

HERRAMIENTA SOFTWARE DE APOYO A UN MODELO DE AULA BASADO EN IoT PARA LA ASIGNATURA DE PROGRAMACIÓN EN LAS INGENIERÍAS

JESUS GABRIEL DELGADO LEAL

**PROGRAMA DE INGENIERÍA DE SISTEMAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA,
ELECTRÓNICA, SISTEMAS Y TELECOMUNICACIONES
FACULTAD INGENIERÍAS Y ARQUITECTURA**



UNIVERSIDAD DE PAMPLONA
Pamplona 2021

HERRAMIENTA SOFTWARE DE APOYO A UN MODELO DE AULA BASADO EN IoT PARA LA ASIGNATURA DE PROGRAMACIÓN EN LAS INGENIERÍAS

**Trabajo de grado presentado como requisito para optar al título de
INGENIERO DE SISTEMAS**

Presentado por: JESUS GABRIEL DELGADO LEAL

jegadele@hotmail.com

Director: Ph.D. LUZ MARINA SANTOS JAIMES

lsantos@unipamplona.edu.co



**PROGRAMA DE INGENIERÍA DE SISTEMAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA,
ELECTRÓNICA, SISTEMAS Y TELECOMUNICACIONES
FACULTAD INGENIERÍAS Y ARQUITECTURA
UNIVERSIDAD DE PAMPLONA
Pamplona 2021**

DEDICATORIA

La familia es el pilar fundamental de la sociedad, son los que están en primera línea cuando necesitamos apoyo en cualquier situación o en la toma de una decisión trascendental para nuestras vidas; una de esas decisiones fue emprender este largo camino para formarme como ingeniero de sistemas en la Universidad de Pamplona.

Es por esto que dedico este trabajo a todos los que me apoyaron en esta decisión y en el transcurso de la carrera, a toda mi familia en especial a mis padres, mi hermana Sandra, Blanca, Carlos, Eduardo, mi sobrina Camila y mis demás hermanos que de alguna u otra forma estuvieron presentes y creyeron en mí, gracias por esas muestras de amor, unión y lealtad; sus palabras de aliento, sus buenos deseos y oraciones me dieron fuerzas para no desfallecer cuando me sentí desmotivado y sin alientos.

También agradezco a la familia Unipamplona que hizo parte de esta formación como profesores y compañeros de estudio; especialmente a la profesora Luz Marina mi directora de tesis, quien me acompañó en el desarrollo de este trabajo y algunas materias, agradezco su comprensión, paciencia, consejos y por compartir parte de su gran conocimiento conmigo.

RESUMEN

Se plantea la Industria 4.0 como la cuarta revolución industrial que generará cambios radicales en la sociedad, oportunidades y por supuesto, retos a superar. La Universidad no está ajena a abordar una transformación digital educativa producto de esta revolución, la cual nos lleva a pensar en nuevos modelos educativos. Internet de las cosas (IoT por sus siglas en inglés) permite la utilización de una variedad de dispositivos que utilizados adecuadamente en nuestra vida académica podrían potenciar tanto la calidad de enseñanza, como el aprendizaje; encaminando nuestra Institución hacia la Educación 4.0.

Para hacer frente a los desafíos de la Universidad de Pamplona con el surgimiento de la educación 4.0, este proyecto busca contribuir desarrollando una herramienta software la cual será una aplicación de escritorio que apoyará un proyecto de investigación interno de la Universidad¹, este plantea un modelo de aula basado en IoT que consta de cuatro momentos de aprendizaje: teórico, reto, misión y evaluación; dicha herramienta apoyará solo el momento de reto de este proyecto de investigación y servirá de intermediario entre el estudiante y Arduino, el cual se usará para aplicar conceptos básicos de programación a objetos del mundo real. Todo lo anterior se aplicará a la primera asignatura de los cursos de programación en los programas de ingeniería, y pretende facilitar su aprendizaje, motivando sus estudiantes para que aprendan de una forma creativa y eficiente; aprovechando las bondades de IoT y encaminando así la Universidad de Pamplona hacia la educación 4.0.

Palabras clave: Industria 4.0, educación 4.0, IoT, Arduino, Python.

¹ Proyecto de investigación aprobado en la convocatoria interna de banco de proyectos 2019, “Desarrollo de estrategias de enseñanza aprendizaje en el modelo de educación 4.0 basado en iot para los programas de ingeniería de la Universidad de Pamplona.”

ABSTRACT

Industry 4.0 is proposed as the fourth industrial revolution that will generate radical changes in society, opportunities and of course challenges to overcome. The University is no stranger to tackling an educational digital transformation product of this revolution, which leads us to think about new educational models. Internet of things (IoT for its acronym in English) allows the use of a variety of devices that used properly in our academic life could enhance both the quality of teaching and learning; directing our Institution towards Education 4.0.

To face the challenges of the University of Pamplona with the emergence of education 4.0, this project seeks to contribute developing a software tool which will be a desktop application that will support an internal research project of the University², this project proposes a classroom model based on IoT that consists of four learning moments: theoretical, challenge, mission and evaluation; This tool will support only the challenge moment of this research project and will serve as an intermediary between the student and Arduino, which will be used to apply basic programming concepts to real world objects. All of the above will apply to the first subject of programming courses in engineering programs, and aims to facilitate their learning, motivating their students to learn in a creative and efficient way; taking advantage of the benefits of IoT and thus directing the University of Pamplona towards education 4.0.

Keywords: Industry 4.0, Education 4.0, IoT, Arduino,

² Research project approved in the 2019 project bank internal call, "Development of teaching-learning strategies in the 4.0 education model based on iot for the engineering programs of the University of Pamplona."

Tabla de contenido

1	INTRODUCCIÓN	1
1.1	Planteamiento del problema	1
1.2	Justificación.....	2
1.3	Delimitación.....	3
1.3.1	Objetivo general.....	3
1.3.2	Objetivos específicos.....	3
1.4	Metodología.	3
2	MARCO TEÓRICO Y ESTADO DEL ARTE	4
2.1	Marco teórico	5
2.1.1	Bases y conceptos del proyecto.....	5
2.1.2	Herramientas utilizadas en el proyecto	10
2.1.3	Protocolos usados en el proyecto	12
2.2	Estado del arte.....	19
3	ANÁLISIS Y MODELADO	21
3.1	Investigación de fundamentos teóricos	21
3.2	Propósito del sistema.....	22
3.3	Alcance del sistema.....	22
3.4	Requerimientos funcionales	22
3.4.1	Arquitectura del Sistema de aula IoT	23
3.5	Requerimientos no funcionales	24
3.6	Escenarios	25
3.7	Identificación de casos de uso.....	26
3.7.1	Casos de uso para el rol Docente.....	27
3.7.2	Casos de uso para el rol Estudiante	37
3.8	Interfaz de usuario.....	42
3.9	Elección de las herramientas de desarrollo	46
4	DISEÑO E IMPLEMENTACIÓN.....	47
4.1	Definición de los objetivos de diseño.....	47
4.2	Definición de los almacenes de datos persistentes	47
4.3	Implementación.....	49
4.3.1	Herramientas y tecnologías a usar.....	49
4.3.2	Ejemplo de interfaz gráfica utilizando tkinter	49
4.3.3	Definición de clases del Sistema de aula IoT	52

4.3.4	Interfaz gráfica del sistema de aula IoT	56
5	PRUEBAS	65
5.1	Resultados y análisis de resultados	66
6	CONCLUSIONES	70
7	RECOMENDACIONES Y TRABAJOS FUTUROS	71
8	BIBLIOGRAFÍA.....	72

Índice de ilustraciones

Ilustración 1. Metodología de desarrollo.....	4
Ilustración 2. Diferencias entre educación 1.0, 2.0,3.0 y 4.0.	9
Ilustración 3. Comunicación en paralelo.....	13
Ilustración 4. Protocolo de comunicación serial.	13
Ilustración 5. Protocolo de comunicación serial I2C.	15
Ilustración 6. Esquema del protocolo de comunicación serial I2C.	15
Ilustración 7. Protocolo de comunicación serial SPI.....	16
Ilustración 8. Protocolo de comunicación Firmata.....	17
Ilustración 9. Arquitectura del Sistema de Aula IoT.....	24
Ilustración 10. Diagrama de casos de uso de las funcionalidades según el rol en el Sistema de aula IoT	27
Ilustración 11. Diagrama de casos de uso para la funcionalidad AdministrarGrupos del rol Docente en el Sistema de aula IoT.	28
Ilustración 12. Diagrama de casos de uso para la funcionalidad AdministrarEquipos del rol Docente en el Sistema de aula IoT.	30
Ilustración 13. Diagrama de casos de uso para la funcionalidad CalificarReto del rol Docente en el Sistema de aula IoT.....	34
Ilustración 14. Diagrama de casos de uso para la funcionalidad DatosSolucion del rol Estudiante en el Sistema de aula IoT.	37
Ilustración 15. Diagrama de casos de uso para la funcionalidad SolucionarReto del rol Estudiante en el Sistema de aula IoT.	38
Ilustración 16. Flujo de eventos de la interfaz de usuario.	42
Ilustración 17. Diseño previo de ventana de Loguin para el sistema de aula IoT.....	43
Ilustración 18. Diseño previo de ventana Actualizar usuario y contraseña.....	44

Ilustración 19. Diseño previo de ventana Docente, funcionalidad administrar grupos.	44
Ilustración 20. Diseño previo ventana Docente, funcionalidad administrar equipos.	44
Ilustración 21. Diseño previo de ventana Docente, funcionalidad calificar reto.....	45
Ilustración 22. Diseño previo de ventana Estudiante, funcionalidad datos solución.	45
Ilustración 23. Diseño previo de ventana Estudiante, funcionalidad solucionar reto.....	45
Ilustración 24. Diagrama de clase almacén de datos.	47
Ilustración 25. Modelo Entidad Relación para la base de datos del sistema.	48
Ilustración 26. Modelo Relacional para la base de datos del sistema.	48
Ilustración 27. Creación de una ventana con Tkinter.	49
Ilustración 28. Clase Notebook para crear pestañas en una ventana con tkinter.....	50
Ilustración 29. Añadir pestañas a una ventana con tkinter.	50
Ilustración 30. Ventana con pestañas utilizando tkinter.....	50
Ilustración 31. Agregar widgets a una pestaña con tkinter.....	51
Ilustración 32. Interfaz gráfica del código de la Ilustración 25.....	52
Ilustración 33. Diagrama de clases del diseño del Sistema de aula IoT.....	55
Ilustración 34. Diagrama de flujo del Sistema de Aula IoT.....	55
Ilustración 35. Ventana de inicio del Sistema de aula IoT.....	57
Ilustración 36. Ventana emergente para credenciales de ingreso incorrectas.	57
Ilustración 37. Pestaña actualizar usuario y contraseña.	58
Ilustración 38. Pestaña de la funcionalidad AdministrarGrupos del rol Docente	59
Ilustración 39. Pestaña con la funcionalidad AdministrarEquipos del rol Docente.	60
Ilustración 40. Pestaña de la funcionalidad CalificarReto del rol Docente.	61
Ilustración 41. Pestaña de la funcionalidad DatosSolucion del rol Estudiante.	62
Ilustración 42. Pestaña de la funcionalidad SolucionarReto del rol Estudiante. Fuente: autor.	63
Ilustración 43. Ventana de ejecución del código fuente con el Sistema de aula IoT.	63
Ilustración 44. Pestaña guía de sensores disponibles en el Sistema de aula IoT.....	64

Ilustración 45. Módulo electrónico del Sistemas de aula IoT.	65
Ilustración 46. Esquema módulo electrónico del Sistemas de aula IoT	66
Ilustración 47. Prueba con sensor de temperatura lm35.....	68
Ilustración 48. Prueba con sensor de gas mq135.....	68
Ilustración 49. Prueba número 1 con sensor de luminosidad bh1750.	69
Ilustración 50. Prueba número 2 con sensor de luminosidad bh1750.	69

Índice de tablas

Tabla 1. Caso de uso ingresarUsuario del rol Docente.	28
Tabla 2. Caso de uso mostrarGrupos del rol Docente.	28
Tabla 3. Caso de uso elegirGrupo del rol Docente.....	29
Tabla 4. Caso de uso agregarEstudiante del rol Docente.	29
Tabla 5. Caso de uso quitarEstudiante del rol Docente.....	29
Tabla 6. Caso de uso editarEstudiantes del rol Docente.	30
Tabla 7. Caso de uso de la funcionalidad AdministrarEquipos del rol Docente.	31
Tabla 8. Caso de Uso crearEquipo del rol Docente.	31
Tabla 9. Caso de Uso elegirEquipo del rol Docente.	31
Tabla 10. Caso de Uso editarEquipo del rol Docente.	32
Tabla 11. Caso de Uso eliminarEquipo del rol Docente.	32
Tabla 12. Caso de Uso asignarEstudiante del rol Docente.....	32
Tabla 13. Caso de Uso quitarEstudiante del rol Docente.....	33
Tabla 14. Caso de Uso asignarReto del rol Docente.	33
Tabla 15. Caso de Uso quitarReto del rol Docente.	33
Tabla 16. Caso de Uso calificarReto del rol Docente.	34
Tabla 17. Caso de uso de la funcionalidad CalificarReto del rol Docente. Fuente:autor.....	35
Tabla 18. Caso de Uso verSolucion del rol Docente.....	35
Tabla 19. Caso de Uso verEvidencia del rol Docente.....	35
Tabla 20. Caso de Uso verCodigo del rol Docente.	36
Tabla 21. Caso de Uso calificar del rol Docente.....	36
Tabla 22. Caso de Uso generarReporte del rol Docente.	36
Tabla 23. Caso de uso ingresarUsuario del rol Estudiante.....	37
Tabla 24. Caso de uso integrantesEquipo del rol Estudiante.	38
Tabla 25. Caso de uso datosSolucion del rol Estudiante.....	38

Tabla 26. Caso de uso de la funcionalidad SolucionarReto del rol Estudiante.	39
Tabla 27. Caso de uso descragarReto del rol Estudiante.	39
Tabla 28. Caso de uso seleccionarSensores del rol Estudiante.	40
Tabla 29. Caso de uso ejecutarCodigo del rol Estudiante.	40
Tabla 30. Caso de uso subirEvidencia del rol Estudiante.	41
Tabla 31. Caso de uso enviarSolucion del rol Estudiante.	41

1 INTRODUCCIÓN

Las tecnologías emergentes actualmente tienen un valor muy importante en la educación, por lo cual es necesario que todos los actores involucrados en el proceso de formación las conozcan y se adapten.

Se plantea la Industria 4.0 como la cuarta revolución industrial que generará cambios radicales en la sociedad, oportunidades y por supuesto, retos a superar.

La Universidad no está ajena a abordar una transformación digital educativa y la Educación 4.0 es una respuesta a la necesidad de la evolución tecnológica que pretende formar capital humano con talento, lo que lleva a las Universidades a repensar sobre sus contenidos curriculares y estrategias de enseñanza-aprendizaje, nuevos requerimientos de enseñanza exigen acercar más a los estudiantes a problemas del mundo real, es por esto que se debe trabajar más en competencias del hacer qué sólo del conocer.

Este es un proyecto basado en IoT, el cual utiliza una variedad de dispositivos que usados adecuadamente en nuestra vida académica podrían potenciar tanto la calidad de enseñanza, como el aprendizaje; encaminando nuestra Institución hacia la Educación 4.0.

1.1 Planteamiento del problema

La Universidad de Pamplona en sus programas de ingeniería ofrece cursos como primera asignatura en el área de programación, que permiten despertar en sus estudiantes habilidades para el desarrollo de tareas que ameritan solución algorítmica, los cuales varían su nombre y los semestres en que se dictan según el programa; pero parte de los estudiantes no logran comprenderlos en su totalidad, interesarse o ver su utilidad en la vida real; lo cual les impide aprender de manera creativa y eficiente.

Las tecnologías emergentes actualmente tienen un valor muy importante en la educación, por lo cual es necesario que todos los actores involucrados en el proceso de formación las conozcan y se adapten. Nuevos requerimientos de enseñanza exigen acercar más a los estudiantes a problemas del mundo real, es por esto que se debe trabajar más en competencias del hacer qué sólo del conocer y desarrollar herramientas que conduzcan a una de las facetas de esta evolución tecnológica como lo es la educación 4.0 (Sandoval Carrero, Santos Jaimes, & Acevedo, 2020).

Entonces se plantea la pregunta, ¿Qué beneficio traerá a la Universidad de Pamplona y sus estudiantes una herramienta software que apoya un modelo de aula basado en IoT para facilitar el aprendizaje de la programación en los programas de ingeniería?

1.2 Justificación

La educación 4.0 es una faceta más de la industria 4.0 que lleva a las Universidades a repensar sobre sus contenidos curriculares y estrategias de enseñanza-aprendizaje (Sandoval Carrero, Santos Jaimes, & Acevedo, 2020).

Con el avance de la tecnología y los cambios que está sufriendo la humanidad y su entorno académico, es necesario estar a la vanguardia de la innovación, e IoT es una gran oportunidad para hacerlo. Proporcionar una herramienta software que apoya un modelo de aula basado en IoT y que permite aplicar conceptos básicos de programación a objetos del mundo real, facilita el aprendizaje de la primera asignatura en esta área, lo cual ayuda a mejorar el aprendizaje de sus estudiantes; logrando capturar el interés por el curso al percibir su utilidad en la vida real, permitiéndoles aprender de forma creativa y eficiente, lo cual proporcionará una experiencia más nutrida.

1.3 Delimitación

1.3.1 Objetivo general

Implementar una herramienta software de apoyo a un modelo de aula basado en IoT, para la primera asignatura de programación en las ingenierías.

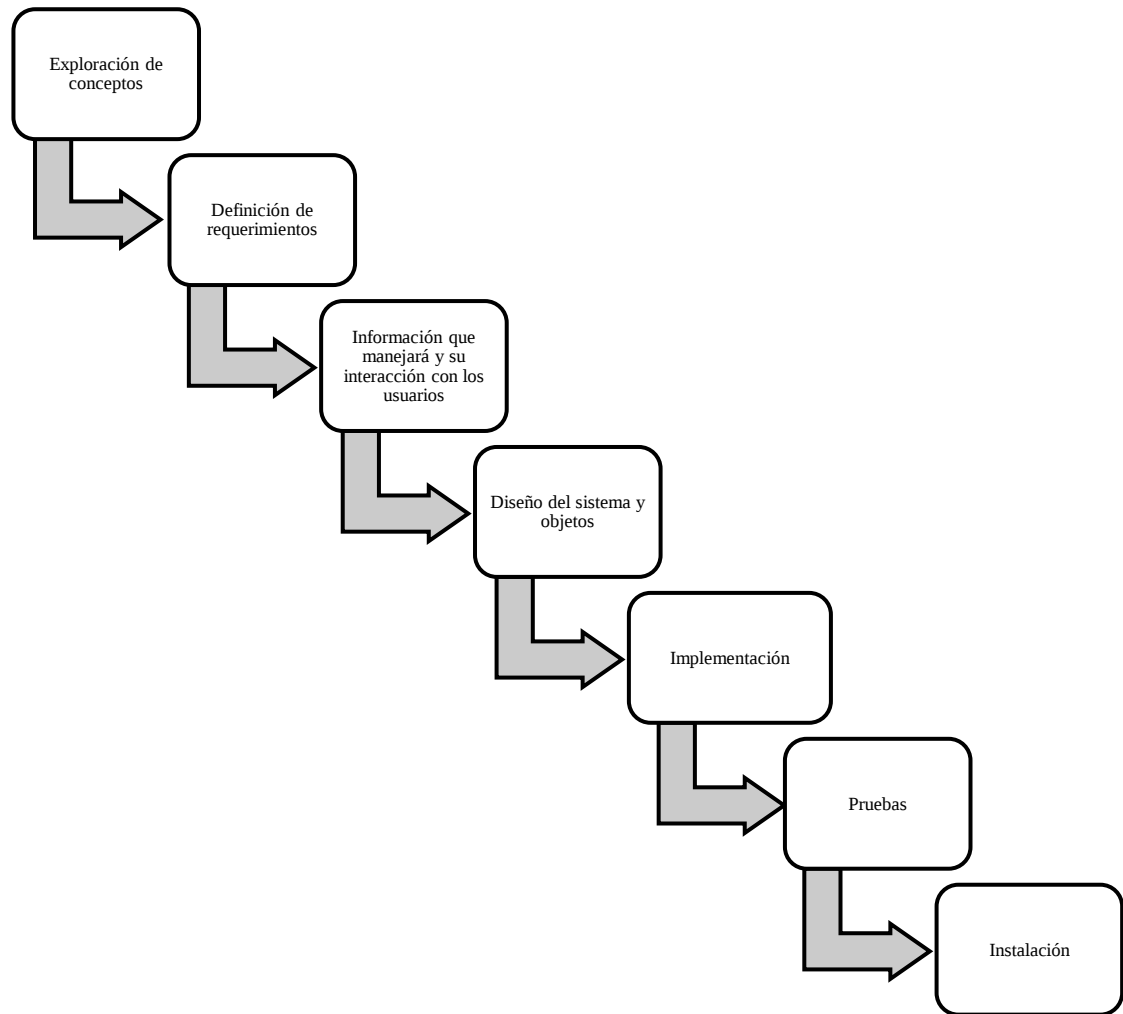
1.3.2 Objetivos específicos

- Hacer un análisis detallado de la herramienta software, posterior a la definición de requerimientos y características que debe cumplir.
- Diseñar la herramienta software y sus diversos componentes de forma tal que satisfaga los requerimientos y características definidos.
- Implementar la herramienta software en el lenguaje de programación Python.
- Comprobar el correcto funcionamiento de la herramienta software mediante pruebas.

1.4 Metodología.

Para la realización de este proyecto se decidió adoptar como metodología de desarrollo el modelo de ciclo de vida en cascada, ya que este es un enfoque centrado en actividades que se ejecutan secuencialmente y es el que más se ajusta al cronograma de actividades. Su característica principal es la verificación constante que evita que cada actividad introduzca requerimientos no deseados o elimine los obligatorios. En la Ilustración 1 se muestra cómo se aplicó este modelo al proyecto.

Se seleccionó el UML (lenguaje de modelado unificado [OMG, 1998]) como notación principal para este proyecto, debido a que tiene una semántica bien definida, proporciona un espectro de notaciones para la representación de diferentes aspectos de un sistema y ha sido aceptado como notación estándar en la industria, lo que evita malas interpretaciones y ambigüedad.



*Ilustración 1. Metodología de desarrollo.
Fuente: (Bruegge & H. Dutoit, 2002).*

2 MARCO TEÓRICO Y ESTADO DEL ARTE

Los avances tecnológicos están cambiando el mercado laboral, según (Florez Olvera, Guzmán Gamez, Martinez Barragán, Ibarra Cruz, & Albear Cortez, 2019), la Industria 4.0 es la cuarta revolución industrial que generará cambios radicales en la sociedad, oportunidades y retos a superar, y la Educación 4.0 es una respuesta a la necesidad de la evolución tecnológica que pretende formar capital humano con talento para competir en el mercado laboral que surge de la Industria 4.0.

Esta cuarta revolución industrial abarca un gran número de tecnologías emergentes entre ellas IoT, la cual permite la utilización de una variedad de dispositivos que utilizados adecuadamente en la vida académica podrían potenciar tanto la calidad de enseñanza, como el aprendizaje.

Un modelo de aula basado en IoT es un tipo de aula que aprovecha las bondades de esta tecnología emergente, aplicando el uso de estos dispositivos al proceso de aprendizaje, logrando que se haga de forma creativa y eficiente.

Este proyecto apoya ese modelo de aula basado en IoT mediante la creación de una aplicación de escritorio, la cual actuará como intermediario entre el estudiante y Arduino para hacer uso de esos dispositivos, y aplicar conceptos básicos de programación por medio del lenguaje Python; de esta forma este proyecto se basa en IoT para encaminar la Universidad de Pamplona hacia la educación 4.0.

Todos estos conceptos y otros relacionados se explican a continuación en el marco teórico, en el estado del arte se presentan algunos proyectos e investigaciones que se basan o utilizan IoT.

2.1 Marco teórico

2.1.1 Bases y conceptos del proyecto

- **Industria 4.0**

El mundo a través de su historia ha sufrido diversas transformaciones y revoluciones que han incluido progresos significativos en la ciencia, la educación, la industria y muchos otros campos; lo cual ha ido modificando el estilo de vida y forma de trabajo de la humanidad.

Según (Basco, Beliz, Coatz, & Garnero, 2018) la economía global transita una nueva fase que se caracteriza por la digitalización y la conectividad. Tecnologías como IoT, computación en la nube, big data, inteligencia artificial, aprendizaje automático, robótica, nanotecnología, impresión 3D, entre otras, refuerzan la importancia de la industria 4.0 a partir de la fabricación de productos personalizados e inteligentes.

El término industria 4.0 se refiere a un nuevo modelo de organización y de control de la cadena de valor a través del ciclo de vida del producto y a lo largo de los sistemas de fabricación apoyado y hecho posible por las tecnologías de la información.

En definitiva, se trata de la aplicación a la industria del modelo IoT. Todos estos términos tienen en común el reconocimiento de que los procesos de fabricación se encuentran en un proceso de transformación digital, una "revolución industrial" producida por el avance de las tecnologías de la información y, particularmente, de la informática y el software. (del Val Roman, 2016)

- **Internet de las cosas (IoT)**

En la comunidad científica existen diversos conceptos para Internet de las cosas (IoT por sus siglas en inglés), pero con base en (Arcos, 2014), IoT trata de una red que interconecta objetos físicos valiéndose del Internet. Los mencionados objetos se valen de sistemas embebidos, o lo que es lo mismo, hardware especializado que le permite no solo la conectividad a Internet, sino que además programa eventos específicos en función de las tareas que le sean dictadas remotamente.

Los sistemas embebidos tratan de chips y circuitos que comparados con, por ejemplo, un Smartphone, podrían parecer muy rudimentarios, pero que cuentan con todas las herramientas necesarias para cumplir labores especializadas muy específicas.

No hay un tipo específico de objetos conectados a IoT. En lugar de eso se les puede clasificar como objetos que funcionan como sensores y objetos que realizan acciones activas. Claro, hay los que cumplen ambas funciones de manera simultánea.

En cualquier caso, el principio es el mismo y la clave es la operación remota. Cada uno de los objetos conectados a Internet tiene una IP específica y mediante esa IP puede ser accedido para recibir instrucciones. Así mismo, puede contactar con un servidor externo y enviar los datos que capture.

- **Educación 4.0**

La industria 4.0 está transformando las relaciones productivas, económicas y comerciales; estos cambios se ven incluso en los entornos educativos, lo que está obligando a revisar los métodos de enseñanza y del proceso de aprendizaje. Estas necesidades y cambios que tiene que cubrir la educación, es lo que se conoce como Educación 4.0. (Perez Romero, Rivera Zárate, & Hernandez Bolaños, 2019)

La Educación 4.0 es una respuesta a la necesidad de la evolución tecnológica que pretende formar capital humano con talento para competir en el mercado laboral que surge de la cuarta revolución industrial (Industria 4.0). En otras palabras, su origen está asociado a una demanda de los empleadores o del sector empresarial; la educación 4.0 carece de una definición teórica y operativa, por lo que suele asociarse a estrategias de aprendizaje emergentes, cuyo fundamento teórico son el conectivismo, el constructivismo y el enfoque educativo basado en competencias. (Florez Olvera, Guzmán Gamez, Martinez Barragán, Ibarra Cruz, & Albear Cortez, 2019)

- **Diferencias entre educación 1.0, 2.0, 3.0, y 4.0**

Como se sabe la educación ha sufrido transformaciones a lo largo de la historia casi siempre ocasionado por la innovación tecnológica y las necesidades de cambio que esta genera, trayendo consigo cambios en los modelos educativos y los paradigmas de enseñanza.

La educación 1.0 es un proceso unidireccional donde el conocimiento que se dispone de los libros seleccionados para el curso, es impartido por los profesores hacia sus alumnos recluidos en un aula y estos trabajan en sus contenidos que se caracterizan por ser estáticos, es decir que no se actualizan y se define como un aprendizaje pasivo y lineal; esta educación 1.0 se caracteriza por centrarse en la evaluación a través de exámenes y el trabajo individual. (Rosique)

La relación todos con todos, que conlleva la puesta en marcha de un modelo de educación 2.0 que altera las relaciones comunicativas tradicionales centradas en un emisor y un receptor cambia por completo el proceso unidireccional de la educación 1.0, presenta una filosofía y una práctica de la educación y de la comunicación basada en el diálogo y en la participación que no requiere solo de tecnologías, sino de un cambio de actitudes y de concepciones. (Aparici, 2011).

La educación 3.0 es aquella en la que los mismos aprendices desempeñarán un papel de creadores del conocimiento compartido a través de las tecnologías de la información generando un aprendizaje autodirigido (Rosique).

Por su parte, la Educación 4.0 se basa en las principales tendencias de innovación y cambio. Los aprendizajes de la revolución 4.0 se centran en las competencias del siglo XXI, tales como la autodirección, la autoevaluación y el trabajo en equipo, es la

respuesta a la necesidad de dar al estudiante el talento para competir en la industria 4.0 con el uso de la tecnología. (UNIDAD POLITÉCNICA PARA LA EDUCACIÓN VIRTUAL, 2018)

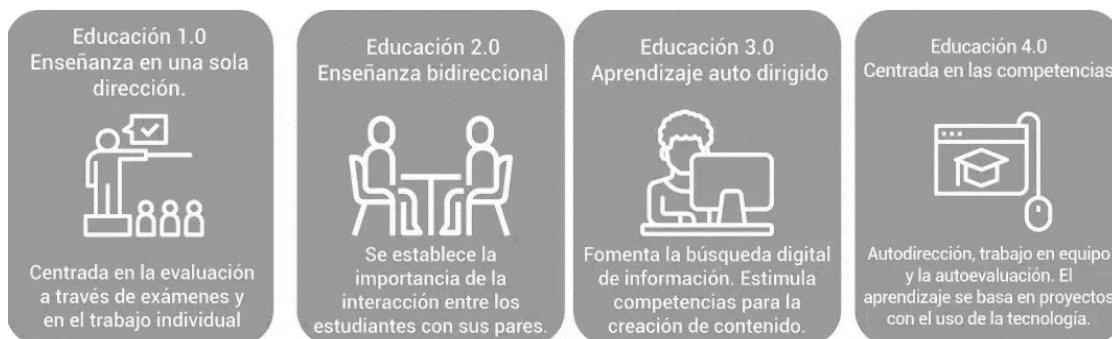


Ilustración 2. Diferencias entre educación 1.0, 2.0, 3.0 y 4.0.

Fuente: <https://docente.4-0.ipn.mx/index.php/educacion-4-0/#:~:text=La%20educaci%C3%B3n%20avanz%C3%B3%20de%201.0,tendencias%20de%20innovaci%C3%B3n%20y%20cambio.>

0/#:~:text=La%20educaci%C3%B3n%20avanz%C3%B3%20de%201.0,tendencias%20de%20innovaci%C3%B3n%20y%20cambio.

- **Universidad 4.0**

Según Pedroza (Pedroza flores, 2018), con el uso de las nuevas tecnologías de la comunicación y con las nuevas tecnologías del aprendizaje, las universidades experimentan cambios profundos y a partir de la transformación digital, estas han dejado de ser lo que eran y se han convertido en un organismo inteligente.

La industria 4.0 lleva a las Universidades a repensar sobre sus contenidos curriculares y estrategias de enseñanza-aprendizaje. Nuevos requerimientos de enseñanza exigen acercar más a los estudiantes a problemas del mundo real y a las tecnologías con que se encontrarán en las empresas. Se debe trabajar más en competencias del hacer qué sólo del conocer.

Es preciso reformar las organizaciones e instituciones establecidas, comprometerse con los nuevos actores emergentes y apoyar a los ciudadanos, cuyas vidas pueden verse afectadas por los cambios que se avecinan. (Pedroza flores, 2018)

- **Aula basada en IoT**

IoT es una tecnología emergente que refuerza la importancia de la Industria 4.0, esta tecnología emergente ofrece una gran variedad de dispositivos que usados adecuadamente en nuestra vida académica podrían potenciar el proceso de enseñanza-aprendizaje. La educación 4.0 es una faceta de la Industria 4.0 y pretende formar capital humano con talento competitivo para esta nueva revolución industrial.

Un modelo de aula basado en IoT, es un tipo de aula que aprovecha las bondades de esta tecnología emergente y propone aplicar conceptos básicos de programación en los dispositivos que ofrece IoT, encaminando así la universidad hacia la Educación 4.0

2.1.2 Herramientas utilizadas en el proyecto

- **Lenguaje de programación:**

Cada ordenador según su diseño interpreta un conjunto de instrucciones elementales, no obstante, para facilitar la tarea del programador se dispone de los lenguajes de programación y estos están determinados por un agregado de palabras claves y símbolos, las cuales deben cumplir unas normas gramaticales para generar las instrucciones u órdenes que expresa cada programa. Las reglas escritas deben tener sintaxis y semántica lo que genera un orden y relación entre las palabras, así como dar significado a las mismas (Rodríguez Sala, y otros, 2003).

En ese orden de ideas, un lenguaje de programación es un lenguaje artificial para la construcción de programas informáticos y se clasifican según el nivel de abstracción, el paradigma de programación, el propósito, la forma de ejecutarse, entre otros, pero se conoce que también se dividen en alto, medio y bajo nivel.

- **Python**

Python es un lenguaje de programación que fue creado a principios de la década de 1990 por Guido van Rossum en Stichting Mathematisch Centrum en los Países Bajos como sucesor de un lenguaje llamado ABC. Guido sigue siendo el autor principal de Python, aunque incluye muchas contribuciones de otros. Administrado por la *Python Software Foundation*, posee una licencia de código abierto, denominada *Python Software Foundation License*. Este es un lenguaje interpretado, es decir, que se ejecuta directamente sin necesidad de compilarse previamente en lenguaje de máquina y requiere de un programa interprete que lea la instrucción del código en tiempo real y lo ejecute. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional (Python Software Foundation, Python Documentation, 2021).

- **Arduino**

Arduino es una compañía de desarrollo de software y hardware libres que diseña y manufactura placas electrónicas, que utilizan un microcontrolador reprogramable con una serie de pines que permiten establecer conexiones entre el controlador y los diferentes sensores, nació en el Ivrea Interaction Design Institute como una herramienta fácil para la creación rápida de prototipos, dirigida a estudiantes sin experiencia en electrónica y programación. Tan pronto como llegó a una comunidad más amplia, la placa Arduino comenzó a cambiar para adaptarse a las nuevas necesidades y desafíos, diferenciando su oferta desde placas simples de 8 bits hasta productos para aplicaciones de IoT, impresión 3D, entornos integrados entre otros (Arduino, 2021).

Existen diversos tipos de placas entre las cuales destacan: Arduino Uno, Arduino DUE, Arduino Leonardo, Arduino Mega, Arduino Micro, Arduino Nano, Arduino Yun, Arduino FIO, Arduino MKR1000, Arduino TRE, Arduino Zero, Arduino Ethernet, entre muchas otras. (Ordiales, 2017)

- **Tk/Tcl**

Proporciona un conjunto de herramientas robusto e independiente de Python para administrar ventanas. Disponible para desarrolladores a través del paquete tkinter y sus extensiones, los módulos tkinter.tix y tkinter.ttk (Python Software Foundation, Interfaces gráficas de usuario con Tk, 2021).

- **Tkinter**

Tkinter es una librería que viene instalada por defecto con el lenguaje de programación Python, el paquete tkinter es una capa delgada orientada a objetos encima de Tcl/Tk, para usarlo no tiene que escribir ningún código Tcl. El paquete tkinter (Tk interface) es la interfaz estándar de Python para el kit de herramientas Tk. Tanto Tk como tkinter están disponibles en la mayoría de las plataformas Unix, así como en los sistemas Windows. Se considera un estándar para la interfaz gráfica de usuario (GUI) para Python y es el que viene por defecto con la instalación para Microsoft Windows (Python Software Foundation, Interfaces gráficas de usuario con Tk, 2021).

2.1.3 Protocolos usados en el proyecto

Como es bien sabido para la comunicación, ya sea en el mundo natural o en el campo de la informática y las redes entre dos actores o más, es necesario el uso de protocolos para que esta comunicación se pueda llevar a cabo.

Un protocolo de comunicación es un sistema de reglas que permiten que dos o más entidades de un sistema se comuniquen entre ellas para transmitir información por algún medio (ELECTRONOBS en Español, 2020), es como un acuerdo previo para establecer la comunicación.

- **Protocolo de comunicación serial**

Es una forma de transmitir los datos seguidos uno tras otro utilizando una sola conexión más la referencia a tierra, nació como alternativa a los problemas de sincronización y cableado que ofrecía la comunicación en paralelo, ya que esta por cada bit necesita una conexión más la referencia a tierra como se aprecia en la Ilustración 3.

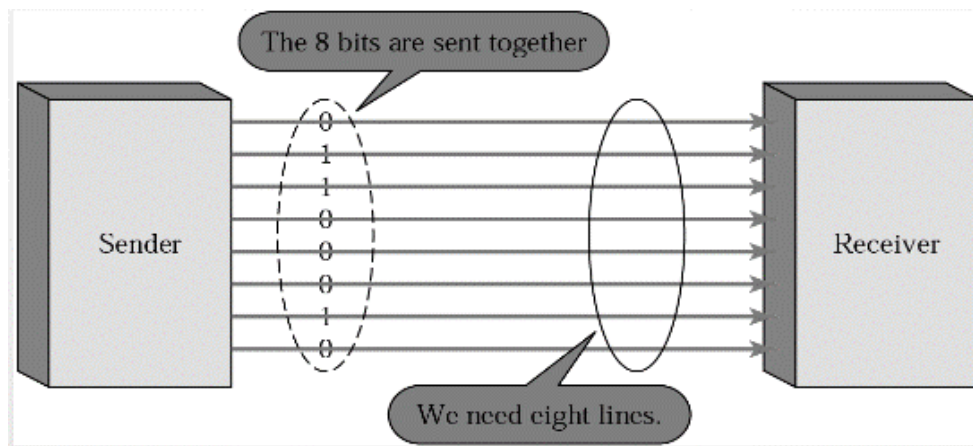


Ilustración 3. Comunicación en paralelo.
Fuente: (Turnero, s.f.)

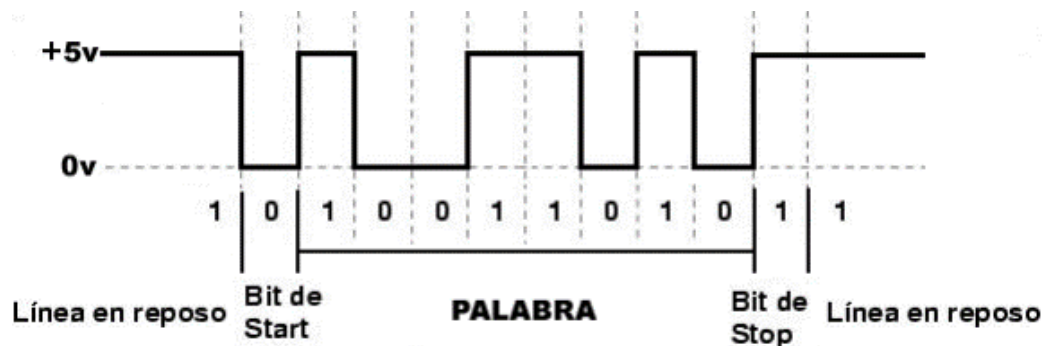


Ilustración 4. Protocolo de comunicación serial.
Fuente: (itq.edu.mx, s.f.)

En la Ilustración 4 se presenta un ejemplo de comunicación en serie: la comunicación UART(Universal Asynchronous Receiver-Transmitter) que es un tipo de comunicación en serie asíncrona y que utiliza la siguiente configuración entre el transmisor (TX) y el receptor (RX).

- ✓ **Velocidad de transmisión:** indica el número de bits que se transfieren por segundo y se mide en baudios, las velocidades más comunes son 115200, 9600 y 4800.
- ✓ **Longitud de los datos:** es la cantidad de bits que tiene el mensaje que se envía, las cantidades más comunes son 5, 7, y 8 bits.
- ✓ **Bit de inicio y parada:** por lo general la señal siempre está en alto (1) y el bit de inicio está representado por un pulso bajo (0), así el receptor sabrá cuando comienzan los datos. El bit de parada será representado por un pulso en alto (1), así después de recibida la cantidad de bits esperada, el receptor sabrá donde terminó el mensaje.

Una comunicación serial es asíncrona si no utiliza reloj, por lo que necesita configurar la velocidad de transmisión, la longitud de cada dato, el bit de inicio y parada, así el receptor sabrá donde empieza y termina cada dato.

La comunicación síncrona utiliza un pulso de reloj por cada bit para saber dónde comienza y termina cada dato, por lo que, si se envía un mensaje con 16 bits, se necesitan 16 pulsos de reloj, entonces será 16 veces más lento, lo cual es el principal inconveniente de la comunicación en serie (ELECTRONOBS en Español, 2020).

- **Protocolo de comunicación serial I2C**

El protocolo de comunicación I2C (Inter-Integrated Circuit) desarrollado por Philips semiconductors, es un tipo de comunicación serial síncrona. Este protocolo permite

la comunicación con varios dispositivos a la vez utilizando solo dos conexiones más la referencia a tierra, un cable es el de datos enviados y tiene el nombre de SDA (Serial Data) y el segundo cable es el reloj SCL (Serial Clock), también es necesario especificar la longitud de los datos y la frecuencia de reloj tanto en el transmisor como en el receptor.

Debido a que el protocolo I2C permite la comunicación con múltiples dispositivos, existe el dispositivo maestro y los dispositivos esclavos, cada dispositivo esclavo tiene una dirección diferente; así el transmisor primero envía la dirección y luego el dato (ELECTRONOBS en Español, 2020).

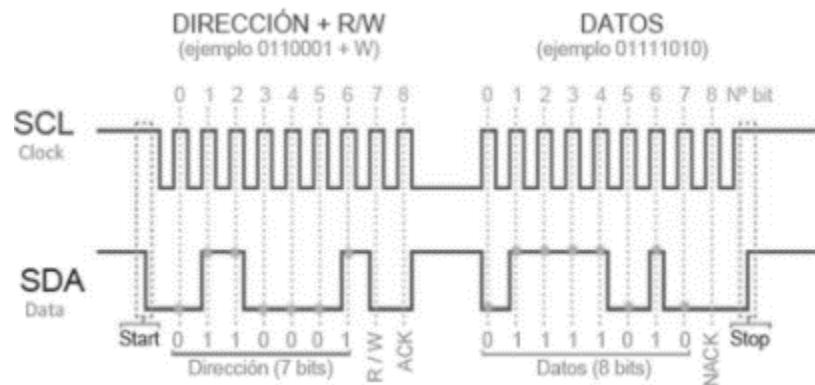


Ilustración 5. Protocolo de comunicación serial I2C.

Fuente: (jecrespon, s.f.)

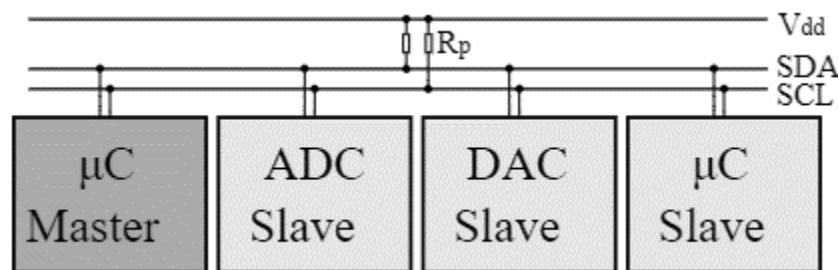


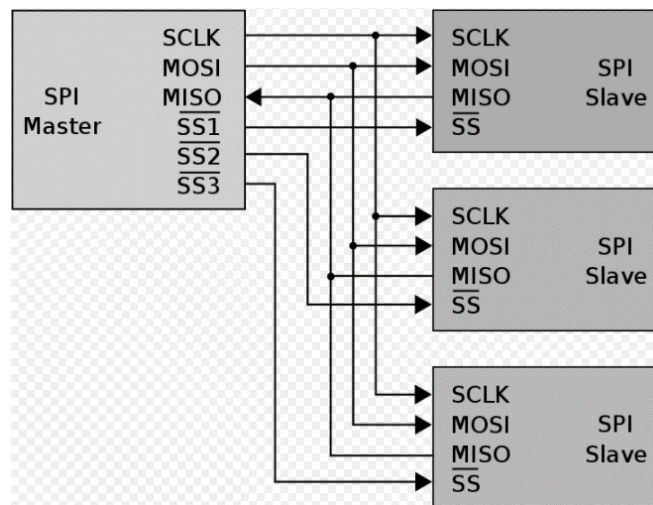
Ilustración 6. Esquema del protocolo de comunicación serial I2C.

Fuente: (jecrespon, s.f.)

- **Protocolo de comunicación SPI**

El protocolo SPI (Serial Peripheral Interface) es un tipo de comunicación síncrona y la transmisión y sincronización de datos se realiza por medio de cuatro conexiones o señales más la referencia a tierra:

- ✓ **SCLK** (Clock): Es el pulso que marca la sincronización. Con cada pulso de este reloj, se lee o se envía un bit. También llamado TAKT (en alemán).
- ✓ **MOSI** (Master Output Slave Input): Salida de datos del Master y entrada de datos al Esclavo.
- ✓ **MISO** (Master Input Slave Output): Salida de datos del Esclavo y entrada al Master.
- ✓ **SS/Select**: Para seleccionar un Esclavo, o para que el Master le diga al Esclavo que se active. También llamada SSTE



*Ilustración 7. Protocolo de comunicación serial SPI.
Fuente: (ELECTRONOBS en Español, 2020)*

Al igual que I2C existe un transmisor maestro y un receptor esclavo, pero el maestro también puede recibir datos utilizando el cable de MISO, por lo que no se usa dirección de esclavo como en I2C sino que se utiliza el SS/Select, el cual el maestro pondrá en bajo con un bit en 0 y luego se envía la señal de reloj y las señales de datos,

por lo que si se desea conectar más dispositivos esclavos se necesitará una conexión de SS/Select para cada módulo (ELECTRONOBS en Español, 2020).

La ventaja es que la comunicación es full dúplex, es decir que se pueden enviar y recibir datos al mismo tiempo, lo que no se puede hacer con I2C ya que es una comunicación simple.

- **Firmata**

Firmata es un protocolo genérico para la comunicación entre microcontroladores y software instalado en un ordenador. Este protocolo se puede implementar en cualquier arquitectura de microcontroladores, así como en cualquier paquete de software. (Firmata | Aprendiendo Arduino, s.f.)

El objetivo de firmata es permitir el control completo de una tarjeta de desarrollo como por ejemplo Arduino, desde un software instalado en un ordenador, sin escribir una sola línea de código de Arduino. En la Ilustración 8 se muestra como interactúan sensores conectados a Arduino con un software cualquiera instalado en un ordenador por medio del protocolo de comunicación Firmata.

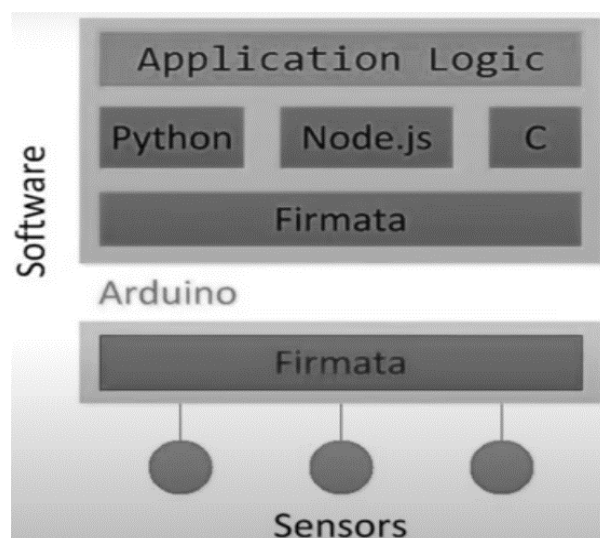


Ilustración 8. Protocolo de comunicación Firmata.
Fuente: <https://www.youtube.com/watch?v=NIEgaMNKj8w>

Hay dos modos de uso de firmata. Un modelo es usar los métodos ofrecidos por la librería firmata dentro del sketch³ para enviar y recibir datos entre el Arduino y el software ejecutándose en el ordenador. Por ejemplo, mandar el valor leído de una entrada analógica.

El segundo modelo y más común es cargar en Arduino el sketch de propósito general llamado “StandardFirmata” o alguna de sus variaciones como “StandardFirmataPlus” o “StandardFirmataEthernet” y usar el software en el ordenador para interactuar con Arduino.

- **PyFirmata**

PyFirmata es una librería de Python que implementa una interfaz para el protocolo Firmata diseñado para comunicar un microcontrolador con un software en un computador, permite controlar Arduino usando solo el lenguaje de programación Python sin escribir una sola línea de código en Arduino (Python Software Foundation, pyFirmata, 2021), es muy usado ya que proporciona muchos métodos simples y efectivos para comunicarse con el protocolo Firmata en los modos análogo, digital y SERVO⁴; pero provee soporte limitado para la comunicación I2C.

- **PyMata**

Es una librería de Python que permite la comunicación con Firmata y ofrece soporte para el protocolo de comunicación I2C (Python Software Foundation, PyMata, 2021).

³ Un Sketch es un archivo que contiene el código fuente de un programa de Arduino y debe estar guardado con la extensión .ino y dentro de una carpeta con el mismo nombre del archivo.

⁴ Es el modo para configurar en Python con PyFirmata el pin, el pulso mínimo, el pulso máximo y el ángulo de un servomotor el cual es un dispositivo que se conecta a Arduino.

2.2 Estado del arte

TITULO	AUTOR	APORTE
Prototipo de un Aula Inteligente aplicando Internet de las Cosas y Modelos de Aprendizaje Profundo.	(Pacheco Gonzáles, Flores Avila, Cano Fuentes, & Tena Chávez, 2018)	El Aprendizaje Profundo tiene un notable desempeño para analizar las grandes cantidades de datos de la actual etapa de transformación digital de la sociedad. Muchas de estas aplicaciones utilizan la enorme infraestructura disponible del Cómputo en la Nube. Sin embargo, para el caso del Internet de las Cosas, existen aspectos para los cuales esta alternativa no siempre es la más efectiva, eficiente y segura, sobre todo en lo que concierne a la privacidad de los datos, consumo de energía y saturación del ancho de banda de la red.
Sistema electrónico con aplicación IoT para monitoreo facial que brinde estimadores de desconcentración del estudiante universitario en el aula a escala de laboratorio.	(Alvear Puertas, 2017)	El sistema de monitoreo facial para determinar la desconcentración en estudiantes universitarios, es un sistema que tiene como objetivo brindar estimadores de desconcentración de los estudiantes de la carrera de Ingeniería en Electrónica y Redes de Comunicación de la Universidad Técnica del Norte.
TIC e IoT para crear un entorno de aprendizaje inteligente para estudiantes de institutos de educación en la India	(Rahman, Himanshi, Deep, & Rahman, 2016)	Los institutos educativos han ido más allá de la forma tradicional de aprendizaje al aceptar las últimas tecnologías y las tendencias modernas, haciendo que el aprendizaje sea más innovador. Las tecnologías interactivas y los servicios de comunicación instantánea han transformado los métodos adoptados para la educación en un medio más interactivo, visualizable y accesible.
Fomento al uso de las TIC en la Universidad Cooperativa de Colombia sede Villavicencio, fortaleciendo la investigación y extensión,	(Murcia Cucaita, 2015)	Uso de las TIC para fomentar la investigación y extensión en los estudiantes del programa ingeniería de sistemas y desde el aula se promueve la investigación, en donde el software y hardware libre en especial android y arduino toman un papel protagonista, fortaleciendo los espacios de aprendizaje aplicando el concepto de IoT.

mediante redes sociales, android, arduino e IoT		
El Internet de las cosas en la educación CS: desafíos actuales y potencial futuro	(Burd B. B., 2018)	Constituye un proyecto cooperativo entre un conjunto de universidades y presenta una revisión general sobre los desafíos tecnológicos que enfrenta la Educación Superior.
Cursos, contenidos y herramientas para internet de las cosas en educación informática.	(Burd B. B., 2017)	Considera cuatro enfoques principales que los educadores informáticos están utilizando para integrar los conceptos y cursos de IoT en sus planes de estudio, resume las opciones y los desafíos relacionados con la enseñanza de IoT y describe algunas herramientas IoT que los docentes pueden utilizar.
Alineación del plan de estudios de pregrado para el aprendizaje de IoT en una Facultad de Ciencias de la Computación	(Guerra Guerra, 2017)	Presenta el plan de estudios de pregrado de la Facultad de Ciencias de la Computación, para que los estudiantes puedan aprender temas de IoT, de acuerdo con las necesidades de la industria peruana.
Internet de las cosas en las instituciones de educación superior	(Rueda Rueda, Manrique, & Cabrera Cruz, 2017)	En este artículo se mencionan las diferentes interacciones entre IoT y las Instituciones de Educación Superior, y define la IoT como herramienta pedagógica.
Internet de las cosas en educación CS: actualización de currículos y exploración de la pedagogía	(Burd B. B., 2018)	Sus resultados proporcionaron un punto de entrada para los educadores con la configuración de un nuevo curso de IoT.

3 ANÁLISIS Y MODELADO

3.1 Investigación de fundamentos teóricos

Antes de iniciar con todas las actividades de este proyecto se realizó una ardua investigación y preparación para su desarrollo con todo lo referente a la comunicación entre el lenguaje de programación Python y Arduino para lo cual se encontró la librería pyfirmata de Python, así como la creación de interfaces gráficas en Python haciendo uso de la librería tkinter y aspectos varios referentes a Python en cuanto a la programación orientada a objetos, la modulación y la creación de paquetes entre muchos otros conceptos necesarios. Se recurrió a diversas fuentes como artículos y páginas de internet y tutoriales de YouTube, fuentes de las cuales se mencionan algunas a continuación.

- Programa y controla con Python tu Arduino (Diéguez)
- Como programar a tu Arduino (Ortiz)
- Aprendiendo Arduino (Firmata | Aprendiendo Arduino, s.f.)
- Curso Python impartido por píldoras informáticas (pildorasinformaticas, s.f.)
- Python 3.x impartido por Sprogramacion (Sprogramacion, s.f.)

También durante el transcurso del desarrollo de las actividades fue necesario acudir a otras fuentes entre muchas otras como Stack Overflow el cual es un sitio de preguntas y respuestas para programadores profesionales y aficionados, la documentación de Python y la documentación de tkinter, esto con el fin de ir solucionando inconvenientes que se presentaban en la marcha.

3.2 Propósito del sistema

Este sistema tiene el propósito de facilitar el proceso de aprendizaje de los estudiantes que apenas empiezan a programar, actuando como intermediario entre el estudiante y Arduino para aplicar conceptos básicos de programación en problemas del mundo real.

3.3 Alcance del sistema

Está dirigido a estudiantes que ven por primera vez programación y se aplicará a la primera asignatura de los cursos en esta área de los programas de ingeniería.

3.4 Requerimientos funcionales

Se describen interacciones entre el sistema y su ambiente, en forma independiente a su implementación. El ambiente incluye al usuario y cualquier otro sistema con el cual interactúe (Bruegge & H. Dutoit, 2002).

El sistema independientemente del rol, debe permitir a los usuarios actualizar su usuario y contraseña si así lo desea.

Este software permitirá a un estudiante ingresar sus datos y acceder al sistema para proceder a dar solución a un reto que previamente ha asignado el docente al equipo y que consiste en un ejercicio práctico, para la solución del reto el software tendrá conexión a una base de datos que contiene la información de los posibles dispositivos que se puedan usar en el reto, el sistema permitirá al estudiante si así lo desea subir una evidencia ya sea en video, foto o documento escrito. El sistema llevará cuenta de la fecha y hora de inicio, así como de su finalización y la cantidad de intentos de ejecución del código en lenguaje de programación Python, una vez solucionado el reto el software guardará los datos de la solución en la base de datos.

Esta base de datos también contiene información de los estudiantes, los docentes, los cursos, los grupos y los equipos; así como de los retos.

El software en el momento que se proceda a dar solución al reto, debe automáticamente identificar el puerto en que se encuentra conectada la placa Arduino al computador.

El docente al igual que el estudiante podrá ingresar al sistema mediante usuario y contraseña, puede crear y eliminar equipos en un grupo, así como asignar o quitar estudiantes y retos a un equipo. Podrá ver la solución dada por un equipo a un reto y dar una calificación, también permitirá al docente si así lo desea generar un reporte de la solución a un reto hecha por un equipo.

3.4.1 Arquitectura del Sistema de aula IoT

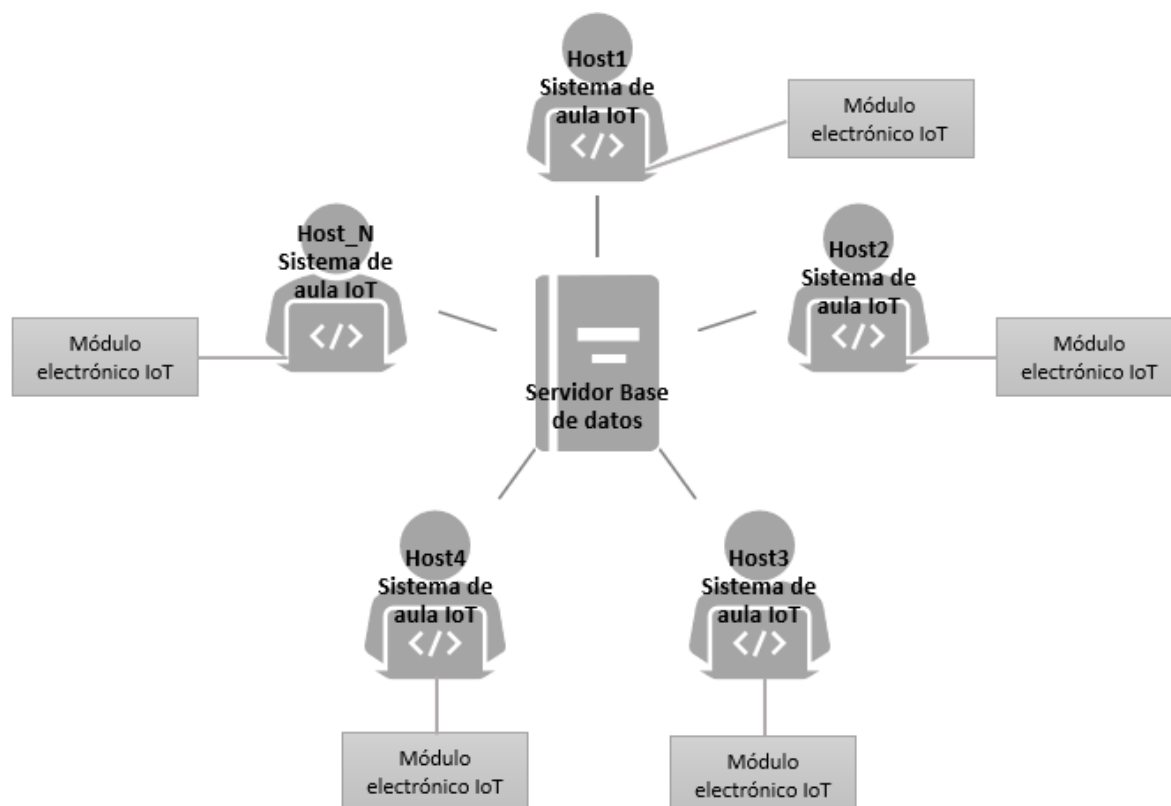
Como ya se mencionó, los requerimientos funcionales describen la interacción del sistema y su ambiente, siendo el ambiente los usuarios u otro sistema.

Este sistema estará conectado a una base de datos que manejará toda la información de cursos, grupos, equipos, estudiantes, docentes, retos y dispositivos, dicha base de datos estará ubicada en un servidor y se accederá a ella por medio de una red de área local LAN⁵.

En cada computador de esta LAN estará instalado el Sistema de aula IoT, y a su vez, cada computador tendrá conectado un módulo electrónico que se presenta en la Ilustración 45 y 46 de este documento, este módulo electrónico está compuesto por diversos dispositivos y sensores que están conectados a una tarjeta Arduino Mega y con los cuales el Sistema de

⁵ LAN (Local Area Network), es una red de computadoras conectadas entre si que abarca un área reducida como una casa, salón de clases, oficinas o edificios.

aula IoT interactúa, guardando información en la base de datos de los retos solucionados por los equipos.



*Ilustración 9. Arquitectura del Sistema de Aula IoT.
Fuente: autor.*

3.5 Requerimientos no funcionales

Se describen aspectos del sistema visibles por el usuario que no se relacionan en forma directa con el comportamiento funcional del sistema (Bruegge & H. Dutoit, 2002).

El software deberá manejar tres roles: administrador, docente y estudiante. Se mostrará una pantalla al usuario para elegir el rol y posteriormente ingresar con un usuario y contraseña, dependiendo del rol con el que ingrese tendrá acceso a ciertas funcionalidades.

- **Rol administrador:** tendrá acceso a los datos de los cursos y sus grupos, así como sus docentes y estudiantes, los cuales podrá crear, editar y eliminar; también a los

retos que existen y los dispositivos, podrá crear nuevos retos, pero no podrá asignarlos a un equipo ni crear equipos.

- **Rol docente:** tendrá acceso solo a la información de los grupos en los que dicta clase, así como de sus estudiantes, podrá crear, eliminar y editar equipos a los cuales podrá asignar o quitar estudiantes y retos que ya están montados en una base de datos, así como dar una calificación a la solución de un reto hecha por un equipo y ver datos de la solución como fecha de inicio y fin, cantidad de intentos de ejecución y sensores utilizados.
- **Rol estudiante:** tendrá acceso solo a la información del equipo al que pertenece y a los retos que se les haya asignado, a los cuales deberá dar solución la cual posteriormente se enviará para que el sistema la suba a la base de datos, también podrá ver datos de las soluciones de los retos hechas por su equipo.

3.6 Escenarios

Un escenario es una descripción narrativa de lo que la gente hace y experimenta cuando trata de utilizar sistemas y aplicaciones de computadora. (Bruegge & H. Dutoit, 2002)

1. Sí al loguearse con alguno de los roles el usuario o contraseña son incorrectas deberá informarse del error e impedirse el acceso.
2. Sí el estudiante no se encuentra asignado a un equipo el sistema no presentará ninguna opción al estudiante y le informará de la situación.
3. Sí el estudiante no conecta la placa Arduino al computador e intenta ejecutar la solución dada al reto, el sistema deberá lanzar una excepción indicando problemas con el puerto para la conexión de la placa.

4. Sí el estudiante hace uso o referencia en su código a una variable de dispositivo no definida para el reto asignado, el sistema deberá informarle del error.
5. Sí el docente intenta asignar un reto a un equipo que no tiene estudiantes asignados el sistema no deberá permitirlo e informarle del error.
6. Sí el docente intenta eliminar un equipo que tiene estudiantes asignados el sistema no deberá permitirlo e informarle del error.

3.7 Identificación de casos de uso

Los casos de uso describen el comportamiento del sistema desde el punto de vista de los actores. Un diagrama de casos de uso está orientado a determinar el qué, y no el cómo de lo que va a realizar el sistema, los actores son entidades externas que interactúan con el sistema y puede ser un usuario u otro sistema (Bruegge & H. Dutoit, 2002). En la Ilustración 10 se muestra un diagrama de casos de uso general para el sistema de aula IoT que describe las funcionalidades a las que tiene acceso un actor según el rol con el que ingrese.

La [sección 3.7.1](#) y [3.7.2](#) describe el flujo de información entre actores y casos de uso y la dependencia o sucesión de un caso de uso a otro.

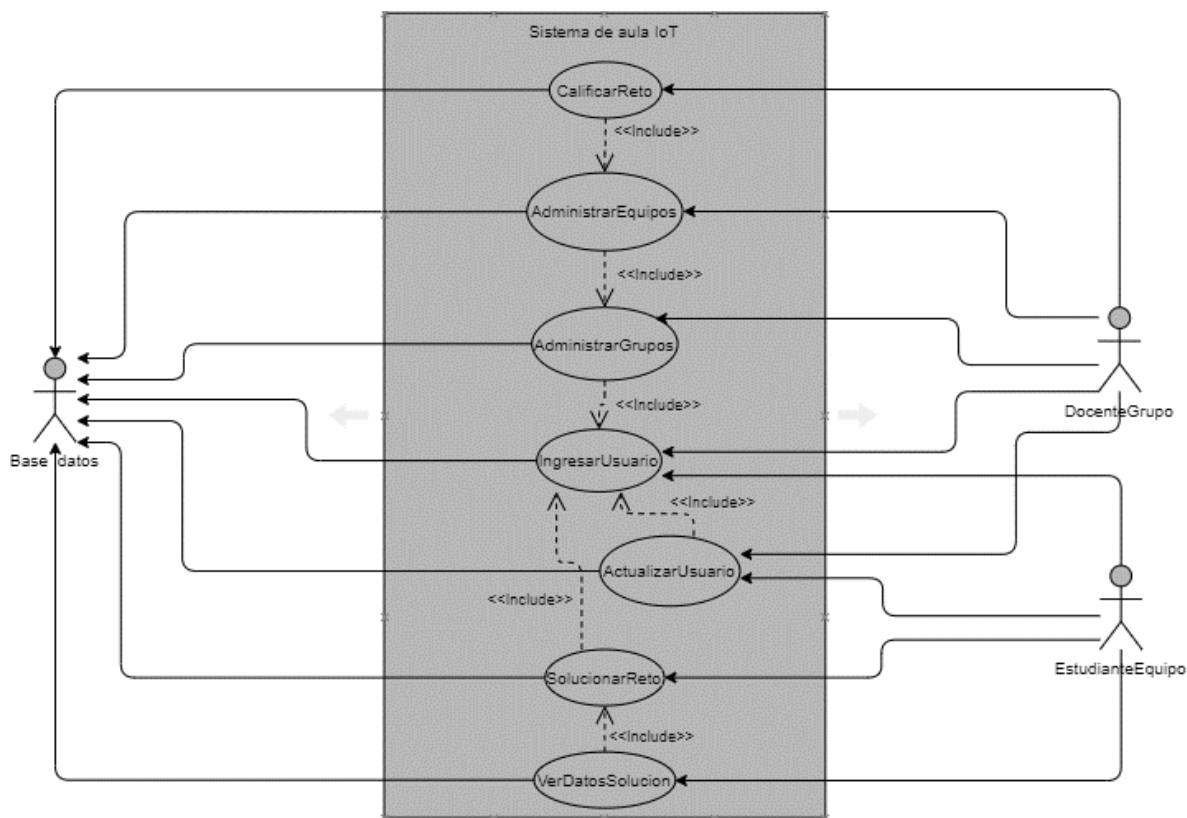


Ilustración 10. Diagrama de casos de uso de las funcionalidades según el rol en el Sistema de aula IoT
Fuente: autor.

3.7.1 Casos de uso para el rol Docente

La Ilustración 10 muestra un diagrama de casos de uso con las funcionalidades a las que tiene acceso cada actor según el rol con el que ingrese al sistema de aula IoT. A continuación, se presentan los diagramas de casos de uso de cada funcionalidad a las que tiene acceso un docente y que se representan en las Ilustraciones 11, 12 y 13, y de las cuales se desprenden más casos de uso que son descritos en las tablas que le siguen a cada funcionalidad (Ilustración 11: tablas 1 a 6, Ilustración 12: tablas 7 a 16, Ilustración 13: tablas 17 a 22).

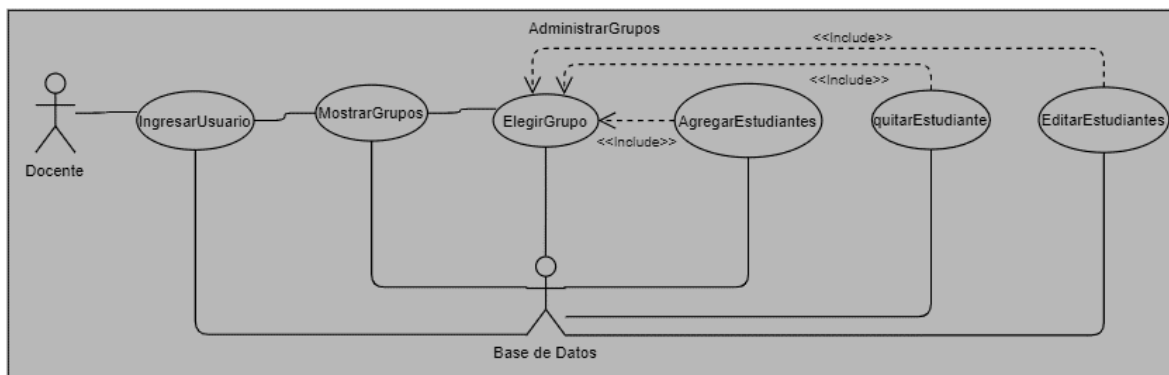


Ilustración 11. Diagrama de casos de uso para la funcionalidad AdministrarGrupos del rol Docente en el Sistema de aula IoT.

Fuente: autor.

Tabla 1. Caso de uso ingresarUsuario del rol Docente.

Nombre del caso de uso	ingresarUsuario.
Actor participante	Iniciado por: Docente.
Condición inicial	Consulta a: Base de datos.
Flujo de eventos	1. El Docente inicia el sistema. 2. El sistema presenta una ventana con los campos para ingresar usuario y contraseña. 3. El Docente ingresa los datos. 4. Sí los datos son correctos el sistema presentará una nueva ventana con los grupos que el docente tiene a cargo. 5. Sí los datos son incorrectos el sistema informará del error.
Condición de salida	
Requerimiento especial	El sistema consulta a la base de datos para validarlos y mostrarlos en pantalla.

Tabla 2. Caso de uso mostrarGrupos del rol Docente.

Nombre del caso de uso	mostrarGrupos.
Actor participante	Iniciado por: ingresarUsuario.
Condición inicial	Consulta a: Base de datos.
Flujo de eventos	1. El Docente inicia el sistema e ingresa usuario y contraseña correctamente. 2. Una vez logueado el docente, el sistema de aula IoT muestra en una nueva ventana los grupos que tiene a cargo.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

Tabla 3. Caso de uso elegirGrupo del rol Docente.

Nombre del caso de uso	elegirGrupo
Actor participante	Iniciado por: Docente. Consulta a: Base de datos.
Condición inicial	1. El Docente elige un grupo de los mostrados en el caso de uso MostrarGrupos y presiona el botón Mostrar.
Flujo de eventos	2. En la misma pestaña del caso de uso MostrarGrupos se muestran los estudiantes que pertenecen al grupo.
Condición de salida	3. El sistema genera una nueva pestaña en la misma ventana para administrar los equipos del grupo.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

Tabla 4. Caso de uso agregarEstudiante del rol Docente.

Nombre del caso de uso	agregarEstudiantes.
Actor participante	Iniciado por: Docente. Consulta a: Base de datos.
Condición inicial	1. El Docente elige un grupo de los mostrados en el caso de uso MostrarGrupos y presiona el botón Agregar estudiante.
Flujo de eventos	2. En una nueva pestaña se muestran los campos para ingresar los datos del nuevo estudiante. 3. El docente ingresa los datos y presiona el botón agregar.
Condición de salida	4. En la misma pestaña del caso de uso MostrarGrupos se muestran los estudiantes que pertenecen al grupo.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

Tabla 5. Caso de uso quitarEstudiante del rol Docente.

Nombre del caso de uso	quitarEstudiante.
Actor participante	Iniciado por: Docente. Consulta a: Base de datos.
Condición inicial	1. El Docente elige un grupo de los mostrados en el caso de uso MostrarGrupos.
Flujo de eventos	2. El Docente elige un estudiante. 3. El Docente presiona el botón Eliminar.
Condición de salida	4. En la misma pestaña del caso de uso MostrarGrupos se muestran los estudiantes que pertenecen al grupo.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

Tabla 6. Caso de uso editarEstudiantes del rol Docente.

Nombre del caso de uso	editarEstudiantes.
Actor participante	Iniciado por: Docente. Consulta a: Base de datos.
Condición inicial	1. El Docente elige un grupo de los mostrados en el caso de uso MostrarGrupos.
Flujo de eventos	2. El Docente elige un estudiante del grupo y presiona el botón Editar. 3. En una nueva pestaña se muestran los campos con los datos del estudiante para ser editados. 4. El docente ingresa los datos y presiona el botón Actualizar.
Condición de salida	5. En la misma pestaña del caso de uso MostrarGrupos se muestran los estudiantes que pertenecen al grupo.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

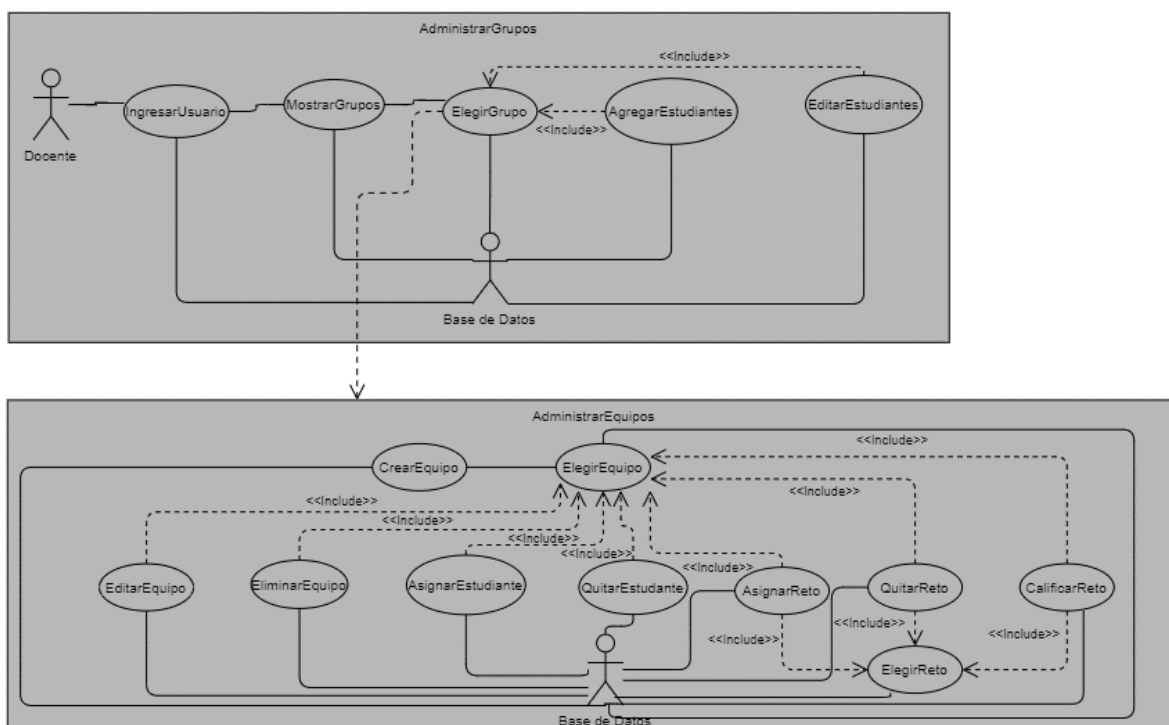


Ilustración 12. Diagrama de casos de uso para la funcionalidad AdministrarEquipos del rol Docente en el Sistema de aula IoT.

Fuente: autor.

Tabla 7. Caso de uso de la funcionalidad AdministrarEquipos del rol Docente.

Nombre del caso de uso	AdministrarEquipos.
Actor participante	Iniciado por: Caso de Uso ElegirGrupo de la funcionalidad AdministrarGrupos. Consulta a: Base de datos.
Condición inicial	1. El Docente elige un grupo de los mostrados en el caso de uso MostrarGrupos y presiona el botón Mostrar.
Flujo de eventos	2. En una nueva pestaña se muestra la funcionalidad AdministrarEquipos con los equipos existentes en el grupo y los retos disponibles.
Condición de salida	3. En la misma pestaña de la funcionalidad AdministrarEquipos se generan los demás casos de uso mostrados en la Ilustración 10.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

Tabla 8. Caso de Uso crearEquipo del rol Docente.

Nombre del caso de uso	crearEquipo.
Actor participante	Iniciado por: Docente. Consulta a: Base de datos.
Condición inicial	1. El Docente ingresa el nombre del nuevo equipo y presiona el botón Crear equipo.
Flujo de eventos	2. En la misma pestaña de la funcionalidad AdministrarEquipos se muestran los equipos existentes en el grupo.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

Tabla 9. Caso de Uso elegirEquipo del rol Docente.

Nombre del caso de uso	elegirEquipo.
Actor participante	Iniciado por: Docente. Consulta a: Base de datos.
Condición inicial	1. El Docente elige un equipo de los mostrados en la funcionalidad AdministrarEquipos y presiona el botón Mostrar.
Flujo de eventos	2. En la misma pestaña de la funcionalidad AdministrarEquipos se muestran los estudiantes que integran el equipo.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

Tabla 10. Caso de Uso editarEquipo del rol Docente.

Nombre del caso de uso	editarEquipo.
Actor participante	Iniciado por: Docente. Consulta a: Base de datos.
Condición inicial	1. El Docente elige un equipo de los mostrados en la funcionalidad AdministrarEquipos.
Flujo de eventos	2. El Docente ingresa el nuevo nombre del equipo. 3. El Docente presiona el botón Cambiar nombre.
Condición de salida	4. En la misma pestaña de la funcionalidad AdministrarEquipos se muestran los equipos existentes en el grupo.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

Tabla 11. Caso de Uso eliminarEquipo del rol Docente.

Nombre del caso de uso	eliminarEquipo.
Actor participante	Iniciado por: Docente. Consulta a: Base de datos.
Condición inicial	1. El Docente elige un equipo de los mostrados en la funcionalidad AdministrarEquipos.
Flujo de eventos	2. El Docente presiona el botón Eliminar.
Condición de salida	3. En la misma pestaña de la funcionalidad AdministrarEquipos se muestran los equipos existentes en el grupo.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

Tabla 12. Caso de Uso asignarEstudiante del rol Docente.

Nombre del caso de uso	asignarEstudiante.
Actor participante	Iniciado por: Docente. Consulta a: Base de datos.
Condición inicial	1. El Docente elige un equipo de los mostrados en la funcionalidad AdministrarEquipos.
Flujo de eventos	2. El Docente elige uno de los estudiantes del grupo. 3. El Docente presiona el botón Asignar estudiante.
Condición de salida	4. En la misma pestaña de la funcionalidad AdministrarEquipos se muestran los estudiantes que integran el equipo.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

Tabla 13. Caso de Uso quitarEstudiante del rol Docente.

Nombre del caso de uso	quitarEstudiante.
Actor participante	Iniciado por: Docente.
	Consulta a: Base de datos.
Condición inicial	1. El Docente elige un equipo de los mostrados en la funcionalidad AdministrarEquipos.
Flujo de eventos	2. El Docente elige uno de los estudiantes del grupo. 3. El Docente presiona el botón Quitar estudiante.
Condición de salida	4. En la misma pestaña de la funcionalidad AdministrarEquipos se muestran los estudiantes que integran el equipo.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

Tabla 14. Caso de Uso asignarReto del rol Docente.

Nombre del caso de uso	asignarReto.
Actor participante	Iniciado por: Docente.
	Consulta a: Base de datos.
Condición inicial	1. El Docente elige un equipo de los mostrados en la funcionalidad AdministrarEquipos.
Flujo de eventos	2. El Docente elige uno de los retos existentes. 3. El Docente presiona el botón Asignar reto.
Condición de salida	4. En la misma pestaña de la funcionalidad AdministrarEquipos se muestran los equipos existentes en el grupo con los retos asignados.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

Tabla 15. Caso de Uso quitarReto del rol Docente.

Nombre del caso de uso	quitarReto.
Actor participante	Iniciado por: Docente.
	Consulta a: Base de datos.
Condición inicial	1. El Docente elige un equipo de los mostrados en la funcionalidad AdministrarEquipos.
Flujo de eventos	2. El Docente elige uno de los retos existentes. 3. El Docente presiona el botón Quitar reto.
Condición de salida	4. En la misma pestaña de la funcionalidad AdministrarEquipos se muestran los equipos existentes en el grupo con los retos asignados.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

Tabla 16. Caso de Uso calificarReto del rol Docente.

Nombre del caso de uso	calificarReto.
Actor participante	Iniciado por: Docente.
Condición inicial	Consulta a: Base de datos.
Flujo de eventos	1. El Docente elige un equipo de los mostrados en la funcionalidad AdministrarEquipos.
Condición de salida	2. El Docente elige uno de los retos existentes. 3. El Docente presiona el botón calificar reto. 4. En la misma ventana de la funcionalidad AdministrarEquipos se muestra una nueva pestaña con la funcionalidad CalificarReto, los datos de la solución del reto y los demás retos asignados al equipo.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

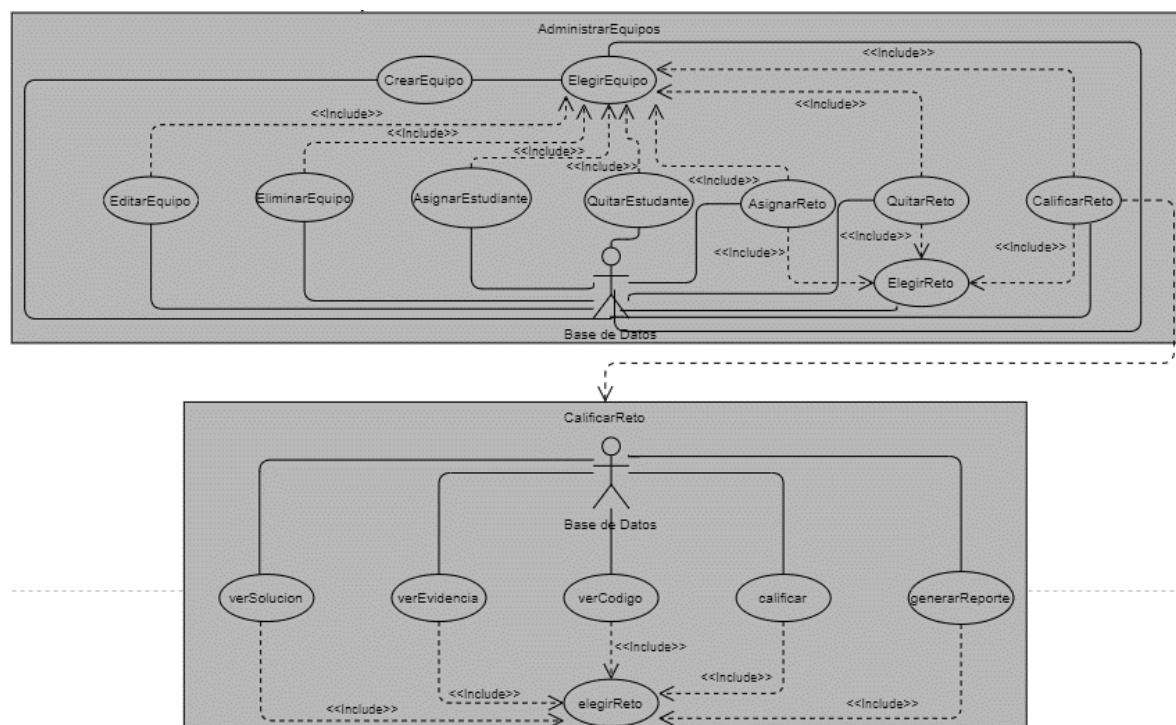


Ilustración 13. Diagrama de casos de uso para la funcionalidad CalificarReto del rol Docente en el Sistema de aula IoT.
Fuente: autor.

Tabla 17. Caso de uso de la funcionalidad CalificarReto del rol Docente. Fuente:autor.

Nombre del caso de uso	CalificarReto.
Actor participante	Iniciado por: Caso de Uso calificarReto de la funcionalidad AdministrarEquipos.
Condición inicial	Consulta a: Base de datos. 1. El Docente elige un equipo de los mostrados en la funcionalidad AdministrarEquipos.
Flujo de eventos	2. El Docente elige uno de los retos existentes. 3. El Docente presiona el botón calificar reto.
Condición de salida	4. En la misma ventana de la funcionalidad AdministrarEquipos se muestra una nueva pestaña con la funcionalidad CalificarReto y los demás casos de uso mostrados en la Ilustración 13.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

Tabla 18. Caso de Uso verSolucion del rol Docente.

Nombre del caso de uso	verSolucion.
Actor participante	Iniciado por: Docente.
Condición inicial	Consulta a: Base de datos. 1. El Docente elige un reto de los asignados al equipo.
Flujo de eventos	2. El Docente presiona el botón Mostrar.
Condición de salida	3. En la misma pestaña de la funcionalidad CalificarReto se muestran los datos de la solución al reto dada por el equipo.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

Tabla 19. Caso de Uso verEvidencia del rol Docente.

Nombre del caso de uso	verEvidencia.
Actor participante	Iniciado por: Docente.
Condición inicial	Consulta a: Base de datos. 1. El Docente elige un reto de los asignados al equipo.
Flujo de eventos	2. El Docente presiona el botón Ver evidencia.
Condición de salida	3. Si existe se descarga el archivo con la evidencia de la solución al reto.
Requerimiento especial	El sistema consulta a la base de datos para descargar el archivo en una carpeta creada por el propio sistema de aula IoT.

Tabla 20. Caso de Uso verCodigo del rol Docente.

Nombre del caso de uso	verCodigo.
Actor participante	Iniciado por: Docente. Consulta a: Base de datos.
Condición inicial	1. El Docente elige un reto de los asignados al equipo.
Flujo de eventos	2. El Docente presiona el botón Ver código.
Condición de salida	3. Si existe se descarga el archivo con el código solución en lenguaje de programación Python.
Requerimiento especial	El sistema consulta a la base de datos para descargar el archivo en una carpeta creada por el propio sistema de aula IoT.

Tabla 21. Caso de Uso calificar del rol Docente.

Nombre del caso de uso	calificar.
Actor participante	Iniciado por: Docente. Consulta a: Base de datos.
Condición inicial	1. El Docente elige un reto de los asignados al equipo.
Flujo de eventos	2. El Docente ingresa la calificación en un campo de texto. 3. Si lo considera necesario el Docente ingresa una observación en un campo de texto.
Condición de salida	4. El Docente presiona el botón Calificar. 5. El sistema actualiza la calificación en la base de datos y muestra los datos de la solución en pantalla.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

Tabla 22. Caso de Uso generarReporte del rol Docente.

Nombre del caso de uso	generarReporte.
Actor participante	Iniciado por: Docente. Consulta a: Base de datos.
Condición inicial	1. El Docente elige un reto de los asignados al equipo.
Flujo de eventos	2. El Docente presiona el botón Generar reporte.
Condición de salida	3. Se descarga un archivo tipo csv con los datos de la solución como curso, grupo, reto, equipo, integrantes del equipo, fecha de inicio y fin, cantidad de intentos y calificación.
Requerimiento especial	El sistema consulta a la base de datos para descargar el archivo en una carpeta creada por el propio sistema de aula IoT.

3.7.2 Casos de uso para el rol Estudiante

En la Ilustración 10 se presenta un diagrama de casos de uso con las funcionalidades a las que tiene acceso cada actor según el rol con el que ingrese al sistema de aula IoT. A continuación, se muestra los diagramas de casos de uso de cada funcionalidad a las que tiene acceso un estudiante y que se representan en las Ilustraciones 14 y 15, de las cuales se desprenden más casos de uso que son descritos en las tablas que le siguen a cada funcionalidad (Ilustración 14: tablas 23 a 25, Ilustración 15: tablas 26 a 31).

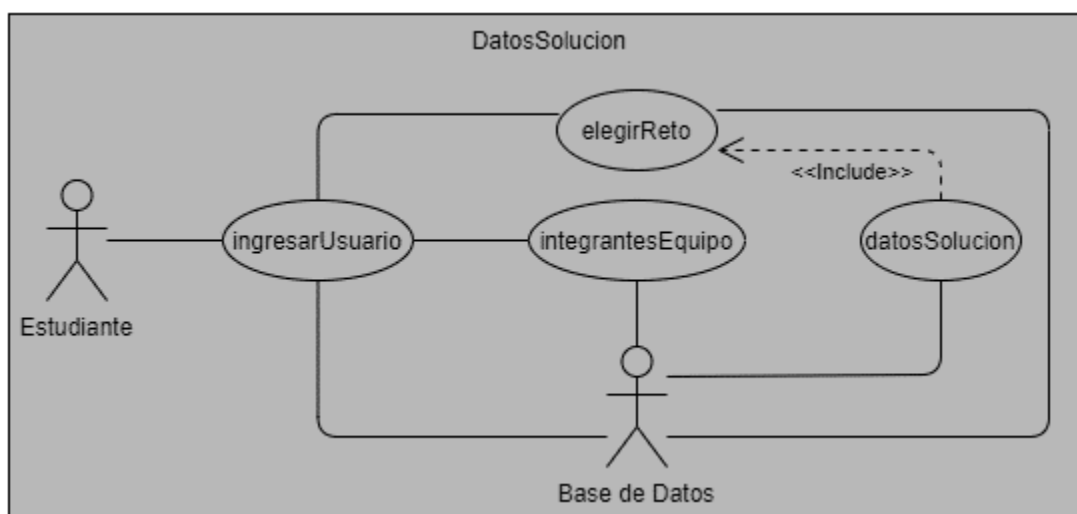


Ilustración 14. Diagrama de casos de uso para la funcionalidad DatosSolucion del rol Estudiante en el Sistema de aula IoT.

Fuente: autor.

Tabla 23. Caso de uso ingresarUsuario del rol Estudiante.

Nombre del caso de uso	ingresarUsuario.
Actor participante	Iniciado por: Estudiante.
Condición inicial	Consulta a: Base de datos.
Flujo de eventos	1. El Estudiante inicia el sistema. 2. El sistema presenta una ventana con los campos para ingresar usuario y contraseña. 3. El Estudiante ingresa los datos. 4. Si los datos son correctos el sistema presentará una nueva ventana con demás casos de uso si no informara del error.
Condición de salida	
Requerimiento especial	El sistema consulta a la base de datos para validarlos y mostrarlos en pantalla.

Tabla 24. Caso de uso integrantesEquipo del rol Estudiante.

Nombre del caso de uso	integrantesEquipo.
Actor participante	Iniciado por: ingresarUsuario. Consulta a: Base de datos.
Condición inicial	1. El Estudiante inicia el sistema e ingresa usuario y contraseña correctamente.
Flujo de eventos	2. Una vez logueado el estudiante, el sistema de aula IoT muestra una nueva ventana con los integrantes del equipo al que el estudiante pertenece.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

Tabla 25. Caso de uso datosSolucion del rol Estudiante.

Nombre del caso de uso	datosSolucion.
Actor participante	Iniciado por: Estudiante. Consulta a: Base de datos.
Condición inicial	1. El Estudiante elige uno de los retos asignados al equipo.
Flujo de eventos	2. El estudiante presiona el botón Mostrar. 3. En la misma pestaña del caso de uso integrantesEquipo se muestran los datos de la solución del reto.
Condición de salida	4. Es deseable que se muestren datos como fecha de inicio y fin, cantidad de intentos, sensores usados y calificación obtenida.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla.

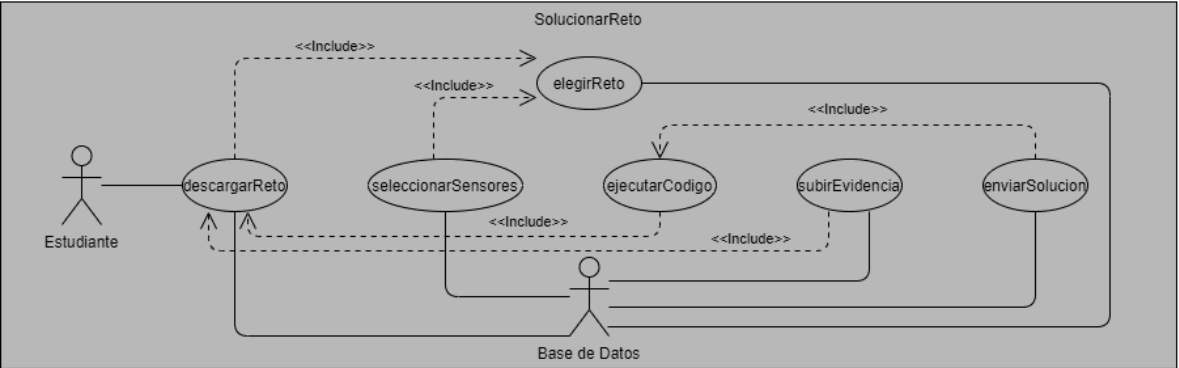


Ilustración 15. Diagrama de casos de uso para la funcionalidad SolucionarReto del rol Estudiante en el Sistema de aula IoT.

Fuente: autor.

Tabla 26. Caso de uso de la funcionalidad SolucionarReto del rol Estudiante.

Nombre del caso de uso	SolucionarReto.
Actor participante	Iniciado por: Caso de Uso ingresarUsuario de la funcionalidad DatosSolucion.
	Consulta a: Base de datos.
Condición inicial	1. El Estudiante ingresa el usuario y contraseña correctamente.
Flujo de eventos	2. El sistema presenta una pestaña en la misma ventana de la funcionalidad DatosSolucion. 3. En esta pestaña se presentan los casos de uso asociados con esta funcionalidad.
Condición de salida	4. Se presentan solo los retos que el equipo tiene asignado, un campo para escribir código y ejecutarlo y las posibilidades de elegir sensores, subir evidencia y enviar la solución.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla, así como para cargarlos en ella.

Tabla 27. Caso de uso descargarReto del rol Estudiante.

Nombre del caso de uso	descargarReto.
Actor participante	Iniciado por: Estudiante.
	Consulta a: Base de datos.
Condición inicial	1. El Estudiante elige uno de los retos que el equipo tiene asignado.
Flujo de eventos	2. El Estudiante presiona el botón Descargar reto.
Condición de salida	3. Se presenta una ventana emergente para que el Estudiante abra el reto en su visor pdf de su preferencia.
Requerimiento especial	El sistema consulta a la base de datos para descargar el archivo en una carpeta creada por el propio sistema de aula IoT.

El reto que el estudiante descargue es un archivo en formato pdf que contiene un enunciado de un problema práctico que simula una situación de la vida real y con la cual el estudiante se podría enfrentar como futuro ingeniero en su ámbito profesional.

Tabla 28. Caso de uso seleccionarSensores del rol Estudiante.

Nombre del caso de uso	seleccionarSensores.
Actor participante	Iniciado por: Estudiante.
Condición inicial	Consulta a: Base de datos.
Flujo de eventos	1. El Estudiante elige uno de los retos que el equipo tiene asignado. 2. El Estudiante elige los sensores disponibles y que considere necesarios para la solución del reto. 3. El Estudiante presiona el botón Seleccionar. 4. Se presentará un campo de texto editable con los sensores que el estudiante haya seleccionado.
Condición de salida	5. Al momento de enviar la solución del reto los sensores seleccionados serán cargados en la base de datos como parte de la solución del reto.
Requerimiento especial	El sistema consulta a la base de datos para mostrarlos en pantalla, así como para cargarlos en ella.

Tabla 29. Caso de uso ejecutarCodigo del rol Estudiante.

Nombre del caso de uso	ejecutarCodigo.
Actor participante	Iniciado por: Estudiante.
Condición inicial	1. El Estudiante ha descargado uno de los retos asignados y que desea solucionar.
Flujo de eventos	2. El Estudiante escribe su código fuente en un campo habilitado para ello dentro de la interfaz. 3. El Estudiante presiona el botón Ejecutar.
Condición de salida	4. Se presentará una ventana emergente en forma de consola donde se mostrará la ejecución del código y la finalización de la ejecución.
Requerimientos especiales	1. El sistema informará al equipo mediante ventanas emergentes si existe algún error en el código. 2. El sistema guardará el código de la última ejecución en una carpeta creada por el mismo Sistema de aula IoT. 3. El sistema llevará cuenta de la cantidad de ejecuciones del código fuente e irá actualizando en la base de datos.

Para el caso de uso ejecutarCodigo de la tabla 29, se debe tener en cuenta que el computador donde se ejecute el Sistema de aula IoT, va a estar conectado a una placa Arduino la cual tendrá una serie de sensores conectados. En el momento que el estudiante ejecute el código, la herramienta software deberá buscar automáticamente el puerto donde esté conectada la placa para que el código pueda ejecutarse.

Tabla 30. Caso de uso subirEvidencia del rol Estudiante.

Nombre del caso de uso	subirEvidencia.
Actor participante	Iniciado por: Estudiante. Consulta a: Base de datos.
Condición inicial	1. El Estudiante ha descargado uno de los retos asignados y que desea agregar evidencia.
Flujo de eventos	2. El Estudiante presiona el botón Subir evidencia.
Condición de salida	3. Se presentará una ventana emergente en la cual podrá seleccionar el archivo que se subirá como evidencia de la solución del reto.
Requerimiento especial	El sistema consulta a la base de datos para cargar el archivo en ella.

Tabla 31. Caso de uso enviarSolucion del rol Estudiante.

Nombre del caso de uso	enviarSolcion.
Actor participante	Iniciado por: Estudiante. Consulta a: Base de datos.
Condición inicial	1. El Estudiante ha ejecutado código para la solución del reto.
Flujo de eventos	2. El Estudiante presiona el botón Enviar solución. 3. El sistema tomará el archivo de la última ejecución especificado en el caso de uso ejecutarCodigo (Tabla 29).
Condición de salida	4. Se presentará una ventana emergente la cual informará del envío exitoso de la solución.
Requerimiento especial	El sistema consulta a la base de datos para cargar el archivo en ella.

3.8 Interfaz de usuario

La interfaz del sistema se diseñó de acuerdo a los casos de uso identificados en la [sección 3.7](#), ya que estos describen el comportamiento del sistema desde el punto de vista de los actores y están orientados a determinar el qué, y no el cómo de lo que va a realizar el sistema.

La interfaz del sistema deberá presentar la opción de elegir un rol e ingresar un usuario y contraseña para posteriormente presentar en pantalla las diferentes funcionalidades a las que tenga acceso el rol de ingreso como se muestra en la Ilustración 16.

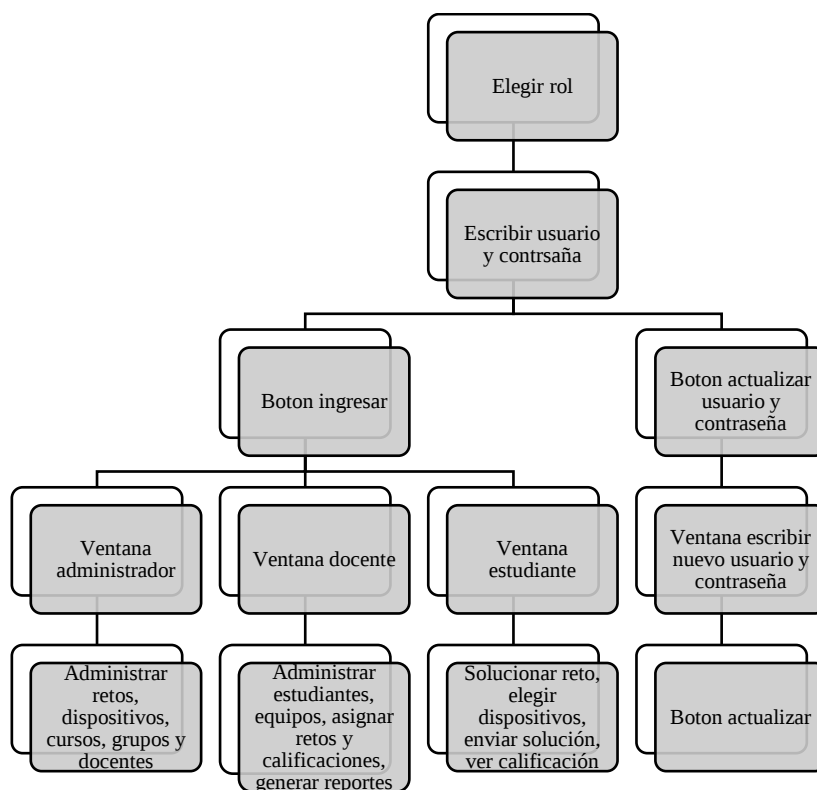


Ilustración 16. Flujo de eventos de la interfaz de usuario.

Fuente: autor

Se diseñaron a lápiz modelos de cómo se pretendía que quedara visualmente la herramienta, estructura, botones, campos de visualización entre otros, todos estos sujetos a modificaciones. Para constancia se anexa Ilustraciones 17 a 23.

Una vez ingresado y validado el usuario y contraseña, el sistema presentará una ventana diferente según el rol y cada funcionalidad a que tenga acceso en un sistema de pestañas en la misma ventana, excepto si el usuario va a actualizar usuario y contraseña a todos se presentará una pestaña en la misma ventana de Loguin como se aprecia en el diseño de la Ilustración 18.

Para los colores de la interfaz se optó por los colores institucionales de la universidad lo cual se verá reflejado en el diseño final quedando así:

- Color de fondo: rojo institucional ("ad3333")
- Botones: azul institucional ("003366")
- Letras: gris institucional ("DADADA")

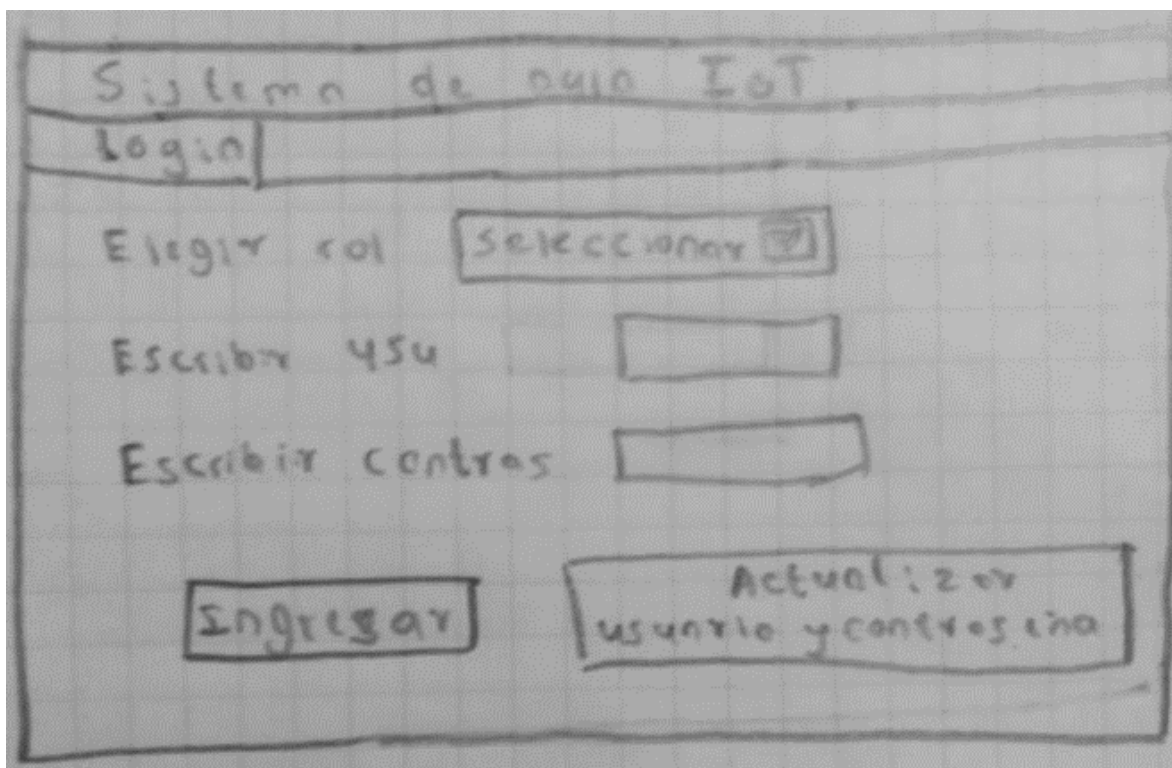


Ilustración 17. Diseño previo de ventana de Loguin para el sistema de aula IoT.
Fuente: autor.

Sistema de aula IoT

Login **Actualizar usuario**

usuario

confirmar usuario

contraseña

confirmar contraseña

Actualizar

Ilustración 18. Diseño previo de ventana Actualizar usuario y contraseña.
Fuente: autor

Nombre del Docente, curso y grupo

Admin grupos **Admin equipos** **Calificar reto**

- Ver y elegir grupos o carga
- Agregar los estudiantes del grupo
- Editar Estudiante del grupo

Ilustración 19. Diseño previo de ventana Docente, funcionalidad administrar grupos.
Fuente: autor

Nombre del docente, curso y grupo

Admin grupos **Admin equipos** **Calificar reto**

- Crear, ver, editar y eliminar equipos
- Asignar y quitar estudiantes a los equipos
- Asignar y quitar retos a los equipos
- Botón para elegir un reto y calificarlo

Ilustración 20. Diseño previo ventana Docente, funcionalidad administrar equipos.
Fuente: autor

Nombre del docente, Curso y grupo		
Administrar grupos	Admin equipos	Calificar reto
• Ver datos de la solución del reto (h. inicio, h. fin, intentos, sensores usados) • Ver evidencia • Ver código solución • Asignar calificación • Generar reporte		

Ilustración 21. Diseño previo de ventana Docente, funcionalidad calificar reto.

Fuente: autor

Equipo nombre		
Datos	Solución	Solucionar reto
• Ver integrantes del equipo • Ver datos de la solución a un reto (fecha inicio, fecha fin, intentos, sensores usados, calificación)		

Ilustración 22. Diseño previo de ventana Estudiante, funcionalidad datos solución.

Fuente: autor

Equipo nombre		
Datos	Solución	Solucionar reto
• Elegir y ver un reto • Elegir sensores • Escribir y ejecutar código python • Enviar evidencia • Enviar solución		

Ilustración 23. Diseño previo de ventana Estudiante, funcionalidad solucionar reto.

Fuente: autor

3.9 Elección de las herramientas de desarrollo

Para la implementación del sistema de aula IoT se debían tener en cuenta tres ejes importantes: el lenguaje de programación a usar, la interfaz de usuario y que este lenguaje se pudiera comunicar o traducir a Arduino.

Es por esto se optó por el lenguaje de programación Python, ya que este ofrece un gran número de librerías y facilidades para el desarrollo de software, además que facilita la comunicación con Arduino y el diseño de interfaces gráficas.

Python es un lenguaje de programación de alto nivel que posee una licencia de código abierto y que se puede aplicar en una gran variedad de campos de producción tecnológica; por otra parte, Arduino es una gran variedad hardware y software libre en placas electrónicas que se pueden utilizar para gran variedad de aplicaciones como IoT, domótica, electrónica, robótica, entre muchas otras, Arduino está basado en el lenguaje de programación c++.

Se puede hacer uso del potencial de estas dos tecnologías para diversas aplicaciones, para lo cual es necesario de un intermediario que las comunique, es ahí donde entra Firmata que es un protocolo que permite la comunicación entre microcontroladores y un software instalado en un computador. Con pyfirmata que es una interface de Python para hacer uso de Firmata se puede comunicar Python y Arduino.

Una vez lograda la comunicación entre estas dos tecnologías es necesario la creación de una interfaz gráfica de usuario (GUI por sus siglas en ingles) que facilite su uso, y con la biblioteca tkinter de Python se puede lograr.

4 DISEÑO E IMPLEMENTACIÓN

El diseño de sistemas consiste en la transformación del modelo de análisis en el modelo de diseño, que toma en cuenta los requerimientos no funcionales y las restricciones descritos anteriormente.

4.1 Definición de los objetivos de diseño

- **Confiabilidad:** no permitirá que un docente pueda ver la información de un grupo al que no dicta clase, así como no permitirá que un estudiante tenga acceso a la solución predefinida del reto y solo podrá acceder al reto que tenga asignado.
- **Informe de errores:** el sistema deberá validar fallas como la no conexión de la placa Arduino al computador, así como el uso inadecuado de variables o dispositivos no definidos o código mal escrito, deberá mostrar un mensaje de advertencia indicando alguno de estos errores por parte del usuario e impedir eliminar equipos que contengan estudiantes asignados.
- **Modificabilidad:** deberá permitir crear nuevos equipos con nuevos estudiantes y reasignar retos por parte del docente, así como reasignar estudiantes a los equipos.

4.2 Definición de los almacenes de datos persistentes

Para llevar la administración de los datos persistentes se optó por usar una base de datos relacional y mysql como sistema de administración de base de datos.

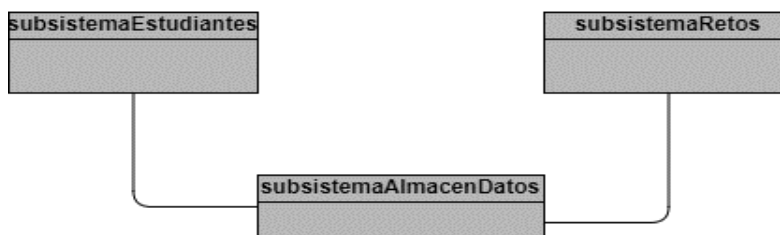


Ilustración 24. Diagrama de clase almacén de datos.
Fuente: autor

En la Ilustración 26 se muestra el modelo relacional de la base de datos para el sistema de aula IoT, en el cual se aprecia que la tabla característica_sensores no tiene relación con ninguna otra, esta tabla se creó solo para guardar el archivo guía de los sensores.

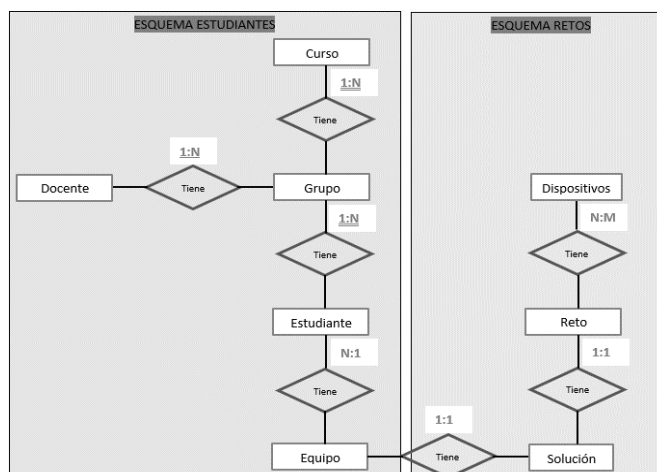


Ilustración 25. Modelo Entidad Relación para la base de datos del sistema.
Fuente: autor.

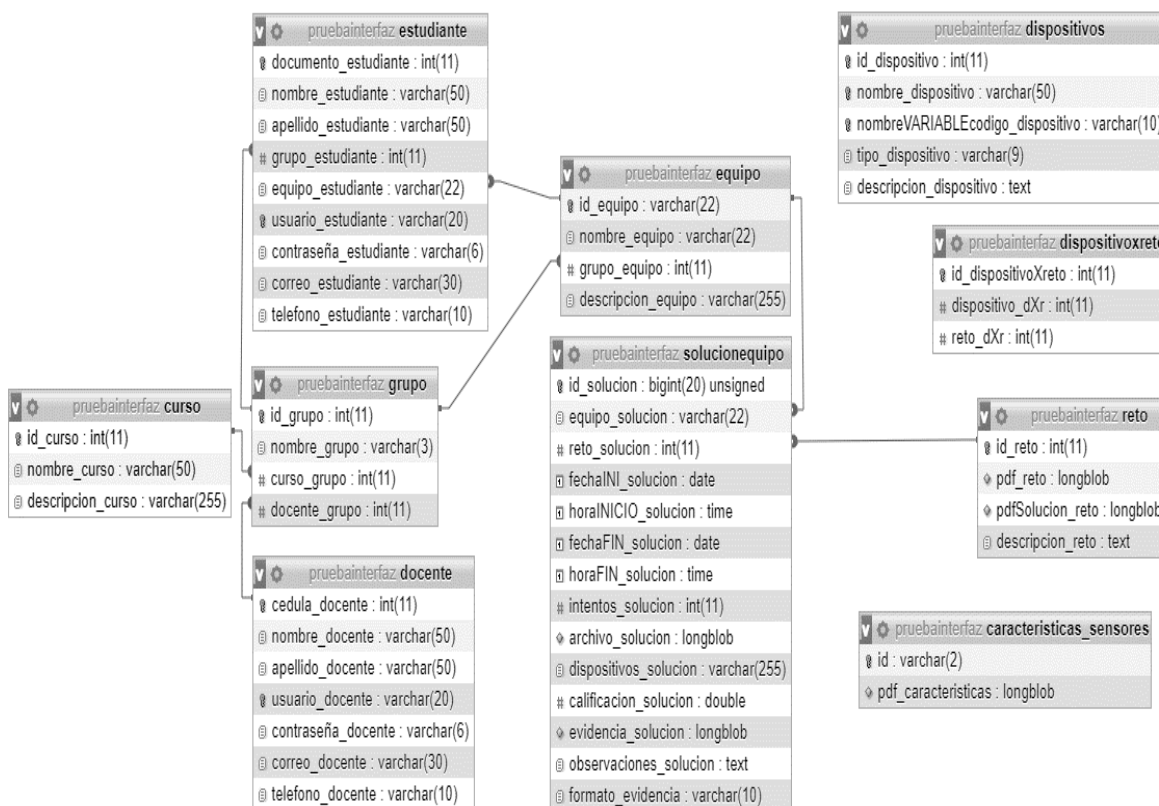


Ilustración 26. Modelo Relacional para la base de datos del sistema.
Fuente: autor.

4.3 Implementación

4.3.1 Herramientas y tecnologías a usar

Para la implementación del sistema se utilizó el lenguaje de programación Python, para el desarrollo de su interfaz gráfica se hizo uso de la librería Tkinter de Python y para la comunicación con Arduino se hizo uso de la interfaz pyfirmata de Python que permite hacer uso del protocolo Firmata para comunicar microprocesadores con software instalado en un dispositivo. En la [sección 3.9](#) se explicó por qué se optó por estas herramientas.

4.3.2 Ejemplo de interfaz gráfica utilizando tkinter

Se utilizó Visual Studio Code para escribir el código fuente del Sistema de aula IoT; debido a lo extenso del código, a continuación, se presenta a manera de ejemplo cómo se crea una interfaz gráfica utilizando la librería tkinter de Python.

1. Para crear un entorno gráfico con **Tkinter** lo primero que se hace es importar la librería **tkinter** y crear una ventana raíz la cual contendrá los demás widgets e invocar el método **mainloop()** para que la interfaz se mantenga ejecutando y a la escucha de nuevos eventos.

```
1 import tkinter as tk
2
3 ventana = tk.Tk()
4 ventana.title("Ventana de prueba")
5 ventana.mainloop()
```

*Ilustración 27. Creación de una ventana con Tkinter.
Fuente: autor.*

2. Para crear pestañas de **tkinter** se debe importar **ttk**, se crea un objeto de la clase **Notebook** y se agrega a la ventana principal.

```
1 import tkinter as tk
2 from tkinter import ttk
3
4 ventana = tk.Tk()
5
6 ventana.title("Ventana de prueba")
7
8 notebook = ttk.Notebook(ventana)
9 notebook.pack(fill="both", expand="yes")
10
11 ventana.mainloop()
```

Ilustración 28. Clase Notebook para crear pestañas en una ventana con tkinter.
Fuente: autor

3. Luego con **ttk** se crea un objeto de la clase **Frame** para agregar las pestañas que se necesiten.

```
1 import tkinter as tk
2 from tkinter import ttk
3
4 ventana = tk.Tk()
5
6 ventana.title("Ventana de prueba")
7
8 notebook = ttk.Notebook(ventana)
9 notebook.pack(fill="both", expand="yes")
10
11 pestania1 = ttk.Frame(notebook)
12 notebook.add(pestania1, text="Pestaña 1")
13
14 pestania2 = ttk.Frame(notebook)
15 notebook.add(pestania2, text="Pestaña 2")
16
17 ventana.mainloop()
```

Ilustración 29. Añadir pestañas a una ventana con tkinter.
Fuente: autor.

Visualmente hasta el momento se lleva lo siguiente

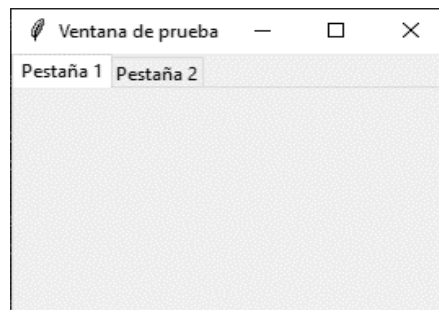


Ilustración 30. Ventana con pestañas utilizando tkinter.
Fuente: autor.

4. Una vez creadas las pestañas y agregadas a la ventana principal, se puede adicionar a estas los widgets que se deseen como campos de texto (**Entry**), etiquetas (**Label**), botones y lo que se considere necesario para la interfaz gráfica como se muestra en la Ilustración 31.

```

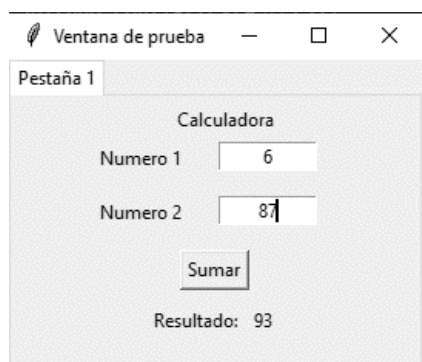
1  import tkinter as tk
2  from tkinter import ttk
3  def sumar(n1,n2):
4      s = n1+n2
5      resultado.config(text=str(s))
6
7  ventana = tk.Tk()
8  ventana.title("Ventana de prueba")
9
10 notebook = ttk.Notebook(ventana)
11 notebook.pack(fill="both",expand="yes")
12 pestania1 = ttk.Frame(notebook)
13 notebook.add(pestania1, text="Pestaña 1")
14
15 tk.Label(pestania1, text="Calculadora").place(x=105,y=5)
16
17 tk.Label(pestania1, text="Numero 1").place(x=55,y=30)
18 entry1 = tk.Entry(pestania1, justify="center", width=10)
19 entry1.place(x=135,y=30)
20
21 tk.Label(pestania1, text="Numero 2").place(x=55,y=65)
22 entry2 = tk.Entry(pestania1, justify="center", width=10)
23 entry2.place(x=135,y=65)
24
25 tk.Button(pestania1,text="Sumar",command=lambda:sumar(int(entry1.get()),int(entry2.get()))).place(x=110,y=100)
26
27 tk.Label(pestania1,text="Resultado:").place(x=90,y=135)
28 resultado = tk.Label(pestania1)
29 resultado.place(x=155,y=135)
30
31 ventana.geometry("270x200")
32 ventana.mainloop()

```

*Ilustración 31. Agregar widgets a una pestaña con tkinter.
Fuente: autor.*

Como se aprecia en la Ilustración 31 los widgets se pueden posicionar en las pestañas por medio del método **place()**, que recibe una coordenada X e Y, pero también se puede usar el método **grid()** o **pack()**; cada uno de ellos ofrece sus ventajas y desventajas, pero se recomienda **place()** ya que este permite posicionar los widgets de acuerdo a los requerimientos.

También se aprecia que el botón hace el llamado a una función mediante la propiedad **command()**, se utiliza lambda para impedir que el botón se ejecute en el flujo del algoritmo sin ser presionado y haga el llamado a la función con parámetros vacíos, esto debido a que en los campos de texto no se ha ingresado ningún número. El código de la Ilustración 31 dará como resultado la interfaz gráfica mostrada en la Ilustración 32.



*Ilustración 32. Interfaz gráfica del código de la Ilustración 25.
Fuente: autor*

4.3.3 Definición de clases del Sistema de aula IoT

Para la codificación del sistema se tomó como punto de partida los casos de uso identificados en la [sección 3.7](#) y se procedió a crear las clases que se encargan de la parte lógica y de interfaz de usuario como lo son la clase Inicio, Docente y Estudiante, estas a su vez llaman a las clases del modelo: la clase Loguin, ConsultasDocente y ConsultasEquipo, que a su vez llaman a la clase Conexión que también hace parte del modelo. Los métodos de las clases del modelo se encargan exclusivamente de hacer las consultas y las modificaciones a la base de datos y devolver los registros solicitados por las clases de la parte lógica que se encargan de procesarlos y mostrarlos en la interfaz. En la Ilustración 33 se muestra mediante un diagrama de clases UML el diseño y el llamado entre clases del Sistema de aula IoT, teniendo en cuenta los casos de uso identificados en la fase de análisis. En la Ilustración 34 se presenta un diagrama de flujo del orden de llamado entre clases en el Sistema de aula IoT.

➤ **CLASES DEL MODELO**

- Clase Conexión: esta clase se encarga de crear un objeto de conexión a la base de datos y es llamada por las demás clases del modelo. Para la conexión a la base de datos se hace uso de MySQL Connector que es el nombre de la familia de controladores MySQL ODBC que proporcionan acceso a una base de datos MySQL usando el estándar industrial de Conectividad de Base de Datos Abierta (Oracle Corporation and/or its affiliates, 2021).
- Clase Loguin: esta clase se encarga de validar las credenciales de ingreso de un usuario. Esta clase crea un objeto de la clase Conexión en cada operación el cual cierra una vez consulta o modifica la base de datos.
- Clase ConsultasDocente: hace parte del modelo y se encarga de hacer toda la parte del CRUD⁶ para llevar a cabo la implementación de los casos de uso identificados en la [sección 3.7.1](#) como administrar grupos y equipos y calificar retos. Esta clase crea un objeto de la clase Conexión en cada operación el cual cierra una vez consulta o modifica la base de datos.
- Clase ConsultasEquipo: hace parte del modelo y se encarga de hacer toda la parte del CRUD para llevar a cabo la implementación de los casos de uso identificados en la [sección 3.7.2](#) como ver datos de una solución ya hecha por el equipo o dar solución a un reto. Esta clase crea un objeto de la clase Conexión en cada operación el cual cierra una vez consulta o modifica la base de datos.

⁶ CRUD, por sus siglas en ingles Create, Read, Update y Delete; es un término que en bases de datos y diseño web es muy utilizado y hace referencia a las acciones que se realizan sobre la base de datos como crear, escribir, actualizar y eliminar registros de las tablas.

➤ **CLASES DE LA PARTE LÓGICA E INTERFAZ DE USUARIO**

- Clase Docente: Implementa los métodos necesarios para tener en cuenta los casos de uso de las funcionalidades identificados en la [sección 3.7](#) (Ilustración 10), para esto se crearon tres pestañas, una por cada funcionalidad: AdministrarGrupos, AdministrarEquipos y CalificarReto; cada una de estas pestañas implementa los casos de uso identificados en la [sección 3.7.1](#) (Ilustraciones 11, 12 y 13). Esta clase crea un objeto de la clase consultasDocente.
- Clase Estudiante: Implementa los métodos necesarios para tener en cuenta los casos de uso de las funcionalidades identificados en la [sección 3.7](#) (Ilustración 10) para esto se crearon dos pestañas, una por cada funcionalidad: DatosSolucion y SolucionarReto; cada una de estas pestañas implementa los casos de uso identificados en la [sección 3.7.2](#) (Ilustraciones 14 y 15). Esta clase crea un objeto de la clase consultasEquipo.
- Clase Inicio: en esta clase se implementaron dos pestañas, una para Loguin y otra para actualizar usuario y contraseña. La pestaña de Loguin se encarga de recibir y validar el usuario y contraseña y el rol con el que desea ingresar, para lo cual crea un objeto de la clase Loguin y así validar los datos; dependiendo del rol de ingreso se creará un objeto de la clase Docente o Estudiante y hará el llamado al constructor de la clase correspondiente. La pestaña de actualizar se encarga de recibir el nuevo usuario y contraseña y actualizarlos en la base de datos.

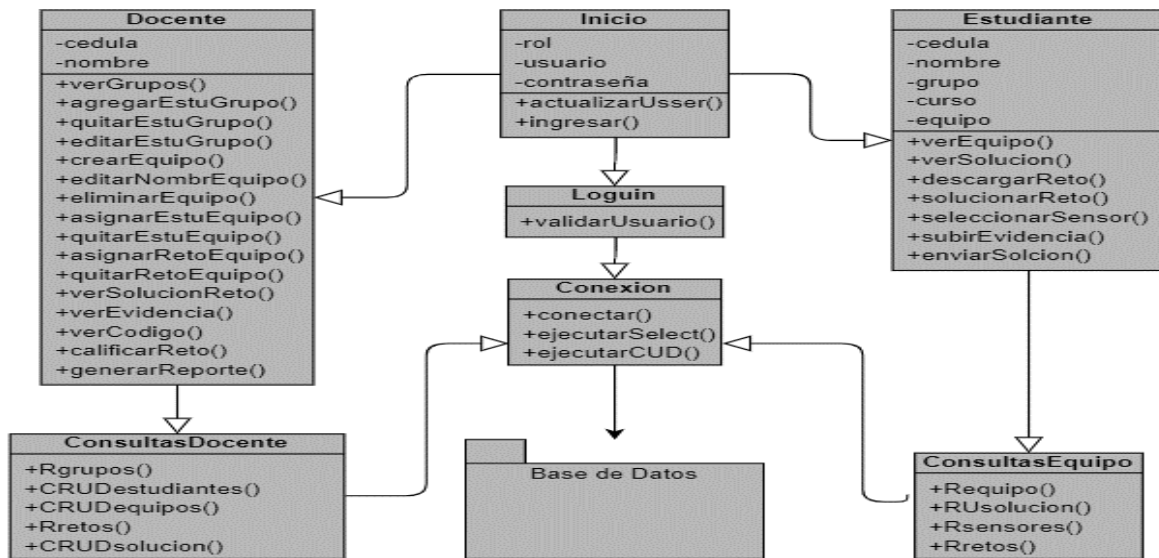


Ilustración 33. Diagrama de clases del diseño del Sistema de aula IoT.

Fuente: autor.

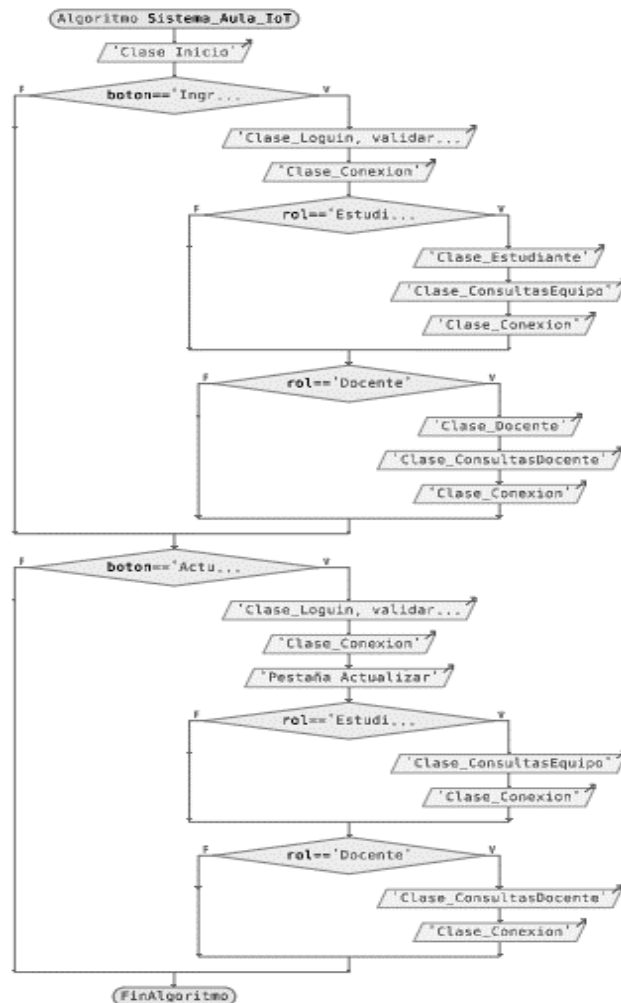


Ilustración 34. Diagrama de flujo del Sistema de Aula IoT.

Fuente: autor.

4.3.4 Interfaz gráfica del sistema de aula IoT

Como ya se ha mencionado para el entorno gráfico se utilizó la librería Tkinter de Python y como editor de código se utilizó Visual Studio Code. Debido a lo extenso del código el cual se organizó en diferentes archivos y módulos, no se presentará en este documento, por lo cual se optó por presentar un ejemplo de cómo se crea una interfaz gráfica en la [sección 4.3.2](#)

Como ya se explicó en la sección anterior, para cada clase de la parte lógica (Inicio, Docente y Estudiante) se creó una ventana, estas ventanas a su vez implementan un sistema de pestañas para organizar las funcionalidades de cada rol, funcionalidades que se especificaron en la [sección 3.7](#), en cada pestaña se implementa los casos de uso identificados en la [sección 3.7.1](#) y [sección 3.7.2](#).

Ya en la [sección 3.8](#) en la Ilustración 16, se había definido el flujo de eventos de cómo debería presentarse la interfaz de usuario que es coincidente con la Ilustración 34 de la [sección 4.3.3](#). Y se habían diseñado a lápiz modelos de cómo se deseaba que presentara visualmente la interfaz (Ilustraciones 17 a 23).

También se había definido que colores debía tener la interfaz gráfica y su distribución quedando así:

- Color de fondo: rojo institucional ("ad3333")
- Botones: azul institucional ("003366")
- Letras: gris institucional ("DADADA")

A continuación, se presentan una serie de ilustraciones con el resultado final de la interfaz gráfica.

La Ilustración 35 muestra la ventana de inicio del Sistema de aula IoT, es implementada por la clase Inicio y contiene un ComboBox para elegir un rol, dos campos de texto para ingresar usuario y contraseña, y dos botones para ingresar o actualizar credenciales de ingreso.



Ilustración 35. Ventana de inicio del Sistema de aula IoT.

Fuente: autor.

Si las credenciales de ingreso son incorrectas entonces el sistema mostrará una ventana emergente con la advertencia.

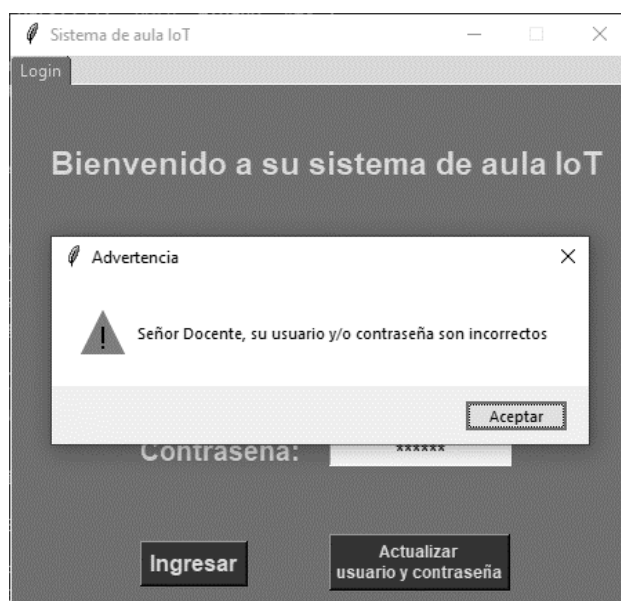
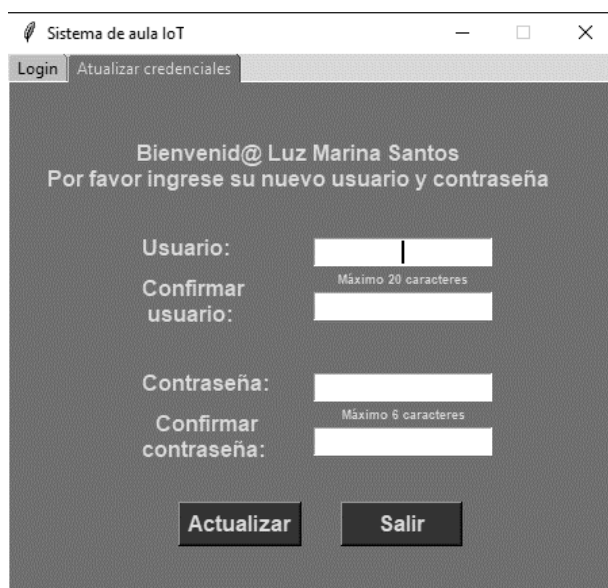


Ilustración 36. Ventana emergente para credenciales de ingreso incorrectas.

Fuente: autor.

Si las credenciales de ingreso son correctas, al presionar el botón Actualizar usuario y contraseña se desplegará una nueva pestaña dentro de la misma ventana para actualizar las credenciales.



*Ilustración 37. Pestaña actualizar usuario y contraseña.
Fuente: autor.*

Si las credenciales de ingreso son correctas, al presionar el botón Ingresar, se creará una nueva ventana dividida por pestañas con las funcionalidades a que tenga acceso según el rol con el que se ingrese.

➤ **FUNCIONALIDADES DEL ROL DOCENTE**

La pestaña Administrar grupos de la Ilustración 38 implementa la funcionalidad AdministrarGrupos identificada en la [sección 3.7.1](#), más precisamente en la Ilustración 11. En esta pestaña se materializan los casos de uso allí especificados, como mostrar los grupos que el docente tiene a cargo, elegir un grupo y agregar, ver, editar y eliminar estudiantes al grupo.

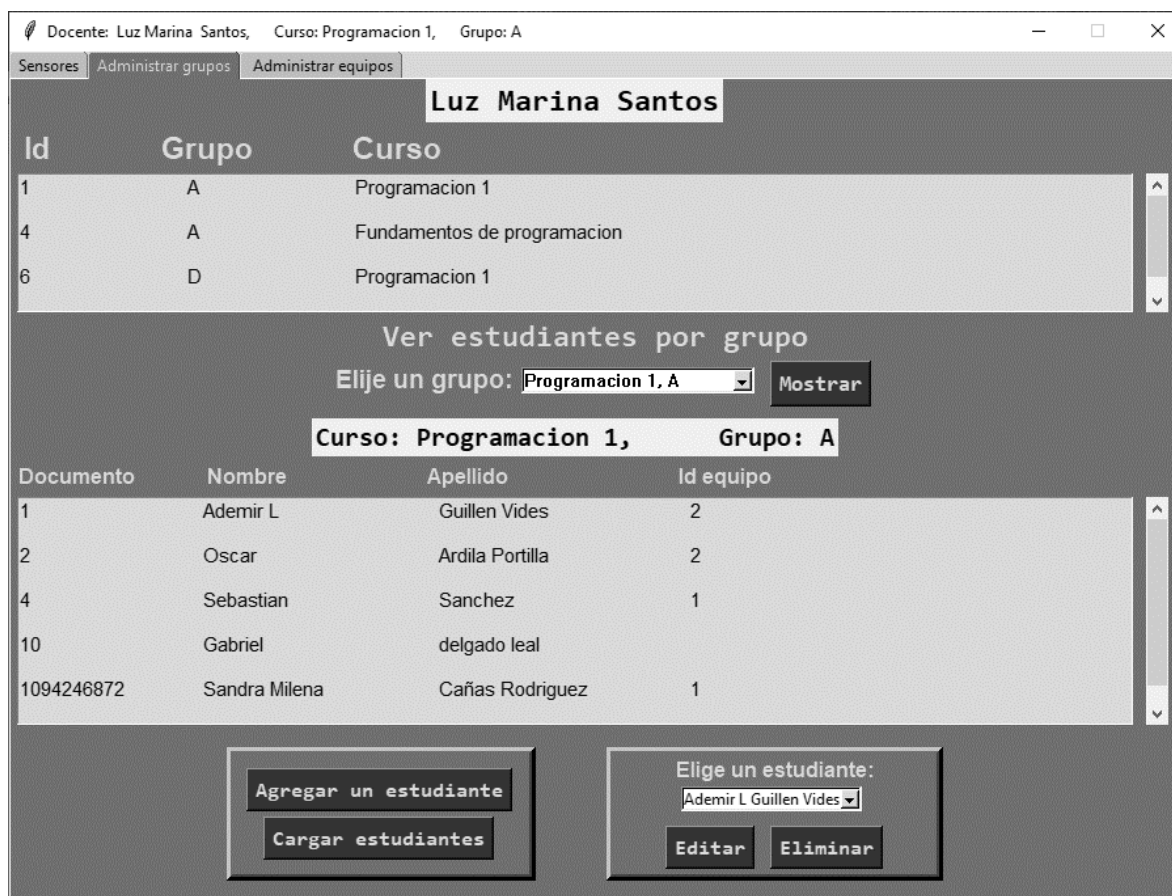


Ilustración 38. Pestaña de la funcionalidad AdministrarGrupos del rol Docente
Fuente: autor.

La pestaña Administrar equipos de la Ilustración 39 muestra la funcionalidad AdministrarEquipos con sus respectivos casos de uso especificados en la Ilustración 12. En esta pestaña el docente puede crear un equipo, elegir un equipo, cambiar su nombre y eliminarlo; agregar quitar y ver estudiantes; asignar, quitar y calificar retos al equipo que haya elegido para ser administrado. Esta pestaña se despliega cuando el docente selecciona uno de los grupos que tiene a cargo, tal y como se muestra en la Ilustración 12 y 10 esta funcionalidad depende de la funcionalidad AdministrarGrupo.

Cuando el docente presiona el botón descargar reto, este se guarda en una carpeta creada por el mismo sistema.

Docente: Luz Marina Santos, Curso: Programacion 1, Grupo: A

Sensores | Administrar grupos | Administrar equipos

Id Equipo	Nombre	Id retos asignados	Id retos calificados
1	equipo1	1, 2,	1, 2,
2	equipo2	2,	

EQUIPOS

Nombre: Máximo 22 caracteres

Crear equipo Cambiar nombre

Elige un equipo:

Eliminar Mostrar

Elige un estudiante:

Asignar estudiante Quitar estudiante

Documento	Nombre	Apellido
4	Sebastian	Sanchez
1094246872	Sandra Milena	Cañas Rodriguez

Estudiantes del Equipo: equipo1, Id equipo: 1

Id reto	Descripción
1	Reto 1. Sensar la temperatura en una incubadora.
2	Reto 2

RETOS

Elige un reto:

Asignar reto Quitar reto Calificar reto Descargar reto

Ilustración 39. Pestaña con la funcionalidad AdministrarEquipos del rol Docente.
Fuente: autor.

La Ilustración 40 muestra la pestaña que implementa la funcionalidad CalificarReto y los casos de uso identificados para esta funcionalidad del rol Docente. Esta se despliega cuando el docente selecciona un equipo y presiona el botón Calificar reto de la funcionalidad administrarEquipos, tal y como se especificó en la identificación de casos de uso, más específicamente en la Ilustración 10 y 13, esta depende de la funcionalidad AdministrarEquipos.

En esta pestaña el docente puede elegir un reto y ver los datos de la solución dada por un equipo presionando el botón mostrar, descargar y ver la evidencia, ver el ultimo código escrito en el lenguaje de programación Python, asignar una calificación y generar un reporte

de la solución al reto en archivo tipo scv, estos archivos se descargan en una carpeta creada por el mismo Sistema de aula IoT.

Docente: Luz Marina Santos, Curso: Programacion 1, Grupo: A

Sensores | Administrar grupos | Administrar equipos | **Calificar reto**

Documento	Nombre	Apellido
4	Sebastian	Sanchez
1094246872	Sandra Milena	Cañas Rodriguez

Integrantes del equipo: equipo1

Elije el reto: **Mostrar**

Datos de la solución del reto 1

Id Solución:	1	Observaciones del docente Faltó punto 3
Fecha inicio:	2021-01-29	
Hora inicio:	16:36:56	
Fecha envío:	2021-04-12	
Hora envío:	18:00:26	
Intentos:	48	
Dispositivos:		
Calificación:	3.6	

Decimales separados por punto, ej: 4.6

Ver evidencia **Calificar** **Ver código** **Generar reporte**

*Ilustración 40. Pestaña de la funcionalidad CalificarReto del rol Docente.
Fuente: autor.*

➤ FUNCIONALIDADES DEL ROL ESTUDIANTE

Una vez que el estudiante haya ingresado sus credenciales correctamente al igual que para el docente, el sistema presenta una ventana dividida por pestañas cada una con las funcionalidades descritas para el estudiante en la [sección 3.7](#) más específicamente en la Ilustración 10.

La Ilustración 41 muestra la implementación de la funcionalidad DatosSolucion con sus respectivos casos de uso. En esta pestaña el estudiante puede elegir un reto y ver los datos de

la solución que como equipo dieron al reto, además de los integrantes del mismo como se especificó en la Ilustración 14.

The screenshot shows a web application window titled 'Equipo equipo1'. It has three tabs: 'Sensores', 'Datos solución', and 'Solucionar Reto'. The 'Datos solución' tab is active. At the top, there is a table with the following data:

Nombre	Apellido	Grupo	Curso
Sebastian	Sanchez	A	Programacion 1
Sandra Milena	Cañas Rodriguez	A	Programacion 1

Below the table, a summary bar displays: **Curso: Programacion 1, Grupo: A, Equipo: equipo1**. Underneath this, it says 'Ver soluciones del equipo'. There is a dropdown menu labeled 'Elige el reto:' with the value '1' selected, and a 'Mostrar' button. To the left of the 'Observaciones del docente' section, there is a list of solution details:

- Id Solución: 1
- Fecha inicio: 2021-01-29
- Hora inicio: 16:36:56
- Fecha envío: 2021-04-12
- Hora envío: 18:00:26
- Intentos: 48
- Dispositivos:
- Calificación: 3.6

To the right of these details is a section titled 'Observaciones del docente' with a text area containing the text 'Faltó punto 3'.

Ilustración 41. Pestaña de la funcionalidad DatosSolucion del rol Estudiante.

Fuente: autor.

La Ilustración 42 presenta la funcionalidad SolucionarReto identificada en la [sección 3.7.2](#) y especificada en la Ilustración 15, aquí se presentan todos los casos de uso que pertenecen a esta funcionalidad. El estudiante puede descargar un reto y proceder a darle solución, el archivo con el reto se descarga en una carpeta creada automáticamente por el Sistema de aula IoT, también puede ir seleccionando dispositivos o sensores que vaya a utilizar para la solución del reto, esto a manera de información de que dispositivos utilizó, puede subir una evidencia como un video, imagen o archivo Word o cualquier otro tipo de evidencia y proceder a enviar la solución que ya crea conveniente.

Cuando el estudiante escribe el código fuente en el lenguaje de programación Python y presiona el botón ejecutar se desplegará una nueva ventana con la consola de Windows para mostrar la ejecución del código como se muestra en la Ilustración 43.

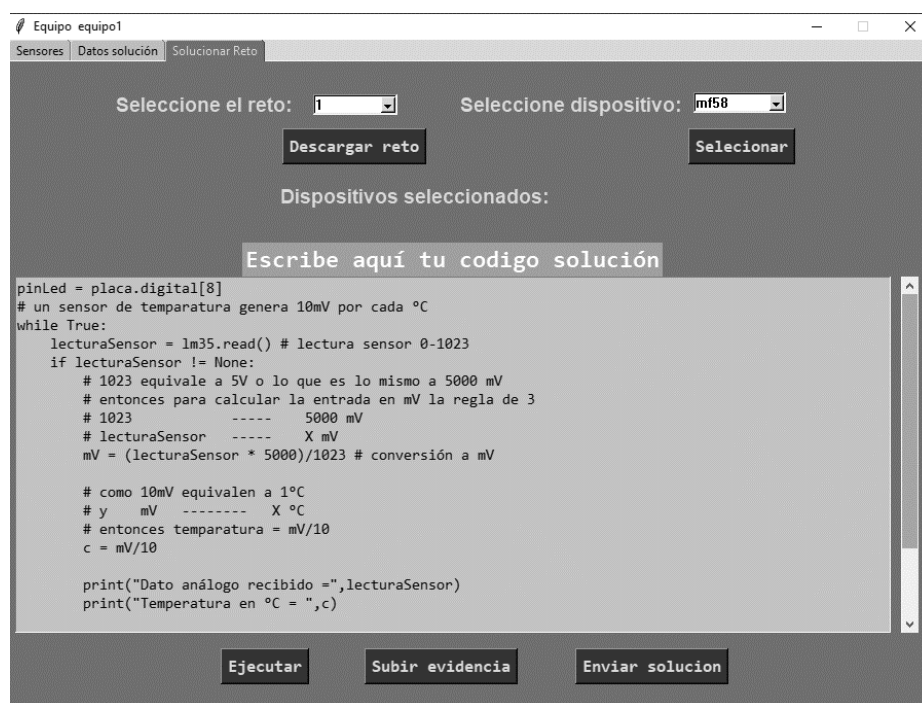


Ilustración 42. Pestaña de la funcionalidad SolucionarReto del rol Estudiante. Fuente: autor.

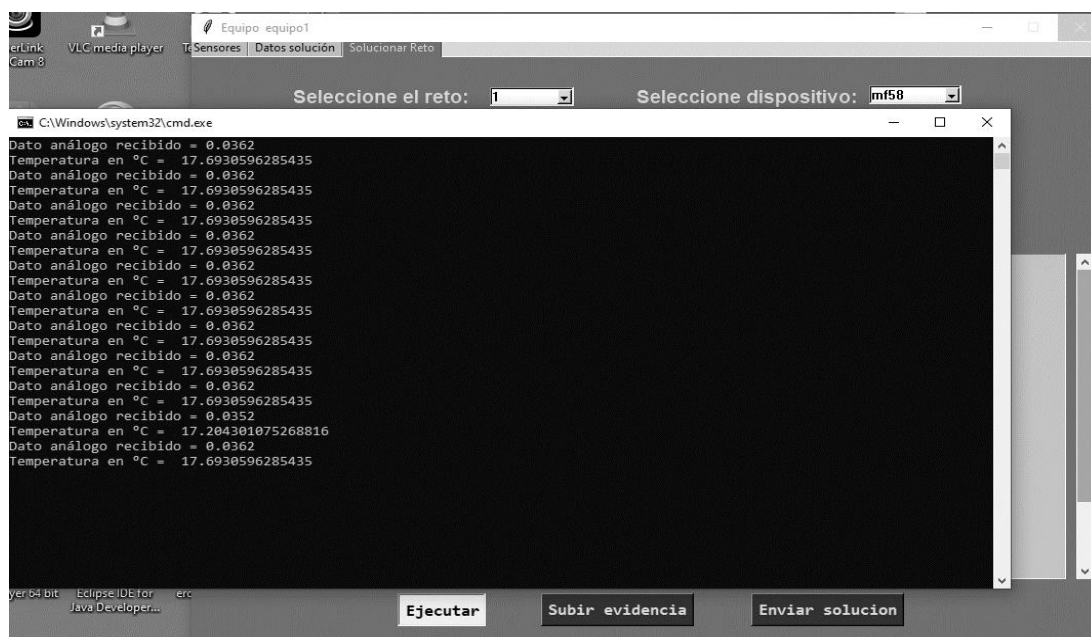
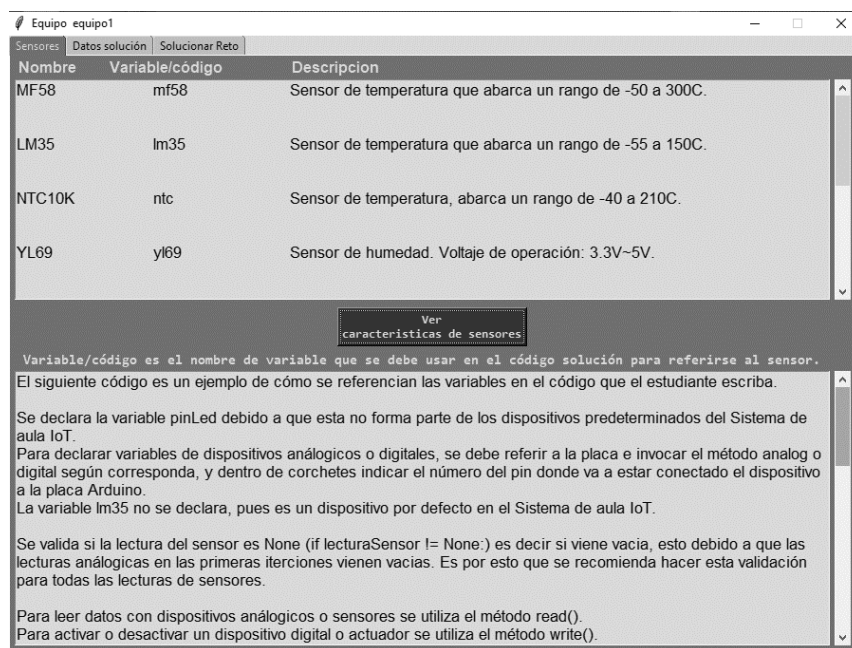


Ilustración 43. Ventana de ejecución del código fuente con el Sistema de aula IoT.
Fuente: autor.

Como ya se ha dicho, el computador donde se esté ejecutando el sistema de aula IoT para el estudiante va a estar conectado a una placa Arduino que a su vez va a tener una serie de dispositivos conectados, las variables que referencian a estos dispositivos ya están definidas previamente en el código fuente del software, para indicarle al mismo en que pin de la placa Arduino se encuentra conectado cada sensor. Esto con el objetivo principal de aplicar conceptos básicos de programación a problemas del mundo real haciendo uso de las bases del IoT.

La Ilustración 44 presenta una pestaña que se incluyó tanto en la interfaz del docente como del estudiante, esta pestaña muestra que sensores están conectados a la placa Arduino y con qué nombre de variable se deben referenciar en el código Python escrito por el estudiante, así como un código de ejemplo para guía del estudiante y un botón para descargar un pdf que contiene información sobre las características de los sensores disponibles en el Sistema de aula IoT.



Nombre	Variable/código	Descripción
MF58	mf58	Sensor de temperatura que abarca un rango de -50 a 300C.
LM35	lm35	Sensor de temperatura que abarca un rango de -55 a 150C.
NTC10K	ntc	Sensor de temperatura, abarca un rango de -40 a 210C.
YL69	yl69	Sensor de humedad. Voltaje de operación: 3.3V~5V.

[Ver características de sensores](#)

Variable/código es el nombre de variable que se debe usar en el código solución para referirse al sensor. El siguiente código es un ejemplo de cómo se referencian las variables en el código que el estudiante escriba.

Se declara la variable pinLed debido a que esta no forma parte de los dispositivos predeterminados del Sistema de aula IoT.

Para declarar variables de dispositivos analógicos o digitales, se debe referir a la placa e invocar el método analog o digital según corresponda, y dentro de corchetes indicar el número del pin donde va a estar conectado el dispositivo a la placa Arduino.

La variable lm35 no se declara, pues es un dispositivo por defecto en el Sistema de aula IoT.

Se valida si la lectura del sensor es None (if lecturaSensor != None:) es decir si viene vacía, esto debido a que las lecturas analógicas en las primeras iteraciones vienen vacías. Es por esto que se recomienda hacer esta validación para todas las lecturas de sensores.

Para leer datos con dispositivos analógicos o sensores se utiliza el método read().
Para activar o desactivar un dispositivo digital o actuador se utiliza el método write().

Ilustración 44. Pestaña guía de sensores disponibles en el Sistema de aula IoT.

Fuente: autor.

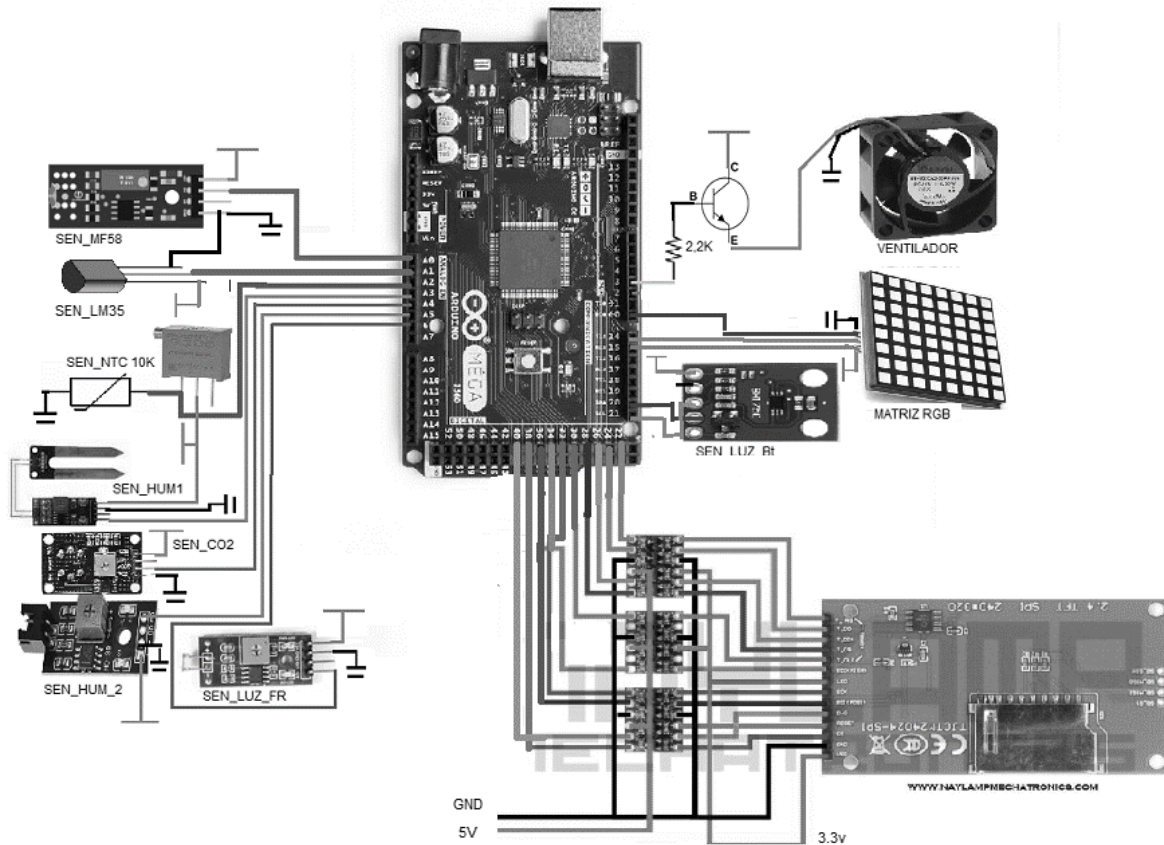
5 PRUEBAS

El objetivo de esta fase es validar el correcto funcionamiento de la herramienta y encontrar diferencias entre el comportamiento especificado y el comportamiento observado. Cabe recalcar que por motivos de la pandemia que se vive a la fecha y su confinamiento, no fue posible realizar las pruebas directamente con los estudiantes, por lo que dichas pruebas se hicieron con el módulo electrónico que se presenta en la Ilustración 45 y 46, el cual forma parte del sistema de aula IoT y que estará conectado el sistema diseñado en este proyecto y con el cual se hicieron las pruebas.

En esta fase se encontró el inconveniente de que la librería de Python PyFirmata no da soporte para comunicación I2C, ya que algunos de los dispositivos conectados al módulo electrónico usan este tipo de comunicación como por ejemplo el sensor de luz BH1750 y la pantalla ili9341. Para solucionar este inconveniente fue necesario incluir la librería PyMata que sí da un amplio soporte para este tipo de comunicación.



*Ilustración 45. Módulo electrónico del Sistemas de aula IoT.
Fuente: (Sandoval Carrero, Santos Jaimes, & Acevedo, 2020)*



*Ilustración 46. Esquema módulo electrónico del Sistemas de aula IoT
Fuente: (Sandoval Carrero, Santos Jaimes, & Acevedo, 2020)*

5.1 Resultados y análisis de resultados

Las ilustraciones 47 a 50 presentan las pruebas que se hicieron desde la interfaz gráfica del sistema de aula IoT con tres sensores del módulo electrónico que se muestra en la Ilustración 45 y 46, en ellas se aprecia el llamado a funciones que se dejaron predeterminadas en el sistema de aula IoT, dichas funciones pueden recibir como parámetro la cantidad de datos que desea censar el estudiante o no recibir ningún parámetro y tomar solo un dato. Una vez que el estudiante presiona el botón ejecutar; en ese momento se despliega una ventana para mostrar la ejecución del código, lo cual bloquea la herramienta software hasta que el código no se termine de ejecutar.

En el código que se presenta en dichas Ilustraciones se hace un llamado a las funciones predeterminadas y se les pasa como parámetro la cantidad de datos que desea tomar; estas funciones retornan una lista con la cantidad de datos tomados y posteriormente se procede a aplicar conceptos básicos de programación como en este caso estructuras repetitivas, condicionales y acceso a los índices de una lista.

También el estudiante podría no pasar ningún parámetro y censar solo un dato, por lo cual el mismo tendría que crear la lista, y dentro de un ciclo repetitivo en cada iteración hacer el llamado a la función e ir guardando cada dato en la lista.

Así como estos ejemplos se pueden hacer más ejercicios para aplicar conceptos básicos de programación interactuando con objetos físicos del mundo real, y es ahí donde entran los retos que estarán predefinidos en el sistema de aula IoT.

En la Ilustración 47 se presenta una prueba hecha con el sensor de temperatura lm35, este sensor genera 10 Volts por cada grado centígrado, cuando se trabaja directamente en Arduino envía un valor al puerto serial de 0 a 1023, donde 1023 que es el máximo valor equivale a 5 Volts, por lo que se debe hacer una regla de tres para convertir este valor a Volts y luego convertir estos Volts a grados centígrados. Este mismo sensor envía a Python por medio del protocolo PyFirmata un valor decimal entre 0 y 1, donde 1 equivale a 5 Volts, por lo que se debe aplicar el mismo procedimiento anterior para obtener la temperatura en grados centígrados, pero haciendo la conversión a Volts con un valor entre 0 y 1.

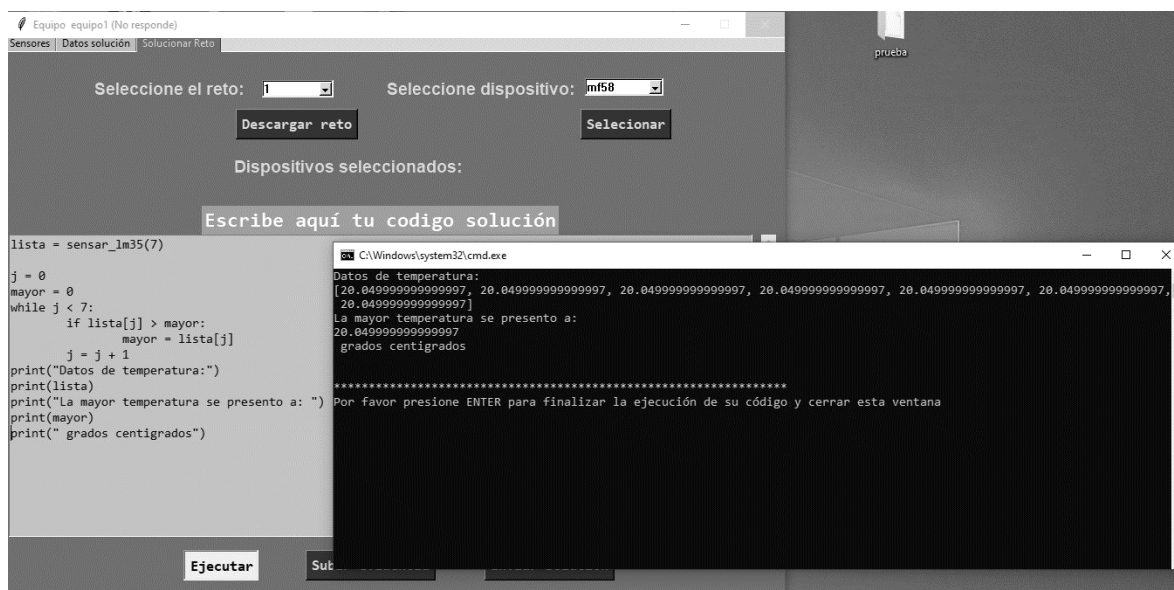


Ilustración 47. Prueba con sensor de temperatura lm35.
Fuente: autor.

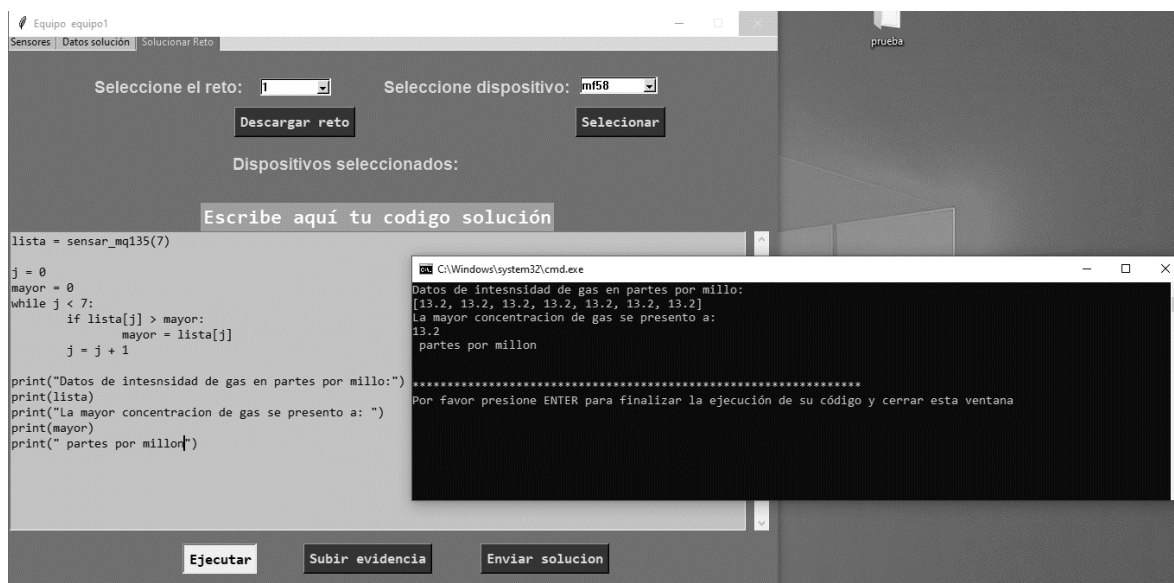


Ilustración 48. Prueba con sensor de gas mq135.
Fuente: autor.

La Ilustración 49 y 50 presentan los datos de luminosidad tomados con el sensor bh1750, en la Ilustración 50 se puede apreciar como la intensidad lumínica aumenta al iluminar el sensor con una linterna, en la impresión de la información que haría el estudiante se apreciaba como escribió datos de temperatura en lugar de luminosidad, esto es parte de los errores que un estudiante podría cometer a falta de concentración.

Este sensor utiliza comunicación I2C y con esto comprobamos el correcto funcionamiento de este protocolo en la comunicación entre Arduino y Python que se da en el software desarrollado en este trabajo.

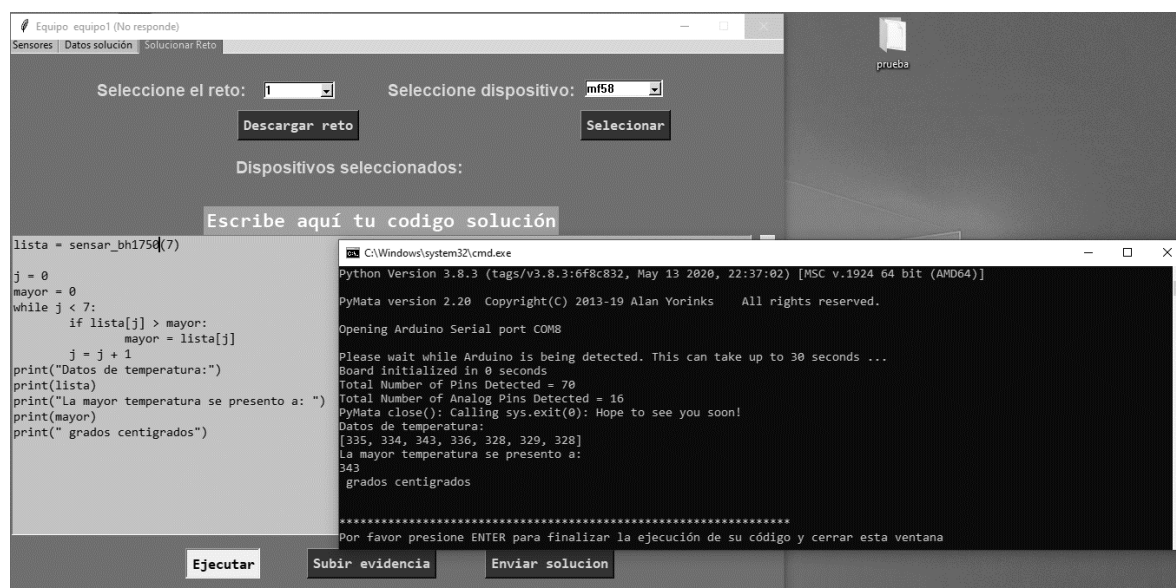


Ilustración 49. Prueba número 1 con sensor de luminosidad bh1750.

Fuente: autor.

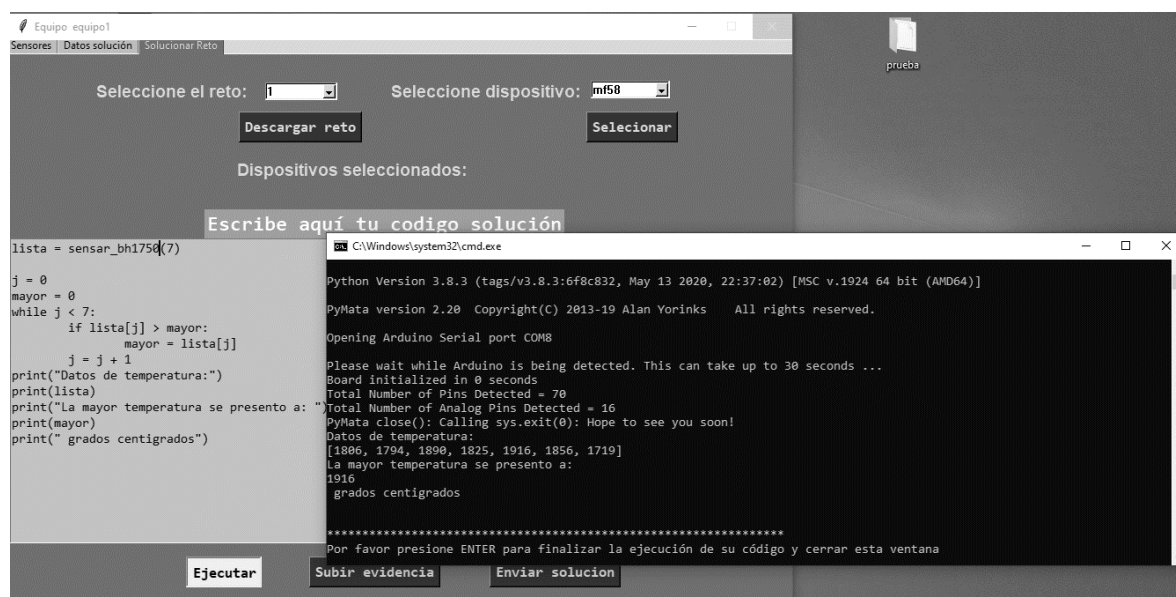


Ilustración 50. Prueba número 2 con sensor de luminosidad bh1750.

Fuente: autor.

6 CONCLUSIONES

- En base al modelo de aula IoT, se pudo establecer que es posible tener un proceso de aprendizaje creativo, ya que el estudiante por medio de la herramienta codificada en este proyecto podrá aplicar conceptos básicos de programación interactuando con objetos físicos del mundo real; por la cual podrá ver como su creatividad e ingenio al aplicar los conocimientos adquiridos en programación darán un resultado tangible.
- El auge de la industria 4.0 e IoT, es una gran oportunidad para motivar a las nuevas generaciones al apasionante mundo de la programación, pues hoy los jóvenes son más tendientes a usar tecnología y más dados a aprender con la práctica sin pasar por alto los fundamentos teóricos. Debido a que en la actualidad el campo laboral es cada día más exigente, es imperativo formar talentos competitivos y enfocarnos no solo en competencias del conocer sino también del hacer.
- El desarrollo de este trabajo fue una experiencia muy enriquecedora profesionalmente, pues fue necesario aplicar conceptos y conocimientos adquiridos en el transcurso de la carrera como: ingeniería del software y diversos conceptos de programación y lógica de programación; todo lo anterior debido a que fue necesario utilizar diversas herramientas como el lenguaje de programación Python y varias de sus librerías, Arduino, protocolos de comunicación, diseño de GUI's, además del manejo de errores y excepciones entre otros.

7 RECOMENDACIONES Y TRABAJOS FUTUROS

Como se pudo apreciar en la sección de pruebas, se tuvo el inconveniente de que la librería de Python PyFirmata que se había seleccionado inicialmente, no da soporte para la comunicación I2C en Arduino; por lo cual fue necesario hacer uso de otra librería de Python llamada PyMata que sí da soporte para este tipo de comunicación, pero esta no es muy eficiente con la lectura de puertos análogos y escritura en puertos digitales como sí lo hace PyFirmata, por lo que fue necesario hacer uso de las dos librerías.

Con base en lo anterior se recomienda si se desea modificar la herramienta desarrollada en este trabajo, buscar una librería más completa que cumpla los requisitos anteriores; esto con el objetivo de hacer la herramienta más liviana en cuanto a tamaño en bytes, pues entre más importaciones se hagan en el código fuente más pesada será; o en su defecto crear su propia librería, lo cual llevará más tiempo de investigación y codificación.

En cuanto a la definición de los almacenes de datos, esta herramienta está guardando los archivos como código del estudiante, evidencia entre otros; directamente en la base de datos, lo que puede generar mucho peso en la misma. Si se desea modificar esta herramienta se podría guardar estos archivos en un servidor y acceder a ellos por medio de la dirección en el mismo; pero se recomienda poner en la balanza seguridad versus eficiencia.

8 BIBLIOGRAFÍA

- Python Software Foundation. (2021). *pyFirmata*. Obtenido de <https://pypi.org/project/pyFirmata/>
- Alvear Puertas, V. (2017). *Sistema electrónico con aplicación IOT para monitoreo facial que brinde estimadores de desconcentración del estudiante universitario en el aula a escala de laboratorio*. Universidad Técnica del Norte, Imbabura. Ecuador. Obtenido de <http://repositorio.utn.edu.ec/handle/123456789/5873>
- Aparici, R. (2011). *PRINCIPIOS PEDAGÓGICOS Y COMUNICACIONALES DE LA EDUCACIÓN 2.0*. Madrid. Obtenido de http://www.educoas.org/portal/la_educacion_digital/laeducacion_145/articles/roberto_aparici.pdf
- Arcos, E. (20 de Octubre de 2014). *hipertextual*. Obtenido de <https://hipertextual.com/archivo/2014/10/internet-cosas/>
- Arduino. (2021). *What is Arduino?* Obtenido de <https://www.arduino.cc/en/Guide/Introduction>
- Basco, A. I., Beliz, G., Coatz, D., & Garnero, P. (2018). *INDUSTRIA 4.0: FABRICANDO EL FUTURO*. Buenos Aires. Obtenido de https://books.google.es/books?hl=es&lr=&id=geiGDwAAQBAJ&oi=fnd&pg=PA8&dq=industria+4.0+&ots=n_IU9owjNH&sig=TW2HUNx4eTgTaPpnjwKm2sK7L8I#v=onepage&q=industria%204.0&f=false
- Bruegge, B., & H. Dutoit, A. (2002). *Ingeniería del software orientado a objetos*. Mexico: Pearson Educación.
- Burd, B. B. (2017). Courses, content, and tools for internet of things in computer science education. *In Proceedings of the 2017 ITiCSE Conference on Working Group Reports*, 125-139.
- Burd, B. B. (Julio de 2018). The internet of things in CS education: updating curricula and exploring pedagogy. *In Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 366-367.
- del Val Roman, J. L. (2016). *Industria 4.0: la transformación digital de la industria*. Facultad de Ingeniería de la Universidad de Deusto. coddinforme. Obtenido de <http://coddii.org/wp-content/uploads/2016/10/Informe-CODDII-Industria-4.0.pdf>
- Diéguez, L. (s.f.). *Kolwidi*. Obtenido de <https://kolwidi.com/blogs/blog-kolwidi/programa-y-controla-con-python-tu-arduino>
- ELECTRONOBS en Español. (11 de Abril de 2020). *You Tube*. Obtenido de <https://www.youtube.com/watch?v=G7aQB6x0LHc>
- Firmata | Aprendiendo Arduino. (s.f.). Obtenido de <https://aprendiendoarduino.wordpress.com/2016/03/06/firmata/>
- Florez Olvera, D. M., Guzmán Gamez, F. J., Martínez Barragán, Y. M., Ibarra Cruz, E., & Albear Cortez, E. (2019). Educación 4.0, origen para su fundamentación. *CONFERENCE PROCEEDINGS 4th Virtual International Conference on Education, Innovation and ICT*. (EDUNOVATIC2019, Ed.) Benemérita Universidad Autónoma de Puebla, México, México: REDINE. Obtenido de

- https://www.researchgate.net/profile/Maria_Cristina_Fernandez-Laso/publication/339984851_Propuesta_de_innovacion_didactica_para_Educacion_Primer_a_basada_en_la_tecnologia_de_Realidad_Aumentada_aplicada_a_la_anatomia_humana_y_la_Prehistoria/links/5e70fc529
- Guerra Guerra, J. &. (June de 2017). Alignment of Undergraduate Curriculum for Learning IoT in a Computer Science Faculty. *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*, 362-362.
- itq.edu.mx. (s.f.). *Desarrollo de sistemas embebidos*. Obtenido de Comunicaciones digitales: Protocolos seriales(uC):
http://www.itq.edu.mx/carreras/IngElectronica/archivos_contenido/Apuntes%20de%20materias/ETD1022_Microcontroladores/4_SerialCom.pdf
- jecrespon. (s.f.). *Aprendiendo Arduino*. Obtenido de
<https://aprendiendoarduino.wordpress.com/2017/07/09/i2c/>
- Murcia Cucaita, J. G. (Julio de 2015). Fomento al uso de las TIC en la Universidad Cooperativa de Colombia sede Villavicencio, fortaleciendo la investigación y extensión, mediante redes sociales, android, arduino e internet of things. Villavicencio, Colombia. Obtenido de
<http://hdl.handle.net/20.500.12494/87>
- Oracle Corporation and/or its affiliates. (2021). *MySQL Connectors*. Obtenido de
<https://www.mysql.com/products/connector/>
- Ordiales, O. (25 de Abril de 2017). *Aula Robótica*. Obtenido de
<http://roboticclassroom.blogspot.com/2017/04/arduino-tipos-de-placas.html>
- Ortiz, A. (s.f.). *edupyton*. Obtenido de <http://edupython.blogspot.com/2014/06/como-programar-tu-arduino.html>
- Pacheco Gonzáles, A., Flores Avila, E., Cano Fuentes, P., & Tena Chávez, A. (2018). *Prototipo de un Aula Inteligente aplicando Internet de las Cosas y*. Instituto Tecnológico de Chihuahua, México.
- Pedroza flores, R. (2018). La universidad 4.0 con currículo inteligente 1.0 en la cuarta revolución industrial. *Revista Iberoamericana para la Investigación y el desarrollo Educativo*, 9(17), 64-90. Obtenido de http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S2007-74672018000200168
- Perez Romero, P., Rivera Zárate, I., & Hernandez Bolaños, M. (2019). La Educación 4.0 de Forma Simple. *Congreso Internacional de Educación*. Ciudad de México, Instituto Politécnico Nacional CIDETEC.
- pildorasinformaticas. (s.f.). *YouTube*. Obtenido de <https://www.youtube.com/watch?v=G2FCfQj-9ig&list=PLU8oAlHdN5BlvPxziopYZRd55pdqFwkeS>
- Python Software Foundation. (2021). *Interfaces gráficas de usuario con Tk*. Obtenido de
<https://docs.python.org/es/3/library/tk.html>
- Python Software Foundation. (2021). *PyMata*. Obtenido de <https://pypi.org/project/PyMata/>

- Python Software Foundation. (2021). *Python Documentation*. Obtenido de <https://docs.python.org/3/license.html>
- Rahman, M., Himanshi, Deep, V., & Rahman, S. (14 de Junio de 2016). TIC e internet de las cosas para crear un entorno de aprendizaje inteligente para estudiantes de institutos de educación en la India. Noida, India.
- Rodriguez Sala, J. J., Santamaria Arana, L., Rabasa Dolado, A., Martines Bonastre, O., Profesores Universidad Miguel Hernandez, Departamento de estadística y matemáticas aplicada, & Área de lenguajes y sistemas informáticos. (2003). *INTRODUCCIÓN A LA PROGRAMACION. TEORÍA Y PRÁCTICA*. San Vicente (Alicante) España: Editorial club universitario.
- Rosique, R. (s.f.). *Un asomo a la Educación y Web 1.0, 2.0 y 3.0*. Obtenido de academia.edu: https://d1wqtxts1xzle7.cloudfront.net/31001786/Un_asomo_a_la_Educacion_y_Web_1.pdf?1363875336=&response-content-disposition=inline%3B+filename%3DUn_asomo_a_la_Educacion_y_Web_1.0_2.0_y.pdf&Expires=1592537028&Signature=D-w4MspS7KLaqa4OWmp8yapwzr6cqLRj1HkM2
- Rueda Rueda, J. S., Manrique, J. A., & Cabrera Cruz, J. D. (Septiembre de 2017). *Researchgate*. Obtenido de <http://www.researchgate.net/publication/319914477>
- Sandoval Carrero, N. S., Santos Jaimes, L. M., & Acevedo, N. (Diciembre de 2020). Estrategia de aprendizaje basado en iot como apoyo a los cursos de programación en ingeniería. *Congreso internacional de electrónica y tecnología avanzada (Cieta)*. Pamplona, Norte de Santander, Colombia: Universidad de Pamplona.
- Sprogamacion. (s.f.). *YouTube*. Obtenido de <https://www.youtube.com/watch?v=VJmthHFJt8g&list=PLdLhejjdnsBbWIjHXmlyv0AJ85q1SMBt>
- Turnero, P. (s.f.). *comunicación de datos*. Obtenido de monografias.com: <https://www.monografias.com/trabajos106/comunicacion-datos-modos/comunicacion-datos-modos.shtml>
- UNIDAD POLITÉCNICA PARA LA EDUCACIÓN VIRTUAL. (2018). *Docente 4.0*. Obtenido de <https://docente.4-0.ipn.mx/index.php/educacuion-4-0/#:~:text=La%20educaci%C3%B3n%20avanz%C3%B3%20de%201.0,tendencias%20de%20innovaci%C3%B3n%20y%20cambio>.