

EasyProg

MENDOZA-AUSTRIA, Luis*†, CABALLERO-TOLENTINO, Juan, MAGGI-NATALE, Carlos, HERNANDEZ-RAMIREZ, Anselmo y ARRIETA-ZUÑIGA, Alejandro

Tecnologico Nacional, Instituto Tecnológico de Pachuca. SEP

ID 1^{er} Autor: *Luis, Mendoza-Austria* / **ORC ID:** 0000-0002-1572-2159, **Researcher ID Thomson:** P4842-2018, **NCBI:** mendozaaustria

ID 1^{er} Coautor: *Juan, Caballero-Tolentino* / **ORC ID:** 0000-0003-2147-3141, **Researcher ID Thomson:** P-6773-2018, **NCBI:** jcaballero

ID 2^{do} Coautor: *Carlos, Maggi-Natale* / **ORC ID:** 0000-0002-3842-409, **Researcher ID Thomson:** P-4747-2018, **arXiv:** zgatoteado

ID 3^{er} Coautor: *Anselmo, Hernandez-Ramirez* / **ORC ID:** 0000-0001-9511-7065, **Researcher ID Thomson:** P-4842-2018, **arXiv:** ahr53

ID 4^o Coautor: *Alejandro, Arrieta-Zuñiga* / **ORC ID:** 0000-0001-9698-788X, **Researcher ID Thomson:** M-9232-2018, **arXiv:** jaaz1967

Recibido: 30 de Septiembre, 2018; Aceptado 01 de Diciembre, 2018

Resumen

El presente trabajo comprende el desarrollo e implementación de una plataforma que proporciona un lenguaje de programación fácil de aprender y utilizar con el cual los niños y los jóvenes puedan introducirse al mundo de la programación. El lenguaje consta de una estructura por bloques, la cual representa una opción fácil de aprender para generar una base sólida de conocimiento. En un futuro no muy lejano, el dominio de la programación de un lenguaje de computación será un conocimiento necesario de toda persona, por lo que, esta plataforma está diseñada para la programación utilizando diferentes tipos de bloques, en los cuales, cada uno representará una instrucción diferente; los programadores combinarán los bloques y agregarán las variables para poder desarrollar sus programas, logrando que los usuarios vayan familiarizándose con las instrucciones más comunes como en los demás lenguajes de programación. En las pruebas realizadas con la aplicación de escritorio, este resultado fue del agrado para los niños que la utilizaron, mostraron gran entusiasmo por la aplicación ya que no les resultó complicada, además de que les generó un interés por la programación.

Programación, Bloques, Lenguaje

Abstract

The present work includes the development and implementation of a platform that provides an easy to learn and use programming language with which children and young people can introduce themselves to the world of programming. The language consists in a structure by blocks, which represents an easy to learn option to generate a solid base of knowledge. In a not so far future, the domain of a computer language programming will be a necessary knowledge for everyone, this platform is designed for programming using different kinds of blocks, in which, each one will represent a different instruction; programmers will combine the blocks and add the variables to develop their programs, letting that the users become familiar with the most common instructions as in the other programming languages. The tests realized with the desk application, result to be enjoyable for the kids that used it, they showed great enthusiasm for the application because it wasn't complicated to use, also, it developed an interest for programming to them.

Programming, Blocks, Language

Citación: MENDOZA-AUSTRIA, Luis, CABALLERO-TOLENTINO, Juan, MAGGI-NATALE, Carlos, HERNANDEZ-RAMIREZ, Anselmo y ARRIETA-ZUÑIGA, Alejandro. EasyProg. Revista de Tecnología Informática. 2018. 2-7: 1-5

* Correspondencia al autor (correo electrónico: mendozaaustrial@hotmail.com)

† Investigador contribuyendo como primer autor

Introducción

El proyecto consta del desarrollo e implementación de una plataforma de software la que proporciona un lenguaje de programación fácil de aprender y utilizar con el cual los niños y los jóvenes pueden introducirse al mundo de programación. El lenguaje consta de una estructura por bloques con la cual es una opción más fácil de aprender y generar una base sólida de conocimiento.

La plataforma no solo introduce a los jóvenes en la programación, también proporciona una serie de ejercicios los cuales mejorarán las habilidades de cada estudiante, conforme se vaya avanzando en los ejercicios irán subiendo de dificultad esto hará que los estudiantes vayan mejorando sus habilidades.

Contiene diferentes tipos de bloques de los cuales cada uno representa una instrucción diferente, los programadores pueden combinar los bloques y agregan las variables para poder desarrollar sus programas. Esto también hace posible que los usuarios vayan familiarizándose con las instrucciones más comunes como en los demás lenguajes de programación. La plataforma también contiene un sistema en el cual los profesores pueden dar cursos de programación a un grupo de alumnos, el docente envía ejercicios y puede aplicar exámenes para evaluar el desempeño del grupo, se contará con modalidades de límite de tiempo para cada ejercicio y finalmente se quedará registrada la calificación.

Antecedentes

La programación, es el proceso de diseñar, codificar, depurar y mantener el código fuente de programas de computadora. El código fuente es escrito en un lenguaje de programación. El propósito de la programación es crear programas que exhiban un comportamiento deseado. El proceso de escribir códigos requiere frecuentemente conocimientos en varias áreas distintas, además del dominio del lenguaje a utilizar, algoritmos especializados y lógica formal. Programar no involucra necesariamente otras tareas tales como el análisis y diseño de la aplicación (pero sí el diseño del código), aunque sí suelen estar fusionadas en el desarrollo de pequeñas aplicaciones.

La capacidad de aprendizaje de los menores es asombrosa. Su mente es como una esponja, ávida de conocimientos que se adquieren prácticamente sin esfuerzo. Por lo que es una gran oportunidad que los niños se vayan familiarizando e introduciendo en el mundo de la programación. Pues al ser menores podrán aprenderán más fácilmente la programación.

No es extraño ver como la programación se convierte en un tópico común en el contexto educativo y a su vez mundial. Día a día, proyectos, empresas y colegios incluyen la programación como parte crucial que deben abordar. El futuro tiene un norte claro y la programación es el factor protagonista, tanto como saber escribir o hablar: ya es un idioma universal. Que los alumnos programen desde el colegio puede sembrar bases positivas en las generaciones futuras.

Objetivo

Desarrollar un software para facilitar la enseñanza y aprendizaje de temas que constituyan las bases de la programación a niños de entre 9 y 15 años de edad, tal que estos puedan desarrollar un interés en el estudio de las ciencias computacionales.

Hipótesis

Lograr que la comunidad de programadores no solo crezca sino que también mejore pues los jóvenes tendrán una gran ventaja al aprender desde temprana edad a programar.

Justificación

La programación es un área que abarca una enorme cantidad de conocimientos, todos los días se desarrollan ideas nuevas y otras quedan obsoletas todo esto requiere que los programadores se actualicen diariamente con el fin de estar a la vanguardia.

Pero al mismo tiempo esto exige una mayor cantidad de tiempo para dominar todos los temas de programación, los seres humanos somos capaces de aprender más rápido a tempranas edades por lo cual es muy recomendable que desde esa edad los chicos empiecen a programar.

Metodología

El proyecto es un software el cual ofrecerá distintas funciones y dinámicas a los usuarios por lo que la metodología que se decidió ocupar es la Metodología en Cascada o Desarrollo en Cascada (ver figura 1) la cual se basa en ordenar rigurosamente las etapas del proceso para el desarrollo de software de tal manera que al inicio de cada etapa el proceso de la anterior ya debió de haber finalizado. Esta metodología es la que pareció mejor para llevar a cabo ya que al final de cada fase se puede realizar una revisión del desempeño del trabajo así como los resultados obtenidos.

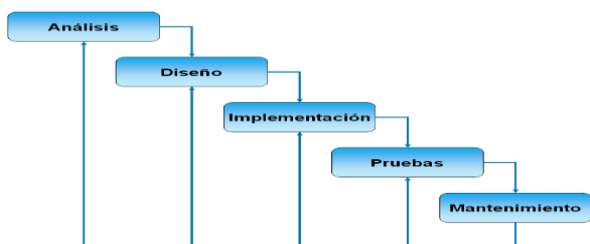


Figura 1 Metodología Cascada

Fuente: Diseño propio

Análisis de requerimientos

Para realizar este proyecto no dispondrá de materiales ya que solo se necesita computadoras o laptops en primer lugar para poder desarrollar el programa, cabe mencionar que se usaran herramientas propias de programación pero estas son software no se requiere ningún hardware especial.

Las computadoras para programar y para ejecutar el software deberán de contar con los requisitos mínimos.

La plataforma se podrá utilizar en equipos con los requisitos mínimos que se listan a continuación:

- Sistema operativo: Windows 8 en adelante
- Memoria RAM: 4 GB
- Procesador: CPU 2 GHz
- Tipo de sistema: 64 bits
- Conexión a internet: Recomendable

En esta fase se analizaron las necesidades del proyecto para determinar qué objetivos debe cubrir.

Es importante señalar que en esta etapa se consensó todo lo que se requiere del proyecto y será aquello lo que seguirá en las siguientes etapas, no pudiéndose requerir nuevos resultados a mitad del proceso de elaboración del proyecto.

Para empezar, se inició con los requisitos con interrogantes como:

¿Qué se espera del proyecto?

¿Qué se tiene para ello?

¿Qué hace falta?

En base a la metodología utilizada y con el tiempo que se tuvo, se distribuyó las múltiples actividades de la siguiente manera: (Ver tabla 1)

Nombre de la tarea	Duración
Investigación de la problemática	7 días
Investigación de los antecedentes	5 días
Describir la problemática	3 días
Identificar requerimientos	6 días
Elaboración del plan de trabajo	3 días
Elaboración de modelos	6 días
Realización del prototipo	15 días
Pruebas de su funcionamiento	7 días
Implementación del producto	13 días
Mantenimiento	16 días

Tabla 1 Cronograma de actividades

Fuente: Diseño propio

Análisis de requisitos del software: Se realizó un análisis de los requisitos fundamentales del software. El software busca enseñar los conceptos y habilidades básicas de un programador a jóvenes de temprana edad. Para poder hacer esto posible se debe buscar la manera de no solo llamar la atención de los usuarios sino también de facilitar el aprendizaje.

Diseño del sistema: Se realizó el diseño del programa el cual incluía tanto la interfaz la cual debía de ser fácil de entender y usar, así como la funcionalidad del programa mismo la que garantizaba que cumpliría con su propósito y que el usuario obtendría lo requerido.

Diseño del programa: En esta fase se realizaron los algoritmos necesarios para el cumplimiento de los requerimientos del usuario así como los análisis necesarios para saber que herramientas usar en la etapa de codificación.

Codificación: En esta fase se implementó el código fuente, haciendo uso de prototipos así como de pruebas y ensayos para corregir errores.

Pruebas: Se realizaron las pruebas para evaluar la funcionalidad del software.

Verificación: En esta fase se implementará el software y los usuarios finales lo utilizarán ya para esto se debieron de haber realizado todas las pruebas pertinentes.

Mantenimiento: Se dará el mantenimiento requerido para mantener al día al software.

Procedimientos y descripciones de actividades

En la primera parte del proyecto se realizó una lista de las características que tendría que tener la plataforma para cumplir con el objetivo de facilitar el aprendizaje de la programación. Esta lista de características (como tener una interfaz de usuario que sea intuitiva, un sistema de programación que sea fácil de entender, etc.) se utilizó posteriormente para la fase de diseño.

En este segundo procedimiento se realizó el diseño de forma general, pensando en que componentes se lograría implementar en la cantidad de tiempo que se tenía a disposición, en esta fase se obtuvo la idea general del funcionamiento del sistema de bloques y como se integraría este sistema con los problemas a resolver.

Después en este tercer paso se empezó a diseñar las características específicas de la plataforma como lo son los colores, dimensiones, controles etc.

En esta siguiente etapa se empezó la codificación de la plataforma empezando por la programación de la interfaz gráfica y a la vez de forma independiente se empezó a programar un sistema de scripting visual basado en bloques para posteriormente integrarlo dentro de la plataforma. Igualmente de forma independiente se empezó a escribir el primer “juego” que se utilizaría para presentar los problemas a resolver de una forma más agradable para el usuario.

Programación

En las siguientes figuras (figura 2, figura 3) se mostrara la programación.

```

index.html x JS index.js package.json
// La siguiente sección de funcionalidad a los botones y PopUp
let games = document.getElementsByClassName("game");
let PopUp = document.getElementsByClassName("containerPopUp")[0];
let PopUpHeader = document.getElementsByClassName("headerPopUp")[0];
let PopUpDesc = document.getElementsByClassName("descriptionPopUp")[0];
let PopUpButton = document.getElementsByClassName("mainbutton")[0];
let infoGames = [
  {name: "laberinto", des: "Un laberinto es un pasatiempo gráfico consistente en trazar un
}
for(let i =0; i<games.length;i++){
  games[i].addEventListener("mouseover",function(ev){
    games[i].classList.add("animated");
    games[i].classList.add("pulse");
  });
  games[i].addEventListener("mouseout",function(ev){
    games[i].classList.remove("pulse");
    games[i].classList.remove("animated");
  });
  games[i].addEventListener("click",function(ev){
    if(infoGames[i]!==undefined){

```

Figura 2 Programación

Fuente: Diseño propio

```

index.html x JS index.js package.json
<script>
// Esta primera parte controla los botones de la barra de titulo
const {remote} = require('electron');
const main = remote.getCurrentWindow()
let titleBar = document.getElementsByTagName("li")
titleBar[0].addEventListener("click",function(){
  main.minimize();
});
titleBar[1].addEventListener("click",function(){
  if (!main.isMaximized())
    main.maximize()
  else
    main.unmaximize()
});
titleBar[2].addEventListener("click",function(){
  main.close();
});
let back = document.getElementById("buttonBack");
back.addEventListener("click",function(ev){
  console.log("v")
  window.history.back();
});

```

Figura 3 Programación

Fuente: Diseño propio

Implementación

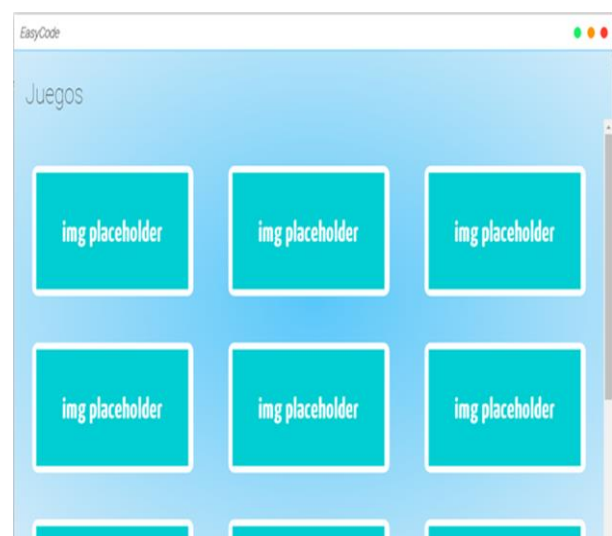


Figura 4 Primer diseño de interfaz

Fuente: Diseño propio

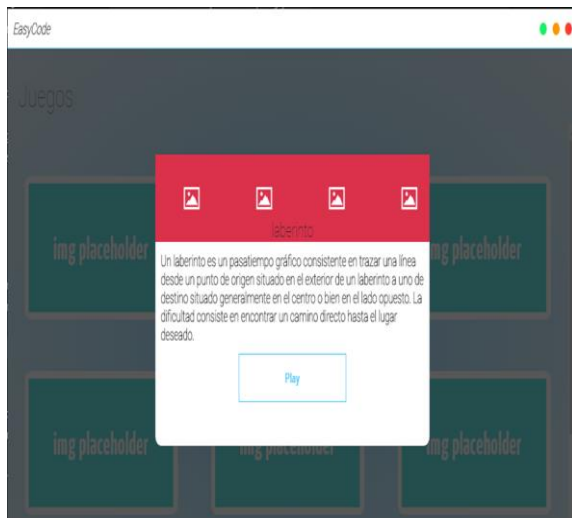
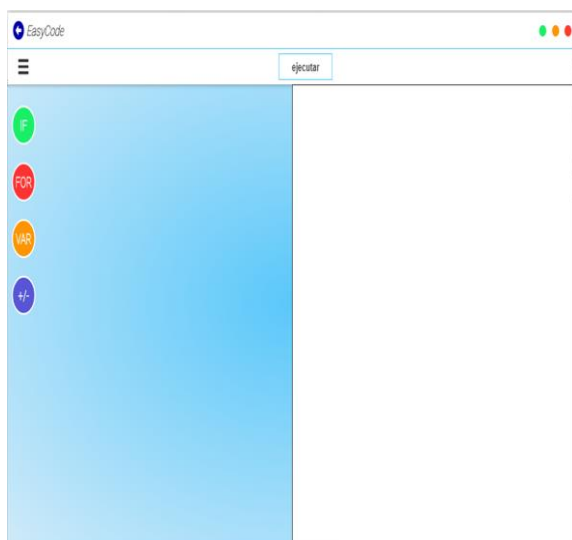


Figura 5 Breve descripción del juego seleccionado
Fuente: Diseño propio



En el primer diseño cada pantalla para los juegos esta dividida en dos partes la parte izquierda donde se podrá añadir código en forma de bloques y la parte derecha donde se encontrará la pantalla de juego, y entre las dos un botón ejecutar que servirá para que el usuario vea su código en el juego

Figura 6 Otra vista de la interfaz
Fuente: Diseño propio

Conclusiones

Después de realizar este proyecto, se llegó a la conclusión de que es muy importante tener los conocimientos adecuados para tener una buena y correcta implementación en el diseño y desarrollo de software, ya que esto juega un papel de gran importancia en la vida laboral.

Actualmente el mundo cambia y se actualiza a cada instante y es indispensable que como ingenieros se tengan las habilidades necesarias para poder diseñar, elaborar y corregir proyectos de software como este.

De igual manera se concluye que es importante aprender a programar a temprana edad, pues los niños aprenden con mayor facilidad y además de que ellos tendrán muchas más posibilidades de crecer y mejorar como programadores.

Recomendaciones

Se recomienda a futuro desarrollar más juegos aptos para niños y jóvenes para que sean implementados en el software con la finalidad de que no caiga en la obsolescencia y por esa razón siga siendo de interés para las nuevas generaciones.

Referencias

Joyanes Aguilar, Luis . (2008). Fundamentos de Programación. Algoritmos Estructura de Datos y Objetos / 4 ed.. España: Mc Graw Hill

Joyanes Aguilar, Luis. (2012). Fundamentos Generales de Programación. España: Mc Graw Hill .

Juan Carlos Casale. (2016). Introducción a la programación. España: Mp edicion.

Tutorialspoint. (2016). Software - Diseño de UI. 1 de marzo de 2018, de tutorialspoint Sitio web: https://www.tutorialspoint.com/es/software_engineering/software_user_interface_design.htm

Universia Argentina. (23 de junio de 2015). 7 consejos para enseñar a programar en el aula. 28 de Febrero de 2018, de universia.com Sitio web: <http://noticias.universia.com.ar/consejos-profesionales/noticia/2015/06/23/1127162/7-consejos-ensenar-programar-aula.html>