ASIGNACIÓN AUTOMÁTICA DE HORARIOS DE CLASE (HERMES)

ARIEL ANDRES REYES TUAY

UNIVERSIDAD INTERNACIONAL DEL TRÓPICO AMERICANO-UNITRÓPICO

FACULTAD DE INGENIERÍAS

PROGRAMA DE INGENIERÍA DE SISTEMAS

YOPAL - CASANARE

2025

ASIGNACIÓN AUTOMÁTICA DE HORARIOS DE CLASE (HERMES)

Nombre del Autor

Ariel Andres Reyes Tuay

Proyecto presentado para optar por el título de Ingeniero de Sistemas

Tutor
Cesar Dayan Martelo Varela, Especialización Tecnológica en Web Máster

UNIVERSIDAD INTERNACIONAL DEL TRÓPICO AMERICANO-UNITRÓPICO

FACULTAD DE INGENIERÍAS

PROGRAMA DE INGENIERÍA DE SISTEMAS

YOPAL - CASANARE

2025

Dedicatoria

Agradecimientos

En primer lugar, quiero extender mi más profundo agradecimiento a la Universidad Internacional del Trópico Americano (Unitrópico) por brindarme la oportunidad de llevar a cabo esta investigación y desarrollo. La visión de la institución de mejorar constantemente la calidad educativa ha sido la fuente de inspiración detrás de este proyecto, y me siento honrado de haber sido parte de este esfuerzo. Mi reconocimiento especial va dirigido a mis queridos docentes y profesores de Unitrópico. Su orientación, conocimientos y experiencia han sido una fuente invaluable de inspiración para este trabajo. Sus enseñanzas y asesoramiento han sido cruciales para la realización de este proyecto. No puedo dejar de agradecer a mis colegas ingenieros, cuya colaboración y dedicación fueron esenciales para llevar a cabo el desarrollo tecnológico de esta aplicación. Sus contribuciones técnicas y creativas han dado forma a la solución tecnológica que presento hoy. El incansable trabajo del personal administrativo y de apoyo de la universidad ha allanado el camino para la implementación exitosa de este proyecto. Su apoyo logístico, administrativo y organizativo ha sido de un valor incalculable. Por último, pero no menos importante, quiero agradecer a todas las personas que, de una manera u otra, han contribuido al desarrollo de este proyecto. Sus aportes y colaboración han sido invaluables.

Tabla de Contenido

Pág

GLOSARIO 17

RESUMEN 20

ABSTRACT 22

Introducción 24

Planteamiento Del Problema 25

Justificación 27

- Para La Administración Y Coordinaciones 27
- Para Los Docentes 27
- Para Los Estudiantes 27
- Para La Universidad En General 28

Delimitación 30

Estado Del Arte 32

- *Celcat Timetabler:* *34*
- *Infosilem* *34*
- *Semestry (Termtime, Roomtime, Mytimetable)* *34*
- *Asc Timetables* *34*
- *Prime Timetable* *34*
- *Ofcourse* *35*
- *Cronos Timetable* *35*
- *Otros* *35*
- *Unitime* *35*
- *Fet (Free Evolutionary Timetabling)* *36*

- *Timefinder* *36*

Brechas Y Limitaciones De Las Soluciones Actuales 37

- *Poca Adaptación A Contextos Específicos* *37*
- *Optimización Limitada* *37*
- *Experiencia De Usuario* *37*

II. Asignación De Horarios Mediante Programación Entera (Pe) 38

A. Fundamentos Y Modelado Con Pe 38

- *Recopilación De Datos* *38*
- *Planificación Y Definición De Variables* *38*
- *Formulación De La Función Objetivo* *38*
- *Establecimiento De Restricciones* *38*

B. Manejo De Restricciones 38

- *Restricciones Duras (Hard Constraints)* *39*
- *Restricciones Blandas (Soft Constraints)* *39*

C. Solvers Y Herramientas De Pe 39

- *Solvers Comerciales* *39*
- *CPLEX (IBM)* *39*
- *Gurobi* *40*
- Solvers De Código Abierto 40
- *C++ Glpk (Gnu Linear Programming Kit)* *40*
- *Open Solver* *40*

Programación Por Restricciones (Cp) 40

- *Google Or-Tools Cp-Sat* *40*

D. Desafíos Y Estrategias En La Aplicación De Pe 40

- *Descomposición Del Problema* *41*
- *Simplificación Del Modelo* *41*
- *Hibridación Con Heurísticas/Metaheurísticas (Ilph/Milph)* *41*
- *Modelado Preciso Y Contextualización* *41*

E. Casos De Estudio Y Benchmarks 41

- *Tesis UANL* *41*
- *Instancias De Udine* *42*
- *Universidad De La Salle (Uls)* *42*
- *Enseñanza Híbrida* *42*
- *Cursos Multi-Sección Y Preferencias* *42*

F. Tendencias Emergentes Y Futuro 42

- *Integración De Pe Con Aprendizaje Automático (Ml)* *42*
- *Alta Tasa De Implementación* *43*
- *Investigación Futura* *43*

Iii. Comparación Y Sinergias: Pe Vs. Otras Técnicas 43

Pe Vs. Metaheurísticas *43*

Pe Vs. Programación Por Restricciones (Cp) *44*

Conclusión Del Estado Del Arte 44

Objetivos 45

Objetivo General 45

Objetivos Específicos 45

Marco Referencial 46

6.1 Marco Teórico 46

6.1.1 46

6.1.2 Sistemas De Gestión Académica Y Planificación De Horarios Universitarios 47

6.1.3 Programación Matemática Y Optimización Aplicada A Horarios 47

6.1.4 Arquitectura De Aplicaciones Web Modernas (Frontend, Backend, Base De Datos) **¡Error! Marcador no definido.**

- *Frontend (Cliente)* ..... ***¡Error! Marcador no definido.***
- *Backend (Servidor/Lógica De Negocio)* ..... ***¡Error! Marcador no definido.***
- *Base De Datos* ..... ***¡Error! Marcador no definido.***

6.1.5 Tecnologías Clave Para El Desarrollo De Software (Angular, Firebase, Python, Bootstrap, Font Awesome) ..... **¡Error! Marcador no definido.**

- *Angular 18* ..... ***¡Error! Marcador no definido.***
- *Firebase 9* ..... ***¡Error! Marcador no definido.***
- *Python* ..... ***¡Error! Marcador no definido.***
- *Bootstrap 5* ..... ***¡Error! Marcador no definido.***
- *Font Awesome* ..... ***¡Error! Marcador no definido.***

6.2 Marco Conceptual 48

6.2.1 Términos Específicos De La Asignación De Horarios 48

- *Asignación De Horarios* *48*
- *Conflictos De Horarios* *48*
- *Optimización* ..... ***¡Error! Marcador no definido.***
- *Prioridades Curriculares* *48*
- *Recursos Académicos* *48*

- *Interfaz De Usuario (Ui)* *48*
- *Lógica De Negocio* *48*

6.2.2 Conceptos Tecnológicos Clave Del Proyecto 49

- *Angular 18* *49*
- *Firebase 9* *49*
- *Python* *49*
- *Programación Entera (Pe)* *49*
- *Bootstrap 5* *49*
- *Font Awesome* *50*
- *Cloud Functions (Firebase)* *50*

6.3 Marco Geográfico ..... **¡Error! Marcador no definido.**

6.4 Marco Demográfico ..... **¡Error! Marcador no definido.**

- *Estudiantes* ..... ***¡Error! Marcador no definido.***
- *Docentes* ..... ***¡Error! Marcador no definido.***
- *Personal Administrativo (Coordinadores, Jefes De Programa, Secretarías):.* ***¡Error! Marcador no definido.***

6.5 Marco Legal 50

Ley 30 De 1992 (Ley General De Educación Superior En Colombia) *50*

Reglamentos Académicos Internos De Unitrópico *50*

Políticas De Privacidad Y Protección De Datos *51*

Metodología 51

7.1 Enfoque Metodológico 51

- *Componente Cualitativo* ..... ***¡Error! Marcador no definido.***

- *Componente Cuantitativo* ..... ***¡Error! Marcador no definido.***
- *Investigación Aplicada De Tipo Constructivo/Desarrollo Tecnológico* ..... ***¡Error! Marcador no definido.***
- *Programación Extrema (Xp)*..... *51*
- *Kanban (Conceptual)*..... ***¡Error! Marcador no definido.***

7.2 Diseño de la Investigación y Desarrollo ..... **¡Error! Marcador no definido.**

Fase 1: Investigación Inicial, Fundamentación Y Planificación (Septiembre 2022 - Diciembre 2022; ~4 Meses)..... **¡Error! Marcador no definido.**

Fase 2: Diseño Conceptual Y Técnico Detallado Del Sistema Hermes (Enero 2023 - Abril 2023; ~4 Meses)..... **¡Error! Marcador no definido.**

Fase 3: Desarrollo Iterativo De Módulos Base Y Funcionalidades Crud (Mayo 2023 - Febrero 2024; ~10 Meses)..... **¡Error! Marcador no definido.**

Fase 4: Desarrollo, Pruebas E Integración Del Algoritmo De Optimización (Noviembre 2023 - Diciembre 2024; ~14 Meses, Con Solapamiento Con Fase 3 Y Extendiéndose En Fase 5) **¡Error! Marcador no definido.**

Fase 5: Pruebas Integrales Del Sistema, Refinamiento Y Optimización (Enero 2025 - Abril 2025, Continuando Parcialmente; ~4 Meses Y En Curso) ..... **¡Error! Marcador no definido.**

Fase 6: Documentación Final, Desarrollo Adicional Y Preparación Para Entrega (Mayo 2025 - Actualidad Y Continuación) ..... **¡Error! Marcador no definido.**

Población y Muestra 52

Población Objeto De Estudio..... 52

Fuentes De Información Para Requisitos Y Validación Conceptual 52

- *Documentos De Propuesta* ..... ***¡Error! Marcador no definido.***

- *Aportes Conceptuales Específicos* *52*
- *Tutor Del Proyecto* ..... ***¡Error! Marcador no definido.***
- *Muestra Para Pruebas (Conceptual)*..... ***¡Error! Marcador no definido.***

7.4 Métodos de Investigación y Desarrollo 52

7.4.1. Métodos De Recolección De Datos Primarios E Información Clave 52

Análisis Documental (Fundamental) 52

- *Documentos De La Propuesta De Grado* ..... ***¡Error! Marcador no definido.***
- *Diagrama Y Audio Explicativo* *52*

Análisis Del Código Fuente Del Proyecto (Auto-Reflexivo Y Continuo) ... **¡Error! Marcador no definido.**

7.4.2. Instrumentos de Recolección y Sistematización de Datos:. **¡Error! Marcador no definido.**

- Fichas De Análisis Documental..... **¡Error! Marcador no definido.**
- Registro De Desarrollo (Bitácora Personal Del Desarrollador) **¡Error! Marcador no definido.**

7.4.3. Plan de Recolección de Datos e Información Clave..... **¡Error! Marcador no definido.**

7.4.4. Técnicas de Procesamiento de Datos Utilizadas..... **¡Error! Marcador no definido.**

Datos Cualitativos (De Documentos, Diagrama, Audio)..... **¡Error! Marcador no definido.**

- *Análisis De Contenido Temático* ..... ***¡Error! Marcador no definido.***
- *Síntesis E Interpretación*..... ***¡Error! Marcador no definido.***

Datos Estructurados Del Sistema Hermes Y Para El Algoritmo:. **¡Error! Marcador no definido.**

- Métricas De Evaluación Del Algoritmo ..... **¡Error! Marcador no definido.**

7.4.5. Técnicas de Análisis de Datos y Desarrollo de la Solución HERMES: .. **¡Error! Marcador no definido.**

- *Análisis De Requisitos* *53*
- *Modelado Del Problema De Optimización Con Or-Tools (Cp-Sat)* ***¡Error! Marcador no definido.***
- *Ciclo De Desarrollo De Software* *53*
- *Análisis De Resultados Del Algoritmo (Iterativo)* ..... ***¡Error! Marcador no definido.***
- *Evaluación Formativa De Usabilidad* ..... ***¡Error! Marcador no definido.***

RESULTADOS 53

8.1 Introducción Breve A Los Resultados ..... **¡Error! Marcador no definido.**

8.2 Recopilación De Datos Para El Sistema Hermes 54

1. *Análisis Documental* *54*
2. *Especificación De Entidades Y Restricciones* *54*
3. *Datos De Prueba* ..... ***¡Error! Marcador no definido.***

8.3 Presentación De Resultados Del Desarrollo De Hermes 54

8.3.1 Aplicación Web Hermes: Arquitectura Y Tecnologías Implementadas 54

8.3.2 Modelo De Datos Implementado En Cloud Firestore 55

8.3.3 Módulos Y Componentes Desarrollados En El Frontend (Angular) 55

- *Módulo De Autenticación (Auth)* *55*
- *Módulos De Gestión De Datos (En Src/App/Pages/)* *55*

Módulo De Horarios (Src/App/Pages/Horarios/) 55

- *Componente De Generación* *56*
- *Componente De Visualización De Horarios Asignados* *56*
- *Componentes Compartidos (Shared)* *56*
- *Rutas* *56*

8.3.4 Backend Api Y Lógica De Negocio (Firebase Cloud Functions - Python) 56

- *Api Endpoints* *56*
- *Lógica De Negocio* *56*

8.3.5 Implementación Y Funcionamiento Del Algoritmo De Generación De Horarios 56

8.4 Análisis De Resultados 57

- *Cumplimiento Del Objetivo De Gestión De Datos* *57*
- *Cumplimiento Del Objetivo De Generación Automática (Núcleo)* *57*
- *Arquitectura Sólida* *58*
- *Eficiencia Potencial* *58*
- *Reducción De Conflictos* *58*

8.5 Comparación Con Literatura Previa 58

Diferencias Y Valor Agregado 58

- *Adaptación Específica Y Resiliencia En El Desarrollo* *58*
- *Pila Tecnológica Integrada Y Accesible* *59*
- *Enfoque En La Funcionalidad Operativa* *59*

8.6 Discusión 59

- Relevancia E Implicaciones Prácticas 59
- Implicaciones Teóricas 59

Limitaciones Del Estudio Y Desarrollo Actual: 59

- *Proceso De Desarrollo Extendido Y Reinicios* *59*
- *Visualización De Horarios En Desarrollo* *60*

Pruebas Exhaustivas Con Datos Completos De Unitrópico 60

- *Optimización De Restricciones Blandas* *60*

- *Gestión De Cambios Dinámicos* *60*

Recomendaciones Para Futuros Trabajos Y Desarrollo En Unitrópico: 60

0. Priorizar Y Finalizar El Módulo De Visualización De Horarios *60*

1. Estabilización Y Pruebas Piloto *60*

2. Documentar Lecciones Del Desarrollo *60*

3. Expandir Restricciones Blandas *61*

4. Capacitación Y Adopción *61*

5. Mantenimiento Y Evolución ..... ***¡Error! Marcador no definido.***

IMPACTOS 61

9.1 Impacto Tecnológico Y Científico 61

Modernización De Procesos Institucionales 61

Adopción Y Validación De Tecnologías Modernas 61

- *Frontend* *62*
- *Backend* *62*
- *Base De Datos* *62*
- *Optimización* *62*
- *Base Para Futuros Desarrollos Tecnológicos* *62*
- *Contribución Al Conocimiento Aplicado (Impacto Científico)* *62*

9.2 Impacto Organizacional, Administrativo Y Económico 63

- *Optimización Del Tiempo Y Los Recursos Humanos* *63*
- *Mejora En La Eficiencia Operativa Y La Toma De Decisiones* *63*
- *Reducción De Errores Y Conflictos Horarios* *63*
- *Transparencia Y Estandarización Del Proceso* *63*

- *Optimización Del Uso De La Infraestructura Física (Impacto Económico Indirecto) 63*
- *Soporte Al Crecimiento Institucional Eficiente 64*

9.3 Impacto Social y Académico (en la Comunidad Unitrópico) 64

- Mejora De La Experiencia Educativa Para Estudiantes 64
- Mejora De Las Condiciones De Trabajo Para Docentes 64
- Reducción Del Estrés Y La Incertidumbre Institucional 65
- Fomento De Una Cultura De Innovación Y Eficiencia 65
- Potencial Mejora En La Equidad 65

9.4 Otros Impactos Considerados 65

- *Impacto Ambiental 65*
- *Impacto Cultural (Institucional) 65*
- *Impacto Político (Institucional) 66*

Índice De Tablas

Cronograma De Actividades 66

Tabla 1 67

Cronograma de Actividades 67

Tabla 2 67

Cronograma Inicial 67

Tabla 2 68

Conclusiones 71

Referencias Bibliográficas 73

Anexos 77

GLOSARIO

Angular 18: Un *framework* de código abierto para construir aplicaciones web del lado del cliente (frontend), desarrollado por Google. Permite crear interfaces de usuario dinámicas y complejas de manera estructurada.

API (Application Programming Interface): Conjunto de definiciones y protocolos que se utiliza para diseñar e integrar software de aplicaciones. Permite la comunicación entre diferentes sistemas o componentes de software.

Bootstrap 5: Un *framework* de código abierto para el desarrollo web *front-end* que facilita la creación de sitios y aplicaciones web responsivos (adaptables a diferentes tamaños de pantalla) y estéticamente atractivos, a través de componentes HTML, CSS y JavaScript predefinidos.

CRUD (Create, Read, Update, Delete): Un acrónimo que describe las cuatro operaciones básicas que se pueden realizar en datos persistentes (por ejemplo, en una base de datos). Son los fundamentos de la interacción con la mayoría de las aplicaciones de gestión de información.

Firebase 9: Una plataforma de desarrollo de aplicaciones móviles y web de Google. Ofrece una variedad de servicios que incluyen una base de datos en tiempo real (Firestore), autenticación, alojamiento y funciones en la nube, simplificando el desarrollo y escalado de aplicaciones.

Font Awesome: Un conjunto de herramientas y bibliotecas de íconos vectoriales escalables que se pueden personalizar con CSS. Se utiliza para añadir elementos gráficos y visuales a las interfaces de usuario de manera sencilla.

Frontend: La parte de una aplicación web o software con la que interactúan directamente los usuarios. Incluye la interfaz de usuario y todos los elementos visuales y de interacción.

Modelo de Programación Entera: Una técnica de optimización matemática utilizada para la toma de decisiones, donde las variables de decisión deben tomar valores enteros. En el contexto de tu proyecto, se utiliza para resolver el problema de asignación de horarios de manera óptima, garantizando que ciertas asignaciones (como aulas o docentes) se realicen como unidades completas y sin fracciones.

Python: Un lenguaje de programación de alto nivel, interpretado y de propósito general. Es conocido por su sintaxis clara y legible, lo que lo hace popular para el desarrollo web, el análisis de datos, la inteligencia artificial y, en tu caso, para algoritmos de optimización.

Unitrópico (Universidad Internacional del Trópico Americano): La institución de educación superior en Yopal, Casanare, Colombia, que es el objeto de estudio y beneficiario del sistema de generación de horarios.

Vista (en desarrollo web): Se refiere a la parte de la interfaz de usuario de una aplicación web que presenta los datos al usuario. Cada "vista" de tu proyecto (login, registro, áreas deportivas, asignaturas, etc.) es una pantalla o sección específica con la que el usuario interactúa.

Web (Aplicación Web): Un programa de software que se ejecuta en un servidor web y se accede a través de un navegador web, lo que permite a los usuarios interactuar con él desde cualquier dispositivo con conexión a internet.

FIREBASE CLOUD FUNCTIONS: Entorno de ejecución serverless que permite ejecutar código de backend en respuesta a eventos emitidos por características de Firebase y solicitudes HTTPS, sin necesidad de gestionar servidores.

FLASK: Microframework ligero para el desarrollo de aplicaciones web en Python. Es comúnmente utilizado para construir APIs y backends simples y rápidos.

FRONTEND: Capa de presentación de una aplicación con la que el usuario interactúa directamente. En aplicaciones web, se construye típicamente con HTML, CSS y JavaScript (o frameworks como Angular).

GOOGLE OR-TOOLS: Suite de software de código abierto desarrollada por Google para resolver problemas de optimización combinatoria, como programación lineal, programación entera, programación por restricciones, y problemas de rutas de vehículos y flujo.

MOCKUP: Modelo a escala o de tamaño real de un diseño o dispositivo, utilizado para la enseñanza, demostración, evaluación del diseño, promoción y otros fines. En desarrollo de software, es una representación visual de la interfaz de usuario.

PYTHON: Lenguaje de programación interpretado, de alto nivel y multipropósito, conocido por su sintaxis clara y legible. Ampliamente utilizado en desarrollo web, ciencia de datos, inteligencia artificial y scripting.

PROGRAMACIÓN LINEAL (PL): Método matemático para determinar una forma de alcanzar el mejor resultado (como ganancia máxima o costo mínimo) en un modelo matemático dado cuyos requisitos están representados por relaciones lineales.

XP (EXTREME PROGRAMMING): Metodología de desarrollo de software ágil que busca producir software de mayor calidad y mejorar la capacidad de respuesta a los cambios en los requisitos del cliente. Se basa en valores como la comunicación, la simplicidad, la retroalimentación, el coraje y el respeto.

RESUMEN

La asignación manual de horarios académicos en la Universidad Internacional del Trópico Americano (Unitrópico) constituye un proceso crítico propenso a ineficiencias, consumo extensivo de tiempo y generación de conflictos, una problemática exacerbada por el reciente incremento en la demanda estudiantil. Este trabajo de grado aborda dicho desafío, contextualizado en el debate académico sobre el Problema de Asignación de Horarios Universitarios (UCTP), reconocido por su complejidad NP-hard y la diversidad de enfoques de optimización matemática y heurística para su resolución. Teóricamente, el proyecto se fundamenta en los principios de la Programación Entera y la Programación por Restricciones para modelar y resolver las complejidades inherentes a la creación de horarios.

El objetivo general fue desarrollar HERMES (Herramienta de Elaboración y Reajuste de Mallas y Esquemas de Horarios), una aplicación web diseñada para la generación automática y optimizada de horarios de clase en Unitrópico. Para alcanzar esta meta, se aplicó una metodología de desarrollo tecnológico adaptativa e incremental, que implicó el diseño y construcción de un sistema con frontend en Angular, backend con Python y Firebase Cloud Functions, y una base de datos en Cloud Firestore. El núcleo de la solución es un algoritmo de optimización implementado con Google, capaz de procesar las entidades académicas y restricciones específicas de la institución.

Como resultado principal, se obtuvo la aplicación web HERMES funcional, que permite la gestión de datos maestros (facultades, carreras, asignaturas, docentes, infraestructura) y la generación de horarios que cumplen con restricciones duras fundamentales y buscan satisfacer criterios de calidad. El impacto esperado de HERMES en Unitrópico incluye una optimización significativa del proceso de asignación de horarios, la reducción de conflictos, un uso más

eficiente de los recursos físicos y académicos, y una mejora general en la experiencia administrativa y educativa para la comunidad universitaria.

Palabras Clave: Programación de Horarios, Optimización, Programación por Restricciones, Aplicaciones Web, Google OR-Tools.

ABSTRACT

The manual academic timetabling process at the Universidad Internacional del Trópico Americano (Unitrópico) constitutes a critical process prone to inefficiencies, extensive time consumption, and conflict generation, a problem exacerbated by the recent increase in student demand. This degree project addresses this challenge, contextualized within the academic debate on the University Course Timetabling Problem (UCTP), renowned for its NP-hard complexity and the diversity of mathematical and heuristic optimization approaches for its resolution. Theoretically, the project is based on the principles of Integer Programming and Constraint Programming to model and solve the inherent complexities of timetable creation.

The general objective was to develop HERMES (Herramienta de Elaboración y Reajuste de Mallas y Esquemas de Horarios – Tool for Elaboration and Readjustment of Curricular Meshes and Schedule Schemes), a web application designed for the automatic and optimized generation of class schedules at Unitrópico. To achieve this goal, an adaptive and incremental technological development methodology was applied, involving the design and construction of a system with an Angular frontend, a backend with Python and Firebase Cloud Functions, and a Cloud Firestore database. The core of the solution is an optimization algorithm implemented with Google OR-Tools (CP-SAT solver), capable of processing academic entities and specific institutional constraints.

As a main result, the functional HERMES web application was obtained, which allows for master data management (faculties, careers, courses, teachers, infrastructure) and the generation of timetables that meet fundamental hard constraints and aim to satisfy quality criteria. The expected impact of HERMES at Unitrópico includes a significant optimization of the timetabling process, a reduction in conflicts, more efficient use of physical and academic

resources, and an overall improvement in the administrative and educational experience for the university community.

Keywords: Timetabling, Optimization, Constraint Programming, Web Applications, Google OR-Tools.

INTRODUCCIÓN

La gestión organizacional en instituciones de educación superior como Unitrópico, caracterizadas por un volumen considerable de estudiantes y programas académicos, es fundamental. Dentro de esta gestión, la creación de horarios de clase representa un proceso de alta complejidad. El presente proyecto surge como respuesta a esta problemática, identificada inicialmente por el Ingeniero Alfonso Tovar, exdirector del programa de Ingeniería de Sistemas. El desarrollo de un sistema web para abordar la complejidad inherente a la generación de horarios académicos se identificó como un desafío técnico significativo.

Actualmente, el proceso de elaboración de horarios en Unitrópico se realiza manualmente mediante hojas de cálculo, una tarea que demanda un considerable número de semanas. Esta metodología manual genera diversas dificultades, tales como el cruce de asignaturas, la subutilización de espacios especializados (laboratorios, áreas deportivas) y dificultades en la asignación de cupos, particularmente tras la transición de la universidad a entidad pública en 2021. Estas circunstancias pueden llevar a que los estudiantes busquen espacios adicionales y los docentes enfrenten situaciones de sobrecupo, evidenciando la ineficiencia del proceso actual.

El objetivo principal de esta tesis es el diseño e implementación de un software web para la generación automática de horarios académicos en Unitrópico. Se utilizarán tecnologías como Angular 18, Firebase 9 y programación entera con Python, con el fin de que el sistema organice eficientemente las asignaturas, los recursos físicos (aulas, laboratorios, áreas deportivas) y las prioridades específicas de cada programa académico.

El alcance del proyecto comprende la gestión de los elementos clave para la programación académica: docentes, asignaturas, programas, facultades, aulas, laboratorios y áreas deportivas. El sistema permitirá la configuración de disponibilidades y restricciones para

dichos elementos, desarrollado en Python, se encargará de procesar estas variables para generar horarios sin conflictos y asegurar el uso eficiente de los recursos. La aplicación web resultante se proyecta como una herramienta intuitiva y accesible desde navegadores. Las limitaciones identificadas durante el desarrollo se relacionaron con la complejidad inherente a los algoritmos, la necesidad de una comprensión detallada de las normativas específicas de Unitrópico y las restricciones temporales propias de un proyecto de grado.

Se espera que los resultados principales de este sistema incluyan una reducción significativa en el tiempo dedicado a la elaboración de horarios y la eliminación de conflictos de programación. Adicionalmente, se proyecta una optimización en el uso de los espacios físicos y una distribución más equitativa de las cargas de trabajo docente, contribuyendo a una mayor eficiencia institucional y una mejor experiencia para estudiantes y profesores.

PLANTEAMIENTO DEL PROBLEMA

La aplicación de la ingeniería de sistemas en el ámbito educativo, los procesos universitarios y la mejora continua de la calidad académica. en instituciones en fase de expansión, como la Universidad Internacional del Trópico Americano (Unitrópico), ubicada en Yopal, Casanare, ciertas tareas administrativas presentan una considerable complejidad. Entre ellas, la creación de horarios de clase se destaca como un proceso crítico que puede generar dificultades operativas. Esta situación no es exclusiva de Unitrópico; numerosas instituciones de educación superior a nivel global, de diversas magnitudes, han enfrentado desafíos similares en la gestión manual de sus horarios, constituyendo una problemática recurrente en el sector.

Actualmente, en Unitrópico, la elaboración de horarios se efectúa manualmente mediante hojas de cálculo. Si bien este método resultaba gestionable cuando la institución tenía menores dimensiones, la transición a universidad pública en 2021, con el consecuente incremento significativo en el número de estudiantes, ha tornado el sistema manual progresivamente insuficiente. El volumen de información y la complejidad de la coordinación se han incrementado sustancialmente. Con múltiples actores y variables involucradas (disponibilidad docente, asignación de aulas, requisitos de asignaturas, capacidad de grupos y particularidades de los programas), la probabilidad de errores y conflictos horarios aumenta. Esta tarea, de alta complejidad, resulta difícil de completar sin incurrir en inconsistencias cuando se aborda de forma enteramente manual.

Las consecuencias de este proceso afectan directamente a la comunidad universitaria. Los estudiantes pueden experimentar dificultades como la superposición de asignaturas, la saturación de aulas que podría requerir la búsqueda de mobiliario adicional, o condiciones que no son óptimas para el aprendizaje. La incertidumbre en la configuración de sus horarios puede generar inconvenientes, afectar la planificación académica y, potencialmente, el progreso en sus estudios. Para el personal docente, la asignación de carga académica y la verificación de disponibilidad de espacios representa un proceso administrativo que consume un tiempo considerable, estimado en varias semanas. Con frecuencia, los horarios resultantes podrían no optimizar el uso de recursos especializados, como laboratorios o áreas deportivas, lo que puede conducir a una utilización no ideal de la infraestructura institucional y a una significativa carga administrativa para coordinadores y facultades. Asimismo, pueden surgir conflictos en las asignaciones horarias de los docentes.

JUSTIFICACIÓN

Este proyecto no es solo un ejercicio técnico; nace de una necesidad muy real y urgente que se vive a diario en la Universidad Internacional del Trópico Americano (Unitrópico). La forma actual de gestionar los horarios de clase es, francamente, un cuello de botella que afecta a todos. Existe una clara brecha entre cómo se hacen las cosas ahora (manualmente, con Excel, con todos los problemas de cruces y la sobrecarga de trabajo administrativo) y cómo deberían ser (un proceso fluido, eficiente y sin estrés). Nuestro sistema busca cerrar esta brecha, pasando de una gestión que genera dolores de cabeza a una que optimiza la vida universitaria.

Los beneficios de implementar esta solución son tangibles y se sentirán en varios frentes:

- **Para La Administración Y Coordinaciones:** Se reducirá drásticamente el tiempo y el esfuerzo dedicados a la creación y ajuste de horarios. Lo que hoy toma semanas, se podrá hacer en cuestión de horas o minutos, liberando al personal para tareas más estratégicas. Esto se traduce en un ahorro de tiempo y recursos humanos considerable.

- **Para Los Docentes:** Tendrán mayor claridad en sus asignaciones y disponibilidad de aulas, minimizando los conflictos de horarios y permitiéndoles concentrarse en la enseñanza. Esto mejorará su satisfacción y eficiencia laboral.

- **Para Los Estudiantes:** Accederán a horarios claros, sin cruces inesperados, que les permitirán planificar mejor su vida académica y personal. Esto se reflejará en una mejor experiencia educativa y una reducción de la frustración, lo que indirectamente podría influir en su rendimiento y permanencia.

- **Para La Universidad En General:** Se logrará una utilización óptima de la infraestructura (aulas, laboratorios, áreas deportivas), que a menudo hoy está subutilizada o mal asignada. La toma de decisiones sobre la planificación académica será mucho más informada y estratégica.

La creación de un sistema de generación de horarios se alinea estratégicamente con la visión de Unitrópico como una universidad pública en crecimiento que busca la excelencia y la eficiencia. Para una institución que ha experimentado un aumento significativo en su población estudiantil desde 2021, la optimización de sus procesos internos no es una opción, sino una necesidad para garantizar una operación fluida y una experiencia de calidad para su comunidad. Este proyecto contribuye directamente a la modernización y al logro de una gestión académica de vanguardia.

Este es el momento adecuado para llevar a cabo este proyecto. La alta demanda actual y la complejidad administrativa que ha generado la masificación de estudiantes hacen que la necesidad de una solución automatizada sea más relevante y oportuna**.** que nunca no se trata de una mejora marginal, sino de una transformación necesaria para que Unitrópico pueda seguir creciendo y ofreciendo la calidad educativa que sus estudiantes merecen en un entorno de mayor volumen.

Los riesgos de no llevar a cabo este proyecto son claros y perjudiciales. Si la situación actual persiste, aumentarán los conflictos en los horarios, la frustración de estudiantes y docentes crecerá, el personal administrativo continuará sobrecargado, y los valiosos recursos físicos de la universidad seguirán subutilizados. En un entorno educativo competitivo, la ineficiencia en la gestión de horarios puede afectar la calidad percibida y la reputación de Unitrópico, limitando su

capacidad de crecimiento y desarrollo futuro. En pocas palabras, no actuar implicaría perpetuar un problema que erosiona la eficiencia y el bienestar de la comunidad universitaria.

De hecho, esta problemática de la asignación manual de horarios y sus consecuencias no es exclusiva de Unitrópico. Universidades en diversas partes del mundo han enfrentado desafíos similares, donde la complejidad de gestionar un número creciente de estudiantes, programas y recursos ha llevado a cuellos de botella operativos. Por ejemplo, instituciones como Purdue University o el MIT, si bien a una escala mucho mayor, han invertido en sistemas complejos como UniTime para optimizar sus procesos de asignación. Estos casos demuestran que la inversión en soluciones automatizadas no es un lujo, sino una necesidad para mantener la eficiencia y la satisfacción de la comunidad académica. La no adopción de una solución tecnológica puede llevar a situaciones de caos en la matriculación, subutilización de aulas y laboratorios, e incluso la pérdida de estudiantes que buscan instituciones con procesos más ágiles.

Nuestra solución, a diferencia de los sistemas comerciales masivos como CELCAT Timetabler o Infosilem que, si bien son robustos, suelen ser muy costosos y a menudo inflexibles a las particularidades de cada institución, busca ser una respuesta a medida y más cercana a la realidad de Unitrópico. Mientras que otras universidades invierten en soluciones pre empaquetadas o en desarrollar algoritmos de programación entera con *solvers* comerciales, nuestro enfoque se diferencia por el desarrollo propio del algoritmo de optimización en Python integrado con una plataforma flexible como Firebase, permitiéndonos adaptar las prioridades específicas de Unitrópico y sus grupos de estudiantes de manera directa. No pretendemos reemplazar un ecosistema completo, sino ofrecer una herramienta quirúrgica que resuelva un problema específico con una lógica de optimización transparente y controlada, evitando la

dependencia de terceros para ajustes finos o reglas muy particulares. Este enfoque nos permite una mayor agilidad y una adaptación precisa a las necesidades locales, algo que a menudo es un reto con soluciones estandarizadas.

Adicionalmente, somos conscientes de que el desarrollo de un sistema de esta envergadura, especialmente con la integración de un modelo de optimización tan sofisticado como la programación entera, conlleva una alta complejidad técnica. Existe la posibilidad real de que, a pesar de los esfuerzos, el proyecto no logre alcanzar todas las funcionalidades planificadas o su finalización completa dentro de los plazos establecidos. Reconocemos este desafío intrínseco al abordar un problema tan complejo, pero consideramos que el aprendizaje derivado de este proceso, la aplicación de metodologías de optimización y el intento de solucionar una problemática tan acuciante en la institución, justifican plenamente la inversión de esfuerzo. No hacer el intento significaría perder la oportunidad de generar una solución valiosa y de comprender a fondo las complejidades de la automatización en entornos universitarios.

DELIMITACIÓN

Para asegurar la viabilidad y el cumplimiento de los objetivos de este proyecto de grado, se establece con precisión su alcance y sus exclusiones. Esta clarificación busca alinear las expectativas y comprender el esfuerzo invertido dentro de un marco definido.

El sistema de generación de horarios propuesto se circunscribe exclusivamente a la Universidad Internacional del Trópico Americano (Unitrópico), sede principal en Yopal, Casanare. El desarrollo del proyecto se enmarca en el periodo académico 2022-2025, con el objetivo de consolidar una versión funcional del sistema. En términos funcionales, el sistema facilitará la

gestión de datos maestros esenciales, tales como docentes, asignaturas, carreras, facultades, aulas, laboratorios, áreas deportivas y bloques. Con base en estos datos, generará horarios considerando las prioridades y restricciones institucionales establecidas. Este proyecto establece una base sólida para futuras mejoras y expansiones.

El proyecto comprende el diseño y desarrollo de una aplicación web accesible mediante navegadores web, así como la implementación de módulos destinados a la administración de la información académica y de infraestructura de Unitrópico. De manera fundamental, incluye el desarrollo de un algoritmo basado en programación entera con Python para la asignación de horarios, diseñado para generar horarios que consideren la disponibilidad de recursos y los tiempos de trabajo. Adicionalmente, se implementa un sistema de autenticación y gestión de roles (administrador, director de programa) para el acceso seguro a la plataforma.

No se incluye el desarrollo de aplicaciones móviles nativas para sistemas operativos iOS o Android; el sistema se limita a una interfaz de aplicación web.

Los recursos para la ejecución de este proyecto, de carácter individual, consisten en un entorno de desarrollo con Angular 18, Firebase 9 (Firestore y Cloud Functions con Python), Bootstrap 5 y Font Awesome. Se dispone de acceso a equipos de cómputo personales y las herramientas de desarrollo de software requeridas. Los recursos económicos son limitados, de origen personal, y cubren principalmente la conectividad, sin financiación externa.

Las limitaciones principales que pueden influir en el desarrollo y la culminación del proyecto incluyen el marco temporal inherente a un proyecto de grado universitario. La complejidad intrínseca del problema de optimización de horarios, que podría necesitar ajustes y simplificaciones del modelo matemático para obtener soluciones en tiempos computacionalmente razonables, constituye un desafío significativo. Asimismo, la correcta

configuración del algoritmo depende críticamente de la claridad y completitud de las reglas y prioridades específicas de Unitrópico. La magnitud del proyecto implica un proceso de aprendizaje continuo y la superación de desafíos técnicos que pueden impactar los tiempos de desarrollo, un factor considerable en un proyecto individual de esta envergadura.

En cuanto a los riesgos y contingencias, se identifica la posibilidad de que el algoritmo no genere soluciones precisas bajo todas las condiciones posibles, especialmente en escenarios de alta restricción, o que el rendimiento del sistema presente desafíos ante volúmenes de datos considerables. Para mitigar estos riesgos, se adopta un enfoque modular y se utilizan Firebase Cloud Functions para el algoritmo en Python, favoreciendo la escalabilidad. Ante la materialización de estos riesgos, la estrategia de contingencia se centra en priorizar las funcionalidades clave del sistema, asegurar un producto mínimo viable que demuestre la base conceptual y la eficacia del algoritmo aplicando principios SOLID, y mantener una documentación técnica exhaustiva de los componentes desarrollados y su lógica.

ESTADO DEL ARTE

La asignación automática de horarios universitarios (UCTP, por sus siglas en inglés) es un problema combinatorio complejo y de gran importancia para las instituciones educativas. Su resolución eficiente impacta directamente en la utilización de recursos, la satisfacción de estudiantes y docentes, y la operatividad general de la universidad Dada su naturaleza NP-difícil, se han explorado diversas aproximaciones para abordarlo Este estado del arte se enfoca en dos vertientes principales: las soluciones de software disponibles en el mercado y el ámbito académico, y el uso de la programación entera (PE) como una técnica de optimización matemática fundamental para resolver este desafío.

Entender dónde estamos parados en el mundo de la tecnología y las soluciones existentes es clave antes de proponer algo nuevo. Por eso, para esta sección, realice una revisión bibliográfica sistemática buscando proyectos y avances relacionados con la generación de horarios y sistemas de gestión educativa. La búsqueda se centró en artículos científicos, informes técnicos, conferencias especializadas y documentación oficial de tecnologías relevantes, publicados principalmente entre 2018 y 2024. El objetivo de esta revisión fue triple: identificar los enfoques metodológicos más innovadores, reconocer los vacíos en el conocimiento que nuestro proyecto podría llenar, y detectar las tendencias tecnológicas que están marcando el paso en este campo, todo para asegurar que nuestra propuesta no solo sea actual, sino que aporte algo realmente valioso.

Tendencias y Avances Recientes**:** El campo de la optimización y la gestión de recursos en entornos educativos ha evolucionado rápidamente. Hemos visto una clara tendencia hacia el uso de aplicaciones web accesibles y con interfaces amigables, lo que facilita su uso por parte de personal no técnico. Esto contrasta con las soluciones de escritorio más antiguas y rígidas. Además, el uso de servicios en la nube para el almacenamiento de datos y la ejecución de lógica compleja, como Firebase o Amazon Web Services, se ha consolidado por su escalabilidad y facilidad de gestión. La programación entera (PE) y otras técnicas de optimización matemática están ganando terreno, demostrando ser herramientas poderosas para resolver problemas complejos de asignación de recursos, asegurando que se cumplan múltiples restricciones y se maximicen objetivos. Las tecnologías de desarrollo *frontend* como Angular y librerías de diseño como Bootstrap y Font Awesome también han avanzado, permitiendo la creación de interfaces de usuario más dinámicas, responsivas y atractivas, esenciales para la aceptación del usuario.

Soluciones Existentes y Comparación con Nuestra Propuesta: Existe una variedad de herramientas de software, tanto comerciales como de código abierto, diseñadas para facilitar la compleja tarea de la creación de horarios universitarios.

A. Software Comercial Hay empresas que ofrecen soluciones robustas con muchas funcionalidades:

- ***Celcat Timetabler:*** Es muy usado por universidades, conocido por su flexibilidad para crear horarios sin conflictos y manejar cambios, además de permitir la publicación y reserva de aulas [1]. También se menciona en investigaciones académicas como una herramienta comercial empleada [2].
- ***Infosilem:*** Una plataforma que ayuda con la programación de cursos y exámenes, verificando conflictos al instante y buscando integrarse con otros sistemas [4].
- ***Semestry (Termtime, Roomtime, Mytimetable):*** Ofrece una solución completa que abarca la planificación, construcción, entrega y operación de horarios, la gestión de aulas con reglas y perfiles personalizables, y una plataforma para la publicación de horarios [5].
- ***Asc Timetables*****:** Puede evaluar millones de combinaciones para dar horarios equilibrados. Permite ajustes manuales, verifica conflictos, facilita la importación de datos y ofrece acceso móvil a los horarios. Soporta la gestión de múltiples edificios y la programación de sustituciones de profesores, y cuenta con una versión *online* [7], [8], [9]. Es utilizado en numerosos países, adaptándose a requisitos regionales [7].
- ***Prime Timetable*****:** Diseñado para diversos niveles educativos, incluyendo universidades, este software funciona en múltiples dispositivos y permite tanto la creación automática como manual de horarios [10], [13]. Sus características incluyen la importación y

exportación de datos, múltiples vistas de horario, manejo de conflictos, consideración de preferencias, asignación de aulas y materias, gestión de descansos y sustituciones, y la capacidad de compartir, imprimir y publicar horarios *online* [10].

- ***Ofcourse*:** Es una solución basada en la nube, enfocada en la creación de horarios de cursos universitarios "casi perfectos". Optimiza los horarios considerando las preferencias de los docentes, los requisitos de los estudiantes y la asignación automática de aulas, tanto para clases presenciales como en línea [15].
- ***Cronos Timetable*:** Una herramienta basada en web que utiliza inteligencia artificial para la generación de horarios escolares y universitarios. Sus funcionalidades incluyen la eliminación de conflictos entre profesores y clases, el respeto a las restricciones de los docentes (como días libres), la minimización de "ventanas" (tiempos muertos) en los horarios de los profesores, y la configuración de diversos parámetros pedagógicos y logísticos [16].
- ***Otros*:** Existen otras herramientas como Coursedog, que ha reportado beneficios en universidades como la reducción de conflictos y ahorros económicos [17], y Arlo, más enfocado en la programación de cursos y capacitación [18]. Untis también es una solución mencionada en el ámbito de la programación académica [17].

B. *Software De Código Abierto (Open Source):* Las soluciones de código abierto ofrecen alternativas valiosas, a menudo con un fuerte respaldo comunitario y académico:

- ***Unitime:*** Es un sistema integral de programación educativa que soporta el desarrollo de horarios de cursos y exámenes, la gestión de cambios, el uso compartido de aulas con otros eventos y la asignación de estudiantes a clases individuales [19], [21]. Desarrollado inicialmente con la colaboración de universidades en Norteamérica y Europa, y ahora un

proyecto de la Fundación Apereo, ha sido implementado con éxito en instituciones grandes como Purdue University y el MIT [22]. UniTime es una aplicación cliente-servidor con interfaz web, independiente de la plataforma (implementada en Java J2EE y base de datos SQL), y permite desde la entrada manual de datos hasta la generación totalmente automatizada de horarios [19]. Su diseño distribuido facilita la descomposición del problema y la coordinación entre múltiples unidades organizativas [19].

- ***Fet (Free Evolutionary Timetabling):*** Este software libre, escrito en C++ con el *framework* Qt, ofrece un algoritmo de generación de horarios totalmente automático, aunque también permite la asignación semiautomática o manual [23]. Es multiplataforma y utiliza un formato XML modular y flexible para los datos de entrada, permitiendo la importación y exportación en formato CSV, y la generación de horarios en HTML, XML y CSV [23]. FET maneja una estructura de estudiantes flexible (años, grupos, subgrupos, e incluso estudiantes individuales) y una vasta paleta de restricciones de tiempo y espacio, cada una con un peso porcentual ajustable para priorizar su cumplimiento [23], [25]. Posee límites muy altos para el número de días, horas, profesores, estudiantes, materias, actividades, aulas y edificios que puede gestionar [25].
- ***Timefinder*:** Aunque actualmente no se mantiene activamente (sus desarrolladores sugieren migrar a FET o UniTime), TimeFinder fue un software libre escrito en Java para optimizar automáticamente horarios de escuelas y universidades [26]. Ofrecía una interfaz gráfica de usuario y la capacidad de importar y exportar datos en formatos XML y texto [26].

Nuestra propuesta se distingue por ser una aplicación web hecha a la medida para Unitrópico. A diferencia de las soluciones genéricas, la nuestra incorpora un algoritmo de programación entera en Python que nos permite manejar las prioridades de materias y optimizar el uso de recursos específicos de la universidad (aulas, laboratorios, áreas deportivas) de una forma que las soluciones manuales o muchos softwares estándar no pueden. La elección de Angular y Firebase nos da una plataforma moderna, escalable y más fácil de mantener, buscando un equilibrio entre personalización y eficiencia.

Brechas Y Limitaciones De Las Soluciones Actuales: A pesar de lo que ofrecen, las soluciones actuales suelen tener varias **brechas**:

- ***Poca Adaptación A Contextos Específicos*:** Pocos sistemas son lo suficientemente flexibles para integrar reglas y prioridades tan particulares como las de Unitrópico, incluyendo la gestión de áreas deportivas o las complejidades de su crecimiento.
- ***Optimización Limitada:*** Muchos sistemas solo asignan horarios sin usar técnicas avanzadas de optimización, lo que resulta en horarios que no son del todo óptimos o tienen conflictos que hay que arreglar manual. La programación entera, aunque conocida, no siempre se aplica a la medida de cada institución.
- ***Experiencia De Usuario*:** Algunos sistemas son muy técnicos y no priorizan una interfaz fácil de usar, lo que dificulta que el personal administrativo se adapte.

Nuestro proyecto busca precisamente llenar estas brechas, ofreciendo una solución que no solo automatice, sino que optimice inteligentemente y sea fácil de usar, mejorando el día a día de quienes hacen los horarios.

II. Asignación De Horarios Mediante Programación Entera (Pe) La Programación Entera (PE) es una técnica de optimización matemática que ha demostrado ser una herramienta poderosa y fundamental en la resolución del problema de asignación automática de horarios universitarios (UCTP).

A. Fundamentos Y Modelado Con Pe El UCTP es inherentemente un problema de optimización combinatoria NP-difícil, lo que lo hace un candidato natural para ser abordado con técnicas de PE [2]. La PE, y en particular la Programación Lineal Entera (PLE) y sus variantes como la Programación Lineal Entera Binaria (PLEB) o Mixta (PLEM), permite formular el problema de manera matemática precisa, buscando soluciones óptimas o con una brecha de optimalidad conocida [1].

Para crear un modelo de PE para horarios, se siguen pasos clave [1]:

- ***Recopilación De Datos:*** Juntar toda la información relevante de la universidad, como políticas, cursos, grupos, aulas, capacidades y preferencias de tiempo.
- ***Planificación Y Definición De Variables:*** Decidir cuáles serán las variables de decisión (a menudo binarias, indicando si una clase se asigna a un horario y aula específicos) y los parámetros del modelo [1].
- ***Formulación De La Función Objetivo*****:** Definir qué se quiere optimizar (ej. maximizar preferencias, minimizar costos asociados a la violación de restricciones blandas, o una combinación de múltiples objetivos) [1].
- ***Establecimiento De Restricciones:*** Traducir todas las reglas y requisitos del problema en ecuaciones o inecuaciones matemáticas.

B. Manejo De Restricciones Las reglas son cruciales y se dividen en dos tipos:

- ***Restricciones Duras (Hard Constraints):*** Son las que deben cumplirse sí o sí para que un horario sea válido [1]. Ejemplos comunes incluyen: un profesor, un grupo de estudiantes o un aula no pueden ser asignados a más de una actividad simultáneamente (no colisiones) [1]; todas las clases planificadas deben ser programadas [29]; las asignaturas deben ser impartidas por docentes con el perfil adecuado [29]; solo se deben programar asignaturas del plan de estudios vigente [29]; no se debe exceder la capacidad de las aulas [1]; se debe respetar la disponibilidad horaria de los docentes [29]; cumplimiento de la jornada escolar y límites de horas para docentes [29]; consideraciones sobre periodos consecutivos o bloques de clases [29].
- ***Restricciones Blandas (Soft Constraints):*** Son condiciones deseables que mejoran la calidad del horario, pero su violación no lo invalida, aunque usualmente incurre en penalizaciones en la función objetivo Ejemplos incluyen: preferencias de horarios de profesores y estudiantes, horarios compactos para los estudiantes, minimizando tiempos muertos [29]; estabilidad de aulas (que una misma asignatura se imparta siempre en la misma aula) [28]; minimizar el número de días de trabajo para los profesores o la fragmentación de sus horarios [28].

C. Solvers Y Herramientas De Pe Para resolver los modelos de PE, se necesita software especializado conocido como *solvers*:

- ***Solvers Comerciales:***
    - ***CPLEX (IBM):*** Es el *solver* de PE más frecuentemente utilizado en la literatura académica para el UCTP [1]. Se destaca por su robustez, velocidad, versatilidad y capacidad para manejar problemas complejos con un gran número de variables y

restricciones (incluso más de 100,000), produciendo consistentemente soluciones óptimas o casi óptimas (típicamente con menos del 1% de desviación) [1].

- ***GUROBI:*** Es el segundo *solver* comercial más popular, conocido por su robustez y eficiencia, a menudo superando a CPLEX en velocidad de procesamiento bruta y manteniendo una alta calidad de solución [1].
- ***Otros solvers y lenguajes de modelado*** como Lingo, AIMMS, GAMS, XPRESS y AMPL también han sido empleados en la investigación del UCTP [1].

- **Solvers De Código Abierto:**
  - ***C++ Glpk (Gnu Linear Programming Kit):*** Una opción de código abierto utilizada en algunas implementaciones [1].
  - ***Open Solver:*** Un complemento para Microsoft Excel, más accesible pero generalmente más lento y menos escalable para instancias grandes en comparación con los *solvers* comerciales [1].

Programación Por Restricciones (Cp):

- ***Google Or-Tools Cp-Sat*:** Aunque no es un *solver* de PE estrictamente, es una potente herramienta de Programación por Restricciones que ha visto aplicaciones recientes en la optimización de horarios universitarios [1].

D. Desafíos Y Estrategias En La Aplicación De Pe A pesar de su poder, la PE enfrenta desafíos, siendo el principal la escalabilidad [1]. A medida que aumenta el tamaño y la complejidad del problema (número de cursos, estudiantes, restricciones), el número de variables y restricciones en el modelo de PE puede crecer exponencialmente, haciendo que la obtención de una solución óptima sea computacionalmente intratable en un tiempo razonable.

Para mitigar estos desafíos, se han desarrollado diversas estrategias:

- ***Descomposición Del Problema*:** Dividir el problema general en subproblemas más pequeños y manejables, que se resuelven de manera secuencial o paralela. Esto puede ser por fases (como en el caso de la UANL, donde primero se asignan unidades de aprendizaje a grupos y horas, y luego docentes y aulas [29]) o por unidades organizativas (ej. por facultad)
- ***Simplificación Del Modelo*:** Reducir el número de variables o restricciones, por ejemplo, considerando inicialmente solo las variables más importantes o relajando temporalmente algunas restricciones [1].
- ***Hibridación Con Heurísticas/Metaheurísticas (Ilph/Milph):*** Utilizar heurísticas para generar soluciones iniciales de buena calidad que luego son refinadas por la PE, o para guiar el proceso de búsqueda del *solver* de PE, especialmente cuando los tiempos de cómputo para soluciones exactas son prohibitivos [1].
- ***Modelado Preciso Y Contextualización*:** El éxito de la aplicación de PE depende críticamente de una formulación matemática que sea a la vez precisa y eficiente, y que esté profundamente adaptada al contexto institucional específico, incluyendo sus restricciones y objetivos particulares [1].

E. Casos De Estudio Y Benchmarks La aplicación de la PE al UCTP ha sido validada en diversos contextos y mediante el uso de conjuntos de datos de prueba (*benchmarks*):

- ***TESIS UANL*:** Se desarrolló un modelo de PLE para la asignación de horarios en una universidad mexicana. Debido a problemas de escalabilidad con un modelo integral, se propuso una metodología de dos fases que demostró ser efectiva para instancias grandes, utilizando GAMS y CPLEX [29].

- ***Instancias De Udine*:** Estos son *benchmarks* conocidos en la comunidad de investigación de UCTP. Se ha reportado la resolución de las cuatro instancias originales a optimalidad probada por primera vez utilizando un enfoque de PE con CPLEX y una técnica de descomposición [28]. El mismo enfoque demostró ser competitivo en las instancias extendidas del ITC2007 [28].
- ***Universidad De La Salle (Uls):*** Se validó un modelo de PE con datos reales de esta universidad, demostrando mejoras en los horarios finales y en la interacción con los usuarios [27].
- ***Enseñanza Híbrida*:** Se ha propuesto un modelo de programación binaria multi-objetivo para abordar la programación de horarios universitarios que considera explícitamente la enseñanza híbrida (presencial y *online*), junto con elecciones de los estudiantes, diversidad de espacios y tiempos de traslado [34].
- ***Cursos Multi-Sección Y Preferencias:*** Investigaciones han desarrollado modelos de PE para manejar simultáneamente cursos con múltiples secciones, la estabilidad de las aulas (asignar un curso consistentemente a la misma aula) y las preferencias de los docentes, logrando resolver problemas a gran escala mediante la extensión de enfoques previos para reducir el espacio de solución [31].

F. Tendencias Emergentes Y Futuro El campo de la asignación de horarios con PE continúa evolucionando:

- ***Integración De Pe Con Aprendizaje Automático (Ml)*:** Esta es una tendencia creciente y prometedora. El ML puede utilizarse para mejorar la eficiencia y la calidad de las soluciones de PE en diversas etapas: selección de instancias y predicción del mejor algoritmo/*solver*, ajuste de parámetros del modelo de PE, predicción de factibilidad o

sugerencia de relajación de restricciones, reducción del espacio de búsqueda para la PE, y posprocesamiento o refinamiento de las soluciones generadas por la PE [1].

- ***Alta Tasa De Implementación*:** Una revisión técnica que cubrió investigaciones de 1990 a 2022 encontró que el 98% de los modelos basados en PE reportaron implementaciones, una tasa significativamente mayor que el 34% encontrado para los algoritmos de metaheurísticas en otra revisión existente [35]. Esto sugiere que la PE, a pesar de su complejidad, es un enfoque que a menudo se lleva a una etapa práctica y demostrable en la investigación académica.
- ***Investigación Futura*:** Se identifican áreas para trabajos futuros, como un mayor énfasis en el uso de algoritmos de PE en otras áreas industriales, una investigación más profunda sobre el uso de modelos de ML para la programación de horarios universitarios para permitir soluciones más basadas en datos, y una comparación más exhaustiva de los enfoques híbridos de PE y ML [1].

Iii. Comparación Y Sinergias: Pe Vs. Otras Técnicas Si bien la PE es una técnica poderosa, es importante considerarla en el contexto de otros enfoques de optimización.

Pe Vs. Metaheurísticas: La PE está diseñada para encontrar soluciones óptimas (o probar la optimalidad), mientras que las metaheurísticas (como algoritmos genéticos, recocido simulado, búsqueda tabú) generalmente buscan encontrar soluciones de muy buena calidad en un tiempo computacional razonable, especialmente para problemas muy grandes donde encontrar la optimalidad exacta es inviable [1]. Como se mencionó, los estudios académicos han reportado una mayor tasa de implementación de soluciones basadas en PE en comparación con las metaheurísticas [1]. Las metaheurísticas pueden ofrecer mayor flexibilidad para manejar ciertos

tipos de restricciones complejas o no lineales que son difíciles de modelar con PE estándar, y pueden ser más rápidas en la exploración de grandes espacios de solución [1].

Pe Vs. Programación Por Restricciones (Cp): La Programación por Restricciones es otro paradigma declarativo para resolver problemas combinatorios. Herramientas como Google OR-Tools CP-SAT son ejemplos de *solvers* de CP potentes que han comenzado a aplicarse al UCTP [1]. Algunas soluciones exitosas, como el sistema UniTime desarrollado en la Universidad de Purdue, se basan en modelos de satisfacción de restricciones y utilizan algoritmos de búsqueda específicos como Iterative Forward Search (IFS) con estadísticas basadas en conflictos y *Branch and Bound*, en lugar de una formulación de PE pura [37]. Estos enfoques también manejan restricciones duras y blandas y han demostrado ser efectivos para problemas universitarios complejos y a gran escala [37]. La hibridación de PE con estas otras técnicas (metaheurísticas, CP, ML) es una dirección de investigación activa y prometedora para aprovechar las fortalezas de cada enfoque.

Conclusión Del Estado Del Arte*:* El estado del arte en la asignación automática de horarios universitarios revela un panorama dinámico y en constante evolución. Por un lado, existe una oferta considerable de software, tanto comercial como de código abierto, que proporciona herramientas cada vez más sofisticadas para ayudar a las instituciones en esta tarea crítica. Estas soluciones varían en sus características, tecnologías subyacentes y enfoques de implementación. Por otro lado, la programación entera se ha consolidado como una metodología matemática fundamental y altamente efectiva para modelar y resolver el UCTP. A pesar de los inherentes desafíos de escalabilidad, los avances en los *solvers* de PE, junto con estrategias de modelado inteligente como la descomposición y la hibridación con otras técnicas (especialmente el aprendizaje automático), continúan ampliando las fronteras de lo que es posible lograr. La alta

tasa de implementación de modelos de PE en la investigación académica subraya su viabilidad y relevancia práctica. El futuro probablemente verá una convergencia aún mayor de estas dos vertientes, con software comercial y de código abierto incorporando algoritmos de optimización más avanzados basados en PE y sus hibridaciones, y con la investigación académica enfocándose en hacer estos métodos aún más potentes, flexibles y adaptables a las crecientes complejidades de las instituciones educativas modernas.

OBJETIVOS

Objetivo General

Desarrollar un sistema web para la generación automática de horarios de clase en la Universidad Internacional del Trópico Americano (Unitrópico), que permita gestionar sus áreas académicas, áreas deportivas y la infraestructura física.

Objetivos Específicos

1. **Analizar** los requerimientos funcionales y no funcionales del sistema web propuesto para generación de horarios de clase, asignaciones de docentes, aulas, laboratorios y áreas deportivas.
2. **Diseñar** la arquitectura del sistema web propuesto, incluyendo el modelo de datos (base de datos no relacional), y adaptable (frontend), y la lógica de negocio que integre el algoritmo de optimización para la asignación de horarios.
3. **Desarrollar** los módulos del sistema web, implementando las funcionalidades mediante Angular 18 y Firebase 9 y programación en Python para garantizar su funcionamiento.

MARCO REFERENCIAL

Vamos a sentar las bases teóricas, conceptuales, geográficas, demográficas y legales que nos permiten entender a fondo el problema de los horarios en Unitrópico y el porqué de nuestra solución nos ayudará a tener un lenguaje común y a asegurarnos de que lo que proponemos tiene un respaldo sólido.

6.1 MARCO TEÓRICO

Las ideas principales para entender el problema de la asignación de horarios y cómo nuestra solución tecnológica puede encajar. Es la teoría la que nos permite entender lo que estamos haciendo, desde las preguntas que nos hacemos hasta cómo vamos a interpretar los resultados. Nos ayuda a no perder el rumbo y mantenernos enfocados en resolver el problema central.

6.1.1 La Ingeniería De Sistemas Y Su Aplicación En La Optimización De Recursos Educativos

La Ingeniería de Sistemas es una disciplina que busca diseñar, implementar y gestionar sistemas complejos, y en el contexto de la educación superior, su aplicación es crucial para optimizar la administración de recursos. Esto incluye desde la infraestructura física (aulas, laboratorios especializados, áreas deportivas) hasta el capital humano (docentes y estudiantes) y la gestión de procesos académicos como la programación de asignaturas. Los principios de la ingeniería de sistemas nos permiten abordar el problema de los horarios no solo como un rompecabezas, sino como un sistema que puede ser mejorado y optimizado. En esencia, aplicamos el pensamiento sistémico para transformar una tarea que consume mucho tiempo y genera frustración en un proceso eficiente que beneficia a toda la comunidad universitaria de Unitrópico.

6.1.2 Sistemas De Gestión Académica Y Planificación De Horarios Universitarios

Los Sistemas de Gestión Académica (SGA) son herramientas tecnológicas diseñadas para automatizar y mejorar la administración de las operaciones en instituciones educativas. Dentro de estos, los módulos de planificación de horarios son esenciales. Estos sistemas buscan simplificar tareas como la matriculación de estudiantes, la asignación eficiente de profesores y aulas, y la gestión de la carga académica. Su teoría se basa en la idea de que una buena organización optimiza el uso de recursos y mejora la experiencia educativa. Un SGA bien diseñado, específicamente en el contexto de la planificación de horarios universitarios, no solo maneja datos, sino que busca crear un entorno más fluido para que estudiantes y docentes se enfoquen en lo importante: el aprendizaje y la enseñanza, minimizando las interrupciones y los conflictos que la gestión manual genera.

6.1.3 Programación Matemática Y Optimización Aplicada A Horarios

La programación matemática es una rama de la matemática aplicada que se utiliza para resolver problemas de optimización, es decir, encontrar la mejor manera de hacer algo cuando se tienen múltiples opciones y restricciones. Dentro de ella, la programación entera (PE) es particularmente relevante para problemas donde las decisiones deben ser tomadas en unidades completas (por ejemplo, asignar un aula completa a una clase, no una fracción de ella). La teoría detrás de la PE nos permite modelar las complejas restricciones y preferencias de un horario universitario (como que un profesor no puede estar en dos sitios a la vez, que ciertas materias tienen prioridad, o que hay que minimizar los tiempos muertos de los estudiantes) y buscar la mejor solución posible, incluso si es un rompecabezas con miles de piezas que sería imposible de calcular a mano. Esta técnica es el "cerebro" detrás de la optimización que buscamos lograr.

6.2 MARCO CONCEPTUAL

6.2.1 Términos Específicos De La Asignación De Horarios

- ***Asignación De Horarios:*** Es el proceso de organizar y distribuir de manera lógica los cursos, estudiantes, profesores y los recursos físicos (como aulas o laboratorios) dentro de un periodo de tiempo determinado, siempre buscando cumplir con una serie de reglas y necesidades.
- ***Conflictos De Horarios*****:** Son las situaciones no deseadas donde hay superposiciones o choques en la asignación de recursos. Por ejemplo, cuando un mismo profesor es asignado a dos clases al mismo tiempo, o dos clases diferentes intentan usar la misma aula en el mismo momento.
- ***Prioridades Curriculares*****:** Se refiere a las reglas o preferencias específicas que tiene la universidad para la asignación de ciertas materias o grupos. Estas prioridades pueden venir de los planes de estudio, de la importancia de ciertas asignaturas, o de decisiones administrativas que deben reflejarse en los horarios.
- ***Recursos Académicos*****:** Son todos los espacios físicos y materiales que la universidad pone a disposición para la enseñanza. En nuestro caso, esto incluye aulas generales, laboratorios especializados y también las áreas deportivas, que a menudo se olvidan en los sistemas de horarios pero que son cruciales para una planificación integral.
- ***Interfaz De Usuario (Ui):*** Es la parte visual y con la que interactúa una persona al usar el software. Son las pantallas, los botones, los menús y los formularios que un usuario ve y maneja para introducir datos y ver los resultados de los horarios.
- ***Lógica De Negocio*****:** Son todas las reglas y los procesos internos que el sistema debe seguir para funcionar correctamente. Es cómo el software "piensa" y toma decisiones, por

ejemplo, cómo valida que un horario es correcto, cómo se calcula la mejor asignación de una materia o cómo se gestionan los roles de usuario.

6.2.2 Conceptos Tecnológicos Clave Del Proyecto

- ***Angular 18*:** El "motor" del *frontend*. Es un *framework* que nos ayuda a construir toda la parte visible de la aplicación web, haciendo que sea dinámica, interactiva y reaccione a lo que el usuario hace en su navegador.
- ***Firebase 9:*** Es como nuestro "cerebro y almacenamiento en la nube". Es una plataforma de Google que nos da una base de datos flexible (Firestore) para guardar toda la información, permite que los usuarios inicien sesión de forma segura y nos deja ejecutar código complejo (nuestro algoritmo de Python) directamente en sus servidores, sin que tengamos que preocuparnos por la infraestructura.
- ***Python:*** El lenguaje de programación que usamos para la "inteligencia" detrás del sistema. Es especialmente bueno para el algoritmo de optimización, ya que nos permite modelar problemas complejos de manera clara y eficiente.
- ***Programación Entera (Pe):*** La técnica matemática avanzada que utiliza Python para encontrar la mejor solución posible a nuestros problemas de horarios, asegurándose de que todas las variables (como la asignación de un salón o un profesor) sean números completos y no fracciones, lo que es esencial para la realidad de los horarios.
- ***Bootstrap 5:*** Una "caja de herramientas de diseño" que usamos para que la aplicación se vea bien y se adapte a cualquier tamaño de pantalla (computador, tablet, celular), de forma rápida y profesional.

- ***Font Awesome*:** Es una "biblioteca de íconos". Son esos pequeños dibujos (como un lápiz, un ojo, una casa) que usamos para hacer que la interfaz sea más intuitiva y atractiva visualmente, facilitando la navegación y comprensión del usuario.
- ***Cloud Functions (Firebase):*** Es una parte de Firebase que nos permite ejecutar nuestro código de Python (el algoritmo de optimización) en los servidores de Google de forma automática y bajo demanda, cada vez que lo necesitamos, sin que tengamos que gestionar un servidor propio.

6.3 MARCO LEGAL

El marco legal es como el conjunto de reglas del juego que nuestro sistema debe respetar. Establece las normativas y regulaciones que rigen la operación de las instituciones de educación superior en Colombia y, por ende, deben ser consideradas en la planificación y generación de horarios. Aunque nuestro sistema es una herramienta tecnológica, su diseño y funcionamiento deben ser coherentes con el marco normativo colombiano y, muy importante, con los reglamentos internos de Unitrópico. Esto incluye:

Ley 30 De 1992 (Ley General De Educación Superior En Colombia): Es la ley fundamental que marca las pautas generales para la organización y el funcionamiento de la educación superior en el país. Nuestro sistema debe encajar dentro de los principios que esta ley establece.

***Reglamentos Académicos Internos De Unitrópico*:** Son las normas específicas de la universidad que dictan cómo se inscriben las asignaturas, cómo se maneja la carga académica de los docentes (máximo de horas, mínimos, etc.), la duración de los periodos de clase, y cómo se usan los espacios educativos. Estas reglas internas son cruciales para el algoritmo de optimización, ya que definen muchas de las restricciones duras y blandas.

***Políticas De Privacidad Y Protección De Datos*:** Si bien el sistema no maneja datos sensibles de estudiantes a gran escala, al gestionar información de docentes y la institución (horarios, disponibilidad), el diseño debe ser consistente con la Ley 1581 de 2012 (Ley de Protección de Datos Personales en Colombia) y sus decretos reglamentarios. Esto asegura que la información se maneje con la debida privacidad y seguridad. Es crucial que el sistema, al generar horarios, no solo sea eficiente y optimizado, sino que también garantice que todas las asignaciones sean válidas y cumplan con las normativas oficiales y la misión institucional.

METODOLOGÍA

La metodología adoptada ha sido necesariamente adaptativa, reconociendo que el desarrollo de un sistema complejo como HERMES, realizado individualmente, ha requerido una gestión flexible del tiempo y los recursos a lo largo de un periodo que se extendió más allá de la planificación inicial de 10 meses, abarcando aproximadamente desde septiembre de 2022 hasta la fecha actual (mayo de 2025), y considerando que el desarrollo continúa en sus fases finales.

7.1 ENFOQUE METODOLÓGICO

El proyecto HERMES se abordó bajo un enfoque metodológico mixto que combina elementos, con un fuerte componente de investigación aplicada de tipo constructivo/desarrollo tecnológico.

Para el ciclo de vida del desarrollo de software, se adoptó un modelo adaptativo, inspirado en los principios de las metodologías ágiles. Aunque el trabajo fue individual, se tomaron elementos de:

- ***Programación Extrema (Xp):*** Se adoptaron principios como la búsqueda de la simplicidad en el diseño, la planificación iterativa (reflejada en la evolución del proyecto por fases y la entrega de funcionalidades incrementales), la refactorización del código

para mantener su calidad a lo largo del tiempo, y la realización de pruebas continuas por parte del desarrollador.

- Este enfoque permitió la flexibilidad necesaria para manejar los desafíos técnicos

POBLACIÓN Y MUESTRA

Población Objeto De Estudio: Personal directivo, administrativo y docente de la Universidad Internacional del Trópico Americano (Unitrópico) involucrado o afectado por el proceso de asignación de horarios.

Fuentes De Información Para Requisitos Y Validación Conceptual:

- ***Aportes Conceptuales Específicos:*** La información clave proporcionada por el Ingeniero Alfonso Tovar, a través de un diagrama explicativo manual ("esquema realizado por el ingeniero alfonso.pdf") y comentarios verbales clarificatorios (audio), fue fundamental en las etapas iniciales para configurar la visión del problema y la estructuración de datos desde una perspectiva institucional. Estos insumos, aunque no producto de entrevistas formales, representaron una valiosa contribución de conocimiento experto del contexto de Unitrópico.

7.4 MÉTODOS DE INVESTIGACIÓN Y DESARROLLO

7.4.1. Métodos De Recolección De Datos Primarios E Información Clave:

Análisis Documental (Fundamental) *Diagrama Y Audio Explicativo***:** EL ARCHIVO "ESQUEMA REALIZADO POR EL INGENIERO ALFONSO.PDF" Y EL AUDIO ASOCIADO DEL INGENIERO TOVAR (ING. ALFONSO) FUERON ANALIZADOS COMO FUENTES DIRECTAS DE 3INFORMACIÓN CONCEPTUAL SOBRE LA ESTRUCTURACIÓN DE DATOS Y LA LÓGICA DE ASIGNACIÓN DESEADA EN UNITRÓPICO.

- Literatura científica y técnica relevante sobre UCTP, algoritmos de optimización, Google OR-Tools, y las tecnologías de desarrollo.

- ***Análisis De Requisitos*:** Traducción de las necesidades identificadas en Unitrópico (a partir de los documentos iniciales y los aportes del Ing. Tovar) en especificaciones técnicas para HERMES.
- ***Ciclo De Desarrollo De Software:*** (Como se describió anteriormente: Diseño Arquitectónico, Diseño de Base de Datos, Desarrollo Frontend, Desarrollo Backend).

RESULTADOS

El objetivo general del presente trabajo de grado fue "Desarrollar una aplicación web, denominada HERMES, para la generación automática y optimizada de horarios de clase de las asignaturas impartidas en los diferentes programas académicos de la Universidad Internacional del Trópico Americano (Unitrópico)". Para alcanzar este objetivo, se siguió una metodología de desarrollo tecnológico adaptativa e incremental, que abarcó desde la investigación inicial y el diseño detallado hasta la implementación iterativa –y en varias ocasiones, la reingeniería y reinicio de componentes fundamentales– del sistema y el algoritmo de asignación preciso. Este proceso se extendió a lo largo de aproximadamente 32 meses, desde septiembre de 2022 hasta la fecha actual.

Los resultados que se presentan a continuación demuestran el esfuerzo por cumplir los objetivos específicos del proyecto. La aplicación web HERMES, con su arquitectura frontend-backend actual, la base de datos estructurada en Cloud Firestore, constituye el principal entregable. Cada componente desarrollado es producto de un aprendizaje continuo y la superación de obstáculos técnicos, buscando siempre un diseño limpio y lo más asequible posible para el usuario final.

8.2 Recopilación De Datos Para El Sistema Hermes

La "recopilación de datos" para el proyecto HERMES se centró en la obtención y estructuración de la información necesaria para modelar el problema de asignación de horarios de Unitrópico y para alimentar el sistema desarrollado. Este proceso se basó en:

1. ***Análisis Documental:*** Los documentos de la propuesta inicial ("propuesta trabajo de grado.docx (1).docx", "horario y salones.docx") y el esquema conceptual del Ing. Tovar (Ing. Alfonso en "esquema realizado por el ingeniero alfonso.pdf") junto con su audio explicativo, fueron cruciales para la definición inicial del problema y los requisitos. La revisión de literatura científica (Estado del Arte) guió la selección de enfoques tecnológicos.
2. ***Especificación De Entidades Y Restricciones:*** A partir del análisis documental y los aportes conceptuales, se identificaron las entidades clave del sistema y las principales restricciones.

8.3 Presentación De Resultados Del Desarrollo De Hermes

Los principales resultados del proyecto se materializan en la aplicación web HERMES y sus componentes tecnológicos. A continuación, se describen los hallazgos y productos más importantes, reflejando el estado actual del desarrollo.

8.3.1 Aplicación Web Hermes: Arquitectura Y Tecnologías Implementadas

Se desarrolló la aplicación web HERMES. La arquitectura y la interfaz de usuario se concibieron bajo una filosofía de **diseño limpio y con el objetivo de ser lo más asequible e intuitiva posible** para los usuarios administradores.

- **Arquitectura General:** El sistema sigue una arquitectura moderna:

- **Frontend (Cliente):** SPA con **Angular v17.3.0** y TypeScript v5.3.2.
- **Backend (Servidor):** Firebase Cloud Functions con Python v3.10 y Flask v3.0.3.
- **Base de Datos:** Cloud Firestore.
- **Autenticación:** Firebase Authentication.

8.3.2 Modelo De Datos Implementado En Cloud Firestore

Se diseñó e implementó un modelo de datos NoSQL en Cloud Firestore. Las principales colecciones son: usuarios, facultades, carreras, asignaturas, docentes, bloques, salones, laboratorios, áreas deportivas, horarios generados (con subcolección asignaciones clase), y configuraciones algoritmo. Este modelo fue refinado a través de las diferentes etapas y reinicios del proyecto para adaptarse mejor a la lógica de la aplicación y del algoritmo.

8.3.3 Módulos Y Componentes Desarrollados En El Frontend (Angular)

El frontend de HERMES se estructuró en módulos funcionales. La actual estructura de módulos y componentes es el resultado de un proceso de desarrollo evolutivo, que incluyó la migración y reingeniería de elementos HTML y CSS en varias etapas para asegurar la funcionalidad y coherencia visual, mientras se reconstruía la lógica en TypeScript sobre bases más sólidas después de cada reinicio del proyecto**.** Los principales módulos son:

- ***Módulo De Autenticación (Auth)*****:** Login y registro.
- ***Módulos De Gestión De Datos (En Src/App/Pages/)*****:** CRUDs para Facultades, Carreras, Asignaturas, Docentes (con gestión de disponibilidad), Bloques, Salones, Laboratorios, Áreas Deportivas.

Módulo De Horarios (Src/App/Pages/Horarios/):

- ***Componente De Generación*:** Permite configurar e iniciar la generación de horarios.
- ***Componente De Visualización De Horarios Asignados*:** Esta funcionalidad se encuentra actualmente en desarrollo intensivo y constituye una de las próximas funcionalidades a completar.
- ***Componentes Compartidos (Shared)*:** Navbar, footer.
- ***Rutas*:** Definidas en src/app/app.routes.ts.

8.3.4 Backend Api Y Lógica De Negocio (Firebase Cloud Functions - Python) El backend en functions/main.py expone una API RESTful con Flask y contiene el algoritmo:

- ***Api Endpoints*:** POST /api/v1/horarios/generar (principal), y endpoints GET para datos de apoyo (/asignaturas, /docentes, etc.).
- ***Lógica De Negocio*:** Funciones para conectar con Firestore, procesar datos para OR-Tools, manejar solicitudes HTTP.

8.3.5 Implementación Y Funcionamiento Del Algoritmo De Generación De Horarios El algoritmo en functions/main.py usa CP-SAT de Google OR-Tools.

1. Recolección y Preparación de Datos: Desde Firestore.
2. Creación del Modelo CP-SAT.
3. Definición de Variables de Decisión Booleanas**.**
4. Establecimiento de Restricciones Duras: Unicidad, no conflictos de docentes/espacios, capacidad, requisitos de aula/laboratorio, disponibilidad docente, carga horaria.
5. Restricciones Blandas: Enfoque inicial en factibilidad; la optimización avanzada de blandas es área de mejora.
6. Función Objetivo: Actualmente implícita (encontrar solución factible).
7. Ejecución del Solver: Con límite de tiempo (540s).

8. Procesamiento y Almacenamiento de la Solución: En Firestore si es OPTIMAL o FEASIBLE. La complejidad de esta implementación fue uno de los factores que contribuyó a los ciclos de reingeniería del proyecto.

8.3.6 Estado Actual de la Visualización de Horarios Asignados

Como se ha indicado, la interfaz de usuario para la visualización detallada e interactiva de los horarios generados por el algoritmo (por ejemplo, en formato de calendario o tabla filtrable por docente, carrera, salón) se encuentra actualmente en fase de desarrollo y constituye una de las próximas funcionalidades a completar en el **sistema HERMES.** Los datos generados por el algoritmo se almacenan correctamente en Firestore, y el frontend cuenta con la estructura (modelos y servicios en src/app/pages/horarios/) para consumir y presentar esta información una vez que los componentes de visualización estén finalizados.

8.4 Análisis De Resultados

Los resultados actuales, producto de un extenso y desafiante proceso de desarrollo, demuestran que el sistema HERMES ha alcanzado un estado funcional robusto en sus componentes centrales.

- ***Cumplimiento Del Objetivo De Gestión De Datos*:** Los módulos CRUD permiten una gestión estructurada de la información base, superando el manejo manual previo.
- ***Cumplimiento Del Objetivo De Generación Automática (Núcleo)*:** El algoritmo genera horarios factibles que respetan las restricciones duras, lo cual es el avance fundamental. El proceso de desarrollo, marcado por varios ciclos de reinicio y refactorización de componentes, permitió una comprensión más profunda de los desafíos técnicos y condujo al diseño actual de los módulos y la interacción con el backend.

- ***Arquitectura Sólida*:** La arquitectura Angular + Firebase (Python) es moderna y escalable. La filosofía de diseño limpio y asequible se ha mantenido como guía.
- ***Eficiencia Potencial*:** La generación algorítmica es significativamente más rápida (minutos) que el proceso manual (semanas).
- ***Reducción De Conflictos*:** El algoritmo, por diseño, evita conflictos definidos por las restricciones duras.

La relevancia de estos resultados radica en que HERMES, a pesar de los desafíos y reinicios en su desarrollo, se perfila como una aplicación web funcional con un motor capaz de abordar el núcleo del problema de asignación de horarios. La finalización de la visualización de horarios consolidará su utilidad.

8.5 Comparación Con Literatura Previa

Al comparar HERMES con los enfoques del Estado del Arte (Capítulo 4):

- **Similitudes:** Utiliza programación por restricciones/entera (CP-SAT) y una arquitectura web, alineándose con tendencias modernas [1], [11], [19].

Diferencias Y Valor Agregado:

- ***Adaptación Específica Y Resiliencia En El Desarrollo*:** HERMES es una solución adaptada a Unitrópico, cuyo desarrollo superó múltiples fallas totales, demostrando un proceso de aprendizaje y reingeniería significativo, un aspecto práctico no siempre detallado en estudios teóricos.

- ***Pila Tecnológica Integrada Y Accesible*:** La combinación Angular + Firebase (Python/Flask) + OR-Tools es una implementación práctica con herramientas modernas y accesibles.
- ***Enfoque En La Funcionalidad Operativa*:** A pesar de los reinicios, el enfoque se mantuvo en lograr un sistema usable para la gestión de datos y la generación de horarios, más allá de la pura exploración algorítmica.

8.6 DISCUSIÓN

El desarrollo de HERMES ha sido un proceso complejo y de gran aprendizaje, con resultados significativos y lecciones importantes.

- **Relevancia E Implicaciones Prácticas:** HERMES tiene el potencial de optimizar drásticamente la creación de horarios en Unitrópico, ahorrando tiempo, mejorando el uso de recursos y reduciendo conflictos. Su diseño busca ser lo más asequible posible para facilitar su adopción.
- **Implicaciones Teóricas:** El proyecto válido la aplicación de CP-SAT de OR-Tools en un contexto real, integrándose en una arquitectura serverless, y subraya la importancia de la resiliencia en proyectos de desarrollo de software complejos.

Limitaciones Del Estudio Y Desarrollo Actual:

- ***Proceso De Desarrollo Extendido Y Reinicios*:** El desarrollo individual se caracterizó por un aprendizaje intensivo y la superación de desafíos técnicos que llevaron a cuatro reinicios importantes del proyecto. Estas reestructuraciones, si bien necesarias para abordar fallas

fundamentales y reconstruir la funcionalidad (migrando componentes HTML/CSS y lógica TypeScript) sobre bases más sólidas, extendieron el cronograma y pudieron haber limitado la exploración de un abanico más amplio de restricciones blandas o funcionalidades avanzadas en la versión actual.

- ***Visualización De Horarios En Desarrollo*:** Como se mencionó, la interfaz gráfica detallada para la visualización e interacción con los horarios generados aún no está completamente implementada y es un paso crítico pendiente.

Pruebas Exhaustivas Con Datos Completos De Unitrópico: La validación final requiere pruebas con el conjunto completo de datos de Unitrópico en un entorno piloto.

- ***Optimización De Restricciones Blandas*:** El foco ha estado en la factibilidad (restricciones duras). La optimización avanzada de múltiples restricciones blandas es un área clara para el trabajo futuro.
- ***Gestión De Cambios Dinámicos*:** No cubre reajustes en tiempo real una vez el horario está en curso.

Recomendaciones Para Futuros Trabajos Y Desarrollo En Unitrópico:

0. ***Priorizar Y Finalizar El Módulo De Visualización De Horarios*:** Es esencial para la usabilidad del sistema.
1. ***Estabilización Y Pruebas Piloto*:** Dada la historia de reinicios, una fase de estabilización del código actual y pruebas piloto exhaustivas en Unitrópico son cruciales.
2. ***Documentar Lecciones Del Desarrollo*:** Capitalizar el aprendizaje de los desafíos y reinicios para futuras mejoras y proyectos.

3. ***Expandir Restricciones Blandas*:** Incorporar gradualmente más criterios de calidad en el algoritmo.
4. ***Capacitación Y Adopción*:** Planificar la capacitación del personal de Unitrópico.

En conclusión, HERMES representa un avance significativo hacia la solución del problema de asignación de horarios en Unitrópico.

IMPACTOS

Este capítulo aborda el alcance del proyecto de grado HERMES y los efectos benéficos que se proyectan a partir de sus resultados. El desarrollo de la aplicación web HERMES para la generación automática de horarios en la Universidad Internacional del Trópico Americano (Unitrópico) no solo representa la consecución de un producto tecnológico, sino que también tiene el potencial de generar impactos significativos en diversas dimensiones de la vida institucional. Estos impactos van más allá de los resultados inmediatos del software y se refieren a cómo la implementación y uso de HERMES pueden influir positivamente en diferentes áreas, grupos de personas y en el entorno operativo de la universidad.

9.1 Impacto Tecnológico Y Científico

Modernización De Procesos Institucionales: HERMES introduce una solución tecnológica avanzada para un proceso crítico que se realizaba manualmente. Esto implica una modernización significativa en la gestión académica de Unitrópico, reemplazando métodos tradicionales y propensos a errores por un sistema digitalizado, automatizado y basado en optimización. Representa la adopción de la transformación digital en un área clave de la administración universitaria.

Adopción Y Validación De Tecnologías Modernas: El proyecto implementa y valida un conjunto de tecnologías contemporáneas en el contexto específico de Unitrópico:

- ***Frontend*:** Angular (v17.3.0) para una interfaz de usuario reactiva y moderna (SPA).
- ***Backend*:** Python (v3.10) sobre una arquitectura serverless con Firebase Cloud Functions, utilizando Flask (v3.0.3) para la API.
- ***Base De Datos*:** Cloud Firestore (NoSQL), que ofrece flexibilidad y escalabilidad.
- ***Optimización*:** Google OR-Tools (v9.9.3992), específicamente el solver CP-SAT, una herramienta de vanguardia para la resolución de problemas de optimización combinatoria. La aplicación exitosa de esta pila tecnológica demuestra su viabilidad y beneficios para resolver problemas complejos en instituciones educativas.
- ***Base Para Futuros Desarrollos Tecnológicos*:** La arquitectura modular de HERMES, la base de datos centralizada y la API desarrollada pueden servir como cimiento para futuras expansiones, como la integración con sistemas de gestión de aprendizaje (LMS), sistemas de inscripción estudiantil, o el desarrollo de módulos analíticos para la planificación académica a largo plazo.
- ***Contribución Al Conocimiento Aplicado (Impacto Científico)*:** Si bien el proyecto no busca generar teoría fundamental nueva, la aplicación y adaptación del solver CP-SAT de Google OR-Tools para el problema específico de asignación de horarios de Unitrópico, con su conjunto particular de entidades y restricciones, constituye un caso de estudio valioso. La documentación del proceso de modelado, los desafíos enfrentados (incluyendo los reinicios del proyecto y la migración de componentes) y las soluciones implementadas pueden aportar al conocimiento práctico en el campo de la ingeniería de software y la investigación de operaciones aplicada a contextos universitarios reales, especialmente en instituciones latinoamericanas con características similares.

9.2 Impacto Organizacional, Administrativo Y Económico

- ***Optimización Del Tiempo Y Los Recursos Humanos*:** Se espera una reducción drástica del tiempo que el personal administrativo y directivo de Unitrópico dedica a la elaboración manual de horarios (pasando de semanas a un proceso que puede completarse en horas o días con el software). Este tiempo liberado puede reorientarse hacia actividades de mayor valor estratégico y académico.
- ***Mejora En La Eficiencia Operativa Y La Toma De Decisiones*:** HERMES facilita un proceso de asignación de horarios más ágil y menos susceptible a errores humanos. Permite a los administradores generar y evaluar diferentes escenarios de horarios, lo que puede conducir a una toma de decisiones más informada sobre la asignación de recursos, la planificación de la oferta académica y la gestión de la capacidad institucional.
- ***Reducción De Errores Y Conflictos Horarios*:** La automatización y la validación sistemática de restricciones por parte del algoritmo disminuyen significativamente la probabilidad de conflictos de solapamiento (docentes, aulas) y otros errores comunes en la asignación manual, reduciendo la necesidad de correcciones y ajustes posteriores que consumen tiempo y generan inconvenientes.
- ***Transparencia Y Estandarización Del Proceso*:** El sistema promueve un proceso más estandarizado y transparente para la asignación de horarios, basado en reglas y criterios definidos y configurables, lo que puede mejorar la percepción de equidad en la distribución de la carga docente y el uso de los espacios.
- ***Optimización Del Uso De La Infraestructura Física (Impacto Económico Indirecto)*:** Al considerar sistemáticamente la capacidad, el tipo y el equipamiento de salones y laboratorios, HERMES puede llevar a una utilización más eficiente de estos recursos

físicos. Un mejor aprovechamiento de la infraestructura existente puede optimizar la inversión realizada y, potencialmente, ayudar a identificar con mayor precisión las necesidades reales de nuevos espacios o a posponer inversiones si se demuestra que los recursos actuales pueden ser gestionados más eficazmente.

- ***Soporte Al Crecimiento Institucional Eficiente*:** Con el aumento de la matrícula estudiantil y la posible expansión de la oferta académica tras su transformación a universidad pública, Unitrópico requiere herramientas que le permitan gestionar esta complejidad de manera eficiente. HERMES está diseñado para ser escalable y puede adaptarse a un mayor volumen de datos y restricciones, apoyando el crecimiento sostenible de la institución.

9.3 IMPACTO SOCIAL Y ACADÉMICO (EN LA COMUNIDAD UNITRÓPICO)

- **Mejora De La Experiencia Educativa Para Estudiantes:** Aunque la visualización de horarios para estudiantes no es una funcionalidad directa de la versión actual de HERMES (que se enfoca en la administración), la generación de horarios más estables, coherentes, con menos conflictos y que aseguren la disponibilidad de los espacios adecuados, indirectamente beneficia a los estudiantes al facilitar su planificación académica y reducir la incertidumbre y los problemas logísticos al inicio de cada semestre.
- **Mejora De Las Condiciones De Trabajo Para Docentes:** HERMES permite considerar la disponibilidad horaria de los docentes y, en futuras iteraciones, podría incorporar de manera más explícita sus preferencias. La reducción de conflictos y la asignación a aulas adecuadas también mejora su entorno de enseñanza. Se reduce la probabilidad de sobrecargas no planificadas o asignaciones inconvenientes.

- **Reducción Del Estrés Y La Incertidumbre Institucional:** Un proceso de creación de horarios más rápido, predecible y eficiente disminuye el nivel de estrés y la incertidumbre que suele acompañar esta tarea al inicio de cada periodo académico, tanto para el personal administrativo como para docentes y estudiantes.
- **Fomento De Una Cultura De Innovación Y Eficiencia:** La implementación exitosa de un proyecto tecnológico como HERMES puede servir como un catalizador para fomentar una cultura de innovación y la búsqueda de soluciones basadas en tecnología para otros desafíos administrativos y académicos dentro de Unitrópico. Promueve la idea de que los procesos pueden ser mejorados y optimizados mediante la ingeniería y el desarrollo de software.
- **Potencial Mejora En La Equidad:** Al basar las asignaciones en un conjunto de reglas y un proceso algorítmico, se puede reducir la subjetividad y promover una distribución más equitativa de los recursos (aulas deseables, horarios convenientes) y de la carga docente.

9.4 Otros Impactos Considerados

- ***Impacto Ambiental*:** Si bien no es un objetivo primario, HERMES podría tener impactos ambientales indirectos positivos. Una mejor planificación y uso de los espacios podría llevar a una optimización del consumo energético en la universidad. Además, la digitalización del proceso y de los horarios podría reducir la necesidad de impresiones masivas de borradores y versiones finales de los horarios.
- ***Impacto Cultural (Institucional)*:** La adopción de HERMES puede impulsar un cambio cultural hacia una planificación más sistemática, proactiva y basada en datos dentro de Unitrópico. Acostumbrar al personal a utilizar herramientas de software para la toma de decisiones complejas es un paso importante en la modernización de la gestión.

- ***Impacto Político (Institucional)*:** Un sistema eficiente y transparente de asignación de horarios puede fortalecer la gobernanza interna y la credibilidad de la administración académica. Las políticas institucionales relacionadas con la asignación de recursos, la carga docente y la planificación académica pueden verse respaldadas y facilitadas por una herramienta como HERMES.

CRONOGRAMA DE ACTIVIDADES

La estimación para el desarrollo de este sistema de generación automática de horarios en Unitrópico fue de **diez meses**, con un inicio proyectado en **2022**. Sin embargo, como ocurre a menudo en proyectos de ingeniería de software de alta complejidad y realizados por una sola persona, la ejecución real se extendió significativamente, culminando en un periodo aproximado de **tres años**.

Tabla 1

Cronograma de Actividades

1. CRONOGRAMA DE ACTIVIDADES										
ACTIVIDAD	**MES 1**	**MES 2**	**MES 3**	**MES 4**	**MES 5**	**MES 6**	**MES 7**	**MES 8**	**MES 9**	**MES 10**
toma de datos	x	x								
análisis			x							
front end				x	x	x	x	x		
back end						x	x	x	x	x
testing								x	x	x

Tabla 2

Cronograma Inicial

CRONOGRAMA DE ACTIVIDADES												
	ACTIVIDADES	Duración (trimestral por columna) Fecha de inicio septiembre 2022 - Fecha fin junio 2025										
		1	2	3	4	5	6	7	8	9	10	11
1	ANTEPROYECTO (Documentación inicial de la propuesta)	**x**										
2	TOMA DE REQUERIMIENTOS (Necesidades a cubrir)	**x**	**x**									

3	ANÁLISIS DE REQUERIMIENTOS (Diseñar las mejores soluciones con base a los requerimientos)	**x**	**x**								
4	DESARROLLO DEL FRONT END (interfaz de usuario)			**x**	**x**						
	RECESO (Tiempo en que no se trabajó en el proyecto por motivos personales)					**x**			**x**		
5	DESARROLLO DEL BACK END (maneja la lógica de la aplicación)						**x**	**x**		**x**	
6	TESTING (Realizar pruebas de la funcionalidad)							**x**		**x**	
7	DOCUMENTACIÓN (Informe de entrega de tesis, documentación del proyecto, modelos de la base de datos)								**x**	**x**	**x**

Nota. Elaboración propia.

Tabla 2

Presupuesto

RECURSO	DESCRIPCIÓN	PRESUPUESTO
Equipo Humano	-Diseñador gráfico	**$**52.200.000.
	-Desarrollador front end	**$**76.000.000.
	-desarrollador back end	**$**54.000.000.
	-desarrollador tester	**$**80.000.000.

Equipos y Software	-2 Computador portátil Lenovo ideapad Gaming (i5 de 10 generación, 16 gb de ram ddr4, disco m.2 de 1 tb, tarjeta gráfica nvidia gtx 1650 ti).	**$**8.798.000.
	-2 computador hp laptop (core i5 13th gen, nvidia geforce gtx 3060, 16gb de ram ddr5, m.2 crucial de 500gb, disco mecánico 1tb).	**$**9.316.000
	-Almacenamiento en firebase por 10 gb usadas para pruebas de datos.	**$**439.230
	-Repositorio de Python(12 meses).	
	-pantalla qbex	**$**193.262
	-mouse g502	**$**680.000
	-curso de angular	**$**211.900
	-curso de Python	**$**379.900
	-curso inteligencia artificial	**$**279.900
|Viajes y Salidas de Campo	-desplazamiento a la universidad	**$**5.000.000

Materiales y suministros	-2 agenda.	**$**36.000
	-2 esferos.	**$**5.000
	-recibo de la luz por mes	**$**3.000.000
	-recibo de internet por mes	**$**1.000.000
	-4 escritorios	**$**2.387.600
	- 4 sillas de escritorio	**$**2.532.512
	- 4 mouse pad redragon p006	**$**239.600
	-4 gafas antireflejo con filtro azul	**$**3.400.000
	- otros	**$**10.000.000
Bibliografía		
TOTAL	**10 meses:**	**$**346.139.204
TOTAL	**36 meses:**	**$**10.169.619.304

Nota. Elaboración Propia.

CONCLUSIONES

El desarrollo de este proyecto de grado se centró en la temática crucial. Para la gestión académica de la Universidad Internacional del Trópico Americano (Unitrópico). La generación automática de horas de clase para las asignaturas, con el objetivo de optimizar asignaciones de recursos y mejorar la planificación académica. Este desafío surgió de las necesidades de superar las ineficiencias del método manual actual.

El desarrollo de este proyecto de grado se centró en una temática crucial para la gestión académica de la Universidad Internacional del Trópico americano Unitrópico, la generación automática de horarios de clase para las asignaturas, con el objetivo de optimizar la asignación de recursos y mejorar la planificación académica. Este desafío surgió de la necesidad de superar las ineficiencias del método manual actual, exacerbadas por la creciente demanda estudiantil. Para abordar este problema, se diseñó y desarrolló una aplicación web automatizada que permite gestionar la información clave de docentes, asignaturas, aulas, laboratorios y áreas deportivas. Esto se logró a través de una metodología ágil e iterativa que, aunque individual, permitió construir una solución funcional y adaptativa. Se utilizaron Angular 18 para una interfaz de usuario intuitiva, Firebase 9 para la gestión robusta de datos en la nube y Python para la implementación del algoritmo de optimización pasado en programación entera.

Los resultados principales del estudio demuestran que el sistema web desarrollado proporciona una plataforma eficiente para la administración de los recursos universitarios y la estructuración de la información necesaria para la asignación de horarios. La interfaz de usuario es amigable y funcional, permitiendo al personal administrativo registrar y actualizar los datos maestros de manera efectiva. Sin embargo, es importante señalar que, aunque se implementó el algoritmo de programación entera en Python y se lograron generar horarios válidos para

escenarios controlados, la alta complejidad inherente a la optimización de horarios a gran escala y la capacidad computacional necesaria para su ejecución al 100% de su potencial aún requieren de futuras investigaciones. Y recursos adicionales. Esto significa que, si bien la base del algoritmo está establecida, su robustez y desempeño óptimo.

En primer lugar, se valida la viabilidad de automatizar y optimizar la gestión de horarios en unitrópico, lo que representa un paso importante hacia la modernización de sus procesos académicos.

En segundo lugar, se demuestra el potencial de la programación entera como una herramienta poderosa para resolver problemas complejos de asignación de recursos en el ámbito educativo, aunque con los desafíos inherentes a su implementación a gran escala. Finalmente, la creación de una plataforma web accesible y con una gestión de datos centralizada sienta las bases para una futura toma de decisiones más informada y estratégica en la universidad.

Las principales contribuciones de este proyecto son la propuesta y el desarrollo de una solución tecnológica hecha a la medida para las necesidades de un hidrópico, ofreciendo una alternativa innovadora al proceso manual que genera conflictos y su utilización de recursos. Además, se aporta un entendimiento práctico de la aplicación de técnicas de optimización avanzadas programación entera en un problema real, evidenciando tanto su potencial como sus desafíos en el contexto de un proyecto de grado. Al comparar los resultados obtenidos con los artículos revisados en el estado del arte, se observa que el enfoque híbrido de desarrollo web moderno angular firebase, combinado con la optimización por programación entera, es una tendencia prometedora, similar a lo que proponen diversas investigaciones y soluciones comerciales. Si bien nuestro sistema logró una eficiencia considerable en la gestión de datos y la capacidad de generar horarios válidos, las limitaciones en la optimización del algoritmo, como se

indicó, reflejan la complejidad mencionada en la literatura sobre el UTP, problema de asignación de horarios universitarios, donde la escalabilidad sigue siendo un reto incluso para sobres comerciales. Uno 2. Esto demuestra que nuestro trabajo se alinea con los desafíos actuales del campo y contribuye al conocimiento existente al validar la aplicación de estas tecnologías en un contexto real, aunque con la honesta identificación de las áreas de mejora.

Referencias Bibliográficas

[1] Eduardo alexander Duque y Juan David Gonzalez, «*Diseño de un modelo de programación lineal para la asignación de aulas en la Institución Universitaria Esumer, Colombia*», https://www.revistaespacios.com/a17v38n10/in173810.htm*l*, Colombia, pp. 1-1, 25 de octubre de 2017.

[2] Rodrigo Hernandez y Jaime Miranda, «*Programación de Horarios de Clases y Asignación de Salas para la Facultad de Ingeniería de la Universidad Diego Portales Mediante un Enfoque de Programación Entera*», Revista de Ingeniería de Sistemas, pp. 121-141, 2008.

[3] Juan Camilo Marín Ángel y Pablo Andrés Maya Duque, «*Modelo lineal para la programación de clases en una institución educativa*» http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1794-91652016000100004, Medellín, pp. 47-71, 16 de junio de 2016.

[4] Augusto cortez vasquez, Gissela Rosales Gerónimo, Raúl Naupari Quiroz, y Hugo Vega Huerta, «*Sistema de apoyo a la generación de horarios basado en algoritmos genéticos*»,

https://sisbib.unmsm.edu.pe/bibvirtual/publicaciones/risi/2010_n1/v7n1/a05v7n1.pdf, pp. 37-55, 2010.

[5] Jose Maria Mejia caballero, «*Asignación de horarios de clases universitarias mediante algoritmos evolutivos*.», *https://manglar.uninorte.edu.co/handle/10584/80#page=1*, pp. 140-149, 15 de julio de 2009.

[6] CARLA LENINCA PACHECO AGÜERO, «*Distribución Óptima de Horarios de Clases utilizando la técnica de Algoritmos Genéticos* », http://jupiter.utm.mx/~tesis_dig/6557, Oaxaca, 11 de agosto de 2000.

[7] John Fredy Franco Baquero, Eliana Mirledy Toro Ocampo, y Ramón Alfonso Gallego Rendón, «*Problema de asignación óptima de salones resuelto con Búsqueda Tabú*», http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0122-34612008000200011, Barranquilla, pp. 149-175, 24 de julio de 2008.

[8] MAURICIO GRANADA E, ELIANA M. TORO OCAMPO, y JOHN F. FRANCO BAQUERO, «*Programación óptima de horarios de clase usando un algoritmo memético*», https://www.redalyc.org/articulo.oa?id=84920491051, Pereira, pp. 255-260, 30 de mayo de 2006.

[9] Linda Lucia Esquivel y Juan Pablo Orjuela, «*Programación de horario escolar con multi-localización y preferencias docentes*», https://dialnet.unirioja.es/servlet/articulo?codigo=7300017, Cali Colombia, pp. 20-29, 2014.

[10] Edmund K. Burke, Michel Gendreau, y Matthew Hyde, «*Hiperheurística: un estudio del estado del arte*», https://www.tandfonline.com/doi/abs/10.1057/jors.2013.71, pp. 1695-1724, 21 de diciembre de 2017.

[11] Md. Mushfiqur Rahman, Sabah Binte Noor, y Fazlul Hasan Siddiqui, «*Programación automatizada de clases a gran escala en MiniZinc*», https://www.researchgate.net/publication/345970845_Automated_Large-scale_Class_Scheduling_in_MiniZinc, pp. 28-38, noviembre de 2020.

[12] Michael W. Carter y Gilbert Laporte, «*Avances recientes en la programación de cursos prácticos*», https://link.springer.com/chapter/10.1007/bfb0055878, pp. 3-19, 1 de enero de 2006.

[13] Stuart J. Russell y Peter Norvig, «*INTELIGENCIA ARTIFICIAL*», https://luismejias21.wordpress.com/wp-content/uploads/2017/09/inteligencia-artificial-un-enfoque-moderno-stuart-j-russell.pdf, Madrid, 2004.

[14] A. Schaerf, «*Generador de horarios universitarios mediante búsqueda tabú*», https://www.scirp.org/reference/referencespapers?referenceid=1938970, pp. 87-127, 26 de diciembre de 2016.

[15] H Paul Williams, «*Construcción de modelos en programación matemática*», https://books.google.com.pa/books?id=YJRh0tOes7UC&printsec=frontcover#v=onepage&q&f=false, 2013.

[16] Laurence Wolsey, «*Programación entera*», https://onlinelibrary.wiley.com/doi/book/10.1002/9781119606475, 19 de septiembre de 2020.

[17] Google, «*OR-Tools - Herramientas de optimización de Google*», https://sourceforge.net/projects/or-tools.mirror/, 13 de mayo de 2025.

[18] Google, «*Angular Documentos*», 2013 - Versión 17.3.13-local+sha.fc6e55b.

[19] Guido Van Rossum, «*La referencia del lenguaje Python*», https://docs.python.org/3/reference/introduction.html.

[20] Google Developers, «Documentación para desarrolladores de Firebase», https://firebase.google.com/docs/reference.

[21] Armin Ronacher de Pocoo, «*Documentación de Flask*.», https://flask.palletsprojects.com/es/stable/quickstart/.

[22] David E. Joslin, «*Optimización de la rueda chirriante* », https://cdn.aaai.org/AAAI/1998/AAAI98-048.pdf.

[23] Marco Chiarandini, Mauro Birattari, Krzysztof Socha, y Olivia Rossi-Doria, «*Un algoritmo híbrido eficaz para la planificación de horarios de cursos universitarios*», https://www.researchgate.net/publication/220366639_An_effective_hybrid_algorithm_for_university_course_timetabling, octubre de 2006.

[24] Pascal Van Hentenryck, «*Satisfacción de restricciones en la programación lógica*», 23 de marzo de 1989.

[25] Google OR Tools, «*Optimización de restricciones*», https://la.mathworks.com/products/optimization.html.

[26] Unitrópico, "*Propuesta Trabajo de Grado: Aplicación web para la generación automática de horarios de clase de las asignaturas de la Unitrópico de Yopal Casanare*," Documento interno, Universidad Internacional del Trópico Americano, Yopal, Casanare, 2022. (Referencia al documento de propuesta que me proporcionaste).

[27] A. Reyes Tuay, *"Esquema realizado por el ingeniero Alfonso*," Documento PDF (diagrama manual), Comunicación personal Ing. Tovar (Ing. Alfonso), aprox. 2022. *(*Referencia al PDF que subiste).

ANEXOS

Anexo A. *documentación del repositorio que seria el link siguiente:* " https://deepwiki.com/Andres-Reyes/Tesis-Grado-Ingenieria-Sistemas "

Anexo B. *Cronograma de actividades*, https://docs.google.com/document/d/1OT1Gy03YD3k5pCb_GyBX2t291WaCPafF/edit?usp=drive_link&ouid=103478930406961399785&rtpof=true&sd=true

Anexo C. esquema realizado por el ingeniero Alfonso https://drive.google.com/file/d/1yzo6zq91fZvZ04pNPqxnCZYe3OtZydLn/view?usp=drive_link

Anexo D. Grabación reunión Ing. Alfonso https://drive.google.com/file/d/1l4fM_RleACN7MM-MFLtlsA-uKgWVxbR0/view?usp=drive_link

Anexo E. Modelo entidad relación https://drive.google.com/file/d/1-4Y6byrgCLcgNz_hffYFZuT35UEvh5oS/view?usp=drive_link

Anexo f. objetos de la base de datos https://docs.google.com/document/d/1mRcjJ8mfIaGgYgj-yzmLV3bFNo1vAD4B/edit?usp=drive_link&ouid=103478930406961399785&rtpof=true&sd=true