UNIVERSIDAD Rafael Belloso Chacín
Revista Electrónica de Estudios Telemáticos
TÉLÉMATIQUE

Ecosistema de Hardware y Software para la introducción de la robótica a niños en etapa escolar

Hardware and software ecosystem for the introduction of robotics to children in school stage

Rodriguez, Paola
Universidad Rafael Urdaneta
paorodriguez24@gmail.com

Parra, Samuel
Universidad Rafael Urdaneta
samuelparra1@gmail.com

Resumen

La investigación tuvo como objetivo el desarrollo de un ecosistema de hardware y software, con el fin de introducir los conocimientos de la robótica a los niños en etapa escolar a través de una aplicación local y su prototipo. El sistema en su totalidad tiene como finalidad mejorar la manera en la que se le presenta la tecnología a los niños, ya que el mismo provee funciones que facilitan el entendimiento de distintas áreas tecnológicas, puesto que además de poder observar el funcionamiento de un robot, estos podrán también programarlo de manera que el robot realice las funciones que ellos deseen. El sistema está basado en la plataforma RaspberryPi, y con el funcionan un conjunto de módulos que permiten su funcionamiento, como lo son el módulo de cómputos proporcionado por el RaspberryPi, ya que el mismo se conecta a la aplicación la cual vendría siendo el módulo de software y un módulo de componentes electrónicos que sería el prototipo como tal. Se basó en autores como Arias, F. (2006), Hernandez, R., Fernandez, C. y Baptista, P. (2014), Hurtado, J. (2010), IBM (2014), Oracle (2018), Bernal (2020) y Sabino (2007) para el nivel, diseño y unidad de análisis de la presente investigación. Se utilizaron lenguajes como Python para la programación del RaspberryPi y del servidor. Javascript y HTML para la aplicación local. Se cumplió en su totalidad con los objetivos de la investigación. Con los resultados obtenidos los investigadores comprobaron que, si se imparten los conocimientos del área robótica a los niños a través de sistemas, se mejora el aprendizaje y la calidad de educación tecnológica impartida.

Palabras Claves: Robótica, RaspberryPi, Automatización, Educación.

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 18 Edición No 2 – Año 2019

Abstract

The research aimed at the development of a hardware and software ecosystem, in order to introduce the knowledge of robotics to school children through a local application and its prototype. The system as a whole aims to improve the way in which technology is presented to children, since it provides functions that facilitate the understanding of different technological areas, since, in addition to being able to observe the operation of a robot, they can also program it so that the robot performs the functions they want. The system is based on the RaspberryPi platform, and with it work a set of modules that allow its operation, such as the computation module provided by the RaspberryPi, since it connects to the application which would be the module of software and an electronic component module which would be the prototype as such. Authors such as Arias, F. (2006), Hernandez, R., Fernandez, C. and Baptista, P. (2014), Hurtado, J. (2010), IBM (2014), Oracle (2018), Bernal (2020) and Sabino (2007) were based on the theoretical part for the level, design and unit of analysis of the present investigation. Languages such as Python were used for RaspberryPi and server programming. Javascript and HTML for the local application. The objectives of the research were fully met, with these results the researchers verified that if knowledge of the robotics area is imparted to children through systems, learning and the quality of technological education are improved.

Keywords: Robotics, RaspberryPi, Automation, Education.

Introducción

La robótica es una rama de la tecnología que estudia el diseño y construcción de máquinas capaces de desempeñar tareas realizadas por el ser humano o que requieren del uso de inteligencia. También es conocida como robótica pedagógica, una disciplina que tiene por objeto la concepción, creación y puesta en funcionamiento de prototipos robóticos y programas especializados con fines pedagógicos (Ruiz-Velasco, 2007). Esta modalidad de la educación crea las mejores condiciones de apropiación de conocimiento que permite a los estudiantes fabricar sus propias representaciones de los fenómenos del mundo que los rodea, facilitando la adquisición de conocimientos acerca de estos fenómenos y su transferencia a diferentes áreas del conocimiento.

La presente investigación tuvo como finalidad desarrollar un sistema para la introducción de la robótica a niños en etapa escolar. El mismo se basó en la tecnología de la robótica, con la finalidad de que los niños puedan manejar y tener contacto directo con esta tecnología. Se plantearon los objetivos lograr su desarrollo y mostrar así los resultados de los mismos, su vez, plantea el desarrollo de este sistema con el uso de una metodología muy efectiva, la cual permite llevar un control sobre el proceso de desarrollo, trabajar colaborativamente y obtener los mejores resultados finales.

De igual forma, para el desarrollo del sistema se hizo una revisión bibliográfica de distintos autores, los cuales sirvieron para guiar el proceso de la investigación y obtener un soporte para lograr los objetivos de la misma, logrando el desarrollo del sistema que favorece a los niños en etapa escolar que estarán manejándolo, con funciones e interfaces amigables y fáciles de comprender para su adecuado uso.

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 18 Edición No 2 – Año 2019

Objetivo general

Desarrollar un Ecosistema de Hardware y Software para la Introducción de la Robótica a niños en etapa escolar.

Metodología de la investigación

Este trabajo de investigación encaja en una investigación tipo proyectiva. Según Hurtado (2010), la investigación de tipo proyectivo tiene como objetivo diseñar o crear propuestas dirigidas a resolver determinadas situaciones. Tomando en cuenta lo anterior, se considera esta investigación de tipo proyectiva ya que es una planificación del área de ingeniería y este posee características como el diseño de programas informáticos elaborada principalmente con los lenguajes de programación JavaScript, HTML y Python, de manera que sean interactivos y de amigable manejo para los niños, siendo así orientados al desarrollo social, y por lo tanto conllevan a un desarrollo tecnológico para la sociedad.

Por otro lado, la presente investigación se enmarca en un diseño de campo no experimental, ya que la misma fue desarrollada obteniendo los datos de fuentes directas en el contexto natural, observando los eventos en un ambiente espontaneo y sin alterar o intervenir en el ambiente de estudio. Para la investigación por lo que se desarrolló una unidad de análisis para la misma; es decir, un sistema, el cual consiste en una aplicación local con fines interactivos junto con la elaboración de un prototipo en físico, los cuales en conjunto se requirieron para lograr los objetivos previamente planteados durante esta investigación. Por este motivo, para la recolección de los datos se planteó una técnica como lo fue la aplicación de una encuesta. Esta se realizó para corroborar si el sistema permitió la introducción de conocimientos básicos de robótica a los niños.

En cuanto a la metodología de desarrollo seleccionada para la realización de la presente investigación, se consideró la metodología Scrum, la cual según Schwaber, K. y Beedle, M. (2002), es un proceso ágil para desarrollar software que fue aplicado por primera vez por Schwaber y Sutherland, en donde se emplean de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Para lograr los objetivos a través de esta metodología, luego de haber realizado un proceso de planificación, se comenzó a trabajar a través de ciclos llamados Sprints, en los cuales se realizan ciclos completos de análisis, diseño, desarrollo y finalmente una fase de pruebas.

Análisis de los resultados

En esta sección se realizó un análisis de los resultados obtenidos en las fases de la investigación definidas para este estudio. En primera instancia se plantearon los requerimientos mínimos a utilizar en este sistema, seguido por la estructuración de los componentes del hardware y del software, además del desarrollo de los algoritmos empleados en la programación. Para finalizar, se muestran algunas simulaciones ilustrativas junto con los diagramas utilizados dentro de su documentación.

Requerimientos del Sistema: Con respecto a los requerimientos físicos mínimos que se deben tener para que el sistema pueda funcionar adecuadamente, primero se debe contar con un dispositivo **(**RaspberryPi-módulos) el cuál se encarga de la conexión entre el módulo de

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 18 Edición No 2 – Año 2019

software y el de componentes electrónicos. Luego, una computadora que cuente con un procesador Dual-Core para poder correr el sistema y por último una conexión a internet.

Arquitectura del Sistema: La International Business Machines ([IBM], 2014) definió la arquitectura del sistema como una representación de un sistema en la que hay una correlación de funciones con componentes de hardware y software, una correlación de la arquitectura de software con la arquitectura de hardware, e interacción humana con estos componentes. Se utilizó la arquitectura cliente servidor, ya que es la que mejor se adapta al funcionamiento y flujo del sistema. Se planteó desde un principio este tipo de arquitectura, ya que la misma encaja perfectamente en el desarrollo del sistema, el cual consta de un cliente que envía un mensaje solicitando un servicio, es decir, realiza una petición, y el servidor se encarga de recibir ese mensaje para luego proporcionar una respuesta. En la figura 1 se muestra en el diagrama de conexiones pertinentes a la arquitectura

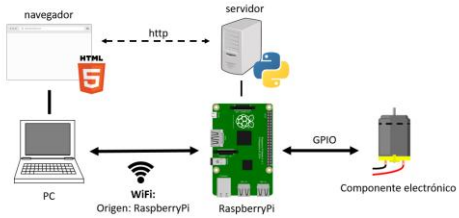

Figura 1. Diagrama de conexiones
Fuente: Elaboración propia (2018)

Seguidamente, se realiza un esquema electrónico conformado por los componentes que integran el sistema, donde se evidencia una computadora, la cual aloja a través del navegador web el software del sistema. Luego se observa el servidor creado en lenguaje python, el cual se encuentra alojando dentro del RaspberryPi, quién se encarga de enviar las instrucciones a los componentes electrónicos utilizados en la arquitectura propuesta. En la figura 2 se aprecia el esquema electronico sugerido.

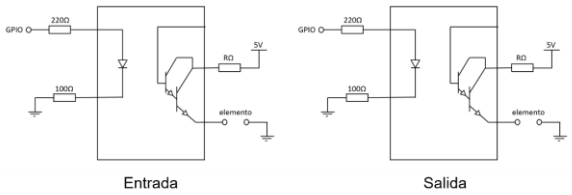

Figura 2. Esquema electrónico
Fuente: Elaboración propia (2018)

Es importante señalar, tal y como se muestra en la figura 3, que se utilizó el standard del Unified Model Language (UML), conocido en español como lenguaje unificado del modelado o lenguaje de modelo unificados, para la diagramación de las interacciones que se darían en cada interfaz, la misma se presenta a continuación, mostrando la interfaz de inicio la cual le indica al usuario los proyectos ya anteriormente creados y le da las opciones de crear un nuevo proyecto y abrir o eliminar un proyecto ya existente.

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 18 Edición No 2 – Año 2019

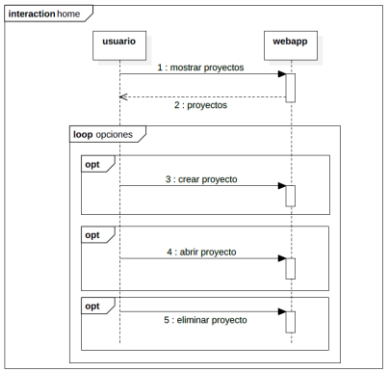

Figura 3. Diagrama de secuencias para explicar el funcionamiento de la interfaz de inicio.
Fuente: Elaboración propia (2018)

Seguido a esto, en la figura 4, se tiene el diagrama de secuencias de la interfaz de crear proyecto para explicar el funcionamiento de la interfaz según lo que serían las peticiones y respuestas que se estarían manifestando dentro del sistema, apegándose al estándar Unified Model Language (UML), en el diagrama se mostrará la interacción del usuario con la interfaz y a su vez, como el usuario puede a través de ella crear un proyecto.

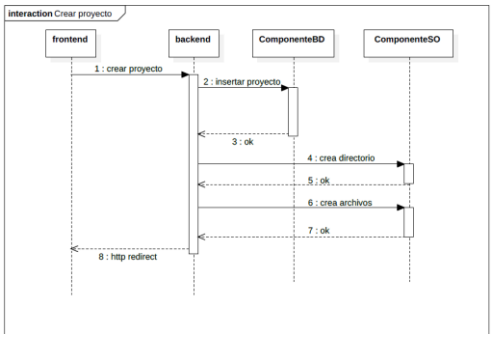

Figura 4. Diagrama de secuencias para explicar el funcionamiento de la interfaz de crear proyecto.
Fuente: Elaboración propia (2018)

Por otro lado, se creó la interfaz en donde se asignan variables, a continuación se muestra el diagrama de secuencias donde se aprecian las opciones que tiene el usuario al momento de interactuar con dicha interfaz, siendo estas, ver las variables asignadas en el sistema, asignar un elemento a un puerto, modificarle el nombre o borrar dicho elemento y guardar el proceso, a continuaciòn se muestra en la figura 5 donde se observan las secuencias que explican el funcionamiento de la interfaz al asignar variables.

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 18 Edición No 2 – Año 2019

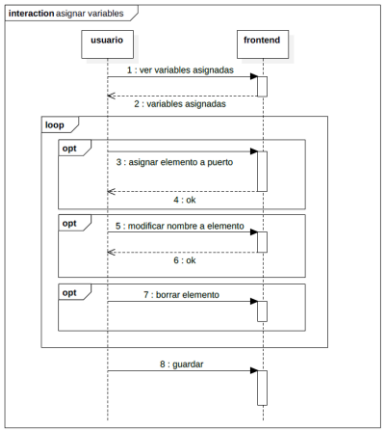

Figura 5. Diagrama de secuencias para explicar el funcionamiento de la interfaz de asignar variables.
Fuente: Elaboración propia (2018)

Seguidamente, se esquematiza el diagrama bajo los mismos estándares que demuestra el funcionamiento de la interfaz donde el usuario programará los componentes para que ejecuten la función establecida por el mismo tal y como se observa en la figura 8. Se observa que el usuario puede ver cómo es la sintáxis del pseudocódigo creado y escribir su códio basandose en los parámetros establecidos por el pseudocódigo. A su vez, se puede hacer una compilación al pseudocódigo que sería la opción de corregir, este pseudocódigo le dirá al usuario si el código creado está permitido o si tiene algún error. Puede además ejecutar el código, pararlo y guardar o eliminar el proyecto.

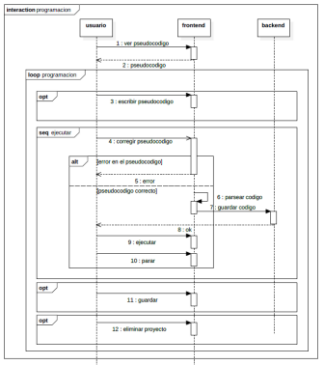

Figura 8. Diagrama de secuencias para explicar el funcionamiento de programación.
Fuente: Elaboración propia (2018)

Diseño de las de las interfaces de la aplicación

Una vez realizada la planificación para el desarrollo del proyecto, se procede con el diseño, el cual es un bosquejo del funcionamiento básico que debe contener la aplicación y tiene como fin propiciar un uso agradable que permita su utilización en todo tipo de usuario, mostrando

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 18 Edición No 2 – Año 2019

claramente los componentes y funciones de la misma, las interfaces de usuario están referidas a la muestra de cómo verá el usuario la aplicación.

Seguidamente se muestra la interfaz de inicio**.** Cabe destacar que, las interfaces de Inicio están referidas a la primera vista que se puede observar al entrar a una plataforma, página web o aplicación. Su función en líneas generales es darle al usuario una introducción al sistema donde están ingresando. En la figura 9 se muestra un wireframe que plantea el diseño de lo que sería luego la interfaz de inicio del sistema. Posteriormente, en la figura 10 se muestra el resultado final de la interfaz de inicio.

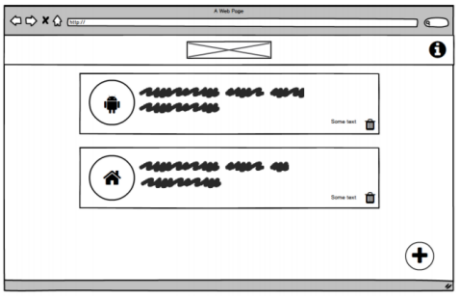

Figura 9. Wireframe de la interfaz de inicio
Fuente: Elaboración propia (2018)

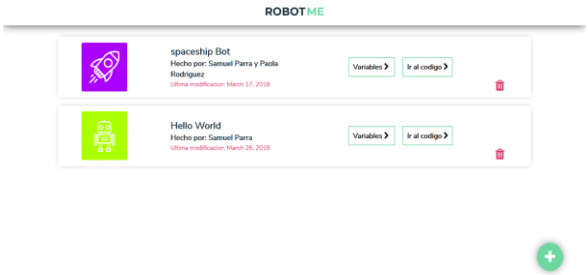

Figura 10. Interfaz de inicio
Fuente: Elaboración propia (2018)

A su vez, se muestra un Wireframe como primer prototipo que ayuda a plantear el diseño estructural de lo que sería luego la interfaz de crear proyecto, así como también se anexa a continuación el resultado final de la interfaz ya realizada y funcionando en el sistema mostrando lo que aparece para el usuario al momento de crear un nuevo proyecto. Se muestran a través de las figuras 11 y 12 que se presenta a continuación.

.

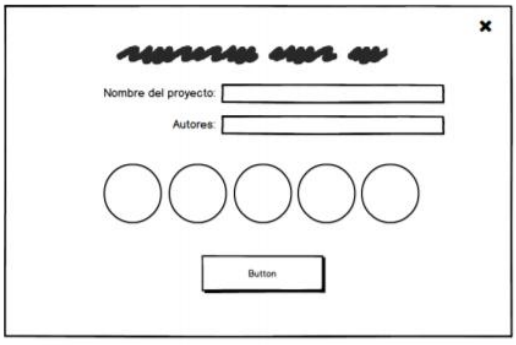

Figura 11. Wireframe de la interfaz de crear proyecto Fuente: Elaboración propia (2018)

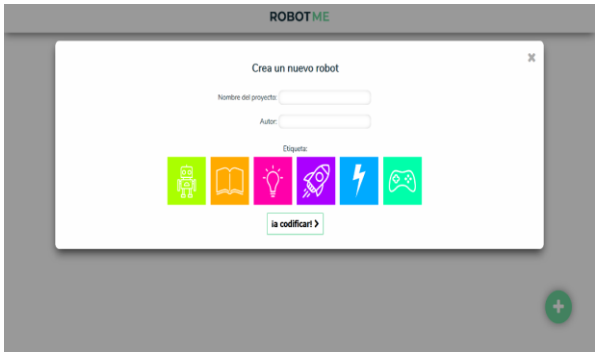

Figura 12. Interfaz de crear el proyecto
Fuente: Elaboración propia (2018)

Por otro lado, es importante documentar, la interfaz que el wireframe que plantea el diseño de lo que sería luego la interfaz de crear proyecto mostrado en la figura 13, el wireframe que plantea el diseño de lo que sería luego la interfaz de programación donde se asignan variables mostrado en la figura 14 y por último Wireframe de la interfaz de programación y su interfaz final mostrado en la figura 15 y 16.

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 18 Edición No 2 – Año 2019

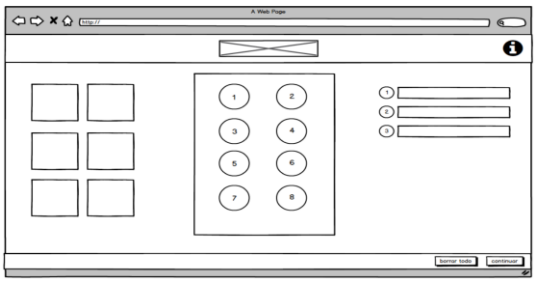

Figura 13. Wireframe de la interfaz de asignar Variables. Fuente: Elaboración propia (2018)

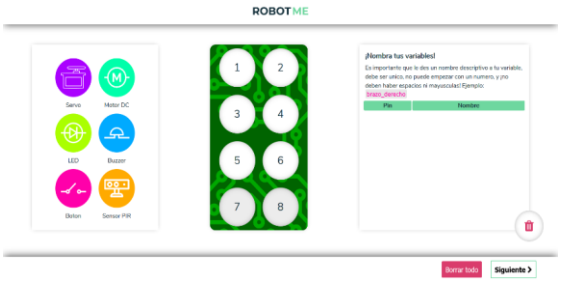

Figura 14. Interfaz de asignar variables Fuente: Elaboración propia (2018)

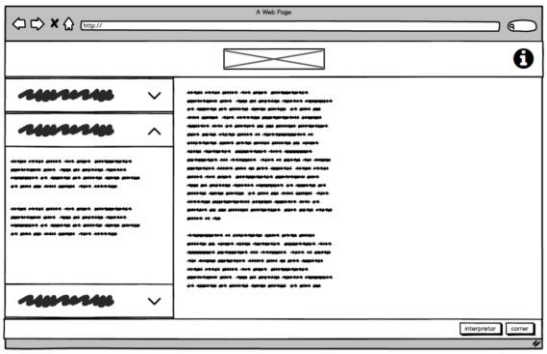

Figura 15. Wireframe de la interfaz de Programación. Fuente: Elaboración propia (2018)

Figura 16. Interfaz de Programación. Fuente: Elaboración propia (2018)

Modelo de datos de la Aplicación

Según Oracle (2018), un modelo de datos es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia. Siguiendo los lineamientos de esta definición, a continuación se muestra el modelo de datos aplicado en la estructura de la base de datos que fue diseñada y se encuentra dentro de la aplicación.

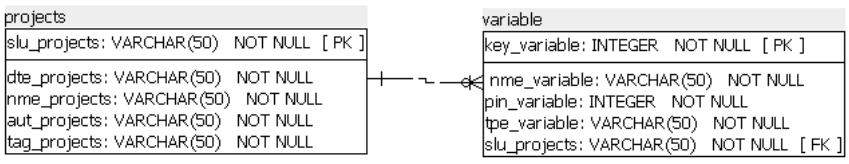

Figura 17. Modelo de datos de la aplicación
Fuente: Elaboración propia (2018)

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 18 Edición No 2 – Año 2019

Codificación de componentes de la Aplicación y Servidor

Primeramente, en la figura 18, se presenta el código que se encuentra en el servidor el cual tiene como finalidad establecer la creación del proyecto. Esto se puede observar a través de la asignación de una identificación, lo cual también es llamado “Slug”, seguidamente a esta función se crea un directorio en el cual podrán ser almacenados los archivos del proyecto, para luego así poder acceder a ellos a través de la Base de Datos previamente mencionada.

```python
@app.route('/new',  methods = ['POST'])
def new_project():
    try:
        #getting the data out of the form and insert the new document and get slug
        name = request.form['name']
        author = request.form['author']
        tag = request.form['tag']
        slug = database.new_project(name,author,tag)
        #create the new directory and files with the slug
        result = command.create_new_project_dir(slug, name, author)
        if (result):
            #redirect to variables
            return 'variables/'+slug
        else:
            return 'ERROR'
        #return 'variables/'+slug

    except (RuntimeError, TypeError, NameError):
        print("Something went wrong creating a new project | views/index.py")
```

Figura 18. Código en el servidor para la creación de un nuevo proyecto
Fuente: Elaboración propia (2018)

Posteriormente a las funciones previamente descritas, se tiene el código el cual se encuentra también dentro del servidor con la finalidad de iniciar el programa traducido del pseudocódigo escrito previamente por el usuario. Este se inicia a través de un evento Websockets para así enviar los datos de salida a la simulación de consola en la interfaz de programación tal y como se muestra a continuación en la figura 19.

```python
@socketio.on('run', namespace='/run')
def run_this(project_slug):
    proc = app.config['PROCESS']
    if proc == None:
        eventlet.spawn(run_code_thread, project_slug=project_slug['data'])
        emit('log', {'data': 'Programa Ejecutandoce'})
    else:
        emit('log', {'data': 'Ya existe un programa ejecutandoce, debes parar el programa anterior para ejecutar este'})

def run_code_thread(project_slug):
    APP_ROOT = os.path.dirname(os.path.abspath(__file__))
    APP_STATIC = os.path.join(APP_ROOT,'/home/pi/robotme/robotme/projects/'+project_slug+'/code.py')
    proc = Popen(['python',APP_STATIC], stdout=PIPE, bufsize=1)
    app.config['PROCESS'] = proc
    print('da')
    for line in iter(proc.stdout.readline,''):
        socketio.emit('log', {'data': line}, namespace='/run')
```

Figura 19. Código dentro del servidor que inicia el programa que el usuario
Fuente: Elaboración propia (2018)

Seguidamente, en la figura 20 se puede visualizar el código en el servidor para detener el programa, ya sea voluntariamente o accidental, traducido desde el cliente a través de un endpoint. Asi como también, se muestra en la figura 21 el código localizado dentro de la

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 18 Edición No 2 – Año 2019

interfaz de usuario para inicializar y detener el programa. Creando y cerrando así el Websocket previamente creado en la siguiente función.

```python
@socketio.on('disconnect')
def kill_socket():
    kill()

@app.route('/code/kill')
def kill_route():
    kill()

def kill():
    proc = app.config['PROCESS']
    print proc
    if proc != None:
        proc.send_signal(signal.SIGINT)
        #proc.terminate()
        app.config['PROCESS'] = None
        print app.config['PROCESS']
        return "Programa terminado"
    else:
        return "No hay programa corriendo"
```

Figura 20. Código dentro del servidor para la interrupción del programa. Fuente: Elaboración propia (2018)

```javascript
const run = () =>{
    $("#console").empty();
    let namespace = '/run';
    console.log(namespace)
    socket = io.connect(location.protocol + '//' + document.domain + ':' + location.port + namespace);
    socket.on('connect', function(e) {
        socket.emit('run', {data: slug});
        console.log(e)
    });
    socket.on('log', function(msg) {
        $('#console').append(output(msg.data));
    });
}

const stop = () =>{
    socket.disconnect()
    $.ajax({
        url : '/code/kill',
        method: 'GET',
        success: (e) =>{
            console.log(e)
            $("#console").append(ouput(e))
        },
        error: (e)=>{
            console.log(e)
        }
    })
}
```

Figura 21. Código en frontend para iniciar y detener el programa Fuente: Elaboración propia (2018)

Se tiene el código encargado de corregir errores de sintaxis del pseudocódigo mostrado en la figura 22, este se ejecuta antes de la traducción a lenguaje Python, de esta manera si hay existencia de errores el pseudocódigo no será traducido. Seguido a esto, en la figura 23 se muestra el código que se tiene el cual se encargará de traducir el pseudocódigo escrito por el cliente a lenguaje Python, para que de esta manera pueda ser satisfactoriamente ejecutado.

```
case "servo":
    {
        if (element.string == "girar") {
            checkIndent(element, i)
            posible_values = ["0", "90", "180"]
            if (servo_vars.indexOf(tokens[index + 2].string) < 0) error("Error: variable de tipo difente a servo", i)
            if (posible_values.indexOf(tokens[index + 4].string) < 0) error("Error: valor de grado no valido", i)
            if (tokens[index + 6].string !== "grados") error("Error: Instruccion incompleta", i)
        }
        return true;
        break;
    }
```

Figura 22. Código para la identificación de errores de sintaxis en el pseudocódigo Escrito por el usuario.
Fuente: Elaboración propia (2018)

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 18 Edición No 2 – Año 2019

Figura 23. Código en el servidor encargado de traducir el pseudocódigo escrito por el usuario a código en Python.
Fuente: Elaboración propia (2018)

Luego, el código previamente traducido a lenguaje Python en el servidor podrá ser enviado como archivo code.py. Este nuevo formato de archivo creado dentro del sistema luego de la traducción dará la posibilidad de ser interpretado por el RaspberryPi, lo cual permitirá ser ejecutado satisfactoriamente de manera que se pueda observar el resultado en el prototipo de los componentes electrónicos en físico.

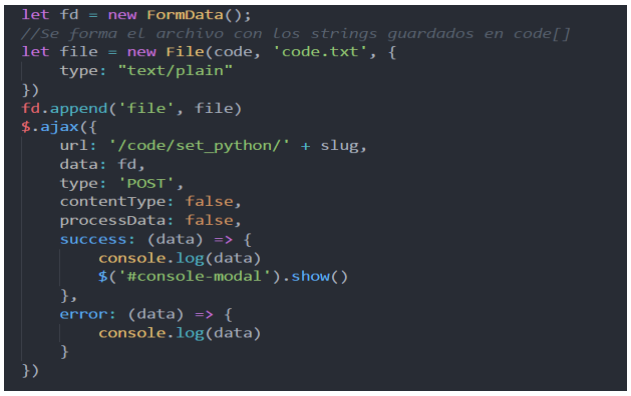

Figura 24. Código traducido a Python enviado como archivo code.py. Fuente: Elaboración propia (2018)

Implementación del Sistema

La implementación es una de las etapas genéricas en el proceso de desarrollo de un sistema informático, como se puede apreciar en la literatura al respecto (Deek y McHugh, 2005; Pardo, Pino, García, Baldassarre, y Piattini, 2013; Ramachandran y de Carvalho, 2011; Stepanek, 2005). En la mayoría de los textos consultados este proceso se asocia a la programación de los modelos que han sido obtenidos en etapas anteriores del ciclo de vida del sistema informático.

Luego de haber completado el proceso de codificación tanto de la aplicación como del RaspberryPi junto con sus módulos y el servidor, se procedió a realizar la implementación del mismo. Se inició con la aplicación local en una laptop con disponibilidad de wifi, luego de esto

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 18 Edición No 2 – Año 2019

se ensambló el dispositivo prototipo y por último se inicializó el servidor, se observó un correcto funcionamiento del sistema.

Pruebas del Sistema

Las pruebas del sistema son una parte del proceso de desarrollo de todo sistema y las mismas demostraran a través de distintas fases el alcance, efectividad y calidad del software que este en desarrollo, estas determinaran cualquier error o fallas en el sistema, encontrar defectos o bugs, y así poder corregirlas a tiempo. Para este sistema se aplicaron las pruebas unitarias, de aceptación y por último la de estrés las cuales se describen a continuación:

Pruebas Unitarias

El concepto de prueba unitaria o de unidad no es un concepto nuevo. Osherove (2009) afirma que este fue introducido en los años 70 por Kent Beck con el lenguaje de programación SmallTalk. También describe las pruebas unitarias como la mejor forma en que un desarrollador puede incrementar la calidad de su código mientras obtiene una mejor compresión de los requerimientos funcionales de una clase o un método.

Las pruebas unitarias se aplican a las distintas unidades del código por separado y así corroborar que cada una funciona. Esta prueba fue aplicada a todo el sistema empezando por la aplicación, donde se probaron las funciones utilizadas en la misma por separado, luego se le fueron aplicadas al servidor donde a través del compilador se verifico si la compilación fue exitosa y por último se le fue aplicada al dispositivo donde se probaron los distintos módulos que el incluye por separado y así corroborar su funcionamiento. Los resultados obtenidos fueron los esperados, un buen funcionamiento del sistema en general. A continuación, se mostrarán pruebas unitarias realizadas de respuestas al servidor y a la base de datos de la aplicación.

Pruebas de Aceptación

Bernal (2020) acota que “las pruebas de aceptación permiten al cliente saber cuándo el sistema funciona, y que los programadores conozcan que es lo que resta por hacer”. Las pruebas de aceptación determinaran la satisfacción del cliente, es decir, servirá para medir la relación entre el usuario y el sistema, verificando así la aceptación del mismo, la rapidez con la que se aprende a usarlo y su adaptabilidad por parte del usuario que lo utilice. Esta prueba se les fue aplicada a los usuarios que utilizaron el sistema y los mismos lo calificaron como un sistema fácil de utilizar, ya que el mismo posee interfaces sencillas y agradables para su uso. Los resultados obtenidos fueron los esperados.

Prueba de Estrés

El propósito de realizar esta prueba es encontrar el punto de quiebre del sistema. Se incrementa el número de usuarios y dispositivos conectados al servidor hasta que el mismo colapse. Esto se realiza con el fin de verificar que el sistema tiene la capacidad de funcionar correctamente al momento de implementarse y el mismo reciba múltiples peticiones de distintos usuarios. Para el sistema, se realizó la prueba conectando cinco (5) usuarios, ya que está pensado para una familia promedio, y dos dispositivos del sistema. Los cinco (5) usuarios realizaron peticiones al mismo tiempo y se obtuvo un tiempo de respuesta aceptable.

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 18 Edición No 2 – Año 2019

Aplicación de instrumento 1

De acuerdo con Sabino (2007), un instrumento de recolección de datos es definido como "cualquier recurso de que se vale el investigador para acercarse a los fenómenos y extraer de ellos información." Esto quiere decir, que para obtener información relevante para la investigación fue necesario el uso de un instrumento de recolección de datos. Pare ello, se elaboró una encuesta cerrada de 7 preguntas que se aplicó a niños en un rango de edad entre 9 y 12 años, así como a sus representantes, que usaron el sistema propuesto. Dicha encuesta se hizo con la finalidad de evaluar si realmente se cumplía con el objetivo de "impartir conocimientos introductorios sobre la robótica a los niños en etapa escolar".

PREGUNTA	SI	NO
¿Es fácil para usted el manejo del sistema?		
¿Considera usted interesante un sistema el cual le permita introducir la robótica a niños en etapa escolar?		
¿Se siente usted a gusto con la idea de que los niños y niñas estén en contacto con la robótica?		
¿Observa usted una actitud receptiva en los niños con respecto al sistema?		
¿Considera usted que el sistema le permite introducir la robótica a los niños en etapa escolar?		
¿Siente usted incomodidad o molestia al momento de usar el sistema?		
¿Tiende usted a indagar más sobre la robótica luego de usar el sistema?		

Resultado del Instrumento 1

Luego de aplicar debidamente la encuesta a usuarios que tuvieron la oportunidad de probar el sistema, habiendo tenido en consideración no solo la opinión de los niños que interactuaron directamente con el prototipo sino también la opinión emitida por los representantes que acompañaban a los niños al momento de probar el sistema propuesto, se realizó el cálculo de las respuestas obtenidas en porcentaje, como se muestra a continuación.

Tabla 1
Instrumento 1

	Preguntas						
	1	**2**	**3**	**4**	**5**	**6**	**7**
SI	**100%**	**100%**	**70%**	**90%**	**100%**	**0%**	**50%**

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 18 Edición No 2 – Año 2019

NO	**0%**	**0%**	**30%**	**10%**	**0%**	**100%**	**50%**

Fuente: Elaboración propia (2018)

Evaluando los resultados obtenidos de la encuesta se puede observar de la pregunta 1 “Es fácil para usted el manejo del sistema” se obtuvo como resultado un 100% de respuestas indicando de manera unánime que sí lo fue, por lo cual se puede concluir sustentándose en estos resultados que efectivamente se cumplió de manera satisfactoria con el objetivo de crear una interfaz amigable y fácil de comprender para los niños.

En la segunda pregunta, acera de si hay o no interés en la existencia de un sistema que se encargue de introducir conocimientos de robótica a niños en etapa escolar, se obtuvo nuevamente un resultado positivo, arrojando un 100% de respuestas afirmativas, Lo cual indica que efectivamente si hay interés en aplicar un sistema con esta temática. Ahora bien, en la tercera pregunta orientada a los padres o supervisores de los niños para corroborar si había receptividad con respecto a los conocimientos que se quieren impartir en los niños, se obtuvo nuevamente un resultado positivo, confirmándose de esta manera que sí se cuenta con la receptividad.

Con la cuarta pregunta se buscaba indagar más a fondo en cómo era el interés de los niños al momento de tener un acercamiento con la robótica, lo cual arrojó resultados positivos en un 90% de los casos encuestados y en la quinta preguntase hizo con la finalidad de corroborar si efectivamente había un acercamiento satisfactorio de los conocimientos de robótica a los niños, en lo cual se obtuvo una respuesta positiva con respecto a esto, demostrando que el sistema cumplía con su objetivo principal.

La sexta pregunta está orientada a los niños era con la finalidad de saber si había algún tipo de incomodidad al momento de usar el sistema, para tener en consideración la experiencia de los usuarios y poder mejorar cualquier aspecto que no estuviese planteado de manera acorde para el público objetivo. Los resultados demostraron que el diseño de experiencia de usuario estaba acorde a lo esperado.

En la última pregunta, orientada nuevamente a los niños que utilizaron el sistema, se les preguntó si luego de utilizar el sistema sintieron curiosidad por indagar más con respecto a la robótica, se obtuvo un resultado ambiguo indicando en un 50% que sí y en un 50% que no. Sin embargo, se consideró como algo positivo debido a que se logró despertar un interés de profundizar con respecto a esta temática en al menos un 50% de los usuarios involucrados. Con los resultados obtenidos en la encuesta, se demostró la respuesta satisfactoria de que fue logrado el objetivo principal planteado anteriormente.

Conclusiones

Luego de haber planteado los requerimientos mínimos del sistema para la introducción de la robótica a niños en etapa escolar, los cuales son necesarios para su correcto funcionamiento, se obtuvo que el sistema no requiere de valores o recursos de muy alto nivel, ya que se debe mantener lo suficientemente sencillo e intuitivo para un amigable manejo, así que se consideran en un nivel estándar para la sociedad por lo que es accesible y beneficioso.

Para la fase de diseño, luego de haber realizado el proceso de planificación, se diseñaron las interfaces a utilizar en la aplicación local, utilizando las ideas planteadas al inicio de la investigación, donde se definió la realización de interfaces sencillas y agradables para su uso por parte del usuario, el cual se sintió satisfecho con el diseño realizado. Se logró crear la

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 18 Edición No 2 – Año 2019

interacción entre el usuario y el sistema satisfactoriamente, debido a que todo usuario busca interfaces cómodas, amigables y fáciles de manejar y eso fue lo que se realizó.

Luego de haber diseñado las interfaces a utilizar en la aplicación local, se procedió a codificar ese diseño, por lo que se usó el estándar HTML5, comprendido por HTML, Javascript, y CSS, en conjunto con la librería JQuery para la creación de una aplicación agradable y cómoda para su uso. Para la codificación del servidor, el cual tiene la función de comunicar el RaspberryPi y la aplicación local con sus módulos, se utilizó el lenguaje Python.

Al haber finalizado el proceso de codificación se procedió a la implementación del sistema y por consiguiente a una fase de pruebas, como lo fueron las pruebas unitarias, las cuales determinarían su efectividad y calidad. Se confirmó lo esperado por parte de los desarrolladores, un sistema con un buen funcionamiento por lo que se determinó que el objetivo referente a las pruebas fue logrado.

Por último, luego de haber realizado las pruebas del sistema, se procedió a realizar la encuesta planteada y así verificar si el sistema permitió efectivamente introducir los conocimientos básicos de la robótica a los niños. La misma se le aplico a un grupo de 10 usuarios donde los mismos utilizaron el sistema. Los usuarios, a través de las preguntas cerradas demostraron positivamente la funcionalidad y el cumplimiento del objetivo del sistema.

Estos resultados obtenidos en las encuestas demuestran que el sistema, además de ser interesante para los usuarios, si permite inducir los conocimientos sobre la robótica según la respuesta de los mismos en la pregunta cinco, y de igual forma los usuarios sienten una completa mejora en cuanto a la educación tecnológica empleada para los niños al hacer uso de este sistema. Por lo que, con estos resultados los investigadores llegaron a la conclusión de que, si se imparten los conocimientos del área robótica a los niños y de igual forma se mejora, a través del sistema, la calidad de educación tecnológica impartida.

Referencias Bibliográficas

Deek, F. P., y McHugh, J. A. M. (2005). *Strategic software engineering: an interdisciplinary approach.* Boca Ratón, Florida: Auerbach Publications.

IBM (2014) *Conceptos de IBM.* Recuperado de: https://www.ibm.com/docs/es/i/7.2?topic=operations-i-concepts

Oracle (2018) *Acerca de Modelado de datos.* Recuperado de: https://docs.oracle.com/cloud/help/es/reportingcs_use/BILPD/GUID-D53FEF32-BEE5-4612-9041-430D09294E65.htm

Hurtado de Barrera, J. (2010). *Guía para la comprensión holística de la ciencia.* Caracas: Fundación Sypal.

Jorge Bernal (2020). *Fundamentos del proceso de Testing Funcional.* Recuperado de: https://www.blmovil.com/fundamentos-del-proceso-de-testing-funcional-iv-pruebas-de-aceptacion/

Osherove, R. (2009). *The Art of Unit Testing.* Manning Publications

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 18 Edición No 2 – Año 2019

Pardo, C., Pino, F. J., García, F., Baldassarre, M. T., y Piattini, M. (2013). From chaos to the systematic harmonization of multiple reference models: A harmonization framework applied in two case studies**.** *Journal of Systems and Software*, 86(1), 125-143.

Ramachandran, M., y de Carvalho, R. A. (2011). *Handbook of Research on Software Engineering and Productivity Technologies: Implications of Globalization* (K. Klinger Ed.). Nueva York: Engineering Science Reference.

Ruiz-Velasco, E. (2007). *Educatrónica: Innovación en el aprendizaje de las ciencias y la tecnología.* México: Ediciones Díaz Santos.

Schwaber, K. and Beedle, M. (2002). *Agile Software Development with Scrum.* New Jersey, United State: Prentice Hall PTR.

Stepanek, G. (2005). *Software Project Secrets: Why Software Projects Fail.* Estados Unidos: APRESS.

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 18 Edición No 2 – Año 2019