

1. Documentación de Sprints

1.1. Sprint 1: Base Técnica y Recolección de Datos

Sprint Planning

Sprint Goal: En el primer sprint, el objetivo fue establecer la base técnica necesaria del hardware para trabajar con el proyecto, implementando el control usando el MPU-6050 y la ESP-32. Esto otorgaría el método para utilizar el control y entrenar el modelo junto con obtener la base inicial de los gestos.

Historias de Usuario Seleccionadas: Se seleccionó la HU-01: Como usuario, quiero tener la habilidad de controlar el autómata usando gestos predefinidos.

Estimaciones del Equipo: La estimación total del equipo fue de 10 SP.

Daily Scrum

Bitácora Consolidada por Miembro:

Haziel López (Hardware y Firmware):

- **¿Qué hice ayer?:** Preparé el código inicial y realicé las primeras conexiones con los dispositivos electrónicos que conforman el hardware.
- **¿Qué haré hoy?:** Me enfocaré en obtener el mecanismo de recolección de datos inicial para comenzar con las pruebas de captura de movimientos.
- **Impedimentos detectados:** Ninguno por el momento.

Jose Armando Pichal Pío (Diseño y Frontend):

- **¿Qué hice ayer?:** Trabajé en la conceptualización del proyecto y en definir los requisitos visuales.
- **¿Qué haré hoy?:** Empezar a trabajar en la idea de lo que sería el modelo del backend y el frontend.
- **Impedimentos detectados:** Ninguno.

Jose Camacho Castelán (Backend y Arquitectura):

- **¿Qué hice ayer?:** Realicé trabajo de conceptualización sobre la arquitectura general del sistema.
- **¿Qué haré hoy?:** Empezar a trabajar en la idea de lo que sería el modelo del backend y el frontend.
- **Impedimentos detectados:** Ninguno.

Sprint Review

Historias completadas: HU-01 (Parcial - La base de datos de gestos fue creada).

Comparación: planeado vs. logrado: Se logró el primer avance importante el segundo día del Sprint 1 al obtener el mecanismo de recolección de datos inicial usando el hardware de la "Varita" que se usará de control remoto. El plan se cumplió en cuanto a la conexión física y el flujo de datos. Se encontró el modelo que se usaría para las interfaces.

Evidencias del incremento funcional: El firmware de la ESP32 enviaba datos del sensor como una cadena separada por comas y el script de Python en la RPi demostró poder leer y separar estos valores.

Feedback del Product Owner: El PO validó el flujo de datos inicial como satisfactorio para iniciar la fase de entrenamiento de la IA.

Sprint Retrospective

Qué funcionó: El proceso de la recolección de datos es primitivo pero funcional. La rápida integración del hardware MPU-6050 con la ESP-32 permitió obtener la primera versión de la "Varita" de control remoto.

Qué no funcionó: El formato de datos actual (cadena separada por comas) es frágil ante la pérdida de conexión o latencia, y no es escalable para un sistema robusto.

Acciones de mejora claras y aplicables:

- **Acción:** Refactorizar el firmware de la Varita y el script de recepción en el Backend para usar un formato de datos más robusto (ej. JSON) que mitigue la fragilidad de la comunicación serial. (Responsable: Jose Camacho y Haziel López).
- **Acción:** Iniciar el diseño y la construcción de una carcasa para el control ("Varita") con el fin de proteger el hardware. (Responsable: Haziel López).

1.2. Sprint 2: Manejo de Datos y Modelos de IA

Sprint Planning

Sprint Goal: Enfocarse en la estabilidad de la comunicación y la creación de la arquitectura de clases del Backend para el procesamiento y manejo de los datos de los sensores, que serviría como piedra angular para la detección de gestos.

Historias de Usuario Seleccionadas: Se seleccionó la HU-08 (ver movimientos en tiempo real - 12 SP) y la HU-02 (manejar el autómata desde un panel de control - 8 SP).

Estimaciones del Equipo: La estimación total del equipo fue de 20 SP.

Daily Scrum

Bitácora Consolidada por Miembro:

Haziel López (Hardware y Comunicación):

- **¿Qué hice ayer?:** Trabajé en la estabilización del firmware y realicé pruebas de transmisión serial para asegurar la consistencia de los datos.
- **¿Qué haré hoy?:** Me enfocaré en la configuración del canal de comunicación estable entre la ESP32 y la Raspberry Pi mediante Serial/Python.
- **Impedimentos detectados:** Algunos problemas menores de latencia que hay que ajustar.

Jose Armando Pichal Pío (Diseño y Frontend):

- **¿Qué hice ayer?:** Nada. Diseñé de manera visual el FrontEnd. Investigué cómo funcionaba el lenguaje de CSS.
- **¿Qué haré hoy?:** Trabajar en el diseño visual del FrontEnd en HTML e implementar CSS.
- **Impedimentos detectados:** Lógica de diseño mal implementada.

Jose Camacho Castelán (Backend y Arquitectura):

- **¿Qué hice ayer?:** Definí el alcance del módulo de IA. Diseñé la arquitectura de las clases. **¿Qué haré hoy?:** Implementaré las clases base y el procesador de IA (AI Processor) para leer los CSVs.
- **Impedimentos detectados:** Ninguno.

Sprint Review

Historias completadas: HU-08 (Parcial - La base del Live Stream se intentó implementar).

Comparación: planeado vs. logrado: Se comenzó con la creación de un programa en Python simple para verificar la recepción de datos. Sin embargo, el modelo desde web (servidor ESP32) resultó ser poco práctico para obtener los datos de forma sencilla. Se optó por cambiar la arquitectura para que la lógica de clasificación y el backend completo se ejecuten en Python 3 en la Raspberry Pi.

Evidencias del incremento funcional: Se decidió desechar la arquitectura de servidor web en la ESP32 y migrar la lógica central a Python 3.

Feedback del Product Owner: El PO aceptó la decisión de pivotar la arquitectura para centralizar la lógica de negocio en la Raspberry Pi, priorizando la mantenibilidad sobre la arquitectura distribuida inicialmente.

Sprint Retrospective

Qué funcionó: La rápida decisión de desechar una arquitectura (servidor web ESP32) que no era viable a largo plazo. El diseño de la arquitectura de clases del AI Processor por parte de Camacho fue sólido.

Qué no funcionó: Se desperdiciaron horas intentando hacer funcionar el modelo de web-server en el microcontrolador.

Acciones de mejora claras y aplicables:

- **Acción:** Evitar complicarnos con hacer programas que se vean imponentes y mejor centrarse en que funcionen. Debemos considerar los costos monetarios y de tiempo, ya que desperdiciarlos en mejoras sin importancia costará más trabajo del necesario. (Responsable: Todo el equipo).
- **Acción:** Iniciar la migración completa a la Raspberry Pi e implementar las colas de comunicación (*threading*) para desacoplar la API del procesamiento de IA. (Responsable: Jose Camacho).

1.3. Sprint 3: Integración de Feedback (Hardware)

Sprint Planning

Sprint Goal: La meta fue enfocar la parte del hardware de feedback, logrando la conexión física entre los elementos electrónicos (OLED, Buzzer) y el backend dentro de la Raspberry Pi 4.

Historias de Usuario Seleccionadas: Se seleccionó la HU-03: Como usuario, quiero obtener feedback del autómata de forma que sepa si está funcionando o no.

Estimaciones del Equipo: La estimación del equipo fue de 7 SP, enfocada principalmente en el tiempo de manipulación física y configuración.

Daily Scrum

Bitácora Consolidada por Miembro:

Haziel López (Hardware y Conexiones):

- **¿Qué hice ayer?:** Comencé el trabajo de conexión y soldadura de los componentes OLED y Buzzer a la Raspberry Pi.
- **¿Qué haré hoy?:** Instalaré el sistema operativo necesario para la Raspberry Pi y realizaré la conexión correcta de todos los pines GPIO.
- **Impedimentos detectados:** Ninguno hasta el momento.

Jose Armando Pichal Pío (Soporte):

- **¿Qué hice ayer?:** Nada.
- **¿Qué haré hoy?:** Ayudé a mi compañero Camacho a investigar sobre la conectividad para el OLED.
- **Impedimentos detectados:** Ninguno.

Jose Camacho Castelán (Software y Configuración):

- **¿Qué hice ayer?:** Instalación del sistema operativo necesario en la RPi.
- **¿Qué haré hoy?:** Conexión de los pines a la pantalla y el buzzer.
- **Impedimentos detectados:** Afortunadamente conseguimos prestada la raspberry, ya que el precio se nos iba de las manos.

Sprint Review

Historias completadas: HU-03 (Parcial - Conexión de OLED completada, buzzer requiere código avanzado).

Comparación: planeado vs. logrado: Se logró conectar la pantalla OLED y se verificó el funcionamiento de los pines. Sin embargo, se determinó que el buzzer aún se tendrá que manejar diferente (con PWM para generar tonos) y no solo con ON/OFF, lo cual es trabajo de software que se pospuso.

Evidencias del incremento funcional: Se verificó la conexión de la pantalla OLED mediante la ejecución de un script de prueba que mostró imágenes estáticas.

Feedback del Product Owner: El PO indicó que este trabajo físico es fundamental para la usabilidad, pero solicitó que el siguiente sprint priorizara la lógica de software para integrar el feedback con el reconocimiento de gestos.

Sprint Retrospective

Qué funcionó: La instalación y configuración de la RPi fue rápida. La conexión a la pantalla fue exitosa.

Qué no funcionó: Este paso no ocupó nada de código, solo fue tener cuidado con las piezas y soldar lo necesario.

Acciones de mejora claras y aplicables:

- **Acción:** Para futuros sprints, combinar la adquisición/conexión de hardware con la implementación de software de bajo nivel para optimizar la duración. (Responsable: Scrum Master).
- **Acción:** Implementar la clase BuzzerController con PWM y la clase AnimatedOLED para manejar la retroalimentación visual animada. (Responsable: Jose Camacho y Haziel López).

1.4. Sprint 4: Desarrollo de la IA y Lógica IoT

Sprint Planning

Sprint Goal: La meta es integrar la lógica de la Inteligencia Artificial con el Backend de Flask, entrenar los modelos de gestos clave para el control IoT y establecer las bases de la comunicación MQTT.

Historias de Usuario Seleccionadas: Se seleccionó la HU-05 (controlar luces con "Lumos 5 SP) y la HU-06 (controlar volumen con Ascendio 5 SP).

Estimaciones del Equipo: La estimación del equipo para este sprint fue de 10 SP.

Daily Scrum

Bitácora Consolidada por Miembro:

Haziel López (Hardware y Comunicación):

- **¿Qué hice ayer?:** Continué la recolección de datos de gestos y estabilicé el firmware para los nuevos movimientos que serán reconocidos.
- **¿Qué haré hoy?:** Asistiré a Camacho en la calibración de los movimientos para asegurar su correcta detección por parte del modelo.
- **Impedimentos detectados:** Ninguno por ahora.

Jose Armando Pichal Pío (Soporte):

- **¿Qué hice ayer?:** Investigué librerías para el frontend y estudié la implementación de MQTT para la comunicación IoT.
- **¿Qué haré hoy?:** Apoyaré en la calibración y trabajaré en el diseño inicial de la interfaz de logs y stream de datos.
- **Impedimentos detectados:** Ninguno.

Jose Camacho Castelán (Backend y IA):

- **¿Qué hice ayer?:** Implementé la clase Classifier y la lógica de threading para el Backend.
- **¿Qué haré hoy?:** Entrenaré los modelos principales como Lumos, Nox, Ascendio y Descendo. En la práctica creé bien la interfaz lista para la producción de modelos de IA, así que fue más fácil de lo esperado.
- **Impedimentos detectados:** Calibrar los movimientos de la varita física para su correcta detección.

Sprint Review

Historias completadas: HU-05 y HU-06 (Ambas completadas en el área de software con modelos entrenados).

Comparación: planeado vs. logrado: El objetivo se cumplió exitosamente. Los modelos de gestos para control IoT fueron entrenados y guardados correctamente. La arquitectura de threading en el archivo principal permite ejecutar la IA sin bloquear el servidor web. La publicación de mensajes MQTT quedó pendiente para el siguiente sprint.

Evidencias del incremento funcional: Se demostró la existencia de los archivos de modelos serializados y la respuesta de la consola de la API de Flask al detectar un gesto correctamente.

Feedback del Product Owner: El PO validó que la precisión inicial de los modelos era aceptable para continuar y solicitó centrar el siguiente sprint en la interfaz para que el usuario final pueda interactuar con el sistema.

Sprint Retrospective

Qué funcionó: La decisión de usar threading resolvió el problema de rendimiento y latencia. El entrenamiento fue rápido gracias a la arquitectura de clases definida.

Qué no funcionó: La calibración del sensor y la recolección de datos sigue siendo la parte más manual y laboriosa.

Acciones de mejora claras y aplicables:

- **Acción:** Implementar el "Historial de gestos" la visualización gráfica del movimiento en la UI para ayudar en futuras fases de entrenamiento (HU-08). (Responsable: Jose Camacho).
- **Acción:** Iniciar el desarrollo de la interfaz de usuario web y la lógica de login. (Responsable: Jose Armando).

1.5. Sprint 5: Panel de Control Web y Gestión de Usuarios

Sprint Planning

Sprint Goal: Entregar el Panel de Control web funcional y listo para el usuario, incluyendo la gestión de acceso (login), la visualización del estado y la funcionalidad de tutoriales.

Historias de Usuario Seleccionadas: Se seleccionó la HU-04 (visualizar estado del autómata - 8 SP) y la HU-09 (tener tutoriales en el panel - 3 SP).

Estimaciones del Equipo: La estimación del equipo para este sprint fue de 11 SP.

Daily Scrum

Bitácora Consolidada por Miembro:

Haziel López (Hardware y Pruebas):

- **¿Qué hice ayer?:** Me enfoqué en el diseño y la construcción de la carcasa inicial del control y el autómata para proteger los componentes.

- **¿Qué haré hoy?:** Realizaré pruebas de integración de punta a punta conectando el hardware con el backend y el frontend operativo.
- **Impedimentos detectados:** Ninguno por el momento.

Jose Armando Pichal Pío (Diseño y Frontend):

- **¿Qué hice ayer?:** Nada.
- **¿Qué haré hoy?:** Desarrollo correcto de la lógica del diseño para el FrontEnd. Implementé la base de datos SQLite para el guardado de los datos del usuario.
- **Impedimentos detectados:** Ninguno.

Jose Camacho Castelán (Backend y Lógica):

- **¿Qué hice ayer?:** Implementé el "Historial de gestos." en la UI, usando la lógica de tablas.
- **¿Qué haré hoy?:** Finalizaré la lógica de login/registro en el archivo principal y la implementación de la visualización gráfica del movimiento (Live Stream).
- **Impedimentos detectados:** Solucionar problemas con un error desaparecido al momento de integrar la lógica de envío de datos a la interfaz en el módulo de entrenamiento.

Sprint Review

Historias completadas: HU-04 y HU-09 (Funcionalidad de login, log, Live Stream e interfaz lista).

Comparación: planeado vs. logrado: El objetivo se cumplió satisfactoriamente. El Panel de Control web es completamente funcional, accesible y protegido por el sistema de login y registro. La visualización del estado mediante logs y conexión funciona perfectamente a través de las colas de polling implementadas en Flask. La sección de tutoriales fue integrada de manera sencilla.

Evidencias del incremento funcional: El Panel de Control muestra la tabla de historial de gestos y el gráfico de movimiento en tiempo real del sensor, confirmando la conexión estable y bidireccional entre el hardware y el frontend.

Feedback del Product Owner: El PO solicitó refinar el estilo visual del frontend y enfatizar los mensajes de error en el estado del autómata, pero aprobó la funcionalidad.

Sprint Retrospective

Qué funcionó: La implementación de la base de datos SQLite para login fue fluida. El trabajo conjunto entre el backend y el frontend para el Live Stream fue muy eficiente.

Qué no funcionó: La optimización de la latencia en el Live Stream requiere aún más trabajo.

Acciones de mejora claras y aplicables:

- **Acción:** Agregar más estilos CSS y animaciones al frontend para mejorar la experiencia del usuario y satisfacer el feedback del PO. (Responsable: Jose Armando).
- **Acción:** Realizar el ajuste fino de los modelos de gestos y disminuir el umbral de detección para reducir falsos negativos. (Responsable: Jose Camacho).

1.6. Sprint 6: Integración Final y Cierre del Producto

Sprint Planning

Sprint Goal: Completar los comandos de juego y de apagado final, lograr la sincronización perfecta de feedback (OLED y Buzzer) con la detección de la IA, y preparar el producto para su entrega final.

Historias de Usuario Seleccionadas: Se seleccionó la HU-07 (apagar con .^vada Kedavra 2 SP) y la HU-10 (inmovilizar con "Petrificus Totalus 3 SP").

Estimaciones del Equipo: La estimación del equipo para este sprint fue de 5 SP.

Daily Scrum

Bitácora Consolidada por Miembro:

Haziel López (Hardware y Pruebas):

- **¿Qué hice ayer?:** Verifiqué la conexión completa del sistema para la lógica de apagado seguro y asistí en la refactorización de los modelos de IA.
- **¿Qué haré hoy?:** Realizaré pruebas finales de todos los comandos de juego implementados y prepararé la documentación técnica del hardware para la entrega final.
- **Impedimentos detectados:** Ninguno, todo marcha según lo planeado.

Jose Armando Pichal Pío (Diseño y Documentación):

- **¿Qué hice ayer?:** Finalicé el diseño de la interfaz y la optimización de estilos.
- **¿Qué haré hoy?:** Trabajar en la documentación del diseño y el manual de usuario, así como en la Bitácora de Cambios.
- **Impedimentos detectados:** Ninguno.

Jose Camacho Castelán (Backend y Lógica):

- **¿Qué hice ayer?:** Entrené los modelos principales y refactoricé todos los modelos para que sean más sencillos de utilizar.
- **¿Qué haré hoy?:** Implementar la clase de integración de feedback y la lógica de apagado en el backend. Puedes poner que hubo tests.
- **Impedimentos detectados:** Cansancio, pero deseo dejar los modelos finos hoy.

Sprint Review

Historias completadas: HU-07 y HU-10 (Completadas, el producto está listo para ser entregado).

Comparación: planeado vs. logrado: Se logró el objetivo completamente. La lógica de feedback para todos los comandos, incluido el comando de apagado con animación de calavera y sonido, está operativa. El producto está listo para la entrega. La única parte pendiente es la conexión de los servos al autómata para que se mueva físicamente, la cual quedó como trabajo futuro.

Evidencias del incremento funcional: El sistema demuestra la secuencia completa: el gesto `.^vada Kedavra.` detectado correctamente, la pantalla OLED muestra la animación de la calavera, el buzzer reproduce el sonido de power-down, y se ejecuta el comando de apagado seguro en la Raspberry Pi.

Feedback del Product Owner: El PO aprueba la funcionalidad central, destacando la usabilidad de la interfaz y la calidad del feedback implementado.

Sprint Retrospective

Qué funcionó: La integración final del software con el hardware fue fluida gracias a la planificación anticipada de las clases (OLED y Buzzer). La sincronización de feedback visual y auditivo mediante la clase integradora fue exitosa.

Qué no funcionó: La falta de un framework de pruebas automatizadas para la detección de gestos obligó a realizar muchas pruebas manuales (físicas) por cada modelo entrenado.

Acciones de mejora claras y aplicables:

- **Acción:** Investigar frameworks de pruebas unitarias para Python que puedan simular la entrada de datos seriales del sensor, con el fin de agilizar la calibración de la IA. (Responsable: Jose Camacho).
- **Acción:** Documentar la arquitectura y el código de manera exhaustiva para facilitar la transferencia de conocimiento. (Responsable: Todo el equipo).