

COMPILACIÓN DE ARTICULOS DE
INVESTIGACIÓN OCTUBRE 2010.

Red Académica Internacional UADY, UAM, WPI,
TAMU, ICA, e invitados.



5

**DESARROLLO DE HERRAMIENTA
CASE PARA EL DISEÑO Y
DESARROLLO DE CAPA DE
PERSISTENCIA DE DATOS PARA
PAQUETES COMPUTACIONALES**

**M.en Ing.Const. Jesús
Nicolas Zaragoza Grifè**

DESARROLLO DE HERRAMIENTA CASE PARA EL DISEÑO Y DESARROLLO DE CAPA DE PERSISTENCIA DE DATOS PARA PAQUETES COMPUTACIONALES

M. en Ing. Const. Jesús Nicolás Zaragoza Grifè

Facultad de Ingeniería, Mérida, Yucatán, México
Correo: zgrife72@tunku.uady.mx

“DESARROLLO DE HERRAMIENTA CASE PARA EL DISEÑO Y DESARROLLO DE CAPA DE PERSISTENCIA DE DATOS PARA PAQUETES COMPUTACIONALES”.

RESUMEN

El presente artículo describe el diseño conceptual de una aplicación denominada FIUADY-Coder que permite generar el código fuente para la capa de persistencia de datos, que forma parte de un paquete computacional de uso general. El código fuente es generado a través de un modelo que permite asociar entidades entre sí que finalmente representan clases de objetos con atributos específicos y métodos preestablecidos para lograr la persistencia de la información relativa a la aplicación en desarrollo, en una base de datos. El sistema fue programado en Visual Basic 6.0 y se utilizó una base de datos de tipo relacional diseñada en Microsoft Access 2003. Finalmente se presenta la herramienta terminada y funcional, así como un ejemplo de código fuente obtenido como resultado. Se concluye que la aportación principal de este trabajo es tener una herramienta computacional que permita ser un apoyo al desarrollador de aplicaciones computacionales, que al utilizarse permita el ahorro de tiempo valioso en tareas rutinarias que no le aportan valor a la aplicación final. Se menciona que la herramienta puede coadyuvar a mejorar la productividad en un departamento de desarrollo de software. También permite que el desarrollador se enfoque más en el análisis de las reglas de negocios que diferencian un paquete computacional de otro, que en la parte del paquete computacional que es necesario diseñar, programar de manera repetitiva cada vez que sea necesario diseñar y desarrollar una nueva aplicación. Ya que este proceso está sujeto a una inversión considerable de tiempo en su diseño y codificación, FIUADY-Coder permite hacer menos costoso el desarrollo de la parte de persistencia de datos. Se hacen recomendaciones para permitir que esta herramienta al utilizar los mismos modelos, genere código fuente en otros lenguajes tales como Java Script, Visual Basic .NET o C#.

Palabras Clave: Herramientas CASE, capa de persistencia, Visual Basic, Ingeniería de Software

INTRODUCCIÓN

El desarrollo de paquetes computacionales ha venido en aumento en las últimas décadas. Hoy en día existe software para casi todas las áreas de la industria. La computación aplicada ha venido a tomar tal auge en las áreas civilizadas del mundo, que ya es imposible pensar en no interactuar con un paquete computacional casi todos los días. El acceso a información de calidad a través del uso de los programas de cómputo permite a usuarios de todo el mundo tomar más y mejores decisiones en sus entornos de trabajo, las cuales coadyuvan a mejorar la eficiencia de las organizaciones. Sin embargo, el desarrollo de tales herramientas computacionales requiere de una inversión considerable de recursos humanos, tecnológicos y de conocimiento. Los programas y sistemas de información han de diseñarse e integrarse de tal manera que éstos se puedan adaptar a las necesidades de las organizaciones y no viceversa. Aquellos sistemas que no permitan la adaptabilidad a las organizaciones están condenados al desuso. El desarrollo de sistemas computacionales confiables y consistentes ha sido una meta de los ingenieros de software en la última década. Existen tendencias hacia la estandarización de los procesos para el levantamiento de requerimientos y para el diseño de los mismos. Incluso han sido desarrollados lenguajes especiales para poder especificar las partes que componen a cada sistema diseñado. Existen muchos lenguajes de programación actualmente, así como también una enorme cantidad de manejadores de bases de datos que permiten la persistencia de la información. Actualmente, existe una tendencia actual a desarrollar software utilizando también software específico para ello. A éste tipo de software se le conoce como CASE (Computer Aided Software Engineering). Las herramientas CASE han venido a revolucionar la manera en que se conciben y se diseñan los sistemas computacionales pudiendo manejar diseños muy complejos que de otra manera serían incosteables de realizar. Las herramientas CASE se cimentan en la especificación de lenguajes de diseño de sistemas como el UML (Unified Modeling Language). Generalmente, estas herramientas permiten a los ingenieros de software y programadores en general, poder concebir primero un diseño muy completo del software a través de un modelo. El modelo del software, creado con la herramienta CASE le permite a los desarrolladores ver y entender la complejidad y las posibles dificultades que pudiera tener la programación y la implementación de dicho paquete computacional (Case, 1985). Esta filosofía de diseño es compartida también en la industria de la construcción en la cual se diseña a través de modelos preconcebidos un edificio para luego volverse realidad al construirse. Esta filosofía ha tomado un auge importante en las dos últimas décadas, la cual se conoce como BIM (Building Information Modeling), la cual tiene aplicaciones múltiples en la industria de la construcción. Ya que existe una tendencia a utilizar herramientas que integren la información con modelos de las construcciones desde varios puntos de vista: costos, tiempo, calidad, sustentabilidad, impacto ambiental, etc. Cabe mencionar, que el desarrollo de aplicaciones BIM para el diseño-construcción conllevan lograr la interoperabilidad de productos tales como Revit (Autodesk, 2011) con aplicaciones y sistemas propios de las organizaciones, por lo que es necesario contar con herramientas que permitan agilizar el diseño de software.

En este artículo se presenta el desarrollo y utilización de un paquete computacional denominado FIUADY-Coder capaz de generar todo el código fuente necesario para la capa de persistencia de un marco de trabajo (Framework) para una aplicación específica. Esto es logrado a través de la modelación de un conjunto de entidades asociadas entre sí dependiendo de la lógica del diseño. Cada entidad tiene un conjunto de atributos que pueden ser persistentes o simples variables que no persisten sus valores en una base de datos. El modelo permite especificar tres tipos de entidades: La entidad persistente, la entidad persisten lateral y la colección de entidades persistentes. En conjunto, el modelo da forma al Framework, que viene siendo la superestructura de una herramienta computacional.

El lenguaje utilizado en el desarrollo de esta herramienta fue Visual Basic 6.0, sin embargo es posible realizar el cambio a lenguajes más modernos como Java Script, Visual Basic .NET o C# sin mayor complicación, ya que la arquitectura es la misma. Este sistema permite que el desarrollador no pierda tiempo escribiendo línea por línea el programa haciendo de esto un proceso proclive a tener errores sustanciales que conlleven a la falla del sistema que se pretenda diseñar con esta herramienta (Fairbanks, 2004). Cabe mencionar, que la capa de persistencia de FIUADY-Coder fue generada y revisada con la misma herramienta, se reescribió utilizando la misma herramienta.

MATERIALES Y MÉTODOS

El desarrollo de este trabajo fue realizado utilizando lenguaje de modelado UML 2.0 (Fowler, 2003) para poder definir las entidades, asociaciones y atributos. Primeramente, entre los requerimientos del paquete computacional, fue necesario definir las entidades persistentes. Estas entidades persistentes principales permiten definir al modelo de la capa de persistencia con el cual el paquete computacional trabaja. Entre las entidades persistentes principales del paquete computacional a desarrollar se encontraron: proyecto, entidad, atributo y enumeración. Estas entidades se muestran en la Figura 1.

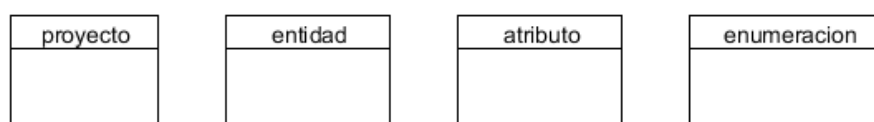


Figura 1. Entidades Persistentes Principales del Paquete Computacional FIUADY-Coder

La entidad proyecto, permite persistir la información acerca del proyecto de capa de persistencia que se pretende modelar. La entidad denominada entidad (válida la redundancia) permite persistir la información acerca de las entidades de las cuales será compuesto el modelo. La entidad atributo, permite persistir la información acerca de cada uno de los atributos persistentes que tendrá cada entidad. La entidad enumeración servirá para cuando sea necesario definir algún atributo de alguna entidad, cuyos valores sean estados finitos.

Por ejemplo: Un atributo puede ser 'DíaDeLaSemana' para el cual los estados finitos de la enumeración serían el conjunto de ítems {'Lunes', 'Martes', 'Miércoles', 'Jueves', 'Viernes', 'Sábado', 'Domingo'}. En la Figura 2 se muestra la asociación entre las cuatro entidades descritas anteriormente.

COMPILACIÓN DE ARTICULOS DE INVESTIGACIÓN OCTUBRE 2010

Red Académica Internacional UADY, UAM, WPI, TAMU, ICA e invitados

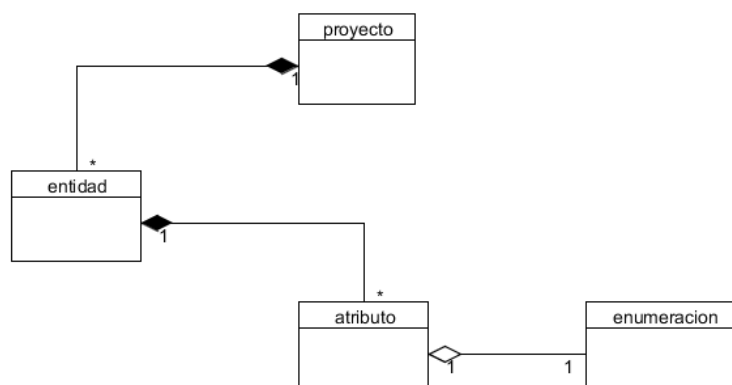


Figura 2. Asociaciones detectadas en la primera fase de diseño de paquete computacional

Como se muestra en la figura 2 se detectó que la entidad proyecto guarda una asociación de uno a muchos con entidad. La entidad a su vez guarda una relación uno a muchos con atributo y finalmente en caso de que el atributo pudiera ser de tipo enumerado la relación es de uno a uno. Hasta este punto del diseño, se detectó que para proyectos varios se podrían reutilizar definiciones parciales del modelo, es decir por ejemplo: al diseñar una aplicación se podría modelar la entidad persona y la cual podría utilizarse para varios modelos que definen modelos de sistemas diferentes.

Entonces, aplicando el principio de la reutilización se llegó a una solución como la que se muestra en la Figura 3.

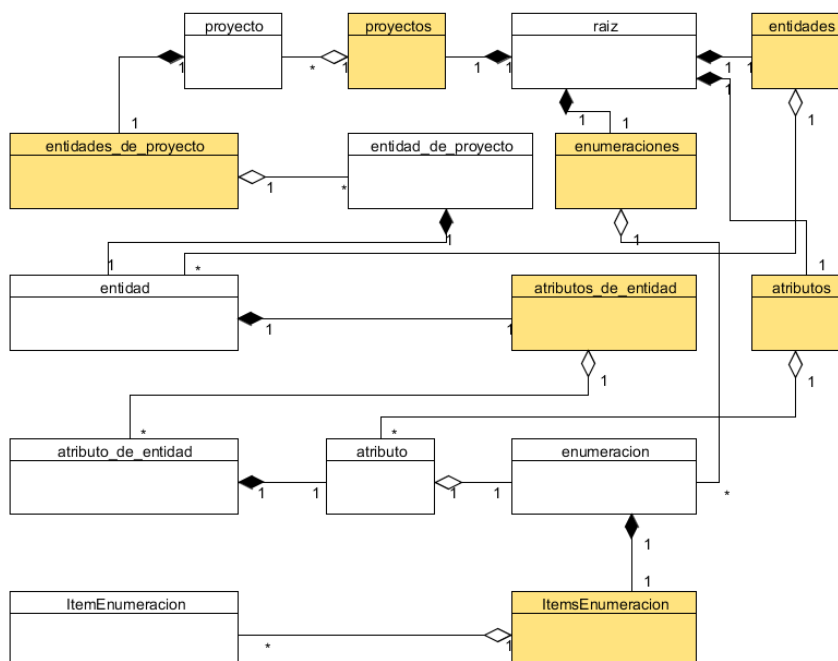


Figura 3. Diagrama de Clases de Persistencia y asociaciones para FIUADY-Coder

De acuerdo al diagrama mostrado en la Figura 3 se tiene que toda la capa de persistencia tiene como base la entidad denominada raíz. La entidad raíz permite la conexión a la base de datos y tiene asociaciones con cuatro entidades de que representan colecciones de cuatro tipos definidos anteriormente (proyecto, entidad, atributo, enumeración). De esta manera cada proyecto tiene a su vez una colección de entidades de proyecto que permiten asociar entidades a un proyecto. Por otro lado cada una de las entidades tiene asociación con una colección de atributos de entidad que permite asociar atributos con entidades. Finalmente existe una colección de enumeraciones que permiten asociarse a un atributo en caso de que éste sea de tipo enumerado.

El código utilizado para programar este modelo definido en la figura 3 fue escrito en Visual Basic 6.0. La base de datos utilizada fue una base de datos de Microsoft Access con extensión MDB.

En la figura 4 se muestra el diagrama entidad relación que define las relaciones y atributos persistentes de cada una de las entidades.

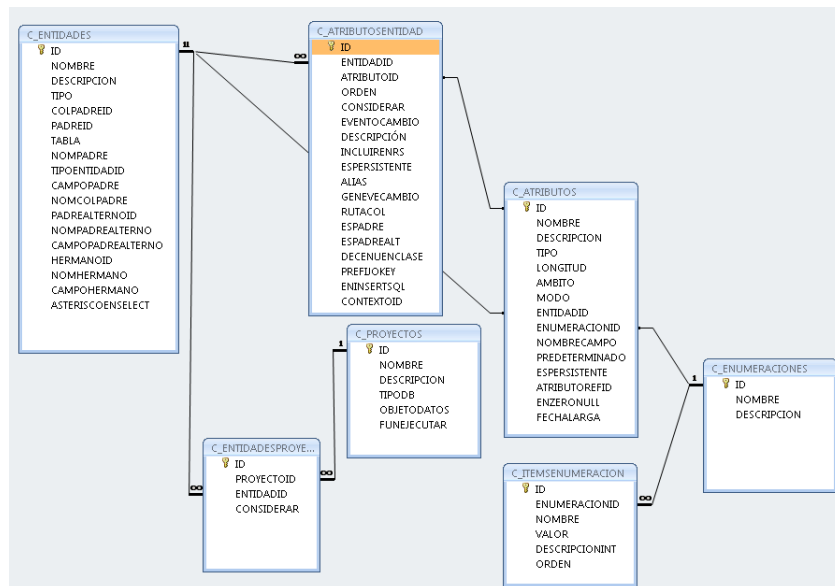


Figura 4 - Diagrama E-R (Entidad-Relación) del modelo de una base de datos para modelos utilizada por FIUADY-Coder

COMPILACIÓN DE ARTICULOS DE INVESTIGACIÓN OCTUBRE 2010

Red Académica Internacional UADY, UAM, WPI, TAMU, ICA e invitados

RESULTADOS Y DISCUSIÓN

El resultado principal de este trabajo fue una aplicación para Windows compatible desde la versión de sistema operativo XP. Esta aplicación permite definir el modelo de la capa de persistencia de una aplicación para Windows. Ya que el modelo ha sido definido, es posible generar el código fuente en lenguaje Visual Basic 6.0 para poder copiarlo en un proyecto y ponerlo a funcionar de inmediato. Utilizando esta herramienta, se ahorra tiempo considerable en tareas rutinarias que permitan a un sistema hacer persistir la información en una base de datos relacional. El conjunto de archivos de código fuente generador para un modelo definido utilizando FIUADY-Coder establece un marco de trabajo de persistencia (Framework) para una aplicación para la cual se realice el modelo. Cabe mencionar que una de las filosofías más importantes que permite el modelado con esta herramienta es la reutilización de patrones que pueden servir para más de una aplicación específica.

En las Figuras 5, 6, 7 se muestran pantallas de la aplicación FIUADY-Coder en diferentes facetas: Catálogos de Entidades y Atributos, Enumeraciones, Generador de código fuente y código fuente ejemplo.

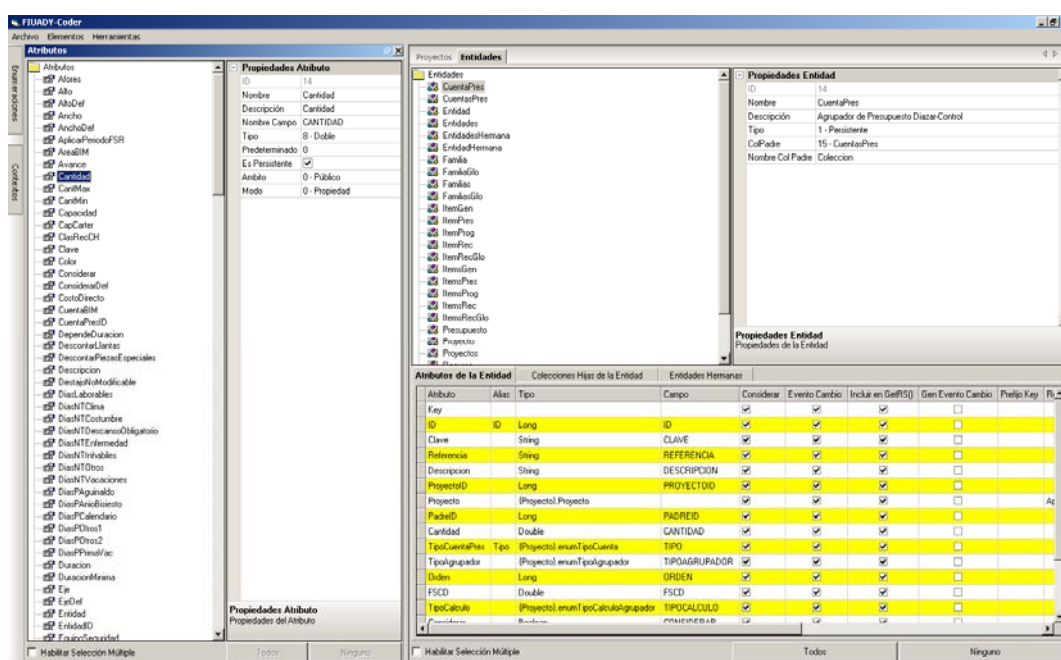


Figura 5. Catálogos de Entidades y Atributos de un modelo en FIUADY-Coder

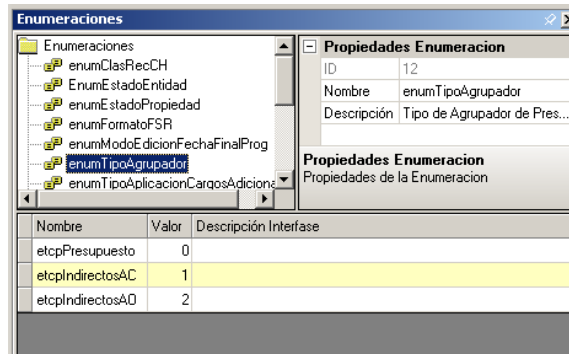


Figura 6. Catálogo de Enumeraciones en FIUADY-Coder

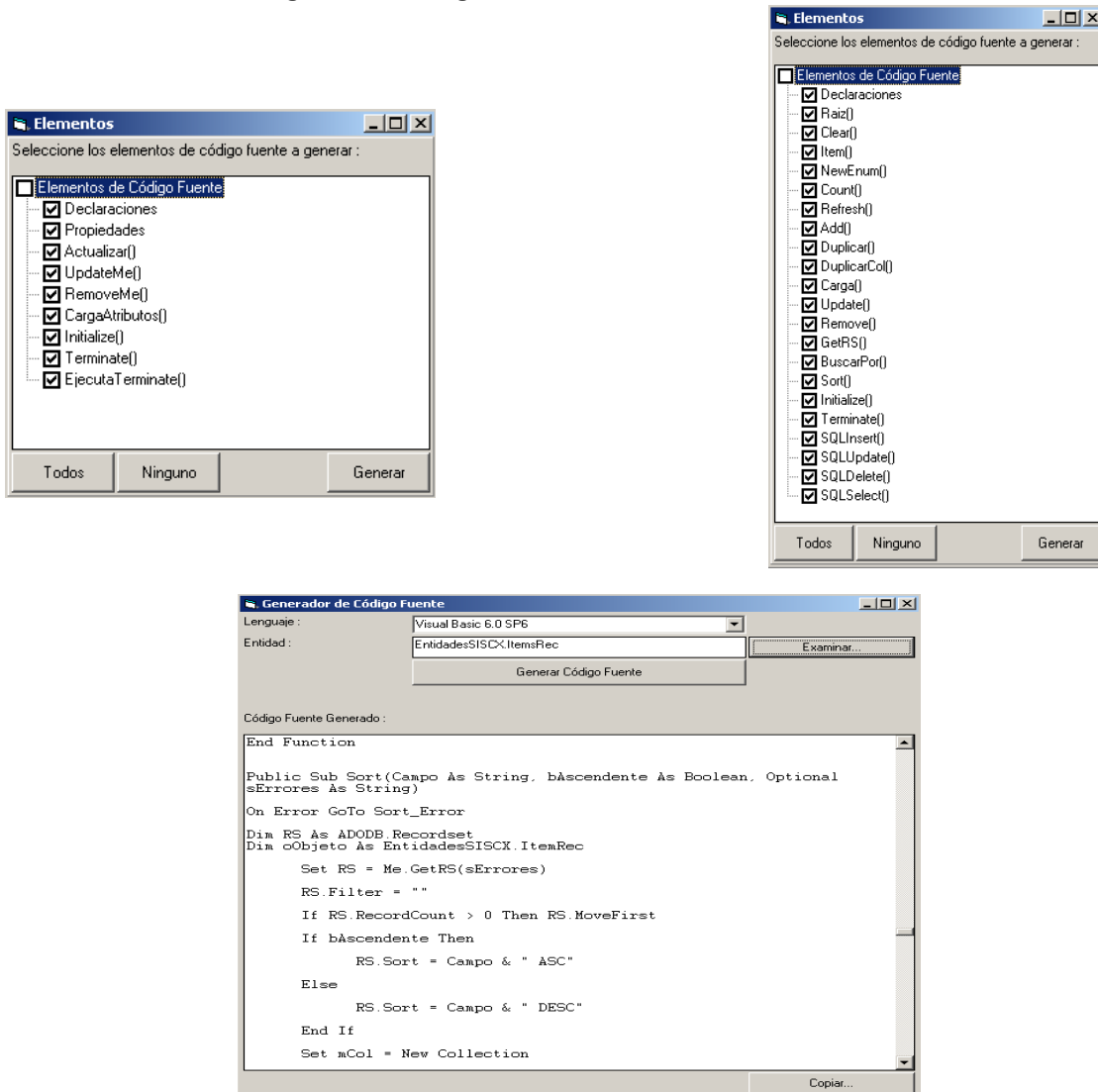


Figura 7. Generador de código Fuente y código fuente ejemplo.

COMPILACIÓN DE ARTICULOS DE INVESTIGACIÓN OCTUBRE 2010

Red Académica Internacional UADY, UAM, WPI, TAMU, ICA e invitados

El código fuente generado por la herramienta FIUADY-Coder ha sido probado en distintas aplicaciones y ha demostrado ser estable. Sin embargo, no significa que pudiera mejorarse en varios aspectos sobre todo aprovechando más las características de los manejadores de bases de datos que existen en el mercado. Se tiene que tener especial cuidado a la hora de especificar el modelo ya que la herramienta no proporciona la gestión de diagramas en la interface gráfica de usuario (GUI por sus siglas en inglés) donde se pueda definir los modelos de acuerdo a las especificaciones de lenguaje de UML 2.0.

En la figura 8 se muestran los atributos persistentes para las entidades las cuales se definen en la tabla C_ENTIDADES de la base de datos de FIUADY-Coder.

Nombre del campo	Tipo de datos	Descripción
ID	Autonumérico	Identificador único de la entidad
NOMBRE	Texto	Nombre de la entidad
DESCRIPCION	Texto	Descripción de la entidad
TIPO	Número	Tipo de la entidad
COLPADREID	Número	Identificador de la Colección padre a la que pertenece la entidad
PADREID	Número	Identificador de la entidad padre a la que pertenece la entidad
TABLA	Texto	Nombre de la tabla donde persistirán los datos de la entidad
NOMPADRE	Texto	Nombre en código fuente que tiene la entidad padre de la entidad
TIPOENTIDADID	Número	Identificador del tipo de entidad en caso de que la entidad sea una entidad colección
CAMPOPADRE	Texto	Nombre del campo padre en la base de datos para la entidad
NOMCOLPADRE	Texto	Nombre en código fuente de la colección padre de la entidad
PADREALTERNOID	Número	Identificador del Padre alternativo a la que pertenece la entidad
NOMPADREALTERNO	Texto	Nombre en código fuente del padre alternativo
CAMPOPADREALTERNO	Texto	Nombre del campo padre alternativo en la base de datos para la entidad
HERMANOID	Número	Identificador de la entidad hermana de la entidad
NOMHERMANO	Texto	Nombre en código fuente de la entidad hermana
CAMPOHERMANO	Texto	Nombre del campo hermano en la base de datos para la entidad
ASTERISCOENSELECT	Sí/No	Indica si se utiliza el asterisco en la consulta de SQL para la selección de datos

Figura 8. Atributos persistentes de las entidades en FIUADY-Coder

En la figura 9 se muestran los atributos persistentes para los atributos los cuales se definen en la tabla C_ATRIBUTOS de la base de datos de FIUADY-Coder.

Nombre del campo	Tipo de datos	Descripción
ID	Autonumérico	Identificador único del Atributo
NOMBRE	Texto	Nombre del Atributo
DESCRIPCION	Texto	Descripción del Atributo
TIPO	Número	Tipo del Atributo de acuerdo a la especificación de código Visual Basic 6.0
LONGITUD	Número	En caso de que el atributo sea de tipo cadena esto indica la longitud máxima
AMBITO	Número	Indica si es Público o Privado.
MODO	Número	Indica si es Propiedad o solo variable
ENTIDADID	Número	Identificador de la entidad que sirve como tipo del atributo
ENUMERACIONID	Número	Identificador de la enumeración de este atributo en caso de ser de tipo enumeración
NOMBRECAMPO	Texto	Nombre del campo en la base de datos en caso de ser un atributo persistente
PREDETERMINADO	Texto	Indica el valor predeterminado del atributo
ESPERSISTENTE	Sí/No	Indica si el atributo es persistente
ATRIBUTOREFID	Número	Atributo Referenciado cuando se trata de un Tipo Entidad
ENZERONULL	Sí/No	Indica que cuando se agregue a la base de datos a través de una consulta si el valor es cero se coloque en vez NULL
FECHALARGA	Sí/No	Indica si el formato de fecha es largo en caso de que el atributo sea de tipo fecha

Figura 9. Atributos persistentes de los atributos en FIUADY-Coder

En la figura 10 se muestran los atributos persistentes para las enumeraciones los cuales se definen en la tabla C_ENUMERACIONES de la base de datos de FIUADY-Coder.

Nombre del campo	Tipo de datos	Descripción
ID	Autonumérico	Identificador único de la enumeración
NOMBRE	Texto	Nombre de la enumeración
DESCRIPCION	Texto	Descripción de la enumeración

Figura 10. Atributos persistentes de las enumeraciones en FIUADY-Coder

En la figura 11 se muestran los atributos persistentes para los ítems de enumeración los cuales se definen en la tabla C_ITEMSENUMERACION de la base de datos de FIUADY-Coder.

Nombre del campo	Tipo de datos	Descripción
ID	Autonumérico	Identificador único del ítem de la enumeración
ENUMERACIONID	Número	Indica la enumeración a la que pertenece el ítem de enumeración
NOMBRE	Texto	Nombre del ítem de la enumeración
VALOR	Texto	Valor del ítem de la enumeración
DESCRIPCIONINT	Texto	Descripción del ítem de la enumeración
ORDEN	Número	Orden del ítem de la enumeración

Figura 11. Atributos persistentes de los ítems de enumeración en FIUADY-Coder

En la figura 12 se muestran los atributos persistentes para los atributos de entidad los cuales se definen en la tabla C_ATRIBUTOSENTIDAD de la base de datos de FIUADY-Coder.

Nombre del campo	Tipo de datos	Descripción
ID	Autonumérico	Identificador único del Atributo de Entidad
ENTIDADID	Número	Identificador de la entidad a la que pertenece el atributo de Entidad
ATRIBUTOID	Número	Identificador del atributo
ORDEN	Número	Orden del atributo de Entidad
CONSIDERAR	Sí/No	Indica si se considera o no
EVENOCAMBIO	Sí/No	Indica si se incluye en el evento Cambio
DESCRIPCIÓN	Texto	Descripción del Atributo Entidad
INCLUIRENRS	Sí/No	Indica si se incluye en la función GetRS()
ESPERSISTENTE	Sí/No	Indica si es persistente
ALIAS	Texto	Alias en caso de conflicto de nombres
GENEVECAMBIO	Sí/No	Indica si se genera evento cambio cuando el atributo cambie de valor
RUTACOL	Texto	Indica la ruta a la colección en caso de que el atributo sea una entidad
ESPADRE	Sí/No	Indica si el atributo es padre
ESPADREALT	Sí/No	Indica si el atributo es padre eterno
DECENUENCLASE	Sí/No	Indica si se declara la enumeración en el código de la entidad a la que pertenece el Atributo de Entidad
PREFIJOKEY	Texto	Indica el prefijo para la llave del atributo
ENINSERTSQL	Sí/No	Indica si se incluye en la consulta de inserción a la base de datos cuando se agregue la entidad
CONTEXTOID	Número	Indica el identificador del contexto al que pertenece el atributo

Figura 12. Atributos persistentes de los atributos de entidad en FIUADY-Coder

COMPILACIÓN DE ARTICULOS DE INVESTIGACIÓN OCTUBRE 2010

Red Académica Internacional UADY, UAM, WPI, TAMU, ICA e invitados

En la figura 13 se muestran los atributos persistentes para los proyectos los cuales se definen en la tabla C_PROYECTOS de la base de datos de FIUADY-Coder.

Nombre del campo	Tipo de datos	Descripción
ID	Autonumérico	Identificador único del proyecto
NOMBRE	Texto	Nombre del Proyecto
DESCRIPCION	Texto	Descripción del Proyecto
TIPODB	Número	Tipo de Base de Datos donde el proyecto hará persistir la información
OBJETODATOS	Texto	Indica el nombre del objeto genérico de datos para la conexión
FUNEJECUTAR	Texto	Nombre de la función para ejecutar las consultas.

Figura 13. Atributos persistentes de los atributos de proyecto en FIUADY-Coder

En la figura 14 se muestran los atributos persistentes para las entidades de proyecto las cuales se definen en la tabla C_ENTIDADESPROYECTO de la base de datos de FIUADY-Coder.

Nombre del campo	Tipo de datos	Descripción
ID	Autonumérico	Identificador único de la Entidad de Proyecto
PROJECTOID	Número	Identificador único del Proyecto al que pertenece la Entidad de Proyecto
ENTIDADID	Número	Identificador de la Entidad
CONSIDERAR	Sí/No	Indica si se considera o no

Figura 14. Atributos persistentes de los atributos de entidades de proyecto en FIUADY-Coder

CONCLUSIONES Y SUGERENCIAS

FIUADY-Coder representa un avance práctico en la manera en la que se desarrollan los paquetes computacionales, logrando mejor consistencia y eliminando los errores humanos de tecleo de código. Por otro lado, esta herramienta permite que el desarrollador se concentre en otras capas tales como la de la interface de usuario para invertir más tiempo y recursos para la usabilidad del programa y también para ponerle mayor énfasis en la capa de reglas de negocios.

Los atributos de las entidades pueden reutilizarse en otras entidades para otros modelos, haciendo así reutilizable mucho del código que se tendría que repetir su escritura haciendo el proceso manual.

Este paquete computacional también permite que en cualquier momento el modelo se vuelva a convertir en código fuente, por lo que cualquier cambio realizado se refleje en el código fuente de la aplicación que se esté desarrollando.

Se sugiere incorporar la capacidad de gestionar diagramas en la GUI para realizar el modelo de manera diagramática según los lineamientos de UML 2.0

También se sugiere adoptar otros lenguajes tales como: Java Script, Visual Basic .NET y C#, para darle mayor amplitud de uso a la herramienta.

Se recomienda utilizar una interfase que permita especificar los diagramas de clases y sus asociaciones para incrementar el índice de usabilidad del paquete computacional.

REFERENCIAS

- 1 Access, Microsoft (2003)
- 2 Case, A. (1985), Computer-aided software engineering (CASE): technology for improving software development productivity, ACM SIGMIS Database, Vol 17- 1, Páginas 35-43.
- 3 Eastman, C. M. Teicholz, P., Sacks, R., and Liston, K. (2008) BIM handbook: A guide to building information modeling for owners, managers, architects, engineers, contractors and fabricators, Wiley Hoboken, N.J.
- 4 Fairbanks, G. (2004) Software Engineering Environment Support for Frameworks
A Position Paper, WoDiSEE workshop at the International Conference on Software Engineering
- 5 Fowler, M. (2003), UML Distilled: A Brief Guide to the Standard Object Modeling Language, Addison-Wesley Professional
- 6 Revit, Autodesk (2011)
- 7 Visual Basic 6.0, Microsoft (1998)
- 8 Windows XP, Microsoft (1998)

ACERCA DEL AUTOR

El M. en Ing. Const. Jesús Nicolás Zaragoza Grifé, estudio la licenciatura en Ingeniería Civil en la Universidad Autónoma de Yucatán, posteriormente obtuvo el grado de Maestro en Ingeniería Construcción en la misma institución. Actualmente es profesor de tiempo completo en la Facultad de Ingeniería de la Universidad Autónoma de Yucatán. Las áreas de interés para el profesor son: el desarrollo de sistemas computacionales, costos de construcción y Building Information Modeling (BIM)

3 **Competer:** es encontrarse, coincidir, pertenecer (no aplicable a nuestros intereses),

Competir: ser adecuado o apto para determinada actividad. Ser competitivo en una tarea, siendo capaz de resolverla de manera