



**UNIVERSIDAD TÉCNICA DE BABAHOYO
FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICA**

PROCESO DE TITULACIÓN

MAYO 2023 - SEPTIEMBRE 2023

EXAMEN COMPLEXIVO DE GRADO O DE FIN DE CARRERA

PRUEBA PRÁCTICA

INGENIERÍA EN SISTEMAS

PREVIO A LA OBTENCION DEL TITULO DE INGENIERO EN SISTEMAS

TEMA:

Estudio Comparativo entre SQL Directo y un ORM

EGRESADO:

Macias Carpio Fersen Arturo

TUTOR:

Ing. Ruiz Parrales Ivan Ruben

AÑO: 2023

RESUMEN

Al Día de hoy Toda aplicación necesita una Bases de Datos para funcionar y en ella poder guardar la información de sus clientes, productos, contabilidad y todo dato para su correcto funcionamiento. Por lo general la lógica de Negocios de una aplicación se Conecta a la fuente de Datos para realizar operaciones de lectura-Escritura y para esto siempre se ha venido utilizando el lenguaje SQL ya que era el único camino disponible hasta la aparición de los ORM , a partir de este punto la Decisión de utilizar un método u otro se ha convertido en un tema de gran interés en el ámbito de desarrollo de software ya que Ambos enfoques tienen como objetivo acceder a las bases de datos, pero difieren en su implementación y en la percepción de los desarrolladores.

SQL, que significa Structured Query Language, es un lenguaje de programación utilizado para gestionar bases de datos relacionales. Permite a los desarrolladores escribir consultas directamente en la base de datos, lo que les brinda un control detallado sobre cómo se accede y manipula la información. SQL es ampliamente aceptado y utilizado, especialmente en aplicaciones empresariales.

Por otro lado, los ORM son herramientas que mapean objetos de un lenguaje de programación a tablas en una base de datos relacional. Estos facilitan la interacción con la base de datos al ocultar la complejidad de las consultas SQL. Algunos ejemplos populares de ORM son Hibernate para Java y Entity Framework para .NET.

El nivel de aceptación entre los desarrolladores varía entre SQL y ORM. Los desarrolladores con experiencia en bases de datos suelen preferir SQL, ya que les permite un control completo y una optimización fina de las consultas. Sin embargo, los ORM son más populares entre los desarrolladores que desean una forma más sencilla y orientada a objetos de trabajar con bases

de datos. Los ORM también son útiles en equipos multidisciplinarios donde no todos los miembros son expertos en SQL.

En cuanto a la popularidad, SQL sigue siendo la elección predominante en el desarrollo de bases de datos, especialmente en aplicaciones de misión crítica. Los ORM son populares en aplicaciones web y proyectos donde la velocidad de desarrollo es prioritaria.

El Estudio Comparativo entre SQL Directo y un ORM permitirá tomar una mejor decisión de que tecnología utilizar para acceder a la fuente de datos dependiendo de las necesidades específicas del proyecto, los requisitos de rendimiento, la portabilidad, la mantenibilidad y otros factores relevantes.

Palabras clave: ORM, Mapeo Objeto-Relacional, SQL, Base de Datos Relacional

SUMMARY

Today, every application requires a database to function and store information about its customers, products, accounting, and other data necessary for its proper operation. Typically, the business logic of an application connects to the data source to perform read-write operations, and traditionally, SQL (Structured Query Language) has been used for this purpose as it was the only available option until the emergence of ORM (Object-Relational Mapping).

From this point on, the decision to use one method over the other has become a topic of great interest in the software development field, as both approaches aim to access databases but differ in their implementation and developers' perceptions.

SQL, which stands for Structured Query Language, is a programming language used to manage relational databases. It allows developers to write queries directly to the database, providing them with detailed control over how information is accessed and manipulated. SQL is widely accepted and used, especially in enterprise applications.

On the other hand, ORM tools map objects from a programming language to tables in a relational database. They simplify interaction with the database by abstracting the complexity of SQL queries. Some popular examples of ORM include Hibernate for Java and Entity Framework for .NET.

The level of acceptance among developers varies between SQL and ORM. Developers with database experience often prefer SQL, as it offers complete control and fine query optimization. However, ORM is more popular among developers who want a simpler, object-oriented way of working with databases. ORM is also useful in multidisciplinary teams where not everyone is an SQL expert.

In terms of popularity, SQL remains the predominant choice in database development, especially in mission-critical applications. ORM is popular in web applications and projects where development speed is a priority.

A comparative study between Direct SQL and ORM will help make a better decision on which technology to use to access the data source, depending on the specific project requirements, performance needs, portability, maintainability, and other relevant factors.

Keywords: ORM, Object-Relational Mapping, SQL, Relational Database.

INTRODUCCIÓN

En la era digital actual, el desarrollo de software se ha convertido en un componente esencial de prácticamente todos los sectores y empresas. Una de las tareas más cruciales en el desarrollo de aplicaciones es la gestión de bases de datos, ya que los datos son la columna vertebral de muchas aplicaciones modernas. En este contexto existen dos enfoques ampliamente utilizados para interactuar con las bases de datos por una parte tenemos el uso de SQL y la otra forma es el empleo de un ORM.

Como Objetivo Principal de la investigación se busca explorar y analizar las ventajas y desventajas de ambas aproximaciones, con el fin de proporcionar una comprensión más clara sobre cuándo es más apropiado emplear un método u otro en el desarrollo de aplicaciones que necesiten interactuar con una base de datos relacional.

El presente Estudio Comparativo entre SQL directo y un ORM se desarrolló siguiendo la línea de investigación relacionada con los Sistemas de información y comunicación, emprendimiento e innovación y la sub línea de investigación con las redes y tecnologías inteligentes de software y hardware.

La Metodología empleada para el caso de estudio es de carácter Descriptivo ya que nos ayudara a recopilar información de diferentes fuentes como libros y sitios web para posteriormente procesarlo y poder presentar los resultados obtenidos, Como técnica de Investigación se empleó la Entrevista en forma virtual a un grupo de Desarrolladores de Software para Conocer sus puntos de vistas dentro de un entorno Empresarial.

En última instancia, la pregunta fundamental que guía esta investigación es la siguiente ¿Cuándo es más adecuado utilizar SQL directo en comparación con un ORM en el desarrollo de aplicaciones que interactúan con bases de datos relacionales? A través de un análisis

profundo y comparativo, esta investigación buscará proporcionar una respuesta basada en evidencia a esta cuestión crítica para el desarrollo de software.

DESARROLLO

La elección entre utilizar SQL directo o un ORM en el desarrollo de aplicaciones de base de datos es una decisión crucial que muchos desarrolladores se cuestionan al iniciar un nuevo proyecto de desarrollo de Software ya que afectara tanto a la eficiencia como a la productividad del mismo. SQL directo implica escribir consultas SQL en su forma nativa, mientras que un ORM proporciona una capa de abstracción que permite interactuar con la base de datos utilizando objetos y lenguajes de programación de alto nivel. Este estudio tiene como objetivo comparar estas dos aproximaciones y determinar cuándo es más apropiado utilizar una u otra.

Base de Datos

De acuerdo con (Oracle, 2023) afirma que “Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático”

Una vez analizada la definición se puede decir que una base de datos es una colección organizada y estructurada de información o datos que se almacenan de manera electrónica en un sistema de computadoras. Estos datos pueden incluir una amplia variedad de tipos de información, como números de teléfono, nombres de clientes, registros de ventas, imágenes, texto, fechas y mucho más. Las bases de datos están diseñadas para almacenar, gestionar, recuperar y manipular datos de manera eficiente y segura.

Sistema Manejador de Bases de Datos

Un Sistema Manejador de Bases de Datos (DBMS por sus siglas en inglés, Database Management System) es un software diseñado para gestionar y administrar bases de datos. Su función principal es proporcionar una interfaz entre los usuarios o las aplicaciones y los datos almacenados en la base de datos. Los DBMS permiten crear, manipular, acceder y mantener bases de datos de manera eficiente y segura .

Un sistema de gestión de bases de datos no solo permite administrar la información y mantener su consistencia, sino que brinda la posibilidad de construir y definir la base de datos a diferentes niveles de abstracción para distintas aplicaciones, facilitando los procesos y su mantenimiento (Europea, 2022)

Base de datos relacional

Según (IBM, 2023) Una base de datos relacional organiza los datos en filas y columnas, que en conjunto forman una tabla. Los datos normalmente se estructuran en varias tablas, que se pueden unir a través de una clave principal o una clave externa.

Y de acuerdo a la definición de (Google, 2023) Una base de datos relacional (RDB) es una forma de estructurar información en tablas, filas y columnas. Un RDB tiene la capacidad de establecer vínculos (o relaciones) entre información mediante la unión de tablas, lo que facilita la comprensión y la obtención de estadísticas sobre la relación entre varios datos.

Podemos decir que una base de datos relacional es un tipo de sistema de gestión de bases de datos (DBMS) que se utiliza para almacenar y administrar datos. Está diseñada en torno al concepto de tablas relacionadas, que son conjuntos de datos organizados en filas y columnas. Estas tablas se utilizan para representar entidades y sus relaciones en un sistema.

Modelo de datos Relacional

Un modelo de datos relacional es un enfoque para organizar y estructurar datos en una base de datos. En este modelo, los datos se representan como tablas, donde cada tabla contiene información sobre una entidad específica (como clientes, productos o pedidos) y las relaciones entre estas entidades, los autores (Oppel & Sheldon, 2016) indican que el modelo relacional, el cual se basa principalmente en los principios matemáticos de la teoría de conjuntos y lógica de predicados, apoya la recuperación de datos sencilla, aplica la integración de datos (la precisión

y coherencia de los datos), y proporciona una estructura de base de datos independiente de las aplicaciones al acceder a los datos almacenados

Lenguaje SQL

El lenguaje de consulta estructurada (SQL) es un lenguaje de programación para almacenar y procesar información en una base de datos relacional. Una base de datos relacional almacena información en forma de tabla, con filas y columnas que representan diferentes atributos de datos y las diversas relaciones entre los valores de datos . (Amazon, 2023)

Comandos SQL

Según (AWS, 2023) Los comandos de lenguaje de consulta estructurada (SQL) son palabras clave o instrucciones SQL específicas que los desarrolladores utilizan para manipular los datos almacenados en una base de datos relacional.

Se los puede clasificar en

- El lenguaje de definición de datos (DDL)
- El lenguaje de consulta de datos (DQL)
- Las instrucciones del lenguaje de manipulación de datos (DML)
- Los administradores de bases de datos usan el lenguaje de control de datos (DCL)
- El motor relacional utiliza el lenguaje de control de transacciones (TCL)

CRUD

De acuerdo a (Microsoft, 2023) “CRUD significa Crear, Leer, Actualizar y Eliminar, que son las cuatro operaciones básicas de base de datos”

Create (Crear) Esta operación implica la creación de nuevos registros o entradas en una base de datos

Read (Leer) La operación de lectura implica recuperar información de la base de datos

Update (Actualizar) Esta operación permite modificar registros o datos existentes en la base de datos

Delete (Eliminar) La operación de eliminación implica la eliminación de registros o datos de la base de datos.

Tabla Con los principales Motores de Base de Datos Relacionales

Base de Datos	Descripción	Empresa	Tipo de Licencia
MySQL	MySQL es una base de datos de código abierto que es ampliamente utilizada en aplicaciones web y empresariales. Ofrecida por Oracle Corporation.	Oracle Corporation	Licencia pública general (GPL) o comercial
PostgreSQL	PostgreSQL es una base de datos de código abierto conocida por su robustez y capacidad de extensión. Es mantenido por la comunidad PostgreSQL.	PostgreSQL Global Development Group	Licencia PostgreSQL (similar a MIT)
Microsoft SQL Server	Microsoft SQL Server es un sistema de gestión de bases de datos desarrollado por Microsoft. Es ampliamente utilizado en entornos empresariales.	Microsoft	Comercial
Oracle Database	Oracle Database es una base de datos empresarial líder desarrollada por Oracle Corporation. Es conocida por su alta disponibilidad y escalabilidad.	Oracle Corporation	Comercial

Tabla 1.Principales Bases de Datos Relacionales . Fuente Propia

ORM

De acuerdo a (César, 2021) ORM son las siglas de O (Object) R (Relational) M (Mapping) que es una técnica de programación que nos permite trabajar con bases de datos relacionales también conocidos como RDMS (Relational Database Management System), como pueden ser SQL Server, Postgresql, MySQL, Oracle..., el cual nos permite convertir nuestros datos de estas bases de datos relacionales a objetos de nuestro lenguaje orientado a objetos.

Según (ESIC, 2018) “Un ORM es un modelo de programación cuya misión es transformar las tablas de una base de datos de forma que las tareas básicas, que realizan los programadores estén simplificadas”

Esto quiere decir que en lugar de escribir consultas SQL directamente para interactuar con la base de datos, los desarrolladores pueden utilizar el ORM para realizar operaciones de inserción, actualización, eliminación y consulta en la base de datos utilizando objetos y métodos de programación.

Modelo de datos orientado a Objetos

El modelo de datos orientado a objetos es una extensión del paradigma de programación orientado a objetos. Los objetos entidad que se utilizan en los programas orientados a objetos son análogos a las entidades que se utilizan en las bases de datos orientadas a objetos puras, pero con una gran diferencia: los objetos del programa desaparecen cuando el programa termina su ejecución, mientras que los objetos de la base de datos permanecen. A esto se le denomina persistencia (Merche, 2017)

Este modelo de datos se utiliza principalmente en sistemas y lenguajes de programación orientados a objetos, como Java, C++, Python y C#. Es especialmente útil cuando se necesita

representar datos complejos y relaciones entre ellos de una manera más natural y estructurada, como en sistemas de información de gestión empresarial (ERP), sistemas de información geográfica (GIS) y sistemas de bases de datos orientados a objetos. Este enfoque proporciona una forma eficiente y organizada de modelar y gestionar datos en aplicaciones que requieren una representación cercana a la realidad y una estructura jerárquica.

De acuerdo a (keepcoding, 2023) afirma que “Una base de datos orientada a objetos es un sistema de bases de datos que puede trabajar con datos complejos, es decir, objetos que reflejan aquellos objetos que se utilizan en el lenguaje de programación orientado a objetos.”

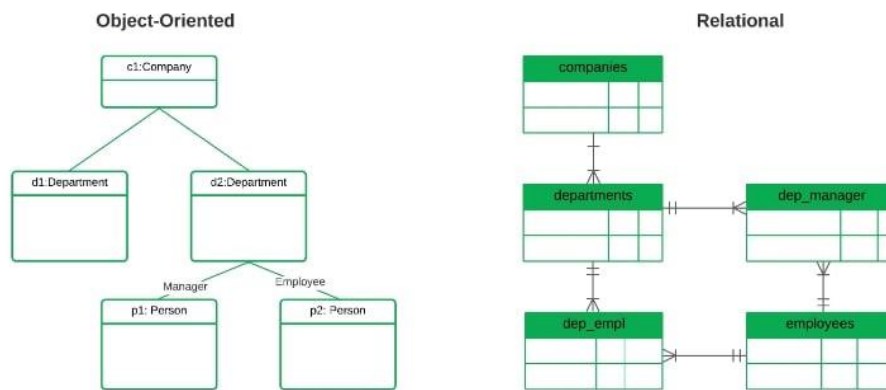


Ilustración 1 . Esquema Orientado a Objetos y Relacional. (keepcoding, 2023)

Mapeo Objeto-Relacional

El mapeo objeto-relacional es una técnica de programación que permite convertir datos del sistema de tipos utilizado por los lenguajes de programación orientado a objetos al utilizado por las bases de datos relacionales (Carlos, 2018)

Las bases de datos relacionales solo permiten guardar tipos de datos primitivos, por lo que no es posible almacenar de forma directa los objetos de la aplicación en las tablas, sin antes realizar algún tipo de conversión (Leonardo, 2021)

Teniendo en cuenta el punto anterior es donde podemos ver uno de los beneficios que ofrecen los ORM al realizar la conversión automática entre los diferentes tipos de datos, para este proceso hacen uso de los patrones de diseño Active Record y Repository

Patrón Repository

El patrón Repository utiliza un repositorio para separar la lógica que recupera los datos y los mapea al modelo de entidades, de la lógica del negocio que actúa en el modelo. El repositorio media entre la capa de fuente de datos y la capa de negocios de la aplicación; encuesta a la fuente de datos, mapea los datos obtenidos de la fuente de datos a la entidad de negocio y persisten los cambios de la entidad de negocio a la fuente de datos. (Enriquez & Busto, 2017)

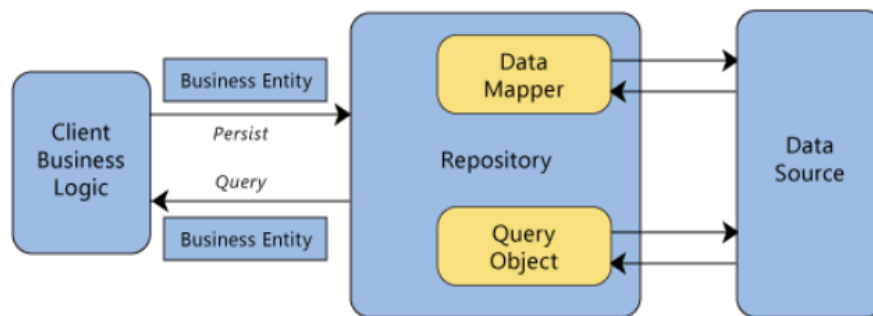


Ilustración 2. Patrón Repository . (Enriquez & Busto, 2017)

Patrón Active Record

Active Record es un patrón en el cual, el objeto contiene los datos que representan a una fila (o tupla) de nuestra tabla o vista, además de encapsular la lógica necesaria para acceder a la base de datos. De esta forma el acceso a datos se presenta de manera uniforme a través de la aplicación (lógica de negocio + acceso a datos en una misma clase (Enriquez & Busto, 2017)

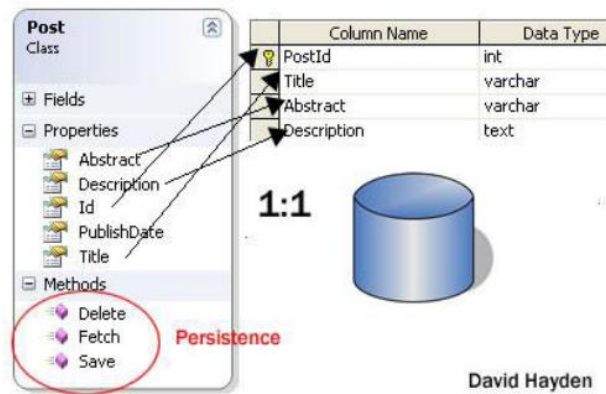


Ilustración 3 .Patrón Active Record. (Enriquez & Busto, 2017)

Tabla Con los Principales ORM

ORM	Descripción	Empresa	Tipo de Licencia
Hibernate	Hibernate es una ORM de código abierto para Java que facilita el mapeo de objetos Java a bases de datos relacionales.	Red Hat	Licencia pública general de GNU (GPL) o LGPL
Entity Framework	Entity Framework es una ORM desarrollada por Microsoft para el ecosistema .NET. Proporciona un mapeo de objetos a bases de datos relacionales.	Microsoft	Licencia MIT
Django ORM	El ORM de Django es parte del framework web Django para Python. Simplifica el acceso y la manipulación de bases de datos relacionales en aplicaciones Django.	Django Software Foundation	Licencia BSD
SQLAlchemy	SQLAlchemy es una ORM de Python que ofrece un enfoque flexible y poderoso para trabajar con bases de datos relacionales.	SQLAlchemy Team	Licencia MIT

Tabla 2. Características Principales ORM . Fuente Propia

De acuerdo al Autor (César, 2021) el uso de un ORM tiene las siguientes Ventajas y Desventajas

Ventajas de usar un ORM

- ✓ No tienes que escribir SQL. Trabajas sólo con el lenguaje que ya conoces
- ✓ Acelera el desarrollo y reduce costes, puesto que nos permite reusar código, tener nuestro código sincronizado con el modelo de datos y, al contrario.
- ✓ Nos permite cambiar de motor de base de datos de forma sencilla.
- ✓ Aunque depende del ORM que uses, por lo general viene con una serie de características como gestión de conexiones, transacciones, migraciones de modelo de datos, transformación de datos, etc.
- ✓ Sabe transformar tus necesidades de acceso a datos al proveedor de base de datos que necesites.
- ✓ Puede que algunas de las consultas las genere mejor que si las hicieras en la Base de Datos directamente, dependiendo del tipo de consulta, obviamente. Puede ser una desventaja el que te genere las consultas, como veremos más adelante.
- ✓ Se ocupan de prevenirnos de problemas como puede ser un ataque de tipo SQL Injection u otros similares.
- ✓

Desventajas de usar un ORM

- ❖ Al no tener que escribir tú las consultas, si se trata de algo complejo, puede dar lugar a generar cierto tipo de consultas muy ineficientes.
- ❖ Tienes que aprender cómo utilizarlos, algunos de ellos tienen cierto grado de complejidad y requiere de tiempo para sacarle el partido necesario para trabajar con ellos de forma eficiente.

- ❖ No todos ellos, pero algunos son pesados, lo cual hace que el rendimiento sea mucho peor que si atacarás directamente a la Base de Datos con una consulta. En ciertos escenarios puede que no esté justificada esta merma de rendimiento y necesites realizar las consultas directamente.
- ❖ En algunos de ellos, la configuración inicial es compleja, ralentizando el desarrollo en exceso.
- ❖ Al no “saber” qué está pasando por debajo, ciertos desarrolladores pueden llegar a perder el control o comprensión de lo que está pasando, con lo que puede ser que ocurran consecuencias no deseadas al realizar cierto tipo de acciones. Por eso el abstraerte tanto de la tecnología subyacente puede ser contraproducente, a veces es una ventaja, pero hay que tener cuidado.

CUADRO COMPARATIVO ENTRE SQL Y ORM

	SQL Directo	ORM
Lenguaje	SQL (Structured Query Language)	Lenguaje de programación (por ej. Python, Java)
Abstracción	Requiere conocimientos específicos de SQL y de la base de datos en uso.	Abstrae la interacción directa con SQL y la base de datos, proporciona una interfaz orientada a objetos.
Mantenimiento	Más propenso a errores y cambios en la estructura de la base de datos pueden requerir modificaciones en múltiples lugares del código.	Mayor facilidad para mantener y actualizar el código, ya que los cambios en la estructura de la base de datos se pueden reflejar automáticamente en el modelo de objetos.
Productividad	Puede ser menos productivo debido a la necesidad de escribir manualmente consultas SQL y gestionar conexiones de base de datos.	Puede aumentar la productividad al proporcionar métodos y clases predefinidos para interactuar con la base de datos.
Seguridad	Vulnerable a ataques de inyección SQL si no se manejan adecuadamente los parámetros de consulta.	Menos vulnerable a ataques de inyección SQL, ya que los ORMs suelen usar consultas parametrizadas o mapeo seguro de objetos a SQL.

Portabilidad	Las consultas SQL pueden variar según la base de datos, lo que puede dificultar la portabilidad del código.	Mayor portabilidad, ya que el ORM puede proporcionar capas de abstracción que ocultan las diferencias en la sintaxis SQL entre diferentes bases de datos.
Rendimiento	En algunos casos, el SQL directo puede ofrecer un mejor rendimiento, ya que permite consultas altamente optimizadas y personalizadas.	Los ORMs pueden introducir cierta sobrecarga de rendimiento debido a la abstracción adicional y las consultas generadas automáticamente, pero en la mayoría de las aplicaciones, la diferencia de rendimiento no es significativa.
Curva de aprendizaje	Puede requerir un conocimiento más profundo de SQL y la base de datos en uso.	Puede ser más fácil de aprender para desarrolladores que están más familiarizados con la programación orientada a objetos que con SQL.

Tabla 3. Tabla Comparativa de Aspectos entre SQL y un ORM . Fuente Propia

CONCLUSIONES

En base a lo Investigación realizada se pudo obtener la siguiente conclusión

El uso de Herramientas ORM simplifica el proceso de mapear el Modelo Relacional de un base SQL al Modelo Orientado a Objetos, permitiendo a los desarrolladores enfocarse en la parte lógica de la solución.

Se Evidencio que al trabajar directamente con entidades en lugar de Sentencias SQL embebidas en el código fuente de las aplicaciones reduce el error humano y evita brechas de seguridad como puede ser una Inyección SQL .

De acuerdo a los resultados obtenidos en la encuesta realizada se pudo comprobar que ambos método tienen sus ventajas y en muchos aspecto el ORM incluso tiene más benéficos, pero esta lejos de reemplazar definitivamente a SQL debido a que presenta limitaciones en algunos aspectos, por tal motivo aún existe un grupo de desarrolladores que indican que el usos de cada método utilizado dependerá del tipo de proyecto que se este abordando puesto que para proyectos sencillos mejor utilizar un ORM mientras que en proyectos donde las consultas sean muy complejas y sobrepasan las capacidades del ORM es mejor utilizar SQL Directo.

Como conclusión se recomienda el uso de un ORM para Bases de datos Relacionales en proyectos donde el tiempo de desarrollo sea un factor importante ya que nos beneficiaremos de algunas de las ventajas que nos proporciona el ORM en temas de productividad.

BIBLIOGRAFIA

- Amazon. (2023). *¿Qué es SQL?* Obtenido de <https://aws.amazon.com/es/what-is/sql/>
- AWS. (2023). *¿Qué son los comandos SQL?* Obtenido de <https://aws.amazon.com/es/what-is/sql/>
- Carlos, Y. (2018). *Herramientas de mapeo objeto-relacional*. Obtenido de <https://www.ceac.es/blog/herramientas-de-mapeo-objeto-relacional-orm>
- César, D. A. (2021). *Qué es un ORM*. Obtenido de <https://openwebinars.net/blog/que-es-un-orm/>
- Enriquez, O. Y., & Busto, H. G. (2017). *Mapeo Objeto / Relacional (ORM)*. Obtenido de <https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/23/21>
- ESIC. (2018). *El ORM como herramienta eficiente de trabajo*. Obtenido de <https://www.esic.edu/rethink/tecnologia/el-orm-como-herramienta-eficiente-de-trabajo>
- Europea, U. (2022). *¿Para qué sirve un gestor de base de datos?* Obtenido de <https://universidadeuropea.com/blog/para-que-sirve-gestor-base-datos/>
- Google. (2023). *¿Qué es una base de datos relacional?* Obtenido de <https://cloud.google.com/learn/what-is-a-relational-database?hl=es-419>
- IBM. (2023). *¿Qué es una base de datos relacional?* Obtenido de <https://www.ibm.com/mx-es/topics/relational-databases>
- keepcoding. (2023). *¿Qué es una base de datos orientada a objetos?* Obtenido de <https://keepcoding.io/blog/base-de-datos-orientada-a-objetos/>

Leonardo, S. G. (2021). *Un análisis comparativo de herramientas ORM*. Obtenido de http://sedici.unlp.edu.ar/bitstream/handle/10915/127331/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y

Merche, M. (2017). *Diseño de Sistemas de Bases de Datos*.

Microsoft. (2023). *Habilitación de operaciones CRUD en ASP.NET Web API 1*. Obtenido de <https://learn.microsoft.com/es-es/aspnet/web-api/overview/older-versions/creating-a-web-api-that-supports-crud-operations>

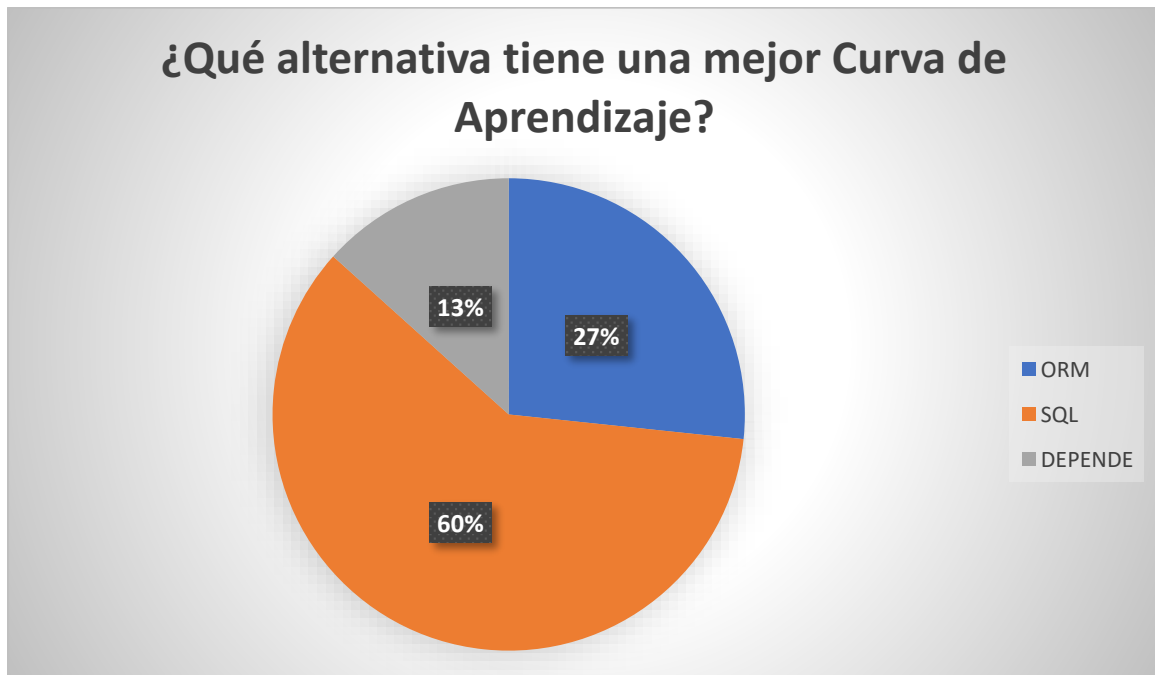
Oppel, A., & Sheldon, R. (2016). *Fundamentos de SQL*. México, D.F: Miguel Ángel Luna Ponce.

Oracle. (2023). *¿Qué es una base de datos?* Obtenido de <https://www.oracle.com/mx/database/what-is-database/>

ANEXOS

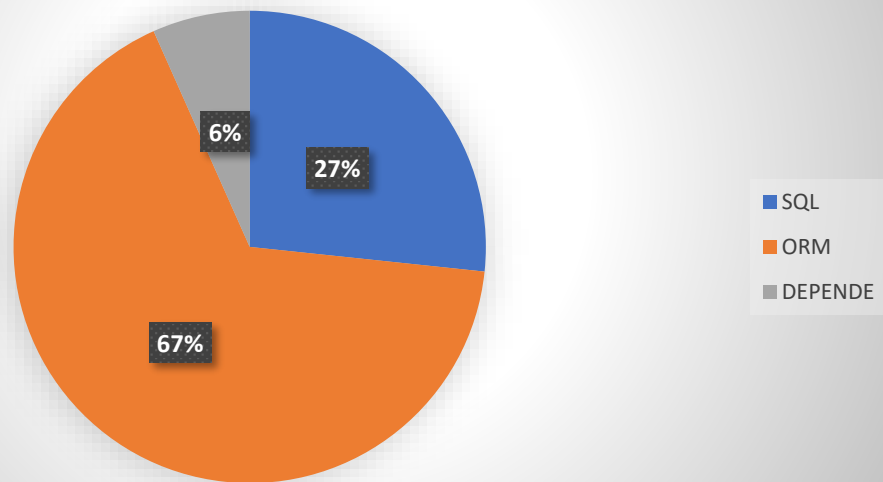
Anexo1.

Los Datos Presentados se obtuvieron por medio la encuesta realizada a 15 Desarrolladores de la Empresa SOLSAP ubicada en la ciudad de Guayaquil



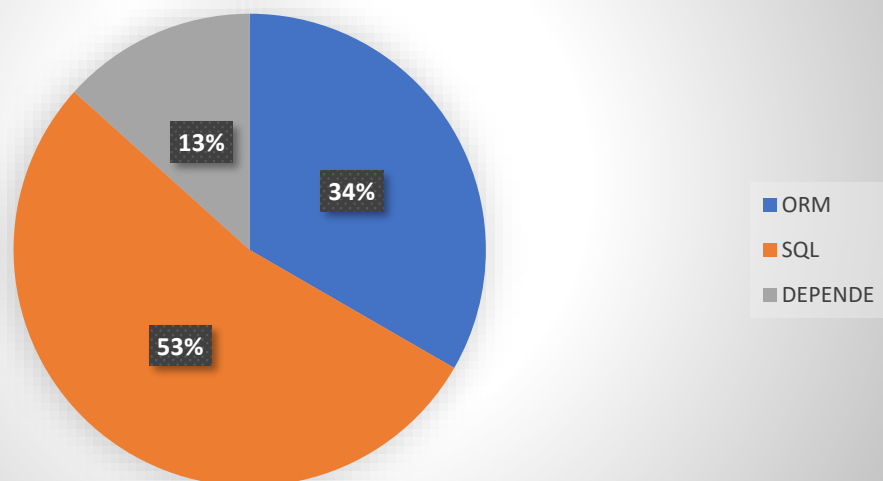
Análisis: Se puede apreciar que en el aspecto de Aprender el lenguaje SQL es mucho más rápido que en aprender a utilizar de forma eficiente un ORM

¿Qué Alternativa permite un Desarrollo más rápido?



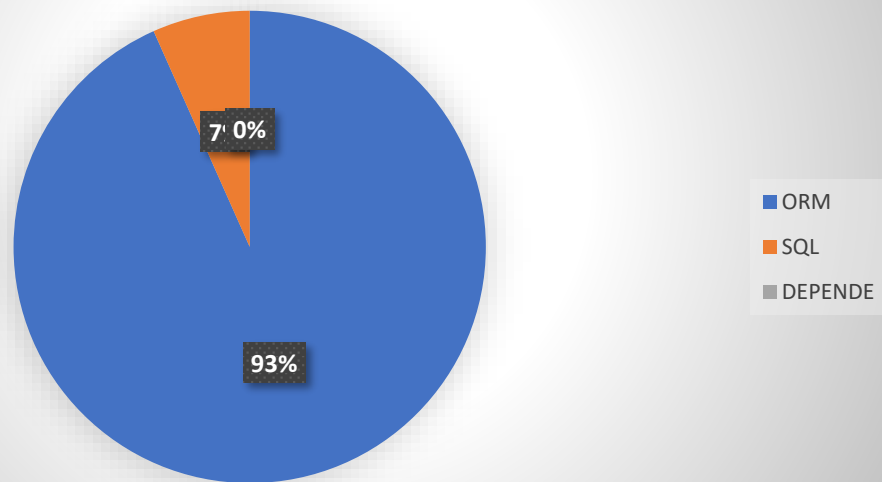
Análisis: El 67% de los desarrolladores encuestados indican que el uso de un ORM les facilita el poder interactuar con la base de datos de una manera más sencilla.

¿Qué alternativa es mejor en cuanto Rendimiento?



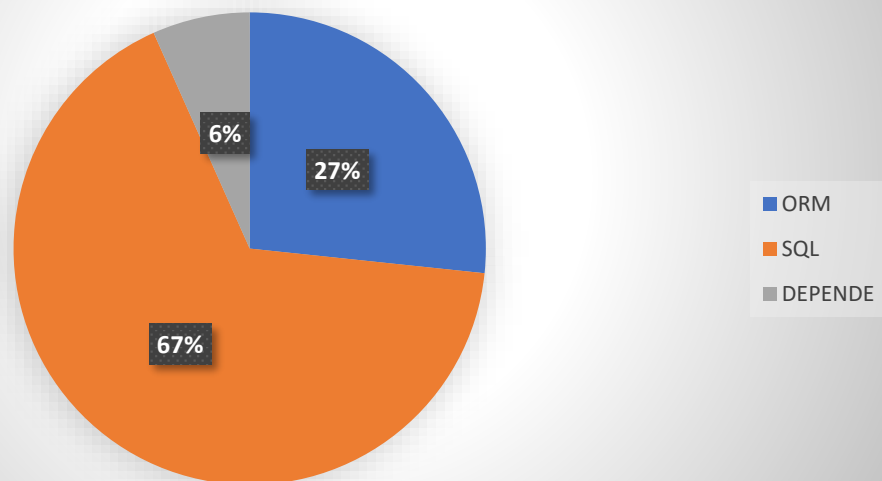
Análisis: De acuerdo a la información recopilada un 53% de los expertos encuentran que las Consultas Elaboradas con instrucciones SQL directas se ejecutan más rápido.

¿Qué alternativa se adapta mejor al Cambio de Base de Datos?



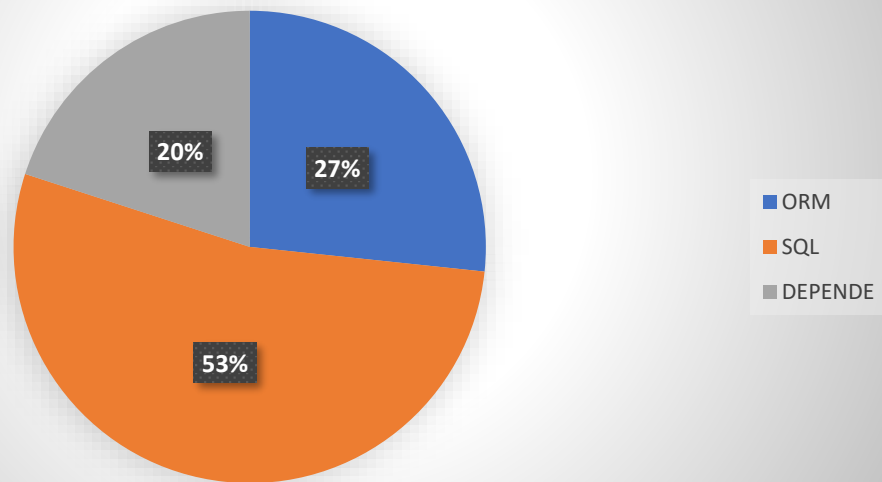
Análisis: Como se puede Observar la mayoría de desarrolladores en el aspecto de presentarse un escenario donde se deba migrar a una nueva base de datos coinciden que el ORM es la mejor alternativa.

¿Qué Alternativa es mejor para trabajar con Reporteria?



Análisis: El 67% de encuestados opinan que es mas sencillo crear una consulta para un reporte donde intervienen varias tablas por medio de la sintaxis SQL.

¿Qué alternativa Prefieren Utilizar en Proyectos Complejos?



Análisis: Un 53% de los Desarrolladores van a preferir utilizar SQL de Forma directa junto a procedimientos almacenados para proyectos que van a requerir mas que sencillas operaciones de CRUD.

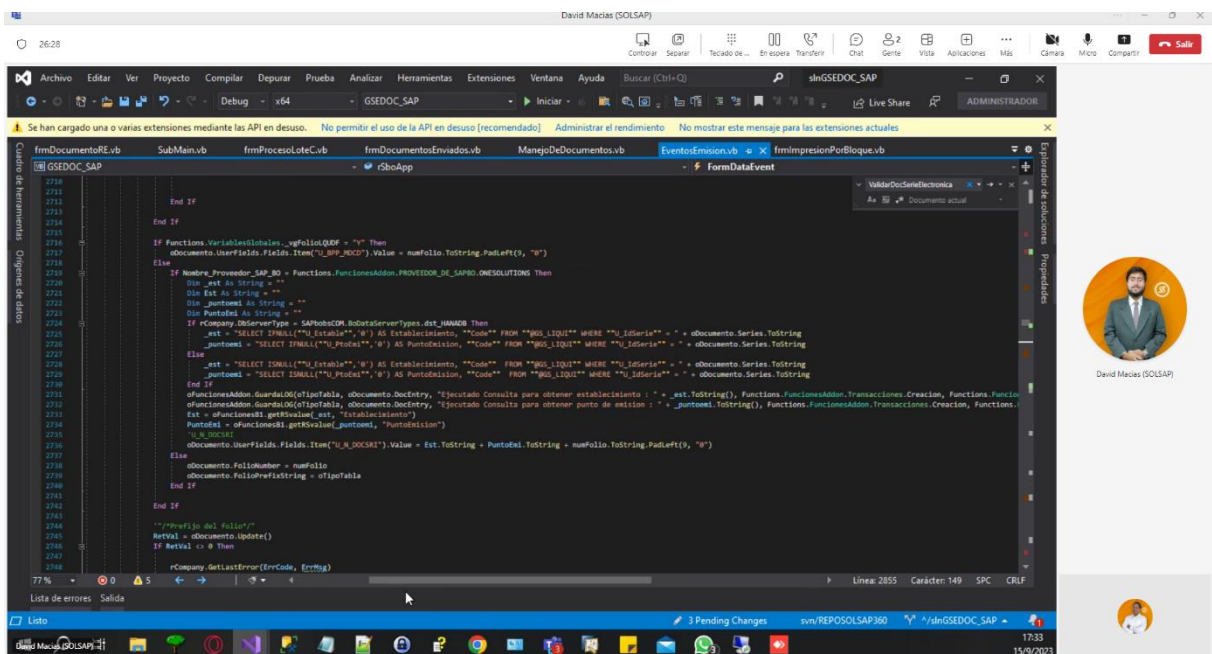


Ilustración 4. Entrevista Virtual a Desarrollador SOL SAP . Fuente Propia

Anexo 2.

Para el Ejercicio Practico se Desarrolló una solución en Visual Studio 2019 formada principalmente por dos proyectos en el primero se integró el ORM Entity Framework muy Utilizado en Entornos .Net y en el segundo Proyecto fue diseñado de una manera mas tradicional creando manualmente las funciones y clases necesarias para Conexión a la base de Datos al igual que las Instrucciones en Lenguaje SQL que permitirá una manipulación directa de los datos que se almacenan en ella.

El motor de Base de Datos que se utilizo es MS SQL SERVER en su versión 2019 y como Base de datos de Test se descargó uno de los archivos de Backup alojados en el repositorio de la Empresa Microsoft, en este caso fue la Base AdventureWorks2019.bak.

Descarga de archivos de copia de seguridad

Use estos vínculos para descargar la base de datos de ejemplo adecuada para su escenario.

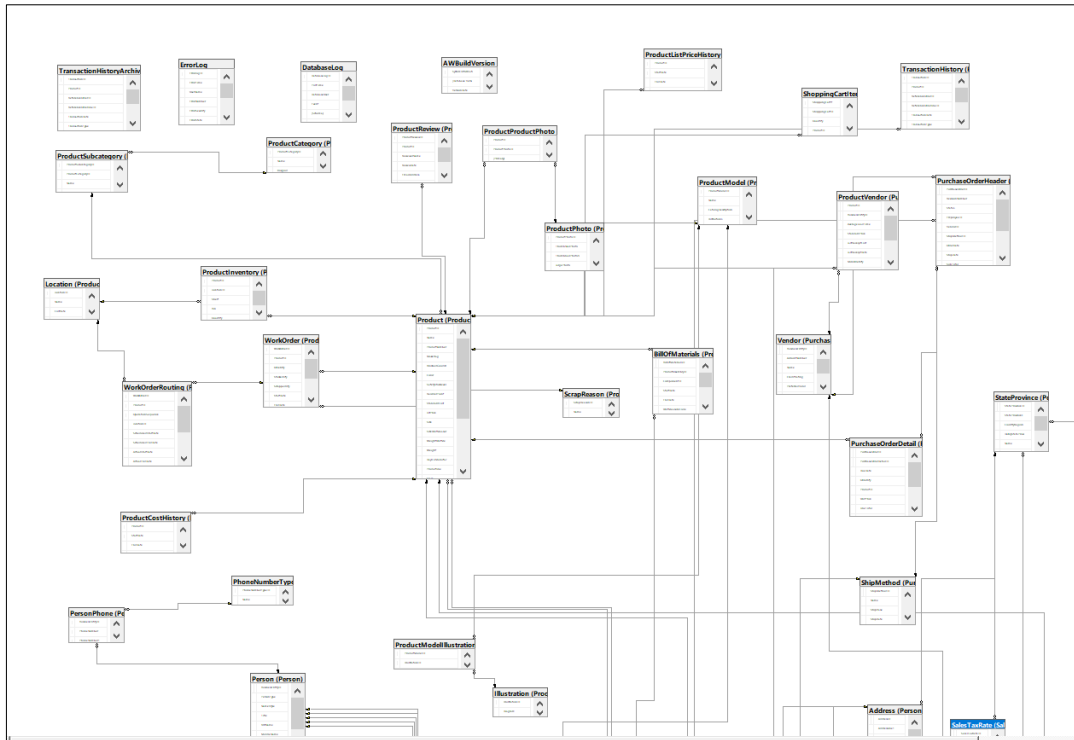
- Los datos **OLTP** son para las cargas de trabajo de procesamiento de transacciones en línea más habituales.
- Los datos de **Data Warehouse (DW)** son para cargas de trabajo de almacenamiento de datos.
- Los datos de **Lightweight (LT)** son una versión ligera y reducida del ejemplo de **OLTP**.

Si no está seguro de lo que necesita, comience con la versión de OLTP que coincida con la versión de SQL Server.

OLTP	Almacenamiento de datos	Ligero
AdventureWorks2022.bak	AdventureWorksDW2022.bak	AdventureWorksLT2022.bak
AdventureWorks2019.bak	AdventureWorksDW2019.bak	AdventureWorksLT2019.bak
AdventureWorks2017.bak	AdventureWorksDW2017.bak	AdventureWorksLT2017.bak
AdventureWorks2016.bak	AdventureWorksDW2016.bak	AdventureWorksLT2016.bak
AdventureWorks2016_EXT.bak	AdventureWorksDW2016_EXT.bak	N/D
AdventureWorks2014.bak	AdventureWorksDW2014.bak	AdventureWorksLT2014.bak
AdventureWorks2012.bak	AdventureWorksDW2012.bak	AdventureWorksLT2012.bak
AdventureWorks2008R2.bak	AdventureWorksDW2008R2.bak	N/D

Se pueden encontrar archivos adicionales directamente en GitHub:

Finalizado el proceso de Restauración en nuestro ambiente Local pudimos generar el diagrama de la base de datos como se muestra a continuación.



Operaciones CRUD desde Proyecto ORM

Para Integrar Entity Framework al proyecto se agregó un nuevo elemento tipo “ADO.NET Entity Data Model” a partir de este punto se debe completar el asistente e indicarle los accesos y base de datos a utilizar

Asistente para Entity Data Model

Elegir la conexión de datos

¿Qué conexión de datos debe usar la aplicación para conectarse a la base de datos?

laptop-n3gs2h7v.AdventureWorks2019.dbo Nueva conexión...

Esta cadena de conexión parece contener datos confidenciales (por ejemplo, una contraseña) que son necesarios para conectarse con la base de datos. Almacenar datos confidenciales en la cadena de conexión puede suponer un riesgo para la seguridad. ¿Desea incluir estos datos en la cadena de conexión?

No, excluir datos confidenciales de la cadena de conexión. Los estableceré en el código de mi aplicación.

Sí, incluir datos confidenciales en la cadena de conexión.

Cadena de conexión:

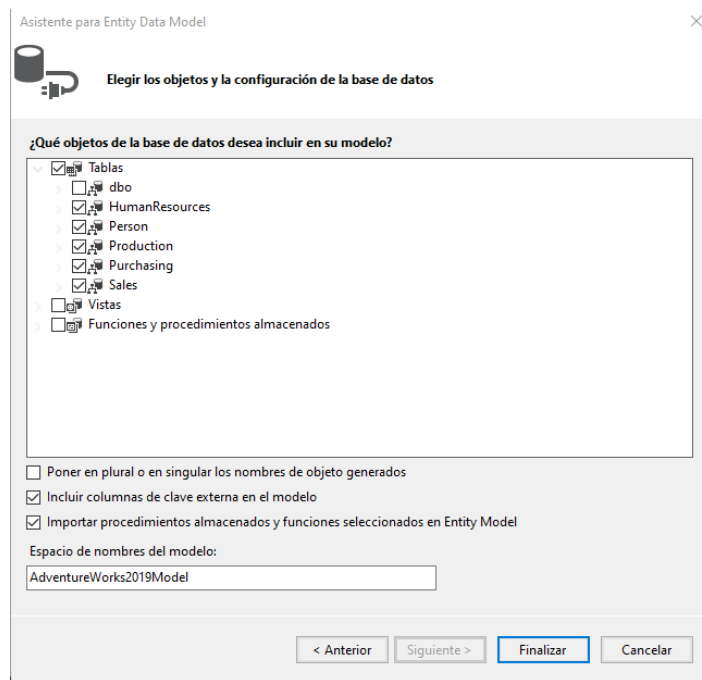
```
metadata=res://*/Modelos.Model1.csdl|res://*/Modelos.Model1.ssd|
res://*/Modelos.Model1.msl;provider=System.Data.SqlClient;provider connection string="data
source=localhost;initial catalog=AdventureWorks2019;user
id=sa;password=*****;MultipleActiveResultSets=True;App=EntityFramework"
```

Guardar configuración de conexión en App.Config como:

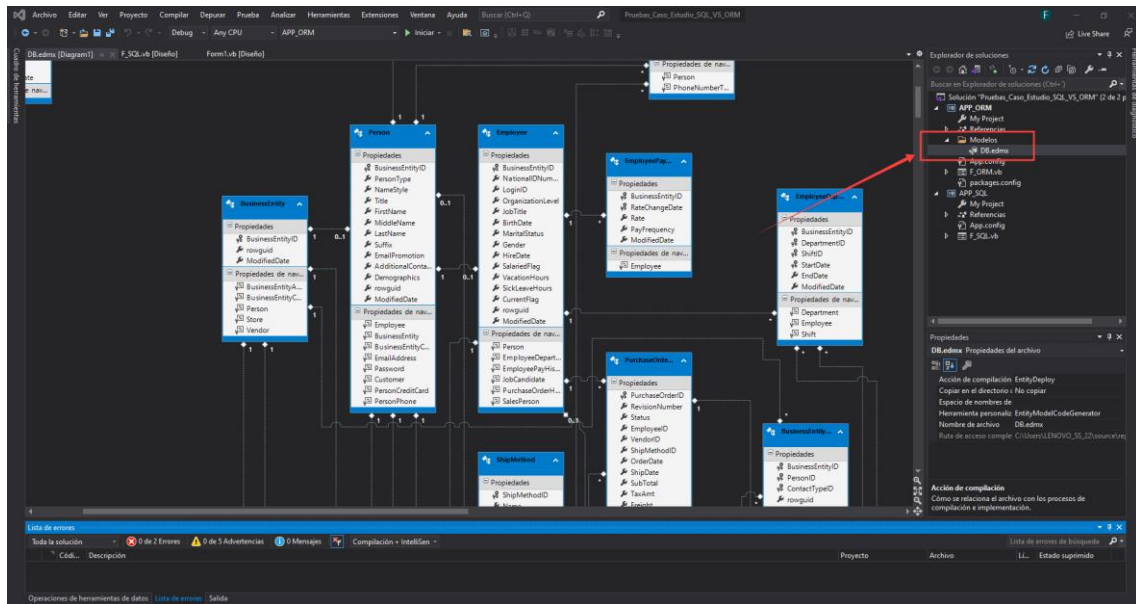
AdventureWorks2019Entities1

< Anterior Siguiente > Finalizar Cancelar

Luego de Ingresar correctamente las credenciales se nos muestra una ventana que permitirá seleccionar las diferentes tablas que posteriormente formaran parte del modelo generado por el ORM



Finalizando el asistente pudimos validar la generación del archivo .edmx que es modelo de datos Entity Data Model XML utilizado en el contexto del desarrollo de aplicaciones .Net



Operaciones CRUD

Al Utilizar el ORM lo primero que se pudo Comprobar es que ya disponemos de manera automática acceso a todas las clases que fueron mapeadas a partir de cada una de las tablas de nuestra base de datos ahorrándonos un tiempo considerable tratándose de una BD que está conformada por varias decenas de tablas.

Las pruebas que se realizaron están enfocadas en la tabla maestra Production.Product de la cual seleccionamos un grupo de campos para efecto de testing

APP_ORM

TEST ORM

ID: 680
 Codigo: FR-R32B-58
 Nombre: HL Road Frame - Lime, 58
 Color: Lime
 Unidad Medida: CM
 Costo: 1059.3100
 Stock: 500

CREAR
 LEER
 ACTUALIZAR
 ELIMINAR

ID	Nombre	Codigo	Color	Costo	Stock	UM	Fecha_Ve
680	HL Road Frame ...	FR-R32B-58	Lime	1059.3100	500	CM	30/4/2008
706	HL Road Frame ...	FR-R32R-58	Red	1059.3100	500	CM	31/5/2011
717	HL Road Frame ...	FR-R32R-62	Red	868.6342	500	CM	31/5/2011
718	HL Road Frame ...	FR-R32R-44	Red	868.6342	500	CM	31/5/2011
719	HL Road Frame ...	FR-R32R-48	Red	868.6342	500	CM	31/5/2011
720	HL Road Frame ...	FR-R32R-52	Red	868.6342	500	CM	31/5/2011
721	HL Road Frame ...	FR-R32R-56	Red	868.6342	500	CM	31/5/2011
722	LL Road Frame ...	FR-R38B-58	Black	204.6251	500	CM	31/5/2011
723	LL Road Frame ...	FR-R38B-60	Black	204.6251	500	CM	31/5/2011

Subrutina para leer los datos , Permite Recuperar la información desde la Base de datos

```
Private Sub LeerProductos()  
    Using db As New AdventureWorks2019Entities  
        Dim personas = From p In db.Product Where p.Color IsNot Nothing And p.SizeUnitMeasureCode IsNot Nothing  
                        Select ID = p.ProductID, Nombre = p.Name,Codigo = p.ProductNumber,  
                                p.Color, Costo = p.StandardCost, Stock = p.SafetyStockLevel, UM = p.SizeUnitMeasureCode,  
                                Fecha_Venta = p.SellStartDate  
        dgvProductos.DataSource = personas.ToList  
    End Using  
End Sub
```

Subrutina para Crear un nuevo Registro

```
1 referencia  
Private Sub CrearProducto()  
    Using db As New AdventureWorks2019Entities  
        Dim producto As New Product  
        producto.Name = txtnombre.Text  
        producto.ProductNumber = txtcod.Text  
        producto.Color = txtcolor.Text  
        producto.StandardCost = txtcosto.Text  
        producto.SafetyStockLevel = txtstock.Text  
        producto.SizeUnitMeasureCode = txtummedida.Text  
        producto.DaysToManufacture = 5  
        producto.SellStartDate = DateTime.Now  
        producto.ReorderPoint = 5  
        db.Product.Add(producto)  
        db.SaveChanges()  
        MessageBox.Show("Datos Guardados con Exito")  
    End Using  
End Sub
```

Subrutina para Modificar un Registro Existente


```

Private Sub ModificarProducto()
    Using db As New AdventureWorks2019Entities
        Dim producto As Product
        producto = db.Product.Find(CInt(txtid.Text))

        producto.Name = txtnombre.Text
        producto.ProductNumber = txtcod.Text
        producto.Color = txtcolor.Text
        producto.StandardCost = txtcosto.Text
        producto.SafetyStockLevel = txtstock.Text
        producto.SizeUnitMeasureCode = txtumedida.Text

        db.Entry(producto).State = Entity.EntityState.Modified
        db.SaveChanges()

        MessageBox.Show("Datos Editados con Exito")
    End Using
End Sub

```

Subrutina para Eliminar un registro Existente

```

1 referencia
Private Sub EliminarProducto()
    Using db As New AdventureWorks2019Entities
        Dim producto As Product

        producto = db.Product.Find(CInt(txtid.Text))

        db.Product.Remove(producto)

        db.SaveChanges()

        MessageBox.Show("Datos eliminados con Exito")
    End Using
End Sub

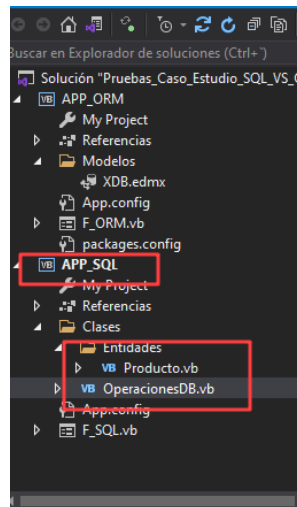
```

Como se pudo observar al utilizar un ORM no es indispensable conocer el Lenguaje SQL el trabajo de acceso a la base de Datos lo Realiza en este caso Entity Framework en segundo plano permitiendo que los desarrolladores realicen operaciones directamente sobre los objetos del modelo.

Operaciones CRUD desde Proyecto SQL

Para el segundo Proyecto debido a que se diseñó para ser implementado de una manera clásica, lo primero que se trabajó fue en la creación de una Clase que permitiera administrar la conexión con la base de datos y una clase que representara la entidad del Producto

El tiempo que conlleva en mapear todas las tablas de la base es significativo razón por lo cual solo se mapeó manualmente la tabla de productos.



Rutina para Leer Datos

```
Private Sub LeerProductos()  
  
    Dim SQL = "Select ProductID as ID,Name as Nombre,ProductNumber asCodigo, "  
    SQL += "Color,StandardCost as Costo,SafetyStockLevel as Stock,SizeUnitMeasureCode as UM, "  
    SQL += "SellStartDate as Fecha_Venta FROM Production.Product "  
    SQL += "WHERE Color IS NOT NULL AND SizeUnitMeasureCode IS NOT NULL"  
  
    Dim reg = opdb.RecuperarRegistros(SQL)  
  
    If reg.Rows.Count > 0 Then  
        dgvProductos.DataSource = reg  
    End If  
End Sub
```

En esta ocasión las Instrucciones en SQL están escritas directamente sobre una Cadena de texto , los desarrolladores deben tener un nivel de conocimiento medio-alto sobre el Lenguaje SQL para no cometer error de sintaxis.

El tiempo que se invirtió en construir una consulta SQL embebida dentro de la lógica de programación fue mucho más largo que en la prueba con el ORM, la única ventaja que se determinó al realizarlo de esta manera fue la flexibilidad de crear una Consulta compleja donde intervenían varias Joins o uniones entre tablas.

Rutina para crear un nuevo registro

```
Private Sub CrearProducto(p As Producto)

    Dim SQL = "INSERT INTO Production.Product(Name,ProductNumber,Color,StandardCost,SafetyStockLevel, "
    SQL += "SizeUnitMeasureCode,DaysToManufacture,SellStartDate,ReorderPoint,ListPrice) "
    SQL += String.Format("VALUES ('{0}','{1}','{2}',{3},{4}','{5}','{6}','{7}','{8}','{9}')", p.Name, p.ProductNumber, p.Color, p.Sta
    Dim reg = opdb.EjecutarQuery(SQL)
    If reg > 0 Then

        MessageBox.Show("Datos Guardados con Exito")

    Else

        MessageBox.Show("No se Guardo el Producto")

    End If

End Sub
```

Rutina para Modificar el valor de un registro

```
1 referencia
Private Sub ModificarProducto(p As Producto, ID As Integer)

    Dim SQL = "UPDATE Production.Product SET "
    SQL += String.Format("Name='{0}',ProductNumber='{1}',Color='{2}',StandardCost={3},SafetyStockLevel={4} ", p.Name, p.Prod
    SQL += "Where ProductID =" & ID

    Dim reg = opdb.EjecutarQuery(SQL)

    If reg > 0 Then

        MessageBox.Show("Datos Modificados con Exito")

    Else

        MessageBox.Show("No se Modifico el Producto")

    End If

End Sub
```

Por último la Rutina de Eliminación

```
Private Sub EliminarProducto(ID As Integer)
    Dim SQL = "DELETE FROM Production.Product Where ProductID=" & ID.ToString

    Dim reg = opdb.EjecutarQuery(SQL)

    If reg > 0 Then
        MessageBox.Show("Datos Eliminados con Exito")
    Else
        MessageBox.Show("No se Elimino el Producto")
    End If
End Sub
```

El resultado de la Ejecución del Proyecto SQL

ID	Nombre	Codigo	Color	Costo	Stock	UM	Fecha_Ve
680	HL Road Frame ...	FR-R92B-58	Lime	1059,3100	500	CM	30/4/2008
706	HL Road Frame ...	FR-R92R-58	Red	1059,3100	500	CM	30/4/2008
717	HL Road Frame ...	FR-R92R-62	Red	868,6342	500	CM	31/5/2011
718	HL Road Frame ...	FR-R92R-44	Red	868,6342	500	CM	31/5/2011
719	HL Road Frame ...	FR-R92R-48	Red	868,6342	500	CM	31/5/2011
720	HL Road Frame ...	FR-R92R-52	Red	868,6342	500	CM	31/5/2011
721	HL Road Frame ...	FR-R92R-56	Red	868,6342	500	CM	31/5/2011
722	LL Road Frame ...	FR-R38B-58	Black	204,6251	500	CM	31/5/2011
723	LL Road Frame ...	FR-R38B-60	Black	204,6251	500	CM	31/5/2011



Babahoyo, 18 de Septiembre del 2023

CERTIFICACIÓN DE PORCENTAJE DE SIMILITUD CON OTRAS FUENTES EN EL SISTEMA DE ANTIPLAGIO

En mi calidad de Tutor del Trabajo de la Investigación de el, **Sr. Macias Carpio Fersen Arturo**, cuyo tema es: **Estudio Comparativo entre SQL Directo y un ORM**, certifico que este trabajo investigativo fue analizado por el Sistema Antiplagio Compilatio, obteniendo como porcentaje de similitud de [8 %], resultados que evidenciaron las fuentes principales y secundarias que se deben considerar para ser citadas y referenciadas de acuerdo a las normas de redacción adoptadas por la institución y Facultad.

Considerando que, en el Informe Final el porcentaje máximo permitido es el 10% de similitud, queda aprobado para su publicación.



Por lo que se adjunta una captura de pantalla donde se muestra el resultado del porcentaje indicado.

ING. IVAN RUIZ PARRALES
DOCENTE TUTOR DE LA FAFI.