



Proyecto Final 5k2 - 2026 - Grupo 1



Autores:

- Blencio Juan Mateo 90317
- Córdoba Oyhamburu Consuelo 85832
- Lypnik Zoi 86495
- Hadjadourian Placereano Jonatan Esteban 99507
- Gil Matías 89765

Profesores:

- Gastañaga, Iris
- Balut, Jorge

SIGA

SISTEMA INTELIGENTE DE GESTIÓN ÁULICA

OPTIMIZACIÓN DE ESPACIOS ÁULICOS

Beneficios que aportan a la gestión



Mejora el uso de
las aulas



Planificación
más ágil



Trazabilidad
y control



Decisiones
basadas en
datos



Notificaciones

Tecnologías clave y Herramientas



Algoritmo inteligente de
asignación



Conteo de asistencia
automático

¿Quiénes usan SIGA?



Subsecretaría
Académica



Auxiliares
Áulicos



Docentes

Problemas que resuelve



Aulas vacías
o subutilizadas



Sobrecupo
de aulas



Trabajo manual
repetitivo



Equipo

- Blencio, Juan Mateo - mateoblencio@gmail.com
- Córdoba, Consuelo - consucordoba9@gmail.com
- Gil, Matías Ezequiel - ezemgil@gmail.com
- Hadjadourian, Jonatan - jonatan.hadjadourian@gmail.com
- Lypnik, Zoi - lypnik.zoi@gmail.com



Sistema Inteligente de Gestión Áulica

Blencio, Juan Mateo

Córdoba Oyhamburu, Consuelo

Gil, Matías Ezequiel

Hadjadourian Placereano, Jonatan Esteban

Lypnik, Zoi

Universidad Tecnológica Nacional, Facultad Regional Córdoba

Abstract

La Subsecretaría Académica de la Universidad Tecnológica Nacional Facultad Regional Córdoba asignaba aulas para clases y exámenes de forma manual, lo cual llevaba mucho tiempo y no se aprovechaba de la mejor manera el espacio. Por ello se desarrolló la plataforma SIGA (Sistema Inteligente de Gestión Áulica) la cual permite asignar fácil e intuitivamente las aulas. Además cuenta con un algoritmo que sugiere asignaciones de forma individual o masiva los cursos o eventos de forma óptima. Cuenta también con un módulo que integra cámaras de seguridad para obtener la cantidad real de asistentes a un curso, un módulo de notificaciones a través de Autogestión y un módulo de estadísticas y reportes.

Este desarrollo web se realizó principalmente en Java, usando una base de datos Postgres y desplegado en un contenedor Docker en los servidores de la facultad.

Como metodología de trabajo se eligió Spring permitiendo la entrega de valor al cliente de forma progresiva.

Como resultado de la implementación se redujo el tiempo para la asignación, se adecuó cada la asignación a la cantidad real de alumnos que asisten y se brindó información importante para la toma de decisiones en cuanto a uso de los espacios.

Palabras claves

Asignación áulica, algoritmo, cámaras, solución de negocio.

Introducción:

Actualmente la asignación de aulas en la UTN FRC se realiza de forma manual, con

soporte digital en diferentes plataformas como Outlook o Excel y de forma fragmentada, lo cual significa que distintos actores asignan distintas aulas y esta información no se comparte. Esto resulta en una inversión muy importante de tiempo dedicada a asignar aulas, especialmente al comienzo de ambos semestres, reservas duplicadas, falta de trazabilidad y, en general, una utilización subóptima de los espacios.

Elementos del Trabajo y metodología

Al inicio del proyecto se realizaron entrevistas con la subsecretaria académica Ing. Analía Guzmán, quien explicó como realizaban la asignación de aulas y las dificultades que enfrentaban. En base a esto se realizó una propuesta que constaba de un turnero digital, con un algoritmo de sugerencias, módulo de notificaciones, estadísticas y control de asistencia por cámaras. Este proyecto recibió el apoyo del secretario académico Ing. Roberto Muñoz y el decano Ing. Federico Olivo Aneiros. Con este respaldo se realizaron pedidos formales al área de cómputos para obtener los accesos de lectura y escritura a datos oficiales de la facultad necesarios para el sistema.

En cuanto a organización del equipo se adoptó la metodología Scrum^[1] con sprints de dos semanas y ceremonias pautadas, utilizando Jira^[2] para gestionar el Product Backlog y las User Stories.

A nivel técnico, se determinó el uso principal de Java^[3] por adecuarse a los requerimientos del proyecto y a las habilidades técnicas del equipo. Para el frontend se utilizó el framework React^[4] y para el backend Spring Boot^[5]. El despliegue se realiza en contenedores Docker^[6] aplicando prácticas de integración continua y despliegue continuo (CI/CD). Para la base de datos se seleccionó PostgreSQL^[7] por compatibilidad con la infraestructura de la facultad, facilitando su posterior mantenimiento. El código y los activos se estructuraron en dos repositorios de GitHub^[8] (Frontend y Backend) con la estrategia de branching gitflow^[9].

Resultados

Para atacar los problemas descritos se desarrolló un turnero web inteligente, el cual se integra con otros sistemas de la facultad como Sysacad, sistema de cámaras y plataforma de formularios.

El core del producto es un turnero donde se pueden asignar eventos (clases, conferencias, cursos) a aulas. En cada asignación queda registrado el tipo de evento, fecha y hora y si corresponde, especialidad, materia y comisión. Estos datos se importan directamente desde el sistema académico de la facultad "Sysacad" mediante una API para que estén siempre actualizados. Las aulas están organizadas en edificios y tienen una capacidad máxima, la cual se carga manualmente de acuerdo a un

relevamiento realizado por la secretaría académica. El sistema permite superposiciones y sobrecupo (cuando una reserva excede la capacidad del aula) pero lo muestra con alertas y colores diferentes. La información de las asignaciones luego impactan en Sysacad, haciendo que la base de datos oficial esté siempre actualizada.

Como herramienta distintiva a cualquier otro sistema de reserva se implementó un algoritmo parametrizable que permite realizar sugerencias para uno o varios cursos, facilitando el trabajo de asignación, en especial al comienzo de cada semestre. El algoritmo se basa en metaheurísticas, por lo que se decidió utilizar la librería Timefold solver^[10]. Éste no resuelve el problema con una fórmula matemática exacta como haría un solver de programación lineal entera. En su lugar, explora el espacio de soluciones probando movimientos (mover un evento a otra aula, intercambiar aulas entre dos eventos) y se queda con el que mejor puntaje da, repitiendo esto dentro de un tiempo límite. Los puntajes de cada asignación penalizan con distintos pesos parametrizables el sobrecupo, cambio de aula para un mismo curso, logra sobrante en el aula, entre otros. Se realizó un trabajo de optimización de estos pesos en base a la importancia relativa que le asigna a cada situación Secretaría Académica.

El sistema cuenta con un módulo de cámaras que toman capturas de las aulas en intervalos regulares, cuentan la cantidad de personas. Estos datos se procesan y promedian a lo largo de varias clases (la cantidad de clases es parametrizable) lo cual genera un dato más confiable

Para realizar un pedido de aula para clase, examen o evento, el interesado puede ingresar a

<https://serviciostid.frc.utn.edu.ar/> y completar el formulario. Esto genera una solicitud en el sistema, que puede ser atendido y notificado de una forma sencilla.

Estas notificaciones se realizan mediante autogestión y afectan a la sección “mis horarios”. De esta forma tanto docentes como alumnos saben inmediatamente a qué aula deben asistir.

Discusión

La implementación del sistema SIGA supone una mejora notable en los tiempos requeridos para la asignación de aulas, y brinda más información tanto a los responsables de asignación mediante estadísticas y gráficos como a docentes y alumnos al ofrecer notificaciones por autogestión. Además la interfaz gráfica es mucho más intuitiva y fácil de usar que el outlook que se usaba anteriormente. El sistema irá mejorando a medida que se incorporen más cámaras, cual permitirá tener información real de la asistencia

Toda transformación requiere un cambio en la cultura, en este caso el cambio afecta a distintas áreas que administraban edificios o aulas de forma independiente y a profesores, los cuales deben incorporar la nueva forma de realizar solicitud de aula.

Conclusión

El desarrollo del sistema SIGA resolvió la fragmentación y demoras en la administración de espacios de la UTN FRC, reemplazando procesos manuales por una solución centralizada, ágil y con total trazabilidad.

La integración directa con Sysacad y Autogestión, sumada al motor de optimización con metaheurísticas y al módulo de visión por cámaras para medir la asistencia real, permite reducir drásticamente los tiempos de asignación y tomar decisiones basadas en datos concretos.

En definitiva, la plataforma no solo optimiza el uso de la infraestructura edilicia, sino que sienta las bases para una modernización operativa y cultural en beneficio de toda la comunidad académica.

Referencias

- [1] SCRUM: (2025, 17 de agosto) Recuperado de: www.scrumguides.org/
- [2] Jira (2025, 17 de agosto) Recuperado de: <https://www.atlassian.com/es/software/jira>
- [3] Java (2025, 17 de agosto) Recuperado de: <https://www.java.com/es/>
- [4] React (2025, 17 de agosto) Recuperado de: <https://es.react.dev/>
- [5] Spring Boot (2025, 17 de agosto) Recuperado de: <https://spring.io/projects/spring-boot>
- [6] Docker (2025, 17 de agosto) Recuperado de: <https://www.docker.com/>
- [7] PostgreSQL (2025, 17 de agosto) Recuperado de: <https://www.postgresql.org/>
- [8] GitHub (2025, 17 de agosto) Recuperado de: <https://github.com/>
- [9] Gitflow (2025, 17 de agosto) Recuperado de: <https://www.atlassian.com/es/git/tutorials/comparing-workflows/gitflow-workflow>
- [10] TimeFold solver (2025, 17 de agosto) Recuperado de: <https://github.com/timefoldai/timefold-solver>

Datos de contacto

- Blencio Juan Mateo - Mateoblencio@gmail.com
- Lypnik Zoi - Lypnik.zoi@gmail.com
- Jonatan Esteban Hadjadourian Placereano jonatan.hadjadourian@gmail.com
- Gil Matías - ezemgil@gmail.com
- Consuelo Córdoba Oyhamburu - consucordoba9@gmail.com

PLANILLA PARA CATALOGAR EL PROYECTO FINAL

AÑO	2026	CURSO Y NRO. DE GRUPO	5K2 - G01
NOMBRE DEL SISTEMA / PROYECTO			
SIGA			
CATEGORÍA (Solución de Negocio / Producto / Proyecto de Impacto Social)			
Solución de Negocio			
HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS			
ÁMBITO DE APLICACIÓN	NOMBRE Y VERSIÓN		
ENTORNO DE DESARROLLO	Antigravity - IntelliJ IDEA		
REPOSITORIOS Y VERSIONADO	GitHub - Git 2.55.0		
PROGRAMACIÓN	Backend: Java 21 (OpenJDK 21.0.12.1 LTS) - Spring Boot 4.0.6 - Maven 3.9.16 Frontend: TypeScript 6.0.3 - React 19.2.6 - Vite 8.0.14 - Node.js 20.19.5 (npm 11.7.0) - Tailwind CSS 4.3.0		
BASE DE DATOS	PostgreSQL 16.14		
COMUNICACIÓN INTERNA	Discord - WhatsApp		
CAPACITACIÓN	YouTube - Documentación oficial		
PRUEBAS DE SISTEMA	Backend: JUnit 6.0.3 - Mockito 5.20.0 - Testcontainers 2.0.5 Frontend: Vitest 4.1.10 - Testing Library 16.3.2 - MSW 2.15.0 - Playwright 1.62.1		
GESTIÓN DEL PROYECTO	Jira - Obsidian		
DOCUMENTACIÓN	Google Drive - Notion		
MODELOS	Draw.io		