

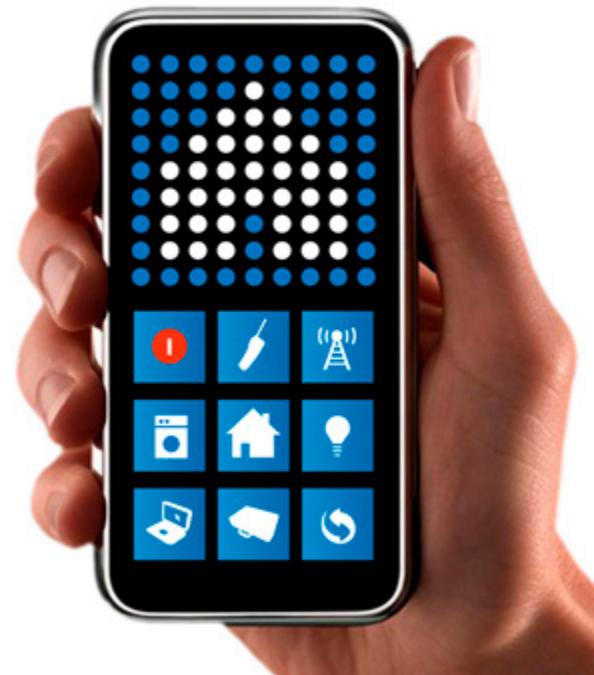
Sistema Inteligente de Seguridad y Administración Doméstica (SISAD)

Comisión

*Acosta, Gonzalo
García, Nicolás
Jasson, Federico*

Introducción

Se desarrolló un sistema domótico operable mediante un teléfono con Android, por Bluetooth.



Características funcionales

- ★ Autenticación de usuarios por contraseña.
- ★ Cerradura eléctrica con cierre automático luego de cierto tiempo.
- ★ Lámpara controlada por sensor de luz: enciende a falta de luz (si así se desea).
- ★ Medición de temperatura y humedad ambiental.
- ★ Control total del sistema mediante Android.

Usuarios

Roles:

- ❖ Usuario común (USER).
- ❖ Usuario administrador (ADMIN).

Todos los usuarios pueden abrir y cerrar la cerradura, activar y desactivar la lámpara, y solicitar la temperatura y la humedad. Además, pueden cambiar su contraseña en el sistema.



Usuarios

Existe un único usuario administrador, con nombre de usuario y contraseña por defecto (configuración de fábrica).

El usuario administrador tiene privilegios para agregar nuevos usuarios o eliminar usuarios ya existentes.



Comunicación

- ❖ La comunicación se lleva a cabo utilizando la tecnología Bluetooth.
- ❖ Por encima del protocolo Bluetooth se diseñó un protocolo específico para este sistema:
 - Modelo cliente-servidor.
 - Stateful (usuarios con sesión).

Cliente
(Android)



Servidor
(Arduino)



| LOGIN | username | password |

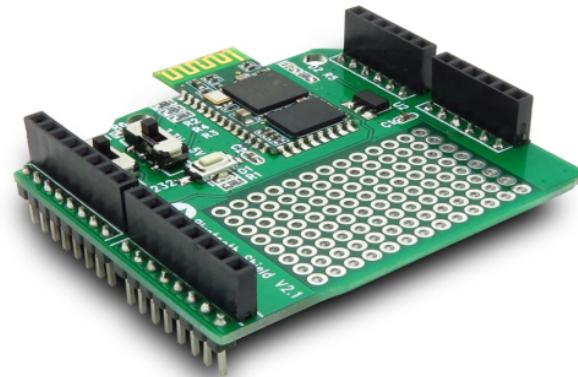
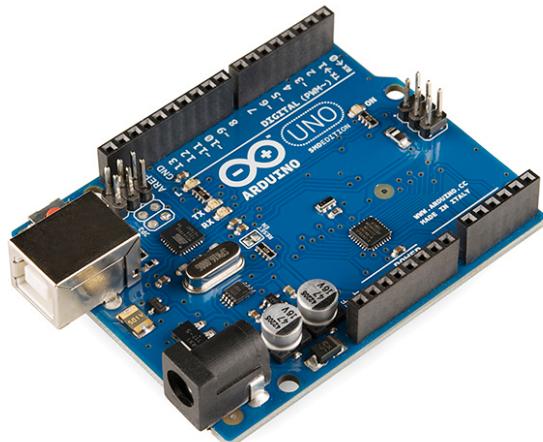
| <response> | <parameter_0> | ... |

| <request> | username | <parameter_0> | ... |

| <response> | <parameter_0> | ... |

Servidor

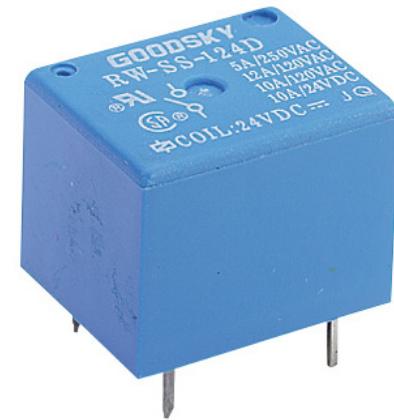
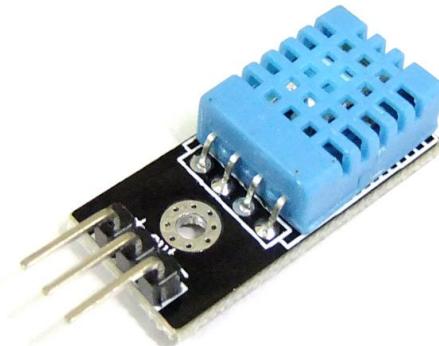
- ❖ Tecnologías:
 - Arduino Uno.
 - Shield Bluetooth ITead.
 - EEPROM del ATmega328P para almacenar la tabla de usuarios en memoria no volátil.
 - Puerto serie para el debugging.



Servidor

❖ Tecnologías:

- Sensor DHT11 de temperatura y humedad.
- Fotodiodo para medir la intensidad de la luz.
- Trabador eléctrico.
- Relé (5V) y lámpara.

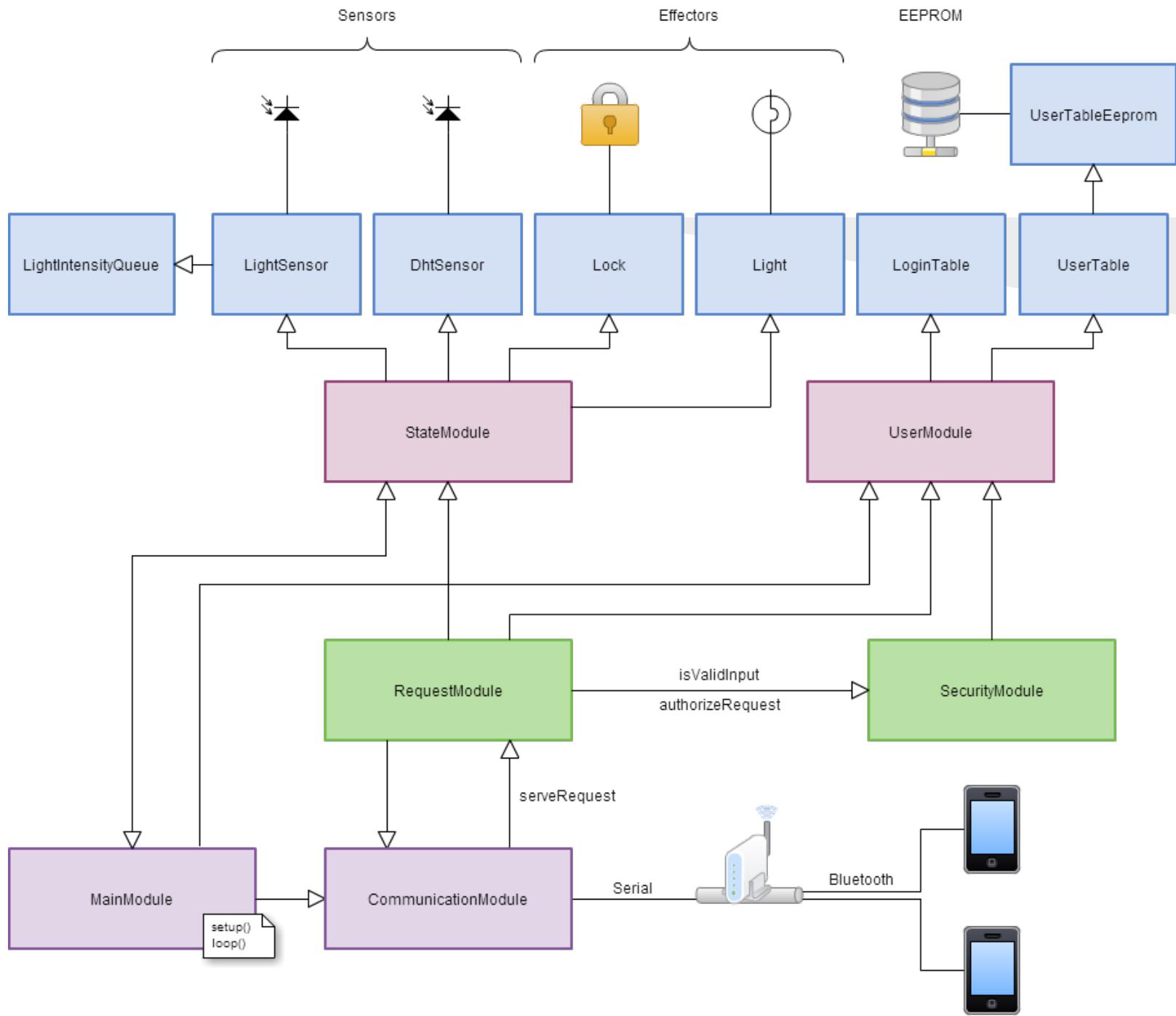


Servidor

- ❖ Responsabilidades:
 - Controla el estado general del sistema.
 - Obtiene datos de los sensores.
 - Acciona los dispositivos físicos.
 - Administra los usuarios del sistema y sus sesiones.
- ❖ Tareas periódicas:
 - Cerrado automático de la cerradura.
 - Medición de temperatura y humedad.
 - Medición de luz ambiental.
 - Disminución del TTL de las sesiones y auto-logout.

Servidor

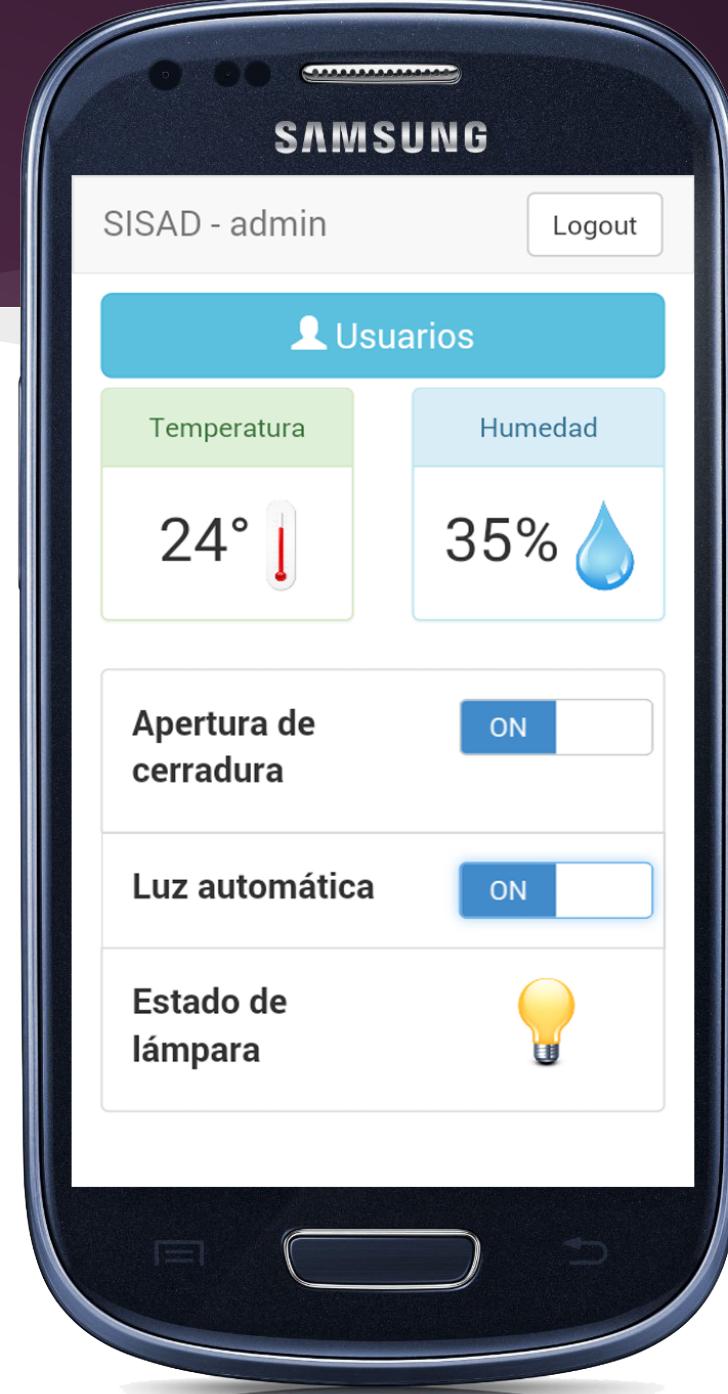
- ❖ Lleva un historial de las últimas mediciones de la luz, para poder distinguir una falta de luz real en el ambiente de una posible obstrucción temporal del sensor. Esto se logra calculando el promedio de estos valores, y apagando/encendiendo la luz cuando éste traspasa un cierto umbral.
- ❖ La EEPROM debe inicializarse en una etapa previa al release.



Cliente

Funcionalidades:

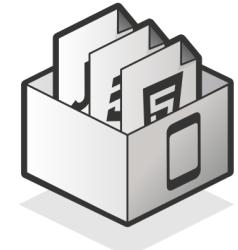
- ★ Iniciar sesión.
- ★ Administrar usuarios.
- ★ Abrir cerradura.
- ★ Activar luz inteligente.
- ★ Visualizar el estado de la luz.
- ★ Visualizar temperatura y humedad.





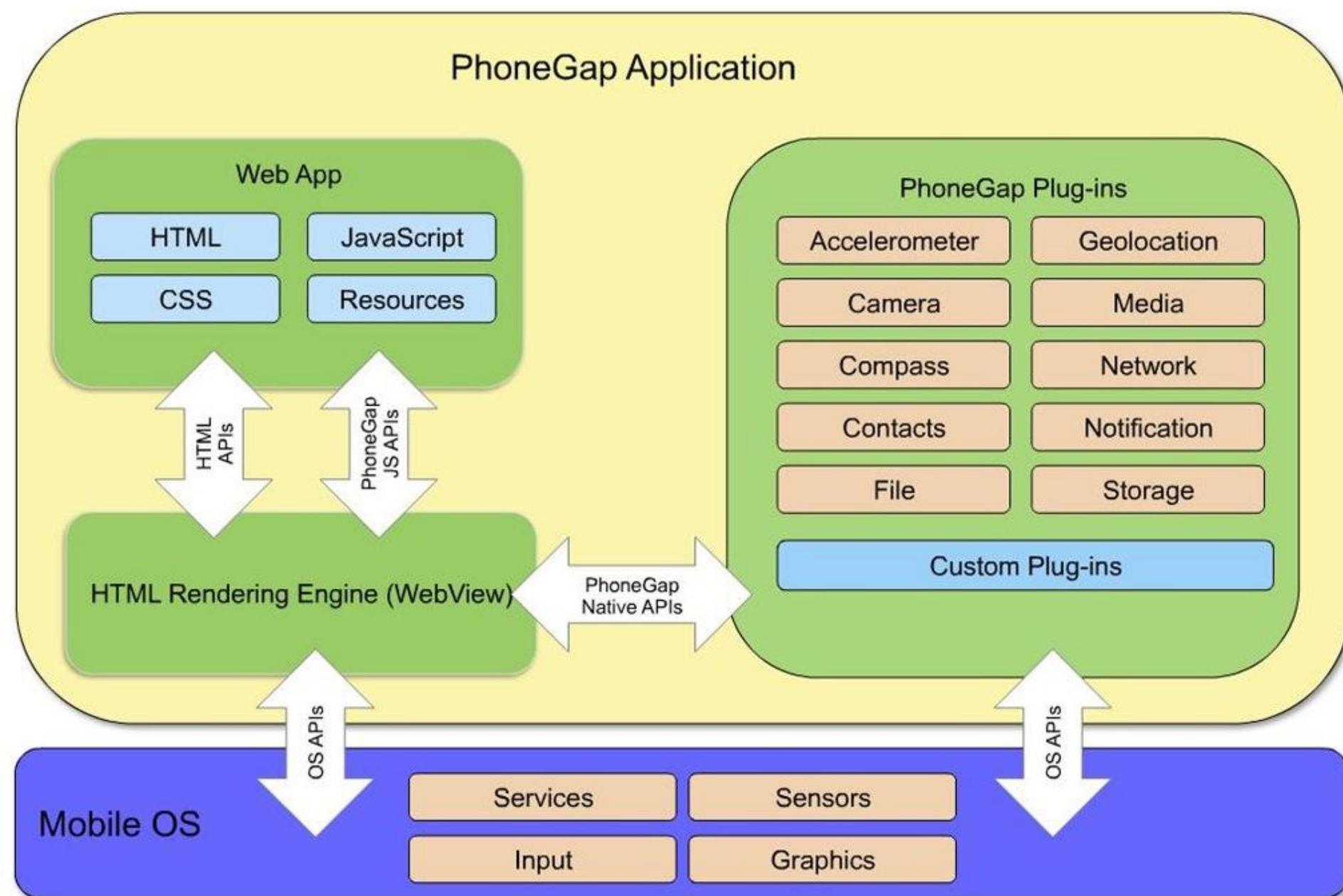
Cliente

- Diseñado para plataformas Android.
- Herramientas utilizadas:
 - Phonegap Framework.
 - Phonegap Developer App.
 - Phonegap Bluetooth Plugin.
 - Backbone.js, BB.LayoutManager.
 - Twitter Bootstrap.
 - Chrome Device Inspector.
 - Samsung Galaxy S3 mini, Android Kitkat 4.4.4.



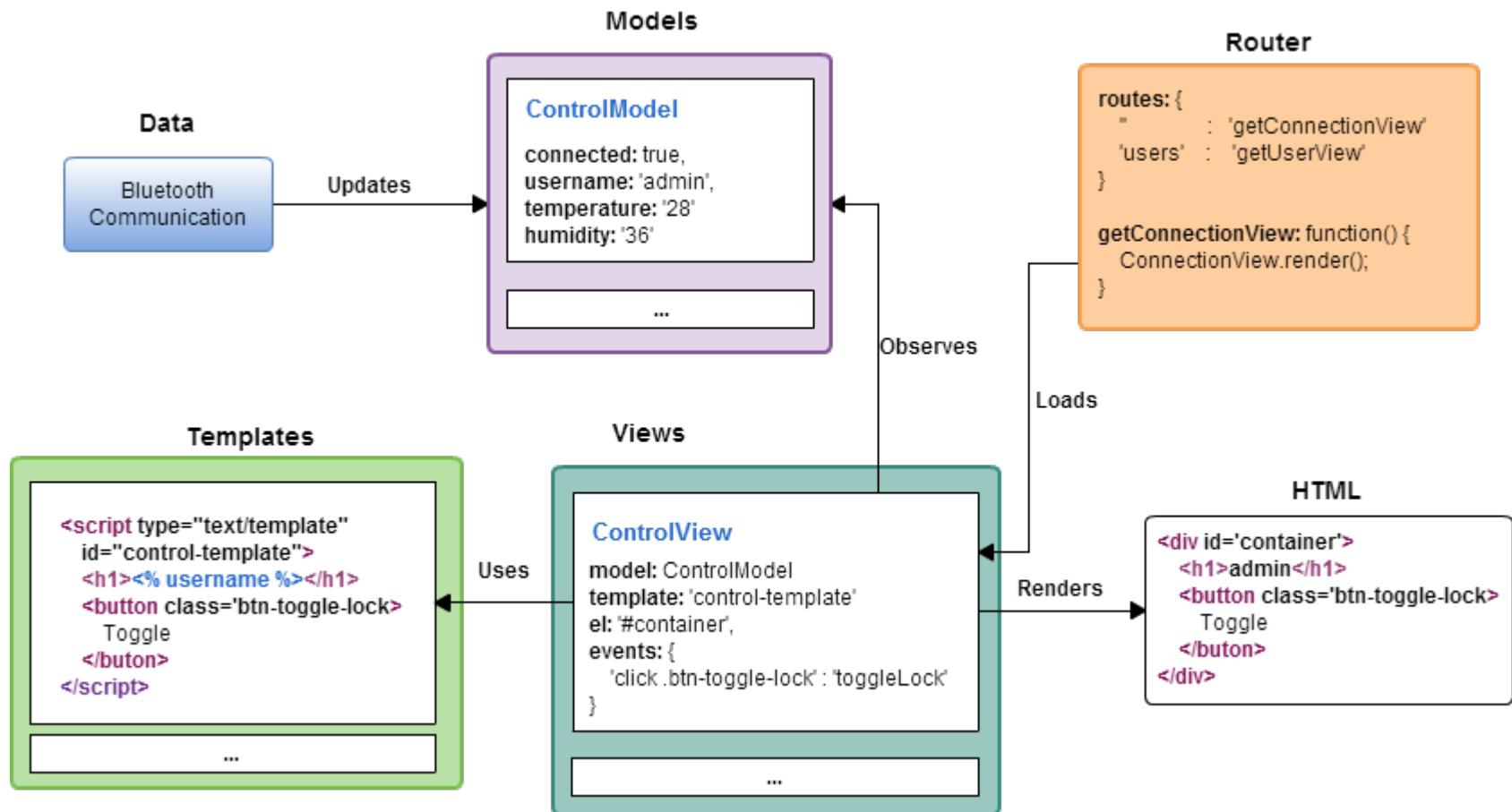
PhoneGap Architecture

PhoneGap Application





BACKBONE.JS





chrome

Developer Tools - file:///android_asset/www/index.html

Elements Network Sources Timeline Profiles Resources Audits Console

Sources Content scr... Snippets

file:///
 └── android_asset/www
 ├── bower_components
 ├── css
 └── js
 ├── controller
 └── model
 ├── bt-connection.js
 ├── bt-manager.js
 ├── communication.js
 └── connection-state.js
 └── view
 ├── connection-view.js
 ├── control-view.js
 ├── list-user-view.js
 ├── logger-view.js
 └── user-view.js
 └── bootstrap-switch.min.js
 └── main.js

// Initialize state refreshing interval
this.set({
 stateCallerId : setInterval(function() {
 Communication.requestState(username)
 }, STATE_REQUEST_INTERVAL)
});

Communication.requestState(username);

Router.navigate("control", true, true);
} else{
 Logger.log("Logged Out.");

 window.localStorage.removeItem('username');
 window.localStorage.removeItem('password');

 Communication.clearQueues();

 clearInterval(this.get('ttlCallerId'));
 clearInterval(this.get('stateCallerId'));

 if(this.get('connected'))
 Router.navigate('login', true, true);
 else
 Router.navigate('', true, true);
}
};

Watch Expressions

Communication.readBuffer: ""
data: <not available>
Communication.parsingFunction:
Communication.sendMessageQueue:
Communication.sendMessageQueue:
Call Stack

Scope Variables

Breakpoints

connection-state.js:51
loggedInChanged: function(){
DOM Breakpoints
XHR Breakpoints
Event Listener Breakpoints
Workers

Console Search Emulation Rendering

User and pass retrieved!
Sent: \$2#admin#12345*
Read: \$1#7*
Logged In.
Sent: \$9#admin*
Read: \$1*
Lock toggled Successfully

connection-state.js:27
logger-model.js:4
logger-model.js:4
logger-model.js:4
logger-model.js:4
logger-model.js:4
logger-model.js:4

Problemas en el desarrollo

- ❑ Problemas de memoria al utilizar la clase String (oficial). ¡NO UTILIZARLA!
- ❑ Problemas al emplear una versión desactualizada de la librería DHTLib (para el sensor de temperatura y humedad).
- ❑ Comportamientos errantes de la lámpara a causa de un transistor quemado.
- ❑ Problemas en la conexión desde Android utilizando Bluetooth.

Limitaciones

- ❖ La única forma de interactuar con el sistema es mediante la aplicación para Android, no hay una opción alternativa.
- ❖ Por limitaciones tecnológicas (y de tiempo), el sistema es vulnerable en cuanto a la seguridad:
 - Las contraseñas se almacenan en texto plano.
 - El protocolo es ingenuo, lo cual lo hace simple y vulnerable a la vez.

Limitaciones

- ❖ No se pudo conseguir una cerradura eléctrica, sino un destrabador eléctrico (como los usados en las entradas de los edificios).



Limitaciones

- ❖ El sensor de luz debe estar “aislado” de la lámpara para evitar que se prenda y apague repetidas veces.



- ❖ El sistema no acepta múltiples conexiones de clientes simultáneamente.

Extensión / Ideas adicionales

- ❖ Permitir que el administrador defina ciertos parámetros del sistema:
 - Tiempo de cierre automático de la cerradura (actualmente fijado en 5 segundos).
 - Tiempo de medición de la luz para calcular el promedio (actualmente fijado en 2 segundos).
- ❖ Incorporar una cerradura eléctrica.
- ❖ Poder controlar todas las luces de la casa.
- ❖ Incorporar una alarma al sistema.

Extensión / Ideas adicionales

- ❖ Incorporar un LCD para visualizar el estado del sistema.
- ❖ Control de calefacción a partir de la temperatura sensada.
- ❖ Comunicación por Internet en lugar de Bluetooth.
- ❖ Botón de reset para inicializar el sistema con las configuraciones de fábrica.

Fin

¿Preguntas?