

Diseño e Implementación de un Sistema de Respuesta Personal empleando Tecnología IoT

Víctor Arce Domínguez*, Monica Flores[‡], Efrén Herrera*, Douglas Plaza* y Jorge Gonzalez[‡]

[‡]Universidad Espíritu Santo, UEES, Facultad de Sistemas, Telecomunicaciones y Electrónica,
Campus Samborondón, Km 2.5 vía La Puntilla, Samborondón, Ecuador

*Escuela Superior Politécnica del Litoral, ESPOL, Facultad de Ingeniería en Electricidad y Computación,
Campus Gustavo Galindo Km 30.5 Vía Perimetral, P.O. Box 09-01-5863, Guayaquil, Ecuador
Email: {*varce, eherrera, douplaza}@espol.edu.ec, {[‡]mfloresm, jgonzales}@uees.edu.ec

Resumen—En el presente documento se detalla el diseño e implementación de un Sistema de Respuesta Personal (SRP). SRP abarca dos subsistemas principales: el Dispositivo de Respuesta al Estudiante (DRE), que es un dispositivo electrónico (hardware), y la plataforma informática para el procesamiento y visualización de datos (software). También se detalla el desarrollo de las metodologías en torno a la tecnología abierta y los problemas técnicos que surgieron durante la implementación del SRP. El sistema integrado (SRP) fue probado mostrando un rendimiento apropiado con un futuro prometedor para ser utilizado como una herramienta diaria en las aulas como medio de evaluación. La tecnología empleada en este trabajo permite el escalado de sistemas en función del número de dispositivos o de la integración de un número considerable de dispositivos en una red WiFi de manera estable. Adicionalmente, la arquitectura abierta del SRP asegura un dispositivo de costo óptimo y asequible para la mayor parte de las Instituciones de Educación en países en vías de desarrollo.

Índice de Términos: Clicker, sistema de respuesta del estudiante, IoT(Internet of Things), hardware abierto.

1. Introducción

Hoy en día, la importancia de un impacto positivo de la educación sobre las sociedades es crucial, principalmente en las economías emergentes [1]. Las economías emergentes necesitan profesionales altamente especializados. Para desarrollar economías basadas en el conocimiento. En este sentido, todas las medidas adoptadas por organizaciones gubernamentales o privadas son valiosas.

La deserción académica puede considerarse como uno de los principales problemas en la educación [2], [3], [4], [5]. Por lo tanto, los métodos y procesos deben reinventarse para estar acorde a los nuevos programas educativos con estudiantes que son considerados como ‘nativos digitales’ debido a las habilidades naturales relacionadas con el uso de dispositivos tecnológicos como ultrabooks, smartphones y tabletas [6] y el creciente campo del Internet de las cosas.

Universidades como La Universidad Espíritu Santo (UEES) y la Escuela Superior Politécnica del Litoral (ESPOL), están tomando medidas para mejorar los métodos de enseñanza-aprendizaje mediante la adopción de tecnología apropiada[7]. En este sentido, el presente trabajo es un aporte a la masificación del uso de SRPs, con la finalidad de mejorar los procesos de enseñanza-aprendizaje de dichas universidades.

El movimiento Open Software y más tarde el Open Hardware permite el desarrollo de tecnologías locales y apropiadas basado en plataformas de prototipado rápido [8]. Arduino®. es uno de los éxitos más relevantes del hardware libre. El Software libre ha permitido la implementación de diferentes plataformas para apoyar a los educadores en la retroalimentación continua de los estudiantes. En los últimos años, se han desarrollado un conjunto de dispositivos que incorporan tecnología IoT los cuales permiten crear entornos completamente comunicados.

En el presente trabajo, diferentes plataformas fueron evaluados para sincronizar correctamente el Dispositivo de Respuesta del Estudiante (DRE) con la plataforma informática. El software utilizado en este documento está basado en software libre: Google Dashboard. Este software permite la selección de plantillas para visualizar datos como lo son gráficas de pastel o diagramas de barras, también permite realizar algunos análisis de los datos.

El documento presenta: primero, una breve explicación general del sistema de respuesta personal. Luego, aspectos relacionados al hardware empleado, elementos utilizados en la implementación del SRP, junto a al desarrollo técnico de los diferentes prototipos. Luego se presenta el software relacionado con el desarrollo de la base de datos y el análisis propio de los datos. Por último, los resultados de la evaluación de la plataforma de respuesta inmediata se presentan.

2. Arquitectura del sistema

El diseño del sistema está basado en software y hardware libre, los mismos que permiten implementar una estructura flexible a cambios, actualizaciones y mejoras de

cualquiera de las etapas que lo conforman, además de presentar un ahorro significativo correspondiente a tiempo de la implementación.

El usuario va a interactuar directamente con el hardware, mientras que la conexión al software se realizará por medio de una red WiFi, de esta manera contando con un módulo de fácil transporte y uso, mientras que desde un solo servidor se podrá acceder a los datos tomados y a su correspondiente interfaz de visualización.

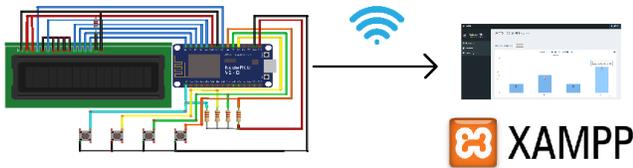


Figura 1. Arquitectura del Sistema

El manejo de la información se la realiza mediante bases de datos albergadas en el servidor principal, cada módulo guarda los datos de forma progresiva para minimizar la pérdida de datos por posible falla del módulo. La figura 1 presenta la interacción hardware y software del presente trabajo. Vale la pena resaltar que en el gráfico se aprecia una tarjeta de desarrollo NodeMcu, sin embargo, el sistema también fué implementado con tecnología Arduino como se lo describe en la siguiente sección.

3. Implementación del Hardware

3.1. Prototipos Iniciales

Para el diseño de los prototipos del sistema se seleccionó la plataforma de prototipos electrónicos de código abierto 'Arduino', específicamente el Arduino Mega 2560, el mismo que se eligió basándose en su flexibilidad, memoria, facilidad de programación y conexión con otros elementos, Arduino como tal permite de forma sencilla el envío y recepción de datos de forma bidireccional, ya sea con sensores o actuadores.

Para realizar la comunicación WiFi, se seleccionó el shield Yun, como tal un shield es una extensión de hardware compatible con la arquitectura de Arduino, la selección fue debido a su versatilidad en la conexión a red, además de albergar un sistema operativo basado en Linux: OpenWrt, el cual permite acceso remoto y crear base de datos locales en cada módulo. En la figura 2 se aprecian las tarjetas de desarrollo Arduino Mega e Iduino Yun.

Para la implementación de los prototipos se usó un Arduino Mega 2560, un Shield Iduino YUN, un protoboard de 830 puntos, 5 pulsadores, 5 resistencias de 1Kohm, 1 resistencia de 330ohm, una pantalla LCD 16x2, y cables tipo jumpers para Arduino.

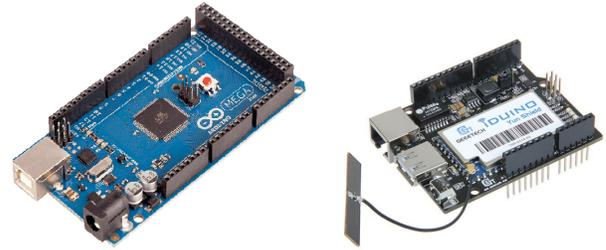


Figura 2. Arduino MEGA 2560 y Shield Iduino YUN

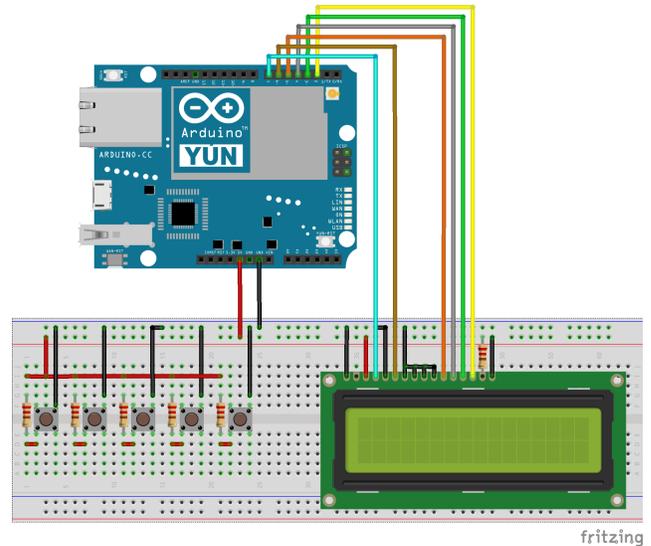


Figura 3. Primer prototipo del módulo. Imagen obtenida usando Fritzing

Después de comprobar el funcionamiento con el primer prototipo se procedió a la elaboración de una placa de circuito impreso 'PCB', para facilitar el transporte y uso del usuario, la misma que se diseñó basándose en los elementos adicionales conectados al Arduino, añadiendo espadines hembra para la conexión con la Arduino mediante cables tipo jumpers.

Con el segundo prototipo se consiguió una mejor estabilidad de la pantalla y reducción de los falsos positivos en los pulsadores, la interfaz permite visualizar las preguntas y las opciones de respuesta, así mismo la opción de contestar con los botones correspondientes.

Dadas las características físicas de las placas y shields de desarrollo Arduino, el tamaño conjunto del módulo se vuelve considerable, dando como resultado la búsqueda de un nuevo hardware que permita reducir considerablemente el tamaño de la implementación.

3.2. Prototipo Final de Hardware

Para la implementación final del módulo de respuesta personal se seleccionó la plataforma NodeMCU®, la misma que esta basada en el chip ESP8266, el cual es una unidad de



Figura 4. Elementos del segundo prototipo.

micro controlador con módulo de conexión WiFi integrada. La figura 5 muestra al dispositivo IoT NodeMcu.

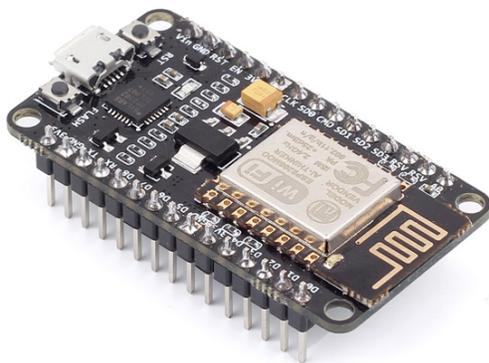


Figura 5. NodeMCU versión 3.

La principales diferencias entre las placas Arduino y la plataforma NodeMCU son tamaño y costo, se podría decir que son mucho mas versátiles y económicas, dado que combina lo mejor de las placas Arduino y la conectividad inalámbrica dando un valor agregado que permite abaratar costos en el desarrollo de dispositivos IoT.

Con la versión final se consiguió un módulo mas pequeño y liviano además con un consumo de corriente reducido, en comparación de los prototipos desarrollados con Arduino.

Para la implementación final del módulo de respuestas se uso un NodeMCU, una pantalla LCD 16x2, resistencias y pulsadores, montados sobre una placa de circuito impreso PCB diseñada para esta fin específico.

4. Implementacion del Software

El desarrollo del software se divide en dos etapas, la primera correspondiente a la página web y manejo de los

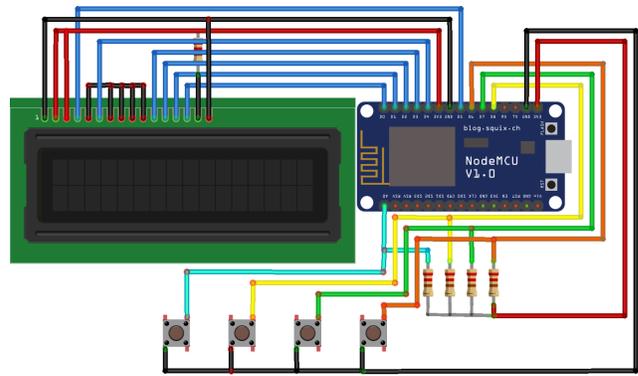


Figura 6. Versión Final de Hardware. Imagen obtenida usando Fritzing

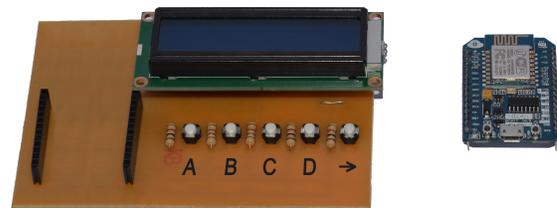


Figura 7. Elementos del prototipo final.

datos, administrados en un servidor local, montado en la aplicación XAMPP, la segunda a la programación y configuración de las diferentes plataformas de desarrollo utilizadas durante el proceso.

4.1. PRIMERA ETAPA

4.1.1. Página Web.. El diseño esta basado en programación html5 y PHP, los mismos que permite el desarrollo de sitios web con contenido dinámico. En el caso particular de nuestro sistema que genera una interfaz gráfica para visualizar los resultados obtenidos en los módulos correspondientes, se escogió como base las aplicaciones 'dashboard' de libre distribución, con el objetivo de obtener una interfaz limpia, fácil de usar y con actualización dinámica.

4.1.2. Base de Datos.. La gestión de datos, se realizó bajo el uso de MySQL, como administrador de los mismos. La estructura lógica de la base de datos se la diseño de forma expandible según las necesidades, la misma que se muestra en la Figure 5.

La base de datos está compuesta por un identificador principal para el módulo, el mismo que concentra los siguientes elementos:

- **IP:** Registra la dirección IP del módulo correspondiente.
- **Número:** Registra el número de pregunta que ha sido contestada.

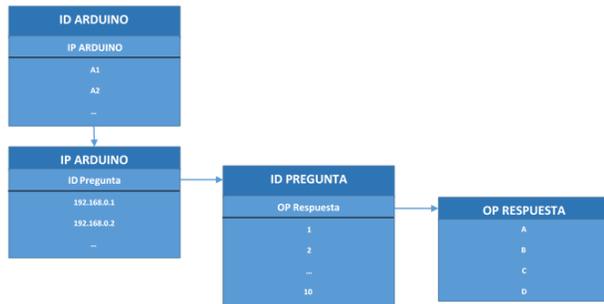


Figura 8. Estructura de la Base de Datos

- **Pregunta:** Registra el contenido de la pregunta propuesta.
- **Respuesta:** Registra la respuesta seleccionada por el usuario.
- **Tiempo:** Registra hora y fecha de cuando se realizó el ingreso a la base de los datos anteriores.

4.2. SEGUNDA ETAPA

4.2.1. Programación de Arduino. La configuración de la tarjeta Arduino mega se la realizó en Arduino IDE, la misma que esta basada en lenguaje C/C++. La programación incluye el gestor de pantalla LCD y TECLADO, para visualizar las preguntas y las opciones de respuestas, así como el teclado para responder.

```
yun-lcd $
#include <LiquidCrystal.h>
#include <Console.h> //Libreria para programar y
String name;
LiquidCrystal lcd(6, 7, 2, 3, 4, 5);
void setup() {
  // Inicia la conexión del puerto de transmisión
  Bridge.begin();
  Console.begin();
  // Espera hasta que la consola este conectada
  while (!Console);
```

Figura 9. Cabecera del Código Implementado en Arduino.

4.2.2. Configuración de Shield YUN. La configuración del Shield YUN, se lo realizó en dos etapas, primero la configuración del sistema Linux, para establecer permisos administrativos, con el objetivo de guardar las respuestas en la base de datos local del módulo, segundo la configuración de red WiFi, para conectarse al servidor principal administrado por XAMPP.

4.2.3. Configuración de NodeMCU. La configuración de la plataforma NodeMCU se realizó en dos etapas, la primera en

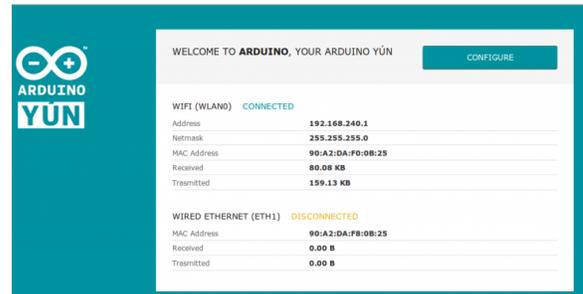


Figura 10. Consola de configuración de la red WiFi para el shield YUN.

lenguaje LUA para establecer configuraciones de IP estática y conexión a la red local. Mientras que la segunda se realizó mediante el Arduino IDE para realizar la programación de control de periféricos, tales como el teclado y la pantalla LCD.

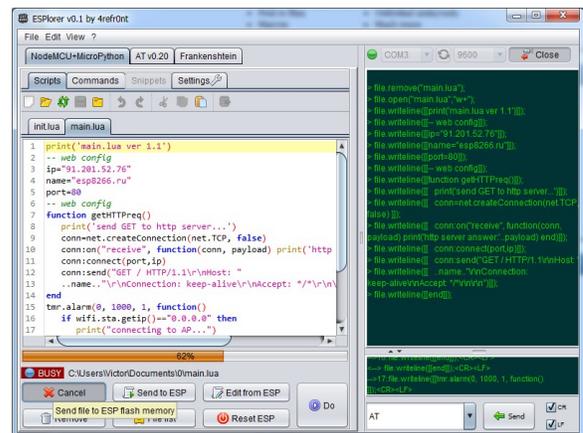


Figura 11. Programación de NodeMCU en LUA con ESPlorer IDE

Cabe mencionar que la principal ventaja de la programación realizada en lenguaje LUA con respecto al lenguaje C++ de Arduino, es la versatilidad de usar scripts para monitorear y accionar el NodeMCU en cualquier momento en tiempo real mediante consola y acceso remoto sin necesidad de estar conectado físicamente al dispositivo, aunque se considera una desventaja que estos cambios o el monitoreo no queda registrado en el dispositivo a menos que se use el ESPlorer IDE.

4.3. USUARIOS

El software en general tiene tres niveles de usuarios, que se describen a continuación:

Invitado:

Cualquier persona que haga uso del módulo para contestar las preguntas a través del teclado.

Encuestador:

Persona que selecciona y carga las preguntas que deberá

contestar el invitado, el encuestador también podrá visualizar los resultados en la página web.

Administrador:

Usuarios que tiene permitido realizar cambios a nivel lógico, de configuración y programación en cada una de las etapas del sistema.

5. Integración del Sistema

La integración del software y hardware, tiene como objetivo facilitar el uso de ambos, tomando como fuertes el manejo de datos a gran escala, una interfaz amigable, un módulo físico de fácil manejo y transporte, a continuación se presentan los dos prototipos usados y la interfaz web con los datos obtenidos.

Finalmente se observa que a través de la aplicación web se puede visualizar diagramas de barras con la cantidad de respuestas por opción, a su vez se puede descargar el mismo como un archivo exportable para posterior análisis.

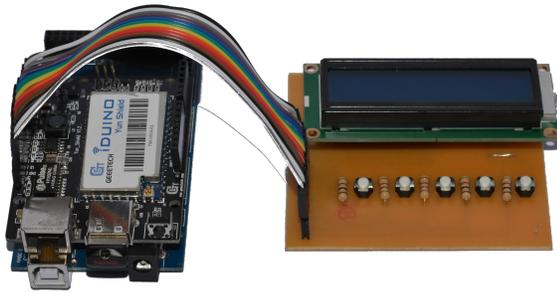


Figura 12. Módulo de Visualización de Preguntas y Respuestas Basado en Arduino



Figura 13. Módulo de Visualización de Preguntas y Respuestas Basado en NodeMCU

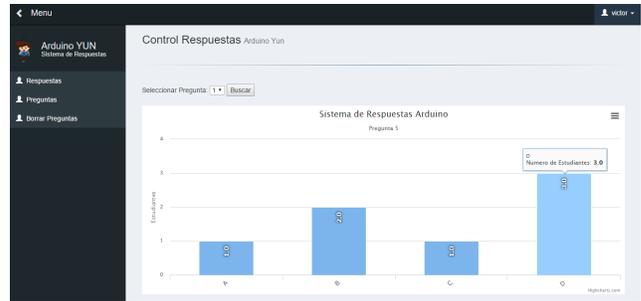


Figura 14. Interfaz Web

El sistema integrado es completamente escalable, lo que implica que soporta desde un solo módulo de respuestas, hasta el límite que permita la red WiFi. En un aula de clases de 25 a 40 personas con un router dedicado, el sistema tendría igual eficiencia como si fuera un solo módulo.

Cabe recalcar que la fuente alimentación del módulo basado en Arduino, debe suministrar la corriente necesaria(800mA) para trabajar en conjunto con el 'shield yun' y mantener una conexión WiFi constante, caso contrario mientras que la Arduino siga encendida y la conexión WiFi se pierda los datos se mantendrán en la memoria externa del sistema, a la misma que podrá acceder un usuario de tipo administrador.

Agradecimientos

El presente trabajo ha sido financiado parcialmente por la institución UEES a través del proyecto con código 2015SIS010 y por la institución ESPOL.

Conclusiones y Recomendaciones

Se ha implementado un sistema de respuesta personal utilizando tecnología de uso libre, la misma que representa una inversión económica óptima, comparado con sistemas comerciales. Aunque no hay una meta comercial involucrada en el desarrollo, ambas versiones fueron probadas en aulas de clases piloto, para recolectar información de los estudiantes. Se espera en próximas pruebas poder abarcar con dispositivos un salón de clase completo de alrededor de 30 estudiantes.

El dispositivo está diseñado como un sistema escalable a través del uso de software y hardware libre, de igual forma el uso de la base de datos en MySQL para almacenar ordenadamente un gran número de parámetros según la necesidad de la institución, los mismos que se pueden acceder desde cualquier nivel de la sección lógica.

El número de dispositivos que pueden escribir en la base de datos está limitado por la capacidad de conexión de la red WiFi y la velocidad de transmisión del protocolo de comunicación, aunque este último puede ser resuelto por la

implementación de un algoritmo de cifrado, apilamiento y transmisión dedicado para este propósito.

El límite en la conexión depende del router pero cabe recalcar que la mayoría de dispositivos nos muestran un límite de entre 30 y 40 dispositivos conectados simultáneamente a la mayor tasa de bits soportada por el mismo, mientras que las cadenas de datos enviadas por los dispositivos de respuesta personal implementados solo usan una pequeña porción de estos, con lo cual podemos garantizar que la cantidad estimada de dispositivos conectados simultáneamente se la puede considerar mayor.

En el presente trabajo se ha conseguido implementar un SRP basado en tecnología IoT. Se han evaluado 2 plataformas siendo la plataforma basada en el sistema de desarrollo NodeMCU la que presenta menos consumo de energía y por el tamaño permite optimizar el volumen ocupado por el dispositivo.

El SRP será empleado en un futuro inmediato en ambientes de aprendizaje donde se evaluará el impacto de su utilización.

Referencias

- [1] I. A. Zualkerman, "Design and implementation of a low-cost classroom response system for a future classroom in the developing world," *Interaction Design and Architecture(s) Journal*, pp. 68–84, 2012.
- [2] A. S. Putra, N. J. Jie, and T. K. Kiong, "Enhancing student involvement in a class using real-time response system," in *Information Technology Based Higher Education and Training (ITHET), 2012 International Conference on*, June 2012, pp. 1–4.
- [3] D. Rajavel and R. Kushalkar, "Clicker: Device independent student response system, developed for classroom and remote learning to provide instant response and feedback," in *Technology for Education (T4E), 2014 IEEE Sixth International Conference on*, Dec 2014, pp. 62–65.
- [4] J. A. Rossiter, "Using clickers in lectures to help identify and teach the control topics students find difficult," in *Control (CONTROL), 2014 UKACC International Conference on*, July 2014, pp. 543–548.
- [5] K. M. Vernaza, "Using personal response system technology and concept check modules to improve students learning experience: A case study," in *2007 37th Annual Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports*, Oct 2007, pp. F1D–16–F1D–17.
- [6] D. Amemado, "Integrating technologies in higher education: the issue of recommended educational features still making headline news," *Open Learning: The Journal of Open, Distance and e-Learning*, vol. 29, no. 1, pp. 15–30, 2014. [Online]. Available: <http://dx.doi.org/10.1080/02680513.2014.908700>
- [7] V. B. H., "El aprendizaje basado en problema (abp) como estrategia didáctica para la enseñanza del solfeo en el nivel superior," *El Artista*, vol. 12, pp. 36–53, 2015.
- [8] A. Powell, "Democratizing production through open source knowledge: From open software to open hardware," 2012.