

Proyecto Final IoT: Percepción

Objetivos Generales:

- Desarrollar competencias en tecnologías IoT aplicando metodologías ágiles.
- Fomentar el trabajo en equipo y la colaboración utilizando herramientas de desarrollo modernas.
- Implementar soluciones tecnológicas innovadoras que aborden problemas reales.

Objetivos de la Unidad:

- Desarrollar la capa de percepción para sistemas IoT.
- Seleccionar sensores e interfaces adecuadas para distintas aplicaciones.
- Prototipar y implementar el controlador de la capa física.

Desarrollo:

Generales:

1. Investigar sobre metodologías ágiles, Scrum y Kanban.
2. Crear cuentas en GitHub y formar grupos de 8 integrantes usando el [link de GitHub Classroom](#).
3. Instalar Git y Visual Studio Code (VsCode).
4. Instalar las extensiones necesarias en VsCode para el desarrollo del proyecto (PlatformIO, Python, Live Share, etc.).

Semana 1: (08/04 - 14/04)

- **Avance del Proyecto Final:** Introducción al IoT, formación de equipos. Primer contacto con herramientas de desarrollo y Git-GitHub.
- **Sprint 1:** Como integrantes del equipo, queremos familiarizarnos con el entorno de desarrollo y herramientas de colaboración para comenzar eficazmente nuestro proyecto IoT.
- **Sprint 1 Backlog:**
 1. Crear cuentas en GitHub y familiarizarse con la plataforma.
 2. Formar equipos de 8 integrantes y asignar roles preliminares.
 3. Instalar Git y configurarlo en las computadoras personales.

4. Instalar Visual Studio Code y explorar la interfaz.
5. Añadir las extensiones necesarias en VsCode (PlatformIO, Live Share, Python, etc.).
6. Realizar un tutorial básico de Git-GitHub para entender commits, branches, merges, y pull requests. [Link a Fundamentos de git-github](#)
7. Configurar un repositorio de equipo en GitHub y practicar con algunos commits y pull requests.
8. Asistir a un webinar o taller introductorio sobre IoT y sus aplicaciones.
9. Investigar sobre metodologías ágiles y diferencias entre Scrum y Kanban.
10. Seleccionar un proyecto de los 24 propuestos o proponer uno nuevo para aprobación del profesor.
11. Realizar una reunión inicial de equipo para discutir intereses y expectativas del proyecto.
12. Preparar una presentación breve de la visión del proyecto para compartir con el profesor y compañeros.

Semana 2: (15/04 - 21/04)

- **Avance del Proyecto Final:** Fundamentos de programación para IoT. Introducción a Arduino y Raspberry Pi.
- **Sprint 2:** Como equipo, queremos comprender los fundamentos de programación específicos para IoT y empezar a experimentar con módulos de desarrollo para establecer las bases de nuestro prototipo.
- **Sprint 2 Backlog:**
 1. Realizar un tutorial básico de programación en C++ con PlatformIO y Python con RTH.
 2. Instalar y configurar Arduino IDE y experimentar sobre Raspberry Pi OS.
 3. Experimentar con ejemplos simples de código para encender un LED con Arduino y Raspberry Pi.
 4. Investigar sobre los periféricos de comunicación de microcontroladores.
 5. Estudiar diferentes tipos de sensores, actuadores y visualizadores compatibles.
 6. Asignar tareas de investigación sobre casos de uso de sensores en proyectos IoT reales.
 7. Realizar una sesión de brainstorming para definir las funcionalidades clave del proyecto seleccionado.
 8. Documentar las especificaciones técnicas preliminares del proyecto en el repositorio de GitHub.
 9. Preparar un informe de avance para presentar al profesor con los resultados de los tutoriales y experimentos realizados.

10. Organizar una reunión de revisión de sprint para discutir el progreso y ajustar el plan de trabajo.
11. Crear un diagrama inicial del sistema propuesto, identificando componentes clave como sensores y actuadores.
12. Iniciar la creación de un kanban en GitHub Projects para organizar y priorizar las tareas de desarrollo.

Semana 3: (22/04 - 28/04)

- **Avance del Proyecto Final:** Sensores en IoT: tipos, selección y aplicación.
- **Sprint 3: Yo como** miembro del equipo de desarrollo del proyecto IoT, **Quiero** diseñar y desarrollar un módulo inicial para nuestro controlador que pueda leer datos de varios sensores y realizar una comunicación básica con actuadores, **Para** establecer una base sólida para el controlador de la capa de percepción, garantizando la precisión y efectividad en la recopilación de datos y la capacidad de respuesta inicial del sistema.
- **Sprint 3 Backlog:**
 1. Establecer especificaciones detalladas para el controlador, incluyendo requisitos de sensores y actuadores.
 2. Seleccionar los protocolos de comunicación adecuados para la interacción entre sensores, actuadores y el controlador.
 3. Diseñar el esquema del controlador con un enfoque en la integración y gestión de múltiples tipos de sensores.
 4. Desarrollar un módulo de software inicial en el módulo de desarrollo que pueda leer datos de varios sensores.
 5. Implementar una función de calibración para los sensores para asegurar la precisión de los datos recogidos.
 6. Realizar pruebas unitarias de los módulos de software para verificar la correcta lectura y procesamiento de datos de los sensores.
 7. Integrar una comunicación inicial con actuadores básicos (como LEDs o motores) para probar la capacidad de respuesta del controlador.
 8. Documentar el progreso y los códigos en el repositorio de GitHub, incluyendo diagramas de circuitos y fragmentos de código.
 9. Preparar una presentación breve del diseño del controlador y los resultados de las pruebas iniciales para la revisión del sprint.

Semana 4: (29/04 - 05/05)

- **Avance del Proyecto Final:** Taller práctico sobre sensores y actuadores.
- **Sprint 4: Yo como** miembro del equipo de desarrollo del proyecto IoT, **Quiero** finalizar el desarrollo de nuestro controlador, implementando completamente los protocolos de comunicación y asegurando una integración eficiente y efectiva con sensores y actuadores, **Para** completar con éxito la capa de percepción de nuestro proyecto, permitiendo que el sistema recopile, procese y responda a los datos de los sensores de manera precisa y confiable, y presentarlo funcionando a la clase y al profesor.
- **Sprint 4 Backlog:**
 1. Finalizar el desarrollo del controlador, asegurando que maneje eficientemente la recopilación de datos de los sensores y el control de actuadores.
 2. Implementar los protocolos de comunicación seleccionados para la interacción efectiva entre el controlador, sensores y actuadores.
 3. Optimizar el código del controlador para mejorar su eficiencia, confiabilidad y capacidad de respuesta.
 4. Realizar pruebas integrales del sistema para validar la funcionalidad completa del controlador en varios escenarios de uso.
 5. Ajustar y calibrar los sensores y actuadores basándose en los resultados de las pruebas para asegurar la precisión y la eficacia.
 6. Desarrollar una interfaz de usuario simple (si es aplicable) para monitorear y controlar la actividad de los sensores y actuadores a través del controlador.
 7. Documentar extensamente el proyecto finalizado, incluyendo el código fuente, esquemas de circuitos, y una guía de usuario.
 8. Crear un video demostrativo del controlador en acción, mostrando cómo interactúa con los sensores y actuadores.
 9. Organizar una presentación final para mostrar el proyecto completado al profesor y a los compañeros, incluyendo una demostración en vivo si es posible.
 10. Recopilar feedback durante la presentación para identificar posibles mejoras o extensiones para futuros proyectos.
 11. Realizar una retrospectiva del sprint para discutir los logros, los desafíos enfrentados y las lecciones aprendidas durante el desarrollo del proyecto.
 12. Planificar los pasos a seguir para continuar el desarrollo del proyecto en las próximas unidades, basándose en los resultados y el feedback recibido.

Metodología de Trabajo:

- Utilizaremos Visual Studio Code con PlatformIO (C++) o RTH (MicroPython) para el desarrollo. Se configurará VsCode con las extensiones necesarias y se integrará con Wokwi para la simulación.
- El trabajo en equipo seguirá metodologías ágiles, utilizando Scrum y Kanban. Los roles dentro del equipo se dividirán en Product Owner, ScrumMaster y Development Team.
- Cada sprint se centrará en las historias de usuario y tareas específicas para ese período, organizando el trabajo a realizar en un backlog y realizando reuniones de planificación y revisión para ajustar el enfoque según sea necesario.