



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**Facultad de Ingeniería en Electricidad y Computación**

**“SISTEMA DE INTEGRACIÓN PARA OBJETOS  
INTELIGENTES”**

**INFORME DE MATERIA INTEGRADORA**

Previo a la obtención del Título de:

**INGENIERO EN CIENCIAS COMPUTACIONALES**  
**ORIENTACIÓN SISTEMAS MULTIMEDIA**

**JAIME ADRIÁN AGUILAR ROMERO**

**KEVIN GUILLERMO CAMPUZANO CASTILLO**

**GUAYAQUIL – ECUADOR**

**AÑO: 2017**

## AGRADECIMIENTOS

Agradezco a Dios y mis padres, especialmente a mi padre por haberme proporcionado una buena educación y apoyo incondicional, enseñándome que con esfuerzo y constancia todo es posible aunque las circunstancias no sean buenas.

Agradezco a mis hermanas que con su apoyo y experiencias compartidas me han ayudado a seguir adelante en cada reto, ofreciéndome lo mejor en cada momento, te agradezco hermana Ángela pues gracias a ti pude concluir esta meta.

Agradezco mis amigos y compañeros por haber compartido alegrías y tristezas, contando con su apoyo siempre.

Mi más sincero agradecimiento al PhD. Domínguez por haberme ayudado y proporcionado elementos para que este proyecto esté finalizado.

Jaime Aguilar

Agradezco a Dios y a mis padres, por darme la oportunidad de vivir esta alocada aventura llamada vida y por estar conmigo en cada paso que doy, por fortalecer mi corazón en los tiempos en los que más necesitaba e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

A mis padres Guillermo Campuzano y Noemí Castillo, por apoyarme siempre ustedes son el pilar fundamental en todo lo que soy, en toda mi educación, tanto académica, como de la vida, por su incondicional apoyo perfectamente mantenido a través del tiempo.

A mis hermanas Leslie y Fiorella Campuzano, por apoyarme en todo momento, las quiero.

A mis amigos Pedro M. y Victor R. por compartir haberme dado la motivación cuando la necesitaba y el respaldo infaltable. A mi compañero de proyecto Adrián A., por haberme apoyado en todas las ocasiones en las que lo necesite sin que yo se lo pidiera.

Kevin Campuzano

## **DEDICATORIA**

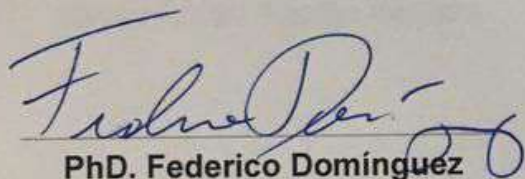
El presente proyecto le dedico a Dios, mis padres y mis hermanas, especialmente a mi hermana Ángela y mi padre Jaime pues su apoyo incondicional me ayudó a seguir adelante en cada reto, cultivando en mi deseos de superación. A mi pareja Fátima pues su apoyo y positivismo me ayudaban a seguir adelante en momentos difíciles.

Jaime Aguilar

El presente proyecto se lo dedico a Dios por darme tan grandiosa oportunidad y a mi familia, por todo el apoyo y consejos que me han brindado tiempo brindado. A mi pareja Elizabeth por el apoyo y comprensión que tuvo para seguir sosteniendo nuestra relación incluso en momentos difíciles.

Kevin Campuzano

## TRIBUNAL DE EVALUACIÓN



**PhD. Federico Domínguez**

PROFESOR EVALUADOR

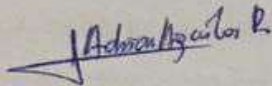


**Ing. Rafael Bonilla**

PROFESOR EVALUADOR

## DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Jaime Adrián Aguilar Romero



Kevin Guillermo Campuzano Castillo

## RESUMEN

En el ámbito de Internet de las cosas, un objeto inteligente es un objeto cotidiano como cafetera, lámpara, puerta de garaje, etc. conectados a inteligentes controlados por personas mediante su Smartphone. El usuario identifica a los objetos inteligentes mediante una etiqueta, etiqueta que tiene que ser única con la que pueda diferenciar e identificar al dispositivo.

En una situación en la que el usuario se encuentre en la necesidad de interactuar con el objeto inmediatamente, no es eficiente que el usuario tenga que recordar que identificador le asignó al dispositivo, además en el caso de que si el usuario quiere realizar acciones en cadena con los dispositivos que tiene a disposición, el usuario tendría que acceder a cada dispositivo y cambiar el estado de dichos objetos, este proceso es efectivo pero no es eficiente.

Al agregar un nuevo dispositivo al sistema intgIO, el usuario podrá crear un marcador fiduciario por medio de una herramienta externa al sistema, como por ejemplo código QR. Con este marcador el usuario ya no tendrá la necesidad de recordar el identificador que le asignó inicialmente al objeto inteligente, solo con escanear el marcador mediante la opción “Buscar Código QR” que proporciona intgIO, el sistema lo inserta automáticamente al tablero.

De una forma intuitiva, el usuario podrá conectar objetos cargados en el tablero, los mismos que poseen etiquetas visuales que facilita la identificación del tipo de objetos inteligentes con los que se está trabajando, así mismo, podrá realizar acciones en cadena, esto quiere decir, que al momento de modificar el estado de un objeto mediante intgIO o mediante manipulación directa a los objetos inteligentes, también se verán modificados los objetos que se encuentren conectados al mismo. Esta conexión es virtual y unidireccional entre objetos, para este proceso se tuvo que tipificar a los objetos.

Los usuarios del sistema intgIO quedaron satisfechos debido a que el uso de la aplicación es intuitiva, pudiendo ingresar objetos, crear relaciones y modificar sus estados mediante las funcionalidades de forma rápida y eficiente.

## ÍNDICE GENERAL

AGRADECIMIENTOS.....	ii
DEDICATORIA.....	iii
TRIBUNAL DE EVALUACIÓN.....	iv
DECLARACIÓN EXPRESA.....	v
RESUMEN.....	vi
CAPÍTULO 1.....	1
1. PLANTEAMIENTO DEL PROBLEMA.....	1
CAPÍTULO 2.....	2
2. ESTADO DEL ARTE.....	2
CAPÍTULO 3.....	5
3. DISEÑO E IMPLEMENTACIÓN.....	5
3.1 TECNOLOGÍAS UTILIZADAS.....	5
3.2 ENTIDADES.....	5
3.2.1 Dispositivo.....	5
3.2.2 Objeto.....	5
3.2.3 Acción.....	6
3.3 IMPLEMENTACIÓN.....	6
CAPÍTULO 4.....	12

4. DISCUSIÓN DE RESULTADOS.....	12
CONCLUSIONES Y RECOMENDACIONES.....	15
BIBLIOGRAFÍA.....	17
ANEXOS.....	18



# CAPÍTULO 1

## 1. PLANTEAMIENTO DEL PROBLEMA

La tecnología en estos tiempos avanza rápidamente junto con las necesidades de las personas que cada día son más exigentes. Exigentes por el hecho de que cuentan con menos tiempo para realizar las tareas cotidianas, como por ejemplo abrir la puerta del garaje al llegar a la casa, encender las luces, poner el estilo de música preferida y descansar. Pero esto toma tiempo, tiempo que puede ser reducido en estos días gracias a la tecnología. Esta es la era del Internet de las Cosas, en la que objetos cotidianos tales como lámparas, televisores, cámaras de seguridad y puertas de garaje, todos ellos con circuitos integrados se encuentran conectados a internet, en donde la persona puede controlarlos a todos con un solo dispositivo de mando.

Para la necesidad descrita anteriormente se desarrolló la aplicación web *IntgIO* la cual permite al usuario ingresar uno o varios objetos inteligentes, poder manipularlos directamente desde la aplicación remotamente, sin tener ningún contacto físico y poder realizar conexiones de forma virtual entre los mismos, estas conexiones no solo serán válidas en la aplicación web, sino que también simularán conexiones entre dispositivos, es decir que si un objeto cambia su estado actual mediante la manipulación física, todos los objetos conectados a él cambiarán automáticamente, ahorrando así tiempo y esfuerzo físico.

Al momento de que el usuario posea muchos objetos inteligentes y desee interactuar inmediatamente con ellos, se encontrará con un gran problema, al ser humano se le dificulta memorizar identificadores, mayor número de identificadores mayor dificultad de memorización, lo cual resulta frustrante que al momento de querer interactuar con un objeto inteligente, no pueda diferenciar o recordar su identificador. *IntgIO* soluciona este problema mediante el uso de marcadores fiduciaros para identificar los objetos debido a que su uso es fácil, sencillo, se ha popularizado en los últimos años y las personas están familiarizadas con los mismos, los cuales se explican más adelante.

## CAPÍTULO 2

### 2. ESTADO DEL ARTE

La realidad aumentada (AR) busca expandir su utilidad la cual es muy aplicable en cualquier sector debido a su facilidad de uso, como por ejemplo la manipulación de objetos. AR utiliza dos interfaces que son de vital importancia, la interfaz tangible y la interfaz gráfica de usuario. La interfaz gráfica de usuario es la representación de la información del sistema por medio de figuras y texto, la interfaz tangible es la que mediante lectura de gestos puede realizar acciones que vayan de acuerdo al mismo.

Estudios demuestran que el usuario tiene que seguir por tres importantes niveles para entender las ventajas que tiene la realidad aumentada.

El primer nivel es entender su significado y funcionalidad, cuando el usuario usa cualquier objeto por primera vez se encuentra con estas inquietudes que las descubre con el uso, este paso resolverá preguntas como: ¿Qué es? y ¿Para qué sirve?

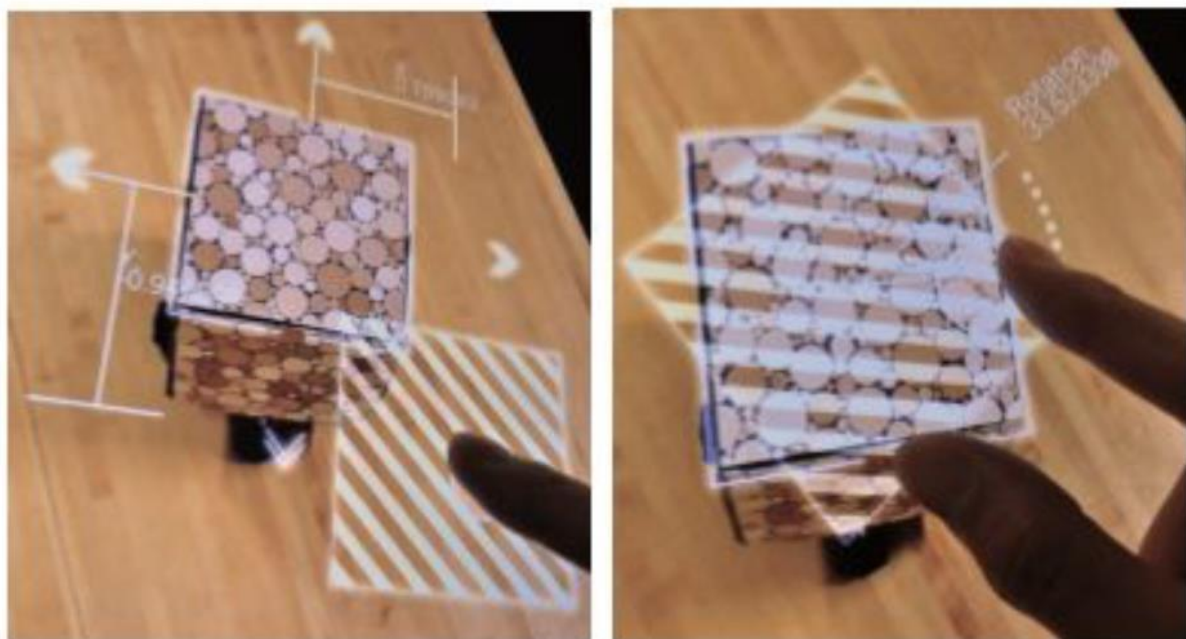
El segundo nivel es programar su configuración tanto a como lo desee el usuario. Esta fase ayuda al usuario a hallar el límite de la funcionalidad del objeto, en este paso se produce una gran interacción visual con el usuario pero también representa una pequeña parte de toda la interacción con el objeto.

El tercer y último paso representa la mayor parte de la interacción entre el usuario y el objeto, es la operación o interacción que se tiene con él. Este paso necesita la atención visual mínima con respecto a la acción y la máxima con respecto al objeto ejecutándola. Ha sido demostrado que imágenes en tiempo real con interacción aumentada en la pantalla proveen una interfaz más intuitiva que las de los sistemas abstractos en los que limitan la acción mediante un componente primitivo [4] , esta es una gran utilidad de sistemas de retroalimentación en tiempo real a través de interfaces tangibles [5] .

ExTouch [7] es un sistema para el control de movimiento de objetos en función del accionamiento espacial, es decir el usuario puede controlar los objetos arrastrando

los gráficos superpuestos en ellos con el gesto táctil en la pantalla, observando los resultados en tiempo real.

Para la implementación de Extouch se utilizó la tecnología de reconocimiento de realidad aumentada basada en imágenes (Unity con la extensión VuforiaTM, que es el encargado de la parte de realidad aumentada) que reconoce una imagen como un plano con objeto 3D obtenido del diccionario de datos registrados, accionando los objetos conectados a través la red inalámbrica.



**Figura 2.1. Muestra las distintas matrices de posición. Izquierda: Traslación en el plano. Derecha: Rotación en el plano [1].**

Para alcanzar la posición deseada y la orientación designada por el usuario, el sistema realiza un cálculo de diferencia de matrices. La posición diferencia o matriz diferencia se calcula a partir de la posición actual del objeto y la posición destino. La posición diferencia se asigna en el control de mando y es enviado al objeto seleccionado, logrando así el movimiento como podemos observar en la Figura 2.1.

Para la implementación de un simple reproductor de música y radio se trabajó el TUI (Interfaz de usuario tangible) y GUI (Interfaz de usuario gráfica), este proyecto consiste en manipular un reproductor música con solo dos tuercas, una controlaba

las estaciones de radio y la otra el volumen del sonido, esto lo realizaba mediante la interfaz tangible una vez que el usuario terminaba el uso en esta interfaz podía utilizar la Interfaz gráfica para un mejor entendimiento y uso alternativo para escoger canciones, reproducir listas de canciones, coger y soltar canciones en una lista a reproducirse. Todos los cambios en TUI influyen en GUI y viceversa.

Cualquier acción que actuaba en una de las interfaces un mensaje era enviado al servidor. El servidor luego enviaba a la otra interfaz en orden para coordinar sus estados. Para la implementación de la interfaz de usuario en realidad aumentada se utilizó el framework Qualcomm Vuforia que puede usar cualquier imagen como un marcador y proyectar una escena 3D encima de él. El dispositivo touch se conecta al server via WIFI y el software usa el protocolo Open Sound Control para la comunicación con el servidor. La representación física del objeto inteligente usa WiFly RN -131 y un microprocesador Atmega32U4 que envía la información de TUI mediante el protocolo OSC hacia el servidor. Como todo elemento básico de TUI, se compone de sliders, push buttons y rotations knobs, el sistema es tan flexible que cualquier objeto con un componente electrónico puede ser parte de la interacción.

OpenHybrid [3] es una plataforma que ayuda a interactuar con cualquier objeto, combinando objetos físicos con los beneficios de una interfaz de usuario flexible, la cual permite:

- Crear contenido de Realidad Aumentada con herramientas HTML.
- Crear la Realidad Aumentada sin ningún conocimiento sobre programación 3D.
- Conectar la funcionalidad de los objetos con un simple arrastrar y soltar.
- Programar el Objeto híbrido físico y conectarlo a la interfaz AR con Arduino.

## CAPÍTULO 3

### 3. DISEÑO E IMPLEMENTACIÓN

#### 3.1 Tecnologías utilizadas

IntgIO es un Sistema de Integración para Objetos Inteligentes, el cual está alojado en un servidor node.js mediante la plataforma Node-red, interfaz de programación de aplicaciones (API) de IOT, la cual se especializa en los protocolos de comunicación como MQTT y HTTP para la comunicación con objetos inteligentes.

Para el guardado de los objetos inteligentes ingresados en intgIO se utiliza la base de datos no relacional Neo4J debido a que proporciona almacenamiento de datos estructurados en grafos, obteniendo así una base de datos flexible con posibilidad de datos altamente interrelacionados (lo cual es indispensable en la aplicación).

Para el manejo gráfico del tablero, utiliza la librería open source *Vis.js* para redes personalizadas basada en Javascript, la cual nos provee una interfaz gráfica en forma de grafos, esta librería fue modificada ligeramente para adaptarse a la aplicación.

Para el manejo de la cámara y escaneo QR en tiempo real, utiliza la librería open source *Instascan*.

#### 3.2 Entidades

IntgIO, se basa en tres estructuras de datos, las cuales se encuentran interconectadas para su correcto funcionamiento y son las siguientes.

##### 3.2.1 Dispositivo

Está representado por el nodo de color rojo y contiene las características del objeto inteligente como puede ser: modelo, tipo, código, tipo de comunicación (MQTT o HTTP), puerto http en el que escucha el objeto (si es comunicación HTTP), tópico al que está suscrito (si es comunicación MQTT), el Dispositivo puede poseer uno o varios Objetos.

##### 3.2.2 Objeto

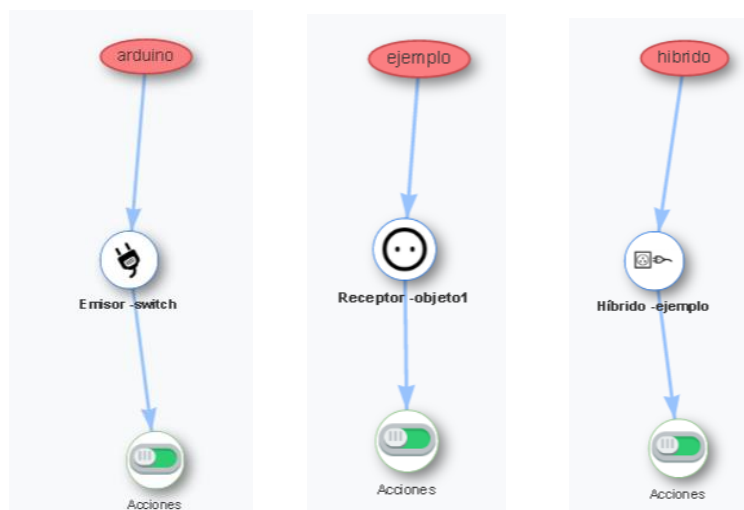
Este nodo puede ser representado de tres formas, la primera con una imagen de un enchufe macho y una etiqueta "*emisor-nombre*", la segunda

mediante una imagen de un enchufe hembra y etiqueta “*receptor-nombre*” y la tercera mediante una imagen de enchufe hembra-macho y la etiqueta “*Híbrido-nombre*”, se representa de esa manera para facilitar al usuario la realización de conexiones entre objetos. Este nodo contiene el factor de comunicación el cual puede ser emisor o receptor, también existe un caso híbrido para objetos que pueden ser emisor y receptor a la vez, el Objeto puede poseer una o varias Acciones.

### 3.2.3 Acción

Este nodo está representado por la imagen de un slider y contiene el estado actual del objeto, la cual puede ser de tipo incremental (0,1,2...,160) o tipo binario(ON/OFF). Este nodo es el único con evento, el cual permite cambiar de estado al dispositivo tanto en la base de datos como en el dispositivo real.

Podemos observar gráficamente cómo se representan estos objetos inteligentes en la Figura 3.2.



**Figura 3.2. Estructura de un objeto inteligente**

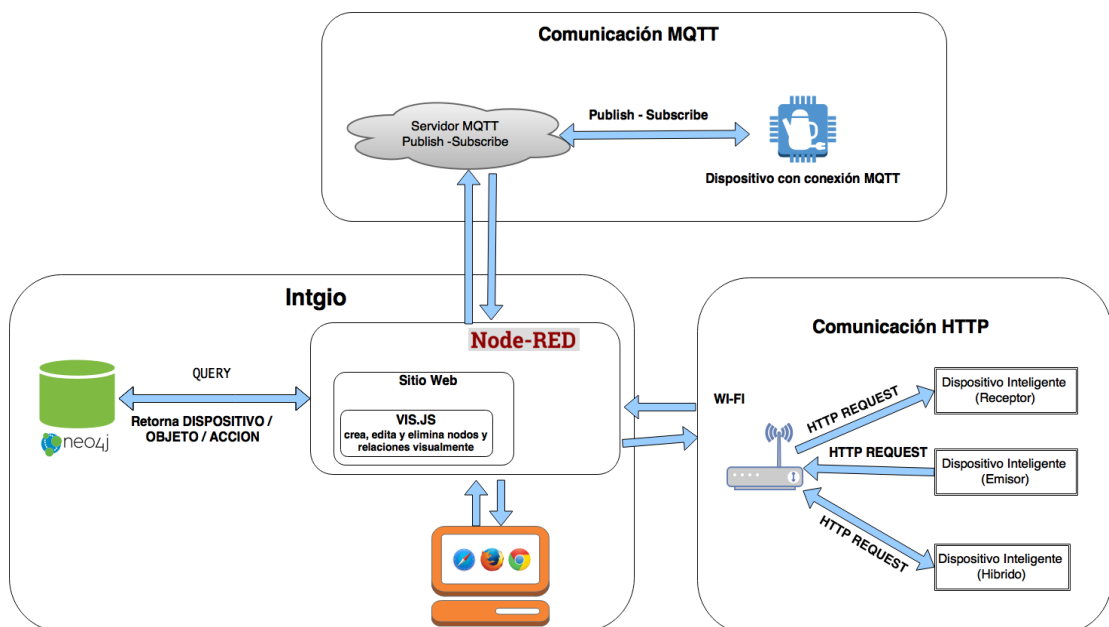
### 3.3 Implementación

La comunicación con el servidor Node-Red se realiza mediante peticiones ajax [6], estas peticiones ejecutan la creación, actualización y eliminación de

dispositivos, objetos y acciones de los objetos inteligentes (dispositivos físicos inteligentes conectados a la red) que se encuentren alojados en la base de datos, de la misma forma se crean o eliminan relaciones entre objetos de diferentes dispositivos.

Al momento de crear, consultar, cambiar estado actual o eliminar un objeto inteligente, Node-Red envía la sentencia a la base Neo4J mediante peticiones HTTP antes de ejecutar cualquier acción sobre el objeto inteligente, el mismo que debe encontrarse registrado y alojado en la red, caso contrario no se reflejarán las acciones a realizar en el mismo.

Este flujo de acciones lo podemos observar en la Figura 3.3.



**Figura 3.3. Arquitectura de intgIO (Sistema de Objetos Inteligentes)**

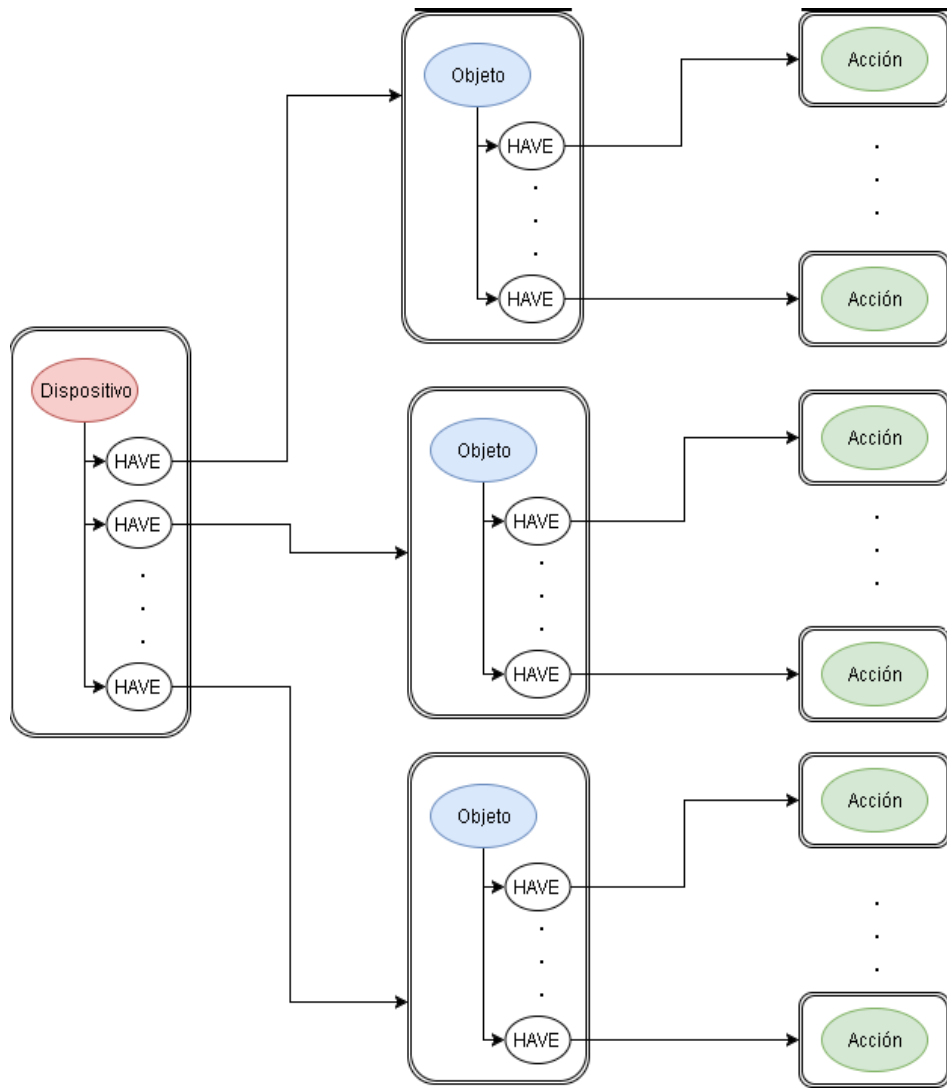
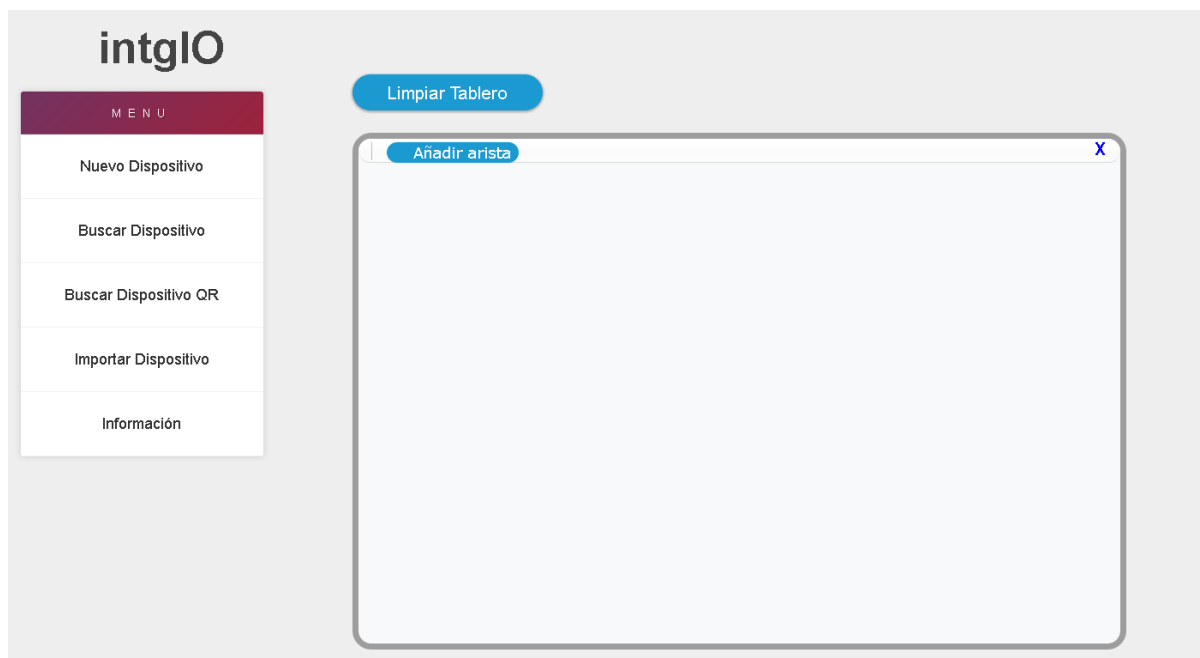


Figura 3.4. Representación de datos en la base Neo4j



La representación de los datos que se registran en Neo4j se muestra de una forma muy similar al tablero de intgLO, lo cual facilita el entendimiento y verificación de los



datos, un nodo de tipo Dispositivo puede tener de cero a muchos Objetos, el nodo de tipo Objeto puede tener de cero a muchas Acciones, el nodo de tipo Acción no posee nodos como lo podemos ver en la Figura 3.4.

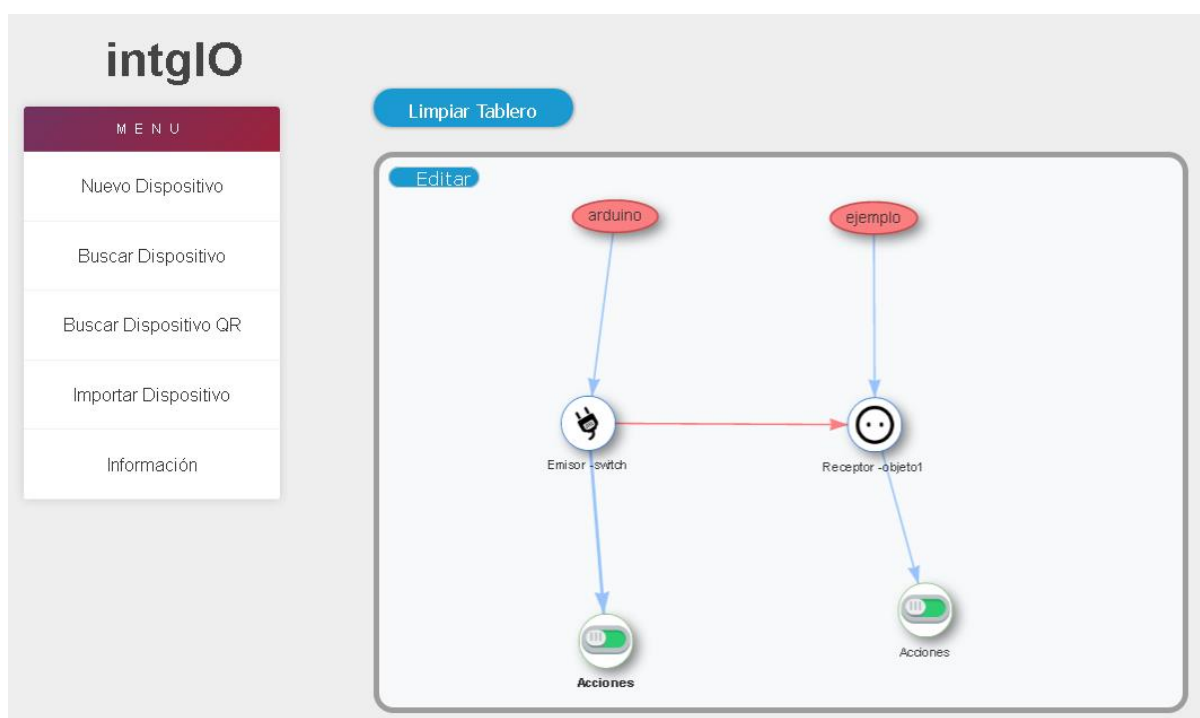
**Figura 3.5. Aplicación web intgLO.**

IntgLO cuenta con varias opciones, las cuales se encuentran en un menú lateral y permiten:

- **Nuevo Dispositivo.**- Crear un dispositivo con sus respectivos objetos y acciones en la base de datos.
- **Buscar Dispositivo.**- Buscar un dispositivo mediante su código para poder ser modificado o agregado al tablero gráfico, también se puede modificar los objetos y acciones relacionadas al dispositivo.
- **Buscar Dispositivo QR.**- Cargar dispositivo al tablero gráfico mediante la cámara y código QR que contiene el código del dispositivo a buscar.

- **Importar Dispositivo.**- Importar un dispositivo con sus respectivos objetos y acciones mediante el ingreso del código del dispositivo y una base de datos externa a la aplicación web (en este caso se utiliza un archivo con formato JSON que contiene varios dispositivos no ingresados en la base de datos de intgIO).
- **Información.**- Información de intgIO y sus creadores.

También posee la opción de “Limpiar Tablero” la cual permite borrar del tablero todos los dispositivos cargados y las opciones para añadir o eliminar aristas o conexiones de dispositivos cargados en el tablero. Además podemos eliminar dispositivos, objetos o acciones permanentemente de la base de datos mediante el tablero una vez seleccionado el elemento que se desea eliminar. Como resultado podemos observar la aplicación web en la Figura 3.5.



**Figura 3.6. Objeto inteligente (arduino) con conexiones virtuales con otro objeto de tipo Receptor.**

Al momento de interconectar objetos como se muestra en la Figura 3.6. Hay que tener en cuenta a qué tipo pertenece ya que existen tres tipos: Emisor, Receptor e Híbrido. El objeto de tipo receptor podrá recibir señales de otro u otros Objetos, el de tipo emisor podrá emitir señal hacia otro u otros objetos y el objeto de tipo híbrido podrá soportar las dos acciones anteriormente explicadas. Llámese señal a cualquier tipo de valor que se envíe o reciba el objeto mediante peticiones GET y este pueda interpretarlo en su manera correspondiente. También hay que tomar en cuenta la dirección de la relación, ya que son unidireccionales, sólo es posible crear la relación en los siguientes casos:

- Emisor  $\Rightarrow$  Receptor
- Emisor  $\Rightarrow$  Híbrido
- Híbrido  $\Rightarrow$  Receptor
- Híbrido  $\Rightarrow$  Híbrido

No es posible crear relaciones entre objetos del mismo tipo a excepción de los híbridos.

Para interactuar con objetos, Node-Red tiene varios protocolos de comunicación, al momento el sistema IntgIO cuenta con: HTTP y MQTT.

Para el protocolo de comunicación HTTP, intgIO envía peticiones GET al dispositivo inteligente receptor, este dispositivo retorna su respuesta, sea esta exitosa o fallida, y en base a aquella respuesta se actualiza el estado de dicho objeto en la base Neo4J.

Para el protocolo de comunicación MQTT, intgIO se suscribe y publica en un tópico determinado el estado que desea cambiar al dispositivo inteligente, el mismo que se encuentra suscrito y escuchando en el servidor MQTT, en el momento en que retorna la respuesta exitosa, se actualiza el estado del dispositivo en Neo4J.

## CAPÍTULO 4

### 4. DISCUSIÓN DE RESULTADOS

El uso de la aplicación intgIO facilita la manipulación de dispositivos de manera remota, así mismo, cambiar el estado actual del dispositivo rápidamente (ejemplo: el usuario dejó prendidas las luces de la casa, mediante intgIO puede cambiar el estado de las mismas y apagarlas con sólo conectarse a internet). El uso de intgIO puede ser esencial al momento de ahorrar tiempo y esfuerzo al usuario.

IntgIO logró la comunicación con dispositivos inteligentes y la interacción entre ellos de manera exitosa, además dio paso a la relación dispositivo padre - hijo, la cual consiste en que un dispositivo es padre o de mando para ejecución de otros dispositivos, el dispositivo padre tiene en objeto de tipo emisor mientras que el o los dispositivos hijos que están conectados virtualmente a él son de tipo receptores. La acción generada en el dispositivo padre generará una reacción en cadena a los dispositivos hijos. Un ejemplo de conexiones lo podemos observar en la Figura 4.7.



**Figura 4.7. Dispositivo inteligente (Photon conectado a switch de poder, a una lámpara y un servo motor).**

Al sistema se le añadió funcionalidades adicionales para una mejor experiencia de usuario, de tal manera de que al usuario se le diera la facilidad de la creación y manipulación de dispositivos, objetos, relaciones y ejecución de acciones de los dispositivos.

Para la carga de un dispositivo al tablero se añadió la funcionalidad de "Buscar Dispositivo QR" la cual mediante la lectura de códigos QR se puede añadir un dispositivo al tablero e interactuar con él fácilmente como se muestra en la Figura 4.8.



**Figura 4.8. Usuario utilizando funcionalidad de Agregación de Dispositivo mediante código QR.**

Se realizaron varias pruebas de usuarios como podemos observar en la Figura 4.9 y Figura 4.10, en la primera prueba el sistema fracasó, no fue por fallas en el sistema si no debido a la falta de experiencia de usuario al tener una interfaz poco intuitiva, el usuario no tenía conocimiento de que acciones podría realizar y cómo realizarlas. Se realizaron los cambios respectivos con la retroalimentación obtenida, la última prueba de usuario fue mejor de que se esperaba, la introducción al sistema fue suficiente para que el usuario entienda que puede realizar y qué acciones podría realizar en el sistema, se tuvo retroalimentación pero no fue con problemas tan severos como en las anteriores a esta. La interfaz de usuario del sistema intgIO es

el resultado de arreglos a las retroalimentaciones obtenidas y toma de decisiones para un mejor diseño y experiencia.



**Figura 4.9. Usuario final realizando pruebas 1. Usuario encendiendo una lámpara desde el sistema web en una tablet.**



**Figura 4.10. Usuario final realizando pruebas 2. Usuario encendiendo varias lámparas mediante un switch, realizada la relación en el sistema web.**

## CONCLUSIONES Y RECOMENDACIONES

El sistema de integración para objetos inteligentes cumple con las expectativas para ser un prototipo, al interactuar con objetos inteligentes y poder relacionarlos entre sí. Una mejora para el sistema intgIO es implementar los demás protocolos de comunicación como son: ftp y serial (COM). Anteriormente explicamos que el sistema se encuentra utilizando los protocolos MQTT y HTTP. En este último protocolo todas las peticiones son bajo el método GET, pero para obtener una mayor seguridad en nuestro sistema sería mejor utilizar método POST.

Otra mejora para el sistema es realizarlo completamente con Realidad aumentada, el framework recomendado a utilizar es Vuforia(Unity) que tiene gran soporte y documentación para su desarrollo. Al tener el sistema implementado con realidad aumentada podrá tener una mejor experiencia de usuario de la que ya cuenta.

Una mejora con respecto al tiempo de comunicación entre los dispositivos y el sistema es que la forma en que se envían los datos sea multicast, por ejemplo si existe un objeto inteligente con más de una conexión virtual actualmente se demora un segundo por cada objeto conectado y las acciones se ejecutan sucesivamente, esto ocurre debido a que se están enviando las peticiones de forma unicast mediante una pila, al momento de ser enviados de forma multicast el tiempo de comunicación será casi inmediato en todos los dispositivos a la vez.

Un punto crítico a tomar es con respecto a la seguridad del sitio web, es importante utilizar la encriptación de datos como por ejemplo el uso de https, con lo cual se protege la integridad de los datos enviados mediante las peticiones, además que facilitará el uso de la cámara en los dispositivos.

Para tener mayor control en el sistema es recomendable que posea un login al ingreso de la aplicación, o al menos una clave de confirmación al momento de cargar un objeto inteligente o cambiar de estado a los dispositivos. Esta versión fue trabajada de forma local y por ser un prototipo no fue crítico este punto, en futuros trabajos implementando acceso global desde internet es necesaria esta seguridad, así mismo deberá tener su sección administrativa para el control de usuarios.

El uso de esta tecnología tiene sus desventajas, debido a que estos objetos se encuentran conectados a internet y pueden ser manipulados remotamente desde cualquier dispositivo, tienen una alta probabilidad de ser hackeados por piratas informáticos. Uno de los casos más relevantes en cuanto al hackeo de objetos inteligentes es el de la compañía de vehículos *Tesla*, en el cual la compañía de seguridad informática Keen Security Lab hackeó el sistema del vehículo logrando tener acceso a él y realizando acciones remotas [8].



## BIBLIOGRAFÍA

- [1] "exTouch: Spatially-Aware Embodied Manipulation of Actuated Objects ...."  
<http://tmg-trackr.media.mit.edu/publishedmedia/Papers/522-exTouch%20SpatiallyAware%20Embodied/Published/PDF>
- [2] (n.d.). Smarter Objects: Using AR technology to Program Physical Objects .... Se recuperó el febrero 16, 2017 de [http://fluid.media.mit.edu/sites/default/files/WIP\\_Smarter\\_Objects\\_0.pdf](http://fluid.media.mit.edu/sites/default/files/WIP_Smarter_Objects_0.pdf)
- [3] (n.d.). Open Hybrid | Fluid Interfaces. Se recuperó el febrero 16, 2017 de <http://fluid.media.mit.edu/projects/open-hybrid>
- [4] (n.d.). object-oriented video: interaction with real-world objects through live .... Se recuperó el febrero 16, 2017 de [http://dl.acm.org/ft\\_gateway.cfm?id=143011](http://dl.acm.org/ft_gateway.cfm?id=143011)
- [5] (2012, mayo 5). Evaluating the benefits of real-time feedback in mobile augmented .... Se recuperó el febrero 16, 2017 de <http://dl.acm.org/citation.cfm?id=2208706>
- [6] (n.d.). jQuery.ajax() | jQuery API Documentation. Se recuperó el febrero 16, 2017 de <http://api.jquery.com/jquery.ajax/>
- [7] (n.d.). Welcome to the Exclusive Touch. Se recuperó el febrero 21, 2017 de <https://www.extouch.com/>
- [8] (n.d.). Chinese Company hacks Tesla car remotely. Se recuperó el septiembre 20, 2016 de <http://www.cnbc.com/2016/09/20/chinese-company-hacks-tesla-car-remotely.html>

## ANEXOS

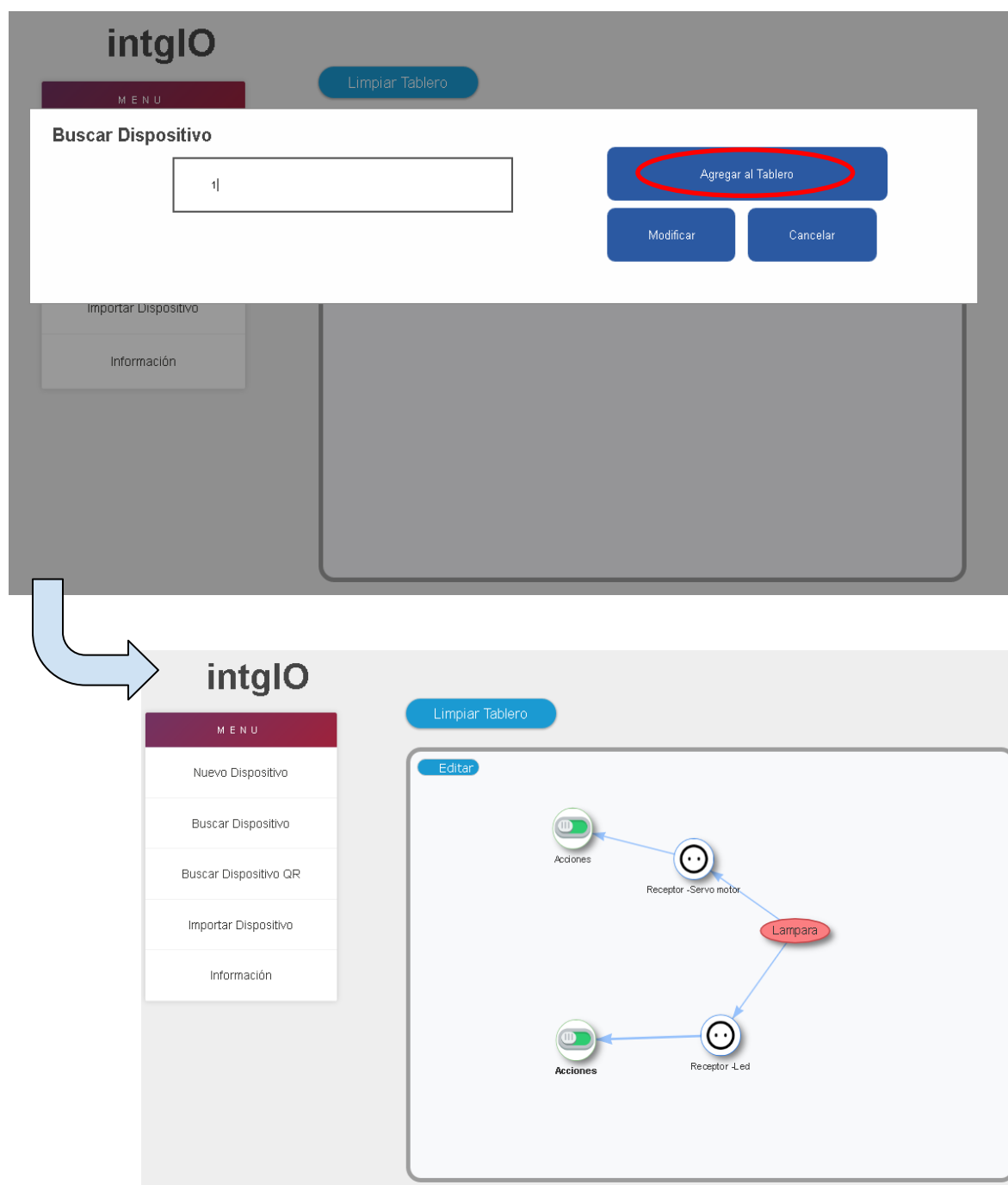
Registrar un objeto inteligente mediante la opción “Nuevo Dispositivo”.

The diagram illustrates the process of registering an intelligent object through three sequential forms:

- Nuevo Dispositivo:** This form contains input fields for 'Código Dispositivo', 'Modelo Dispositivo', 'Nombre Dispositivo', 'Dirección IP', and 'Puerto'. A dropdown menu for 'Tipo de Comunicación' is set to 'TCP'. At the bottom, there are 'Cancelar' and 'Continuar' buttons. The 'Continuar' button is circled in red.
- Nuevo Acción:** This form contains input fields for 'Código de Acción', 'Objeto Relacionado' (dropdown), and 'Tipo de Acción' (dropdown, set to 'Incremental'). It also has a 'Descripción' text area. At the bottom, there are 'Cancelar' and 'Crear' buttons. The 'Crear' button is circled in red.
- Nuevo Objeto:** This form contains input fields for 'Nombre Objeto' and 'Código Objeto'. A dropdown menu for 'Tipo de Objeto' is set to 'Emisor'. At the bottom, there are 'Cancelar' and 'Crear' buttons. The 'Crear' button is circled in red.

Blue arrows indicate the flow from 'Nuevo Dispositivo' to 'Nuevo Acción', and from 'Nuevo Acción' to 'Nuevo Objeto'. Red circles highlight the 'Continuar', 'Crear', and 'Crear' buttons respectively.

**Agregar un objeto inteligente al tablero mediante la opción “Buscar Dispositivo” → “Agregar al tablero”.**



**Modificar un objeto inteligente mediante la opción “Buscar Dispositivo” → “Modificar”.**

The interface is titled "intgLO" and includes a "Limpiar Tablero" button. The "Buscar Dispositivo" section contains a search input field with "1" and buttons for "Agregar al Tablero", "Modificar", and "Cancelar". The "Editar Dispositivo" section shows fields for "Código Dispositivo" (1), "uno", "Lampara", "Tipo de Comunicación" (TCP), "200.126.23.139", and "3636", along with "Guardar Cambios", "Objetos", and "Cerrar" buttons. The bottom section features a table with columns "Código", "Nombre", "tipo", and "Editar", and a "Volver a Dispositivo" button.

Código	Nombre	tipo	Editar
3	Servo motor	Receptor	Modificar
2	Led	Receptor	Modificar

**Editar Objeto**

3

Servo motor

Tipo de Objeto: Receptor

Modificar Acciones Volver

Agregar Acción

Código	tipo	Editar
	incremental	Modificar

Volver a Objeto

**Editar Acción**

Código de Acción

5

Tipo de Acción: Incremental

Descripción

motor

Volver Modificar