

Tipo de artículo: Artículo original
Temática: Desarrollo de Aplicaciones Informáticas
Recibido: 30/10/2013 | Aceptado: 6/03/2014

Intérprete de programas de usuario para el cálculo de parámetros petrofísicos

Interpreter of user programs for petrophysical parameters calculation

Ridel Oscar García-Mora ^{1*}, Yusleidy Valera-González¹

¹ CEDIN. Departamento de Construcción de Componentes. Universidad de las Ciencias Informáticas, Carretera a San Antonio de Los Baños, km 2 ½, Torrens, Boyeros, La Habana, Cuba. CP.: 19370

* Autor para la correspondencia: rmora@uci.cu

Resumen

La determinación del volumen de hidrocarburos que pueden ser extraídos por un pozo determinado de vital importancia para la industria petrolera. En la evaluación petrofísica de formaciones son empleados de conjunto: los datos tomados directamente en el pozo mediante los registros geofísicos, y aplicaciones informáticas que permiten procesar estos datos para realizar interpretaciones, cálculo de nuevas propiedades que no pueden ser leídas directamente del pozo y visualización de la información para el análisis de la misma. El presente trabajo tiene como objetivo permitir el desarrollo y ejecución de programas de usuario para el cálculo de parámetros petrofísicos en el Sistema de Análisis e Interpretación de Registros de Pozos de Petróleo. Para desarrollar este trabajo se utilizaron métodos científicos como el analítico-sintético, modelado y observación. Además, se utilizó el RUP, lenguaje de modelado UML, la herramienta *CASE Visual Paradigm*, el lenguaje de programación C++ y como marco de desarrollo Qt, y como entorno de desarrollo de *QT Creator*. Como resultado se obtuvo el intérprete de programas de usuario y un editor de código con completamiento y resaltado de sintaxis para hacer el proceso de desarrollo de los programas más amigable al usuario. Con la integración del intérprete al Sistema de Análisis e Interpretación de Registros de Pozos de Petróleo se logra que el mismo se adapte mejor a las necesidades de cada usuario en particular.

Palabras clave: Intérprete, petrofísica, petróleo, programa.

Abstract

Determining the volume of oil that can be extracted for a given well is vital for the oil industry. In the petrophysical evaluation of training set are used : data taken directly in the well using geophysical logs and process applications that allow these data to make interpretations , calculation of new properties that cannot be read directly from the well and display the information for analysis of the same . The present work aims to enable the development and execution of user programs for calculating petrophysical parameters in the System Analysis and Interpretation of Oil Well Records. To develop this work using scientific methods such as analytic-synthetic, modeling and observation. In addition, we used the RUP, UML modeling language, the CASE tool Visual Paradigm, the programming language C and as Qt development framework, and as a development environment for QT Creator. The result was the interpreter of user programs and a code editor with syntax highlighting, completion and to make the development process more user-friendly programs. With the integration of the interpreter to System Analysis and Interpretation of Oil Well Logging is done that it is better suited to the needs of each individual user.

Keywords: *Interpreter; oil; petrophysics; program.*

Introducción

La determinación de parámetros petrofísicos como: el volumen de arcilla, la porosidad total, la porosidad efectiva y la saturación de agua entre otros son esenciales para el cálculo de reservas de petróleo que pueden ser extraídos por un pozo determinado. Los cálculos para determinar los parámetros petrofísicos son realizados utilizando fórmulas matemáticas que son conocidas por los especialistas en petrofísica. Este es un ejemplo del procedimiento para calcular el volumen de arcilla a partir del registro de rayos gamma:

Donde:

$$I_{GR} = \frac{(Gr_{log} - Gr_{min})}{(Gr_{max} - Gr_{min})} \quad (1)$$

Gr_{log} : Valor del registro de rayos gamma a una profundidad dada.

Gr_{max} : Valor del registro Gr frente a una formación arcillosa.

Gr_{min} : Valor del registro Gr frente a una formación limpia de arcilla.

I_{GR} : Índice de rayos gamma.

$V_{sh} = I_{GR}$ (2): Respuesta lineal, estimado de primer orden.

$V_{sh} = 0.08 * (2^{3.7 * I_{GR}} - 1)$ (3): Larionov (1969), rocas del período terciario.

$$V_{sh} = \frac{I_{GR}}{1 - I_{GR}} \quad (4): \text{Steiber (1970).}$$

$$3-2* I_{GR}$$

$$V_{sh} = 1.7-[3.38-(I_{GR} - 0.7)^2]^{0.5} \text{ (5): Clavier (1971).}$$

$$V_{sh} = 0.33*(2^{2*I_{GR}} - 1) \text{ (6): Larionov (1969), rocas antiguas.}$$

Evaluando en la formula (1) los valores de Gr_{log} , Gr_{min} y Gr_{max} para una profundidad dada, se obtiene el valor de I_{GR} y de acuerdo a la formación de rocas presente en la profundidad para la cual se calculó el valor de I_{GR} se evalúa una de las formulas: (2), (3), (4), (5) o (6). Obteniéndose así el valor del volumen de arcilla (V_{sh}) de las formaciones de rocas para la profundidad que se esté evaluando (*Faculty of Mathematics and Natural Sciences*, 2012). Existen varias vías más para calcular el volumen de arcilla, todas con parámetros en las formulas matemáticas que dependen en alto grado de las características de las formaciones de rocas que se estén analizando, y esto no solo sucede con este parámetro petrofísico también pasa con los cálculos de porosidad y saturación de agua y con otros muchos parámetros más no mencionados en este trabajo. En varias soluciones de software analizadas se pudo apreciar que son capaces de comunicarse con programas externos desarrollados por los usuarios usando lenguajes de programación como C++, Pascal y Fortran para calcular parámetros petrofísicos (Senergy, 2007). También la *suite ofimática Microsoft Office* le brinda la posibilidad al usuario de escribir macros en *Visual Basic* para agregar funcionalidades a los documentos, así como modificar su información. Es por lo antes expuesto que al dar una solución para el cálculo de los parámetros petrofísicos en un sistema de análisis e interpretación de registros de pozos de petróleo, además de proveer un módulo con modelos matemáticos predefinidos para el cálculo, es necesario darle la posibilidad al usuario de introducir sus propios modelos. Los modelos matemáticos son introducidos por el usuario mediante la creación de pequeños programas que al ser ejecutados acceden a la información que tiene almacenada el sistema en memoria para realizar los cálculos y la modifican para guardar los cambios de la nueva información generada como resultado de su ejecución.

A partir de que el Sistema para el Análisis e Interpretación de Registros de Pozo de Petróleo (AnPer) no cuenta con un módulo para la ejecución de programas de usuario, se plantea el diseño e implementación del mismo. Este módulo debe permitirle al usuario desarrollar programas para calcular parámetros petrofísicos empleando un lenguaje de programación de fácil uso. Además el módulo debe tener tiempos de respuesta eficientes y bajo consumo de memoria, permitiendo que los usuarios puedan desarrollar sus programas de manera rápida y sencilla.

Materiales y métodos

Conceptos asociados al tema abordado:

El registro de pozo es una grabación contra profundidad de alguna de las características de las formaciones de atravesadas, hechas por aparatos de medición (herramientas) en el hoyo del pozo (SCRIB, 2012), que son estudiadas y analizadas en la Petrofísica para conocer más de su naturaleza.

Las características de las rocas grabadas en los registros de pozos brindan una información limitada acerca del mismo por lo que se hace necesario calcular otras características a partir de las que ya son conocidas.

La porosidad se define como el volumen de poros por cada unidad volumétrica de formación, o sea, es la fracción del volumen de total de una muestra que es ocupada por poros o huecos, es símbolo de la porosidad es Φ o PHI y se expresa en fracciones (v/v) o en porcentaje (%). Generalmente, los cálculos se realizan utilizando los valores en forma fraccionaria (York, 2011).

El volumen de arcilla se define como el porcentaje neto de arcilla presente en una formación. Este parámetro puede ser determinado a partir de los registros de rayos gamma (GR) o del potencial espontáneo (SP) (Jávita, 2007).

La permeabilidad es la medida de la facilidad con que los fluidos que fluyen a través de una formación. La unidad de la permeabilidad es el Darcy que se define como: la cantidad de fluido que pasa a través de un cm^2 de área de formación en 1 segundo, bajo la acción de una atmósfera de presión, teniendo el fluido una unidad de viscosidad. Si el fluido que pasa es un cm^3 se dice que la permeabilidad es de 1 Darcy. Comúnmente se usa el milidarcy (md) ya que el Darcy es una unidad muy grande (Castillo, 2011).

La saturación de una formación es el porcentaje del volumen poroso ocupado por el fluido en consideración. Por lo tanto, la saturación de agua es la fracción o porcentaje del volumen poroso que contiene agua de formación. La saturación de petróleo o gas es la fracción del volumen poroso que contiene petróleo o gas. Los poros deben saturarse con algún fluido. De esta manera la suma de las saturaciones de los diferentes fluidos contenidos en la roca es igual al 100% (Castillo, 2011).

Soluciones de software analizadas:

Techlog: plataforma de software enfocada a proveer soluciones y crear flujos de trabajos relacionados con la información del pozo, sin importar la dimensión y dominio técnico, en colaboración con las diferentes disciplinas (Carrasco, 2012). *Techlog* está diseñado para ejecutarse en sistema operativo *Windows* y fue desarrollado por la

compañía *Schlumberger* utilizando los lenguajes de programación C++ y *Python*. Además, en su construcción son empleadas las bibliotecas Qt. Esta plataforma permite acceder a datos provenientes de núcleos, registros, imágenes, fluidos entre otros, posibilitando tanto a geofísicos, geólogos y petrofísicos visualizar, analizar, interpretar y editar los datos del pozo en una sola plataforma de software. *Techlog* incorpora además, varios módulos de visualización tanto en 2D como en 3D y una colección extensa de módulos para la interpretación. Requiriendo para su uso de una licencia propietaria, y se recomienda para su buen funcionamiento, 16.0GB de Memoria RAM y un procesador de 8*2.6GHz.

Interactive Petrophysics (IP): software para el análisis e interpretación de registros de pozos, ideal tanto para geólogos como petrofísicos. Está diseñado para ejecutarse en sistema operativo *Windows* y fue desarrollado empleando el *framework* de desarrollo *.NET* por la compañía *Senergy*. Este sistema brinda una variedad de funcionalidades para el análisis petrofísico entre las que se destacan: visualizar la información en varias vistas como: histogramas, *crossplot* y pistas, procesamiento de registro de imagen y un módulo de análisis, que cubre por completo el flujo de trabajo de procesamiento de datos, para el análisis estadístico de los resultados. Además, permite la ejecución de programas de usuario y la estimación de propiedades petrofísicas a partir de interpretaciones. “Su principal funcionalidad es el motor de interpretación gráfica, que permite al usuario realizar una rápida interpretación de múltiples zonas, mediante el ajuste de parámetros en gráficos de registro, *crossplots* e histogramas” (Senergy, 2008). IP requiere para su uso de una licencia propietaria, y se recomienda para su buen funcionamiento, 1.0GB de Memoria RAM y un procesador de doble núcleo con 2.6GHz.

Petrolog: software especializado en la información del registro de pozo, y en el análisis de esta información, mediante imágenes que permitan realizar muchas de las tareas asociadas a la gestión y evaluación de los datos de registros de pozos. *Petrolog* está diseñado para ejecutarse en sistema operativo *Windows* y fue desarrollado empleando el *framework* de desarrollo *.NET* utilizando lenguaje C#. Este sistema brinda una variedad de funcionalidades para el análisis petrofísico como son gestión de datos discretos, editor gráfico, minería de datos, procesamiento de múltiples capas, análisis de vistas rápidas (*Quicklook*), análisis de múltiples minerales, análisis de Redes Neuronales, y gráficos para todos los tipos de registros: histogramas, diagramas cruzados (2D y 3D) y proyecciones estereográficas. *Petrolog* requiere para su uso de una licencia propietaria y se recomienda para su buen funcionamiento, 2.0GB de Memoria RAM y un procesador Intel 3.2 GHz.

Métodos teóricos:

Análítico-Sintético: Se utiliza para analizar teorías y elementos bibliográficos relacionados con los intérpretes y los lenguajes de programación, permitiendo la extracción de los elementos más importantes que dan inicio a la investigación.

Modelación: Es utilizado para representar gráficamente la solución que se propone.

Métodos empíricos:

Observación: Se emplea para estudiar las características y comportamientos de las soluciones similares, permitiendo obtener información relevante sobre el proceso de interpretación de programas.

Metodología de desarrollo RUP (Proceso Unificado de Desarrollo)

Es una metodología tradicional de desarrollo de software guiado por los casos de uso, centrado en la arquitectura, iterativo e incremental diseñado como un marco para los métodos y herramientas de UML (Pressman, 2008). El Proceso Unificado de Desarrollo define cuatro fases entre las que se encuentra:

- 1) inicio que abarca la comunicación con el cliente y las actividades de planeación, y destaca el desarrollo y el refinamiento de casos de uso como un modelo primario.
- 2) elaboración que abarca la comunicación con el cliente y las actividades de modelado con un enfoque en la creación de modelos de análisis y diseño, con énfasis en las definiciones de clase y representaciones arquitectónicas.
- 3) construcción que refina y después traduce el modelo de diseño en componentes de software implementados.
- 4) transición que transfiere el software del desarrollador al usuario final para realizar las pruebas y obtener la aceptación.

RUP define nueve disciplinas a realizar en cada fase del proyecto, entre las que se encuentra seis de procesos como son: Modelado del negocio, Análisis de requisitos, Análisis y Diseño, Implementación, Prueba y Distribución. Además, cuenta con tres de soporte entre las que esta Gestión de configuración y cambios, Gestión del proyecto y Gestión del entorno. En esta metodología se define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto.

Características principales de RUP:

- Está formado por componentes de software.
- Unifica los mejores elementos de metodologías existentes.
- Preparado para desarrollar grandes y complejos proyectos.

- Orientado a Objetos.
- Utiliza el Lenguaje Unificado de Modelado (UML) como lenguaje de representación visual.

Lenguaje Unificado de Modelado (UML)

UML es un lenguaje estándar de modelado visual que se usa para especificar, construir, documentar y visualizar artefactos de un sistema de software. UML permite a los desarrolladores visualizar el resultado de su trabajo en esquemas o diagramas estandarizados. Por ejemplo, símbolos o iconos característicos utilizados para capturar los requisitos. Estos iconos no son más que una notación gráfica, es decir, una síntesis; sin embargo detrás de esta notación gráfica, UML especifica un significado, es decir, una semántica.

Dicho lenguaje proporciona un vocabulario que incluye tres categorías: elementos, relaciones y diagramas, conteniendo aspectos conceptuales tales como procesos de negocios, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables (Jacobson, *et al.*, 2000).

Visual Paradigm

Herramienta CASE (del inglés, *Computer Aided Software Engineering*) con licencia gratuita, que propicia un conjunto de ayudas para el desarrollo de programas informáticos dando soporte al modelado visual con UML (del inglés, *Unified Modeling Language*), desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación (*International, Visual Paradigm*, 2008).

Visual Paradigm ofrece:

- Entorno de creación de modelos conformes a UML.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales Entornos de Desarrollo Integrado (IDE).
- Disponibilidad en múltiples plataformas.
- Extensible mediante desarrollo de nuevos Módulos (*plug-ins*).

Lenguaje de programación C++:

C++ es un lenguaje imperativo, orientado a objetos, derivado de C. Al igual que C; C++ está muy ligado al *hardware* subyacente, manteniendo una considerable potencia para la programación a bajo nivel; pero se le han añadido elementos que permiten también un estilo de programación con alto nivel de abstracción. Debido a esto; C++ brinda la posibilidad de crear clases, plantillas, sistema de espacios de nombres y funciones en línea, posee un mecanismo para el manejo de excepciones, permite la sobrecarga de operadores y utiliza operadores para el manejo de memoria. Este lenguaje de programación está estandarizado por la Organización Internacional de Estándares (ISO) y cuenta con una biblioteca estándar de alta calidad (Stroustrup, 2013).

Framework (marco de trabajo) de desarrollo:

Qt es un *framework* multiplataforma, que se utiliza para el desarrollo de aplicaciones, está escrito en C++; sin embargo, es posible utilizarlo con otros lenguajes como C#, PHP, Python, y Ruby. Qt ofrece bibliotecas de código para: creación de interfaces gráficas de usuario, acceso a bases de datos, manipulación de contenido XML (*eXtensible Markup Language*), comunicación en red, visualización con *OpenGL*, entre otras. Además; extiende el lenguaje de programación C++, a través de macros y meta-información, agregando nuevas características como: el bucle *foreach*, la sentencia *forever* e introspección (Digia, 2012).

Qt Script (módulo del framework Qt):

QtScript es un motor de secuencias de comandos que ha formado parte del conjunto de herramientas Qt desde la versión 4.3. El lenguaje de programación se basa en el ECMAScript (ECMA International, 2013) con un par de desviaciones y extensiones. Esta biblioteca cuenta con el motor, y una API C++ para evaluar código *QtScript*, este módulo permite exponer métodos y objetos de clases derivadas de *QObject* para que puedan ser utilizadas en el código *QtScript* (Digia, 2012).

Metodología Computacional

Antes de comenzar a desarrollar el intérprete fue necesario realizar un levantamiento de la información del sistema que debía poder ser accedida y/o modificada desde los programas de usuario.

Información del sistema que debe ser accesible desde los programas de usuario:

- Profundidad inicial de lectura del fichero de registro de pozo.

- Profundidad final de lectura del fichero de registro de pozo.
- Paso de lectura del fichero de registro de pozo.
- Valor nulo del fichero de registro de pozo.
- Cantidad de curvas que contiene el fichero de registro de pozo.
- Índice que ocupa una curva determinada en el fichero de registro de pozo.
- Nombre de una curva determinada.
- Unidad de medida de una curva determinada.
- Código de una curva determinada.
- Descripción de una curva determinada.
- Valor mínimo de una curva determinada.
- Valor máximo de una curva determinada.
- Arreglo de valores de una curva determinada.
- Cantidad de parámetros que contiene el fichero de registro de pozo.
- Índice que ocupa un parámetro determinado en el fichero de registro de pozo.
- Nombre de un parámetro determinado.
- Unidad de medida de un parámetro determinado.
- Valor de un parámetro determinado.
- Descripción de un parámetro determinado.
- Cantidad de conjuntos de zonas que contiene el fichero de registros de pozo.
- Índice que ocupa un conjunto de zonas determinado dentro del fichero de registro de pozo.
- Nombre de un conjunto de zonas determinado.
- Enlace de un conjunto de zonas determinado.
- Tipo de un conjunto de zonas determinado.
- Cantidad de zonas que contiene un conjunto de zonas determinado.
- Índice que ocupa una zona determinada en un conjunto de zonas.
- Profundidad inicial de una zona determinada en un conjunto de zonas.
- Profundidad final de una zona determinada en un conjunto de zonas.

Información del sistema que se puede modificar desde los programas de usuario:

- Valor nulo del fichero de registro de pozo.

- Nombre de una curva determinada.
- Unidad de medida de una curva determinada.
- Código de una curva determinada.
- Descripción de una curva determinada.
- Arreglo de valores de una curva determinada.
- Nombre de un parámetro determinado.
- Unidad de medida de un parámetro determinado.
- Valor de un parámetro determinado.
- Descripción de un parámetro determinado.
- Nombre de un conjunto de zonas determinado.
- Tipo de un conjunto de zonas determinado.
- Enlace de un conjunto de zonas determinado.
- Nombre de una zona determinada en un conjunto de zonas.
- Profundidad inicial de una zona determinada en un conjunto de zonas.
- Profundidad final de una zona determinada en un conjunto de zonas.

Además de los requerimientos de información expuestos anteriormente también se definieron algunas funciones que permitieran que el proceso de codificación fuera más amigable al usuario:

Funciones de utilidades:

- Retornar arreglo de valores nulos.
- Retornar posición en el arreglo de valores de curva de un valor de profundidad.
- Adicionar una nueva curva al fichero de registro de pozo.
- Eliminar una curva del fichero de registro de pozo.
- Adicionar un parámetro al fichero de registro de pozo.
- Eliminar un parámetro del fichero de registro de pozo.
- Adicionar un conjunto de zonas al fichero de registro de pozo.
- Eliminar un conjunto de zonas del fichero de registro de pozo.
- Adicionar una zona a un conjunto de zonas existente.

- Eliminar una zona de un conjunto de zonas existentes.

Posteriormente cada uno de estos requerimientos de información fue definido como una función C++ con un prototipo específico que define *QtScript* para que pueda ser accesible desde el intérprete. El siguiente es un ejemplo de la función que devuelve la profundidad inicial de lectura del fichero de registro de pozo:

```
QScriptValue JavaScriptCodeExecuter::getRecordStartDepth(QScriptContext *context, QScriptEngine *engine)
{
    QScriptValue v;
    Q_UNUSED(engine)
    if(!testArgumentCount(context,0))
    {
        v = record->getStartDepth();
    }
    return v;
}
```

Figura 1. Código C++ de la función que devuelve la profundidad inicial de lectura del registro de pozo.

Para registrar una función como la anterior en el *QScriptEngine* se hace de la siguiente manera:

```
engine->globalObject().setProperty("getRecordStartDepth",engine->newFunction(getRecordStartDepth,0));
```

Figura 2. Código C++ para registrar una función *QScriptEngine*.

Resultados y discusión

Como primer resultado de esta investigación se realizó un estudio del estado del arte de los módulos para el cálculo de parámetros petrofísicos, permitiendo conocer las herramientas, lenguajes de programación y requerimientos de *hardware* de las soluciones informáticas existentes en el mundo. Además se realizó un modelo de negocio empleando diagrama de dominio que permitió realizar con mayor facilidad la ingeniería de requisitos lo que permitió capturar un total de 62 requisitos funcionales y 2 requisitos no funcionales de rendimiento. A partir de los requisitos funcionales y de los casos de uso se realizó el diagrama de diseño del módulo. Debido a la utilización del módulo *QtScript* en el desarrollo del intérprete, la solución cuenta con un número pequeño de clases lo cual puede apreciarse en la figura 4, esto redujo el tiempo de desarrollo en 75 % con respecto a haber desarrollado una solución que contara con su propio lenguaje. Se desarrolló un editor de código con completamiento y resaltado de sintaxis para hacer más amigable al usuario el proceso de escritura de los programas (Figura 3).

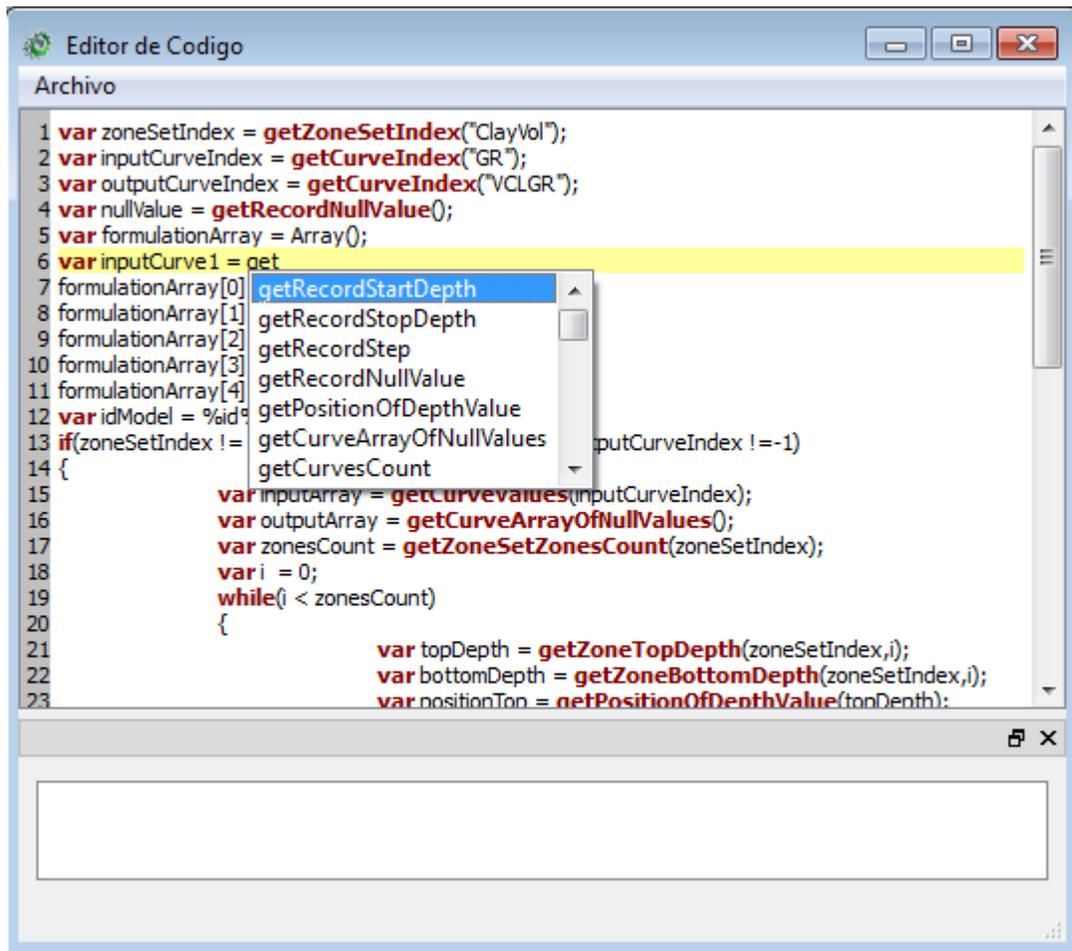


Figura 3. Editor de código.

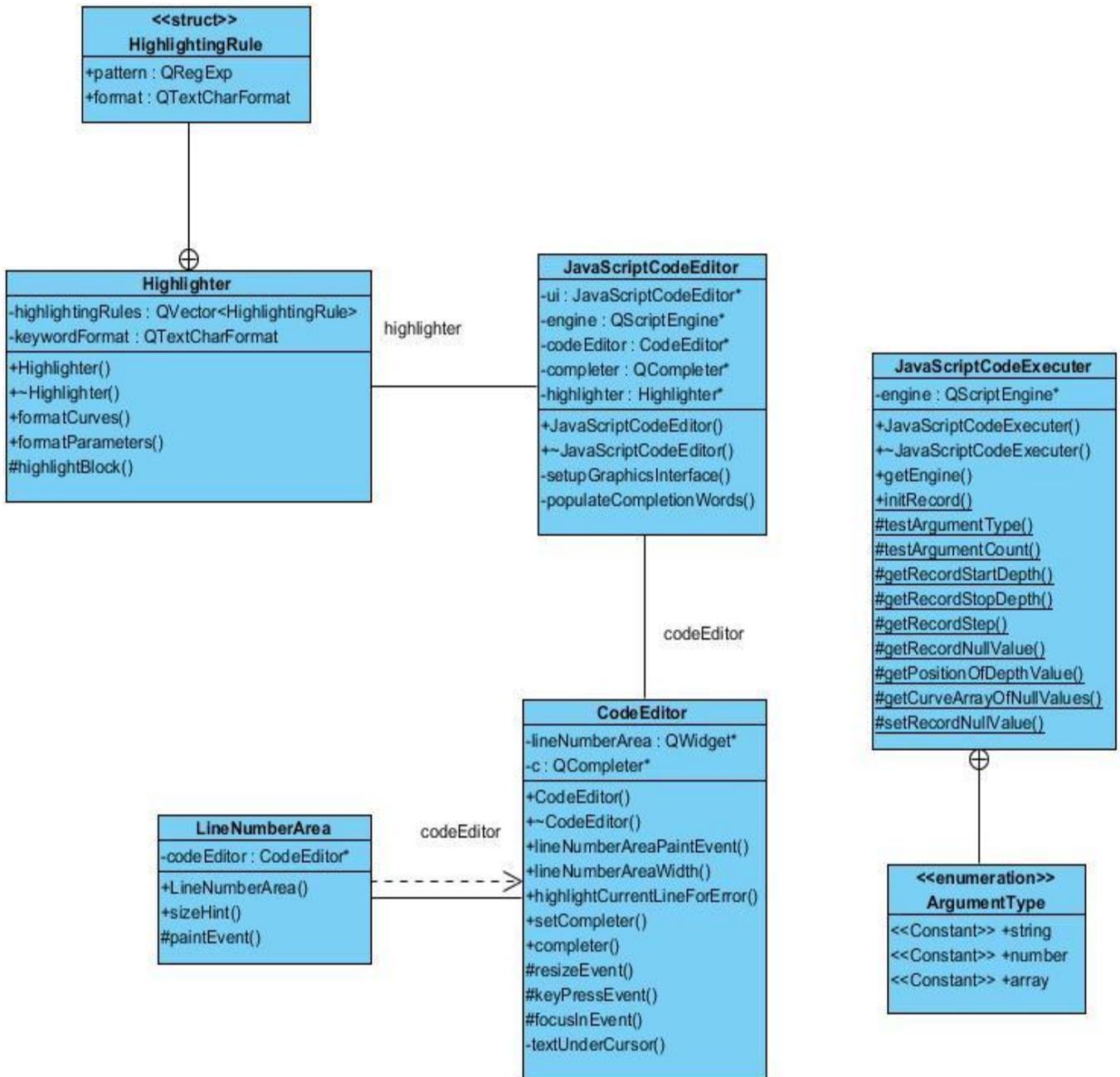


Figura 4. Diagrama de clases del intérprete.

Conclusiones

Con el desarrollo del presente trabajo se logró crear un intérprete de programas de usuario para el cálculo de parámetros petrofísicos, empleando herramientas y tecnologías libres lo cual contribuye a alcanzar la soberanía tecnológica de nuestro país. Esta investigación profundiza en una de las diferentes variantes que se pueden adoptar para hacer que una aplicación construida empleando las librerías Qt sea modificable por el usuario a través de *scripting*. La solución desarrollada como parte de este trabajo le permite al Sistema para el Análisis e Interpretación de Registros de Pozos de Petróleo (AnPer) contar con una mayor flexibilidad para calcular parámetros petrofísicos, permitiendo que el sistema pueda ser utilizado para evaluar formaciones de rocas para las cuales no cuenta con modelos matemáticos definidos.

Referencias

- CARRAZCO, J. M. *Una solución dedicada al pozo para el ciclo de vida de un activo*. México DF : Centro de Publicaciones de Pemex, 2012. p. 22-27.
- CASTILLO, J. JIMENEZ. *Registros Geofísicos*. Mexico: Grupo CEDIP, 2011. p. 25-35.
- DIGIA. 2012. acerca de Qt. *Qt*. [en línea] Digia, 2012 [Consultado el: 3 de octubre de 2013]. Disponible en: [\[http://qt.digia.com/Product/\]](http://qt.digia.com/Product/).
- QT PROJECT. *Qt Project*. [en línea] Digia, 2012 [Consultado el: 25 de septiembre de 2013]. Disponible en: [\[http://qt-project.org/doc/qt-5.0/qtscript/qtscript-index.html\]](http://qt-project.org/doc/qt-5.0/qtscript/qtscript-index.html).
- ECMA International. ECMAScript Language Specification. [en línea] 2011 [Consultado el: 4 de octubre de 2013]. Disponible en: [\[http://www.ecma-international.org/publications/standards/Ecma-262.htm\]](http://www.ecma-international.org/publications/standards/Ecma-262.htm).
- Faculty of Mathematics and Natural Sciences. 2010. Porosity Logs. *University of Oslo*. [en línea] 2010 [Consultado el: 6 de octubre de 2013]. Disponible en: [\[http://www.uio.no/studier/emner/matnat/geofag/GEO4250/v08/undervisningsmateriale/Lectures/BWLA%20-%20Porosity%20Logs.pdf\]](http://www.uio.no/studier/emner/matnat/geofag/GEO4250/v08/undervisningsmateriale/Lectures/BWLA%20-%20Porosity%20Logs.pdf).
- International, Visual Paradigm. 2008. Visual Paradigm. *Visual Paradigm*. [en línea] Visual Paradigm International, 2008 [Consultado el: 28 de septiembre de 2013]. Disponible en: [\[http://www.visual-paradigm.com/product/vpum/features/\]](http://www.visual-paradigm.com/product/vpum/features/).
- JACOBSON, Ivar, BOOCH, GRADY y RUMBAUGH, J. *El Proceso Unificado de Desarrollo de Software*. Madrid: Addison-Wesley, 2000. p. 407-418. ISBN: 84-7829-036-2.

- JÁVITA, T. A. Y. REVELO. *Actualización de Reservas y Ubicación de Pozos de Desarrollo en el Campo Tapi-Tetete*. Quito: Escuela Politécnica Nacional, 2007. p. 21-28.
- Pressman, R. S. *Ingeniería del Software*. Nueva York: McGraw-Hill, 2008. p. 86-88. ISBN 970-60-5473-3.
- SCRIB. 2012. SCRIB. *SCRIB*. [en línea] 2012 [Consultado el: 15 de septiembre de 2013]. Disponible en: [<http://es.scribd.com/doc/26228291/Registro-o-Perfilaje-de-Pozos>].
- SENERGY. 2007. *Interactive Petrophysics Users Manual*. Kincardneshire: PGL, 2007. p. 1257-1302.
- SENERGYWORLD. 2008. *SenergyWorld*. [en línea] Senergy, 2008. [Consultado el: 5 de octubre de 2013]. Disponible en: [<http://www.senergyworld.com/software/interactive-petrophysics>].
- STROUSTRUP, BJARNE. *The C++ Programming Language*. New York: Addison-Wesley, 2013. p. 13-16. ISBN 978-0321563842.
- YORK, E. M. BISBÉ. 2011. *Elementos Básicos de Geofísica de Pozos*. La Habana: Empresa de Extracción y Perforación de Petróleo Occidente, 2011. p. 15-19.