

BUILDING THE FUTURE WITH SC: SMART CLASSROOM 2.0

Mansilla Luis¹, Nonino Fabian¹, Spooner Federico¹
Ing. Pedro López²

Desarrollador e investigador del proyecto¹
{lmansilla022, fabian.nonino, spoonerfederico}@gmail.com
Docente Coordinador²
pedro.lopez@uai.edu.ar
Universidad Abierta Interamericana - Rosario, Santa Fe, Argentina

RESUMEN

Smart Classroom 2.0 intenta brindar espacios educativos tecnológicamente enriquecidos y adaptados que apoyen de manera proactiva tanto a docentes como alumnos en el desarrollo de las clases. Nuestro prototipo puede proveer soluciones adaptativas que se ajusten y generen beneficios para los usuarios de dichos entornos.

En este artículo se propone tanto un modelo de Aula Inteligente, como las herramientas necesarias para automatizar muchas de las tareas cotidianas que se ejecutan en este ámbito. Agregamos también un módulo de Aula Virtual para mejorar aún más la experiencia de los alumnos.

Palabras Clave: Aula Inteligente, Aula Virtual, ahorro energético, asistencia controlada.

CONTEXTO

Luego de investigar durante todo 2017 y principios de 2018 sobre el proyecto Arduinos, decidimos realizar un trabajo que aplique los conocimientos que fuimos incorporando.

Enfocamos nuestros esfuerzos a lograr un prototipo de aula virtual con funciones avanzadas para luego realizar pruebas de

campo dentro de nuestra facultad y en base a evaluaciones posteriores tener la posibilidad de implementarlo en otras dependencias/facultades.

A la hora de pensar sobre cuál iba a ser nuestra investigación o proyecto, se nos ocurrió pensar que actividades dentro del ámbito de nuestras clases podrían automatizarse.

INTRODUCCION

Tradicionalmente, a la hora de comenzar una clase en particular, un alumno o profesor, obtiene información de donde se va a impartir una clase; en bedelía, algún transparente o consultando a otros compañeros. Siendo estos medios, muchas veces poco confiables, y ante un cambio de cronograma, la notificación en tiempo y forma del mismo es casi siempre nula, obligando a los involucrados a perder mucho tiempo hasta dar con la clase o aula esperada.

Por otro lado, muchas veces llegamos y ya se encuentran encendidos luces, proyectores, ventiladores y/o aires acondicionados que quedaron de otra clase, o nosotros descuidamos los mismos sin apagarlos al terminar. Esto se traduce en desgaste de los equipos y consumo de luz innecesario.

El sistema de aula inteligente que proponemos, llega para brindar asistencia autónoma a estos

casos cotidianos.

Programando el aula con el cronograma de horarios en los que va a estar activa durante la jornada, profesor afectado y listado de alumnos inscriptos. La misma posee toda la información necesaria para funcionar. El sistema de control chequea fecha y hora en cada momento, y si se encuentra en horario de inactividad, procederá a apagar todo sistema eléctrico que tenga bajo su control y restablecerlos al comienzo del cursado o tiempo antes según se parametricé.

Tanto alumnos como profesores podrán chequear en la correspondiente aplicación móvil o vía web. Que aula fue asignada para el cursado en cuestión.

Al llegar a la misma, antes de entrar, se anunciarán mediante un control biométrico a la entrada del aula. Dejando así constancia de ingreso y por ende asistencia a la clase. Los alumnos atrasados o el mismo profesor puede en todo momento chequear quienes ya se encuentran en el aula.

Llegando al tiempo de finalización, el sistema alertará a la aplicación móvil del profesor, recordándole que su tiempo de clase está terminando, para que pueda así formular una conclusión o notificar que su clase se va a prolongar. De esta forma. No se requiere presencia física de los bedeles en cada aula, muchas veces interrumpiendo la clase para tomar asistencia. En caso de que el sistema biométrico falle, El profesor desde su aplicación móvil podrá tomar asistencia o ajustar los presentes.

ESTADO DEL ARTE

A continuación se hace referencia a diferentes proyectos que fueron tomados como punto de partida a la hora de llevar adelante la realización del presente trabajo.

El trabajo número uno se basa en un modelo de ambiente inteligente adaptativo apuntado principalmente a la educación [1].

En el número dos, se plantea el desarrollo de un i-campus o campo inteligente el cual intenta redefinir la forma de educar utilizando la informática como bastión, se muestra un caso de aplicación puntual en la Universidad

de Essex[2].¹

Desde la investigación número tres, se muestran los impactos que tienen las TIC no sólo en el ámbito educativo, sino en la sociedad en general. Se habla de las ventajas de la educación a distancia y además se plantean “Dos vertientes en la relación entre las TICs y la educación”[3].

El trabajo número cuatro, presenta una metodología de aprendizaje llamada ABP “Aprendizaje Basado en Problemas” que intenta promover el desarrollo de la enseñanza de la tecnología, aplicando para ello hardware opensource, poniendo foco especialmente en Arduinos[4].

Tomando como referencia los trabajos anteriores, se presenta una tabla comparativa entre los diferentes trabajos y las características más importantes del presente trabajo de investigación.

Tabla 1. Comparativa de trabajos anteriores con el proyecto actual.

Trabajo	C 1	C 2	C3	C 4	C 5	C6
Ambientes Inteligentes en Contextos Educativos: Modelo y Arquitectura	X	X				X
The Intelligent Classroom: Beyond Four Walls	X	X		X		X
TIC: Su uso como Herramienta para el Fortalecimiento y el Desarrollo de la Educación Virtual		X		X		X
ABP (Aprendizaje Basado en Problemas) para la enseñanza de proyectos tecnológicos interdisciplinarios basados en Arduino		X	X			X
Trabajo Actual	X	X	X	X	X	X

C1: Smart Classroom C2: TIC en la educación C3: Arduinos C4: Aula Virtual C5: Eficiencia Energética C6: Recursos didácticos

¹ <https://www.essex.ac.uk/>

ANALISIS

Protocolo de comunicación entre dispositivos y el servidor: Cada nodo de control del aula, se comunicaran con el servidor mediante el uso del protocolo TCP/IP. Serán dotados de un módulo ESP8266 para enlace inalámbrico bajo el estándar IEEE 802.11 b/g/n, o un módulo ENC28J60 para enlace alámbrico bajo el estándar IEEE 802.3 según sea el caso.

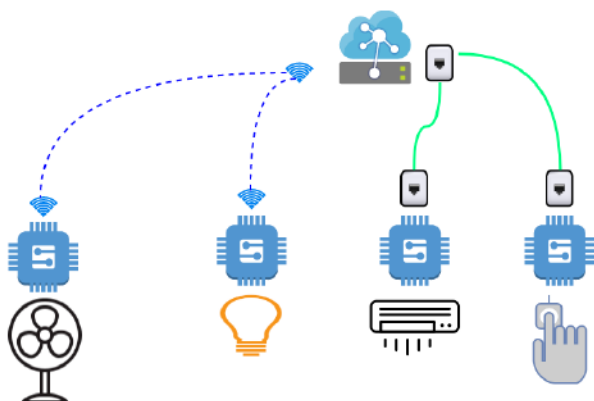
Dichos mensajes se enviarán via **HTTP** en formato **JSON**. Donde cada endpoint interpretará y dará una respuesta en este mismo formato.

ESP8266 ESP-01: Es un módulo autónomo capaz de establecer un enlace WIF autenticándose en un router remoto y establecer comunicaciones simples bajo tcp/ip. Entre otras cosas, descryptar mensajes JSON, tomar decisiones y brindarle estados a su GPIO en consecuencia.

ENC28J60: Es un módulo capaz de establecer un enlace Ethernet y establecer comunicaciones simples bajo tcp/ip, y comunicar el resultado vía SPI hacia Arduino, quien se encargará de decidir e interactuar con el dispositivo a controlar.

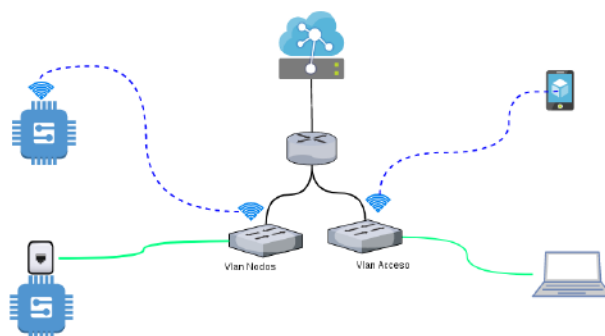
Arduino Pro Mini: Microcontrolador elegido por su relación costo/capacidad. Siendo su corazón un ATMEGA 328, posee 14 entrada/salidas digitales, 6 entradas analógicas, Protocolos SPI I2C y UART.

Gracias a estas características, es más que suficiente para interactuar con los diferentes dispositivos a controlar, quedando holgado para futuras implementaciones.



Red: La red bajo la que se comuniquen todos los nodos de control de cada aula estará aislada de la red de acceso al servidor vía celulares o navegadores.

Para ello, el diseño lógico de la red constará de 2 vlans diferentes, una para los nodos de control y otra bajo la cual el servidor reciba todas las peticiones de aplicación (web o celular). Para evitar así que problemas típicos de redes abiertas al exterior, como inyección de dhcp, tormentas de broadcast entre otros, comprometan el funcionamiento de los nodos. De esta forma, tanto celulares como navegadores, interactuarán exclusivamente con el servidor. Y este será el responsable de notificar a los nodos de control.



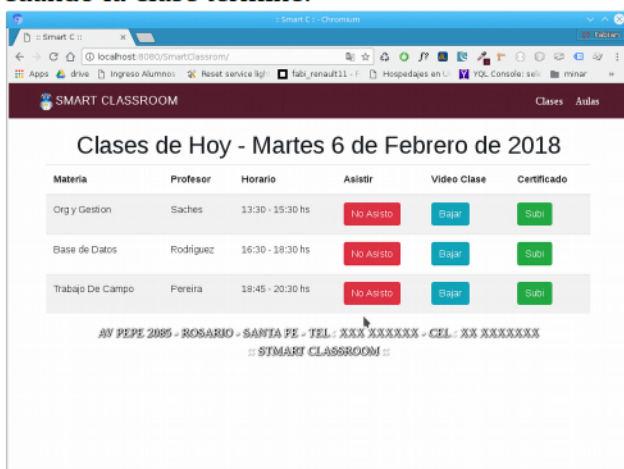
Funcionamiento: El bedel accederá a la plataforma y asignará un aula en particular, el cronograma de toda la semana... especificando por día, materia a dictar, horario de inicio/fin y profesor que la dicta. Con dicha información, el servidor tendrá conciencia de los horarios de inactividad y enviará, cuando llegue el momento, a los nodos de control de dicha aula, la acción correspondiente, tanto de encendido como de apagado entre otras.

En cualquier momento, un alumno o profesor podrá consultar, en qué aula tiene asignada la materia y en qué horario. A medida van ingresando al aula, tanto alumnos como profesores se identificarán biometricamente al ingresar, esto enviará al servidor la notificación que un alumno o el profesor, se encuentran en el aula a la espera de la clase. Esta información contribuye a la asistencia, donde el bedel no tendrá más que pedirle al sistema el listado de los presentes. Tanto de alumnos como el profesor. En todo momento

un alumno podrá ver si sus compañeros o el mismo profesor ya se encuentran en el aula o no.

Al concluir la clase, el profesor confirma que la misma ha terminado vía la app y si fuese necesario, el servidor nuevamente se comunica con los nodos de control, apagando todo sistema que consume energía innecesariamente.

Aula Virtual: En caso que un alumno no pueda asistir a una clase, lo informa mediante la app o via web, la misma le devolverá un acceso que se habilitará a bajar el video cuando la clase termine.



En el salón existirá una cámara IP la cual se activará y desactivará siguiendo el itinerario tal cual los demás módulos inteligentes, la cámara grabará toda la clase para guardarla en un storage, dejándola disponible para el alumno que faltó a clases, pueda bajar el video y verlo para así no perderla

Junto con esta funcionalidad, el alumno también puede subir un certificado médico o algún otro documento que justifique su ausencia.

DESARROLLO

Escenario 1:

Al momento de seleccionar el hardware para comunicar el Arduinos con el servidor principal, decidimos utilizar un módulo bluetooth **HC05**. El cual vía comandos AT, permite configurarse de modo cliente/servidor.

Comunicación Arduinos-Servidor por Bluetooth

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial BT1(10, 11); // RX | TX
void setup()
{ pinMode(8, OUTPUT); // Al poner en HIGH
  forzaremos el modo AT
  pinMode(9, OUTPUT); // cuando se alimenta
  de aquí
  digitalWrite(9, HIGH);
  delay (500); // Espera antes de encender
  el modulo
  Serial.begin(9600);
  Serial.println("Levantando el modulo HC-05");
  digitalWrite (8, HIGH); //Enciende el modulo
  Serial.println("Esperando comandos AT:");
  BT1.begin(57600);
}
void loop()
{ if (BT1.available())
  Serial.write(BT1.read());
  if (Serial.available())
  BT1.write(Serial.read());
}
```

Luego de varias pruebas descubrimos que la comunicación cliente-servidor era solamente punto a punto por lo que no soportaba su instalación en un **ambiente multipunto**.

Esta realidad nos hizo reflexionar sobre la practicidad del Bluetooth para realizar este tipo de conexiones y nos dimos cuenta que utilizando un módulo WiFi o Ethernet no sólo tendríamos la posibilidad de establecer sesiones multipunto, sino que además tanto los dispositivos de red Ethernet/Wifi son de uso cotidiano/facilita su adaptabilidad.

Escenario 2:

En la búsqueda de un medio de conectividad que cumpla con los nuevos requisitos, encontramos dos módulos que nos permitieron lograr tanto la conectividad alámbrica como inalámbrica de manera transparente sin modificar el hardware de base.

Tanto el módulo ENC28J60 (ethernet) como el módulo ESP8266 ESP-01 se comunican con el arduino bajo el mismo estándar y protocolo SPI.

Lo cual nos permite dependiendo el medio, optar por una tecnología u otra, al mismo costo y mínimo impacto

Ambos módulos son autónomos en cuanto a tecnologías de red se refiere, no hay necesidad de estudiar o controlar las capas del modelo OSI a bajo nivel, su chipset se encarga de ello y nos brinda via SPI al arduino el resultado de

los eventos .

En lo que a WIFI se refiere, el chip gestiona y controla seguridad, encriptación, frecuencia y control de ruido. Despreocupámonos de todo control del medio .

Setup inicial de módulos y variables Ethernet

```
//libreria para el cómodo uso de enc28j60
#include <EtherCard.h>
//libreria para el cómodo uso de json
#include <ArduinoJson.h>
//definimos la mac del nodo
static byte mac[] = { 0xDD, 0xDD, 0xDD, 0x00, 0x01, 0x05 };
//definimos su ip
static byte ip[] = { 192, 168, 30, 3 };
//estado global del nodo
int estado =0;
const int ENCENDIDO=1;
const int APAGADO=2;
const int ESTATUS=3;
byte Ethernet::buffer[700];
void setup() {
  Serial.begin(9600);
  if (!ether.begin(sizeof Ethernet::buffer, mac, 10))
    Serial.println("Error al acceder a enc28j60");
  else
    Serial.println("Inicializado");
  if (!ether.staticSetup(ip))
    Serial.println("No se pudo establecer la dirección IP");
  Serial.println();
}
```

Setup inicial de módulos y variables WIFI

```
//libreria para el cómodo uso de ESP8266
#include <ESP8266WiFi.h>
//libreria para el cómodo uso de json
#include <ArduinoJson.h>
//Declaramos el puerto 80 para escucha de peticiones
WiFiServer server(80)
int estado =0;
const int ENCENDIDO=1;
const int APAGADO=2;
const int ESTATUS=3;
void setup()
{
  Serial.begin(115200);
  WiFi.begin("ssid", "password");
  Serial.print("Conectando");
  while (WiFi.status() != WL_CONNECTED)
  {
    //espera hasta que el modulo enlace con el router
    delay(500);
    Serial.print(".");
  }
  Serial.println();
  //declara ip estática del nodo y su mascara
  IPAddress ip(192,168,30,4);
```

```
IPAddress subnet(255,255,255,0);
WiFi.config(ip, gateway, subnet);
Serial.print("Conectado! ");
//módulo conectado, inicia servicio
server.begin();
}
```

En esta nueva arquitectura, el servidor tendrá conciencia de que nodos se van suscribiendo al sistema, y comenzará a interactuar con ellos... estando ellos solo escuchando órdenes y brindando resultados de las mismas. Enviará mensajes de Keep Alive para saber si los mismos se encuentran activos y refrescar sus estados.

Para nuestro primer escenario, se desarrolló un código simple en java que envía mensajes http vía puerto 80 a las ips que conoce como nodos.

Del lado del arduino, recibe dichos mensajes, actúa en consecuencia y brinda su respuesta ACK / NACK

Como se puede notar, en ningún momento lidiamos con el medio físico, ya sea ethernet o wifi en el control. Lo que nos dio la transparencia que buscamos al delegar esto al hardware dedicado.

Función “Responder”

```
static word responder()
{
  BufferFiller bfill = ether.tcpOffset();
  bfill.emit_p(PSTR("HTTP/1.0 200 OK\r\n"
    "Content-Type: application/json;charset=utf-8"
    "\r\n"
    "{\"nodo\":$S, \"estado\":$D}"
    "\r\n"), ip, estado);
  return bfill.position();
}
```

Programa principal Módulo Ethernet

```
void loop()
{
  //lee el último paquete recibido, lo guarda en el
  buffer y devuelve el tamaño
  word len = ether.packetReceive();
  //extrae del buffer (si es tcp termina la
  conversación)
  word pos = ether.packetLoop(len);

  if (pos)
  {
    if (strstr((char *)Ethernet::buffer + pos, "GET /?"
on=1") != 0) {
      //ejecutar subrutina para encender,
```

```

dependiendo lo que se esté queriendo automatizar
    estado= ENCENDIDO;

}

if (strstr((char *)Ethernet::buffer + pos, "GET /?
on=0") != 0) {
    //ejecutar subrutina para apagar el dispositivo
    estado=APAGADO;
}
if (strstr((char *)Ethernet::buffer + pos, "GET /?
estado=1") != 0) {
    //petición de estados
}
ether.httpServerReply(responder());
}
}

```

Programa principal Módulo WIFI

```

void loop()
{
//le pregunta al server si no hay algún cliente
disponible
// si no es así termina el bucle y vuelve a comenzar
WiFiClient client = server.available();
if (!client){
    return;
}
while(!client.available())
delay(1);
String req = client.readStringUntil('\r');
client.flush();
//lee la request del cliente (su url completa) y
comprueba si es la url esperada
//en caso que lo sea, setea la variable val en 1 o en
0 según corresponda
int val;
if ( req.indexOf("?on=1") != -1)
//ejecutar subrutina para encender, dependiendo lo
que se esté queriendo automatizar
    estado= ENCENDIDO;

else if (req.indexOf("?on=0") != -1)
{
    //ejecutar subrutina para apagar el dispositivo
    estado=APAGADO;
}
else if (req.indexOf("?estado=1") != -1)
{
    //petición de estados
}
client.print(responder());
}
}

```

Beneficios: Gracias a esta continua interacción entre alumnos, profesores y bedeles sobre el sistema de aulas inteligentes. La información es actualizada y propagada de manera inmediata. Donde casos como por ejemplo, la necesidad de un cambio de aula de manera inesperada se vuelve mucho más simple, ya

que el sistema tiene siempre la última información sobre qué aula está desocupada y hasta cuando. A la hora de aplicar el cambio de aula. Tanto Alumnos como profesores pueden chequear la reasignación y remitirse a la nueva aula casi sin pérdida de tiempo. Reduciendo así la pérdida de tiempo de clase ante eventualidades de este estilo.

Por otro lado, aunque parezca ínfimo, la reducción del consumo de un aula por muy cortos periodos de tiempo, este multiplicado por todas las aulas del establecimiento, por todo el año, demuestra una reducción notable del consumo.

Otra gran ventaja es que no se requiere perder tiempo de clases en la toma de asistencia, ni del traslado de responsables al aula, muchas veces interrumpiendo la clase en momentos inapropiados.

Ya no se requiere de un gran personal de bedelía para cubrir todas las aulas, que en edificios grandes, son entre 3 a 6 bedeles para tomar a tiempo todas las asistencias.

LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

El proyecto pertenece a una de las 5 líneas de investigación de **CAETI Rosario**² denominada “Sociedad del Conocimiento y Tecnologías aplicadas a la Educación”. Las producciones de esta línea tienen la intención de “...potenciar las sinergias originadas en la vinculación del contenido, el conocimiento y el aprendizaje para lograr que los contenidos y conocimientos sean completos, accesibles, interactivos y utilizables en el tiempo. Se generarán avances en términos de usabilidad, accesibilidad, escalabilidad y costo de los métodos y tecnologías que manejan la creación, distribución y aprehensión del conocimiento...”[12]

CONCLUSION

Hemos desarrollado esta propuesta en función a problemáticas que en algún momento tanto nosotros como compañeros y docentes hemos

² <http://caeti.uai.edu.ar/lineas.aspx>

padecido en algún momento. Pensando en el medio ambiente y el ahorro energético, que es un tema que preocupa cada vez más y no esta de mas que la tecnología ayude a las políticas de concientización, que no siempre tienen en la gente el efecto esperado.

Bajo el Concepto de OpenSource, invitamos a cualquier establecimiento ya sea tecnológico o no, que desee implementarlo, tenga las herramientas para enriquecer el mismo o como ejemplo para el estudio y formación de profesionales en el ámbito de la electrónica y sistemas.

FORMACION DE RECURSOS HUMANOS

El presente trabajo, fue realizado por alumnos del Centro de Altos Estudios en Tecnologías Informáticas de la Universidad Abierta Interamericana regional rosario (CAETI Rosario) del último año de la carrera de 5 Ingeniería en Sistemas Informáticos: Luis Mansilla, Fabian Nonino y Federico Spooner. El mismo cuenta con la dirección del Ing. Pedro López, docente de Universidad Abierta interamericana.

BIBLIOGRAFIA

- [1] Hernández-Calderón, José-Guillermo, Edgard Benítez-Guerrero, and Carmen Mezura-Godoy. "Ambientes inteligentes en contextos educativos: modelo y arquitectura." *Research in Computing Science* 77 (2014): 55-65.
- [2] Dooley, James, et al. "The Intelligent Classroom: Beyond Four Walls." (2011): 457-468.
- [3] Rosario, Jimmy. "TIC: Su uso como Herramienta para el Fortalecimiento y el Desarrollo de la Educación Virtual." *DIM: Didáctica, Innovación y Multimedia* 8 (2006).
- [4] RIVERA, YAIR ENRIQUE. "ABP (APRENDIZAJE BASADO EN PROBLEMAS) PARA LA ENSEÑANZA DE

PROYECTOS TECNOLOGICOS INTERDISCIPLINARES BASADOS EN ARDUINO." Encuentro Internacional de Educación en Ingeniería ACOFI 2014. 2014.

[5] Cabero, Julio, et al. "Nuevas tecnologías aplicadas a la educación." Madrid, Síntesis (2000).

[6] Introducción a JSON
<https://json.org/json-es.html>

[7] Android Things
<https://developer.android.com/things/index.html>

[8] IoT and iOS - Lessons Learned
<https://www.dotconferences.com/2017/01/hugues-bernet-rollande-iot-and-ios-lessons-learned>

[9] D. McMaster, K. McCloghrie. "Definitions of Managed Objects for IEEE 802.3 Repeater Devices"

<https://tools.ietf.org/html/rfc2108>

[10] P. Calhoun, M. Montemurro, D. Stanley "Control and Provisioning of Wireless Access Points (CAPWAP) Protocol Binding for IEEE 802.11"

<https://tools.ietf.org/html/rfc5416>