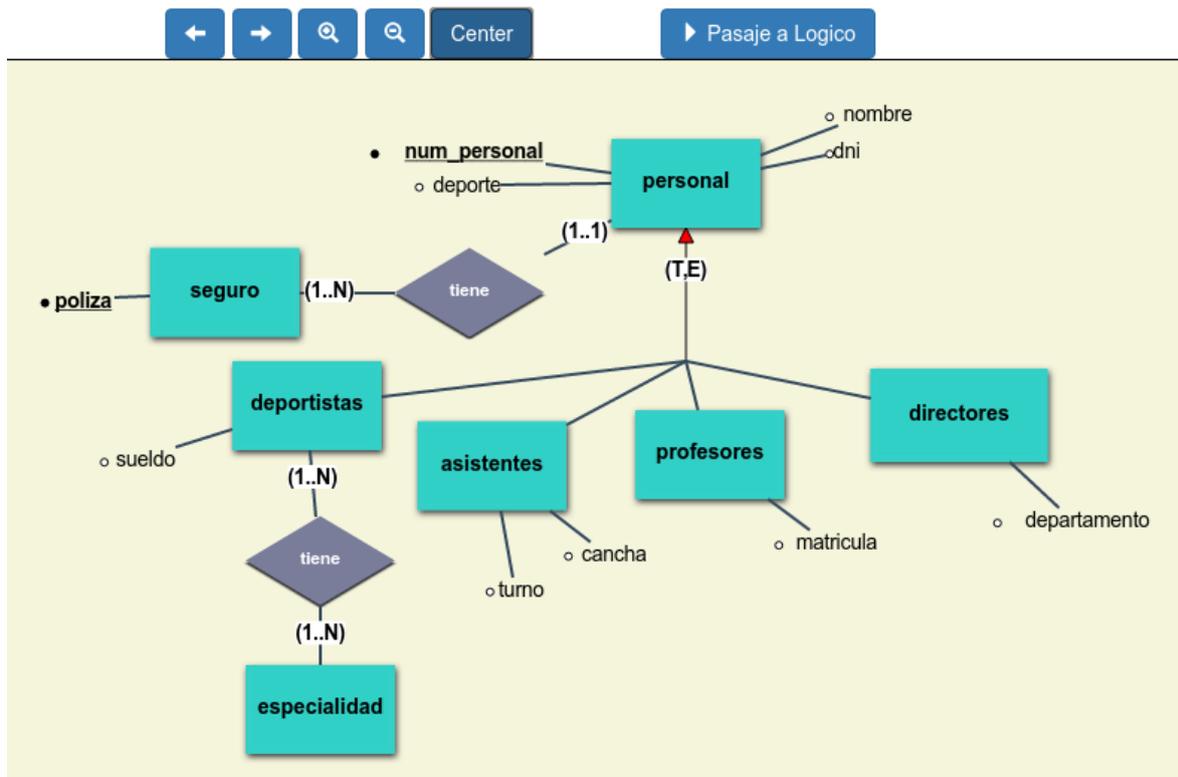


Fig. 37 Sección "Modelo Conceptual"

Como se ilustró en la Figura 36, en la Figura 37 se muestra como se relacionan los colores en el texto con los colores de los elementos en el Modelo Conceptual. Más adelante veremos que esta relación continúa también en el Modelo Lógico.

Por último, en esta sección es en donde se inicia el pasaje al Modelo Lógico, el cuál dispondrá de una serie de pasos para lograrlo, y se basará enteramente en el Modelo Conceptual que el usuario haya diseñado.

### SECCIÓN MODELO LÓGICO

Aquí se diseña y se realiza el pasaje al Modelo Lógico, el cual estará enteramente basado en el Modelo Conceptual previamente diseñado. Esta sección consta de 3 pasos:

1. Eliminación de Atributos Compuestos.
2. Eliminación de Atributos Polivalentes.

### 3. Eliminación de Jerarquías.

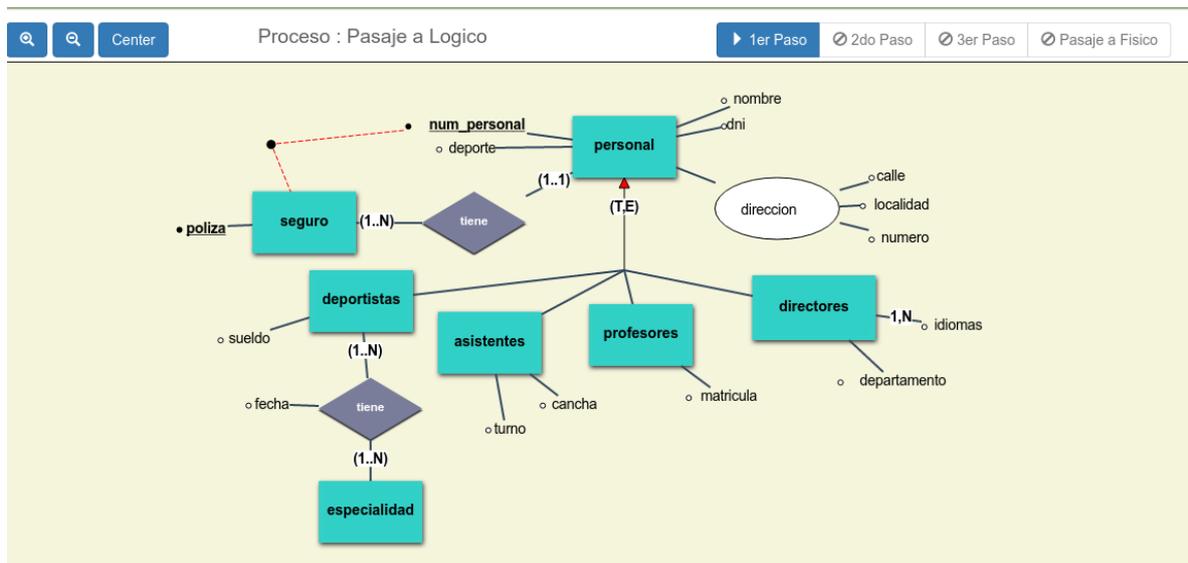


Fig. 38 Sección "Modelo Lógico"

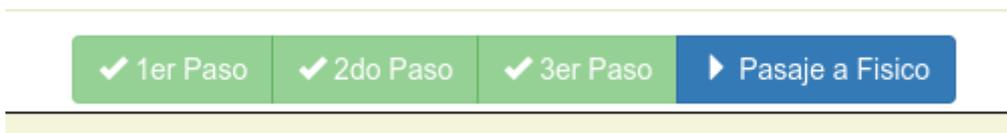


Fig. 39 Pasaje a Físico habilitado

El último paso de esta sección, es el inicio del pasaje al Modelo Físico. Como se ilustra en la Figura 39, luego de realizar los primeros 3 pasos, se habilita el botón titulado "Pasaje a Físico", el cual permite realizar dicha acción.

#### SECCIÓN MODELO FÍSICO

En esta última sección, se encuentra el Modelo Físico, el cual se basa en el diseño previo del Modelo Lógico.

Tablas :

**personal** : [ [personal\\_id](#) | [num\\_personal](#) | deporte | dni | nombre | direccion\_calle\_numero\_localidad | [F\\_seguro\\_id](#) ]

- *Identificadores Compuestos*

- *id\_compuesto* : [ seguro\_id | num\_personal ]

**deportistas** : [ [deportistas\\_id](#) | sueldo | [F\\_personal\\_id](#) ]

**directores** : [ [directores\\_id](#) | departamento | [F\\_personal\\_id](#) ]

**profesores** : [ [profesores\\_id](#) | matricula | [F\\_personal\\_id](#) ]

**asistentes** : [ [asistentes\\_id](#) | turno | cancha | [F\\_personal\\_id](#) ]

**especialidad** : [ [especialidad\\_id](#) ]

**seguro** : [ [seguro\\_id](#) | [poliza](#) ]

**idiomas** : [ [idiomas\\_id](#) | idiomas ]

*Fig. 40 Sección "Modelo Físico"*

En la Figura 40, se puede apreciar un ejemplo de cómo se muestra el Modelo Físico resultante, de todo el proceso previamente descrito, en donde se detalló una especificación, luego se realizó el diseño del Modelo Conceptual, en base al cual se generó, con ayuda de la herramienta, el Modelo Lógico, para finalmente poder generar las tablas físicas que se utilizaran en la base de datos.

#### 6.5.4 AGREGAR/EDITAR COMPONENTES: ENTIDADES, RELACIONES Y ATRIBUTOS

Comenzando con el diseño del Modelo Conceptual, lo primero que se necesita es añadir componentes al Modelo. Para esto mismo se encuentran las acciones de “Agregar Entidad”, “Agregar Atributo” y “Agregar Relación”.



Fig. 41 Opciones para agregar los elementos en el Modelo Conceptual

En la Figura 41, se puede apreciar que existen 2 maneras de hacer esto. Ambas requieren del seleccionado de la palabra que se necesita relación con el nuevo elemento. Luego el usuario puede optar por hacer un “click derecho” sobre la palabra seleccionada, lo cual desplegará un menú con las opciones, o bien, puede elegir entre las mismas opciones que están ubicadas en el margen superior izquierdo del enunciado.

Una vez hecho esto, la herramienta dará la posibilidad al usuario de poder completar los datos necesarios para poder describir el elemento deseado.

Fig. 42 Ejemplo creación de elemento Atributo.

La Figura 42 muestra un ejemplo, en el cual el usuario selecciona “Agregar Atributo” y la herramienta solicita que se le complete la “cardinalidad”, el “dominio” y la “entidad” a la cual pertenece.

A su vez, la herramienta cuenta con una serie de validaciones, con el objetivo de ayudar al usuario a completar de manera correcta el modelo, pero a su vez motivándolo a conocer las reglas teóricas de cómo debería agregar cada tipo de entidad.

The screenshot shows a web interface for editing an entity. The title is "Entidad". There are two input fields at the top: "Texto referencia:" containing "club" and "Alias" containing "club". Below these are three tabs: "Jerarquia" (active), "Atributos", and "Identificadores Compuestos". Under the "Jerarquia" tab, there are two dropdown menus: "Entidad padre con cobertura:" and "Hija de :", both set to "-Seleccione-". At the bottom, there are three buttons: "Eliminar" (red), "Guardar" (green), and "Cancelar" (red).

Fig. 43 Ejemplo edición de un componente Entidad.

Una vez añadido el componente al Modelo Conceptual, se lo podrá editar haciendo “doble click” sobre el mismo. La herramienta volverá a desplegar las mismas opciones que cuando se crea uno nuevo, pero esta vez con los datos cargados del componente en cuestión.

Una vez terminada cada acción, y después de haber guardado los cambios, se verán reflejadas las modificaciones sobre el Modelo Conceptual.

## Edición alternativa de Atributos

Entidad

Texto referencia: Alumno

Alias: Alumno

Jerarquia Atributos Identificadores Compuestos

Nombre	Alias	Tipo	Cardinalidad	Eliminar
numero	num_legajo	Texto	1 1	<input checked="" type="checkbox"/> Identificador
nombre	nombre	Texto	1 1	<input type="checkbox"/> Identificador

Eliminar Guardar Cancelar

Fig. 44 Ejemplo Edición de una Entidad: solapa Atributos

Tanto en la edición de una Entidad, como en la de una Relación, se encuentra la posibilidad de poder editar los atributos de las mismas. Como se puede observar en la Figura 44, se encuentra la sección *Atributos*, en la cual se visualizan los atributos pertenecientes a la componente. En dicha sección, se le da la posibilidad al usuario de modificar algunas cualidades del atributo, así también como la posibilidad de eliminarlo.

### 6.5.5 CREACION DE IDENTIFICADORES COMPUESTOS

Además de poder editar Atributos, dentro de la edición de Entidades, la herramienta provee la posibilidad de crear o editar Identificadores Compuestos, los cuales pueden ser Externos o Internos.

Fig. 45 Edición Entidad: Identificadores Compuestos

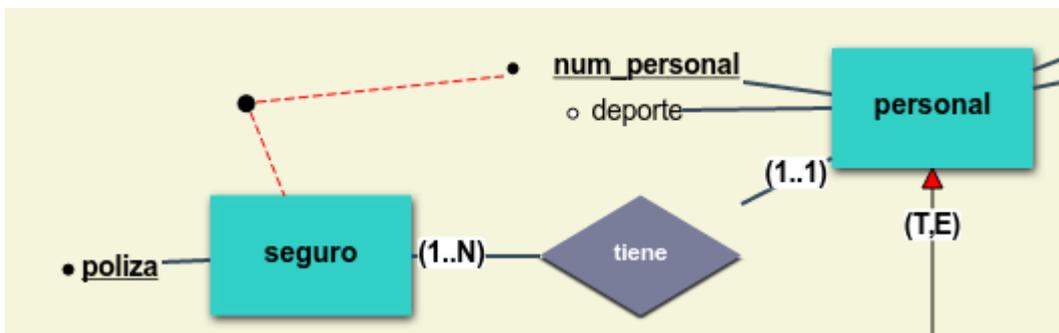


Fig. 46 Identificados compuesto – línea punteada roja-

En la Figura 45 se observa que la entidad “personal” contiene un Identificador Compuesto, el cual se puede editar o eliminar desde esta misma ventana. En el modelo de la Figura 46 se verá reflejado con una línea punteada en color rojo, uniendo los atributos y entidades previamente seleccionados.

#### 6.5.6 VALIDACIONES

Durante todo el proceso de diseño de la base de datos, la herramienta ofrece una serie de

validaciones, con el objetivo de reducir las probabilidades de que el usuario cometa errores. Esto se realiza también, con el objetivo de que el usuario realice un aprendizaje de reglas elementales del diseño.

The screenshot shows the 'Atributo' form in the CasER 3.0 interface. At the top, there are tabs for 'Enunciado' and 'Modelo Conceptual'. The form has several sections:

- Texto referencia :** A text input field containing 'legajo'.
- Alias :** A text input field containing 'legajo'.
- Requerido :** A red icon with an exclamation mark and the text 'Requerido'.
- Dominio :** A dropdown menu with '-Dominio-' selected. To its right are two checkboxes: 'compuesto' (unchecked) and 'Identificador' (checked).
- Entidad :** A dropdown menu with '-Entidad-' selected.

A tooltip message is displayed over the 'Requerido' section, stating: 'Los atributos 'identificadores' no pueden tener 0 como cardinalidad minima.'

Fig. 47 Validación Formulario: Ejemplo atributo

A continuación se detallarán algunos ejemplos de validaciones con las que cuenta CasER 3.0:

✓ Validación de Formularios:

En los distintos formularios de Creación y Edición de Componentes, la herramienta cuenta con validación de los valores, e incluso de la combinación de los mismos. En la Figura 47 se ilustra un ejemplo de combinación de valores, en donde el usuario ingresa cardinalidad mínima igual a 0, y selecciona que el atributo es un identificador; como esto rompe con las reglas de identificadores, la herramienta avisa de esto mismo al usuario.

✓ Validación de Identificadores Compuestos:

El usuario sólo podrá crear identificadores compuestos, siempre y cuando la entidad posea más de 2 atributos o esté relacionada con otra entidad que posea atributos.

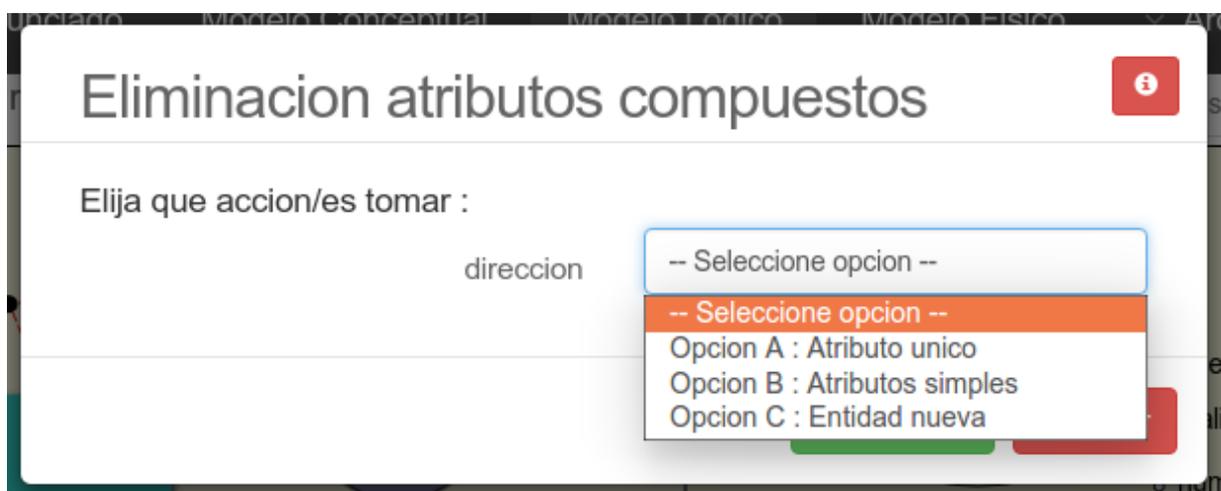
✓ Validación de Integridad:

Durante el proceso de diseño de los Modelos, la herramienta realiza una validación de integridad de los Modelos. La misma está orientada a reducir las inconsistencias propias de las modificaciones que van sufriendo los modelos durante el proceso. Un ejemplo de esta validación, es el caso de cuando se elimina una Entidad, se realiza una validación

para que no queden Relaciones huérfanas (con menos de 2 Entidades), o que no queden Identificadores Externos sin Entidades externas relacionadas.

#### 6.5.7 PASAJES DE MODELOS: SEMIAUTOMATICOS Y AUTOMATICOS

La herramienta Caser 3.0, cuenta con la posibilidad de realizar los pasajes entre modelos de manera Automática y Semiautomática.



*Fig. 48 Pasaje semiautomático: Ejemplo intervención del usuario*

Cuando se habla de la posibilidad de hacer un pasaje entre modelos de manera Semiautomática, significa que si bien la herramienta automatiza ciertos pasos para agilizar la tarea, se requiere de todas maneras que el usuario tome algunas decisiones para completar el pasaje (la Figura 48 muestra un ejemplo de cómo se le da la posibilidad al usuario de tomar una decisión). Este es el caso del pasaje del Modelo Conceptual al Modelo Lógico, en el que se le pregunta al usuario que desea hacer con los Atributos Compuestos, en el cual podría elegir entre: agrupar en un único atributo, dejar cada uno de sus atributos como atributos simples o crear una nueva Entidad representando el Atributo Compuesto.

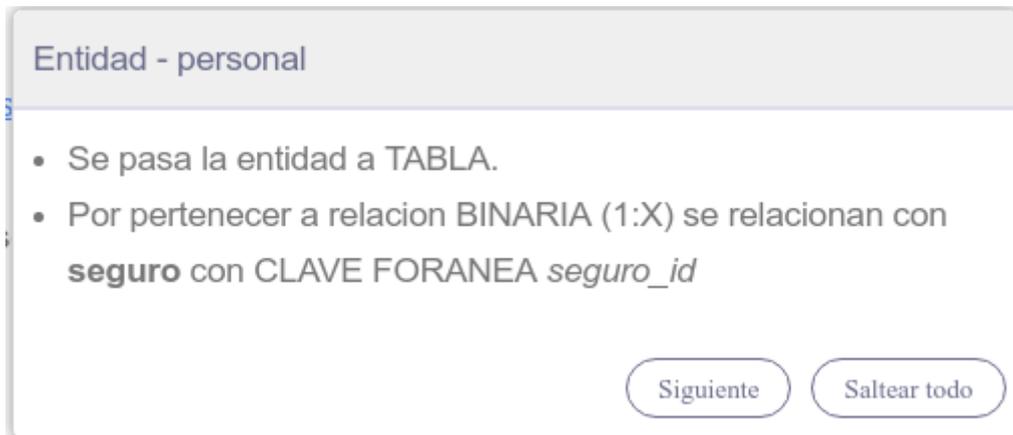


Fig. 49 Pasaje Automático: No existe intervención del usuario, solo se informa las acciones

En cambio cuando se habla del pasaje de manera Automática, básicamente lo que significa es que no requerirá intervención alguna por parte del usuario. La herramienta posee la lógica suficiente para poder tomar las decisiones necesarias para completar con la tarea (la Figura 49 ilustra la manera en que la herramienta realiza las acciones, manteniendo informado al usuario en todo momento). El pasaje del Modelo Lógico al Modelo Físico se realiza de esta manera. En dicho pasaje, en base al Modelo Lógico, la herramienta sabe cómo transformar los distintos casos, a las tablas resultantes. Por ejemplo, tiene implementada la lógica de crear una sola tabla cuando la relación entre 2 entidades tiene una cardinalidad con los pares (1, 1).

#### 6.5.8 EJEMPLO PRÁCTICO

A continuación se expondrá un ejemplo del diseño de una base de datos para el manejo de información del personal de un club, utilizando la herramienta.

Se mostrarán extractos de imágenes de las secciones de CasER en las cuales se trabaja para poder ver con mayor facilidad como se va gestando el diseño de la BD.

Nombre del proyecto **Ejemplo Práctico: Personal (Tesis Caser 3.0)**

Enunciado Selección : "tiene"

Agregar Entidad    Agregar Relacion    Agregar Atributo

En el caso del club, se tiene como ejemplo la jerarquía de **personal** del club, las categorías de personal son: **deportistas** (**sueldo**), **asistentes** (**turno** **cancha**), **profesores** (**matricula**), **directores** (**departamento**), **nombre** **dni**, **num** personal, **deporte** .

El personal **tiene** seguro, y el **seguro** una **poliza**.

El deportista **tiene** una **especialidad**, **fecha** de recibida

El personal lo dirige un director y habla muchos idiomas

El persona tiene una **direccion** compuesta por **calle** **numero** **localidad**

Fig. 50 Ejemplo Práctico: Planteo de la problemática a resolver

En primer lugar se plantea la problemática a resolver como se muestra en la Figura 50, dándose a conocer los detalles del tema en cuestión. Una vez que se tiene toda la información pertinente se puede empezar a diseñar la base de datos. El segundo paso, es empezar a identificar las entidades con sus atributos, y las relaciones que se pueden dar entre ellas, forjándose así el Modelo Conceptual.

Los elementos identificados, entidades, relaciones y atributos, son marcados con un color correspondiente al elemento representado. Esto permite identificar de manera rápida, los atributos que son marcados en el texto, para encontrar la congruencia del planteo y el modelo.

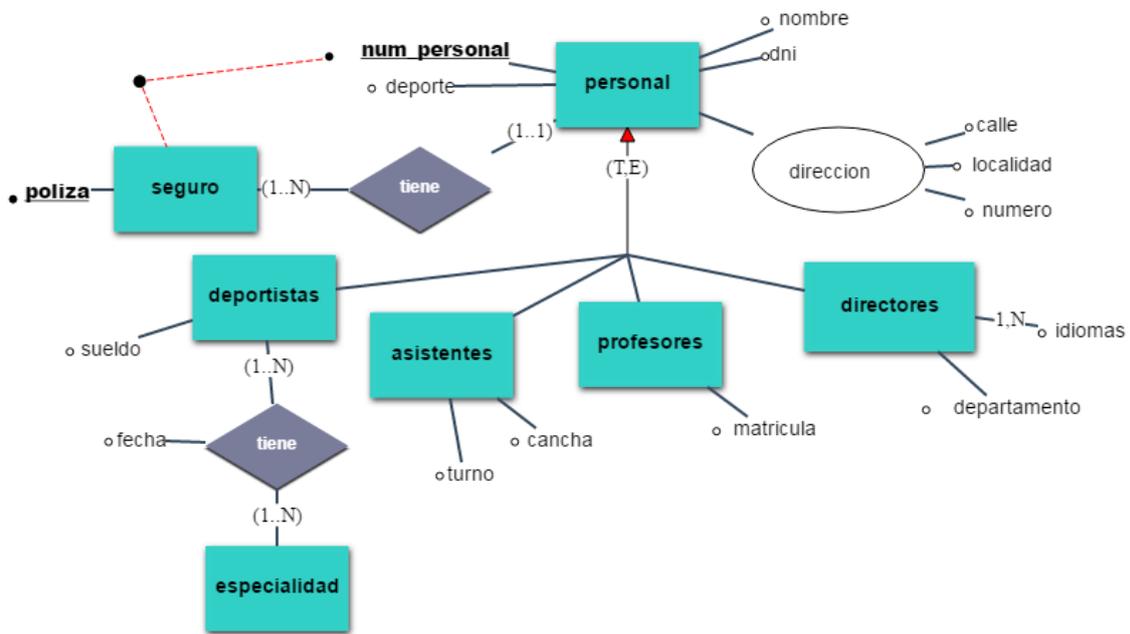


Fig. 51 Ejemplo Práctico: Modelo Conceptual

La Figura 51, muestra el Modelo Conceptual terminado. Una vez culminado el primer modelo del diseño, se puede comenzar con el pasaje al Modelo Lógico, se debe tener en cuenta que una vez adentrado en esta etapa, no se puede volver a modificar ni el enunciado, ni el Modelo Conceptual.

El pasaje de Modelo Conceptual al Lógico, como se explicó con anterioridad, es asistido y semiautomático. Los pasos ejecutados en este pasaje se hacen de manera ordenada, y dando opciones para cada caso.

En este ejemplo en particular, se tienen atributos polivalentes como es el caso de “idiomas”, atributos compuestos, como la “dirección” y una jerarquía a resolver.

Estas cuestiones se irán resolviendo a medida que se avance en el proceso de pasaje de modelos.

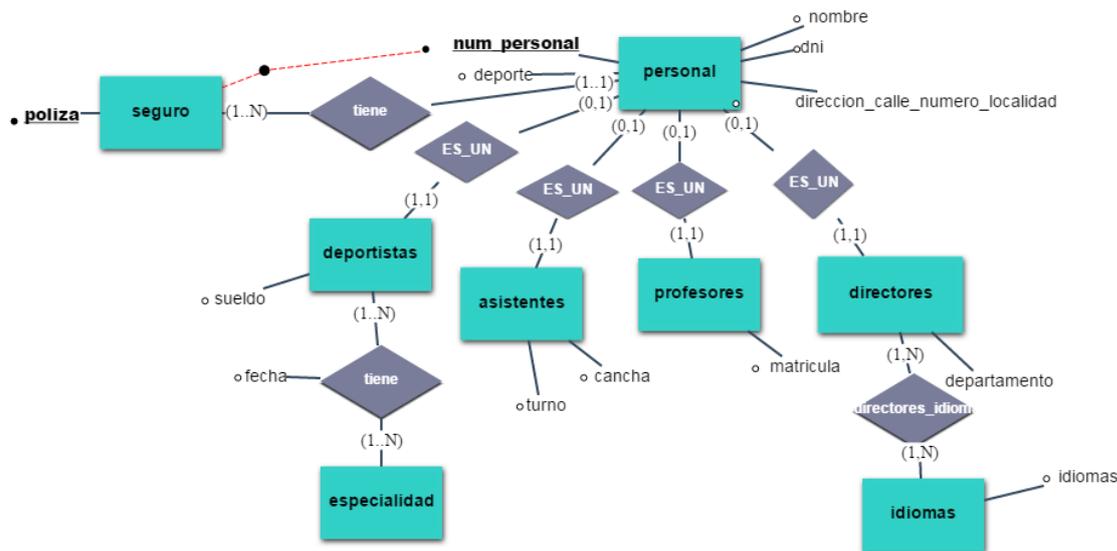


Fig. 52 Modelo Lógico.

La Figura 52 muestra el Modelo Lógico terminado. Los atributos polivalentes se convirtieron en entidades, relacionadas con la entidad dueña del atributo en cuestión. Los atributos compuestos, se unificaron en uno solo. Y la jerarquía se resolvió dejando todos los elementos y forjándose entre ellas la relación “ES\_UN”. Cada una de éstas resoluciones fueron decisión del usuario.

Tablas :

**personal** : [ [personal\\_id](#) | [num\\_personal](#) | [deporte](#) | [dni](#) | [nombre](#) | [direccion\\_calle\\_numero\\_localidad](#) | [F\\_seguro\\_id](#) ]

- *Identificadores Compuestos*
- *id\_compuesto* : [ [seguro\\_id](#) | [num\\_personal](#) ]

**deportistas** : [ [deportistas\\_id](#) | [sueldo](#) | [F\\_personal\\_id](#) ]

**directores** : [ [directores\\_id](#) | [departamento](#) | [F\\_personal\\_id](#) ]

**profesores** : [ [profesores\\_id](#) | [matricula](#) | [F\\_personal\\_id](#) ]

**asistentes** : [ [asistentes\\_id](#) | [turno](#) | [cancha](#) | [F\\_personal\\_id](#) ]

**especialidad** : [ [especialidad\\_id](#) ]

**seguro** : [ [seguro\\_id](#) | [poliza](#) ]

**idiomas** : [ [idiomas\\_id](#) | [idiomas](#) ]

Fig. 53 Modelo Físico

Para culminar con el diseño, el último paso a dar, es la conversión del Modelo Lógico al Modelo Físico. La Figura 53 es un extracto de este modelo como se ve en la herramienta, y la Figura 54 muestra cómo la herramienta muestra el dominio de uno de los atributos llevado a Tabla.

Tablas :



**personal** : [ [personal\\_id](#) | [num\\_personal](#) | [deporte](#) | dni | nombre | direccion\_calle\_numero\_localidad | [F\\_seguro\\_id](#) ]

- **Identificadores Compuestos**
  - [id\\_compuesto](#): [ seguro\_id | num\_personal ]

Fig. 54 Modelo físico – ejemplo atributo deporte de tipo texto.

A priori se expuso que la versión 3.0 de CasER había incorporado el concepto de dominio, los cuales han ido seleccionándose a medida que se creaba un atributo. Estos se ven en el modelo físico, posándose con el mouse sobre el atributo.

#### 6.5.9 FORMAS DE COMPARTIR Y ACCEDER A LOS EJERCICIOS

En esta sección se explorara una de las ventajas de CasER 3.0 como herramienta web.

A priori se mencionó que la misma está integrada con Google Drive<sup>1</sup>, esto quiere decir que los trabajos realizados se pueden compartir con otras personas. Cabe destacar que la persona con quien se comparta dicho trabajo no requiere una cuenta en CasER, con tener una cuenta de Gmail bastará.

Existen 2 formas de compartir los trabajos realizados. Uno, de la misma forma en que se comparten los archivos guardados en Drive. Y la segunda manera es desde la herramienta.

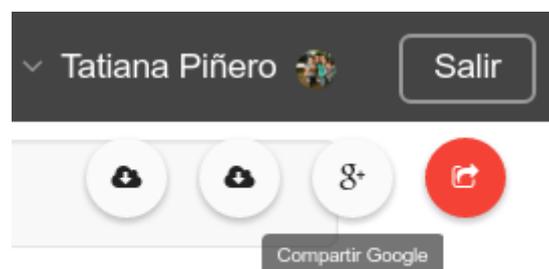


Fig. 55 Compartir un proyecto con otras personas

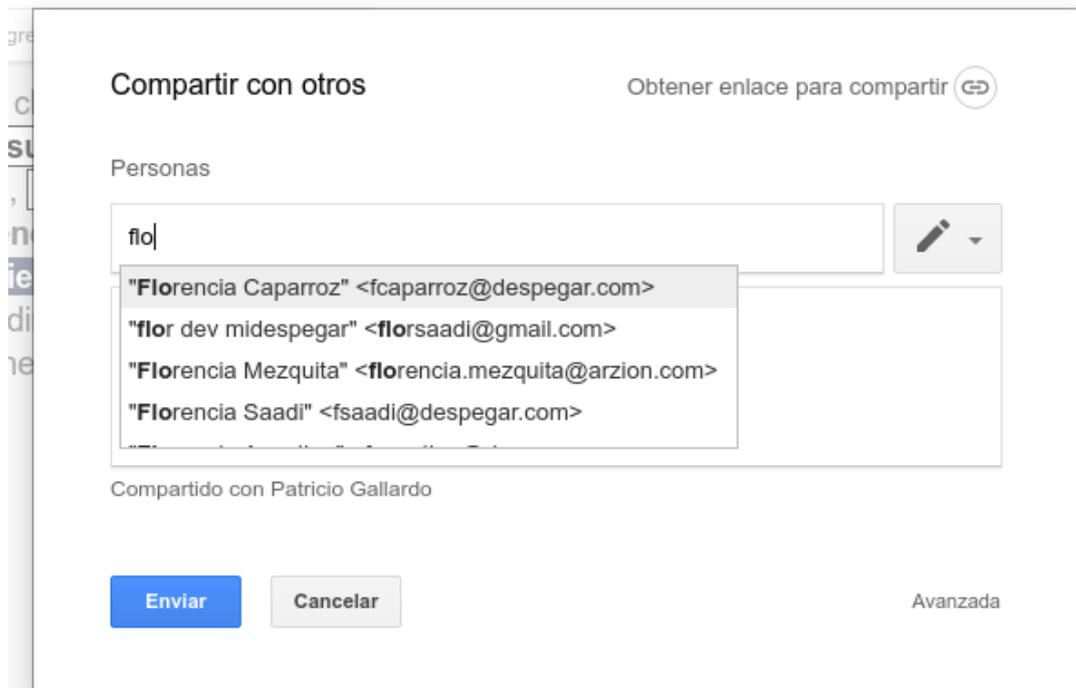


Fig. 56 Ventana emergente para compartir un proyecto

En la Figuras 56 y 57 se observa la manera de compartir un trabajo práctico. Los proyectos cuando se guardan en Google Drive, se generan con una extensión .caser, estos mismos son detectados por la herramienta a la hora de abrirlos.



Fig. 57 Acceder a un proyecto.

Desde la aplicación, como muestra la Figura 57, se despliega una serie de opciones, entre ellas, la de *Abrir*, si se selecciona dicha opción abrirá un PopUp como el de la Figura 34, en donde todos los archivos .caser guardados en Drive, serán accesibles.

Se observa también las opciones:

- ✓ Nuevo
- ✓ Guardar
- ✓ Guardar local
- ✓ Duplicar
- ✓ Eliminar

Nuevo: Se abrirá una nueva pestaña donde podrá crear un proyecto nuevo desde el inicio. Si en el mismo momento en el que el usuario decide crear un proyecto tenía uno abierto en el cual estaba trabajando, no lo perderá; ya que se abrirá una nueva pestaña del navegador donde podrá comenzar con el nuevo diseño, y en la pestaña donde se encontraba situado, quedará el proyecto en el que estaba trabajando al momento de crear uno nuevo.

Guardar: Esta opción antes mencionada, permitirá guardar los proyectos en la cuenta de Google con la que se haya iniciado sesión, donde quedarán a disposición de los usuarios para luego compartirlos, o accederlos. Los proyectos generados se guardan con la extensión .caser

Guardar Local: Se crea una copia local para guardar en el directorio del usuario, también con la extensión .caser.

Duplicar: Esta opción puede ser muy útil, cuando se quiere diseñar una base de datos del mismo problema, pero de diferentes maneras. Lo que hace esta opción, es abrir una nueva pestaña, con el mismo contenido que se tiene en el proyecto abierto. Al momento de guardarlo, se guardará como un proyecto diferente al original del cual se generó la copia.

Eliminar: Como su nombre lo indica, elimina el proyecto. Antes de eliminarlo realmente, se abre un PopUp de confirmación, ya que el proyecto eliminado no podrá ser recuperado, a no ser que se haya guardado una copia local a priori. Esta opción borrará el proyecto del repositorio<sup>6</sup> de google Drive.

### 6.5.10 EXPORTAR UN EJERCICIO

Esta opción es otro de los aportes que tiene la nueva versión de CasER.

Permite extender la posibilidad de compartir el proyecto con las personas que no poseen cuenta de Gmail; esta opción, exporta el modelo lógico o conceptual, a una imagen .png, que podrá ser guardada de manera local, para luego imprimirlo, o enviarlo por correo a cualquier cuenta.

En el momento en que se selecciona la opción de exportar tanto el modelo conceptual como el lógico se abre una pestaña nueva con el modelo, haciendo click derecho sobre la misma, se podrá seleccionar la opción de guardar, de la misma manera en que se guardan el resto de las imágenes de internet.

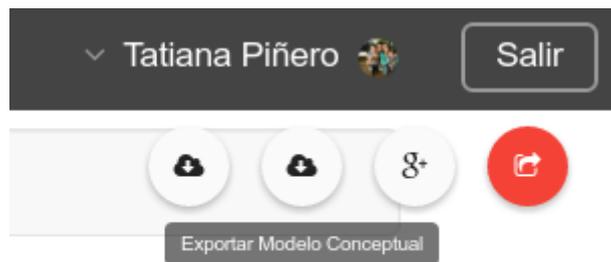


Fig. 58 Exportar un proyecto

## CAPITULO 7: CONCLUSIONES Y TRABAJOS FUTUROS

### 7.1 CONCLUSIONES

En la actualidad las bases de datos constituyen un elemento imprescindible en cualquier empresa u organización. Es por ésto que el diseño de la misma resulta ser un proceso clave y crítico.

En la asignatura Introducción a las Bases de datos, éste es uno de los temas más importantes que se aborda, y es necesario que los alumnos cuenten con un buen sistema de aprendizaje, para poder tener naturalizado en qué consiste el diseño de las bases de datos.

Esta tarea se puede realizar de dos maneras, con papel y lápiz, iterando sobre gráficos que muchas veces crecen de manera exponencial, siendo arduo el proceso y quedando poco claro el modelo logrado, y hasta incluso muchas veces se debe rehacer por completo el diseño.

Otra manera de realizar las prácticas de este tema es utilizando una herramienta que actúe como asistente en esta tarea.

La primera herramienta creada para este propósito, como una tesina de grado por alumnos de la Facultad de Informática de la UNLP, fue CASER 1.0, en el año 2009. Esto constituyó un paso fundamental, a partir de una especificación de requerimientos y marcando palabras claves se puede generar automáticamente el modelo conceptual. Mientras que las tareas de diseño lógico y físico se completaban con el uso tradicional de papel y lápiz. Esto motivó a una segunda versión de la herramienta que permite completar el proceso de diseño en forma asistida, con la cual el Diseño Conceptual, el Diseño Lógico y el Diseño Físico se realiza sin utilizar papel y lápiz y en modo semiautomático.

Dichas versiones cumplieron con su propósito de ayudar al alumno en la asistencia del diseño de una base de datos, pero tenían una limitación muy importante en cuanto a las nuevas modalidades de aprendizaje colaborativo que están surgiendo en la actualidad. CasER 1 y 2, solo se podían utilizar si se instalaba previamente (cuya descarga no se encuentra disponible a través de internet), el cual se ejecutaba solo bajo el sistema Operativo Windows.

Habiendo una necesidad de tener mayor alcance en el acceso a la herramienta, y los diseños realizados en ella desde cualquier punto del mundo, en cualquier dispositivo que tuviera internet,

sin la necesidad de contar con un programa previamente instalado, como así también la posibilidad de aplicar características del aprendizaje colaborativo y a distancia, se pensó así una manera de mejorar esta situación, y fue entonces cuando se desarrolló una tercera versión llamada “CasER 3.0: Modelado de Bases de datos en la nube”.

Ésta nueva versión al igual que la 2.0 asiste al usuario en el proceso de diseño de una bases de datos, teniendo en cuenta el diseño de un Modelo Conceptual, Lógico, y Físico. Pero al tener otro enfoque, en cuanto a su uso, su alcance y otras funcionalidades, fue necesario desarrollar ésta nueva herramienta de manera íntegra, dado que las tecnologías que se usaron en las anteriores son obsoletas y no cumplen con los requisitos para poder extenderla y así poder transformarla de una aplicación de escritorio, a un aplicación web. Por lo cual, se desarrollaron nuevamente y de manera íntegra, todas las funcionalidades existentes y se implementaron nuevas funcionalidades para darle un valor agregado en cuanto a funcionalidad.

Como conclusión se mencionarán solo las mejoras que se han incluido en esta nueva versión y las ventajas que ellas traen:

- ✓ Asistencia al alumno en las distintas fases del modelado de datos brindando un esquema de trabajo ordenado para la transición entre los distintos modelos, de manera más intuitiva y con una interfaz más amigable, aportando comodidad y agilidad a la hora de diseñar una base de datos.
  
- ✓ Cambio en la interfaz gráfica, mejorando de manera notoria el Look&Feel de la herramienta, haciendo más fácil el aprendizaje de su uso.
  
- ✓ Accesibilidad desde cualquier dispositivo móvil que posea internet.
  
- ✓ Actualización transparente para el usuario. No es necesario la descarga y la instalación de nuevas versiones. Las aplicaciones webs se actualizan sin necesidad del que el usuario haga nada al respecto.
  
- ✓ Ésta herramienta está integrada con Google Drive lo cual trae ventajas:
  - En cuanto a la utilización académica, los alumnos podrán entregar trabajos prácticos de manera rápida y cómoda, accediéndolos desde su cuenta de

google, sin la necesidad de tener sus trabajos, guardados en un dispositivo externo.

- Los archivos .caser que se comparten poseen permisos otorgados por su dueño, permisos de edición y/o lectura. Ésto es muy cómodo cuando se desea realizar una corrección sobre dicho diseño, los encargados de realizarlas podrán hacerlo sobre el mismo diseño compartido sin necesidad de generar un nuevo archivo, estos cambios se verán reflejados cuando el dueño del archivo lo actualice. De ésta manera las correcciones son ágiles.
  - Esta ventaja no solo se utiliza en una relación de alumno-profesor, se puede compartir con otros colegas, y obtener otros puntos de vista sobre el trabajo realizado, y se puede continuar generando o haciendo cambios de manera remota, y así diseñar de manera conjunta sin estar en el mismo espacio físico y trabajando sobre el mismo diseño, sobre el mismo archivo.
- ✓ También está integrado con google analytics, lo cual permite obtener estadísticas de su uso. Antes esta funcionalidad no existía, y no se tenía un reporte del uso de la herramienta. Ahora con esta nueva funcionalidad se puede consultar cuantas personas, lo utilizan, si las personas son alumnos, aficionados, o de otra rama y si pertenecen a la UNLP o a otra universidad. Ésto es posible, gracias a que el usuario cuando utiliza por primera vez la herramienta, debe crearse una cuenta completando unos datos que serán almacenados en una base, de la cual se podrá consultar con posterioridad.

Actualmente en el mercado (libre o comercial) no se encuentra una aplicación web educativa con estas características, dado que todos los asistentes disponibles comienzan desde el diseño lógico, y no cuentan con el el diseño conceptual tan necesario para quienes intentan construir una Base de Datos sin experiencia previa, y tan aconsejable para quienes ya tienen experiencia.

## 7.2 TRABAJOS FUTUROS

Es importante remarcar que en la mayoría de los cursos donde se utiliza el producto, SQL es el tema inmediato posterior al modelado, y contar con generación automática de código resultaría de gran ayuda en el proceso de enseñanza del tema. Dicho esto, uno de los trabajos futuros a realizarse puede ser la generación de código SQL de creación de tablas, a partir de las tablas resultantes del modelo físico.

Como se mencionó con anterioridad, es una herramienta web, y como tal, se podría desarrollar un chat interno a la aplicación, donde se puedan discutir temas pertinentes sobre el desarrollo a medida que se va diseñando una base de datos en paralelo.

También queda pendiente la posibilidad de poder editar por más de un usuario a la vez en tiempo real, con control de versiones. Lo cual habilitaría la posibilidad de que un Profesor pueda corregir de manera remota a un Alumno, y este cambio se vea reflejado en tiempo real.

También sería muy útil poder generar “comentarios”, tanto en el texto, como en los diagramas. Ayudaría a la legibilidad de dichos diagramas, y a poder explicar el “por qué” de cada decisión del diseño de una base de datos, quedando todo documentado y en un mismo lugar.

Por último, se podría generar un historial de cambios, que ayude a documentar los pasos realizados.

## BIBLIOGRAFIA

- [1] Diseño Conceptual de Bases de Datos. Autor: Jorge Sánchez . Editorial: openlibra. Año 2004.
- [2] Introducción al Diseño de Bases de Datos. Autor: Dolors Costal Costa. Editorial: UOC PAPERS  
ISBN: 8484299929 9788484299929. Año: 2002
- [3] Bases de datos relacionales y modelado de datos. IFCT0310. Autora: María Yolanda Jiménez Capel. Editorial: ic editorial, ISBN 9788416433308. Año 2014
- [4] Introducción a las Bases de Datos, Fundamentos y Diseño. Autores: Rodolfo Bertone y Pablo Thomas. Editorial Pearson, ISBN 978-987615136-8. Mayo 2011
- [5] “CasER 2.0. Herramienta para el diseño de Bases de Datos.”, Autoras: Durán, Alejandra; Rius, María Florencia. Tesina de Grado 2009, Facultad de informática - UNLP
- [6] Diseño Conceptual de Bases de Datos: un enfoque de entidades-interrelaciones. Autores: Carlo Batini, Shamkant B. Navathe, Stefano Ceri. Editorial Addison Wesley. ISBN: 0201601206 9780201601206. Año 1994
- [7] Procesamiento de Bases de Datos: Fundamentos, diseño e implementación. Autor: David M. Kroenke. Editorial: Prentice-Hall. ISBN: 968-880-696-X. Año: 1996.
- [8] Introducción a los sistemas de bases de datos. Autor: Chris J. Date. Editorial: Pearson Publications Company. ISBN: 97896844444195. Año: 2001.
- [9] Fundamentos de Bases de Datos. Autor: Korth Silberchatz. Editorial: Mc Graw-Hill/Interamericana de España, S.A.U. ISBN: 84-481-8644-1. Año: 2006
-