**UNIVERSIDAD ANDRÉS BELLO**

**FACULTAD DE INGENIERÍA**

**INGENIERÍA EN COMPUTACIÓN E INFORMÁTICA**



**Robótica Educativa a Distancia**

**GONZALO CONTRERAS CANEO**

PROYECTO DE TÍTULO PARA OPTAR AL TÍTULO

INGENIERO EN COMPUTACIÓN E INFORMÁTICA

VIÑA DEL MAR – CHILE

JULIO 2018

Contenido

Contenido 2

Resumen 8

CAPÍTULO 1 - INTRODUCCIÓN 9

1.1 Identificación del problema 10

1.2 Objetivo general 11

1.3 Objetivos específicos 11

1.4 Diagrama de Ishikawa 12

CAPÍTULO 2 – FUNDAMENTACIÓN DEL TEMA 13

2.1 Alternativas de solución 13

2.2 Factibilidades 14

2.2.1 Factibilidad Económica 14

2.2.2 Factibilidad Técnica 14

2.2.3 Factibilidad Operativa 15

2.2.4 Factibilidad Legal 15

2.3 Descripción de la solución propuesta 15

2.3.1 Esquema de alto nivel 17

CAPÍTULO 3 - METODOLOGÍA 18

3.1 Metodología de Gestión 18

3.1.1 Tabla de tareas a desarrollar 19

3.2 Metodología de Desarrollo 24

3.3 Métricas 25

3.4 Planificación 27

3.5 Plan de Pruebas 29

3.6 Gestión de Configuración 30

3.7 Plan gestión de Cambios 31

3.8 Plan gestión de Riesgos 31

3.9 Ambientes de Desarrollo 33

CAPÍTULO 4 - RESULTADOS 34

4.1 Análisis de información 34

4.3 Pruebas de desarrollador 35

4.3.1 Hitos 1-3 35

4.3.2 Hito 4 35

4.3.3 Hito 5 36

4.3.4 Hito 6 37

4.4 Evidencia Liberación del Producto 39

4.5 Evidencia Gestión de Tickets 40

4.6 Evidencia Gestión de Riesgos 42

4.7 Evidencia Control de Versiones 44

4.8 Evidencia Control de Cambios 45

4.9 Evidencia de Desarrollo 47

CAPÍTULO 5 – CONCLUSIONES 50

5.1 Conclusiones 50

5.2 Problemas abiertos 50

5.3 Trabajo futuro 51

REFERENCIAS 52

APÉNDICE A 53

1.1 Preparación EV3 53

1.1.1 Guía de instalación 53

1.1.2 Elementos necesarios 53

1.1.3 Preparación 54

1.1.4 Archivos 58

1.2 Configuración Cámara IP 59

1.2.1 Creación cuenta 59

1.2.2 Configuración cámara 61

APÉNDICE B 63

1.1 Nuevos tickets surgidos 63

1.3 Análisis de nuevos riesgos 64

1.3 Gestión de Cambios 66

1.4 Desarrollo Remoto 66

1.4.1 Creación Cuenta y Dominio en noip.com 66

1.4.2 Modificación Modem 69

1.5 Modificación de la web 74

1.6 ¿Cómo funciona el proyecto? 76

1.6.1 Situación Previa 76

1.6.2 Situación Actual 77

1.6.3 Comparaciones 79

1.7 Diagrama de Paquetes 80

1.8 Conclusiones 83

**INDICE DE FIGURAS**

Ilustración 1 Ambiente de desarrollo 11

Ilustración 2 Diagrama de Ishikawa 12

Ilustración 3 Diseño explicativo del proceso 17

Ilustración 4 Modelo Iterativo e Incremental 25

Ilustración 5 Arquitectura cliente-servidor de la solución propuesta 35

Ilustración 6 Repositorio utilizado para liberación 39

Ilustración 7 Documentos dentro del servidor de la UNAB 40

Ilustración 8 Archivos y guías necesarias para replicar el proyecto 40

Ilustración 9 Evidencia gestión de tickets 42

Ilustración 10 Evidencia gestión de Riesgos 44

Ilustración 11 Código Python movimientos 48

Ilustración 12 Código HTML con instrucciones 49

[Ilustración A.1 Inicio de EV3 54](#_Toc521328007)

[Ilustración A.2 Iniciar nueva sesión desde mobaxterm 55](#_Toc521328008)

[Ilustración A.3 Inicio de sesión por primera vez 56](#_Toc521328009)

[Ilustración A.4 Acceso a los archivos 57](#_Toc521328010)

[Ilustración A.5 Archivos necesarios dentro del EV3 58](#_Toc521328011)

[Ilustración A.6 Registro de una cámara 60](#_Toc521328012)

[Ilustración A.7 Registro de la cámara en la página 60](#_Toc521328013)

[Ilustración A.8 Datos necesarios para modificar en la cámara 61](#_Toc521328014)

[Ilustración A.9 Interfaz DCS-933L D-Link 62](#_Toc521328015)

[Ilustración B.1 Tickets aparecidos en la etapa final del proyecto 64](#_Toc521328016)

[Ilustración B.2 Riesgos aparecidos 65](#_Toc521328017)

[Ilustración B.3 Página principal noip.com 67](#_Toc521328018)

[Ilustración B.4 Dominios creados 68](#_Toc521328019)

[Ilustración B.5 Creación nombre del dominio 68](#_Toc521328020)

[Ilustración B.6 Hostnames creados en una cuenta 69](#_Toc521328021)

[Ilustración B.7 Página principal configuración modem 70](#_Toc521328022)

[Ilustración B.8 Firewall "activado" en el modem 71](#_Toc521328023)

[Ilustración B.9 EV3 conectado en la red 71](#_Toc521328024)

[Ilustración B.10 Agregar regla al modem a IP 192.168.0.23 72](#_Toc521328025)

[Ilustración B.11 MobaXterm mostrando manejo de otra ubicación 73](#_Toc521328026)

[Ilustración B.12 Nueva página 74](#_Toc521328027)

[Ilustración B.13 Dreamweaver IDE 75](#_Toc521328028)

[Ilustración B.14 Diagrama de Despliegue situación anterior 76](#_Toc521328029)

[Ilustración B.15 Diagrama de Despliegue situación actual 77](#_Toc521328030)

[Ilustración B.16 Diagrama de Paquetes 80](#_Toc521328031)

[Ilustración B.17 Diagrama Paquetes webserver 81](#_Toc521328032)

[Ilustración B.18 Modelo nueva página 82](#_Toc521328033)

**INDICE DE TABLAS**

Tabla 1 Métricas y su método de validación 27

Tabla 2 Gestión de Riesgos 33

Tabla 3 Prueba de desarrollo 1 36

Tabla 4 Prueba de desarrollo 2 36

Tabla 5 Prueba de desarrollo 1 37

Tabla 6 Prueba de desarrollo 1 37

Tabla 7 Prueba de desarrollo 2 38

Tabla 8 Prueba de desarrollo 3 38

Tabla 9 Control de versiones 44

Tabla 10 Control de cambios 47

[Tabla B.1 Mitigación nuevos riesgos 65](#_Toc521328044)

# Resumen

Actualmente existen múltiples kits educacionales destinados a entregar nociones introductorias de programación, junto con fomentar el raciocinio lógico y pensamiento algorítmico mediante plataformas robóticas, en distintos niveles de profundidad y para público de distintas edades, tales como Bee-bot para el público pre-escolar, Lego Mindstorms o VEX Robotics para aquellos alumnos que recién se están introduciendo al área de manera un poco más formal, o múltiples kits que hacen uso de tarjetas de desarrollo como Arduino para aquellos alumnos que desean experimentar la programación autónoma de plataformas móviles mediante la incorporación de código con lenguajes de texto estructurado. A pesar de la alta oferta de este tipo de kits educacionales, el costo de adquisición de éstas aún representa un alto valor para muchos establecimientos y más aún para personas naturales.

En este proyecto se busca dar solución al problema de la baja tasa de utilización de los kits educacionales debido a la poca disponibilidad de éstos por parte de los usuarios quienes no logran tener acceso a ellos ya que las instituciones proveen estos implementos únicamente dentro de sus espacios físicos, el costo y la dificultad que trae la programación, a través de una plataforma web en donde el usuario podrá interactuar con los diversos elementos que el establecimiento posee de forma remota a través de la plataforma web la cual se implementó en donde el usuario se conecta desde una red externa a la dirección publica de una red, redirigiendo la conexión al robot que tiene montado un sistema operativo basado en Linux donde se programa el servidor web y todos los elementos importantes para hacer funcionar los motores del robot con el cual se está trabajando.

# CAPÍTULO 1 - INTRODUCCIÓN

Hoy en día muchos establecimientos a nivel nacional cuentan entre sus métodos de educación didáctica variados elementos tales como kits Lego Mindstorms, Arduinos, Raspberry PI, etc, para facilitar a sus estudiantes el aprendizaje en relación con la programación.

La robótica educativa es una disciplina que tiene por objeto la concepción, creación y puesta en funcionamiento de prototipos robóticos y programas especializados con fines pedagógicos [1]. La robótica educativa crea las mejores condiciones de apropiación de conocimiento que permite a los estudiantes fabricar sus propias representaciones de los fenómenos del mundo que los rodea, facilitando la adquisición de conocimientos acerca de estos fenómenos y su transferencia a diferentes áreas del conocimiento [2].

En Chile, hay una escasa cantidad de empresas las cuales brindan enseñanza sobre la programación de robots. La mayoría de estas se especializan más en la venta y distribución de los diferentes kits que poseen.

NeoEduca es una de las empresas la cual ofrece cursos presenciales para profesores para que estos puedan implementar los conocimientos aprendidos en sus respectivos establecimientos. También cuenta con cursos para jóvenes y niños que estén interesados en aprender a armar, programar y conectar los distintos elementos.

La fundación Mustakis es otra empresa que entre sus ramas se encuentra la robótica educativa la cual cuenta con distintos talleres que se van categorizando según el nivel de dificultad. La empresa ofrece sus cursos presenciales para el aprendizaje de la programación de robots y actualmente cuenta con una gran cantidad de alumnos inscritos para dicha actividad.

La Universidad Andrés Bello no es ajeno a este tipo de enseñanzas por lo que cuenta con diversos kits educacionales los cuales están a plena disposición de alumnos y profesores para su uso dentro del establecimiento.

## Identificación del problema

Los establecimientos cuentan con una cantidad limitada de kits educacionales ya sean Lego Mindstorms, unidades de tarjetas de desarrollo como Arduino, Raspberry Pi, etc., los cuales están a disposición de alumnos y docentes exclusivamente para su uso dentro de la institución.

Los kits legos Mindstorms son unos productos que permiten acercar al alumno de forma lúdica a la construcción de plataformas robóticas móviles en forma de juguetes, fabricado por la empresa LEGO y bajo distintas versiones que además de incluir distintos modelos del microcontrolador a programar, incorporan distintos tipos de sensores y mecanismos de comunicación, donde la característica principal por sobre otro tipo de kits, es que la construcción de la estructura física que soporta al robot construido se logra mediante la unión de piezas y la programación de acciones de forma interactiva. Pueden ser usados para construir un modelo de sistema integrado con partes electromecánicas controladas por computador. Hasta la fecha han existido tres generaciones de dichos robots los cuales son: el bloque RCX, el bloque NXT y el EV3 siendo estos dos últimos los bloques que se van a utilizar la mayor parte del tiempo, para el desarrollo de este proyecto para que sean de acceso para los usuarios finales.

Arduino es una tarjeta de desarrollo que actúa como interfaz entre un microcontrolador ATMEGA y pines de entrada/salida digitales o análogos (señal análoga que se obtiene al usar de forma interna un conversor apropiado), que además es de código abierto basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquier interesado en crear objetos o entornos interactivos. Arduino puede sentir el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante control de luces, motores y otros artefactos [3].

El problema principal encontrado fue que dichos kits no son aprovechados en su totalidad a causa de diversos motivos en los que se encuentran principalmente la distancia de los alumnos respecto a las instituciones que ofrecen estas actividades, debido a su ubicación geográfica, la poca disponibilidad de los docentes o también el escaso espacio físico dentro del establecimiento, ya que se puede aprovechar un espacio para reutilizar por diferentes usuarios, cuyo número se maximiza en el tiempo al considerar la participación remota.

El laboratorio 216 de la facultad de Viña del Mar cuenta por el momento con variados kits, entre los que destacan 1 kit EV3, 3 NXT, varios Arduinos y unos cuantos Raspberry PI, los cuales son usados para los talleres y ramos que las carreras de Ingeniería en Computación e Informática y Civil Informática imparte a sus alumnos. Dichos kits solo pueden ser utilizados dentro de los terrenos de la Universidad, por lo que los alumnos no tienen la posibilidad de acceder en cualquier minuto a estas plataformas para experimentar con su uso y programación.

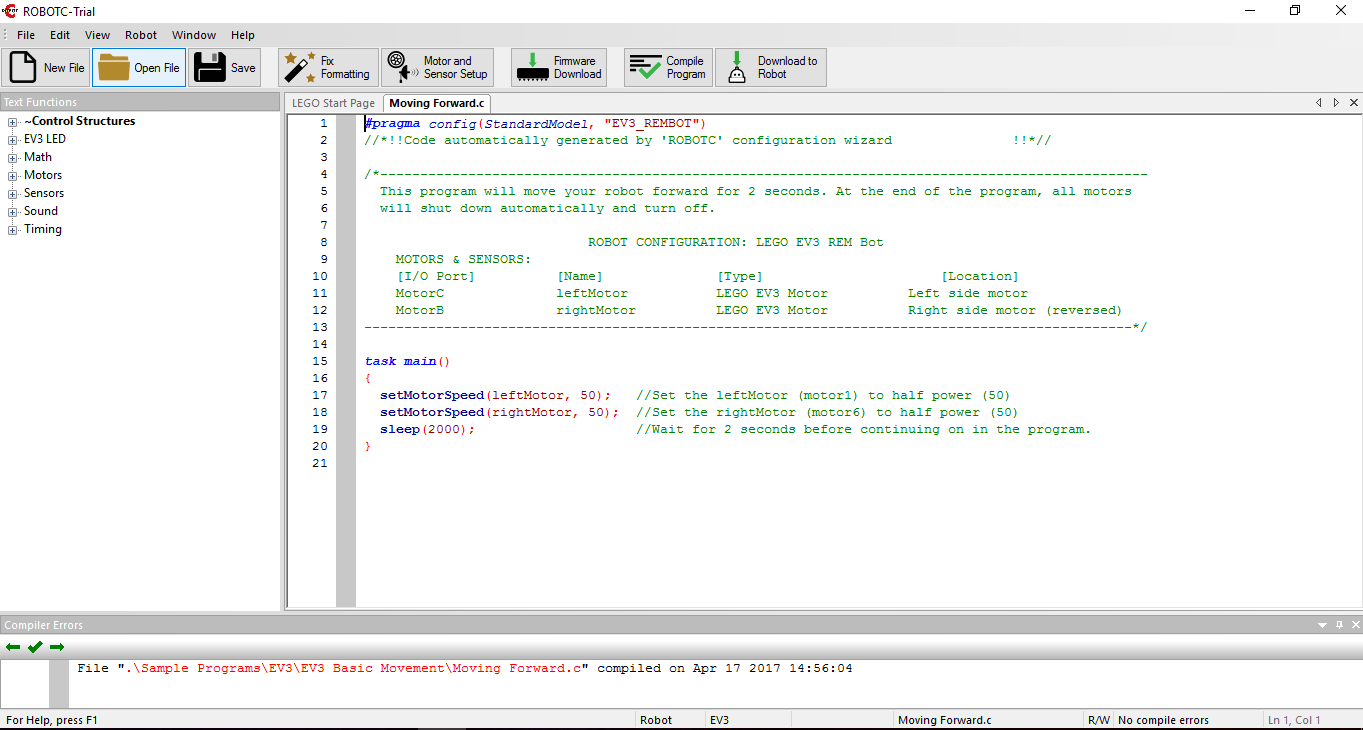


Ilustración 1 Ambiente de desarrollo

## 1.2 Objetivo general

Mejorar el acceso y tasa de ocupación de los kits educacionales mediante su uso de forma remota.

## 1.3 Objetivos específicos

1. Aumentar el acceso y tasa de uso de los kits educacionales.
2. Aumentar la cobertura en términos de alumnos beneficiados debido a disponibilidad temporal y geográfica.
3. Flexibilizar el horario dedicado por el docente para el diseño de las actividades.
4. Diseñar y montar una plataforma virtual la cuál permita a los usuarios (alumnos y docentes) la utilización de los kits educacionales de forma remota.

## 1.4 Diagrama de Ishikawa

La identificación de las diferentes causas que dan origen al problema abordado en este proyecto de título se observa en el diagrama de Ishikawa mostrado en la siguiente ilustración:

C:\Users\Gonzalo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Untitled Diagram (4).png

Ilustración 2 Diagrama de Ishikawa

# CAPÍTULO 2 – FUNDAMENTACIÓN DEL TEMA

## 2.1 Alternativas de solución

En el 2004, el Departamento de Física, Ingeniería en Sistemas y Teoría de la Señal de la Universidad de Alicante, España, ha desarrollado su propia plataforma de un laboratorio de robótica a distancia y la evaluación del impacto en los docentes [4]. El trabajo referenciado consiste en proporcionar al alumno de dos herramientas las cuales son un entorno para la especificación y simulación de algoritmos llamado Visual y de un robot industrial que recibe datos llamado Robolab [5].

Como fue mencionado anteriormente, existen en Chile instituciones las cuales ofrecen talleres de robótica para quienes estén interesados en esta materia. Fueron dos las empresas destacadas que poseen una gran cantidad de alumnos inscritos para sus cursos, las cuales son:

NeoEduca, y sus métodos de enseñanza son y puntos positivos rescatables:

* Sesiones que se basan en el desarrollo de desafíos y experimentos los cuales pueden ser transferibles al aula
* Su programa de capacitación contempla al menos 16 horas cronológicas
* Los talleres están diseñados para niños desde los 6 años gasta los 18 años
* Algunas de las materias que enseñan en el curso son:
  + Algoritmos básicos
  + Armado de robot
  + Desplazamiento del robot
  + Uso de sensores

Fundación Mustakis siendo los puntos positivos más rescatables los siguientes:

* Formato de trabajo contemplan 8 talleres semestrales de un ciclo anual
* Se realizan en 5 sedes universitarias a lo largo de Chile: Universidad de Chile (Santiago), Universidad Técnica Federico Santa María (Valparaíso y Concepción), Universidad Austral (Puerto Montt) y Universidad de la Frontera (Temuco)
* Visitas a diferentes industrias que trabajen con tecnologías de robótica aplicada.

## 2.2 Factibilidades

### 2.2.1 Factibilidad Económica

Para este proyecto no se considerará la factibilidad económica debido a que no se tiene información para que se realice un análisis financiero detallado ni que se realice un movimiento de mercado ya que para realizar una factibilidad económica se debe tener en consideración dos factores importantes que son los costos, tanto fijos como variables y las ventas, siendo este ultimo el factor clave al no poder realizarse este estudio.

Esto se debe a que el proyecto no está planeada su comercialización sino más bien está pensado en ser usado en las instituciones para aumentar el acceso a los kits educacionales que ellos poseen.

### 2.2.2 Factibilidad Técnica

La solución al ser una plataforma web fue desarrollada en los lenguajes de programación JavaScript, HTML y CSS.

Al necesitar la plataforma comunicarse con un robot Lego Mindstorms, fue necesario también contar con un sistema llamado EV3DEV, el cual es un sistema operativo basado en Debian-Linux el cual logra funcionar en variados Lego Mindstorms, para permitir de igual manera la programación en distintos lenguajes, en el caso de este proyecto Python.

De igual manera se necesitó una librería extra descargable para el control remoto del EV3 por medio de un intérprete, llamado RPyC

Todas las herramientas mencionadas anteriormente, al estar disponibles de forma gratuita y disponibles en la web se puede concluir que el proyecto si es factible técnicamente.

### 2.2.3 Factibilidad Operativa

La solución, al ser una aplicación web, no se requiere que se instalen aplicaciones extras a excepción de un navegador web.

También parte de la solución requiere que el usuario tenga instalada una librería en Python para poder comunicarse con el EV3 de forma remota a través de un intérprete, dicha librería requerida es llamada RPyC por sus siglas Remote Python Control.

Para los usuarios finales de este proyecto, no se encontrarán con problemas para operar la plataforma y la instalación de la librería se puede hacer de manera simple, siguiendo el manual de instalación fue subido al repositorio en el cual se está trabajando. Por lo que el proyecto es factible operativamente.

### 2.2.4 Factibilidad Legal

Ya que todas las herramientas que son necesarias para la realización de este proyecto se pueden encontrar de forma gratuita y de libre acceso (OpenSource) sin la necesidad de adquirir licencias de por medio, se puede concluir que es factible legalmente la realización de éste.

## 2.3 Descripción de la solución propuesta

La solución propuesta consistió en desarrollar una página web la cual permita a los alumnos poder manipular los kits educacionales de forma remota y de manera i

interactiva, siendo guiado en la misma plataforma con la utilización de conceptos básicos de la programación de robots.

También se desarrolló una forma de manipular el robot por medio de cualquier ambiente de desarrollo en base a Python la cuál contenga la versión 3 o superior de este.

Uno de los puntos positivos de esta propuesta, es que el alumno no tendría la necesidad de encontrarse en el mismo espacio físico de los kits para poder desarrollar las distintas funcionalidades que estos poseen, además de que los mismos alumnos pueden ver su disponibilidad para realizar las distintas tareas que se encuentran en la web, de la misma manera, es una alternativa que beneficia a los docentes cuyo tiempo es escaso y no pueden realizar más talleres.

Las grandes diferencias entre las alternativas y la propuesta son principalmente:

* La dimensión de los robots, con los cuales se están trabajando en el caso de la Universidad de Alicante, ya que estos sirven para operar objetos de gran tamaño para su transporte de un lado del laboratorio a otro.
* El modo de aprendizaje de los usuarios a quienes está enfocado este proyecto es de forma remota utilizando los robots que los establecimientos poseen, sin la necesidad de estar presencialmente en un taller como son los casos de la Fundación Mustakis y NeoEduca.

## 2.3.1 Esquema de alto nivel

El esquema que se presenta a continuación tiene por objetivo explicar cómo funciona la solución propuesta en este proyecto.

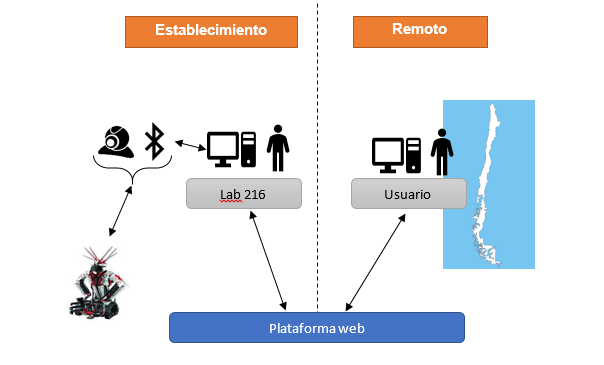


Ilustración 3 Diseño explicativo del proceso

Se tiene a un usuario que se representa al lado derecho en el diseño, el cual será capaz de conectarse independiente de su ubicación geográfica a la plataforma que se desarrolló y la cual hace como servidor el computador del laboratorio 216 de la Universidad Andrés Bello facultad de Viña del Mar. Al momento de conectarse a la plataforma, se tiene la opción de ingresar a un host que contiene las imágenes en vivo de una cámara IP que servirá para que el usuario sea testigo de lo que está realizando. En la misma plataforma va a ser capaz de manipular los motores principales del robot EV3 para que sea capaz de cumplir tareas en las que se requiera utilizar el movimiento de éste.

# CAPÍTULO 3 - METODOLOGÍA

### 3.1 Metodología de Gestión

Este proyecto fue ejecutado utilizando una metodología ágil para la gestión ya que esta permite trabajar de manera flexible y principalmente ayuda a la detección temprana de errores y da lugar a la flexibilidad y adaptación para nuevos requisitos o cambios que se encuentren en estos.

Dentro de lo anterior mencionado, se analizó en profundidad el uso de la metodología llamada Kanban [6], la cuál es una metodología que gestiona el cómo se va llevando a cabo el proyecto, gracias a la utilización de tarjetas que representan las tareas que deben ser realizadas, las que están en desarrollo y las que están finalmente terminadas; todas las actividades van puestas en un tablero. Al utilizar esta metodología es importante limitar la cantidad de trabajo y calcular la cantidad de tareas simultaneas que se están realizando, ya que uno de los principios más importantes son priorizar el trabajo que se está haciendo en vez de cada vez ir agregando más tareas, es decir, no se puede abrir una tarea mientras otra no esté finalizada.

Kanban se basa en una serie de principios que logra que se diferencie con otro tipo de metodologías conocidas como ágiles: Calidad garantizada, reducción del desperdicio, mejora continua y flexibilidad.

Para implementar esta metodología se tienen que tener en cuenta los siguientes aspectos:

* Definir el flujo de trabajo: para esto, debemos simplemente crear nuestro tablero el cual sea accesible para todo el equipo de trabajo, el cual contará con una cantidad de columnas igual a los estados por los que pasa una tarea desde que se inicia hasta que se finaliza.
* Visualizar las fases del ciclo de producción: cuyo objetivo es clarificar al máximo el trabajo que se debe realizar. Para ello se deben dividir las tareas en más pequeñas para así agilizar el proceso de producción, las cuales son pegadas como “post-it” en el tablero.
* Stop Starting, Start Finishing: como mencionado anteriormente, es el principio de terminar las tareas que aún están pendientes antes de empezar unas nuevas para no colapsar en trabajo al equipo.
* Control de Flujo: nos permite realizar un seguimiento del trabajo realizado, almacenando la información que nos proporcionan las tarjetas.

En conclusión, se decidió utilizar Kanban por encima de otras metodologías ágiles, por su flexibilidad en la entrada de tareas, por ser una ayuda en el seguimiento de estas tareas y por la facilidad de ser supervisada por el equipo de trabajo, en este caso, el profesor guía.

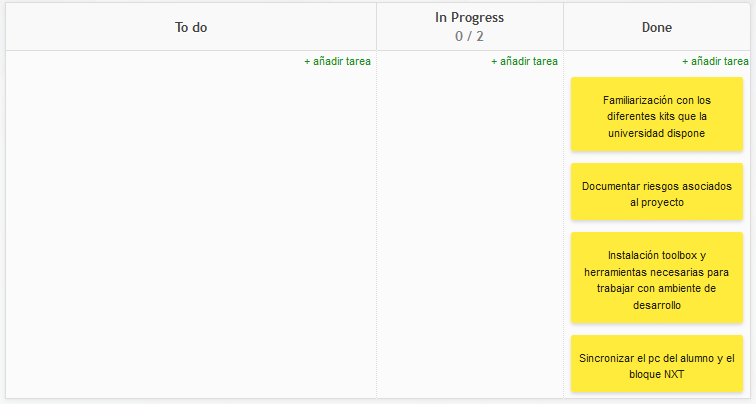
Para el desarrollo de este proyecto usando esta metodología, se utilizó una herramienta online con la que se pueden generar tablas y asignar las tareas correspondientes.

#### 3.1.1 Tabla de tareas a desarrollar

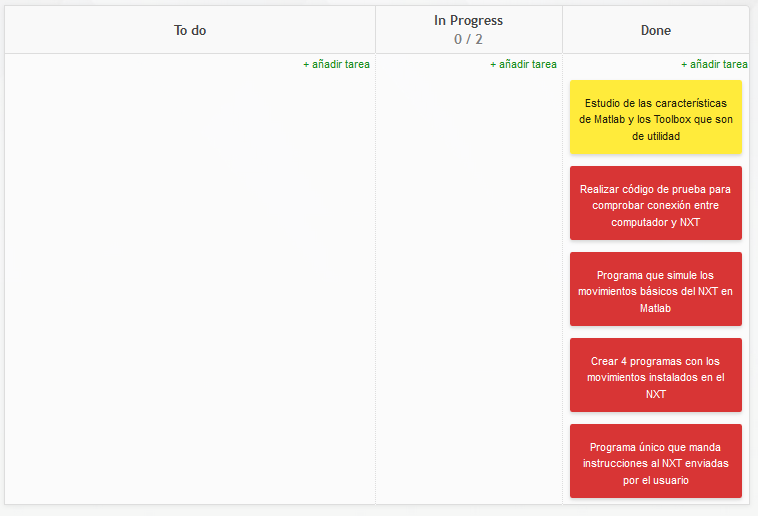
Después de cada reunión, en este caso con el profesor guía, se asignaban una serie de tareas las cuales debían llevarse a cabo y analizar la posibilidad de implementarlas en el proyecto.

Dichas tareas serán presentadas a continuación separadas en los hitos presentados a lo largo de este proyecto:

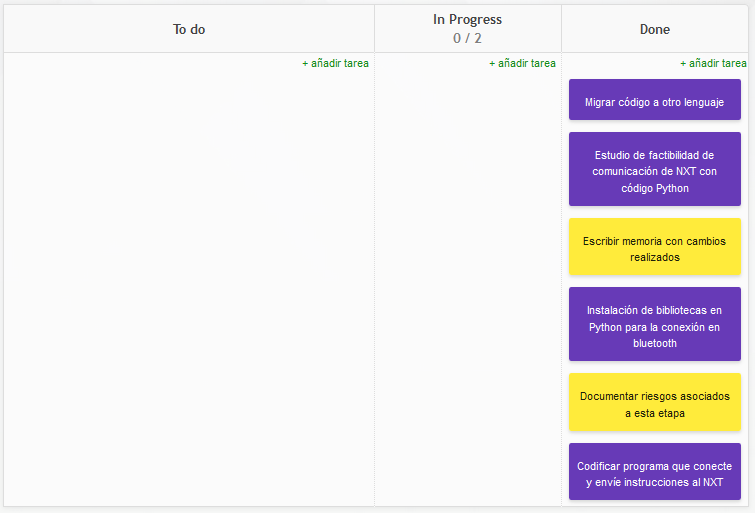
**3.1.1.1 Hito 1**



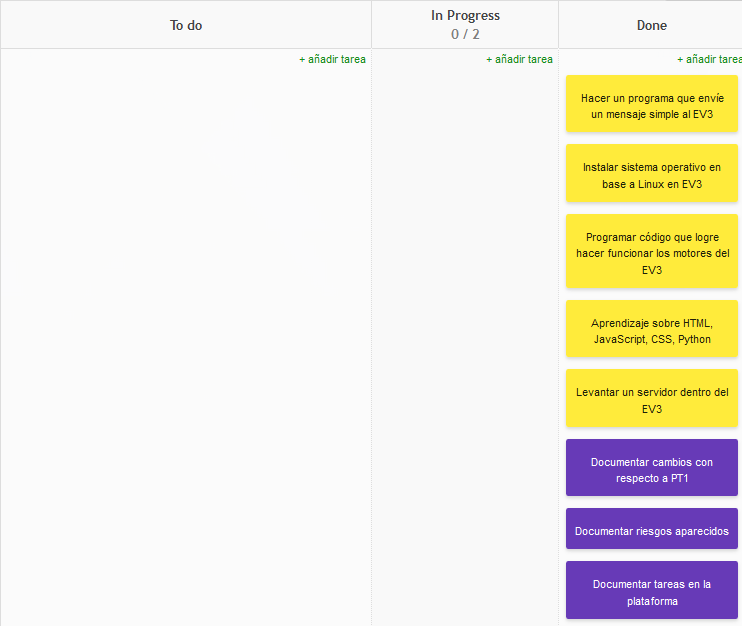
**3.1.1.2 Hito 2**



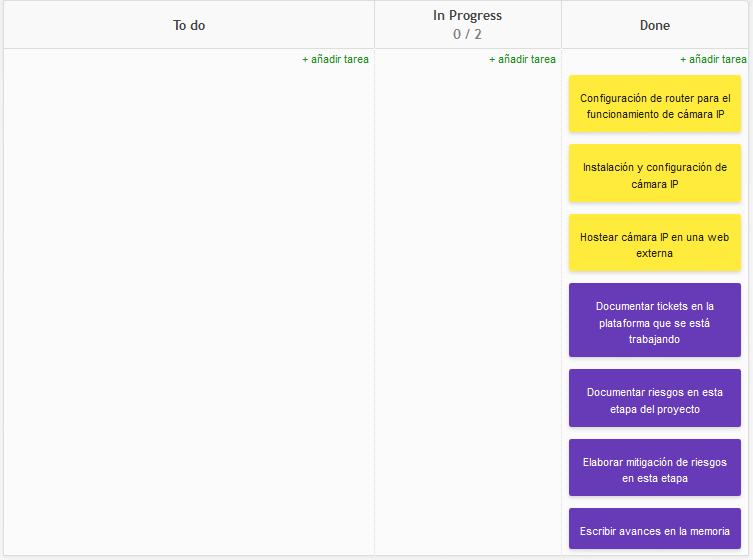
**3.1.1.3 Hito 3**



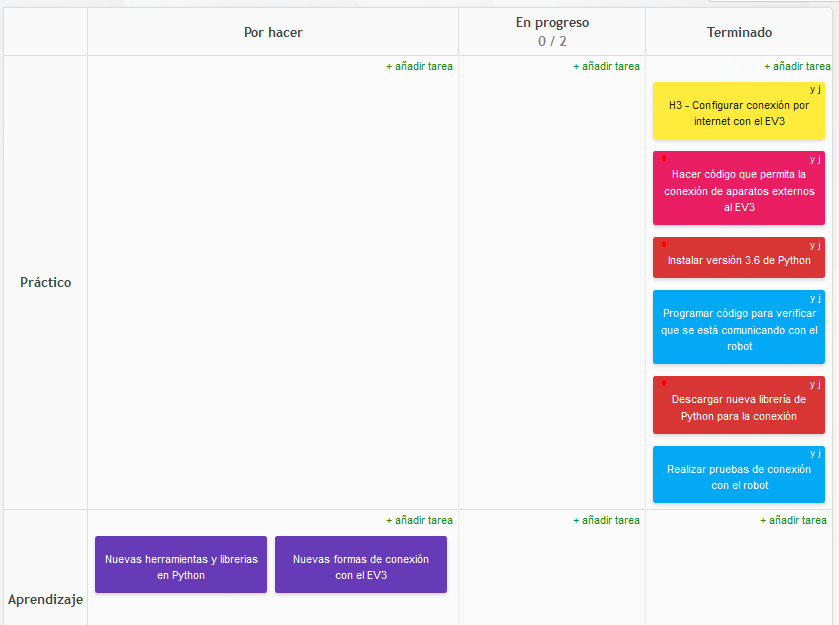
**3.1.1.4 Hito 4**



**3.1.1.5 Hito 5**



**3.1.1.6 Hito 6**



### 3.2 Metodología de Desarrollo

Para la realización de este proyecto se trabajará con la metodología tradicional iterativo e incremental, en donde se divide en varias tareas agrupadas en etapas que se irán repitiendo en cada iteración.

Al terminar el ciclo o iteración de una tarea se hará entrega del producto al cliente el cual efectuará sus comentarios correspondientes de forma de feedback, para así lograr la reducción de riesgos en la ejecución de este. Ante la posibilidad de que el cliente no esté de acuerdo con el producto entregado al finalizar el incremento, se pueden realizar los cambios correspondientes y así lograr que el proyecto avance de acorde con lo que él vea conveniente.

Se decidió trabajar con este tipo de metodología debido a que los requisitos del proyecto fueron dados al comenzarlo y la probabilidad de que cambien durante la ejecución de este es demasiado baja, por lo que se descarta la utilización de una ágil ya que este tipo de metodología es fundamental en un proyecto en donde los requisitos varíen durante la puesta en marcha del desarrollo del producto.

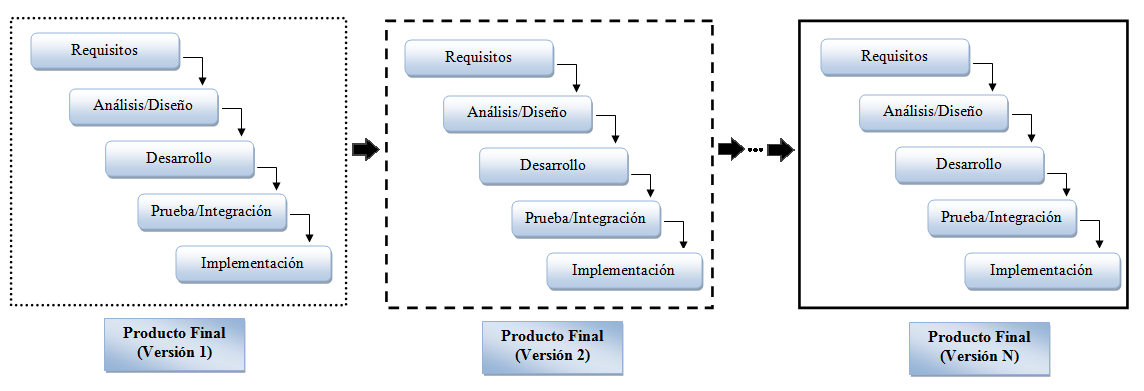


Ilustración 4 Modelo Iterativo e Incremental

## 3.3 Métricas

Para cuantificar las características funcionales del proyecto, se ha decidió utilizar métricas relacionadas con los objetivos específicos para hacer más medible el éxito del proyecto.

OE1: Aumentar el acceso y tasa de uso de los kits educacionales.

**Situación Actual:** solo se puede usar por una persona a la vez.

**Después del Proyecto:** ese mismo se puede utilizar, pero idealmente que al menos dos usuarios se logren conectar al mismo tiempo.

OE2: Aumentar la cobertura en términos de disponibilidad temporal y geográfica.

**Situación Actual:** disponibilidad temporal, los kits educacionales solo pueden ser utilizados dentro de un bloque de clase o taller en específico. Disponibilidad geográfica, solo puede ser utilizado dentro de los terrenos de la universidad.

**Después del Proyecto:** disponibilidad temporal, el kit podrá ser utilizado a cualquier hora durante lo que dure el horario escolar (ejemplo: 10 horas desde que empieza la primera clase de la mañana 8:30 hasta que termina la última clase diurna 19:00). Disponibilidad geográfica, usuarios de cualquier ubicación podrán acceder al kit.

OE3: Flexibilizar el horario dedicado por el docente para el diseño de las actividades.

**Situación Actual:** el docente puede que disponga de 1 hora 30 minutos por cada curso para enseñarles a distintos estudiantes sobre el uso o programación del kit.

**Después del Proyecto:** cada usuario puede aprender por su cuenta durante todo el periodo que el kit esté disponible a partir de una clase de 1 hora 30 minutos previamente diseñada por el profesor.

OE4: Diseñar y montar una plataforma virtual la cuál permita a los usuarios (alumnos y docentes) la utilización de los kits educacionales de forma remota.

**Situación Actual:** solo se puede programar el kit mientras esté conectado mediante USB con el computador.

**Después del Proyecto:** los usuarios son capaces de ingresar a la web independiente de donde se encuentre, para realizar tareas con relación al kit educacional.

|  |  |  |
| --- | --- | --- |
| **Objetivo Específico** | **Métrica** | **Validación** |
| OE1 | Cantidad de kits pedidos por día promedio | El número de pedidos sea mayor a 0 |
| OE2 | Número de regiones conectadas por IP | Que la cantidad de regiones registradas por IP sean mayores que 1 |
| OE3 | Satisfacción de flexibilidad horaria | Encuesta de satisfacción |
| OE4 | De tipo cumple/no cumple | Cumple con lo que se ha desarrollado |

Tabla 1 Métricas y su método de validación

## 3.4 Planificación

La realización del proyecto se divide en 6 hitos que son ejecutados durante un año académico, los cuales se separan en 3 hitos en cada semestre.

Se detallará a continuación lo que se pretende realizar en cada hito para lograr completar el proyecto.

**3.4.1 Hito 1**

Para dar inicio al proyecto, se necesitaba saber con certeza de que una conexión entre el computador y un kit educacional (NXT en un principio) era realmente posible de ejecutar, sin importar de qué manera se hiciera. Para ello se realizó un programa básico el cuál era una simulación de los movimientos básicos de los motores el cual fue programado en el lenguaje C. Programa el cuál fue transferido al NXT a través de un cable USB.

De esta manera se cumplieron las metas que se pretendían como la familiarización de esta nueva tecnología para el alumno encargado del proyecto, y además de que la conexión entre el kit educacional y un computador es realmente posible de establecer.

**3.4.2 Hito 2**

Una vez comprobada la conexión entre el NXT y el computador por USB, se procede a investigar si ahora se puede dar un paso más en relación con la conectividad entre estos dos elementos. Una de las formas de interacción que se encontraron, fue la conexión vía bluetooth entre ambos. Durante este hito se enlazaron el notebook y el kit y luego se procedió a escribir código que requiriera la utilización de los motores y que este programa fuera enviado por bluetooth desde el computador para ser recibido por el NXT para que este último lo ejecutara. Esta vez el lenguaje utilizado para esta etapa fue un lenguaje propio de la herramienta Matlab.

**3.4.3 Hito 3**

Ya avanzados en la conectividad entre el NXT y el computador, se necesitaba un código que fuera capaz de recibir instrucciones de un usuario por mediante el teclado para recrear lo que se vería en la solución final propuesta (la plataforma web). El nuevo código consiste en un menú con instrucciones de cómo quiere que se mueva el robot (avanzar, retroceder, derecha, izquierda y detener) en donde el usuario al presionar una de estas instrucciones, eran enviadas al NXT de forma remota sin la necesidad de una conexión por cables. En este hito fue necesario cambiar el lenguaje de programación propio de Matlab al lenguaje Python.

**3.4.4 Hito 4**

Para la segunda parte del proyecto, en este hito se comenzó el desarrollo de la plataforma web la cual servirá para interactuar entre el kit educacional y el usuario de forma remota.

En esta parte se tuvieron que ejecutar cambios con respecto a la tecnología que se estaba utilizando durante los primeros tres hitos, siendo el NXT reemplazado por su versión más reciente el EV3.

A su vez, fue dejado de lado el sistema operativo con el que se trabajaba anteriormente (en un principio se utilizaba el sistema que venía predeterminado) por un sistema operativo en base a Linux el cuál ayudaba a la detección de adaptadores de redes Wifi, lo que era de vital importancia para la correcta ejecución del proyecto.

Una vez cambiado el sistema operativo, se procedió a crear un web server dentro del mismo EV3.

La conexión entre el EV3 y el computador esta vez será realizada a través de SSH, con la ayuda de un programa llamado MobaXterm.

**3.4.5 Hito 5**

Para esta parte del proyecto, se comenzó con la puesta en funcionamiento de una cámara IP, la cuál será llamada dentro de la web principal en donde se manipula

el robot. La cámara fue subida a un host externo a la plataforma principal para que el usuario pudiera acceder a ella cuando sea requerido.

**3.4.6 Hito 6**

Para la última etapa del proyecto, se puso en funcionamiento una forma alternativa de manipular el kit educacional. Dicha alternativa consiste en la programación por parte del usuario, esto para satisfacer uno de los objetivos específicos que consiste en ser un apoyo para el docente al momento de planificar el diseño de actividades en relación con la programación.

En esta oportunidad, el usuario en vez de manipular directamente el robot será capaz de generar código el cual será enviado directamente a este y lo interpretará al estar correctamente escrito. Todo en base a una posible actividad requerida por el docente en sus clases en línea.

## 3.5 Plan de Pruebas

**3.5.1 Hitos 1-3**

Al inicio de este proyecto, lo que se realizó fue un trabajo de investigación para verificar si era factible la conexión de los diferentes kits educacionales que la Universidad dispone (NXT y EV3). Al ser más un trabajo de investigación, hubo poca programación asociada a esta etapa del proyecto, los códigos realizados durante la ejecución de estos hitos fueron netamente para comprobar que si existía una forma de comunicar los kits educacionales a través de distintos equipos que tengan como requisito mínimo el reconocimiento de aparatos por bluetooth.

**3.5.2 Hitos 4-6**

Para esta parte del proyecto, se programó la plataforma web en donde el usuario será capaz de interactuar con el kit educacional con el que se encuentra trabajando actualmente siendo este el Lego Mindstorms EV3. Dicha plataforma contendrá los botones en donde el usuario al presionar uno, el robot interpretará el

movimiento que le fue enviado. De igual manera fue necesario agregar un enlace en donde el usuario podrá ingresar a imágenes en vivo de lo que está realizando el kit, la cámara estará alojada en un host externo.

También, fue puesto en funcionamiento una herramienta la cual permitirá a los usuarios realizar código en el lenguaje Python escrito desde cualquier ambiente de desarrollo que tenga la versión 3.6 de este lenguaje, el cual será enviado al robot para que sea leído e interpretado por éste.

Se evaluará el comportamiento de las funcionalidades mencionadas en Hitos 4-6 para comprobar el correcto funcionamiento de estas en base a distintas pruebas las cuales serán realizadas por el ejecutor de este proyecto.

## 3.6 Gestión de Configuración

La gestión de la configuración nos permite asegurar la calidad durante todo momento del producto el cual se está trabajando, el cual será llevado a cabo durante toda la ejecución del proyecto hasta que éste se encuentre totalmente finalizado.

Lo que conlleva que se tiene que tener un control del código del proyecto de manera que se pueda acceder de manera rápida a las versiones anteriores en caso de que ocurra algún problema crítico. Para esto puede ser utilizado algunas herramientas de libre acceso para el control de versiones como los son los GIT utilizando el software de, por ejemplo, GitHub, que nos permiten almacenar todas las versiones de los códigos que se están trabajando y en donde se puede acceder a la información desde cualquier ubicación.

La memoria, de la misma manera que los repositorios, es una herramienta muy importante al momento de la gestión de la configuración ya que nos permite guardar la planificación del proyecto y también documentar y dejar en evidencia todo el desarrollo que se ha trabajado como también los cambios que van ocurriendo a medida que el proyecto avance.

## 3.7 Plan gestión de Cambios

El proceso de control de cambios es importante en los proyectos para evaluar si un cambio en la línea base de este genera un impacto muy importante durante su desarrollo.

El proceso será evaluado, en este caso, por el profesor guía y el alumno, siendo el primero quien determinará si el cambio es apropiado y será el encargado de aprobarlo o rechazarlo.

El proceso consta de los siguientes pasos:

* Solicitud de cambio: tanto el alumno como el profesor guía podrán sugerir un cambio en el proyecto si este es beneficioso para la ejecución del proyecto
* Análisis de impacto: se analizará el impacto que el cambio traerá al proyecto en términos de implementación, si retrasa o beneficia el tiempo y si es factible de implementar
* Aprobar o rechazar: finalmente será el profesor guía quien será el encargado de tomar la decisión final si el cambio propuesto por el alumno o él mismo traerá beneficios al proyecto

Registro del cambio: se documentará si el cambio fue aceptado o rechazado por el profesor guía manteniendo un orden y un historial por si la solicitud vuelve ser elevada

## 3.8 Plan gestión de Riesgos

Para llevar a cabo el plan de riesgos, se definieron los posibles riesgos que puede ir ocurriendo a medida que el proyecto se va ejecutando, a los cuales se les asignará una gravedad, a quien irá asignado ese riesgo, un plan de mitigación y los efectos que pueden causar.

Este proceso se mantendrá mientras esté abierto el proyecto, al final de cada iteración se ira evaluando si es necesario modificar los riesgos ya encontrado o si es necesario agregar nuevos que puedan poner el riesgo éste. Gracias a esto se puede llevar un monitoreo de los riesgos para que estos no afecten de manera grave la ejecución del proyecto.

Para esta gestión se utilizó la herramienta Proyeqtor, la cual permite documentar los riesgos que fueron encontrados o los que serán encontrados después de la respectiva reevaluación, para mantener un orden y estar constantemente monitoreándolos.

A continuación, se muestra una tabla con los riesgos definidos anteriormente con el detalle respectivo:

|  |  |  |  |
| --- | --- | --- | --- |
| **Riesgo** | **Efecto** | **Responsable** | **Mitigación** |
| Pérdida de trabajo por el extravío o daño del equipamiento del alumno | Retraso en las entregas semanales y retraso en la construcción del producto | Alumno | Realizar respaldos de todo tipo de avance que se realice al proyecto en algún repositorio o en alguna plataforma de almacenamiento. |
| El pc no soporte la conexión vía bluetooth con el NXT | Retraso en la programación de código, lo que se traduce en atraso general del proyecto | Alumno | Buscar alternativas de conexión entre el computador y el NXT |
| Atraso por desconocimiento técnico del o los lenguajes a utilizar | Retraso general del proyecto | Alumno | Investigar acerca de los lenguajes permitidos en los kits y aprender lo básico en la programación |
| Atraso en las entregas por diversos motivos | Retraso en el proyecto y verse en la obligación de modificar el cronograma | Alumno | Trabajar más horas para compensar el trabajo no realizado |

Tabla 2 Gestión de Riesgos

## 3.9 Ambientes de Desarrollo

Para la realización del proyecto fueron utilizados los siguientes ambientes de trabajo:

Hardware:

* Un computador portátil:
  + Intel(R) Core(TM) i3-3210M CPU @ 2.50 GHz
  + Memoria Física 6 GB
  + Sistema Operativo Windows 10 64 bits

Software:

* Plataformas:
  + RobotC
  + MatLab
  + Python Shell 3.6
  + MobaXterm

Lenguajes de Programación:

* Programación de funcionalidades del robot:
  + C
  + Lenguaje propio de MatLab
  + Python
* Plataforma web:
  + HTML
  + CSS
  + JavaScript

# CAPÍTULO 4 - RESULTADOS

## 4.1 Análisis de información

Se detectó que la dificultad para el desarrollo de esta plataforma web para este proyecto se concentraba, más que nada, en la prueba de los distintos mecanismos de comunicación involucrados entre cada una de las partes (conexión remota desde el pc del usuario al pc servidor y conexión entre el pc servidor y el robot a controlar) y con los kits educacionales en los cuales se está realizando el proyecto. Esto se debe principalmente a que el sistema operativo nativo del kit EV3 solo se puede desarrollar en un lenguaje de programación el cual no permite interactuar en un ambiente web, por lo que se tuvo que investigar de diferentes alternativas para el trabajo a realizar, que en este caso fue necesario instalar un sistema operativo extra al kit el cual está en base a Linux y puede ser programado en varios lenguajes tales como Python, Java, JavaScript, C++, C, etc.

La plataforma constará en la utilización de las “flechas” en donde cada una representará una instrucción para los motores del kit EV3 por lo que se necesita principalmente que el servidor que está dentro del mismo EV3 esté siempre activo mientras se requiera utilizar la plataforma, con la ayuda de un SSH el cual permitirá el levantamiento de dicho servidor. A su vez, se necesita que el kit esté siempre conectado a una red para evitar posibles caídas del servidor, añadiendo que la batería esté con la carga suficiente para evitar de igual manera que se desactive el servidor.

**4.2 Arquitectura**

A continuación, se presenta la arquitectura cliente-servidor de forma detallada para la solución que fue propuesta:

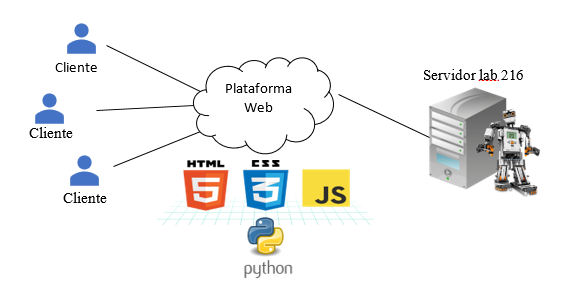


Ilustración 5 Arquitectura cliente-servidor de la solución propuesta

## 4.3 Pruebas de desarrollador

### 4.3.1 Hitos 1-3

Como la programación en esta etapa del proyecto consistió básicamente en la comprobación de que la comunicación era posible, no fue necesario realizar pruebas durante este periodo ya que era más que nada una etapa de investigación y no tendría un impacto relevante en la segunda parte del proyecto. Los programas realizados durante esta etapa no eran los que finalmente iba a ser puestos en funcionamiento en la finalización, por lo que un plan de pruebas y pruebas más detalladas no eran necesarias.

### 4.3.2 Hito 4

|  |  |
| --- | --- |
| **N° Prueba:** 1 | **Nombre:** entrar\_web |
| **Objetivo:**  Al momento de subir el server, se puede acceder a la plataforma de manera correcta | |
| **Pre-requisitos:**  -levantar el servidor | **Datos Entrada:** |
| **Procedimientos:** | |
| **Resultado esperado:**  La plataforma abre de manera correcta al ser levantado el servidor | **Resultado Obtenido:** Aprobado |

Tabla 3 Prueba de desarrollo 1

|  |  |
| --- | --- |
| **N° Prueba:** 2 | **Nombre:** movimientos |
| **Objetivo:**  Las flechas que envían las instrucciones de movimientos de los motores al robot cumplan su función | |
| **Pre-requisitos:**  -entrar a la plataforma | **Datos Entrada:**  -hacer click en las flechas que están en la plataforma |
| **Procedimientos:** | |
| **Resultado esperado:**  El robot, al hacer click en las flechas que simulan el movimiento, efectivamente realiza funciones en sus motores | **Resultado Obtenido:** Aprobado |

Tabla 4 Prueba de desarrollo 2

### 4.3.3 Hito 5

|  |  |
| --- | --- |
| **N° Prueba:** 1 | **Nombre:** estabilidad\_cámara |
| **Objetivo:**  El host de la cámara debe funcionar de manera correcta y está disponible cuando sea requerido al momento que la página principal redireccione al host | |
| **Pre-requisitos:**  -Host de la cámara IP esté disponible | **Datos Entrada:** |
| **Procedimientos:** | |
| **Resultado esperado:**  Al presionar el enlace que contiene el host, este se abre sin problemas | **Resultado Obtenido:** Aprobado |

Tabla 5 Prueba de desarrollo 1

### 4.3.4 Hito 6

|  |  |
| --- | --- |
| **N° Prueba:** 1 | **Nombre:** conexión |
| **Objetivo:**  El robot EV3 acepte la conexión en un ambiente de desarrollo Python 3.6 | |
| **Pre-requisitos:** | **Datos Entrada:**  -Línea de código que realiza la conexión con el EV3 |
| **Procedimientos:** | |
| **Resultado esperado:**  El robot acepta la conexión y se puede acceder a los motores por línea de código | **Resultado Obtenido:** Aprobado |

Tabla 6 Prueba de desarrollo 1

|  |  |
| --- | --- |
| **N° Prueba:** 2 | **Nombre:** movimientos |
| **Objetivo:**  El robot EV3 acepte instrucciones de movimientos básicos programadas en Python 3.6 | |
| **Pre-requisitos:** | **Datos Entrada:**  -Línea de código que realiza movimientos básicos en los motores |
| **Procedimientos:** | |
| **Resultado esperado:**  El robot logró realizar los movimientos básicos que fueron programados y enviados a éste | **Resultado Obtenido:** Aprobado |

Tabla 7 Prueba de desarrollo 2

|  |  |
| --- | --- |
| **N° Prueba:** 3 | **Nombre:** error\_código |
| **Objetivo:**  El robot no se vea afectado al momento de enviar código con errores de sintaxis | |
| **Pre-requisitos:** | **Datos Entrada:**  -Línea de código escrita de forma incorrecta |
| **Procedimientos:** | |
| **Resultado esperado:**  El código no es interpretado por el robot y este a su vez no muestra fallas al momento de ingresar nuevo código | **Resultado Obtenido:** Aprobado |

Tabla 8 Prueba de desarrollo 3

## 4.4 Evidencia Liberación del Producto

Para la liberación del proyecto para ser instalado y utilizado para cualquier usuario, se usó el repositorio Github en donde se encontrarán las versiones del producto con un manual de instalación y los pasos a seguir para su correcta utilización.

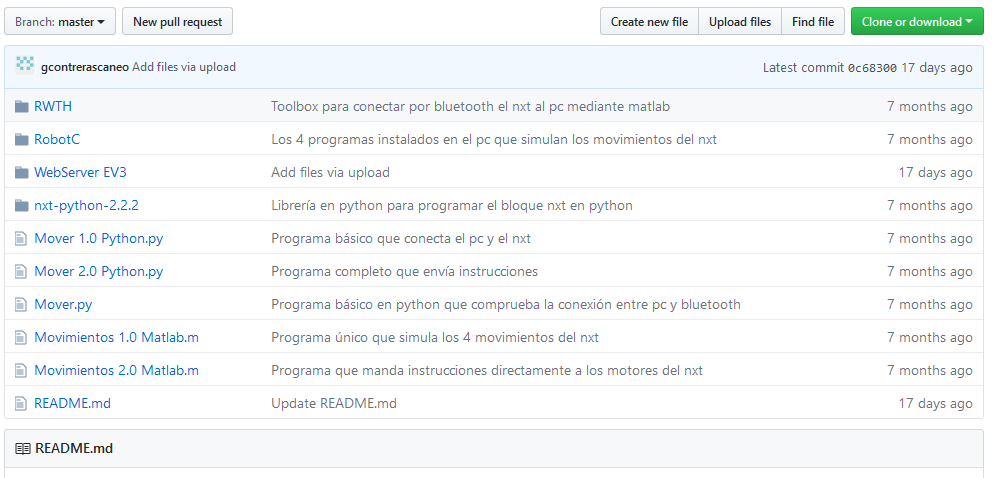


Ilustración 6 Repositorio utilizado para liberación

De igual manera, se liberó el producto en el servidor de la Universidad para que esté disponible para todo aquel quien quiera descargar y poner en marcha todas las funcionalidades que el proyecto en si entrega a sus usuarios.



Ilustración 7 Documentos dentro del servidor de la UNAB

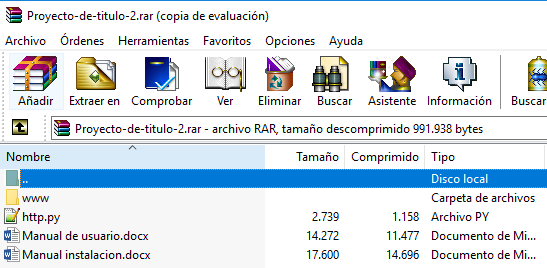
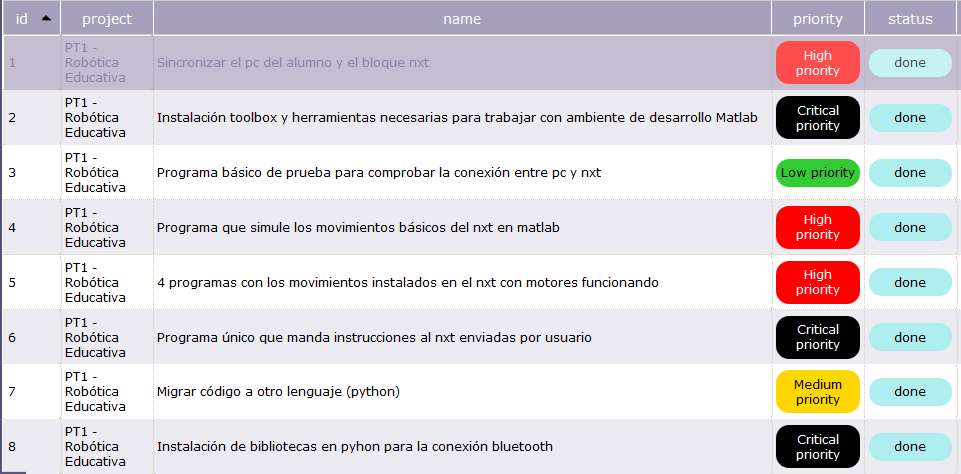


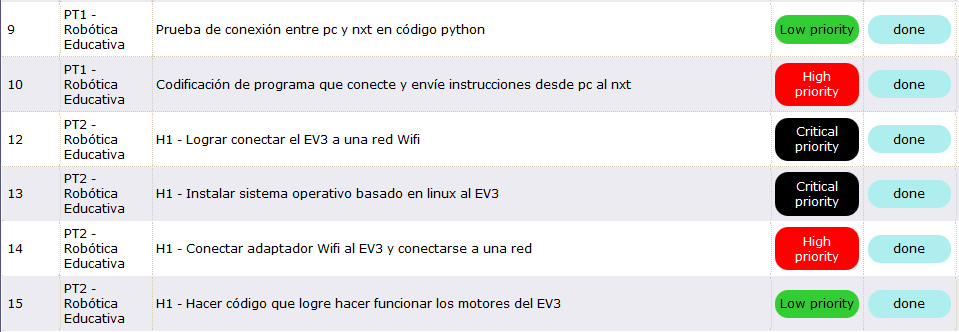
Ilustración 8 Archivos y guías necesarias para replicar el proyecto

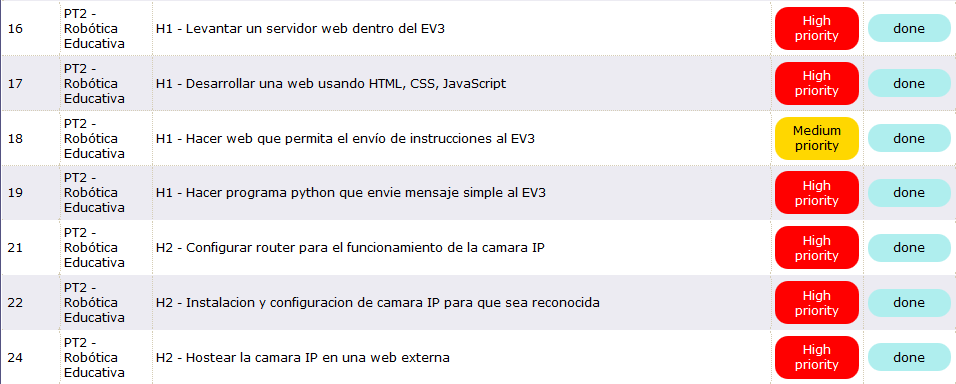
## 4.5 Evidencia Gestión de Tickets

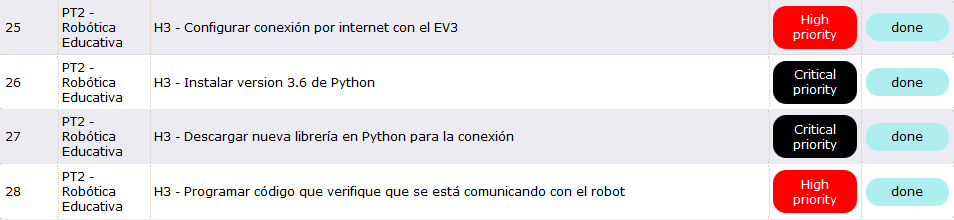
Se utilizó la herramienta Projeqtor para llevar a cabo un orden en las tareas que fueron desempeñadas durante todo el proyecto.

El detalle de las actividades se mostrará en las siguientes ilustraciones:









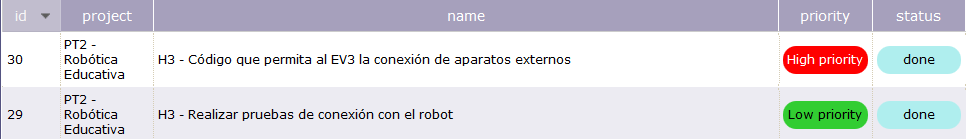
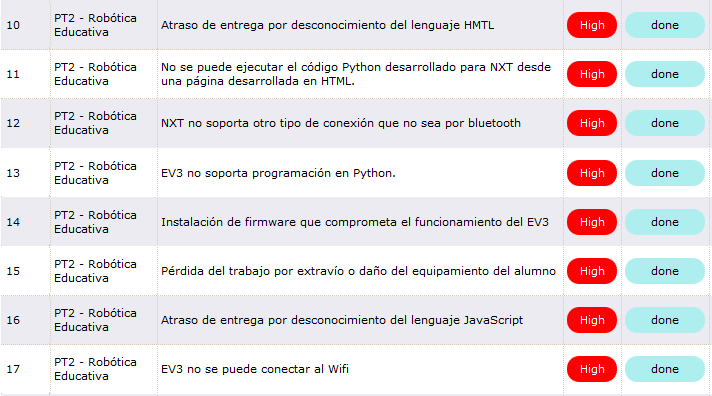


Ilustración 9 Evidencia gestión de tickets

## 4.6 Evidencia Gestión de Riesgos

Para la planificación del proyecto se utilizó la herramienta Projeqtor en donde se documentaron las tareas a cumplir durante el ciclo del proyecto, el plan de pruebas y su respectiva gestión de riesgos los cuales fueron apareciendo a medida que el proyecto avanzaba.





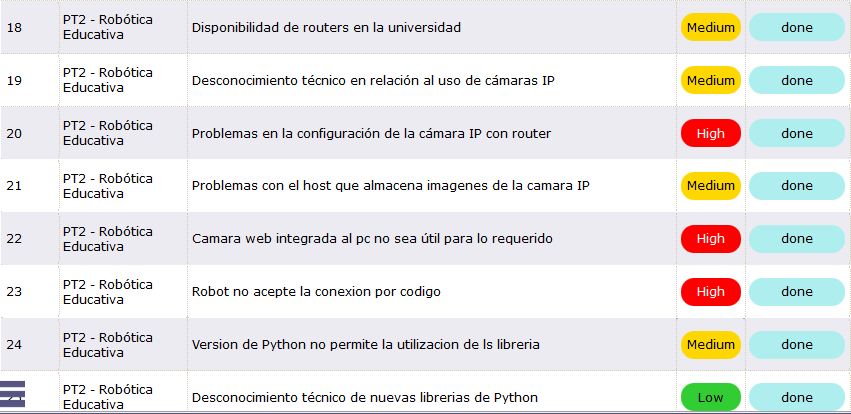


Ilustración 10 Evidencia gestión de Riesgos

## 4.7 Evidencia Control de Versiones

Para la evidencia del control de versiones se realizó una tabla que se detalla a continuación:

**4.7.1 Hitos 1-3**

|  |  |  |
| --- | --- | --- |
| **Versión** | **Función** | **Resultado** |
| Versión 1.0 | Código simple en Matlab que permite la conexión por bluetooth con el NXT | Finalizado |
| Versión 2.0 | Código simple migrado de Matlab a Python que permita la conexión por bluetooth | Finalizado |
| Versión 3.0 | Código operativo donde un usuario puede interactuar con el kit NXT | Finalizado |

Tabla 9 Control de versiones

**4.7.2 Hitos 4-6**

|  |  |  |
| --- | --- | --- |
| **Versión** | **Función** | **Resultado** |
| Versión 3.0 | Código que logre comunicarse con el robot por medio de wifi | Finalizado |
| Versión 4.0 | Código simple que logre hacer funcionar los motores del EV3 | Finalizado |
| Versión 5.0 | Código que pueda ser capaz de levantar un web server | Finalizado |
| Versión 5.1 | Código que pueda ser capaz de levantar un web server y que contenga instrucciones que sean interpretadas por el EV3 | Finalizado |
| Versión 5.2 | Código que pueda ser capaz de levantar un web server, que contenga instrucciones y que pueda tenga un link que redireccione a un host de la cámara IP para que registre movimientos | Finalizado |

## 4.8 Evidencia Control de Cambios

La siguiente tabla muestra los cambios que se efectuaron durante toda la ejecución de este proyecto:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Actual** | **Solicitud** | **Impacto** | **Situación** | **Por** |
| C1 | El lenguaje de programación del robot es C | El lenguaje de programación quiere ser cambiado para optimizar las funcionalidades del robot a Matlab | BAJO | APROBADO | Miguel Solis |
| C2 | El lenguaje de programación es Matlab | El lenguaje propio de Matlab debe ser migrado a uno que sea compatible con la programación web, en este caso se cambió a Python. | ALTO | APROBADO | Miguel Solis |
| C3 | El robot utilizado es un kit NXT | Cambiar el kit utilizado actualmente por un kit EV3 debido a la mayor cantidad de funcionalidades que posee. | MEDIO | APROBADO | Miguel Solis |
| C4 | Cambio de servidor donde se aloja la web | El robot debería alojarse en la facultad, pero al no contar con salidas de redes necesarias se optó por cambiar el lugar del servidor de manera temporal | ALTO | APROBADO | Miguel Solis |

Tabla 10 Control de cambios

## 4.9 Evidencia de Desarrollo

Al desarrollar este proyecto se utilizaron diversas herramientas que ayudaron a que sea posible su finalización de forma correcta. A su vez, fue desarrollado código que permitió que todo fuera posible.

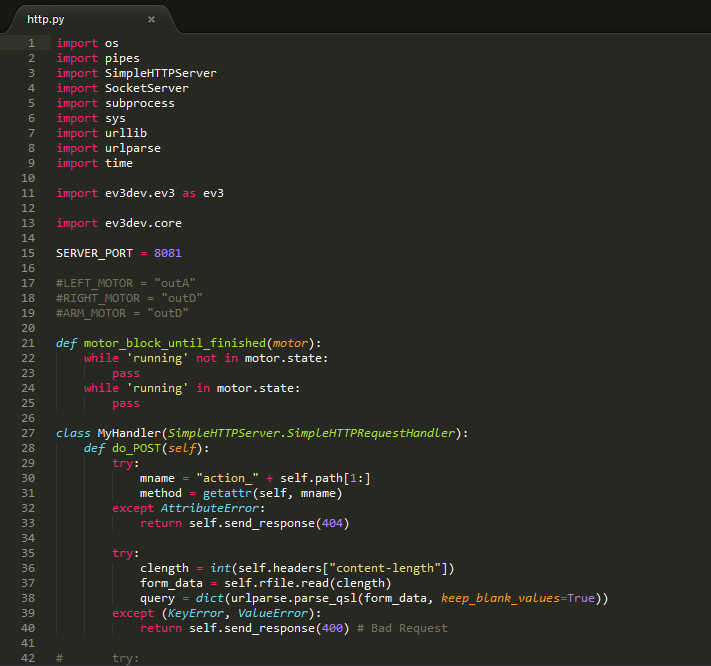


Ilustración 11 Código Python movimientos

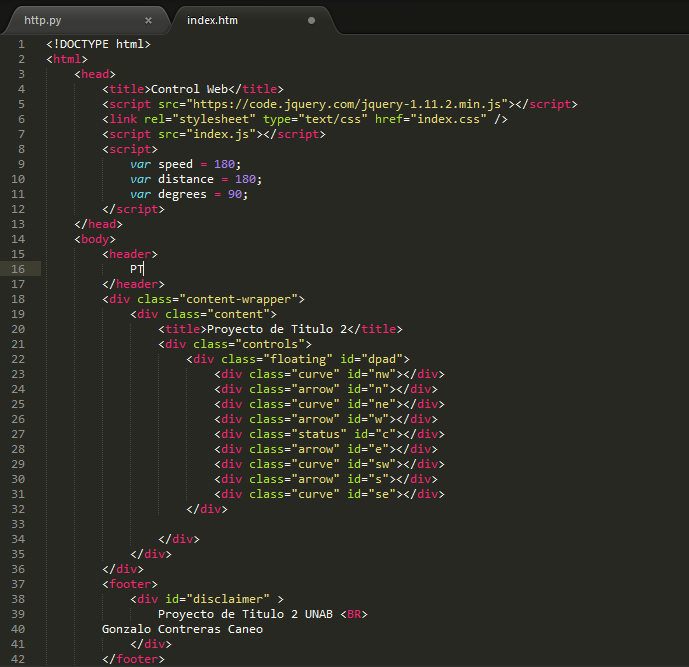


Ilustración 12 Código HTML con instrucciones

# CAPÍTULO 5 – CONCLUSIONES

## 5.1 Conclusiones

Teniendo en consideración los objetivos del proyecto y el trabajo realizado, se procede a analizar el éxito del proyecto.

Con respecto al aumento del acceso y la tasa de uso de los kits, se puede decir que con la funcionalidad de la plataforma web se cumple de manera satisfactoria este objetivo dado que el kit EV3 estará en ejecución en donde el usuario podrá interactuar con este de forma remota.

En general, la plataforma web cumple con tres de los cuatro objetivos específicos que se establecieron al comienzo del proyecto para que este sea declarado un éxito. El usuario es capaz de manipular uno de los kits que la institución dispone sin la necesidad de estar en el mismo espacio físico que estos.

La otra funcionalidad la cual se implementó cumple con el objetivo de flexibilizar el horario dedicado por el docente de manera satisfactoria.

En conclusión, se puede decir que el proyecto en términos de funcionalidades versus objetivos específicos fue un éxito dado a que el trabajo realizado cubre con todos los términos que se daban para que el proyecto pudiera ser aceptado.

## 5.2 Problemas abiertos

Los problemas abiertos en este proyecto tienen relación a la investigación y el de agregar una funcionalidad.

Como parte de la investigación, se debe encontrar una manera alternativa de levantamiento del servidor de la plataforma web en la cual se está trabajando. También a nuevas formas de conexión entre los aparatos computacionales con el kit educacional EV3 utilizando como intermediario, por ejemplo, la tecnología de los minicomputadores también conocidos como Raspberry Pi para el levantamiento de servidores.

En temas de funcionalidad, agregar una forma de programar por parte del usuario sin tener que tener instalada una librería en Python que no es de conocimiento general de docentes o estudiantes, por lo que implementar esta nueva funcionalidad reduciría la traba de los usuarios al momento de utilizar el proyecto en el cual se trabajó.

Se podría considerar el problema con la cámara IP con la cual se está trabajando para capturar en vivo la manipulación por parte del usuario al kit, el problema recae en la calidad de reproducción del video que se ve con una calidad que deja que desear, debido a la antigüedad del aparato que se está utilizando.

## 5.3 Trabajo futuro

Según lo planteado anteriormente como problemas abiertos, se adjunta una lista con el trabajo a futuro a desarrollar para el proyecto:

* Investigación de comunicación entre los kits EV3 y/o NXT con los Raspberry PI para el levantamiento de plataformas web y nuevas formas de interacción entre ellos.
* Trabajar con la plataforma llamada Jupyter, la cual permite la programación por medio de la web en lenguaje Python. Ayudará al usuario a no tener que instalar nuevas librerías en su equipo para poder programar.

# REFERENCIAS

[1] Ruiz, Sánchez, Enrique, Velazquez (2007). [innovación en el aprendizaje de las ciencias y la tecnología](https://dialnet.unirioja.es/servlet/libro?codigo=638909). Madrid: Díaz de Santos. p. <https://dialnet.unirioja.es/servlet/libro?codigo=638909>. [ISBN](https://es.wikipedia.org/wiki/ISBN) [978-84-7978-822-3](https://es.wikipedia.org/wiki/Especial:FuentesDeLibros/978-84-7978-822-3). Consultado el 30 de mayo de 2018.

[2] Sánchez, Flor Ángela Bravo; Guzmán, Alejandro Forero (2012). «La robótica como un recurso para facilitar el aprendizaje y desarrollo de competencias generales». *Education in the Knowledge Society (EKS)* **13** (2): 120-136. ISSN 2444-8729.

[3] Herrador, R. E. (2009). Guia de usuario de arduino. Universidad de Córdoba, 8.

[4]http://www.disclab.ua.es/robolab/labvir.htm

[5] Torres, F., Puente, S. T., Pomares, J., Candelas, F. A., & Ortiz, F. G. (2001). Robolab: Laboratorio virtual de robótica básica a través de Internet. *Primeras Jornadas de Trabajo sobre Enseñanza Vía Internet/Web de la Ingeniería de Sistemas y Automática (EIWISA'01), Madrid*.

[6] Japan Management Association (1989). Kanban. New York: Routledge.

# APÉNDICE A

## Preparación EV3

### Guía de instalación

En el siguiente apartado se explicarán los elementos necesarios para la correcta instalación y puesta en funcionamiento del servidor cuando sea posible tener acceso a una red que permita una comunicación desde el exterior.

### 1.1.2 Elementos necesarios

Para la instalación y puesta en funcionamiento pensando en la obtención de un EV3 totalmente nuevo, son necesarios distintos softwares y hardware para que sea posible el funcionamiento del proyecto, los cuales son:

**Hardware**

* Una memoria microSD de no más de 16 GB
* Un adaptador Wifi compatible con el EV3 (el usado en el proyecto es un TL-WN722N TP-Link)
* Un computador con acceso a internet

**Software**

* Sistema operativo ev3dev
* Win32 Disk Imager
* Mobaxterm

### 1.1.3 Preparación

Primero se debe descargar la última versión del sistema operativo en base a Debian (una distribución de Linux). Esta se puede encontrar en la página oficial ev3dev.org en su sección de descargas.

Una vez descargado, se debe usar el win32 Disk Manager (o cualquier herramienta que permita grabar imágenes) y grabar el sistema en la tarjeta microSD.

Al terminar el proceso de grabación, se procede a insertar el adaptador wifi y la tarjeta microSD en las ranuras que posee el EV3.

Ahora esperar a que el sistema operativo cargue dentro del EV3.

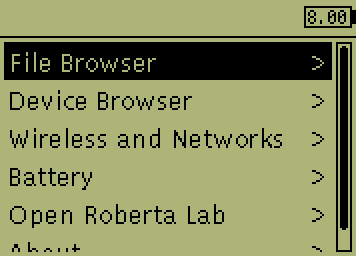


Ilustración A.1 Inicio de EV3

Cuando el sistema operativo cargue de forma satisfactoria, se verá una pantalla de inicio como la de la ilustración A.1

Como se ve en la ilustración A.1 se tiene que acceder a Wireless and Networks para (si el EV3 está totalmente nuevo) habilitar la señal wifi, y proceder a conectarlo al internet.

Una vez conectado el EV3 en la red, éste nos dará una dirección IP con la cual podremos acceder con la ayuda del programa SSH.

Al tener todo preparado, es necesario ahora contar con la herramienta mobaxterm que como fue explicado anteriormente, es una plataforma que permite usar comandos Linux en el sistema operativo Windows, que cuenta con la creación de sesiones SSH la cual es vital para la comunicación entre el computador y el EV3.

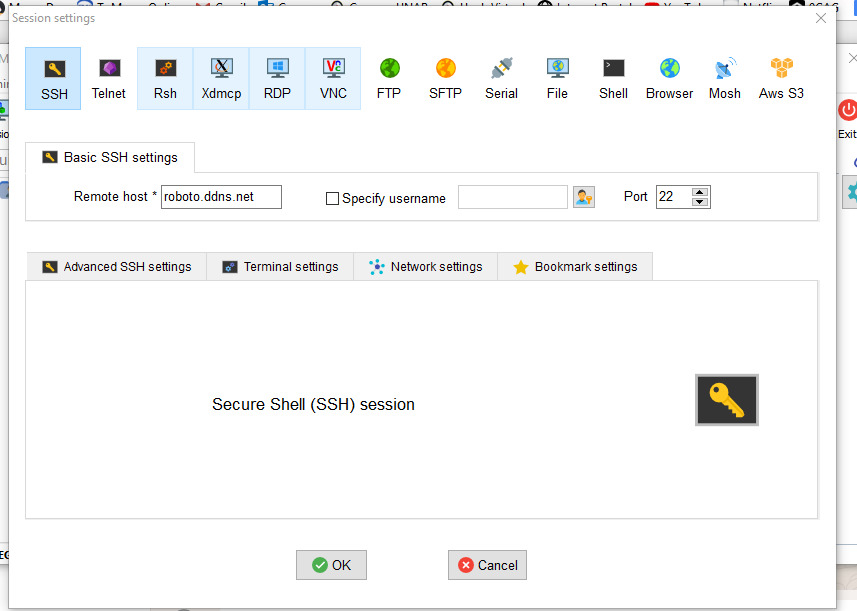


Ilustración A. Iniciar nueva sesión desde mobaxterm

Al presionar en el botón Session hacer clic donde dice SSH, luego completar el campo Remote Host con la IP mostrada en el robot (para que funcione de manera local) o el nombre de dominio que está guardado la IP pública del modem (en el caso de este proyecto roboto.ddns.net). Al iniciar por primera vez pedirá lo siguiente en la terminal del proyecto:

* Login as: robot
* [robot@roboto.ddns.net’s](mailto:robot@roboto.ddns.net's) password: maker

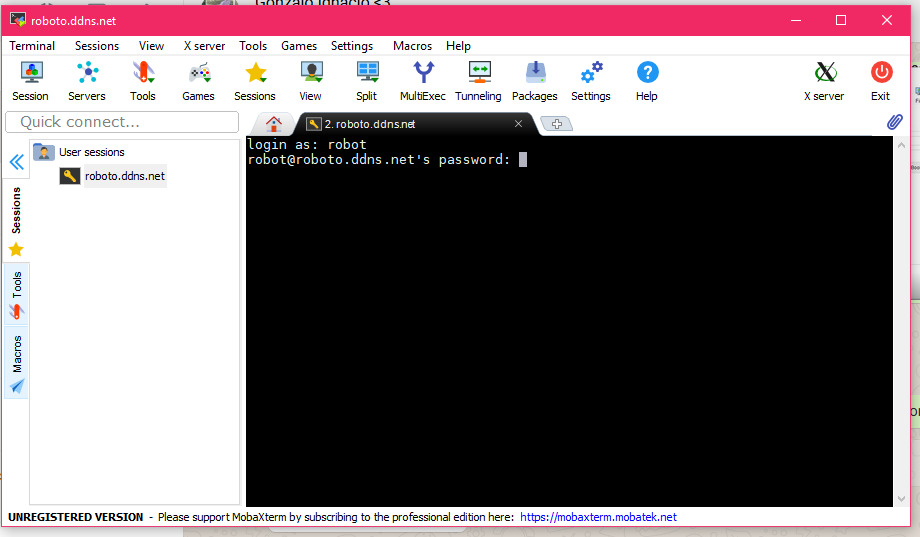


Ilustración A. Inicio de sesión por primera vez

Ahora has iniciado sesión en la plataforma y tienes acceso a los archivos dentro del robot.

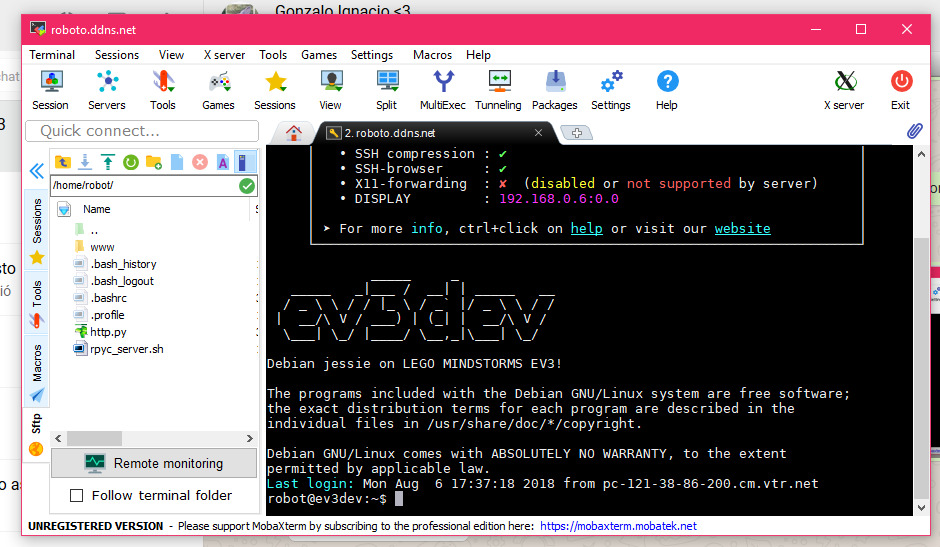


Ilustración A. Acceso a los archivos

En la terminal de mobaxterm es necesario correr los siguientes comandos:

* + sudo apt-get upgrade
  + sudo apt-get install python-ev3dev

El primer comando es recomendado realizar antes de instalar cualquier librería para instalar las ultimas actualizaciones.

El segundo comando permite instalar las librerías que el sitio ev3dev nos entrega que son necesarias para su programación en Python.

Luego de que se instalaran las últimas actualizaciones y las librerías, es necesario copiar los archivos desde el computador al EV3

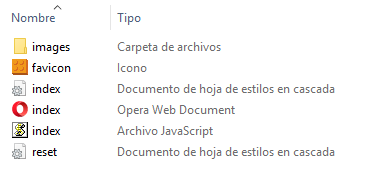


Ilustración A.5 Archivos necesarios dentro del EV3

Todos estos archivos pueden ser encontrados en el servidor de la universidad.

Una vez se tienen los archivos guardados dentro de la memoria del robot, se puede levantar el servidor al escribir el comando “python http.py” en la terminal del programa mobaxterm y de esa forma se logra levantar el servidor de manera local.

Para lograr la conexión remota, es necesario seguir los pasos que se encuentran en los puntos 1.4.1 y 1.4.2 en el apéndice A de este proyecto.

### 1.1.4 Archivos

Los archivos necesarios para la instalación y puesta en funcionamiento de este proyecto se pueden encontrar en el servidor de la Universidad Andrés Bello en el siguiente enlace:

<http://proyectos.informatica-unab-vm.cl/PT/Contreras-Caneo/Proyecto-de-titulo-2.rar>

O en su defecto se pueden obtener los mismos archivos necesarios para el levantamiento del web server y también de la página principal que se trabajó en el Github personal del alumno:

https://github.com/gcontrerascaneo/PT\_2

En dicho enlace se encontrarán con:

* http.py es un script hecho en python el cuál es el encargado de levantar el servidor web en el puerto 8081 utilizando la dirección IP que el EV3 muestra. Además, es el encargado de enviar directamente los valores de los parámetros de velocidad y tiempo a los motores del EV3.
* index.htm es la página principal en donde se verán los botones que simularán los movimientos.
* index.js es el programa encargado de una vez haciendo clic en algunas de las flechas del .htm, envía los parámetros al script en python.
* index.css hace que el cuadro con las flechas quede en ese orden en particular
* Carpeta con imágenes con las flechas que se ven reflejada en la plataforma.

## Configuración Cámara IP

Uno de los elementos importantes en este proyecto, es la incorporación de una cámara con el fin de registrar los movimientos realizados por los usuarios finales de esta plataforma para que ellos mismos sean testigos de lo que están haciendo dentro de la plataforma. Dicho esto, se procede a explicar los pasos que fueron ejecutados para instalar la cámara IP que se está utilizando que será por medio de un host.

### Creación cuenta

Como fue mencionando anteriormente, fue necesario la utilización de un host para que almacene las imágenes que la cámara IP está registrando al momento de ser utilizada la plataforma.

Para ello, utilizaremos el host de la página sensr.net y se procederá en crear una cuenta.

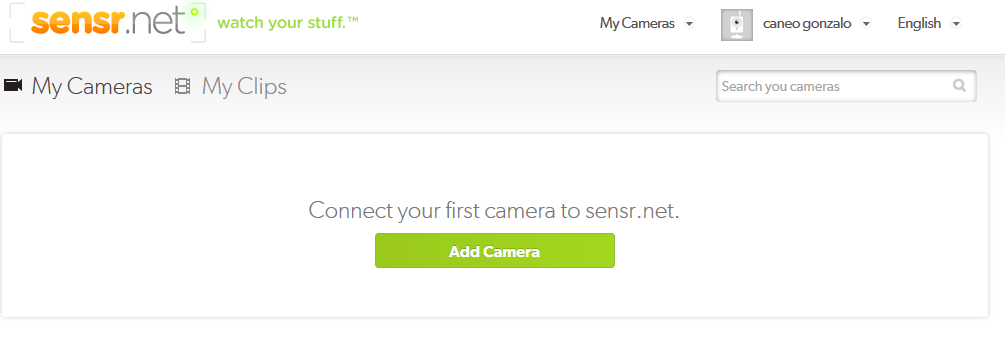


Ilustración A. Registro de una cámara

Una vez creada la cuenta se ingresa a la pestaña “My Cameras” y al presionar Add Camera se ingresarán los datos necesarios:

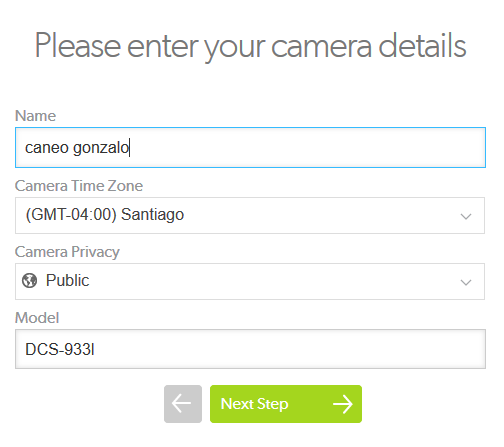


Ilustración A. Registro de la cámara en la página

Luego de registrar el nombre de la cámara, ubicación de esta, privacidad y el modelo la plataforma ahora enseñará una información para que ahora se realice la configuración dentro de la cámara.

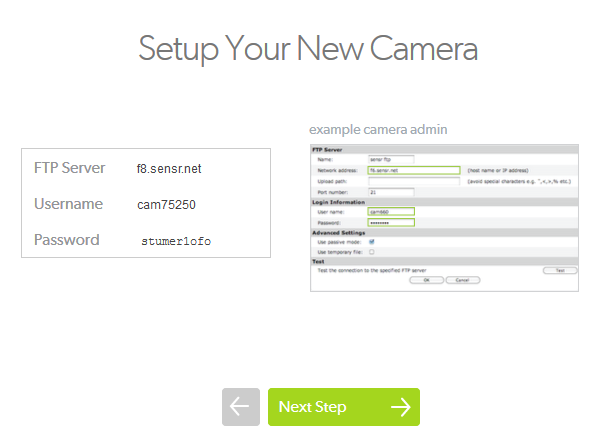


Ilustración A. Datos necesarios para modificar en la cámara

### Configuración cámara

Luego de tener creada la cuenta se utilizan las credenciales entregadas en la Ilustración A.8:

* FTP Server
* Username
* Password

Y son escritas en la interfaz de la cámara en la pestaña configuración y después presionar la opción FTP:



Ilustración A. Interfaz DCS-933L D-Link

Nota: la Ruta es donde guardará las imágenes registradas (dentro del mismo host) por lo que se requerirá agregar un nombre el cual se desea y dejar el Modo Pasivo siempre marcado en Sí.

# APÉNDICE B

Para la realización de esta parte de la actividad, se toma en consideración que el servidor del robot está montado en el domicilio del estudiante ya que para que el funcionamiento remoto se necesita que la red tenga una salida a internet, motivo único por el cual podría no ser factible utilizarlo de forma remota en alguna institución y más aún se puede instalar en cualquier domicilio particular, por lo que este documento servirá para dejar en evidencia los pasos que fueron realizados para que la plataforma funcionara de forma remota y que esta a su vez sea replicable en la Universidad una vez solucionado el problema de las redes disponibles dentro de la institución.

Una vez finalizada la exposición de este proyecto surgieron observación como el tema del funcionamiento remoto de este, es que se otorgó la oportunidad de corregir ciertos puntos para que el proyecto fuera finalmente declarado un éxito.

Entre los puntos los cuales fueron considerados a trabajar se encuentran:

* Mejorar la estética de la página web que se desarrolló.
* Establecer una conexión remota al EV3.
* Realizar diagrama de paquete y de despliegue

## 1.1 Nuevos tickets surgidos

Basados las observaciones una vez finalizado el proyecto, lo primero que se realizo fue dejar en evidencia el trabajo que se espera realizar en la herramienta en la cual se está trabajando.

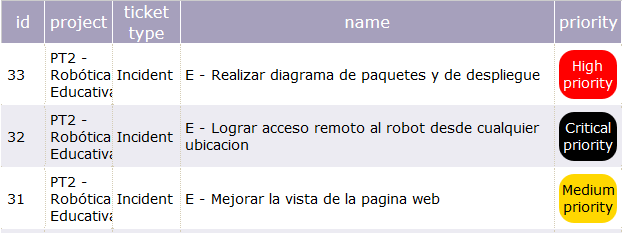


Ilustración B.1 Tickets aparecidos en la etapa final del proyecto

Como se observa, la prioridad crítica de las tareas solicitadas se encuentra el lograr el acceso remoto de la plataforma web que se desarrolló, por lo que se dio inicio al trabajo inmediato de esa solicitud.

Luego se encuentra el mejorar las arquitecturas dándole un mayor énfasis en la explicación de cómo funciona el proyecto, cuáles son los pasos que sigue este al momento de poner en funcionamiento el servidor hasta que el usuario es capaz de conectarse a la plataforma y manipular el kit.

Finalmente está el no menos importante mejoramiento estético de la plataforma web, trabajo el cual se le dio una prioridad media con respecto a las otras dos debido a que, a diferencia de las tareas con prioridad alta o crítica, esta no afecta el correcto funcionamiento de la plataforma.

## Análisis de nuevos riesgos

Una vez realizada la documentación de las tareas solicitadas por la comisión en la herramienta con la cual se está llevando a cabo la planificación del proyecto, se analizaron los posibles riesgos que podrían presentarse durante la ejecución de alguna de las tareas.

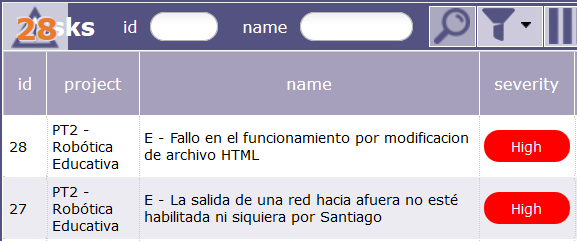


Ilustración B.2 Riesgos aparecidos

Como se puede observar en la ilustración B.2, fueron dos los riesgos que se encontraron al momento de analizar las tareas. Ambos riesgos fueron catalogados con alta severidad, siendo uno de ellos el que podría afectar directamente con el funcionamiento de la plataforma (riesgo con ID 28) por lo que se tomaron medidas para contrarrestar cualquier eventualidad negativa que pueda poner en peligro el funcionar del proyecto. El riesgo con ID 27 es el más importante a mitigar, puesto que es la base del proyecto y necesario para poder cumplir de manera satisfactoria con los objetivos específicos mencionados en el inicio de éste.

El plan de mitigación de igual manera fue documentado en la herramienta que se está utilizando y a continuación se mencionará el plan que se llevó a cabo:

|  |  |
| --- | --- |
| ID Riesgo | Mitigación |
| 27 | Como no es posible tener una salida de red desde la Universidad al exterior ni siquiera por Santiago como asegura el experto en redes de la facultad de Viña del Mar, se decidió optar por levantar un servidor temporal mientras la situación es regularizada por los encargados de esa área. Por lo tanto, el robot estará alojado temporalmente en una residencia en la cual será posible acceder de forma remota para que el usuario sea capaz de manipular el robot desde cualquier ubicación geográfica. |
| 28 | Tener siempre el archivo respaldado en caso de que estos eventos ocurran. |

Tabla B.1 Mitigación nuevos riesgos

Como fue propuesto por la comisión el día viernes 13 de julio de 2018, para comprobar que el proyecto funcionara de forma remota y que usuarios a pesar de no encontrarse en la misma ubicación física del robot puedan acceder a esta, se instalara el servidor de la plataforma en el domicilio del alumno para que los profesores de la Universidad puedan acceder desde la facultad y poder manipularlo de forma remota, evidenciando mediante una cámara que realmente es el docente quien está controlando el EV3.

Al lograr esto, quedaría en evidencia que es posible pasar a una red pública la plataforma que se desarrolló por lo que su réplica en la universidad sería posible de realizar al momento de que se habilite la salida de la red al exterior.

## 1.3 Gestión de Cambios

Como se puede apreciar en la tabla 10 del documento de memoria, el cambio fue solicitado por el alumno y aprobado por el profesor guía, por lo que se inició el proceso de trabajar en este.

## 1.4 Desarrollo Remoto

### 1.4.1 Creación Cuenta y Dominio en noip.com

Una de las tareas asignadas, y la más importante a desarrollar, fue lograr que la plataforma sea usada de forma remota por los usuarios. En un principio se acordó entre la comisión y el alumno de montar el servidor en su hogar debido a la imposibilidad de acceder a la red de la Universidad desde afuera, por lo que se optó por dicha opción para que quede en evidencia que es factible y replicable en cualquier ambiente, mientras los encargados de las redes dentro de la institución normalicen la situación.

Para el desarrollo de esta actividad, primero fue necesario crear una cuenta en la plataforma noip.com el cual es un servicio de DNS (Domain Name System)

dinámico que permite identificar, en este caso, el servidor en el cual se está trabajando con un nombre de dominio más fácil de recordar en lugar de utilizar la dirección IP.

Una vez creada la cuenta en la plataforma de no ip, se procede a crear un nuevo Dynamic DNS (DDNS) de esta forma:

1. Siendo redireccionado al dashboard de la plataforma se encontrará al lado izquierdo de la pantalla lo siguiente.

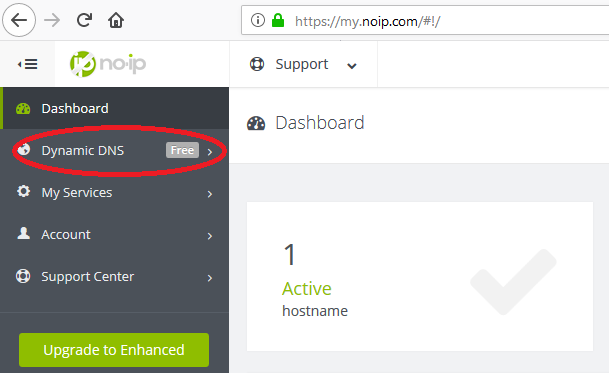


Ilustración B.3 Página principal noip.com

1. Al entrar a la pestaña Dynamic DNS se encontrará en lugar donde se efectuará la creación del nombre del dominio.

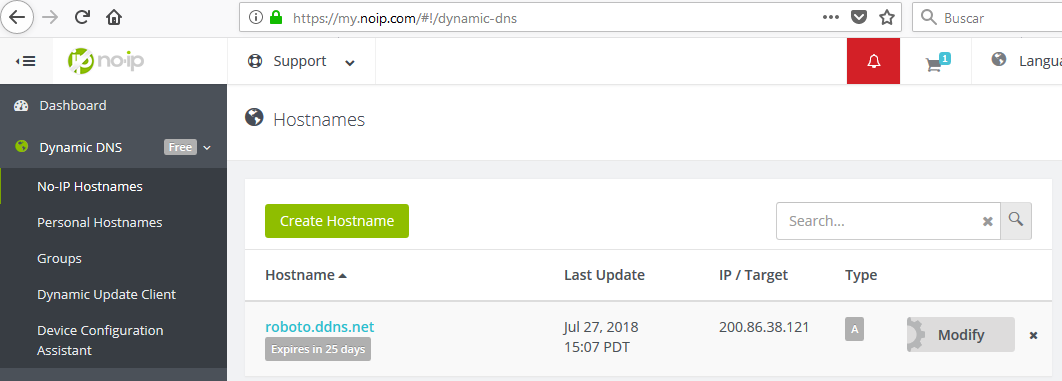


Ilustración B.4 Dominios creados

1. Al presionar Create Hostname se podrá dar inicio a la creación del nombre del dominio y se ingresará la dirección IP a donde esta apuntará. En el campo IPv4 es donde se anotará la dirección IP pública, generalmente el campo es rellenado de manera automática tomando en cuenta tu dirección IP.

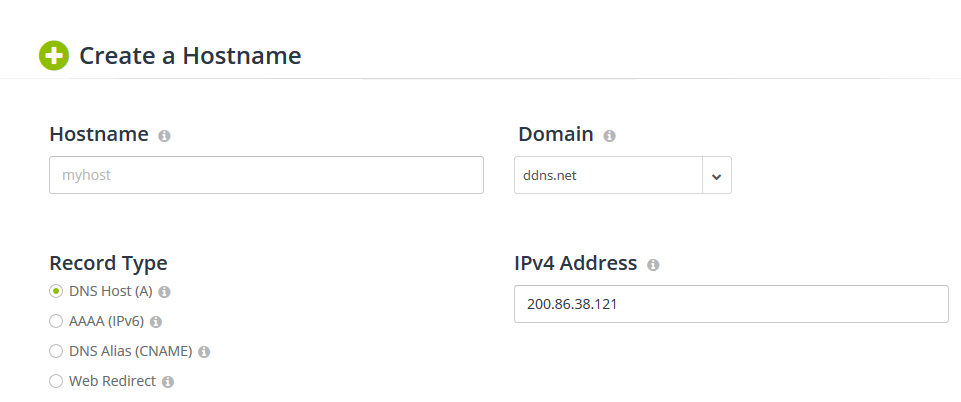


Ilustración B.5 Creación nombre del dominio

1. Al presionar Create Hostname se tendrá lista la primera parte que es la creación de la cuenta para tener un dominio que ingrese por un nombre y no la dirección IP.

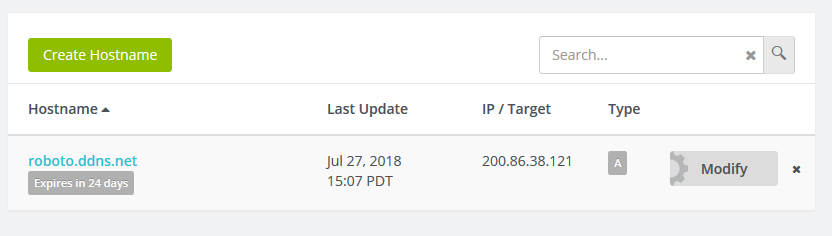


Ilustración B.6 Hostnames creados en una cuenta

Una vez finalizados estos pasos, se tiene un dominio (en el caso para este proyecto) denominado roboto.ddns.net que apuntará a la IP pública del domicilio del alumno que servirá finalmente para conectarse de manera remota a la red.

### 1.4.2 Modificación Modem

Para poder tener acceso remoto al servidor donde se encuentra alojado el producto, fue necesario realizar algunas modificaciones en el modem con el cual se llevó a cabo este trabajo. Para esto, fue necesario usar el modem Arris TG4982LG el cual fue proporcionado por el proveedor de internet que en este caso es VTR. Cabe destacar que esta configuración está hecha en un modem entregado por VTR y no necesariamente otro usuario tendrá el mismo modelo, por lo que algunas pestañas en la configuración pueden ser distintas a las mostradas en este apéndice, pero la teoría seguiría siendo la misma.

Cabe destacar que esta configuración fue hecha a partir del modem que fue entregado por VTR directamente, y que para funcionar en el caso de tener un router deberán habilitarse algunos puertos y permitir el forwarding entre el modem y el router.

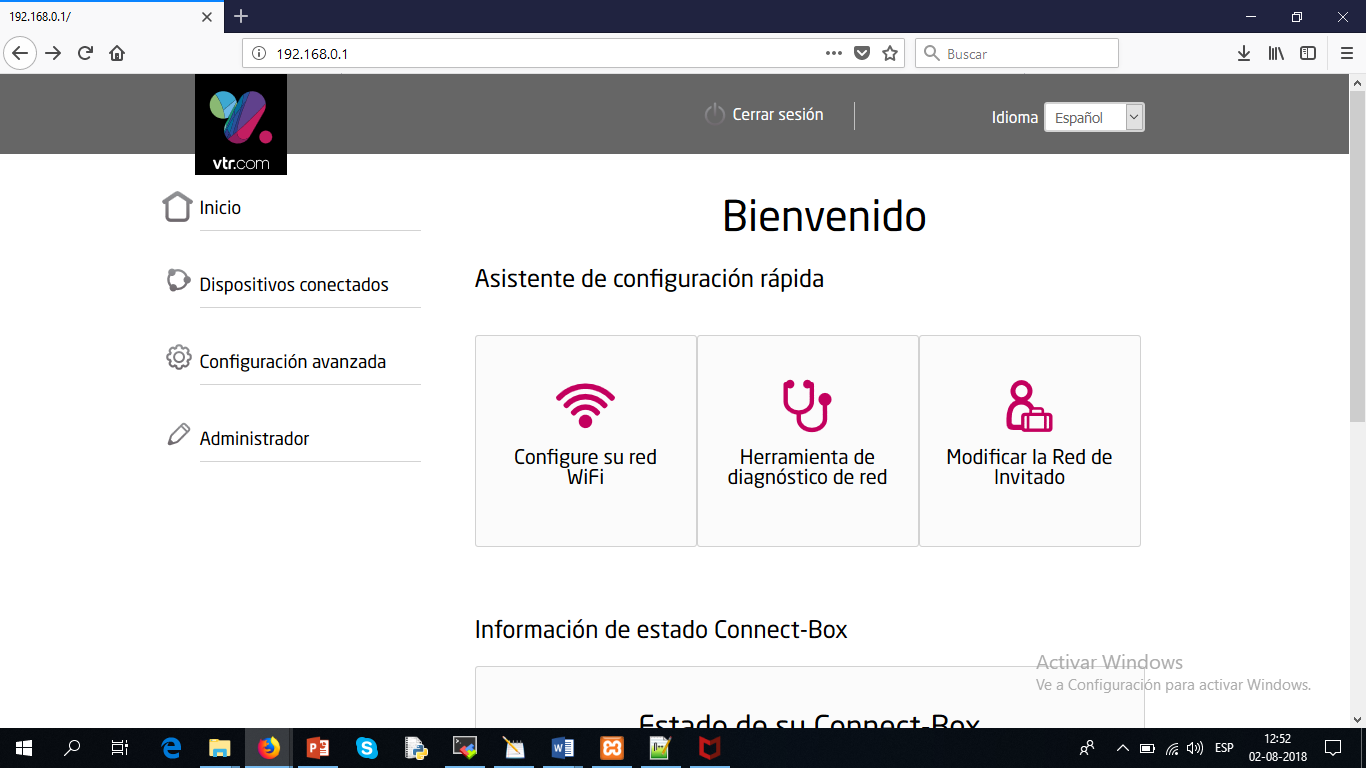


Ilustración B.7 Página principal configuración modem

Entonces para conseguir la comunicación remota fue necesario modificar los siguientes campos dentro del modem para que fuera posible:

1. Primero es importante mencionar que en este modem hay que dejar activada la configuración de seguridad Firewall porque es uno de los modelos de modem que viene con un error de fábrica que hace que al estar marcada la opción, se deshabilite el Firewall



Ilustración B.8 Firewall "activado" en el modem

1. El robot EV3 debe estar conectado en la red que el modem distribuye para tener registrada la dirección IP de este, al momento de querer indicar esta para ser accesible desde una ubicación remota.

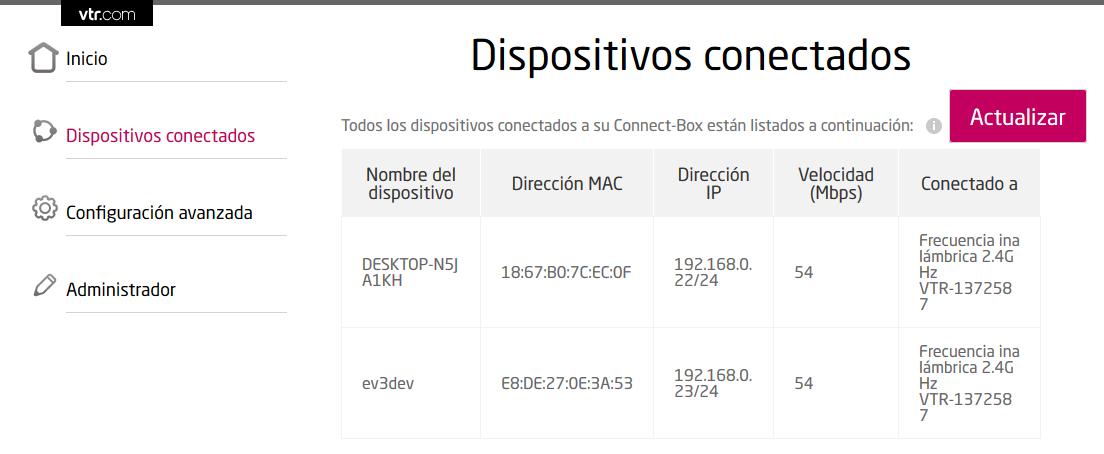


Ilustración B.9 EV3 conectado en la red

1. Para que se logre la conexión remota de la plataforma, es necesario apuntar a la dirección IP del robot en la pestaña Configuración

Avanzada -> DHCP y cambiar el campo por la IP que muestra el EV3 en este caso .23



Ilustración B.10 Agregar regla al modem a IP 192.168.0.23

Una vez realizado todos los pasos previos, al levantar la plataforma con el nombre de dominio previamente establecido (roboto.ddns.net:8081) y en el puerto 8081 donde se levanta al momento de ejecutar el comando es posible ingresar desde cualquier ubicación a este para que pueda moverse.

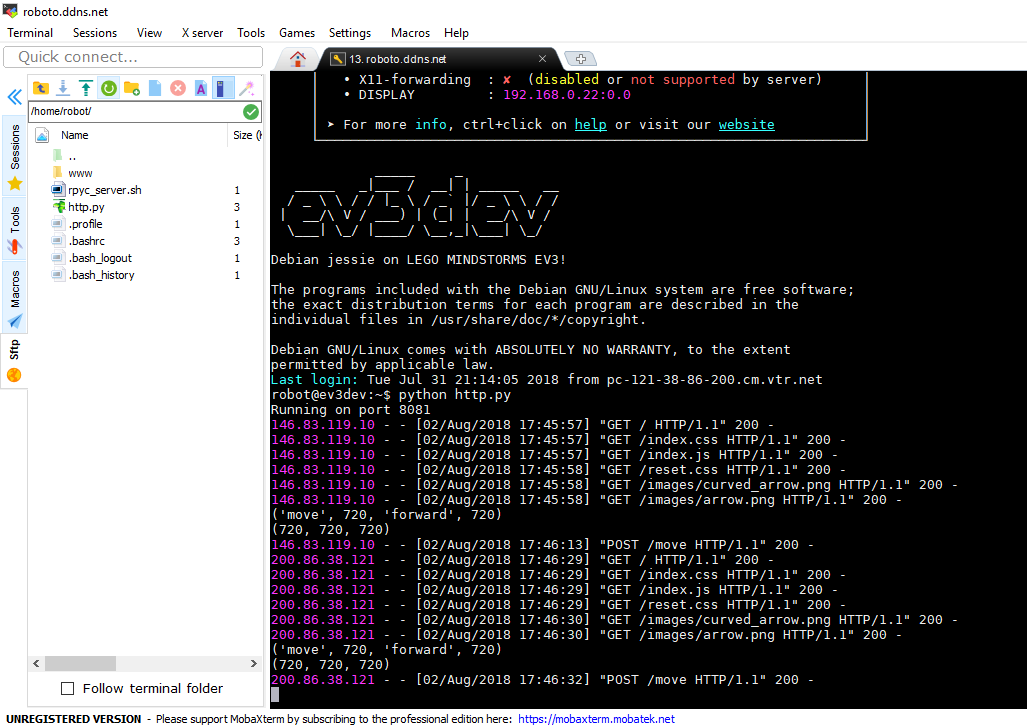


Ilustración B.11 MobaXterm mostrando manejo de otra ubicación

Como se puede apreciar en la ilustración B.11, se solicitó al profesor guía que accediera a la plataforma desde su ubicación actual siendo esta Antofagasta, mostrando así que la dirección desde donde se conecta es la 146.83.119.10.

En el caso contrario en la misma ilustración se aprecia la IP 200.86.38.121 que corresponde a la del estudiante de manera local que al mismo tiempo probó la plataforma para dejar evidenciado que es real la manipulación remota del EV3.

## 1.5 Modificación de la web

Otra de las tareas de gran importancia fue la modificación de la plataforma en donde el usuario ingresa. Si bien la plataforma en si no fue modificada, se optó por crear una página explicativa que finalmente redireccionará a la plataforma en donde se manipula el EV3.

Esto para que los usuarios entiendan cual es la finalidad del proyecto y el por qué se está realizando.

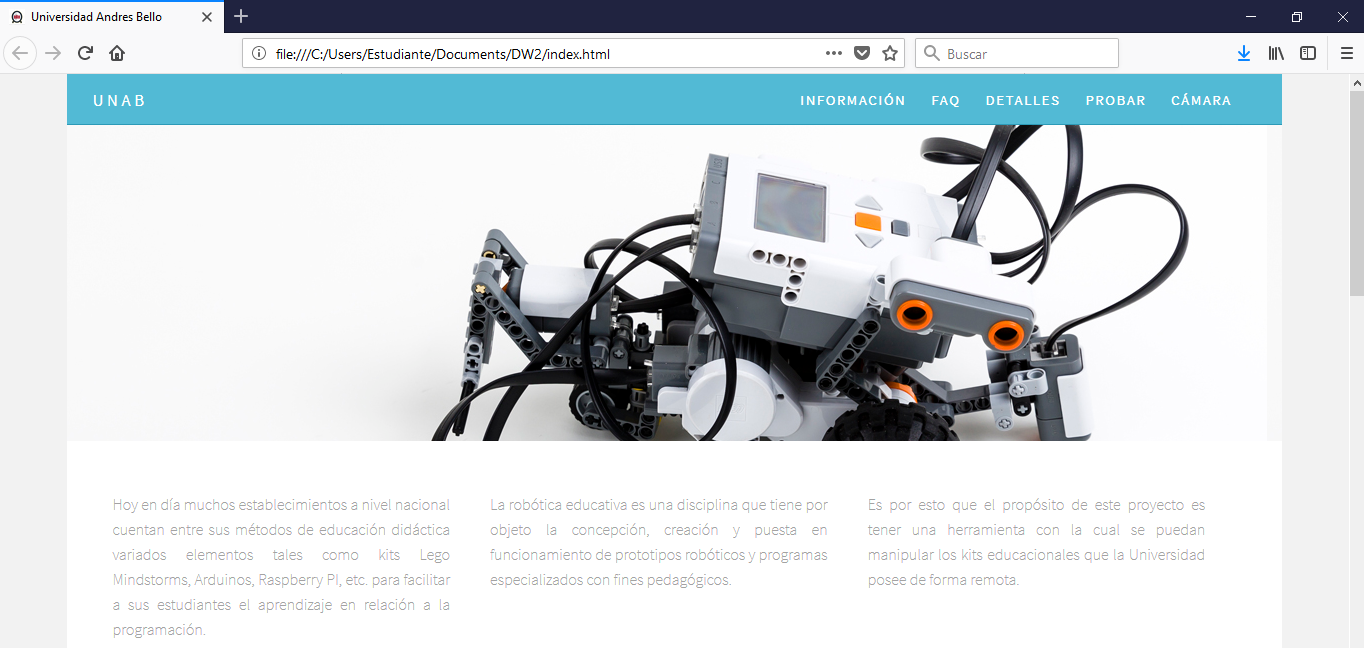


Ilustración B. Nueva página

De esta forma los usuarios se nutrirán de información con respecto a los objetivos del por qué el nacimiento e implementación de este proyecto.

Para la realización de esta nueva página se está utilizando el ambiente de desarrollo integrado (IDE) llamado Dreamweaver.

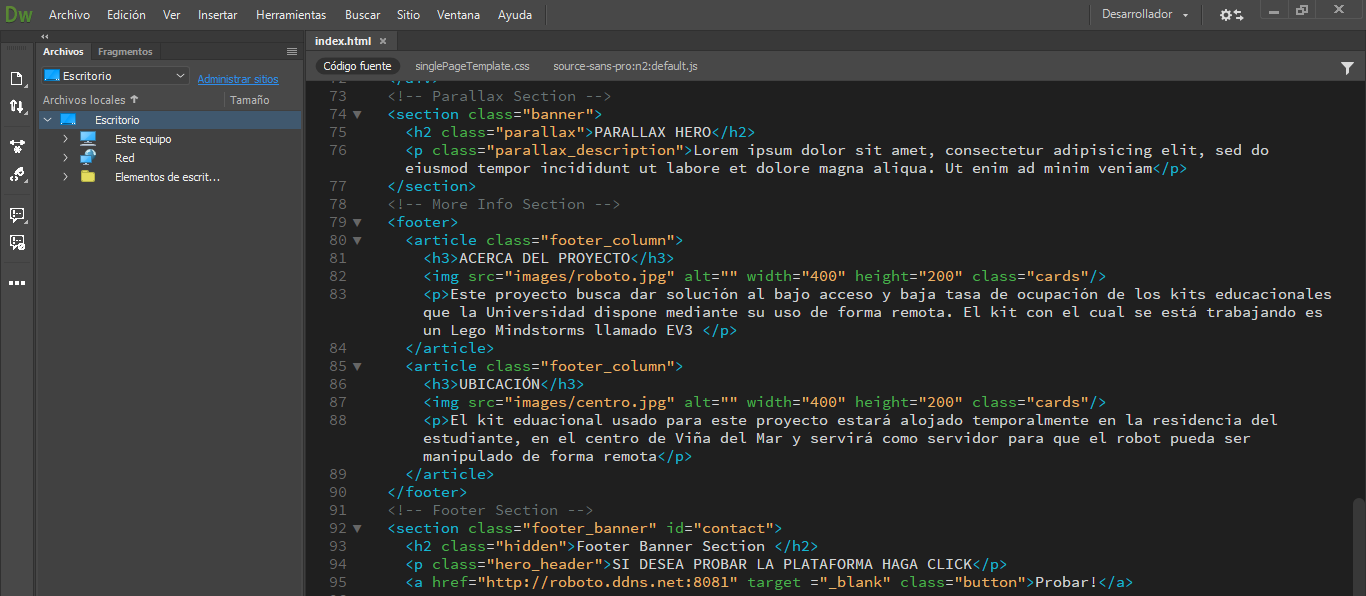


Ilustración B. Dreamweaver IDE

## 1.6 ¿Cómo funciona el proyecto?

### 1.6.1 Situación Previa

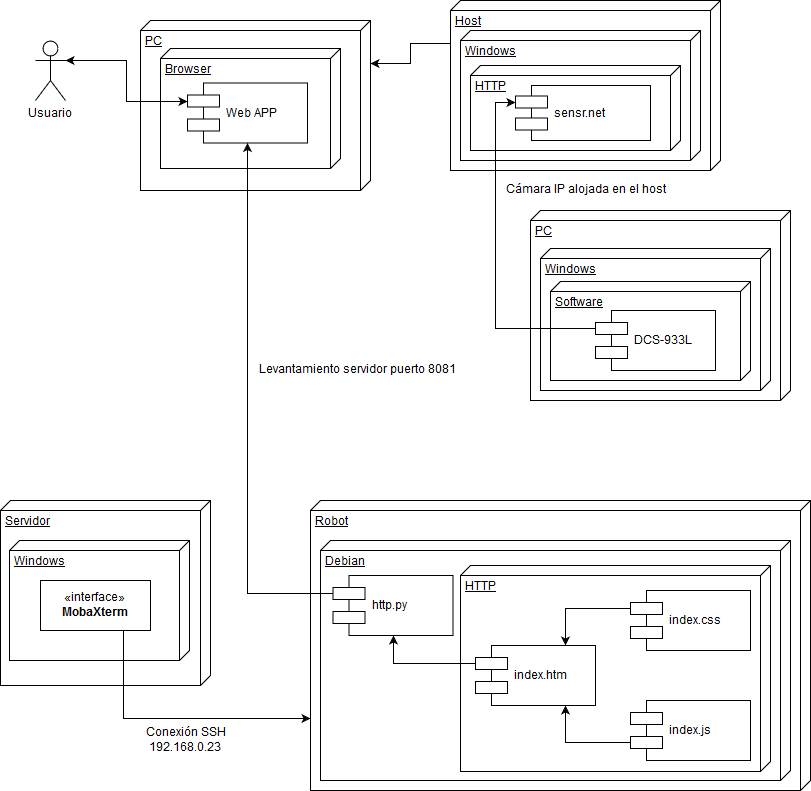


Ilustración B.14 Diagrama de Despliegue situación anterior

Como se puede apreciar en la imagen, antes de generar la solución de la comunicación remota existe la ausencia de un intermediario entre el robot y el browser, lo que producía que solo usuarios conectados a la misma red que el robot fuesen capaces de manipularlo, convirtiendo el proyecto en un laboratorio de manera local.

### 1.6.2 Situación Actual

Para la explicación de cómo funciona el proyecto desde el levantamiento del servidor hasta que el usuario ingresa a la plataforma actual, será detallado en el diagrama de despliegue que fue confeccionado.

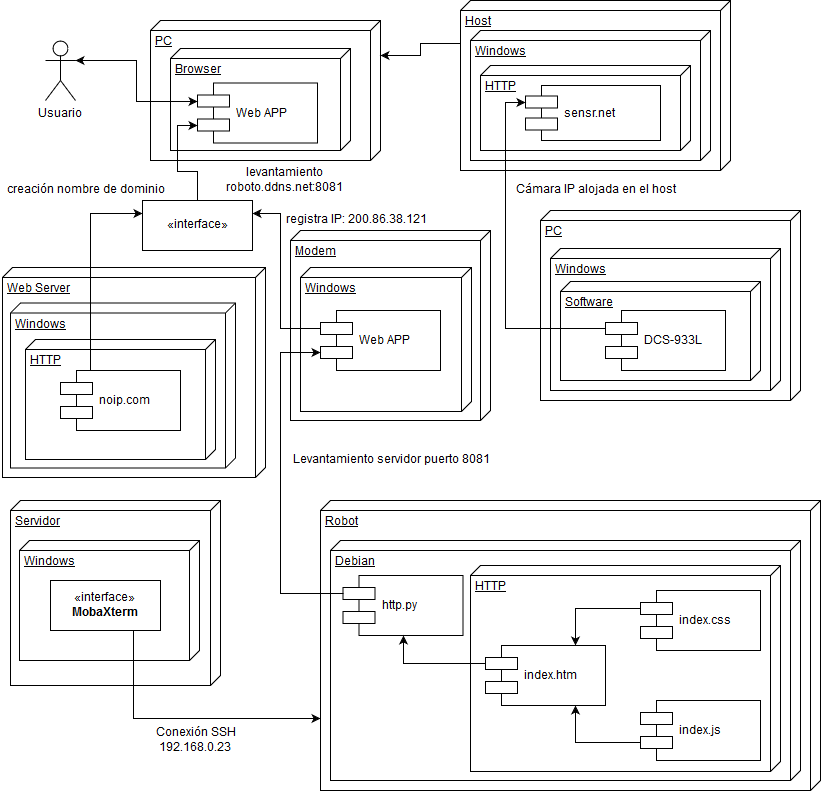


Ilustración B.15 Diagrama de Despliegue situación actual

Por lo que se puede apreciar en la ilustración B.15 recién presentada, existen varios elementos que logran en conjunto que la plataforma web pueda funcionar de manera correcta.

Empezando desde el servidor que está alojado temporalmente en el domicilio del estudiante, el cual se está utilizando el computador personal con las siguientes características; Intel Core i3-3210M CPU @ 2.50 GHz con Windows 10 de 64 bits, memoria física de 6 gigabytes, para el levantamiento del servicio.

Esto se realiza a través de un programa denominado Mobaxterm, el cual es un terminal para Windows en ambiente Linux, que permite, entre todas sus características, una conexión por SSH a la IP privada del robot que está conectado en la red siendo en este caso 192.168.0.23 permitiendo, como lo mencionado anteriormente, utilizar los comandos que el sistema operativo Linux posee. Al realizar la conexión por SSH con el robot, es posible ahora navegar por los archivos que este posee desde el ordenador. Era necesario un programa con ambiente Linux como mobaxterm ya que el sistema operativo instalado en el robot se trata de Debian, una distribución de Linux.

Una vez conectado en el robot, se observará que existen 4 archivos fundamentales para que el levantamiento del web server sea posible.

* **http.py** es el script creado en Python que sirve para levantar el servidor web en HTTP y en el puerto 8081. También sirve como el receptor de los parámetros que recibe de los archivos index.htm e index.js para generar los movimientos de los motores que permite que el robot se mueva. Es en este archivo en donde se pueden crear las excepciones en el caso de que se requiera limitar la cantidad de usuarios conectados simultáneamente. Este archivo se ejecuta con el comando “python http.py” en la terminal del programa mobaxterm.
* **index.htm** es el script en donde se da la forma a la plataforma, posee las imágenes de las flechas de instrucciones y contiene también los parámetros (velocidad, grado de giro del motor y tiempo) que serán enviados al robot.
* **index.js** es el script encargado de mandar los parámetros establecidos en el archivo index.htm una vez se presiona una de las flechas que simulan

uno de los movimientos del robot en la página. Estos parámetros son enviados al archivo http.py para que el robot se encargue de interpretarlo.

* **Archivos varios** también se encuentra una carpeta la cual contiene los archivos en formato .jpg de las flechas y el favicon que son mostrados en la plataforma.

Cuando se levanta el web server con el comando anteriormente mencionado, la IP pública del modem que está conectado el robot es registrado en la plataforma noip.com para generar un Dynamic DNS. El modem con el cual se está trabajando también fue necesario realizarle modificaciones como fue mencionado en el apartado 1.4.2 de este documento.

Al tener ya configurado el modem y el DDNS, solo basta con que el usuario se conecte con el nombre de dominio que se eligió, en este caso, se debería conectar a roboto.ddns.net:8081

Otro elemento importante para esta plataforma es la cámara para que el usuario sea capaz de ver que es él mismo quien está manejando el robot, por lo que fue necesario contar con una cámara IP DCS 933L de la marca D-link la cual está instalada frente al robot para registrar sus movimientos. la cámara por su parte está alojada en un host gratis para el acceso de los usuarios que quieran verla durante la ejecución de los movimientos del robot, dicho host es la página sensr.net.

### 1.6.3 Comparaciones

Como se puede apreciar en la ilustración B.14, se produce la ausencia de un intermediario que logre una configuración entre el robot y la web, dándose de esa manera que este proyecto funcione de forma local. Por políticas de la universidad, no se puede tener una red que pueda ser accedida desde el exterior por lo que es necesario utilizar el primer modelo ya que de igual manera funciona, pero de manera local.

En cambio, en la ilustración B.15 existe la presencia del intermediario que es capaz de amarrar la dirección IP del robot y hacerla visible al exterior al momento de montar el servidor.

## 1.7 Diagrama de Paquetes

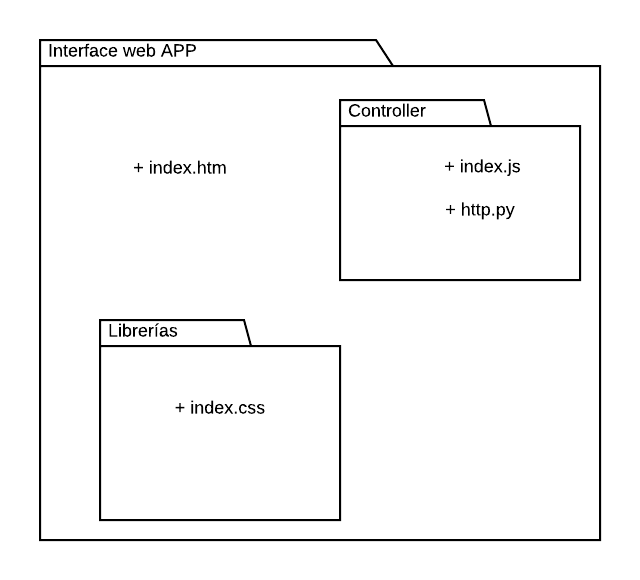


Ilustración B.16 Diagrama de Paquetes

En el diagrama mostrado en la ilustración B.16 se muestra las dependencias de todos los paquetes que están integrados en el robot y lo necesario hasta que el servidor web es montado con su nombre de dominio.

En el caso de la ilustración B.16 se muestran los archivos necesarios para la interfaz del control web, siendo necesarios el archivo http.py para el levantamiento y el index.css para el orden de las flechas con las cuales será controlado el robot. También es necesario el script desarrollado en javascript para enviar los parámetros de tiempo, velocidad y giro al ev3 al momento de presionar un botón de la plataforma.

Una de las ideas que se formularon al momento de querer evitar que múltiples usuarios se conecten a la plataforma es que la página se actualice cada cierto tiempo en una cantidad de segundos evitando de esta manera que un usuario que se encuentra conectado e inactivo no le quite la oportunidad a alguien quien quiera empezar a trabajar en la plataforma. Para esta alternativa solo falta hacer la modificación del archivo index.htm

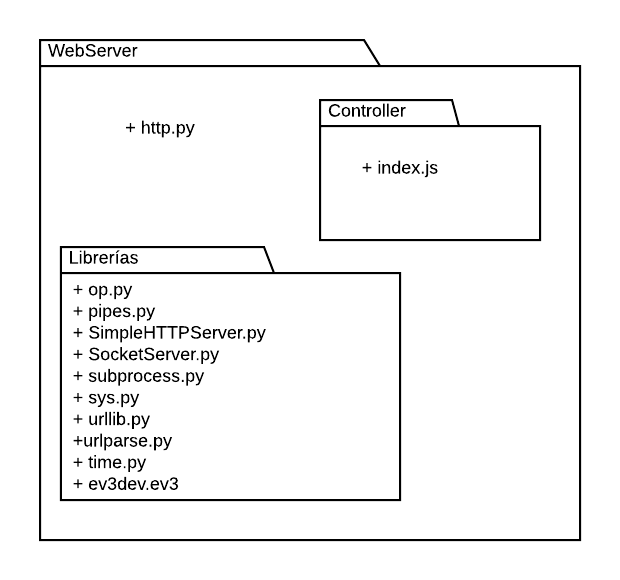


Ilustración B. Diagrama Paquetes webserver

Del mismo modo, se tiene el diagrama de paquetes para el levantamiento del servidor, el cual necesitará importar las librerías que están instaladas dentro del sistema operativo de ev3dev. También es necesario contar con el script index.js ya que este al recibir los parámetros mencionados anteriormente, son enviados al archivo en python para que sea interpretado por el robot. Las librerías solo deben ser importadas en el archivo http.py para que sean utilizadas.

Para trabajar en los scripts no fue necesaria la utilización de un entorno de programación en específico, por lo que se utilizó el mismo programa mobaxterm que una de sus funcionalidades es la capacidad de crear archivos dentro del mismo robot y ser guardado con la extensión que uno necesite.

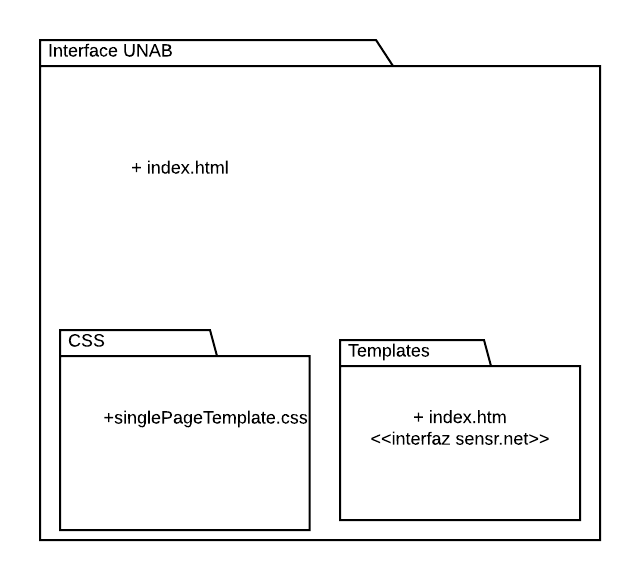


Ilustración B. Modelo nueva página

Para la nueva página que se creo mencionado en el apartado 1.5 de este anexo, se explicó que se realizó una nueva plataforma para que los estudiantes se nutrieran con información con respecto al proyecto. Para eso, se realizó una plataforma para la universidad en donde se encontrarán los diversos motivos por el cual se implementó el proyecto.

Para esto, fue necesario utilizar los archivos del robot index.htm y la interfaz de la cámara IP para que en la nueva plataforma sean redireccionados a la plataforma de control del robot.

A diferencia de los otros dos modelos, la realización de esta nueva plataforma se utilizó una IDE llamada Dreamweaver.

## 1.8 Conclusiones

A pesar de que el servidor está montado en el domicilio del alumno que está desarrollando este proyecto, fue posible desarrollar la tarea principal y la más importante como lograr la comunicación remota de la plataforma web hacia el exterior.

Gracias a la comunicación remota que fue capaz de lograrse, es que el proyecto cumple con todos los objetivos que fueron planteados al inicio de éste.

Es por esto por lo que se puede concluir que el proyecto es factible de realizar y de replicar, utilizando el anexo presentado, una vez que los problemas de las redes dentro de la universidad sean solucionados.