

Análisis del impacto de la Herramienta CASE Progranimate en estudiantes del campus de las ingenierías región Veracruz de la Universidad Veracruzana como parte de las herramientas de apoyo para aprender a programar

Espinoza Maza, Jonathan de Jesús
Universidad Veracruzana, México
jespinoza@uv.mx

Rodríguez Rodríguez, Luis Alberto
Universidad Veracruzana, México
alberodriguez@uv.mx

Macgluf Issasi, Arturo
Universidad Veracruzana, México
amacgluf@uv.mx

Fragoso Terán, Juan Manuel
Universidad Veracruzana, México
alberodriguez@uv.mx

Aguirre Orozco, Ulises Gonzalo
Universidad Veracruzana, México
uaguirre@uv.mx

Resumen - El razonamiento humano es el protagonista en el proceso de aprendizaje de la programación y el apoyo con diversas técnicas y elementos forman parte de este. Estos elementos se relacionan con las herramientas que puedan servir como guía para obtener la solución computacional a un problema, tales como los diagramas de flujo, pseudocódigos y el desarrollo en algún lenguaje de programación. Este artículo pretende servir de apoyo a las tareas de la construcción de soluciones a problemas con programación.

Palabras clave: Tecnología; Razonamiento humano; Herramienta CASE; Progranimate;

Abstract - The human reasoning is the leading factor in the programming learning process, and the support with a variety of techniques and elements are part of the success, these elements are related to the tools that can work as a guide to obtain the computing solution to a problem such as flow diagrams, pseudo codes and development of some programming language. This article aims to support the tasks about building solutions to programming issues.

Keywords: Technology; Human reasoning; CASE tool; Progranimate;

INTRODUCCIÓN

Realizar diagramas desde cero sin una herramienta asistida o de manera natural (manual, es decir con lápiz y papel) como parte previa al desarrollo o a la documentación de una solución a un problema, conlleva a tener ciertas desventajas, como la inversión de tiempo que pudiera turnarse excesivo si la situación se encuentra contra reloj, calcular el espacio de escritura en un documento (hablando de un desarrollo a mano alzada), o bien la falta de certeza de que las figuras estén bien posicionadas o a su vez, la sintaxis incluida en ella sea la correcta para corroborar si la solución va por buen camino o no, en fin, es un sin número de posibles situaciones que hacen que aplicar diagramas de flujo no sea tan atractivo.

No obstante, existen actualmente diversas herramientas que permiten simplificar el tiempo y mejorar la experiencia con esta importante técnica previa al desarrollo a través de un lenguaje de programación.

LAS HERRAMIENTAS CASE Y SU IMPORTANCIA EN LAS SOLUCIONES DE PROBLEMAS

Desde los inicios de la última década del siglo pasado, los analistas comenzaron a beneficiarse de las herramientas de productividad, llamadas herramientas de Ingeniería de Software Asistida por Computadora CASE (Computer-Aided Software Engineering), las cuales fueron creadas con la finalidad de mejorar el trabajo rutinario mediante apoyo automatizado. (Kendall & Kendall, 2005)

Razones para el uso de las herramientas CASE

De acuerdo con Kendall&Kendall (2005), se argumentan 4 características inherentes de las herramientas CASE para su aplicación en la solución de problemas o en su defecto para el desarrollo de soluciones computacionales:

- Aumento en la productividad del analista: Las herramientas CASE dan al analista la posibilidad de realizar planeación, análisis y diseño por medio de gráficos, con el propósito de construir aplicaciones locales o en línea, incluyendo base de datos para los casos en los que se requieran.

- Mejora la comunicación analista-usuario: El éxito de las soluciones computacionales radica en la comunicación entre analistas y usuarios durante el ciclo de vida del desarrollo de la solución. Gracias al uso de herramientas CASE, se desprende que esta acción fomenta una mayor y más eficiente comunicación entre este binomio.
- Integración de las actividades del ciclo de vida: Uno de los objetivos de las herramientas CASE es integrar las actividades y dar continuidad de una fase a otra durante el ciclo de vida de desarrollo de sistemas o de las soluciones computacionales, siendo de mayor utilidad cuando se requiere iteraciones de retroalimentación y modificaciones.
- Evaluación precisa de los cambios en el mantenimiento. Una de las razones más importantes para el uso de las CASE, es que permiten a los usuarios realizar el análisis y evaluación del impacto en los cambios en la fase de mantenimiento.

Una vez definida la importancia de las Herramientas CASE, se pueden mencionar algunos ejemplos comunes que existen actualmente

al alcance de cualquier usuario que desee modelar las soluciones para la diversidad de problemas a resolver, desde el software con licenciamiento hasta los libres, por ejemplo, uno de las CASE más comunes con licencia es Microsoft Visio, o si hablamos de software libre, PSeint, DFD, Raptor son opciones a elegir, entre otros más que existen el mundo del Software y de las CASE.

Progranimate como herramienta CASE aplicado a la enseñanza de los Algoritmos Computacionales y Programación.

Hablar de Progranimate es hablar de una potente herramienta que tiene su origen en la Universidad de Carolina Occidental (Western Carolina University), desarrollada por el Dr. Andrew Scott, Profesor Asistente en dicha Universidad, y pensada en programadores novatos y profesores.

Progranimate es una herramienta de resolución de problemas visuales interactiva y que incluye un atractivo generador de código.

Está dirigido a enseñar los conceptos básicos de la programación y al fortalecimiento de las habilidades de resolución de problemas y de lectura de códigos de los programadores novatos. Esto está en relación con los conceptos imperativos (no orientados a objetos) de variables, selección de secuencias, iteración y matrices.

Progranimate utiliza el diagrama de flujo para visualizar la programación y el proceso de ejecución. Esto tiene como objetivo fomentar modelos efectivos y precisos de conceptos de programación, algoritmos y ejecución de acuerdo con su autor. (Scott, 2019)

La interfaz de Progranimate es amigable e intuitiva, permite realizar soluciones relacionadas con estructuras, secuencias selectivas, repetitivas y de datos (vectores y matrices), mismos que permiten ofrecer una solución amigable, en donde el diagrama de flujo es ejecutado para poder visualizar el flujo de los datos y sus interacciones.

Además, permite visualizar automáticamente de acuerdo con la forma de ir construyendo el diagrama

de flujo, el código de programación en 5 lenguajes distintos de programación tales como; Java, JavaScript, Visual Basic.Net, Visual Basic 6.0 y Pascal, además de ofrecer tres pseudocódigos; Pseudo Java, Código Simple 1 y Código Simple 2, mismo que da una perspectiva distinta para ir conociendo las diferentes sintaxis de cada uno de los mencionados.

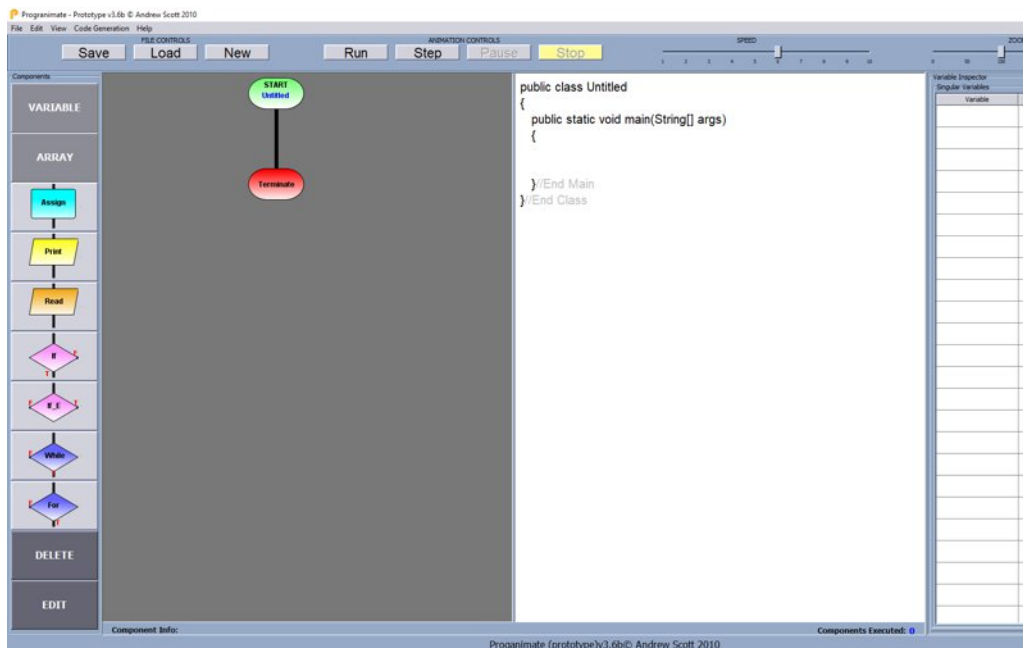
Todas estas opciones complementadas por un inspector de variables, que permite visualizar el comportamiento de los datos y de los resultados que se van obteniendo desde el inicio hasta el final del diagrama de flujo.

Conociendo Progranimate.

La Figura 1, muestra los elementos que contiene la interfaz de la herramienta CASE Progranimate. Dicha interfaz está compuesta por los siguientes elementos:

- **Barra de Submenús.** Archivo, Edición, Vista, Generación de Código y Ayuda.
- **Controles de archivo.** Con la finalidad de salvar el avance en el diseño de una solución, Progranimate ofrece guardar, cargar y crear nuevos archivos con extensión .prg.

Figura 1. Interfaz de Progranimate.



Fuente: *Obtenido de Progranimate (Scott, 2019)*

- **Control de Animación.** Los elementos situados en esta sección permiten inicializar, ir paso a paso, pausar o detener la ejecución de la solución con la finalidad de verificar el comportamiento de cada elemento y si el resultado es el esperado.
- **Barra deslizante de velocidad.** Permite aumentar o disminuir la velocidad de la ejecución en tiempo real del diagrama de flujo, con la finalidad de visualizar la interacción entre elementos.
- **Barra deslizante de Zoom.** Permite ampliar o disminuir el tamaño del diagrama de flujo, según sea el tamaño de este.
- **Inspector de variables.** Permite visualizar las variables, sus valores y el tipo con el que fueron declaradas, mismos valores que van cambiando de acuerdo con la ejecución de sus componentes hasta llegar a la solución final.
- **Componentes ejecutados.** Muestra el total de componentes ejecutados una vez que se “corre” el diagrama de flujo, asimismo se relaciona con el total de líneas ejecutadas para la solución.
- **Panel de código.** Permite conocer el código de programación de acuerdo con el lenguaje seleccionado, con la finalidad de familiarizar la sintaxis con el usuario, así como la inserción de cada componente del diagrama de flujo, esta es una de las cualidades

que hace atractivo a esta herramienta CASE con tecnología Cliente-Servidor.

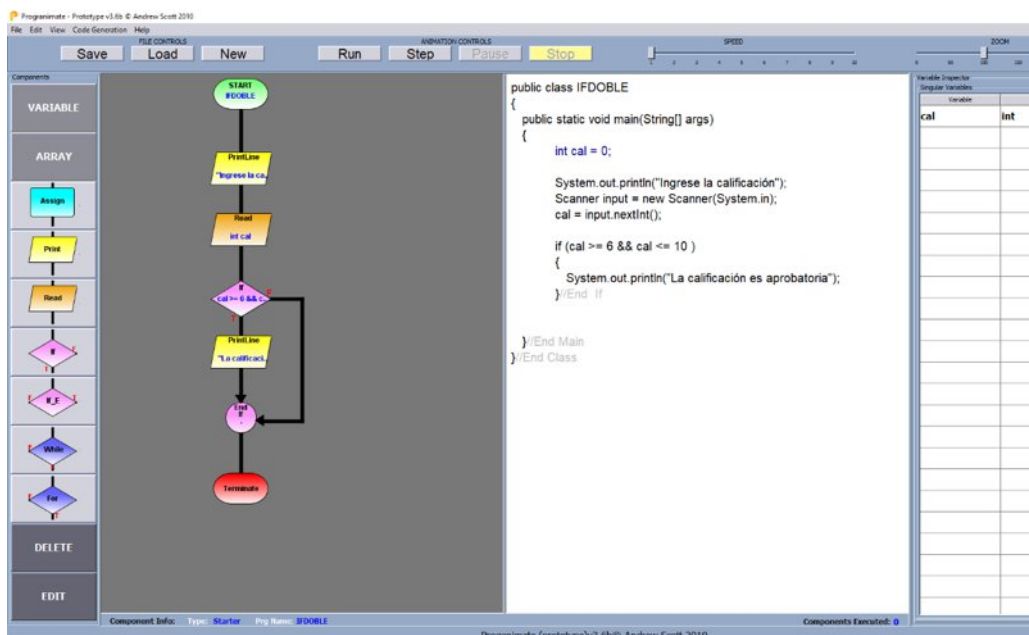
- **Información de los componentes.** Es una especie de barra de estado, que muestra el tipo de componente seleccionado y el contenido de este.
- **Panel de Componentes.** Muestra la simbología aplicable a estructuras secuenciales, selectivas, repetitivas y de datos, además el botón para añadir o eliminar variables y la edición de cada símbolo, mismo al que también se accede al hacer doble clic sobre el símbolo una vez insertado en el panel central del diagrama de flujo.

- **Área de diseño del Diagrama de Flujo.** En esta parte se van insertando los símbolos que se encuentran en el Panel de Componentes, mismo que al dar clic en cualquiera de ellos y posteriormente dar clic el último que se encuentra a partir de Inicio o Start, se van posicionando para dar forma a la solución deseada.

Aplicando Progranimate.

Una vez explicado a detalle la interfaz de Progranimate, se resolverán con ayuda de la herramienta los tres ejercicios anteriores relacionados con las estructuras selectivas, simple, doble y múltiple.

Figura 2. Solución a un problema con estructura selectiva simple.



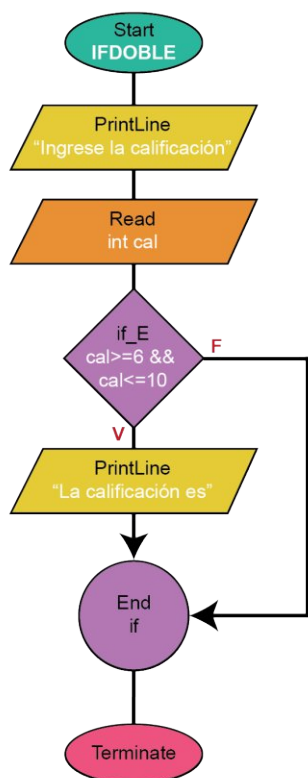
Fuente: Elaboración propia con base en la herramienta CASE Progranimate (Scott, 2019).

Ejemplo de solución para una Estructura Selectiva Simple con Progranimate.

La Figura 2 muestra la solución de un problema referente a una calificación para saber si es aprobatoria utilizando una estructura selectiva simple con Progranimate.

La Figura 3 muestra la solución a la estructura selectiva simple presentada en la Figura 2 utilizando la

Figura 3. Diagrama de flujo de una estructura selectiva simple.



Fuente: Elaboración propia con base en la herramienta CASE Progranimate (Scott, 2019).

herramienta CASE Progranimate (Scott, 2019) (el diagrama de flujo se exportó en formato .bmp).

La Figura 4 indica la sintaxis de lenguaje Java de una estructura selectiva simple exportada mediante la herramienta CASE Progranimate (Scott, 2019), esta herramienta permite obtener el código fuente en un formato .txt a partir del diagrama de flujo generado.

Figura 4. Sintaxis de lenguaje Java, de una estructura selectiva simple.

```

if_simple.txt: Bloc de notes
Archivo Edición Formato Ver Ayuda
public class IFDOBLE
{
    public static void main(String[] args)
    {
        int cal = 0;

        System.out.println("Ingrese la calificación");
        Scanner input = new Scanner(System.in);
        cal = input.nextInt();

        if (cal >= 6 && cal <= 10 )
        {
            System.out.println("La calificación es aprobatori
        } //End If

    } //End Main
} //End Class
  
```

Fuente: Elaboración propia con base en la herramienta CASE Progranimate (Scott, 2019).

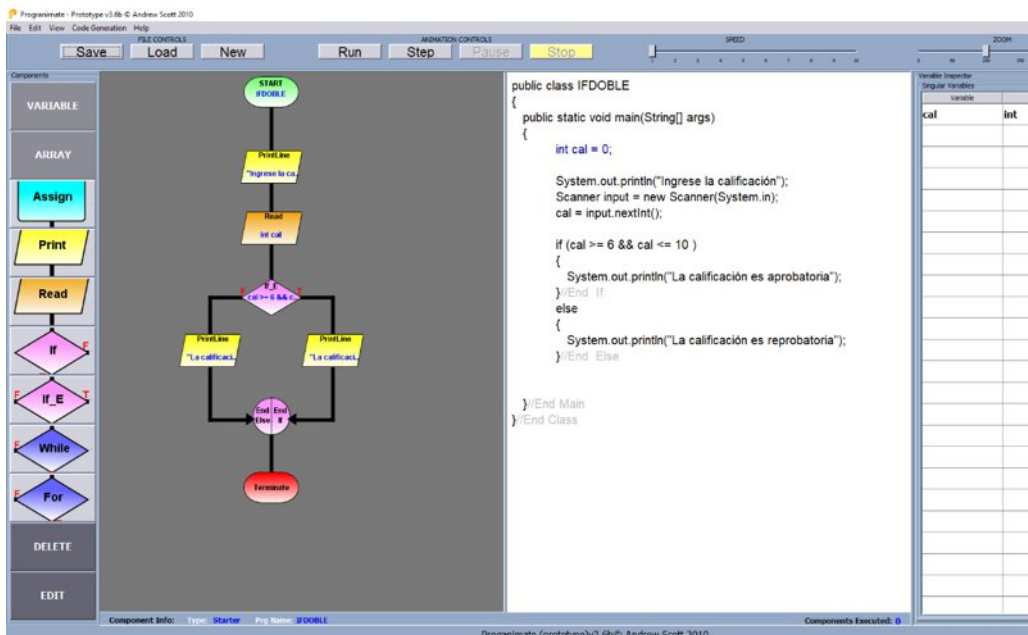
Ejemplo de solución para una Estructura Selectiva Doble con Progranimate.

La Figura 5 muestra la solución de un problema referente a una calificación para saber si el estudiante se encuentra aprobado o reprobado con base en la evaluación de la variable utilizando una estructura selectiva doble con la herramienta CASE Progranimate (Scott, 2019).

La Figura 6, muestra el diagrama de flujo de la solución a la estructura selectiva doble presentada en la Figura 5 utilizando la herramienta CASE Progranimate (Scott, 2019) (la Figura fue exportada en formato .bmp).

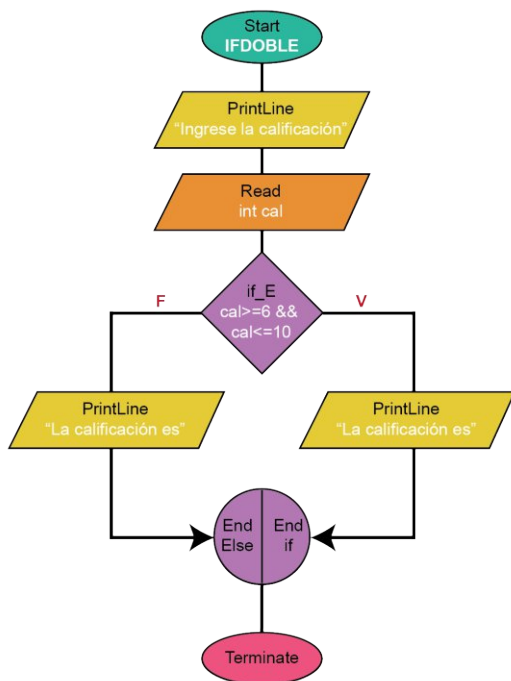
La Figura 7 indica la sintaxis de lenguaje Java de una estructura selectiva doble exportada mediante la herramienta CASE Progranimate (Scott, 2019), esta herramienta permite obtener el código fuente en un formato .txt a partir del diagrama de flujo generado en la Figura 6.

Figura 5. Solución a un problema con estructura selectiva doble.



Fuente: Elaboración propia con base en la herramienta CASE Progranimate (Scott, 2019).

Figura 6. Diagrama de flujo de la solución a la estructura selectiva doble.



Fuente: *Elaboración propia con base en la herramienta CASE Progranimate (Scott, 2019).*

Ejemplo de solución para una Estructura Selectiva Múltiple con Progranimate.

La Figura 8 muestra la solución a una estructura selectiva múltiple que muestra la solución a una calificación para saber si el Estudiante se encuentra aprobado o reprobado con base en la evaluación de la variable, misma que identifica si el valor ingresado es cero o un número negativo, utilizando la herramienta CASE Progranimate (Scott, 2019).

Figura 7. Sintaxis de lenguaje Java, de una estructura selectiva doble.

```

ifdoble.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
public class IFDOBLE
{
    public static void main(String[] args)
    {
        int cal = 0;

        System.out.println("Ingrese la calificación");
        Scanner input = new Scanner(System.in);
        cal = input.nextInt();

        if (cal >= 6 && cal <= 10 )
        {
            System.out.println("La calificación es aprobatoria");
        } //End If
        else
        {
            System.out.println("La calificación es reprobatoria");
        } //End Else

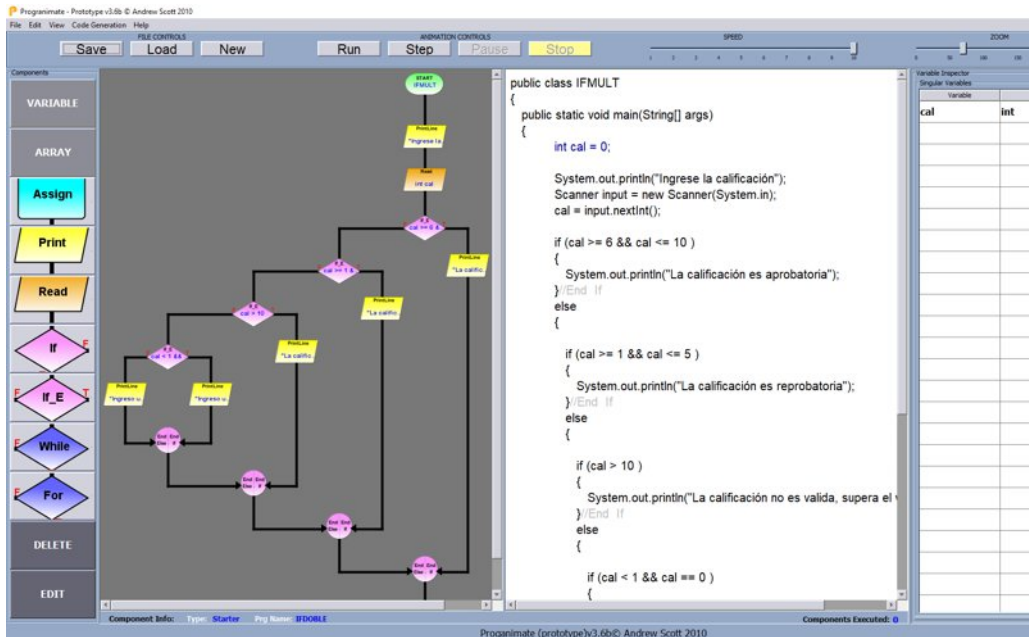
    } //End Main
} //End Class
  
```

Fuente: *Elaboración propia con base en la herramienta CASE Progranimate (Scott, 2019).*

La Figura 9, muestra el diagrama de flujo de la solución a la estructura selectiva múltiple presentada en la Figura 8 utilizando la herramienta CASE Progranimate (Scott, 2019) (la Figura fue exportada en formato .bmp).

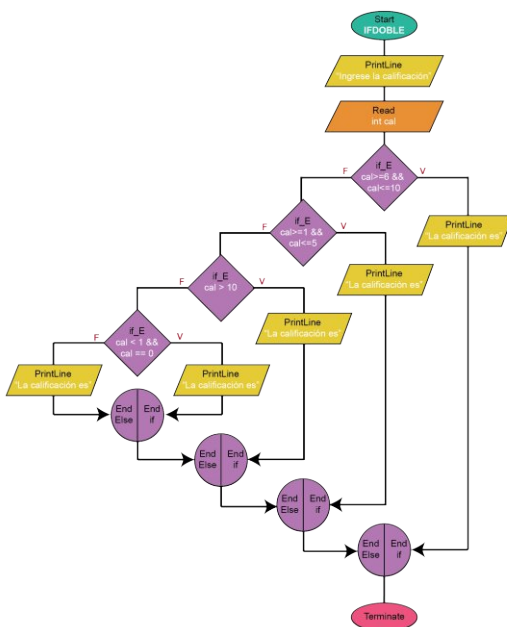
La Figura 10 indica la sintaxis de lenguaje Java de una estructura selectiva múltiple exportada mediante la herramienta CASE Progranimate (Scott, 2019), se muestra el código fuente generado a partir del diagrama de flujo de la Figura 9.

Figura 8. Solución a un problema con estructura selectiva múltiple.



Fuente: Elaboración propia con base en la herramienta CASE Progranimate (Scott, 2019).

Figura 9. Diagrama de flujo de la solución a la estructura selectiva múltiple.



Fuente: Elaboración propia con base en la herramienta CASE Progranimate (Scott, 2019).

Figura 10. Sintaxis de lenguaje Java, de una estructura selectiva doble.

```

#Multitab Bloc de notas
Archivo Edición Formato Ver Ayuda
public class IFMULT
{
    public static void main(String[] args)
    {
        int cal = 0;

        System.out.println("Ingrese la calificación");
        Scanner input = new Scanner(System.in);
        cal = input.nextInt();

        if (cal >= 6 && cal <= 10 )
        {
            System.out.println("La calificación es aprobatoria");
        } //End If
        else
        {
            if (cal >= 1 && cal <= 5 )
            {
                System.out.println("La calificación es reprobatoria");
            } //End If
            else
            {
                if (cal > 10 )
                {
                    System.out.println("La calificación no es valida, supera el valor 10");
                } //End If
            } //End Else

            if (cal < 1 && cal == 0 )
            {
                System.out.println("Ingreso un cero, no es un valor aceptable");
            } //End If
        } //End Else
    } //End Main
} //End Class

```

Fuente: Elaboración propia con base en la herramienta CASE Progranimate (Scott, 2019).

RESULTADOS

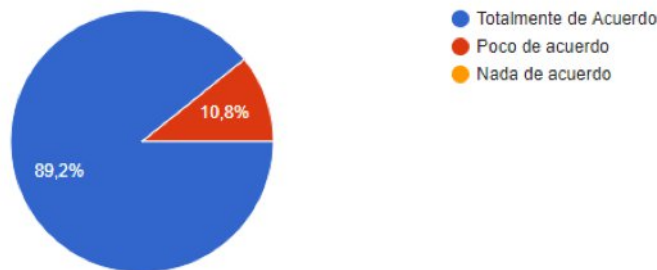
Después de haber utilizado la Herramienta CASE Progranimate como parte del acompañamiento académico durante el periodo febrero – julio 2019 (periodo efectivo comprendido del 18 de febrero al 12 de junio para el Campus de las Ingenierías Región Veracruz (Universidad Veracruzana, 2019), se aplicó un

instrumento de evaluación para analizar el impacto de la herramienta Progranimate en tres secciones de la Experiencia Educativa (materia) Algoritmos Computacionales y Programación con diversos Estudiantes de los distintos Programas Educativos Iniciación Disciplinar, obteniendo los resultados mostrados de la Figura 11 a la 17.

Figura 11. Pregunta relacionada con la simplicidad de la Aplicación.

¿Su uso es simple?

65 respuestas

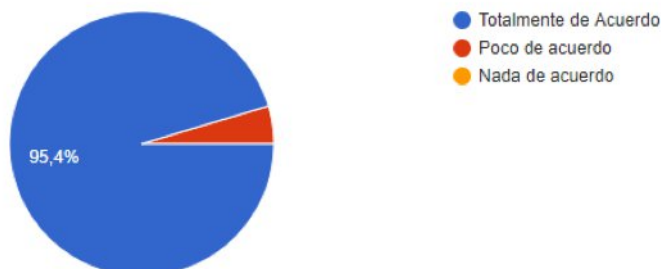


Fuente: Elaboración propia.

Figura 12. Pregunta relacionada con la interfaz de la aplicación.

¿La interfaz es amigable/intuitiva?

65 respuestas

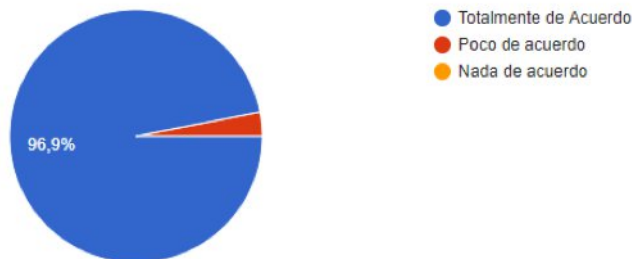


Fuente: Elaboración propia.

Figura 13. Pregunta relacionada con la aportación al conocimiento respecto al tema abordado.

¿La aportación al tema que presenta la aplicación, es útil?

65 respuestas

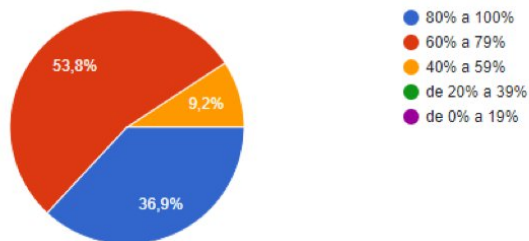


Fuente: *Elaboración propia.*

Figura 14. Resultados del impacto significativo en el proceso de enseñanza - aprendizaje al utilizar Progranimate en las diversas sesiones de clase.

¿En qué nivel; la Herramienta Progranimate le permitió aprender a programar?

65 respuestas

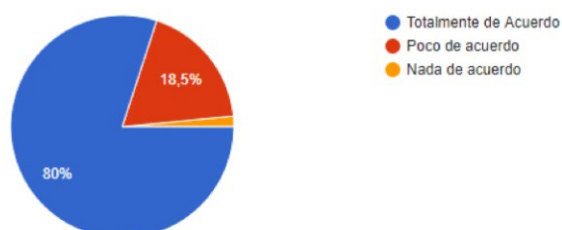


Fuente: *Elaboración propia.*

Figura 15. Pregunta relacionada con el binomio lógica-programación usando Progranimate como herramienta que facilita el apoyo en el aprendizaje.

¿Considera usted que la incorporación de diversos lenguajes de programación, le facilita el uso para relacionar la lógica con la sintaxis del lenguaje seleccionado?

65 respuestas

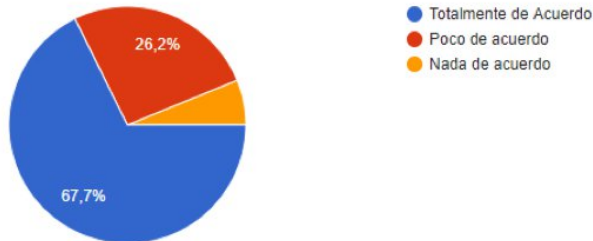


Fuente: *Elaboración propia.*

Figura 16. Pregunta relacionada el nivel de aceptación con la herramienta a través de la arquitectura cliente-servidor.

¿Está de acuerdo con que esta aplicación sea en línea? (Es decir, que la pueda ejecutar en cualquier computadora con acceso a internet)

65 respuestas

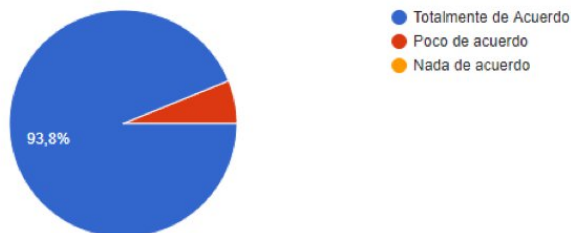


Fuente: Elaboración propia.

Figura 17. Pregunta relacionada con la aceptación de la idea de crear una herramienta con autoría mexicana, en específico de la UV.

¿Le gustaría que la Universidad Veracruzana creara una herramienta similar?

65 respuestas



Fuente: Elaboración propia.

CONCLUSIÓN

El proceso de resolución de problemas utilizando herramientas computacionales, conlleva de manera natural una serie de pasos que le permiten al desarrollador o solucionador de problemas, comprender y analizar de una manera sencilla pero profunda, cada uno de los elementos que integran la solución, hasta llegar al desarrollo o nivel de programación de dicha solución. Por lo que la importancia de utilizar

herramientas como Progranimate, representan una alternativa efectiva en el proceso de enseñanza-aprendizaje en el largo camino del desarrollo, ya que permite ejercitar, mejorar e incrementar el uso de la lógica para tener un mejor razonamiento en el momento de aplicar soluciones puntuales a la diversidad de problemas que se puedan presentar en la vida del estudiante, desarrollador o solucionador de problemas.

El uso de Progranimate contribuye de manera importante en el proceso de enseñanza – aprendizaje de la programación, dando una perspectiva integral al educando de el comportamiento de una solución y relacionando de manera directa e interactiva con una diversidad de lenguajes de programación, permitiendo al estudiante aplicar dichos conocimientos en otro tipo de lenguajes que no se incluyen en esta herramienta, siendo así un referente para construir nuevas CASE que permitan aportar a este universo de conocimiento.

BIBLIOGRAFÍA

Apolinario García, G. C. (2006). *Algoritmos y Diagramas de Flujo Aplicados en C++*. Perú: Megabyte.

Berenguel Gómez, J. (2016). *Desarrollo de aplicaciones web distribuidas UF1846*. Arganda del Rey, Madrid: Paraninfo.

Cairó, O. (2006). *Metodología de la Programación. Algoritmos, diagramas de flujo y programas*. Ciudad de México: Alfaomega.

Ceballos Sierra, F. J. (1990). *Curso de programación con C. Microsoft C*. Ciudad de México: Macrobite.

Deitel, H., & Deitel, P. (2008). *Cómo Programar en C++*. Ciudad de México: Pearson Educación.

Gottfried, B. (2005). *Programación en C*. Ciudad de México: Mc Graw Hill.

International Organization for Standardization ISO. (2011). *ISO*. Consultado el 12 de abril de 2011. Obtenido de ISO: <http://www.iso-9899.info/n1570.html>

Joyanes Aguilar, L. (2008). *Fundamentos de Programación*. Ciudad de México: Mc Graw Hill.

Joyanes Aguilar, L. (2013). *Fundamentos generales de programación*. Ciudad de México: Mc Graw Hill.

Joyanes Aguilar, L., & Sánchez García, L. (2006). *Programación en C++. Un enfoque práctico*. Aravaca: Mc Graw Hill.

Kendall, K. E., & Kendall, J. (2005). *Análisis y diseño de sistemas*. Ciudad de México: Pearson Educación.

Scott, A. (2019). *Progranimate*. Obtenido de Progranimate: www.progranimate.com

Universidad Veracruzana (2019). *Calendario Ajustado y Oficial para el Campus de las Ingenierías*. Consultado el 20 de junio de 2019. Obtenido de Facultad de Ingeniería Mecánica y Ciencias Navales, Región Veracruz, Universidad Veracruzana.: <https://www.uv.mx/veracruz/fimcn/noticias/calendarioajustado/>