

Desarrollo de una herramienta matemática como apoyo a la educación básica en el estado de Guerrero (México) utilizando Monodevelop con C#

Development of a Mathematic Tool as a Support to the Basic Education in the State of Guerrero, Mexico, Using Monodevelop with C#

Feliciano Morales Severino*

Joel Abisai Serrano Reyes**

Feliciano Morales Angelino***

Fecha de recepción: 8 de agosto del 2012

Fecha de aceptación: 10 de septiembre del 2012

Resumen

Se dan a conocer los resultados de un proyecto de algunos integrantes del Cuerpo Académico de Tecnologías de Información y Comunicaciones de la Unidad Académica de Ingeniería de la Universidad Autónoma de Guerrero, que se está llevando a cabo para apoyar a los estudiantes de educación básica en el estado de Guerrero, en el área de matemáticas; así mismo, fomentar en las instituciones educativas y en los estudiantes el uso de software libre, ya que no genera ningún costo. Actualmente las primarias trabajan en el área de las matemáticas de manera tradicional, sin la ayuda de ninguna herramienta de software que permita desarrollar más habilidades en el área de las matemáticas, por lo que el índice de aprovechamiento es bajo en un gran sector de estudiantes. Con esta aplicación, a la cual se denominó Mathe-

* Unidad Académica de Ingeniería de la Universidad Autónoma de Guerrero. sevefelici@hotmail.com

** Unidad Académica de Ingeniería de la Universidad Autónoma de Guerrero. kalel_krptn@hotmail.com

*** Unidad Académica de Ingeniería de la Universidad Autónoma de Guerrero. af_morales@hotmail.com

mathsLab, se pretende contribuir con el rendimiento académico de los estudiantes.

En este tipo de instituciones educativas solo se utiliza el sistema operativo Windows y Microsoft Office; no se promueve el uso de herramientas diferentes, lo que impide una visión diferente en las nuevas generaciones en cuanto al uso de herramientas de software libre.

Mediante esta aplicación se pretende coadyuvar al incremento del índice de aprovechamiento en el área de las matemáticas; la aplicación servirá de apoyo en el aprendizaje en las operaciones aritméticas de multiplicación, división y fracciones, principalmente.

Se pretende, también, motivar a los estudiantes en el uso del software de fuente abierta, ya que esta aplicación se está desarrollando con este tipo de tecnología.

Para ello se quiere crear una aplicación que se ejecutará en el ambiente Linux (específicamente Ubuntu), lo cual servirá de ayuda en el aprendizaje de los estudiantes. Al mismo tiempo, esto les permitirá conocer el entorno Linux.

Además se describe cómo es que se está desarrollando esta aplicación, mediante el entorno de desarrollo MonoDevelop con el lenguaje C#, es decir, todo en software libre.

Palabras clave: IDE, desarrollo de software, software propietario, software libre, C#, GTK#, programación, portabilidad, paradigmas, multiplataforma, Proyecto Mono, MonoDevelop.

Abstract

The purpose of this article is that of publishing the results of a project implemented by members of the Academic Body of Information Technologies and Communications of the Engineering Academic Unit belonging to the Universidad Autónoma de Guerrero. This project is being carried out to support basic education students to improve their abilities in math, in the state of Guerrero, Mexico; likewise, to incite educative institutions and students to use free software, since this one does not create a cost for its use. Nowadays, in primary education, math is being taught in a traditional manner without the help of software tools that can help improve the ability of the student in this area. Therefore, academic progress is low in a large group of students. That is why, this app, which is known as "MathematicsLAB", aims to contribute to the academic progress of the students.

Besides, educational institutions only use Microsoft operative system and programs, such as Windows Office, not promoting the use of alternative tools, preventing the development of a different viewpoint in new generations, regarding the use of free software tools.

With this project, it is expected to contribute to the increase of the academic progress in the area of mathematics, by the use of this app, which will mainly support the learning of arithmetic operations such as multiplication, division and fractions.

It is also pretended to incite students to use open-source software, since this app is being developed with this kind of technology.

To accomplish that, an app, which will be able of working in a Linux environment (especially Ubuntu), will be developed. When helping students with the learning of math, they will get familiar with a Linux environment.

However, it is going to be describe how this app is being developed, using the IDE known as MonoDevelop and the C# programming language, in other words, this app will be developed using only free software tools.

Key words: IDE, software development, proprietary software, free software, C#, GTK#, programming, portability, paradigms, multiplatform, Mono Project, MonoDevelop.

1. Introducción

1.1. Software

El desarrollo de software es el proceso de diseñar, escribir, depurar y mantener el código fuente de programas computacionales. El código fuente se escribe tradicionalmente en un solo lenguaje de programación y su propósito es crear programas que exhiban cierto comportamiento deseado. El proceso de escribir código requiere frecuentemente conocimientos en varias áreas distintas, además del dominio del lenguaje por utilizar, algoritmos especializados y lógica formal.

En 1970 menos del 1% de las personas hubiera sido capaz de definir el significado de "software". En la actualidad la mayoría de los profesionales y muchos miembros del público creen entender la definición de software. Pero ¿en realidad lo hacen?

Una definición de software en un libro de texto podría ser la siguiente: "el software se forma con 1) las instrucciones que al ejecutarse proporcionan las características, funciones y el grado de desempeño deseados; 2) las estructuras de datos que permiten que manipulen la información de manera adecuada; y 3) los documento que describen la operación y uso de programas".

Existen muchas definiciones para la descripción de software; sin embargo, se requiere una definición formal, y quizá la más apropiada sea la definición dada por el Instituto de Ingenieros en Electricidad y Electrónica (IEEE) en su estándar 729: "es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación".

Tomando en cuenta la anterior definición, es claro que, cuando nos referimos a programas, nos estamos refiriendo a sus estados: código fuente, binario o ejecutable; pero no solamente eso: la documentación, los datos a procesar e incluso la información de usuario forman parte del software. En pocas palabras, software abarca todo lo intangible, todo lo que no es físico [1], [2]. Por lo tanto, el software tiene características muy diferentes a las del hardware:

1. El software se desarrolla o construye; no se manufactura en el sentido clásico.
2. El software no se *desgasta*.
3. A pesar de que la industria tiene una tendencia hacia la construcción por componentes, la mayoría del software aún se construye a la medida.

1.2. El papel evolutivo del software

Para entender claramente cómo es que ahora se plantean diferentes formas de solucionar problemas de automatización por medio del proceso de desarrollo de software, se debe identificar cómo ha evolucionado este proceso.

El software desempeña dos papeles importantes, el de producto y el de vehículo. Como producto ofrece la potencia de cómputo presentada como hardware de una computadora o, de manera más amplia, por una red de computadoras accesibles mediante hardware local. Sin importar dónde se encuentre el software, ya sea en dispositivos móviles como los celulares o dentro de una computadora, este se considera como un transformador; realiza la producción, el manejo, la adquisición, la modificación, el despliegue o la transmisión de información que puede ser tan simple como un bit o tan compleja como una presentación multimedia. En su

papel como vehículo para la entrega de un producto, el software actúa como base para el control de una computadora (sistemas operativos), la comunicación de información (redes) y la creación y control de otros programas (utilerías de software y ambientes).

El software maneja el producto más importante en estos tiempos: la información. Transforma datos de manera que sean útiles en un contexto específico; maneja información de negocios para mejorar su desempeño, pero lo más importante es que proporciona los medios para adquirir información en todas las formas posibles.

En tan solo 50 años el rol del software ha cambiado de manera significativa. El surgimiento de nuevas tecnologías y las mejoras a las tecnologías ya existentes, como el incremento en las capacidades de memoria y almacenamiento, han propiciado el surgimiento de sistemas más elaborados y complejos basados en computadoras.

El desarrollo de software ha crecido tanto que ahora se considera un factor dominante en la economía del mundo industrializado. El programador *solitario*, aquel que por cuenta propia desarrollaba cualquier aplicación que le era solicitada, ha sido reemplazado por equipos de especialistas en software, en los que cada miembro se enfoca en una parte de la tecnología requerida para desarrollar una aplicación compleja.

La preocupación de la industria en cuanto a tiempo de obtención del software, los altos costos de construcción de este, el tiempo y esfuerzo invertido en el mantenimiento de programas ya existentes han conducido a la adopción de prácticas como la ingeniería del software [1], [2].

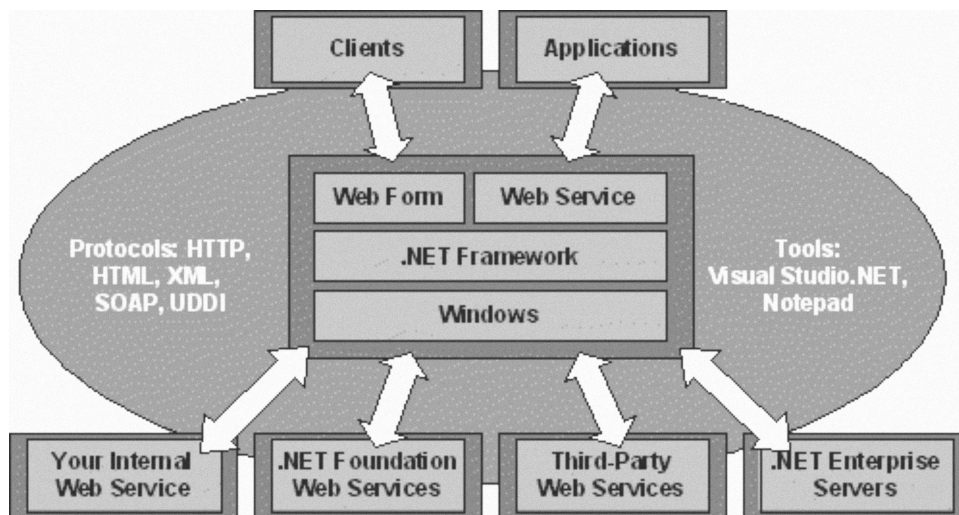
2. Plataforma .NET

Con este paradigma, se trata de que el código generado por un lenguaje pueda funcionar fácilmente con el código generado por otro lenguaje [3].

La plataforma .NET es la propuesta que inició Microsoft (y posteriormente se usó también como *open source*) para competir con la plataforma Java. Esta plataforma está diseñada para que se puedan desarrollar componentes de software utilizando casi cualquier lenguaje de programación, de forma que lo que se escriba en un lenguaje pueda utilizarse desde cualquier otro de la manera más transparente posible (utilizando servicios web como *middleware*). Esto es, en vez de estar limitados a un único lenguaje de programación, se permite cualquier lenguaje de programación, siempre y cuando se adhiera a unas normas comunes establecidas para la plataforma .NET en su conjunto. De hecho, existen compiladores de múltiples lenguajes muy poderosos para la plataforma .NET.

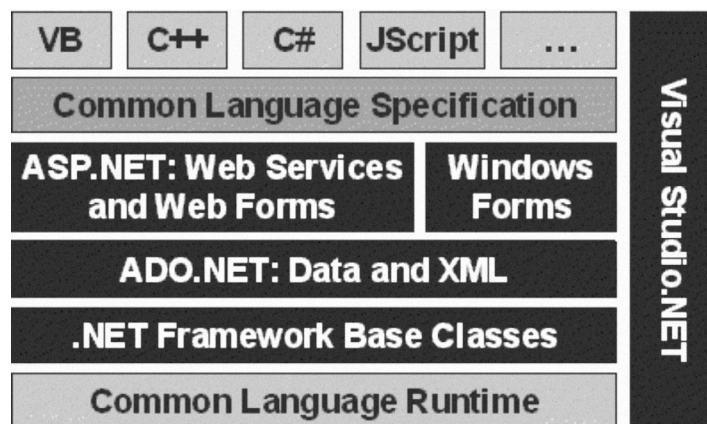
La plataforma .NET apuesta por un futuro en el que las aplicaciones se ejecutan de manera distribuida en internet. Así, una aplicación se ejecuta en un solo servidor y no existen múltiples copias de esta. Además, una misma aplicación puede *adornarse* con distintas interfaces para que, desde diferentes dispositivos (teléfonos móviles, PDA, portátiles, etc.), pueda accederse a esta. La plataforma .NET no es más que un conjunto de tecnologías para desarrollar y utilizar componentes que nos permitan crear formularios web, servicios web y otras aplicaciones en distintos sistemas operativos [4], como se muestra en la figura 1.

Figura 1. Plataforma.NET para Windows



Con el fin de crear aplicaciones para la plataforma .NET, tanto servicios web como aplicaciones tradicionales (aplicaciones de consola, aplicaciones de ventanas, servicios de cualquier sistema operativo, etc.), Microsoft y Mono han publicado el denominado kit de desarrollo de software conocido como .NET Framework, que contiene el CLR (Common Language Runtime), el .NET Framework Clases y características avanzadas como ADO.NET (para acceso a bases de datos), ASP.NET (para generar páginas activas), WinForms (para construir aplicaciones Windows) o soluciones para Linux y para Mac. Adicionalmente puede emplearse Visual Studio.NET o entornos de desarrollo integrados basados en Mono que permiten hacer todo lo anterior desde una interfaz visual basada en ventanas [5], [6], como se muestra en la figura 2.

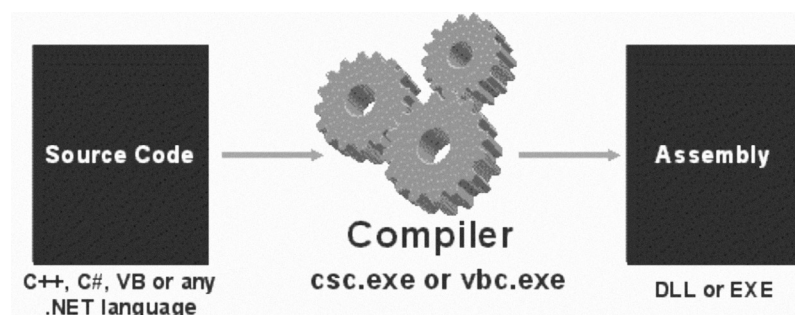
Figura 2. Entorno del Framework para Windows.



2.1. Common Language Runtime (CLR)

El CLR es el núcleo de la plataforma .NET ya que es el encargado de gestionar la ejecución de código compilado para la plataforma .NET. Puede parecerse a la máquina virtual de Java [7], [8], [9] (figura 3).

Figura 3. Common Language Runtime (CLR)



Fuente: elaboración propia.

Las dos principales características del CLR son:

1. Ejecución multiplataforma: el CLR actúa como una máquina virtual; se encarga de ejecutar las aplicaciones diseñadas para la plataforma .NET. Su especificación está abierta, por lo que cualquier plataforma para la que exista una versión del CLR podrá ejecutar cualquier aplicación .NET. Microsoft ha desarrollado versiones del CLR para la mayoría de las versiones de Windows. Por otro lado, Microsoft ha firmado un acuerdo con Correl para portar el CLR a Linux y también hay terceros que están desarrollando de manera independiente versiones de libre distribución del CLR para Linux. Asimismo, dado que la arquitectura del CLR está totalmente abierta, es posible que en el futuro se diseñen versiones de este para otros sistemas operativos.

2. Integración de lenguajes: desde cualquier lenguaje para el que exista un compilador que genere código para la plataforma .NET es posible utilizar código generado para esta usando cualquier otro lenguaje, como si se tratara de código escrito usando el primero [5], [6].

2.2. Common Intermediate Language (CIL)

Pronunciado “sil” o “kil”, anteriormente llamado Microsoft Intermediate Language o MSIL, es el lenguaje de programación de más bajo nivel en el Common Language Infrastructure y en el .NET Framework. Los lenguajes del .NET Framework compilan a CIL. CIL es un lenguaje ensamblador orientado a objetos, se basa en pilas y es ejecutado por el CLR. Existen varios compiladores de lenguajes que generan este tipo de código [7], [9], [10], [11].

2.3. Common Language Infrastructure (CLI)

La infraestructura de lenguaje común es una especificación estandarizada que describe un entorno virtual para la ejecución de aplicaciones, cuya principal característica es la de permitir que aplicaciones escritas en distintos lenguajes de alto nivel puedan luego ejecutarse en múltiples plataformas tanto de hardware como de software sin necesidad de reescribir o recompilar su código fuente [4], [10], [11].

2.4. Proyecto Mono

Mono es un proyecto de implementación del Framework .NET de Microsoft utilizando código libre (*open source*), gestionado por Ximian y basado en las especificaciones definidas en ECMA.

Actualmente, el proyecto se puede considerar bastante maduro en muchos aspectos. Además se están desarrollando partes que no son específicamente de la plataforma, como ADO.NET, WinForms y ASP.NET.

Uno de los objetivos del proyecto Gnome de Ximian es el poder disponer en entornos Unix de herramientas comparables y competitivas con las existentes en el mercado de aplicaciones de escritorio. Al desarrollar Gnome se pudo comprobar que era importante disponer de independencia respecto a los lenguajes de programación, para lo cual se diseñaron las Gnome API de forma que sean fácilmente utilizables desde diferentes lenguajes, mediante la utilización sencilla de *wrappers*. Actualmente la API de Gnome está disponible para múltiples lenguajes de programación [5], [12], [13].

Más tarde se comenzó a usar Corba con el objetivo de definir interfaces para los distin-

tos componentes y encapsular las API. Este trabajo derivó en la creación de Bonobo, del que actualmente existen interfaces desde C, Perl, Python y Java.

La plataforma .NET busca unos objetivos muy similares a los que persigue Gnome: ofrecer independencia de lenguaje a los programadores; es también más madura, documentada, más amplia en su ámbito de actuación y tiene un diseño consistente. Cualquier API que se escriba utilizando un lenguaje que genere código para el CLR puede usarse desde cualquier otro lenguaje que genere código para esta plataforma [5], [13].

2.5. MonoDevelop

MonoDevelop es un entorno de desarrollo diseñado principalmente para C# y otros lenguajes NET. MonoDevelop permite a los desarrolladores escribir rápidamente aplicaciones de escritorio y Web ASP.NET en Linux, Windows y Mac OSX. MonoDevelop hace que sea fácil para los desarrolladores de aplicaciones .NET creadas con Visual Studio implementarlas para Linux y mantener una base de código único para todas las plataformas.

A finales del 2003 un grupo de desarrolladores provenientes de la comunidad Mono comenzó a migrar a SharpDevelop Linux y a usar las librerías GTK.

MonoDevelop ha sido absorbido por el Proyecto Mono y está siendo activamente mantenido por Novell y la comunidad Mono. MonoDevelop se empaqueta y distribuye junto a Mono desde la segunda beta de Mono 1.0. MonoDevelop 3.0.2 es la última versión de MonoDevelop, liberada el 24 de mayo del 2012.

Desde la versión 2.2 MonoDevelop ya dispone de un instalador para Windows y

MacOS. Ofrece así un completo soporte multiplataforma.

MonoDevelop se distribuye juntamente con Mono para Mac OSX y ambos funcionan de manera nativa (sin requerir otro software adicional). MonoDevelop se empaqueta para Solaris sobre SPARC y x86 pero es mantenido por grupos de la comunidad OpenSolaris. Finalmente, MonoDevelop es también mantenido por la comunidad FreeBSD [14].

2.5.1. Características

Multi-plataforma: compatible con Linux, Windows y Mac OSX.

Entorno de desarrollo configurable: diseños de ventanas totalmente personalizados, combinaciones de teclas definidas por el usuario, herramientas externas.

Soporta varios lenguajes: C#, Visual Basic.Net, C/C++, Vala.

Depurador integrado: para depurar aplicaciones Mono y nativas.

Diseñador Visual GTK#: para construir fácilmente aplicaciones GTK#.

ASP.NET: para crear proyectos web con soporte completo de código y prueba en XSP el servidor web de Mono [14].

2.6. C# para Mono

C# (pronunciado “si sharp” en inglés) es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de progra-

mación diseñados para la infraestructura de lenguaje común [8], [12].

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

El nombre C Sharp fue inspirado por la notación musical, donde ‘#’ (sostenido, en inglés sharp) indica que la nota (C es la nota do en inglés) es un semitono más alta, sugiriendo que C# es superior a C/C++. Además, el signo ‘#’ se compone de cuatro signos ‘+’ pegados.

Aunque C# forma parte de la plataforma .NET, esta es una API, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el marco Mono-DotGNU, el cual genera programas para distintas plataformas como Windows, Unix, Android, iOS, Windows Phone, Mac OS y GNU/Linux [15].

C# es un lenguaje de programación de los más nuevos, diseñado para crear un amplio número de aplicaciones empresariales que se ejecutan en .NET Framework. Supone una evolución de Microsoft C y Microsoft C++; es sencillo, moderno, proporciona seguridad de tipos y está orientado a objetos. El código creado mediante C# se compila como código administrado, lo cual significa que se beneficia de los servicios de Common Language Runtime. Estos servicios incluyen interoperabilidad entre lenguajes, recolección de elementos no utilizados, mejora de la seguridad y mayor compatibilidad entre versiones.

C# se presenta como Visual C# en el conjunto de programas Visual Studio .NET. Visual C# utiliza plantillas de proyecto, diseñado-

res, páginas de propiedades, asistentes de código, un modelo de objetos y otras características del entorno de desarrollo. La biblioteca para programar en Visual C# es .NET Framework [8], [16].

Todas estas características están disponibles en el C# de Mono.

3. Desarrollo de la aplicación MathematicsLAB

Como se comentó en el resumen, esta herramienta es el resultado de un proyecto de algunos integrantes del Cuerpo Académico de Tecnologías de Información y Comunicaciones de la Unidad Académica de Ingeniería de la Universidad Autónoma de Guerrero, con la que se pretende apoyar a los estudiantes de educación básica en el estado de Guerrero. Es el primer avance, con el cual se van a hacer las primeras pruebas con el fin de poder implementar nuestros indicadores para poder argumentar si se logra o no el objeti-

vo de contribuir al mejoramiento del rendimiento escolar de los estudiantes.

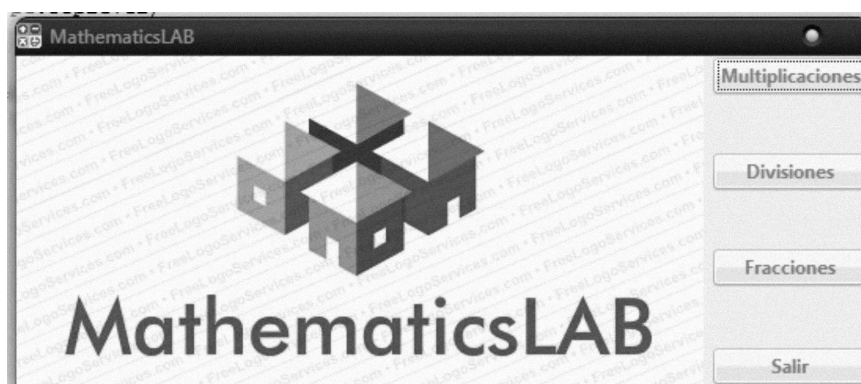
Con la información que se recopiló, el problema de más peso está en la resolución de divisiones, multiplicaciones y operaciones con fracciones, por lo que esta primera versión ataca estos problemas principalmente.

Esta herramienta está diseñada completamente en .Net para software libre, específicamente utilizando el IDE MonoDevelop, programando en lenguaje C# y el sistema operativo ubuntu.

Todas las interfaces son muy sencillas, ya que se está haciendo hincapié en la usabilidad para que sea fácil de utilizar por los estudiantes y se toma en cuenta que estos son principiantes en el manejo de las computadoras.

El funcionamiento de la aplicación se describe en las figuras 4, 5, 6, 7, 8, 9 y 10.

Figura 4. Pantalla inicial de la aplicación MathematicsLAB



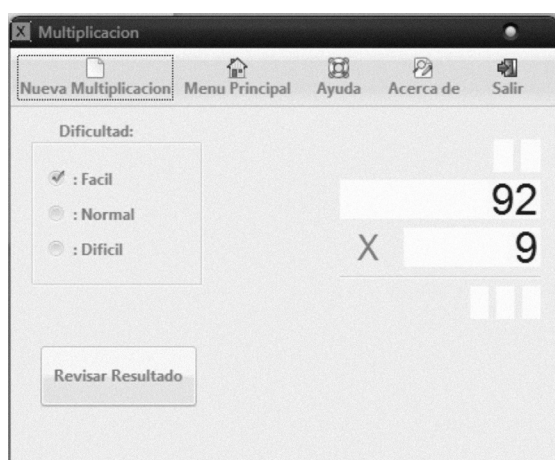
Fuente: elaboración propia.

En la figura 4 se muestra la pantalla inicial para optar por algunas de las tres operaciones: multiplicaciones, divisiones, fracciones, o salir de la aplicación. Es una interfaz muy sencilla para decidir qué tipo de operación se quiere practicar.

Estas opciones nos permiten regresar al menú principal, generar una nueva multiplicación, mostrar ayuda relacionada con el programa, ver información relacionada con este y, por último, una opción para cerrar el programa por completo.

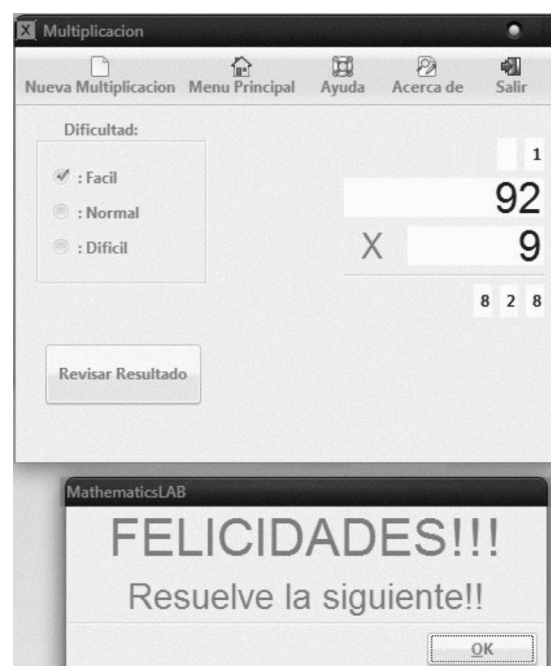
Figura 5. Interfaz de las operaciones de multiplicaciones

Figura 6. Resolución de una multiplicación exitosa



Fuente: elaboración propia.

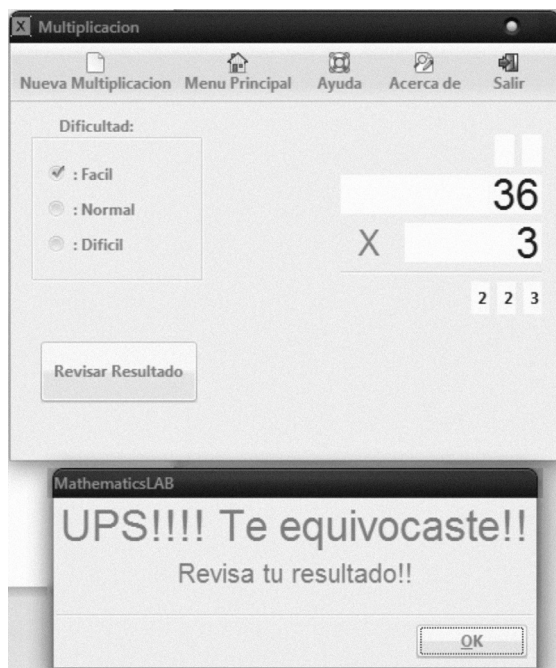
En la figura 5 se muestra una interfaz que sirve para la práctica de las multiplicaciones. Dicha interfaz cuenta con tres niveles: fácil, normal y difícil. Se tienen cuadros de texto libres para llevar a cabo la operación, así como un botón para comprobar la correcta o incorrecta solución de la operación. Además, el programa cuenta con una barra de opciones rápidas y sencillas, fáciles de reconocer. Di-



Fuente: elaboración propia.

En caso de haber resuelto la operación de manera exitosa, se desplegará un cuadro de diálogo, el cual elogiará al usuario por resolver la operación de manera correcta (figura 6).

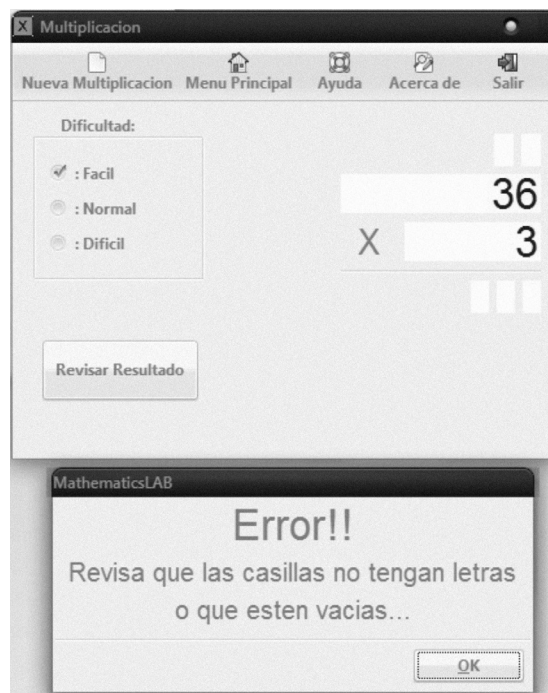
Figura 7. Resolución de una multiplicación fallida



Fuente: elaboración propia.

En caso de haber fallado la respuesta a dicha operación, se dará un aviso mediante un cuadro de diálogo. Se ofrece la oportunidad de regresar y tratar de resolver la operación (figura 7).

Figura 8. Interfaz de error de captura



Fuente: elaboración propia.

No todos los errores se deben al desarrollo de la operación matemática. Puede ser que el usuario pretenda comprobar su resultado sin haber introducido algo en las casillas de resultado, o que haya introducido letras por error. En dicho caso, la aplicación avisará si alguno de los dos casos anteriores se presenta (figura 8).

Al momento de hacer uso de la opción salir, una ventana pedirá confirmar la selección. De esta manera, se evitará que el usuario cierre la aplicación por error (figura 9).

Figura 9. Interfaz de cierre de la aplicación por error



Fuente: elaboración propia.

Figura 10. Interfaz de las operaciones de divisiones



Fuente: elaboración propia.

Para el caso de las operaciones con divisiones, se ofrecen las mismas opciones que en las multiplicaciones, pero la operación es totalmente diferente (figura 10).

Para el caso de las operaciones con fracciones, básicamente cuenta con las mismas opciones que tiene la interfaz para resolver las multiplicaciones o las divisiones.

4. Referencias

- [1] R. Pressman, *Ingeniería del software. Un enfoque práctico*, 6ª ed. 2012.
- [2] Disponible en: <http://alvaroblogdetrabajo.blogspot.mx/2011/03/definicion-de-software.html>
- [3] H. Schildt, C#. *Manual de referencia*. McGraw-Hill, 2003.
- [4] S. F. Morales, "Interoperabilidad entre lenguajes de programación", *Vínculos*, Universidad Distrital Francisco José de Caldas, Bogotá, pp. 186-195.
- [5] Patxi Echarte.
- [6] Disponible en: <http://elvex.ugr.es/decsai/csharp/dotnet/index.xml>
- [7] P. J. Deitel y H. M. Deitel, *Java. Cómo programar*, 7ª ed. Pearson-Prentice Hall.
- [8] F. J. Ceballos, *Enciclopedia de Microsoft Visual C#, 3ª ed.* Alfaomega Ra-Ma.
- [9] A. Aho et al., *Compiladores. Principios, técnicas y herramientas*. Pearson y Addison Wesley. Disponible en www.eslomas.com/index.php/archives/2005/05/11/introduccion-plataforma-net-y-mono/
- [10] J. A. González Seco, C#. *El nuevo lenguaje de internet*. Disponible en <http://mygnet.com>
- [11] MSDN, *Interoperabilidad entre lenguajes*. Disponible en: <http://msdn.microsoft.com/es-es/library/a2c7tshk.aspx>
- [12] F. J. Ceballos, *Aplicaciones .NET multiplataforma (Proyecto Mono)*. Alfaomega Ra-Ma.

- [13] Disponible en: www.wikilearning.com/tutorial/introduccion_a_la_plataforma_net_y_mono-proyecto_mono/11990-3
- [14] Disponible en: <http://monodevelop.com/>
- [15] Disponible en: http://es.wikipedia.org/wiki/C_Sharp
- [16] Disponible en: [http://msdn.microsoft.com/es-es/library/aa287558\(v=vs.71\).aspx](http://msdn.microsoft.com/es-es/library/aa287558(v=vs.71).aspx)

Apéndice I. Código del módulo divisiones

En este apartado agregamos solo el código para el módulo que permite resolver divisiones, a fin de mostrar cómo es que abordamos este problema, ya que no podemos agregar todo el código porque incrementaríamos de manera exorbitante la extensión del artículo.

```
using System;
using Pango;

namespace MathematicsLAB
{
    class NumAleatorios{

        public int UnaCifra (){
            // Creamos un objeto random
            Random r = new Random();
            // Generamos un número entre dos valores dados
            int aleatorio = r.Next(1, 10);
            return aleatorio;
        }

        public int DosCifras (){
            // Creamos un objeto random
            Random r = new Random();
            // Generamos un número entre dos valores dados
            int aleatorio = r.Next(11, 99);
            return aleatorio;
        }

        public int TresCifras ()
        {
            // Creamos un objeto random
            Random r = new Random();
            // Generamos un número entre dos valores dados
            int aleatorio = r.Next(100, 999);
            return aleatorio;
        }
    }
}
```

```

public int CuatroCifras ()
{
    // Creamos un objeto random
    Random r = new Random();
    // Generamos un número entre dos valores dados
    int aleatorio = r.Next(1000, 9999);
    return aleatorio;
}

public partial class Divisiones : Gtk.Window
{
    public Divisiones () :
        base(Gtk.WindowType.Toplevel)
    {
        this.Build ();
        this.WidthRequest = 420;
        this.HeightRequest = 260;
        entryDividendo.ModifyFont      (FontDescription.FromString
("Arial 25"));
        entryDivisor.ModifyFont (FontDescription.FromString ("Arial
20"));
        radiobuttonFacil.Active = false;
    }

    protected void OnQuitActionActivated (object sender, EventArgs
e)
    {
        MathematicsLAB.DialogoSaliir d = new MathematicsLAB.Dialogo-
Saliir ();
        d.Show ();
    }

    protected void OnRadiobuttonFacilClicked (object sender, Even-
tArgs e)
    {
        //Limpiar casillas
        entryCoscientel.Text = "";
        entryCosciente2.Text = "";

        entryCosciente3.Visible = false;
        entryCosciente4.Visible = false;
        entryResto3.Visible = false;
        entryResto5.Visible = false;
        entryResto6.Visible = false;
        entryResto7.Visible = false;
        entryResto8.Visible = false;
        entryResto9.Visible = false;
        NumAleatorios num = new NumAleatorios ();
        int divisor, dividendo;
        divisor = num.UnaCifra ();

```

```

        dividendo = num.DosCifras ();
        entryDivisor.Text = divisor + "";
        entryDividendo.Text = dividendo + "";
    }
    protected void OnRadiobuttonNormalClicked (object sender, EventArgs e)
    {
        //Limpiar casillas
        entryCosciente1.Text = "";
        entryCosciente2.Text = "";
        entryCosciente3.Text = "";

        entryCosciente3.Visible = true;
        entryCosciente4.Visible = false;
        entryResto3.Visible = true;
        entryResto5.Visible = true;
        entryResto7.Visible = true;
        entryResto6.Visible = false;
        entryResto8.Visible = false;
        entryResto9.Visible = false;
        NumAleatorios num = new NumAleatorios ();
        int dividendo, divisor;
        divisor = num.UnaCifra ();
        dividendo = num.TresCifras ();
        entryDividendo.Text = dividendo + "";
        entryDivisor.Text = divisor + "";
    }
    protected void OnRadiobuttonDificilClicked (object sender, EventArgs e)
    {
        //Limpiar casillas
        entryCosciente1.Text = "";
        entryCosciente2.Text = "";
        entryCosciente3.Text = "";
        entryCosciente4.Text = "";

        entryCosciente3.Visible = true;
        entryCosciente4.Visible = true;
        entryResto3.Visible = true;
        entryResto4.Visible = true;
        entryResto5.Visible = true;
        entryResto6.Visible = true;
        entryResto7.Visible = true;
        entryResto8.Visible = true;
        entryResto9.Visible = true;
        NumAleatorios num = new NumAleatorios ();
        int dividendo, divisor;
        divisor = num.DosCifras ();
        dividendo = num.CuatroCifras ();
    }

```

```

        entryDividendo.Text = dividendo + "";
        entryDivisor.Text = divisor + "";
    }
    protected void OnButtonResultadoClicado (object sender, Even-
tArgs e)
    {
        string resultado;
        int ResultadoDiv, DivisionNum;

        try {

            if (radiobuttonFacil.Active == true) {

                resultado = entryCoscientel.Text + entryCoscien-
te2.Text + entryCosciente3.Text + entryCosciente4.Text;
                DivisionNum = int.Parse (resultado);
                ResultadoDiv = int.Parse (entryDividendo.Text) /
int.Parse (entryDivisor.Text);

                if (DivisionNum == ResultadoDiv){
                    MathematicsLAB.Exito exito = new Exito ();
                    exito.Run ();

                    entryCoscientel.Text = "";
                    entryCosciente2.Text = "";
                    entryResto1.Text = "";
                    entryResto2.Text = "";
                    entryResto4.Text = "";

                    NumAleatorios num = new NumAleatorios ();
                    int divisor, dividendo;

                    divisor = num.UnaCifra ();
                    dividendo = num.DosCifras ();
                    entryDivisor.Text = divisor + "";
                    entryDividendo.Text = dividendo + "";
                }else {
                    MathematicsLAB.Equivocacion equi = new Equivo-
cacion ();
                    equi.Run ();
                }
            }
            else if (radiobuttonNormal.Active == true){
                resultado = entryCoscientel.Text + entryCoscien-
te2.Text + entryCosciente3.Text + entryCosciente4.Text;
                DivisionNum = int.Parse (resultado);
                ResultadoDiv = int.Parse (entryDividendo.Text) /
int.Parse (entryDivisor.Text);

                if (DivisionNum == ResultadoDiv){

```



```
entryResto8.Text = "";
entryResto9.Text = "";

NumAleatorios num = new NumAleatorios ();
int divisor, dividendo;

divisor = num.DosCifras ();
dividendo = num.CuatroCifras ();

entryDivisor.Text = divisor + "";
entryDividendo.Text = dividendo + "";

    }else {
        MathematicsLAB.Equivocacion equi = new Equivocacion ();
        equi.Run ();
    }

    } catch {
        MathematicsLAB.CasillaVacía casillavacia = new CasillaVacía ();
        casillavacia.Run ();
    }
}
}
```