

Evaluador de calidad de escritura de código fuente

ESPINOZA-GALICIA, Carlos*†, GÓMEZ-LÓPEZ, Williams y REYES-LÓPEZ, Rubén

Instituto Tecnológico Superior de Huichapan, El Saucillo S/N CP 42400, Huichapan, Hidalgo, México

Recibido Abril 13, 2017; Aceptado Junio 16, 2017

Resumen

En las empresas de desarrollo de software, la calidad de los proyectos realizados es de suma importancia. Esta calidad implica que el objetivo del software sea cumplido y también que el código fuente sea mantenible, escalable y reutilizable, por lo cual es de interés contar con un evaluador de calidad de escritura de código fuente, dicho proyecto evalúa que se usen las buenas prácticas que se requieren para el desarrollo del proyecto, evaluando la sintaxis que el programador utiliza para escribir y nombrar paquetes, clases, métodos y variables que son comparadas contra reglas preestablecidas para esto. Como resultado emite una serie de reportes los cuales evalúan la calidad de un proyecto, módulo o incluso ser utilizada por un desarrollador, fungiendo esto como una herramienta para el área de Calidad y el mismo desarrollador. El software fue escrito en Visual C# usando una base de datos NoSQL. Hasta el momento se ha empleado en proyectos escolares siendo de utilidad para que los mismos alumnos mejoren la calidad del software escrito y como herramienta de evaluación para el docente. Trabajos futuros contempla el mejoramiento del proyecto, inclusión de otros lenguajes y su utilización con proyectos más grandes, así como su interacción con Git.

Buenas Practicas, Calidad de Software, NoSQL

Abstract

In software development companies, the quality of the projects carried out is of the utmost importance. This quality ensures the fulfilment of the software objective, the maintainability, scalability and reusability of the source code. That is why it is of high interest to have a source code writing quality evaluator. This project evaluates good practices that are required for the project through the assessment of the syntax that the developer uses in writing and naming packages, classes, methods and variables, and then compare them against pre-established rules for the development. As a result, it gives out a series of reports that evaluate the quality of a project, module or even of a developer himself, turning out to be a useful tool for the Quality area and for the developer. The software was written in Visual C # using a NoSQL database. So far, this quality evaluator has been used in school projects and turned out to be a useful tool for students because it helped them improve the quality of their written software; but it was also used as an evaluation tool for teachers. Future work includes the improvement of the project, inclusion of other programming languages and their use with larger projects, as well as their interaction with Git.

Good Practices, Software Quality, NoSQL

Citación: ESPINOZA-GALICIA, Carlos, GÓMEZ-LÓPEZ, Williams y REYES-LÓPEZ, Rubén. Evaluador de calidad de escritura de código fuente. Revista de Tecnología Informática 2017, 1-1: 30-35

* Correspondencia al Autor (Correo Electrónico: cespinoza@iteshu.edu.mx)

† Investigador contribuyendo como primer autor.

Introducción

La evolución de la tecnología ha dado paso al crecimiento de distintos mercados a nivel mundial, principalmente el desarrollo de software el cual creció 226% dentro del sector de Tecnologías de la Información y Comunicación en México hasta el año 2012. (González Bañales, 2016)

El desarrollo de software es un área de la ingeniería en sistemas computacionales en donde hay una gran cantidad de técnicas para lograr un producto de calidad.

Existen técnicas de desarrollo como eXtream Programming, Scrum, Cascada, entre otras; también metodologías como la de desarrollo en capas e incluso patrones de diseño (Bishop, 2008) como MVC o MVVM que permiten un desarrollo más ordenado, todo esto y muchas otras cosas con en el fin de que el resultado sea el esperado por el cliente y que además pueda ser fácilmente extensible y mantenible.

Sin embargo, todo esto es para cumplir con las exigencias del cliente, pero que pasa cuando dicho software es realizado por un grupo grande de desarrolladores, los cuales están creando y editando parte del código fuente, los cuales, para que esto se de en forma adecuada, han surgido los estándares de programación que son reglas que cada empresa u organización define para la escritura del código fuente en el desarrollo del software.

Si bien los estándares de programación dan pauta de las reglas a seguir, esto no implica que los desarrolladores lo sigan fielmente, entonces surge la pregunta ¿cómo saber si los desarrolladores siguen el estándar de programación? y otra tal vez más importante, ¿se puede evaluar a los desarrolladores en el uso del estándar definido por la empresa?; justo aquí es donde surge el interés y el objetivo del evaluador de calidad de código escrito.

La forma en que se presenta el artículo es el siguiente: Metodología, apartado en donde se menciona el desarrollado y los resultados del mismos en su primera fase y como última sección las conclusiones y referencias.

Objetivo

Desarrollar un evaluador de calidad de escritura de código basado en buenas prácticas para aquellas personas encargadas de la revisión corrección y mantenimiento de código, y que trabajara a partir de estándares definidos y buenas prácticas de programación para .NET.

Objetivos específicos

- Desarrollar una aplicación que permita visualizar de manera gráfica los resultados de la evaluación del código.
- Darle a la aplicación la capacidad de corregir los errores de escritura de código de manera automática.
- Proveer al usuario un reporte final que certifique la calidad del código.

Metodologia

Para realizar una aplicación que permita la detección, corrección y mantenimiento del código es necesario utilizar algún modelo para el desarrollo de dicho software, en este caso se utiliza el modelo en espiral (Boehm, 1986), donde el esfuerzo del desarrollo es iterativo, es decir, en el punto en que termina un esfuerzo de desarrollo, comienza otro, además en cada ejecución del desarrollo se implementan cuatro pasos (Figura 1):

Determinar objetivos, analizar los riesgos, desarrollar y probar y planificar lo realizado.



Figura 1 Metodología en espiral (Boehm, 1986)

Por lo que, en cada etapa, se realiza un subproyecto que complementa al anterior y donde todos en conjunto, generen el entregable total.

El analizador de calidad de escritura de código permite a los desarrolladores de software agilizar la etapa de revisión de código y mejorar la calidad del software bajo un cierto estándar de programación automatizando el proceso de evaluación y corrección de código permitiendo así el ahorro de recursos destinados a esta tarea así como el tiempo invertido reduciendo también el costo de desarrollo del software y dando como resultado estadísticas, gráficos, mejora directa en el código y posiblemente certificaciones de calidad.

El programa que se desarrolló (llamado CodEval) fue estructurado mediante las siguientes etapas:

1. **Carga de directorio:** el usuario selecciona la ubicación del proyecto para obtener todos los archivos cuya extensión es .cs

2. **Evaluación de código:** al proporcionar un archivo con extensión cs, el software identifica clases, objetos, métodos, variables, etc. para compararlos con un estándar de programación proporcionado.

3. **Detección de errores:** a través de una comparación es capaz de detectar errores en la redacción de código.

4. **Estadísticas e informes:** A partir de la información obtenida en la etapa 2, se generan reportes e informes de interés para el usuario.

5. **Corrección de errores:** Se analiza la información obtenida y proporciona correcciones para el código fuente.

Para el evaluador de calidad de escritura de código se usó ROSLYN donde, de acuerdo a Alessandro Del Sole, proporciona compiladores de código abierto para C# y Visual Basic (Del Sole, 2016), esto con el fin de reconocer fácilmente los nombres de variables, métodos, clases, y demás objetos dentro de la solución.

El analizador lleva a cabo una verificación del código con el fin de encontrar errores referentes a un cierto estándar de programación. Los tipos de errores encontrados se catalogan en:

- Variables locales
- Variables globales
- Nombre de métodos
- Nombre de clases
- Instancias de objetos
- Controles
- Ensamblados

Para este proyecto se contemplan tres fases:

Fase 1: Una aplicación de escritorio. Se desarrollará una aplicación en la cual el usuario podrá cargar un directorio con código de formato .cs para evaluarlo de acuerdo con un estándar establecido o proporcionado, entregando como resultado estadísticas y correcciones de código.

Fase 2: Versión en la nube. Se desarrollará un portal web que podrá realizar la funcionalidad de la aplicación con directorios cargados desde la nube. Por ejemplo, a través de repositorios como GitLab, GitHub, entre otros.

Fase 3: Soporte de más lenguajes de programación. Se pretende extender la evaluación para aplicarse en lenguajes de programación como Python, Ruby, Pascal, PHP, entre otros.

Actualmente el proyecto se encuentra en proceso de transición a la segunda fase, sin embargo, para la primera fase de la solución del proyecto tiene la siguiente estructura:

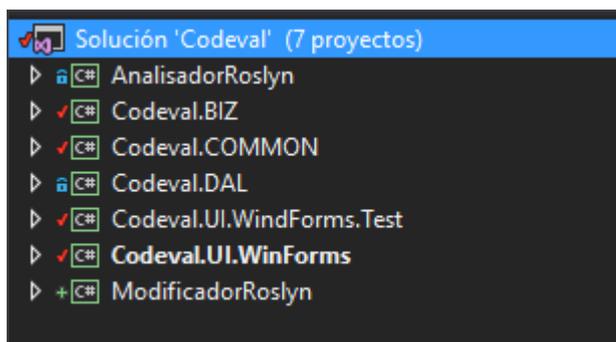


Figura 2 Estructura del proyecto Analizador de Calidad de escritura de código fuente. (Construcción propia)

Como se observa en la Figura 2, la solución presenta una estructura basada en capas (Microsoft Corporation, 2009), además de proyectos de apoyo, los cuales se describen a continuación:

AnalizadorRoslyn: proyecto que realiza la revisión del proyecto a evaluar, después de ejecutar devuelve una lista de Tokens a ser evaluados.

- Codeval.BIZ: presenta la lógica de negocio para el funcionamiento de la aplicación y de la interfaz de usuario
- Codeval.COMMON: contiene la definición de los objetos que se usan en todas las capas.
- Codeval.DAL: Contiene los métodos CRUD (1&1 Internet Inc) para el acceso a la base datos NoSQL (Acens) la cual contiene los datos de comparación para la evaluación
- Codeval.UI.WinForms.Test: presenta pruebas unitarias para la funcionalidad del proyecto.
- Codeval.UI.WinForms: presenta la interfaz grafica del proyecto
- ModificadorRoslyn: Proyecto que realiza la modificación del código según la solicitud de la persona que evalúa el código.

Resultados

Después de un año de trabajo, se tiene la primera fase de la aplicación, la cual toma un directorio seleccionado por el usuario (Figura 3 (1)), verifica todos los archivos de código (Figura 3 (2)) y los compara con la configuración proporcionada con base al estándar de calidad usado por el equipo de desarrollo, presenta un listado de objetos, métodos, clases, propiedades, enumeraciones, etc., que concuerdan y los que no con las reglas introducidas, se da la opción a la persona que evalúa el código de realizar un informe o bien las correcciones pertinentes al código (Figura 3 (3)).

Al final se muestra un reporte de los hallazgos del programa, el cual podrá ser proporcionado al desarrollador para su conocimiento.

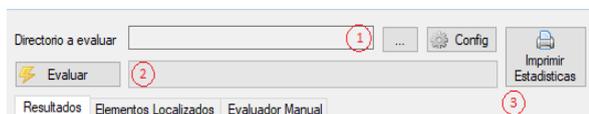


Figura 3 Pantalla principal (parte superior) de Codeval (Construcción propia)

En la Figura 4 se muestra el reporte específico de los errores encontrados después de que el archivo .cs es evaluado por el programa, identificando el archivo, la línea, el tipo, grado y la descripción del error.

Archivo	Línea	Tipo	Impacto	Descripción
Cuenta.cs	15	10	10	Variable mal definida
Cuenta.cs	9	40	20	Doc. incompleta
Moneda.cs	70	50	30	Función NumeroATexto ya existía
CambioDivisa.cs	14	30	30	Utiliza metodo de 2 capas abajo

Figura 4 Reporte de errores generado por Codeval (Construcción propia)

En la figura 5 se observa la cantidad de ocurrencias por el tipo y grado de error.

Desarrollador	Tipo de error										Impacto		
	10	20	30	40	50	60	10	20	30				
1243 Bazarro Chavez Martín	1	1	1	1	1		1	1	3				
1222 Neri Flores Jimena	2	1		4			5	2					
3456 Garcia Jimenez Juan	1		1				2						

Figura 5 Reporte de resumen de errores de Codeval (Construcción propia)

Finalmente, el reporte muestra graficas de los errores encontrados en el archivo, clasificando el tipo de error y el impacto de cada uno.

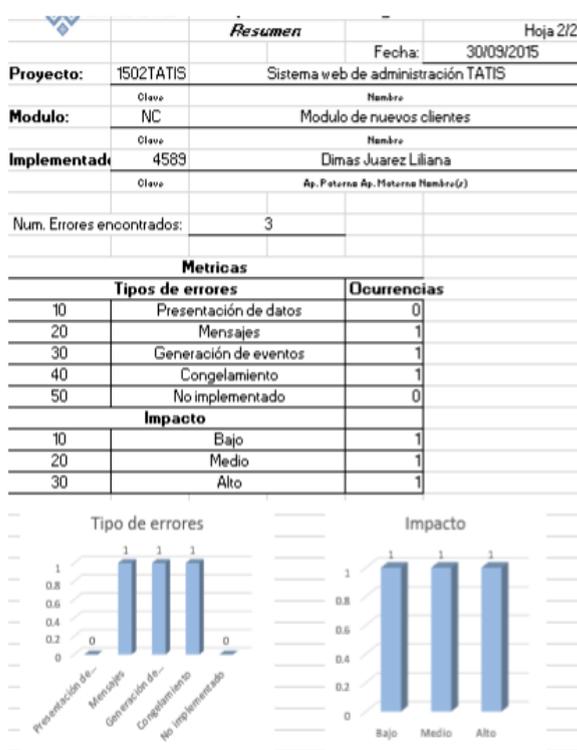


Figura 6 Extracto del Reporte final de tipo de errores y su impacto. (Construcción propia)

Conclusiones y trabajos futuros

El desarrollo de software es una industria muy valiosa dentro del mercado mundial, anualmente se facturan alrededor de 300 mil millones de dólares por lo que es importante procurar éxito en los proyectos de software utilizando estándares y normas de calidad.

Realizar un analizador de código es una solución y automatización a los problemas de codificación bajo estos estándares, permitiendo evaluar si los desarrolladores utilizan el estándar propuesto lo que presupone una calidad en el desarrollo de software generando que el mismo sea mantenible y extensible, incluso por desarrolladores nuevos que no han tenido contacto con el proyecto.

Por el momento se ha usado en proyectos escolares donde se evalúa la correcta escritura de código por parte de estudiantes, sirviendo como una herramienta para mejorar las técnicas de programación de cada alumno.

Como se menciono antes, actualmente se encuentra en la primera fase, como trabajos futuros se pretende llevar este proyecto a la nube y trabajar directamente con repositorios GIT y con otros lenguajes de programación.

Referencias

1&1 Internet Inc. (s.f.). *CRUD: la base de la gestión de datos*. Obtenido de 1&1: <https://www.1and1.mx/digitalguide/paginas-web/desarrollo-web/crud-las-principales-operaciones-de-bases-de-datos/>

Acens. (s.f.). *Bases de datos NoSQL. Qué son y tipos que nos podemos encontrar*. Obtenido de AcensWhitePapers: <https://www.acens.com/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>

Bishop, J. (2008). *C# 3.0 Design Patterns*. Sebastopol, CA: O'Reilly Media, Inc.

Boehm, B. (1986). A Spiral Model of Software Development an Enhancement. *ACM SIGSOFT Software Engineering Notes*, 14-24.

Del Sole, A. (2016). *Rollyn Succinctly*. Morrisville, NC: SynCFusion, Inc.

González Bañales, L. (8 de Julio de 2016). *SG BUZZ Conocimiento para crear Software Grandioso*. Recuperado el 2 de Agosto de 2016, de Un vistazo a la Industria Mundial del Software: <https://sg.com.mx/content/view/213>
Microsoft Corporation. (2009). *Microsoft Application Architecture Guide, Patterns & Practices, 2nd Edition*. Redmond: Microsoft Corporation.