

UNIVERSIDAD LAICA “ELOY ALFARO” DE MANABÍ

FACULTAD DE CIENCIAS INFORMÁTICAS



TEMA:

AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA “ELOY ALFARO” DE MANABÍ.

**TRABAJO DE TITULACIÓN MODALIDAD PROYECTO INTEGRADOR,
PREVIO A LA OBTENCIÓN DEL TÍTULO DE:**

INGENIERO EN SISTEMAS

AUTORES:

REYES ZAMBRANO WELLINGTON STEVEN

QUIROZ MOREIRA JONATHAN JESÚS

DIRECTOR:

ING. PEDRO DELGADO FRANCO Mg.

MANTA – MANABI – ECUADOR

2017



CERTIFICACIÓN

En mi calidad de director de trabajo de titulación de la Facultad de Ciencias Informáticas de la Universidad Laica “Eloy Alfaro” de Manabí certifico:

Haber dirigido y revisado el trabajo de titulación sobre el tema “AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA “ELOY ALFARO” DE MANABÍ”, considero que el mencionado trabajo investigativo cumple con los requisitos y tiene los méritos suficientes para ser sometido a la evaluación del jurado examinador que las autoridades de la Facultad de Ciencias Informáticas Designen.

La autoría del tema desarrollado, corresponde a los señores REYES ZAMBRANO WELLINGTON STEVEN, QUIROZ MOREIRA JONATHAN JESÚS, estudiantes con estudios concluidos en el periodo académico 2016-2017, quienes se encuentran aptos para la defensa.

Particular que certifico para los fines, salvo disposición de Ley en contrario.

Lo certifico:

ING.PEDRO DELGADO FRANCO Mg.
Director de trabajo de Titulación



UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ
Creada el 13 de noviembre de 1985 mediante Decreto Ley No.10, publicado en el Registro Oficial No. 313
FACULTAD DE CIENCIAS INFORMÁTICAS
Creada, Resolución H. Consejo Universitario del 11 de Julio del 2001



TRABAJO DE TITULACIÓN MODALIDAD PROYECTO INTEGRADOR,
PREVIO A LA OBTENCIÓN DEL TÍTULO DE: INGENIERO EN SISTEMAS

“AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE
INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE
DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE
CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA “ELOY ALFARO”
DE MANABÍ.”

Tribunal examinador que declara APROBADO el Grado de INGENIERO
EN SISTEMAS, de los señores: QUIROZ MOREIRA JONATHAN JESUS,
REYES ZAMBRANO WELLINGTON STEVEN

Ing. Delgado Muentes Wilian, Mg. _____

Ing. Machuca Avalos Mike, Mg. _____

Ing. Molina Loor Winther, Mg. _____

Manta, 21 de agosto de 2017



DECLARACIÓN EXPRESA

La responsabilidad del contenido de este trabajo de titulación, cuyo tema es:

“AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA “ELOY ALFARO” DE MANABÍ”, corresponde a Reyes Zambrano Wellington Steven, con C.I. 131402948-7 y Quiroz Moreira Jonathan Jesús, con C.I. 131076276-8 exclusivamente y los derechos patrimoniales de la misma a la Universidad Laica “Eloy Alfaro” de Manabí.

REYES ZAMBRANO WELLINGTON STEVEN

C.I # 131402948-7

QUIROZ MOREIRA JONATHAN JESÙS

C.I # 131076276-8



DEDICATORIA

Dedico este trabajo a mi Madre, Sra. Ángela Quiroz Moreira que con su esfuerzo y dedicación me ha apoyado a lo largo de toda mi vida brindándome la oportunidad de recibir estudios. Siendo madre y padre a la vez dedicada en todo momento en mi formación académica brindándome su confianza y pudiendo así cumplir la meta de ser Profesional.

A mi esposa Sra. Andrea Cadena Oviedo y a mi hija Ágata Quiroz Cadena por estar conmigo en cada momento que quise tirar la toalla y brindadme ese apoyo en los momentos de tristeza y alegría junto a nuestros familiares.

A mi tío Máximo Quiroz Moreira por siempre estar hay en todo momento de mi vida porque si su apoyo y consejos no sería la persona que soy en estos momentos.

A mis profesores que han estado en mi formación académica al largo de mi vida en especial a mi director de trabajo de titulación ING. Pedro Delgado Franco, por su tiempo, experiencia y motivación para la culminación de este trabajo de titulación.

“Quiero lograrlo porque me dijeron que no podría hacerlo.”

Carl Brashear 19 de enero de 1931 – 25 de julio de 2006.

Quiroz Moreira Jonathan Jesús



DEDICATORIA

Dedico este trabajo a mi madre, la Sra. Ángela Regina Zambrano Pazmiño, que con su amor y apoyo pudo lograr que sea una buena persona y estudiante, que, brindando toda su sabiduría, pudo lograr realizar el sueño de ambos.

A mi padre, el Sr. Carlos Bernardo Reyes Macías, que, estando conmigo pudo lograr lo que ni otro ser hizo, ya que nunca me faltó un alimento, un lápiz o un cuaderno necesarios para mi aprendizaje y crecimiento.

A mi novia, la Srta. Génesis Dayanna Mendoza Moreira, quien estuvo conmigo en todo momento desde el primer día en que nos conocimos, brindándome su amor y apoyo incondicional, los cuales me permitieron dar ese esfuerzo necesario y no rendirme en el camino, siendo una gran compañía y ayuda idónea en mi vida y si Dios lo permite, en un futuro siendo mi gran esposa.

A mis amigos, compañeros y demás familiares, los cuales me brindaron su apoyo en el momento que lo necesitaba, ya que todas y cada una de las personas que han llegado a mi vida, han sido de bendición para poder llegar hasta donde estoy, son grandes personas a las cuales admiro y agradezco por estar ahí conmigo.

“La vida es una serie de colisiones con el futuro; no es una suma de lo que hemos sido, sino de lo que anhelamos ser”.-José Ortega y Gasset.

Reyes Zambrano Wellington Steven



AGRADECIMIENTO

Primeramente, agradezco a dios por permitirme llegar a estos momentos y brindarme su bendición en todo momento a mi mamá por su apoyo incondicional: Ángela Quiroz Moreira por ser el pilar principal a lo largo de este largo camino para alcanzar uno de mis sueños, por todo el sacrificio que tuvo que pasar para brindarme el estudio.

Al Ing. Pedro Delgado Franco director de esta investigación, por la orientación y el seguimiento continuo de la misma, por el apoyo y la motivación brindaba a lo largo del desarrollo del proyecto integrador.

Finalmente, a la Sra. Decana Lcda., Dolores Muños Verduga y a los profesores de la facultad de ciencias Informáticas por brindar los conocimientos necesarios para superar los obstáculos y cumplir las metas propuestas.

Quiroz Moreira Jonathan Jesús



AGRADECIMIENTO

Quiero dar gracias primeramente a Dios por permitirme haber llegado hasta este punto en mi vida, dándome sabiduría, fuerzas y salud para lograr las metas que he logrado hasta el momento.

Agradezco a mis padres por apoyarme en todo momento de mi vida, desde que llegué a este mundo y hasta que me vaya, por brindarme su amor y comprensión, aconsejándome en todo momento para tomar las mejores decisiones y ser una mejor persona, a mi madre la Sra. Ángela Regina Zambrano Pazmiño, y a mi padre que en todo el momento que estuvo conmigo, el Sr. Carlos Bernardo Reyes Macías.

Al ing. Pedro Delgado Loor, tutor y guía del presente trabajo, por el saber guiarnos hasta este punto y haber finalizado con esmero y correctamente el trabajo de investigación, dedicando su tiempo dentro y fuera de su jornada laboral.

A la Sra. Decana Lcda. Dolores Muñoz Verduga, a los profesores, personal administrativo, secretarias, y conserjes de la facultad, quienes estuvieron ayudándome en toda la jornada estudiantil, conociendo todas y cada una de las etapas universitarias que hasta el momento fueron necesarias para poder lograr superar las pruebas que estaban en el camino.

Reyes Zambrano Wellington Steven



INDICE

DECLARACIÓN EXPRESA	IV
DEDICATORIA	V
AGRADECIMIENTO	VII
RESUMEN.....	XV
INTRODUCCIÓN	XVI
UBICACIÓN Y CONTEXTUALIZACIÓN DE LA PROBLEMÁTICA.....	XVII
PROBLEMATIZACIÓN	XVIII
Planteamiento del problema.....	XVIII
OBJETIVOS	XIX
Objetivo General	XIX
Objetivos Específicos.....	XIX
JUSTIFICACION	XX
CAPÍTULO I:	- 1 -
MARCO TEÓRICO.....	- 1 -
1.1.- INTRODUCCIÓN	- 2 -
1.2.- ANTECEDENTES DE INVESTIGACIÓN RELACIONADA AL TEMA	- 2 -
1.3.- DEFINICIÓN CONCEPTUAL	- 3 -
1.3.1.- Cerradura	- 3 -
1.3.1.1.- Definición	- 3 -
1.3.1.2.- Partes de una cerradura	- 4 -
1.3.1.3.- Tipos de cerraduras	- 5 -
1.3.2.- Teléfono inteligente	- 6 -
1.3.2.1.- Definición	- 6 -
1.3.2.2.- Teléfono inteligente a utilizar	- 6 -
1.3.2.3.- Teléfono inteligente y su sistema operativo	- 6 -
1.3.3- Automatización.....	- 7 -
1.3.3.1.- Definición	- 7 -
1.3.3.2.- Automatización programable	- 7 -
1.3.4.- Lenguajes de programación	- 8 -
1.3.4.1.- Definición	- 8 -
1.3.4.2.- Lenguaje de programación Java	- 8 -
1.3.4.3.- Uso del lenguaje Java para el dispositivo	- 8 -
1.3.4.4.- Lenguaje Java y dispositivo móvil	- 9 -
1.3.4.5.- Lenguaje Java y computadora personal	- 9 -
1.3.5.- FUNDAMENTOS BÁSICOS DE ELECTRÓNICA	- 9 -



1.3.5.1.- Definición.....	- 9 -
1.3.6.- Circuito eléctrico.....	- 10 -
1.3.6.1.- Definición.....	- 10 -
1.3.6.2.- Componentes del circuito electrónico	- 11 -
1.3.7.- RELAY	- 12 -
1.3.7.1.- Definición.....	- 12 -
1.3.7.2.- Tipos de relés	- 12 -
1.3.7.3.- Modulo rele	- 13 -
1.3.8.- ELECTRONICA DE CONTROL	- 14 -
1.3.8.1.- Definición.....	- 14 -
1.3.9.- PLATAFORMA ELECTRÓNICA ARDUINO	- 14 -
1.3.9.1.- Definición.....	- 14 -
1.3.9.2.- Modelos de Arduino	- 15 -
1.3.9.3.- Arduino uno	- 17 -
1.3.9.4.- Características Arduino Uno R3.....	- 17 -
1.3.9.5.- Ide Arduino.....	- 17 -
1.4.- CONCLUSIONES	- 18 -
CAPÍTULO II:	- 20 -
MARCO METODOLÓGICO.....	- 20 -
2.1. – INTRODUCCIÓN	- 21 -
2.2. TIPOS DE INVESTIGACIÓN	- 21 -
2.3. MÉTODOS DE INVESTIGACIÓN	- 22 -
2.4. HERRAMIENTAS DE RECOLECCIÓN DE DATOS	- 22 -
2.4.1 ENCUESTA - ENTREVISTA - OBSERVACIÓN / OTRAS	- 22 -
2.5. FUENTES DE INFORMACIÓN DE DATOS	- 23 -
2.5.1. FUENTES PRIMARIAS – FUENTES SECUNDARIAS	- 23 -
2.6 INSTRUMENTAL OPERACIONAL	- 24 -
2.6.1 ESTRUCTURA Y CARACTERISTICAS DE LOS INSTRUMENTOS DE RECOLECCIÓN DE DATOS.....	- 24 -
2.7. ESTRATEGIA OPERACIONAL PARA LA RECOLECCIÓN Y TABULACIÓN DE DATOS	- 25 -
2.7.1. PLAN DE RECOLECCIÓN, TABULACIÓN, ANÁLISIS E INTERPRETACIÓN DE DATOS	- 25 -
2.8. MUESTREO	- 26 -
2.8.1. SEGMENTACIÓN Y TAMAÑO DE LA MUESTRA	- 26 -
2.9. PRESENTACIÓN Y ANÁLISIS DE LOS RESULTADOS.....	- 27 -
2.9.1. PRESENTACIÓN Y DESCRIPCIÓN DE LOS RESULTADOS OBTENIDOS	- 27 -
ENCUESTA REALIZADA A LOS ESTUDIANTES DE LA FACCI QUE RECIBEN CATEDRA EN LOS LABORATORIOS DE INFORMATICA	- 28 -
II. INFORMACIÓN SOBRE TARJETAS ARDUINO.....	- 30 -
3. ¿Conoce usted sobre las tarjetas ARDUINO?	- 30 -
II. VINCULACIÓN	- 36 -



CAPÍTULO III:.....	- 40 -
MARCO PROPOSITO	- 40 -
3.1 – INTRODUCCIÓN	- 41 -
3.2 – DESCRIPCIÓN DE LA PROPUESTA	- 41 -
3.4 – DETERMINACIÓN DE RECURSOS	- 42 -
3.4.1. Recursos Humanos	- 42 -
3.4.2. Recursos Tecnológicos	- 43 -
3.4.3. Recursos Económicos	- 43 -
3.5. ETAPAS DE ACCIÓN PARA EL DESARROLLO DE LA PROPUESTA ..	- 44 -
3.1.1. Selección del entorno de desarrollo del sistema computacional.	- 44 -
3.1.2. Programación y desarrollo del sistema computacional	- 45 -
3.1.3. Metodología de programación	- 45 -
3.1.3.1. Selección de la Metodología de programación	- 47 -
3.2. Organización de los requisitos	- 47 -
3.3. Construcción del sistema automático computacional	- 48 -
3.5. ETAPAS DE ACCIÓN PARA EL DESARROLLO DE LA PROPUESTA ..	- 49 -
3.5.1. Análisis de requerimientos	- 49 -
3.5.2. Identificación de componentes y herramientas a utilizar	- 49 -
3.5.2.1. Selección de materiales para la estructura	- 50 -
3.5.2.2. Selección de materiales electrónicos	- 50 -
3.5.2.3. Selección del entorno de desarrollo	- 51 -
3.5.2.4. Selección del hardware Arduino	- 51 -
3.5.2.5 Selección del software de desarrollo del sistema computacional	- 52 -
3.3.1. Diseño y características del dispositivo	- 52 -
3.3.1.1. Construcción de la estructura del proyecto de automatización	- 52 -
3.3.1.2. Construcción de la parte electrónica del proyecto de automatización	- 53 -
3.3.2. Fase de análisis sobre el funcionamiento del Proyecto	- 53 -
3.3.2.1. Casos de uso	- 53 -
3.5.4. Implementación	- 55 -
3.5.4.1. Capacitación a los administradores del laboratorio de informática	- 55 -
3.5.4.2. Manual de usuario	- 56 -
3.5.5. Pruebas	- 56 -
3.5.5.1. Pruebas funcionales	- 56 -
3.5.5.1. Pruebas de desempeño	- 56 -
CAPÍTULO IV:	- 58 -
VALIDACIÓN DE RESULTADOS	- 58 -
4.1 – INTRODUCCIÓN	- 59 -



4.2 – SEGUIMIENTO Y MONITOREO DE RESULTADOS	- 59 -
4.2.1 Automatización de puertas en los laboratorios de informática - 59	
-	
4.2.2 Funciones del proyecto de Automatización de puertas en los laboratorios de informática	- 59 -
CONCLUSIONES	- 61 -
RECOMENDACIONES	- 62 -
BIBLIOGRAFÍA	- 63 -
ANEXOS	- 65 -
GLOSARIO	- 91 -



INDICE DE TABLAS

TABLA 1: TARJETA ARDUINO.	PÁG. 15
TABLA 2. ESPECIFICACIONES ARDUINO UNO R3.....	PÁG. 17
TABLA 3. PLAN DE RECOLECCIÓN DE DATOS.	PÁG. 26
TABLA 4. POBLACIÓN DE ESTUDIANTES.....	PÁG. 27
TABLA 5. RESULTADOS ENCUESTA PREGUNTA 1.....	PÁG. 28
TABLA 6. RESULTADOS ENCUESTA PREGUNTA 2.....	PÁG. 29
TABLA 7. RESULTADOS ENCUESTA PREGUNTA 3.....	PÁG. 30
TABLA 8. RESULTADOS ENCUESTA PREGUNTA 4.....	PÁG. 31
TABLA 9. RESULTADOS ENCUESTA PREGUNTA 5.....	PÁG. 32
TABLA 10. RESULTADOS ENCUESTA PREGUNTA 6.....	PÁG. 33
TABLA 11. RESULTADOS ENCUESTA PREGUNTA 7.....	PÁG. 34
TABLA 12. RESULTADOS ENCUESTA PREGUNTA 8.....	PÁG. 35
TABLA 13. RESULTADOS ENCUESTA PREGUNTA 9.....	PÁG. 36
TABLA 14. RESULTADOS ENCUESTA PREGUNTA 10.....	PÁG. 37
TABLA 15. RECURSOS HUMANOS.....	PÁG. 42
TABLA 16. RECURSOS TECNOLÓGICOS.....	PÁG. 43
TABLA17. RECURSOS ECONÓMICOS.....	PÁG. 44
TABLA 18: SELECCIÓN DE HARDWARE ARDUINO.....	PÁG. 51
TABLA 19: PRUEBAS DE DESEMPEÑO.....	PÁG. 57



INDICE DE GRAFICOS

ILUSTRACIÓN 1: RESULTADOS ENCUESTA PREGUNTA 1.....	PAG 28
ILUSTRACIÓN 2: RESULTADOS ENCUESTA PREGUNTA 2.....	PAG 29
ILUSTRACIÓN 3: RESULTADOS ENCUESTA PREGUNTA 3.....	PAG 30
ILUSTRACIÓN 4: RESULTADOS ENCUESTA PREGUNTA 4.....	PAG 31
ILUSTRACIÓN 5: RESULTADOS ENCUESTA PREGUNTA 5.....	PAG 32
ILUSTRACIÓN 6: RESULTADOS ENCUESTA PREGUNTA 6.....	PAG 33
ILUSTRACIÓN 7: RESULTADOS ENCUESTA PREGUNTA 7.....	PAG 34
ILUSTRACIÓN 8: RESULTADOS ENCUESTA PREGUNTA 8.....	PAG 35
ILUSTRACIÓN 9: RESULTADOS ENCUESTA PREGUNTA 9.....	PAG 36
ILUSTRACIÓN 10: RESULTADOS ENCUESTA PREGUNTA 10.....	PAG 37
ILUSTRACIÓN 11. CASO DE USO 1 SISTEMA PRINCIPAL.....	PAG 53
ILUSTRACIÓN 12. CASO DE USO 2 APLICACIÓN MÓVIL.....	PAG 54



RESUMEN

Los avances tecnológicos y la evolución en la elaboración de proyectos electrónicos impulsaron el desarrollo de este proyecto de titulación con el objetivo de automatizar los laboratorios de informática de la Facultad de Ciencias Informáticas y ser la base para futuros proyectos a corto o largo plazo.

En este proyecto se diseñará y construirá un prototipo para la automatización de los laboratorios de informática, utilizando para el desarrollo de la tarjeta principal la tarjeta electrónica ARDUINO la cual es una plataforma Open Hardware, la cual consta de varios modelos entre los cuales se escogió ARDUINO UNO R3 esta versión cuenta con requisitos necesarios para la construcción de este prototipo proporcionando una comunicación serial TTL, que permite la conexión entre el dispositivo y un computador y otras conexiones para los diferentes componentes en este caso un módulo bluetooth para poder controlarlo mediante un dispositivo móvil.

El prototipo de automatización constará de un Sistema Principal de control el cual estará instalado en un computador y será el encargado de gestionar la comunicación entre ambos a través de la conexión serial TTL, el dispositivo móvil tendrá acceso mediante una aplicación desarrollada para controlar la aperturas y clausuras de forma remota de los laboratorios mediante el bluetooth.



INTRODUCCIÓN

El Presente Proyecto integrador tiene como objetivo diseñar y construir un Sistema para la automatización de puertas en los laboratorios de informática para la apertura remota mediante dispositivos móviles y computacionales en la Facultad de Ciencias Informática de la Universidad Laica "Eloy Alfaro" de Manabí el cual será instalado en la Facultad de Ciencias Informáticas, dando solución a las necesidades de los administradores de los laboratorios.

Este proyecto de automatización es de gran utilidad a los estudiantes y personal docente de la Facultad de Ciencias Informática principalmente por el desarrollo manejo y programación de micro-controladores, electrónica de potencia, módulos para tarjetas electrónicas, tanto en conocimientos teóricos como prácticos, para lo cual, primeramente se seleccionan los componentes necesarios después de una búsqueda exhaustiva, dando como resultado que se brinde una mayor seguridad a los estudiantes, docentes y personal administrativo, los costos y valores que se obtendrían o utilizarían también tienen que cooperar con la facultad, para que el presente proyecto sea de simple manejo y composición, capacitado a los docentes y personal administrativo en dicha área, logrando como uno de los puntos principales traer la automatización a la facultad, para que los demás proyectos, o el mismo, puedan ser mejorados y recreados a gran escala, para así llegar a tener una facultad automática, trayendo consigo, un solo proceso de análisis y tareas programadas, brindando como resultado una domótica total en el lugar.

A los controladores, microcomponentes y demás proyectos a realizar se realizará un proceso de análisis para que el proyecto integrador sea, controlable, amoldable y extensible a nuevos cambios y actualizaciones, brindando un mejor soporte y comportamiento.



UBICACIÓN Y CONTEXTUALIZACIÓN DE LA PROBLEMÁTICA

El siguiente proyecto se realizará en la Facultad de Ciencias Informáticas (FACCI) perteneciente a la Universidad Laica “Eloy Alfaro” de Manabí, la cual se encuentra ubicada en la Avenida Circunvalación, en la provincia de Manabí ciudad de Manta, parroquia Manta, la carrera permite obtener un título profesional de **INGENIERÍA EN SISTEMAS** y especializarse en las diferentes ofertas que ofrece tales como desarrollo móvil, desarrollo en los diferentes lenguajes de programación, Infraestructuras y redes de información, electrónica y sistemas digitales.

Este proyecto se realizó en los laboratorios de Informáticas de la Facultad de Ciencias Informáticas, en estos laboratorios se brinda un espacio para las diferentes materias que se imparten. Gracias a la implementación de este proyecto se obtendrá acceso con mayor rapidez en el momento de ingresar a los mismo aumentando el tiempo de catedra y disminuyendo el tiempo de espera para su ingreso.



PROBLEMATIZACIÓN

Planteamiento del problema

En la actualidad, la seguridad física en muchos lugares es crucial, dependiendo del tipo de acto que se realice, sea en el lugar que se encuentre, siempre va a estar desprotegido todo aquello que se desea conservar, inclusive puede que sin darse cuenta ya se ha perdido casi todo. Ahora, al momento de ir a un lugar o al momento de querer acceder a un sitio, puede que por muy seguro que esté, uno mismo puede quedar fuera de esto.

Llegará el momento en el que se esté muy ocupado o solo se necesite tener acceso a un lugar y no se pueda, ya que el lugar esté cerrado o no se tiene la llave para acceder al sitio, o que el sitio esté muy lejos y se esté muy ocupado como para llegar.



OBJETIVOS

Objetivo General

“Automatizar las puertas de los laboratorios de informática de la Facultad de Ciencias Informáticas (FACCI), basado en cerraduras eléctricas y sistemas informáticos para mejorar los procesos de gestión estudiantil, de seguridad y laboral de la Facultad de Ciencias Informáticas de la Universidad Laica Eloy Alfaro de Manabí”

Objetivos Específicos

- Investigar los tipos y las diferentes estructuras de las cerraduras y herramientas eléctricas que puedan implementarse en los laboratorios para un gestiónamiento más factible.
- Desarrollar un sistema que permita manipular sin complicaciones la apertura y clausura de las cerraduras eléctricas.
- Implementar herramientas de automatización tanto en el Hardware como software.
- Instalar, configurar los equipos y accesorios que se eligieron en la implementación de los laboratorios.



JUSTIFICACION

Cada día se aproxima al futuro, nuevas tecnologías aparecen y otras desaparecen, lo que antes era regular o manual se está volviendo automático, más sencillo y más factible, en el caso de la facultad, se realizará la automatización de las puertas mediante dispositivos eléctricos y electrónicos, lo cual va más allá de un simple botón.

En ciertos momentos en los cuales el profesor tiene su plan de trabajo para el día establecido, o para un día en concreto tomar una evaluación, pero lo que no se esperarían es que ese día en especial los laboratorios estén ocupados o en mantenimiento por los usuarios.

Por esto, al obtener el sistema de automatización de puertas en los laboratorios, se estaría pensando en tres cosas, el primero es que ayudaría a gestionar a los estudiantes al momento del intercambio de clases, el segundo punto es que los equipos estarían más seguros, ya que el único modo para ingresar sería por la puerta al girar la perilla, pero en este caso, se estaría ,as seguro porque se controlaría desde una pc, se mostraría un mensaje si estarían ocupados los equipos, los laboratorios o simplemente no estarían disponibles, todo esto se podría ver, observar o controlar desde el pc o del dispositivo móvil, mejorando así el control sobre estos.

CAPÍTULO I:
MARCO TEÓRICO



1.1.- INTRODUCCIÓN

En este capítulo se conocerá los términos y conceptos respectivos a utilizar en el proyecto de titulación, así mismo, los componentes y como estará constituido cada opción y apartado sobre el tema, por lo cual, esto ayudará a entender cómo funciona cada una de las partes del proyecto a realizar, logrando comprender de la mejor manera que es lo que se quiere implementar, como funciona y a donde se quiere llegar.

1.2.- ANTECEDENTES DE INVESTIGACIÓN RELACIONADA AL TEMA

El proyecto integrador al cual estamos apuntando tiene muchas posibilidades de uso y creación, dicho estas posibilidades, se tendrían que crear muestras para poder empezar con una base sólida, y como lo dice la señorita Zambrano Peñarrieta Ángela en su tesis "Creación de un laboratorio electrónico"

"La posibilidad de crear un dispositivo con circuitos eléctricos y electrónicos conlleva a un gran sinnúmero de posibilidades, por lo que se pueden crear muchas formas de realizar un prototipo o un trabajo en sí, para esto se necesita estudiar a fondo los procesos de su creación y configuración manual haciendo referencia a un centro de estudio electrónico." (DOLORES, 2013).

Entonces, lo primero que necesitaríamos es un prototipo para las bases del proyecto, posiblemente con un método de prototipo o preparación por etapas para estudiar a detalle las composiciones que tendrá el proyecto, ya que en nuestro entorno, todo está en constante cambio, y es como lo dicen los señores Efrain e Ignacio en su tesis "Software de seguridad y control para el ingreso y salida vehicular en el Ala de Combate 23 de la ciudad de Manta utilizando tecnologías Identificación de radio frecuencia"



“En la actualidad, nos encontramos inmersos en un mundo en el cual las tecnologías son de ámbitos cambiantes en un corto periodo de tiempo, mismos que producen cambios interesantes en nuestro medio aportando mejoras continuas de control y seguridad permitiendo automatizar procesos e información con excelentes niveles de calidad y eficiencia.” (ENCALADA & PONCE, 2015)

Al realizar el estudio a estudiantes y docentes, se pueden encontrar beneficios a la hora de utilizar los laboratorios de cómputo, es por esto que debe brindarse una mayor seguridad, tal y como lo dicen las Srtas. Alexandra y María Cagua en su tesis “Software de control para el uso de los laboratorios de cómputo de la Universidad Laica "Eloy Alfaro" de Manabí”.

“Los laboratorios de cómputo ayudan a las instituciones a tener una gran área de investigación y desarrollo, aquella comprende la utilización de tecnologías de la información y la comunicación (TIC), para poder realizar trabajos y/o proyectos que sea fijado por las asignaturas o docentes. Estas tecnologías están a disposición del colectivo universitario, las cuales le permiten desenvolver sus clases de forma práctica, interactiva y novedosa.” (BRIONES CAGUA CLAUDIA & BRIONES CAGUA MARIA, 2015)

Al llegar a todos estos antecedentes se puede proceder a utilizar varias metodologías y tipos de procesos / investigaciones para el correcto funcionamiento del proyecto, teniendo en cuenta los puntos principales y seguridad en las etapas del mismo.

1.3.- DEFINICIÓN CONCEPTUAL

1.3.1.- Cerradura

1.3.1.1.- Definición

Es un mecanismo complejo de metal que se incorpora en el lugar el cual se desea proteger su contenido, sea desde una casa en el que se colocaría en



las puertas, en los armarios o en otros objetos pequeños como lo son maletas, cajones de escritorio, un pequeño cofre y otro objeto en el cual se pueda colocar.

En efecto, el mecanismo brinda seguridad para su contenido, pero este se puede activar o desactivar mediante una llave, normalmente creadas de metal o de bronce, talladas para abrir una de estas cerraduras de manera personal, logrando así que no toda la cerradura existente pueda abrir con cualquier llave, sino, que una llave específicamente pueda abrir una cerradura. (Mellado, 1857)

1.3.1.2.- Partes de una cerradura

- **La caja:** Es una caja creada de metal o acero que contiene los elementos de una cerradura, esta es fabricada por medio del estampado de una chapa de acero doble, dándole así una mayor fuerza y rigidez para una mejor seguridad.
- **El pestillo:** Se trata de la parte que se mantiene fuera de la cerradura, por lo general se encuentra a los lados de esta y es la que atraviesa al agujero que se encuentra aún lado de la puerta o también llamado muelle, esta es corrediza ya que al momento de cerrar o abrir el cerrojo tiene que moverse de un lado a otro
- **El muelle:** Esta parte de la cerradura es indispensable para mantener la seguridad, por lo que se trata de la parte que retiene al pestillo al momento de colocarse he impedir que este se mueva de un lado a otro sin la acción o el movimiento de la llave, ya que el muelle cuenta con un fijador que se coloca en el pestillo y este solo se levanta al girar la llave.
- **Guardas:** Éstas sirven para que al momento de insertar una llave en la cerradura y moverla no permita la acción de abrirse si la llave ingresada



no es la que está creada para esa cerradura, ya que cada cerradura tiene diferentes guardas y cada llave tiene que tener las mismas especificaciones de estas guardas.

- **Resortes:** Estos cumplen la función de mantener fijas algunas piezas de la cerradura, como lo son el pestillo, la guarda y la palanca de vinculación y así poder mantenerlas en una posición que puedan moverse y trabajar.
- **Nuez:** Esta pieza de la cerradura sirve para ejercer una pequeña fuerza en el pestillo de empuje por medio de una manija. Tiene un agujero en la parte del centro en la que pasa el vástago de la manija para así poder girarla y poder ejercer la fuerza donde el pestillo de la cerradura se mueve y permite que se pueda mover y abrir la cerradura.

1.3.1.3.- Tipos de cerraduras

- **Cerradura mecánica:** Es la que se encuentra en muchos lados y sitios, son las primeras utilizadas desde su creación, la cual para acceder al contenido que protege se tiene que hacer uso obligatorio de la llave o el pestillo de este, por lo que no existe otro medio externo para poder abrirla.
- **Cerradura automática:** Es una mejora de las cerraduras mecánicas, ya que estas de aquí funcionan con un pequeño pulso eléctrico para poder ejercer una pequeña fuerza y así mover el pestillo, puede ser desde un botón o el dispositivo que se prefiera, funcionando así desde un lugar cercano o lejano dentro del hogar o donde se encuentre instalado, también consta de una llave para poder abrir la cerradura al momento de salir del sitio, por si se está del otro lado del botón o dispositivo que permitirá abrirla automáticamente.



1.3.2.- Teléfono inteligente

1.3.2.1.- Definición

Es un dispositivo móvil utilizado para realizar tareas personales o acciones personales para el gusto de las personas, como lo es reproducción de música, navegación web, mensajería instantánea, mensajería normal, llamadas a otras personas, con el fin de mantener una funcionalidad que acepte el cliente, por lo cual existen diferentes tipos de teléfonos para el gusto de cada persona, cada uno tiene una complejidad distinta a la otra y un funcionamiento distinto.

1.3.2.2.- Teléfono inteligente a utilizar

Cada uno de los teléfonos inteligentes tiene una funcionalidad distinta, ya que los hay desde teléfonos sencillos que permite solo las operaciones básicas, como lo son llamadas o mensajería, pero no permiten otras acciones, pero por otro lado, están los Smartphone los cuales permiten muchas funciones más, por lo que permiten ir más allá de lo que se estima, como lo es conectarse a una red y búsquedas en internet, teniendo así un sistema operativo que lo controle y pueda realizar estas funciones.

1.3.2.3.- Teléfono inteligente y su sistema operativo

Estos dispositivos cuentan con diferentes tipos de sistemas operativos para controlarlos, variando la marca o su fabricante, tiene un sistema que lo identifica, por lo que no todas las aplicaciones las ejecutaría cualquier sistema operativo del dispositivo, es lo que hay que tener en cuenta al momento de crear o desarrollar algo para estos, por lo que si se crea para una plataforma en específico, se debe primero analizar para cual será.



1.3.3- Automatización

1.3.3.1.- Definición

Se trata de convertir una tarea manual en una tarea que se realice por sí misma con tareas pre-programadas por el usuario mediante un objeto o aparato emisor listas para enviar a un aparato receptor, ejecutando así una serie de tareas automáticamente, reemplazando así la tarea que el usuario realizaría manualmente.

Estas, están encadenadas con otros procesos por los cuales tiene que pasar antes de llegar al resultado esperado, por lo que si se automatiza cualquier objeto, se tiene que crear una "ruta", pasando así por una serie de ciclos y procesos para que llegue al fin y obtener el resultado, por lo que si decimos en un ejemplo que si una persona encendía un televisor de manera manual, mediante el botón de este aparato, ahora lo hará desde un control remoto el cual un botón tiene configurada una tarea específica, enviando así una señal al aparato receptor y este ejecuta la tarea antes mencionada.

1.3.3.2.- Automatización programable

Son tareas automatizadas pero no de manera simple, ya que el usuario en esta acción podrá modificar algunas tareas que ya estaban ejecutándose en el sistema automático, logrando así que cambie el curso de a donde se quería llegar con el objeto, es decir, si automáticamente un objeto A estaba programado para que solo llegue al punto A, con una modificación externa, sea desde otro dispositivo o en el mismo objeto, al pulsarlo o ejecutarlo, el objeto A podrá ir al punto B y al volverse a presionar volvería a ir al punto A y viceversa, a esto es lo que se le llama automatización programable.



1.3.4.- Lenguajes de programación

1.3.4.1.- Definición

Según la definición teórica, como lenguaje se entiende a un sistema de comunicación que posee una determinada estructura, contenido y uso. La programación es, en el vocabulario propio de la informática, el procedimiento de escritura del código fuente de un software. De esta manera, puede decirse que la programación le indica al programa informático qué acción tiene que llevar a cabo y cuál es el modo de concretarla. (Porto, 2012)

Como lo dice el autor, el lenguaje de programación tiene una estructura la cual está comprendida por varios procesos o pasos, en los que estos ejecutan una acción cada uno, para así obtener un resultado esperado, y de estos se obtiene el programa, aunque existen muchos lenguajes de programación, es recomendable utilizar el que más se personalice al gusto.

1.3.4.2.- Lenguaje de programación Java

La principal característica de Java es la de ser un lenguaje compilado e interpretado. Todo programa en Java ha de compilarse y el código que se genera byte-codes es interpretado por una máquina virtual. De este modo se consigue la independencia de la máquina, el código compilado se ejecuta en máquinas virtuales que si son dependientes de la plataforma. (Fernández, 2009)

1.3.4.3.- Uso del lenguaje Java para el dispositivo

Este lenguaje es utilizado para crear los distintos programas que se utilizan en los dispositivos móviles con un sistema operativo específico, el cual es Android, por lo general estos dispositivos ejecutan una app la cual les permite realizar ciertas funciones, y es ahí en donde actuaría el o las aplicaciones para el proyecto.



En este lenguaje se pueden sacar o realizar dos tipos de aplicaciones, de escritorio, el cual se puede ejecutar en cualquier sistema operativo en la actualidad, es decir, es un sistema con una plataforma libre, por lo que tiene una ventaja sobre los demás sistemas operativos logrando así que cualquier desperfecto que tenga la computadora con respecto al sistema que utilice, pueda ejecutarse.

1.3.4.4.- Lenguaje Java y dispositivo móvil

El lenguaje java tiene una opción para que, al momento de crear una aplicación, sea esta en un teléfono, o en una computadora, y al crearlo en el teléfono, este permite controlar la cerradura automática con el teléfono inteligente, por lo que automáticamente se convertiría en un control remoto para la cerradura.

1.3.4.5.- Lenguaje Java y computadora personal

En referencia al punto anterior, java también permite crear una aplicación para la computadora personal, por lo que también se puede ejecutar el programa que se crea desde una computadora, permitiendo así tener otro punto para acceder al dispositivo controlado, cumpliendo con las mismas funcionalidades que el dispositivo móvil.

1.3.5.- FUNDAMENTOS BÁSICOS DE ELECTRÓNICA

1.3.5.1.- Definición

- **Corriente:** Es la circulación de cargas o electrones a través de un circuito eléctrico cerrado, que se mueven siempre del polo negativo al polo positivo de la fuente de suministro de fuerza electromotriz (FEM).



- **Voltaje:** La tensión es la diferencia de potencial de una línea con resto a la otra, también se conoce como la velocidad de electrones que circulan a través de un conductor. La unidad de medida de la tensión es el voltio (V).
- **Resistencia Eléctrica:** La resistencia eléctrica es toda oposición que encuentra la corriente a su paso por un circuito eléctrico cerrado, atenuando o frenando el libre flujo de circulación de las cargas eléctricas o electrones La unidad de medida es el OHM.
- **Electrónica analógica:** La electrónica analógica es una rama de la electrónica que estudia los sistemas cuyas variables varían de una forma continua en el tiempo.
- **Electrónica digital:** La electrónica digital es una rama de electrónica en la cual se estudia o se aplica solo dos estados de valores, magnitudes o tensiones: alto-bajo, cero-uno. En la representación digital los valores no se denotan por valores proporcionales, sino por símbolos llamados dígitos.

1.3.6.- Circuito eléctrico

1.3.6.1.- Definición

Un circuito es una red eléctrica (interconexión de dos o más componentes, tales como resistencias, inductores, condensadores, fuentes, interruptores y semiconductores) que contiene al menos una trayectoria cerrada. Los circuitos



que contienen solo fuentes, componentes lineales (resistores, condensadores, inductores) y elementos de distribución lineales (líneas de transmisión o cables) que pueden analizarse por métodos algebraicos para determinar su comportamiento en corriente directa o en corriente alterna. (www.herrajesdelsureste.com, 2017).

1.3.6.2.- Componentes del circuito electrónico

- **Conector DB9 hembra:** El conector DB9 (originalmente DE-9) es un conector analógico de 9 clavijas de la familia de conectores D-Subminiature (D-Sub o Sub-D). El conector DB9 se utiliza principalmente para conexiones en serie, ya que permite una transmisión asíncrona de datos según lo establecido en la norma RS-232 (RS-232C). (Vialfa-Carlos, 2016)
- **Resistencias:** Es un pequeño compuesto de silicio, el cual permite modificar la corriente eléctrica que pasa por ahí para que el dispositivo final no se queme o se dañe, estos varían con su capacidad de salida, siendo así desde menor a mayor corriente.
- **Condensadores:** Son componentes electrónicos que almacenan energía eléctrica para utilizarlas en un circuito en el momento adecuado. Está fabricado, básicamente, por un par de armaduras distanciadas por un material aislante denominado dieléctrico. La capacidad de un condensador radica en almacenar mayor o menor número de cargas cuando está sujeto a una tensión.
- **Transistor:** El transistor es un componente electrónico semiconductor utilizado ampliamente para generar una señal de salida en respuesta a otra



señal de entrada. Efectúa varias configuraciones como de amplificador oscilador, conmutador o rectificador. (Braun, 1982)

- **Diodo:** Es un componente electrónico que permite el paso de la corriente en un solo sentido. Los diodos más empleados en los circuitos electrónicos actuales son los diodos fabricados con material semiconductor. Entre los diodos que están ganando popularidad en el mercado comercial es el diodo tipo LED, por su bajo consumo eléctrico. (Tooley, Mike 2012)
- **Malla:** Común mente llamado análisis de mallas, es una técnica usada para determinar la tensión o la corriente de cualquier elemento de un circuito plano. La ventaja de usar esta técnica es que crea un sistema de ecuaciones para resolver el circuito, minimizando en algunos casos el proceso de utiliza para hallar una tensión o una corriente. (Nilsson, James W, 2002)

1.3.7.- RELAY

1.3.7.1.- Definición

El Relay es un interruptor operado magnéticamente. El relé se activa o desactiva (dependiendo de la conexión) cuando el electroimán (que forma parte del relé) es energizado (le ponemos un voltaje entre sus terminales para que funcione). Esta operación causa que exista conexión o no, entre dos o más terminales del dispositivo (el relé). Esta conexión se logra con la atracción o repulsión de un pequeño brazo, llamado armadura, por el electroimán. (Unicrom, 2016)

1.3.7.2.- Tipos de relés

- **Relés de tipo armadura:** Pese a ser los más antiguos siguen siendo los más utilizados en multitud de aplicaciones. Un electro imán provoca la basculación de una armadura al ser activado, cerrando o abriendo los



contactos dependiendo de si es N.A (normalmente abierto) o N.C (normalmente cerrado).

- **Relés de núcleo móvil:** A diferencia del anterior modelo estos están formados por un émbolo en lugar de una armadura. Debido a su mayor fuerza de atracción, se utiliza un solenoide para cerrar sus contactos. Es muy utilizado cuando hay que controlar altas corrientes.
- **Relé tipo reed o de lengüeta:** están constituidos por una ampolla de vidrio, con contactos en su interior, montados sobre delgadas láminas de metal. Estos contactos conmutan por la activación de una bobina, que se encuentra alrededor de la mencionada ampolla.
- **Relés polarizados o bi estables:** se componen de una pequeña armadura, solidaria a un imán permanente. El extremo inferior gira dentro de los polos de un electro imán, mientras que el otro lleva una cabeza de contacto. Al excitar el electro imán, se mueve la armadura y provoca el cierre de los contactos. Si se polariza al revés, el giro será en sentido contrario, abriendo los contactos o cerrando otro circuito.
- **Relés multitensión:** son la última generación de relés que permiten por medio de un avance tecnológico en el sistema electromagnético del relé desarrollado y patentado por Relaygo, a un relé funcionar en cualquier tensión y frecuencia desde 0 a 300 AC/DC reduciendo a un solo modelo las distintas tensiones y voltajes que se fabricaban hasta ahora.

1.3.7.3.- Modulo rele

Este módulo rele cuenta con varios modelos dependiendo de sus variantes existen de 2, 4, 16, etc. su única diferencia entre ellos es el número de pines de datos de entrada que incorporan de tipo en su estructura base consta



de 2 entradas que suelen estar situada a cada extremo de la fila de sus pines de entrada el primero es GND (Tierra) y VCC (Corriente), hay varios modelos donde difieren los voltajes de entrada el voltaje recomendable debe ser de 5V si se usa a través de las salidas de voltaje de una placa Arduino si se usa una fuente externa de alimentación esta tendría que ser de 12V.(arubia45, 2013)

1.3.8.- ELECTRONICA DE CONTROL

1.3.8.1.- Definición

Los sistemas electrónicos de control son aquellos dedicados a obtener una salida requerida de un sistema o proceso determinado. En un sistema general se disponen de una serie de entradas que se obtienen del sistema a controlar, y se realiza el diseño un sistema para que, a partir de estas entradas, se modifique un establecido parámetro en el sistema, con lo que las señales anteriores volverán a su estado normal ante cualquier situación.

1.3.9.- PLATAFORMA ELECTRÓNICA ARDUINO

1.3.9.1.- Definición

Arduino es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar.

Está pensado para artistas, diseñadores, como hobby y para cualquiera 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 el control de luces, motores y otros artefactos. (arduino.cl, 2016)

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea



controlando luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en wiring) y el entorno de desarrollo Arduino (basado en processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software. Las placas pueden ser hechas a mano o compradas montadas de fábrica; el software puede ser descargado de forma gratuita. Los ficheros de diseño de referencia están disponibles bajo una licencia abierta, así pues, es libre de adaptarlos según las necesidades. (Artero, 2013, pág. 61-62)

1.3.9.2.- Modelos de Arduino

Desde su creación en el año 2005 hasta la actualidad la plataforma ARDUINO ha evolucionado implementando diferentes modelos, estos modelos van distribuidos desde lo más esencial hasta lo más complejo es por esta razón que ARDUINO se ajusta al nivel del Usuario, a continuación, se muestran las tarjetas electrónicas ARDUINO en la siguiente tabla:

Tabla 1: Tarjeta ARDUINO.

Modelo	Micro controlador	Voltaje del Sistema	Digital I/O	Entradas Análogas	Memoria Flash	Interfaz de programación
Arduino Due	AT91SAM3X8 E	3.3V	54	12	512kb	Nativa USB
ARDUINO LEONARDO	ATmega32U4	5V	20	12	32kb	Nativa USB
ARDUINO UNO - R3	ATmega328	5V	14	6	32kb	USB vía ATmega16U2
REDBOARD	ATmega328	5V	14	6	32kb	USB vía FTDI
ARDUINO PRO 3.3V/8MHZ	ATmega328	3.3V	14	6	32kb	Cabecera compatible con FTDI



ARDUINO PRO 5V/16MHZ	ATmega328	5V	14	6	32kb	Cabecera compatible con FTDI
MEGA PRO 3.3V	ATmega2560	3.3V	54	16	256kb	Cabecera compatible con FTDI
MEGA PRO 5V	ATmega2560	5V	54	16	256kb	Cabecera compatible con FTDI
ARDUINO MINI 05	ATmega328	5v	14	6	32kb	Cabecera Serial
ARDUINO PRO MINI 3.3V/8MH	ATmega328	3.3V	14	6	32kb	Cabecera compatible con FTDI
ARDUINO PRO MINI 5V/16MHZ	ATmega328	5v	14	6	32kb	Cabecera compatible con FTDI
ARDUINO FIO	ATmega328P	3.3V	14	8	32kb	Cabecera compatible con FTDI o Inalámbrica via Xbee
MEGA PRO MINI 3.3V	ATmega2560	3.3V	54	16	256kb	Cabecera compatible con FTDI
PRO MICRO 5V/16MHZ	ATmega32U4	5v	12	4	32kb	Nativa USB
PRO MICRO 3.3V/8MH Z	ATmega32U4	3.3V	12	4	32kb	Nativa USB
LILYPAD ARDUINO 328 MAIN BOARD	ATmega328	3.3V	14	6	32kb	Cabecera compatible con FTDI
LILYPAD ARDUINO SIMPLE BOARD	ATmega328	3.3V	9	4	32kb	Cabecera compatible con FTDI

Fuente: ARDUINO Página oficial.



1.3.9.3.- Arduino uno

ARDUINO UNO R3 es un modelo de placa ARDUINO para empezar con la electrónica y la codificación. Está basado en el microcontrolador Tmega328P que incorpora nuevas Funciones. Cuenta con 14 pines digitales de entrada y salida I/O (de las cuales 6 podrán utilizar como salidas PWM), 6 entradas analógicas, una memoria flash de 32kb, un cristal de cuarzo de 16 MZ. Una alimentación de 5 V. (Artero, 2013, pág. 71)

1.3.9.4.- Características Arduino Uno R3

Tabla 2. Especificaciones Arduino uno R3

ESPECIFICACIONES ARDUINO UNO R3	
Microcontrolador	ATmega328
Voltaje de funcionamiento 5V Voltaje de entrada	5V
Voltaje de entrada (limite)	7-12V
Pines E/S digitales	14 (de las cuales 6 podrán utilizar como salidas PWM)
Pines de entrada analógica	6 entradas analógicas
Memoria Flash	32k de memoria Flash
EEPROM	1024 bytes
Velocidad de reloj	16MHz

Fuente: Arduino Página Oficial.

1.3.9.5.- Ide Arduino

Arduino contiene un entorno de desarrollo Integrado (IDE) que permite a través de un conjunto de herramientas desarrollar programas de una manera sencilla el entorno de desarrollo está basado en los entornos de Processing y Wiring que son de código abierto.

ARDUINO contiene un entorno interactivo de desarrollo que permite programar sencillamente el microcontrolador de la tarjeta. Este lenguaje de



programación se basa en C/C++ y se reduce con el uso de las bibliotecas que tiene Arduino. (Artero, 2013, pág.145).

1.4.- CONCLUSIONES

Se pueden realizar los procesos de automatización planteados en el proyecto tratando con los materiales antes descritos.

De los diferentes tipos e infraestructuras de las cerraduras existentes, se elige una en la cual tenga una mayor probabilidad de salida al momento crítico.

Se desarrolla el sistema basado en el programa escogido y se estructurará e implementó el sistema automático manualmente con las partes.

Se implementa el sistema en la facultad, establecido previamente en un lugar seguro, para un cuidado diario.

Se configura el sistema, equipos y accesorios para que tenga una conexión estable entre los equipos.

CAPÍTULO II:
MARCO METODOLÓGICO



2.1. – INTRODUCCIÓN

En esta parte de nuestro trabajo se permitirá conocer el marco investigativo, o en otros conceptos, el marco metodológico, el cual previa realización del trabajo se tiene que tomar en cuenta que técnicas de investigación se utilizará para continuar con la obtención de información.

Para realizar nuestra investigación se utilizarán fuentes de información primaria como un paso principal, aplicando técnicas de observación, y de entrevistas a las personas involucradas para continuar con nuestro trabajo exploratorio en la obtención de información.

2.2. TIPOS DE INVESTIGACIÓN

El tipo de investigación que se realizó para el trabajo fue:

Exploratorio: Se utilizó este tipo de investigación por lo que se realizó un estudio exploratorio del área que se verá afectada, por lo que se necesitaba encontrar la problemática del congestionamiento del sitio y a su vez, como se podría mejorar para solucionar el problema, investigando cada una de las posibles opciones y sus consecuencias.

Analítico: Se utilizó este tipo de investigación por lo que nuestro proyecto estaba en la etapa de estudio y al tener varios procesos de creación, se necesitaba estudiar qué impacto tendrá el proyecto de automatización en los estudiantes y en el área afectada.



2.3. MÉTODOS DE INVESTIGACIÓN

Para poder encontrar la problemática del sitio se tuvo que utilizar un método que permitiera ayudarnos a empezar a construir los hechos del problema del área para poder tener una idea mucho más clara y saber el cómo se puede mejorar los resultados obtenidos, para esto se usa el siguiente método.

Método inductivo - deductivo: Se tuvo que utilizar este método para observar el aglomera miento del sitio, y del como algunas veces no está disponible, para su mejor uso y seguridad, en sí de la automatización de la puerta de los laboratorios de computación, que ocasiona el no tener más seguridad y el cómo podría mejorar en gran manera su actualización o automatización.

Analítico – Sintético: Se determinará el proceso de análisis y la sintaxis para establecer el orden y la manera en que se llevan cada uno de los procesos utilizados en la obtención de los datos para su luego procesamiento.

2.4. HERRAMIENTAS DE RECOLECCIÓN DE DATOS

2.4.1 ENCUESTA - ENTREVISTA - OBSERVACIÓN / OTRAS

La herramienta de recolección de datos que se utilizó para esta labor fueron las siguientes:

- **Observación:** En esta parte nos situamos en un punto estratégico del lugar en el que se podían observar los 3 laboratorios disponibles y en el que poco a poco se iban agrupando las personas a tal punto de no saber por dónde estar. Además de los laboratorios, se podía observar que las cámaras a veces no estaban funcionando por lo que le brindaba poca seguridad al sitio y en otras ocasiones se observaba que



los laboratorios estaban ocupados o apartados para eventos sin que los profesores lo supieran, impidiendo cumplir con su labor el cual era de dar ciertos tipos de clases con la utilización de los laboratorios, cambiando así su método de enseñanza a otra forma.

- **Entrevista:** Podría decirse que esta herramienta también es esencial, esta herramienta la utilizamos para acercarnos a los estudiantes que utilizan estos laboratorios para preguntarles de cómo se sienten al utilizarlos, de cuáles son los inconvenientes que tienen al momento de ingresar al mismo y también se preguntó al administrador del laboratorio sobre el funcionamiento de este, sus medidas de seguridad y que ocasiona el no tener información abierta a los estudiantes o docentes.

2.5. FUENTES DE INFORMACIÓN DE DATOS

2.5.1. FUENTES PRIMARIAS – FUENTES SECUNDARIAS

Una de las fuentes primarias para la información de datos fue la utilización personal del laboratorio, poniendo así al usuario en el ambiente el cual se efectuaba la problemática, otra de las formas fue la observación de las cámaras de seguridad del sitio, pudiendo observar que no estaban habilitadas.

Una de las fuentes secundaria fue la pregunta a los estudiantes de cómo se sentían al momento de querer ingresar al aula, por lo que a ellos mismos, siendo usuarios de los laboratorios tenían más experiencia en cuanto a la utilización del mismo, preguntando entonces al administrador de los laboratorios el funcionamiento del mismo y a las personas que utilizaban para poder realizar una observación mejorada del mismo.



Otra de las fuentes fueron la información encontrada en los textos y bibliografías de las tesis antes realizadas por distintos estudiantes, por lo que se pudo comprender el funcionamiento de estos dispositivos y herramientas, así como la utilización de la herramienta de internet para investigar el funcionamiento similar de nuestro proyecto de tesis.

2.6 INSTRUMENTAL OPERACIONAL

2.6.1 ESTRUCTURA Y CARACTERISTICAS DE LOS INSTRUMENTOS DE RECOLECCIÓN DE DATOS

Observación: Para realizar la investigación del problema en sí, se tuvo que definir primordialmente qué es lo que se iba a investigar, y después de realizar la encuesta se pudo obtener un mejor margen de lo que origina el problema y así poder investigarlo de manera más profunda, tomando así el tiempo necesario para reconocer qué es lo que origina el problema y como poder resolverlo, documentando todos los datos obtenidos en un cuaderno y grabaciones, para poder luego recolectar esa información, ordenarla y clasificarla para un mejor uso.

Entrevista: Siendo uno de los mejores métodos también para la obtención de datos, con esta herramienta pudimos de manera verbal, obtener datos de cómo se origina el problema de la aglomeración en los pasillos, de cómo podría mejorar la automatización el problema y el cómo o que mejoras se le puede hacer al laboratorio para que este tenga mayor seguridad y cómo debería de ser.



2.7. ESTRATEGIA OPERACIONAL PARA LA RECOLECCIÓN Y TABULACIÓN DE DATOS

2.7.1. PLAN DE RECOLECCIÓN, TABULACIÓN, ANÁLISIS E INTERPRETACIÓN DE DATOS

Con el plan de recolección de datos se pudo empezar con la investigación del mismo, por lo que se buscaron las mejores maneras o las mejores herramientas para realizar la concentración de los resultados obtenidos a través de la herramienta de observación y de investigación del problema, en la encuesta se utilizaron preguntas de selección de opciones para facilitar la tabulación y análisis de los resultados.

Los datos obtenidos y guardados o anotados fueron adjuntados o se adjuntan, lo que son anotaciones o grabaciones se recopilan y se seleccionan las respuestas que tengan mayor concordancia o relevancia para obtener un punto específico de la automatización de la puerta de los laboratorios, encontrando o tratando de encontrar una solución a la misma.

Una vez obtenidos los datos principales de la tabulación se comparan con la problemática de la automatización de la puerta de los laboratorios, dando así un resultado de cómo mejorar esa área de la facultad y el cómo brindar una mayor seguridad a la misma, sin que el rendimiento académico se vea afectado para el conteo de datos, se utilizó la herramienta de Microsoft Excel, y se procedió a la tabulación para mostrar los resultados en cuadros y gráficos estadísticos que facilitaran el análisis e interpretación. Las tareas establecidas en proceso se detallan a continuación.



Tabla 3. Plan de recolección de datos

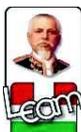
Actividad	Descripción
Contacto Inicial	Contacto e investigación preliminar con los involucrados.
Formato para aplicación de encuestas	Reunión con los administradores de los Laboratorios de informática, y el Tutor del proyecto integrador para compartir los objetivos de la investigación y solicitar autorización para aplicar la encuestas y entrevista.
Aplicación encuestas, Observación	Conversación con los estudiantes para dar a conocer los motivos de investigación y solicitar su colaboración.
Preparar instrumentos	Estructura y revisión de los instrumentos para aplicar las encuestas y entrevista.
Aplicar instrumentos	Inicio de la aplicación de los instrumentos a los estudiantes de los niveles 4, 6, 8 ubicados en los laboratorios de informática tiempo de aplicación de la encuesta y entrevista 2 días.
Procesar información	Se tabulo la información determinado datos estadísticos y gráficos para su posterior análisis.
Elaborar informes	Publicación de resultados y establecer medidas.

Autores: Reyes Wellington – Quiroz Jonathan

2.8. MUESTREO

2.8.1. SEGMENTACIÓN Y TAMAÑO DE LA MUESTRA

El total de la muestra que se escogió para esta investigación y entrevista tiene un total de 30 personas, por lo que fueron 10 personas por laboratorio, se utilizaron 3 laboratorios de informática, estas personas son estudiantes o



profesores que requerían la utilización de dichos laboratorios para recibir sus clases y sirvieron para recolectar la información necesaria, los datos y aplicar las herramientas antes mencionadas.

Tabla 4. Población de Estudiantes

Estudiantes	Docentes	Semestre	Número Total
9	1	4	10
9	1	6	10
9	1	8	10
		Población	30

Autores: Reyes Wellington – Quiroz Jonathan

2.9. PRESENTACIÓN Y ANÁLISIS DE LOS RESULTADOS

Se realizó el respectivo análisis de los datos obtenidos mediante las técnicas de recolección de datos y estos resultados serán desplegados de manera que se pueda comprender y demostrar del porqué de su contenido, sin exceptuar ningún proceso previo a estos datos.

2.9.1. PRESENTACIÓN Y DESCRIPCIÓN DE LOS RESULTADOS OBTENIDOS

Se realizó el análisis respectivo de los datos obtenidos mediante las técnicas de recolección de los datos y los resultados obtenidos serán desplegados mediante sus respectivos gráficos estadísticos.



ENCUESTA REALIZADA A LOS ESTUDIANTES DE LA FACCI QUE RECIBEN CATEDRA EN LOS LABORATORIOS DE INFORMÁTICA

I. INFORMACIÓN RESPECTO A LOS LABORATORIOS DE LA FACULTAD DE CIENCIAS INFORMÁTICA.

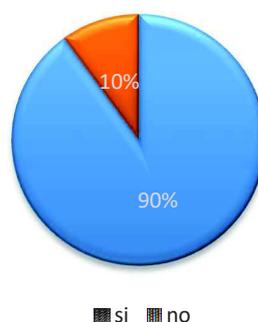
1. ¿Considera usted que los laboratorios de informática deberían ser automatizados?

Tabla 5. Resultados Encuesta Pregunta 1.

Alternativas	F	%
SI	27	90%
NO	3	10%
TOTAL	30	100%

Fuente: Estudiantes

Ilustración 2. Resultados Encuesta Pregunta 1.



Autores: Reyes Wellington – Quiroz Jonathan

Análisis: Se puede observar en la gráfica que el 90% de los estudiantes consideran que se debería automatizar los laboratorios de informática, mientras que el 10% restante no están de acuerdo con la automatización en los laboratorios de informática, en relación al resultado se evidencia que un alto porcentaje del estudiante apoya el proyecto de automatización.



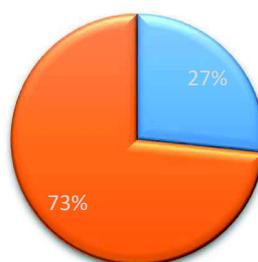
2. ¿Esta conforme con el tiempo de apertura de los laboratorios de informática al momento de recibir clases?

Tabla 6. Resultados Encuesta Pregunta 2.

Alternativas	F	%
SI	8	27%
NO	22	73%
TOTAL	30	100%

Fuente: Estudiantes

Ilustración 3. Resultados Encuesta Pregunta 2.



■ si ■ no

Autores: Reyes Wellington – Quiroz Jonathan

Análisis: Se puede observar en la gráfica que el 73% de los estudiantes no está de acuerdo con el tiempo que se tardan en abrir los laboratorios de informática, mientras que el 27% restante están conformes con el tiempo de apertura de los laboratorios de informática.



II. INFORMACIÓN SOBRE TARJETAS ARDUINO.

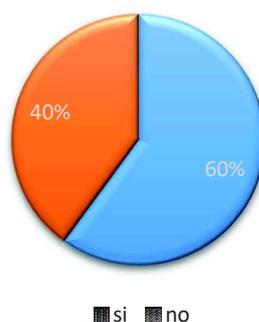
3. ¿Conoce usted sobre las tarjetas ARDUINO?

Tabla 7. Resultados Encuesta Pregunta 3.

Alternativas	F	%
SI	18	60%
NO	12	40%
TOTAL	30	100%

Fuente: Estudiantes

Ilustración 4. Resultados Encuesta Pregunta 3.



Autores: Reyes Wellington – Quiroz Jonathan

Análisis: Se puede observar en la gráfica que el 60% de los estudiantes tiene conocimiento sobre Arduino, mientras que el 40% restante no cuentan con conocimientos sobre dichos temas, en relación al resultado se evidencia que un alto porcentaje del estudiante conoce sobre Arduino.



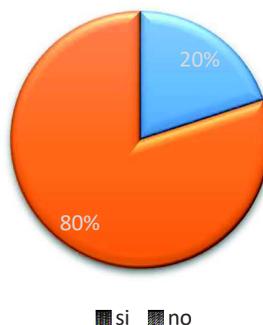
4. ¿Tiene conocimiento sobre Open Source?

Tabla 8. Resultados Encuesta Pregunta 4.

Alternativas	F	%
SI	6	20%
NO	24	80%
TOTAL	30	100%

Fuente: Estudiantes

Ilustración 5. Resultados Encuesta Pregunta 4.



Autores: Reyes Wellington – Quiroz Jonathan

Análisis: Se puede observar en la gráfica que el 80% de los estudiantes no cuenta con conocimiento sobre open source, mientras que el 20% restante cuentan con conocimientos sobre dichos temas, en relación al resultado se evidencia que un alto porcentaje del estudiante no conoce sobre open source.



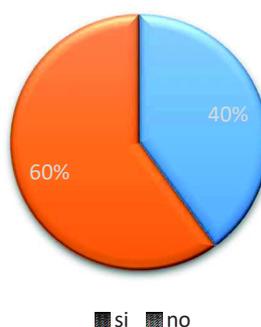
5. ¿Tiene conocimiento sobre Hardware libre?

Tabla 9. Resultados Encuesta Pregunta 5.

Alternativas	F	%
SI	12	40%
NO	18	60%
TOTAL	30	100%

Fuente: Estudiantes

Ilustración 6. Resultados Encuesta Pregunta 5.



Autores: Reyes Wellington – Quiroz Jonathan

Análisis: Se puede observar en la gráfica que el 60% de los estudiantes no cuenta con conocimiento sobre software libre, mientras que el 40% restante cuentan con conocimientos sobre dichos temas, en relación al resultado se evidencia que un alto porcentaje del estudiante no conoce sobre Hardware libre.



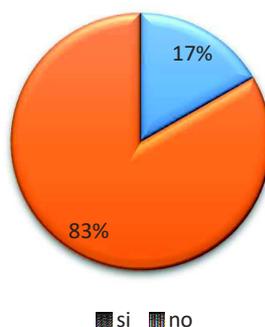
6. ¿Conoce el lenguaje de programación que utiliza Arduino?

Tabla 10. Resultados Encuesta Pregunta 6.

Alternativas	F	%
SI	12	17%
NO	18	83%
TOTAL	30	100%

Fuente: Estudiantes

Ilustración 7. Resultados Encuesta Pregunta 6.



Autores: Reyes Wellington – Quiroz Jonathan

Análisis: Se puede observar en la gráfica que el 83% de los estudiantes no cuenta con conocimiento sobre el lenguaje de programación que utiliza Arduino mientras que el 17% restante cuentan con conocimientos sobre dichos temas, en relación al resultado se evidencia que un alto porcentaje del estudiante no conoce sobre el lenguaje de programación de Arduino.



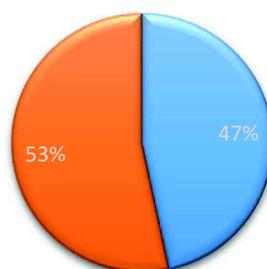
7. ¿Tiene conocimiento sobre Software libre?

Tabla 11. Resultados Encuesta Pregunta 7.

Alternativas	F	%
SI	14	47%
NO	16	53%
TOTAL	30	100%

Fuente: Estudiantes

Ilustración 8. Resultados Encuesta Pregunta 7.



■ si ■ no

Autores: Reyes Wellington – Quiroz Jonathan

Análisis: Se puede observar en la gráfica que el 53% de los estudiantes no cuenta con conocimiento sobre software libre, mientras que el 47% restante cuentan con conocimientos sobre dichos temas, en relación al resultado se evidencia que un alto porcentaje del estudiante no conoce sobre software libre.



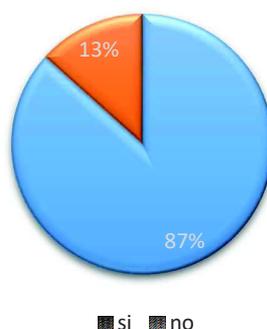
8. ¿Conoce usted el termino automatización?

Tabla 12. Resultados Encuesta Pregunta 8.

Alternativas	F	%
SI	36	87%
NO	4	13%
TOTAL	30	100%

Fuente: Estudiantes

Ilustración 9. Resultados Encuesta Pregunta 8.



Autores: Reyes Wellington – Quiroz Jonathan

Análisis: Se puede observar en la gráfica que el 87% de los estudiantes cuenta con conocimiento sobre el término de automatización, mientras que el 13% restante cuentan con conocimientos sobre dichos temas, en relación al resultado se evidencia que un alto porcentaje del estudiante no conoce sobre automatismo.



III. VINCULACIÓN

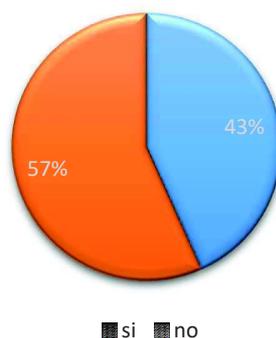
9. ¿Cree usted que el proyecto de automatización de los laboratorios de informática ayudaría a impulsar el desarrollo de nuevos proyectos basados en Open Source (Código Abierto)?

Tabla 13. Resultados Encuesta Pregunta 9.

Alternativas	F	%
SI	13	43%
NO	17	57%
TOTAL	30	100%

Fuente: Estudiantes

Ilustración 10. Resultados Encuesta Pregunta 9.



Autores: Reyes Wellington – Quiroz Jonathan

Análisis: Se puede observar en la gráfica que el 57% de los estudiantes no concuerda que el proyecto de automatización ayudara a la creación de nuevos proyectos basados en open source, mientras que el 43% restante respalda que se crearan nuevos proyectos basados en código abierto.



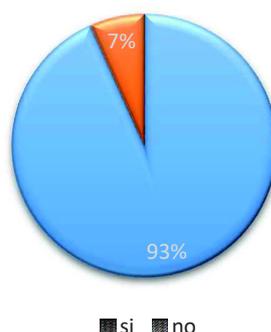
10. En base a las necesidades que tenga la comunidad ¿Piensa usted que podría implementarse un proyecto de automatización para mejorar la calidad de los servicios que brindan las entidades publica o privada?

Tabla 14. Resultados Encuesta Pregunta 10.

Alternativas	F	%
SI	28	93%
NO	2	7%
TOTAL	30	100%

Fuente: Estudiantes

Ilustración 11. Resultados Encuesta Pregunta 10.



Autores: Reyes Wellington – Quiroz Jonathan

Análisis: Se puede observar en la gráfica que el 93% de los estudiantes concuerda que el proyecto de automatización ayudara a la creación de nuevos proyectos basados en open source, mientras que el 7% restante no respalda que se crearan nuevos proyectos basados en código abierto.



2.9.2. INFORME FINAL DEL ANALISIS DE RESULTADOS

Una vez realizada la encuesta con los resultados obtenidos arrojaron los siguientes parametros mediante las tecnicas de investigacion utilizadas lo cual se deduce lo siguiente:

El 90% de los encuestados consideran que los laboratorios deberían ser automatizados.

El 73% de los encuestados no está conforme con el tiempo de apertura de los laboratorios al momento de recibir clases es consciente en la tardanza al momento de su apertura.

El 60% de los encuestados conoce sobre las tarjetas Arduino sea de manera directa o indirecta.

El 80% de los encuestados no tiene conocimiento sobre open source en su mayoría tiende a confundirlo con software libre siendo conceptos diferentes.

El 60% de los encuestados no tiene conocimiento sobre hardware libre, probablemente por qué no asocia este hardware a los ámbitos open source.

Respecto a la programación el 83% de los encuestados desconoce el IDE con que trabaja Arduino al confundirlo con NetBeans IDE.

El 53% de los encuestados no tiene conocimiento sobre software libre, probablemente por qué en su mayoría utilizan el sistema operativo Windows o Mac OS.



El 87% de los encuestados tiene conocimiento sobre el concepto de automatización.

El 57% de los encuestados no creen que con el proyecto de automatización de los laboratorios de informática dará la pauta para aumentar la creación de proyectos basados en open source.

El 93% de los encuestados creen que este tipo de proyectos de automatización se pueden aplicar a mejorar los servicios brindamos en entidades públicas o privadas.

CAPÍTULO III:
MARCO PROPOSITO



3.1 – INTRODUCCIÓN

En el presente capítulo se abordará la propuesta del desarrollo del proyecto de automatización de puertas en los laboratorios de informática para la apertura remota mediante dispositivos móviles y computacional en la facultad de ciencias informáticas de la Universidad Laica “Eloy Alfaro de Manabí” será implementado en los laboratorios de informática, describiendo las fases de desarrollo e implementación. Además, se presentarán los recursos humanos, tecnológicos e económicos que han intervenido para el desarrollo de la propuesta, se tratarán puntos importantes como las diferentes fases de desarrollo e implementación del proyecto, factores determinantes para la ejecución del proyecto, además los beneficios tangibles e intangibles que tendrá los laboratorios de informática con la implementación de este proyecto.

3.2 – DESCRIPCIÓN DE LA PROPUESTA

Después de haber identificado el problema en el capítulo I, con sus causas y efectos que genera, se realizó un análisis y se planteó la propuesta de automatización de puertas en los laboratorios de informática para la apertura remota mediante dispositivo móvil y computacional en la facultad de ciencias informáticas de la Universidad Laica “Eloy Alfaro” de Manabí.

Por lo tanto, este proyecto de electrónica será de gran utilidad para los laboratorios de informáticas de esta manera se disminuirá el tiempo de respuesta que se tiene al momento de la apertura manual de los laboratorios obteniendo como resultado que los estudiantes y docentes tengan acceso a los mismo en el menor tiempo posible de manera eficiente y eficaz, mejorando los procesos de gestión laboral y seguridad.



3.3 – OBJETIVOS

Este proyecto tiene como principal objetivo desarrollar un sistema de automatización para los laboratorios de informática, los materiales de construcción de la parte electrónica estará conformada por microcontroladores módulos bluetooth que permitirá la comunicación entre dispositivos móviles y la parte controladora del sistema, contara con una conexión USB entre ambas parte el computador y el sistema principal donde se controlara el funcionamiento de las puertas en los laboratorios.

La finalidad de este proyecto de investigación, es la contribución y enseñanza sobre electrónica y sobre programación de micro controladores utilizados sobre una placa Arduino que es la base para el prototipo, de proyectos de electrónica, robótica.

3.4 – DETERMINACIÓN DE RECURSOS

3.4.1. Recursos Humanos

Estos recursos son indispensables para la parte investigativa e implementación del proyecto, los cuales se muestran a continuación:

Tabla 15. Recursos Humanos.

Nombre	Rol desempeñado en el proyecto
REYES ZAMBRANO WELLINGTON STEVEN	Tesista
QUIROZ MOREIRA JONATHAN JESÙS	Tesista
ING. PEDRO DELGADO	Asesor Tutor

Autores: Reyes Wellington – Quiroz Jonathan



3.4.2. Recursos Tecnológicos

Tomando en consideración que la tecnología es la parte Fundamental del proyecto se detallan en la tabla los recursos necesarios.

Tabla 16. Recursos Tecnológicos

Cantidad	Descripción
2	Tarjeta Arduino Uno R3
1	Computador
1	Impresora
1	Flash memory
1	Librerías de Open CV
1	Ide Arduino
1	Servicio de Internet

Autores: Reyes Wellington – Quiroz Jonathan

3.4.3. Recursos Económicos

Después de los análisis de los recursos utilizados y su factibilidad con lo que contará el proyecto, esto para determinar si es factible dentro de los parámetros de construcción.



Tabla17. Recursos Económicos

Ítem	Descripción	%	Costo comercial	Costo Proyecto
1	Humanos		\$ 1.150,00	\$ -
2	Mano de obra (Estructura de la parte electrónica)		\$ 150,00	\$ -
3	Mano de obra (Desarrollo del sistema, aplicación móvil)		\$ 1.000,00	\$ -
4	Materiales		\$ 800,50	\$ 800,50
5	Materiales utilizados en la construcción		\$ 800,50	\$ 800,50
6	Utilitarios		\$ 420,00	\$ 420,00
7	Resmas de papel A4		\$ 20,00	\$ 20,00
8	impresiones y copias		\$ 200,00	\$ 200,00
9	suministros de oficina		\$ 110,00	\$ 110,00
10	empaste de documentos		\$ 80,00	\$ 80,00
11	anillados		\$ 10,00	\$ 10,00
12	otros		\$ 490,00	\$ 490,00
13	servicio de electricidad		\$ 60,00	\$ 60,00
14	servicios de internet		\$ 150,00	\$ 150,00
15	viáticos		\$ 280,00	\$ 280,00
	Subtotal (1+4+6+12)		\$ 2.860,50	\$ 1.710,50
	Gastos adicionales (Imprevistos) (5%)	5%	\$ 143,03	\$ 85,53
	total		\$ 3.003,53	\$ 1.796,03

Autores: Reyes Wellington – Quiroz Jonathan

3.5. ETAPAS DE ACCIÓN PARA EL DESARROLLO DE LA PROPUESTA

3.1.1. Selección del entorno de desarrollo del sistema computacional.

El entorno de desarrollo escogido para la realización del proyecto de automatización del sistema computacional es de código abierto, el programa llamado netbeans, el cual nos permite realizar programas en entornos de código abierto, dando una facilidad para uso en cualquier sistema operativo instalado en el equipo computacional, siempre y cuando permita la utilización de la plataforma java, los cuales, hasta el momento están en todos los sistemas existentes.



Una vez seleccionado el programa se decidió que el método o metodología utilizada para la creación y desarrollo del mismo el cual es la metodología de Cascada, el cual nos permite mirar o ir a un final, permitiendo un inicio, seguimiento, diseño y desarrollo del sistema, además de la metodología de prototipo que también se utilizó, ya que esta nos permite crear programas básicos pero funcionales para probar e ir utilizando los dispositivos Arduino para un manejo correcto desde un principio.

3.1.2. Programación y desarrollo del sistema computacional

El proceso de desarrollo del sistema computacional que utilizará el sistema de apertura remota fue puesto en marcha al seleccionar el software de programación que se utilizará, los casos de uso que utiliza el sistema fueron tomados en las encuestas realizadas a los estudiantes, personal docente y administrativo, para lo cual se realizó una recopilación de requerimientos y se seleccionaron los funcionales, dando así paso al inicio del desarrollo del programa. Este es un programa orientado a objetos (POO) el cual nos permite realizar una interfaz gráfica para que el usuario promedio pueda manejarlos, sin la necesidad de la utilización de código.

3.1.3. Metodología de programación

Existen diferentes tipos de metodologías para la creación del sistema principal que controlará los dispositivos de automatización.

Modelo de cascada: Este es un modelo que nos permite realizar procesos, o al etapa de desarrollo es hacia abajo, o en caída, el cual nos permite realizar un análisis de las necesidades, el diseño, la implantación, realizar pruebas o validaciones, integración y mantenimiento. Este modelo está dividido en secuencias con una que otra superposición.



Metodología de prototipo: Este nos permite desarrollar prototipos a medida que se va realizando el proyecto, a gran ventaja que, sin estar terminado nos permite dar una probada del producto final, sin tener todo lo referente al mismo, lo cual nos permite paso a paso mejorar el sistema hasta estar concluido.

Metodología incremental: Este nos permite tener una estrategia para mantener controlada la complejidad y riesgos del sistema, siendo así, una metodología en otra, es decir, la suma de varias metodologías en cascada forman la metodología incremental, por lo que se definen los requisitos antes de proceder con el avance, utilizando primeramente una metodología en cascada para realizar el concepto inicial de software, el análisis de las necesidades y el diseño de la arquitectura del mismo.

Metodología en espiral: En esta metodología, la atención se centra en la evaluación, en el riesgo y la mejoría del sistema, permitiendo así dividir el proyecto en segmentos tan pequeños que estos proporcionen la facilidad durante el cambio de proceso de desarrollo, siendo esto que cada ciclo inicia con la identificación de las partes interesadas, condiciones de ganancia y concluye en la revisión y finalización.

Metodología XP: Esta metodología de desarrollo de ingeniería de software nos permite adoptar las mejores metodologías de desarrollo de acuerdo a lo que se pretende realizar en el proyecto o sistema y aplicarlo de manera dinámica durante el ciclo de vida del sistema.

Metodología SCRUM: Esta es una metodología se aplica para obtener un mejor resultado para trabajar colaborativamente, en equipo para obtener una excelente respuesta en cuanto al estudio del mismo, obteniendo así resultados flexibles, definidos y productivos para un mejor uso.



3.1.3.1. Selección de la Metodología de programación

La metodología utilizada en la creación del sistema fue de tipo prototipo, el cual nos permitía crear prototipos poco a poco para las pruebas tanto de hardware como de software, probando así los equipos de Arduino y el sistema automático, dando así un correcto funcionamiento del mismo cuando este mismo esté culminado.

Así mismo, se utilizó en ciertas partes la metodología de cascada, el cual nos permitía observar el proyecto ya finalizado, no sin antes hacer un análisis, diseño, implementación, pruebas y hacer un empujón en los horarios, planificación, presupuestos. Un estricto manejo se mantiene durante toda la vida del sistema, del proyecto o semestre, lo cual permite retroalimentarse gracias al usuario, sin dejar de ser obsoleto el sistema propuesto.

3.2. Organización de los requisitos

Los requisitos obtenidos para la creación del sistema computacional se recogieron a medida que la encuesta se iba realizando, tomando en cuenta opiniones tanto de estudiantes como de docentes y personal administrativo, siendo estos requisitos buenos y malos para el sistema en cuestión, por lo que al finalizar la recolección de requisitos, se organizó para tener los requisitos funcionales y no funcionales.

Entre los requisitos funcionales tenemos:

- Registro de los horarios
- Función de registro una vez por semestre
- Bloqueo al sistema principal
- Función manual desde el computador parecido al sistema móvil



- Estado de abierto o cerrado avisando el cómo se encuentra el o los laboratorios.

Entre los requisitos no funcionales tenemos:

- El sistema tiene que ser accesible para todos
- El sistema debe ser en un código para una sola plataforma
- Que solo permita abrir y cerrar desde el pc dependiendo del usuario

3.3. Construcción del sistema automático computacional

El sistema computacional no tuvo costo efectivo, por lo que todo fue manejado en código libre, permitiendo así una mayor operatividad, siendo esto una ventaja aún mayor. Este proyecto se realizó pensando en los diferentes tipos de sistemas operativos, por lo que este sistema tendría que funcionar bajo cualquier circunstancia en la que se encuentre.

La conexión a base de datos utilizada se encuentra en dos partes:

Base de datos MySQLite: En este se encuentran los datos de usuario que se registran en el sistema y el puerto de comunicaciones que utilizará el sistema Arduino uno para su manejo, permitiendo utilizar sólo una vez la opción de conexión.

Archivo de Texto Plano: En este se encuentran los datos de los horarios establecidos para los laboratorios 201, 202, 203 y 206, los cuales serán almacenados una sola vez por semestre si así lo desea el usuario, permitiendo así el proceso de automatización para el sistema, siendo esto que sólo necesite del usuario una sola vez cada semestre.



La programación realizada en el software libre se encuentra en dos partes:

Archivo de conexión: En esta parte se encuentran las líneas de conexión que necesita el sistema para tomar y guardar los datos de usuario y puerto de comunicaciones que necesite el sistema Arduino.

Archivos de código: En esta parte se encuentran los diferentes comandos, tiempos y diseños que necesita el programa para tener la funcionalidad de "automático" por lo que al tomar los datos necesarios de los diferentes de archivos de texto plano, estos tienen un tiempo de refresco o de actualización de 15 segundos, es decir, que una vez concluido este tiempo, se vuelve a tomar los datos almacenados y tiempos, y, dependiendo de esto, se tiene una apertura o cerramiento de las puertas de los laboratorios.

3.5. ETAPAS DE ACCIÓN PARA EL DESARROLLO DE LA PROPUESTA

3.5.1. Análisis de requerimientos

En esta fase se establecieron los requisitos necesarios para la construcción del proyecto de automatización. Tales como el hardware, software para que ambos sean de código abierto (open source). Se escogieron los componentes electrónicos y las partes estructurales.

3.5.2. Identificación de componentes y herramientas a utilizar

En esta fase se establece el diseño físico y componentes del sistema del proyecto de automatización se escogieron los siguientes componentes y materiales:



3.5.2.1. Selección de materiales para la estructura

Madera:

La madera es un material moldeable, resistente estas características lo hacen idóneo para la construcción de estructuras en general, en este proyecto de automatización se utilizó para la construcción de la estructura que contendrá el sistema electrónico y partes internas de la estructura.

3.5.2.2. Selección de materiales electrónicos

Al realizar un estudio donde se analizaron las funciones que desempeñaría un proyecto de automatización y en base a esos resultados se escogieron módulos y componentes que aportarían a cumplir las funciones del proyecto estos componentes se detallan a continuación:

- Módulo Relay 8 canales.
- Módulo Bluetooth hc-05.
- Borneras.
- Fuente de poder Cctv 12 V.
- Jumper cable macho hembra.
- Wire Jumper Head Conector hembra.
- Ventilador 5 V.
- Cable # 10 flexible.
- Amarras plásticas negras pequeñas.
- Espagueti termo incogible 5mm.
- Jack macho para alimentación.
- Cerraduras Electromagnéticas.



3.5.2.3. Selección del entorno de desarrollo

El entorno de desarrollo escogido para la realización del proyecto de automatización es "Processing", el cual es un entorno de desarrollo que fue base para la creación del entorno de desarrollo de ARDUINO "Proseccing", es de código abierto y su finalidad es la de enseñar los fundamentos de programación en un contexto visual, todo esto siendo útil para el proceso de control mediante el dispositivo móvil.

3.5.2.4. Selección del hardware Arduino

Al plantear el proyecto se escogió la plataforma electrónica ARDUINO para el diseño y construcción del proyecto siendo una plataforma de código abierto, se pueden realizar adaptaciones según las necesidades, esta plataforma electrónica cuenta con varios modelos de la cuales se utilizó el más acorde a los requerimientos de este Proyecto.

Tabla 18: Selección de Hardware Arduino

	ARDUINO UNO	ARDUINO MEGA 2560	ARDUINO DUE
Microcontrolador	Atmega328	Atmega2560	AT91SAM3X8E
Pines Digitales I/O	14	54	54
Memorias Flash	32 KB	256 KB	512 KB
SRAM	2 KB	8 KB	66 KB
Velocidad de reloj	16 MHZ	16 MHZ	84 MHZ

Autores: Reyes Wellington – Quiroz Jonathan

En la tabla anterior se muestra una comparación de los tipos de placas más utilizados de la plataforma ARDUINO, para el proyecto de automatización



se escogió el modelo ARDUINO UNO R3 ATMEGA328 por ser el más adecuado por contar con los suficientes requerimientos para realizar las diferentes conexiones, además cuenta con una memoria flash y velocidad de reloj necesarias para obtener una respuesta funcional.

3.5.2.5 Selección del software de desarrollo del sistema computacional

La selección del sistema que utilizarán las computadoras de la facultad pasó por un proceso de descarte, en el cual se llegó a estimar varias opciones de plataformas, y a su vez, se tomaron las opciones de sistemas operativos, dando como resultado la elección de un sistema versátil y moldeable que funcione con todas las plataformas y sistemas existentes hasta el momento, reduciendo costos y mejorando el tiempo de uso de los mismo, llegando a utilizar el programa de desarrollo de Netbeans, el cual, en su plataforma JAVA, nos permite ejecutarse en los dos sistemas operativos más utilizados hasta el momento, siendo esto una forma rápida en tiempo y menor costo.

3.3.1. Diseño y características del dispositivo

En esta fase se realizaron análisis de la estructura que contendrían los diferentes componentes internos y externos, considerando las funciones que cumpliría, al finalizar este análisis se optó por darle forma rectangular, en donde la estructura constaría de una tapa superior que contendrá un ventilador que servirá como disipador de aire y puertos para las conexiones eléctricas y de conexiones USB y los componentes electrónicos que se utilizaron.

3.3.1.1. Construcción de la estructura del proyecto de automatización

Después de finalizar las fases de requerimiento y la fase de diseño de la parte estructural del proyecto, se llegó a determinar que los materiales escogidos



para la fabricación serían la madera la madera sirvió para diseñar la estructura que serviría de contenedor de las partes electrónicas.

3.3.1.2. Construcción de la parte electrónica del proyecto de automatización

La construcción del circuito del proyecto de automatización se comenzó después de realizar pruebas en un ambiente controlado a los componentes electrónicos y después de un análisis previo a la construcción, este análisis sirvió para ajustar el circuito a la estructura, el componente principal es la placa electrónica ARDUINO UNO R3 ATMEGA328, la cual es la encargada de procesar las instrucciones que permiten el control del proyecto de automatización.

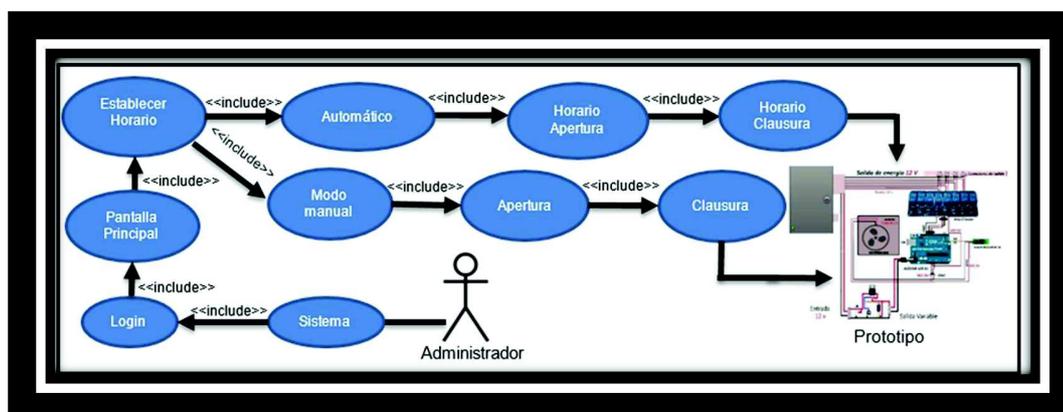
3.3.2. Fase de análisis sobre el funcionamiento del Proyecto

En esta fase se detallará las funcionalidades del proyecto a continuación se detallará a través de un diagrama de caso de uso.

3.3.2.1. Casos de uso

A continuación, se detallarán los diagramas de casos.

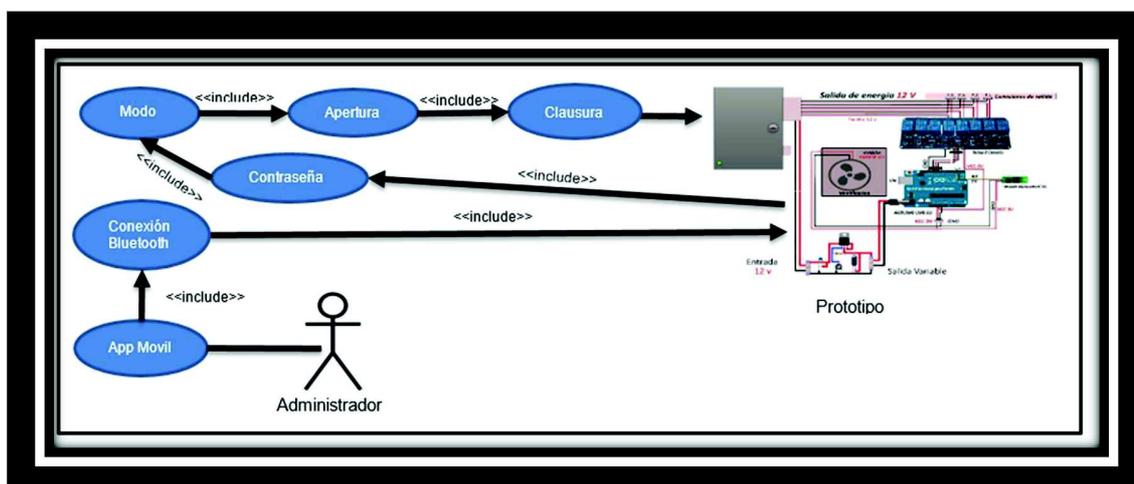
Ilustración 11. Caso de uso 1 Sistema Principal.





En el diagrama de caso de uso 1 se muestra las funciones principales para el manejo de los laboratorios para realizar estas operaciones el administrador debe asignar los parámetros que muestra el sistema principal estos parámetros son procesados por la tarjeta Arduino uno permitiendo la apertura y clausura de los laboratorios de dos modos el modo automático que permite ingresar la hora de apertura y clausura sin la necesidad de estar presente al momento de la ejecución y el modo manual que permite realizar la misma función pero con diferencia que se tiene que estar presente para su ejecución.

Ilustración 12. Caso de uso 2 Aplicación móvil.



Autores: Reyes Wellington – Quiroz Jonathan

En el diagrama de caso de uso 2 se muestra la función que desempeña la aplicación móvil cuenta con dos medidas de seguridad la primera es la opción de búsqueda del módulo bluetooth el cual se encuentra en el ubicado en el proyecto el cual tiene que ser emparejado para su correcto funcionamiento la segunda medida de seguridad es la del código de acceso de emparejamiento que fue modificada para disminuir el acceso no autorizado una vez realizado el



acceso se podrá usar las funciones de apertura y clausura de los laboratorios de forma independiente del sistema principal.

3.5.4. Implementación

Una vez concluido el proyecto de automatización y realizadas las pruebas pertinentes, se realizó la implementación se implementó en las instalaciones del laboratorio de informática de la FACCI, además se realizó una capacitación sobre el manejo del proyecto de automatización.

1. Primeramente, se determinó con los administradores de los laboratorios de informática, un lugar adecuado para ubicar el proyecto de automatización, donde no obstaculice a los estudiantes ni a los docentes.
2. En segunda instancia, se procedió a instalar la aplicación del sistema principal en el computador y la aplicación Android en el dispositivo móvil, para poder realizar el funcionamiento del proyecto de automatización.

3.5.4.1. Capacitación a los administradores del laboratorio de informática

La capacitación se realizó a los administradores de los laboratorios de informática y docentes invitados, para que conocieran y aprendieran el funcionamiento del sistema principal del proyecto de automatización.

En la capacitación los administradores y los docentes invitados aprendieron el funcionamiento y las etapas de construcción del sistema.



3.5.4.2. Manual de usuario

Para el uso del sistema del proyecto de automatización se añade un manual de usuario, el mismo, que brinda las facilidades y pasos necesarios para un correcto manejo. Los puntos principales a tratar en este manual son:

- Configuración del sistema principal.
- Diagramas electrónicos.
- Programación de Arduino UNO R3 ATMEGA328.
- Programación del sistema Principal.
- Fotos de la construcción del proyecto de automatización.

3.5.5. Pruebas

En esta etapa se realizaron las pruebas funcionales y el desempeño del proyecto de automatización.

3.5.5.1. Pruebas funcionales

Las pruebas funcionales se realizaron en los laboratorios de informática en estas pruebas el proyecto de automatización funciono satisfactoriamente, se probaron las aperturas de puertas por las funciones que realiza el sistema principal, la conexión mediante la aplicación móvil respondió satisfactoriamente en sus diferentes procesos.

3.5.5.1. Pruebas de desempeño

Para estas pruebas se analizaron diferentes tipos de respuestas los cuales determinan el desempeño del proyecto de automatización y del sistema principal para ejecutar las instrucciones.



Tabla 19: Pruebas de desempeño

Tipo	Descripción
Plataformas	El sistema principal es multiplataforma Windows, Linux, su conexión es mediante USB y su aplicación móvil se ejecuta mediante conexión bluetooth en el sistema operativo Android.
Eficiencia	La eficiencia del sistema de automatización de puertas es del 99.9% en tiempo real.

Autores: Reyes Wellington – Quiroz Jonathan

CAPÍTULO IV:
VALIDACIÓN DE RESULTADOS



4.1 – INTRODUCCIÓN

En el presente capítulo se evidenciará la validación de los resultados del proyecto la propuesta del desarrollo de automatización de puertas en los laboratorios de informática para la apertura remota mediante dispositivos móviles y computacional en la Facultad de Ciencias Informáticas de la Universidad Laica "Eloy Alfaro" de Manabí, mostrando el cumplimiento de los objetivos propuestos y detallando los resultados de la implementación realizada, en este capítulo se redactaran las conclusiones y recomendaciones.

4.2 – SEGUIMIENTO Y MONITOREO DE RESULTADOS

4.2.1 Automatización de puertas en los laboratorios de informática

Es un proyecto electrónico que se ha implementado en las instalaciones del laboratorio de informática perteneciente a la Facultad de ciencias informáticas FACCI, este proyecto se construyó para servir como una herramienta de automatización la cual permita mejorar el tiempo de respuesta en las aperturas de los mismo.

4.2.2 Funciones del proyecto de Automatización de puertas en los laboratorios de informática

El proyecto de Automatización consta de las siguientes funciones:

- El módulo Bluetooth HC-05 permite la comunicación con dispositivos pudiendo ser configurado en 2 modos: esclavo y maestro, permitiendo cambiar su nombre por defecto al igual que su contraseña de acceso.



- El módulo relay 8 canales es un dispositivo que consta de 8 relay con sus respectivas salidas de voltaje tiene como función cortar o activar el flujo de energía dependiendo de qué orden reciba.
- La placa Arduino es una placa de hardware libre que se utiliza para prototipos en general por su fácil manejo y soporte esto brindara las ventajas de modificar sus salidas o pines dependiendo de las necesidades que se requieran esta placa será la encargada de controlar los módulos que tenga conectados.
- Un Ventilador común que servirá como disipador de aire para mantener una temperatura recomendable en los componentes.



CONCLUSIONES

Al finalizar el presente proyecto se define si los objetivos planteados fueron alcanzados, entonces se puede concluir que:

- Se implementó un proyecto de automatización de puertas en los laboratorios de informática para la apertura remota mediante dispositivo móvil y computacional en la facultad de ciencias informáticas de la Universidad Laica “Eloy Alfaro” de Manabí.
- Se desarrolló un Firmware para “Arduino UNO R3”, el cual permite controlar a la placa de acuerdo a las instrucciones que están programadas. Además de un sistema principal que permite agregar un cronograma de uso en los laboratorios automatizando la apertura y clausura de las puertas dependiendo del horario que se haya ingresado.
- Con la implementación de este proyecto de automatización de las puertas en los laboratorios de informática los estudiantes y docentes tendrán mayor eficiencia reduciendo el tiempo de respuesta al momento de apertura de los laboratorios.



RECOMENDACIONES

Una vez concluido el desarrollo de este proyecto se recomienda lo siguiente:

- Verificar las conexiones al momento de conectar la parte electrónica al computador que tendrá el sistema principal porque una mala configuración puede perjudicar el correcto funcionamiento del dispositivo.
- Tener cuidado con la fuente de energía un incorrecto manejo puede causar choques eléctricos o quemaduras ya que tiene una entrada de energía 110 a 220 V.
- En el manual de usuario se detalla el funcionamiento y configuración de la interfaz del sistema principal y de la aplicación móvil se recomienda leer el manual de usuario antes de iniciar el sistema.
- Este proyecto puede servir como base para futuros proyectos de electrónica o domótica a mayor escala a implementarse en la FACCI.
- Utilizar una placa Raspberry Pi 3 modelo B como servidor local para albergar el sistema que controla las puertas.



BIBLIOGRAFÍA

1. Briones Cagua, A., & Briones Cagua, M. A. (2015). Software de control para el uso de los laboratorios de cómputo de la Universidad Laica "Eloy Alfaro" de Manabí. *Software de control para el uso de los laboratorios de cómputo de la Universidad Laica "Eloy Alfaro" de Manabí*. Chone, Manabí, Ecuador.
2. Vialfa- Carlos. (17 de Octubre de 2016). <http://es.ccm.net>. Obtenido de <http://es.ccm.net/contents/176-conector-db9>
3. cerrajero, M. d. (s.f.). Obtenido de <http://manualcerrajero.com/Cerradurascomunes.html>
4. Definicion.de. (s.f.). Obtenido de <http://definicion.de/lenguaje-de-programacion/>
5. Dolores, Z. P. (2013). Creacion de un laboratorio electrónico. Ecuador.
6. Enclada, E. J., & Ponce, A. I. (Mayo de 2015). Software de seguridad y control para el ingreso y salida vehicular en el Ala de Combate 23 de la ciudad de Manta utilizando tecnologías Identificación de radio frecuencia. *Software de seguridad y control para el ingreso y salida vehicular en el Ala de Combate 23 de la ciudad de Manta utilizando tecnologías Identificación de radio frecuencia*. Manta, Manabí, Ecuador.
7. Santos, M. M. (s.f.).
8. es.ccm.net. (s.f.). Obtenido de <http://es.ccm.net/contents/176-conector-db9>
9. Fernández, O. B. (16 de ebero de 2009). Obtenido de <http://www3.uji.es/~belfern/pdidoc/IX26/Documentos/introJava.pdf>
10. Fernández, O. B. (s.f.). Obtenido de <http://www3.uji.es/~belfern/pdidoc/IX26/Documentos/introJava.pdf>
11. Quiroz, Z. M. (s.f.).
12. Mellado, F. d. (1857). Diccionario de artes y manufacturas. Obtenido de <https://es.wikipedia.org/wiki/Cerradura#Referencias>



13. Porto, J. P. (2012). *http://definicion.de*. Obtenido de <http://definicion.de/lenguaje-de-programacion/>
14. Viñas, L. P. (1999). Circuitos y Dispositivos Electrónicos. Barcelona España: Editorial UPC
15. Melgar, E. R., & Diez, C. C. (2012). Arduino and Kinect Projects. New York EEUU: Editorial Apress.
16. Gomariz, S., Biel, D., Matas, J., & Reyes, M. (1998). Teoría de Control: Diseño Electrónico. Barcelona España: Editorial UPC.
17. Cousot, S., & Stanley, D. E. (10 de 2002). OPENCV: Processing and Java Library. Recuperado el 17 de 10 de 2013, de <http://ubaa.net/shared/processing/opencv/>
18. Roldan, J. A. (2001). Fundamentos de Electrotecnia y Electrónica Básica. Manta Ecuador: Editorial Suceso
19. Joniuz. (8 de Julio de 2014). Internet de las cosas. Obtenido de Midiendo temperatura y humedad con Arduino: <http://www.internetdelascosas.cl/2014/07/08/midiendo-temperatura-yhumedad/>
20. Artero, O. T. (2013). ARDUINO curso práctico de formación. México: Editorial Alfaomega.
21. ARDUINO. (15 de Enero de 2015). ARDUINO Página oficial. Obtenido de MPU-6050 Accelerometer + Gyro: <http://playground.arduino.cc/Main/MPU-6050>
22. Geekfactory. (14 de Enero de 2015). Módulo Bluetooth HC-05. Obtenido de Bluetooth HC-05 y HC-06 Tutorial de Configuración: <http://www.geekfactory.mx/radio/bluetooth-hc-05-y-hc-06-tutorial-deconfiguracion/>
23. Herrador, R. E. (2009). Guía de usuario de ARDUINO. Cordoba: Creative Commons Attribution-Noncommercial-Share Alike 3.0.
24. Barrientos, A., & Peñin, L. (1996). Fundamentos de robótica. Madrid España: Editorial Mc Graw Hill.
25. Barea, R. (s.f.). Introducción a la robotica. Madrid España: Departamento de Electrónica. Universidad de Alcalá.



AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ.

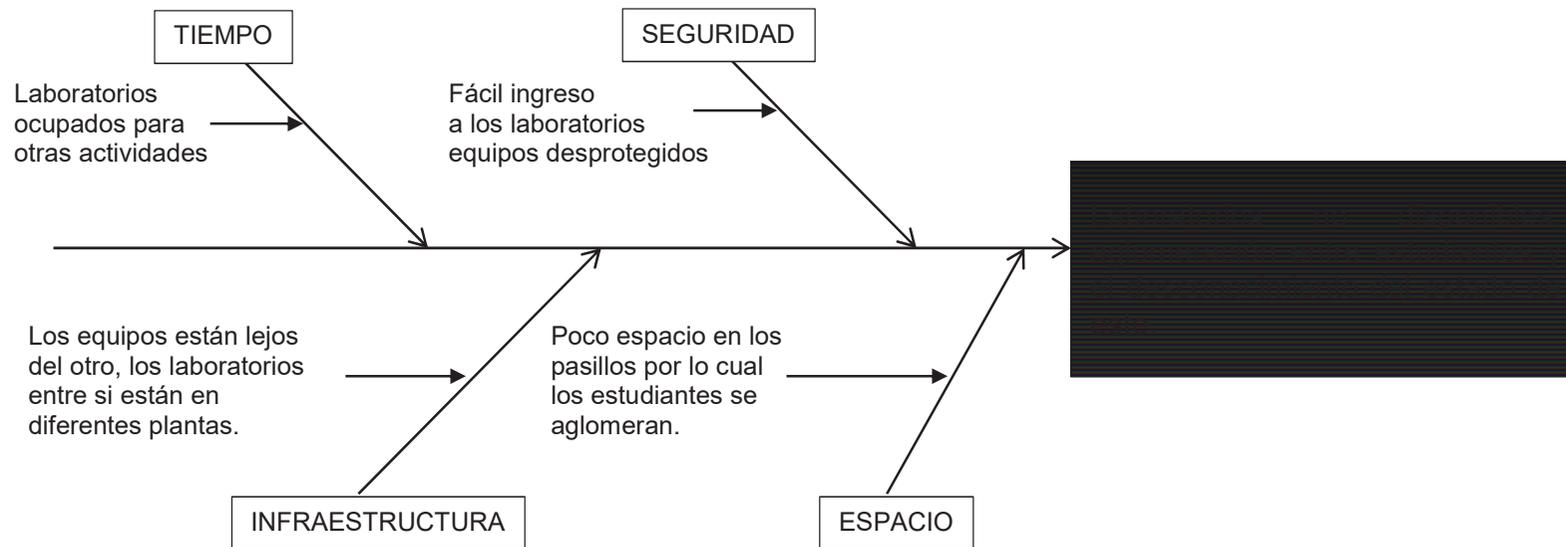


ANEXOS



Diagrama Causa – Efecto

Ilustración 1. Diagrama Causa Efecto.



Fuente: FACCI.

Autor: REYES ZAMBRANO WELLINGTON STEVEN
QUIROZ MOREIRA JONATHAN JESUS



AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ.



ANEXO 1: Solicitud de permiso de Implementación y capacitación.

Manta, 10 de Mayo del 2017

Sra. Lcda.
Dolores Muñoz Verduga, Mg. Decana
DECANA FACULTAD DE CIENCIAS INFORMÁTICAS ULEAM
Manta

De nuestras consideraciones:

Los estudiantes, QUIROZ MOREIRA JONATHAN JESÚS con C.I 131076276-8 y REYES ZAMBRANO WELLINGTON STEVEN, con C.I 131402948-7; nos dirigimos a usted para informarle que se ha culminado la elaboración de nuestro trabajo de titulación en su parte teórica y práctica, para certificar que nuestro trabajo de investigación es técnicamente operativo, se realizará la instalación implementación dentro del área designada, y posteriormente la capacitación al personal administrativo y docente de nuestro trabajo de titulación cuyo tema es: "AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA "ELOY ALFARO DE MANABÍ". Por lo que solicitamos se nos autorice proceder con la respectiva instalación, o se nos asigne una fecha para proceder con su autorización.

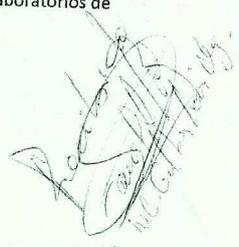
La instalación costara de dos fases:

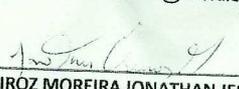
FASE1: Instalación de la parte física del proyecto en la sala de administración de los laboratorios de informática.

FASE2: Instalación del sistema principal y aplicación móvil para el manejo de los laboratorios de informática.

Por su atención a la presente, le quedo muy agradecido.

Atentamente:


REYES ZAMBRANO WELLINGTON STEVEN
C.I.131402948-7
E-mail: e1314029487@live.uleam.edu.ec


QUIROZ MOREIRA JONATHAN JESUS
C.I.131076276-8
E-mail: e1310762768@live.uleam.edu.ec

UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ
FACULTAD DE CIENCIAS INFORMÁTICAS
Recepción de Documentos

FECHA:	10 / 05 / 2017
HORA:	19:54
RECIBIDO POR:	Wendy
N° GUIA:	316
FIRMA:	



ANEXO 2: Registro de asistencia a la capacitación.

REGISTRO DE ASISTENCIA

Registro de asistencia a la capacitación del proyecto integrador "AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ", realizado el día 13 de julio del 2017 en el horario de 14:30 a 16:00 PM

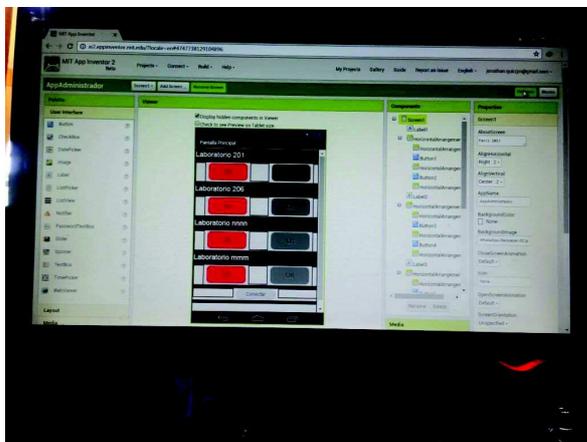
No	NOMBRE Y APELLIDOS	FIRMA
1	Francisco Toala Sotomayor	
2	Fabian ANCHUNDIA CORDERA	
3	Donny Pacheco	
4	Edgar Leon Muñoz	
5		
6		
7		
8		
9		
10		
11		



AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ.



ANEXO 3: Implementación.



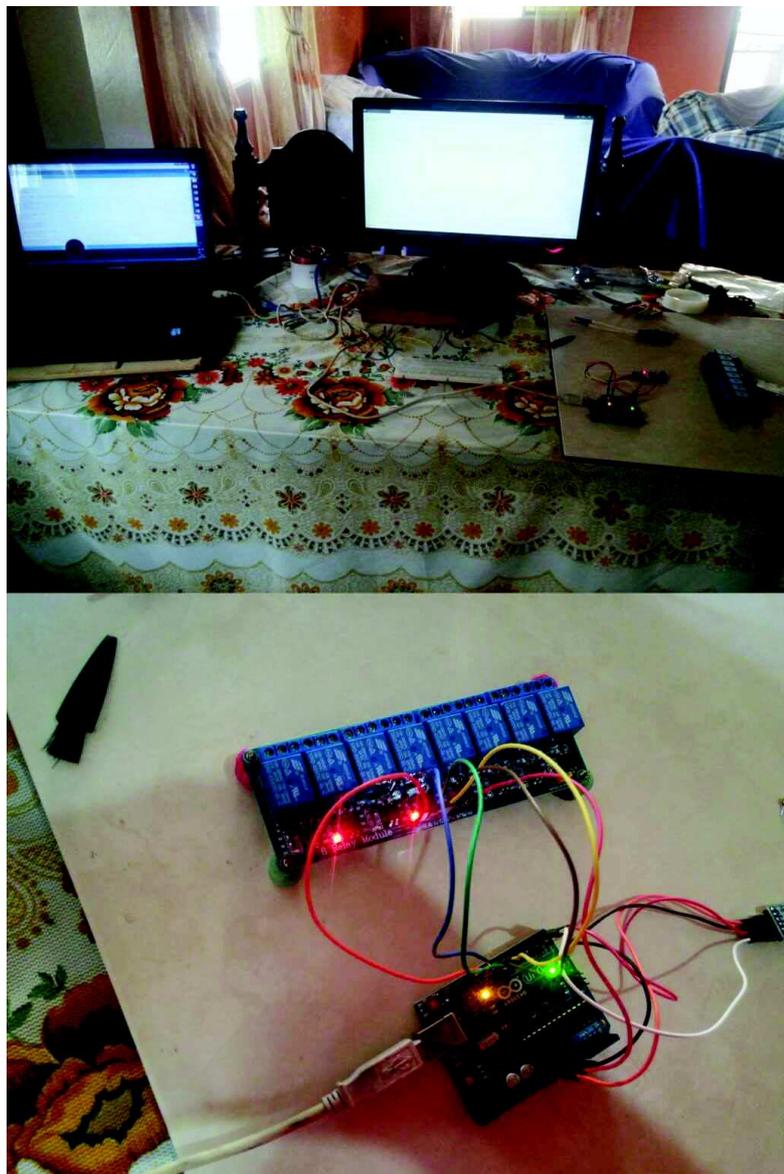


AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ.



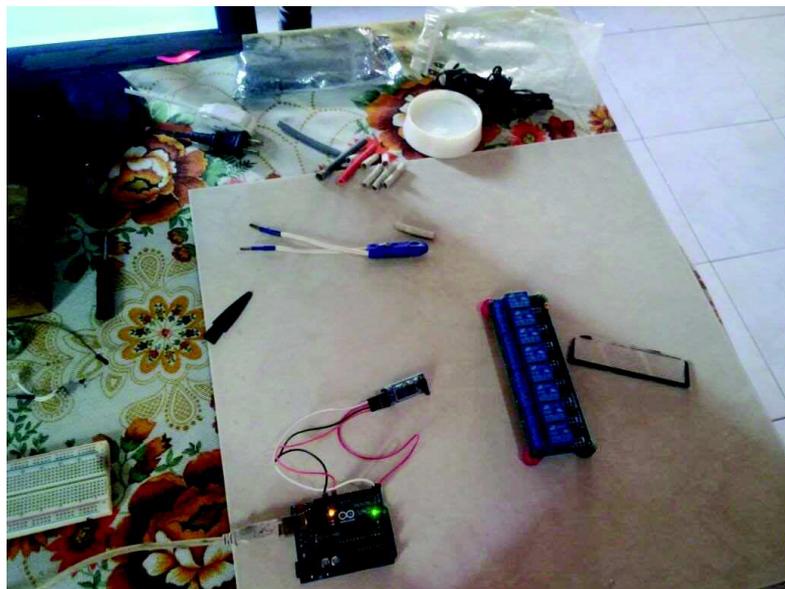


AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ.





AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ.





AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ.



ANEXO 4: Capacitación.





AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ.





AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ.





AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ.





AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ.





ANEXO 5: Presupuesto de Ensamblaje

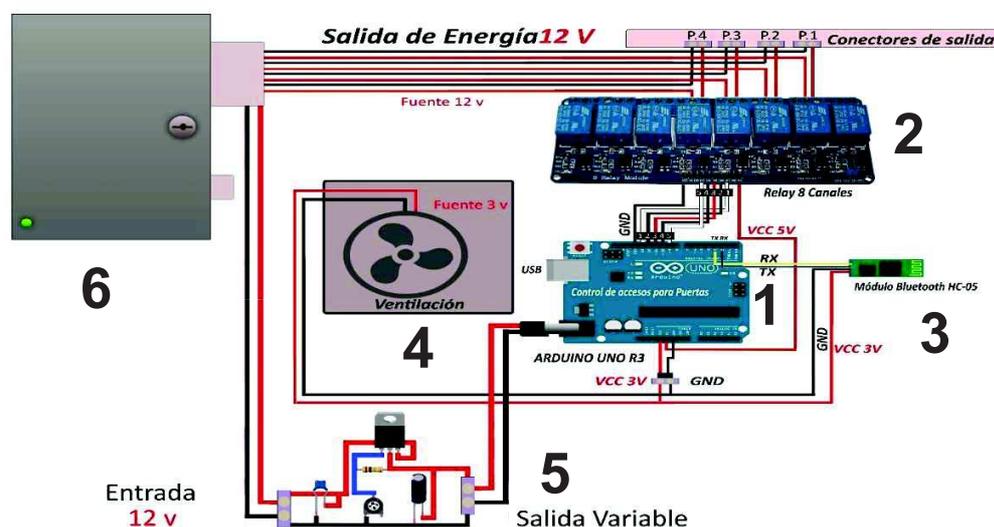
Presupuesto de ensamblaje

Item	Descripción	U	V. Unitario	Costo Total
1	Arduino uno r3	2	\$ 35,00	\$ 70,00
2	Módulo Relay 8 canales	1	\$ 30,00	\$ 30,00
3	Módulo Relay 1 canal	1	\$ 10,00	\$ 10,00
4	Modulo Bluetooth HC-05	1	\$ 24,00	\$ 24,00
5	Cerradura Electromagnética	4	\$ 64,00	\$ 256,00
6	Bornera 2 pines	12	\$ 0,30	\$ 3,60
7	Bornera 3 pines	16	\$ 0,40	\$ 6,40
8	Tarjeta perforada	2	\$ 1,00	\$ 2,00
9	Conector Para Fuente Tipo macho	4	\$ 1,00	\$ 4,00
10	Conector Para Fuente Tipo Hembra	4	\$ 0,40	\$ 1,60
11	Madera 1X1 Metro	1	\$ 20,00	\$ 20,00
12	Hoja Adhesiva	2	\$ 1,00	\$ 2,00
13	Fuente de alimentación 12 v	1	\$ 60,00	\$ 60,00
14	Tornillo galvanizado para madera 4x20mm	10	\$ 0,09	\$ 50,00
15	Resistencia 1k	1	\$ 0,10	\$ 0,10
16	Velocímetro 10 k	1	\$ 2,00	\$ 2,00
17	Regulador LM78L05	1	\$ 1,00	\$ 1,00
18	Condensador electrolítico 1uf	1	\$ 0,30	\$ 0,30
19	Condensador cerámico 0,1uf	1	\$ 0,20	\$ 0,20
20	Jumper cable macho hembra	2	\$ 0,20	\$ 0,40
21	Ventilador 5 V	1	\$ 0,50	\$ 0,50
22	Cable # 20 flexible. 60 metros	1	\$ 30,00	\$ 30,00
23	Espagueti termoencogible 5mm 1m	1	\$ 0,80	\$ 0,80
24	Amaras plásticas negras pequeñas.	20	\$ 0,10	\$ 2,00
25	Canaletas	20	\$ 1,50	\$ 30,00
26	Extensión USB active 10 metros	2	\$25,00	\$50,00
27	Botones normalmente cerrados	4	\$0.50	\$2,00
28	Batería 12 V	1	\$20.00	\$20,00
TOTAL				\$ 678,90

Autores: Reyes Wellington – Quiroz Jonathan



ANEXO 6: Diagrama electrónico.



Autores: Reyes Wellington – Quiroz Jonathan

Información de los componentes

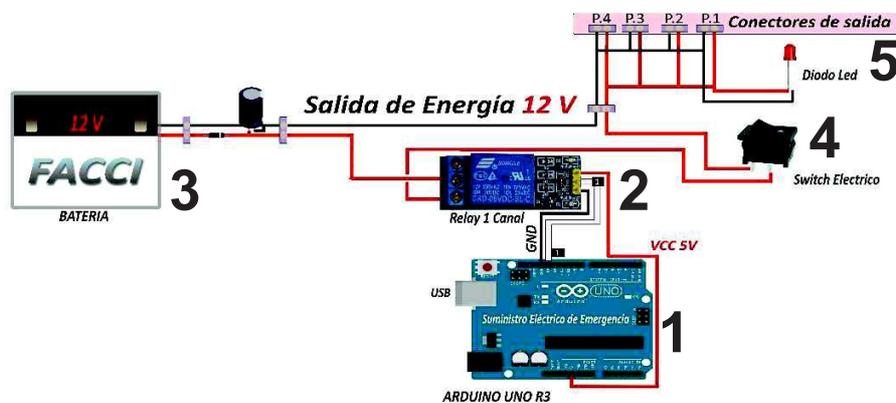
TEM	DISPOSITIVOS
1	Arduino uno R3 328
2	Relay 8 canales
3	Modulo Bluetooth HC-05
4	Ventilador
5	Regulador de voltaje variable
6	Fuente de alimentación

Información del Diagrama

INFORMACIÓN
RX: Comunicación Serial de recepción
TX: Comunicación Serial de transmisión
GND: Conexión a tierra
VCC: Voltaje de corriente continúa
USB: Universal Serial Bus



ANEXO 7: Diagrama electrónico respaldo de energía.



Autores: Reyes Wellington – Quiroz Jonathan

Información de los componentes

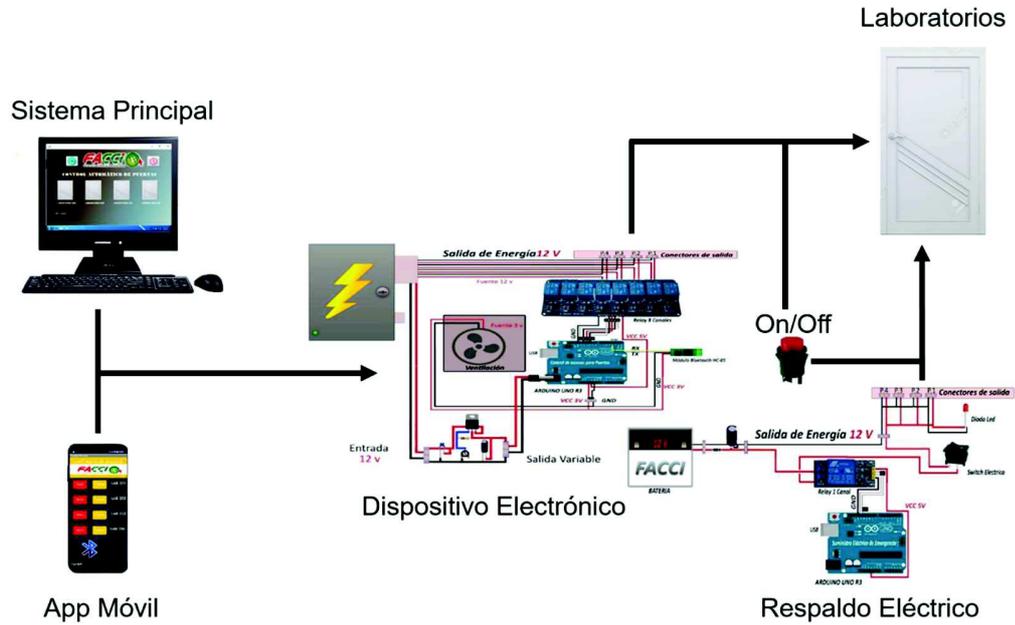
TEM	DISPOSITIVOS
1	Arduino uno R3 328
2	Relay 1 canal
3	Batería 12 V
4	Switch Eléctrico
5	Diodo Led

Información del Diagrama

INFORMACIÓN
P: Puertas
GND: Conexión a tierra
VCC: Voltaje de corriente continua
USB: Universal Serial Bus



ANEXO 8: Diagrama de Funcionamiento.



ANEXO 9: Fotos de construcción de la parte Estructural / Electrónica.



Módulo Bluetooth HC-05.



AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ.



Módulo Relay 8 canales.



Placa Arduino Uno R3



Fuente 12v



AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ.



Estructura del prototipo.



Vista interna de la parte electrónica.



Testeo de la parte electrónica.



ANEXO 10: Fotos de Instalación de la parte Estructural.



Vista delantera de la parte electrónica



Cerradura electromagnética instalada en los Laboratorios.



ANEXO 11: Ilustraciones.

Ilustración 14. Cerradura mecánica



Fuente: www.sidhe.es

Ilustración 15. Cerradura eléctrica



Fuente: www.ferreteriaxerez.com

Ilustración 16. Teléfono Inteligente



Fuente: www.dpyak.com

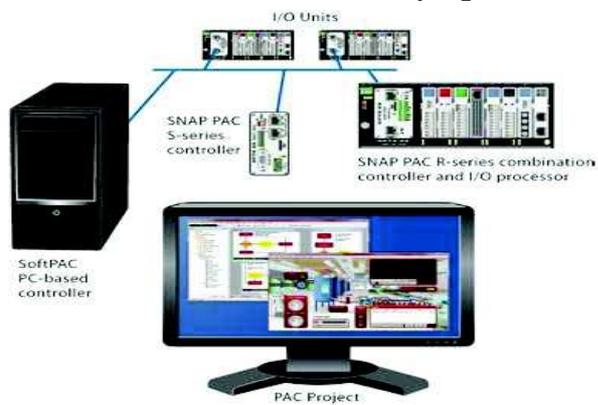


Ilustración 17. Automatización



Fuente: www.iesve.org

Ilustración 18. Automatización programable



Fuente: www.industriales.ws

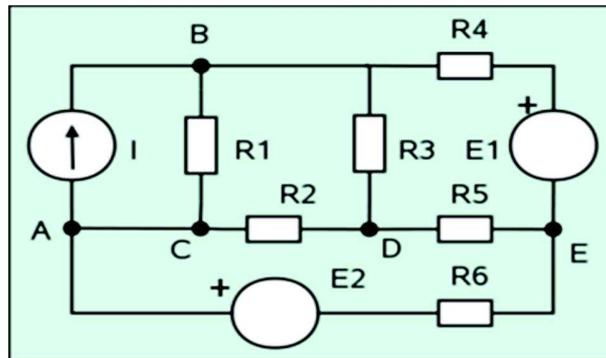
Ilustración 19. Lenguajes de programación



Fuente: www.campusmvp.es



Ilustración 20. Circuito Eléctrico



Fuente: jorbu.blogdiario.com

Ilustración 21. Relay



Fuente: arubia45.blogspot.com

Ilustración 22. Arduino Logo



Fuente: arduino.cc



ANEXO 12: Encuestas.

ENCUESTA PARA REALIZACIÓN DEL PROYECTO

AUTOMATIZACIÓN DE PUERTAS EN LOS LABORATORIOS DE INFORMÁTICA PARA LA APERTURA REMOTA MEDIANTE DISPOSITIVOS MÓVILES Y COMPUTACIONAL EN LA FACULTAD DE CIENCIAS INFORMÁTICAS DE LA UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ.

La información recolectada será confidencial, gracias por su colaboración. Las preguntas planteadas son cerradas, le pedimos que conteste con la mayor sinceridad posible, además no hay repuesta correctas e incorrectas, esta refleja su opinión.

De antemano ¡Muchas gracias por su colaboración!

I. INFORMACIÓN RESPECTO A LOS LABORATORIOS DE LA FACULTAD DE CIENCIAS INFORMÁTICA.

1. ¿Considera usted que los laboratorios de informática deberían ser automatizados?

SI () NO ()

2. ¿Esta conforme con el tiempo de apertura de los laboratorios al momento de recibir clases?

SI () NO ()



II. INFORMACIÓN SOBRE TARJETAS ARDUINO.

3. ¿conoce usted sobre las tarjetas ARDUINO?

SI () NO ()

4. ¿Tiene conocimiento sobre Open Source?

SI () NO ()

5. ¿Tiene conocimiento sobre Hardware libre?

SI () NO ()

6. ¿Conoce el lenguaje de programación que utiliza Arduino?

SI () NO ()

7. ¿Tiene conocimiento sobre Software libre?

SI () NO ()

8. ¿conoce usted el termino automatización?

SI () NO ()



III. VINCULACIÓN

9. ¿Cree usted que el proyecto de automatización de los laboratorios de informática ayudaría a impulsar el desarrollo de nuevos proyectos basados en Open Source (Código Abierto)?

SI () NO ()

10. En base a las necesidades que tenga la comunidad ¿Piensa usted que podría implementarse un proyecto de automatización para mejorar la calidad de los servicios que brindan las entidades publica o privada?

SI () NO ()



GLOSARIO

ULEAM: Universidad Laica Eloy Alfaro de Manabí.

FACCI: Facultad de Ciencias Informáticas.

ARDUINO: Plataforma electrónica de hardware y software para la construcción de prototipos.

BLUETOOTH: Tipo de tecnología para redes inalámbricas que permite la transmisión de voz y datos entre distintos dispositivos mediante una radiofrecuencia segura.

MÓDULO RELAY: Módulo que permite controlar el encendido y apagado de cualquier aparato que se conecte a una fuente de alimentación eléctrica externa

IDE: Integrated Development Environment (Entorno de Desarrollo Integrado).

FIRMWARE: Son instrucciones que controla físicamente al hardware.

RX: Comunicación Serial de recepción.

TX: Comunicación Serial de transmisión.

GND: Conexión a tierra.

VCC: Voltaje de corriente continua.



USB: Universal Serial Bus.

WIRING: Plataforma electrónica que sirvió como base para la construcción de las placas ARDUINO.

PROCESSING: Lenguaje de programación utilizado en el desarrollo de prototipos electrónicos.

OPEN SOURCE: (Código abierto) es el término con el que se conoce al software distribuido y desarrollado libremente.

MANUAL DE USUARIO

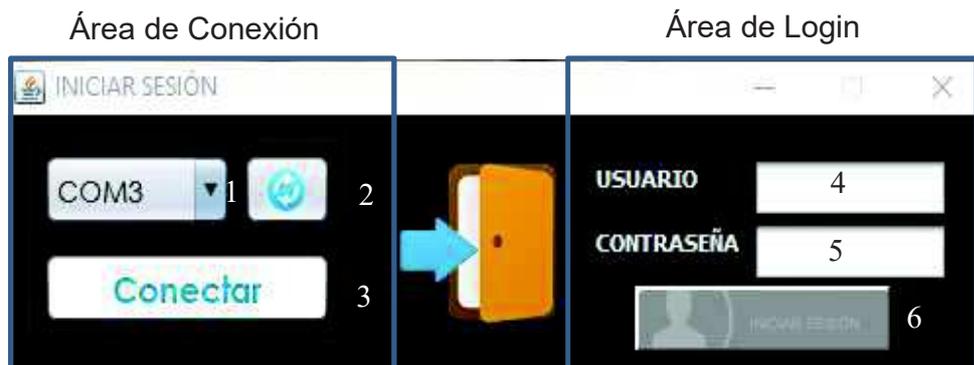
INTRODUCCIÓN

En este Manual de usuario se conocerá el funcionamiento de la parte electrónica como del software en sus interfaces (Control Principal, App Móvil) del sistema principal se dividirá en dos modos automático y manual en la app móvil se detallará la Vinculación con el dispositivo bluetooth y posterior conexión con la app para su uso de manera correcta se conocerá el ensamblado de la parte física y los componentes en su fabricación, el software utilizado en su programación y su código fuente.

Los términos y condiciones de estos Software tanto en su versión de escritorio como móvil corresponden legalmente a la Facultad de Ciencias Informáticas.

SISTEMA DE CONTROL

PANTALLA DE CONEXIÓN / INICIO DE SESIÓN



ÁREA DE CONEXIÓN

- **1:** Esta opción se procederá a buscar el puerto de la conexión USB del dispositivo anteriormente conectado el puerto varia de equipo en equipo.
- **2:** Esta opción nos permitirá actualizar cuando no aparezca el puerto de conexión anteriormente explicado en el punto No. 1.
- **3:** Esta opción se procederá a conectar el sistema con la parte electrónica la cual controla el funcionamiento de apertura y clausura de las puertas.

NOTA: Para poder acceder al inicio de sesión se debe cumplir los puntos del área de conexión.

ÁREA DE LOGIN

- **4:** Ésta área sirve para identificar el Usuario que Administrara los recursos del sistema de control.
 - **5:** Ésta área sirve para identificar La contraseña que complementa a la del usuario.
- 6:** Esta opción permite ingresar al sistema de control teniendo en cuenta haber cumplido los campos de Usuario y Contraseña explicados en los puntos No. 4, 5.

PANTALLA PRINCIPAL

Área Manual

Área Automática



ÁREA MANUAL

Clausurados

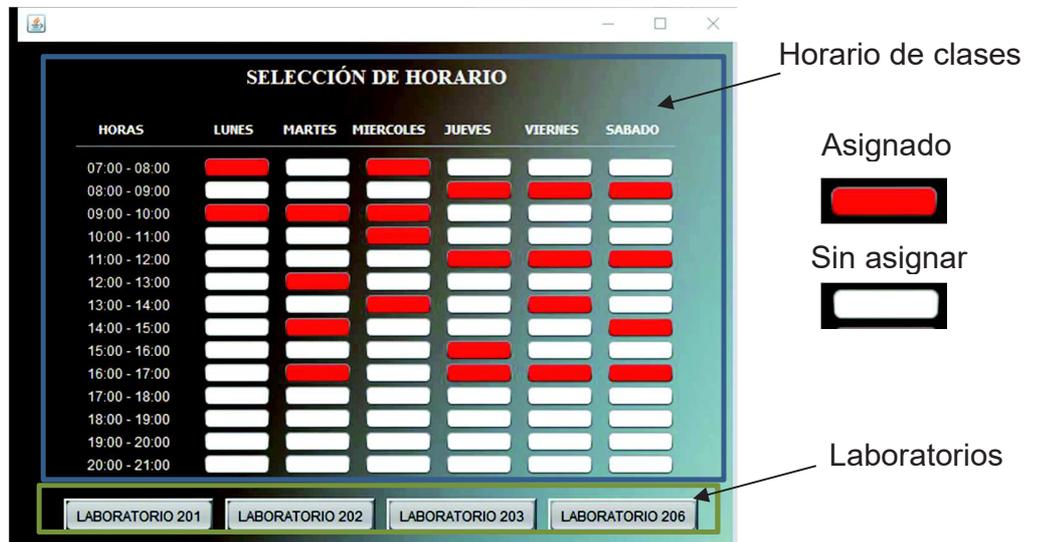


En funcionamiento



La opción manual consta de 4 botones que indican los diferentes laboratorios por defecto están en modo de clausura y se tiene que hacer click en el laboratorio deseado para su apertura al momento de estar en funcionamiento el botón cambia su color original.

ÁREA AUTOMÁTICA



The screenshot shows a window titled "SELECCIÓN DE HORARIO" with a grid of class times and days. The grid has columns for LUNES, MARTES, MIERCOLES, JUEVES, VIERNES, and SABADO, and rows for hours from 07:00 - 08:00 to 20:00 - 21:00. Red boxes indicate assigned times, and white boxes indicate unassigned times. A legend on the right shows a red box for "Asignado" and a white box for "Sin asignar". Below the grid are buttons for LABORATORIO 201, LABORATORIO 202, LABORATORIO 203, and LABORATORIO 206. Arrows point from the text "Horario de clases" and "Laboratorios" to the grid and buttons respectively.

HORAS	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO
07:00 - 08:00	Asignado	Sin asignar	Asignado	Sin asignar	Sin asignar	Sin asignar
08:00 - 09:00	Sin asignar	Sin asignar	Sin asignar	Asignado	Asignado	Asignado
09:00 - 10:00	Asignado	Asignado	Asignado	Sin asignar	Sin asignar	Sin asignar
10:00 - 11:00	Sin asignar	Sin asignar	Asignado	Sin asignar	Sin asignar	Sin asignar
11:00 - 12:00	Sin asignar	Sin asignar	Sin asignar	Asignado	Asignado	Asignado
12:00 - 13:00	Sin asignar	Asignado	Sin asignar	Sin asignar	Asignado	Sin asignar
13:00 - 14:00	Sin asignar	Sin asignar	Asignado	Sin asignar	Asignado	Sin asignar
14:00 - 15:00	Sin asignar	Asignado	Sin asignar	Sin asignar	Sin asignar	Asignado
15:00 - 16:00	Sin asignar	Sin asignar	Sin asignar	Asignado	Asignado	Sin asignar
16:00 - 17:00	Sin asignar	Asignado	Sin asignar	Asignado	Asignado	Asignado
17:00 - 18:00	Sin asignar					
18:00 - 19:00	Sin asignar					
19:00 - 20:00	Sin asignar					
20:00 - 21:00	Sin asignar					

HORARIO DE CLASES

Ésta área sirve para seleccionar el horario de funcionamiento del sistema por día y hora (Horario Continuo).

LABORATORIOS

Ésta área se complementa con la del horario de clases una vez configurado el horario de acuerdo al cronograma de clases en cada uno de los laboratorios estos se guardan automáticamente.

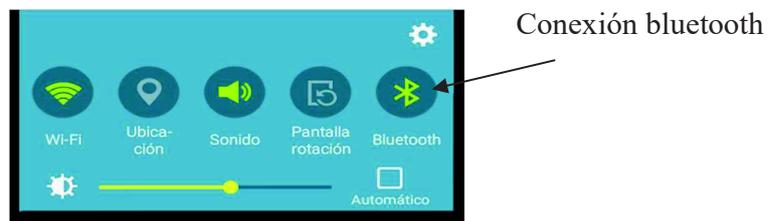
NOTA: En esta área solo se configura una vez por Semestre dado que el sistema administra la apertura y clausura de los laboratorios siguiendo el cronograma asignado en el horario.

APP MÓVIL

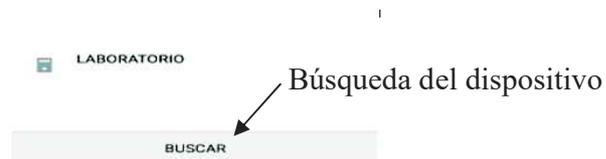
CONEXIÓN BLUETOOTH / VINCULACIÓN

Es esta fase el primer paso es buscar el dispositivo bluetooth y hacer el respectivo emparejamiento. Como se muestra en la siguiente imagen.

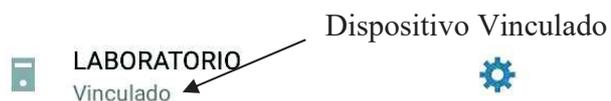
- **1:** Primero procederemos a activar la opción de bluetooth.



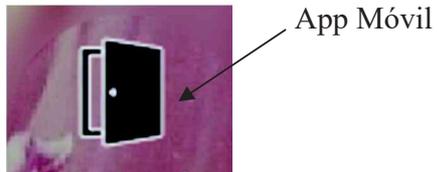
- **2:** luego se buscará el dispositivo bluetooth con el nombre de LABORATORIO para posteriormente emparejarlo a continuación aparecerá un cuadro de dialogo que pedirá el ingreso de la contraseña La cual fue proporcionada a los administradores de los laboratorios.



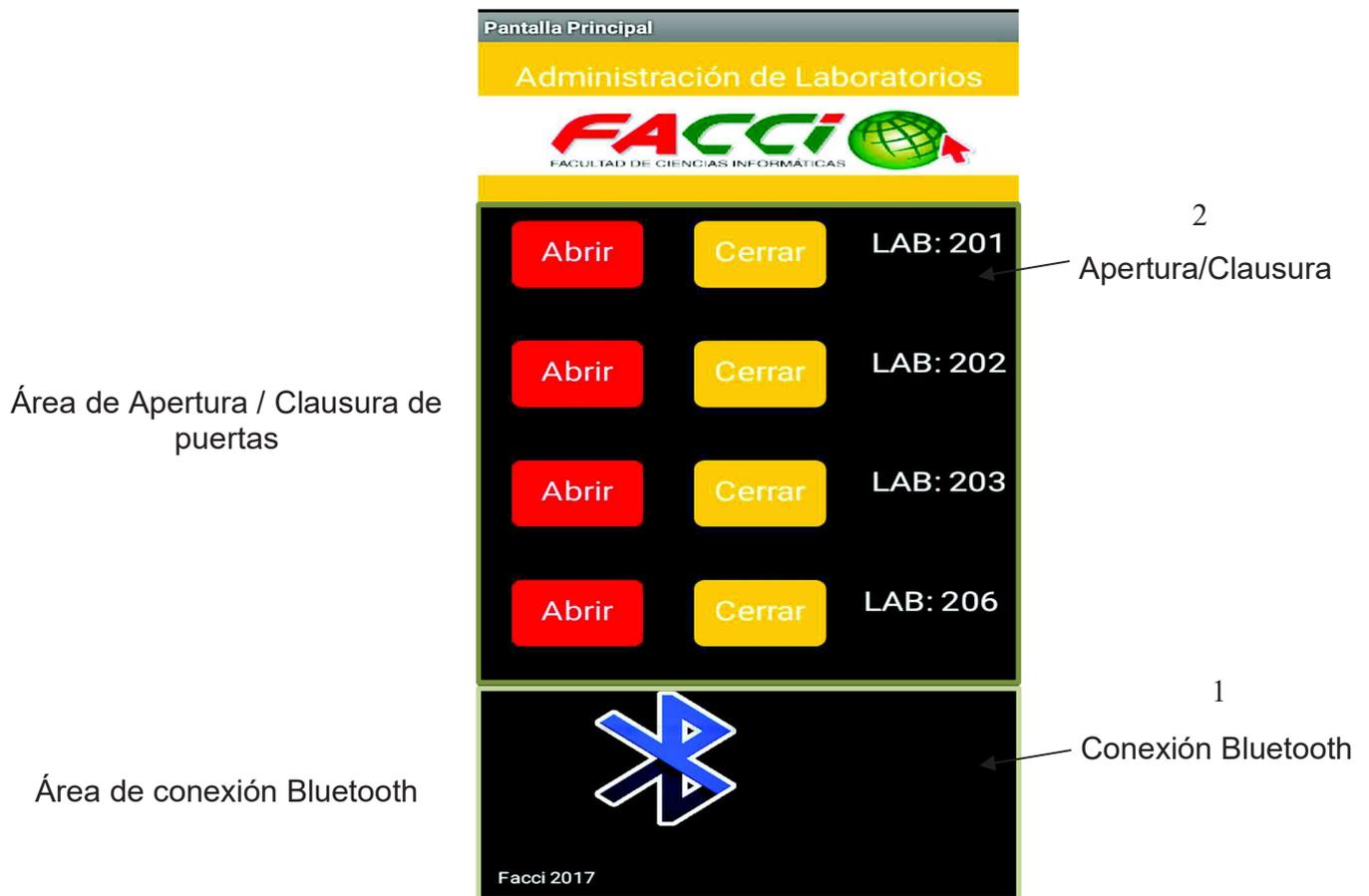
- **3:** A continuación, si ingreso correctamente la contraseña el dispositivo bluetooth saldrá vinculado al móvil terminando con la fase de emparejamiento.



- **4:** En este paso se buscará el icono de la app que tiene como logotipo una puerta cuyo nombre es administración de laboratorios luego se procederá a abrir la app.



PANTALLA PRINCIPAL



ÁREA DE CONEXIÓN BLUETOOTH

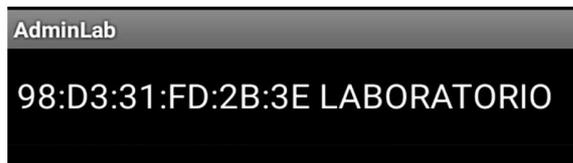
En esta área se procederá a la conexión del dispositivo bluetooth con la app para hacerlo se seguirá los siguientes pasos:

CONEXIÓN BLUETOOTH

Como primeros pasos para la conexión con el dispositivo bluetooth se seleccionará el botón que contiene el logotipo de bluetooth.



Posteriormente aparecerá una pantalla donde nos indicara que dispositivo están vinculados seleccionaremos el dispositivo que en pasos anteriores se vinculó con el dispositivo móvil y se finalizará la conexión dando paso al manejo de la app.

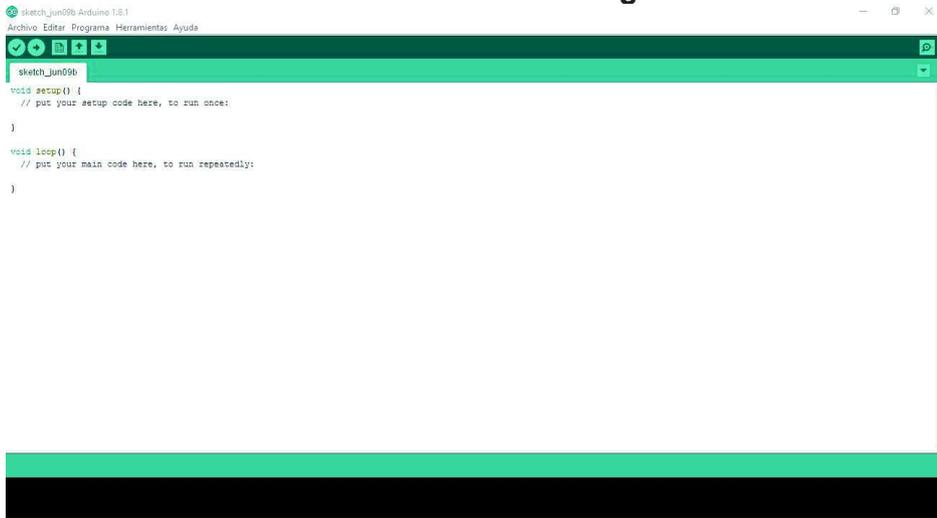


ÁREA DE APERTURA / CLAUSURA DE PUERTAS

En esta área se realizará la apertura y clausura de las puertas de manera manual la cual se divide por laboratorios cada laboratorio cuenta con 2 botones uno de apertura y otro de clausura los cuales permiten gestionar de manera independiente a cada uno.

IDE DE PROGRAMACIÓN / CODIGO FUENTE

Para la programación de la placa arduino se utilizó el Ide que proporciona arduino el cual está basado en Processing



CODIGO FUENTE ARDUINO

```
#include <SoftwareSerial.h>
SoftwareSerial Serial_2 (2, 3); // Crea nueva conexión pin2(RX) a TX y
Pin3(TX) a RX para el modulo bluetooth
```

```
int estado=0;
int retardo=100;
int analogo=0;
```

```
void setup(){
```

```
  pinMode(13,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(11,OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(8,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(3,OUTPUT);
  Serial.begin(9600); // se inicia con una frecuencia de 9600 baudios
```

```

    delay(1000);
}
void loop(){
if(Serial.available(>0){ // Si el puerto serie esta habilitado
    estado = Serial.read(); // Lee lo que llega por el puerto Serie
}

if(estado== 'a'){ // on/off de los pin 13 al pin 6
    digitalWrite(13,HIGH);
}
if(estado== 'b' ){
    digitalWrite(13,LOW);
}
if(estado== 'c'){
    digitalWrite(12,HIGH);
}
if(estado== 'd' ){
    digitalWrite(12,LOW);
}
if(estado== 'e'){
    digitalWrite(11,HIGH);
}
if(estado== 'f' ){
    digitalWrite(11,LOW);
}
if(estado== 'g' ){
    digitalWrite(10,HIGH);
}
if(estado== 'h' ){
    digitalWrite(10,LOW);
}
if(estado== 'i' ){
    digitalWrite(9,HIGH);
}
if(estado== 'j' ){
    digitalWrite(9,LOW);
}
if(estado== 'k' ){
    digitalWrite(8,HIGH);
}
if(estado== 'l' ){
    digitalWrite(8,LOW);
}
if(estado== 'm' ){
    digitalWrite(7,HIGH);
}

```

```

}
if(estado== 'n' ){
    digitalWrite(7,LOW);
}
if(estado== 'o' ){
    digitalWrite(6,HIGH);
}
if(estado== 'p' ){
    digitalWrite(6,LOW);
}

```

```

pinMode(13,OUTPUT);
pinMode(12,OUTPUT);
pinMode(11,OUTPUT);
pinMode(10,OUTPUT);
pinMode(9,OUTPUT);
pinMode(8,OUTPUT);
pinMode(7,OUTPUT);
pinMode(6,OUTPUT);
pinMode(5,OUTPUT);
pinMode(4,OUTPUT);
pinMode(3,OUTPUT);
Serial.begin(9600);
delay(1000);
}

```

```

if(Serial_2.available(>0){ //Nuevo puerto para la conexion Bluetooth
// Si el puerto serie esta habilitado
    estado = Serial_2.read(); // Lee lo que llega por el puerto Serie
}

```

```

if(estado== 'a'){ // on/off de los pin 13 al pin 6
    digitalWrite(13,HIGH);
}
if(estado== 'b' ){
    digitalWrite(13,LOW);
}
if(estado== 'c'){
    digitalWrite(12,HIGH);
}
if(estado== 'd' ){
    digitalWrite(12,LOW);
}
if(estado== 'e'){
    digitalWrite(11,HIGH);
}

```

```
}
if(estado== 'f' ){
    digitalWrite(11,LOW);
}
if(estado== 'g' ){
    digitalWrite(10,HIGH);
}
if(estado== 'h' ){
    digitalWrite(10,LOW);
}
if(estado== 'i' ){
    digitalWrite(9,HIGH);
}
if(estado== 'j' ){
    digitalWrite(9,LOW);
}
if(estado== 'k' ){
    digitalWrite(8,HIGH);
}
if(estado== 'l' ){
    digitalWrite(8,LOW);
}
if(estado== 'm' ){
    digitalWrite(7,HIGH);
}
if(estado== 'n' ){
    digitalWrite(7,LOW);
}
if(estado== 'o' ){
    digitalWrite(6,HIGH);
}
if(estado== 'p' ){
    digitalWrite(6,LOW);
}

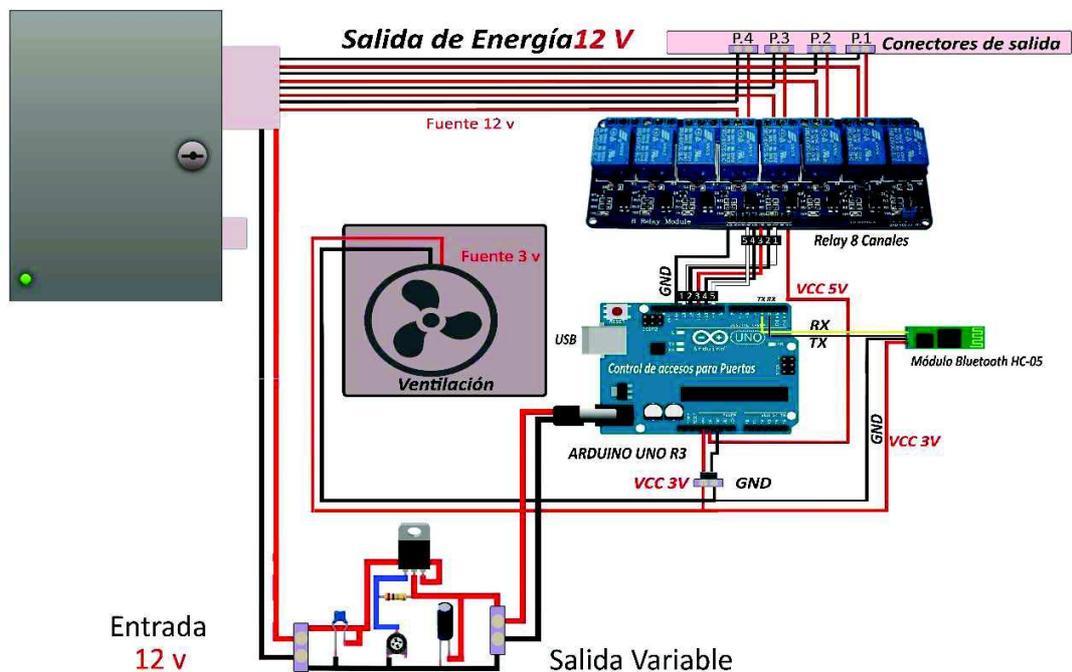
}

delay(retardo);
}
```

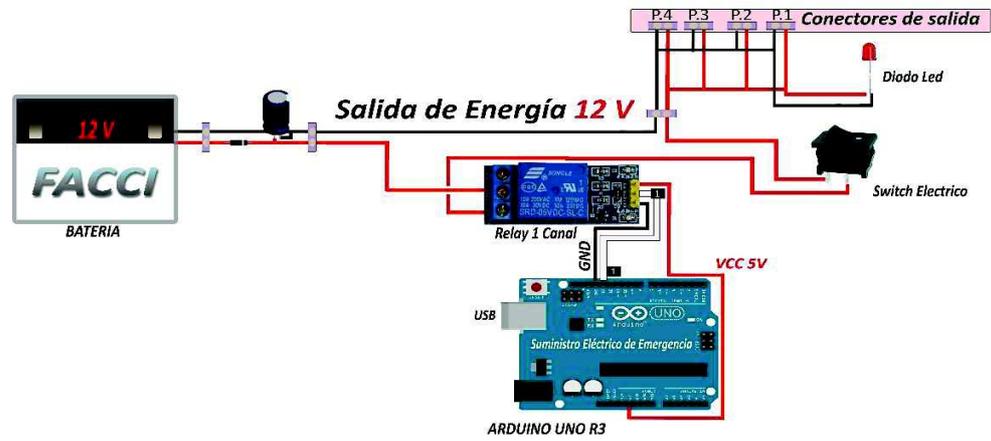
ENSAMBLADO

Para la fabricación de la estructura se utilizó madera y se estructuro de tal manera que pueda albergar todos los componentes a continuación se mostrara los procesos de fabricación y ensamblado.

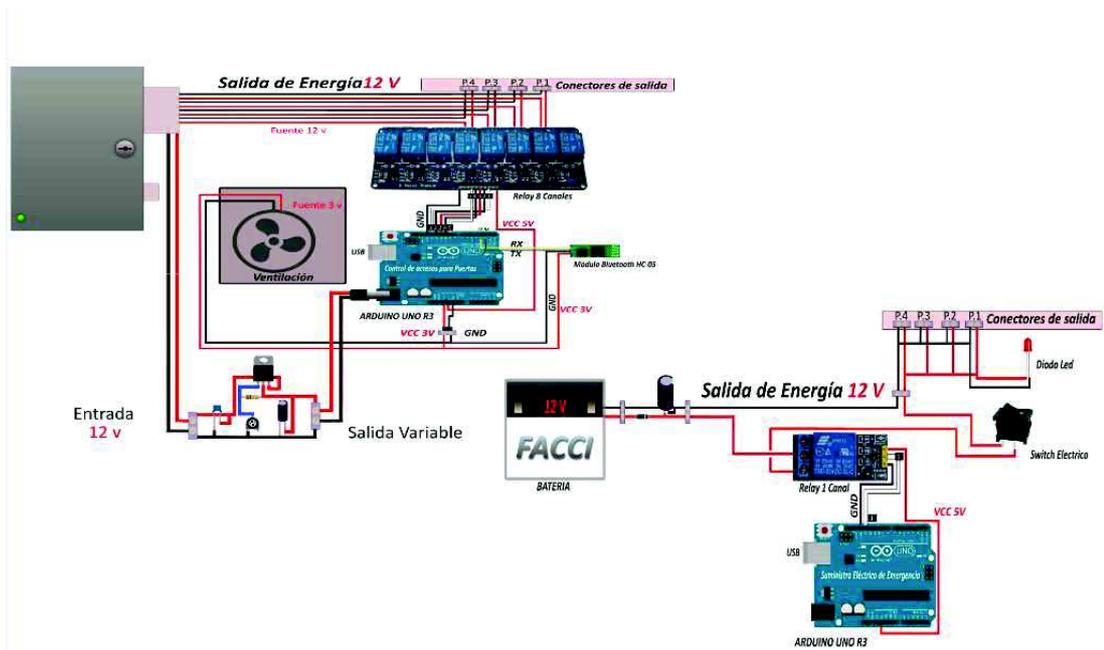
Esquema del Circuito Principal



Esquema del Circuito Respaldo de Energía



Esquema Final



ESTRUCTURA DEL SISTEMA

INDICE

Formulario Usuario	3
Formulario Principal	3
Formulario Formulario	4
.....	4
Formulario Horario.....	4
Clase Base.....	5
Clase principal Arduino Switch	5
Clase Control	5
Clase Controls.....	25
Clase Formulario	26
Clase Horario	32
Clase Usuario.....	78

Formulario Usuario



Formulario Principal



Formulario Formulario



Formulario Horario

HORAS	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO
07:00 - 08:00	Red	White	Red	White	White	White
08:00 - 09:00	White	White	White	Red	Red	Red
09:00 - 10:00	Red	Red	Red	White	White	White
10:00 - 11:00	White	White	Red	White	White	White
11:00 - 12:00	White	White	White	Red	Red	Red
12:00 - 13:00	White	Red	White	White	White	White
13:00 - 14:00	White	White	Red	White	Red	White
14:00 - 15:00	White	Red	White	White	White	Red
15:00 - 16:00	White	White	White	Red	White	White
16:00 - 17:00	White	Red	White	Red	Red	Red
17:00 - 18:00	White	White	White	White	White	White
18:00 - 19:00	White	White	White	White	White	White
19:00 - 20:00	White	White	White	White	White	White
20:00 - 21:00	White	White	White	White	White	White

LABORATORIO 201 LABORATORIO 202 LABORATORIO 203 LABORATORIO 206

Clase Base

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Base;
import java.sql.Connection;
import java.sql.DriverManager;
import javax.swing.JOptionPane;
/**
 *
 * Base principal
 */
public class Base {
    Connection conect=null;
    public Connection conexion(){
        try {
            Class.forName("org.sqlite.JDBC");
            conect=DriverManager.getConnection("jdbc:sqlite:sistemapuerta.db");
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null,"ERROR AL CONECTAR LA BASE DE DATOS"+e);
        }
        return conect;
    }
}
```

Clase principal Arduino Switch

```
package Switch;
public class ArduinoSwitch {
    public static void main(String[] args) {
        Usuario Ventana = new Usuario();
        Ventana.setVisible(true);
    }
}
```

Clase Control

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Switch;

import Base.Base;
```

```

import java.awt. Image;
import java.io.*;
import static java.lang.Thread.*;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Timer;
import java.util.TimerTask;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.Icon;
import javax.swing.ImageIcon;
import panamahitek.Arduino.PanamaHitek_Arduino;

/**
 *
 * @author stev-
 */
public final class Control extends javax.swing.JFrame {
    int banderaconecion=0;
    int banderamanual=0;
    int velocidad=30;
    String[][] matriz201 = new String[6][24];
    String[][] matriz202 = new String[6][24];
    String[][] matriz203 = new String[6][24];
    String[][] matriz206 = new String[6][24];
    String compare="0";
    private boolean encendido;
    private final SimpleDateFormat formato;
    ResultSet RS;
    Base CC = new Base();
    Connection CN = CC.conexion();
    PanamaHitek_Arduino Arduino = new PanamaHitek_Arduino();
    Timer timer =new Timer();

/**
 * Creates new form Control
 */
public Control() {
    initComponents();
    encendido = true;
    formato = new SimpleDateFormat("HH:mm:ss");
    this.leer201();

```

```

        this.leer202();
        this.leer203();
        this.leer206();
        this.dia();

    }
    void arduino(){
        String CAP="";
        String sql="SELECT * FROM puerto WHERE COD=1";

        try {
            Statement st = CN.createStatement();
            ResultSet RS = st.executeQuery(sql);
            while(RS.next())
            {
                CAP=RS.getString("COD_PUERTO");
            }
            Arduino.arduinoTX(CAP, 9600);
        } catch (Exception ex) {
            Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    public void cerrado(){
        cerrado201();
        cerrado202();
        cerrado203();
        cerrado206();
    }

    public void cerrado201(){
        ImagenIcon imagen2= new ImagenIcon("C:\\SistemaPuerta\\cerradot.jpg");
        Icon icono1=new ImagenIcon(imagen2.getImage().getScaledInstance(lbl201.getWidth(),
        lbl201.getHeight(), Image.SCALE_DEFAULT));
        lbl201.setIcon(icono1);
        this.repaint();
    }
    try {Arduino.sendData("b");} catch (Exception ex)
    {Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);}
    }
    public void cerrado202(){
        ImagenIcon imagen2= new ImagenIcon("C:\\SistemaPuerta\\cerradot.jpg");
        Icon icono1=new ImagenIcon(imagen2.getImage().getScaledInstance(lbl202.getWidth(),
        lbl202.getHeight(), Image.SCALE_DEFAULT));
        lbl202.setIcon(icono1);
        this.repaint();
    }

```

```

try {Arduino.sendData("d");} catch (Exception ex)
{Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);}
}

public void cerrado203(){
    ImagenIcon imagen2= new ImagenIcon("C:\\SistemaPuerta\\cerradot.jpg");
    Icon icono1=new ImagenIcon(imagen2.getImage().getScaledInstance(lbl203.getWidth(),
lbl203.getHeight(), Image.SCALE_DEFAULT));
    lbl203.setIcon(icono1);
    this.repaint();
try {Arduino.sendData("f");} catch (Exception ex)
{Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);}
}

public void cerrado206(){
    ImagenIcon imagen2= new ImagenIcon("C:\\SistemaPuerta\\cerradot.jpg");
    Icon icono1=new ImagenIcon(imagen2.getImage().getScaledInstance(lbl206.getWidth(),
lbl206.getHeight(), Image.SCALE_DEFAULT));
    lbl206.setIcon(icono1);
    this.repaint();
try {Arduino.sendData("h");} catch (Exception ex)
{Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);}
}

public void abierto201(){
    ImagenIcon imagen2= new ImagenIcon("C:\\SistemaPuerta\\abiertot.jpg");
    Icon icono1=new ImagenIcon(imagen2.getImage().getScaledInstance(lbl201.getWidth(),
lbl201.getHeight(), Image.SCALE_DEFAULT));
    lbl201.setIcon(icono1);
    this.repaint();
    try {Arduino.sendData("a");} catch (Exception ex)
{Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);}
}

public void abierto202(){
    ImagenIcon imagen2= new ImagenIcon("C:\\SistemaPuerta\\abiertot.jpg");
    Icon icono1=new ImagenIcon(imagen2.getImage().getScaledInstance(lbl202.getWidth(),
lbl202.getHeight(), Image.SCALE_DEFAULT));
    lbl202.setIcon(icono1);
    this.repaint();
try {Arduino.sendData("c");} catch (Exception ex)
{Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);}
}

public void abierto203(){
    ImagenIcon imagen2= new ImagenIcon("C:\\SistemaPuerta\\abiertot.jpg");

```

```

        Icon icono1=new ImagenIcon(imagen2.getImage().getScaledInstance(lbl203.getWidth(),
lbl203.getHeight(), Image.SCALE_DEFAULT));
        lbl203.setIcon(icono1);
        this.repaint();
try {Arduino.sendData("e");} catch (Exception ex)
{Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);}
}
public void abierto206(){
    ImagenIcon imagen2= new ImagenIcon("C:\\SistemaPuerta\\abierto.jpg");
    Icon icono1=new ImagenIcon(imagen2.getImage().getScaledInstance(lbl206.getWidth(),
lbl206.getHeight(), Image.SCALE_DEFAULT));
    lbl206.setIcon(icono1);
    this.repaint();
try {Arduino.sendData("g");} catch (Exception ex)
{Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);}
}
void iniciar(){
    int contador=0;

    Timer timer;
    TimerTask tarea;

    int velmil=velocidad*1000;

    tarea=new TimerTask() {
        @Override
        public void run() {
            switch (contador) {
                case 0:
                    if (banderaconexion==0){
                        horario();
                    }else{
                        velocidad=30;
                        if(banderaconexion==1){
                            arduino();
                            banderaconexion=0;
                            horario();
                        }
                    }
                }

            break;
            default:
                throw new AssertionError();

```

```
    }  
  }  
};
```

```
    this.timer.scheduleAtFixedRate(tarea, velmil, velmil);
```

```
}
```

```
public void leer201(){
```

```
    File archivo = null;  
    FileReader Fr = null;  
    BufferedReader br = null;  
    try {  
        archivo = new File("C:\\SistemaPuerta\\LAB201.txt");  
        Fr = new FileReader(archivo.toString());  
        br = new BufferedReader(Fr);  
        String linea;  
        String delimiter = " ";  
        int numlinea=0;  
        while (((linea = br.readLine()) != null)) {  
            String a[]=linea.split(delimiter);  
            for (int l = 0; l < a.length; l++) {  
                matriz201[numlinea][l] = a[l];  
            }  
            numlinea++;  
        }  
    } catch (IOException e) {  
    }  
}
```

```
public void leer202(){
```

```
    File archivo = null;  
    FileReader Fr = null;  
    BufferedReader br = null;  
    try {  
        archivo = new File("C:\\SistemaPuerta\\LAB202.txt");  
        Fr = new FileReader(archivo.toString());  
        br = new BufferedReader(Fr);  
        String linea;  
        String delimiter = " ";  
        int numlinea=0;
```

```

while (((linea = br.readLine()) != null)) {
    String a[]=linea.split(delimiter);
    for (int l = 0; l < a.length; l++) {
        matriz202[numlinea][l] = a[l];
    }
    numlinea++;
}
} catch (IOException e) {
}
}
public void leer203(){

File archivo = null;
FileReader Fr = null;
BufferedReader br = null;
try {
    archivo = new File("C:\\SistemaPuerta\\LAB203.txt");
    Fr = new FileReader(archivo.toString());
    br = new BufferedReader(Fr);
    String linea;
    String delimiter = " ";
    int numlinea=0;
    while (((linea = br.readLine()) != null)) {
        String a[]=linea.split(delimiter);
        for (int l = 0; l < a.length; l++) {
            matriz203[numlinea][l] = a[l];
        }
        numlinea++;
    }
} catch (IOException e) {
}
}
public void leer206(){

File archivo = null;
FileReader Fr = null;
BufferedReader br = null;
try {
    archivo = new File("C:\\SistemaPuerta\\LAB206.txt");
    Fr = new FileReader(archivo.toString());
    br = new BufferedReader(Fr);
    String linea;
    String delimiter = " ";
    int numlinea=0;
    while (((linea = br.readLine()) != null)) {

```

```

        String a[]=linea.split(delimiter);
        for (int l = 0; l < a.length; l++) {
            matriz206[numlinea][l] = a[l];
        }
        numlinea++;
    }
} catch (IOException e) {
}
}

public void dia(){
    Calendar now = Calendar.getInstance();
    String[] strDays = new String[]{
        "Domingo", "Lunes", "Martes", "Miercoles", "Jueves", "Viernes", "Sabado"};
    lblDia.setText(strDays[now.get(Calendar.DAY_OF_WEEK) - 1]);

}

public void horario(){

do {

String dia;
dia=lblDia.getText();
Calendar calendario = Calendar.getInstance();
int hora;
hora =calendario.get(Calendar.HOUR_OF_DAY);
if (hora<7 || hora>21){
    if(banderamanual==1){

    }else{
        cerrado();
    }

    try{
sleep(30000);
encendido=false;

} catch (InterruptedException ex) {
}

} else{

```

```

if ("Lunes".equals(dia)) {
    int tiempo2=calendario.get(Calendar.HOUR_OF_DAY);
    String L201=matriz201[0][calendario.get(Calendar.HOUR_OF_DAY)];
    String L202=matriz202[0][calendario.get(Calendar.HOUR_OF_DAY)];
    String L203=matriz203[0][calendario.get(Calendar.HOUR_OF_DAY)];
    String L206=matriz206[0][calendario.get(Calendar.HOUR_OF_DAY)];

//-----201
    int horario201=0;
    horario201=Integer.parseInt(L201);
    if (tiempo2==horario201) {
        abierto201();
    }else{
        cerrado201();
    }
}

//-----202
    int horario202=0;
    horario202=Integer.parseInt(L202);
    if (tiempo2==horario202) {
        abierto202();
    }else{
        cerrado202();
    }
}

//-----203
    int horario203=0;
    horario203=Integer.parseInt(L203);
    if (tiempo2==horario203) {
        abierto203();
    }else{
        cerrado203();
    }
}

//-----206
    int horario206=0;
    horario206=Integer.parseInt(L206);
    if (tiempo2==horario206) {
        abierto206();
    }else{
        cerrado206();
    }
}

try{
    sleep(2000);
    encendido=false;
}

```

```

} catch(InterruptedException ex) {
}
}

    if ("Martes".equals(dia)) {
    int tiempo2=calendario.get(Calendar.HOUR_OF_DAY);
    String L201=matriz201[1][calendario.get(Calendar.HOUR_OF_DAY)];
    String L202=matriz202[1][calendario.get(Calendar.HOUR_OF_DAY)];
    String L203=matriz203[1][calendario.get(Calendar.HOUR_OF_DAY)];
    String L206=matriz206[1][calendario.get(Calendar.HOUR_OF_DAY)];

//-----201
    int horario201=0;
    horario201=Integer.parseInt(L201);
    if (tiempo2==horario201) {
        abierto201();
    }else{
        cerrado201();
    }

//-----202
    int horario202=0;
    horario202=Integer.parseInt(L202);
    if (tiempo2==horario202) {
        abierto202();
    }else{
        cerrado202();
    }

//-----203
    int horario203=0;
    horario203=Integer.parseInt(L203);
    if (tiempo2==horario203) {
        abierto203();
    }else{
        cerrado203();
    }

//-----206
    int horario206=0;
    horario206=Integer.parseInt(L206);
    if (tiempo2==horario206) {
        abierto206();
    }else{

```

```

        cerrado206();
    }

    try{
        sleep(2000);
        encendido=false;

    } catch(InterruptedException ex) {
    }

    }
    if ("Miercoles".equals(dia)) {
        int tiempo2=calendario.get(Calendar.HOUR_OF_DAY);
        String L201=matriz201[2][calendario.get(Calendar.HOUR_OF_DAY)];
        String L202=matriz202[2][calendario.get(Calendar.HOUR_OF_DAY)];
        String L203=matriz203[2][calendario.get(Calendar.HOUR_OF_DAY)];
        String L206=matriz206[2][calendario.get(Calendar.HOUR_OF_DAY)];

//-----201
        int horario201=0;
        horario201=Integer.parseInt(L201);
        if (tiempo2==horario201) {
            abierto201();
        }else{
            cerrado201();
        }

//-----202
        int horario202=0;
        horario202=Integer.parseInt(L202);
        if (tiempo2==horario202) {
            abierto202();
        }else{
            cerrado202();
        }

//-----203
        int horario203=0;
        horario203=Integer.parseInt(L203);
        if (tiempo2==horario203) {
            abierto203();
        }else{
            cerrado203();
        }
    }

```

```

//-----206
int horario206=0;
horario206=Integer.parseInt(L206);
if (tiempo2==horario206) {
    abierto206();
} else {
    cerrado206();
}

try{
sleep(2000);
encendido=false;

} catch (InterruptedException ex) {
}

}
if ("Jueves".equals(dia)) {
int tiempo2=calendario.get(Calendar.HOUR_OF_DAY);
String L201=matriz201[3][calendario.get(Calendar.HOUR_OF_DAY)];
String L202=matriz202[3][calendario.get(Calendar.HOUR_OF_DAY)];
String L203=matriz203[3][calendario.get(Calendar.HOUR_OF_DAY)];
String L206=matriz206[3][calendario.get(Calendar.HOUR_OF_DAY)];

//-----201
int horario201=0;
horario201=Integer.parseInt(L201);
if (tiempo2==horario201) {
    abierto201();
} else {
    cerrado201();
}

//-----202
int horario202=0;
horario202=Integer.parseInt(L202);
if (tiempo2==horario202) {
    abierto202();
} else {
    cerrado202();
}

//-----203
int horario203=0;
horario203=Integer.parseInt(L203);
if (tiempo2==horario203) {

```

```

    abierto203();
    }else{
    cerrado203();
}

//-----206
int horario206=0;
horario206=Integer.parseInt(L206);
if (tiempo2==horario206) {
    abierto206();
    }else{
    cerrado206();
}

try{
sleep(2000);
encendido=false;

} catch(InterruptedException ex) {
}

}
if ("Viernes".equals(dia)) {
int tiempo2=calendario.get(Calendar.HOUR_OF_DAY);
String L201=matriz201[4][calendario.get(Calendar.HOUR_OF_DAY)];
String L202=matriz202[4][calendario.get(Calendar.HOUR_OF_DAY)];
String L203=matriz203[4][calendario.get(Calendar.HOUR_OF_DAY)];
String L206=matriz206[4][calendario.get(Calendar.HOUR_OF_DAY)];

//-----201
int horario201=0;
horario201=Integer.parseInt(L201);
if (tiempo2==horario201) {
    abierto201();
    }else{
    cerrado201();
}

//-----202
int horario202=0;
horario202=Integer.parseInt(L202);
if (tiempo2==horario202) {
    abierto202();
    }else{
    cerrado202();
}

```

```

}
//-----203
int horario203=0;
horario203=Integer.parseInt(L203);
if (tiempo2==horario203) {
    abierto203();
} else {
    cerrado203();
}

//-----206
int horario206=0;
horario206=Integer.parseInt(L206);
if (tiempo2==horario206) {
    abierto206();
} else {
    cerrado206();
}

try{
sleep(2000);
encendido=false;

} catch(InterruptedException ex) {
}

}
if ("Sabado".equals(dia)) {
int tiempo2=calendario.get(Calendar.HOUR_OF_DAY);
String L201=matriz201[5][calendario.get(Calendar.HOUR_OF_DAY)];
String L202=matriz202[5][calendario.get(Calendar.HOUR_OF_DAY)];
String L203=matriz203[5][calendario.get(Calendar.HOUR_OF_DAY)];
String L206=matriz206[5][calendario.get(Calendar.HOUR_OF_DAY)];

//-----201
int horario201=0;
horario201=Integer.parseInt(L201);
if (tiempo2==horario201) {
    abierto201();
} else {
    cerrado201();
}

//-----202
int horario202=0;

```

```

    horario202=Integer.parseInt(L202);
    if (tiempo2==horario202) {
        abierto202();
    }else{
        cerrado202();
    }
//-----203
    int horario203=0;
    horario203=Integer.parseInt(L203);
    if (tiempo2==horario203) {
        abierto203();
    }else{
        cerrado203();
    }

//-----206
    int horario206=0;
    horario206=Integer.parseInt(L206);
    if (tiempo2==horario206) {
        abierto206();
    }else{
        cerrado206();
    }

try{
    sleep(2000);
    encendido=false;

} catch(InterruptedException ex) {
}
}
}
}while(encendido);}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
private void initComponents() {

```

```

jLabel1 = new javax.swing.JLabel();
btnCelular = new javax.swing.JButton();
btnhorario = new javax.swing.JButton();
jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
lblDia = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
jLabel7 = new javax.swing.JLabel();
jLabel8 = new javax.swing.JLabel();
lbl201 = new javax.swing.JLabel();
lbl202 = new javax.swing.JLabel();
lbl203 = new javax.swing.JLabel();
lbl206 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setBackground(new java.awt.Color(0, 102, 102));
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Switch/logo.png")));
// NOI18N
getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(190,
20, 370, 90));

btnCelular.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Switch/mobile_phone_14388_opt.png"))); //
NOI18N
btnCelular.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
btnCelular.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
btnCelular.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCelularActionPerformed(evt);
    }
});
getContentPane().add(btnCelular, new
org.netbeans.lib.awtextra.AbsoluteConstraints(110, 40, 68, -1));

btnhorario.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Switch/Web-BV-
Horario_opt.png"))); // NOI18N
btnhorario.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
btnhorario.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));

```

```

btnhorario.setMaximumSize(new java.awt.Dimension(57, 57));
btnhorario.setMinimumSize(new java.awt.Dimension(57, 57));
btnhorario.setPreferredSize(new java.awt.Dimension(57, 57));
btnhorario.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnhorarioActionPerformed(evt);
    }
});
getContentPane().add(btnhorario, new
org.netbeans.lib.awtextra.AbsoluteConstraints(570, 40, 70, 60));

jLabel2.setBackground(new java.awt.Color(255, 255, 255));
jLabel2.setFont(new java.awt.Font("Bodoni MT Black", 1, 24)); // NOI18N
jLabel2.setForeground(new java.awt.Color(255, 255, 255));
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("CONTROL AUTOMÁTICO DE PUERTAS");
getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 130,
710, -1));

jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
jLabel3.setForeground(new java.awt.Color(255, 255, 255));
jLabel3.setText("DIA:");
getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(50,
360, -1, -1));

lblDia.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
lblDia.setForeground(new java.awt.Color(255, 255, 255));
lblDia.setText(" ");
getContentPane().add(lblDia, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 360,
90, -1));

jLabel5.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
jLabel5.setForeground(new java.awt.Color(255, 255, 255));
jLabel5.setText("LABORATORIO 206");
getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(540,
300, -1, -1));

jLabel6.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
jLabel6.setForeground(new java.awt.Color(255, 255, 255));
jLabel6.setText("LABORATORIO 201");
getContentPane().add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
300, -1, -1));

jLabel7.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
jLabel7.setForeground(new java.awt.Color(255, 255, 255));

```

```

jLabel7.setText("LABORATORIO 202");
getContentPane().add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(220,
300, -1, -1));

jLabel8.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
jLabel8.setForeground(new java.awt.Color(255, 255, 255));
jLabel8.setText("LABORATORIO 203");
getContentPane().add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(380,
300, -1, -1));

lbl201.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Switch/cerradot.jpg"))); // NOI18N
lbl201.setText("jLabel9");
lbl201.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
getContentPane().add(lbl201, new org.netbeans.lib.awtextra.AbsoluteConstraints(60, 180,
90, 110));

lbl202.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Switch/cerradot.jpg"))); // NOI18N
lbl202.setText("jLabel9");
lbl202.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
getContentPane().add(lbl202, new org.netbeans.lib.awtextra.AbsoluteConstraints(220,
180, 90, 110));

lbl203.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Switch/cerradot.jpg"))); // NOI18N
lbl203.setText("jLabel9");
lbl203.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
getContentPane().add(lbl203, new org.netbeans.lib.awtextra.AbsoluteConstraints(380,
180, 90, 110));

lbl206.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Switch/cerradot.jpg"))); // NOI18N
lbl206.setText("jLabel9");
lbl206.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
getContentPane().add(lbl206, new org.netbeans.lib.awtextra.AbsoluteConstraints(540,
180, 90, 110));

jLabel4.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Switch/19832806_1531862570167961_23954
0534_n.jpg"))); // NOI18N

```

```

jLabel4.setText("jLabel4");
getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0,
710, 440));

```

```

pack();
} // </editor-fold> // GEN-END: initComponents

```

```

private void btnCelularActionPerformed(java.awt.event.ActionEvent evt) // GEN-
FIRST:event_btnCelularActionPerformed

```

```

try {
    Arduino.killArduinoConnection();
    banderaconecion=0;
    banderamanual=1;
    timer.cancel();
} catch (Exception ex) {
    Logger.getLogger(Usuario.class.getName()).log(Level.SEVERE, null, ex);
}
Formulario jframe=new Formulario();
jframe.setVisible(true);
this.setVisible(false);

```

```

// TODO add your handling code here:

```

```

} // GEN-LAST:event_btnCelularActionPerformed

```

```

private void btnhorarioActionPerformed(java.awt.event.ActionEvent evt) // GEN-
FIRST:event_btnhorarioActionPerformed

```

```

Horario jframe=new Horario();
jframe.setVisible(true);
// TODO add your handling code here:
} // GEN-LAST:event_btnhorarioActionPerformed

```

```

/**

```

```

 * @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(Control.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {

```

```

java.util.logging.Logger.getLogger(Control.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {

```

```

java.util.logging.Logger.getLogger(Control.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

```

```

java.util.logging.Logger.getLogger(Control.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
}
//</editor-fold>

```

```

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    @Override
    public void run() {
        new Control().setVisible(true);
    }
});
}

```

```

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton btnCelular;
private javax.swing.JButton btnhorario;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
public static javax.swing.JLabel lbl201;

```

```
public static javax.swing.JLabel lbl202;
public static javax.swing.JLabel lbl203;
public static javax.swing.JLabel lbl206;
private javax.swing.JLabel lblDia;
// End of variables declaration//GEN-END:variables
}
```

Clase Controls

```
package Switch;
```

```
import java.awt.Color;
import java.io.IOException;
import java.net.URISyntaxException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JButton;
import javax.swing.JComboBox;
```

```
/**
 *
 */
```

```
public class Controls {
```

```
    public void enableButton(JButton boton) {
        boton.setEnabled(true);
        boton.setBackground(new Color(255, 204, 51));
```

```
    }
```

```
    public void disableButton(JButton boton) {
        boton.setEnabled(false);
        boton.setBackground(new Color(204, 204, 204));
```

```
    }
```

```
    public void disableConnectionPanel(JButton boton, JComboBox combo) {
        combo.setEnabled(false);
        boton.setEnabled(false);
```

```
    }
```

```
    public void enableConnectionPanel(JButton boton, JComboBox combo) {
        combo.setEnabled(true);
        boton.setEnabled(true);
```

```
}
```

```
}
```

Clase Formulario

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package Switch;  
  
import Base.Base;  
import java.awt.Color;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.awt.event.MouseAdapter;  
import java.awt.event.MouseEvent;  
import java.sql.Connection;  
import java.sql.ResultSet;  
import java.sql.Statement;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import panamahitek.Arduino.PanamaHitek_Arduino;  
  
/**  
 *  
 * @author stev-  
 */  
public class Formulario extends javax.swing.JFrame {  
    ResultSet RS;  
    Base CC = new Base();  
    Connection CN = CC.conexion();  
    PanamaHitek_Arduino Arduino = new PanamaHitek_Arduino();  
    /**  
     * Creates new form Formulario  
     */  
    public Formulario() {  
        this.setUndecorated(true);  
        initComponents();  
  
        String CAP="";
```



```

    }
});
getContentPane().add(jToggleButton1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 120, 200, -1));

jToggleButton2.setBackground(new java.awt.Color(255, 0, 0));
jToggleButton2.setText("LABORATORIO 202");
jToggleButton2.setMargin(new java.awt.Insets(10, 14, 10, 14));
jToggleButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jToggleButton2ActionPerformed(evt);
    }
});
getContentPane().add(jToggleButton2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 180, 200, -1));

jToggleButton3.setBackground(new java.awt.Color(255, 0, 0));
jToggleButton3.setText("LABORATORIO 203");
jToggleButton3.setMargin(new java.awt.Insets(10, 14, 10, 14));
jToggleButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jToggleButton3ActionPerformed(evt);
    }
});
getContentPane().add(jToggleButton3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 240, 200, -1));

jToggleButton4.setBackground(new java.awt.Color(255, 0, 0));
jToggleButton4.setText("LABORATORIO 206");
jToggleButton4.setMargin(new java.awt.Insets(10, 14, 10, 14));
jToggleButton4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jToggleButton4ActionPerformed(evt);
    }
});
getContentPane().add(jToggleButton4, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 300, 200, -1));

btncerrar.setText("X");
btncerrar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btncerrarActionPerformed(evt);
    }
});

```

```

        getContentPane().add(btncerrar, new org.netbeans.lib.awtextra.AbsoluteConstraints(180,
360, -1, -1));

        jLabel3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Switch/logo.png")));
// NOI18N
        jLabel3.setToolTipText("");
        jLabel3.setBorder(javax.swing.BorderFactory.createEtchedBorder());
        getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 20,
360, 90));

        jLabel4.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Switch/19668219_1531438200210398_13829
47556_n.jpg"))); // NOI18N
        getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0,
380, 400));

        pack();
} // </editor-fold> // GEN-END: initComponents

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) // GEN-
FIRST:event_jButton1ActionPerformed
    if (jToggleButton1.isSelected()){ try {
        jButton1.setBackground(Color.yellow);
        Arduino.sendData("a");

    } catch (Exception ex) {
        Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);
    } } else{
        try {
            jButton1.setBackground(Color.red);
            Arduino.sendData("b");

        } catch (Exception ex) {
            Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

// TODO add your handling code here:
} // GEN-LAST:event_jButton1ActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) // GEN-
FIRST:event_jButton2ActionPerformed
    if (jToggleButton2.isSelected()){ try {
        jButton2.setBackground(Color.yellow);
        Arduino.sendData("c");

```

```

} catch (Exception ex) {
    Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);

    } else{
    try {
        jToggleButton2.setBackground(Color.red);
        Arduino.sendData("d");
    } catch (Exception ex) {
        Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);

    }
} // TODO add your handling code here:
} //GEN-LAST:event_jToggleButton2ActionPerformed

private void jToggleButton3ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_jToggleButton3ActionPerformed
if (jToggleButton3.isSelected()){ try {
    jToggleButton3.setBackground(Color.yellow);
    Arduino.sendData("e");
} catch (Exception ex) {
    Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);

} } else{
    try {
        jToggleButton3.setBackground(Color.red);
        Arduino.sendData("f");
    } catch (Exception ex) {
        Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);
    }
} // TODO add your handling code here:
} //GEN-LAST:event_jToggleButton3ActionPerformed

private void jToggleButton4ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_jToggleButton4ActionPerformed
if (jToggleButton4.isSelected()){ try {
    jToggleButton4.setBackground(Color.yellow);
    Arduino.sendData("g");
} catch (Exception ex) {
    Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);

} } else{

```

```

        try {
            jToggleButton4.setBackground(Color.red);
            Arduino.sendData("h");
        } catch (Exception ex) {
            Logger.getLogger(Formulario.class.getName()).log(Level.SEVERE, null, ex);
        }
    } // TODO add your handling code here:
} //GEN-LAST:event_jToggleButton4ActionPerformed

private void btncerrarActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_btncerrarActionPerformed
try {
    Arduino.killArduinoConnection();
    } catch (Exception ex) {
        Logger.getLogger(Usuario.class.getName()).log(Level.SEVERE, null, ex);
    }
    Control control=new Control();
    control.banderaconeccion=1;
    control.banderamanual=0;
    control.velocidad=1;
    control.iniciar();
    control.setVisible(true);
    dispose();// TODO add your handling code here:

} //GEN-LAST:event_btncerrarActionPerformed

/**
 * @param args the command line arguments
 */

// Variables declaration - do not modify //GEN-BEGIN:variables
private javax.swing.JButton btncerrar;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JToggleButton jToggleButton1;
private javax.swing.JToggleButton jToggleButton2;
private javax.swing.JToggleButton jToggleButton3;
private javax.swing.JToggleButton jToggleButton4;
// End of variables declaration //GEN-END:variables
}

```

Clase Horario

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Switch;

import java.awt.Color;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import javax.swing.JOptionPane;

/**
 *
 * @author stev-
 */
public class Horario extends javax.swing.JFrame {
    String H[][]= new String[6][24];

    /**
     * Creates new form Horario
     */
    public Horario() {
        initComponents();
        for (int i = 0; i < 6; i++) {
            for (int j = 0; j < 24; j++) {
                H[i][j]=String.valueOf(0);
            }
        }
    }
    void blanco(){
        L1.setSelected(false);
        L1.setBackground(Color.WHITE);
        L2.setSelected(false);
        L2.setBackground(Color.WHITE);
        L3.setSelected(false);
        L3.setBackground(Color.WHITE);
        L4.setSelected(false);
        L4.setBackground(Color.WHITE);
        L5.setSelected(false);
        L5.setBackground(Color.WHITE);
        L6.setSelected(false);
        L6.setBackground(Color.WHITE);
        L7.setSelected(false);
        L7.setBackground(Color.WHITE);
        L8.setSelected(false);
        L8.setBackground(Color.WHITE);
    }
}
```

```
L9.setSelected(false);
L9.setBackground(Color.WHITE);
L10.setSelected(false);
L10.setBackground(Color.WHITE);
L11.setSelected(false);
L11.setBackground(Color.WHITE);
L12.setSelected(false);
L12.setBackground(Color.WHITE);
L13.setSelected(false);
L13.setBackground(Color.WHITE);
L14.setSelected(false);
L14.setBackground(Color.WHITE);
```

```
M1.setSelected(false);
M1.setBackground(Color.WHITE);
M2.setSelected(false);
M2.setBackground(Color.WHITE);
M3.setSelected(false);
M3.setBackground(Color.WHITE);
M4.setSelected(false);
M4.setBackground(Color.WHITE);
M5.setSelected(false);
M5.setBackground(Color.WHITE);
M6.setSelected(false);
M6.setBackground(Color.WHITE);
M7.setSelected(false);
M7.setBackground(Color.WHITE);
M8.setSelected(false);
M8.setBackground(Color.WHITE);
M9.setSelected(false);
M9.setBackground(Color.WHITE);
M10.setSelected(false);
M10.setBackground(Color.WHITE);
M11.setSelected(false);
M11.setBackground(Color.WHITE);
M12.setSelected(false);
M12.setBackground(Color.WHITE);
M13.setSelected(false);
M13.setBackground(Color.WHITE);
M14.setSelected(false);
M14.setBackground(Color.WHITE);
```

```
X1.setSelected(false);
X1.setBackground(Color.WHITE);
X2.setSelected(false);
X2.setBackground(Color.WHITE);
X3.setSelected(false);
X3.setBackground(Color.WHITE);
X4.setSelected(false);
X4.setBackground(Color.WHITE);
X5.setSelected(false);
X5.setBackground(Color.WHITE);
```

```
X6.setSelected(false);
X6.setBackground(Color.WHITE);
X7.setSelected(false);
X7.setBackground(Color.WHITE);
X8.setSelected(false);
X8.setBackground(Color.WHITE);
X9.setSelected(false);
X9.setBackground(Color.WHITE);
X10.setSelected(false);
X10.setBackground(Color.WHITE);
X11.setSelected(false);
X11.setBackground(Color.WHITE);
X12.setSelected(false);
X12.setBackground(Color.WHITE);
X13.setSelected(false);
X13.setBackground(Color.WHITE);
X14.setSelected(false);
X14.setBackground(Color.WHITE);
```

```
J1.setSelected(false);
J1.setBackground(Color.WHITE);
J2.setSelected(false);
J2.setBackground(Color.WHITE);
J3.setSelected(false);
J3.setBackground(Color.WHITE);
J4.setSelected(false);
J4.setBackground(Color.WHITE);
J5.setSelected(false);
J5.setBackground(Color.WHITE);
J6.setSelected(false);
J6.setBackground(Color.WHITE);
J7.setSelected(false);
J7.setBackground(Color.WHITE);
J8.setSelected(false);
J8.setBackground(Color.WHITE);
J9.setSelected(false);
J9.setBackground(Color.WHITE);
J10.setSelected(false);
J10.setBackground(Color.WHITE);
J11.setSelected(false);
J11.setBackground(Color.WHITE);
J12.setSelected(false);
J12.setBackground(Color.WHITE);
J13.setSelected(false);
J13.setBackground(Color.WHITE);
J14.setSelected(false);
J14.setBackground(Color.WHITE);
```

```
V1.setSelected(false);
V1.setBackground(Color.WHITE);
V2.setSelected(false);
V2.setBackground(Color.WHITE);
```

```
V3.setSelected(false);
V3.setBackground(Color.WHITE);
V4.setSelected(false);
V4.setBackground(Color.WHITE);
V5.setSelected(false);
V5.setBackground(Color.WHITE);
V6.setSelected(false);
V6.setBackground(Color.WHITE);
V7.setSelected(false);
V7.setBackground(Color.WHITE);
V8.setSelected(false);
V8.setBackground(Color.WHITE);
V9.setSelected(false);
V9.setBackground(Color.WHITE);
V10.setSelected(false);
V10.setBackground(Color.WHITE);
V11.setSelected(false);
V11.setBackground(Color.WHITE);
V12.setSelected(false);
V12.setBackground(Color.WHITE);
V13.setSelected(false);
V13.setBackground(Color.WHITE);
V14.setSelected(false);
V14.setBackground(Color.WHITE);
```

```
S1.setSelected(false);
S1.setBackground(Color.WHITE);
S2.setSelected(false);
S2.setBackground(Color.WHITE);
S3.setSelected(false);
S3.setBackground(Color.WHITE);
S4.setSelected(false);
S4.setBackground(Color.WHITE);
S5.setSelected(false);
S5.setBackground(Color.WHITE);
S6.setSelected(false);
S6.setBackground(Color.WHITE);
S7.setSelected(false);
S7.setBackground(Color.WHITE);
S8.setSelected(false);
S8.setBackground(Color.WHITE);
S9.setSelected(false);
S9.setBackground(Color.WHITE);
S10.setSelected(false);
S10.setBackground(Color.WHITE);
S11.setSelected(false);
S11.setBackground(Color.WHITE);
S12.setSelected(false);
S12.setBackground(Color.WHITE);
S13.setSelected(false);
S13.setBackground(Color.WHITE);
S14.setSelected(false);
```

```

S14.setBackground(Color.WHITE);

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();
    jLabel7 = new javax.swing.JLabel();
    jLabel8 = new javax.swing.JLabel();
    jLabel9 = new javax.swing.JLabel();
    jLabel10 = new javax.swing.JLabel();
    jLabel11 = new javax.swing.JLabel();
    jLabel13 = new javax.swing.JLabel();
    jLabel14 = new javax.swing.JLabel();
    jLabel15 = new javax.swing.JLabel();
    jLabel16 = new javax.swing.JLabel();
    jLabel17 = new javax.swing.JLabel();
    jLabel12 = new javax.swing.JLabel();
    jLabel18 = new javax.swing.JLabel();
    jLabel19 = new javax.swing.JLabel();
    jLabel20 = new javax.swing.JLabel();
    jLabel21 = new javax.swing.JLabel();
    jLabel22 = new javax.swing.JLabel();
    L6 = new javax.swing.JToggleButton();
    S1 = new javax.swing.JToggleButton();
    jSeparator1 = new javax.swing.JSeparator();
    L1 = new javax.swing.JToggleButton();
    M1 = new javax.swing.JToggleButton();
    X1 = new javax.swing.JToggleButton();
    J1 = new javax.swing.JToggleButton();
    V1 = new javax.swing.JToggleButton();
    L2 = new javax.swing.JToggleButton();
    L3 = new javax.swing.JToggleButton();
    L4 = new javax.swing.JToggleButton();
    L5 = new javax.swing.JToggleButton();
    L14 = new javax.swing.JToggleButton();
    L11 = new javax.swing.JToggleButton();
    L10 = new javax.swing.JToggleButton();
    L9 = new javax.swing.JToggleButton();

```

```
L8 = new javax.swing.JToggleButton();
L7 = new javax.swing.JToggleButton();
L12 = new javax.swing.JToggleButton();
L13 = new javax.swing.JToggleButton();
M6 = new javax.swing.JToggleButton();
M2 = new javax.swing.JToggleButton();
M3 = new javax.swing.JToggleButton();
M4 = new javax.swing.JToggleButton();
M5 = new javax.swing.JToggleButton();
M14 = new javax.swing.JToggleButton();
M11 = new javax.swing.JToggleButton();
M10 = new javax.swing.JToggleButton();
M9 = new javax.swing.JToggleButton();
M8 = new javax.swing.JToggleButton();
M7 = new javax.swing.JToggleButton();
M12 = new javax.swing.JToggleButton();
M13 = new javax.swing.JToggleButton();
X6 = new javax.swing.JToggleButton();
X2 = new javax.swing.JToggleButton();
X3 = new javax.swing.JToggleButton();
X4 = new javax.swing.JToggleButton();
X5 = new javax.swing.JToggleButton();
X14 = new javax.swing.JToggleButton();
X11 = new javax.swing.JToggleButton();
X10 = new javax.swing.JToggleButton();
X9 = new javax.swing.JToggleButton();
X8 = new javax.swing.JToggleButton();
X7 = new javax.swing.JToggleButton();
X12 = new javax.swing.JToggleButton();
X13 = new javax.swing.JToggleButton();
J6 = new javax.swing.JToggleButton();
J2 = new javax.swing.JToggleButton();
J3 = new javax.swing.JToggleButton();
J4 = new javax.swing.JToggleButton();
J5 = new javax.swing.JToggleButton();
J14 = new javax.swing.JToggleButton();
J11 = new javax.swing.JToggleButton();
J10 = new javax.swing.JToggleButton();
J9 = new javax.swing.JToggleButton();
J8 = new javax.swing.JToggleButton();
J7 = new javax.swing.JToggleButton();
J12 = new javax.swing.JToggleButton();
J13 = new javax.swing.JToggleButton();
V6 = new javax.swing.JToggleButton();
V2 = new javax.swing.JToggleButton();
V3 = new javax.swing.JToggleButton();
V4 = new javax.swing.JToggleButton();
V5 = new javax.swing.JToggleButton();
V14 = new javax.swing.JToggleButton();
V11 = new javax.swing.JToggleButton();
V10 = new javax.swing.JToggleButton();
V9 = new javax.swing.JToggleButton();
```

```

V8 = new javax.swing.JToggleButton();
V7 = new javax.swing.JToggleButton();
V12 = new javax.swing.JToggleButton();
V13 = new javax.swing.JToggleButton();
S6 = new javax.swing.JToggleButton();
S2 = new javax.swing.JToggleButton();
S3 = new javax.swing.JToggleButton();
S4 = new javax.swing.JToggleButton();
S5 = new javax.swing.JToggleButton();
S14 = new javax.swing.JToggleButton();
S11 = new javax.swing.JToggleButton();
S10 = new javax.swing.JToggleButton();
S9 = new javax.swing.JToggleButton();
S8 = new javax.swing.JToggleButton();
S7 = new javax.swing.JToggleButton();
S12 = new javax.swing.JToggleButton();
S13 = new javax.swing.JToggleButton();
btnl202 = new javax.swing.JButton();
btnl201 = new javax.swing.JButton();
btnl203 = new javax.swing.JButton();
btnl206 = new javax.swing.JButton();
jLabel23 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
jLabel1.setForeground(new java.awt.Color(255, 255, 255));
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("SELECCIÓN DE HORARIO");
getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(-4, 20,
630, -1));

jLabel2.setForeground(new java.awt.Color(255, 255, 255));
jLabel2.setText("07:00 - 08:00 ");
getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
100, -1, 20));

jLabel3.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
jLabel3.setForeground(new java.awt.Color(255, 255, 255));
jLabel3.setText("SABADO ");
getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(510,
70, -1, -1));

jLabel4.setForeground(new java.awt.Color(255, 255, 255));
jLabel4.setText("09:00 - 10:00 ");
getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
140, 80, 20));

jLabel5.setForeground(new java.awt.Color(255, 255, 255));
jLabel5.setText("08:00 - 09:00 ");

```

```
getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
120, 80, 20));
```

```
jLabel6.setForeground(new java.awt.Color(255, 255, 255));
jLabel6.setText("13:00 - 14:00 ");
getContentPane().add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
220, -1, 20));
```

```
jLabel7.setForeground(new java.awt.Color(255, 255, 255));
jLabel7.setText("12:00 - 13:00 ");
getContentPane().add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
200, -1, 20));
```

```
jLabel8.setForeground(new java.awt.Color(255, 255, 255));
jLabel8.setText("10:00 - 11:00 ");
getContentPane().add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
160, -1, 20));
```

```
jLabel9.setForeground(new java.awt.Color(255, 255, 255));
jLabel9.setText("11:00 - 12:00 ");
getContentPane().add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
180, -1, 20));
```

```
jLabel10.setForeground(new java.awt.Color(255, 255, 255));
jLabel10.setText("20:00 - 21:00 ");
getContentPane().add(jLabel10, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
360, -1, 20));
```

```
jLabel11.setForeground(new java.awt.Color(255, 255, 255));
jLabel11.setText("19:00 - 20:00 ");
getContentPane().add(jLabel11, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
340, -1, 20));
```

```
jLabel13.setForeground(new java.awt.Color(255, 255, 255));
jLabel13.setText("18:00 - 19:00 ");
getContentPane().add(jLabel13, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
320, -1, 20));
```

```
jLabel14.setForeground(new java.awt.Color(255, 255, 255));
jLabel14.setText("17:00 - 18:00 ");
getContentPane().add(jLabel14, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
300, -1, 20));
```

```
jLabel15.setForeground(new java.awt.Color(255, 255, 255));
jLabel15.setText("16:00 - 17:00 ");
getContentPane().add(jLabel15, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
280, -1, 20));
```

```
jLabel16.setForeground(new java.awt.Color(255, 255, 255));
jLabel16.setText("15:00 - 16:00 ");
getContentPane().add(jLabel16, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
260, -1, 20));
```

```

jLabel17.setForeground(new java.awt.Color(255, 255, 255));
jLabel17.setText("14:00 - 15:00 ");
getContentPane().add(jLabel17, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
240, -1, 20));

jLabel12.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
jLabel12.setForeground(new java.awt.Color(255, 255, 255));
jLabel12.setText("HORAS ");
getContentPane().add(jLabel12, new org.netbeans.lib.awtextra.AbsoluteConstraints(70,
70, -1, -1));

jLabel18.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
jLabel18.setForeground(new java.awt.Color(255, 255, 255));
jLabel18.setText("LUNES ");
getContentPane().add(jLabel18, new org.netbeans.lib.awtextra.AbsoluteConstraints(170,
70, -1, -1));

jLabel19.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
jLabel19.setForeground(new java.awt.Color(255, 255, 255));
jLabel19.setText("MARTES ");
getContentPane().add(jLabel19, new org.netbeans.lib.awtextra.AbsoluteConstraints(230,
70, -1, -1));

jLabel20.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
jLabel20.setForeground(new java.awt.Color(255, 255, 255));
jLabel20.setText("MIERCOLES ");
getContentPane().add(jLabel20, new org.netbeans.lib.awtextra.AbsoluteConstraints(290,
70, -1, -1));

jLabel21.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
jLabel21.setForeground(new java.awt.Color(255, 255, 255));
jLabel21.setText("JUEVES ");
getContentPane().add(jLabel21, new org.netbeans.lib.awtextra.AbsoluteConstraints(370,
70, -1, -1));

jLabel22.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
jLabel22.setForeground(new java.awt.Color(255, 255, 255));
jLabel22.setText("VIERNES ");
getContentPane().add(jLabel22, new org.netbeans.lib.awtextra.AbsoluteConstraints(440,
70, -1, -1));

L6.setBackground(java.awt.Color.white);
L6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        L6ActionPerformed(evt);
    }
});
getContentPane().add(L6, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 200,
60, 20));

S1.setBackground(java.awt.Color.white);

```

```

S1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        S1ActionPerformed(evt);
    }
});
getContentPane().add(S1, new org.netbeans.lib.awtextra.AbsoluteConstraints(510, 100,
60, 20));
getContentPane().add(jSeparator1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(50, 90, 520, 10));

L1.setBackground(java.awt.Color.white);
L1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        L1ActionPerformed(evt);
    }
});
getContentPane().add(L1, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 100,
60, 20));

M1.setBackground(java.awt.Color.white);
M1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        M1ActionPerformed(evt);
    }
});
getContentPane().add(M1, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 100,
60, 20));

X1.setBackground(java.awt.Color.white);
X1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        X1ActionPerformed(evt);
    }
});
getContentPane().add(X1, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 100,
60, 20));

J1.setBackground(java.awt.Color.white);
J1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        J1ActionPerformed(evt);
    }
});
getContentPane().add(J1, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 100,
60, 20));

V1.setBackground(java.awt.Color.white);
V1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        V1ActionPerformed(evt);
    }
});

```

```
getContentPane().add(V1, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 100, 60, 20));
```

```
L2.setBackground(java.awt.Color.white);  
L2.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        L2ActionPerformed(evt);  
    }  
});
```

```
getContentPane().add(L2, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 120, 60, 20));
```

```
L3.setBackground(java.awt.Color.white);  
L3.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        L3ActionPerformed(evt);  
    }  
});
```

```
getContentPane().add(L3, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 140, 60, 20));
```

```
L4.setBackground(java.awt.Color.white);  
L4.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        L4ActionPerformed(evt);  
    }  
});
```

```
getContentPane().add(L4, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 160, 60, 20));
```

```
L5.setBackground(java.awt.Color.white);  
L5.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        L5ActionPerformed(evt);  
    }  
});
```

```
getContentPane().add(L5, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 180, 60, 20));
```

```
L14.setBackground(java.awt.Color.white);  
L14.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        L14ActionPerformed(evt);  
    }  
});
```

```
getContentPane().add(L14, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 360, 60, 20));
```

```
L11.setBackground(java.awt.Color.white);  
L11.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        L11ActionPerformed(evt);  
    }  
});
```

```

    }
    });
    getContentPane().add(L11, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 300,
60, 20));

    L10.setBackground(java.awt.Color.white);
    L10.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            L10ActionPerformed(evt);
        }
    });
    getContentPane().add(L10, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 280,
60, 20));

    L9.setBackground(java.awt.Color.white);
    L9.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            L9ActionPerformed(evt);
        }
    });
    getContentPane().add(L9, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 260,
60, 20));

    L8.setBackground(java.awt.Color.white);
    L8.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            L8ActionPerformed(evt);
        }
    });
    getContentPane().add(L8, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 240,
60, 20));

    L7.setBackground(java.awt.Color.white);
    L7.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            L7ActionPerformed(evt);
        }
    });
    getContentPane().add(L7, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 220,
60, 20));

    L12.setBackground(java.awt.Color.white);
    L12.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            L12ActionPerformed(evt);
        }
    });
    getContentPane().add(L12, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 320,
60, 20));

    L13.setBackground(java.awt.Color.white);
    L13.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            L13ActionPerformed(evt);
        }
    });
    getContentPane().add(L13, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 340,
60, 20));

    M6.setBackground(java.awt.Color.white);
    M6.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            M6ActionPerformed(evt);
        }
    });
    getContentPane().add(M6, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 200,
60, 20));

    M2.setBackground(java.awt.Color.white);
    M2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            M2ActionPerformed(evt);
        }
    });
    getContentPane().add(M2, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 120,
60, 20));

    M3.setBackground(java.awt.Color.white);
    M3.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            M3ActionPerformed(evt);
        }
    });
    getContentPane().add(M3, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 140,
60, 20));

    M4.setBackground(java.awt.Color.white);
    M4.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            M4ActionPerformed(evt);
        }
    });
    getContentPane().add(M4, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 160,
60, 20));

    M5.setBackground(java.awt.Color.white);
    M5.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            M5ActionPerformed(evt);
        }
    });
    getContentPane().add(M5, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 180,
60, 20));

```

```
M14.setBackground(java.awt.Color.white);
M14.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        M14ActionPerformed(evt);
    }
});
getContentPane().add(M14, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 360,
60, 20));
```

```
M11.setBackground(java.awt.Color.white);
M11.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        M11ActionPerformed(evt);
    }
});
getContentPane().add(M11, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 300,
60, 20));
```

```
M10.setBackground(java.awt.Color.white);
M10.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        M10ActionPerformed(evt);
    }
});
getContentPane().add(M10, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 280,
60, 20));
```

```
M9.setBackground(java.awt.Color.white);
M9.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        M9ActionPerformed(evt);
    }
});
getContentPane().add(M9, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 260,
60, 20));
```

```
M8.setBackground(java.awt.Color.white);
M8.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        M8ActionPerformed(evt);
    }
});
getContentPane().add(M8, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 240,
60, 20));
```

```
M7.setBackground(java.awt.Color.white);
M7.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        M7ActionPerformed(evt);
    }
});
```

```

    getContentPane().add(M7, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 220,
60, 20));

    M12.setBackground(java.awt.Color.white);
    M12.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            M12ActionPerformed(evt);
        }
    });
    getContentPane().add(M12, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 320,
60, 20));

    M13.setBackground(java.awt.Color.white);
    M13.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            M13ActionPerformed(evt);
        }
    });
    getContentPane().add(M13, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 340,
60, 20));

    X6.setBackground(java.awt.Color.white);
    X6.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            X6ActionPerformed(evt);
        }
    });
    getContentPane().add(X6, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 200,
60, 20));

    X2.setBackground(java.awt.Color.white);
    X2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            X2ActionPerformed(evt);
        }
    });
    getContentPane().add(X2, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 120,
60, 20));

    X3.setBackground(java.awt.Color.white);
    X3.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            X3ActionPerformed(evt);
        }
    });
    getContentPane().add(X3, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 140,
60, 20));

    X4.setBackground(java.awt.Color.white);
    X4.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            X4ActionPerformed(evt);
        }
    });

```

```

    }
  });
  getContentPane().add(X4, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 160,
60, 20));

  X5.setBackground(java.awt.Color.white);
  X5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      X5ActionPerformed(evt);
    }
  });
  getContentPane().add(X5, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 180,
60, 20));

  X14.setBackground(java.awt.Color.white);
  X14.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      X14ActionPerformed(evt);
    }
  });
  getContentPane().add(X14, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 360,
60, 20));

  X11.setBackground(java.awt.Color.white);
  X11.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      X11ActionPerformed(evt);
    }
  });
  getContentPane().add(X11, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 300,
60, 20));

  X10.setBackground(java.awt.Color.white);
  X10.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      X10ActionPerformed(evt);
    }
  });
  getContentPane().add(X10, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 280,
60, 20));

  X9.setBackground(java.awt.Color.white);
  X9.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      X9ActionPerformed(evt);
    }
  });
  getContentPane().add(X9, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 260,
60, 20));

  X8.setBackground(java.awt.Color.white);
  X8.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            X8ActionPerformed(evt);
        }
    });
    getContentPane().add(X8, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 240,
60, 20));

    X7.setBackground(java.awt.Color.white);
    X7.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            X7ActionPerformed(evt);
        }
    });
    getContentPane().add(X7, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 220,
60, 20));

    X12.setBackground(java.awt.Color.white);
    X12.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            X12ActionPerformed(evt);
        }
    });
    getContentPane().add(X12, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 320,
60, 20));

    X13.setBackground(java.awt.Color.white);
    X13.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            X13ActionPerformed(evt);
        }
    });
    getContentPane().add(X13, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 340,
60, 20));

    J6.setBackground(java.awt.Color.white);
    J6.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            J6ActionPerformed(evt);
        }
    });
    getContentPane().add(J6, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 200,
60, 20));

    J2.setBackground(java.awt.Color.white);
    J2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            J2ActionPerformed(evt);
        }
    });
    getContentPane().add(J2, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 120,
60, 20));

```

```
J3.setBackground(java.awt.Color.white);
J3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        J3ActionPerformed(evt);
    }
});
getContentPane().add(J3, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 140,
60, 20));
```

```
J4.setBackground(java.awt.Color.white);
J4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        J4ActionPerformed(evt);
    }
});
getContentPane().add(J4, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 160,
60, 20));
```

```
J5.setBackground(java.awt.Color.white);
J5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        J5ActionPerformed(evt);
    }
});
getContentPane().add(J5, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 180,
60, 20));
```

```
J14.setBackground(java.awt.Color.white);
J14.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        J14ActionPerformed(evt);
    }
});
getContentPane().add(J14, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 360,
60, 20));
```

```
J11.setBackground(java.awt.Color.white);
J11.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        J11ActionPerformed(evt);
    }
});
getContentPane().add(J11, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 300,
60, 20));
```

```
J10.setBackground(java.awt.Color.white);
J10.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        J10ActionPerformed(evt);
    }
});
```

```
getContentPane().add(J10, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 280, 60, 20));
```

```
J9.setBackground(java.awt.Color.white);  
J9.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        J9ActionPerformed(evt);  
    }  
});
```

```
getContentPane().add(J9, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 260, 60, 20));
```

```
J8.setBackground(java.awt.Color.white);  
J8.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        J8ActionPerformed(evt);  
    }  
});
```

```
getContentPane().add(J8, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 240, 60, 20));
```

```
J7.setBackground(java.awt.Color.white);  
J7.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        J7ActionPerformed(evt);  
    }  
});
```

```
getContentPane().add(J7, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 220, 60, 20));
```

```
J12.setBackground(java.awt.Color.white);  
J12.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        J12ActionPerformed(evt);  
    }  
});
```

```
getContentPane().add(J12, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 320, 60, 20));
```

```
J13.setBackground(java.awt.Color.white);  
J13.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        J13ActionPerformed(evt);  
    }  
});
```

```
getContentPane().add(J13, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 340, 60, 20));
```

```
V6.setBackground(java.awt.Color.white);  
V6.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        V6ActionPerformed(evt);  
    }  
});
```

```
    }  
  });  
  getContentPane().add(V6, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 200,  
60, 20));
```

```
V2.setBackground(java.awt.Color.white);  
V2.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        V2ActionPerformed(evt);  
    }  
});  
getContentPane().add(V2, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 120,  
60, 20));
```

```
V3.setBackground(java.awt.Color.white);  
V3.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        V3ActionPerformed(evt);  
    }  
});  
getContentPane().add(V3, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 140,  
60, 20));
```

```
V4.setBackground(java.awt.Color.white);  
V4.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        V4ActionPerformed(evt);  
    }  
});  
getContentPane().add(V4, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 160,  
60, 20));
```

```
V5.setBackground(java.awt.Color.white);  
V5.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        V5ActionPerformed(evt);  
    }  
});  
getContentPane().add(V5, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 180,  
60, 20));
```

```
V14.setBackground(java.awt.Color.white);  
V14.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        V14ActionPerformed(evt);  
    }  
});  
getContentPane().add(V14, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 360,  
60, 20));
```

```
V11.setBackground(java.awt.Color.white);  
V11.addActionListener(new java.awt.event.ActionListener() {
```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            V11ActionPerformed(evt);
        }
    });
    getContentPane().add(V11, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 300,
60, 20));

    V10.setBackground(java.awt.Color.white);
    V10.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            V10ActionPerformed(evt);
        }
    });
    getContentPane().add(V10, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 280,
60, 20));

    V9.setBackground(java.awt.Color.white);
    V9.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            V9ActionPerformed(evt);
        }
    });
    getContentPane().add(V9, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 260,
60, 20));

    V8.setBackground(java.awt.Color.white);
    V8.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            V8ActionPerformed(evt);
        }
    });
    getContentPane().add(V8, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 240,
60, 20));

    V7.setBackground(java.awt.Color.white);
    V7.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            V7ActionPerformed(evt);
        }
    });
    getContentPane().add(V7, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 220,
60, 20));

    V12.setBackground(java.awt.Color.white);
    V12.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            V12ActionPerformed(evt);
        }
    });
    getContentPane().add(V12, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 320,
60, 20));

```

```
V13.setBackground(java.awt.Color.white);
V13.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        V13ActionPerformed(evt);
    }
});
getContentPane().add(V13, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 340,
60, 20));
```

```
S6.setBackground(java.awt.Color.white);
S6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        S6ActionPerformed(evt);
    }
});
getContentPane().add(S6, new org.netbeans.lib.awtextra.AbsoluteConstraints(510, 200,
60, 20));
```

```
S2.setBackground(java.awt.Color.white);
S2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        S2ActionPerformed(evt);
    }
});
getContentPane().add(S2, new org.netbeans.lib.awtextra.AbsoluteConstraints(510, 120,
60, 20));
```

```
S3.setBackground(java.awt.Color.white);
S3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        S3ActionPerformed(evt);
    }
});
getContentPane().add(S3, new org.netbeans.lib.awtextra.AbsoluteConstraints(510, 140,
60, 20));
```

```
S4.setBackground(java.awt.Color.white);
S4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        S4ActionPerformed(evt);
    }
});
getContentPane().add(S4, new org.netbeans.lib.awtextra.AbsoluteConstraints(510, 160,
60, 20));
```

```
S5.setBackground(java.awt.Color.white);
S5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        S5ActionPerformed(evt);
    }
});
```

```
getContentPane().add(S5, new org.netbeans.lib.awtextra.AbsoluteConstraints(510, 180, 60, 20));
```

```
S14.setBackground(java.awt.Color.white);  
S14.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        S14ActionPerformed(evt);  
    }  
});
```

```
getContentPane().add(S14, new org.netbeans.lib.awtextra.AbsoluteConstraints(510, 360, 60, 20));
```

```
S11.setBackground(java.awt.Color.white);  
S11.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        S11ActionPerformed(evt);  
    }  
});
```

```
getContentPane().add(S11, new org.netbeans.lib.awtextra.AbsoluteConstraints(510, 300, 60, 20));
```

```
S10.setBackground(java.awt.Color.white);  
S10.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        S10ActionPerformed(evt);  
    }  
});
```

```
getContentPane().add(S10, new org.netbeans.lib.awtextra.AbsoluteConstraints(510, 280, 60, 20));
```

```
S9.setBackground(java.awt.Color.white);  
S9.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        S9ActionPerformed(evt);  
    }  
});
```

```
getContentPane().add(S9, new org.netbeans.lib.awtextra.AbsoluteConstraints(510, 260, 60, 20));
```

```
S8.setBackground(java.awt.Color.white);  
S8.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        S8ActionPerformed(evt);  
    }  
});
```

```
getContentPane().add(S8, new org.netbeans.lib.awtextra.AbsoluteConstraints(510, 240, 60, 20));
```

```
S7.setBackground(java.awt.Color.white);  
S7.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        S7ActionPerformed(evt);  
    }  
});
```

```

    }
    });
    getContentPane().add(S7, new org.netbeans.lib.awtextra.AbsoluteConstraints(510, 220,
60, 20));

    S12.setBackground(java.awt.Color.white);
    S12.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            S12ActionPerformed(evt);
        }
    });
    getContentPane().add(S12, new org.netbeans.lib.awtextra.AbsoluteConstraints(510, 320,
60, 20));

    S13.setBackground(java.awt.Color.white);
    S13.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            S13ActionPerformed(evt);
        }
    });
    getContentPane().add(S13, new org.netbeans.lib.awtextra.AbsoluteConstraints(510, 340,
60, 20));

    btnl202.setBackground(new java.awt.Color(153, 153, 153));
    btnl202.setText("LABORATORIO 202");

    btnl202.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));
    btnl202.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnl202ActionPerformed(evt);
        }
    });
    getContentPane().add(btnl202, new org.netbeans.lib.awtextra.AbsoluteConstraints(180,
400, 130, 30));

    btnl201.setBackground(new java.awt.Color(153, 153, 153));
    btnl201.setText("LABORATORIO 201");

    btnl201.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));
    btnl201.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnl201ActionPerformed(evt);
        }
    });
    getContentPane().add(btnl201, new org.netbeans.lib.awtextra.AbsoluteConstraints(40,
400, 130, 30));

    btnl203.setBackground(new java.awt.Color(153, 153, 153));
    btnl203.setText("LABORATORIO 203");

```

```

btnl203.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));
    btnl203.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnl203ActionPerformed(evt);
        }
    });
    getContentPane().add(btnl203, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 400, 130, 30));

```

```

    btnl206.setBackground(new java.awt.Color(153, 153, 153));
    btnl206.setText("LABORATORIO 206");

```

```

btnl206.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));
    btnl206.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnl206ActionPerformed(evt);
        }
    });
    getContentPane().add(btnl206, new org.netbeans.lib.awtextra.AbsoluteConstraints(460, 400, 130, 30));

```

```

    jLabel23.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Switch/19832806_1531862570167961_239540534_n.jpg"))); // NOI18N
    jLabel23.setText("jLabel23");
    getContentPane().add(jLabel23, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 620, 450));

```

```

    pack();
} // </editor-fold> // GEN-END: initComponents

```

```

private void btnl202ActionPerformed(java.awt.event.ActionEvent evt) { // GEN-FIRST:event_btnl202ActionPerformed
    String ruta = ("C:\\SistemaPuerta\\LAB202.txt");
    try{
        FileWriter arch = new FileWriter(ruta);
        PrintWriter escribir = new PrintWriter(arch);
        for(int i=0; i<6; i++){
            for(int j=0; j<24; j++){
                escribir.print ( H[i][j] + " ");
            }
            escribir.println();
        }
        escribir.close();
        blanco();
    }
    catch(IOException e){
        JOptionPane . showMessageDialog ( null , "Error \n" + e);
    }
}

```

```

} //GEN-LAST:event_btnI202ActionPerformed

private void L1ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_L1ActionPerformed
if (L1.isSelected()){
    H[0][7]=String.valueOf(7);
    L1.setBackground(Color.RED);
} else{
    H[0][7]=String.valueOf(0);
    L1.setBackground(Color.WHITE);
}

// TODO add your handling code here:
} //GEN-LAST:event_L1ActionPerformed

private void btnI201ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_btnI201ActionPerformed
String ruta = ("C:\\SistemaPuerta\\LAB201.txt");
try{
    FileWriter arch = new FileWriter(ruta);
    PrintWriter escribir = new PrintWriter(arch);
    for(int i=0; i<6; i++){
        for(int j=0; j<24; j++){
            escribir.print ( H[i][j] + " ");
        }
        escribir.println();
    }
    escribir.close();
    blanco();
}
catch(IOException e){
    JOptionPane . showMessageDialog ( null , "Error \n" + e);
} // TODO add your handling code here:
} //GEN-LAST:event_btnI201ActionPerformed

private void L2ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_L2ActionPerformed
if (L2.isSelected()){
    H[0][8]=String.valueOf(8);
    L2.setBackground(Color.RED);
} else{
    H[0][8]=String.valueOf(0);
    L2.setBackground(Color.WHITE);
} // TODO add your handling code here:
} //GEN-LAST:event_L2ActionPerformed

private void L3ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_L3ActionPerformed
if (L3.isSelected()){
    H[0][9]=String.valueOf(9);
    L3.setBackground(Color.RED);
} else{

```

```

        L3.setBackground(Color.WHITE);
        H[0][9]=String.valueOf(0);
    } // TODO add your handling code here:
    }//GEN-LAST:event_L3ActionPerformed

    private void L4ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_L4ActionPerformed
        if (L4.isSelected()){
            H[0][10]=String.valueOf(10);
            L4.setBackground(Color.RED);
        }else{
            L4.setBackground(Color.WHITE);
            H[0][10]=String.valueOf(0);
        } // TODO add your handling code here:
    } //GEN-LAST:event_L4ActionPerformed

    private void L5ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_L5ActionPerformed
        if (L5.isSelected()){
            H[0][11]=String.valueOf(11);
            L5.setBackground(Color.RED);
        }else{
            L5.setBackground(Color.WHITE);
            H[0][11]=String.valueOf(0);
        } // TODO add your handling code here:
    } //GEN-LAST:event_L5ActionPerformed

    private void L6ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_L6ActionPerformed
        if (L6.isSelected()){
            H[0][12]=String.valueOf(12);
            L6.setBackground(Color.RED);
        }else{
            L6.setBackground(Color.WHITE);
            H[0][12]=String.valueOf(0);
        } // TODO add your handling code here:
    } //GEN-LAST:event_L6ActionPerformed

    private void L7ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_L7ActionPerformed
        if (L7.isSelected()){
            H[0][13]=String.valueOf(13);
            L7.setBackground(Color.RED);
        }else{
            L7.setBackground(Color.WHITE);
            H[0][13]=String.valueOf(0);
        } // TODO add your handling code here:
    } //GEN-LAST:event_L7ActionPerformed

    private void L8ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_L8ActionPerformed
        if (L8.isSelected()){

```

```

        H[0][14]=String.valueOf(14);
        L8.setBackground(Color.RED);
    }else{
        L8.setBackground(Color.WHITE);
        H[0][14]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_L8ActionPerformed

private void L9ActionPerformed(java.awt.event.ActionEvent evt) //GEN-
FIRST:event_L9ActionPerformed
    if (L9.isSelected()){
        H[0][15]=String.valueOf(15);
        L9.setBackground(Color.RED);
    }else{
        L9.setBackground(Color.WHITE);
        H[0][15]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_L9ActionPerformed

private void L10ActionPerformed(java.awt.event.ActionEvent evt) //GEN-
FIRST:event_L10ActionPerformed
    if (L10.isSelected()){
        H[0][16]=String.valueOf(16);
        L10.setBackground(Color.RED);
    }else{
        L10.setBackground(Color.WHITE);
        H[0][16]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_L10ActionPerformed

private void L11ActionPerformed(java.awt.event.ActionEvent evt) //GEN-
FIRST:event_L11ActionPerformed
    if (L11.isSelected()){
        H[0][17]=String.valueOf(17);
        L11.setBackground(Color.RED);
    }else{
        L11.setBackground(Color.WHITE);
        H[0][17]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_L11ActionPerformed

private void L12ActionPerformed(java.awt.event.ActionEvent evt) //GEN-
FIRST:event_L12ActionPerformed
    if (L12.isSelected()){
        H[0][18]=String.valueOf(18);
        L12.setBackground(Color.RED);
    }else{
        L12.setBackground(Color.WHITE);
        H[0][18]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_L12ActionPerformed

```

```

private void L13ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_L13ActionPerformed
    if (L13.isSelected()){
        H[0][19]=String.valueOf(19);
        L13.setBackground(Color.RED);
    }else{
        L13.setBackground(Color.WHITE);
        H[0][19]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_L13ActionPerformed

```

```

private void L14ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_L14ActionPerformed
    if (L14.isSelected()){
        H[0][20]=String.valueOf(20);
        L14.setBackground(Color.RED);
    }else{
        L14.setBackground(Color.WHITE);
        H[0][20]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_L14ActionPerformed

```

```

private void btn1203ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_btn1203ActionPerformed
String ruta = ("C:\\SistemaPuerta\\LAB203.txt");
try{
    FileWriter arch = new FileWriter(ruta);
    PrintWriter escribir = new PrintWriter(arch);
    for(int i=0; i<6; i++){
        for(int j=0; j<24; j++){
            escribir.print ( H[i][j] + " ");
        }
        escribir.println();
    }
    escribir.close();
    blanco();
}
catch(IOException e){
    JOptionPane . showMessageDialog ( null , "Error \n" + e);
} // TODO add your handling code here:
} //GEN-LAST:event_btn1203ActionPerformed

```

```

private void btn1206ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_btn1206ActionPerformed
String ruta = ("C:\\SistemaPuerta\\LAB206.txt");
try{
    FileWriter arch = new FileWriter(ruta);
    PrintWriter escribir = new PrintWriter(arch);
    for(int i=0; i<6; i++){
        for(int j=0; j<24; j++){
            escribir.print ( H[i][j] + " ");
        }
    }
}

```

```

        escribir.println();
    }
    escribir.close();
    blanco();
}
catch(IOException e){
JOptionPane . showMessageDialog ( null , "Error \n" + e);
} // TODO add your handling code here:
} //GEN-LAST:event_btnI206ActionPerformed

private void M1ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_M1ActionPerformed
    if (M1.isSelected()){
        H[1][7]=String.valueOf(7);
        M1.setBackground(Color.RED);
    }else{
        M1.setBackground(Color.WHITE);
        H[1][7]=String.valueOf(0);
    }
    // TODO add your handling code here:
} //GEN-LAST:event_M1ActionPerformed

private void J1ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_J1ActionPerformed
    if (J1.isSelected()){
        H[3][7]=String.valueOf(7);
        J1.setBackground(Color.RED);
    }else{
        J1.setBackground(Color.WHITE);
        H[3][7]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_J1ActionPerformed

private void M2ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_M2ActionPerformed
    if (M2.isSelected()){
        H[1][8]=String.valueOf(8);
        M2.setBackground(Color.RED);
    }else{
        M2.setBackground(Color.WHITE);
        H[1][8]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_M2ActionPerformed

private void M3ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_M3ActionPerformed
    if (M3.isSelected()){
        H[1][9]=String.valueOf(9);
        M3.setBackground(Color.RED);
    }else{
        M3.setBackground(Color.WHITE);
        H[1][9]=String.valueOf(0);
    }
}

```

```

        } // TODO add your handling code here:
    } //GEN-LAST:event_M3ActionPerformed

    private void M4ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_M4ActionPerformed
        if (M4.isSelected()){
            H[1][10]=String.valueOf(10);
            M4.setBackground(Color.RED);
        }else{
            M4.setBackground(Color.WHITE);
            H[1][10]=String.valueOf(0);
        } // TODO add your handling code here:
    } //GEN-LAST:event_M4ActionPerformed

    private void M5ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_M5ActionPerformed
        if (M5.isSelected()){
            H[1][11]=String.valueOf(11);
            M5.setBackground(Color.RED);
        }else{
            M5.setBackground(Color.WHITE);
            H[1][11]=String.valueOf(0);
        } // TODO add your handling code here:
    } //GEN-LAST:event_M5ActionPerformed

    private void M6ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_M6ActionPerformed
        if (M6.isSelected()){
            H[1][12]=String.valueOf(12);
            M6.setBackground(Color.RED);
        }else{
            M6.setBackground(Color.WHITE);
            H[1][12]=String.valueOf(0);
        } // TODO add your handling code here:
    } //GEN-LAST:event_M6ActionPerformed

    private void M7ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_M7ActionPerformed
        if (M7.isSelected()){
            H[1][13]=String.valueOf(13);
            M7.setBackground(Color.RED);
        }else{
            M7.setBackground(Color.WHITE);
            H[1][13]=String.valueOf(0);
        } // TODO add your handling code here:
    } //GEN-LAST:event_M7ActionPerformed

    private void M8ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_M8ActionPerformed
        if (M8.isSelected()){
            H[1][14]=String.valueOf(14);
            M8.setBackground(Color.RED);
        }
    }

```

```

        }else{
            M8.setBackground(Color.WHITE);
            H[1][14]=String.valueOf(0);
        } // TODO add your handling code here:
    }//GEN-LAST:event_M8ActionPerformed

    private void M9ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_M9ActionPerformed
        if (M9.isSelected()){
            H[1][15]=String.valueOf(15);
            M9.setBackground(Color.RED);
        }else{
            M9.setBackground(Color.WHITE);
            H[1][15]=String.valueOf(0);
        } // TODO add your handling code here:
    }//GEN-LAST:event_M9ActionPerformed

    private void M10ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_M10ActionPerformed
        if (M10.isSelected()){
            H[1][16]=String.valueOf(16);
            M10.setBackground(Color.RED);
        }else{
            M10.setBackground(Color.WHITE);
            H[1][16]=String.valueOf(0);
        } // TODO add your handling code here:
    }//GEN-LAST:event_M10ActionPerformed

    private void M11ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_M11ActionPerformed
        if (M11.isSelected()){
            H[1][17]=String.valueOf(17);
            M11.setBackground(Color.RED);
        }else{
            M11.setBackground(Color.WHITE);
            H[1][17]=String.valueOf(0);
        } // TODO add your handling code here:
    }//GEN-LAST:event_M11ActionPerformed

    private void M12ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_M12ActionPerformed
        if (M12.isSelected()){
            H[1][18]=String.valueOf(18);
            M12.setBackground(Color.RED);
        }else{
            M12.setBackground(Color.WHITE);
            H[1][18]=String.valueOf(0);
        } // TODO add your handling code here:
    }//GEN-LAST:event_M12ActionPerformed

    private void M13ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_M13ActionPerformed

```

```

if (M13.isSelected()){
    H[1][19]=String.valueOf(19);
    M13.setBackground(Color.RED);
}else{
    M13.setBackground(Color.WHITE);
    H[1][19]=String.valueOf(0);
} // TODO add your handling code here:
} //GEN-LAST:event_M13ActionPerformed

private void M14ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_M14ActionPerformed
    if (M14.isSelected()){
        H[1][20]=String.valueOf(20);
        M14.setBackground(Color.RED);
    }else{
        M14.setBackground(Color.WHITE);
        H[1][20]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_M14ActionPerformed

private void X1ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_X1ActionPerformed
    if (X1.isSelected()){
        H[2][7]=String.valueOf(7);
        X1.setBackground(Color.RED);
    }else{
        X1.setBackground(Color.WHITE);
        H[2][7]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_X1ActionPerformed

private void X2ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_X2ActionPerformed
    if (X2.isSelected()){
        H[2][8]=String.valueOf(8);
        X2.setBackground(Color.RED);
    }else{
        X2.setBackground(Color.WHITE);
        H[2][8]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_X2ActionPerformed

private void X3ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_X3ActionPerformed
    if (X3.isSelected()){
        H[2][9]=String.valueOf(9);
        X3.setBackground(Color.RED);
    }else{
        X3.setBackground(Color.WHITE);
        H[2][9]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_X3ActionPerformed

```

```

private void X4ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_X4ActionPerformed
    if (X4.isSelected()){
        H[2][10]=String.valueOf(10);
        X4.setBackground(Color.RED);
    }else{
        X4.setBackground(Color.WHITE);
        H[2][10]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_X4ActionPerformed

```

```

private void X5ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_X5ActionPerformed
    if (X5.isSelected()){
        H[2][11]=String.valueOf(11);
        X5.setBackground(Color.RED);
    }else{
        X5.setBackground(Color.WHITE);
        H[2][11]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_X5ActionPerformed

```

```

private void X6ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_X6ActionPerformed
    if (X6.isSelected()){
        H[2][12]=String.valueOf(12);
        X6.setBackground(Color.RED);
    }else{
        X6.setBackground(Color.WHITE);
        H[2][12]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_X6ActionPerformed

```

```

private void X7ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_X7ActionPerformed
    if (X7.isSelected()){
        H[2][13]=String.valueOf(13);
        X7.setBackground(Color.RED);
    }else{
        X7.setBackground(Color.WHITE);
        H[2][13]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_X7ActionPerformed

```

```

private void X8ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_X8ActionPerformed
    if (X8.isSelected()){
        H[2][14]=String.valueOf(14);
        X8.setBackground(Color.RED);
    }else{
        X8.setBackground(Color.WHITE);
    }
}

```

```

        H[2][14]=String.valueOf(0);
    }    // TODO add your handling code here:
} //GEN-LAST:event_X8ActionPerformed

private void X9ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_X9ActionPerformed
    if (X9.isSelected()){
        H[2][15]=String.valueOf(15);
        X9.setBackground(Color.RED);
    }else{
        X9.setBackground(Color.WHITE);
        H[2][15]=String.valueOf(0);
    }    // TODO add your handling code here:
} //GEN-LAST:event_X9ActionPerformed

private void X10ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_X10ActionPerformed
    if (X10.isSelected()){
        H[2][16]=String.valueOf(16);
        X10.setBackground(Color.RED);
    }else{
        X10.setBackground(Color.WHITE);
        H[2][16]=String.valueOf(0);
    }    // TODO add your handling code here:
} //GEN-LAST:event_X10ActionPerformed

private void X11ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_X11ActionPerformed
    if (X11.isSelected()){
        H[2][17]=String.valueOf(17);
        X11.setBackground(Color.RED);
    }else{
        X11.setBackground(Color.WHITE);
        H[2][17]=String.valueOf(0);
    }    // TODO add your handling code here:
} //GEN-LAST:event_X11ActionPerformed

private void X12ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_X12ActionPerformed
    if (X12.isSelected()){
        H[2][18]=String.valueOf(18);
        X12.setBackground(Color.RED);
    }else{
        X12.setBackground(Color.WHITE);
        H[2][18]=String.valueOf(0);
    }    // TODO add your handling code here:
} //GEN-LAST:event_X12ActionPerformed

private void X13ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_X13ActionPerformed
    if (X13.isSelected()){
        H[2][19]=String.valueOf(19);

```

```

        X13.setBackground(Color.RED);
    }else{
        X13.setBackground(Color.WHITE);
        H[2][19]=String.valueOf(0);
        } // TODO add your handling code here:
    } //GEN-LAST:event_X13ActionPerformed

private void X14ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_X14ActionPerformed
    if (X14.isSelected()){
        H[2][20]=String.valueOf(20);
        X14.setBackground(Color.RED);
    }else{
        X14.setBackground(Color.WHITE);
        H[2][20]=String.valueOf(0);
        } // TODO add your handling code here:
    } //GEN-LAST:event_X14ActionPerformed

private void J2ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_J2ActionPerformed
    if (J2.isSelected()){
        H[3][8]=String.valueOf(8);
        J2.setBackground(Color.RED);
    }else{
        J2.setBackground(Color.WHITE);
        H[3][8]=String.valueOf(0);
        } // TODO add your handling code here:
    } //GEN-LAST:event_J2ActionPerformed

private void J3ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_J3ActionPerformed
    if (J3.isSelected()){
        H[3][9]=String.valueOf(9);
        J3.setBackground(Color.RED);
    }else{
        J3.setBackground(Color.WHITE);
        H[3][9]=String.valueOf(0);
        } // TODO add your handling code here:
    } //GEN-LAST:event_J3ActionPerformed

private void J4ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_J4ActionPerformed
    if (J4.isSelected()){
        H[3][10]=String.valueOf(10);
        J4.setBackground(Color.RED);
    }else{
        J4.setBackground(Color.WHITE);
        H[3][10]=String.valueOf(0);
        } // TODO add your handling code here:
    } //GEN-LAST:event_J4ActionPerformed

```

```

private void J5ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_J5ActionPerformed
    if (J5.isSelected()){
        H[3][11]=String.valueOf(11);
        J5.setBackground(Color.RED);
    }else{
        J5.setBackground(Color.WHITE);
        H[3][11]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_J5ActionPerformed

```

```

private void J6ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_J6ActionPerformed
    if (J6.isSelected()){
        H[3][12]=String.valueOf(12);
        J6.setBackground(Color.RED);
    }else{
        J6.setBackground(Color.WHITE);
        H[3][12]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_J6ActionPerformed

```

```

private void J7ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_J7ActionPerformed
    if (J7.isSelected()){
        H[3][13]=String.valueOf(13);
        J7.setBackground(Color.RED);
    }else{
        J7.setBackground(Color.WHITE);
        H[3][13]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_J7ActionPerformed

```

```

private void J8ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_J8ActionPerformed
    if (J8.isSelected()){
        H[3][14]=String.valueOf(14);
        J8.setBackground(Color.RED);
    }else{
        J8.setBackground(Color.WHITE);
        H[3][14]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_J8ActionPerformed

```

```

private void J9ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_J9ActionPerformed
    if (J9.isSelected()){
        H[3][15]=String.valueOf(15);
        J9.setBackground(Color.RED);
    }else{
        J9.setBackground(Color.WHITE);
        H[3][15]=String.valueOf(0);
    }
}

```

```

        } // TODO add your handling code here:
    } //GEN-LAST:event_J9ActionPerformed

    private void J10ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_J10ActionPerformed
        if (J10.isSelected()){
            H[3][16]=String.valueOf(16);
            J10.setBackground(Color.RED);
        }else{
            J10.setBackground(Color.WHITE);
            H[3][16]=String.valueOf(0);
        } // TODO add your handling code here:
    } //GEN-LAST:event_J10ActionPerformed

    private void J11ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_J11ActionPerformed
        if (J11.isSelected()){
            H[3][17]=String.valueOf(17);
            J11.setBackground(Color.RED);
        }else{
            J11.setBackground(Color.WHITE);
            H[3][17]=String.valueOf(0);
        } // TODO add your handling code here:
    } //GEN-LAST:event_J11ActionPerformed

    private void J12ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_J12ActionPerformed
        if (J12.isSelected()){
            H[3][18]=String.valueOf(18);
            J12.setBackground(Color.RED);
        }else{
            J12.setBackground(Color.WHITE);
            H[3][18]=String.valueOf(0);
        } // TODO add your handling code here:
    } //GEN-LAST:event_J12ActionPerformed

    private void J13ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_J13ActionPerformed
        if (J13.isSelected()){
            H[3][19]=String.valueOf(19);
            J13.setBackground(Color.RED);
        }else{
            J13.setBackground(Color.WHITE);
            H[3][19]=String.valueOf(0);
        } // TODO add your handling code here:
    } //GEN-LAST:event_J13ActionPerformed

    private void J14ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_J14ActionPerformed
        if (J14.isSelected()){
            H[3][20]=String.valueOf(20);
            J14.setBackground(Color.RED);
        }
    }

```

```

        }else{
            J14.setBackground(Color.WHITE);
            H[3][20]=String.valueOf(0);
        } // TODO add your handling code here:
    }//GEN-LAST:event_J14ActionPerformed

    private void V1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_V1ActionPerformed
        if (V1.isSelected()){
            H[4][7]=String.valueOf(7);
            V1.setBackground(Color.RED);
        }else{
            V1.setBackground(Color.WHITE);
            H[4][7]=String.valueOf(0);
        } // TODO add your handling code here:
    }//GEN-LAST:event_V1ActionPerformed

    private void V2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_V2ActionPerformed
        if (V2.isSelected()){
            H[4][8]=String.valueOf(8);
            V2.setBackground(Color.RED);
        }else{
            V2.setBackground(Color.WHITE);
            H[4][8]=String.valueOf(0);
        } // TODO add your handling code here:
    }//GEN-LAST:event_V2ActionPerformed

    private void V3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_V3ActionPerformed
        if (V3.isSelected()){
            H[4][9]=String.valueOf(9);
            V3.setBackground(Color.RED);
        }else{
            V3.setBackground(Color.WHITE);
            H[4][9]=String.valueOf(0);
        } // TODO add your handling code here:
    }//GEN-LAST:event_V3ActionPerformed

    private void V4ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_V4ActionPerformed
        if (V4.isSelected()){
            H[4][10]=String.valueOf(10);
            V4.setBackground(Color.RED);
        }else{
            V4.setBackground(Color.WHITE);
            H[4][10]=String.valueOf(0);
        } // TODO add your handling code here:
    }//GEN-LAST:event_V4ActionPerformed

    private void V5ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_V5ActionPerformed

```

```

if (V5.isSelected()){
    H[4][11]=String.valueOf(11);
    V5.setBackground(Color.RED);
}else{
    V5.setBackground(Color.WHITE);
    H[4][11]=String.valueOf(0);
} // TODO add your handling code here:
} //GEN-LAST:event_V5ActionPerformed

private void V6ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_V6ActionPerformed
    if (V6.isSelected()){
        H[4][12]=String.valueOf(12);
        V6.setBackground(Color.RED);
    }else{
        V6.setBackground(Color.WHITE);
        H[4][12]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_V6ActionPerformed

private void V7ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_V7ActionPerformed
    if (V7.isSelected()){
        H[4][13]=String.valueOf(13);
        V7.setBackground(Color.RED);
    }else{
        V7.setBackground(Color.WHITE);
        H[4][13]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_V7ActionPerformed

private void V8ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_V8ActionPerformed
    if (V8.isSelected()){
        H[4][14]=String.valueOf(14);
        V8.setBackground(Color.RED);
    }else{
        V8.setBackground(Color.WHITE);
        H[4][14]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_V8ActionPerformed

private void V9ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_V9ActionPerformed
    if (V9.isSelected()){
        H[4][15]=String.valueOf(15);
        V9.setBackground(Color.RED);
    }else{
        V9.setBackground(Color.WHITE);
        H[4][15]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_V9ActionPerformed

```

```

private void V10ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_V10ActionPerformed
    if (V10.isSelected()){
        H[4][16]=String.valueOf(16);
        V10.setBackground(Color.RED);
    }else{
        V10.setBackground(Color.WHITE);
        H[4][16]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_V10ActionPerformed

```

```

private void V11ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_V11ActionPerformed
    if (V11.isSelected()){
        H[4][17]=String.valueOf(17);
        V11.setBackground(Color.RED);
    }else{
        V11.setBackground(Color.WHITE);
        H[4][17]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_V11ActionPerformed

```

```

private void V12ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_V12ActionPerformed
    if (V12.isSelected()){
        H[4][18]=String.valueOf(18);
        V12.setBackground(Color.RED);
    }else{
        V12.setBackground(Color.WHITE);
        H[4][18]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_V12ActionPerformed

```

```

private void V13ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_V13ActionPerformed
    if (V13.isSelected()){
        H[4][19]=String.valueOf(19);
        V13.setBackground(Color.RED);
    }else{
        V13.setBackground(Color.WHITE);
        H[4][19]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_V13ActionPerformed

```

```

private void V14ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_V14ActionPerformed
    if (V14.isSelected()){
        H[4][20]=String.valueOf(20);
        V14.setBackground(Color.RED);
    }else{
        V14.setBackground(Color.WHITE);
    }
}

```

```

        H[4][20]=String.valueOf(0);
    }    // TODO add your handling code here:
} //GEN-LAST:event_V14ActionPerformed

private void S1ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_S1ActionPerformed
    if (S1.isSelected()){
        H[5][7]=String.valueOf(7);
        S1.setBackground(Color.RED);
    }else{
        S1.setBackground(Color.WHITE);
        H[5][7]=String.valueOf(0);
    }    // TODO add your handling code here:
} //GEN-LAST:event_S1ActionPerformed

private void S2ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_S2ActionPerformed
    if (S2.isSelected()){
        H[5][8]=String.valueOf(8);
        S2.setBackground(Color.RED);
    }else{
        S2.setBackground(Color.WHITE);
        H[5][8]=String.valueOf(0);
    }    // TODO add your handling code here:
} //GEN-LAST:event_S2ActionPerformed

private void S3ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_S3ActionPerformed
    if (S3.isSelected()){
        H[5][9]=String.valueOf(9);
        S3.setBackground(Color.RED);
    }else{
        S3.setBackground(Color.WHITE);
        H[5][9]=String.valueOf(0);
    }    // TODO add your handling code here:
} //GEN-LAST:event_S3ActionPerformed

private void S4ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_S4ActionPerformed
    if (S4.isSelected()){
        H[5][10]=String.valueOf(10);
        S4.setBackground(Color.RED);
    }else{
        S4.setBackground(Color.WHITE);
        H[5][10]=String.valueOf(0);
    }    // TODO add your handling code here:
} //GEN-LAST:event_S4ActionPerformed

private void S5ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_S5ActionPerformed
    if (S5.isSelected()){
        H[5][11]=String.valueOf(11);

```

```

        S5.setBackground(Color.RED);
    }else{
        S5.setBackground(Color.WHITE);
        H[5][11]=String.valueOf(0);
    }    // TODO add your handling code here:
} //GEN-LAST:event_S5ActionPerformed

private void S6ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_S6ActionPerformed
    if (S6.isSelected()){
        H[5][12]=String.valueOf(12);
        S6.setBackground(Color.RED);
    }else{
        S6.setBackground(Color.WHITE);
        H[5][12]=String.valueOf(0);
    }    // TODO add your handling code here:
} //GEN-LAST:event_S6ActionPerformed

private void S7ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_S7ActionPerformed
    if (S7.isSelected()){
        H[5][13]=String.valueOf(13);
        S7.setBackground(Color.RED);
    }else{
        S7.setBackground(Color.WHITE);
        H[5][13]=String.valueOf(0);
    }    // TODO add your handling code here:
} //GEN-LAST:event_S7ActionPerformed

private void S8ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_S8ActionPerformed
    if (S8.isSelected()){
        H[5][14]=String.valueOf(14);
        S8.setBackground(Color.RED);
    }else{
        S8.setBackground(Color.WHITE);
        H[5][14]=String.valueOf(0);
    }    // TODO add your handling code here:
} //GEN-LAST:event_S8ActionPerformed

private void S9ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_S9ActionPerformed
    if (S9.isSelected()){
        H[5][15]=String.valueOf(15);
        S9.setBackground(Color.RED);
    }else{
        S9.setBackground(Color.WHITE);
        H[5][15]=String.valueOf(0);
    }    // TODO add your handling code here:
} //GEN-LAST:event_S9ActionPerformed

```

```

private void S10ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_S10ActionPerformed
    if (S10.isSelected()){
        H[5][16]=String.valueOf(16);
        S10.setBackground(Color.RED);
    }else{
        S10.setBackground(Color.WHITE);
        H[5][16]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_S10ActionPerformed

```

```

private void S11ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_S11ActionPerformed
    if (S11.isSelected()){
        H[5][17]=String.valueOf(17);
        S11.setBackground(Color.RED);
    }else{
        S11.setBackground(Color.WHITE);
        H[5][17]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_S11ActionPerformed

```

```

private void S12ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_S12ActionPerformed
    if (S12.isSelected()){
        H[5][18]=String.valueOf(18);
        S12.setBackground(Color.RED);
    }else{
        S12.setBackground(Color.WHITE);
        H[5][18]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_S12ActionPerformed

```

```

private void S13ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_S13ActionPerformed
    if (S13.isSelected()){
        H[5][19]=String.valueOf(19);
        S13.setBackground(Color.RED);
    }else{
        S13.setBackground(Color.WHITE);
        H[5][19]=String.valueOf(0);
    } // TODO add your handling code here:
} //GEN-LAST:event_S13ActionPerformed

```

```

private void S14ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_S14ActionPerformed
    if (S14.isSelected()){
        H[5][20]=String.valueOf(20);
        S14.setBackground(Color.RED);
    }else{
        S14.setBackground(Color.WHITE);
        H[5][20]=String.valueOf(0);
    }
}

```

```
    } // TODO add your handling code here:  
} //GEN-LAST:event_S14ActionPerformed
```

```
// Variables declaration - do not modify //GEN-BEGIN:variables
```

```
private javax.swing.JToggleButton J1;  
private javax.swing.JToggleButton J10;  
private javax.swing.JToggleButton J11;  
private javax.swing.JToggleButton J12;  
private javax.swing.JToggleButton J13;  
private javax.swing.JToggleButton J14;  
private javax.swing.JToggleButton J2;  
private javax.swing.JToggleButton J3;  
private javax.swing.JToggleButton J4;  
private javax.swing.JToggleButton J5;  
private javax.swing.JToggleButton J6;  
private javax.swing.JToggleButton J7;  
private javax.swing.JToggleButton J8;  
private javax.swing.JToggleButton J9;  
private javax.swing.JToggleButton L1;  
private javax.swing.JToggleButton L10;  
private javax.swing.JToggleButton L11;  
private javax.swing.JToggleButton L12;  
private javax.swing.JToggleButton L13;  
private javax.swing.JToggleButton L14;  
private javax.swing.JToggleButton L2;  
private javax.swing.JToggleButton L3;  
private javax.swing.JToggleButton L4;  
private javax.swing.JToggleButton L5;  
private javax.swing.JToggleButton L6;  
private javax.swing.JToggleButton L7;  
private javax.swing.JToggleButton L8;  
private javax.swing.JToggleButton L9;  
private javax.swing.JToggleButton M1;  
private javax.swing.JToggleButton M10;  
private javax.swing.JToggleButton M11;  
private javax.swing.JToggleButton M12;  
private javax.swing.JToggleButton M13;  
private javax.swing.JToggleButton M14;  
private javax.swing.JToggleButton M2;  
private javax.swing.JToggleButton M3;  
private javax.swing.JToggleButton M4;  
private javax.swing.JToggleButton M5;  
private javax.swing.JToggleButton M6;  
private javax.swing.JToggleButton M7;  
private javax.swing.JToggleButton M8;  
private javax.swing.JToggleButton M9;  
private javax.swing.JToggleButton S1;  
private javax.swing.JToggleButton S10;  
private javax.swing.JToggleButton S11;  
private javax.swing.JToggleButton S12;  
private javax.swing.JToggleButton S13;
```

```
private javax.swing.JToggleButton S14;
private javax.swing.JToggleButton S2;
private javax.swing.JToggleButton S3;
private javax.swing.JToggleButton S4;
private javax.swing.JToggleButton S5;
private javax.swing.JToggleButton S6;
private javax.swing.JToggleButton S7;
private javax.swing.JToggleButton S8;
private javax.swing.JToggleButton S9;
private javax.swing.JToggleButton V1;
private javax.swing.JToggleButton V10;
private javax.swing.JToggleButton V11;
private javax.swing.JToggleButton V12;
private javax.swing.JToggleButton V13;
private javax.swing.JToggleButton V14;
private javax.swing.JToggleButton V2;
private javax.swing.JToggleButton V3;
private javax.swing.JToggleButton V4;
private javax.swing.JToggleButton V5;
private javax.swing.JToggleButton V6;
private javax.swing.JToggleButton V7;
private javax.swing.JToggleButton V8;
private javax.swing.JToggleButton V9;
private javax.swing.JToggleButton X1;
private javax.swing.JToggleButton X10;
private javax.swing.JToggleButton X11;
private javax.swing.JToggleButton X12;
private javax.swing.JToggleButton X13;
private javax.swing.JToggleButton X14;
private javax.swing.JToggleButton X2;
private javax.swing.JToggleButton X3;
private javax.swing.JToggleButton X4;
private javax.swing.JToggleButton X5;
private javax.swing.JToggleButton X6;
private javax.swing.JToggleButton X7;
private javax.swing.JToggleButton X8;
private javax.swing.JToggleButton X9;
private javax.swing.JButton btnl201;
private javax.swing.JButton btnl202;
private javax.swing.JButton btnl203;
private javax.swing.JButton btnl206;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel jLabel19;
```

```

private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel20;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel22;
private javax.swing.JLabel jLabel23;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JSeparator jSeparator1;
// End of variables declaration//GEN-END:variables
}

```

Clase Usuario

```

package Switch;
import Base.Base;
import java.awt.Color;
import java.awt.Font;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Iterator;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import panamahitek.Arduino.PanamaHitek_Arduino;

/**
 *
 *
 */
public class Usuario extends javax.swing.JFrame {
//Se instancia la Librería Arduino
    ResultSet RS;
    Base CC = new Base();
    Connection CN = CC.conexion();
    PanamaHitek_Arduino Arduino = new PanamaHitek_Arduino();
    Controls c = new Controls();

```

```

public Usuario() {

    initComponents();

    getPorts();

    //Se inicia la conexión con Arduino para transmisión de datos
}
private void close() {
    dispose();
}
public void getPorts() {
    jComboBoxPorts.removeAllItems();
    if (Arduino.getPortsAvailable() > 0) {
        List lst = Arduino.getSerialPorts();
        for(int i=0; i<lst.size(); i++){
            jComboBoxPorts.addItem(lst.get(i));
        }
        //Arduino.getSerialPorts().forEach(i -> jComboBoxPorts.addItem(i));
        jButtonConectar.setEnabled(true);
        jButtonConectar.setBackground(new Color(255, 255, 255));
    } else {
        jButtonConectar.setEnabled(false);
        jButtonConectar.setBackground(new Color(204, 204, 204));
    }

}

void LIMPIAR(){
    //PARA LIMPIAR LOS CAMPOS DE TEXTOS
    txtusuario.setText("");
    txtcontrasena.setText("");
}

void Acceder(String USUARIO,String CONTRASENA){

    String CAP="";
    String sql="select * from login WHERE USUARIO='"+USUARIO+"' AND
CONTRASENA='"+CONTRASENA+"' ";
    String sql2 ="UPDATE puerto SET COD_PUERTO = '"+jComboBoxPorts.getSelectedItem()+"'
WHERE COD = 1";

    try {
        //
        Statement st = CN.createStatement();

```

```

st.execute(sql2);
ResultSet RS = st.executeQuery(sql);

//PARA VERIFICAR QUE USUARIO
while(RS.next())
{
    CAP=RS.getString("TIPO_USUARIO");
}
if(CAP.equals("Admin"))
{
    try {
        Arduino.killArduinoConnection();

    } catch (Exception ex) {
        Logger.getLogger(Usuario.class.getName()).log(Level.SEVERE, null, ex);
    }

    JOptionPane.showMessageDialog(null, "Bienvenido");
    Switch.Control INGRESO= new Switch.Control();
    INGRESO.banderaconeccion=1;
    INGRESO.velocidad=1;
    INGRESO.iniciar();
    INGRESO.setVisible(true);

    close();

}

if(CAP.equals("VENDEDOR"))
{
//

}if(!CAP.equals("Admin"))&& (!CAP.equals("VENDEDOR"))
{
    JOptionPane.showMessageDialog(this, "El Usuario O la Contraseña Son Incorrectas
Vuelva a Intentar");
    txtusuario.requestFocus();
    LIMPIAR();
}

}catch (SQLException ex) {

```

```

    }
}
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN: initComponents
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    txtusuario = new javax.swing.JTextField();
    btniniciar = new javax.swing.JButton();
    txtcontrasena = new javax.swing.JPasswordField();
    jPanel3 = new javax.swing.JPanel();
    JComboBoxPorts = new javax.swing.JComboBox();
    jButtonRefresh = new javax.swing.JButton();
    jButtonConectar = new javax.swing.JButton();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
    setTitle("INICIAR SESIÓN");
    setBackground(new java.awt.Color(51, 51, 51));
    setResizable(false);
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    jPanel1.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseMoved(java.awt.event.MouseEvent evt) {
            jPanel1MouseMoved(evt);
        }
    });
    jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
    getContentPane().add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -
1, -1));

    jLabel1.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    jLabel1.setForeground(new java.awt.Color(255, 255, 255));
    jLabel1.setText("USUARIO");
    getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(300,
20, -1, 20));

    jLabel2.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    jLabel2.setForeground(new java.awt.Color(255, 255, 255));

```

```

jLabel2.setText("CONTRASEÑA");
getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(300,
50, -1, 20));
getContentPane().add(txtusuario, new
org.netbeans.lib.awtextra.AbsoluteConstraints(380, 20, 100, -1));

btniniciar.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Switch/BOTON_INICIARSESION_opt.png"))); //
NOI18N
btniniciar.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
btniniciar.setEnabled(false);
btniniciar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btniniciarActionPerformed(evt);
    }
});
getContentPane().add(btniniciar, new org.netbeans.lib.awtextra.AbsoluteConstraints(320,
80, 130, 30));
getContentPane().add(txtcontrasena, new
org.netbeans.lib.awtextra.AbsoluteConstraints(380, 50, 100, -1));

jPanel3.setBackground(new java.awt.Color(51, 51, 51));
jPanel3.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jComboBoxPorts.setFont(new java.awt.Font("Century Gothic", 0, 14)); // NOI18N

jButtonRefresh.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jButtonRefresh.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/images/refreshicon.png"))); // NOI18N
jButtonRefresh.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonRefreshActionPerformed(evt);
    }
});

jButtonConectar.setBackground(new java.awt.Color(255, 255, 255));
jButtonConectar.setFont(new java.awt.Font("Century Gothic", 1, 17)); // NOI18N
jButtonConectar.setForeground(new java.awt.Color(0, 153, 204));
jButtonConectar.setText("Conectar");
jButtonConectar.setEnabled(false);
jButtonConectar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonConectarActionPerformed(evt);
    }
}

```



```

        jLabel4.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Switch/Login_37128_opt.png"))); // NOI18N
        getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(200,
20, -1, -1));

        jLabel5.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Switch/19668219_1531438200210398_13829
47556_n.jpg"))); // NOI18N
        jLabel5.setText("jLabel5");
        getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0,
500, 120));

        pack();
    } // </editor-fold> //GEN-END: initComponents

    private void jPanel1MouseClicked(java.awt.event.MouseEvent evt) { //GEN-
FIRST:event_jPanel1MouseClicked

    } //GEN-LAST:event_jPanel1MouseClicked

    private void jButtonConectarActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_jButtonConectarActionPerformed

    if (jButtonConectar.getText().equals("Desconectar")) {
        try {
            Arduino.killArduinoConnection();
            jButtonConectar.setText("Conectar");
            c.enableConnectionPanel(jButtonRefresh, jComboBoxPorts);

        } catch (Exception ex) {
            Logger.getLogger(Usuario.class.getName()).log(Level.SEVERE, null, ex);
        }

    } else {

        try {
            Arduino.arduinoTX(jComboBoxPorts.getSelectedItem().toString(), 9600);
            jButtonConectar.setText("Desconectar");
            c.disableConnectionPanel(jButtonRefresh, jComboBoxPorts);
            this.btniniciar.setEnabled(true);
        } catch (Exception ex) {
            Logger.getLogger(Usuario.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

```

} //GEN-LAST:event_jButtonConectarActionPerformed

private void jButtonRefreshActionPerformed(java.awt.event.ActionEvent evt) //GEN-
FIRST:event_jButtonRefreshActionPerformed
    getPorts();
} //GEN-LAST:event_jButtonRefreshActionPerformed

private void btniniciarActionPerformed(java.awt.event.ActionEvent evt) //GEN-
FIRST:event_btniniciarActionPerformed
    String USUARIO=txtusuario.getText();
    String CONTRASENA=new String(txtcontrasena.getPassword());
    Acceder(USUARIO, CONTRASENA);

    // TODO add your handling code here:
} //GEN-LAST:event_btniniciarActionPerformed

/**
 * @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(Usuario.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {

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

```

```

    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Usuario.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Usuario.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Usuario().setVisible(true);
    }
});
}
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton btniniciar;
private javax.swing.JButton jButtonConectar;
private javax.swing.JButton jButtonRefresh;
private javax.swing.JComboBox jComboBoxPorts;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel3;
private javax.swing.JPasswordField txtcontrasena;
private javax.swing.JTextField txtusuario;
// End of variables declaration//GEN-END:variables
}

```