

UNIVERSIDAD NACIONAL ANDRÉS BELLO

FACULTAD DE INGENIERÍA

INGENIERÍA EN COMPUTACIÓN E INFORMÁTICA

**DESARROLLO DE PLATAFORMA PARA TELE-OPERACIÓN DE ROBOTS DE BAJO COSTO PARA AMBIENTE EDUCATIVO**

**Seminario de investigación para optar al grado de**

**Licenciado en Ingeniería en Computación e Informática.**

Autor: Pablo Sánchez Pino

Profesor guía: Miguel Solís Cid

VIÑA DEL MAR-CHILE

2018

INDICE

[RESUMEN 5](#_Toc514039285)

[INTRODUCCIÓN 5](#_Toc514039286)

[CAPITULO I - PROBLEMATICA 6](#_Toc514039287)

[1.1. DEFINICION DEL PROBLEMA 6](#_Toc514039288)

[1.2. DIAGRAMA ISHIKAWA DE SITUACIÓN ACTUAL DEL SISTEMA 6](#_Toc514039289)

[1.3. CINCO PORQUÉ 7](#_Toc514039290)

[1.4. OBJETIVOS 7](#_Toc514039291)

[Objetivos Generales 7](#_Toc514039292)

[Objetivos Específicos 7](#_Toc514039293)

[Métricas 7](#_Toc514039294)

[1.5. ALCANCES LIMITACIONES 8](#_Toc514039295)

[Alcances 8](#_Toc514039296)

[Limitaciones 8](#_Toc514039297)

[CAPITULO II - MARCO METODOLÓGICO 9](#_Toc514039298)

[2.1. METODOLOGÍA. 9](#_Toc514039300)

[2.2. ALTERNATIVAS DE SOLUCION. 9](#_Toc514039301)

[2.3. FACTIBILIDAD TECNICA. 10](#_Toc514039302)

[2.4. DISEÑO DE ALTO NIVEL. 10](#_Toc514039303)

[2.5. REQUERIMIENTOS. 10](#_Toc514039304)

[CAPITULO III - PLAN DE PROYECTO 11](#_Toc514039305)

[3.1. PLANIFICACION 11](#_Toc514039307)

[3.2. PLAN DE GESTION DE CONFIGURACION 11](#_Toc514039308)

[3.3. PLAN DE GESTION DE CAMBIOS 12](#_Toc514039309)

[3.4. PLAN DE GESTION DE RIESGOS 12](#_Toc514039310)

[3.5. AMBIENTE DE DESARROLLO 13](#_Toc514039311)

[3.6. PLAN DE PRUEBAS 13](#_Toc514039312)

[CAPITULO IV – ANALISIS 14](#_Toc514039313)

[4.1. ANALISIS 14](#_Toc514039315)

[4.1.1. FASE I 14](#_Toc514039316)

[4.1.2. FASE II 15](#_Toc514039317)

[4.2. DISEÑO DETALLADO 15](#_Toc514039318)

[4.3. PLAN DE PRUEBAS DE DESARROLLO DE SOFTWARE 17](#_Toc514039319)

[4.3.1. ALCANCE DE LAS PRUEBAS FASE I. 17](#_Toc514039320)

[4.3.1.1. Elementos de pruebas 17](#_Toc514039321)

[4.3.1.2. Funcionalidades por probar 17](#_Toc514039322)

[4.3.1.3. Funcionalidades por no probar 18](#_Toc514039323)

[4.3.1.4. Enfoque de pruebas 18](#_Toc514039324)

[4.3.1.5. Criterios de aceptación o rechazo 18](#_Toc514039325)

[4.3.1.6. Criterios de suspensión 18](#_Toc514039326)

[4.3.1.7. Criterios de reanudación 19](#_Toc514039327)

[4.3.2. Recursos 19](#_Toc514039328)

[4.3.2.1. Requerimientos de Hardware 19](#_Toc514039329)

[4.3.2.2. Requerimientos Software 19](#_Toc514039330)

[4.3.3. Metodología Pruebas 19](#_Toc514039331)

[4.3.4. Dependencias y Riesgos 19](#_Toc514039332)

[4.3.5. Referencias 20](#_Toc514039333)

[4.3.6. Avance Técnico Fase I 20](#_Toc514039334)

[4.3.7. Código Prototipo Fase I 21](#_Toc514039335)

[4.3.8. Pruebas de Aceptación Fase I 24](#_Toc514039336)

[4.4. PRUEBAS DE SISTEMA 25](#_Toc514039337)

[4.5. ALCANCE DE LAS PRUEBAS FASE II. 25](#_Toc514039338)

[4.5.1.1. Elementos de pruebas 25](#_Toc514039339)

[4.5.1.2. Funcionalidades por probar 25](#_Toc514039340)

[4.5.1.3. Funcionalidades por no probar 25](#_Toc514039341)

[4.5.1.4. Enfoque de pruebas 25](#_Toc514039342)

[4.5.1.5. Criterios de aceptación o rechazo 26](#_Toc514039343)

[4.5.1.6. Criterios de suspensión 26](#_Toc514039344)

[4.5.1.7. Criterios de reanudación 26](#_Toc514039345)

[4.5.2. Recursos 26](#_Toc514039346)

[4.5.2.1. Requerimientos de Hardware 26](#_Toc514039347)

[4.5.2.2. Requerimientos Software 26](#_Toc514039348)

[4.5.3. Metodología Pruebas 27](#_Toc514039349)

[4.5.4. Dependencias y Riesgos 27](#_Toc514039350)

[4.5.5. Referencias 27](#_Toc514039351)

[4.5.6. Pruebas Funcionales Fase II 27](#_Toc514039352)

[4.5.7. Avance Técnico Fase II 30](#_Toc514039353)

[4.5.7.1. Prototipado I 30](#_Toc514039354)

[4.5.7.2. Prototipado II 33](#_Toc514039355)

[4.5.8. Pruebas de Aceptación Fase II 35](#_Toc514039356)

[4.5.9. Fase III 36](#_Toc514039357)

[4.6. Evidencia Gestión de Riesgo 37](#_Toc514039358)

[4.7. Evidencia Gestión de Cambio 37](#_Toc514039359)

[5. POSTMORTEM 37](#_Toc514039360)

[5.1. TRABAJOS FUTUROS 38](#_Toc514039361)

# RESUMEN

El uso de las Tecnologías de la Información y Comunicación ha demostrado ser útil en ambientes educativos, no sólo entregando herramientas tecnológicas para la resolución de problemas o estimulando cierto tipo de razonamiento, sino también fomentando el interés de jóvenes a explorar un conocimiento más profundo en áreas de la ciencia y tecnología. En este contexto, la robótica educativa presenta un novedoso y atractivo enfoque para presentar conceptos desde el pensamiento secuencial y razonamiento lógico hasta la programación de autómatas que son capaces de realizar su propia inferencia del entorno o desarrollar un aprendizaje continuo. A pesar de la alta convocatoria que tienen este tipo de actividades, el alcance respecto al número de alumnos beneficiados se limita en torno a la capacidad física de la sala utilizada o la cantidad de equipamiento disponible. En este proyecto de título, como objetivo a largo plazo se propone la implementación de un laboratorio de robótica educativa operado de forma remota, maximizando así el tiempo de utilización de algún kit educativo de bajo costo, permitiendo que los alumnos experimenten con éste de forma independiente a su ubicación geográfica.

# INTRODUCCIÓN

El concepto de “enseñanza tradicional” consiste en establecer sesiones presenciales a las cuales los alumnos deben asistir en determinado horario, ubicación y laboratorio que cuente con los dispositivos físicos necesarios para llevar a cabo este concepto.

Para este caso, concepto de “enseñanza tradicional” no es practico debido a sus capacidades limitadas como por ejemplo número de equipos disponibles, ubicación geográfica, horarios académicos entre algunos. Otro punto que se debe considerar es que se trata de una tecnología de alto costo que hará menos viable su implementación. Como resultado de la práctica “enseñanza tradicional”, debemos considerar que el acceso a estas tecnologías estará disponible solo a un grupo limitado de alumnos, además el proceso de aprendizaje será más lento, limitando el desarrollar, creación y experimentación de nuevas funcionalidades en el ambiente educativo de la robótica. No obstante, con la utilización de internet es posible superar estas limitaciones mediante laboratorios virtuales con acceso remoto que permiten a los alumnos realizar actividades prácticas desde ubicaciones remotamente distantes en horarios mucho más flexibles.

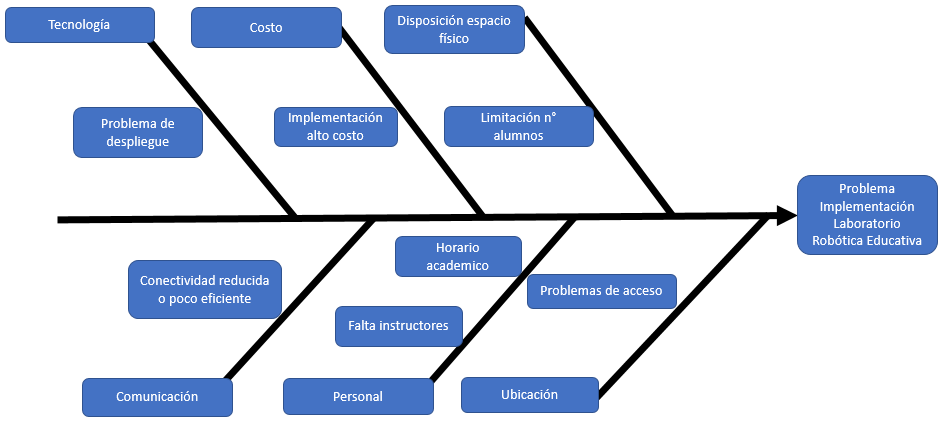
# CAPITULO I - PROBLEMATICA

## DEFINICION DEL PROBLEMA

Debido al poco alcance que ofrece la robótica educativa producto de costos de dispositivos y posterior implementación, limitación número de alumnos, espacio físico y ubicación geográfica laboratorios, entre otros, a momento no se ha logrado desarrollar plataforma o sistema remoto que permite soslayar estas limitaciones objeto permitir que un gran número de alumnos tenga acceso a estos dispositivos escaso e implementados en un laboratorio especializado que pueda ofrecer practica remota, además no se cuenta con herramientas que permita la retroalimentación visual de lo las funciones que ejecute el robot A ello debemos considerar que los métodos tradicionales de acceso remoto (RDP Microsoft, VNC, TeamViewer, etc.) requieren de mayores recursos de ancho de banda o conexiones más veloces para su utilización.

La tele operación de robots permitirá a los alumnos acceder a un curso de robótica y probar sus algoritmos de control en el robot real, ubicado en un laboratorio remoto, disponible las 24 horas del día y del cual se podrá obtener retroalimentación online mediante streaming de video.

## DIAGRAMA ISHIKAWA DE SITUACIÓN ACTUAL DEL SISTEMA



## CINCO PORQUÉ

1. ¿Por qué no se cuenta con un laboratorio de robótica? *Porque el alcance de su implementación es muy limitado.*
2. ¿Por qué el alcance de su implementación es muy limitado? *Porque su implementación tiene un alto costo versus beneficio o resultados que se puedan obtener.*
3. ¿Por qué su implementación tiene un alto costo versus beneficio o resultados que se puedan obtener? *Porque solo será accesible a un número limitado de alumnos.*
4. ¿Por qué solo será accesible a un número limitado de alumnos? *Porque dependerá de la disponibilidad horaria para practica y ubicación geográfica alumnos*
5. ¿Por qué dependerá de la disponibilidad horaria para practica y ubicación geográfica alumnos? *Porque no se cuenta con plataforma que entregue las herramientas para la tele operación remota, de bajo costo, en ambiente educativo.*

## OBJETIVOS

### Objetivos Generales

* Desarrollar plataforma o sistema que permita la tele operación de robots a bajo costo objeto poder dar mayor valor a la robótica educativa permitiendo poner esta tecnología al alcance de un mayor número de alumnos independiente de su ubicación geográfica.

### Objetivos Específicos

* Plataforma deberá permitir autenticar usuarios.
* Compatible con mayoría navegadores webs disponibles.
* Plataforma web deberá ser intuitiva y de fácil operación para alumno.
* La interfaz de usuario debe ajustarse a las características de la web de la institución.
* Plataforma web deberá disponer se retroalimentación mediante streaming de video.
* Implementación plataforma debe considerar utilización hardware disponible en organización, objeto mantener costos bajos.
* Establecer canal de comunicación cifrado acceso remoto.

### Métricas

* Disponibilidad plataforma días lunes a viernes, horario académico (08:00 – 22:30 hrs).
* Acceso y operación recurrente dependiendo de cantidad de robots disponibles.
* Requerimiento ancho de banda mínimo 512Kbp.
* El sistema debe tener una disponibilidad del 98,99% de las veces en que un usuario intente accederlo.
* El tiempo para iniciar o reiniciar el sistema no podrá ser mayor a 5 minutos.
* La tasa de tiempos de falla del sistema no podrá ser mayor al 2% del tiempo de operación total.
* El promedio de duración de fallas no podrá ser mayor a 2 horas
* La probabilidad de falla del Sistema no podrá ser mayor a 2%.

## ALCANCES LIMITACIONES

### Alcances

Objeto lograr mejor entendimiento proyecto, el alcance de este se dividirá en dos categorías.

* Software: Considera utilización de programa Scratch que es un lenguaje de programación gratuito, altamente intuitivo y amigable para utilizar en ambiente educativo. Por otra parte, se debe desarrollar una WebApp que permita acceso remoto objeto alumnos puedan ejecutar Scratch desde servidor laboratorio y a su vez entregue retroalimentación vía streaming de video.
* Hardware: Proyecto considera la utilización de Mbot que consisten en kit de robótica ideal para que los alumnos se inicien en la robótica, programación y electrónica basado en Arduino y Scratch, compatible con mayoría de dispositivos móviles y además permiten conectividad mediante tecnología Bluetooth o 2,4G. Por otra parte, se debe considerar la implementación de un servidor en laboratorio que permita gestionar las sesiones remotas, conectividad con Mbot y permita a los alumnos ejecutar software Scratch.

### Limitaciones

Objeto lograr mantener los costos bajos, al igual que los alcances, limitaciones se dividirán en dos categorías.

* Software: Aprovechar ventaja Scratch es un software gratuito para uso tanto personal como educativo, de esta forma se puede integrar a proyecto sin limitarse por costos de licencias.
* Hardware: Proyecto no considera adquisición de hardware propio, sistema o plataforma considera instalación en hardware que se encuentre disponible en organización.

# CAPITULO II - MARCO METODOLÓGICO



## METODOLOGÍA.

Tanto para etapa de *Desarrollo de Software* como para etapa de *Gestión de Proyecto* se implementará metodología ágil la cual permite mejorar la eficiencia en la gestión y la calidad del producto final, tener la capacidad de respuesta al cambio en los productos y sus definiciones, entregas tempranas y la retroalimentación continua durante la construcción del producto. Se ha decidido utilizar la metodología *Scrum “Adaptado”* debido a que se ajustará al factor tiempo permitiendo la rápida iteración y como resultado una rápida retroalimentación., además permite enfocarse exclusivamente en los problemas planteados objeto garantizar la calidad del producto. Otro factor de suma importancia es la fácil adaptabilidad y flexibilidad que entrega *Scrum* en la cual podremos implementar cambios y agregar nuevas capacidades en caso de ser necesario, de forma más rápida y eficaz.

## ALTERNATIVAS DE SOLUCION.

Durante proceso de investigación, en su mayoría, se encuentran sistema de tele operación similares los cuales se enfocan en ambientes industriales (brazos robóticos) o médicos los cuales tienen en común arquitectura cliente servidor. En algunos casos sistema entregan retroalimentación mediante video o fotogramas.

Se encuentra sistema que podría servir de alternativa el cual está basado en placa Arduino con módulo Xbee el cual integra comandos de voz para realizar la operación remota dispositivos.

Cabe señalar que alternativas investigadas solo hacen referencia a la operación remota, ya sea mediante web browser o algún aplicativo que permite la tele operación. Para nuestro caso el gran reto será implementar e integrar acceso remoto y permitir la programación inalámbrica de los dispositivos robots.

Alternativa propuesta consiste en implementar plataforma o sistema con disponibilidad 24x7 basado en arquitectura cliente servidor, que permita el acceso remoto mediante WebApp al servidor ubicado en laboratorio en cual residirá software para programación de los dispositivos robot. Objeto evitar la utilización de cables y permitir mayor autonomía de laboratorio servidor deberá establecer comunicación inalámbrica con dispositivos robots para efectuar carga de programación. Además, considera integrar retroalimentación mediante streaming de video objeto validar y certificar programación y movimientos dispositivos robots.

## FACTIBILIDAD TECNICA.

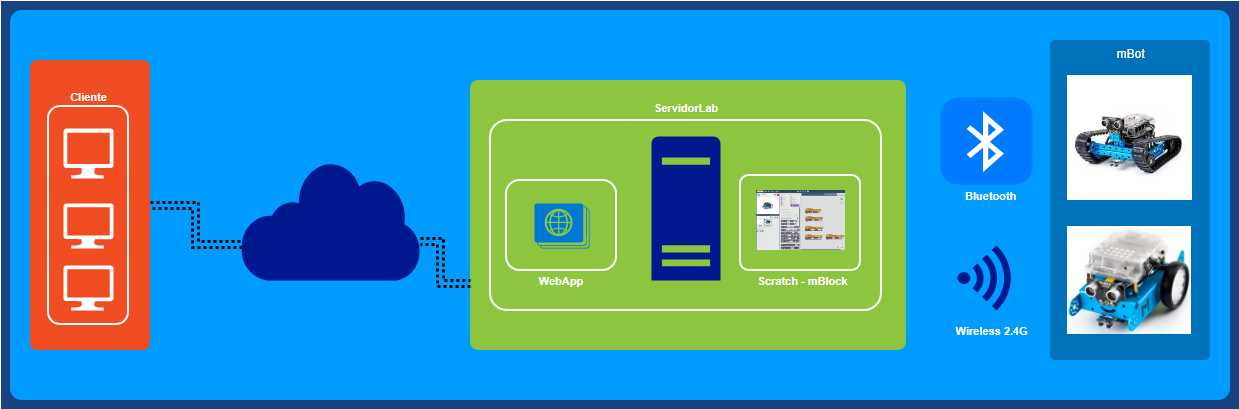
Proyecto considera y plantea la utilización de siguiente hardware.

* Servidor capacidad promedio disponible por UNAB.
  + Interfaz inalámbrica.
* Robot los cuales consideramos siguientes alternativas.
  + Dispositivos Mbot basado en Arduino con modulo inalámbrico.
  + Kit armable Arduino con modulo inalámbrico.

Para desarrollo lógica se considera utilización siguiente software.

* Software Arduino.
* Software Scratch.
* Software acceso remoto.
* Debian 9

## DISEÑO DE ALTO NIVEL.



## REQUERIMIENTOS.

Revisar documento ***Anexo n°1 “Especificación Requisitos Sistema”*** definido por estándar IEEE-830-1998 el cual entrega una descripción completa del comportamiento de la plataforma o sistema que se desarrollara.

# CAPITULO III - PLAN DE PROYECTO



## PLANIFICACION

Se considera la utilización de herramienta de planificación EDT o Estructura de Desglose del Trabajo la cual consiste en una descomposición jerárquica, orientada al producto entregable del trabajo que será ejecutado por el equipo del proyecto, para lograr los objetivos del proyecto y crear los productos entregables requeridos. Organiza y define el alcance total al subdividir el trabajo en porciones de trabajo más pequeñas y fáciles de manejar, llamados paquetes de trabajo, que pueden programarse, costearse, supervisarse y controlarse. ***Anexo n°2 “EDT Tele Operación de Robots Ambiente Educativo”*** y ***Anexo n°2 “Cronograma Proyecto”.***

## PLAN DE GESTION DE CONFIGURACION

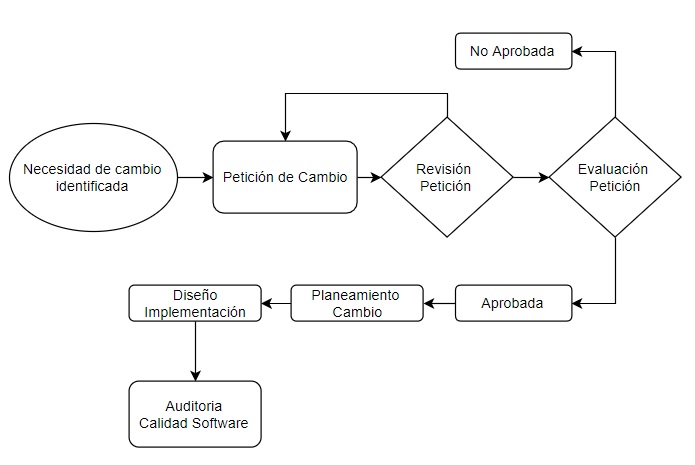
Identificación de la configuración: Establecer mecanismos de identificación de elementos de hardware y software que puedan ser más susceptibles durante desarrollo del proyecto objeto establecer trazabilidad y seguimiento durante siguientes etapas.

Control de Cambio de Configuración: Identificar y registrar peticiones de cambios efectuados durante desarrollo proyecto objeto priorizar, analizar, evaluar dichas peticiones para su eventual aprobación o rechazo.

Auditoria y revisión de configuración: Validar y comprobar que cambios efectuados durante desarrollo proyecto cumplan y logren con lo que sistema pretende ser. La auditoría se realizará cuando se estime oportuna y debe resolver siguientes interrogantes.

* ¿Se han realizado las peticiones de cambios? En caso de que la respuesta sea negativa, ¿por qué?
* ¿Se están aplicando estándares de la Ingeniería del Software para la resolución de los cambios, como reutilización y flexibilidad de código?
* ¿Se incluyen en los informes de cambios la fecha y los autores del mismo?

## PLAN DE GESTION DE CAMBIOS



## PLAN DE GESTION DE RIESGOS

|  |  |  |  |
| --- | --- | --- | --- |
| RIESGO | IMPACTO | PROBABILIDAD | ¿POR QUÉ? |
| HARDWARE | ALTO | MEDIO | * GR-01 Hardware entregado para proyecto no cumple con estándar mínimo solicitado. * GR-02 Entrega tardía hardware afectando desarrollo cronograma y planificación proyecto. * GR-03 Hardware defectuoso u obsolescencia tecnológica. * GR-04 Fallas de implementación. * GR-05 Hardware no soporta software. * GR-06 Incompatibilidad tecnológica. |
| SOFTWARE | ALTO | ALTO | * GR-07 No cumple con requerimientos. * GR-08 Problema integración con otras aplicaciones. * GR-09 Problemas ejecución (latencia). |
| CAMBIO  REQUERIMIENTOS | ALTO | BAJO | * GR-10 Deficiente “Especificación de Requerimientos”. * GR-11 Retraso especificación. * GR-12 Cambios realizados etapas avanzadas proyecto. |
| DISEÑO | ALTO | ALTO | * GR-13 No cumple con necesidades sistema. * GR-14 No se ajusta a solución propuesta y desarrollada. * GR-15 Inconsistencia con hardware y software. |
| IMPLEMENTACION | ALTO | MEDIO | * GR-16 No disponer de espacio físico adecuado para montaje hardware requerido. * GR-17 No lograr integración sistema. |

## AMBIENTE DE DESARROLLO

Desarrollo sistema o plataforma considera utilización siguientes herramientas y lenguajes para su construcción.

* Robot considera utilización para primera fase software Arduino – Python.
* Desarrollo WebApp considera lenguaje de programación PHP.
* Servidor bajo plataforma Debian 9.
* RubyonRails.
* NetBeans.

## PLAN DE PRUEBAS

Para primera fase de proyecto las pruebas consistirán en la validación de la comunicación inalámbrica entre robot y pc - servidor, para ello se considera los siguientes ítems como parte del plan de pruebas.

* Efectuar programación mediante software Arduino para establecer canal de comunicación inalámbrico.
* Establecer comunicación inalámbrica mediante comandos AT desde Monitor Serie software Arduino.
* Validar conectividad inalámbrica mediante de software Putty.
* Identificar puerto serial “COM”.
* Ejecutar software Arduino, seleccionar puerto identificado punto anterior y cargar sketch en dispositivo.

# CAPITULO IV – ANALISIS

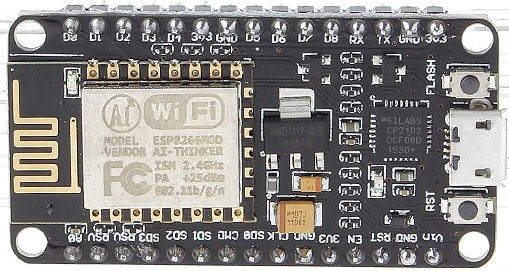


## ANALISIS

## FASE I

Una vez finalizada etapa se logra determinar elección tecnología para establecer comunicación inalámbrica robot y pc-servidor. Alternativas investigadas:

* Módulo ESP8266: chip de bajo costo Wi-Fi con una pila TCP/IP completa y un microcontrolador.
* Módulo HC-06: dispositivo bluetooth se puede utilizar para dar conectividad inalámbrica a través de una interfaz serial TTL entre Microcontroladores (tales como PICs o Arduino) y otros dispositivos como PC, laptops o celulares Smartphone.

*Modulo Bluetooth HC-06 Modulo Wifi ESP8266Mod*

Para ambos módulos se debe establecer en primera instancia la programación cableada y carga sketch Arduino que permite levantar canal de comunicación serial inalámbrico.

En caso de módulo ESP8266, aunque presenta mayores ventajas por sobre módulo HC-06, lograr establecer canal de comunicación inalámbrico presento mayores inconvenientes los cuales pueden generar un gran número de puntos de falla para su utilización como programación o carga inalámbrica robot y además se debe levantar servicios extras para lograr cometido.

En caso de Modulo HC-06 presenta menor complejidad establecer comunicación inalámbrica, como observación se hace presente que se debe soldar de forma manual pin 32 de modulo, no requiere levantar interfaces externas para efectuar carga sketch robot minimizando riesgos de problemas de comunicación y posibles puntos de falla.

Como resultado de investigación y pruebas se determina módulo HC-06 Bluetooth es la solución ideal para establecer programación o carga sketch inalámbrica en robot.

## FASE II

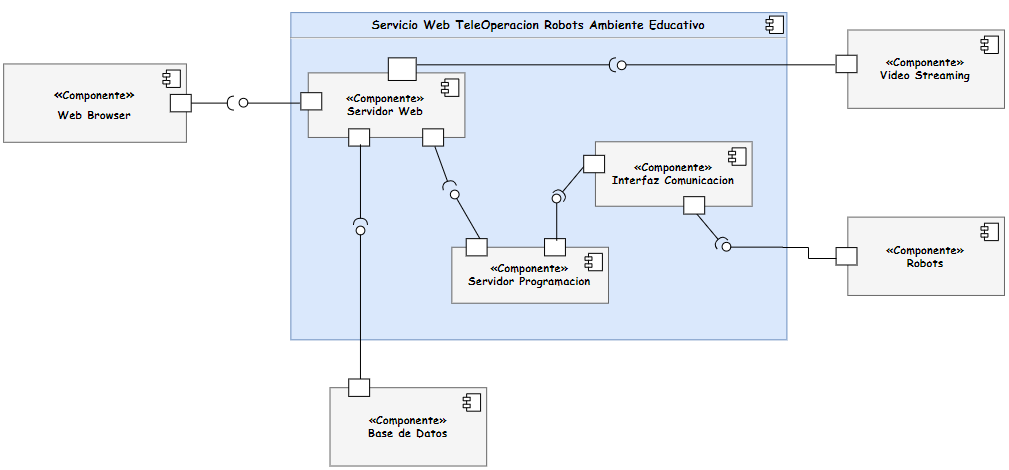
Esta fase de proyecto considera desarrollo e implementación de una WebApp que garantice el acceso remoto utilizando internet como medio de transmisión de datos permitiendo que sistema se encuentre disponibles para los usuarios independiente de su ubicación geográfica o acotado a un horario especifico

Para desarrollo de esta fase se considerará la utilización de Ruby on Rails que es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby. Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código que con otros frameworks y con un mínimo de configuración. El lenguaje de programación Ruby permite la metaprogramación, de la cual Rails hace uso, lo que resulta en una sintaxis que muchos de sus usuarios encuentran muy legible, además cuenta con siguientes características:

* Permite agilizar el proceso de desarrollo.
* Ayuda a simplificar tareas comunes y repetitivas.
* No tiene ningún costo de licencia.
* Amplia variedad de librerías adicionales o gemas que proporcionan funcionalidades comunes en proyectos web
* Fácilmente adaptable a metodología de desarrollo iterativo incremental.

## DISEÑO DETALLADO

La arquitectura utilizada para el diseño de la aplicación es una arquitectura Web de tres capas. La capa de los clientes, la capa del servidor de aplicaciones y la capa de la base de datos.

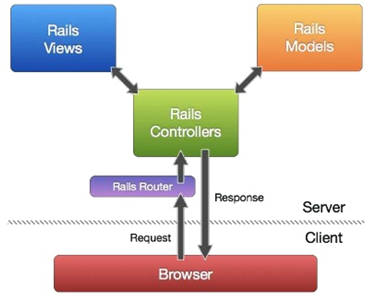


En la capa de cliente se sitúan como su nombre indican todas las máquinas cliente de la aplicación. Los clientes no contienen nada de la lógica de negocio de la aplicación. Toda la lógica de la aplicación está situada en la segunda capa o nivel, es decir, en el servidor de aplicación y en la base de datos se almacena toda la información que maneja la aplicación de manera permanente.

Cabe destacar, que los puestos de los clientes utilizan un modelo de cliente ligero. Debido a que toda la lógica de la aplicación se encuentra situada en el servidor de aplicaciones, los puestos de los clientes solo deben tener el software necesario para que el usuario pueda invocar la funcionalidad del servidor de aplicaciones y pueda visualizar el resultado de la ejecución. Debido a que se ha utilizado el protocolo HTTP, este software será un navegador.

Como es lógico, las comunicaciones entre las tres capas variarán dependiendo de las capas que se comunican. La comunicación entre los puestos de clientes y el servidor de aplicaciones se realizará a través de Internet. En cuanto a la comunicación del servidor de aplicaciones con la base de datos se realizará mediante una red interna de la entidad.

Para gestionar la interacción entre estas tres capas se utilizará patrón de arquitectura MVC (Model-View-Controller) el cual más que un patrón es una filosofía de diseño de aplicaciones en la que se define la organización independiente del modelo. La vista corresponde al entorno de usuario, el controlador es el motor de la aplicación y el modelo es la capa de datos



## PLAN DE PRUEBAS DE DESARROLLO DE SOFTWARE

Historial de Versiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Autor** | **Descripción** |
| **20-09-17** | **1.0** | **Pablo Sánchez Pino** | **Pruebas Desarrollo Fase I** |
| **10-05-18** | **2.0** | **Pablo Sánchez Pino** | **Pruebas Desarrollo Fase II** |

Información Proyecto

|  |  |
| --- | --- |
| Proyecto | Tele operación Remota Robots Ambiente Educativo |
| Fecha | 20 – Septiembre - 2017 |
| Cliente | Seminario Titulo |
| Patrocinador principal | UNAB |
| Líder de proyecto | Pablo Sánchez Pino |

Aprobaciones

|  |  |  |
| --- | --- | --- |
| **Nombre y Apellido** | **Cargo** | **Fecha** |
| **Pablo Sánchez Pino** | **Líder Proyecto** | **20 – 09 - 17** |
| **Pablo Sánchez Pino** | **Líder Proyecto** | **10 – 05 - 18** |

## ALCANCE DE LAS PRUEBAS FASE I.

## Elementos de pruebas

* Arduino.
* Diseño circuito
* Servomotor
* Módulo HC-06.
* Software Arduino.
* Programación base.
* Comunicación pc – Arduino.
* Cable USB.
* Conexión bluetooth.
* Operación inalámbrica Arduino.

## Funcionalidades por probar

* Conexión pc – Arduino.
* Carga sketch Arduino.
* Establecer canal de comunicación inalámbrico.
* Establecer paridad dispositivo externo (dispositivo móvil) – Arduino mediante Bluetooth.
* Efectuar operación Arduino (Led – Servomotores).

## Funcionalidades por no probar

Listado de las funcionalidades que no se van a probar durante Fase I sistema.

* Acceso remoto.
* Acceso por rol usuario Alumno – Instructor – Administrador.
* Selección actividad.
* Diseño actividad.
* Designar actividad.
* Gestión usuarios – estadísticas.
* Editar registros.
* Autentificación usuarios.
* Editar Usuarios.
* Tele operación Arduino.
* Retroalimentación por streaming de video.

## Enfoque de pruebas

Debido al alcance de la Fase I de este proyecto solo se considerará desarrollo de pruebas funcionales enfocados particularmente en esta fase. Pruebas de desempeño, interfaces y no funcionales se realizarán durante Fase 2 y Fase 3 respectivamente según correspondan.

Dejando claro párrafo anterior, se lista pruebas a realizar por parte usuario final.

* Ejecutar software Arduino en pc local. (USB)
* Efectuar o modificar código programación base en lenguaje de programación Arduino.
* Efectuar carga sketch Arduino.
* Establecer conexión Bluetooth dispositivo móvil – Arduino.
* Ejecutar movimientos mediante aplicación “ArduDroid”.

## Criterios de aceptación o rechazo

Debido al acotado alcance de las pruebas para fase I se considerará completar al 100% pruebas funcionales, de forma secuencial, anteriormente indicadas.

## Criterios de suspensión

A continuación, se establece claramente en qué condiciones o fallas críticas se puede detener ejecución de pruebas.

* No lograr comunicación pc – Arduino por defecto cable USB en mal estado o desconfiguración puerto serial.
* Problemas de código programación.
* No lograr paridad dispositivo móvil – Arduino.

## Criterios de reanudación

Objeto continuar con ejecución de pruebas, en caso de existir condiciones anteriormente indicadas, se deberá subsanar de forma inmediata y secuencial descartando fallas mayores que puedan ocasionar la completa inutilidad del sistema.

## Recursos

## Requerimientos de Hardware

Lista de los requerimientos de equipos, hardware y red necesarios para completar las actividades de pruebas Fase I sistema.

* Pc.
* Cable USB.
* Arduino UNO.
* Modulo Bluetooth HC-06.
* Servomotor.
* Protoboard – cables conexión.
* Dispositivo móvil con Bluetooth.

## Requerimientos Software

Lista de los requerimientos de software necesarios para completar las actividades de pruebas Fase1 sistema.

* S.O. Windows 7 en adelante.
* Software Arduino.
* Aplicación ArduDroid.

## Metodología Pruebas

Para esta etapa se considerará mayormente efectuar pruebas de “Caja Negra” objeto evaluar como también validar entradas y salidas sistema correspondiente a la operación dispositivo Arduino.

Además, considerar efectuar pruebas, en menor medida, de “Caja Gris” con la finalidad de evaluar la especificación de alto nivel del sistema y código programación Arduino

## Dependencias y Riesgos

Aquí se listan los riesgos asociados con el proceso de pruebas Fase1 de software.

* Disponibilidad de recursos.
* Restricciones de tiempo.
* Premisas que resulten no ser ciertas.

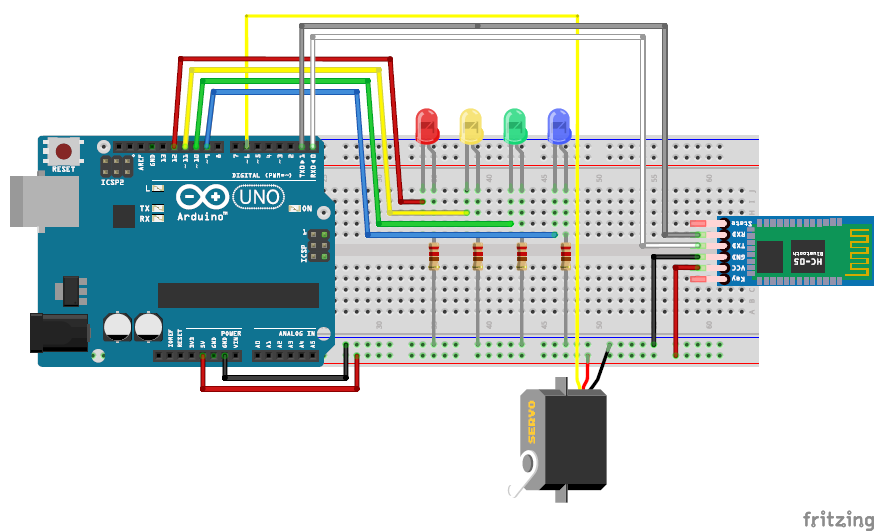
## Referencias

* Memoria Seminario.
* Documento Especificación Requisitos del Sistema.
* Cronograma Fase n°1.
* EDT.

## Avance Técnico Fase I

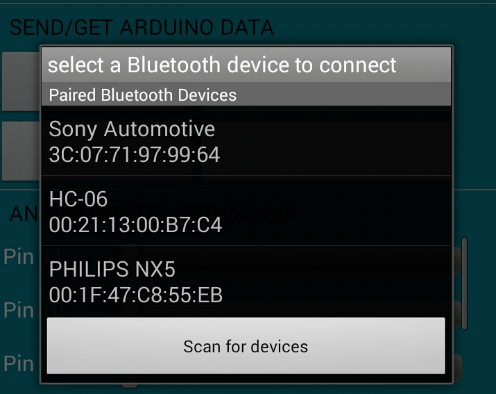
Para esta primera fase se desarrolla e implementa prototipo robot basado en Arduino sobre el cual se ejecuta código programación base que permite integración de distintos componentes de forma sencilla y simple. Prototipado 1 considera integración modulo Bluetooth HC-06 para efectuar activación led y servomotor una vez que se establece comunicación inalámbrica Arduino – Smartphone el cual cuenta con aplicación ArduDroid instalada que permite efectuar operación remota led y servomotor.

A continuación, se presenta diagrama conexionado prototipo Arduino.

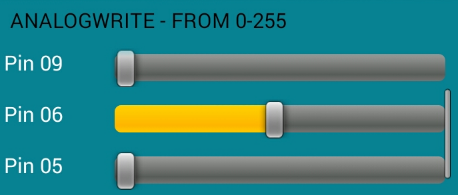


Para la programación de dispositivo Arduino se utilizó código base el cual se puede modificar de forma sencilla y simple para integración de distintos módulos o dispositivos. Antes de realizar carga considerar siguientes pasos:

* Desconectar modulo HC-06 para efectuar carga sketch en Arduino objeto evitar que se produzcan errores durante carga.
* Una vez efectuada carga sketch se debe conectar modulo HC-06 el cual comenzara a parpadear led integrado de color rojo indicando que dispositivo se encuentra listo.
* En smarphone habilitar conexiones Bluetooth y ejecutar aplicación ArduDroid.
* En app ArduDroid establecer paridad con modulo HC-06. (pass por defecto 1234)
  + *“Connect me to a Bluetooth device”.*
  + *HC – 06*



* Efectuada paridad encender y apagar led seleccionando salidas digitales 9/10/11/12/13. Salida digital 6 permite manejar servomotor en sentido reloj o contra reloj.

## Código Prototipo Fase I

/\*

PROYECTO TESIS

PROGRAMADOR: Pablo Sanchez Pino

Desconectar modulo Bluetooth mientras carga Sketch, luego conectar

ARDUINO BLUETOOTH

Pin 0 (RX) TXD

Pin 1 (Tx) RXD

5V VCC

GND GND

Descargar aplicacion Android en Play Store: ArduDroid

\*/

#define CARACTER\_INICIO\_CMD '\*'

#define CARACTER\_FINAL\_CMD '#'

#define CARACTER\_DIV\_CMD '|'

#define ESCRITURA\_DIGITAL\_CMD 10

#define ESCRITURA\_ANALOGA\_CMD 11

#define TEXTO\_CMD 12

#define LECTURA\_ARDUDROID\_CMD 13

#define MAX\_COMMAND 20

#define MIN\_COMMAND 10

#define LONGITUD\_ENTRADA\_STRING 40

#define ESCRITURA\_ANALOGICA\_MAX 255

#define PIN\_ALTO 3

#define PIN\_BAJO 2

#include <Servo.h>//Incluye la libreria Servo

Servo servo1;//Crea el objeto servo1 con las caracteristicas de Servo

String inText;

void setup() {

Serial.begin(9600);

Serial.println("Tele Operacion Arduino");

Serial.flush();

servo1.attach(7,600,1500);

}

void loop()

{

Serial.flush();

int ard\_command = 0;

int pin\_num = 0;

int pin\_value = 0;

char get\_char = ' '; //lee serial

// esperar a que los datos entren

if (Serial.available() < 1) return;//retornar al Loop().

// analizar entrada de indicador de inicio de comando

get\_char = Serial.read();

if (get\_char != CARACTER\_INICIO\_CMD) return;//volver loop ().

// parse incoming command type

ard\_command = Serial.parseInt(); // read the command

// analizar el tipo de comando entrante

pin\_num = Serial.parseInt(); // leer el pin

pin\_value = Serial.parseInt(); // leer el valor

// 1) OBTENER COMANDO DE TEXTO PARA ARDUDROID

if (ard\_command == TEXTO\_CMD){

inText =""; // borra variable para nueva entrada

while (Serial.available()) {

char c = Serial.read(); //recibe byte de la memoria intermedia serie

delay(5);

if (c == CARACTER\_FINAL\_CMD) { // si la cadena completa ha sido leida

// add your code here

break;

}

else {

if (c != CARACTER\_DIV\_CMD) {

inText += c;

delay(5);

}

}

}

}

// 2) OBTENER DATOS DE digitalWrite ARDUDROID

if (ard\_command == ESCRITURA\_DIGITAL\_CMD){

if (pin\_value == PIN\_BAJO) pin\_value = LOW;

else if (pin\_value == PIN\_ALTO) pin\_value = HIGH;

else return; // error en el valor de PIN. regresar.

set\_digitalwrite( pin\_num, pin\_value); //Eliminar el comentario para utilizarla

return; // regrese al inicio de loop()

}

// 3) GET analogWrite DATA FROM ARDUDROID

if (ard\_command == ESCRITURA\_ANALOGA\_CMD) {

analogWrite( pin\_num, pin\_value );

// add your code here

return; // Done. return to loop();

}

// 4) Enviar datos a ARDUDROID

if (ard\_command == LECTURA\_ARDUDROID\_CMD) {

// char send\_to\_android[] = " Coloca el texto aquí." ;

// Serial.println(send\_to\_android); // Ejemplo: Envío de texto

Serial.print(" Analogo 0 = ");

Serial.println(analogRead(A0)); //Leer y enviar valor analógico del Pin de Arduino

return; // Listoe. regrese al loop();

}

}

// 2a) seleccionar el pin # solicitado para la acción digitalWrite

void set\_digitalwrite(int pin\_num, int pin\_value)

{

switch (pin\_num) {

case 13:

pinMode(13, OUTPUT);

digitalWrite(13, pin\_value);

// adicione su código aquí, para este número de pin del Arduino

break;

case 12:

pinMode(12, OUTPUT);

digitalWrite(12, pin\_value);

// adicione su código aquí, para este número de pin del Arduino

break;

case 11:

pinMode(11, OUTPUT);

digitalWrite(11, pin\_value);

// adicione su código aquí, para este número de pin del Arduino

break;

case 10:

pinMode(10, OUTPUT);

digitalWrite(10, pin\_value);

// adicione su código aquí, para este número de pin del Arduino

break;

case 9:

pinMode(9, OUTPUT);

digitalWrite(9, pin\_value);

// adicione su código aquí, para este número de pin del Arduino

break;

case 8:

pinMode(8, OUTPUT);

digitalWrite(8, pin\_value);

// adicione su código aquí, para este número de pin del Arduino

break;

case 7:

servo1.write(0);// Gira el servo a 0 grados

delay(2000);// Espera 700 mili segundos a que el servo llegue a la posicion

servo1.write(90);// Gira el servo a 90 grados

delay(2000);

servo1.write(180);//Gira el servo a 180 grados

delay(2000);

case 6:

servo1.write(0);// Gira el servo a 0 grados

delay(2000);// Espera 700 mili segundos a que el servo llegue a la posicion

servo1.write(90);// Gira el servo a 90 grados

delay(2000);

servo1.write(180);//Gira el servo a 180 grados

delay(2000);

break;

case 5:

servo1.write(0);// Gira el servo a 0 grados

delay(2000);// Espera 700 mili segundos a que el servo llegue a la posicion

servo1.write(90);// Gira el servo a 90 grados

delay(2000);

servo1.write(180);//Gira el servo a 180 grados

delay(2000);

break;

case 4:

pinMode(4, OUTPUT);

digitalWrite(4, pin\_value);

// adicione su código aquí, para este número de pin del Arduino

break;

case 3:

pinMode(3, OUTPUT);

digitalWrite(3, pin\_value);

// adicione su código aquí, para este número de pin del Arduino

break;

case 2:

pinMode(2, OUTPUT);

digitalWrite(2, pin\_value);

// adicione su código aquí, para este número de pin del Arduino

break;

// por defecto

// si nada más fue seleccionado, hacer el defecto (default)

// default es opcional

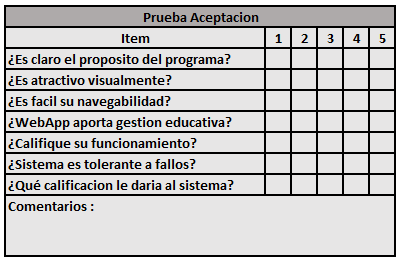
}

}

## Pruebas de Aceptación Fase I

Considera la participación usuaria frente al sistema, para esta fase, el ambiente de prueba no contará con una persona encargada de guiar al usuario, sino que su entorno de prueba será privado y la ejecución la estimará el usuario en un ambiente propicio. Se podrá incluir además sugerencias en cuanto a la interacción y manipulación del sistema

* Ejecutar software Arduino en pc local. (USB)
* Efectuar o modificar código programación base en lenguaje de programación Arduino.
* Efectuar carga sketch Arduino.
* Establecer conexión Bluetooth dispositivo móvil – Arduino.
* Ejecutar movimientos mediante aplicación “ArduDroid”.



## PRUEBAS DE SISTEMA

Siguientes etapas proyecto considerara efectuar pruebas funcionales pendientes como también pruebas no funcionales las cuales debido al acotado alcance de Fase I no fueron abordadas.

## ALCANCE DE LAS PRUEBAS FASE II.

## Elementos de pruebas

* Arduino.
* Software Arduino.
* Comunicación pc – Arduino.
* Cable USB.
* WebApp

## Funcionalidades por probar

* Conexión pc – Arduino.
* Carga sketch Arduino.
* Establecer canal de comunicación.
* Efectuar operación Arduino (Led – Blink).
* Acceso por rol usuario Alumno – Instructor – Administrador
* Autentificación usuarios.
* Editar Usuarios.

## Funcionalidades por no probar

Listado de las funcionalidades que no se van a probar durante Fase1 sistema.

* Selección actividad.
* Diseño actividad.
* Designar actividad.
* Gestión usuarios – estadísticas.
* Editar registros.
* Tele operación Arduino.
* Retroalimentación por streaming de video.

## Enfoque de pruebas

Fase II considerar y priorizara en mayor medida pruebas de desempeño, interfaces y no funcionales según correspondan.

Dejando claro párrafo anterior, se lista pruebas a realizar por parte usuario final.

* Acceso WebApp por rol
* Editar usuarios por parte de usuario Administrador
* Validar login Profesor o Instructor
* Usuario Profesor o Instructor crear catedra – clases
* Validar login usuario Alumno.
* Establecer conexión arduino.
* Usuario Alumno ejecutar Led-Blink.

## Criterios de aceptación o rechazo

Fase 2 considerará completar al 100% pruebas funcionales, de forma secuencial, anteriormente indicadas.

## Criterios de suspensión

A continuación, se establece claramente en qué condiciones o fallas críticas se puede detener ejecución de pruebas.

* No lograr comunicación pc – Arduino por defecto cable USB en mal estado o desconfiguración puerto serial.
* Problemas codificación.
* Insuficiencia de hardware propio.

## Criterios de reanudación

Objeto continuar con ejecución de pruebas, en caso de existir condiciones anteriormente indicadas, se deberá subsanar de forma inmediata y secuencial descartando fallas mayores que puedan ocasionar la completa inutilidad del sistema.

## Recursos

## Requerimientos de Hardware

Lista de los requerimientos de equipos, hardware y red necesarios para completar las actividades de pruebas Fase 2 sistema.

* Pc - Cable USB.
* Arduino UNO.
* Protoboard – Led - cables conexión.

## Requerimientos Software

Lista de los requerimientos de software necesarios para completar las actividades de pruebas Fase II sistema.

* S.O. Windows 7 en adelante.
* Software Arduino.
* WebApp.
* Web Browser.
* NetBeans IDE 8.2.
* NodeJS.
* RubyonRails.
* Sublime Text.
* Postman.

## Metodología Pruebas

Para esta etapa se considerará efectuar pruebas de “Caja Gris” con la finalidad de evaluar la especificación de alto nivel del sistema y código programación Arduino, además de pruebas funcionales las cuales se detallan en pto. **4.5.6**

## Dependencias y Riesgos

Aquí se listan los riesgos asociados con el proceso de pruebas Fase II de software.

* Disponibilidad de recursos.
* Restricciones de tiempo.
* Premisas que resulten no ser ciertas.
* Hardware propio limitado.
* Poca experiencia en codificación y lenguaje utilizado.

## Referencias

* Memoria Seminario.
* Documento Especificación Requisitos del Sistema.
* Cronograma Fase II.
* EDT.

## Pruebas Funcionales Fase II

|  |  |
| --- | --- |
| Prueba Funcional | PF01 |
| **Descripción:** Autentificación usuarios independiente de su rol | |
| **Prerrequisitos**   * Browser Internet Explorer versión 9 o superior, Mozilla Firefox, Google Chrome. * Acceso solo usuarios registrados. | |
| **Pasos:**   * Ejecutar browser. * Ingresar nombre usuario en WebApp. * Ingresar contraseña en WebApp. * Acceso sistema. | |
| **Resultado esperado:** Sistema debe permitir a usuario, independiente de su rol, acceder remotamente a plataforma. | |
|  | |
| Prueba Funcional | PF02 |
| **Descripción:** Selección actividad usuario “Alumno” | |
| **Prerrequisitos**   * Browser Internet Explorer versión 9 o superior, Mozilla Firefox, Google Chrome. * Usuario “Alumno” registrados. | |
| **Pasos:**   * Acceso sistema. * Revisar existencia o disponibilidad actividad práctica. * Seleccionar actividad. * Visualizar actividad. * Opcional selección actividad libre. | |
| **Resultado esperado:** Sistema deberá desplegar actividades disponibles o permitir seleccionar actividad libre. | |
|  | |
| Prueba Funcional | PF03 |
| **Descripción:** Ingreso actividad usuario “Alumno” | |
| **Prerrequisitos**   * Browser Internet Explorer versión 9 o superior, Mozilla Firefox, Google Chrome. * Usuario “Alumno” registrados. | |
| **Pasos:**   * Acceso sistema. * Selección tipo de actividad. * Ingresar actividad. * Efectuar actividad. | |
| **Resultado esperado:** Sistema deberá permitir el ingreso a usuario “Alumno” a actividades practicas o actividad libre. | |

|  |  |
| --- | --- |
| Prueba Funcional | PF03 |
| **Descripción:** Gestión actividades prácticas. | |
| **Prerrequisitos**   * Browser Internet Explorer versión 9 o superior, Mozilla Firefox, Google Chrome. * Usuario “Instructor” registrados. | |
| **Pasos:**   * Acceso sistema * Seleccionar “Subir actividad”. (plantilla con descripción actividad y base código lenguaje Arduino) * Confirmar actividad. * Visualizar y validar actividad | |
| **Resultado esperado:** Sistema deberá distinguir por rol a usuario “Instructor” y diferenciarlo de rol usuario “Alumno”, asegurando pantalla inicio sistema cuente con opción para “Subir actividad”. | |

|  |  |
| --- | --- |
| Prueba Funcional | PF03 |
| **Descripción:** Designar actividades. | |
| **Prerrequisitos**   * Browser Internet Explorer versión 9 o superior, Mozilla Firefox, Google Chrome. * Usuario “Instructor” registrados. | |
| **Pasos:**   * Seleccionar actividad a designar previamente subida a sistema. * Seleccionar “Asignar actividad” * Seleccionar si asignación actividad será realizar por Alumno – Grupo – Curso. * Confirmar “Asignar actividad2. | |
| **Resultado esperado:** Sistema deberá permitir a usuario “Instructor” designar actividad, predefinidas, por alumno, grupo o curso. | |

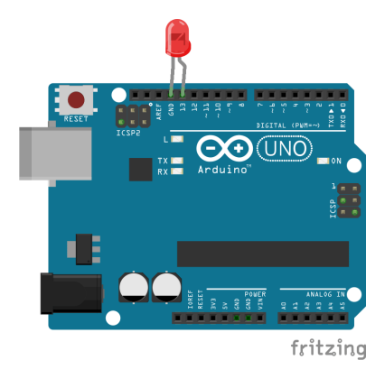
|  |  |
| --- | --- |
| Prueba Funcional | PF03 |
| **Descripción:** Agregar usuarios | |
| **Prerrequisitos**   * Browser Internet Explorer versión 9 o superior, Mozilla Firefox, Google Chrome. * Acceso solo usuarios registrados. (Alumno – Instructor) * Software Arduino. * Bluetooth | |
| **Pasos:**   * Acceso sistema * Seleccionar “Ingresar usuarios”. * Completar planilla con datos nuevo usuario válidos UNAB. * Enviar mail de confirmación notificando nombre usuario y contraseña | |
| **Resultado esperado:** Sistema deberá permitir que solamente usuario “Administrador” pueda dar de alta a nuevos usuarios en plataforma | |

|  |  |
| --- | --- |
| Prueba Funcional | PF03 |
| **Descripción:** Editar registros | |
| **Prerrequisitos**   * Browser Internet Explorer versión 9 o superior, Mozilla Firefox, Google Chrome. * Usuario “Administrador” registrado. | |
| **Pasos:**   * Acceso sistema * Seleccionar “Registros”. * Descargar plantilla con datos usuarios registrados en sistema. | |
| **Resultado esperado:** Sistema deberá permitir solamente usuario “Administrador” tener acceso a registros de usuarios y sus actividades objeto editar o eliminar en caso de ser necesario. | |

## Avance Técnico Fase II

## Prototipado I

Esta fase se subdivide en el desarrollo de dos prototipos. Primer prototipado consiste en un simple Led-Blink el cual se activa y desactiva mediante aplicación desarrollada en JavaScript, utilizando librerías para lograr establecer la comunicación serial, permitiendo encender o apagar led en pin 13 Arduino.



Para la programación de dispositivo Arduino se utilizó código base el cual se puede modificar de forma sencilla y simple para integración de distintos módulos o dispositivos.

//Creado por Pablo Sanchez Pino

void setup() {

//Iniciamos la comunicación con el puerto serie

pinMode(13,OUTPUT);

Serial.begin(9600);

}

void loop() {

if (Serial.available() >0){

int input = Serial.read();

if (input =='1') {

digitalWrite(13, HIGH);

}

else

{

if (input =='0')

digitalWrite(13, LOW);

}

}

}

A continuación, se especifica codificación en JavaScript ejecutado en NetBeans IDE que entrega interfaz con la cual podemos encender o apagar led 13 arduino.

/\*

\* Creado por Pablo Sanchez Pino

\*/

package GUI;

import java.util.logging.Level;

import java.util.logging.Logger;

import panamahitek.Arduino.PanamaHitek\_Arduino;

public class Formulario extends javax.swing.JFrame {

PanamaHitek\_Arduino arduino = new PanamaHitek\_Arduino();

public Formulario() {

initComponents();

try {

arduino.arduinoTX("COM5", 9600);

} catch (Exception ex) {

Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);

}

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jButton1.setText("ON");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton2.setText("OFF");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(43, 43, 43)

.addComponent(jButton1)

.addGap(18, 18, 18)

.addComponent(jButton2)

.addContainerGap(45, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(84, 84, 84)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton1)

.addComponent(jButton2))

.addContainerGap(52, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try {

arduino.sendData("1");

} catch (Exception ex) {

Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

try {

arduino.sendData("0");

} catch (Exception ex) {

Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Formulario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Formulario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Formulario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Formulario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Formulario().setVisible(true);

}

});

}

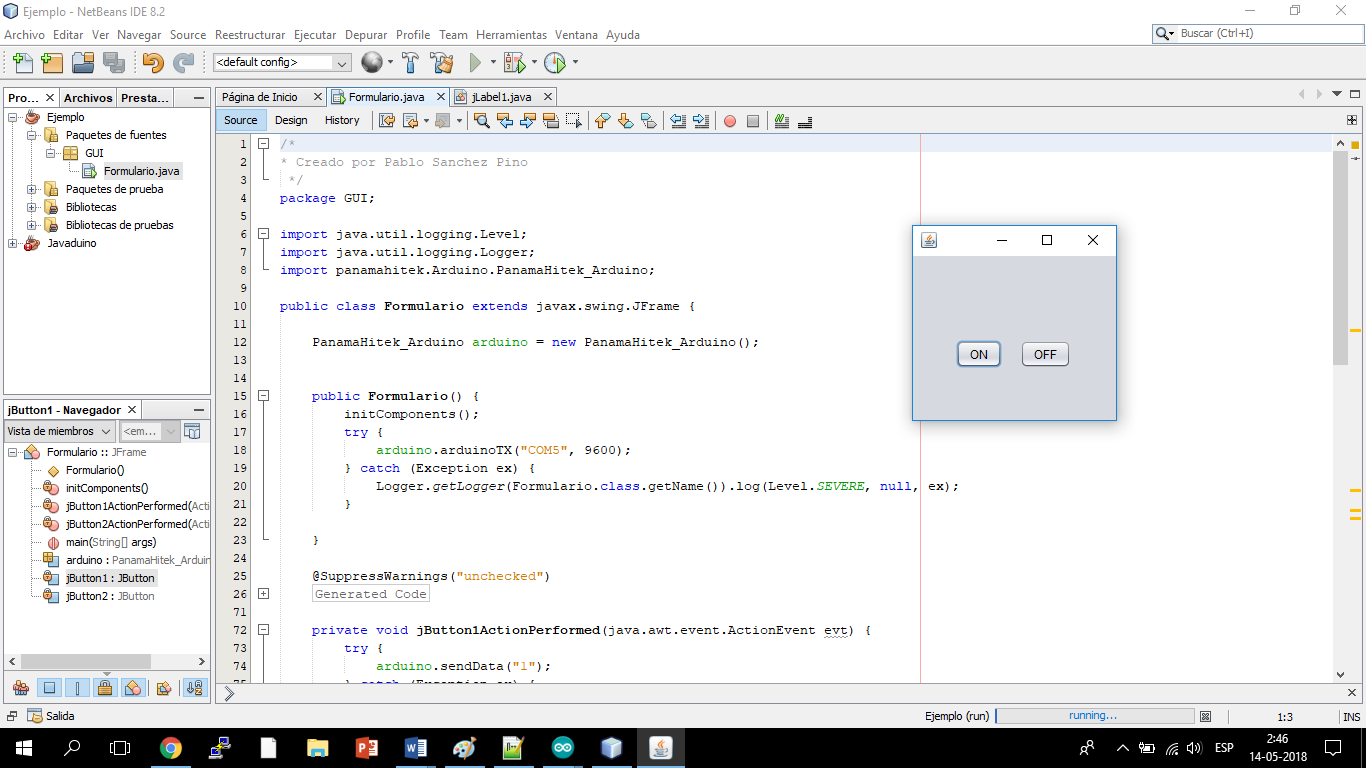
// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

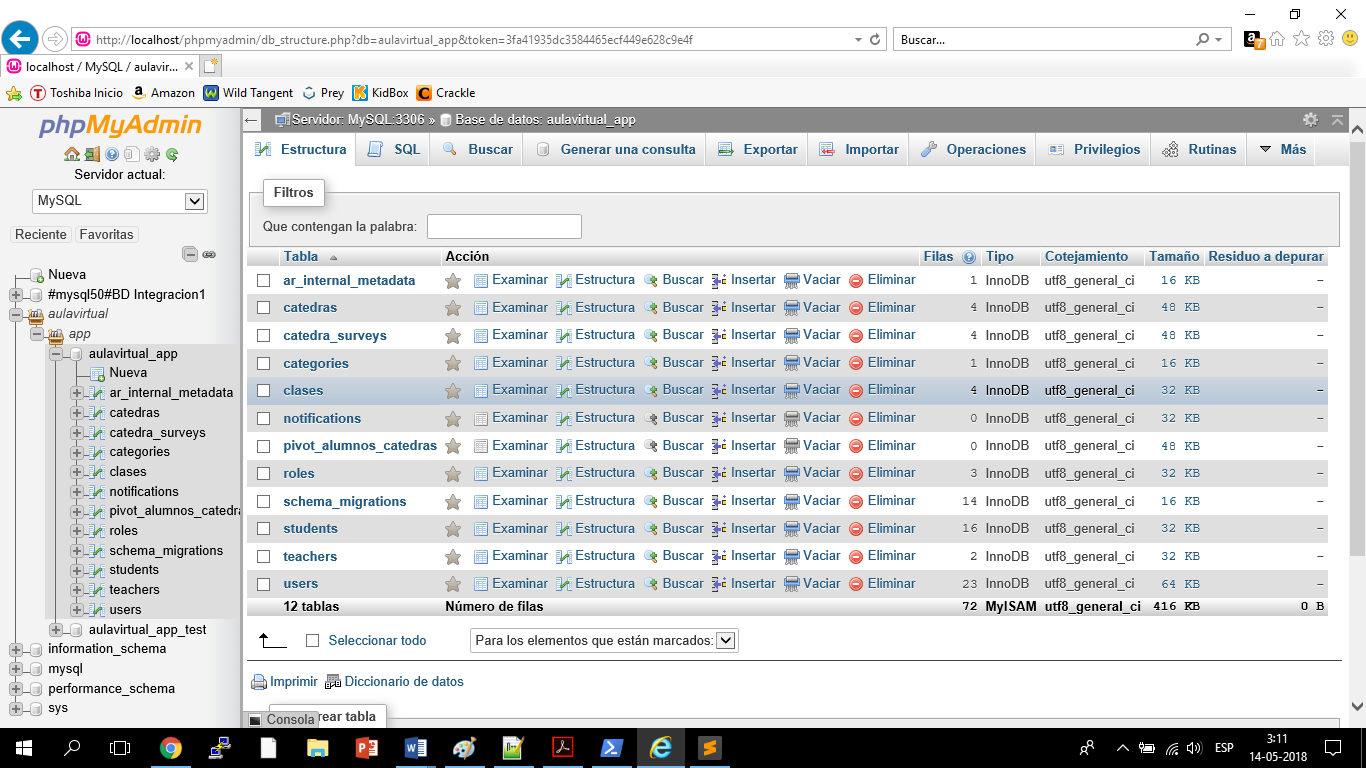
// End of variables declaration

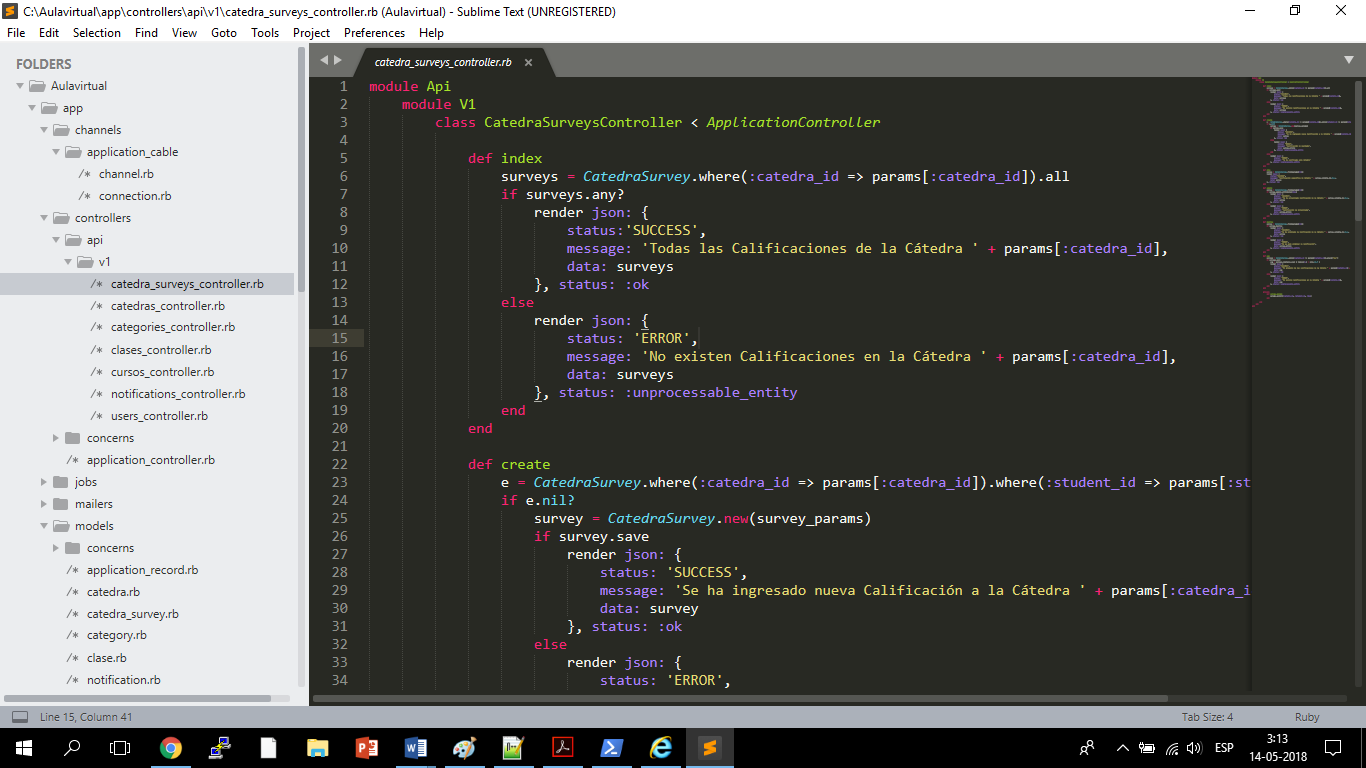
}

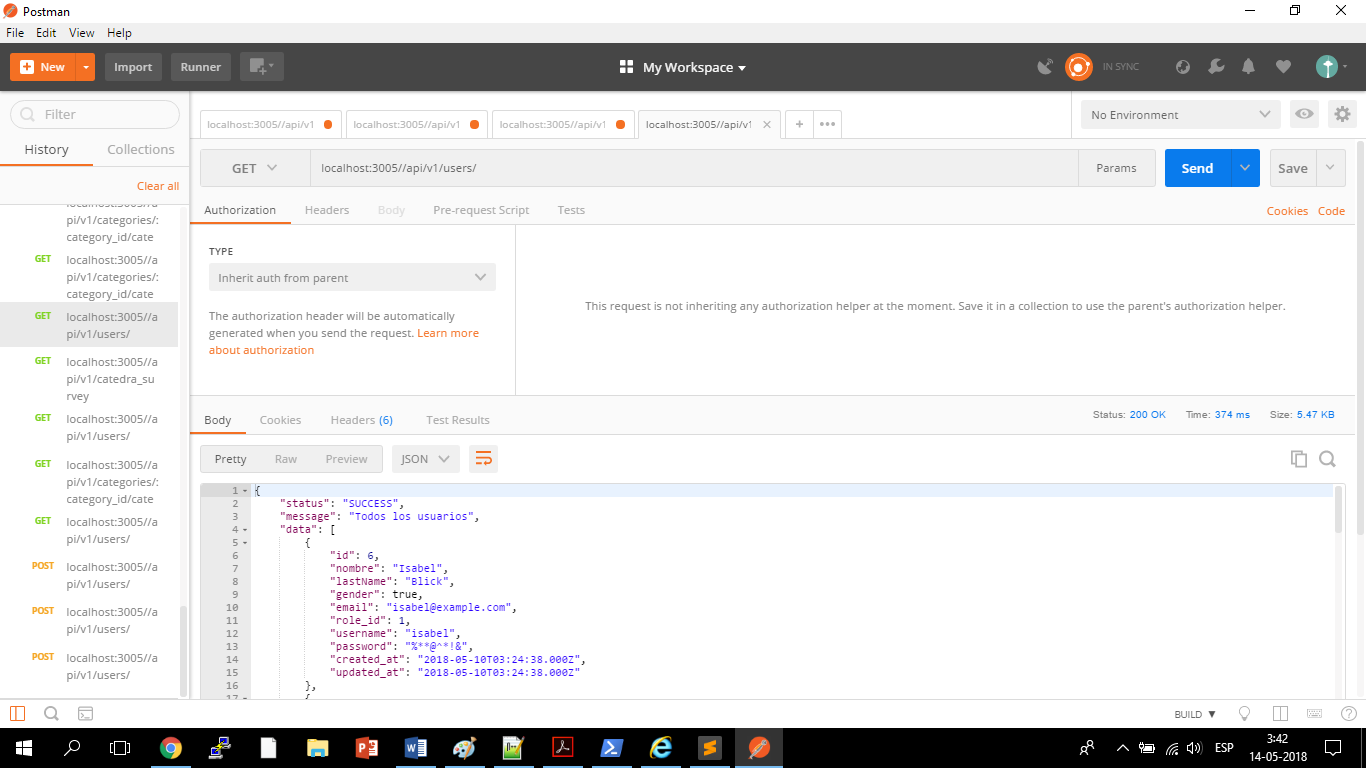


## Prototipado II

Para este prototipo se desarrolla mediante framework Ruby on Rails lo que será la codificación backend futura aplicación web y correspondiente base de datos para gestionar usuarios. Debido a que desarrollo se encuentra en proceso, se puebla tabla usuarios y mediante postman se efectuaran pruebas objeto validar base de datos y codificación backend.

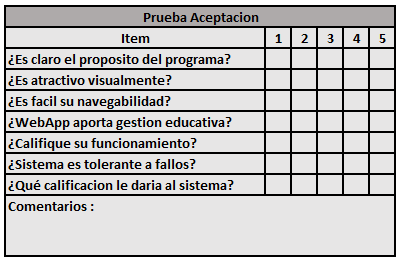






## Pruebas de Aceptación Fase II

Al igual que en Fase I se considera la participación del usuario frente al sistema, para esta fase, el ambiente de prueba no contará con una persona encargada de guiar al usuario, sino que su entorno de prueba será privado y la ejecución la estimará el usuario en un ambiente propicio. Se podrá incluir además sugerencias en cuanto a la interacción y manipulación del sistema



## Fase III

|  |  |
| --- | --- |
| Prueba Integración | PF03 |
| **Descripción:** Comunicación servidor - Arduino | |
| **Prerrequisitos**   * Browser Internet Explorer versión 9 o superior, Mozilla Firefox, Google Chrome. * Usuario “Alumno” registrado. * Software Arduino. | |
| **Pasos:**   * Acceso sistema. * Selección tipo de actividad. * Ingresar actividad. * Efectuar actividad. * Establecer canal Bluetooth de comunicación * Efectuar operación remota Arduino | |
| **Resultado esperado:** Sistema deberá permitir usuario “Alumno” levantar canal de comunicación inalámbrico para efectuar operación y movimientos con dispositivo Arduino. | |

|  |  |
| --- | --- |
| Prueba Integración | PF03 |
| **Descripción:** Comunicación servidor - Arduino | |
| **Prerrequisitos**   * Browser Internet Explorer versión 9 o superior, Mozilla Firefox, Google Chrome. * Usuario “Instructor” registrado. * Software Arduino. | |
| **Pasos:**   * Acceso sistema * Seleccionar “Subir actividad”. (plantilla con descripción actividad y base código lenguaje Arduino) * Confirmar actividad. * Establecer canal Bluetooth de comunicación * Efectuar operación remota Arduino para validar operación. | |
| **Resultado esperado:** Sistema deberá permitir usuario “Instructor” levantar canal de comunicación inalámbrico para efectuar operación y movimientos con dispositivo Arduino para validar y certificar actividad práctica. | |

|  |  |
| --- | --- |
| Prueba Integración | PF03 |
| **Descripción:** Gestión de estadísticas | |
| **Prerrequisitos**   * Browser Internet Explorer versión 9 o superior, Mozilla Firefox, Google Chrome. * Usuario “Administrador” registrado. * Software Arduino. | |
| **Pasos:**   * Acceso sistema * Seleccionar “Estadísticas”. * Descargar plantilla con datos usuarios registrados en sistema. | |
| **Resultado esperado:** Sistema deberá permitir solamente a usuario “Administrador” obtener reporte de estadísticas actividades efectuadas por usuarios “Alumno” – “Instructor”. Además, entregar reportes usuarios “Alumno” según corresponda para evaluación por parte de usuarios “Instructor” | |

## Evidencia Gestión de Riesgo

*“Riesgo se define como la probabilidad que una amenaza transforme en un desastre”*

Fase I proyecto se consideró la utilización de módulo ESP8266 para establecer canal de comunicación inalámbrico con Arduino. Durante proceso de desarrollo prototipo Fase1 integración módulo ESP8266 con dispositivo Arduino dificulto levantar la comunicación inalámbrica, además no fue del todo eficaz y optimo debido a complejidad de configuraciones y programación Arduino.

Para ello se tomó en cuenta “PLAN DE GESTION DE RIESGOS” anteriormente abordado en esta tesis y se trazó con riesgo de HARDWARE (GR-04 Fallas de implementación - GR-06 Incompatibilidad tecnológica) e IMPLEMENTACION (GR-17 No lograr integración sistema) con un alto impacto en desarrollo sistema

## Evidencia Gestión de Cambio

* Identificación necesidad de cambio: Reemplazo tecnológico Wireless (ESP8266) a Bluetooth (HC-06).
* Evaluación petición cambio: “Aprobado” modulo bluetooth cumple requisitos funcionales y no funcionales Fase 1 proyecto.
* Diseño implementación: Simple y sencilla implementación, no requiere conexiones adicionales. Software Arduino cuenta con librería para ejecutar código base de programación.
* Auditoria calidad: Se logra establecer canal inalámbrico Arduino – Smartphone vía bluetooth logrando la tele operación remota.

## POSTMORTEM

Fase I de esta tesis ha presentado un sistema de tele operación para dispositivo Arduino basado en comunicación por bluetooth. La solución implementada garantiza la manipulación remota mediante smartphone y aplicación ArduDroid.

Se creo prototipo basado en dispositivo Arduino que permite demostrar la tele operación remota de los componentes electrónicos utilizados para pruebas (led – servomotor). Además, se implementó código de programación lenguaje Arduino simple y sencillo, objeto poder modificarlo para integración de distintos componentes.

Esta etapa se implementó completamente con software libre, demostrando que es posible realizar una implementación de mediana complejidad con software libre no necesariamente propietario o comercial logrando mantener proyecto a “Bajo Costo”. Por otra parte, dispositivos Arduino cuentan con una serie de kit de robótica y comunicación inalámbrica bluetooth incorporada que se pueden utilizar en ambiente educativo.

* Fase II se desarrolla 2 prototipos, el primero consiste en una aplicación desarrollada en JavaScript la cual enciende y apaga led en Arduino de la cual se espera integrar prontamente a webapp. Segundo prototipo consiste en desarrollo, en progreso, backend webapp que permita el acceso remoto vía internet al ServidorLab, se genera base de datos, controladores, ruta, entre otros, con la finalidad de poder efectuar gestión usuarios.

## TRABAJOS FUTUROS

Varios son los trabajos que implementar, por ello proyecto se dividió en 3 Fases las cuales se ajustan a calendario académico, principalmente etapas o fases siguientes son las siguientes

* Fase III “Integración”: Completada y visada fases anteriores se deberá efectuar desarrollo frontend e integración completa del sistema objeto usuarios, independientes de su rol, puedan realizar, desarrollar o designar actividades prácticas de robótica educativa y obtener retroalimentación mediante streaming de video de dichas prácticas.