

UNIVERSIDAD TECNOLÓGICA DE XICOTEPEC DE JUÁREZ

(Tamaño 12)



“ClothesGuard”

(Tamaño 16)

TESINA QUE PRESENTA

(Tamaño 12)

NOMBRE DEL LOS ESTUDIANTE

*Uriel Abdallah Medina Torres
Karen Lizbeth Negrete Hernández
Jonathan Baldemar Ramírez Reyes
Christian Paul Rodríguez Pérez*

(Tamaño 12)

***TÉCNICO SUPERIOR UNIVERSITARIO
EN DESARROLLO DE SOFTWARE
MULTIPLATAFORMA***

AGOSTO, 2025

DEDICATORIA

Este proyecto está dedicado a todas aquellos que apoyan y creen en uso de tecnologías innovadoras para facilitar nuestra vida cotidiana.

Gracias por motivarnos a seguir creando soluciones que combinan creatividad, sostenibilidad y tecnología.

DRIGYNIE

AGRADECIMIENTOS

El equipo desea expresar su más sincero agradecimiento a todas las personas que, de una u otra forma, contribuyeron al desarrollo y culminación de este proyecto.

Agradecemos especialmente a quienes nos brindaron su orientación académica y profesional, compartiendo sus conocimientos y experiencia para guiarnos en cada etapa del proceso.

Las palabras nunca serán suficientes para agradecer a las personas que nos apoyan en nuestras vidas.

Este logro es también de ustedes.

Gracias por acompañarnos en este camino.

Con aprecio,

DRIGYNIE

Resumen

El proyecto del Tendedero Inteligente surge para solucionar las dificultades que enfrentan comunidades como Xicotepec de Juárez, donde el clima húmedo y las lluvias frecuentes complican el secado de la ropa mediante métodos tradicionales. La solución propuesta integra tecnologías modernas que automatizan el proceso, permitiendo un control eficiente y adaptado a las condiciones ambientales en tiempo real.

- **Necesidad:**

El sistema responde a la carencia de soluciones automatizadas para optimizar el secado de ropa en condiciones climáticas adversas. Con la detección temprana de factores como humedad, lluvia y temperatura, se evita la intervención manual constante y se protege la calidad de las prendas.

- **Metodología:**

Se desarrollará un prototipo modular y compacto, equipado con sensores de humedad y temperatura. El proyecto se gestionará utilizando la metodología Scrum, lo que permitirá iteraciones ágiles y la integración continua de mejoras basadas en el feedback del usuario. Además, el dispositivo se conectará a una aplicación móvil para facilitar el control remoto y la automatización del proceso.

- **Objetivo Principal:**

Optimizar el proceso de secado de ropa en ambientes adversos, ajustando automáticamente el funcionamiento del tendedero según las condiciones climáticas. Esto se traduce en una mayor eficiencia, reducción del consumo energético y preservación de las prendas, proporcionando una solución práctica, económica y accesible tanto para contextos urbanos como rural

Abstract

The Intelligent Clothesline (ClothesGuard) project addresses the challenges encountered in drying clothes under adverse climatic conditions, such as high humidity and frequent rain, as experienced in communities like Xicotepec de Juárez. Traditional drying methods depend heavily on favorable weather and manual intervention, often resulting in inefficient drying and potential damage to garments. This project presents an innovative, automated solution that optimizes the drying process by integrating modern IoT technologies.

A modular and compact prototype was developed incorporating an ESP32 microcontroller, which serves as the central processing unit, along with sensors for temperature, humidity, and rain detection. A motorized mechanism, controlled by the microcontroller, automatically extends or retracts the clothesline in response to environmental data. Communication between the hardware and user interfaces is established via WebSockets, enabling real-time data transmission to a mobile application and a web platform. The software side was developed using agile methodologies, with frameworks such as React Native and Angular, and a Node.js backend connected to a MongoDB database for efficient data handling.

Field and laboratory tests validated the functionality and reliability of the system, confirming that the automated process not only enhances the efficiency of clothes drying but also reduces energy consumption and prolongs the lifespan of garments. The prototype's modular design further ensures ease of installation and scalability for broader applications.

In summary, ClothesGuard demonstrates that the integration of automated control and real-time environmental monitoring can provide a practical, cost-effective, and sustainable alternative to traditional clothes drying methods. This solution has the potential to significantly improve household routines and support energy-efficient practices in both urban and rural settings.

Tabla de Contenidos

Capítulo 1 Introducción e información general	1
Planteamiento del Problema	1
Objetivo General del Proyecto.....	2
Objetivos Específicos.....	3
Justificación del Proyecto.	4
Limitaciones y Alcances.....	¡Error! Marcador no definido.
Capítulo 2. MARCO TEÓRICO Y MARCO CONTEXTUAL.	6
Marco Teórico.....	6
Marco Contextual.....	16
Capítulo 3 Metodología y Desarrollo	18
Capítulo 4 Resultados y Conclusiones.....	32
Resultados	¡Error! Marcador no definido.
Conclusiones	¡Error! Marcador no definido.
Recomendaciones	32
Lista de referencias	34
Apéndice	36
Vita.....	¡Error! Marcador no definido.

Tabla de Figuras

Ilustración 1. Tendedero tradicional de ropa	19
Ilustración 2. Sensor de lluvia YL- 83	20
Ilustración 3 Sensor de temperatura y humedad DHT11	21
Ilustración 4. Motor a pasos 28BYJ-48 (5V).....	21
Ilustración 5. Microcontrolador ESP32	21
Ilustración 6. MongoDB	21
Ilustración 7, código Arduino de nuestro proyecto.....	23
Ilustración 8 Wireframes de la api mobile en figma.....	24
Ilustración 9 Configuración de pines en el código Arduino	25
Ilustración 10 Conexión con una red de Internet	25
Ilustración 11Mensaje de conexión a red Wifi	25
Ilustración 12. Función siendo llamada de forma constante dentro de setup().....	26
Ilustración 13. Motor a pasos 1 ensamblado en un poste	29
Ilustración 14. Poste Principal con Sensor de agua, Distancia, Humedad y Motor a pasos	29
Ilustración 15. Vista aérea de la maqueta ensamblada	30

Capítulo 1

Introducción e información general

El proyecto de Tendedero Inteligente busca desarrollar una solución innovadora para optimizar el proceso de secado de ropa, especialmente en comunidades como Xicotepec de Juárez, donde el clima húmedo y las lluvias frecuentes dificultan esta tarea cotidiana. Este tendedero está diseñado para adaptarse a las condiciones climáticas cambiantes, aprovechando tecnologías modernas para automatizar el proceso de secado y facilitar el control por parte del usuario.

El tendedero inteligente estará equipado con sensores de humedad y temperatura que permitirán detectar las condiciones climáticas en tiempo real. A través de una conexión a internet o una aplicación móvil, el usuario podrá controlar el tendedero, ajustando su funcionamiento según las necesidades específicas, como la activación de un sistema de ventilación o la configuración de alarmas para notificar el tiempo estimado de secado.

Además, el diseño del tendedero será modular y compacto, permitiendo su uso en espacios pequeños, lo que lo hace ideal para viviendas en zonas urbanas o rurales con limitaciones de espacio. Se utilizarán materiales resistentes y de bajo consumo energético, contribuyendo a la sostenibilidad ambiental al evitar el uso excesivo de secadoras eléctricas que generan altos costos de energía. Este proyecto tiene como objetivo no solo mejorar la eficiencia del proceso de secado de ropa, sino también ofrecer una solución accesible, económica y amigable con el medio ambiente, adaptándose a las particularidades del clima y las necesidades de los usuarios.

Planteamiento del problema

El secado tradicional de la ropa mediante un tendedero convencional presenta diversas limitaciones. En muchos hogares, el proceso requiere la intervención manual para monitorear y retirar la ropa, lo que implica depender de condiciones climáticas favorables y de la disponibilidad de tiempo. Además, la exposición prolongada a la intemperie en condiciones inadecuadas —como lluvia o alta humedad— puede deteriorar la calidad de las prendas. Actualmente, no existe un sistema integrado que automatice este proceso basado en variables ambientales o que permita la intervención manual en momentos oportunos. Surge, por tanto, la necesidad de desarrollar un tendedero inteligente que, mediante la integración de sensores y programación, detecte de forma automática condiciones adversas (lluvia, exceso de humedad o transcurrido un tiempo determinado) y active el proceso de secado, o bien, responda a comandos directos del usuario para iniciar o detener la acción.

Objetivo General del Proyecto.

Desarrollar un tendedero inteligente que optimice el proceso de secado de ropa en condiciones climáticas adversas, como la humedad y las lluvias frecuentes en Xicotepec de Juárez, mediante el uso de tecnologías innovadoras que permitan automatizar el control de temperatura y humedad, reduciendo el consumo energético, mejorando la eficiencia del secado y ofreciendo una solución accesible y sostenible para los usuarios.

Objetivos Específicos.

- Diseñar y desarrollar un prototipo de tendedero inteligente que utilice sensores de humedad y temperatura para ajustar el proceso de secado según las condiciones climáticas.
- Implementar un sistema de control automático y manual a través de una aplicación móvil, que permita al usuario monitorear y gestionar el funcionamiento del tendedero en tiempo real.
- Optimizar el uso de energía del tendedero inteligente, utilizando tecnologías de bajo consumo energético, como ventiladores eficientes y aprovechamiento de energía solar cuando sea posible.
- Crear un diseño modular y adaptable que permita el uso del tendedero en distintos tipos de viviendas, especialmente en espacios reducidos, garantizando su funcionalidad y facilidad de instalación.
- Evaluar el rendimiento del tendedero en diversas condiciones climáticas, realizando pruebas de secado en escenarios con alta humedad y lluvias, y ajustando el prototipo según los resultados obtenidos.
- Contribuir a la sostenibilidad ambiental al ofrecer una alternativa de secado eficiente que reduzca la dependencia de secadoras eléctricas, minimizando el impacto ambiental y el consumo de energía.

Justificación del Proyecto.

El secado tradicional de la ropa mediante un tendedero convencional suele presentar diversas limitaciones ya que en muchos hogares este proceso requiere la intervención manual para monitorear y retirar la ropa, lo que esto implica depender de las condiciones climáticas favorables y la disponibilidad de tiempo, la necesidad de desarrollo de un tendedero inteligente representa una solución innovadora y práctica frente a los métodos tradicionales de secado. Este sistema no solo optimizaría el proceso, reduciendo la intervención manual y el tiempo de espera, sino que también protegería las prendas de posibles daños ocasionados por condiciones ambientales desfavorables.

Entre los beneficios destacados se encuentran:

- **Eficiencia y Automatización:** Al activar el secado de forma automática según parámetros ambientales, se garantiza una mayor eficiencia en el uso del tiempo y en el manejo de la ropa.
- **Protección de las Prendas:** La detección de condiciones como la lluvia o la alta humedad permite evitar la exposición prolongada de la ropa a factores que pueden deteriorarla.
- **Adaptabilidad y Confort:** La posibilidad de intervenir manualmente a través de un comando, sumada a la respuesta automática del sistema, proporciona flexibilidad y mejora la experiencia del usuario.
- **Innovación y Sostenibilidad:** La integración de tecnologías inteligentes en el hogar no solo moderniza las tareas cotidianas, sino que también contribuye a un uso más racional de recursos y energía, fomentando prácticas sostenibles.

Alcances

1. Automatización del secado – Sistema que ajusta la exposición al sol o activa un mecanismo de secado eléctrico según la humedad detectada.
2. Sensores climáticos – Capacidad de detectar lluvia, temperatura y humedad para retraer la ropa automáticamente en caso de mal clima.
3. Multiplataforma – Compatible con aplicaciones móviles en distintos sistemas operativos (iOS, Android)
4. Adaptabilidad a las circunstancias – Puede configurarse según necesidades específicas, como diferentes tipos de ropa o condiciones climáticas.
5. Control remoto – Posibilidad de gestionar el tendido y recogida de ropa a distancia mediante una aplicación.
6. Eficiencia energética – Puede usar energía solar o sistemas de bajo consumo para reducir el impacto ambiental.

Limitaciones

1. Peso – Capacidad máxima de carga limitada, lo que puede restringir la cantidad de ropa que se puede tender a la vez.
2. Energía (Necesita electricidad) – Depende de una fuente de energía eléctrica, lo que puede afectar su funcionamiento en caso de cortes de luz.
3. Distancia – Puede haber restricciones en la comunicación con la aplicación o el control remoto si está fuera del alcance de la red Wi-Fi o Bluetooth.
4. Espacio necesario – Requiere un área adecuada para su instalación, lo que puede ser un problema en espacios pequeños.
5. Mantenimiento – Los sensores, motores y mecanismos necesitan revisiones periódicas para garantizar su correcto funcionamiento.
6. Durabilidad de los componentes electrónicos – Riesgo de desgaste o fallas debido a la exposición al sol, la humedad y otros factores externos.

Capítulo 2. MARCO TEÓRICO Y MARCO CONTEXTUAL.

Marco Teórico

El proyecto busca desarrollar un sistema automatizado para un tendedero inteligente, orientado a optimizar el proceso de secado de ropa en entornos con condiciones climáticas adversas. La solución propuesta integra sensores de humedad y temperatura, que permiten determinar el momento óptimo para el secado, y módulos de conectividad que facilitan el monitoreo en tiempo real. Asimismo, se implementará una plataforma web y móvil para que los usuarios puedan controlar el funcionamiento del tendedero, ajustar parámetros, recibir alertas sobre cambios climáticos y visualizar datos clave, lo que contribuye a preservar la calidad de las prendas y a reducir el consumo energético. La gestión del proyecto se apoyará en metodologías ágiles, como SCRUM, favoreciendo iteraciones rápidas y una colaboración efectiva entre los miembros del equipo.

1. METODOLOGÍA

La metodología es un conjunto estructurado de procedimientos, técnicas y herramientas que se utilizan para guiar el desarrollo de una investigación o proyecto. No se limita únicamente a la selección de métodos, sino que abarca el planteamiento general, la planificación, la ejecución y la evaluación de un proceso, permitiendo que los resultados sean consistentes y verificables [1].

Sirve para asegurar que la investigación o proyecto se realice de manera organizada y sistemática. Esto permite que los hallazgos sean confiables y que otros investigadores o equipos de desarrollo puedan replicar el proceso, lo cual es fundamental para la validación científica y la mejora continua en el ámbito de la investigación y la gestión de proyectos [2].

Características

- **Sistematicidad:** Sigue un conjunto de pasos ordenados y estructurados para garantizar la coherencia del estudio [1].
- **Objetividad:** Busca reducir sesgos mediante procedimientos verificables y replicables [2].
- **Flexibilidad:** Se adapta a diferentes contextos según la naturaleza del estudio o proyecto [3].
- **Validez y confiabilidad:** Asegura que los resultados sean precisos y puedan ser utilizados para la toma

de decisiones [4].

- Reproducibilidad: Permite que otros investigadores o desarrolladores sigan el mismo proceso y obtengan resultados comparables [5].

2. METODOLOGÍAS EN PROGRAMACIÓN

Las metodologías en programación son marcos de trabajo y enfoques estructurados que guían el proceso de desarrollo de software. Estas metodologías definen fases o etapas que van desde la planificación, el análisis de requerimientos, el diseño, la implementación, las pruebas hasta el mantenimiento. No solo se enfocan en la codificación, sino también en la documentación, la gestión de cambios y el aseguramiento de la calidad [6].

Sirven para organizar y estandarizar el proceso de desarrollo, facilitando la colaboración en equipos, mejorando la calidad del producto final y permitiendo una gestión más predecible y controlada de los proyectos de software. Esto se traduce en una reducción de errores, mayor eficiencia y la posibilidad de adaptarse a cambios en los requerimientos del proyecto [7].

Características:

- Estructuración del proceso: Definen etapas secuenciales o iterativas (análisis, diseño, implementación, pruebas y mantenimiento) [6,7].
- Documentación: Promueven la elaboración de documentación detallada que facilita la comunicación y la gestión [6].
- Control de calidad: Incluyen estrategias de pruebas y revisiones para garantizar el correcto funcionamiento del producto [7].
- Gestión de cambios: Permiten adaptarse ante modificaciones en los requerimientos [6].
- Flexibilidad versus rigidez: Mientras algunos modelos (como el cascada) son más rígidos, las metodologías ágiles permiten respuestas rápidas a imprevistos [7].

3. SCRUM

Scrum es un marco de trabajo ágil diseñado para gestionar proyectos complejos, principalmente

en el desarrollo de software. Se fundamenta en ciclos iterativos llamados sprints y establece roles, eventos y artefactos que estructuran el proceso de desarrollo de manera colaborativa y transparente [Schwaber & Sutherland, 2020].

Sirve para mejorar la capacidad del equipo de responder a cambios rápidos y entregar de manera continua incrementos de valor en el producto final. Permite una comunicación fluida y una retroalimentación constante, lo que facilita la adaptación y la mejora continua durante el desarrollo del proyecto [Schwaber & Sutherland, 2020].

Características:

- Roles definidos:
 - Scrum Master: Facilita el proceso y elimina impedimentos [Schwaber & Sutherland, 2020].
 - Product Owner: Define y prioriza los requisitos del producto [Schwaber & Sutherland, 2020].
 - Development Team: Se encarga del desarrollo del producto [Schwaber & Sutherland, 2020].
- Eventos clave: Incluyen la planificación del sprint, reuniones diarias (daily scrum), revisión y retrospectiva del sprint [Schwaber & Sutherland, 2020].
- Artefactos: Se incluyen el Product Backlog, el Sprint Backlog y el Incremento [Schwaber & Sutherland, 2020].
- Transparencia y adaptación: Basado en la inspección continua y la adaptación a cambios [Schwaber & Sutherland, 2020].

4. FIGMA

Figma es una herramienta de diseño de interfaces basada en la nube que permite a los usuarios crear, prototipar y colaborar en el desarrollo de productos digitales en tiempo real.

Sirve para facilitar la creación y validación de diseños de interfaces de usuario. Al estar basado en la nube, permite que múltiples miembros del equipo (diseñadores, desarrolladores y otros stakeholders) trabajen de manera simultánea, compartan comentarios y realicen modificaciones en tiempo real, lo que agiliza el proceso creativo y de toma de decisiones.

Características:

- Interfaz intuitiva: Ofrece una experiencia amigable que potencia la creatividad.
- Diseño vectorial: Permite la creación y manipulación de gráficos escalables.
- Prototipado interactivo: Facilita la simulación de interacciones de usuario.
- Colaboración en tiempo real: Varios usuarios pueden trabajar simultáneamente.
- Acceso basado en la nube: Los proyectos se pueden gestionar desde cualquier dispositivo.
- Integración de bibliotecas de componentes: Favorece la reutilización y coherencia visual.

5. WIREFRAMES

Los wireframes son representaciones esquemáticas y simplificadas de la estructura y el diseño de una interfaz digital. Actúan como un plano o boceto que define la disposición de los elementos sin incluir detalles de diseño final.

Sirven para visualizar y planificar la arquitectura de una aplicación o sitio web antes de invertir tiempo en el desarrollo de gráficos y contenido definitivo. Facilitan la comunicación entre diseñadores, desarrolladores y clientes, permitiendo iterar y validar la estructura y funcionalidad de la interfaz en etapas tempranas.

Características:

- Estructura visual básica: Representan la jerarquía y distribución de elementos.
- Simplicidad: Se centran en la funcionalidad y organización, sin distracciones de diseño.
- Herramienta de comunicación: Facilitan la discusión y validación de ideas entre los involucrados.
- Iteración rápida: Permiten ajustar el diseño antes de invertir en detalles gráficos[9].

6. ARDUINO

Arduino es una plataforma de electrónica de código abierto que integra hardware (placas con microcontroladores) y software (entorno de desarrollo integrado o IDE) para facilitar la creación de prototipos interactivos.

Sirve para desarrollar proyectos de electrónica de manera sencilla, permitiendo a usuarios de todos los niveles experimentar y construir dispositivos que interactúan con el entorno físico. Es ampliamente utilizado en educación, arte digital, automatización y proyectos de Internet de las Cosas (IoT).

Características:

- Hardware: Placas electrónicas con microcontroladores (por ejemplo, el ATmega328 en el Arduino Uno) que cuentan con pines de entrada/salida para conectar sensores y actuadores.
- Arduino IDE: Entorno de desarrollo integrado que permite escribir, compilar y cargar código en la placa, facilitando la experimentación y desarrollo [10].

7. MONGODB

MongoDB es un sistema de gestión de bases de datos NoSQL orientado a documentos que almacena la información en un formato similar a JSON, utilizando el formato BSON para mayor eficiencia [MongoDB, 2023].

Sirve para gestionar y almacenar grandes volúmenes de datos de manera flexible y escalable. Es ideal para aplicaciones que requieren manejar datos semiestructurados o no estructurados, como redes sociales, análisis de datos en tiempo real, y aplicaciones web de alto rendimiento [MongoDB, 2023].

Características:

- Modelo de datos flexible: Permite esquemas dinámicos y anidados gracias al uso de BSON [MongoDB, 2023].
- Escalabilidad horizontal: Utiliza particionamiento (sharding) para distribuir datos en múltiples servidores [MongoDB, 2023].
- Alta disponibilidad: Implementa réplicas para garantizar la redundancia y la continuidad del servicio [MongoDB, 2023].

- Índices avanzados: Optimiza el rendimiento en las consultas [MongoDB, 2023].
- Consultas ricas: Soporta operaciones de agregación y búsqueda de texto [MongoDB, 2023].
- Compatibilidad: Proporciona controladores para diversas plataformas y lenguajes [MongoDB, 2023].

8. LENGUAJE DE PROGRAMACIÓN

Un lenguaje de programación es un sistema formal de comunicación que permite a los desarrolladores escribir instrucciones y algoritmos para que una computadora ejecute tareas específicas [1]. Cada lenguaje posee su propia sintaxis y semántica, definiendo cómo se expresan las operaciones y es Sirve para construir programas y sistemas que automatizan procesos, resuelven problemas y permiten el desarrollo de aplicaciones y herramientas digitales. Los lenguajes de programación facilitan la interacción entre humanos y máquinas, permitiendo la implementación de lógicas complejas y algoritmos. tructuras de control.

Características:

- Sintaxis y semántica: Reglas que estructuran y dotan de significado a las instrucciones.
- Abstracción: Facilitan el manejo de conceptos complejos mediante estructuras y funciones.
- Portabilidad: Muchos lenguajes pueden ejecutarse en diversas plataformas.
- Soporte de librerías y frameworks: Amplían las funcionalidades básicas del lenguaje [1].

9. TYPESCRIPT

TypeScript es un lenguaje de programación de código abierto desarrollado por Microsoft, que extiende JavaScript incorporando un sistema de tipos estático opcional [Microsoft, 2023]. Esto permite detectar errores de forma temprana y mejorar la claridad y mantenibilidad del código [Rosenwasser & Baylor, 2022].

Sirve para desarrollar aplicaciones complejas de forma más estructurada y robusta, permitiendo a los desarrolladores escribir código más predecible y fácil de mantener. Al compilar a JavaScript estándar, TypeScript es compatible con todos los navegadores y entornos de ejecución, lo que lo hace ideal para proyectos de gran escala [Microsoft, 2023; Rosenwasser & Baylor, 2022].

Características:

- Tipado estático opcional: Permite declarar tipos para reducir errores en tiempo de ejecución [Microsoft, 2023].
- Compatibilidad con JavaScript: Es un superconjunto de JavaScript, haciendo que el código JavaScript sea válido en TypeScript [Microsoft, 2023].
- Interfaces y tipos avanzados: Facilita la definición de estructuras de datos claras y reutilizables [Microsoft, 2023].
- Soporte para programación orientada a objetos: Permite utilizar clases, interfaces y herencia [Microsoft, 2023].
- Transpilación a JavaScript: Convierte el código TypeScript a JavaScript estándar para su ejecución [Microsoft, 2023].
- Mejora en la experiencia del desarrollador: Proporciona autocompletado y detección temprana de errores [Rosenwasser & Baylor, 2022].

10. HTML

HTML (HyperText Markup Language) es el lenguaje de marcado que se utiliza para estructurar y organizar el contenido en la web. Define la estructura básica de páginas y aplicaciones web a través de etiquetas que indican encabezados, párrafos, listas, enlaces, imágenes y otros elementos multimedia.

Sirve como la columna vertebral de cualquier sitio web, permitiendo definir la jerarquía y disposición del contenido. Es esencial para la creación de páginas web, ya que junto con CSS y JavaScript, permite construir sitios interactivos y visualmente atractivos [15].

Características:

- Estructuración de contenido: Define la organización de textos, imágenes y otros elementos mediante etiquetas [15].
- Compatibilidad universal: Es reconocido y soportado por todos los navegadores web [15].
- Integración con otros lenguajes: Funciona conjuntamente con CSS y JavaScript para generar sitios

dinámicos [15].

11. CSS

CSS (Cascading Style Sheets) es un lenguaje utilizado para describir la presentación y el diseño de documentos HTML [16]. Separa el contenido estructurado de su apariencia visual, permitiendo la aplicación de estilos de forma organizada y jerárquica [16].

Sirve para mejorar la apariencia de las páginas web, controlando aspectos como colores, fuentes, márgenes y disposición de los elementos. Esto facilita la creación de diseños coherentes y atractivos, y permite actualizar la estética de un sitio sin modificar su estructura subyacente [16].

Características:

- Separación de contenido y estilo: Permite modificar la apariencia sin alterar la estructura del contenido [16].
- Estilos en cascada: Aplica estilos de manera jerárquica, donde las reglas pueden heredarse o sobrescribirse según se requiera [16].
- Compatibilidad: Es el estándar de la presentación en la web, reconocido por todos los navegadores modernos [16].

12. ANDROID STUDIO

Android Studio es el entorno de desarrollo integrado (IDE) oficial de Google para la creación de aplicaciones en Android [13]. Está basado en IntelliJ IDEA y ofrece un conjunto completo de herramientas para el desarrollo, prueba y depuración de aplicaciones móviles [13].

Sirve para facilitar el proceso de desarrollo de aplicaciones nativas para dispositivos Android, proporcionando herramientas de diseño, simulación y análisis que optimizan el ciclo de creación y mejora de aplicaciones móviles [13].

Características:

- Basado en IntelliJ IDEA: Ofrece un entorno robusto y familiar para desarrolladores [13].
- Herramientas integradas: Facilita la edición, compilación y depuración de aplicaciones [13].

- Compatibilidad con emuladores: Permite probar aplicaciones en dispositivos virtuales y físicos [13].
- Optimización para Android: Incluye herramientas específicas para el diseño y desarrollo en la plataforma [13].

13. REACT

React es una biblioteca de JavaScript desarrollada por Facebook que se utiliza para construir interfaces de usuario interactivas y dinámicas [17]. Se basa en un enfoque declarativo y modular que divide la UI en componentes reutilizables [17].

Sirve para desarrollar aplicaciones web modernas y de alto rendimiento, facilitando la actualización eficiente de la interfaz de usuario mediante la manipulación del Virtual DOM y promoviendo una arquitectura basada en componentes que mejora la escalabilidad y mantenibilidad del código [17,1819]

Características:

- Basado en componentes: Permite dividir la interfaz en partes reutilizables y modulares [17].
- Virtual DOM: Optimiza las actualizaciones minimizando la manipulación directa del DOM real [18].
- Enfoque declarativo: Facilita la descripción de la UI en función de su estado, lo que simplifica la gestión y depuración [17].
- Unidireccionalidad en el flujo de datos: Contribuye a una gestión predecible del estado [19].
- Amplio ecosistema: Cuenta con numerosas herramientas, bibliotecas y una comunidad activa [19].

14. REACT NATIVE

React Native es un framework de código abierto basado en React que permite desarrollar aplicaciones móviles utilizando JavaScript [11]. Con este enfoque, se pueden crear aplicaciones que se comportan de forma nativa en iOS y Android a partir de una única base de código [11].

Sirve para facilitar el desarrollo de aplicaciones móviles, permitiendo a los desarrolladores aprovechar sus conocimientos en JavaScript y React para crear productos que ofrecen una experiencia

similar a la de las aplicaciones nativas, reduciendo tiempo y esfuerzo en el desarrollo multiplataforma [11].

Características:

- Uso de JavaScript: Permite aprovechar el lenguaje ampliamente conocido en el desarrollo web [11].
- Componentes reutilizables: Facilita la creación modular de interfaces, mejorando la mantenibilidad [11].
- Flujo de datos unidireccional: Contribuye a una gestión predecible del estado [11].
- Recarga en tiempo real: Permite ver de inmediato los cambios durante el desarrollo [11].

15. EXPO

Expo es un marco de desarrollo basado en React Native que simplifica la creación de aplicaciones móviles para iOS y Android [20]. Proporciona un conjunto de herramientas, servicios y APIs preconfiguradas que eliminan la necesidad de lidiar con la configuración nativa compleja [20].

Sirve para agilizar el desarrollo, prueba y distribución de aplicaciones móviles. Permite a los desarrolladores centrarse en escribir código en JavaScript sin preocuparse por detalles nativos, facilitando el acceso a funcionalidades del dispositivo y actualizaciones Over-The-Air (OTA) [20][22].

Características:

- Desarrollo sin configuración nativa: Permite crear aplicaciones sin instalar herramientas como Xcode o Android Studio [20].
- Acceso a API del dispositivo: Facilita la utilización de funciones como la cámara, la ubicación y notificaciones [20].
- Actualizaciones OTA: Permite actualizar aplicaciones directamente sin pasar por las tiendas oficiales [22].
- Compatibilidad con React Native: Utiliza la misma base de código para desarrollar aplicaciones de forma eficiente [20].

16. EXPO GO

Expo Go es una aplicación complementaria al marco Expo que permite ejecutar y probar aplicaciones móviles desarrolladas con Expo de manera inmediata en dispositivos físicos [12].

Sirve para facilitar la iteración y validación del desarrollo de aplicaciones móviles, ya que permite a los desarrolladores ver los cambios realizados en el código en tiempo real sin necesidad de compilar una versión nativa completa [12].

Características:

- Integración con React Native: Simplifica el proceso de prueba y desarrollo [12].
- Optimización del flujo de trabajo: Permite visualizar actualizaciones en tiempo real, acelerando el ciclo de desarrollo [12].
- Compatibilidad multiplataforma: Facilita el desarrollo para iOS y Android desde una única base de código [12].

17. ANGULAR

Angular es una plataforma y framework de código abierto para el desarrollo de aplicaciones web de una sola página (SPA) [14]. Utiliza HTML y TypeScript para construir aplicaciones dinámicas, estructuradas y modulares, aprovechando la arquitectura basada en componentes [14].

Sirve para desarrollar aplicaciones web robustas y escalables, permitiendo a los desarrolladores organizar el código de manera modular, reutilizar componentes y gestionar la complejidad mediante la inyección de dependencias. Angular está diseñado para ofrecer experiencias de usuario fluidas en aplicaciones de alta interactividad [14].

Características:

- Uso de TypeScript: Mejora la escalabilidad y mantenibilidad del código [14].
- Arquitectura basada en componentes: Permite el desarrollo modular y la reutilización de código [14].
- Inyección de dependencias: Facilita la gestión e integración de servicios y funcionalidades [14].
- Optimización para aplicaciones SPA: Diseñado para ofrecer experiencias de usuario fluidas y eficientes [14].

Marco Contextual

El estudio se desarrolla en la comunidad de **Xicotepec de Juárez**, una localidad caracterizada por un clima predominantemente húmedo y con frecuentes episodios de lluvia, condiciones que dificultan el secado tradicional de la ropa. Este entorno, donde tanto zonas urbanas como rurales conviven, presenta desafíos diarios relacionados con la dependencia del clima para tareas cotidianas como el secado de prendas. Estas condiciones afectan especialmente a hogares que utilizan métodos tradicionales de secado, los cuales dependen exclusivamente de la variabilidad del clima, obligando a los usuarios a realizar constantes intervenciones manuales para monitorear y ajustar el proceso de secado.

El proyecto de Tendedero Inteligente busca ofrecer una solución integral a estos problemas, adaptándose a las condiciones específicas de Xicotepec de Juárez. La integración de sensores de humedad y temperatura, junto con el uso de tecnologías de bajo consumo energético, permitirá automatizar y optimizar el proceso de secado, reduciendo la intervención manual y protegiendo la calidad de las prendas. Esta innovación no solo mejora la calidad de vida de los habitantes, sino que también promueve el uso racional de recursos y la sostenibilidad ambiental en la región.

Capítulo 3 Metodología y Desarrollo

1. Metodología Scrum

Para el desarrollo del Tendedero Inteligente se optó por implementar la metodología Scrum, debido a su capacidad para gestionar proyectos complejos de forma ágil y adaptable. Este enfoque permite estructurar el proyecto en ciclos cortos de 1 a 2 semanas, en los cuales se revisa constantemente el avance, se identifican áreas de mejora y se ajustan los procesos según los resultados obtenidos.

La elección de Scrum se fundamenta en la necesidad de integrar de manera coordinada la parte de hardware (sensores, actuadores y módulos de conectividad) y la de software (programación, interfaces y gestión de datos). Con este método, el equipo puede abordar cada etapa del desarrollo sin sentirse abrumado, ya que se centra en objetivos específicos en cada sprint, permitiendo corregir desviaciones y optimizar la funcionalidad del sistema en tiempo real.

En cada sprint se llevan a cabo las siguientes fases:

1. **Recolección de Requerimientos y Análisis:** Se identifican las necesidades del proyecto, se definen las funcionalidades esenciales del tendedero inteligente y se establecen los criterios de éxito.
2. **Diseño:** Se elabora un prototipo inicial, se planifican los componentes tanto de hardware como de software y se establecen las interfaces de usuario y control.
3. **Desarrollo y Programación:** Se realiza la implementación de los módulos de sensores, el desarrollo de la plataforma web y móvil, y la integración de sistemas de comunicación entre ellos.
4. **Pruebas y Validación:** Se ejecutan pruebas de funcionamiento, se recopilan datos de rendimiento y se verifican los ajustes de los sensores y algoritmos de control.
5. **Implementación y Mantenimiento:** Se lanza el sistema en un entorno controlado y se establecen protocolos para el mantenimiento continuo y la incorporación de mejoras basadas en el feedback de los usuarios.

2. Análisis

Identificación del Problema Real

Para definir el problema a resolver, realizamos una lluvia de ideas en equipo, evaluando situaciones cotidianas en las que la tecnología podría facilitar tareas domésticas. Uno de los aspectos más discutidos fue el secado de ropa, ya que es un proceso que depende de múltiples factores externos y, en muchas ocasiones, no es eficiente.

A partir de esta reflexión, identificamos varios inconvenientes con los métodos tradicionales de secado:

- Factores climáticos impredecibles: No hay manera de prever si lloverá o si las condiciones ambientales serán óptimas para secar la ropa.
- Daño en las prendas por exposición excesiva: Dejar la ropa demasiado tiempo al sol deteriora los tejidos y los colores, reduciendo la vida útil de las prendas.
- Recolección tardía por descuido: Muchas veces, la ropa permanece en el tendedero más tiempo del necesario porque el usuario olvida retirarla, lo que la deja expuesta al polvo o cambios climáticos repentinos.
- Problemas en espacios reducidos: En hogares pequeños, el área de secado es limitada, y no siempre se puede optimizar el espacio disponible.
- Desperdicio de energía en sistemas de secado convencionales: El uso de secadoras eléctricas representa un alto consumo energético, incrementando los costos de electricidad.



Ilustración 1. Tendedero tradicional de ropa

Propuesta de Solución: Tendedero Inteligente

Tras identificar estas problemáticas, definimos la idea de un tendedero automatizado que incorpore sensores y un sistema inteligente para mejorar la eficiencia del secado. Para validar su viabilidad, realizamos encuestas y recopilamos opiniones de potenciales usuarios, lo que nos permitió definir las funcionalidades clave del sistema:

- Monitoreo de humedad en la ropa y condiciones climáticas: Un sensor de agua detectará si la ropa aún está húmeda, mientras que sensores ambientales medirán temperatura y humedad relativa para predecir el tiempo de secado.
- Automatización del proceso: Un mecanismo motorizado permitirá extender o retraer el tendedero en función del clima y el estado de la ropa.
- Alertas inteligentes: El sistema enviará notificaciones a una aplicación móvil cuando la ropa esté seca o si se detectan condiciones adversas como lluvia.

Selección de Tecnología

Con base en estos requerimientos, se evaluaron diferentes tecnologías y se eligieron los siguientes componentes:

- Sensor de agua: Se utilizará un sensor de humedad para detectar lluvia y retirar automáticamente la ropa.

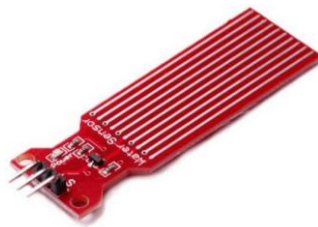


Ilustración 2. Sensor de lluvia YL- 83

- Sensores ambientales: Se empleará un DHT11 para medir temperatura y humedad, permitiendo estimar el tiempo óptimo de secado.



Ilustración 3 Sensor de temperatura y humedad DHT11

- Mecanismo motorizado: Se optó por un motor paso a paso, ya que permite un control preciso para desplegar o retraer el tendedero de manera automática.



Ilustración 4. Motor a pasos 28BYJ-48 (5V)

- Microcontrolador: Se seleccionó la ESP32, debido a su conectividad WiFi y Bluetooth, facilitando la comunicación con la aplicación móvil.

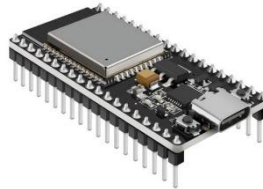


Ilustración 5. Microcontrolador ESP32

- Aplicación y almacenamiento de datos: Se usará Node.js y MongoDB para el backend y gestión de datos, junto con React Native para el desarrollo de la app web, Expo y Expo go para la aplicación móvil permitiendo el control remoto del tendedero.



Ilustración 6. MongoDB

3. Diseño

Diseño del Hardware

El hardware del Tendedero Inteligente se compone de los siguientes elementos principales:

- **ESP32:** Este microcontrolador es el corazón del sistema. Gracias a su conectividad WiFi y Bluetooth, procesa los datos obtenidos de los sensores y controla los actuadores.
- **Sensor DHT11:** Mide la temperatura y la humedad relativa del ambiente, proporcionando información vital para determinar las condiciones de secado.
- **Sensor de lluvia:** Detecta la presencia de agua sobre su superficie y envía una señal al ESP32 para activar el mecanismo de protección en caso de precipitaciones.
- **Motores paso a paso 28BYJ-48:** Se utilizan para mover el tendido de forma precisa, extendiéndolo o retrayéndolo según sea necesario.
- **Driver ULN2003:** Actúa como interfaz entre los motores y el ESP32, asegurando que se suministre la corriente y el voltaje adecuados para el funcionamiento correcto de los motores.

Diseño de Software

El software del sistema se divide en tres grandes bloques:

1. **Firmware en el ESP32:**

Programado en C++ mediante Arduino IDE, se implementaron librerías como AccelStepper para el control de los motores y DHT para la lectura de datos ambientales. Se configuró la conectividad WiFi y se implementó el protocolo WebSocket, que permite la comunicación en tiempo real entre el ESP32 y las aplicaciones (web y móvil). El firmware organiza las tareas en un bucle principal que se encarga de la adquisición de datos, la activación de los actuadores y la transmisión de información a través del canal WebSocket.

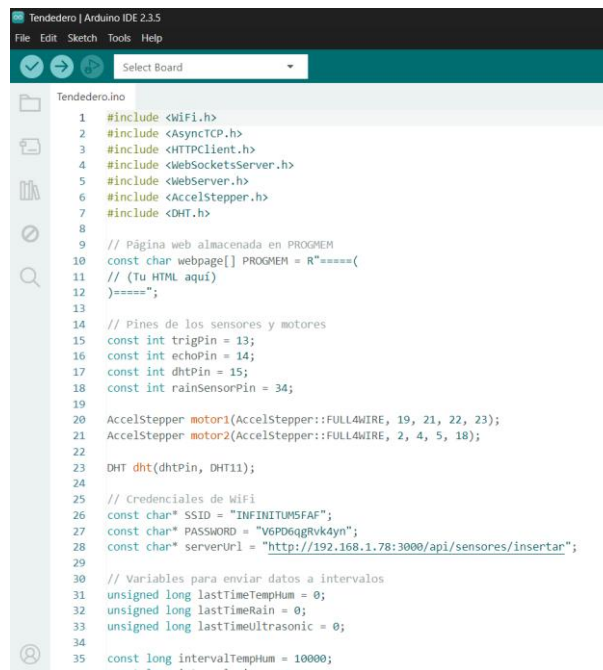


Ilustración 7, código Arduino de nuestro proyecto

2. Aplicación Web

Diseñada para mostrar datos procesados, la aplicación web (desarrollada con Angular, HTML, CSS y JavaScript) presenta información en tablas y gráficos interactivos, permitiendo el análisis detallado de las condiciones ambientales a lo largo del tiempo. Se utiliza Node.js en el backend, junto con librerías como amCharts, para la visualización dinámica de datos.

Se diseñó un panel principal que muestra datos en tiempo real mediante tablas y gráficos, junto con opciones para visualizar el historial de mediciones y analizar tendencias.

3. Aplicación móvil

Desarrollada con React Native y Expo, la aplicación móvil se centra en el control directo del tendedero. Permite a los usuarios enviar comandos para extender o retraer el tendedero en tiempo real, facilitando la operación manual o el cambio de modo de funcionamiento, y ofreciendo una interfaz adaptada para dispositivos táctiles.

El diseño de la interfaz se realizó mediante la elaboración de bocetos y prototipos en Figma y Adobe XD, se creó una interfaz enfocada en el control remoto del tendedero, con botones de acción y notificaciones en tiempo real, permitiendo una operación intuitiva y eficiente.

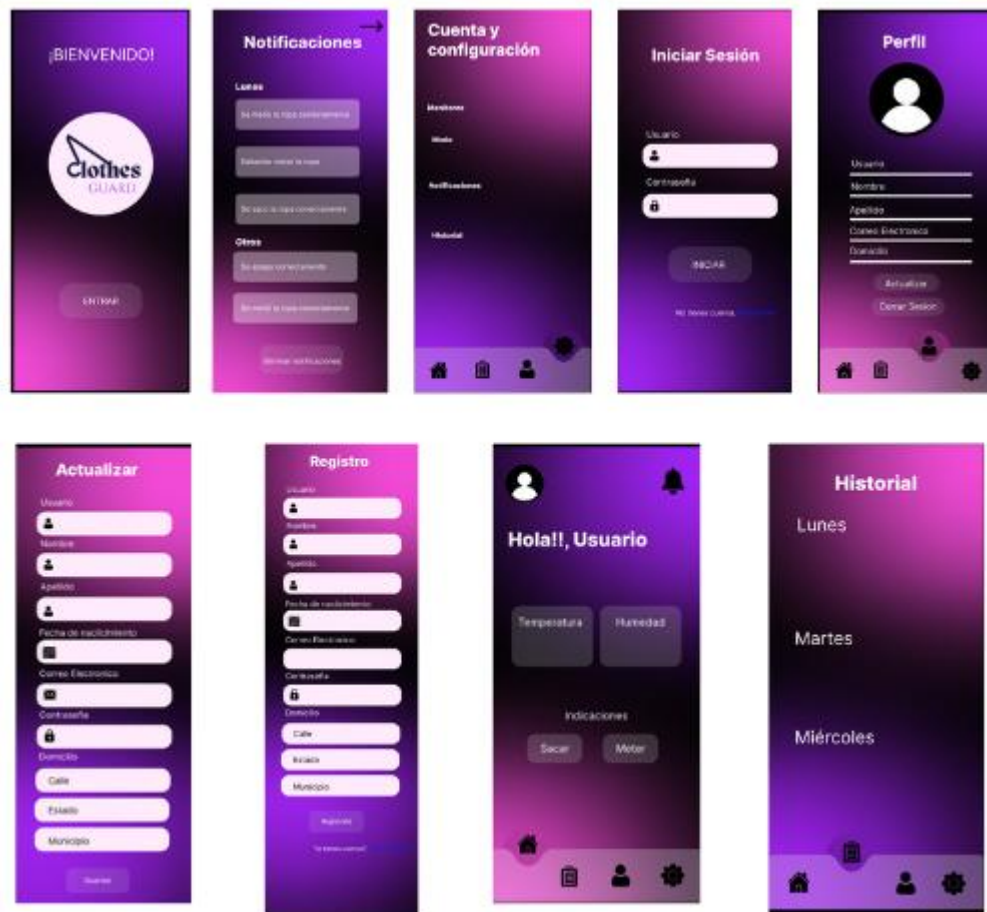


Ilustración 8 Wireframes de la api mobile en figma

4. Desarrollo/Programación.

Funcionamiento del Código en el ESP32

El firmware del ESP32 se desarrolló de manera modular y cronológica para integrar todos los componentes del sistema. Su funcionamiento se describe en las siguientes etapas:

1. Inicialización de Componentes:

Se configuran los pines del ESP32 para conectar los sensores y actuadores, estableciendo los parámetros de calibración necesarios para obtener lecturas precisas.

```

Tendedero | Arduino IDE 2.3.5
File Edit Sketch Tools Help

Select Board

Tendedero.ino
1  #include <WiFi.h>
2  #include <AsyncTCP.h>
3  #include <HTTPClient.h>
4  #include <WebSocketsServer.h>
5  #include <WebServer.h>
6  #include <AccelStepper.h>
7  #include <DHT.h>
8
9  // Página web almacenada en PROGMEM
10 const char webpage[] PROGMEM = R"=====(
11 // (Tu HTML aquí)
12 )=====";
13
14 // Pines de los sensores y motores
15 const int trigPin = 13;
16 const int echoPin = 14;
17 const int dhtPin = 15;
18 const int rainSensorPin = 34;
19
20 AccelStepper motor1(AccelStepper::FULL4WIRE, 19, 21, 22, 23);
21 AccelStepper motor2(AccelStepper::FULL4WIRE, 2, 4, 5, 18);
22
23 DHT dht(dhtPin, DHT11);

```

Ilustración 9 Configuración de pines en el código Arduino

2. Conexión a la Red WiFi:

El ESP32 se conecta a la red doméstica utilizando credenciales preestablecidas, implementando un mecanismo de reconexión automática para asegurar la continuidad de la comunicación.

```

25 // Credenciales de WiFi
26 const char* SSID = "INFINITUM5FAF";
27 const char* PASSWORD = "V6PD6qgRvk4yn";
28 const char* serverUrl = "http://192.168.1.78:3000/api/sensores/insertar";
29

```

Ilustración 10 Conexión con una red de Internet

```

96 // Función para conectar a WiFi
97 void connectWiFi() {
98     Serial.print("Conectando a WiFi...");
99     WiFi.begin(SSID, PASSWORD);
100     unsigned long startTime = millis();
101     while (WiFi.status() != WL_CONNECTED && millis() - startTime < 20000) {
102         delay(1000);
103         Serial.print(".");
104     }
105
106     if (WiFi.status() == WL_CONNECTED) {
107         Serial.println("\n✅ Conexión WiFi exitosa");
108         Serial.print("📡 Dirección IP: ");
109         Serial.println(WiFi.localIP());
110     } else {
111         Serial.println("\n❌ No se pudo conectar al WiFi. Reiniciando...");
112         ESP.restart();
113     }
114 }
115

```

Ilustración 11 Mensaje de conexión a red Wifi

```

51 void setup() {
52     Serial.begin(115200);
53     pinMode(trigPin, OUTPUT);
54     pinMode(echoPin, INPUT);
55     pinMode(rainSensorPin, INPUT);
56
57     motor1.setMaxSpeed(1500);
58     motor1.setAcceleration(500);
59     motor2.setMaxSpeed(1500);
60     motor2.setAcceleration(500);
61
62     dht.begin();
63     connectWiFi();
64 }

```

Ilustración 12. Función siendo llamada de forma constante dentro de setup()

3. Adquisición y Procesamiento de Datos:

El dispositivo realiza lecturas periódicas de la temperatura, la humedad y la detección de lluvia. Los datos se procesan, se filtran y se empaquetan en formato JSON para su envío a través de WebSockets.

4. Activación de Motores Según Condiciones Ambientales:

Basándose en los datos procesados, el sistema determina si es necesario extender o retraer el tendido. Por ejemplo, si se detecta lluvia, el firmware activa el mecanismo de retracción; en condiciones óptimas, extiende el tendido para facilitar el secado.

5. Comunicación en Tiempo Real:

Los datos y el estado del sistema se transmiten de forma continua a las aplicaciones (web y móvil) mediante WebSockets. La interfaz recibe esta información en tiempo real y, a su vez, puede enviar comandos que son ejecutados inmediatamente por el ESP32.

Desarrollo del Frontend y de la Aplicación Móvil

Aplicación Web (Angular)

- Creación y Configuración del Proyecto:

Se ejecutó el comando `ng new Angular` para generar la estructura base del proyecto.

- Instalación de Dependencias:

Se instalaron librerías y frameworks esenciales como Express, CORS, y amCharts para la visualización de datos.

- Generación de Componentes:

Se utilizaron comandos como `ng g c <nombre_del_componente>` para crear módulos, páginas

informativas y componentes dinámicos.

- Implementación del Diseño:

Se integraron plantillas y paletas de colores obtenidas de Tailwind Components, asegurando un diseño tecnológico, creativo y atractivo para el usuario.

Aplicación Móvil (React Native)

- Fase de Diseño y Planificación:

Se realizaron bocetos y prototipos iniciales con Canva, Adobe XD y Figma, definiendo la paleta de colores y estructura de la aplicación.

- Desarrollo del Frontend:

Se creó la pantalla principal, de login, registro y vistas adicionales (menú, notificaciones, historial y perfil) usando React Native y Expo, garantizando una experiencia de usuario fluida.

- Implementación de Navegación e Interacciones:

Se integró el sistema de navegación (@react-navigation) y se empleó axios para la comunicación con la API, permitiendo un control en tiempo real del tendido.

Desarrollo del Backend y la API

El backend fue diseñado para establecer una comunicación fluida entre el frontend y la base de datos mediante una API RESTful. La arquitectura del backend se organizó en varias carpetas con funciones específicas:

- Controller:

Gestiona las solicitudes de los usuarios y coordina la lógica de negocio.

- DAO (Data Access Object):

Facilita la interacción con la base de datos, permitiendo operaciones de lectura y escritura de datos.

- Models:

Define la estructura y esquemas de datos, utilizando la librería Mongoose para MongoDB.

- Router:

Se encarga de gestionar las rutas y solicitudes HTTP que llegan desde el frontend.

Archivos clave como app.js, index.js y databases.js se utilizan para inicializar el servidor, gestionar la conexión a la base de datos y coordinar la ejecución de los diferentes módulos. Esta arquitectura modular y escalable permite una fácil implementación de nuevas funcionalidades y el mantenimiento continuo del sistema.

Gestión de Base de Datos y MongoDB Atlas

Para la gestión de datos se optó por MongoDB Atlas, una solución en la nube que facilita la administración, escalabilidad y seguridad del sistema. En este entorno se implementaron las siguientes estrategias:

- **Cluster Dedicado:**

Se creó un cluster exclusivo para el proyecto ClothesGuard, configurado para mantener la integridad y rendimiento de los datos.

- **Colecciones Esenciales:**

- Usuarios: Almacena credenciales, datos personales y configuraciones de los usuarios.
- Sensores y Actuadores: Registra los datos generados por los dispositivos IoT, permitiendo el monitoreo en tiempo real.
- Notificaciones: Guarda los mensajes y alertas enviados a los usuarios.

Desarrollo de la Maqueta (Prototipo)

Para validar la viabilidad técnica y funcional del Tendedero Inteligente, se realizó la construcción de una maqueta que integró de forma física todos los componentes del sistema. Este prototipo permitió evaluar de manera práctica la interacción entre los sensores, el microcontrolador, los actuadores y la interfaz de usuario.

- **Ensamblaje del Hardware:**

La maqueta se constituyó a partir de la integración del ESP32, que actúa como unidad de procesamiento central, con el sensor de humedad y temperatura DHT11, el sensor de lluvia y los motores paso a paso modelo 28BYJ-48. Los módulos se conectaron utilizando un driver ULN2003, el cual aseguraba la correcta entrega de corriente y voltaje a los motores. Durante

el ensamblaje se verificó la disposición de los componentes en una placa de pruebas, optimizando la distribución para facilitar tanto la conexión de cables como la posterior integración con la carcasa del prototipo.



Ilustración 13. Motor a pasos 1 ensamblado en un poste

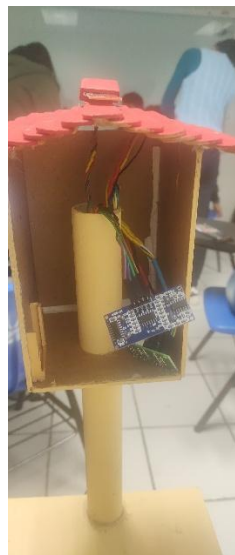


Ilustración 14. Poste Principal con Sensor de agua, Distancia, Humedad y Motor a pasos

- Verificación de la Conectividad y Funcionamiento:

Se realizaron pruebas iniciales para confirmar la conectividad del ESP32 a la red WiFi, asegurando la transmisión continua de datos a las aplicaciones web y móviles mediante el protocolo WebSocket. Adicionalmente, se calibraron los sensores para que los valores de humedad, temperatura y detección de lluvia fueran precisos y consistentes, lo que permitió

validar la lógica de activación del mecanismo motorizado para extender o retraer el tendedero.

- **Integración y Ajustes Físicos:**

Una vez integrados los componentes, se procedió a montar la maqueta en una estructura provisional que simulaba la instalación real del sistema. Este montaje incluyó una carcasa protectora para el ESP32 y los sensores, diseñada para minimizar la interferencia de factores externos (como la luz solar directa o la acumulación de polvo) y que facilitara el acceso a los componentes para tareas de mantenimiento y verificación. Durante esta etapa se realizaron ajustes en la ubicación y fijación de cada módulo, garantizando un ensamblaje sólido y funcional.



Ilustración 15. Vista aérea de la maqueta ensamblada

6. Pruebas/Testing.

Pruebas de Hardware

- **Sensores:** Se verificó la precisión de los datos del DHT11 y del sensor de lluvia.
- **Motores:** Se realizaron pruebas de carga para asegurar su operación precisa sin sobrecalentamiento.

Pruebas de Software

- **Conectividad WiFi:** Se evaluó la estabilidad de la conexión y el mecanismo de reconexión.
- **Comunicación en Tiempo Real:** Se validó el correcto funcionamiento de los WebSockets en el envío y recepción de datos.
- **Interfaz de Usuario:** Se probó la usabilidad en ambas aplicaciones (web y móvil).

Pruebas de Integración Backend-Frontend

Se utilizaron herramientas como Postman para probar la API, garantizando que la comunicación entre el frontend y el backend fuese confiable y eficiente.

7. Implementación/Mantenimiento.

Instalación del Sistema

- El prototipo se instaló en un entorno real para evaluar su funcionamiento operativo, verificando:
- La correcta conexión del ESP32 a la red WiFi.
- La estabilidad en la transmisión de datos hacia las aplicaciones.
- La respuesta del sistema ante condiciones ambientales variables.

Capacitación y Soporte

Se desarrollaron manuales, tutoriales y se impartieron sesiones de capacitación para asegurar el correcto uso y mantenimiento del sistema por parte de los usuarios.

Plan de Mantenimiento

Se establecieron protocolos para:

- La limpieza periódica de sensores.
- La inspección y alineación de motores.
- La actualización remota del software.
- El monitoreo continuo del desempeño del sistema para detectar y corregir anomalías de manera proactiva.

Capítulo 4 Resultados y Conclusiones

Resultado

Tras aplicar las correcciones pertinentes, los resultados obtenidos demostraron un desempeño satisfactorio y un avance significativo hacia el cumplimiento de los objetivos del proyecto:

- **Conectividad y Comunicación en Tiempo Real:** Se logró establecer una comunicación fluida entre el ESP32 y las aplicaciones web y móviles mediante el uso de WebSockets, permitiendo la visualización en tiempo real de los datos procedentes de los sensores.
- **Integración y Funcionalidad de Sensores:** Los sensores de humedad, temperatura y lluvia operaron de manera eficiente tras la calibración final, posibilitando la detección oportuna de las condiciones ambientales y la activación automática del sistema.
- **Mecanismo de Control Automatizado:** El sistema de extensión y retracción del tendedero funcionó de forma correcta, respondiendo a los cambios ambientales y protegiendo la ropa al retirar el tendido ante la presencia de precipitaciones.
- **Interfaz de Usuario Amigable:** Las aplicaciones desarrolladas ofrecieron una experiencia de usuario fluida, donde se pueden consultar datos históricos y recibir notificaciones, lo que facilita el monitoreo y control remoto del sistema.

Conclusiones

Durante el desarrollo del proyecto ClothesGuard se logró integrar satisfactoriamente un sistema automatizado que optimiza el proceso de secado en condiciones adversas. A pesar de los retos iniciales, la implementación de las correcciones adecuadas permitió establecer una conexión estable entre hardware y software, garantizando la visualización en tiempo real de los datos y la respuesta oportuna del sistema ante cambios ambientales. El prototipo, al integrar sensores, comunicación en tiempo real y control automatizado, demuestra ser una solución funcional y sostenible, que no solo cumple con los objetivos planteados, sino que también mejora la calidad de vida de los usuarios y contribuye al uso eficiente de la energía.

Recomendaciones

- Se recomienda realizar un mantenimiento periódico del sistema, prestando especial atención a la limpieza y recalibración de los sensores para asegurar su óptimo rendimiento a lo largo del tiempo.

- Es aconsejable revisar y actualizar el firmware del ESP32 de manera regular, incorporando mejoras basadas en la retroalimentación recibida durante las pruebas y en los avances tecnológicos.
- Dado que el prototipo integra diversas tecnologías (sensores, plataformas web y móviles), se sugiere documentar detalladamente cada proceso de integración y desarrollo, lo cual facilitará futuras ampliaciones o adaptaciones del sistema.
- Se recomienda capacitar a los usuarios mediante manuales o tutoriales que expliquen el funcionamiento, los procedimientos de mantenimiento y el manejo correcto del sistema, asegurando así un uso eficiente y prolongado del dispositivo.
- Considerando la variabilidad de las condiciones climáticas, es importante realizar estudios de campo periódicos para ajustar los parámetros del sistema, garantizando la optimización del secado en distintas zonas geográficas.
- Finalmente, se sugiere evaluar la posibilidad de integrar fuentes de energía alternativas, como paneles solares, con el fin de incrementar la autonomía del sistema y reforzar el compromiso del proyecto con la sostenibilidad ambiental.

Lista de referencias

- [1] Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2014). Metodología de la investigación (6ª ed.). McGraw-Hill.
- [2] Creswell, J. W., & Creswell, J. D. (2018). Research design: Qualitative, quantitative, and mixed methods approaches (5ª ed.). SAGE Publications.
- [3] Yin, R. K. (2018). Case study research and applications: Design and methods (6ª ed.). SAGE Publications.
- [4] Babbie, E. (2020). The practice of social research (15ª ed.). Cengage.
- [5] Robson, C., & McCartan, K. (2016). Real world research (4ª ed.). Wiley.
- [6] Pressman, R. S. (2010). Ingeniería del Software: Un enfoque práctico (7ª ed.). McGraw-Hill.
- [7] Sommerville, I. (2011). Ingeniería del Software (9ª ed.). Addison-Wesley.
- [8] Larman, C. (2004). Agile and Iterative Development: A Manager's Guide. Addison-Wesley.
- [9] Figma Inc. (n.d.). Figma. Recuperado el 26 de marzo de 2025, de <https://www.figma.com>
- [10] Banzi, M., & Shiloh, M. (2011). Getting started with Arduino: The open source electronics prototyping platform. Maker Media, Inc.
- [11] Novick, J. (2017). [Título del artículo o fuente sobre React Native]. [Detalles de publicación, si están disponibles].
- [12] Expo. (n.d.). Expo Go. Recuperado de <https://expo.dev>
- [13] Li, H., Chen, X., & Zheng, Q. (2017). Android application development framework based on Android Studio. En 2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA) (pp. 278–282). IEEE.
- [14] Angular Team. (n.d.). Angular. Recuperado de <https://angular.io>
- [15] W3C. (n.d.). HTML. Recuperado de <https://www.w3.org/standards/webdesign/htmlcss>
- [16] W3C. (n.d.). CSS. Recuperado de <https://www.w3.org/Style/CSS>
- [17] Facebook. (2013). React: A JavaScript library for building user interfaces. Facebook Open Source. Recuperado de <https://react.dev/>
- [18] Freeman, A. (2021). Pro React 17. Apress.
- [19] Banks, A., & Porcello, E. (2017). Learning React: Modern patterns for developing React apps. O'Reilly Media.
- [20] Expo. (n.d.). Expo documentation. Recuperado de <https://docs.expo.dev>

[21] Wojciechowski, A. (2021). Mastering React Native: Build well-structured and maintainable applications with React Native. Packt Publishing.

[22] Expo. (2023). Expo Application Services (EAS). Recuperado de <https://expo.dev/eas>

[Microsoft] Microsoft. (2023). TypeScript Documentation. Microsoft Docs. Recuperado de <https://www.typescriptlang.org>

[MongoDB] MongoDB. (2023). What is MongoDB? Recuperado de <https://www.mongodb.com>

[Schwaber & Sutherland] Schwaber, K., & Sutherland, J. (2020). The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. Scrum.org.

]

Apéndice

Las tablas y figuras pueden ir en el apéndice como se mencionó anteriormente. También es posible usar el apéndice para incluir datos en bruto, instrumentos de investigación y material adicional.