

Herramienta de Mapeo Objeto Relacional y la Productividad de la Empresa Interfases Software Group

Object Relational Mapping Tool and Business Productivity Software Interfaces Group

Freddy Patricio Baño Naranjo^{*}, Washington Bryan Chingo Esquivel[†], y Fausto Alberto Viscaino[‡]
 Universidad Regional Autónoma de los Andes UNIANDÉS, Ecuador
 bryanbyr@gmail.com, {freddybn,faustoviscaino}@uniandes.edu.ec

Fecha de recepción: 6 de agosto de 2016 — **Fecha de aceptación:** 16 de agosto de 2016

Resumen—El desarrollo de software busca mejorar la productividad de las empresas por medio de la automatización y el uso de herramientas, las empresas que se dedican al desarrollo de software tienen como objetivo ayudar a otras empresas en la automatización y desarrollo de herramientas para este fin, pero muchas veces las mismas olvidan su propia productividad. Las Herramientas ORM (Object Relational Mapping) se han ideado con este fin, al evitar repetir muchas líneas de programación en la capa de abstracción, pero es necesario considerar que cada una tiene su estándar o su propio lenguaje por lo cual se requeriría tiempo adicional para ocuparlas. El desarrollo de este ORM tiene como objetivo proveer una herramienta acorde a las necesidades y el estándar de programación de la empresa Interfases, logrando que durante el desarrollo de software, este se centre en otros aspectos más relevantes, como el diseño de la base de datos, interfaz de usuario y la capa de negocios, olvidándonos casi por completo de la capa de abstracción. Con el uso de la herramienta ORM se logró reducir el tiempo de desarrollo de la aplicación de prueba.

Palabras Clave—ORM, Productividad, Desarrollo de Software, Bases de Datos

Abstract—Software development seeks to improve business productivity through automation and the use of tools, companies engaged in software development are aimed at helping other companies in the automation and development of tools for this purpose, but often they forget their own productivity. The ORM Tools (Object Relational Mapping) have been devised for this purpose, to avoid repeating many lines of programming abstraction layer, but it is necessary to consider that each has its standard or your own language by which additional time is required to occupy them. The development of these ORM aims to provide a tool according to the needs and programming standard enterprise interfaces, making for software development, this focus on other more important aspects, such as the design of the database, user interface and business layer, forgetting almost entirely abstraction layer. Using the ORM tool was reduced development time of the test application.

Keywords—ORM, Productivity, Software Development, Databases.

INTRODUCCIÓN

La productividad en todo sistema de operación de bienes o servicios obedece a la relación que guardan los resultados obtenidos para con los recursos empleados en el logro de los mismos, este factor es de vital importancia ya que de ser favorable se estará en condiciones de permanecer en el mercado cada vez más competitivo. Esta premisa ha llevado a las empresas a establecer diversos mecanismos de control con la convicción de elevar su desempeño a través de la mejora de sus indicadores de operación.

En la ciudad de Ambato se encuentra funcionando la empresa Interfases desde el año 2005, su labor principal es desarrollar sistemas informáticos a medida. En varias conversaciones con el jefe de desarrollo y en observaciones directas se han podido deducir algunas dificultades sustanciales que tienen que ver con el desarrollo de sistemas, entre ellas podemos mencionar.

Demoras excesivas en la definición del CRUD (create, read, update, delete) de la base de datos, sus conexiones y su entorno

inicial. Elevado número de instrucciones definidas paso a paso por los diferentes programadores.

Inconsistencias cuando se hace un desarrollo individual para luego poder empatar los módulos respectivos. Estas dificultades repercuten directamente en la productividad de la empresa, elevando los costos de producción del software.

El presente trabajo investigativo tiene como objetivo general: Desarrollar una herramienta informática de tipo ORM para que en base a su utilización se logre acelerar los procesos del desarrollo de software y con ello elevar la productividad en la empresa Interfases de la ciudad de Ambato.

Para lograr este objetivo general se plantean los siguientes objetivos específicos:

- Fundamentar científicamente la teoría que tiene que ver con el mapeo de objetos relacionales (ORM) y la productividad empresarial.
- Diagnosticar los niveles de eficiencia que tiene la empresa durante el desarrollo de sistemas.

Desarrollar la herramienta O.R.M

El proyecto se orienta en la línea de investigación denominada Desarrollo de software y programación de sistemas.

Se considera que con la implementación de esta herramienta ORM se logrará una aceleración en los procesos de desarrollo

^{*}Ingeniero en Sistemas e Informática, Magister en Informática.

[†]Ingeniero en Sistemas e Informática.

[‡]Ingeniero en Sistemas e Informática, Magister en Gerencia Informática Con Mención en Desarrollo de Software y Redes.

de software, una economía en el tiempo necesario para generar líneas de código y obviamente la disminución de los errores de programación, todo esto permitirá el mejoramiento de la productividad laboral y por ende la reducción de los costos para la empresa. Por todos estos aspectos positivos se justifica plenamente el desarrollo de esta investigación.

El fundamento científico del presente proyecto investigativo tiene que ver con los siguientes aspectos:

Ingeniería de software

La ingeniería del software es la aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento del software, que es la aplicación de la ingeniería del software (IEEE, 1990).

La ingeniería de Software es el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales (Pressman and Troya, 1988).

En base a estas definiciones podemos decir que la ingeniería de software es una disciplina que nos ayuda a llevar un desarrollo organizado de software asegurando su calidad.

El software se desarrolla

El software es un elemento lógico y se diferencia del hardware que es un elemento físico, es por ello que el software no se fabrica en el sentido clásico. Aunque existen similitudes entre el desarrollo del software y la construcción del hardware, ambas actividades son fundamentalmente distintas (INTECO, 2009).

Cada producto software es diferente porque se construye para cumplir los requisitos únicos de un cliente. Cada software necesita, por lo tanto, ser construido usando un enfoque de ingeniería (INTECO, 2009).

El software no se estropea

Los defectos no detectados harán que falle el programa durante las primeras etapas de su vida. Sin embargo, una vez que se corrigen (suponiendo que no se introducen nuevos errores) los fallos disminuyen (INTECO, 2009).

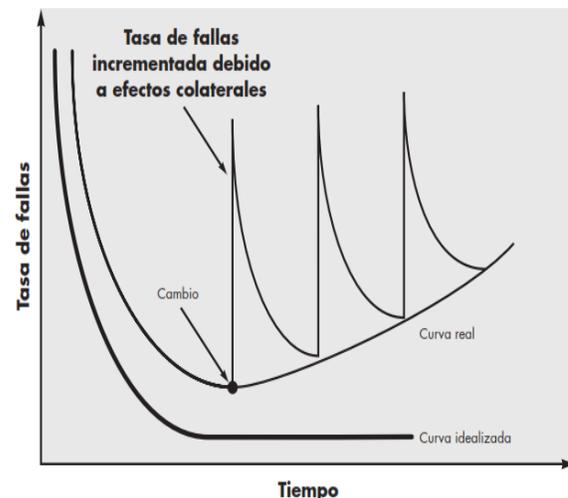


Figura 1. Curvas de falla del software.

Fuente: (Pressman and Troya, 1988).

El software no se estropea, pero se deteriora. Durante su vida, el software sufre cambios (mantenimiento). Conforme se hacen los cambios, es bastante probable que se introduzcan nuevos defectos, lo que hace que el software se vaya deteriorando debido a los cambios (INTECO, 2009).

El software se construye a medida

A medida que evoluciona una disciplina de ingeniería, se crea un conjunto de componentes estandarizados para el diseño. Los tornillos estándar y los circuitos integrados pre construidos son sólo dos de los miles de componentes estándar que utilizan los ingenieros mecánicos y eléctricos conforme diseñan nuevos sistemas. Los componentes reutilizables han sido creados para que el ingeniero pueda concentrarse en los elementos verdaderamente innovadores de un diseño; es decir, en las partes de éste que representan algo nuevo (Pressman and Troya, 1988).

Aunque la mayoría de la industria tiende a ensamblar componentes, en el caso del software, la mayoría se construye a medida. Aunque la reutilización y ensamblaje de componentes está aumentando, el software con frecuencia se construye de acuerdo a los requisitos específicos de un cliente (INTECO, 2009).

ORM

A diferencia de ERM y UML, ORM (Object-Role Modeling) es construir a partir de una base lingüística utilizando frases estructuradas en lenguaje natural como punto de partida. ORM no hace uso de la noción de atributos, pero considera el universo de trabajo en términos de objetos que juegan papeles. Los objetos se clasifican en grupos de tipos de entidades y tipos de valor (Voß, 2013).

Las entidades pueden ser cualquiera de los conceptos abstractos (posiblemente) y son referencias de valores. Por ejemplo, una persona (entidad) puede ser referenciada por su nombre (valor). Modelado comienza con ejemplos concretos

que se dividen en hechos elementales y se convierten a predicados. Este proceso será ejemplificado fi cado en la siguiente (Voß, 2013).

Mapeo Objeto Relacional

El mapeo objeto-relacional es una técnica de programación para convertir datos del sistema de tipos utilizado en un lenguaje de programación orientado a objetos al utilizado en una base de datos relacional. En la práctica esto crea una base de datos virtual orientada a objetos sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (esencialmente la herencia y el polimorfismo) (Del Busto and Enriquez, 2012).

Las bases de datos relacionales solo permiten guardar tipos de datos primitivos (enteros, cadenas de texto, etc.) por lo que no se pueden guardar de forma directa los objetos de la aplicación en las tablas, sino que estos se deben de convertir antes en registros, que por lo general afectan a varias tablas. En el momento de volver a recuperar los datos, hay que hacer el proceso contrario, se deben convertir los registros en objetos. Es entonces cuando ORM cobra importancia, ya que se encarga de forma automática de convertir los objetos en registros y viceversa, simulando así tener una base de datos orientada a objetos (Del Busto and Enriquez, 2012).

Tecnología y Productividad

La primera revolución en la información fue el invento de la escritura, la segunda la generó el libro, la tercera la imprenta y, por último, la cuarta y actual revolución en la información es resultado de las innovaciones tecnológicas de las últimas seis décadas en el campo de la electrónica y las comunicaciones, y está modificando muchos aspectos de la vida. La posibilidad de intercambiar información y comunicarse instantáneamente, a escala mundial y a bajo costo, a través de internet y otros medios satelitales, ha provocado el debilitamiento de las fronteras y el mutuo contacto de los pueblos del mundo (Pulido, 2010).

Hoy en día podemos decir que casi cualquier actividad humana eficiente viene marcada por una característica común, el uso de algún tipo de tecnología que hace que el proceso sea más eficiente, más rápido en su entrega o se puedan obtener resultados en mayor cantidad y precisión. Desde consultar una agenda de teléfonos ubicada en la Red con el último modelo de Smartphone, poder video chatear con la otra parte del mundo con un mínimo de infraestructura de comunicaciones e incluso evaluar el crecimiento y engorde de cerdos ibéricos mediante chips implantados bajo su piel y con sistemas de radiofrecuencia. Este progreso hace que la sociedad use la tecnología como un medio para mejorar sus procesos, su calidad de vida u obtener mejores resultados (PortalProgramas.com, 2011).

DESARROLLO

Métodos

La modalidad investigativa aplicada fue de tipo cuali-cuantitativa, es decir un híbrido entre modalidad cualitativa y

cuantitativa. Los aspectos cualitativos fueron determinados en base a observación directa en la empresa, mientras que los aspectos cuantitativos se lograron en base a la investigación de campo.

Los tipos de investigación aplicada son:

Bibliográfica: utilizada para lograr el fundamento científico relacionado con metodologías ORM y con productividad laboral.

De campo: llevada cabo en el mismo sitio donde se obtuvieron la manifestación de la situación problemática, es decir en la empresa Interfases de la ciudad de Ambato.

Los métodos investigativos aplicados fueron:

El analítico sintético: aplicado para elaborar el fundamento teórico, se procedió a la recopilación de información, luego se la analizó y finalmente se la sintetizó en el marco teórico.

Inductivo - deductivo. Usado para generalizar una solución a partir de una solución específica.

La población involucrada en la problemática es bastante reducida, se limita a su Gerente y a 2 desarrolladores, por lo tanto no hizo falta el cálculo de una muestra.

La técnica aplicada para la investigación de campo fue la entrevista, se utilizó la guía de entrevista como instrumento de investigación.

De las entrevistas realizadas se determinó lo siguiente:

- Efectivamente se tienen demoras en el desarrollo de algunos procesos relacionados con diseño, desarrollo de CRUD y conexión a las bases de datos.
- Son usuales los errores durante el ensamblado de módulos cuando se desarrolla en forma individual por diferentes desarrolladores.
- Estas dificultades inciden en el nivel productivo de la empresa, ya que retrasan los tiempos planificados.

Desarrollo de la herramienta

Al momento de desarrollar software siempre se busca garantizar la calidad del mismo por lo cual la metodología seleccionada para su desarrollo debe tomar en cuenta los diferentes factores involucrados en el desarrollo.

Para el desarrollo de software en este caso se ha seleccionado PSP (Proceso de software personal) ya que se concentra en las prácticas de trabajo de los ingenieros de forma individual, se aplica a programas pequeños de menos de 10000 líneas de código.

Proceso de Software Personal

La estructura del proceso de PSP se muestra a partir de una declaración requisitos, el primer paso en el proceso de PSP es

la planificación. Mientras que los ingenieros están siguiendo el script para hacer el trabajo, registran su tiempo y datos de defectos en los registros de tiempo y defectos. Al final del trabajo, durante la fase de post-mortem (PM), que resumen los datos de tiempo y defectos de los registros, medir el tamaño del programa, y entran estos datos en el formulario de resumen del plan.

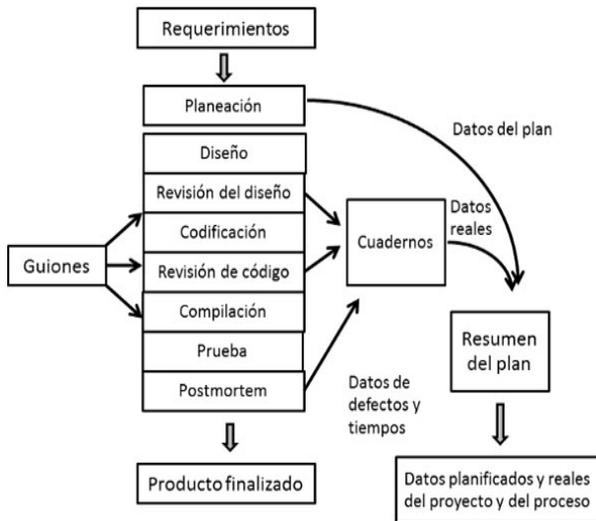


Figura 2. Flujo de procesos del Modelo PSP.
Fuente: (Puello, 2014).

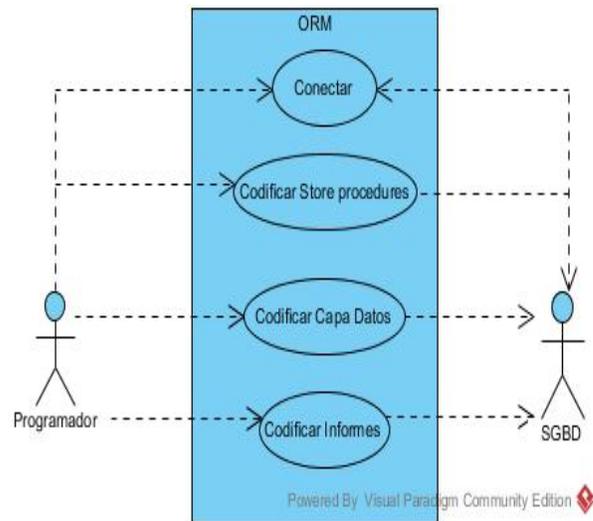


Figura 3. Caso de uso de Codificación.
Fuente: Elaboración Propia.

Diccionario de Datos de Actores

Tabla 1. Descripción de los Actores

No	Actor	Descripción
1	Programador	Persona encargada de codificar los store procedure, las clases, la capa de abstracción y la presentación de datos en los informes.
2	SGBD	Sistema encargado de la gestión de las bases de datos (Pueden ser MySQL, SQL o PostgreSQL).

Fuente: Elaboración propia.

Descripción de los Procesos

Tabla 2. Descripción de los Procesos

No	Proceso	Descripción
1	Conectar	El programador se conecta al SGBD.
2	Codificar Store Procedure	En este proceso el programador codifica los SP en base a las tablas del Sistema Gestor de Bases de Datos.
3	Codificar Capa de Datos	En este proceso el programador codifica las transacciones que el programa lleva a cabo con la base de datos.
4	Codificar Informes	En este proceso el programador codifica la presentación de ciertos datos de interés del usuario.

Fuente: Elaboración propia.

Declaración Requisitos

De acuerdo a los requerimientos de la empresa se han desarrollado los siguientes diagramas: diagramas de caso de uso, diagramas de secuencia y diagramas de clases, los mismos que nos ayudaran a desarrollar los distintos módulos del software.

Caso de Uso de Codificación

El caso de uso indica cómo se realiza por lo general el proceso de codificación de la empresa el cual se da en base a las tablas que se encuentran en el sistema gestor de base de datos.

Caso de uso Conectar

El caso de uso indica cómo se realiza por lo general el proceso conexión al sistema gestor de base de datos.

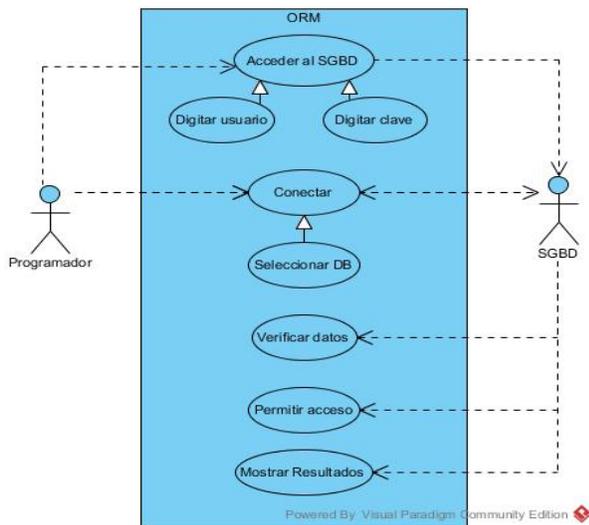


Figura 4. Caso de Uso Conexión
Fuente: Imagen Propia

Aplicación de la herramienta

Luego del desarrollo de la herramienta, esta se ha convertido en la primera implementación de un ORM desarrollado en la Carrera de Sistemas de Uniandes, se ha probado el mismo en la empresa interfasas. A continuación describimos de una manera breve el mismo.

Formulario Login

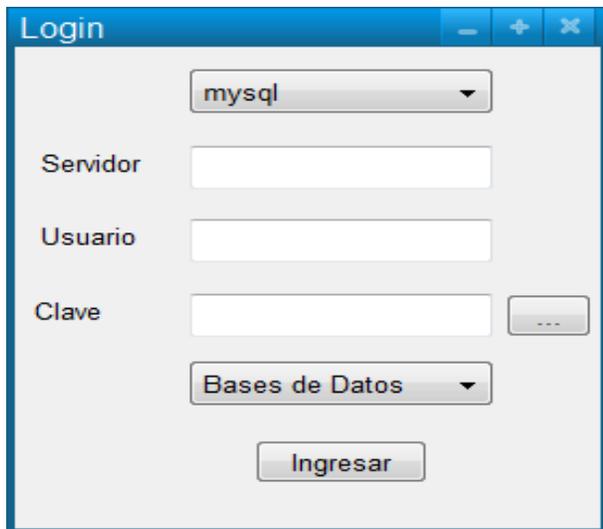


Figura 5. Formulario Login
Fuente: Imagen Propia

En el formulario de login podemos encontrar el JComboBox que contiene los Sistemas Gestores de Bases de datos compatibles con el ORM, un JTextField en el cual se ingresa la dirección del SGBD, dos JTextField para usuario y contraseña, un JButton para el acceso al SGBD que nos devuelve la Bases de Datos disponibles al segundo JComboBox y por ultimo un JButton con el cual se realiza la conexión a la Base de Datos específica y nos muestra el siguiente formulario.

Formulario Básico

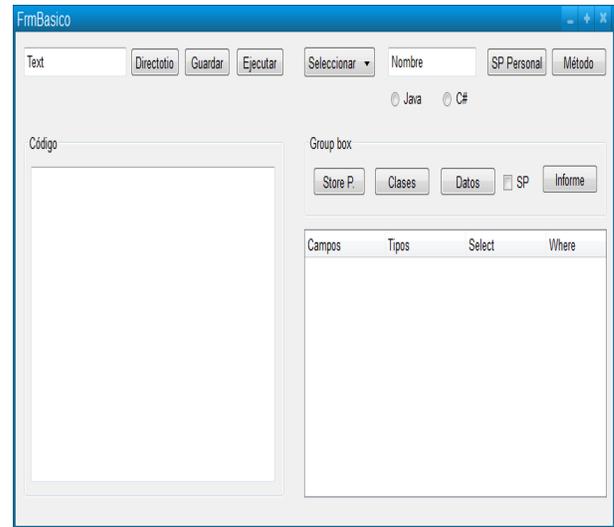


Figura 6. Diseño Formulario Básico
Fuente: Imagen Propia

Este formulario contiene un JTextField que nos permite ingresar el directorio donde se va a guardar los Store Procedure y las clases de las distintas opciones elegidas a través del JButton Directorio o bien sea del Guardar, el JButton Ejecutar nos permitirá crear los Store Procedure en los distintos SGBD.

A continuación se encuentra un JComboBox Donde se presentan la diferentes tablas que se encuentran en la Base de Datos seleccionada al final tenemos dos JButton que nos permiten crear un Store Procedure o un segmento de código personalizado por medio del JTable de la parte inferior en la cual podemos seleccionar los campos que se van a usar, el nombre de estos Segmentos lo podremos en el JTextField que se encuentra antes de estos componentes y bajo el mismo dos JRadioButton donde seleccionar el lenguaje de programación en que se generarán todos los códigos que precisemos.

Antes de la JTable se hallan agrupadas la mayoría de las acciones básicas y principales del ORM a través de 4 JButton, el primero “Store P.” nos permite generar los Store Procedure para insertar, actualizar, listar, seleccionar y eliminar una tabla de la Base de Datos, luego se encuentra el botón “Clases” que nos generan la tabla como un objeto con sus atributos y respectivos métodos get y set, el botón “Datos” nos crea la capa de abstracción de Datos en base a los Store Procedure o directamente a través de consultas directas en clase a su la tenemos un JCheckBox que nos permitirá elegir como se realizará esta abstracción, al final de este grupo termina con el botón “Informe” que nos presentará el Panel Opciones.

Por ultimo tenemos un JTextArea que presentara todo el código o Store Procedure generados.

Panel Opciones

El Panel Opciones se muestra a través de un mensaje con opción de Aceptar contiene un JTree que muestra las tablas con sus respectivos campos y dos JButton con los cuales

podemos seleccionar los campos a sus respectivos JList que representan al select y where de una consulta compleja.

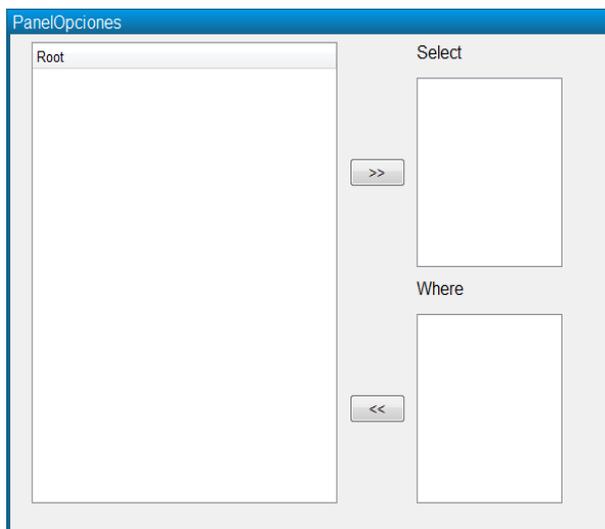


Figura 7. Panel Opciones

Fuente: Imagen Propia

REFERENCIAS BIBLIOGRÁFICAS

- Del Busto, H. G. and Enriquez, O. Y. (2012). Mapeo objeto/relacional (orm). *Revista Telemática*, 10(3):1-7.
- IEEE (1990). The Standish Group.
- INTECO, I. N. d. T. d. I. C. (2009). Laboratorio nacional de calidad del software–servicios de apoyo a la empresa. PortalProgramas.com (2011). Valoración del software libre en la sociedad.
- Pressman, R. S. and Troya, J. M. (1988). *Ingeniería del software*. Number 001.64 P74s. McGraw Hill.
- Puello, Plinio y Oviedo, S. D. y. F. D. A. (2014). Metodología para el diagnóstico de prácticas del modelo de proceso personal de software. *Información tecnológica*, 25:57 – 66.
- Pulido, H. G. (2010). Calidad total y productividad.
- Voß, J. (2013). Describing data patterns. a general deconstruction of metadata standards.

RESULTADOS

Luego de la implementación de la herramienta ORM en la empresa se han obtenido los siguientes resultados:

- Reducción de un 30 % del tiempo en el proceso de elaboración de CRUD de la base de datos, conexión a la misma y más.
- Estandarización en la nomenclatura de la capa de acceso a los datos.
- La compatibilidad de los módulos se elevó en un 25 % debido a la disminución de errores en las líneas de código
- El tiempo del proceso de desarrollo fue sustancialmente inferior en comparación con proyectos anteriores.

CONCLUSIONES

- La ingeniería de software es una disciplina que nos ayuda a comprender que es el software, su ciclo de vida desde su concepción hasta su discontinuación y las diferentes formas en que este ciclo puede manejarse. Además nos permite desarrollar software de calidad en base a métodos y herramientas.
- Lenguajes de Modelado Conceptual nos indican de manera más concreta los métodos y herramientas que mencionamos en la ingeniería de software y como han mejorado y disminuido el tiempo de desarrollo de software.
- El uso de la tecnología apoya la productividad de las empresas en todo ámbito que se pueda imaginar, por lo tanto el área de desarrollo de software también debe hacer uso de toda herramienta tecnológica que le ayude en el desarrollo rápido de software y verificando que este sea de Calidad.
- El uso de la aplicación ORM logro mejorar la productividad de la empresa Interfaces, ya que se desarrollan los productos de software en menor tiempo y asegurando la calidad del mismo.