TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

INDI Web Client

Desarrollo de un prototipo de cliente Web INDI para el control de instrumental astronómico

Autor
Pablo Torrecillas Ortega

Tutor
Prof. Dr. Sergio Alonso Burgos

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

—

Granada, 11 de Diciembre de 2015

INDI Web Client

Desarrollo de un prototipo de cliente Web INDI para el control de instrumental astronómico.

Autor
Pablo Torrecillas Ortega

Tutor
Prof. Dr. Sergio Alonso Burgos

INDI Web Client: Desarrollo de un prototipo de cliente Web INDI para el control de instrumental astronómico

Pablo Torrecillas Ortega

Palabras clave: INDI, *Software* Libre, Cliente Web, Internet, Astronomía, Control, Prototipo, Navegador

Resumen

Este proyecto ha sido realizado ante la necesidad de conseguir diferentes herramientas para el control de instrumental astronómico y facilitar el trabajo tanto a profesionales como a *amateurs* en el campo de la astronomía. Una vez analizado el estado del arte y los medios que se utilizan en este momento, se ha desarrollado un proyecto encaminado a conseguir los objetivos planteados a través de un prototipo de cliente web viniendo a completar de esta forma las diferentes aplicaciones que ya existen.

El objetivo del proyecto es poder realizar la función control de un observatorio astronómico situado en cualquier lugar del mundo con conexión a Internet.
Las tareas que se pueden realizar son las siguientes: recepción y procesamiento de datos de diferente naturaleza, envío de los mismos y cambio de parámetros y valores. Este trabajo viene a ser la base para otras funciones posteriores que se pueden desarrollar a partir del prototipo de cliente web realizado.

Para ello, se ha utilizado como punto de partida el protocolo INDI que ya se encuentra desarrollado. INDI es, a grandes rasgos, un protocolo que facilita el control en el tiempo y en el espacio, así como la adquisición de datos y el intercambio entre diferentes dispositivos *hardware* y sus interfaces *software*.

La utilización de este protocolo, lleva aparejado una enorme ventaja y es el poder emplear diferentes dispositivos obteniendo resultados con el mismo beneficio, independientemente de la naturaleza del dispositivo utilizado, abriéndose así un abanico de posibilidades ante el problema que se podría plantear si se tuviera que utilizar un dispositivo concreto y único, pues de esa forma se acotarían enormemente las posibilidades si no se cuenta con el dispositivo en cuestión y por tanto no se podría llevar a cabo la función control. Cabe destacar que este prototipo de cliente es multiplataforma y por tanto es capaz de ejecutarse en cualquier máquina que disponga de navegador web.

La aportación de este proyecto, basado en un prototipo de cliente web, es una nueva e importante posibilidad que viene a unirse a las ya existentes, complementando la forma de obtener resultados utilizando otras herramientas.

Dado que la difusión de Internet se hace ya de forma cotidiana y es fácil encontrar un punto de acceso al mismo, prácticamente en cualquier punto del planeta el astrónomo tanto *amateur* como profesional, solo necesitaría un navegador web y la conexión a Internet para poder trabajar y controlar todos los aspectos astronómicos que desee.

Este proyecto aporta una innovación de forma sencilla, rápida y poco costosa en la forma de trabajar en la obtención de datos y su posterior manipulación en el campo de la astronomía.

Hay que hacer justa mención al *software* libre y su filosofía pues han sido pilares básicos en el trabajo que hoy se presenta, ya que se han utilizado todas las herramientas de dicho *software* para el desarrollo del prototipo del cliente web.

Como conclusión, cabe destacar que este proyecto sienta las bases para conseguir llegar a ser un cliente web como producto final utilizando el trabajo desarrollado y que se iniciaba creando un prototipo de cliente web basado en INDI.

Project Title: Project Subtitle

First name, Family name (student)

Keywords: Keyword1, Keyword2, Keyword3, ...

Abstract

Write here the abstract in English.

Yo, **Pablo Torrecillas Ortega**, alumno de la titulación Grado de Ingeniería Informática de **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI XXXXXXXXX, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Pablo Torrecillas Ortega

Granada a 11 de diciembre de 2015.

D. **Sergio Alonso Burgos**, Profesor del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado ***INDI Web Client, Desarrollo de un prototipo de cliente Web INDI para el control de instrumental astronómico***, ha sido realizado bajo su supervisión por **Pablo Torrecillas Ortega**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 11 de diciembre de 2015.

Tutor:

Prof. Dr. Sergio Alonso Burgos

Agradecimientos

Poner aquí agradecimientos...

Índice de figuras

Índice de cuadros

Índice general

Capítulo 1

Introducción

1.1. Astronomía

La astronomía es la ciencia que se ocupa del estudio de los cuerpos celestes del universo, los planetas y sus satélites, los cometas y meteoroides, las estrellas y la materia interestelar, los sistemas de materia oscura, estrellas, gas y polvo llamados galaxias y los cúmulos de galaxias, así como sus movimientos, los fenómenos ligados a ellos y las leyes que los rigen.

La palabra, como tal, proviene del latín astronomía. La astronomía ha formado parte de la historia de la humanidad considerándola como la ciencia más antigua. Civilizaciones como la azteca, la maya y la inca, así como la egipcia, la china y la griega alcanzaron un grado tal de conocimientos que son tenidos por fundamentales para la posterior evolución de esta disciplina, considerándola como esencial para otras ciencias como la matemática o la física.

Su registro y la investigación de su origen se produce a partir de la información que llega de ellos a través de la radiación electromagnética o de cualquier otro medio.

Es una de las pocas ciencias en las que los *amateurs* también pueden desempeñar un papel activo, especialmente en el descubrimiento y seguimiento de fenómenos como curvas de luz de estrellas variables, descubrimiento de asteroides y cometas, etc.

En sus inicios, la astronomía tenía una aplicación práctica para conocer los ciclos de los astros y establecer medidas de tiempo que permitieran determinar, entre otras cosas, el momento propicio para la siembra y la cosecha. En los pueblos antiguos, los astros se consideraban como divinidades y el estudio de sus posiciones resultaba esencial para determinar sus influencias sobre los acontecimientos terrenales. Por este conjunto de razones la astronomía fue, en todas las civilizaciones del pasado, una ciencia tanto al servicio del poder civil como del religioso.

Antiguamente se ocupaba, únicamente, de la observación y predicciones

de los movimientos de los objetos visibles a simple vista, quedando separada durante mucho tiempo de la Física. En Sajonia-Anhalt, Alemania, se encuentra el famoso Disco celeste de Nebra, que es la representación más antigua conocida de la bóveda celeste. Quizá fueron los astrónomos chinos quienes dividieron, por primera vez, el cielo en constelaciones. Los antiguos griegos hicieron importantes contribuciones a la astronomía, entre ellas, la definición de magnitud. La astronomía precolombina poseía calendarios muy exactos y parece ser que las pirámides de Egipto fueron construidas sobre patrones astronómicos muy precisos.

Fue probablemente Eratóstenes quien diseñara la esfera armilar que es un astrolabio para mostrar el movimiento aparente de las estrellas alrededor de la tierra.

La astronomía observacional estuvo casi totalmente estancada en Europa durante la Edad Media, a excepción de algunas aportaciones como la de Alfonso X el Sabio con sus tablas alfonsíes, o los tratados de Alcabitius, pero floreció en el mundo con el Imperio persa y la cultura árabe. Al final del siglo X, un gran observatorio fue construido cerca de Teherán (Irán), por el astrónomo persa Al-Khujandi, quien observó una serie de pasos meridianos del Sol, lo que le permitió calcular la oblicuidad de la eclíptica. También en Persia, Omar Khayyam elaboró la reforma del calendario que es más preciso que el calendario juliano acercándose al Calendario Gregoriano.

Durante siglos, la visión geocéntrica de que el Sol y otros planetas giraban alrededor de la Tierra no se cuestionó. En el Renacimiento, Nicolás Copérnico propuso el modelo heliocéntrico del Sistema Solar. Su trabajo De Revolutionibus Orbium Coelestium fue defendido, divulgado y corregido por Galileo Galilei y Johannes Kepler, autor de Harmonices Mundi, en el cual se desarrolla por primera vez la tercera ley del movimiento planetario.

Galileo añadió la novedad del uso del telescopio para mejorar sus observaciones. La disponibilidad de datos observacionales precisos llevó a indagar en teorías que explicasen el comportamiento observado. Al principio sólo se obtuvieron reglas como las leyes del movimiento planetario de Kepler, descubiertas a principios del siglo XVII.

Fue Isaac Newton quien extendió hacia los cuerpos celestes las teorías de la gravedad terrestre y conformando la Ley de la gravitación universal, inventando así la mecánica celeste, con lo que explicó el movimiento de los planetas y consiguiendo unir el vacío entre las leyes de Kepler y la dinámica de Galileo. Esto también supuso la primera unificación de la astronomía y la física.

Tras la publicación de los Principios Matemáticos de Isaac Newton (que también desarrolló el telescopio reflector), se transformó la navegación marítima. A partir de 1670, utilizando instrumentos modernos de latitud y los mejores relojes disponibles se ubicó cada lugar de la Tierra en un planisferio o mapa, calculando para ello su latitud y su longitud. Los requerimientos de la navegación supusieron un empuje para el desarrollo progresivo de obser-

vaciones astronómicas e instrumentos más precisos, constituyendo una base de datos creciente para los científicos.

A finales del siglo XIX se descubrió que, al descomponer la luz del Sol, se podían observar multitud de líneas de espectro , regiones en las que había poca o ninguna luz.

Se descubrió que las estrellas eran objetos muy lejanos y con el espectroscopio se demostró que eran similares al Sol, pero con una amplia gama de temperaturas, masas y tamaños. La existencia de la Vía Láctea como un grupo separado de estrellas no se demostró hasta el siglo XX, junto con la existencia de galaxias externas y, poco después, la expansión del universo, observada en el efecto del corrimiento al rojo. La astronomía moderna también ha descubierto una variedad de objetos exóticos como los quásares, púlsares, radiogalaxias, agujeros negros, estrellas de neutrones, y ha utilizado estas observaciones para desarrollar teorías físicas que describen estos objetos.

Durante el siglo XX, la espectrometría avanzó, en particular, como resultado del nacimiento de la física cuántica, necesaria para comprender las observaciones astronómicas y experimentales.

La astronomía moderna se divide en varias ramas: astrometría, el estudio mediante la observación de las posiciones y los movimientos de estos cuerpos; mecánica celeste, el estudio matemático de sus movimientos explicados por la teoría de la gravedad; astrofísica, el estudio de su composición química y su condición física mediante el análisis espectral y las leyes de la física, y cosmología, el estudio del Universo como un todo.

1.2. Observatorio e Instrumental Astronómico

La denominación y el edificio conocido como Observatorio Astronómico, lugar desde donde se estudian y controlan los cambios, los movimientos y las leyes que rigen los astros, ha sufrido grande cambios con el paso del tiempo. En la antigüedad, dado que la astronomía estaba ligada a la religión y por tanto a los templos, estos eran los lugares que servían para hacer de ellos observatorios astronómicos.

Fue ya en la Edad Media cuando el observatorio comenzó a ser un lugar en el cual se reunían los astrónomos y en él se fueron disponiendo los diferentes instrumentos o herramientas que facilitaban el estudio de aquellas personas, profesionales o *amateurs*, que se dedicaban a esta disciplina. Después de las primeras décadas del siglo XX, los astrónomos se veían en la obligación de alejarse de la ciudad debido a la contaminación lumínica y química que en ella se produce. Es en este momento histórico cuando comenzaron a construirse los observatorios astronómicos, siguiendo en la actualidad ubicándose en lugares desérticos y elevados para conseguir trabajar con un cielo oscuro, libre de contaminación lumínica y consiguiendo que el

número de días serenos sea más elevado.

Por otra parte, se denomina instrumental astronómico al conjunto de instrumentos a disposición del astrónomo para complementar y facilitar sus observaciones. A continuación, se pasa a describir parte del instrumental más importante que podemos encontrar en un observatorio.

1.2.1. Telescopio

Figura 1.1: Telescopio Astronómico (http://blog.astroaficion.com/)

El telescopio es un instrumento cuya función principal es recoger la luz de un objeto lejano y ampliarlo. Está considerado como el artífice de la astronomía moderna.

Históricamente se le atribuye el descubrimiento a Hans Lippershey que era un fabricante de lentes, alemán, que también era científico, astrónomo e inventor. Cuando Galileo Galilei escuchó la noticia de la creación del telescopio, decidió diseñar y construir uno, y en 1609 lo registró como el primer telescopio astronómico. Gracias a él, se pudo llevar a a cabo la destacada observación del día 7 de enero de 1610 donde se pudieron observar la existencia de cuatro de las lunas de Júpiter girando entorno al planeta. En un principio el nombre que se le puso al instrumento fue de “lente espía”, y el 14 de abril de 1611 Giavanni Demisiani propuso el nombre de telescopio durante una cena que se brindó en honor a Galileo. Galileo decidió llevar a esa cena el telescopio y los comensales pudieron observar las lunas de Júpiter. El nombre por el que se conoce este instrumento, Telescopio, proviene del griego con el prefijo tele que significa “lejos” y se le añade la raíz griega skop que significa “ver”.

Además de poder ampliar los objetos lejanos para verlos con mayor detalle, gracias al telescopio se pueden observar cuerpos celestes de débil lumi-

nosidad y que son invisibles a simple vista, ya que el objetivo del telescopio es capaz de percibir más luz que el ojo humano. Cuanto mayor es el diámetro del objetivo, más luz capta. Dicho diámetro suele referenciarse como la "apertura del telescopio" y de ello depende el poder que tenga de resolución el objetivo y por tanto el telescopio.

Los primeros telescopios que se consolidaron fueron los de tipo kepleriano que disponían de unas longitudes focales de hasta 30 o 40 centímetros, con el objetivo de conseguir grandes aumentos. A principios del siglo XVIII comenzaron a fabricarse los telescopios con lente y un espejo cóncavo a su vez. A partir de ese momento comenzó un debate sobre los telescopios, su utilidad y cual de ellos ofrecía mayor precisión, ya que existían dos tipos, los telescopios reflectores, en los que la luz es reflejada y dirigida hacia el foco, y los telescopios refractores, en donde la luz es refractada pasando por el objetivo. A mediados del siglo XX se terminó la disputa cuando triunfaron de manera definitiva los telescopios reflectores.

1.2.2. Cámara CCD

Figura 1.2: Cámara CCD (http://bitran.co.jp/)

Las siglas de cámara CCD (*Charge-Coupled Device*) proceden del inglés y significan dispositivo de carga acoplada. Los primeros dispositivos que se crearon fueron inventados por Willard Boyle y George E. Smith, el 17 de octubre de 1969 dentro de los laboratorios Bell y ambos fueron distinguidos recientemente con el Premio Nobel de Física, en el año 2009, gracias a los dispositivos CCD.

Dicha cámara funciona como un detector de estado sólido. Este funcionamiento hace que la cámara sea muy eficiente ya que hace mucho más fácil la obtención y el procesamiento de las imágenes astronómicas. La cámara CCD funciona como un conjunto de circuitos eléctricos de condensadores enlazados y acoplados que son sensibles a la radiación electromagnética. Estos

condensadores pueden transferir su carga eléctrica a uno o varios condensadores que se encuentren a su lado en el circuito impreso. El CCD registra la ubicación de cada fotodiodo sobre el que incide un fotón de rayos X. Un fotón es un paquete que contiene radiación electromagnética. A su vez, el CCD es capaz de registrar la energía que depende de su frecuencia, y por tanto, de su longitud de onda.

Los sensores CCD tienen una estructura de células que son sensibles a la luz en forma de mosaico y a cada una de esas células se le llama *píxel*. Cada *píxel* es una estructura detectora en la cual se pueden almacenar fotones. Desde el CCD la imagen es procesada por la cámara y a su vez registrada en la tarjeta de memoria.

1.2.3. Montura

Figura 1.3: Montura (http://astrofacil.com/)

La montura es la parte mecánica. Sirve como estructura para montar y sujetar el tubo del telescopio y así poder realizar la operación de enfoque para poder detectar y seguir a un cuerpo celeste. Se requiere que la montura sea de mucha firmeza y a su vez tenga mucha suavidad en los movimientos para poder conseguir que las observaciones sean perfectas.

Existen varios tipos de monturas entre las que destacan: Altacimutales y Ecuatoriales.

La montura altacimutal es aquella que hace referencia al sistema de coordenadas celestes altacimutales. Requiere un continuo ajuste ya que tienen

absoluta libertad para moverse en altura (de arriba hacia abajo) y en acimut (de derecha a izquierda).

La montura ecuatorial es aquella que hace referencia al sistema de coordenadas celestes ecuatoriales. Tiene un eje ecuatorial que está alineado con el eje terráqueo y algunas de estas monturas pueden disponer de un pequeño motor capaz de realizar un giro completo en 24 horas. Dispone además de otro eje, llamado declinación y es regular al primero. Cuando se encuentra enfocado un astro y el motor se ha accionado, el tubo del telescopio sigue automáticamente el movimiento de la bóveda celeste y el objeto enfocado permanecerá fijo en el interior del campo visual del telescopio. Por este motivo tan importante, la montura ecuatorial es muy eficaz en la astronomía ya que permite exposiciones muy prolongadas.

1.2.4. Enfocador

Figura 1.4: Enfocador (http://tienda.lunatico.es/)

El enfocador es una de las piezas fundamentales del telescopio ya que gracias a él, nos permitirá ver las imágenes formadas tras la reflexión de la luz en el espejo primario y su desviación por el espejo secundario.

Para ver las imágenes se necesitará un juego de oculares. La combinación de la longitud focal de los oculares y la longitud focal del telescopio, nos dará como resultado el número de aumentos total que tenemos en nuestro sistema.

Uno de los enfocadores más populares que existe actualmente en el mercado es el de tipo *Crayford*. Estos enfocadores son bastante caros ya que tienen una relación de velocidad alta que mejora el ajuste de enfoque.

1.2.5. Rueda Portafiltros

La rueda portafiltros es un objeto, generalmente de aluminio, que contiene en su interior filtros diferentes para poder cambiar de forma eficaz la visual o la astrofotografía. No hace falta quitar la cámara para cambiar el filtro ya que tan solo con un pequeño giro en la rueda se selecciona el filtro poniéndose en el lugar correcto y se puede seguir observando. Como mínimo,

Figura 1.5: Rueda Portafiltros (http://astrocity.es/)

se requiere que tenga cuatro filtros si se quiere realizar astrofotografía con cámaras CCD blanco y negro. Para ello, se necesitarán el filtro azul, el filtro rojo y el filtro verde (RGB), y posiblemente un filtro para los infrarrojos.

1.2.6. Cúpula

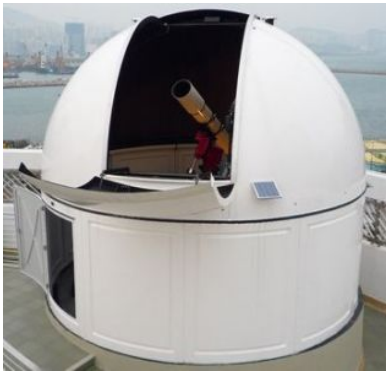

Figura 1.6: Cúpula Astronómica (http://apt.com.es/)

Una cúpula es una estructura principalmente semiesférica o de techo deslizante. La cúpula permite alojar en su interior el instrumental astronómico y protegerlo. Ya sea de un tipo o de otro, está formada por una o más escotillas que permiten su abertura y, de esta forma, poder realizar una observación.

Figura 1.7: Óptica Adaptativa (http://www.valkanik.com/)

1.2.7. Óptica Adaptativa

La óptica adaptativa es una técnica que permite, mediante el uso de óptica deformable, corregir en tiempo real los defectos que vienen de la atmósfera terrestre en la imagen observada con el telescopio. Es una técnica muy importante para todos los astrónomos ya que se pueden obtener imágenes mucho más nítidas. Un símil que hacen muchos astrónomos para entender esto es que la técnica es comparable con mirar un objeto situado en el fondo de una piscina con agua y sin agua. La óptica adaptativa es capaz de eliminar las perturbaciones y por tanto equivaldría a observar desde el espacio. Ganar nitidez con la óptica en las imágenes significa concentrar en un menor número de puntos sensibles del detector los pocos fotones que llegan de los objetos débiles o lejanos y eso es, dicho con otras palabras, que la posibilidad de verlos es mayor.

1.2.8. Estación Meteorológica

Figura 1.8: Estación Meteorológica (http://comprawifi.com/)

Una estación meteorológica es una instalación cuya función principal es medir y registrar diversos datos meteorológicos. Dichos datos se utilizan

para elaborar predicciones meteorológicas a partir de modelos numéricos. A su vez, la estación está formada por sensores e instrumentos que sirven para la obtención de todos los datos entre los que destacan el termómetro, el barómetro, el pluviómetro, el psicómetro o la veleta.

1.2.9. Controlador de Dispositivos

Hoy en día, en los observatorios, se pueden controlar los diversos instrumentos de un observatorio de diferentes formas como pueden ser:

- Controlados directamente desde el propio dispositivos.
- Mediante el uso de un PC y desde ahí se controlarían todos los aspectos.
- Mediante el uso de herramientas de control remoto. En este caso situarían al astrónomo en un lugar fuera del observatorio y desde ese lugar se podrían controlar los diferentes dispositivos que permitan realizar el trabajo del astrónomo.

1.3. Plataformas para la creación de *software* astronómico

Actualmente existen dos grandes protocolos para la creación de *software* astronómico. Estos dos claros protocolos son ASCOM e INDI. La función de ambos protocolos es llegar a poder controlar todo el instrumental astronómico desde una misma máquina.

1.3.1. ASCOM

El protocolo ASCOM está formado por un conjunto de desarrolladores y fabricantes de instrumentales astronómicos. Lo que buscan lograr es la compatibilidad plug-and-play, independientemente del lenguaje que se utilice entre el *software* y los instrumentos astronómicos. Para ello, crean una capa para separar los dispositivos del *software* que utilizan los dispositivos. Otro objetivo es conseguir hacer que cualquier driver pueda ser utilizado en cualquier lenguaje de programación.

Tiene un gran inconveniente y es que actualmente solo funciona en sistemas operativos de *Microsoft Windows* y dificulta su desarrollo ya que solamente está abierto a un único sistema operativo.

Por otra parte, hay que detallar que ASCOM no está pensado para funcionar remotamente pero muchos astrónomos recurren al escritorio remoto de *Microsoft Windows* para así poder trabajar remotamente. [1]

1.3.2. INDI

INDI es un protocolo de *Software Libre* diseñado para apoyar el control, la automatización, la adquisición de datos y el intercambio entre los dispositivos *hardware* y las interfaces *software*. Se puede usar tanto en dispositivos reales como en dispositivos virtuales.

Su biblioteca permite controlar cualquier dispositivo con un driver INDI mediante el paso de archivos del tipo XML.

INDI significa "*Instrument-Neutral-Distributed-Interface*". INDI fue creado por *Elwood C. Downey* del *ClearSky Institute* [6].

Cómo funciona INDI

Hoy en día, los sistemas de control que se desarrollan se hacen para una gama de dispositivos concretos o incluso para un dispositivo específico. Si se modifica algún parámetro del dispositivo o de la gama de dispositivos, el *software* debe modificarse para asentar dicho cambio. En otras palabras, hay una fuerte conexión entre el *software* y el *backend* del *hardware*.

Para que no exista esa fuerte conexión entre el *software* y el *backend* del *hardware* se desarrolla INDI, ya que los clientes son totalmente conscientes de las capacidades de los diferentes dispositivos que tienen conectados a través del protocolo INDI. Esto hace que en tiempo de ejecución, se construya una interfaz gráfica en función a las características que envíe el dispositivo. Dicha interfaz por tanto, se construye de forma dinámica.

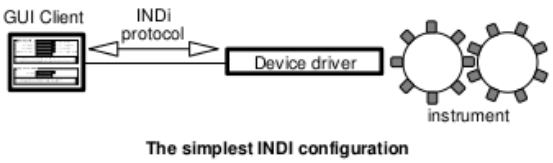

Figura 1.9: Configuración de INDI (http://indilib.org/)

Para que INDI funcione con total perfección se definen tres sustantivos muy importantes que hacen que encajen todas las piezas correctamente. Estos sustantivos son: *drivers*, servidor y cliente.

Drivers, Servidores y Clientes de INDI

- El *driver* de INDI es un controlador que se comunica directamente con el dispositivo. Permite conectarse a uno o más dispositivos físicos y es el responsable de definir todos los clientes. Los *drivers* envían una lista de propiedades a los clientes para que éstos puedan así definir dinámicamente sus interfaces gráficas.

- El servidor es un programa que ejecuta los diferentes *drivers* de los distintos dispositivos. Hace de conexión entre los dispositivos y los clientes. Por regla general está alojado en un ordenador en el cual están todos los dispositivos conectados. También el servidor permite que estén los dispositivos conectados en otras máquinas y éstas envíen los datos al servidor. Todo el intercambio de datos que se realiza entre el servidor y los *drivers* se consigue mediante el uso del protocolo INDI.
- El cliente es un programa que hace dinámicamente la interfaz gráfica. Puede estar conectado con uno o más servidores. Una vez que conecta con uno o más servidores el cliente solicita toda la información y el servidor se la da. Seguidamente, el cliente procesa los datos y los muestra creando la interfaz que se menciona anteriormente.

Descripción del protocolo

El protocolo INDI define un conjunto de 5 propiedades que son las que envían los diferentes *drivers* para formar la interfaz del cliente. Estas propiedades están muy relacionadas ya que una o todas las propiedades pueden estar presentes en un dispositivo. A continuación se pasa a detallar cada una de las propiedades:

- **Textos:** Son cadenas de caracteres ordenados arbitrariamente.
- **Números:** Son cantidades numéricas. Además de la cantidad numérica se envían otros parámetros que sirven para el formato de su visualización y configuración.
- **Switchs:** Son propiedades que están encendidas o apagadas. Estas propiedades pueden ser de tres tipos diferentes:
  - **Una de muchas:** de todas las opciones se tiene que seleccionar obligatoriamente una.
  - **Como máximo una:** de todas las opciones se puede seleccionar como máximo una.
  - **Cualquiera de muchas:** se podrán seleccionar todas las que se deseen.
- **Lights:** Son propiedades que pueden estar en cada uno de los cuatro estados que se definen:
  - **Inactivo:** Luz de color gris.
  - **Alerta:** Luz de color rojo.
  - **Ocupado:** Luz de color amarillo.
  - **Ok:** Luz de color verde.

- **Blob:** Son propiedades que tienen objetos binarios arbitrariamente, como imágenes.

Simuladores

INDI dispone de un conjunto de simuladores que sirven para testear todo lo que se va haciendo sin necesidad de tener un dispositivo físico. Con él se pueden realizar todo tipo de operaciones sin variar nada con respecto al dispositivo real. Entre ellos destacan:

- Simulador de telescopio.
- Simulador de CCD.
- Simulador de rueda portafiltros.
- Simulador de enfocador.
- Simulador de GPS.

1.4. Estado del arte. *Software* con INDI y sin él

Actualmente existen diferentes *software* para realizar la función control de un observatorio. Hay que destacar que podemos diferenciarlos en dos grandes grupos: Los *software* que no hacen uso de la biblioteca INDI y los que si hacen uso de dicha biblioteca.

A continuación se hace un pequeño detalle de algunos de estos *software*:

1.4.1. Clientes que no hacen uso de INDI

- **Maxim DL:** Dispone de una completa integración del observatorio y controla todo el equipo astronómico. Es compatible con ASCOM y el astrónomo puede crear sus propios preajustes. A su vez, se puede supervisar y controlar el observatorio mediante una webcam en la cúpula, interruptores remotos y vigilancia del clima. Es uno de los software sin INDI más completo que existe. [7]
- **Images Plus:** Es un *software* que en un principio sirve para realizar el tratamiento de astrofotografía pero que después se ha ido mejorando y hoy en día permite hacer uso y control de dispositivos. Su uso es muy limitado ya que solamente permite hacer uso de algunos dispositivos concretos. También funciona con ASCOM. [5]
- **Astroplanner:** Permite planificar y ejecutar una sesión de observación al astrónomo. Se pueden introducir los sitios que se desean observar, importarlos desde archivos o buscar en el catálogo del que dispone

el programa. Se puede usar tanto en MAC OS X (10.4 o superior) y en *Windows* (XP o superior). Funciona también con ASCOM en la versión para *Windows* pero solamente para el control de telescopios. [2]

1.4.2. Clientes que hacen uso de INDI [4]

- **KStars:** Es una herramienta de *Software Libre* de astrofotografía incluida en multiplataforma. Proporciona una simulación gráfica precisa del cielo nocturno desde cualquier punto de la tierra. Además, proporciona herramientas de cálculo astronómico y mucha información que ayudará y facilitará el trabajo al astrónomo.

- **Ekos:** Software que lleva a cabo tareas de astrofotografía bajo la plataforma Linux. Ekos es también una herramienta de *Software Libre* y forma parte del *software* de Kstars mencionado anteriormente.

- ***Remote Observatory*:** Es un cliente INDI multilenguaje de *Software Libre* para la plataforma Android. Se encuentra en un continuo desarrollo y permite controlar cualquier número de servidores así como múltiples interfaces.

Como conclusión final diremos, que vistos algunos de los clientes que se pueden usar para llevar a cabo el control de un observatorio hay que resaltar que ninguno de ellos es capaz de llevar dicho control desde el navegador, y por tanto, necesitaríamos de un pc con unos requerimientos mínimos puesto que todos los pcs que existen en la actualidad los superarían, y una instalación del *software* que, en nuestro caso, seleccionásemos.

1.5. Conexión: *Socket* y *WebSocket*

Al comenzar a trabajar en el apartado de las conexiones del cliente web y el servidor, advertimos que necesitábamos crear un socket desde el prototipo del cliente para que funcionase correctamente, ya que lo que se pretendía, y esa era nuestra búsqueda, que la conexión fuese persistente entre el cliente y el servidor, y además, que ambas partes pudiesen enviar datos en cualquier momento. Para ello se hizo uso del *websocket*.

1.5.1. *Socket*

Un *Socket* de Internet es un método en la comunicación por el cual un cliente y un servidor pueden intercambiar cualquier flujo de datos y que generalmente se hace de manera fiable y ordenada. Es usado para la familia de protocolos *TCP/IP*.

El *Socket* constituye el mecanismo para la entrega de paquetes de datos que provienen de la tarjeta de red a los procesos o hilos apropiados. Además,

queda definido por un par de direcciones IP, una local y otra remota, un protocolo de transporte y un par de números que son el puerto local y el puerto remoto. [11]

1.5.2. *WebSocket*

Un *WebSocket* es una tecnología que proporciona un canal de comunicación bidireccional sobre un único *socket TCP*. Está diseñada para ser implementada en navegadores y en servidores web, sin embargo, puede utilizarse por cualquier aplicación cliente/servidor.

En el ámbito del cliente, *WebSocket* está ya implementado en *Mozilla Firefox 8*, *Google Chrome 4* y *Safari 5*. Para versión móvil de *Safari* en el *iOS 4.2* y en *Internet Explorer 10*. [12]

Para ejecutar un *websocket*, tan solo hay que lanzar en el terminal la siguiente instrucción:

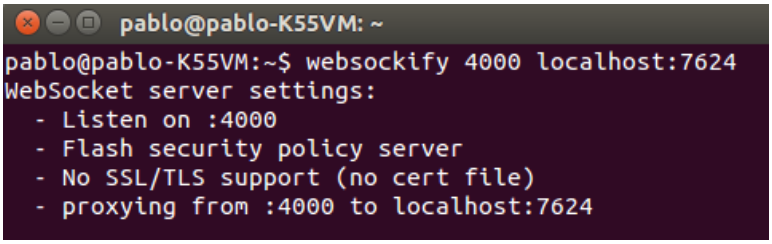

Figura 1.10: Lanzamiento de un *WebSocket*

Con eso estaremos "*WebSockificando*" el puerto 4000 del servidor y el puerto 7624 de nuestra máquina.

Las comunicaciones de origen cruzado son un protocolo moderno creado directamente para *WebSocket*. Aunque sigue siendo necesario que se aseguren las comunicaciones entre clientes y servidores de forma segura, *WebSocket* permite la comunicación entre las partes de cualquier dominio.

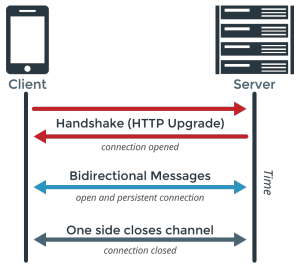

Figura 1.11: Diagrama de funcionamiento de un *WebSocket*

Capítulo 2

Objetivos

El objetivo fundamental de este proyecto es desarrollar un prototipo de cliente web basado en INDI que sea capaz de realizar la función control sobre un observatorio astronómico. Dicho prototipo será multiplataforma y utilizará tecnologías de carácter *Software Libre*. Se pretende además, que en la medida de lo posible se añadan nuevas funcionalidades en un futuro y que se convierta en un cliente web y no en un prototipo.

2.1. Objetivos Principales y Secundarios

A continuación se detallan los objetivos principales del prototipo del cliente web:

- **OBJ-1:** Crear un prototipo de cliente que sea capaz de realizar el control de cualquier dispositivo INDI sin apreciar su naturaleza.
- **OBJ-2:** Permitir la conexión con un servidor concreto mediante su IP y su puerto.
- **OBJ-3:** Efectuar la gestión de múltiples dispositivos que estén conectados a un servidor.
- **OBJ-4:** Desarrollar un prototipo de cliente multiplataforma, que permita hacer uso del prototipo del cliente en los principales navegadores que existen en el mercado.
- **OBJ-5:** Utilizar *Software Libre* en la aplicación como herramienta de trabajo.
- **OBJ-6:** Ofrecer la posibilidad de que cualquier desarrollador pueda aportar su capacidad y conocimiento al prototipo.
- **OBJ-7:** Aportar una nueva forma de acceder a los datos de un observatorio astronómico.

- **OBJ-8:** Optimizar el uso de instrumentos de astronomía facilitando una nueva herramienta de obtención de datos.

Por otra parte, se busca conseguir los siguientes objetivos secundarios:

- **OBJ-S-1:** Adaptar el prototipo de cliente web a las propiedades estándares de INDI.
- **OBJ-S-2:** Adecuar el prototipo a los estándares de HTML, CSS, Jquery y JavaScript.
- **OBJ-S-3:** Desarrollar el prototipo de modo que funcione correctamente con el mayor número posible de dispositivos conectados al servidor.
- **OBJ-S-4:** Conseguir que la interfaz sea maximizable y minimizable independientemente del número de dispositivos conectados al servidor.
- **OBJ-S-5:** Difundir el prototipo de cliente web en la página web de INDI.
- **OBJ-S-6:** Divulgar el prototipo de cliente web en el foro de INDI

2.2. Materias y Herramientas para el Desarrollo

Este trabajo no se hubiera podido desarrollar sin haber obtenido unos conocimientos básicos y en profundidad en las siguiente materias:

- Fundamentos de la Ingeniería del Software para realizar el análisis y la ingeniería de requisitos.
- Infraestructuras Virtuales para realizar el conjunto de pruebas y simulaciones.
- Transmisión de Datos y Redes de Computadores para configurar las conexiones y la red a la hora de hacer uso del prototipo.
- Tecnologías Web para poder realizar una web en HTML y darle un estilo propio con CSS.
- Programación Web para poder complementar aún más la materia de Tecnologías Web.
- Diseño de Aplicaciones para Internet para desarrollar una aplicación multiplataforma y que funcione en cualquier ordenador con conexión a internet.

Por último, hay que hacer mención a los importantes e imprescindibles conocimientos obtenidos en otros campos como:

- Astronomía, para entender todo lo que se va buscando en ella.
- Equipos astronómicos, para entender cómo funcionan y los resultados que se esperan de ellos.
- Websocket, para establecer un canal bidireccional entre el cliente y el servidor.
- SublimeText, para poder desarrollar el código fuente del trabajo.
- LaTeX, para poder realizar la documentación del proyecto.
- GitHub, para realizar un control de versiones del proyecto tanto en la parte de desarrollo del código fuente como en la parte de desarrollo de la documentación.

2.3. Alcance de los Objetivos

El prototipo de cliente web debe cumplir con todos los objetivos específicos que más arriba exponemos y que nos hemos planteado ante el reto que supone este trabajo. El que así sea es de relevante importancia dado que actualmente no existe ningún cliente web basado en INDI para controlar un observatorio astronómico.

Así mismo, una vez que se finalice el proyecto, se desea concluir con el prototipo y que pase a ser un cliente web estable y que los profesionales y estudiosos en la materia que nos ocupa puedan hacer uso y disfrute del mismo.

Esta es una de las razones principales por las que se ha desarrollado este prototipo. Se ha desarrollado bajo *Software Libre* con la intención de que cualquier desarrollador pueda continuar con el proyecto bajo los estándares de *Software Libre*.

Por otra parte, se intentará que dicho prototipo sea publicado en la página web oficial de INDI para que los diferentes usuarios puedan verlo, interesarse y contribuir a su desarrollo final.

2.4. Interdependencia de los Objetivos

El objetivo fundamental y que se debe conseguir con este proyecto es el **OBJ-1** ya que es el desarrollo del prototipo de cliente web basado en INDI.

Todos los objetivos planteados tienen entidad propia e independencia en sí mismo, exceptuando los objetivos secundarios **OBJ-S-1** y **OBJ-S-2**, de relevante importancia, puesto que seguir los estándares hará que se optimice notablemente el funcionamiento del prototipo.

Capítulo 3

Planificación

Este proyecto se ha planificado siguiendo los puntos que a continuación se detallan y que permiten trabajarlo de una manera ordenada a la vez que dan información al lector facilitándole una visión amplia y exhaustiva, así como la comprensión de la posterior exposición del mismo. A continuación se detallan los diferentes puntos de la planificación que han sido estudiados para este proyecto

3.1. Metodología de Desarrollo

La metodología utilizada ha sido el modelo iterativo que es el que más y mejor se ajusta a las características del proyecto. La elección de este tipo de metodología se justifica por los siguientes aspectos de la misma:

- Permite un entendimiento incremental del proyecto.
- Habilita una fácil retroalimentación al usuario.
- Dispone de objetivos parciales y metas concretas.
- El proceso es medido conforme avanzan todas las implementaciones.

Además de la utilización del modelo iterativo que por sus características específicas es el más idóneo para este trabajo, hay que mencionar que para el control de versiones del proyecto se ha estado utilizando el software Git y para guardar los datos se ha recurrido al servicio GitHub.

3.2. Fases

En este punto se van a exponer de manera más exhaustiva y se va a entrar en detalle, cada una de las fases que se desarrollan para completar el proyecto viéndose, de esta forma, el progreso del mismo.

Figura 3.1: Modelo Iterativo (http://adsi.foroactivo.com//)

3.2.1. Planteamiento del Problema

- Descripción: Exposición del problema que se desea resolver, pautas a seguir y familiarización con las tecnologías.
- Apartado: Capítulo 1.

3.2.2. Especificaciones del Proyecto

- Descripción: Obtención de los requisitos funcionales y requisitos no funcionales del proyecto.
- Apartado: Capítulo 4.

3.2.3. Planificación

- Descripción: Estimación temporal, estimación de costes económicos y estimación de recursos humanos.
- Apartado: Capítulo 3.

3.2.4. Ingeniería

- Descripción: Análisis de los requisitos y diseño del sistema a desarrollar.
- Apartado: Capítulos 4 y 5.

3.2.5. Construcción

- Descripción: Implementación del proyecto.
- Apartado: Capítulo 5.

3.2.6. Pruebas

- Descripción: Pruebas internas del proyecto
- Apartado: Capítulo 6.

3.3. Estimación Temporal

Seguidamente se hace una estimación temporal del proyecto detallando cada una de las fases de la sección anterior.

- **Planteamiento del Problema:**
  - Descripción de los objetivos a grandes rasgos.
  - Comprender el protocolo INDI.
  - Entender los diferentes instrumentos que intervendrán en el protocolo.
  - Planteamiento de las posibles tecnologías que se usarán en el proyecto.

Estimación: 20 horas.

- **Especificaciones del Proyecto:**
  - Conocer perfectamente las necesidades del usuario.
  - Establecer los objetivos que deben plantearse ante este proyecto.
  - Extracción de los requisitos funcionales.

- Extracción de los requisitos no funcionales.

Estimación: 30 horas.

- **Análisis y Diseño:**
  - Análisis de los requisitos que debe reunir este proyecto.
  - Diagramas del proyecto.
  - Metodología empleada en el desarrollo del proyecto

Estimación: 40 horas.

- **Construcción:**
  - Establecimiento de las herramientas, las plataformas, los diferentes tipos de lenguajes y el *software* que se va a utilizar
  - Creación de la interfaz para conectarse a un servidor INDI concreto mediante su IP y su puerto.
  - Creación de la interfaz para mostrar en una ventana el dispositivo con sus grupos y sus diferentes propiedades.
  - Creación de la interfaz para mostrar, en la pantalla, varias ventanas con diferentes dispositivos conectados al servidor simultáneamente.
  - Posibilidad de cambiar cualquier parámetro de un dispositivo conectado.
  - Posibilidad de enviar información al servidor con los parámetros que hayan sido modificados.

Estimación: 110 horas.

- **Pruebas:**
  - Pruebas de integración.
  - Pruebas de sistema.
  - Pruebas de aceptación.

Estimación: 10 horas.

- **Documentación:**
  - Documentación del código del prototipo del cliente.
  - Documentación del proyecto.
  - Manual de usuario del prototipo.

Estimación: 40 horas.

Tarea a realizar	**Tiempo estimado (horas)**
Planteamiento del problema	20 horas
Especificaciones del proyecto	30 horas
Análisis y diseño	40 horas
Construcción	110 horas
Pruebas	10 horas
Documentación	40 horas
Total	**240 horas**

Cuadro 3.1: Estimación temporal

3.4. Recursos Humanos

Gracias a que se ha elegido en la realización y posterior utilización de este proyecto un *Software Libre*, toda persona que tenga unos conocimientos mínimos en programación, astronomía e INDI, que desee realizar una aportación puede hacerlo en el repositorio del proyecto.

El desarrollo del proyecto ha sido realizado por una única persona y, que en este caso, coincide con el autor del presente documento.

Como para la realización del proyecto, desde su inicio hasta la finalización del mismo, se han utilizado unos 4 meses aproximadamente, se podría decir que el proyecto realizado por una única persona supondría un trabajo desarrollado a lo largo de un periodo de cuatro meses. Por el contrario, si se dispusiera de un equipo formado por cuatro personas, se obtendría el mismo resultado con una duración de un mes.

3.5. Recursos *Software* Reutilizables

Los siguientes recursos hacen mención a diferentes desarrollos software creados por otras empresas o por otros desarrolladores.

Como el objetivo fundamental del proyecto es la creación de un prototipo de cliente web capaz de controlar un observatorio astronómico desde cualquier lugar del planeta donde exista conexión a internet, se tendrán que tener en cuenta durante el desarrollo, todas aquellas herramientas que

permitan realizar el prototipo HTML y aquellas bibliotecas de software libre que se encuentran en internet. Dichas bibliotecas se podrán consultar y utilizar, tanto para añadir funcionalidad al prototipo de cliente, como para realizar los diferentes elementos gráficos del proyecto.

3.6. Estimación de Costes Económicos

En esta sección se va a proceder a la estimación de todos los costes económicos para poder llevar a cabo el proyecto:

3.6.1. Licencias *Software*

Dado que el proyecto se realiza mediante herramientas de desarrollo de *Software libre* no es necesario realizar el pago de ninguna licencia *software*.

3.6.2. Recursos Humanos

Representan la totalidad de personas que forman el equipo y que han contribuido al desarrollo y finalización de este proyecto. El tiempo que se ha empleado para este trabajo arroja un total de 240 horas.

Dado que la ley prohíbe de forma expresa a los Colegios de Profesionales para disponer e informar sobre los honorarios, se tomarán como honorarios a percibir el sueldo medio que recibiría un Graduado en Ingeniería Informática. Partiendo de esta premisa, se hará una estimación de que el precio por hora será de unos 18€. El resultado final que se obtiene analizando el coste económico del equipo de trabajo, en este proyecto, sería la cantidad de 4320€.

3.6.3. Material

Dado que INDI dispone de diferentes simuladores con los que poder realizar todas las pruebas necesarias para comprobar que el prototipo de cliente funciona, el coste total del material ha sido de 0€.

Por otra parte, se puede utilizar una cámara *reflex nikon D3200* con un objetivo 18-55mm valorada en 339,99€según Amazon. [8]

Por último, no se le puede restar al material un elemento imprescindible y de vital importancia para este proyecto como es un ordenador. Gracias al ordenador se podrá llevar a cabo la función de programación, documentación del código, buscar información, etcétera. El precio aproximado es de 450€.

Como conclusión, en cuanto a la estimación de costes económicos referidos a la realización de este proyecto, obtenemos los siguientes resultados:

Gastos	**Coste Estimado en euros**
Licencias *software*	0€
Recursos humanos	4320€
Material	790€
Total	**5110€**

Cuadro 3.2: Estimación de coste económico

3.7. Temporización

Se ha elaborado una estimación temporal acorde a las tareas realizadas en la sección de estimación temporal y atendiendo también al número de días. Para mostrarlo de una forma más sencilla, se ha creado un diagrama de Gantt.

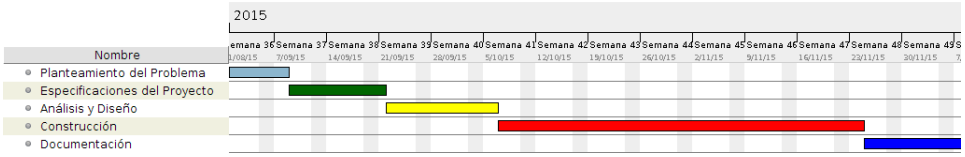

Figura 3.2: Diagrama de Gantt

Capítulo 4

Analisis

En este capitulo se realizará un análisis del proyecto para poder hacer así el futuro diseño del mismo. Se describirán las diferentes partes que se han desarrollado.

4.1. Análisis de Requisitos

El análisis de los requisitos es el conjunto de todas las tareas que formarán las distintas necesidades de un software. Todos estos requisitos se tomarán por parte tanto del cliente como del desarrollador del proyecto ya que tiene que existir una viabilidad en el proyecto.

Se realiza un análisis de requisitos para que al llegar a la fase de diseño se tenga un software óptimo, sin contradicciones o sin ambigüedades, por ejemplo.

Los requisitos se van generando a partir de las diferentes entrevista que se tienen con el cliente.

4.1.1. Requisitos Funcionales

- **RF-1:** Conectarse a un servidor INDI.
- **RF-2:** Mostrar todos los dispositivos conectados en el servidor INDI.
- **RF-3:** Obtener las propiedades de los dispositivos conectados.
- **RF-4:** Agrupar las propiedades de los dispositivos por diferentes grupos.
- **RF-5:** Editar las diferentes propiedades de INDI:
  - **RF-5.1:** Propiedad Text.
  - **RF-5.2:** Propiedad Number.
  - **RF-5.3:** Propiedad Switch.

  - **RF-5.4:** Propiedad Blob.
  - **RF-5.5:** Propiedad Light.

- **RF-6:** Activar la recepción de Blob en cada dispositivo conectado.

- **RF-7:** Desactivar la recepción de Blob en cada dispositivo conectado.

4.1.2. Requisitos No Funcionales

- **RNF-1:** Utilización de licencias libres y herramientas de desarrollo del proyecto de Software Libre para publicar el proyecto como tal.

- **RNF-2:** El sistema solamente podrá conectarse con dispositivos que se encuentren conectados físicamente con el servidor.

- **RNF-3:** Solamente se mostrarán las interfaces de los dispositivos de los cuales se hayan obtenido sus propiedades mediante el fichero XML.

- **RNF-4:** Se deben emplear los estándares de los diferentes lenguajes.

4.2. Descripción de Actores

Los actores son los personajes que participarán en el sistema. Tendrán a su disposición todas las funciones del sistema para poder interactuar con él.

A continuación se detalla el único usuario que tendrá el sistema:

AC-1: Usuario

- **Descripción:** Persona que utilizará el prototipo de cliente realizando con él las distintas funcionalidades que posee.

- **Características:** Es el único usuario del sistema y el que llevará a cabo todas las acciones.

- **Relaciones:** Ninguna.

- **Atributos:** Ninguno.

- **Comentarios:** El usuario tiene la responsabilidad de hacer una buena utilización del prototipo de cliente. Deberá tener, en mayor o menor medida, conocimientos sobre la Astronomía, pues el manejo del prototipo de cliente web se ha diseñado para trabajar con él, tanto a nivel profesional como a nivel *amateur*.

4.3. Descripción de Casos de Uso

Los diferentes casos de uso nos permiten conocer como los actores se desenvuelven en el sistema, es decir, las diferentes funcionalidades que tienen disponibles en el sistema.

4.3.1. CU-1: Conectarse a un servidor

- **Actores:** Usuario (I)
- **Tipo:** Primario, Esencial
- **Precondición:**
- **Postcondición:** Se realiza la conexión con el servidor.
- **Propósito:** Conectarse a un servidor.
- **Resumen:** El usuario se conectará a un servidor mediante su IP y puerto correspondiente.
- **Autor:** Pablo Torrecillas Ortega
- **Versión:** 1.0

Curso normal			
Actor		**Sistema**	
1	Usuario: Introduce la IP y el puerto y pulsa sobre el botón conectar.		
		2	El sistema almacena la IP y el puerto.

Cuadro 4.1: CU-1. Conectarse a un servidor

4.3.2. CU-2: Obtener los diferentes dispositivos

- **Actores:** Usuario(I)
- **Tipo:** Primario, Esencial
- **Precondición:** Debe existir una conexión establecida previamente.
- **Postcondición:** Se mostrarán, en caso de que existan, los dispositivos conectados.

- **Propósito:** Obtener los dispositivos del servidor.
- **Resumen:** El usuario obtiene los diferentes dispositivos que están conectados al servidor en ese momento.
- **Autor:** Pablo Torrecillas Ortega
- **Versión:** 1.0

Curso normal			
Actor		**Sistema**	
1	Usuario: Introduce la IP y el puerto y pulsa sobre el botón conectar.		
		2	El sistema almacena la IP y el puerto. Seguidamante, muestra los dispositivos conectados

Cuadro 4.2: CU-2. Obtener los diferentes dispositivos

4.3.3. CU-3: Finalizar la aplicación

- **Actores:** Usuario (I)
- **Tipo:** Secundario, Esencial
- **Precondición:** Que esté abierta la ventana del prototipo de cliente.
- **Postcondición:** Finalizará la conexión con los dispositivos del servidor.
- **Propósito:** Finalizar la aplicación.
- **Resumen:** El usuario finaliza la aplicación cerrando la conexión con el servidor.
- **Autor:** Pablo Torrecillas Ortega
- **Versión:** 1.0

4.3.4. CU-4: Mostrar un dispositivo

- **Actores:** Usuario (I)
- **Tipo:** Primario, Esencial
- **Precondición:** Que exista una conexión previamente bien realizada.

Curso normal			
Actor		**Sistema**	
1	Usuario: Cierra la ventana del navegador.		
		2	El sistema cierra el navegador de cliente y finaliza la aplicación.

Cuadro 4.3: CU-3. Finalizar la aplicación

- **Postcondición:** Se mostrará la ventana de un dispositivo concreto.
- **Propósito:** Mostrar un dispositivo en el navegador.
- **Resumen:** EL usuario seleccionará un dispositivo concreto del conjunto de ventanas de dispositivos.
- **Autor:** Pablo Torrecillas Ortega
- **Versión:** 1.0

Curso normal			
Actor		**Sistema**	
1	Usuario: Selecciona una ventana del conjunto de ventanas de dispositivos para mostrar todas sus propiedades y trabajar con él.		
		2	El sistema muestra la ventana seleccionada.

Cuadro 4.4: CU-4. Mostrar un dispositivo

4.3.5. CU-5: Editar la propiedad *text*

- **Actores:** Usuario (I)
- **Tipo:** Primario, Esencial
- **Precondición:** Que exista un dispositivo conectado con el servidor y con una propiedad *text* que se pueda modificar.
- **Postcondición:** Se modificará la propiedad *text* y se enviará el valor al servidor.

- **Propósito:** Editar la propiedad *text*.
- **Resumen:** El usuario modificará una propiedad *text* en su correspondiente cuadro de modificación.
- **Autor:** Pablo Torrecillas Ortega
- **Versión:** 1.0

Curso normal			
Actor		**Sistema**	
1	Usuario: Pulsa el cuadro de modificación del texto.		
2	Usuario: Escribe el valor de texto a modificar y pulsa el botón actualizar		
		3	El sistema cambia el valor del texto y lo envía al servidor.

Cuadro 4.5: CU-5. Editar la propiedad *text*

4.3.6. CU-6: Editar la propiedad *number*

- **Actores:** Usuario (I)
- **Tipo:** Primario, Esencial
- **Precondición:** Que exista un dispositivo conectado con el servidor y con una propiedad *number* que se pueda modificar.
- **Postcondición:** Se modificará la propiedad *number* y se enviará el valor al servidor.
- **Propósito:** Editar la propiedad *number*.
- **Resumen:** El usuario modificará una propiedad *number* en su correspondiente cuadro de modificación.
- **Autor:** Pablo Torrecillas Ortega
- **Versión:** 1.0

Curso normal			
Actor		**Sistema**	
1	Usuario: Pulsa el cuadro de modificación de la propiedad numérica		
2	Usuario: Escribe el valor del número a modificar y pulsa el botón actualizar		
		3	El sistema cambia el valor del número y lo envía al servidor.

Cuadro 4.6: CU-6. Editar la propiedad *number*

4.3.7. CU-7: Editar la propiedad *switch*

- **Actores:** Usuario (I)
- **Tipo:** Primario, Esencial
- **Precondición:** Que exista un dispositivo conectado con el servidor y con una propiedad *switch* que se pueda modificar.
- **Postcondición:** Se modificará la propiedad *switch* y se enviará el valor al servidor.
- **Propósito:** Editar la propiedad *switch*.
- **Resumen:** El usuario modificará una propiedad *switch* en su correspondiente cuadro de modificación.
- **Autor:** Pablo Torrecillas Ortega
- **Versión:** 1.0

4.3.8. CU-8: Editar la propiedad *blob*

- **Actores:** Usuario (I)
- **Tipo:** Primario, Esencial
- **Precondición:** Que exista un dispositivo conectado con el servidor y con una propiedad *blob* que se pueda modificar.
- **Postcondición:** Se modificará la propiedad *blob* y se enviará el valor al servidor.
- **Propósito:** Editar la propiedad *blob*.

Curso normal			
Actor		Sistema	
1	Usuario: Selecciona una o muchas de las opciones dependiendo del tipo de switch que sea.		
		2	El sistema muestra la o las opciones seleccionadas y envía los valores al servidor.

Cuadro 4.7: CU-7. Editar la propiedad *switch*

- **Resumen:** El usuario modificará una propiedad *blob* en su correspondiente cuadro de modificación.
- **Autor:** Pablo Torrecillas Ortega
- **Versión:** 1.0

Curso normal			
Actor		Sistema	
1	Usuario: Selecciona la propiedad que desea desactivar.		
		2	El sistema muestra la propiedad modificada y los valores al servidor.

Cuadro 4.8: CU-8. Editar la propiedad *blob*

4.3.9. CU-9: Activar la recepción de *blob*

- **Actores:** Usuario (I)
- **Tipo:** Secundario, Esencial
- **Precondición:** Que exista un dispositivo conectado con el servidor.
- **Postcondición:** Se activará la recepción de *blob*
- **Propósito:** Activar blob
- **Resumen:** El usuario activará la recepción de *blob* de un dispositivo concreto que por defecto viene desactivada
- **Autor:** Pablo Torrecillas Ortega

- **Versión:** 1.0

Curso normal			
Actor		**Sistema**	
1	Usuario: Pulsa el botón de recepción de *blob* de un dispositivo concreto.		
		2	El sistema deja seleccionado el botón y activa la recepción de blob para el dispositivo.

Cuadro 4.9: CU-9. Activar la recepción de *blob*

4.3.10. CU-10: Desactivar la recepción de *blob*

- **Actores:** Usuario (I)
- **Tipo:** Secundario, Esencial
- **Precondición:** Que exista un dispositivo conectado al servidor y tenga activada la recepción de *blob*.
- **Postcondición:** Se desactivará la recepción de *blob* para ese dispositivo.
- **Propósito:** Desactivar la recepción de *blob*.
- **Resumen:** El usuario desactivará la recepción de *blob* de un dispositivo concreto que tenía activada la recepción.
- **Autor:** Pablo Torrecillas Ortega
- **Versión:** 1.0

4.3.11. CU-11: Guardar un *blob*

- **Actores:** Usuario (I)
- **Tipo:** Secundario, Esencial
- **Precondición:** Que exista al menos un dispositivo conectado en el servidor y tenga activada la recepción de *blob*.
- **Postcondición:** Se almacenará un *blob* en la memoria de la máquina.
- **Propósito:** Guardar un *blob*

Curso normal			
Actor		Sistema	
1	Usuario: Pulsa el botón para desactivar la recepción.		
		2	El sistema quita la selección del botón y desactiva la recepción de blob para el dispositivo.

Cuadro 4.10: CU-10. Desactivar la recepción de *blob*

- **Resumen:** El usuario guardará un *blob* en la memoria para poder hacer uso de él.
- **Autor:** Pablo Torrecillas Ortega
- **Versión:** 1.0

Curso normal			
Actor		Sistema	
1	Usuario: Pulsa el botón para guardar un *blob*.		
		2	El sistema almacena la información proporcionada por el *blob*.

Cuadro 4.11: CU-11. Guardar un *blob*

4.3.12. CU-12: Cambiar de grupo de propiedades

- **Actores:** Usuario (I)
- **Tipo:** Primario, Esencial
- **Precondición:** Que exista un dispositivo conectado al servidor y que tenga al menos dos grupos de propiedades.
- **Postcondición:** Se cambiará la vista de la pestaña según los elementos que tenga.
- **Propósito:** Cambiar de grupo de propiedades.
- **Resumen:** El usuario cambiará de pestaña en la ventana de un dispositivo.

- **Autor:** Pablo Torrecillas Ortega
- **Versión:** 1.0

Curso normal			
Actor		**Sistema**	
1	Usuario: Pulsa el nombre de la pestaña para cambiar de grupo de propiedades.		
		2	El sistema muestra la pestaña seleccionada con el conjunto de propiedades que ésta posee.

Cuadro 4.12: CU-12. Cambiar de grupo de propiedades

4.3.13. CU-13: Cerrar un dispositivo

- **Actores:** Usuario (I)
- **Tipo:** Secundario, Esencial
- **Precondición:** Que exista al menos un dispositivo conectado en el servidor.
- **Postcondición:** Se cerrará el dispositio y no se podrá trabajar con él a menos que se vuelva a realizar la conexión.
- **Propósito:** Cerrar un dispositivo.
- **Resumen:** El usuario cerrará un dispositivo que está conectado con el servidor eliminando la posibilidad de realizar más acciones con él.
- **Autor:** Pablo Torrecillas Ortega
- **Versión:** 1.0

4.4. Diagrama de Casos de Uso

Los diagramas de casos de uso sirven para especificar la comunicación y comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas.
Los diagramas de casos de uso, se utilizan también para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo. [**?**]

Curso normal			
Actor		**Sistema**	
1	Usuario: Pulsa sobre el botón X de la ventana del dispositivo para cerrarlo		
		2	El sistema cierra le ventana del dispositivo que se ha seleccionado.

Cuadro 4.13: CU-13. Cerrar un dispositivo

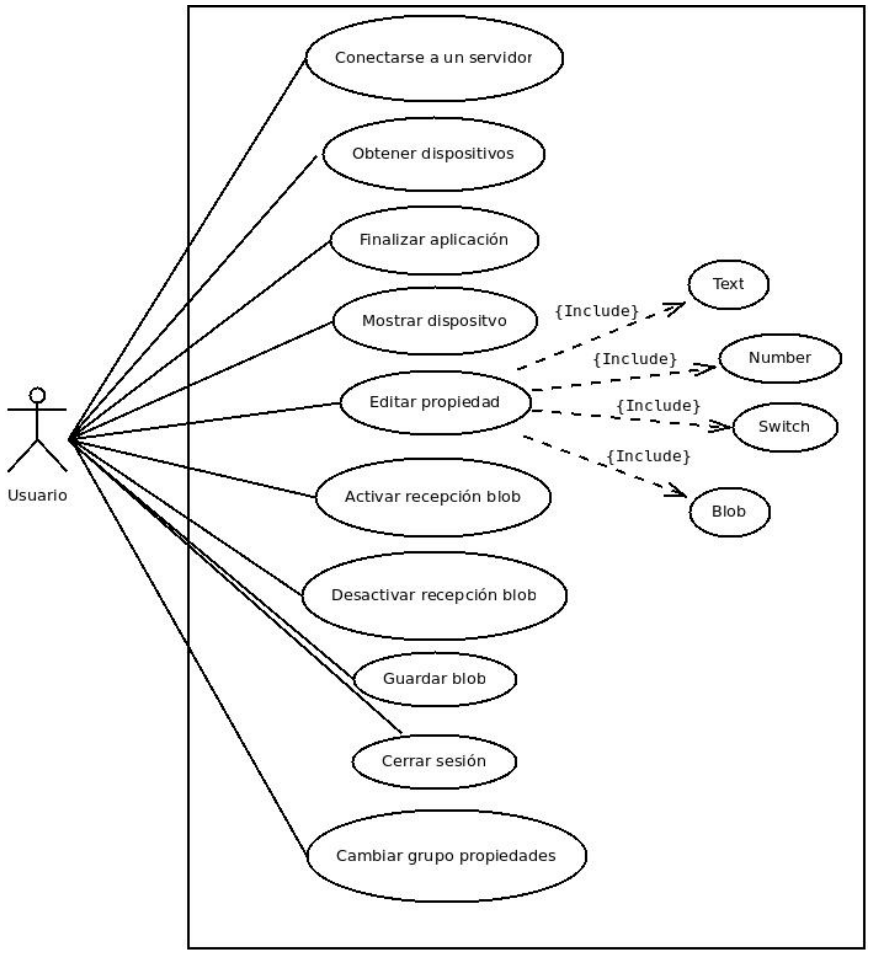

Figura 4.1: Diagrama de Casos de Uso

Capítulo 5

Diseño e Implementación

Este proyecto se realiza atendiendo a dos partes, ambas muy importantes y claramente diferenciadas. Por una parte, se ha creado el diseño del prototipo de cliente y por otra parte se ha elaborado el boceto de la interfaz con su posterior diseño de interfaz de usuario.

5.1. Diseño del Prototipo del Cliente

El diseño del prototipo del cliente es la pieza fundamental del proyecto ya que sobre dicho prototipo giran todas las cosas.

La función principal del prototipo es permitir conectarnos con cualquier servidor INDI y una vez que establecemos la conexión, mediante el WebSocket, es fundamental que la comunicación se establezca de forma continua con el fin de recibir los diferentes dispositivos que se encuentran conectados con el servidor en concreto y así poder intercambiar mensajes con los parámetros que se van modificando en ambas partes.

5.1.1. Métodos en el Fichero *connection.js*

El cometido de estos métodos es establecer la conexión con el servidor. También, en ese archivo, se pueden encontrar los métodos para poder realizar el envío de valor de los parámetros modificados, ya sea de tipo *text*, *number*, *switch* o *blob*, al servidor mediante el *WebSocket*, que se crea en el momento de la conexión.

5.1.2. Métodos en el Fichero *parserDefVector.js*

Este fichero se establece en dos grandes bloques. En primer lugar, se encuentran los métodos mediante los cuales se va creando la interfaz con los datos que se reciben del fichero XML, como pueden ser, entre otros, los nombres de los dispositivos, los nombres de las propiedades, los apelativos de

los grupos de las propiedades, los valores que tienen dichas propiedades, etc. En segundo lugar, se hallan las funciones auxiliares para crear dicha interfaz y que sirven de apoyo a los métodos de análisis y parseo de las propiedades.

Una vez que llegan esos datos se crean las diferentes ventanas que corresponden a cada dispositivo siempre y cuando estos dispositivos no estén ya creados. Tanto para las propiedades de **texto** como para las de **número** se realiza el mismo proceso.

A continuación, se pone una bombilla del color correspondiente según el estado que envíe el fichero XML que proviene del servidor, y el nombre de la propiedad. Este conjunto se ha denominado *propertyBox*. Seguidamente, mediante el parseo del fichero XML, se va recorriendo cada uno de los nodos que lo van formando y para cada uno de ellos se muestra el nombre que viene descrito mediante el campo *label* y su correspondiente valor del nodo. Al conjunto formado por el nombre y el valor se le ha denominado *elementBoxContainer*.

Posteriormente, y según el tipo de permiso que tenga esa propiedad, se pone o no el botón de actualizar para que en el futuro se puedan modificar los campos y poder así enviarle el mensaje al servidor.

Por último, se añade a las diferentes pestañas la propiedad, según el grupo al que pertenece la propiedad que se está analizando.

Para el análisis y posterior parseo de la propiedad ***switch*** se sigue el mismo itinerario con el que se ha procedido en los *text* y en los *number*, con la diferencia de que en la parte del *elementBoxContainer*, se hacen diferentes tipos de llamadas para cada una de las diferentes reglas que pueden contener las propiedades *switch*:

- ***One Of Many:*** se crea un desplegable que siempre obliga al usuario a tener una opción seleccionada en ella.
- ***At Most One:*** se crea un desplegable que permite al usuario si así lo desea o le es factible, solo seleccionar una de todas las opciones que se muestran en él.
- ***Any Of Many:*** se crea un conjunto de opciones que le permitirán al usuario seleccionar todas las que se deseen.

Según el tipo de permiso que tenga esa propiedad, se dispone o no, el botón de actualizar para que en el futuro se puedan modificar los campos y poder enviarle el mensaje al servidor.

Para el análisis y parseo de la propiedad ***blob***, se ha seguido también el mismo procedimiento que en las diferentes propiedades pero en este caso, se ha hecho una pequeña variante en la parte del *elementBoxContainer*. Dentro de ella se muestra, igualmente, el nombre y el valor de la propiedad y además, se añaden dos campos nuevos que son el tamaño del *blob* y su correspondiente formato. De igual forma, en su parte derecha, y siempre dependiendo del tipo de permiso que corresponda, se añade o no el botón de actualizar para enviar los datos al servidor.

En último lugar, hay que decir que en la parte del análisis y su correspondiente parseo sobre la propiedad ***light***, se realiza lo mismo que en las diferentes propiedades salvo en el *elementBoxContainer* que se añade el nombre de la propiedad y el color de la bombilla correspondiente al valor que se envía en ese nodo.

Hay que destacar que esta propiedad es siempre del tipo solo lectura y por tanto, no se agregará nunca un botón ya que no se pueden cambiar los valores que tiene.

Por otra parte, hay que destacar los métodos auxiliares para la creación del interfaz. Estos métodos son de igual o más importancia que los anteriores ya que sin ellos no se podrían crear las ventanas, poner los diferentes tipos de interruptores o incorporar las diferentes pestañas a las ventanas de los dispositivos.

5.1.3. Métodos en el Fichero *parserSetVector.js*

Están constituidos por un total de cinco métodos, uno para cada una de las diferentes propiedades que dispone INDI. La función principal de estos cinco métodos es común a todos y es la de almacenar los diferentes valores que se cambian en el prototipo de cliente para poder enviárselos al servidor. Actualmente, en el prototipo, la función de envío se está realizando sobre los propios botones, sobre los desplegables o sobre los *checkbox* que se han ido creando dinámicamente en el fichero anterior. En esos botones se realiza la llamada que corresponda al envío según la propiedad y se hace todo el envío desde el fichero connection.js como se ha mencionado anteriormente.

5.1.4. Métodos en el Fichero *parser.html*

Es el punto de partida de todo el proceso. Desde ahí se establece la conexión con el servidor y se crea un bucle mediante el cual se va enviando la información a las diferentes funciones según el tipo de propiedad que se esté recibiendo o enviando. También dispone de un conjunto de funciones

auxiliares que son comunes tanto a *parserDefVector.js* como a *parserSetVector.js*. Estas pueden ser, por ejemplo, el obtener el nombre de las funciones o conseguir lograr el tipo de bombilla a poner, entre otras.

5.2. Bocetos de la Interfaz

La interfaz de usuario es el medio con que el usuario puede comunicarse, en este caso, con la computadora mediante su navegador y comprende todos los puntos de contacto entre el usuario y el equipo. Por regla general, suelen ser fáciles de entender y fáciles de interactuar con ellas, aunque en el ámbito de la informática es preferible referirse a ellas como que suelen ser "amigables e intuitivas" porque es complejo a la vez que subjetivo, nombrarlas con el adjetivo "fácil". [**?**]

A lo largo de cada una de las diferentes iteraciones del proceso se han ido mejorando las interfaces ya que se ha ido incrementando el número de elementos y por tanto, cada vez había que hacer la interfaz con un grado de dificultad añadido.

En la primera iteración se elaboró todo el diseño en papel para ir constituyendo un esquema de lo que sería la futura la interfaz de usuario.

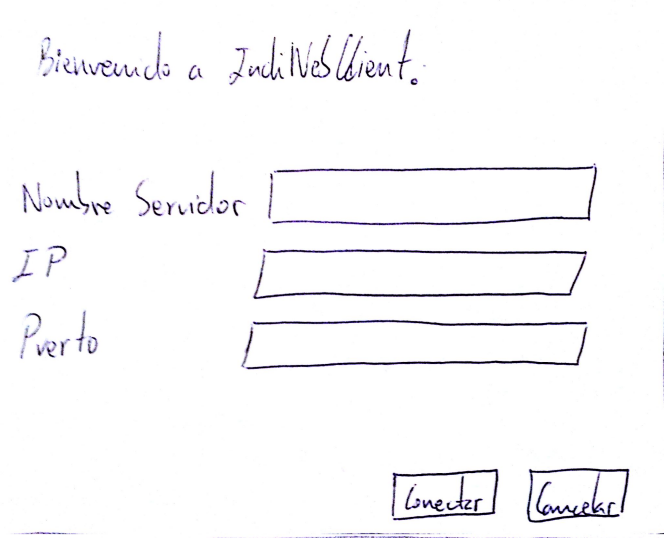

Figura 5.1: Boceto de la Interfaz (I)

Seguidamente, y en las iteraciones posteriores, se comenzó a plasmar el trabajo del papel en el navegador comenzando a crear las primeras ventanas

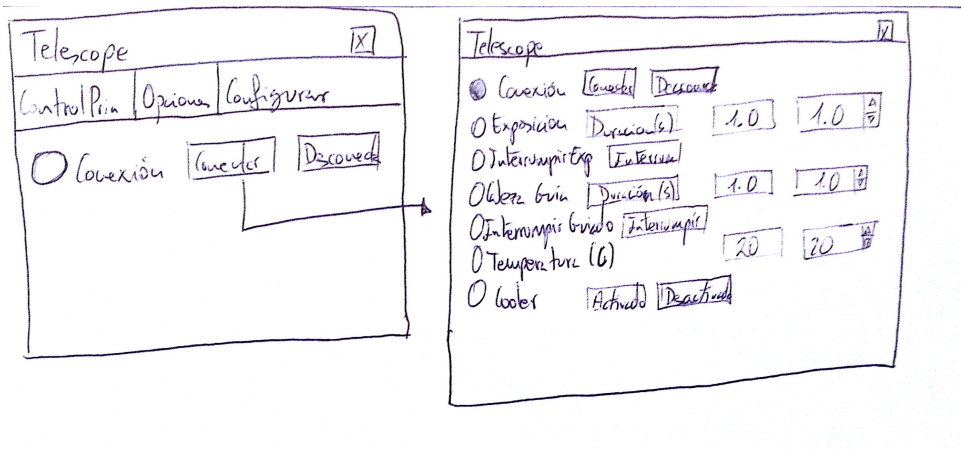

Figura 5.2: Boceto de la Interfaz (II)

para ir proyectando como colocar los diferentes elementos que las iban a integrar.

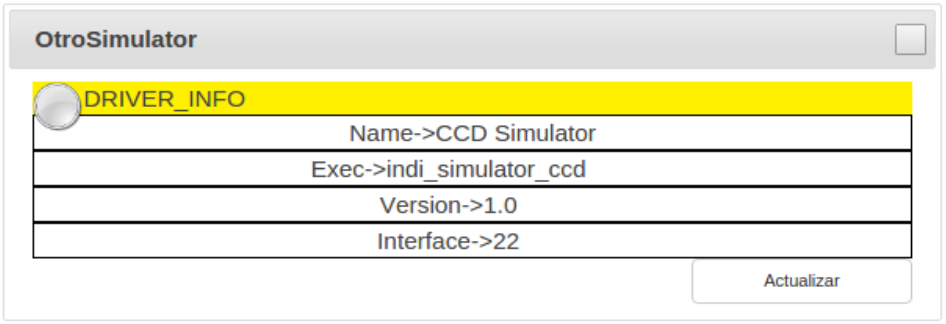

Figura 5.3: Primeras Ventanas de la IU (I)

Posteriormente, se consiguió el diseño final de la interfaz del prototipo del cliente web.

5.3. Diseño de la Interfaz

Para obtener unos mejores resultados, se han ido desarrollando las interfaces por separado:

- Interfaz de conexión.
- Interfaz de propiedad.

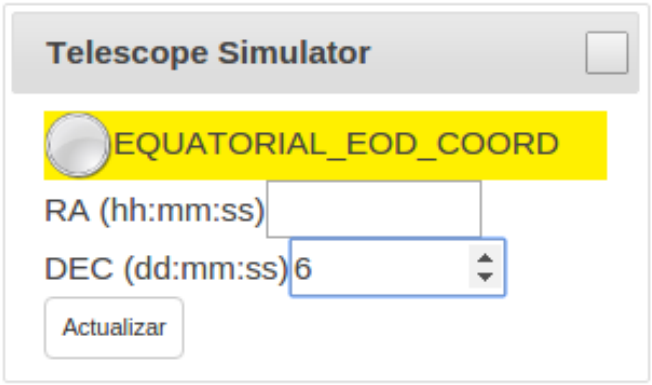

Figura 5.4: Primeras Ventanas de la IU (II)

- Interfaz de grupo de elementos.
- Interfaz de dispositivo.

5.3.1. Interfaz de Conexión

Se define como la pantalla principal del prototipo de cliente. En ella se muestran dos cuadros de texto para introducir la IP y el puerto del servidor con el que se desea conectar.

5.3.2. Interfaz de Propiedad

Es una interfaz que tendrá, aproximadamente para cada propiedad, la misma estructura. En la parte izquierda se observa como aparece la bombilla del estado de la propiedad y seguidamente su nombre. A continuación, se encuentra el nombre y el valor de la propiedad o nombres y valores, dependiendo de si tiene más de uno. Por último se encuentra el botón de actualizar, para el caso de que su permiso lo permita.

Para los *switch* en la parte central, seguido de la bombilla y el nombre, se encontrará el desplegable con las diferentes opciones o casillas para marcar, según el caso de su regla de *switch*, ya sea una u otra.

Para los *blob*, su parte central, estará formada por el nombre y el valor, y además existirán dos campos extra que serán el tamaño y el formato del propio *blob*.

Para los *light*, en su parte central, se encontrará una o más bombillas con su correspondiente nombre cada una.

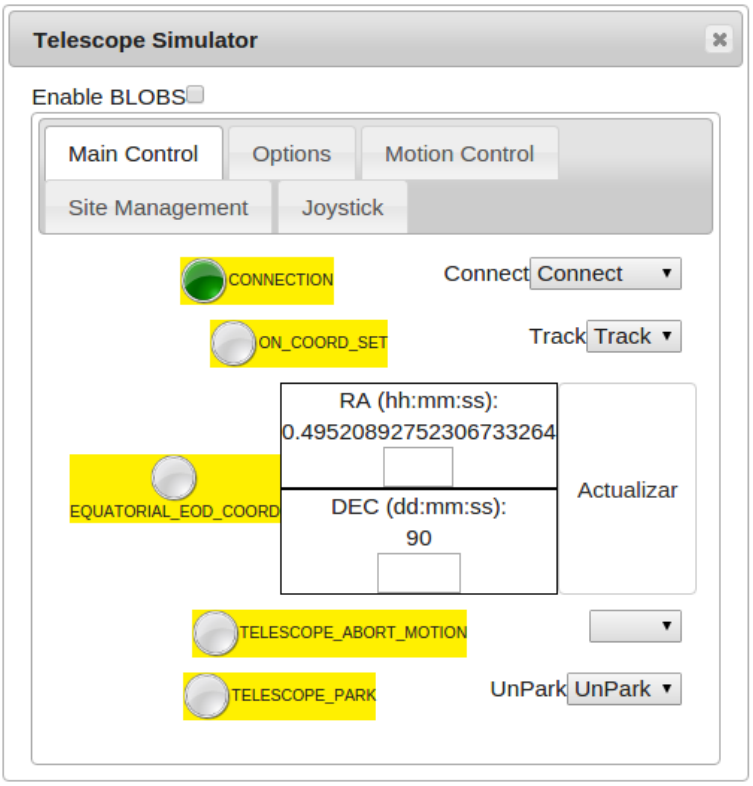

Figura 5.5: Interfaz de Usuario Actual de Dispositivo

5.3.3. Interfaz de Grupos de Propiedades

La interfaz del grupo de propiedades comprende las diferentes pestañas según se agrupan las propiedades. Cuando llega un mensaje en *XML* del servidor se analiza y se comprueba si para el dispositivo que ha enviado el mensaje hay ya creada una ventana, que en caso de no existir, se crea y a continuación el grupo se almacena como una pestaña en la cual irán todas las propiedades de ese grupo. Se comprueba si la pestaña correspondiente a ese grupo existe y en caso de existir, se muestra la propiedad en esa pestaña. En caso contrario, si se da la circunstancia de que no existiera se crea una nueva pestaña y se muestra en ella la propiedad que envía en ese momento el servidor.

5.3.4. Interfaz de Dispositivo

Se entiende por interfaz de dispositivo la ventana encargada de mostrar toda la información del dispositivo conjuntamente con las diferentes interfaces precedentes descritas. En ella se pueden encontrar las diferentes

propiedades de un dispositivo en cada una de las pestañas según correspondan a un grupo o a otro. Esta interfaz es la que se muestra cuando se realiza la conexión con el servidor INDI para cada uno de los dispositivos conectados. Es la interfaz que tiene mayor importancia puesto que en ella se muestra toda la información correspondiente al dispositivo.

5.4. Implementación

5.4.1. Licencia

Una de las principales características de este proyecto es que, desde un primer momento, se pensó en la elección de *Software Libre* como herramienta y soporte para su desarrollo y posterior utilización. Se ha elegido la licencia **GPL v3** por los siguientes motivos:

- Es *Software Libre*.
- Obliga a redistribuir con la misma licencia.
- Está aprobada por *Open Source Initiative*.
- Se compromete y obliga a poder disponer del material de forma ilimitada y gratuita para toda aquella persona que lo solicite.

5.4.2. Desarrollo de Código

Para todo el desarrollo del código se ha utilizado el editor de textos y código Sublime Text [10]. Las razones que hemos tenido para hacer esta elección han sido las siguientes: el tener un tamaño muy reducido, poseer un arranque casi inmediato y poder ampliar su funcionalidad a través de *plugins*.

Este editor de textos y código no es gratuito. Eso es cierto, sin embargo, para el desarrollo de este proyecto que se presenta, he utilizado una versión de prueba que es de tiempo ilimitado. [3]

Por otra parte, y para tener un control de versiones se ha utilizado *Git*, así como para guardar los datos se ha recurrido al servicio *GitHub* [9].

Todo el código fuente se puede encontrar en el repositorio del proyecto.

Capítulo 6

Pruebas

En cada itereación de la planificación se han realizado una mínima parte de pruebas ya que al trabajar sobre un prototipo no se podía efectuar una batería de pruebas puesto que los resultados obtenidos podrían no ser los esperados. Es por ello, que las pruebas realizadas han sido pruebas internas y se ha procedido cuidando mucho los detalles en ellas para no entorpecer el funcionamiento del prototipo.

Las pruebas internas que se han materializado han sido las siguientes:

- Pruebas de unitarias.
- Pruebas de integración.
- Pruebas de sistema.
- Pruebas de aceptación.

6.1. Pruebas Unitarias

Son aquellas pruebas que se han efectuado sobre módulos sencillos, sin estar conectados unos con otros y para verificar el resultado que se obtiene.

6.1.1. Conexión con el servidor INDI

Objetivo de la prueba: Comprobar que la conexión es correcta con el servidor INDI desde el prototipo de cliente que está usándose en el navegador del ordenador.

Fases de la prueba:

- Conexión a un servidor INDI: Se conecta con el servidor INDI y se comprueba que el funcionamiento de la conexión es satisfactoria. El servidor recibe valores esperados y ejecuta correctamente.

- Conexión y finalización del servidor: Se comprueba cómo reacciona el prototipo de cliente cuando el servidor está caído. Se descubrió que el resultado que se obtenía al intentar cambiar algún tipo de parámetro en un dispositivo, era que no se almacenaban los valores ni respondía el servidor.

6.1.2. Propiedades con el Servidor INDI

Objetivo de la prueba: Comprobar que las propiedades funcionan correctamente dentro de los dispositivos con datos erróneos.

Fases de la prueba:

- Datos fuera del rango establecido en la propiedad *number*: Se prueba que el servidor no almacena un dato fuera del rango de una determinada propiedad.
- Datos en formato erróneo en la propiedad textitnumber: Se comprueba que el servidor no modifica datos en otro formato que no sea el que se le indica al poner el cursor sobre la casilla de modificación numérica y, por tanto, no los almacena.

6.1.3. Grupos de Propiedades

Objetivo de la prueba: Poder realizar diferentes pruebas con el conjunto de grupos de propiedades.

Fases de la prueba:

- Mostrar diferentes grupos para un dispositivo: Se comprueba que se crean tantas pestañas como grupos de propiedades existen para un dispositivo.
- Añadir una propiedad nueva a un grupo de nueva creación: Se introduce una nueva pestaña y dentro de ella se almacena dicha propiedad.
- Añadir una propiedad nueva a un grupo previamente establecido: Se añade esa propiedad al grupo que ya está creado y al mismo tiempo se muestran los elementos que ya existen en él.

6.2. Pruebas de Integración

Las pruebas de integración realizadas han sido muy sencillas. En estas pruebas todos los módulos están interconectados unos con otros y se efectúan para comprobar que todas las conexiones están bien practicadas y, por consiguiente, se obtendrán unos resultados satisfactorios. Las diferentes pruebas de integración se han realizado con el servidor INDI sobre un

simulador de telescopio. Para ello se han elaborado dos tipos de pruebas: Pruebas de conexión con el servidor y pruebas de manejo de las propiedades del dispositivo.

6.2.1. Pruebas de conexión con el servidor

Objetivo de la prueba: Conseguir que la función principal de esta prueba sea que el prototipo de cliente se conecte con el servidor sin ningún tipo de problema.

Fases de la prueba:

- Se realiza la conexión con el servidor y a su vez con el simulador del telescopio. Como resultado se obtiene que se muestre el telescopio con todas sus propiedades y con sus grupos de propiedades. Al apagar el servidor, se comprueba cómo no se obtienen resultados por parte de éste y el envío de cambio de valores no se realiza puesto que no hay conexión. No se muestran mensajes si se aborta o falla la conexión.

6.2.2. Pruebas de Manejo de las Propiedades del Dispositivo

Objetivo de la prueba: Comprobar que todas las propiedades del dispositivo funcionan correctamente cuando se envían parámetros de las diferentes propiedades.

Fases de la prueba: Para cada propiedad, se cambian algunos parámetros y se comprueba que el envío al servidor es satisfactorio y éste lo recepciona y modifica los parámetros enviados.

- Propiedad *Text*: Se cambia el texto de la propiedad y se comprueba que se realizó el cambio adecuadamente.

- Propiedad *Number*: Se cambia el número por valores que sean correctos y se comprueba que el cambio ha sido positivo. Por otra parte, también se realizan dos comprobaciones más, una para el valor numérico fuera de rango y otra para el valor numérico con un formato incorrecto. Para ambas pruebas, se puede apreciar cómo no se envían los parámetros al servidor y cómo no se modifican dichos parámetros en el prototipo del cliente ya que no los trata como números correctos.

- Propiedad *Switch*: Se sustituirán los parámetros de los diferentes *switch* que de los que dispone el simulador del telescopio. Se puede apreciar, claramente, que no es posible introducirle valores erróneos ya que si la propiedad es, por ejemplo, de la regla de *al menos uno*, siempre tendremos que seleccionar uno y no será posible dejarlo en blanco. Para las propiedades con la regla *uno de muchos* o *cualquiera de muchos*, tampoco es posible forzar a un error.

Como resultado se comprueba que la propiedad *switch* funciona correctamente en todos los aspectos.

- Propiedad *Blob*: Se verifica que la recepción de *blob* ha sido satisfactoria. No se muestra nada sobre ella, tan solo que se puede guardar.

Como **nota importante**, hay que destacar que para la **propiedad light** no se realizan pruebas de integración ya que es una propiedad del tipo solo lectura y no se puede interactuar con ella. Por supuesto, se ha realizado la prueba de comprobación de que si llega un mensaje XML del servidor INDI con una propiedad *light*, dicha propiedad se muestra correctamente en la ventana del dispositivo del cual proviene.

Bibliografía

[1] Ascom - standars for astronomy. http://www.ascom-standards.org/. Última revisión: 23 de noviembre de 2015.

[2] Astroplanner. http://www.astroplanner.net/requirements.html/. Última revisión: 23 de noviembre de 2015.

[3] Beneficios de sublime text. http://hipertextual.com/archivo/2014/04/sublime-text-vs-brackets/. Última revisión: 7 de noviembre de 2015.

[4] Clientes que usan indi. http://www.indilib.org/about/clients.html/. Última revisión: 24 de noviembre de 2015.

[5] Images plus. http://www.mlunsold.com/ILControl.html/. Última revisión: 23 de noviembre de 2015.

[6] Indi. http://www.indilib.org/. Última revisión: 28 de noviembre de 2015.

[7] Maxim dl. http://www.cyanogen.com/maximmain.php/. Última revisión: 23 de noviembre de 2015.

[8] Nikon d3200 de amazon. http://www.amazon.es/Nikon-D3200-pantalla-estabilizador-importado/dp/B00I3M5U5M/ref=sr_1_1?ie=UTF8&qid=1448363877&sr=8-1&keywords=nikond3200. Última revisión: 24 de noviembre de 2015.

[9] Página web oficial de github. https://github.com/. Última revisión: 7 de noviembre de 2015.

[10] Página web oficial de sublime text. http://www.sublimetext.com/. Última revisión: 7 de noviembre de 2015.

[11] Socket de internet. https://es.wikipedia.org/wiki/Socket_de_Internet. Última revisión: 16 de octubre de 2015.

[12] Websocket de internet. https://es.wikipedia.org/wiki/WebSocket. Última revisión: 17 de octubre de 2015.