



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**“SAEPRI: SISTEMA DE AUTO ENSEÑANZA DE
PROGRAMACIÓN INTERACTIVA”**

INFORME DE PROYECTO INTEGRADOR

Previa a la obtención del Título de:

**INGENIERO EN CIENCIAS COMPUTACIONALES
ORIENTACIÓN SISTEMAS MULTIMEDIA**

STELLA CAROLINA ANDRADE ARTEAGA

KEYLA ESTEFANIA FIGUEROA ASTUDILLO

GUAYAQUIL – ECUADOR

AÑO: 2016

AGRADECIMIENTOS

Agradezco a quienes creyeron en nosotras, y a quienes siguen creyendo. A quienes están delante de nosotros en el camino y aun así nos apoyaron. Por sobre todo, a mi compañera de proyecto, por acompañarme sin falta en este trayecto de la vida.

Stella Andrade Arteaga

Agradezco a Dios por ser mi mayor fuente de inspiración, mi motor para continuar durante cada obstáculo en mi vida. A mis padres por su paciencia, apoyo incondicional, porque con sabiduría supieron corregirme y aconsejarme para ser una persona de bien. Agradezco a mi abuelita que siempre está orando por mi salud y bienestar. A mis hermanos por todos los momentos vividos. A mis tíos y primos por el aliento y la buena vibra. Mis sinceros agradecimientos también a mis amigos por ser mis cómplices y brindarme su mano cuando lo necesitaba.

Agradezco a mis maestros por su gran labor académica.

A mi compañera de proyecto por su esfuerzo, dedicación y por confiar en mí en el transcurso de nuestra vida universitaria

Keyla Figueroa Astudillo

DEDICATORIA

Dedicado a aquellos que necesiten aquella validación de sus motivaciones, a aquellos que necesitan esa chispa para continuar. Por sobretodo, dedicado a aquellos que apenas empiezan, a los 'niños' de todas las edades que desean empezar aprendiendo algo totalmente nuevo para ellos.

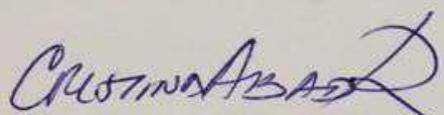
Stella Andrade Arteaga

Dedicado a Dios quien hace sentir su presencia en mi vida día a día. Dedicado a mi familia que son mi principal felicidad.

Dedicado a aquellos que se dedican a esta área para que este trabajo sirva de inspiración en la innovación de las formas de aprendizaje y preparación educativa de las futuras generaciones.

Keyla Figueroa Astudillo

TRIBUNAL DE EVALUACIÓN



PhD. Cristina Abad Robalino

PROFESOR EVALUADOR



PhD. Xavier Ochoa Chehab

PROFESOR EVALUADOR

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

Stella Andrade

Stella Andrade Arteaga

Keyla Figueroa

Keyla Figueroa Astudillo

RESUMEN

En este proyecto presentamos el diseño e implementación de un sistema para el autoaprendizaje de programación, llamado Sistema de Auto Enseñanza de Programación Interactiva. SAEPRI fue propuesto para solucionar el problema de la falta de atención en la enseñanza de temas relacionados a la programación y conceptos derivados de las ciencias computacionales en la educación temprana. SAEPRI busca solucionar el problema permitiendo que el usuario conozca un poco acerca del área a tratar y se incentive a profundizar en los temas que involucra el mundo de la computación. Para su implementación se usó Blockly como el concepto principal del proyecto, en forma de juego de puzzle, tratando de esta manera de lograr captar la atención del usuario. En el futuro se espera agregar más tópicos e incluir medallas o un tipo de premiación que enganche al usuario a continuar interactuando con la plataforma.

ÍNDICE GENERAL

AGRADECIMIENTOS.....	ii
DEDICATORIA	iii
TRIBUNAL DE EVALUACIÓN	iv
DECLARACIÓN EXPRESA.....	v
RESUMEN	vi
ÍNDICE GENERAL.....	vii
CAPÍTULO 1	1
1. LA PROGRAMACIÓN COMO MÉTODO DE APRENDIZAJE	1
1.1. Causas.....	1
1.2. Efectos	2
1.3. Soluciones Similares.....	2
1.3.1. StarLogo	2
1.3.2. Scratch	3
CAPÍTULO 2.....	5
2. DISEÑO DE LA SOLUCIÓN.....	5
2.1. Arquitectura de la solución	6
2.1.1. Diagrama de Arquitectura.....	6
2.1.2. Diagrama de Casos de Uso.....	6
2.1.3. Diseño de Prototipos	7
2.2. Tecnologías	8
2.2.1. Aplicación Web.....	8
2.2.2. Blockly	8
2.2.3. Otras librerías JavaScript	9
2.2.4. Web Server.....	9
CAPÍTULO 3.....	10

3. DESARROLLO DE LA SOLUCIÓN: GEORGE, TU ASISTENTE DE APRENDIZAJE.....	10
3.1. Asistente del juego	10
3.2. Niveles de dificultad.....	11
3.2.1. Nivel Básico.....	11
3.2.2. Nivel Intermedio.....	12
3.2.3. Nivel Avanzado.....	13
3.3. Puntuaciones.....	13
CONCLUSIONES Y RECOMENDACIONES	15
BIBLIOGRAFÍA.....	16

CAPÍTULO 1

1. LA PROGRAMACIÓN COMO MÉTODO DE APRENDIZAJE

En Ecuador es poco común que en las escuelas primarias y secundarias se enseñen nociones básicas de computación y programación, limitándose a enseñar lo básico del hardware y software comúnmente usados (y hasta obsoletos). Esta es una limitante para la sociedad actual con la vigente economía del conocimiento, donde lo que se busca es generar soluciones automatizadas como medios que permitan facilitar la comunicación e interacción.

Estudios y profesionales en el área de computación [1]-[2] han demostrado que profundizar en el aprendizaje de programación ayuda a desarrollar el pensamiento lógico en los niños, fomenta la creatividad, estimula el pensamiento, y despierta el interés por los detalles y minuciosidad en los niños.

Por ello es importante incentivar la incorporación de programación como una asignatura más dentro del plan académico. La enseñanza temprana de programación no solo ayuda a mejorar las habilidades cognitivas, sino también que los menores se familiaricen a las nuevas tecnologías y medios programables a los que se encuentren rodeados.

Lo que se busca otorgar con nuestra solución es una plataforma libre y accesible para la comunidad, donde, de una forma divertida, se enseñe a los niños las bases y nociones de la computación y programación: lógica, algoritmos, diagramas de flujo, y componentes básicos de programación.

1.1. Causas

En la actualidad el avance de la tecnología y el acceso de sistemas automatizados es un tema que involucra a todas las generaciones. Niños y adultos están interactuando día a día con objetos tecnológicos, muchas veces sin cuestionarse lo que está sucediendo mientras realizan ciertas acciones en sus dispositivos.

El problema principalmente está en que las instituciones educativas por lo general no incluyen este tipo de instrucción en la enseñanza de los niños y el desinterés que existe en avivar la curiosidad de estos conceptos en la educación temprana.

1.2. Efectos

Uno de los efectos de no instruir tempranamente a los niños en programación es que los limita al uso de las ofertas del mercado, de sólo depender de las soluciones ya existentes. Además de desconocer las vías para crear sus propias soluciones.

A largo plazo, la falta de conocimiento en estas áreas que se encuentran en constante desarrollo implica una pérdida valiosa de futuros profesionales en computación y emprendedores de soluciones tecnológicas.

1.3. Soluciones Similares

1.3.1. StarLogo

StarLogo [3] es una **aplicación** de escritorio desarrollado en Java y C.

StarLogo se basa en el modelamiento y simulación del entorno natural, usando gráficos 3D, sonidos y bloques que deben ser colocados en orden, como en forma de rompecabezas.



Figura 1.1 Banner de StarLogo [3]

1.3.2. Scratch

Scratch [4] se ofrece de forma gratuita, presentando una interfaz de programación fácil e intuitiva, donde cada acción se apila o agrupa como bloques de lego en pseudocódigo. Está dirigido para edades entre los 8 y 16 años.



Figura 1.2 Banner de Scratch [4]

1.3.3. Code.org

Code.org es una organización sin fines de lucro que busca expandir el acceso a las ciencias computacionales en todos los currículos de las primarias.

Code.org [5] lanzó “Hora de Código”, que promueve a las personas disciplinarse en las ciencias computacionales dedicando una hora de estudio, a través de los tutoriales y guías que la organización ofrece.

Esta organización es muy popular por su participación en el desarrollo de algunos personajes de Disney y juegos como Angry Birds, Minecraft, entre otros.



Figura 1.3 Banner de “Hora del Código” [5]

El documento presente está organizado de la siguiente manera. En el Capítulo 2 se presentará la metodología de la solución desarrollada, explicando las bases que se tomaron en cuenta para la implementación del proyecto, así como las especificaciones técnicas. Luego, en el Capítulo 3, se muestran los resultados de la aplicación. Al final de este documento se detallan las conclusiones del proyecto, así como recomendaciones y trabajos a futuro.

CAPÍTULO 2

2. DISEÑO DE LA SOLUCIÓN

La solución está pensada para fines académicos, dirigida a todos los usuarios que desconozcan del área de computación, pero enfocada principalmente a niños de primaria (de 6 a 10 años). Por ello, la aplicación fue concebida como un juego, para así ganar la atención de los usuarios más pequeños, motivándolos a continuar profundizando en el estudio de la computación.

El siguiente paso que se tomó para desarrollar la solución es la accesibilidad, siendo la base necesaria por asentar para que la solución alcance a una cantidad considerable de usuarios. Aunque las aplicaciones de escritorio poseen sus propias ventajas sobre las aplicaciones web o móviles y viceversa [7], al final se optó por realizar una aplicación web interactiva por las siguientes razones:

- No existiría la necesidad de instalar una aplicación más en el dispositivo, por lo que no abarcaría espacio ni consumiría recursos del usuario.
- Estaría desarrollado con los últimos estándares web, por lo que no habrían muchas discrepancias entre los distintos navegadores existentes, volviéndose una aplicación portable.
- La tecnología responsive haría posible la visualización de la aplicación no solo en un navegador de escritorio, sino también en dispositivos móviles como tablets.
- Disponibilidad de librerías javascript que facilitaban el desarrollo de la solución.
- Posibilidad de volver la aplicación multijugador en línea, con puntuaciones globales.

De las soluciones similares encontradas, tomamos inspiración de Scratch y su forma de programar en bloques, después de encontrar la librería para JavaScript llamada Blockly, desarrollada por Google y que también fue inspirada por Scratch.

Así, fue desarrollándose el juego como un juego de puzzles, donde el jugador tuviera que seguir las indicaciones de cada nivel para lograr el objetivo. Se dividió el juego

en tres dificultades o áreas de estudio, las cuales comprenden tres de los conceptos base necesarios en la computación.

De forma más específica, la solución, en primera instancia, fue desarrollada como una aplicación web estática sin el uso de un framework en particular, usando HTML5 para la estructura, CSS3 para los estilos, JavaScript y múltiples librerías para el dinamismo. Luego, para el componente multijugador, se agregó un webservice basado en Java, una base de datos MySQL, y como servidor web Apache Tomcat.

2.1. Arquitectura de la solución

2.1.1. Diagrama de Arquitectura

La Figura 2.1 muestra el diseño de arquitectura a la solución propuesta donde los usuarios, con previa conexión a internet, se comunican con el servidor que envía y recibe la información de perfil de los usuarios registrados en la base.

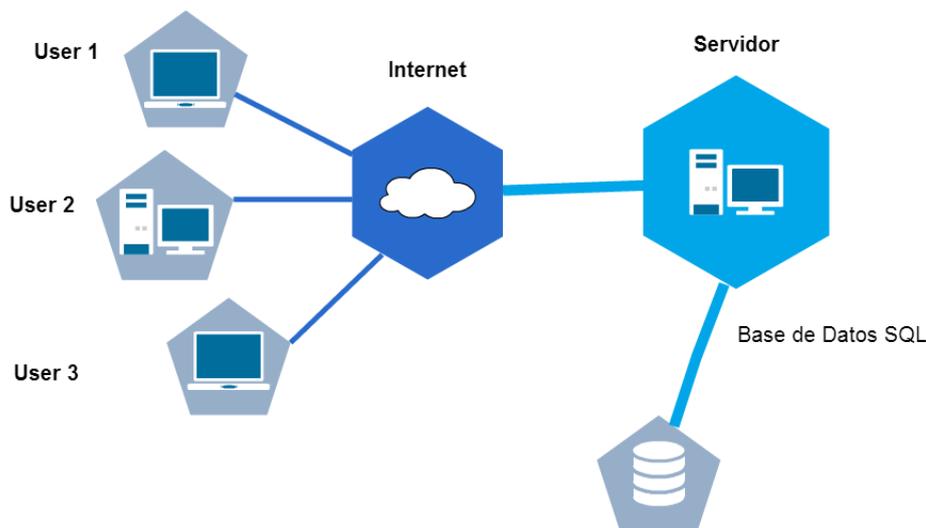


Figura 2.1 Diseño de arquitectura de la solución

2.1.2. Diagrama de Casos de Uso

La Figura 2.2 muestra un resumen de las funciones con las que el usuario puede interactuar con el sistema, y cómo éste sistema le responde.

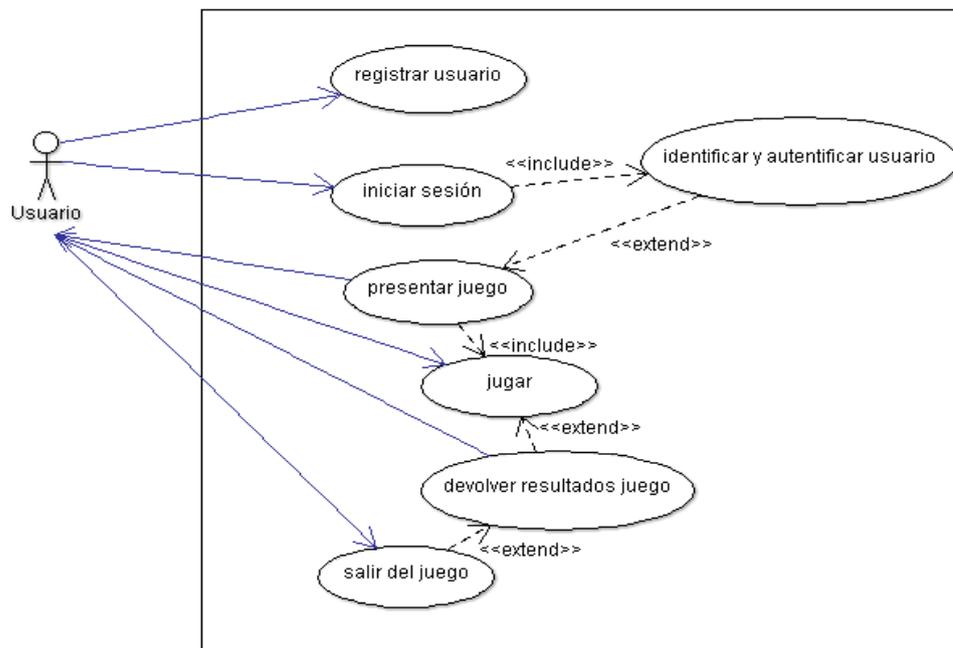


Figura 2.2 Diagrama de Casos de Uso

2.1.3. Diseño de Prototipos

Ya para el diseño gráfico en sí de la aplicación, se diseñaron mock-ups de las pantallas del juego. En la Figura 2.3 se muestra el prototipo de la pantalla principal de un juego: en la barra superior estaría la información del usuario, con su nombre de usuario, puntuación, enlaces para configuración y cerrar sesión; y en el canvas central se dejó el espacio a un costado para que se ejecuten las animaciones al completar correctamente un juego.

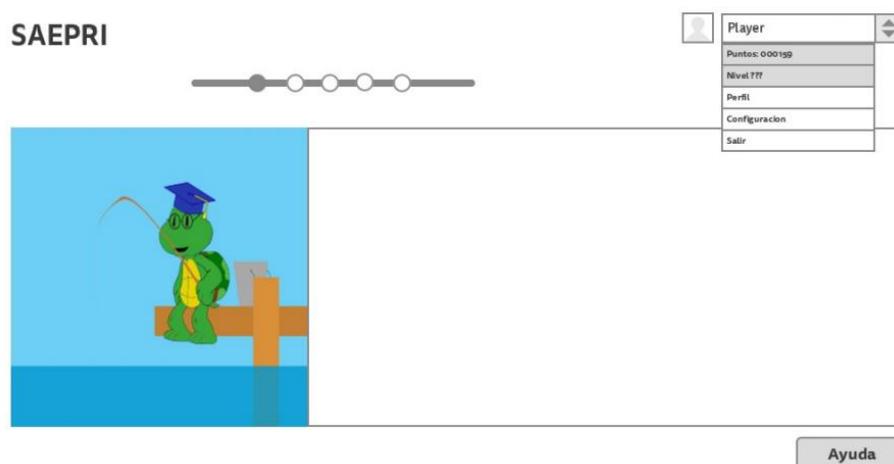


Figura 2.3 Mock-up de la pantalla principal de un juego

2.2. Tecnologías

2.2.1. Aplicación Web

Los estándares web HTML5 y CSS3 fueron utilizados para el desarrollo y diseño de la aplicación web. Adicionalmente se utilizó el framework Bootstrap [7] para el diseño responsive del sistema.

JavaScript fue usado para la funcionalidad e interacción del sitio web con el usuario, no solo en su forma original o JQuery, si no con el uso de librerías especializadas detalladas a continuación.

2.2.2. Blockly

Desarrollado por Google, Blockly [8] es una librería diseñada para construir editores de programación visuales. Los bloques previamente creados con código de programación detrás (ya sea en JavaScript, PHP, Python, o algún otro) se ajustan uno con otro, al final devolviendo un bloque de código ordenado y funcional, y a veces hasta ejecutándolo. Por sí solo, Blockly no fue diseñado para enseñar a programar a niños, pero sí sirve para desarrollar aplicaciones y juegos con esos fines, generando bloques específicos para las actividades que los niños deben

resolver, acompañado de librerías multimedia que muestran los resultados.

2.2.3. Otras librerías JavaScript

Al enfocarse en la parte visual e interactiva de la aplicación, se usaron librerías que manejen sonido e imágenes en la web. Así, se usó la librería SoundJS [9] para manejar los sonidos de respuesta del juego, y la librería FabricJS [10] para el manejo de imágenes en el canvas con los que responde el juego.

Una librería indispensable en la etapa de desarrollo fue soap client, para realizar las peticiones al web services.

2.2.4. Web Server

Del lado del servidor, el sitio web fue alojado en el contenedor web Apache Tomcat. Los perfiles del usuario, así como los logs del juego se encuentran almacenados en el motor de base de datos MySQL.

En cuanto la comunicación del servidor con el usuario, los métodos del web service fueron desarrollados en Java utilizando el protocolo de acceso Soap basado en el formato XML.

Para la persistencia de datos de sesión de un usuario, se usaron las cookies de los navegadores, las cuales mantienen la información temporalmente en el navegador.

CAPÍTULO 3

3. DESARROLLO DE LA SOLUCIÓN: GEORGE, TU ASISTENTE DE APRENDIZAJE.

Como resultado a la propuesta se creó como solución el Sistema de Auto Enseñanza de Programación Interactiva, llamado desde ahora en este documento como SAEPRI por sus siglas. En esta plataforma, se levantarían juegos tipo puzzle para que los niños aprendan los conceptos básicos de la programación y las ciencias computacionales de una forma diferente y entretenida. El objetivo de la solución no es principalmente que los jugadores terminen perfilándose como futuros programadores, si no que se despierte en ellos la curiosidad por conocer más, y se desarrolle su pensamiento lógico-matemático.

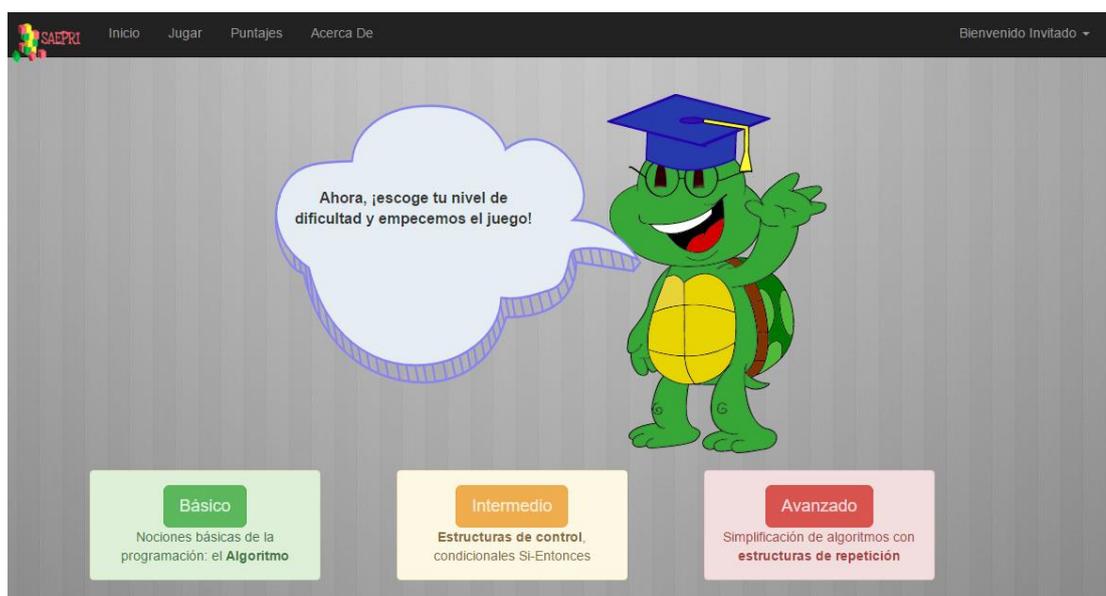


Figura 3.1 Pantalla de bienvenida al usuario.

3.1. Asistente del juego

Al ser una aplicación de entretenimiento dirigida hacia niños de escuela, fue necesario diseñar una mascota entrañable y acorde al tema propuesto.

Nombrado “George”, esta tortuga politécnica será quien explique los conceptos por aprender en cada nivel de dificultad; indicará en cada juego lo que se necesita cumplir para pasar con éxito, e indicará también cuando el jugador se equivoca o acierta correctamente.



Figura 3.2 George, el guía de aprendizaje en SAEPRI.

3.2. Niveles de dificultad

SAEPRI en su versión actual consta de tres niveles, que van del nivel fácil a difícil. Cada dificultad está determinada por temas básicos, característicos de la computación. Actualmente, cada nivel contiene dos lecciones con problemas diferentes, referente a los tópicos que van a estudiarse en el nivel.

3.2.1. Nivel Básico

El primer nivel consta de juegos basados en el concepto de algoritmo, donde el jugador tiene que ordenar de forma correcta los bloques establecidos en el juego para que los pasos se ejecuten en orden. En la Figura 3.3 se ve la captura de la pantalla inicial del Nivel Básico, donde George el galápago explica brevemente el concepto de algoritmo, acompañado de un ejemplo usando los bloques del juego.



Figura 3.3 Pantalla principal del nivel básico

3.2.2. Nivel Intermedio

El segundo nivel consta de juegos basados en los conceptos de las estructuras condicionales. Se presentan nuevos bloques de condición, donde se tendrá que poner la acción a realizar dependiendo si un evento ocurre o no. En la Figura 3.4 se ve la captura de la pantalla inicial del Nivel Intermedio, donde se explica brevemente el concepto de las estructuras condicionales y los bloques Sí-Entonces.

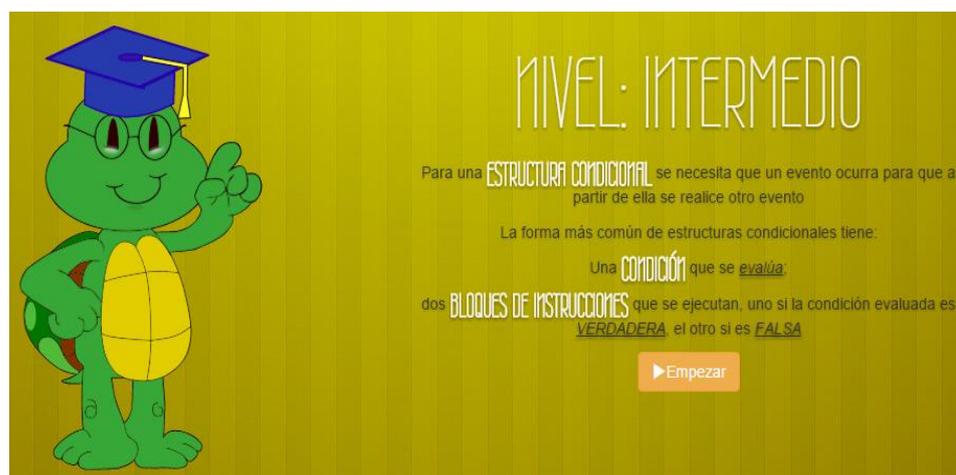


Figura 3.4 Pantalla principal del nivel intermedio

3.2.3. Nivel Avanzado

El último nivel consta de juegos basados en los conceptos de estructuras iterativas o bucles. Como en el nivel Intermedio, el nivel Avanzado presenta un nuevo tipo de bloque que maneja los bucles. En la Figura 3.5 se muestra la pantalla principal del Nivel Avanzado, donde se explica brevemente el concepto de estructuras iterativas.



Figura 3.5 Pantalla principal del nivel avanzado

3.3. Puntuaciones

Para cada lección existe un puntaje máximo definido de acuerdo al número de movimientos exactos para completar un objetivo en un tiempo límite de un minuto.

Si el jugador cumple el objetivo sin cometer errores en un tiempo menor al planteado, tendrá una bonificación de 120 puntos.

Existen varios casos de penalizaciones; cuando el jugador comete errores, cuando el jugador completa el juego en un tiempo mayor al establecido, cuando el jugador realiza una cantidad de movimientos fuera del umbral del movimiento esperado para ese juego. La penalización será de -5 puntos, de acuerdo a la cantidad de acciones cometidas según el caso. En el caso de que la penalización sea por superar el tiempo límite, se reducirá 50 puntos del puntaje establecido para la lección.

Cuando el jugador complete el juego de una lección en varios intentos, se toma en cuenta el juego con mayor puntaje de aquella lección.

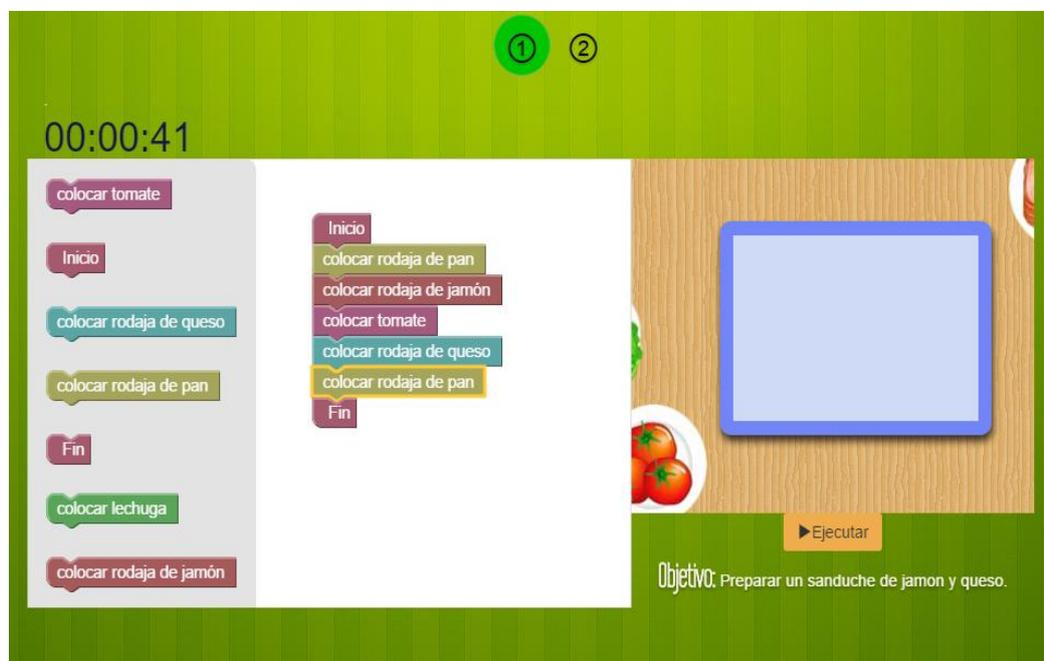


Figura 3.6 Pantalla del juego

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1. Si bien ya existen múltiples soluciones similares a la planteada en este documento -ya sea en la red u otorgado por otras empresas-, sería adecuado adaptar cada solución a la necesidad del usuario final, realizando actividades y juegos especializados para los niños de nuestro país.
2. Las ciencias computacionales pueden descubrir un gran potencial en la niñez, dando la oportunidad de formar adultos emprendedores e investigadores. Proyectos como este deben trabajar para que no solo los niños se interesen en las ciencias computacionales, sino también los mismos profesores, para que agreguen la materia al pènsum académico en las escuelas.

Recomendaciones

1. A lo que se podría aspirar con proyectos como este es el apoyo de organizaciones y ministerios encargados y preocupados por la educación de los niños y jóvenes, los cuales podrían hacer estudios, no sólo de mercado, sino también del pènsum académico necesario, y así desarrollar los juegos que abarquen las necesidades del alumnado. Ser una solución puramente académica sin fines de lucro es lo que se espera que este proyecto alcance a ser.
2. A futuro, se podría mejorar la interfaz para que tenga un diseño más uniforme, atractivo, y ligero, acorde con el tema de la aplicación.
3. Otra de las mejoras que pueden realizarse es desarrollar un backend para facilitar y hacer uniforme la subida de más juegos por nivel.

BIBLIOGRAFÍA

- [1] J. C. López García, *“Algoritmos y Programación (Guía para docentes)”*, segunda edición, Ed. Eduteka, 2007, 2009.
- [2] *Computational Thinking Leadership Toolkit*, 1st ed.: CSTA and ISTE, 2011.
- [3] (2008) StarLogo [Online]. Disponible en:
http://education.mit.edu/portfolio_page/starlogo-tng/
- [4] (2005) Scratch [Online]. Disponible en: <https://scratch.mit.edu/>
- [5] (2015) Code.org [Online]. Disponible en: <http://code.org>
- [6] D. Bychkov (2013, Junio 07). *“Desktop vs. Web Applications: A Deeper Look and Comparison”* [Blog Online]. Disponible en: <http://www.seguetech.com/blog/2013/06/07/desktop-vs-web-applications-deeper-comparison>
- [7] (2011) Bootstrap [Online]. Disponible en: <http://getbootstrap.com/>
- [8] (2012) Blockly [Online]. Disponible en: <https://developers.google.com/blockly/>
- [9] (2014) SoundJS [Online]. Disponible en: <http://www.createjs.com/soundjs>
- [10] (2014) FabricJS [Online]. Disponible en: <http://fabricjs.com/>