El Modelamiento (o modelado) de Datos es un proceso previo, pero indispensable, que se debería tener en cuenta al momento de abordar un problema siempre que en éste se hallen involucrados datos y se cuente con las herramientas digitales para manipularlos. El objetivo principal del modelamiento es encontrar una forma de **estructurar la información, ordenarla y finalmente procesarla** en una o varias bases de datos. Una vez hecho el modelado, resulta más "sencillo", en términos generales, analizar la información y manipularla de acuerdo a los objetivos propios de cada institución. Las ventajas de tener la información estructurada y ordenada son evidentes: su visualización será fácil de comprender, la comunicación entre los desarrolladores y los representantes de la institución no presentará muchas brechas, se enriquece en calidad las bases de datos diseñadas, entre otras.

El proceso de Modelamiento de Datos **consta de tres fases** claramente definidas y que se siguen de manera secuencial. Se inicia con el *Modelamiento a nivel Conceptual*, con los resultados de esta primer fase se da paso al *Modelamiento a nivel Lógico* y finalmente se concluye con el *Modelamiento a nivel Físico*. En la siguiente tabla se listan las características de cada una de las fases, pudiendo así visualizar lo que diferencia a cada una de ellas.

Modelamiento a nivel **Conceptual**	Modelamiento a nivel **Lógico**	Modelamiento a nivel **Físico**
Su objetivo es generar un boceto del camino a seguir, es por ello que: • Se buscar identificar las entidades del sistema y empresariales de nivel superior y la relación entre ellas. • La persona encargada es el *Arquitecto de datos*. • Lo que se logra en esta fase es el conocimiento, ojalá completo, del negocio.	Como uno de los objetivos es llevar la información a bases de datos, se definen entonces: • Entidades y relaciones. • Atributos propios. • Las claves primarias. • Las claves foráneas. • Unidades de información. • Las clases. También se busca establecer la *cardinalidad*, esto es, la cantidad de elementos (en términos de proporción) que participan en la relación entre dos o más entidades.	Permite visualizar, con base en la información conseguida en la fase previa, la forma en la que se desea implementar el diseño de la solución. Como consecuencia se consigue una esquematización con: • Tablas y columnas. • Tipos de datos. • Visualizaciones. • Restricciones. • Los demás procesos que se integran en una base de datos.
Se tiene entonces una especie de plan de acción, éste es el insumo que recibe la siguiente fase.	Se obtiene una visión más general, con un mayor contexto y un mayor nivel de detalle del diseño que se está creando.	Se consigue un resultado final, que no es más que el modelo de la implementación de la base de datos.

¿Qué son las *DDL, DML, DCL* y *TCL*?

Recordemos que SQL (**S**tructured **Q**uery **L**anguaje) es una herramienta organizar, manejar y recuperar información almacenada por una base de datos computacional. Como lenguaje computacional -no lenguaje de programación- es utilizado para interactuar con bases de datos, pero no con cualquiera si no con aquellas que reciben el nombre de *bases de datos relacionales*.

La interacción por medio de SQL puede ser entendida al considerar que en éste las instrucciones, el término preciso es **sentencias,** están divididas en 4 categorías:

1. **DDL** (**D**ata **D**efinition **L**anguaje): Como su nombre lo indica, Lenguaje de Definición de Datos, corresponde a la categoría de sentencias que son utilizadas para definir el esquema de la base de datos, sirven para crear, modificar y eliminar la estructura de los objetos de la base de datos, pero no la información. Por lo general los usuarios finales (usuario común) no utilizan las sentencias de esta categoría, que está compuesta por:
    - **CREATE**: Esta instrucción es utilizada para ***crear*** ya sea la base de datos o sus objetos, estos últimos pueden ser tablas, índices, funciones, vistas, entre otros.
    - **DROP**: Con ella se pueden ***eliminar*** objetos de la base de datos.
    - **ALTER**: Es empleada para ***alterar*** la estructura de la base de datos.
    - **TRUNCATE**: Sirve para ***remover*** todos los registros de una tabla, incluyendo todos los espacios para los registros son removidos.
    - **COMMENT**: Esta instrucción es útil para ***agregar comentarios*** al diccionario de datos.
    - **RENAME**: Como su nombre lo sugiere, es empleada para *renombrar* un objeto existente en la base de datos.

2. **DML** (**D**ata **M**anipulation **L**anguaje): En esta categoría, cuyo nombre se traduce como Lenguaje de Manipulación de Datos, se encuentran las instrucciones que permiten la manipulación de la información en la base de datos, es decir, permiten modificar la información almacenada en la base de datos. Esta categoría está compuesta por:
    - **SELECT**: Su utilidad es la de ***recuperar*** información de la base de datos.
    - **INSERT**: Es empleada para ***insertar*** información a una tabla.

- **UPDATE**: Se utiliza para ***actualizar*** la información existente en una tabla.
- **DELETE**: Con ella se pueden ***eliminar*** registros de una tabla de la base de datos.
- **MERGE**: Gracias a esta instrucción se pueden ***ejecutar al mismo tiempo*** las sentencias *INSERT,UPDATE* y *DELETE*. Esta sentencia suple una posible "falencia" de las tres sentencias recién mencionadas, la de tener que utilizar a cada una de ellas de manera separada.
- **LOCK TABLE**: Por medio de ella se puede determinar la *simultaneidad* de una tabla.
- **CALL**: Se emplea para ***llamar*** (ejecutar) un subprograma, ya sea uno PL/SQL o uno JAVA.
- **EXPLAIN PLAIN**: Utilizada par describir el c***amino de acceso*** a la información.

3. **DCL** (**D**ata **C**ontrol **L**anguaje): El objetivo de las sentencias agrupadas en esta categoría es el de determinar los permisos y acceso a los objetos. Las sentencias que pertenecen a la categoría DCL son:
    - **GRANT**: Con ella se le ***otorga*** privilegios de acceso a la base de datos.
    - **REVOKE**: Esta sentencia deshace lo de la anterior, es decir, ***revoca*** los privilegios de acceso otorgados al usuario.
4. **TCL** (**T**ransaction **C**ontrol **L**anguage). Como una última categoría tenemos a aquella que agrupa a las sentencias que son usadas para manejar transacciones en la base de datos. Estas sentencias son empleadas para manejar los cambios hechos por las sentencias DML y además permiten que éstas sean agrupadas, indicándole así a la base de datos cuándo se deben ejecutar todas las operaciones y cuándo ninguna.
    - **COMMIT**: Esta sentencia se emplea para ***grabar*** (guardar) todos los cambios (cualquier transacción) en la base de datos.
    - **ROLLBACK**: Sentencia que permite ***deshacer*** los cambios de toda la transacción.
    - **SAVEPOINT**: Con esta sentencia se puede ***grabar temporalmente*** una transacción con el objetivo y en caso de ser necesario, de podernos devolver hasta este punto, esto para evitar tener que rehacer todo desde cero.

¿Qué son los requerimientos de software?

Los requerimientos de software, también conocidos como *requisitos de software*, es un conjunto de aspectos que se deben de tener en cuenta al momento de modelar e implementar el sistema que se espera desarrollar. Por la generalidad que enmarca suele ser pertinente hacer alguna distinción, clasificar lo que se quiere, todo con el objetivo de verificar que no se ha descuidado algún aspecto relevante o en su defecto, de no ser redundante en otro.

Los requisitos son identificados, por lo menos su mayoría, en la etapa de desarrollo conocida como *especificación de requisitos de* software, que junto con los casos de uso, permiten tener una imagen global de lo que se espera del sistema a desarrollar. Los requerimientos pueden ser clasificados como sigue:

- **Requisitos de usuarios**: Son aquellas necesidades que los usuarios expresan de manera verbal. Por la forma en la que el analista las conoce, resulta evidente el cuidado que se debe tener al momento de levantar estos requisitos, siempre tratando de evitar un posible malentendido del lenguaje.
- **Requisitos del sistema:** Una vez conocidas las necesidades del usuario y con una idea general de lo que se desea hacer, los requisitos del sistema no son más que las componentes (hardware o software adicional) que el sistema deberá poseer para llevar a cabo las tareas esperadas.
- **Requisitos funcionales**: Aquí están encasillados todos los servicios que el sistema deberá proporcionar al finalizar cada una de sus ejecuciones. Su nombre es bastante sugerente, las funcionalidades que debe hacer el sistema.
- **Requisitos no funcionales**: Aparte de que el sistema deberá efectuar algunas tareas (para las que fue diseñado en un principio), también deberá hacer otras tareas que no pueden ni deben ser descuidadas, éstas hacen referencia a lo que se conoce como *requisito no funcional.* Los requisitos no funcionales son aquellas partes del software que pueden no estar relacionadas directamente con las actuaciones del usuario o con su funcionamiento, pero que son importantes pues ellas permiten alcanzar atributos como rendimiento, seguridad, portabilidad, entre otros.

¿Qué son los riesgos de software?

Se entiende por riesgo a todo evento o condición incierta que, una vez ocurrida, produce un efecto (por lo general negativo) en por lo menos uno de los objetivos del proyecto. Es así que podemos considerar que un riesgo es un *problema potencial*, esto con la idea de darle la importancia que debería tener y para ello es vital:

1. Poder identificarlo.
2. Evaluar su probabilidad de aparición o en su defecto, determinar una medida que ayude a dimensionar qué tan factible es de que el riesgo ocurra.
3. Intentar hacer una estimación de su impacto en el proyecto.
4. Finalmente, pero no menos importante, establecer como mínimo un plan de contingencia para el escenario en el que el riesgo sí ocurre.

Puesto que en un proyecto de software son varios los agentes involucrados, cada uno de ellos podría, y en efecto lo hace, aportar cierto riesgo a la ejecución del proyecto, de aquí que una clasificación de los riesgos en torno a sus participantes o agentes involucrados sea adecuada.

- **Riesgos asociados con los usuarios:** Quizás el más destacable en esta categoría sea la posible falta de compromiso por parte de la organización (la gerencia o personas encargadas), de la mano de la poca participación de los usuarios. Salvo en algunas excepciones, estos factores de riesgo están fuera del alcance del líder del proyecto.
- **Factores de riesgo asociados al líder del proyecto:** Aquí se puede mencionar el más relevante, que es la posible incapacidad del líder para juzgar el alcance del sistema. Otro es el escenario en el que el líder realice una identificación pobre de la funcionalidad requerida. Por su relación directa con ambos riesgos, el líder del proyecto debería ser capaz de controlar estos riesgos.
- **Factores de riesgo asociados a la ejecución del proyecto:** En esta categoría deben ser incluidos:
    - La posible presencia de personal inadecuado, aquellos con poco o nulo conocimiento del tema, por ejemplo.
    - La elección de una metodología de desarrollo no apropiada para abordar el problema.
    - Errores al momento de haber efectuado la definición de roles y responsabilidades.

        - Una pobre planeación y control del proyecto.
    - Al igual que ocurre en los riegos asociados al líder del proyecto, es éste quien debería considerarlos y más aún, controlarlos.
- **Factores de riesgo asociados al entorno:** Por lo general aquí se encuentran todos los riesgos que serían consecuencia directa de un posible cambio en la gerencia organizacional.

Vale la pena mencionar que son los factores de riesgos asociados al líder junto con los asociados a la ejecución, los que afectan de manera directa el hecho de que los proyectos sean o no culminados en el tiempo y con el presupuesto previsto.

Bibliografía

1. Para entender mejor la definición de modelado y los diferentes tipos de modelo que existen, se acudió a las siguientes fuentes de información:

    - Anónimo. (2022, 8 abril). *¿Qué es el modelado de datos? Concepto, beneficios y tipos*. Ceupe. Recuperado 22 de julio de 2022, de https://www.ceupe.com/blog/modelado-de-datos.html?dt=1658351221547

    - Araneda, P. (2021, 31 marzo). *Capítulo 4 Modelamiento de Datos | Base de Datos*. www.bookdown.org. Recuperado 22 de julio de 2022, de https://bookdown.org/paranedagarcia/database/modelamiento-de-datos.html

2. Para complementar los tópicos de SQL, se buscó apoyo en las siguientes fuentes:

    - GeeksforGeeks. (2021b, 30 septiembre). *SQL | DDL, DQL, DML, DCL and TCL Commands*. Recuperado 23 de julio de 2022, de https://www.geeksforgeeks.org/sql-ddl-dql-dml-dcl-tcl-commands/

    - GeeksforGeeks. (2021a, septiembre 9). *MERGE Statement in SQL Explained*. Recuperado 23 de julio de 2022, de https://www.geeksforgeeks.org/merge-statement-sql-explained/

    - *SQL Server MERGE: The Essential Guide to MERGE Statement*. (2020, 11 abril). SQL Server Tutorial. Recuperado 23 de julio de 2022, de https://www.sqlservertutorial.net/sql-server-basics/sql-server-merge/

    - Groff, J. P. (2022). *SQL THE COMPLETE REFERENCE* (3.ª ed.). TMH.

3. Lo referente a requisitos de software fue respaldado por las siguientes fuentes:

    - Wikipedia contributors. (2022, 3 julio). *Software requirements specification*. Wikipedia. Recuperado 23 de julio de 2022, de https://en.wikipedia.org/wiki/Software_requirements_specification

- colaboradores de Wikipedia. (2021, 8 octubre). *Especificación de requisitos de software*. Wikipedia, la enciclopedia libre. Recuperado 23 de julio de 2022, de https://es.wikipedia.org/wiki/Especificaci%C3%B3n_de_requisitos_de_software

4. Para concluir se buscó apoyo en las siguientes fuentes, esto con la intención de entender todo lo relacionado con riesgos de un proyecto de desarrollo de software:

- Villarreal, C. (2022, 19 julio). *7 errores comunes en Proyectos de Desarrollo de Software*. Northware. Recuperado 23 de julio de 2022, de https://www.northware.mx/blog/7-errores-comunes-en-proyectos-de-desarrollo-de-software/#:%7E:text=%C2%BFQu%C3%A9%20es%20un%20riesgo%20en,de%20los%20objetivos%20del%20proyecto

- *Riesgo del software*. (s. f.). Ingeniería de Software. Recuperado 23 de julio de 2022, de https://ingenieriasoft.webcindario.com/gestion-y-planificacion-de-proyectos/planificacion-de-proyectos-de-software/riesgo-del-software.html