



Universidad Tecnológica del Centro de Veracruz

Programa Educativo de Ingeniería en Tecnologías de la Información

Reporte de Titulación en Ingeniería de Tecnologías de la Información

Centro de Convenciones William O. Jenkins

**Implementación de Una Aplicación Móvil y Servomotores para
Aumentar y Controlar el Angulo de Visión de las Cámaras de
Vigilancia en el Centro de Convenciones William O. Jenkins**

Presenta: Aldo Fernández Rivera

Puebla, Puebla a abril de 2016



Universidad Tecnológica del Centro de Veracruz

Ingeniería en Tecnologías de la Información

Asesor Industrial: Ing. Juan Carlos Xicoténcatl Correa

Asesor Académico: Ing. Víctor Jacinto Mejía Lara

Alumno: Aldo Fernández Rivera

Contenido

1.1	Glosario del Proyecto	4
1.2	Requerimientos Técnicos	5
1.3	Modelo de Casos de Uso	6
1.4	Pruebas de Aceptación.....	6
2	Modelo de Diseño.....	8
2.1	Modelo de Despliegue.....	8
2.2	Diagramas de Secuencia.....	9
2.3	Modelo de Interfaces de Usuario	9
2.3.1	Menú Principal	9
2.3.2	Interfaces de los Salones o Zonas	11
2.3.3	Interfaz de Streaming	11
2.4	Placa de Pruebas.....	12
2.5	Esquema.....	13
2.6	Placa de Circuito Impreso.....	14
2.7	Modelo de Datos.....	15
3	Construcción	17
3.1	Prototipo.....	17
3.2	Configuración del módulo ESP8266	18
3.3	Código Arduino.....	24
3.4	Aplicación Móvil (ExpoCam).....	25
3.4.1	Login.....	25
3.4.2	Menú IOS	25
3.4.3	Menú Android.....	26
3.4.4	Cámaras Centro de Convenciones.....	27
3.4.5	Cámaras Centro Expositor	28
3.4.6	Streaming	29
3.4.7	Código Fuente.....	30
4	Metodología	33
4.1	AUP	33
4.2	Entregables Mínimos.....	33
5	Evaluación de Resultados	33
6	Conclusiones	33

7 Referencias	34
8 Apéndices o Anexos	35

Figura 2.1 Caso de uso.	6
Figura 2.1 Diagrama de despliegue.....	8
Figura 2.2 Diagrama de secuencia.	9
Figura 2.3 Menú principal	10
Figura 2.4 Interfaz salón la oriental	11
Figura 2.5 Interfaz de streaming	12
Figura 2.6 Placa de Pruebas	13
Figura 2.7 Esquema electrónico.....	14
Figura 2.8 Placa de circuito impreso	15
Figura 2.9 Diagrama relacional	16
Figura 3.1 Prototipo	17
Figura 3.2 Prototipo	17
Figura 3.3 Prototipo	18
Figura 3.4 código Arduino configuración ESP8266	18
Figura 3.5 Inicio del ESP8266	19
Figura 3.6 ESP8266 Comando AT	19
Figura 3.7 ESP8266 comando AT+CWMODE?.....	20
Figura 3.8 ESP8266 comando AT cambio de modo	20
Figura 3.9 ESP8266 comando AT+CWLAP	21
Figura 3.10 ESP8266 comando AT+CWJAP	21
Figura 3.11 ESP8266 comando AT+CIFSR.....	22
Figura 3.12 ESP8266 comando AT+CIPMUX=1	22
Figura 3.13 ESP8266 comando AT+CIPSERVER=1,80	23
Figura 3.14 ESP8266 obteniendo datos por GET	23
Figura 3.15 ExpoCam login.....	25
Figura 3.16 Menú IOS.....	26
Figura 3.17 Menú Android	27
Figura 3.18 Cámaras Centro de Convenciones	28
Figura 3.19 Cámaras Centro Expositor.....	29
Figura 3.20 Streaming de un DVR	30
Figura 3.21 app.js	31
Figura 3.22 Controller.js.....	31

Modelado de Requerimientos

1.1 Glosario del Proyecto

1. **RT:** Requerimiento técnico.
2. **WLAN:** Red inalámbrica de área local.
3. **VLAN:** Red virtual de área local.

4. **IP:** Etiqueta numérica para la identificación lógica dentro de una red.
5. **IOS:** Sistema operativo móvil perteneciente a la compañía Apple.
6. **Circuito electrónico:** Serie de componentes eléctricos conectados entre sí.
7. **Servo:** Dispositivo que tiene la capacidad de ubicarse en cualquier posición dentro de un rango de operación.
8. **Streaming:** Transmisión de video en tiempo real.

1.2 Requerimientos Técnicos

- RT1 La aplicación móvil puede ser utilizada si y sólo si se encuentra autenticado en una WLAN.
- RT2 El movimiento de las cámaras debe ocurrir no mayor a 3 segundos el 90% del tiempo.
- RT3 La visualización del vídeo puede tener un retraso de no más de 3 segundos el 90% del tiempo.
- RT4 Cada uno de los circuitos electrónicos debe pertenecer a una VLAN y se les debe asignar una IP estática.
- RT5 La aplicación sólo se podrá instalar en sistemas operativos Android 4.0.3, IOS 5 y BlackBerry OS 7.1 o superiores.

1.3 Modelo de Casos de Uso

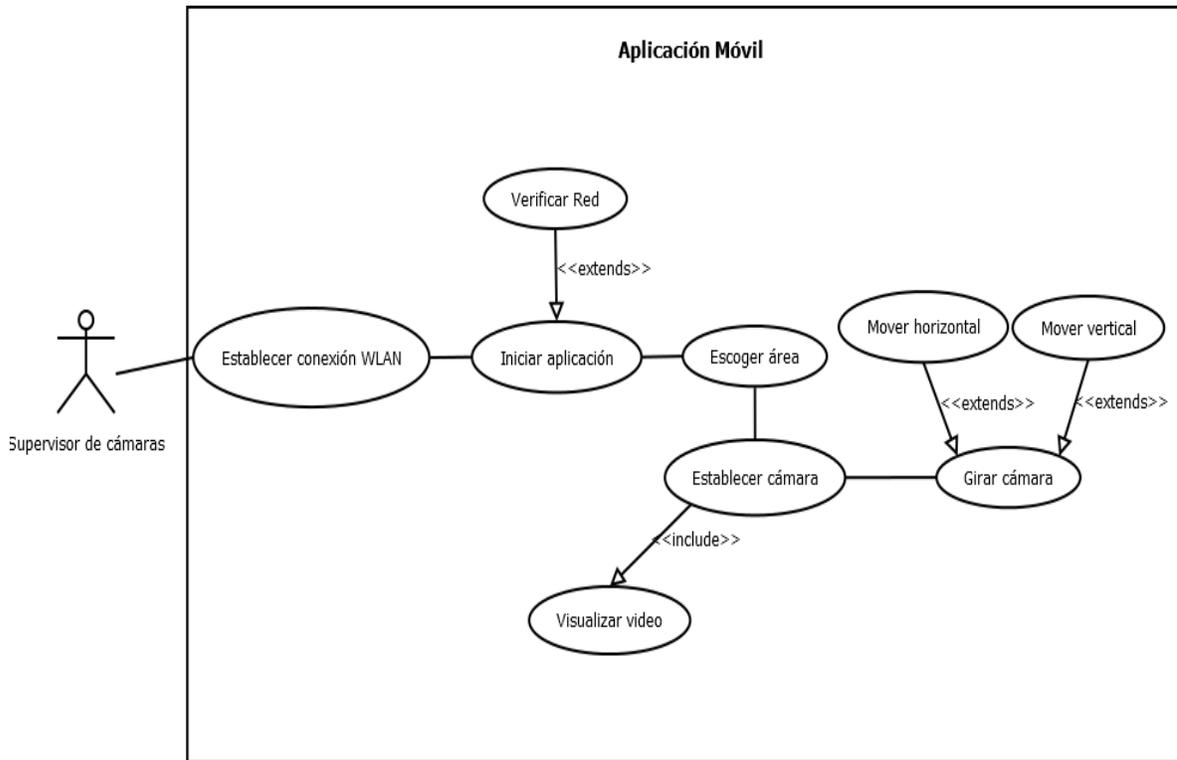


Figura 2.1 Caso de uso.

1.4 Pruebas de Aceptación

ID	P001
Descripción	El inicio de la aplicación móvil debe realizar una verificación de la WLAN conectada.
Preparación	<ol style="list-style-type: none"> 1. Autenticarse las siguientes WLAN: <ul style="list-style-type: none"> • INFINITUM7D0C • Puebla-2a • Puebla-2b • Puebla-2c • AMTW
Instrucciones	<ol style="list-style-type: none"> 1. Prender el celular. 2. Ir a ajustes. 3. Ir a Wi-Fi. 4. Activar Wi-Fi. 5. Autenticarse en las redes mencionadas.
Resultados Esperados	Al iniciar la aplicación móvil, ésta debe cargar exitosamente y mostrar un menú.

ID	P002
Descripción	Visualización de vídeo de las cámaras.
Preparación	<ol style="list-style-type: none"> 1. Iniciar Aplicación 2. Elegir las diferentes zonas: <ul style="list-style-type: none"> • Analco • La Luz • El Alto • La Violeta • La Oriental • La Esperanza • El Parián • La Concordia • El Chacuaco • Angelópolis 3. Elegir cámara <ul style="list-style-type: none"> • Elegir Cámara • Cámara 1 • Cámara 2 • Cámara 3 • Etc.
Instrucciones	<ol style="list-style-type: none"> 1. Iniciar la aplicación 2. Elegir una de las zonas. 3. Elegir una cámara perteneciente a la zona.
Resultados Esperados	Se debe visualizar el vídeo de la cámara seleccionada que pertenece a cierta zona.

ID	P003
Descripción	Movimiento de las cámaras.
Preparación	<ol style="list-style-type: none"> 1. Iniciar Aplicación 2. Elegir las diferentes zonas: <ul style="list-style-type: none"> • Analco • La Luz • El Alto • La Violeta • La Oriental • La Esperanza • El Parián • La Concordia • El Chacuaco • Angelópolis 3. Elegir Cámara <ul style="list-style-type: none"> • Cámara 1

	<ul style="list-style-type: none"> • Cámara 2 • Cámara 3 • Etc.
Instrucciones	<ol style="list-style-type: none"> 1. Iniciar la aplicación 2. Elegir una de las zonas. 3. Elegir una cámara perteneciente a la zona. 4. Presionar uno de los vectores que salen a los lados.
Resultados Esperados	Las cámaras se deben ir moviendo a cierta velocidad conforme a la dirección indicada.

2 Modelo de Diseño

2.1 Modelo de Despliegue

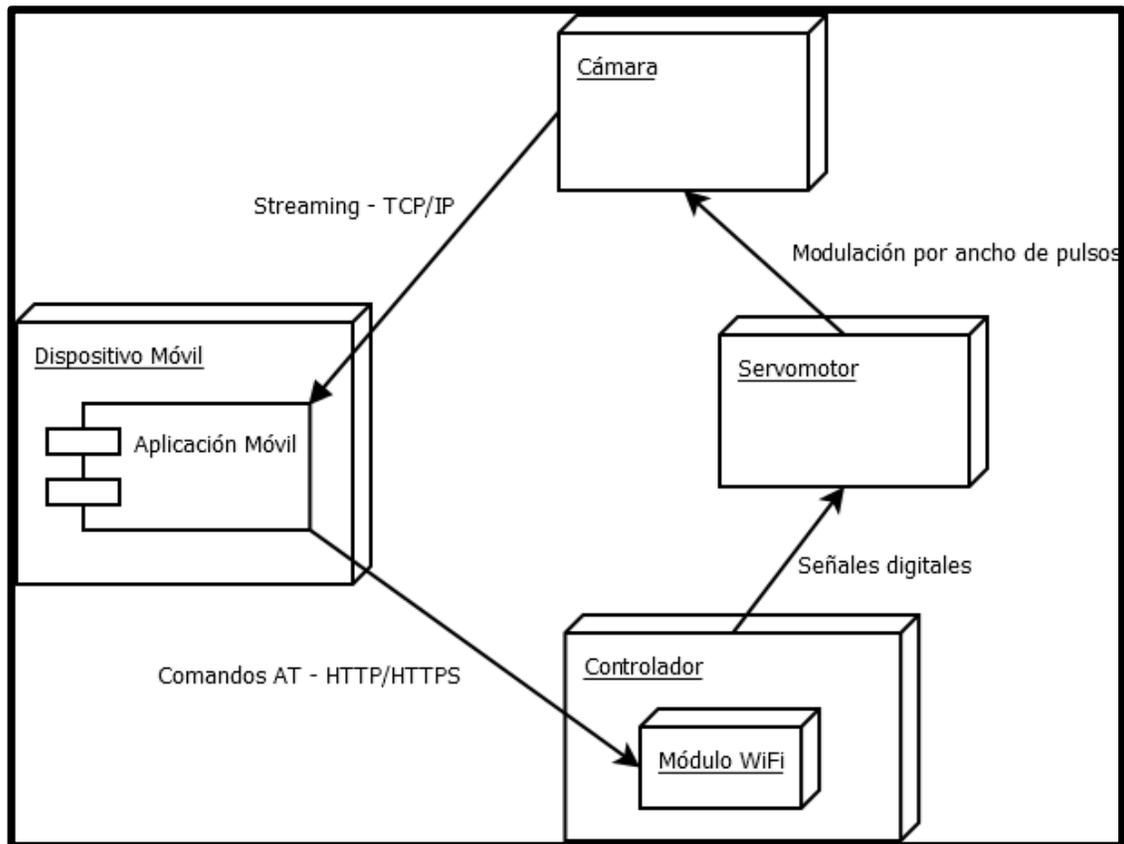


Figura 2.1 Diagrama de despliegue.

2.2 Diagramas de Secuencia

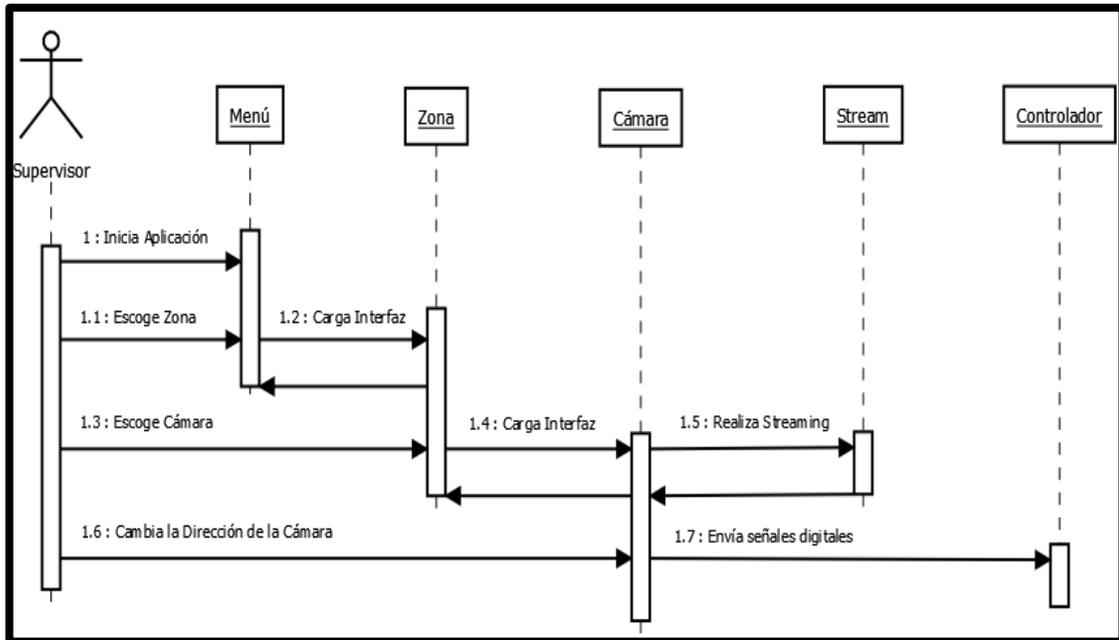


Figura 2.2 Diagrama de secuencia.

2.3 Modelo de Interfaces de Usuario

2.3.1 Menú Principal

El objetivo de la interfaz principal es identificar cada una de las zonas de una manera diferente y sencilla en comparación con otras aplicaciones similares. En total son 10 zonas, las cuales son las siguientes:

1. **Analco:** Es parte del salón San Francisco, cuenta con un espacio de 31.72m x 40.90m. Es utilizado para tipos de montaje como: banquetes, stands y coctelería.
2. **La Luz:** Al igual que el salón Analco, La luz es una parte del salón San Francisco con un espacio de 21.02m x 40.90m.
3. **El Alto:** Es la última parte del salón San Francisco y tiene un espacio de 20.59m x 40.90m que en total con la unión de los tres salones llega a tener un espacio de 73.33m x 40.90m.
4. **La Violeta:** Su espacio es de 12.25m x 11.56m. Éste salón se puede dividir en dos salas del mismo tamaño y se suelen utilizar para oficinas, salas de prensa o reuniones de trabajo.
5. **La Esperanza:** Cuenta con un espacio de 5.56m x 15.60m, es uno de los salones más chicos del recinto y se suele ocupar para salas de prensa entre otros eventos pequeños.

6. **La Oriental:** Su espacio es de 12.25m x 11.56m, ubicado entre el salón la Esperanza y la Violeta. El Salón se puede dividir en 2 salones: La oriental I y la Oriental II de 6 metros de ancho cada uno.
7. **El Parián:** Es el salón más pequeño que ofrece el centro de convenciones. Su espacio es de 5.84m x 6.40m y en su mayoría se ocupa para eventos muy pequeños.
8. **La Concordia:** Con un espacio de 6.1m x 6.35m, es el segundo salón más pequeño con el que cuenta el recinto, ubicado al lado del salón parían.
9. **El Chacuaco:** Su nombre se debe a la chimenea que aún conserva de la fábrica “La Mascota”. En esta plazuela se encuentra el acceso al balcón del puente peatonal que une al CCP con el Barrio del Artista, El Parián y los Sapos.
10. **Auditorio Angelópolis:** El cual cuenta con el equipo más moderno, cabina de audio y video, traducción simultánea, baño propio, lobby, con capacidad para 264 personas.



Figura 2.3 Menú principal

2.3.2 Interfaces de los Salones o Zonas

Ésta interfaz muestra la arquitectura y las diferentes cámaras que se encuentran dentro del salón o zona seleccionada en el menú principal. Cada una de las cámaras contará con una referencia a su propio streaming.



Figura 2.4 Interfaz salón la oriental

2.3.3 Interfaz de Streaming

La interfaz de streaming visualiza la transmisión de vídeo y controla el movimiento de la cámara que fue seleccionada en el menú de zonas. Ésta interfaz abarca casi la totalidad de los dispositivos móviles con el objetivo de proyectar un video con la mayor resolución posible.

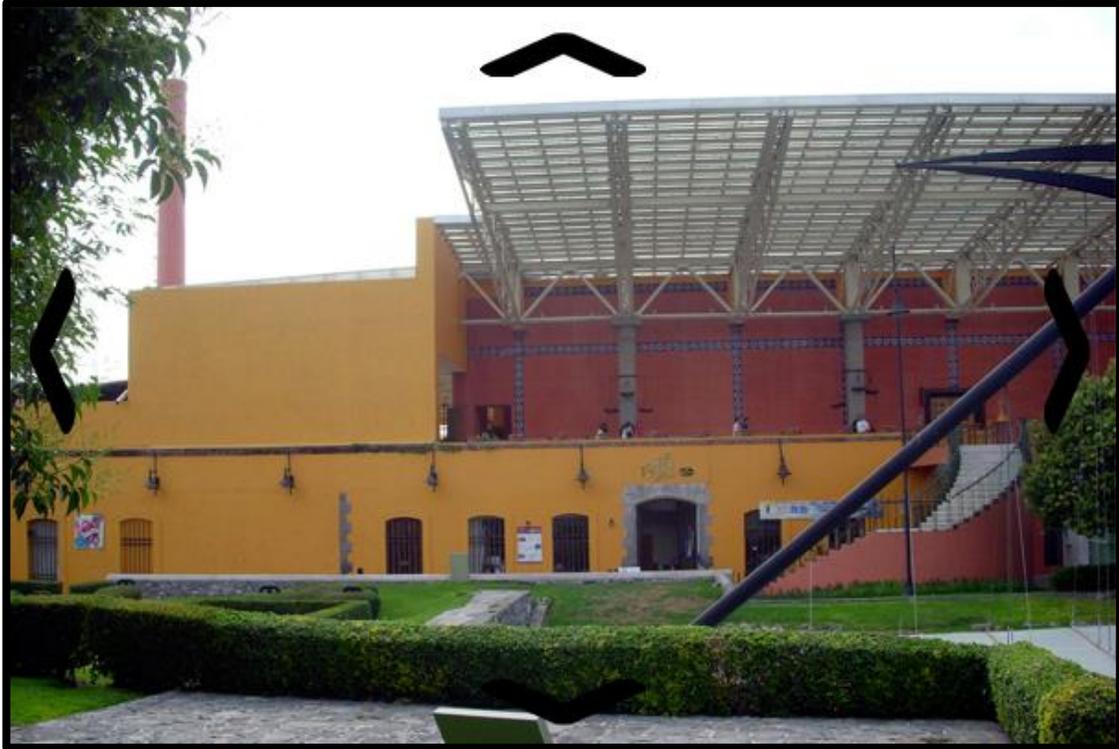


Figura 2.5 Interfaz de streaming

2.4 Placa de Pruebas

En la figura 3.1, se puede observar la integración de los servomotores con el módulo ESP8266, mediante una placa de pruebas y un microcontrolador (Arduino Uno).

La placa de pruebas está compuesta de 5 tipos de cables, cada color representa una función:

Rojo: Conductor de corriente directa de 5 voltios.

Negro: Conductor de conexión a tierra.

Amarillo: Línea de control por la que se envía la señal codificada para comunicar el ángulo en el que se debe posicionar.

Verde: Conductor de transmisión de señal.

Naranja: Conductor de recepción de señal.

Arduino Uno cuenta con 13 pines digitales que pueden interpretar y emitir señales. De los 13 disponibles sólo se ocupan 4 pines.

Dos de ellos son el pin 9 y el pin 8 que son para enviar datos a los servomotores y se comunican por medio de un cable amarillo. Los otros dos pines (3 y 2) son para transmitir y recibir datos del módulo ESP8266, el pin 3 utiliza un cable color naranja (RX) y el pin 2 un cable de color verde (TX).

Por otro lado, se encuentran los pines de poder (corriente eléctrica). De los 5 pines totales de poder sólo se utilizan 2 pines, el pin de 5 volts que se utiliza mediante un cable de color rojo y un pin de tierra (GND) que por norma se utiliza con un cable de color negro.

Como complemento electrónico se puede utilizar una resistencia de 440 ohms con una tolerancia de +/- 5 o un regulador de voltaje con el fin de reducir el voltaje en 3.3 volts para la alimentación del módulo ESP8266 ya que un voltaje superior a esta quemaría el componente.

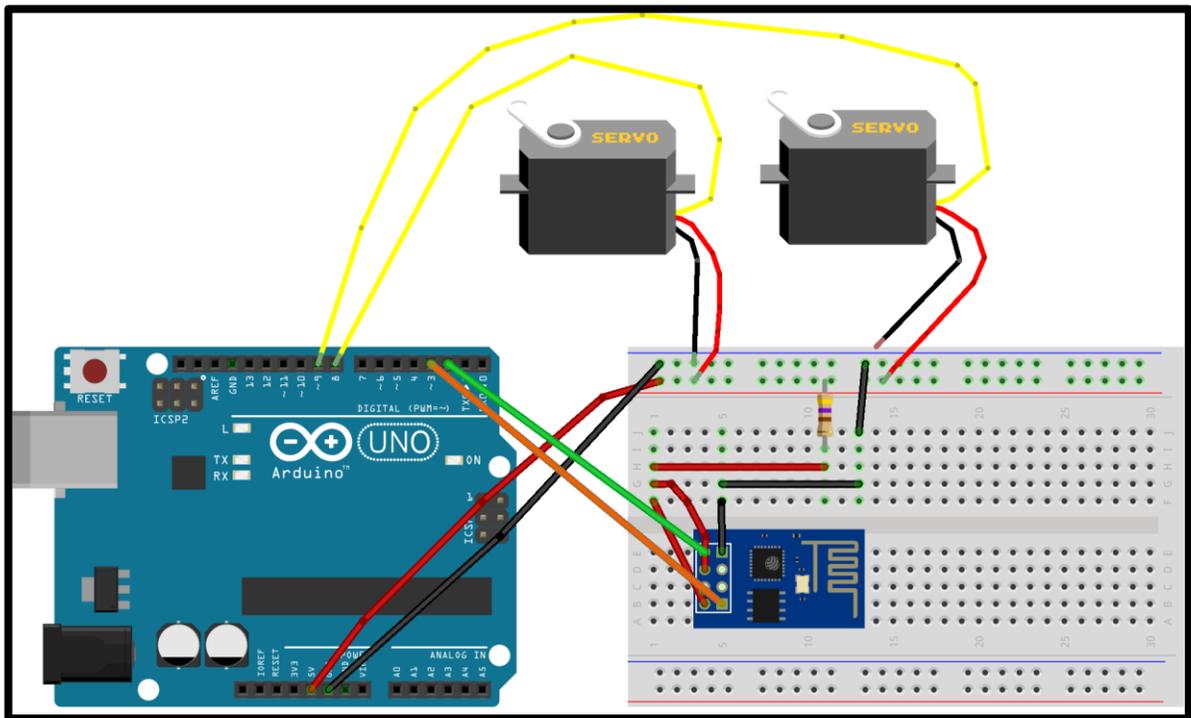


Figura 2.6 Placa de Pruebas

2.5 Esquema

La figura 4.1 se representa el esquema electrónico con información más detallada de las entradas y salidas de los componentes.

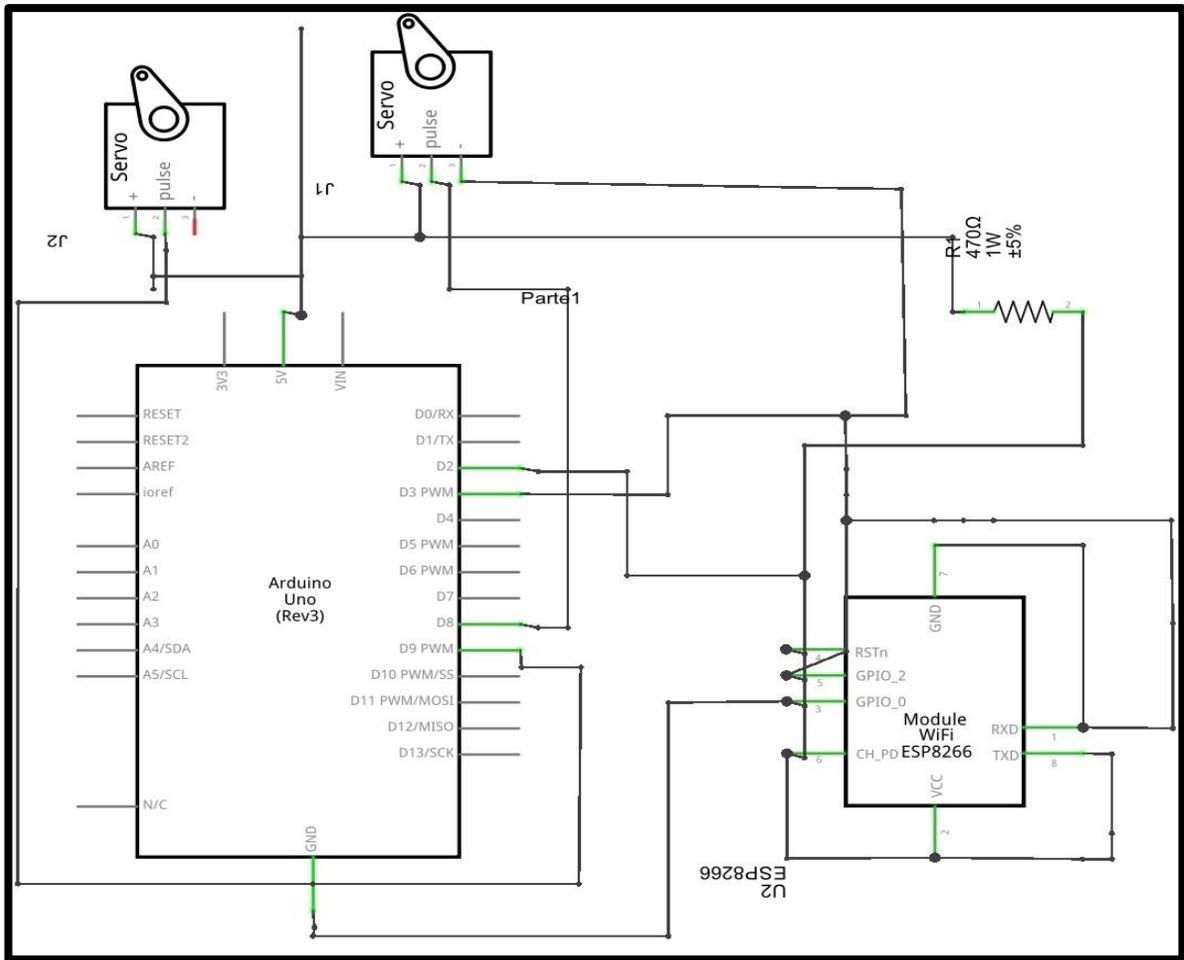


Figura 2.7 Esquema electrónico

2.6 Placa de Circuito Impreso

La figura 5.1 representa el prototipo de un diseño para realizar una placa de circuito impreso.

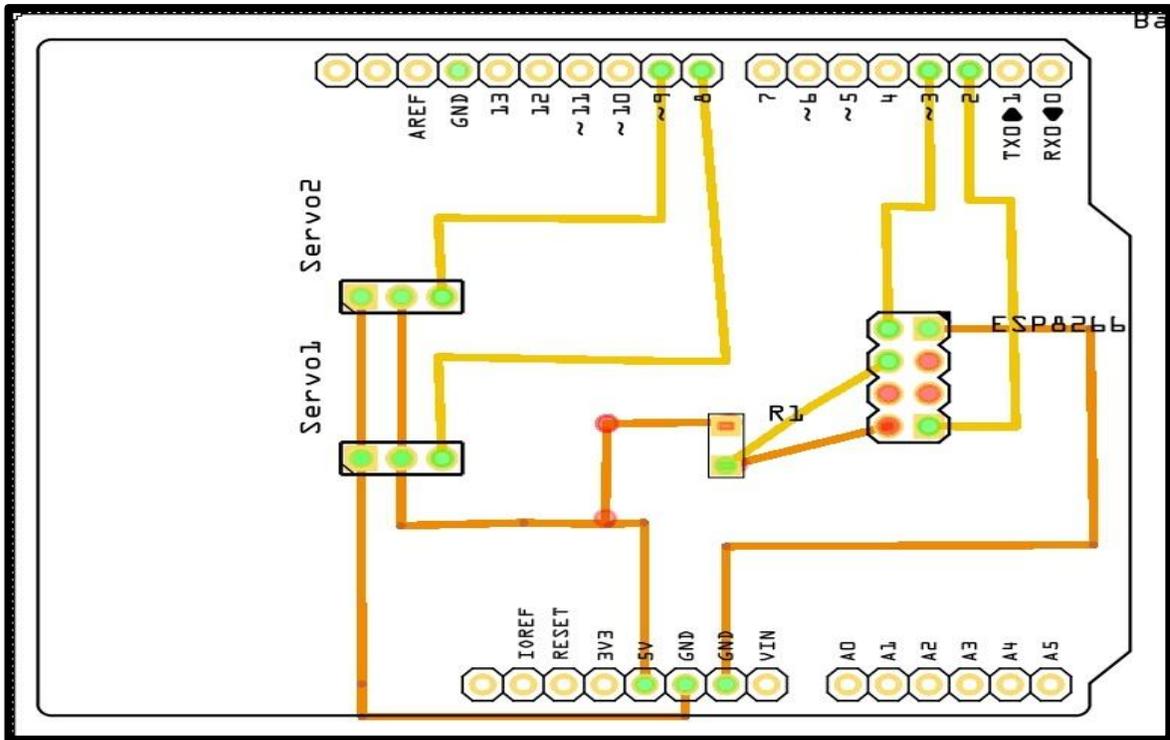


Figura 2.8 Placa de circuito impreso

2.7 Modelo de Datos

Como se puede observar en la figura 2.9, el diagrama relacional contiene 4 tablas: tblDvrs, tblUsers, tblDepartments y tblDvr_users, todas las tablas con diferentes funciones.

Tabla de usuarios(tblUsers)

Esta tabla se encarga de almacenar todos los usuarios que se desean agregar, contiene atributos como: user(usuario), pass(contraseña) y active que habilita o deshabilita el usuario. Las contraseñas ingresadas en el campo "pass" se encriptan por cuestiones de seguridad en el formato MD5.

Tabla de departamentos(tblDepartments)

Almacena los departamentos con los que cuenta convenciones y parques como, por ejemplo: Centro de convenciones William O. Jenkins o Centro expositor de Puebla, entre otros. Cuenta con atributos como: name(nombre), info(información), img(imagen) y active que al igual que la tabla tblUsers habilita o deshabilita departamentos.

Tabla de grabadoras de video digital(tblDvrs)

Contiene algunos datos de los DVR's como es su ip y una relación con la tabla de departamentos(tblDepartments) para la identificación de los DVR's pertenecientes a cada departamento. También se pueden habilitar o deshabilitar en caso de que algún DVR deje de operar.

Tabla de asignación de grabadoras de video digital a usuarios(tblDvr_users)

Es una tabla de asignación en la cual se agregan usuarios junto con el DVR al que tiene acceso. Este proceso se puede repetir varias veces es decir un usuario puede tener varios DVR's, pero no puede tener el mismo DVR dos veces debido a la llave primaria compuesta que contiene la tabla. Al igual que todas las tablas se pueden habilitar y deshabilitar la asignación de DVR's mediante el campo active.

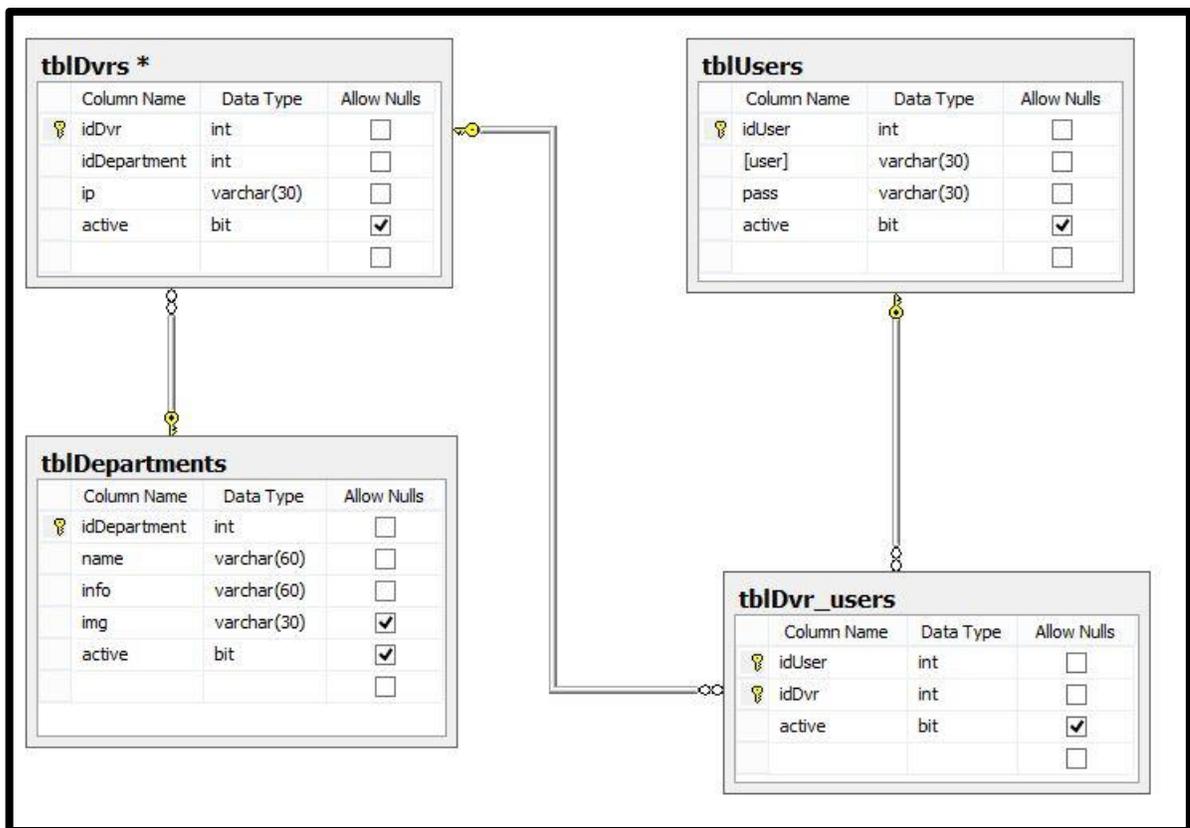


Figura 2.9 Diagrama relacional

3 Construcción

3.1 Prototipo

En las figuras 3.1, 3.2, 3.3 se puede observar un prototipo. Este está compuesto de 2 servomotores con un torque de 4.3 kg/cm a 5 volts, un ESP8266 con una alimentación de 3.3 volts, 2 Protoboards unidas, algunos jumpers y un Arduino Uno.

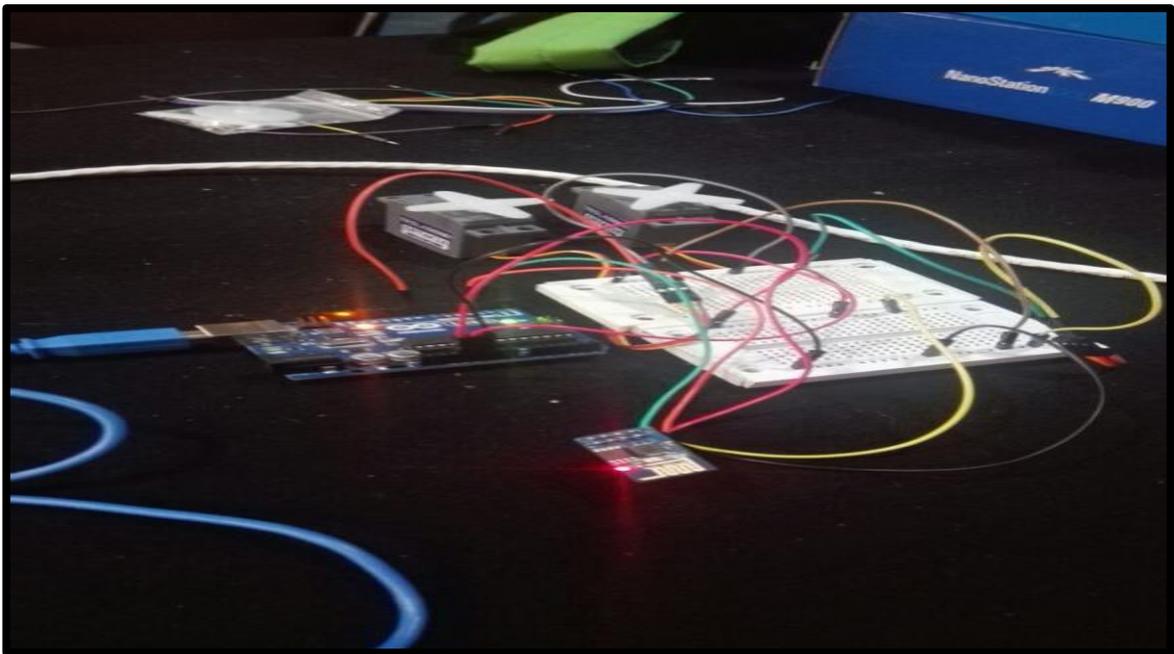


Figura 3.1 Prototipo

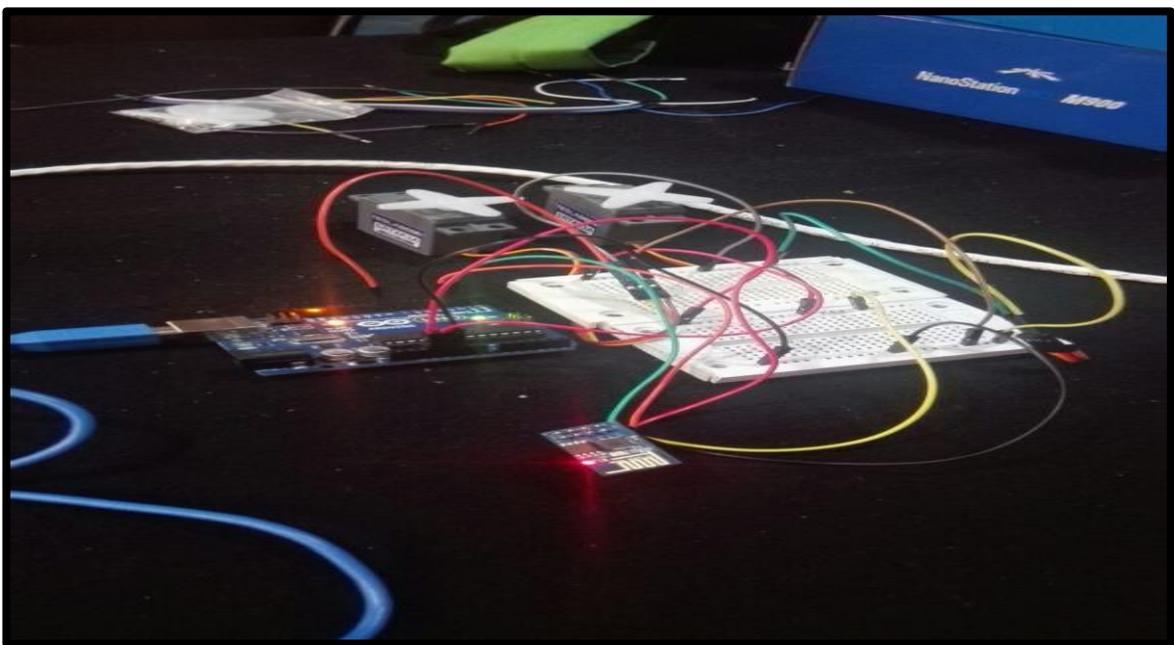


Figura 3.2 Prototipo

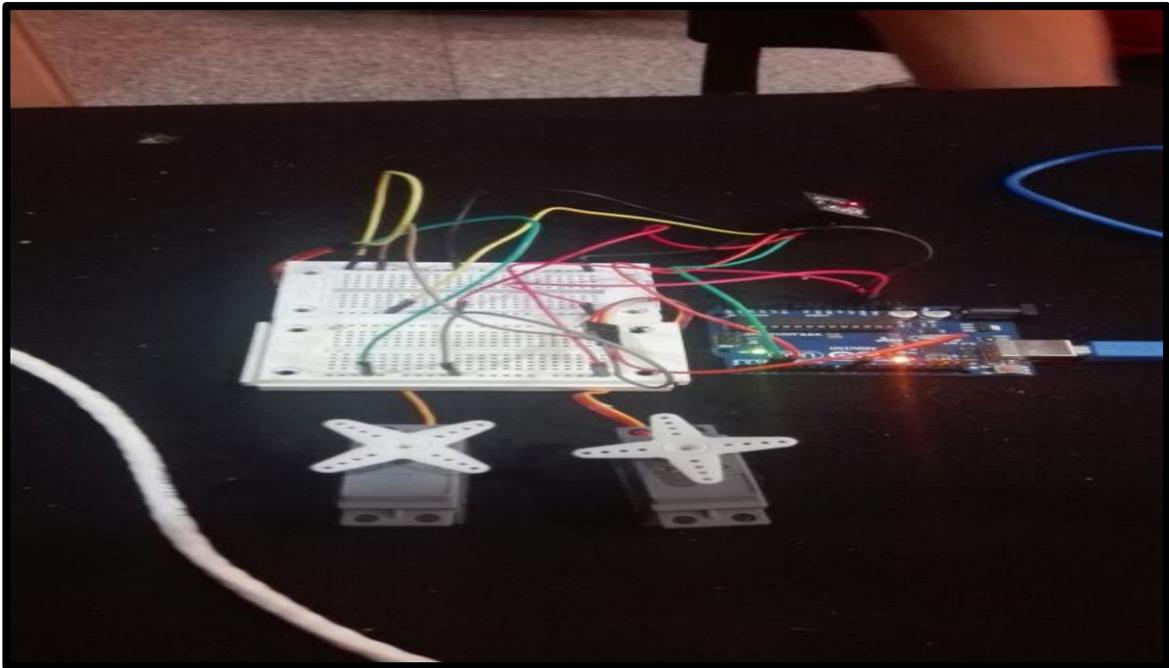


Figura 3.3 Prototipo

3.2 Configuración del módulo ESP8266

Para realizar la configuración del ESP8266 se debe realizar un prototipo como en la figura 2.6. Una vez establecido el prototipo, se configura mediante código la comunicación entre el módulo ESP8266 y el Arduino. Se inicializa el serial y el software serial a una transmisión digital de 115200 baudios. El código se muestra en la figura 3.4.

```
1 #include <SoftwareSerial.h>
2 SoftwareSerial esp(3, 2) // RX | TX
3
4 void setup() {
5     Serial.begin(115200);
6     esp.begin(115200);
7 }
8
9 void loop() {
10     if(esp.available()){
11         char c = esp.read();
12         Serial.print(c);
13     }
14     if(Serial.available()){
15         char c = Serial.read();
16         esp.print(c);
17     }
18 }
19 }
```

Figura 3.4 código Arduino configuración ESP8266

Una vez compilado el código, se abre el monitor serie del IDE de Arduino y se configura la transmisión digital a 115200 baudios, debe de generar un mensaje como en la figura 3.5.

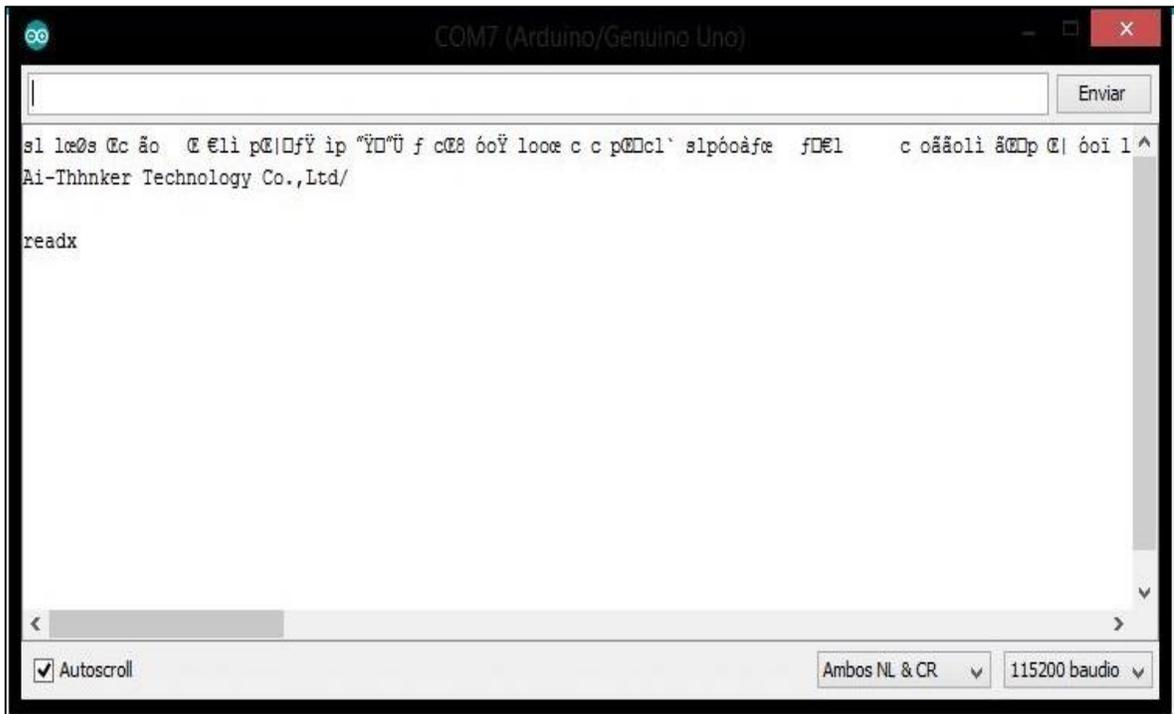


Figura 3.5 Inicio del ESP8266

Para verificar que la configuración de recepción y transmisión están correctas, se envía el comando "AT", el modulo responde con un sencillo "OK", para indicar que tenemos línea abierta como se muestra en la figura 3.6.

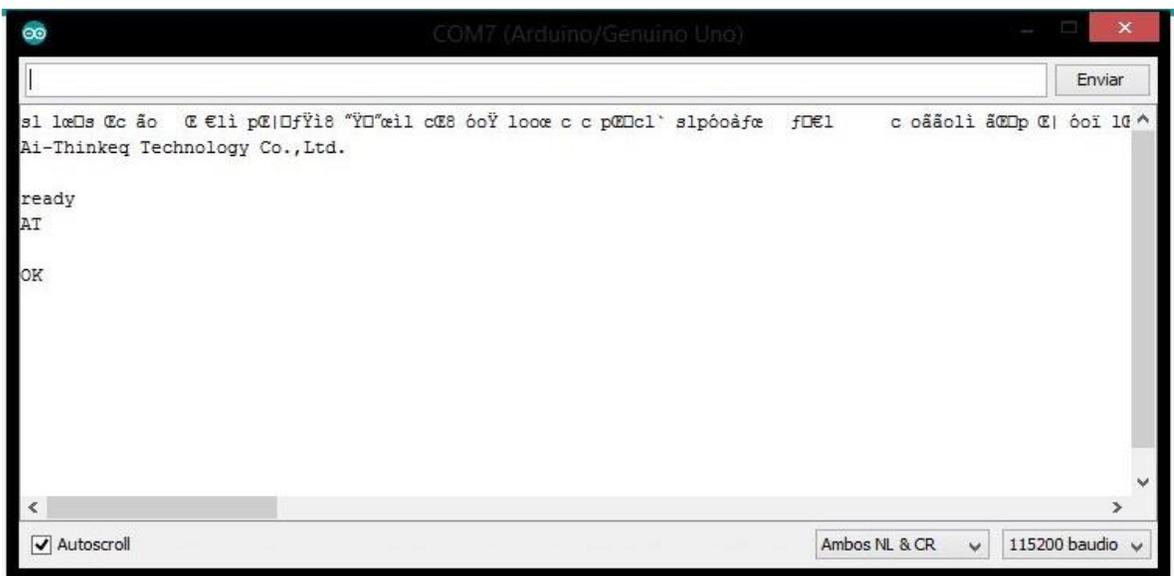


Figura 3.6 ESP8266 Comando AT

Para verificar el modo de fábrica en el que se encuentra el módulo ESP8266 se inserta la instrucción AT+CWMODE como se muestra en la figura 3.7.



Figura 3.7 ESP8266 comando AT+CWMODE?

Si el modulo responde diferente a "+CWMODE:3" se inserta la instrucción AT+CWMODE=3, que es el modo adecuado para identificar las redes WLAN. Si el cambio fue realizado el modulo debe responder "OK" así como se muestra en la figura 3.8.

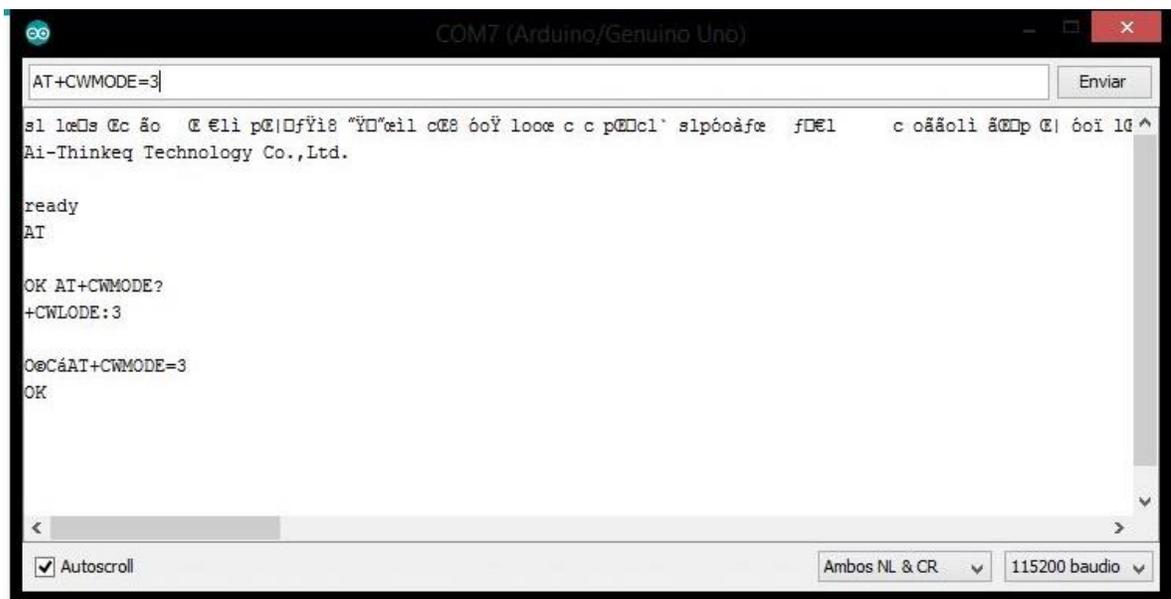


Figura 3.8 ESP8266 comando AT cambio de modo

Para identificar todas las redes WLAN disponibles se inserta el comando AT+CWLAP, el modulo debe responder algo más o menos parecido a la figura 3.9.



Figura 3.9 ESP8266 comando AT+CWLAP

Una vez identificada la red a la cual se va a conectar se inserta el comando AT+CWJAP="SSID","CONTRASEÑA", en este caso el SSID es "BOSH-PBX", por cuestiones de seguridad no se muestra la contraseña de la red "BOSH-PBX", pero el modulo debe responder "WIFI CONNECTED, WIFI GOT IP" como se muestra en la figura 3.10.

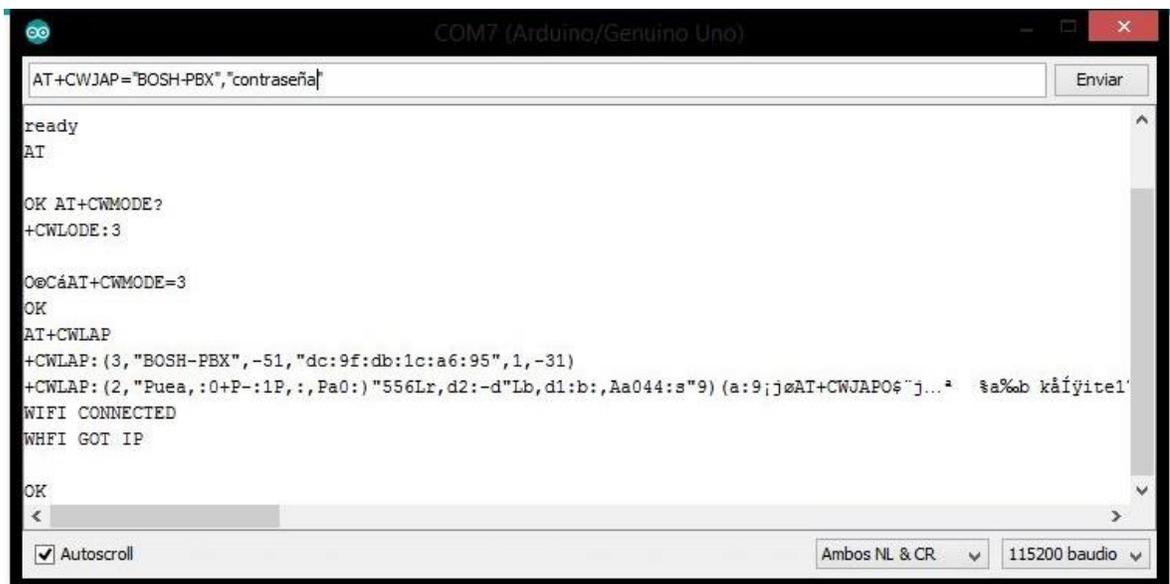
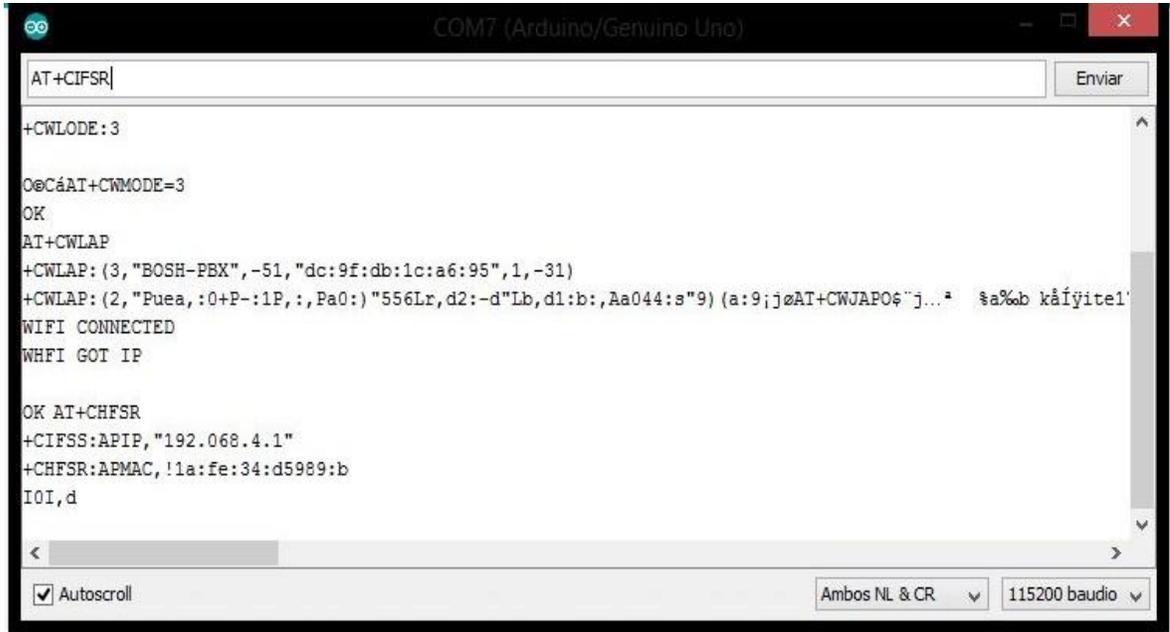


Figura 3.10 ESP8266 comando AT+CWJAP

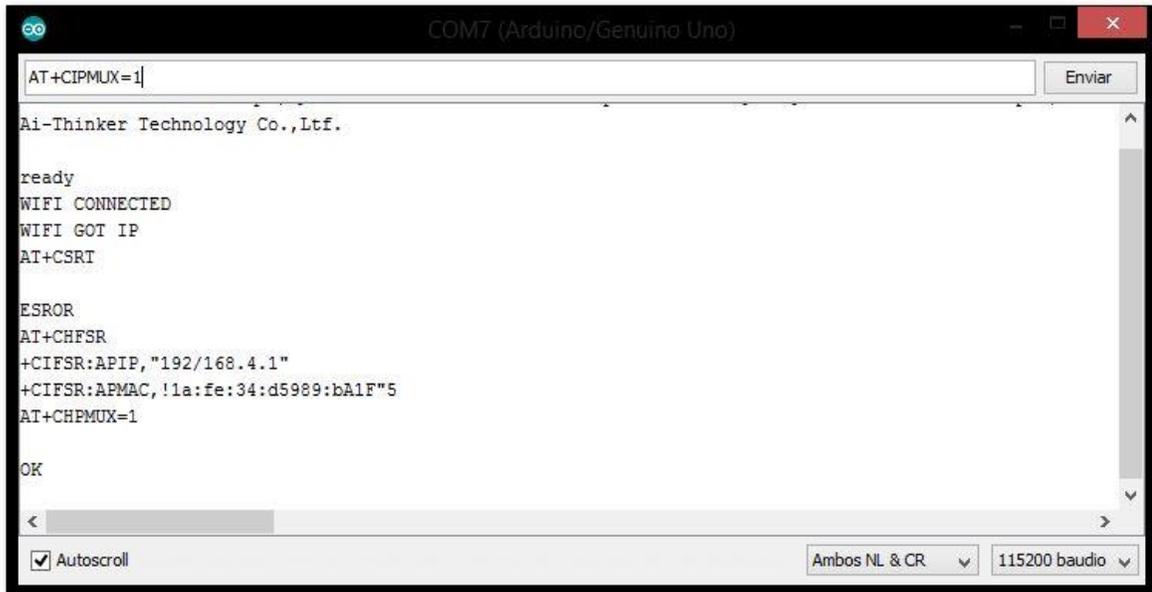
Para obtener la IP que tiene asignada el módulo ESP8266, se inserta el comando AT+CIFSR, en este caso la IP asignada es la 192.168.4.1 así como se muestra en la figura 3.11.



```
COM7 (Arduino/Genuino Uno)
AT+CIFSR
+CWL0DE:3
O@CáAT+CWM0DE=3
OK
AT+CWLAP
+CWLAP: (3, "BOSH-PBX", -51, "dc:9f:db:1c:a6:95", 1, -31)
+CWLAP: (2, "Puea, :0+P-:1P, :, Pa0:) "556Lr,d2:-d"Lb,d1:b:,Aa044:s"9) (a:9;jj@AT+CWJAP0$`j...` $a%b káíyte1'
WIFI CONNECTED
WIFI GOT IP
OK AT+CIFSR
+CIFSR:APIP,"192.068.4.1"
+CIFSR:APMAC,!1a:fe:34:d5989:b
IOI,d
```

Figura 3.11 ESP8266 comando AT+CIFSR

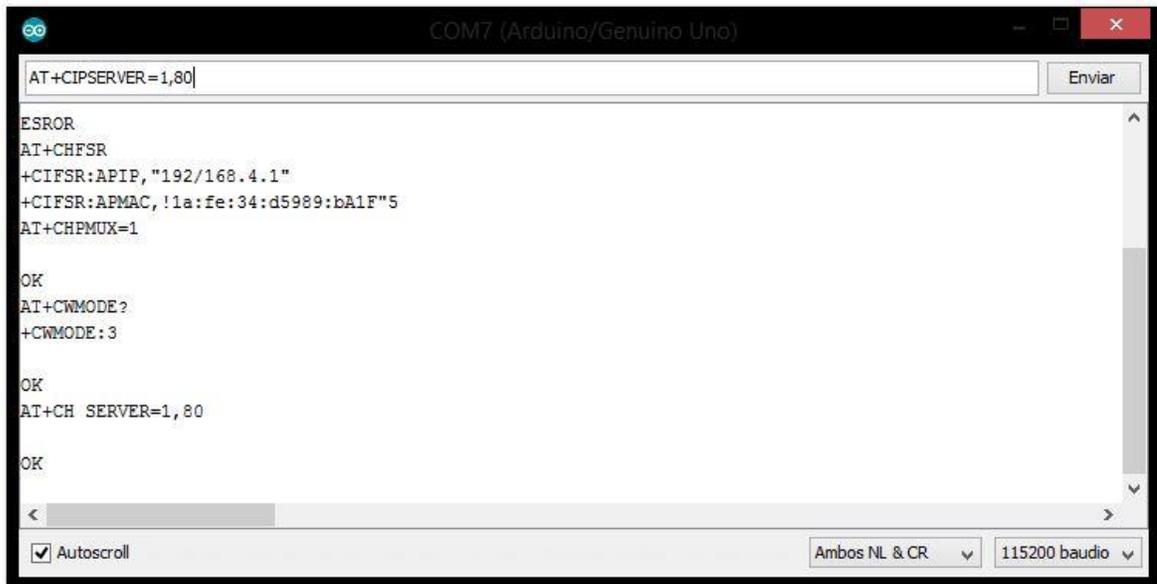
El comando que se muestra en la figura 3.12 habilita múltiples conexiones simultaneas. Al igual como todos los comandos AT, si los comandos son correctos el modulo responde con un "OK".



```
COM7 (Arduino/Genuino Uno)
AT+CIPMUX=1
Ai-Thinker Technology Co.,Ltf.
ready
WIFI CONNECTED
WIFI GOT IP
AT+CSRT
ESROR
AT+CHFSR
+CIFSR:APIP,"192/168.4.1"
+CIFSR:APMAC,!1a:fe:34:d5989:bA1F"5
AT+CHPMUX=1
OK
```

Figura 3.12 ESP8266 comando AT+CIPMUX=1

Para habilitar el servicio web del módulo ESP8266 se inserta el comando AT+CIPSERVER=1,80 donde 1 es el número de servicio y 80 es el número del puerto, así como se muestra en la figura 3.13.



```
COM7 (Arduino/Genuino Uno)
AT+CIPSERVER=1,80
Enviar
ESROR
AT+CHFSR
+CIFSR:APIP, "192/168.4.1"
+CIFSR:APMAC, '1a:fe:34:d5989:bA1F'5
AT+CHPMUX=1
OK
AT+CWMODE?
+CWMODE:3
OK
AT+CH SERVER=1,80
OK
Autoscroll
Ambos NL & CR
115200 baudio
```

Figura 3.13 ESP8266 comando AT+CIPSERVER=1,80

El último paso por realizar es verificar que el módulo interpreta los parámetros que se mandan por GET, para ello en un navegador se escribe la IP del módulo ESP8266 junto con los parámetros a enviar. En este caso los parámetros son 2: “servo1” que controla la rotación del servo número 1 y “servo2” que realiza la misma función, pero para el servomotor número 2. La URL a utilizar es: “192.168.4.1/servo1=130, servo2=70”, el módulo debe responder un mensaje como se muestra en la figura 3.14



```
COM7 (Arduino/Genuino Uno)
WIFI CONNECTED
WIFI GOT IP
AT+CIPMUX=1
OK
AT+CIPSERVER=1,80
OK
0, CONNDCT
1, CONNECT
+HPD,0,410:GET /servo1=130,servo2=70 HTTP/1.1
Host: 192.168/4.1
n-h-c, hp;eesrlw)/To0Sifpe81, CLOSED
Autoscroll
Ambos NL & CR
115200 baudio
```

Figura 3.14 ESP8266 obteniendo datos por GET

3.3 Código Arduino

Una vez configurando el módulo ESP8266, se compila el código final que se muestra en la figura Este último realiza la comunicación entre ESP8266 y los servomotores.

```
1 #include <SoftwareSerial.h>
2 #include <Servo.h>
3
4 SoftwareSerial esp(3, 2); // RX | TX
5 Servo servol, servo2;
6 String espMsg;
7
8 void setup() {
9     esp.begin(115200);
10    servol.attach(8);
11    servo2.attach(9);
12 }
13
14 void loop() {
15     if(esp.available()){
16         espMsg += esp.read();
17         if(espMsg.indexOf("servol") >= 0){
18             String param = getParam("servol");
19             moveServo(1, param.toInt());
20             espMsg.replace("servol="+param, "");
21         }
22         if(espMsg.indexOf("servo2" >= 0)){
23             String param = getParam("servo2");
24             moveServo(2, param.toInt());
25             espMsg.replace("servo2="+param, "");
26         }
27     }
28 }
```

```
29
30 String getParam(String paramName) {
31     int index = espMsg.indexOf(paramName);
32     paramName += "=";
33     index += paramName.length();
34     return espMsg.substring(index, index+3);
35 }
36
37 void moveServo(int servo, int rotation){
38     if(servo == 1){
39         servol.write(rotation);
40     }else if(servo == 2){
41         servo2.write(rotation);
42     }
43 }
44 }
```

3.4 Aplicación Móvil (ExpoCam)

La aplicación móvil ExpoCam está desarrollada en Angular JS, HTML5 y Ionic Framework que permite crear aplicaciones multiplataforma.

3.4.1 Login

En la figura 3.15 se puede observar el login de la aplicación ExpoCam. Por cuestiones de seguridad los campos tienen una longitud mínima de 6 caracteres y máxima de 16. Para lograr la comunicación entre los dispositivos y una base de datos se creó un servicio web con la tecnología ASP.NET alojado en un servidor web IIS 8.0.



Figura 3.15 ExpoCam login

3.4.2 Menú IOS

En la figura 3.16 se observa el menú en un dispositivo IOS. El menú por defecto es el menú de departamentos que muestra los departamentos o franquicias disponibles que tiene habilitado el usuario desde la base de datos.



Figura 3.16 Menú IOS

3.4.3 Menú Android

En la figura 3.17 se visualiza el menú de un dispositivo Android que a diferencia del menú de IOS, la barra de navegación se encuentra en la parte superior. La finalidad de este diseño es mantener la apariencia de una aplicación nativa.

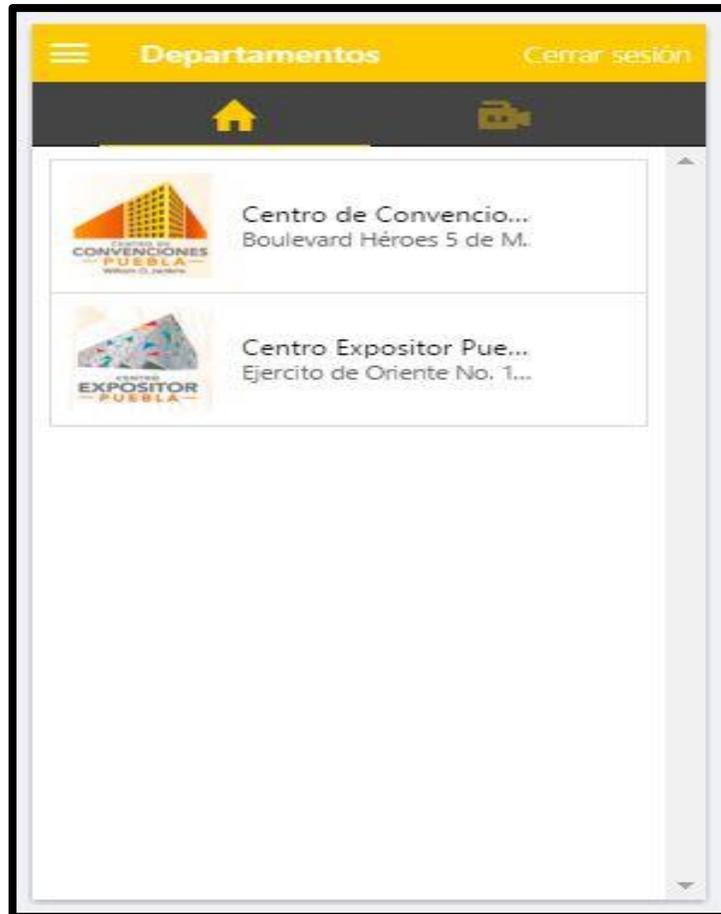


Figura 3.17 Menú Android

3.4.4 Cámaras Centro de Convenciones

Al seleccionar un departamento se puede visualizar en la barra lateral todos los DVR's a los que tiene acceso el usuario desde la base de datos. En este caso se visualizan los DVR's disponibles para el departamento de centro de convenciones William O. Jenkins. Cada DVR's a su vez está compuesto de varias cámaras.

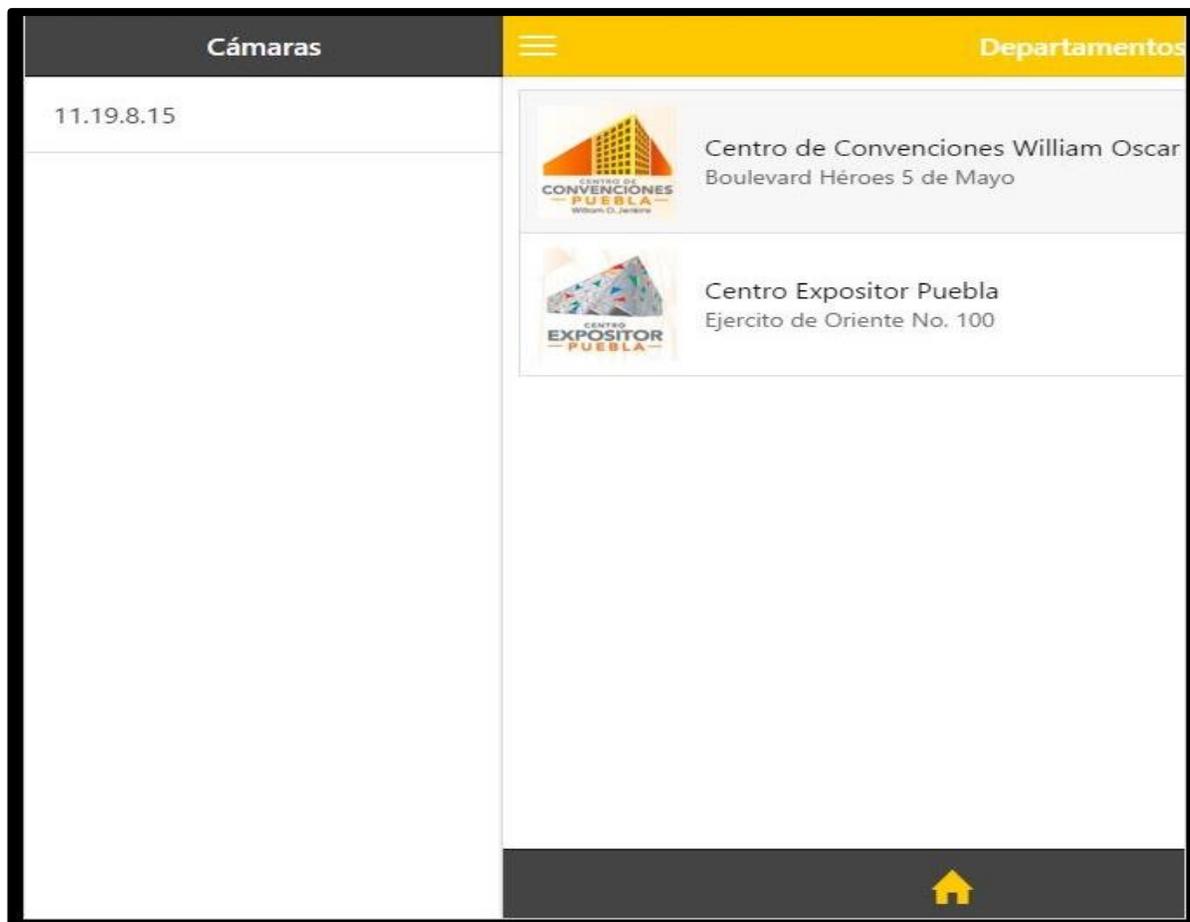


Figura 3.18 Cámaras Centro de Convenciones

3.4.5 Cámaras Centro Expositor

La figura 3.19 muestra todas las IP's de los DVR's disponibles para el usuario ubicadas en el Centro Expositor de Puebla.

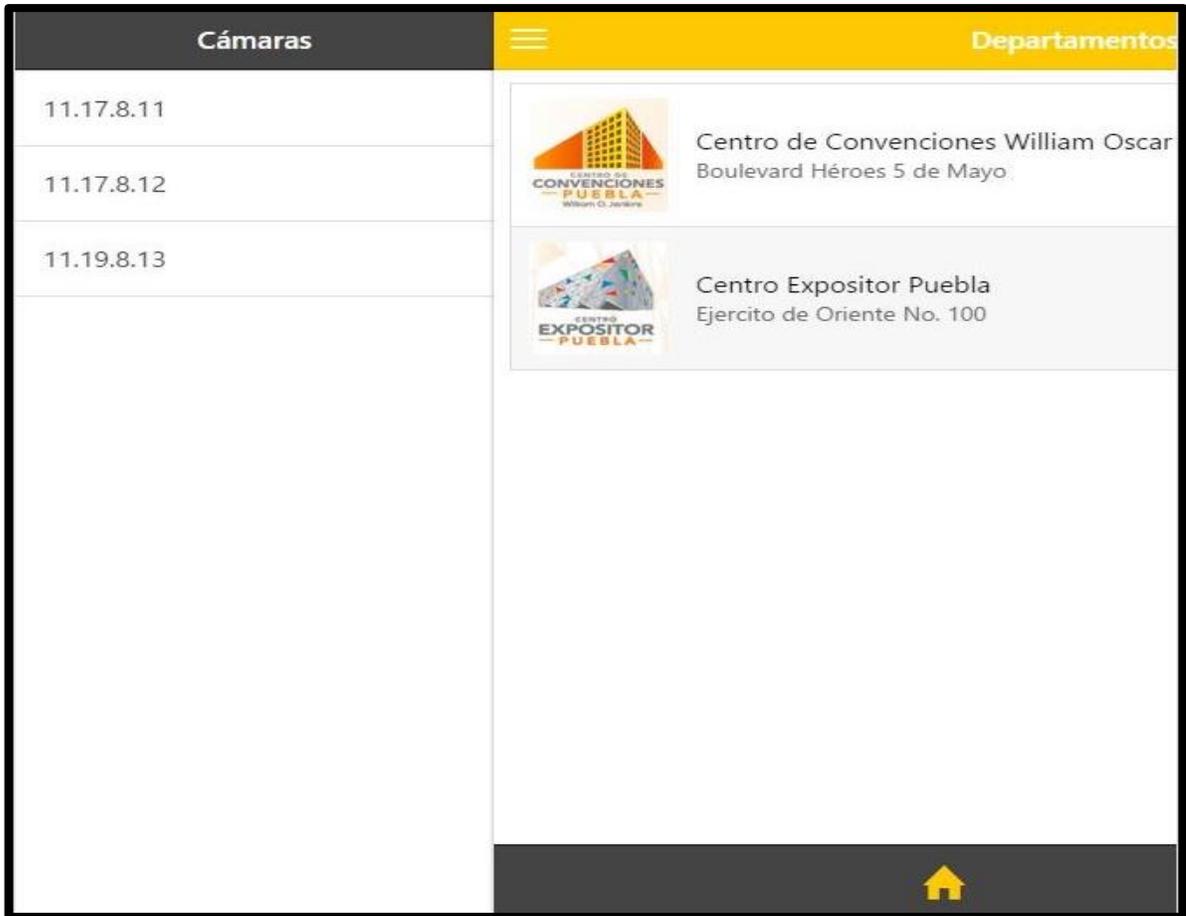


Figura 3.19 Cámaras Centro Expositor

3.4.6 Streaming

Al seleccionar un DVR la aplicación pasa a otra ventana llamada “Streaming” que visualiza y controla las cámaras, así como se puede observar en la figura.

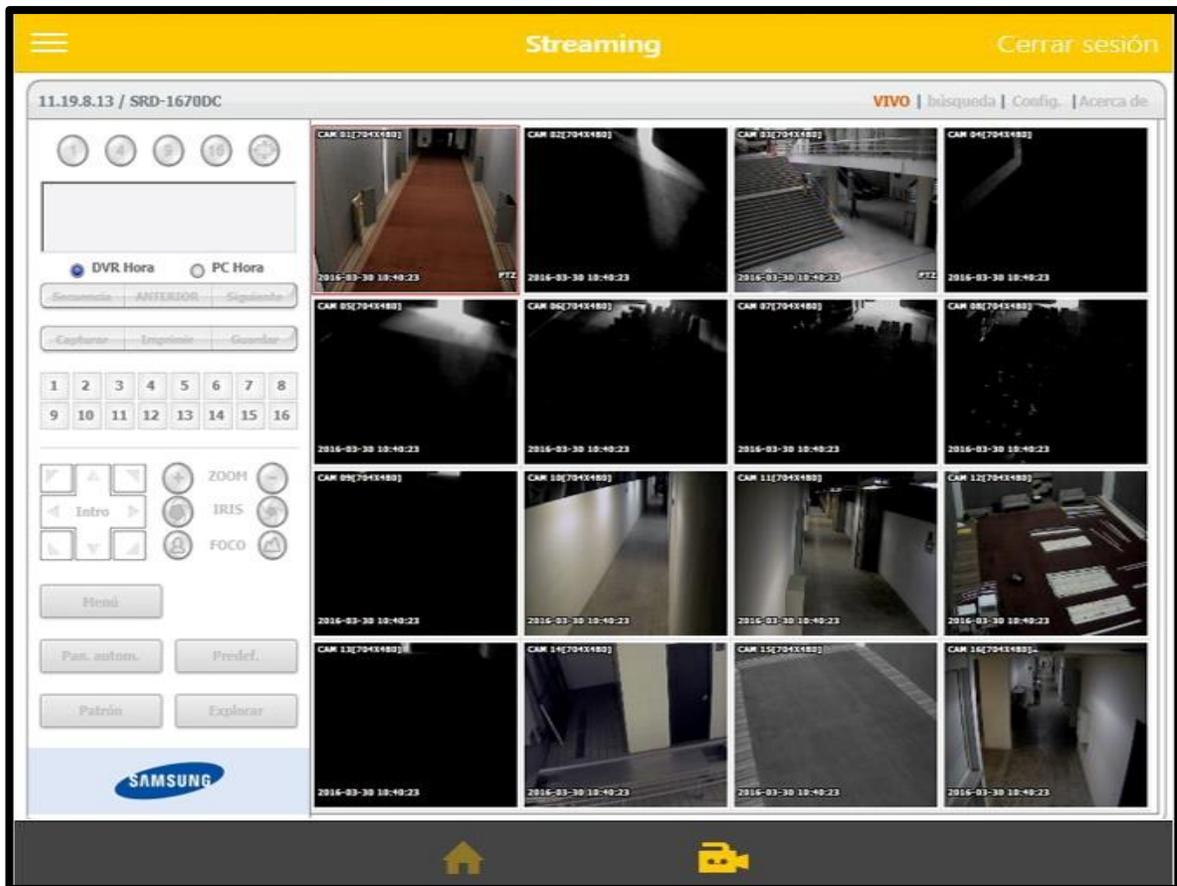


Figura 3.20 Streaming de un DVR

3.4.7 Código Fuente

Por cuestiones de la empresa sólo se muestran pequeños fragmentos de código de la aplicación ExpoCam.

El código que se muestra en la figura 3.21 realiza la función de cargar los templates(vistas) y controladores dependiendo del estado en el que se encuentre.

```

angular.module('starter', ['ionic', 'starter.controllers', 'starter.services'])
.run(function($ionicPlatform) {
  $ionicPlatform.ready(function() {
    if (window.cordova && window.cordova.plugins && window.cordova.plugins.Keyboard) {
      cordova.plugins.Keyboard.hideKeyboardAccessoryBar(true);
      cordova.plugins.Keyboard.disableScroll(true);
    }
    if (window.StatusBar) {
      StatusBar.styleDefault();
    }
  });
});

.config(function($stateProvider, $urlRouterProvider) {
  $stateProvider
    .state('login', {
      url: '/login',
      templateUrl: function($state){
        if(getSession('user').isActive){
          $state.go('sidebar.tab.depts');
        }else{
          return 'templates/login.html';
        }
      },
      controller: 'loginCtrl'
    })
    .state('sidebar', {
      abstract: true,
      templateUrl: function($state){
        if(getSession('user').isActive){
          return 'templates/sidebar.html';
        }else{
          $state.go('login');
        }
      },
      controller: 'sidebarCtrl'
    })
});

```

Figura 3.21 app.js

En la figura 3.22 se muestra un pequeño fragmento del archivo Controller.js, que como su nombre lo indica se inicializan todos los controladores.

```

angular.module('starter.controllers', [])
.controller('deptsCtrl', function($scope, deptService) {
  $scope.depts = deptService.all();
  $scope.onClick = function(objDept){
    $scope.$emit('listDvrs', objDept);
  }
});

.controller('loginCtrl', function($scope, $http, $state, $ionicPopup){
  $scope.loader = 'ng-hide';
  $scope.login = function(acc){
    if(validate(acc, 2, 6, 16)){ // validate method is in functions.js - parameters obj, objLength, minLength, maxLength
      $scope.loader = null;
      var url = "http://192.168.1.100:8080/CamService.asmx/";
      var params = {user: acc.user, pass: acc.pass};
      webService($http, url+'login', params, function(response){
        //this method is in functions.js
        if(response.isActive){
          setSession('user', response);
          serviceConstruct(response.idUser, $http, url+'query', $state);
        }else{
          popup($ionicPopup, 'Error de inicio de sesión', 'Por favor verifique su usuario y contraseña.');
```

Figura 3.22 Controller.js

4 Transición

4.1 Suite de pruebas de regresión

ID	PR001
Descripción	Se manda los parámetros de: <ul style="list-style-type: none">• Servo1=40• Servo2=160 a la dirección 192.168.4.1
Preparación	Prender el módulo ESP8266 y compilar el código proporcionado en la sección Código Arduino.
Instrucciones	<ol style="list-style-type: none">1. Entrar a la red proporcionada por el ESP8266.2. Ir a la dirección IP 192.168.4.13. Ingresar el valor de los parámetros servo1 y servo2 separados por comas
Resultados Esperados	El servomotor 1 debe girar 40° y el servomotor 2 160°.

ID	PR002
Descripción	La aplicación móvil debe mostrar los DVR's a los que tiene acceso el usuario y al seleccionarlo debe visualizar las cámaras que administra.
Preparación	Iniciar sesión en la aplicación móvil ExpoCam.
Instrucciones	<ol style="list-style-type: none">1. Seleccionar el departamento.2. Seleccionar un DVR.
Resultados Esperados	Al seleccionar un DVR en la lista de cámaras se deben visualizar varias cámaras que son administradas por el DVR.

4.2 Notas

- La versión actual de la aplicación Expocam es la versión 1.0.
- Se generó un APK para Android 4.0.3 o superior y un IPA para IOS 5 o superior.
- Las cuentas de usuario junto con la información a visualizar se insertan desde SQL Server 2008.

5 Metodología

5.1 AUP

Es un enfoque al desarrollo de software basado en el Rational Unified Process (RUP) de IBM. El ciclo de vida de Agile UP es serial en lo grande e iterativo en lo pequeño, liberando entregables incrementales en el tiempo.

5.2 Entregables Mínimos

Inicio	Elaboración	Construcción	Transición
Modelo de requerimientos	Modelo de diseño	Código fuente	Suite de pruebas de regresión
	Scripts de instalación		Documentación del sistema
			Sistema
			Notas

6 Evaluación de Resultados

Con la realización del proyecto se contemplan diferentes beneficios, la más importante a destacar es el aumento de seguridad en la video vigilancia ya que dicho proyecto permite al personal monitorear y controlar las cámaras desde un dispositivo móvil siempre y cuando el usuario se encuentre autenticado en una de varias redes proporcionadas por todo el recinto. Esto con el fin de evitar conexiones externas que puedan llegar a lucrar con la aplicación.

La aplicación también es administrable, es decir se puede agregar, modificar y eliminar el contenido que visualizan los usuarios, esto hace que la aplicación se adapte a futuros cambios que puedan surgir en la empresa.

7 Conclusiones

El desarrollo de los módulos de la aplicación móvil y los servomotores fue culminado con éxito, evaluando los requerimientos funcionales y los objetivos establecidos durante el inicio del proyecto. Este proyecto funge como una herramienta de alto valor dentro de la organización, permitiendo la satisfacción del personal por la sencillez y utilidad de la aplicación móvil.

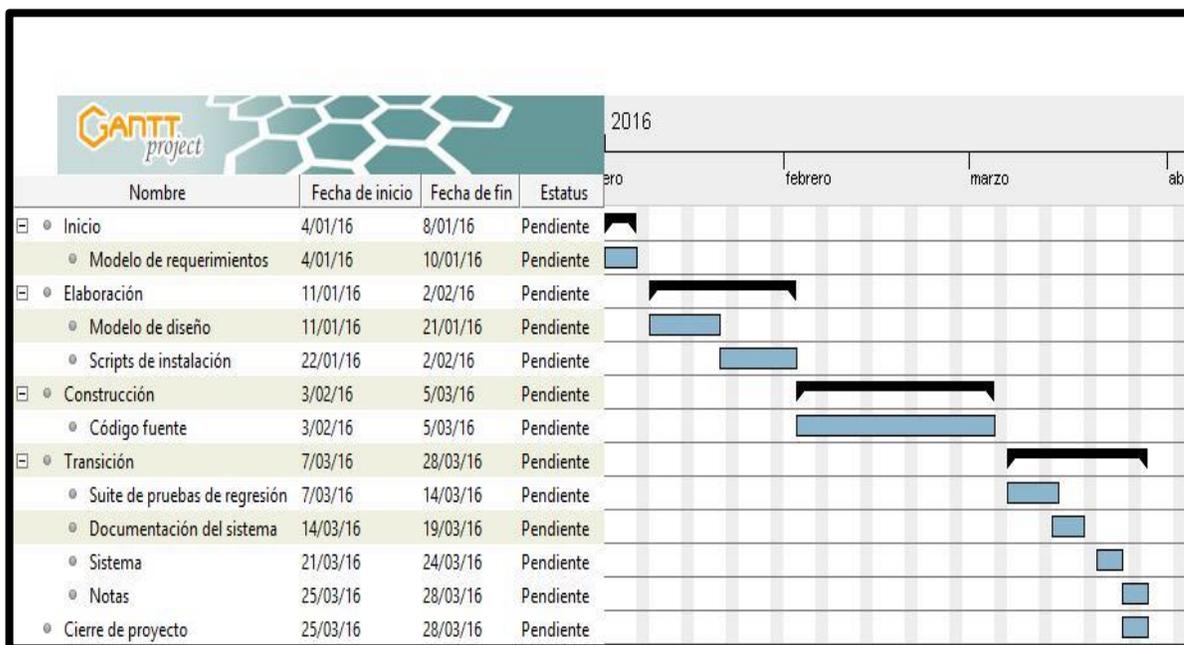
La aplicación móvil ayuda a ciertas personas que cuentan con el puesto y los privilegios, de poder monitorear y controlar las cámaras de video vigilancia, aumentando la seguridad de la empresa, ya que no se necesita estar en un departamento en específico para monitorear las cámaras del recinto.

El proyecto cumplió mis expectativas como profesional además de seguir explotando mis conocimientos técnicos y empíricos.

8 Referencias

- [1] «Prometec,» [En línea]. Available: <http://www.prometec.net/arduino-wifi/>.
- [2] «Pyroelectro,» [En línea]. Available: http://www.pyroelectro.com/tutorials/arduino_multi_servo/hardware.html.
- [3] «Instructables,» [En línea]. Available: <http://www.instructables.com/id/Arduino-Webserver-Control-Lights-Relays-Servos-etc/>.
- [4] «Random nerd tutorials,» [En línea]. Available: <http://randomnerdtutorials.com/arduino-webserver-with-an-arduino-ethernet-shield/>.
- [5] «Ionic framework,» [En línea]. Available: <http://ionicframework.com/docs/>.
- [6] «Sistemas orp,» [En línea]. Available: <http://www.sistemasorp.es/2014/11/11/programando-un-arduino-remotamente-con-el-modulo-esp8266/>.
- [7] «Arduino,» [En línea]. Available: <https://www.arduino.cc/en/Tutorial/UDPSendReceiveString>.
- [8] «Introducción al Agile UP,» [En línea]. Available: <http://www.cc.una.ac.cr/AUP/html/overview.html>.

9 Apéndices o Anexos



Material	Precios
Arduino uno (1)	360
Esp8266	125
Jumpers	60
Protoboard (1)	60
Servomotor (2)	180
Regulador de voltaje	25
Monto Total Máximo: 710	