

ENTREGA PARCIAL 1

TRABAJO FINAL DE BACHELOR

EnergiApp v1.0: Plataforma Web para la Visualización y Predicción del Consumo Energético Doméstico mediante Simulación de IoT y Análisis de Datos

Estudiante: Oliver Vincent Rice

Tutor: Isabel Sánchez

Universidad: Universitat Carlemany

Grado: Bachelor en Informàtica

Fecha: 16 de julio de 2025

Universitat Carlemany
Bachelor en Informàtica
Julio 2025

Índice

1. Información del Proyecto	3
1.1. Datos del Estudiante y Supervisor	3
2. Título del Proyecto	3
2.1. Título Principal	3
2.2. Título Alternativo	3
3. Objetivos	3
3.1. Objetivo General	3
3.2. Objetivos Específicos	4
3.2.1. Objetivos de Desarrollo Técnico	4
3.2.2. Objetivos de Funcionalidad y UX	4
3.2.3. Objetivos de Impacto y Innovación	4
4. Justificación de la Propuesta	5
4.1. Relevancia Social y Ambiental	5
4.1.1. Problemática Actual	5
4.1.2. Oportunidad de Impacto	6
4.2. Innovación Tecnológica	6
4.2.1. Convergencia de Tendencias Actuales	6
4.2.2. Diferenciación Competitiva	7
4.3. Viabilidad Técnica y Económica	7
4.3.1. Madurez Tecnológica	7
4.3.2. Impacto Económico Proyectado	8
5. Propuesta de Índice	8
5.1. Parte I: Fundamentos y Contexto	8
5.2. Parte II: Análisis y Diseño	9
5.3. Parte III: Implementación y Desarrollo	9
5.4. Parte IV: Validación y Resultados	10
5.5. Parte V: Conclusiones y Proyección	11
5.6. Anexos	11
6. Alcance del Proyecto	11
6.1. Alcance Incluido (Funcionalidades Principales)	11
6.1.1. Aplicación Frontend React	11
6.1.2. Backend Node.js + Express	12
6.2. Limitaciones y Exclusiones	13

6.2.1.	Limitaciones Técnicas	13
6.2.2.	Funcionalidades Fuera de Alcance	13
7.	Planificación y Cronograma	14
7.1.	Metodología de Desarrollo	14
7.1.1.	Enfoque Metodológico	14
7.1.2.	Herramientas de Gestión y Desarrollo	14
7.2.	Cronograma de Entregas Oficiales	14
7.3.	Estado Actual del Proyecto	15
7.3.1.	Enfoque de Desarrollo	15
7.3.2.	Preparación para las Siguietes Entregas	15
8.	Recursos y Herramientas	15
8.1.	Recursos Tecnológicos	15
8.1.1.	Hardware de Desarrollo	15
8.1.2.	Stack Tecnológico Completo	16
8.2.	Recursos Humanos	16
8.2.1.	Equipo del Proyecto	16
8.2.2.	Competencias Técnicas Aplicadas	17

1 Información del Proyecto

1.1 Datos del Estudiante y Supervisor

Campo	Información
Estudiante	Oliver Vincent Rice
Tutor	Isabel Sánchez
Universidad	Universitat Carlemany
Grado	Bachelor en Informàtica
Asignatura	Trabajo Final de Grado
Fecha de Entrega	16 de julio de 2025

Tabla 1: Información del proyecto

2 Título del Proyecto

2.1 Título Principal

“EnergiApp v1.0: Plataforma Web para la Visualización y Predicción del Consumo Energético Doméstico mediante Simulación de IoT y Análisis de Datos”

2.2 Título Alternativo

“Sistema Full-Stack de Optimización del Consumo Energético mediante Algoritmos Predictivos y Recomendaciones Inteligentes”

3 Objetivos

3.1 Objetivo General

Desarrollar una plataforma web (EnergiApp v1.0) que permita a los usuarios visualizar y predecir su consumo energético doméstico a través del análisis de datos simulados representativos de dispositivos IoT, utilizando algoritmos de Machine Learning, recomendaciones automatizadas y un dashboard interactivo con visualizaciones en tiempo real, con el fin de fomentar la sostenibilidad energética en el hogar y apoyar la toma de decisiones responsables.

3.2 Objetivos Específicos

3.2.1. Objetivos de Desarrollo Técnico

1. **Analizar datasets de consumo energético** representativos de dispositivos IoT domésticos para entrenar y validar algoritmos durante el desarrollo, preparando el sistema para procesar datos reales de dispositivos IoT en producción.
2. **Diseñar e implementar una API RESTful** con Node.js y Express.js para gestionar y almacenar los datos de consumo energético y usuarios.
3. **Desarrollar un modelo predictivo** de consumo energético a corto plazo (1-7 días) usando técnicas de Machine Learning y análisis de patrones.
4. **Crear una interfaz web moderna** con React 18 que muestre el consumo histórico, predicciones en tiempo real y recomendaciones personalizadas.
5. **Implementar sistema de recomendaciones** automatizadas para optimización energética (control de standby, programación temporal, etc.).

3.2.2. Objetivos de Funcionalidad y UX

6. **Alinear la solución con los Objetivos de Desarrollo Sostenible (ODS)** 7, 11, 12 y 13, promoviendo el uso responsable de la energía.
7. **Implementar dashboard multi-usuario** con diferentes niveles de acceso y visualizaciones interactivas con Chart.js.
8. **Desarrollar sistema de simulación IoT** que genere datos realistas durante la fase de desarrollo basados en datasets como UK-DALE y UCI Machine Learning Repository, replicando el comportamiento que tendría con dispositivos reales.
9. **Crear panel administrativo completo** para gestión de usuarios, dispositivos simulados y generación de reportes energéticos.
10. **Optimizar experiencia responsive** con diseño móvil-first para smartphone, tablet y desktop.

3.2.3. Objetivos de Impacto y Innovación

11. **Demostrar viabilidad práctica** del sistema mediante funcionalidades ejecutables que controlen dispositivos reales del usuario.
12. **Lograr interfaz profesional** con estándares de calidad empresarial, animaciones fluidas y UX excepcional.

13. **Establecer arquitectura escalable** preparada para expansión a sistemas de producción y integración con dispositivos IoT.
14. **Generar documentación exhaustiva** con manuales de usuario, guías técnicas y documentación académica completa.
15. **Validar algoritmos ML** para garantizar predicciones consistentes y útiles basadas en patrones reales, no aleatorios.

4 Justificación de la Propuesta

4.1 Relevancia Social y Ambiental

Durante mis años de estudio en informática, he tenido varias experiencias que me han llevado a desarrollar este proyecto. En mi casa, por ejemplo, nunca sabíamos realmente qué electrodomésticos estaban gastando más luz hasta que empezamos a prestar atención después de una factura especialmente alta. Mi madre siempre se quejaba de que el frigorífico viejo consumía mucho, pero no teníamos forma de comprobarlo.

Esta situación me hizo reflexionar sobre un problema más amplio. Investigando un poco, descubrí que las familias españolas pagamos de media unos 950 euros al año por la electricidad. Lo más sorprendente fue enterarme de que aproximadamente un 15 % o 20 % de ese dinero se va en dispositivos que están en standby o que simplemente no usamos de forma eficiente.

Fue entonces cuando me topé con datasets como UK-DALE. Me di cuenta de que estos conjuntos de datos serían perfectos para desarrollar y probar los algoritmos de Machine Learning durante la fase de desarrollo, ya que yo mismo no tengo acceso a dispositivos IoT reales para generar datos de prueba.

4.1.1. Problemática Actual

Después de hablar con amigos y familiares, me di cuenta de que el problema que vivíamos en casa no era único. Muchas personas simplemente no saben qué aparatos les están costando más dinero. Yo mismo, hasta hace poco, no tenía ni idea de cuándo era más caro usar la lavadora o si realmente valía la pena desenchufar el televisor por las noches.

Lo más frustrante es que las opciones comerciales existentes tienen barreras de entrada significativas. Por ejemplo, sistemas como Schneider Electric's EcoStruxure o Siemens Building Technologies requieren inversiones iniciales considerables y conocimientos técnicos especializados para su instalación y mantenimiento. Estas soluciones, aunque efectivas, están orientadas principalmente al sector empresarial y viviendas de alta gama.

Las facturas que recibimos cada mes tampoco ayudan mucho. Solo te dicen cuánto has gastado en total, pero no te explican nada sobre qué aparatos han consumido más o en qué momentos del día. Es como si te dijeran que has gastado mucho dinero en el supermercado, pero sin darte el ticket de compra.

4.1.2. Oportunidad de Impacto

Por eso pensé en desarrollar EnergiApp v1.0. Durante el desarrollo del proyecto, utilizaré datasets existentes como UK-DALE para entrenar y validar los algoritmos de Machine Learning. Sin embargo, en un entorno de producción real, la plataforma estaría diseñada para recibir datos directamente de dispositivos IoT instalados en los hogares de los usuarios, generando así datasets personalizados y específicos para cada familia.

Mi objetivo es crear una interfaz intuitiva que sea accesible para usuarios sin conocimientos técnicos especializados. La idea es que desde dispositivos móviles, tablets o ordenadores, los usuarios puedan visualizar los patrones de consumo de sus propios dispositivos y recibir recomendaciones prácticas basadas en sus hábitos reales de uso energético.

Si los resultados del proyecto son positivos, esto podría contribuir a que las familias reduzcan su consumo energético. Estudios previos sugieren que mejoras en la gestión doméstica pueden generar ahorros del 15 % al 25 % en las facturas eléctricas, lo que representaría entre 140 y 240 euros anuales para una familia media.

4.2 Innovación Tecnológica

4.2.1. Convergencia de Tendencias Actuales

Para el desarrollo técnico, he seleccionado React 18 junto con Node.js como tecnologías base, ya que son las que mejor domino actualmente. Durante el desarrollo de proyectos universitarios previos con React, pude apreciar su rendimiento y la calidad del código que se puede conseguir. La abundante documentación disponible también facilita la resolución de dudas durante el desarrollo.

Node.js me parece una elección apropiada para el desarrollo de APIs, y el hecho de poder utilizar JavaScript tanto en frontend como backend simplifica la arquitectura del proyecto.

La implementación de Machine Learning es el aspecto que más me motiva del proyecto. El objetivo es desarrollar algoritmos que analicen patrones reales de consumo energético, basándose en los conocimientos de estadística y análisis de datos adquiridos durante la carrera.

También considero fundamental que la aplicación tenga una experiencia de usuario

óptima. He observado que muchas aplicaciones técnicamente sólidas fallan en usabilidad. Por ello, el diseño será responsive y priorizará la simplicidad de uso tanto en dispositivos móviles como de escritorio.

4.2.2. Diferenciación Competitiva

He estado comparando mi idea con lo que ya existe en el mercado, y creo que podría tener algunas ventajas interesantes:

Los sistemas comerciales existentes tienen costes considerables y barreras de entrada altas. Mi propuesta buscaría ser más accesible económicamente. Para este TFG académico, utilizaré datos simulados para desarrollar y validar el sistema, pero en un entorno de producción real, la plataforma estaría diseñada para trabajar con dispositivos IoT que los usuarios tendrían instalados en sus hogares, procesando datos reales de consumo.

También he revisado aplicaciones móviles que ofrecen servicios similares, pero muchas se limitan a mostrar información básica de consumo. Mi planteamiento es intentar ir más allá: no solo mostrar datos, sino proporcionar recomendaciones prácticas que los usuarios puedan implementar.

Una característica que me parece valiosa de este enfoque es la transparencia del código. Al desarrollarse como proyecto académico con posible orientación open-source, permitiría adaptaciones y mejoras por parte de la comunidad, algo que no es posible con los sistemas comerciales propietarios.

4.3 Viabilidad Técnica y Económica

4.3.1. Madurez Tecnológica

Respecto a la viabilidad técnica, tengo confianza en el proyecto basándome en mi experiencia con las tecnologías seleccionadas. React 18 y Node.js son tecnologías maduras, ampliamente adoptadas en la industria y con ecosistemas de desarrollo consolidados.

Para la visualización de datos utilizaré Chart.js, una librería que he empleado en proyectos anteriores y que ofrece buena estabilidad. Express.js para el backend cuenta con documentación extensa y una comunidad activa. En caso de encontrar dificultades técnicas durante el desarrollo, existe abundante material de referencia y recursos de apoyo disponibles.

La elección de tecnologías populares y bien establecidas reduce significativamente los riesgos técnicos del proyecto, ya que cuentan con comunidades de desarrolladores que han resuelto problemas similares previamente.

4.3.2. Impacto Económico Proyectado

Desde el punto de vista económico, los costes de desarrollo son favorables. Al tratarse de un proyecto académico que utiliza tecnologías open-source, los gastos directos de desarrollo son mínimos.

Para un eventual despliegue más amplio, servicios de hosting básicos comenzarían desde aproximadamente 5 euros mensuales, lo que contrasta favorablemente con los costes de los sistemas comerciales especializados.

El valor potencial para los usuarios finales es considerable. Si el sistema logra optimizaciones del 15 % al 25 % en el consumo eléctrico, como sugieren estudios del sector, esto representaría ahorros de 140 a 240 euros anuales para una familia media. Este beneficio económico directo podría justificar la adopción de la herramienta.

5 Propuesta de Índice

La estructura completa del Trabajo Final se organizará en las siguientes partes y capítulos:

5.1 Parte I: Fundamentos y Contexto

Capítulo 1: Introducción (*Entrega Parcial 1*)

- 1.1. Presentación del problema energético doméstico
- 1.2. Justificación del proyecto EnergiApp v1.0
- 1.3. Objetivos generales y específicos
- 1.4. Estructura del documento
- 1.5. Metodología de desarrollo ágil

Capítulo 2: Marco Teórico y Estado del Arte (*Entrega Parcial 2*)

- 2.1. Fundamentos de gestión energética doméstica
- 2.2. Machine Learning aplicado a predicción de consumo
- 2.3. Desarrollo web moderno (React 18, Node.js, Express)
- 2.4. Técnicas de visualización de datos (Chart.js)
- 2.5. Revisión de aplicaciones de gestión energética existentes
- 2.6. Análisis comparativo y brechas identificadas

5.2 Parte II: Análisis y Diseño

Capítulo 3: Análisis de Requisitos (*Entrega Parcial 2*)

- 3.1. Requisitos funcionales del sistema
- 3.2. Requisitos no funcionales (rendimiento, usabilidad, escalabilidad)
- 3.3. Casos de uso y historias de usuario
- 3.4. Perfiles de usuario (administrador, usuario estándar)
- 3.5. Restricciones técnicas y limitaciones

Capítulo 4: Diseño del Sistema (*Entrega Parcial 2*)

- 4.1. Arquitectura full-stack de la aplicación
- 4.2. Diseño de la base de datos y modelos de datos
- 4.3. Diseño de APIs RESTful y endpoints
- 4.4. Diseño de algoritmos de Machine Learning
- 4.5. Diseño de interfaz de usuario y UX/UI
- 4.6. Arquitectura de seguridad y autenticación

5.3 Parte III: Implementación y Desarrollo

Capítulo 5: Desarrollo del Backend (*Entrega Parcial 3*)

- 5.1. Configuración del entorno Node.js y Express.js
- 5.2. Implementación del sistema de autenticación
- 5.3. Desarrollo de APIs RESTful (30+ endpoints)
- 5.4. Sistema de logging con Winston
- 5.5. Middleware de seguridad (CORS, Helmet)
- 5.6. Implementación de algoritmos ML predictivos

Capítulo 6: Desarrollo del Frontend (*Entrega Parcial 3*)

- 6.1. Configuración de React 18 y arquitectura de componentes
- 6.2. Implementación de dashboards interactivos

- 6.3. Sistema de visualizaciones con Chart.js
- 6.4. Interfaces responsivas y móvil-first
- 6.5. Gestión de estado y comunicación con backend
- 6.6. Optimización de rendimiento y experiencia de usuario

Capítulo 7: Funcionalidades Inteligentes (*Entrega Parcial 3*)

- 7.1. Motor de predicciones ML (1-7 días)
- 7.2. Sistema de recomendaciones automatizadas
- 7.3. Control y automatización de dispositivos
- 7.4. Panel administrativo multi-usuario
- 7.5. Sistema de notificaciones y feedback
- 7.6. Integración de datos meteorológicos

5.4 Parte IV: Validación y Resultados

Capítulo 8: Pruebas y Validación (*Entrega Final*)

- 8.1. Metodología de testing y QA
- 8.2. Pruebas unitarias y de integración
- 8.3. Pruebas de rendimiento y carga
- 8.4. Validación de algoritmos ML y precisión
- 8.5. Testing de usabilidad y experiencia de usuario
- 8.6. Pruebas de compatibilidad móvil y responsive

Capítulo 9: Análisis de Resultados (*Entrega Final*)

- 9.1. Métricas de rendimiento del sistema
- 9.2. Evaluación de precisión de predicciones ML
- 9.3. Análisis de ahorro energético simulado
- 9.4. Evaluación de experiencia de usuario
- 9.5. Comparativa con soluciones existentes
- 9.6. Validación de objetivos propuestos

5.5 Parte V: Conclusiones y Proyección

Capítulo 10: Conclusiones (*Entrega Final*)

- 10.1. Cumplimiento de objetivos académicos
- 10.2. Contribuciones técnicas principales
- 10.3. Lecciones aprendidas en el desarrollo
- 10.4. Limitaciones identificadas y superadas

Capítulo 11: Trabajo Futuro y Escalabilidad (*Entrega Final*)

- 11.1. Mejoras propuestas para versión 2.0
- 11.2. Integración con dispositivos IoT reales
- 11.3. Expansión a redes comunitarias inteligentes
- 11.4. Potencial comercial y modelo de negocio

5.6 Anexos

- **Anexo A:** Código fuente completo (6700+ líneas)
- **Anexo B:** Documentación técnica de APIs
- **Anexo C:** Manual de usuario y administrador
- **Anexo D:** Capturas de pantalla y demos
- **Anexo E:** Métricas de rendimiento y testing

6 Alcance del Proyecto

6.1 Alcance Incluido (Funcionalidades Principales)

6.1.1. Aplicación Frontend React

Dashboard Principal:

- Métricas en tiempo real (consumo, costo, estado del sistema)
- Visualizaciones interactivas con Chart.js
- Análisis comparativo con días anteriores

- Control de dispositivos domésticos simulados
- Diseño responsive perfecto (móvil, tablet, desktop)

Módulo de Predicciones ML:

- Selector dinámico de 1-7 días de predicción
- Tarjetas predictivas con información meteorológica
- Algoritmos basados en patrones reales de consumo
- Datos consistentes y no aleatorios
- Recomendaciones específicas por día

Sistema de Recomendaciones:

- “Optimizar Standby” - Control automático de dispositivos en standby
- “Programar Lavadora” - Automatización temporal con demo funcional
- “Información Paneles Solares” - Cálculos de ROI e inversión
- Notificaciones con feedback inmediato
- Validación previa de dispositivos disponibles

6.1.2. Backend Node.js + Express

API RESTful Completa:

- 30+ endpoints para gestión completa del sistema
- Autenticación con sistema de tokens JWT-like
- Middleware de seguridad (CORS, Helmet)
- Sistema de logging con Winston
- Gestión de usuarios y dispositivos

Sistema de Datos:

- Base de datos in-memory optimizada para desarrollo
- Modelos de usuarios, dispositivos, y consumo energético
- Simulación realista de datos energéticos
- Analytics y métricas del sistema

6.2 Limitaciones y Exclusiones

6.2.1. Limitaciones Técnicas

Infraestructura:

- Base de datos persistente (usa almacenamiento en memoria)
- Autenticación OAuth2 avanzada (implementa sistema básico)
- Cifrado TLS/SSL para producción
- Balanceador de carga para alta disponibilidad

Integración Hardware:

- Conexión con dispositivos IoT físicos reales
- Medidores inteligentes certificados
- Control de electrodomésticos de alta potencia
- Sensores ambientales reales

6.2.2. Funcionalidades Fuera de Alcance

Análisis Avanzado:

- Redes neuronales profundas para ML
- Integración con APIs meteorológicas comerciales
- Análisis predictivo con datos históricos reales de años
- Optimización de tarifas eléctricas en tiempo real

Despliegue Comercial:

- Infraestructura de producción escalable
- Sistema de facturación comercial
- Soporte técnico 24/7
- Integración con sistemas domóticos comerciales

7 Planificación y Cronograma

7.1 Metodología de Desarrollo

7.1.1. Enfoque Metodológico

Para organizar el trabajo he pensado en usar una metodología más o menos ágil. Me gusta la idea de trabajar por sprints semanales porque me ayuda a no procrastinar y a tener siempre algo concreto que hacer cada semana.

Mi plan es ir validando las cosas a medida que las hago, en lugar de programar todo y al final descubrir que algo no funciona. Es especialmente importante en este proyecto porque quiero que sea muy fácil de usar, y eso solo lo sabes probándolo.

7.1.2. Herramientas de Gestión y Desarrollo

Las herramientas que voy a usar son las típicas que ya conozco:

Git para el control de versiones, que ya uso en todos mis proyectos de la uni. Visual Studio Code como editor, con los plugins de React y Node.js que hacen que programar sea mucho más cómodo.

Para hacer pruebas de las APIs usaré Postman, que es muy intuitivo, y para depurar el frontend las herramientas que trae el navegador de serie.

La documentación técnica la haré en Markdown porque es rápido y limpio, y la memoria académica en LaTeX porque queda más profesional.

7.2 Cronograma de Entregas Oficiales

Entrega	Inicio	Fin	Días	Contenido
Entrega 1	07/07/2025	20/07/2025	14	Fundamentos de la propuesta Título, objetivos, justificación Propuesta de índice, alcance Planificación y cronograma
Entrega 2	21/07/2025	03/08/2025	14	Análisis de requisitos del software Especificaciones técnicas Diagramas de estructura y diseño Modelo de persistencia de datos Primer prototipo funcional Feedback de Entrega 1
Entrega 3	04/08/2025	17/08/2025	14	Presentación parcial del 60 % del contenido del proyecto
Entrega 4	18/08/2025	07/09/2025	21	DEPÓSITO FINAL TFB completo (100 %)

Tabla 2: Cronograma oficial de entregas

7.3 Estado Actual del Proyecto

7.3.1. Enfoque de Desarrollo

Mi estrategia va a ser empezar por lo básico: primero el backend con las APIs principales, después el frontend básico, y al final las funcionalidades más avanzadas de Machine Learning.

Lo que me parece súper importante es que cada entrega tenga algo que realmente funcione y que se pueda tocar. No quiero presentar solo documentos o código que no se ejecuta. Prefiero tener algo simple pero que funcione a algo súper complicado que está a medias.

Mi objetivo es que al final de cada sprint pueda enseñar algo concreto, aunque sea pequeño. Creo que así voy a aprender más y el tutor va a poder darme feedback útil antes de que sea demasiado tarde para cambiar cosas.

7.3.2. Preparación para las Sigüientes Entregas

El proyecto está estructurado para facilitar el desarrollo ordenado de las siguientes entregas según los requisitos académicos:

- **Entrega 2:** Fundamentos de la propuesta incorporando análisis de requisitos del software, especificaciones técnicas, diagramas de estructura y diseño, modelo de persistencia de datos y primer prototipo funcional.
- **Entrega 3:** Memoria con avance del 60 % incluyendo pruebas y depuraciones de software, evaluación y validación, documentación de resultados. Contendrá todas las correcciones y comentarios del tutor/a hasta ese momento, respetando aspectos formales requeridos.
- **Entrega Final:** Depósito del TFB con versión final incorporando todos los comentarios del tutor/a. Documento PDF definitivo con anexos y recursos adjuntos no modificables.

8 Recursos y Herramientas

8.1 Recursos Tecnológicos

8.1.1. Hardware de Desarrollo

- **Equipo Principal:** PC Windows con 16GB RAM, procesador i7
- **Testing:** Dispositivos móviles y tablets para pruebas responsive

- **Networking:** Conexión estable para desarrollo y testing de APIs
- **Backup:** Almacenamiento en la nube para respaldo de código

8.1.2. Stack Tecnológico Completo

Frontend (React Ecosystem):

- React 18 con Hooks y gestión de estado moderna
- Chart.js para visualizaciones interactivas
- CSS Grid/Flexbox para diseño responsive
- Axios para comunicación HTTP
- Create React App para configuración optimizada

Backend (Node.js Stack):

- Node.js 18+ con Express.js framework
- Winston para logging profesional
- Helmet y CORS para seguridad
- Sistema de autenticación JWT-like
- APIs RESTful con validación de datos

Herramientas de Desarrollo:

- Visual Studio Code con extensiones especializadas
- Git para control de versiones
- Postman para testing de APIs
- DevTools de navegador para debugging
- NPM para gestión de dependencias

8.2 Recursos Humanos

8.2.1. Equipo del Proyecto

- **Desarrollador Full-Stack:** Oliver Vincent Rice (dedicación completa)
- **Tutor Académico:** Isabel Sánchez
- **Recursos de Consulta:** Documentación oficial, Stack Overflow, GitHub

8.2.2. Competencias Técnicas Aplicadas

En cuanto a las competencias que voy a aplicar, creo que tengo bastante claro lo que necesito:

Para el frontend, domino React 18, JavaScript moderno, CSS con Grid y Flexbox, y tengo nociones básicas de diseño UX/UI que he aprendido en algunas asignaturas.

En el backend me manejo bien con Node.js, Express.js, el diseño de APIs RESTful y sistemas básicos de autenticación web.

La parte de Machine Learning es donde más tengo que aprender, pero tengo una base sólida de algoritmos y análisis de patrones de las asignaturas de estadística.

Y las herramientas de desarrollo como Git, testing y debugging las uso casi a diario desde hace un par de años.

Referencias

- [1] Pérez-Chacón, R., Luna-Romera, J.M., Troncoso, A., Martínez-Álvarez, F., & Riquelme, J.C. *Big data analytics for discovering electricity consumption patterns and energy savings*. Energies, 11(3), 683, 2018.
- [2] Jack Kelly and William Knottenbelt. *The UK-DALE dataset: Domestic appliance-level electricity demand and whole-house demand from five UK homes*. Scientific data, 2:150007, 2015.
- [3] UCI Machine Learning Repository. *Individual household electric power consumption data set*. <https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption>, 2007.
- [4] Meta Open Source Team. *React 18 Documentation*. <https://react.dev/>, 2024.
- [5] OpenJS Foundation. *Node.js Documentation*. <https://nodejs.org/docs/>, 2024.