

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA DE SISTEMAS**

### **DESARROLLO DE UNA APLICACIÓN WEB DE PLANIFICACIÓN ACADÉMICA PARA LA FACULTAD DE INGENIERÍA DE SISTEMAS**

#### **GESTIÓN DE ESPACIOS, HORAS Y PREPLANIFICACIÓN**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO/A EN  
SOFTWARE**

**MAHATMA ISRAEL QUIJANO ZANIPATTINI**

**mahatma.quijano@epn.edu.ec**

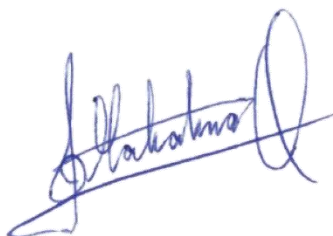
**DIRECTOR: CARLOS EDUARDO ANCHUNDIA VALENCIA**

**carlos.anchundia@epn.edu.ec**

**DMQ, septiembre 2022**

## CERTIFICACIONES

Yo, Mahatma Israel Quijano Zanipattini declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



---

**Mahatma Israel Quijano Zanipattini**

Certifico que el presente trabajo de integración curricular fue desarrollado por Mahatma Israel Quijano Zanipattini, bajo mi supervisión.



---

**Carlos Eduardo Anchundia Valencia**  
**DIRECTOR**

## **DECLARACIÓN DE AUTORÍA**

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Mahatma Israel Quijano Zanipattini

Carlos Eduardo Anchundia Valencia

Luis Alejandro Llanganate Valencia

Diana Stefanía López Prikhodko

## **DEDICATORIA**

Dedico el presente trabajo de integración curricular a mis compañeros y los demás estudiantes de la Facultad de Ingeniería de Sistemas, con quienes cada semestre hemos sido partícipes del proceso de matrículas y hemos sido testigos de la problemática relacionada a la planificación académica.

## **AGRADECIMIENTO**

Agradezco a la Facultad de Ingeniería de Sistemas y a su cuerpo docente por el conocimiento impartido a lo largo de mi formación profesional.

Agradezco a mis compañeros y amigos de la carrera por el apoyo incondicional durante cada semestre.

Agradezco a Luis Alejandro Llanganate Valencia y Diana Stefanía López Prikhodko, colaboradores del trabajo de integración curricular. Sin su ayuda, este trabajo no habría sido posible.

Agradezco a Carlos Eduardo Anchundia Valencia, director del presente trabajo, por el conocimiento compartido con nosotros que nos ha guiado durante este proyecto.

# ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	1
DECLARACIÓN DE AUTORÍA .....	2
DEDICATORIA .....	3
AGRADECIMIENTO.....	4
ÍNDICE DE CONTENIDO.....	5
ÍNDICE DE FIGURAS .....	7
ÍNDICE DE TABLAS.....	8
RESUMEN.....	9
ABSTRACT.....	10
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	11
1.1 Descripción del proyecto .....	11
1.2 Descripción del componente .....	12
1.3 Objetivo general.....	13
1.4 Objetivos específicos.....	13
1.5 Alcance .....	14
2 MARCO TEÓRICO .....	14
2.1 Planificación académica .....	14
2.2 Planificación académica en la Facultad de Ingeniería de Sistemas de la Escuela Politécnica Nacional.....	15
2.3 Desarrollo dirigido por comportamiento .....	16
3 METODOLOGÍA.....	18
3.1 Investigación sobre la problemática .....	18
Investigación de fuentes bibliográficas.....	18
Entrevistas a <i>stakeholders</i> .....	18
3.2 Análisis de requerimientos.....	20
Historias de usuario .....	20

Estimación de las historias .....	21
Planificación en <i>sprints</i> .....	22
3.3 Selección de herramientas y diseño de la arquitectura .....	22
3.4 Desarrollo de los <i>sprints</i> .....	24
<i>Sprint 1</i> .....	24
<i>Sprint 2</i> .....	27
<i>Sprint 3</i> .....	33
4 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES .....	39
4.1 Resultados .....	39
4.2 Conclusiones.....	40
4.3 Recomendaciones .....	41
5 REFERENCIAS BIBLIOGRÁFICAS .....	42
6 ANEXOS.....	43
ANEXO I. DOCUMENTO DE REVISIÓN SISTEMÁTICA PARA LA INVESTIGACIÓN DE FUENTES BIBLIOGRÁFICAS.....	43
ANEXO II. DOCUMENTO DE EXTRACCIÓN DE INFORMACIÓN REALIZADA DURANTE LA INVESTIGACIÓN DE FUENTES BIBLIOGRÁFICAS.....	43
ANEXO III. BACKLOG DEL TRABAJO DE INTEGRACIÓN CURRICULAR .....	44
ANEXO IV. INFORMACIÓN DEL PROYECTO EN LA HERRAMIENTA NOTION .....	46
ANEXO V. REPOSITORIO DEL FRONTEND .....	46
ANEXO VI. REPOSITORIO DEL BACKEND.....	46
ANEXO VII. REUNIÓN INFORMATIVA DEL SEMESTRE 2022-A.....	46
ANEXO VIII. PRESENTACIÓN CON LA NUEVA SUBDECANA DE LA FIS Y EL COORDINADOR DE LA CARRERA DE INGENIERÍA DE SOFTWARE.....	46
ANEXO IX. FEATURES DE CADA HISTORIA.....	47
ANEXO X. DOCUMENTACIÓN DE LA API EN SWAGGER.....	49
ANEXO XI. INTERFACES DE USUARIO PARA CADA HISTORIA.....	50
ANEXO XII. ENTREVISTA A LOS STAKEHOLDERS REALIZADA POR CARLOS ANCHUNDIA.....	56

ANEXO XIII. ENTREVISTA AL SUBDECANO REALIZADA POR EL EQUIPO DE DESARROLLO .....	56
ANEXO XIV. ENTREVISTA A ALEX ULLOA.....	56

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Esquema general del proceso del proyecto. En color anaranjado se encuentra lo contemplado en el componente “gestión de espacios, horas y preplanificación”.....	12
<b>Figura 2.</b> Principales actividades y salidas de BDD. Tomado de [5].....	17
<b>Figura 3.</b> Estructura de una característica y un escenario (Gherkin). Tomado y traducido de [5]. .....	17
<b>Figura 4.</b> Valores de estimación usados para las historias de usuario. ....	21
<b>Figura 5.</b> Estimación de las historias de usuario del componente “gestión de espacios, horas y preplanificación”. ....	21
<b>Figura 6.</b> Diagrama de la arquitectura propuesta. ....	23
<b>Figura 7.</b> Estructura del archivo .env. ....	23
<b>Figura 8.</b> Estructura inicial de los pasos de un escenario utilizando Cucumber y TypeScript. ....	24
<b>Figura 9.</b> Archivo cucumber.js usado para la ejecución de pruebas.....	25
<b>Figura 10.</b> Configuración del archivo package.json del backend para la ejecución de pruebas.....	25
<b>Figura 11.</b> Estructura de la carpeta de pruebas (features y step definitions). ....	26
<b>Figura 12.</b> Archivo con las funciones BeforeStep y AfterStep necesarias para la ejecución de pruebas. Se muestra la versión final de este archivo.....	26
<b>Figura 13.</b> Roles del proyecto vistos en la tabla “rol”.....	28
<b>Figura 14.</b> Ejemplo de archivo CSV para la creación de múltiples espacios físicos. ....	28
<b>Figura 15.</b> Ejecución de las pruebas del primer escenario de la historia M1.....	29
<b>Figura 16.</b> Ejecución de las pruebas del segundo escenario de la historia M1. ....	29
<b>Figura 17.</b> Prototipo del módulo de gestión de espacios físicos. ....	30
<b>Figura 18.</b> Organización usada para los módulos en frontend.....	30
<b>Figura 19.</b> Interfaz gráfica del módulo de gestión de espacios físicos. Pantalla principal. ....	31
<b>Figura 20.</b> Ejecución de la prueba del escenario de la historia M2. Versión del sprint 2. ....	32
<b>Figura 21.</b> Prototipo del módulo de horas no disponibles. ....	32
<b>Figura 22.</b> Interfaz gráfica del módulo de horas no disponibles. Pantalla de selección. Versión sprint 2.....	33



<b>Figura 23.</b> Ejecución de las pruebas de los dos escenarios de la historia M2. ....	34
<b>Figura 24.</b> Interfaz gráfica del módulo de horas no disponibles. Pantalla de solicitud. Versión sprint 3.....	35
<b>Figura 25.</b> Interfaz gráfica del módulo de horas no disponibles. Pantalla de lista de solicitudes para el jefe de departamento. ....	35
<b>Figura 26.</b> Interfaz gráfica del módulo de horas no disponibles. Pantalla para aprobar o rechazar una solicitud específica. ....	36
<b>Figura 27.</b> Ejemplo de correo de aprobación de la solicitud de horas no disponibles.....	36
<b>Figura 28.</b> Ejemplo de correo de rechazo de la solicitud de horas no disponibles. ....	37
<b>Figura 29.</b> Ejecución de la prueba del escenario de la historia M3.....	38
<b>Figura 30.</b> Interfaz gráfica del módulo de registro de alumnos por asignatura. Pantalla de registro.....	38
<b>Figura 31.</b> Diagrama comparativo de historias de usuario planificadas y realizadas. ....	40

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Historias de usuario del componente “gestión de espacios, horas y preplanificación”.....	20
<b>Tabla 2.</b> Planificación inicial de cada historia de usuario por cada sprint. ....	22
<b>Tabla 3.</b> Nueva planificación de las historias de usuario por sprint. ....	27

## RESUMEN

La Facultad de Ingeniería de Sistemas de la Escuela Politécnica Nacional se encarga de formar profesionales con conocimientos sobre Tecnologías de Información y Comunicaciones. Como parte de sus procesos académicos, la facultad lleva a cabo semestralmente el proceso de planificación académica. Este proceso aborda la generación de horarios en base al cuerpo docente, las carreras ofertadas, las diferentes asignaturas de estas, los espacios físicos disponibles, las restricciones de horas no disponibles de cada docente y el número de estudiantes que cursará cada asignatura. Actualmente, las autoridades encargadas realizan la planificación académica manualmente con un *software* de escritorio, razón por la cual se ven obligadas a intercambiar los archivos creados hasta obtener un horario que cuente con toda la información requerida. Para solucionar este problema se desarrolló una aplicación web para la planificación académica, aplicando la metodología ágil, el marco de trabajo Scrum y un desarrollo dirigido por comportamiento (BDD). La aplicación consiste en tres componentes, de los cuales el presente documento detalla el denominado “gestión de espacios, horas y preplanificación”. Este componente trata la información de los espacios físicos disponibles para impartir clases, la gestión de solicitudes realizadas por docentes para indicar aquellas horas no disponibles para sus actividades de docencia, y el registro del número de estudiantes que cursa cada asignatura durante un semestre y su relación con el proceso de preplanificación académica. La creación de este componente facilitó el registro y manipulación de esta información a través de una aplicación web concurrente e intuitiva.

**PALABRAS CLAVE:** Planificación académica, desarrollo dirigido por comportamiento, metodología ágil, Scrum, *stakeholder*, espacios físicos, horas no disponibles, preplanificación académica.

## **ABSTRACT**

The Facultad de Ingeniería de Sistemas at Escuela Politécnica Nacional, is in charge of training professionals with Information and Communications Technology. As part of their academic processes, this faculty does the academic planning process each semester. This process takes in count generating timetables according to the existing staff, the offered careers, the curriculum of each of these careers, the available instructional spaces, the constraints related to teacher's unavailable hours and the number of students who will attend each subject. Currently, the authorities in charge do the academic planning manually using a desktop software, reason why they are required to exchange the created files until they obtain a timetable which has all the required information. To solve this problem, we developed a web application for the academic planning, using the agile methodology, the Scrum framework, and a behavior-driven development (BDD). This application consists of three components, of which this document details the one called "management of spaces, hours and preplanning". This component takes in count the information of the available instructional spaces, the management of the teacher's requests indicating their unavailable hours for teaching activities, and the registering process of the number of students attending each subject during a semester and its relationship with the academic preplanning process. The creation of this component made it easier to register and manipulate this information through a concurrent and intuitive web application.

**KEYWORDS:** Academic planning, behavior-driven development, agile methodology, Scrum, stakeholder, instructional spaces, unavailable hours, academic preplanning.

# 1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

## 1.1 Descripción del proyecto

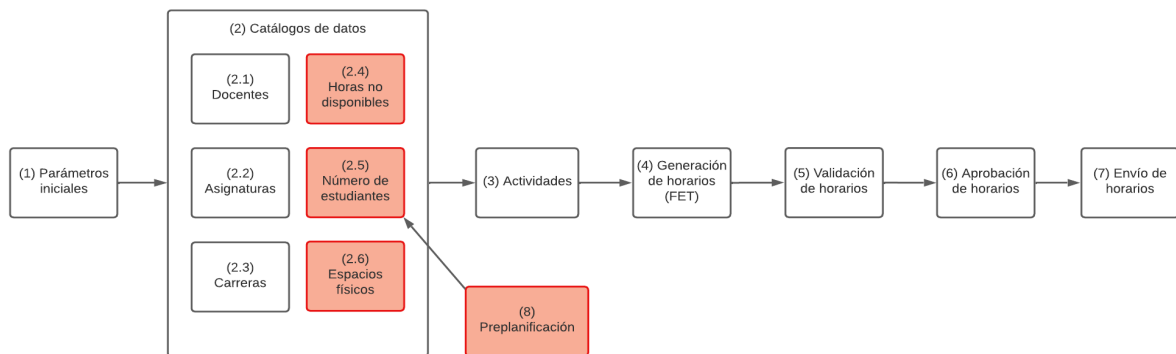
Según el Estatuto de la Escuela Politécnica Nacional, el subdecano de cada Unidad Académica es responsable de dirigir y coordinar las actividades de docencia. Una de sus actividades es la creación de la planificación académica semestral. Esta planificación debe ser aprobada por parte de los consejos de los departamentos adscritos a la facultad, sirviendo como un insumo para el proceso de matriculación de los estudiantes [1].

Para la creación de la planificación académica se deben considerar diversas restricciones como, por ejemplo: la gestión de espacios físicos de aulas y laboratorios; y el registro de las horas académicas disponibles del cuerpo docente [2]. Actualmente, los directores de cada carrera y subdecano de la Facultad de Ingeniería de Sistemas utilizan el *software* FET para el proceso de creación de la planificación académica. El motor del *software* FET permite, a través de su algoritmo heurístico, la organización de actividades en las que se asignan tanto docentes, horarios y asignaturas para los diferentes niveles [1]. No obstante, el uso de este *software* presenta varias limitaciones respecto al acceso concurrente de usuarios. Por lo tanto, los involucrados no pueden acceder simultáneamente a la información; es decir, los usuarios se ven obligados a intercambiar constantemente los archivos que genera este programa de escritorio para avanzar con sus actividades de planificación.

Adicionalmente, el *software* FET necesita información sobre el número de estudiantes a inscribirse por cada asignatura. Para esto, existe un proceso denominado preplanificación académica que se realiza después de la primera mitad del semestre en curso. Este proceso se lo lleva a cabo mediante una encuesta en Microsoft Forms en la que los estudiantes deben indicar qué asignaturas posiblemente escogerán el siguiente semestre. Sin embargo, los datos recogidos por el formulario son ingresados manualmente en el *software* FET y tienen un factor de incertidumbre que produce: (1) inexactitud de su elección realizada debido al momento en que se ejecuta la encuesta; (2) respuestas incongruentes al no controlar las relaciones de prerrequisitos y correquisitos de sus asignaturas; o (3) respuestas inoportunas.

Para solventar estos problemas del proceso de planificación académica, se propuso el desarrollo de una aplicación web que contemple todos los aspectos mencionados. En la Figura 1 se observa un esquema general de las secciones contempladas en la propuesta.

Aquellas secciones correspondientes al componente del presente documento se encuentran coloreadas en la misma figura.



**Figura 1.** Esquema general del proceso del proyecto. En color anaranjado se encuentra lo contemplado en el componente “gestión de espacios, horas y preplanificación”.

## 1.2 Descripción del componente

La solución propuesta contempla la creación de un componente para la “gestión de espacios, horas y preplanificación” que consistirá en tres módulos. El primer módulo denominado “gestión de espacios físicos” (Figura 1-2.6) permitirá el registro de las diferentes aulas y laboratorios disponibles en la Facultad de Ingeniería de Sistemas por parte de un usuario gestor de espacios. Se busca que esta información sirva de entrada para el subsistema FET de tal manera que se asignen los espacios físicos de acuerdo con las asignaturas registradas para la generación de horarios.

El módulo para el registro de horas no disponibles del docente (Figura 1-2.4) busca que el cuerpo docente tenga acceso a la aplicación, de tal manera que cada uno de sus integrantes indique por sí mismo su disponibilidad entre lunes y sábado para la realización de sus actividades de docencia. Para cumplir con esto, se propone la creación de un usuario docente con el fin de ingresar al aplicativo previo a la generación de horarios de la planificación académica. Esta información servirá como entrada al subsistema FET para indicar las restricciones entre los docentes y las horas en las que pueden impartir sus asignaturas correspondientes.

Finalmente, el módulo de preplanificación académica (Figura 1-8) busca tomar como entrada la información referente al número de estudiantes matriculados en cada asignatura existente en la malla curricular de cada carrera en un semestre dado. Estos datos se

procesarán tomando en cuenta las restricciones de aquellas asignaturas que tengan relaciones de prerrequisitos y correquisitos entre sí, de tal manera que se obtenga un cálculo del número aproximado de estudiantes que tomarán las asignaturas correspondientes al siguiente nivel académico del semestre subsecuente (Figura 1-2.5). Esta información se presentará de tal manera que permita identificar la cantidad de paralelos necesarios para cada asignatura al generar horarios, tomando en cuenta restricciones de aforo.

Con esta información procesada se podrá contar con datos fiables y oportunos que servirán como entradas para el subsistema FET para la posterior obtención de los horarios del semestre correspondiente a la planificación académica.

### **1.3 Objetivo general**

Contar con una aplicación web, usable e intuitiva, que se encuentre disponible para que múltiples usuarios realicen de manera concurrente la preplanificación y planificación académicas semestrales, mediante el procesamiento de datos fiables y oportunos para la gestión de espacios físicos y horas no disponibles.

### **1.4 Objetivos específicos**

1. Permitir a múltiples usuarios realizar concurrentemente el proceso de planificación académica mediante la disponibilidad del componente en la web.
2. Agilizar el registro de horas no disponibles para las actividades de docencia, previo a la generación de horarios, al incluir el rol de docente como parte del proceso de planificación académica.
3. Facilitar el registro de espacios físicos y horas no disponibles durante el proceso de planificación académica mediante una aplicación fácil de usar e intuitiva.
4. Disminuir la incertidumbre de los cupos disponibles para el semestre subsecuente a partir del procesamiento de la información del semestre en curso.

## 1.5 Alcance

El trabajo de integración curricular busca la creación de una aplicación web para la planificación académica que permita la generación de horarios mediante la información detallada en la sección 2.2 y el *software* FET como servicio en la nube. Esta aplicación debe contar con una interfaz intuitiva y permitir que varios usuarios la usen de manera concurrente. Para esto, el proyecto cuenta con tres componentes que abarquen cada aspecto indicado en la Figura 1.

El componente “gestión de espacios, horas y preplanificación”, detallado en la sección 1.2, contemplará la entrega de una aplicación web que cuente con los módulos de “gestión de espacios físicos”, “registro de horas no disponibles” y “preplanificación académica”. Para el uso de estos módulos, se considera la creación de roles para un gestor de espacios físicos, los docentes y los coordinadores de carrera. Igualmente, considera el despliegue de la aplicación web en un servidor de la Facultad de Ingeniería de Sistemas, una vez que se implemente cada componente del trabajo de integración curricular.

Sin embargo, el componente del presente documento no contemplará la migración de datos de horarios generados en semestres anteriores con el *software* FET. Tampoco considera la integración con ningún otro sistema externo al trabajo de integración curricular.

## 2 MARCO TEÓRICO

### 2.1 Planificación académica

Las instituciones de educación superior cuentan con una gran cantidad de información que debe ser procesada cada semestre para la correcta gestión de aulas, estudiantes, docentes, personal administrativo, así como la generación de horarios y diferentes tipos de reportes. Este proceso se denomina comúnmente como “planificación académica” y es diferente en cada institución. Además de lo mencionado anteriormente, se pueden considerar aspectos de inscripciones, boletas de calificaciones, carga horaria y asistencia de docentes, entre otros [3]. De esta manera, constituye un proceso de suma importancia tanto para el personal administrativo de la institución como para los docentes y estudiantes que pertenecen a esta.

Sin embargo, la complejidad de manejar toda esta información ha llevado a la creación de diferentes proyectos en busca de mejores soluciones. Tal es el caso de Reyes [4], quien

propuso una aplicación web para la planificación académica de la Universidad de Guayaquil, con el fin de solventar esta problemática. Al igual que Leonardini [3], esta propuesta se centra en la búsqueda de disminuir el tiempo y los recursos necesarios para la planificación académica, dada la falta de automatización en los procesos. Es así como se puede observar la importancia en el uso de tecnologías que reemplacen los procesos manuales y contemplen todas las restricciones dictadas por las normas de la institución.

## **2.2 Planificación académica en la Facultad de Ingeniería de Sistemas de la Escuela Politécnica Nacional**

Cada semestre, el subdecano de la Facultad de Ingeniería de Sistemas (FIS) junto a los consejos respectivos llevan a cabo el proceso de planificación académica [1]. A diferencia de lo explicado en la sección 2.1, la planificación académica de la FIS se limita a la generación de horarios de cada asignatura tomando en cuenta los siguientes aspectos:

1. El listado de docentes que impartirá clases a los estudiantes de la FIS durante el semestre a planificar.
2. El listado de carreras ofertadas por la FIS.
3. El listado de asignaturas incluidas en la malla curricular de cada una de las carreras ofertadas por la FIS.
4. La jornada laboral que comprende los días y horas de la semana en la que los docentes imparten clases a sus estudiantes.
5. Los espacios físicos que corresponden a las diferentes aulas y laboratorios de la facultad.
6. Los grupos y subgrupos de cada nivel académico, es decir los paralelos a los que los estudiantes son asignados cada semestre.
7. Las horas no disponibles de cada docente, las cuales indican a qué momento no se les debe asignar actividades de docencia.
8. Un proceso de preplanificación académica que consiste en determinar el número de estudiantes probables para cada asignatura, de manera que se estime la cantidad de cupos necesarios para el semestre subsiguiente.



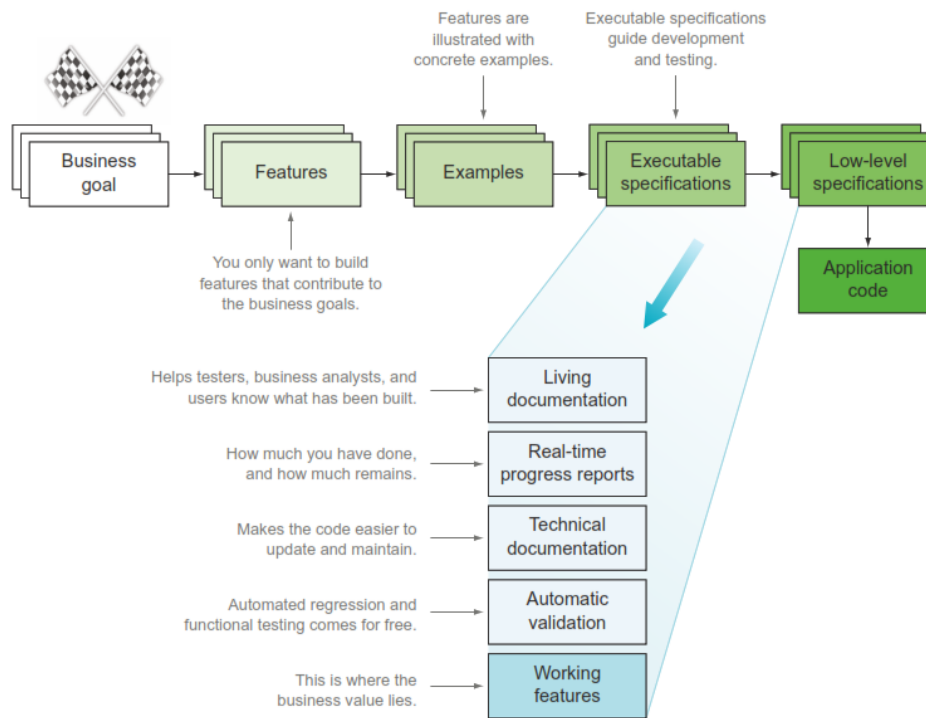
En base a esta información, la FIS genera los horarios de un semestre, los aprueba y distribuye. Estos horarios son creados mediante el *software* de escritorio FET de manera manual entre el subdecano y los coordinadores de carrera.

Por esta razón, cualquier solución propuesta para mejorar la planificación académica de la FIS deberá enfocarse en todos estos aspectos, al igual que hacer énfasis en una herramienta de *software* que realice la creación de horarios. Debido a que estos procesos son manuales en la actualidad, es necesario considerar el uso de tecnologías que realicen la planificación académica de la manera más automática posible.

### **2.3 Desarrollo dirigido por comportamiento**

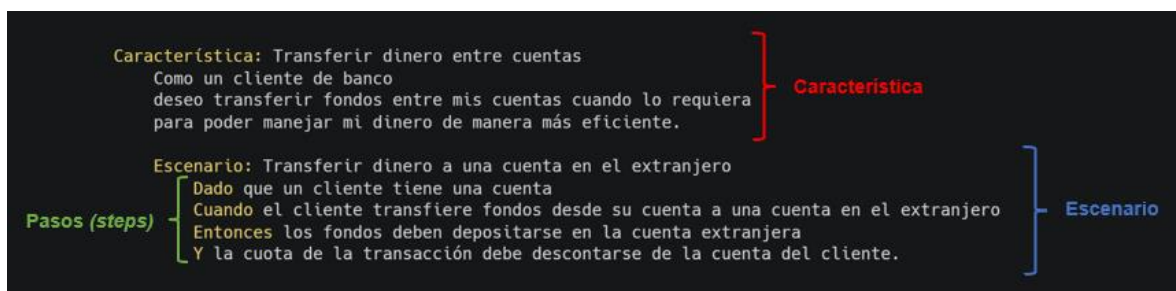
También conocido como BDD (*behavior-driven development*), es un conjunto de prácticas de ingeniería de *software* basadas en el desarrollo dirigido por pruebas (TDD) que busca que las entregas de *software* tengan más valor para el cliente [5]. Así, al igual que TDD, BDD indica que es necesario comenzar por crear pruebas por cada característica esperada del *software*, las cuales fallarán al no estar implementadas todavía. Sin embargo, Ferguson indica que muchos desarrolladores suelen tener problemas al decidir dónde empezar o qué pruebas hacer a continuación. Es por esto por lo que BDD se centra en escribir pruebas de aceptación como ejemplos que sean fáciles de entender para todo el equipo, a partir de los objetivos del negocio [6].

Ferguson [5] señala que es necesario crear solo aquellas características (*features*) que contribuyan con los objetivos de negocio. Las características son porciones tangibles de la funcionalidad necesaria para cumplir con estos objetivos [5]. Luego, se crean ejemplos (también denominados escenarios) en lenguaje natural que ilustren la funcionalidad respectiva, los cuales permiten la creación de especificaciones ejecutables que corresponden a las pruebas que guiarán el desarrollo [5]. Esto quiere decir, que en BDD es necesario crear únicamente aquellas pruebas que provean valor al negocio.



**Figura 2.** Principales actividades y salidas de BDD. Tomado de [5].

Las actividades indicadas en la Figura 2 se logran gracias a un lenguaje en común que facilite la comunicación entre los miembros del proyecto y los *stakeholders*, el cual suele denominarse como Gherkin [5]. Este lenguaje representa en lenguaje natural las características (Figura 3 en rojo) y escenarios (Figura 3 en azul). Estos escenarios están a su vez formados por pasos o *steps* (Figura 3 en verde).



**Figura 3.** Estructura de una característica y un escenario (Gherkin). Tomado y traducido de [5].

Para transformar este lenguaje en las especificaciones ejecutables que dirijan el desarrollo, es común el uso de distintas herramientas de BDD como por ejemplo Cucumber, del que se hablará más adelante.

### **3 METODOLOGÍA**

A continuación, se explicará a detalle cada etapa llevada a cabo por parte del equipo de desarrollo y el autor del componente del presente documento, durante la realización del proyecto.

#### **3.1 Investigación sobre la problemática**

El equipo de desarrollo del presente trabajo de integración curricular llevó a cabo una serie de actividades con el fin de comprender la problemática relacionada al proceso actual de planificación académica de la FIS. Estas actividades se detallan a continuación:

##### **Investigación de fuentes bibliográficas**

Como primer paso, el equipo de desarrollo investigó el concepto de “planificación académica”, los problemas comunes que este proceso puede tener en diferentes instituciones, y las soluciones propuestas por otras personas. Para esto, el equipo planteó previamente una estrategia de búsqueda de fuentes, así como un proceso de evaluación y priorización de estas. Este proceso se encuentra detallado en el ANEXO I.

Toda esta información fue recolectada (ANEXO II) y posteriormente analizada. De esta manera, se obtuvieron las siguientes ideas:

1. El concepto de “planificación académica” comprende diversos temas y es diferente en cada institución educativa. Puede incluir inscripciones, horarios, calificaciones, jornadas, etc. Por esta razón, se decidió que era indispensable conocer más a detalle el alcance de la planificación académica en la FIS, lo cual se explicó anteriormente en este documento.
2. Las fuentes investigadas [2], [3] y [4] señalaron que los problemas vienen de la complejidad de la información y de la falta de automatización de los procesos.

##### **Entrevistas a *stakeholders***

Con el fin de comprender las necesidades de los interesados (*stakeholders*) y el valor del negocio, el equipo de desarrollo y el director del presente trabajo realizaron diversas entrevistas a los involucrados. Estos últimos son:

1. José Francisco Lucio Naranjo, subdecano de la FIS;
2. Julio César Sandobalín Guamán, coordinador de la carrera de Ingeniería en software;
3. Josafá de Jesús Aguiar Pontes, coordinador de la carrera de Ingeniería de computación;
4. y Alex Javier Ulloa Arévalo, graduado de la FIS quien realizó previamente un proyecto bajo la misma problemática del presente documento.

A continuación se indican las entrevistas realizadas:

1. **Entrevista al subdecano y los coordinadores de carrera de la FIS.** Esta fue llevada a cabo por Carlos Anchundia, director del presente trabajo, con el fin de comprender cómo se realiza actualmente el proceso de planificación académica, sus problemas y qué es lo que necesitan los *stakeholders*. Se encuentra en el ANEXO XII.
2. **Entrevista al subdecano de la FIS.** Esta fue realizada por el equipo de desarrollo del proyecto con el objetivo de aclarar dudas referentes a los usuarios involucrados, restricciones a tomar en cuenta, proceso de preplanificación, entre otros asuntos que ayudaron a la redacción del presente documento. Se encuentra en el ANEXO XIII.
3. **Entrevista al tesista Alex Ulloa.** Esta última entrevista se realizó para conocer cuál fue la propuesta del tesista y cómo buscó dar solución a la problemática planteada. De igual manera, el entrevistado mencionó los problemas surgidos durante la realización de su trabajo de tesis, como lo son los errores que se debe controlar en el *software* FET y la falta de documentación de su trabajo. Se encuentra en el ANEXO XIV.

Gracias a esta información, el equipo de desarrollo determinó la problemática y el alcance de la solución a proponer, descritos en el documento de plan de trabajo de integración curricular. Así, se decidió también los roles de Scrum del proyecto, con el subdecano de la FIS como *product owner* y Alejandro Llanganate como *scrum master*. Esto con el fin de mantener una comunicación activa con el subdecano, y obtener su retroalimentación durante el desarrollo.

## 3.2 Análisis de requerimientos

### Historias de usuario

En esta etapa se representaron los requerimientos del proyecto mediante el uso de historias de usuario, a partir de la información recolectada. De esta manera, se formó el *product backlog* desarrollado en los *sprints* detallados en la sección 3.4. Este se lo puede visualizar a más detalle en el enlace del ANEXO III. A continuación, se muestran las historias correspondientes al componente del presente documento.

**Tabla 1.** Historias de usuario del componente “gestión de espacios, horas y preplanificación”.

Identificador en el backlog	Identificador en el presente documento	Título	Descripción
11	M1	Registro de espacios físicos	Como gestor de espacios físicos deseo registrar las diferentes aulas y laboratorios, ya sea de forma masiva a partir de un archivo o individualmente a través de un formulario, para distribuir los cursos de manera adecuada así evitando problemas por falta de infraestructura y evitar aglomeraciones.
12	M2	Registro de horas no disponibles del docente	Como docente deseo solicitar al jefe de departamento la consideración de mis horas no disponibles, seleccionadas en una matriz, por cada día de la semana para que el horario del periodo subsecuente que me sea asignado pueda acoplarse a mis actividades académicas, de docencia y de investigación.
13	M3	Registro de alumnos por asignatura	Como coordinador deseo ingresar a través de un formulario el número de estudiantes que cursarán cada una de las asignaturas del semestre actual, para poder planificar los horarios en base a los cupos que deberán estar disponibles para las asignaturas de dicho semestre.
13	M4	Preplanificación académica de horarios	Como coordinador deseo calcular el número de estudiantes que puedan matricularse en cada asignatura del semestre subsecuente a partir de la información de los estudiantes del semestre actual para contar con datos fiables que ayuden a disminuir la incertidumbre de los cupos que deberán estar disponibles el semestre subsecuente y determinar un número más certero para las asignaturas del siguiente nivel académico.

Por simplicidad, en el presente documento se hará referencia a cada historia usando el identificador de la segunda columna de la Tabla 1. Cabe mencionar que estas historias sufrieron diversos cambios como parte del proceso evolutivo de la metodología ágil. Sin embargo, en la Tabla 1 se muestran las versiones finales de cada historia.

### Estimación de las historias

Para estimar las historias del *product backlog*, dentro del equipo de desarrollo se utilizó la técnica del *planning poker* con apoyo del sitio web *Planning Poker Online*<sup>1</sup>. En esta reunión, cada integrante del equipo escogió un número de la serie de Fibonacci referente a la estimación de cada historia. Debido a las diferencias en la opinión de cada desarrollador, fue necesario llegar a un consenso sobre estos valores. La decisión de estas estimaciones fue evolucionando durante el desarrollo. Los valores usados para esto se muestran en la Figura 4.

Valor	Descripción
1	La HU tiene una complejidad baja, ya que su dependencia con el resto del sistema es bastante baja y su tiempo/esfuerzo de creación es relativamente bajo.
2	La HU maneja una variedad más amplia de aspectos a tomar en cuenta para poder llegar a satisfacer al cliente y depende de otras HU o procesos pequeños para poder ser completada.
3	La HU tiene una mayor complejidad, puesto que se espera una mayor cantidad de funcionalidades y operaciones para satisfacer al cliente. Además de relacionarse con la parte interactiva del usuario (núcleo de actividades fundamentales).

**Figura 4.** Valores de estimación usados para las historias de usuario.

Tomando en cuenta estos valores y las historias de la Tabla 1, se tiene la estimación por cada historia del componente del presente documento:

Historia de usuario	Estimación
M1: Registro de espacios físicos	2
M2: Registro de horas no disponibles del docente	3
M3: Registro de alumnos por asignatura	3
M4: Pre-planificación académica de horarios	3

**Figura 5.** Estimación de las historias de usuario del componente “gestión de espacios, horas y preplanificación”.

<sup>1</sup> Planning Poker Online, sitio web para facilitar el uso de planning poker entre varios usuarios. <https://planningpokeronline.com/>

## Planificación en *sprints*

En este punto, se consideró la planificación de tres *sprints* de tres semanas. Además, se contó con un *sprint* 0 en el que se realizaron todas las actividades de investigación de fuentes, entrevistas y análisis de requerimientos mencionados previamente en el documento.

Tomando en cuenta las estimaciones de la Figura 5, se decidió la planificación de cada *sprint* con sus respectivas historias de usuario indicada en la Tabla 2. Cabe mencionar que los identificadores usados para cada historia se lo usaron con el propósito de evitar confusiones entre los miembros del equipo de desarrollo. Esta planificación fue cambiando a lo largo del desarrollo como se detallará más adelante.

**Tabla 2.** Planificación inicial de cada historia de usuario por cada *sprint*.

Identificador	Sprint
M1	1
M2	1
M3	2
M4	3

### 3.3 Selección de herramientas y diseño de la arquitectura

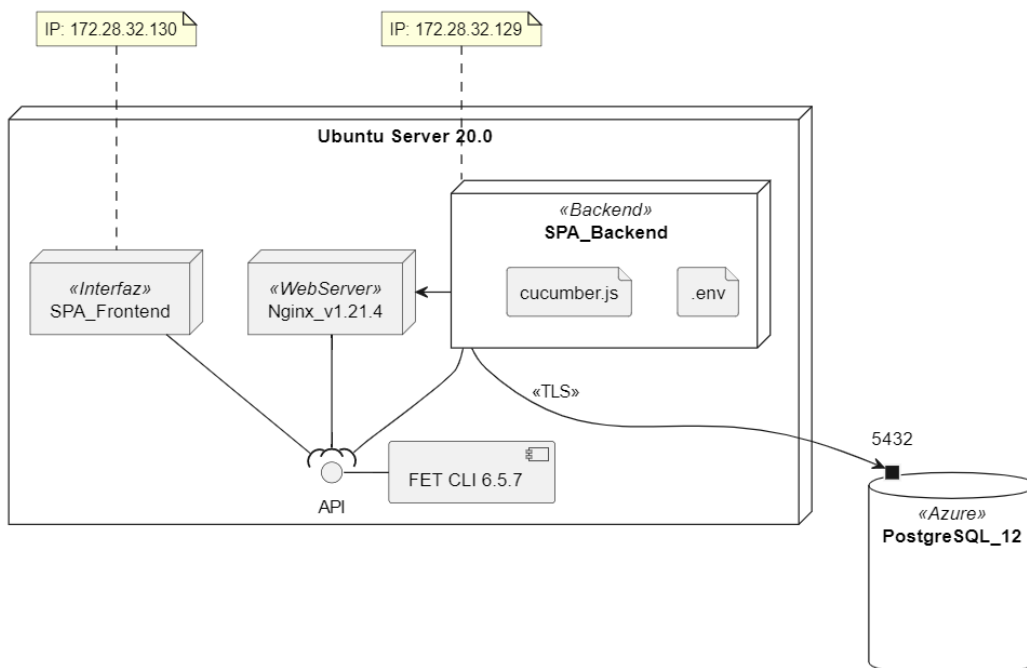
Se usó la herramienta Notion con el fin de documentar la información obtenida hasta este punto y la información futura. Esto se decidió debido a la facilidad que ofrece la herramienta para la creación y configuración de espacios de trabajo que se adapten al equipo. De esta manera, fue sencillo estructurar la información recolectada y la organización de las historias de usuario y sus actividades. El enlace de acceso a esta herramienta se encuentra en el ANEXO IV.

Para el desarrollo del *backend* del proyecto se decidió usar el *framework* NestJS para la creación de una API RESTful. Adicionalmente, se usó la librería TypeORM para la creación de una base de datos relacional y sus respectivas entidades. Esta base de datos se ubicó en un recurso de PostgreSQL en Microsoft Azure, haciendo uso de los beneficios como estudiante de la EPN. Finalmente, se usó la herramienta Swagger para la documentación de la API.

Por el lado del *frontend* se usó el *framework* Angular para la creación de una aplicación web de una sola página (SPA) que permite la organización por módulos y la reutilización

de componentes. De igual manera, se usó la librería Angular Material para agilizar la creación de componentes y para mantener una apariencia consistente en la aplicación. Por otro lado, la creación de prototipos de las interfaces gráficas fue con la herramienta Figma.

En la Figura 6, se muestra el diagrama propuesto para la arquitectura. Cabe señalar que las direcciones IP fueron provistas por la FIS. Sin embargo, estas no están siendo utilizadas en el código debido a que la aplicación no ha sido desplegada.



**Figura 6.** Diagrama de la arquitectura propuesta.

El archivo **cucumber.js** indicado en el diagrama se lo explicará más adelante (sección 3.4 – *sprint 1*). Mientras que el archivo **.env** (Figura 7) almacena valores de configuración del aplicativo, incluyendo las credenciales de acceso a la base de datos, una cadena secreta para la generación del token JWT y las credenciales del correo electrónico remitente.

```
.env
planificacion-academica-fis-backend > .env
1 SPA_DB_HOST=spa-server.postgres.database.azure.com
2 SPA_DB_PORT=5432
3 SPA_DB_USERNAME=
4 SPA_DB_PASSWORD=
5 SPA_DB_DATABASE=postgres
6 JWT_SECRET=
7
8 MAIL_HOST=smtplib-mail.outlook.com
9 MAIL_USER=
10 MAIL_PASSWORD=
11 MAIL_PORT=587
```

Las anotaciones en la imagen indican:

- Las líneas 1 a 5 (SPA\_DB\_HOST, SPA\_DB\_PORT, SPA\_DB\_USERNAME, SPA\_DB\_PASSWORD, SPA\_DB\_DATABASE) están agrupadas por una llave roja y etiquetadas como "Credenciales de acceso para la base de datos".
- La línea 6 (JWT\_SECRET) está etiquetada con una flecha verde como "Clave para la generación de tokens JWT".
- Las líneas 8 a 11 (MAIL\_HOST, MAIL\_USER, MAIL\_PASSWORD, MAIL\_PORT) están agrupadas por una llave azul y etiquetadas como "Credenciales de la dirección de correo electrónico".

**Figura 7.** Estructura del archivo **.env**.



Para la realización de las pruebas, se aplicó BDD en conjunto a la herramienta Cucumber, con la librería **cucumber-js**<sup>2</sup>. Esta herramienta lee las especificaciones desde archivos denominados *features* (con la sintaxis Gherkin), los examina en busca de *escenarios* y los ejecuta [6]. Cada uno de estos escenarios corresponde a una lista de pasos (*step definitions*) determinados por las palabras clave: “*dado*”, “*cuando*”, “*entonces*” e “*y*”. Por ejemplo, los pasos correspondientes al escenario de la Figura 3, escritos en el lenguaje TypeScript, se verían similares a la Figura 8. Como se puede apreciar, la cláusula “*dado*” corresponde a una función **Given**, la cláusula “*cuando*” a la función **When** y la cláusula “*entonces*” a la función **Then**. Cabe señalar que la cláusula “*y*” puede corresponder a cualquiera de estas tres funciones, dependiendo de cuál cláusula le preceda. En este caso, dado que la cláusula “*entonces*” precede a la cláusula “*y*” en la Figura 3, entonces esta última corresponde a la función **Then**.

```

Dado {
  Given('que un cliente tiene una cuenta', function () {
    // Se escribe código que convierta la frase de arriba en acciones concretas
    return 'pending';
  });
}

Cuando {
  When('el cliente transfiere fondos desde su cuenta a una cuenta en el extranjero', function () {
    // Se escribe código que convierta la frase de arriba en acciones concretas
    return 'pending';
  });
}

Entonces {
  Then('los fondos deben depositarse en la cuenta extranjera', function () {
    // Se escribe código que convierta la frase de arriba en acciones concretas
    return 'pending';
  });
  Then('la cuota de la transacción debe descontarse de la cuenta del cliente', function () {
    // Se escribe código que convierta la frase de arriba en acciones concretas
    return 'pending';
  });
}
Cláusula "y"

```

Figura 8. Estructura inicial de los pasos de un escenario utilizando Cucumber y TypeScript.

### 3.4 Desarrollo de los *sprints*

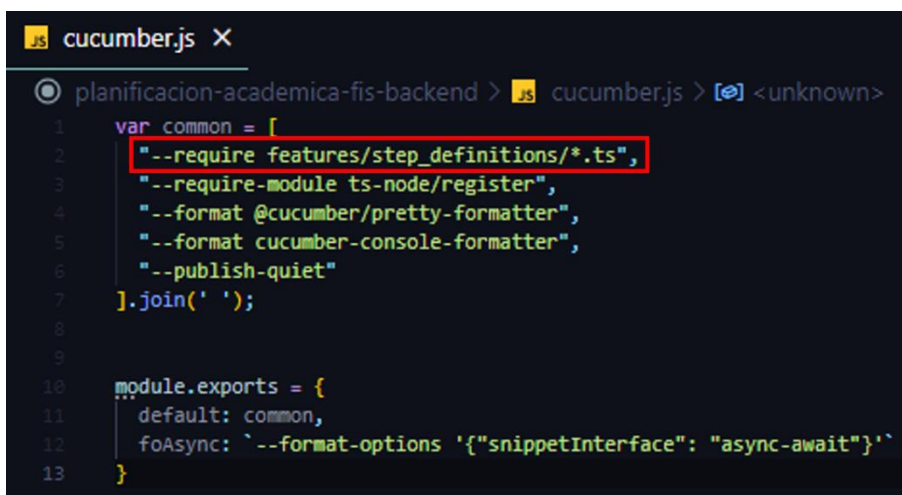
Después de lo realizado en el *sprint 0*, mencionado anteriormente, se procede a indicar las actividades realizadas en cada uno de los *sprints* del proyecto.

#### **Sprint 1**

En este *sprint* se comenzó por la creación de los repositorios para el *frontend* y *backend* (ANEXO V y ANEXO VI, respectivamente). En el *backend* fue necesario crear un archivo denominado **cucumber.js** (como se muestra en la Figura 6) con el fin de poder hacer uso de la herramienta Cucumber en NestJS. Este contiene diversos comandos de

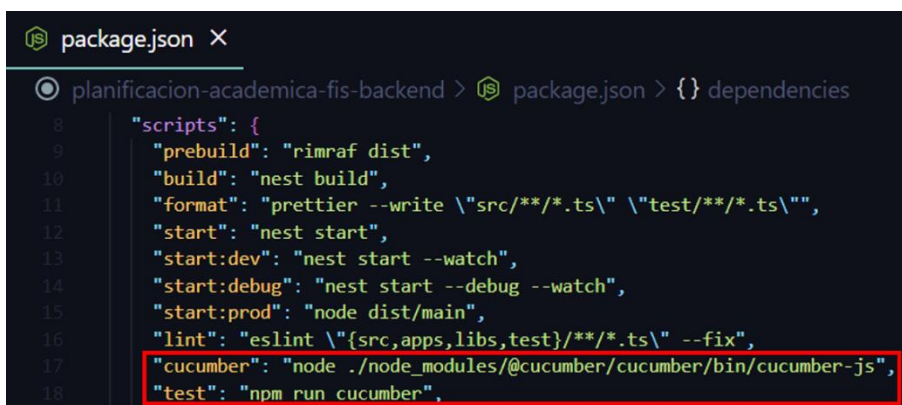
<sup>2</sup> Cucumber-js, librería basada en Cucumber para su uso en JavaScript.  
<https://github.com/cucumber/cucumber-js>

configuración, entre los cuales uno indica la ubicación de los *features* y *steps* (Figura 9, segunda línea).



```
1 var common = [  
2   "--require features/step_definitions/*.ts",  
3   "--require-module ts-node/register",  
4   "--format @cucumber/pretty-formatter",  
5   "--format cucumber-console-formatter",  
6   "--publish-quiet"  
7 ].join(' ');  
8  
9  
10 module.exports = {  
11   default: common,  
12   foAsync: `--format-options '{"snippetInterface": "async-await"}'`  
13 }
```

Figura 9. Archivo cucumber.js usado para la ejecución de pruebas.



```
8 "scripts": {  
9   "prebuild": "rimraf dist",  
10  "build": "nest build",  
11  "format": "prettier --write \"src/**/*.ts\" \"test/**/*.ts\"",  
12  "start": "nest start",  
13  "start:dev": "nest start --watch",  
14  "start:debug": "nest start --debug --watch",  
15  "start:prod": "node dist/main",  
16  "lint": "eslint \"{src,apps,libs,test}/**/*.ts\" --fix",  
17  "cucumber": "node ./node_modules/@cucumber/cucumber/bin/cucumber-js",  
18  "test": "npm run cucumber",  
19 }
```

Figura 10. Configuración del archivo package.json del *backend* para la ejecución de pruebas.

Las pruebas se ubicaron dentro del repositorio del *backend* en un directorio denominado **features** (Figura 11). Aquí, se crearon los archivos con extensión *.feature* correspondientes a cada característica del proyecto (obtenidas de las historias de usuario). Dentro de este directorio, se tiene una carpeta denominada **step\_definitions** donde se ubican los archivos de las pruebas en TypeScript. Por otro lado, en el archivo **package.json** se colocaron los comandos necesarios para la ejecución como se ve en las líneas 17 y 18 de la Figura 10. De esta manera, la ejecución de las pruebas se la realiza mediante el siguiente comando desde el directorio raíz del *backend*:

```
npm test .\features\[nombre del archivo .feature]
```



Figura 11. Estructura de la carpeta de pruebas (*features* y *step definitions*).

Por otro lado, el equipo de desarrollo presentó problemas al momento de ejecutar cada *step definition* de Cucumber debido a que para hacer uso de los controladores y servicios de NestJS era necesario primero levantar el servidor. Esto no se hacía automáticamente al ejecutar las pruebas, razón por la cual se hizo uso de las funciones **BeforeStep** y **AfterStep** (Figura 12) provistas por la librería cucumber-js. Estas funciones se ejecutan antes y después de todas las pruebas, razón por la cual se utilizaron para levantar y cerrar el servidor.

```

beforeStep_afterStep.ts X
planificacion-academica-fis-backend > features > step_definitions > beforeStep_afterSt
1  import { ValidationPipe } from '@nestjs/common';
2  import { NestFactory } from '@nestjs/core';
3  import { AppModule } from '../src/app.module';
4  import configuracion from '../src/config/configuracion';
5  import { AfterStep, BeforeStep } from '@cucumber/cucumber';
6
7  BeforeStep(async function () {
8    // Levantar aplicacion
9    const app = await NestFactory.create(AppModule, {
10     logger: false,
11   });
12
13   // Para que se apliquen Los DTOs
14   app.useGlobalPipes(new ValidationPipe({ whitelist: true, transform: true }));
15
16   await app.listen(configuracion().http.port);
17
18   this.app = await app;
19 });
20
21 AfterStep(async function () {
22   // Cerrar aplicacion
23   this.app.close();
24 });

```

Figura 12. Archivo con las funciones *BeforeStep* y *AfterStep* necesarias para la ejecución de pruebas. Se muestra la versión final de este archivo.

Debido a los problemas mencionados, al tiempo de aprendizaje de las herramientas indicadas en la sección 3.3 y la carga académica, no fue posible al final de este *sprint* cumplir con la entrega planificada (Tabla 2). Por esta razón, se realizó una nueva planificación tomando en cuenta los siguientes *sprints*. De esta manera se llegó a la siguiente planificación:

**Tabla 3.** Nueva planificación de las historias de usuario por *sprint*.

Identificador	Sprint
M1	2
M2	2
M3	3

Como se puede observar en la Tabla 3, la historia de usuario M4 fue descartada. Esto se decidió debido a que en la reunión informativa del semestre 2022-A (ANEXO VII), el subdecano José Lucio indicó que el proceso de preplanificación se lo llevaría a cabo con el uso de un *software* desarrollado por Josafá Aguiar, coordinador de la carrera de Ingeniería de computación. Esto último se confirmó posteriormente en una reunión (ANEXO VIII) por parte del coordinador de la carrera de Ingeniería de software, Julio Sandobalín, indicando que la preplanificación académica se descartaría del proyecto debido a su complejidad y la nueva solución de Josafá Aguiar.

## ***Sprint 2***

### **Historia M1: Registro de espacios físicos**

Esta historia consiste en el registro de los diferentes espacios físicos que serán utilizados para la generación de horarios por parte de un usuario gestor de espacios físicos. De esta manera, se tomaron en cuenta dos escenarios:

- 1. Se agrega un espacio físico.** Cuando el gestor de espacios físicos registra a través de un formulario un nuevo espacio físico, este se almacena en la base de datos únicamente si su nombre no se ha repetido.
- 2. Se agrega un archivo con múltiples espacios físicos.** El gestor de espacios físicos tiene la posibilidad de subir un archivo CSV con la información de un espacio físico (nombre, tipo, facultad, aforo). De este archivo, se registran solo aquellos espacios físicos cuyos nombres no estén repetidos.

Tomando en cuenta esto, se creó el *feature* correspondiente para las pruebas con BDD como se muestra en el ANEXO IX. Luego, se procedió a la creación de los *steps* correspondientes. De esta manera, se determinó la necesidad de crear las entidades para: espacios físicos, tipos de aula y facultades. Asimismo, se estableció que los servicios de la API deberían controlar que espacios físicos no se dupliquen. Los diferentes métodos del controlador generado en la API se encuentran documentados con Swagger como se muestra en el ANEXO X.

Para la ejecución de estos es necesario que el usuario cuente con el rol de “gestor de espacios físicos”. Por esta razón, se crearon roles en la base de datos como se ve en la Figura 13.

	id [PK] uuid	nombre character varying
1	9465fa4e...	COORDINADOR
2	1903845...	DOCENTE
3	0f19e08e...	SUBDECANO
4	1b88827...	GESTOR_ESPACIOS_FISICOS
5	b69c0146...	JEFE_DE_DEPARTAMENTO

**Figura 13.** Roles del proyecto vistos en la tabla “rol”.

Por otro lado, los archivos usados para el segundo escenario se encuentran en el directorio **documents\_test** (Figura 11) y tienen la estructura mostrada en la Figura 14. Como se puede observar, estos archivos deben tener los datos “nombre”, “tipo”, “facultad” y “aforo” (en este orden) separados por “;”.

```

espacios_fisicos_prueba1.csv
planificacion-academica-fis-backend > features > documents_test > espacios_fisicos_prueba1.csv
1 nombre;tipo;facultad;aforo
2 DeltaPrueba;LABORATORIO;Facultad Prueba Espacios Fisicos;25
3 LambdaPrueba;LABORATORIO;Facultad Prueba Espacios Fisicos;25
4 SIS401Prueba;AULA;Facultad Prueba Espacios Fisicos;25
5 SIS402Prueba;AULA;Facultad Prueba Espacios Fisicos;30
6 SIS403Prueba;AULA;Facultad Prueba Espacios Fisicos;15

```

**Figura 14.** Ejemplo de archivo CSV para la creación de múltiples espacios físicos.

Con esta implementación, se puede observar (Figura 15 y Figura 16) que la ejecución de las pruebas es exitosa:

```

=====
Característica: Registro de espacios físicos

Como gestor de espacios físicos deseo registrar las diferentes aulas y laboratorios, ya sea de forma
masiva a partir de un archivo o individualmente a través de un formulario, para distribuir los cursos
de manera adecuada así evitando problemas por falta de infraestructura y evitar aglomeraciones.

Esquema del escenario: Se agrega un espacio físico
✓ Dado que existe un espacio físico llamado "BetaPrueba"
✓ Cuando se agrega un espacio físico llamado "AlfaPrueba"
✓ Entonces al consultar la base de datos se observan "2" registros.

=====

Característica: Registro de espacios físicos

Como gestor de espacios físicos deseo registrar las diferentes aulas y laboratorios, ya sea de forma
masiva a partir de un archivo o individualmente a través de un formulario, para distribuir los cursos
de manera adecuada así evitando problemas por falta de infraestructura y evitar aglomeraciones.

Esquema del escenario: Se agrega un espacio físico
✓ Dado que existe un espacio físico llamado "BetaPrueba"
✓ Cuando se agrega un espacio físico llamado "BetaPrueba"
✓ Entonces al consultar la base de datos se observan "1" registros.

2 escenarios (2 passed)
6 steps (6 passed)
0m09.918s (executing steps: 0m09.785s)
Run finished at 2022/09/04 08:13

```

Figura 15. Ejecución de las pruebas del primer escenario de la historia M1.

```

=====
Característica: Registro de espacios físicos

Como gestor de espacios físicos deseo registrar las diferentes aulas y laboratorios, ya sea de forma
masiva a partir de un archivo o individualmente a través de un formulario, para distribuir los cursos
de manera adecuada así evitando problemas por falta de infraestructura y evitar aglomeraciones.

Esquema del escenario: Se agrega un archivo con múltiples espacios físicos
✓ Dado que existe un espacio físico llamado BetaPrueba
✓ Y existe un espacio físico llamado AlfaPrueba
✓ Y existe un espacio físico llamado GammaPrueba
✓ Y existe un espacio físico llamado SI5502Prueba
✓ Cuando se importe el archivo "espacios_fisicos_prueba1.csv"
✓ Entonces al consultar la base de datos se observan "9" espacios físicos.

=====

Característica: Registro de espacios físicos

Como gestor de espacios físicos deseo registrar las diferentes aulas y laboratorios, ya sea de forma
masiva a partir de un archivo o individualmente a través de un formulario, para distribuir los cursos
de manera adecuada así evitando problemas por falta de infraestructura y evitar aglomeraciones.

Esquema del escenario: Se agrega un archivo con múltiples espacios físicos
✓ Dado que existe un espacio físico llamado BetaPrueba
✓ Y existe un espacio físico llamado AlfaPrueba
✓ Y existe un espacio físico llamado GammaPrueba
✓ Y existe un espacio físico llamado SI5502Prueba
✓ Cuando se importe el archivo "espacios_fisicos_prueba2.csv"
✓ Entonces al consultar la base de datos se observan "7" espacios físicos.

2 escenarios (2 passed)
12 steps (12 passed)
0m16.188s (executing steps: 0m16.040s)
Run finished at 2022/09/04 08:25

```

Figura 16. Ejecución de las pruebas del segundo escenario de la historia M1.

Con esto, se procede a la implementación del *frontend* del módulo de gestión de espacios físicos. Para esto se empezó por la creación de un prototipo de la interfaz gráfica que se muestra en la Figura 17.



**Figura 17.** Prototipo del módulo de gestión de espacios físicos.

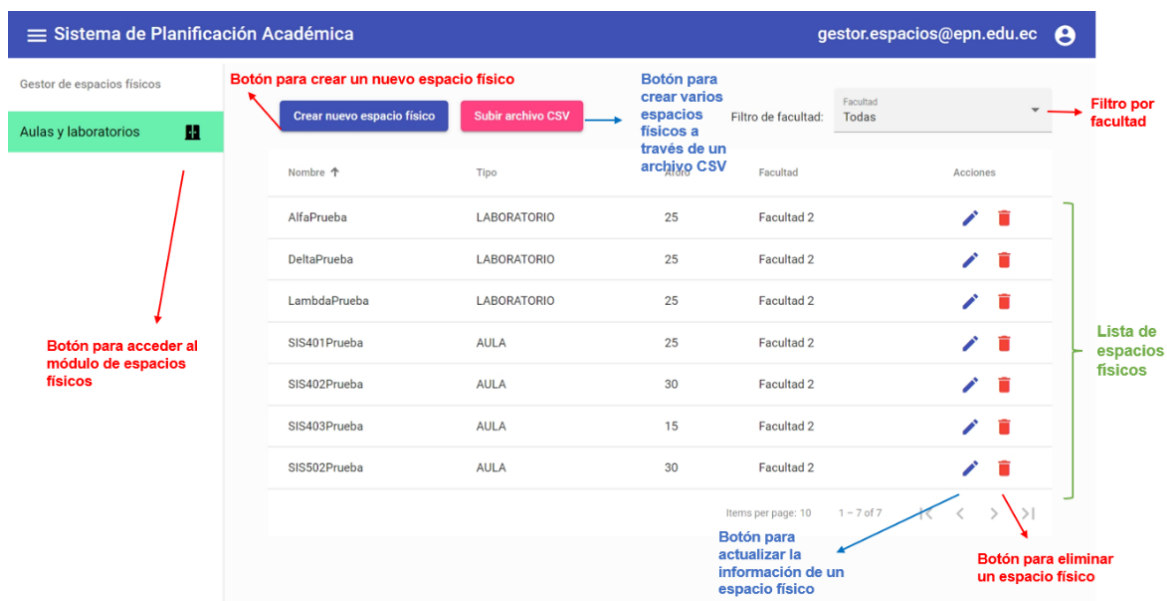
En este prototipo (y los siguientes) se puede notar una barra lateral con botones para acceder a cada módulo. A la derecha de esta barra se tiene el contenido principal del módulo. En el caso de la Figura 17, se cuenta con un botón para crear nuevos espacios físicos, un botón “Subir archivo CSV” para crear varios espacios físicos al cargar un archivo con la estructura de la Figura 14. Por otro lado, se muestra la lista de espacios físicos, donde cada uno cuenta con botón para actualizar su información y otro para eliminarlo.

Posteriormente, se continuó con la creación de los distintos servicios que realizan peticiones a la API ya creada (*login*, obtención de espacios físicos, etc.). Posteriormente, se creó el módulo correspondiente como se indica en la Figura 18. Esta organización será utilizada en los demás módulos mencionados en el presente documento.



**Figura 18.** Organización usada para los módulos en *frontend*.

A continuación, se muestra la interfaz gráfica de la pantalla principal para la historia M1. En esta, se permite visualizar todos los espacios físicos existentes, filtrarlos o eliminarlos (Figura 19). De igual manera, se pueden crear y actualizar registros. Igualmente, se cuenta con un botón denominado “Subir archivo CSV” para cargar múltiples registros a la vez.



**Figura 19.** Interfaz gráfica del módulo de gestión de espacios físicos. Pantalla principal.

Cabe señalar que sí se tomaron en cuenta mensajes de error, de confirmación e información correspondientes a las diferentes respuestas obtenidas por el servidor. Sin embargo, por motivo de extensión no se los presenta en este documento.

### Historia M2: Registro de horas no disponibles del docente

Durante el *sprint 2*, se tomó en cuenta el siguiente escenario:

- **Se agregan horas no disponibles de un docente.** Cuando un docente indica en pantalla cuáles son sus horas no disponibles para actividades de docencia, estas se registran para ser usadas posteriormente por el *software* FET.

Con esto en cuenta, se creó el *feature* para esta historia (ANEXO IX). Luego, se procedió a la creación de los *steps* correspondientes. Así, se determinó la necesidad de crear las entidades para: horas no disponibles, docentes, jornadas laborales y semestres. De esta manera, dependiendo de la jornada laboral establecida para el semestre en curso, un docente podrá elegir las horas no disponibles que desee.



En la Figura 20 se muestra la ejecución de la prueba creada.

```
Característica: Registro de horas no disponibles del docente

Como docente deseo señalar en una matriz mis horas no disponibles por
cada día de la semana para que el horario del periodo subsecuente que
me sea asignado pueda acoplarse a mis actividades académicas, de
docencia y de investigación.

Escenario: Se agregan horas no disponibles de un docente
✓ Dado que existe un docente con correo docente.prueba@epn.edu.ec
✓ Y un semestre con jornada laboral de lunes a sábado desde las 7h hasta las 20h con almuerzo a las 13h
✓ Cuando se agregan como horas no disponibles el día Martes de 14 a 15
✓ Y el día Martes de 15 a 16
✓ Y también el día Jueves de 7 a 8
✓ Entonces al consultar la base de datos de horas no disponibles se observan 3 registros.
```

Figura 20. Ejecución de la prueba del escenario de la historia M2. Versión del *sprint* 2.

Posteriormente, se continuó con la implementación del *frontend* del módulo de horas no disponibles. De igual manera que con la historia anterior, se creó un prototipo de la interfaz gráfica.

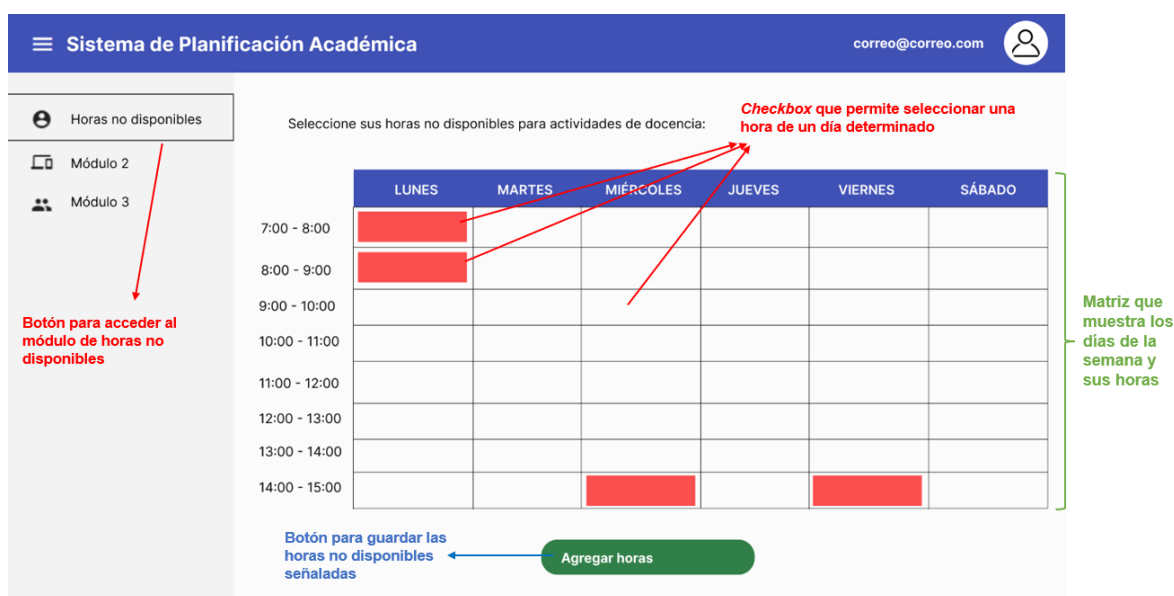
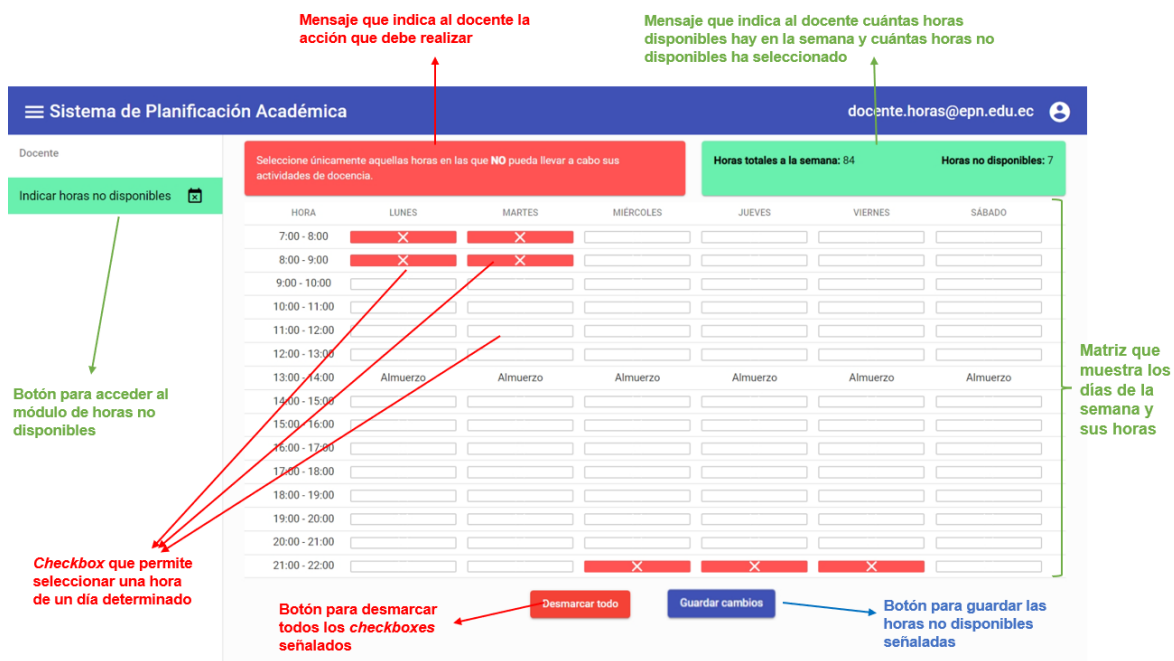


Figura 21. Prototipo del módulo de horas no disponibles.

En el prototipo de la Figura 21 se muestra una matriz que indica los días de la semana y las horas de la jornada laboral. Se ideó de esta manera para que el docente seleccione las casillas correspondientes a sus horas no disponibles y posteriormente presione el botón de la parte inferior para registrar estas horas. La Figura 22 muestra la implementación de esta interfaz. Adicionalmente a lo presentado en el prototipo, se cuenta en la parte superior con mensajes informativos para el docente. Asimismo, se cuenta con un botón “Desmarcar todo” en la parte inferior que permite retirar las casillas seleccionadas, de manera que facilite al docente el uso de la interfaz.



**Figura 22.** Interfaz gráfica del módulo de horas no disponibles. Pantalla de selección. Versión *sprint 2*.

### **Sprint 3**

En este punto del desarrollo surgieron cambios en los roles de Scrum. Esto se debió a que durante el periodo 2022-A, se realizaron las elecciones de las autoridades de la FIS. Fue así como el anterior subdecano, José Lucio, ocupó el puesto de decano de la facultad. Por esta razón, y según el Estatuto de la Escuela Politécnica Nacional [1], quien pasa a estar a cargo de la planificación académica de la FIS es la nueva subdecano, Monserrate Intriago. Sin embargo, debido a que el coordinador de la carrera de Ingeniería de software tiene más experiencia sobre la problemática a resolver, Julio Sandobalín estuvo dispuesto a tomar el puesto de *product owner*. Todo esto se decidió en la reunión del ANEXO VIII.

#### **Cambio en la historia M2: Registro de horas no disponibles del docente**

Tras la reunión mencionada, surgieron cambios en la historia M2. El *product owner* indicó que las horas no disponibles de un docente deben pasar primero por un proceso de solicitud. Luego, el jefe de departamento es quien decide si acepta o rechaza esta solicitud, de tal manera que estos datos se usen o no en la generación de horarios con el *software* FET. Por esta razón, se hicieron cambios en el *feature* correspondiente tomando en cuenta dos escenarios:

1. **Se aprueba la solicitud de horas no disponibles de un docente.** Busca probar que el jefe de departamento tenga la posibilidad de aprobar una solicitud y que al hacerlo, esta se registre como aprobada y se notifique al docente mediante correo electrónico.
2. **Se rechaza la solicitud de horas no disponibles de un docente.** Este escenario busca que el jefe de departamento pueda rechazar una solicitud y que al hacerlo, esta sea eliminada y se notifique al docente mediante correo electrónico.

Estos cambios en el feature se ven reflejados en el ANEXO IX. Ahora, el *feature* incluye el concepto de solicitud. En base a esto, se incluyó el rol “jefe de departamento” (Figura 13) y una nueva entidad correspondiente a las solicitudes. Cada solicitud le pertenece a un docente y al semestre en el que se realizó la solicitud. De igual manera, se incluyó un valor que indica el estado de la solicitud (“aprobada” o “por revisar”).

En la documentación de los métodos de la API de esta historia (ANEXO X) se puede observar que se cuenta con una petición GET que permite obtener todas aquellas horas no disponibles que hayan sido aprobadas. Con esta petición, se pueden obtener los datos que se van a usar para la generación de horarios.

La Figura 23 muestra la ejecución de las pruebas de los nuevos escenarios.

```

=====
Característica: Registro de horas no disponibles del docente

Como docente deseo solicitar al jefe de departamento la consideración de mis horas no disponibles,
seleccionadas en una matriz, por cada día de la semana para que el horario del periodo subsecuente
que me sea asignado pueda acoplarse a mis actividades académicas, de docencia y de investigación.

Escenario: Se aprueba la solicitud horas no disponibles de un docente
✓ Dado que existe un docente con correo mahatma.quijano@epn.edu.ec
✓ Y un semestre con jornada laboral de lunes a sábado desde las 7h hasta las 20h con almuerzo a las 13h
✓ Cuando se solicitan como horas no disponibles el día Martes de 14 a 15
✓ Y el día Martes de 15 a 16
✓ Y también el día Jueves de 7 a 8
✓ Y el jefe de departamento aprueba la solicitud
✓ Entonces al consultar la base de datos de horas no disponibles se observan 3 registros
✓ Y el docente recibe un correo electrónico indicando la aprobación de su solicitud.

=====
Característica: Registro de horas no disponibles del docente

Como docente deseo solicitar al jefe de departamento la consideración de mis horas no disponibles,
seleccionadas en una matriz, por cada día de la semana para que el horario del periodo subsecuente
que me sea asignado pueda acoplarse a mis actividades académicas, de docencia y de investigación.

Escenario: Se rechaza la solicitud horas no disponibles de un docente
✓ Dado que existe un docente con correo mahatma.quijano@epn.edu.ec
✓ Y un semestre con jornada laboral de lunes a sábado desde las 7h hasta las 20h con almuerzo a las 13h
✓ Cuando se solicitan como horas no disponibles el día Martes de 14 a 15
✓ Y el día Martes de 15 a 16
✓ Y también el día Jueves de 7 a 8
✓ Y el jefe de departamento rechaza la solicitud
✓ Entonces al consultar la base de datos de horas no disponibles se observan 0 registros
✓ Y el docente recibe un correo electrónico indicando el rechazo de su solicitud.

2 escenarios (2 passed)
16 steps (16 passed)
0m25.001s (executing steps: 0m24.844s)

```

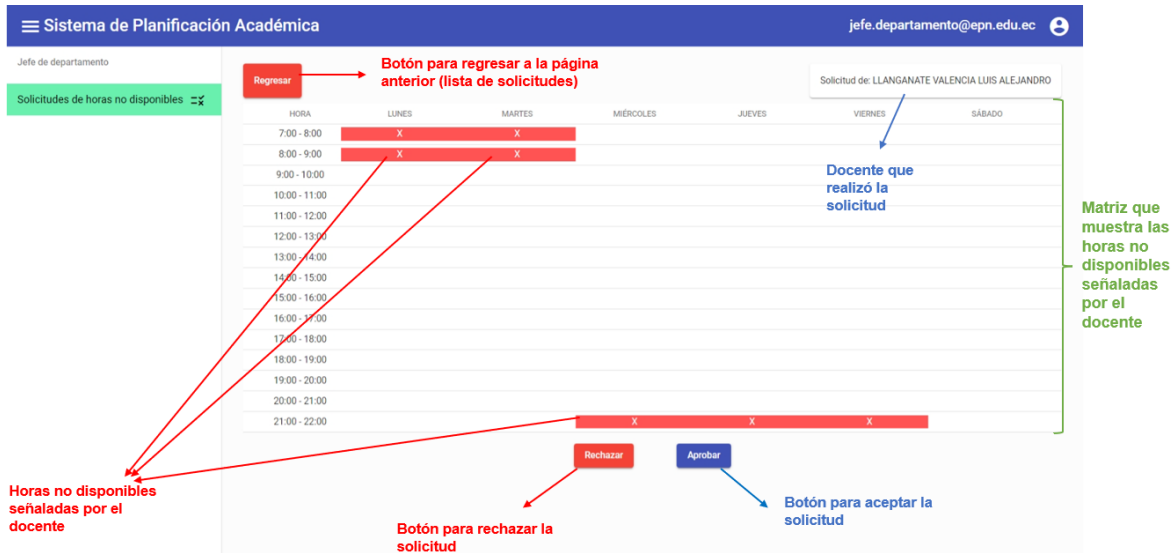
**Figura 23.** Ejecución de las pruebas de los dos escenarios de la historia M2.

Posteriormente, se implementaron los cambios necesarios en el *frontend*. A diferencia de la interfaz creada anteriormente (Figura 22), se indica al docente mediante el nombre del botón “Solicitar horas no disponibles” (Figura 24) que lo seleccionado corresponde a una solicitud. De igual manera, en la barra lateral izquierda se indica que se trata de una solicitud.

**Figura 24.** Interfaz gráfica del módulo de horas no disponibles. Pantalla de solicitud.  
Versión *sprint 3*.

Además, se creó un nuevo componente (Figura 25) en el que un usuario con el rol “jefe de departamento” puede visualizar las solicitudes creadas por cada docente. Luego, puede acceder a una solicitud específica y decidir si la aprueba o rechaza (Figura 26).

**Figura 25.** Interfaz gráfica del módulo de horas no disponibles. Pantalla de lista de solicitudes para el jefe de departamento.

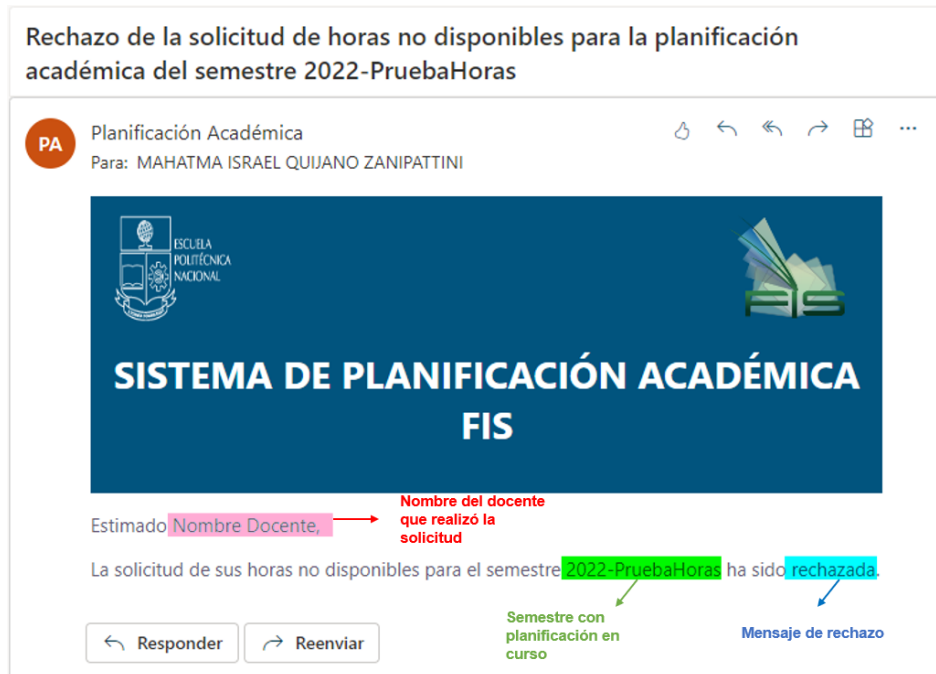


**Figura 26.** Interfaz gráfica del módulo de horas no disponibles. Pantalla para aprobar o rechazar una solicitud específica.

Esta decisión se notifica al docente mediante un correo electrónico de aprobación (Figura 27) o de rechazo (Figura 28).



**Figura 27.** Ejemplo de correo de aprobación de la solicitud de horas no disponibles.



**Figura 28.** Ejemplo de correo de rechazo de la solicitud de horas no disponibles.

### Historia M3: Registro de alumnos por asignatura

Esta historia consiste en registrar el número de alumnos por cada asignatura durante un semestre. Estos datos sirven para la creación de actividades en la generación de horarios con el *software* FET. En este caso, el usuario encargado de realizar esta acción tiene el rol de “coordinador”. Es importante mencionar que esta historia constituye el punto de conexión con el proceso de preplanificación académica, puesto que este último consiste en vincular el número de estudiantes estimado para un semestre.

Para esta historia se planteó su respectivo *feature* (ANEXO IX), con el siguiente escenario:

- **Se indica el número de estudiantes de un semestre en curso.** Dado que hay un semestre cuya planificación está en curso y se han registrado varias asignaturas, cuando el coordinador de carrera indique el número de estudiantes por cada asignatura, estas se registran para ser usadas posteriormente en la generación de horarios.

Posteriormente, se crearon los *steps* correspondientes. Así, se determinó la necesidad de crear una entidad para el número de estudiantes, que se asocie al semestre en curso y a la asignatura correspondiente. El ANEXO X muestra la documentación correspondiente a la API.

Una vez implementado el código del *backend*, se procede a ejecutar la prueba creada (Figura 29).

```
=====
Característica: Registro de alumnos por asignatura

Como coordinador deseo ingresar a través de un formulario el número de estudiantes que cursarán cada una
de las asignaturas del semestre actual, para poder planificar los horarios en base a los cupos que deberán
estar disponibles para las asignaturas de dicho semestre.

Escenario: Se indica el número de estudiantes de un semestre en curso
✓ Dado que existe un semestre cuya planificación está en curso
✓ Y existen las asignaturas de AsignaturaPrueba1, AsignaturaPrueba2, AsignaturaPrueba3 y AsignaturaPrueba4
✓ Cuando se indica el número de estudiantes 27, 10, 40 y 5, respectivamente,
✓ Entonces al consultar la base de datos de número de estudiantes se observan 4 registros.

1 escenario (1 passed)
4 steps (4 passed)
0m07.790s (executing steps: 0m07.675s)
```

Figura 29. Ejecución de la prueba del escenario de la historia M3.

A continuación, se procedió a la implementación del *frontend*. Para esto se creó un nuevo módulo que va a poder ser accedido por un usuario con rol “coordinador”. El componente principal para este módulo consiste en una lista de todas las asignaturas registradas, donde podrá ingresar el número de estudiantes por cada una de ellas (Figura 30).

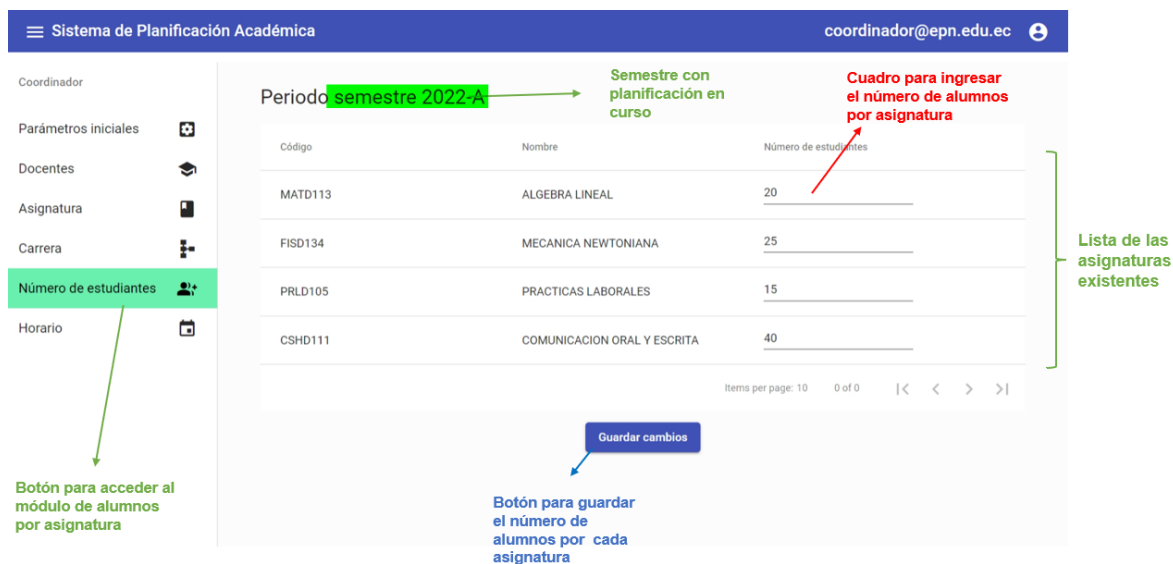


Figura 30. Interfaz gráfica del módulo de registro de alumnos por asignatura. Pantalla de registro.

## 4 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

### 4.1 Resultados

El proceso investigativo realizado anteriormente (sección 3.1) permitió que el equipo de desarrollo comprenda el concepto de “planificación académica”, así como el alcance que esto tiene dentro de la FIS. Es gracias a esto que se pudo entender las reglas de negocio alrededor de la problemática. De esta manera, fue posible establecer los objetivos tanto del proyecto como del componente del presente documento. Esto con el fin de esclarecer aquello que es de valor para los clientes.

Asimismo, la técnica de historias de usuario facilitó el entendimiento de los requerimientos obtenidos tras el proceso investigativo. Fue así como se pudo realizar la estimación pertinente que permitió con la que se realizó la planificación. A pesar de que a lo largo del desarrollo ha habido cambios para la historia M2 (como se explica en el sprint 3 de la sección 3.4), así como también la decisión de descartar la historia M4, el equipo ha sabido adaptarse. Esto es gracias al uso de la metodología ágil y su naturaleza de aceptar cambios. Así, se realizó una nueva planificación de las historias con el fin de entregar valor al cliente (Tabla 3).

Por otro lado, las herramientas utilizadas (sección 3.3) agilizaron el proceso de desarrollo. En particular, la herramienta Notion permitió al equipo de desarrollo gestionar sus historias de usuario y sus actividades. De esta forma, se cuenta cierto grado de trazabilidad a lo largo del proyecto. Mientras que el uso de BDD y la herramienta Cucumber permitieron enfocar el desarrollo en aquello que da valor al cliente. Gracias a esto, se crearon las pruebas necesarias que guiaron el desarrollo.

La Figura 31 presenta un diagrama comparativo de aquellas historias de usuario planificadas inicialmente y aquellas que sí se realizaron. Como se puede observar, hay una historia que no se realizó. Esta es la historia M4 la cual, como se explicó en la sección 3.4 en el *sprint 1*, fue descartada debido a la complejidad del proceso de preplanificación académica, siendo el *product owner* quien indicó que no se la tome en cuenta. Sin embargo gracias a lo realizado en la historia M3, es posible vincular el proceso actual (manual con el uso de Microsoft Forms) o un futuro proceso de preplanificación, con el componente de esta historia.





**Figura 31.** Diagrama comparativo de historias de usuario planificadas y realizadas.

## 4.2 Conclusiones

El componente “gestión de espacios, horas y preplanificación” consiste en una aplicación web intuitiva y fácil de usar que permite su uso de manera concurrente para diferentes usuarios. De esta manera, cada usuario puede colaborar para la planificación académica a diferencia del proceso actual con el *software* FET.

El uso concurrente por parte de varios usuarios es posible debido al uso de tecnologías web con las que se estableció un *backend* y un *frontend*. De esta manera, la comunicación se realiza a través de una API y los datos necesarios para la generación de horarios cuentan con persistencia en una base de datos. Si bien la aplicación creada no ha sido desplegada, esto se debe a que no se cuenta aún con uno de los componentes de otro integrante del equipo de desarrollo. Sin embargo, el uso de la API permite a cada usuario manipular los datos en la misma base de datos.

Por otro lado, la creación del módulo de horas no disponibles agiliza el proceso en comparación al realizado actualmente. El hecho de incluir al docente como otro usuario del sistema, permite que este realice la solicitud de horas no disponibles de una manera más sencilla. Igualmente, el jefe de departamento puede revisar todas estas solicitudes en el sistema y aprobarlas o rechazarlas. Adicionalmente, con el docente como usuario se pueden incluir mejoras a futuro como por ejemplo que cada docente pueda observar su horario a través del sistema.

Gracias a la organización por módulos de Angular y la librería de componentes de interfaz de usuario Angular Material, la aplicación web creada cuenta con una interfaz de usuario

fácil de usar. Esto se diferencia por completo del *software* FET, que como se ha mencionado anteriormente, no es intuitivo y requiere de un alto nivel de familiaridad con el programa para poder usarlo correctamente.

No obstante, no se cumplió el último de los objetivos específicos referente a la preplanificación académica. Esto se debió a la indicación del *product owner*, mencionada anteriormente (ANEXO VIII). En esta se señaló que debido a la complejidad del proceso de preplanificación académica y la solución propuesta por el coordinador de la carrera de Ingeniería de computación, Josafá Aguiar, este tema sea descartado del presente trabajo de integración curricular.

### 4.3 Recomendaciones

El mayor problema que se presentó durante el desarrollo del componente del presente documento fue el error en la planificación inicial. Esto se debió principalmente al tiempo que tomó aplicar las herramientas mencionadas anteriormente. El uso de BDD para las pruebas y su adaptación a NestJS tomó más tiempo de lo esperado y, por tanto, la planificación inicial fue alterada. Esto indica que se realizó una estimación incorrecta debido a que no se tomó en cuenta el tiempo de aprendizaje, razón por la cual es necesario enfatizar este hecho para tomarlo en cuenta como una mejora en futuros proyectos.

A pesar de haber optado por la metodología ágil, no se llevaron reuniones todos los días dentro del equipo de desarrollo. Si bien esto se debe a la diferencia en horarios de clases entre cada integrante además de la carga académica, esto supuso falta de comunicación y, en algunos momentos, un retraso en las actividades realizadas. Por esta razón, se debió contar con una mejor organización dentro del equipo para poder llevar a cabo estas reuniones más frecuentemente.

En el código fuente de la aplicación, tanto en *backend* como en *frontend*, no se tomó en cuenta desde un inicio una nomenclatura estándar (ej.: *camelCase*) por lo que existe aún variables por refactorizar, así como también el nombre de algunas columnas en las tablas de la base de datos. Esto surgió como un error por la falta de comunicación, de tal manera que cada integrante del equipo de desarrollo codificaba de diferente manera.

## 5 REFERENCIAS BIBLIOGRÁFICAS

- [1] Consejo Politécnico de la Escuela Politécnica Nacional, «Estatuto de la Escuela Politécnica Nacional (Codificación). Resolución RCP-315-2019.,» 8 Agosto 2019. [En línea]. Available: [https://cei.epn.edu.ec/Documentos/EPN/EstatutoEPN\\_Agosto\\_2019.pdf](https://cei.epn.edu.ec/Documentos/EPN/EstatutoEPN_Agosto_2019.pdf).
- [2] A. J. Ulloa Arévalo, «Desarrollo de una aplicación web de planificación académica para el Subdecanato de la Facultad de Ingeniería de Sistemas,» Quito, 2021.
- [3] W. D. Leonardini Aparicio, «Sistema de planificación y seguimiento académico caso: Unidad Educativa Rvdo. P. Walterstrub,» La Paz, 2009.
- [4] M. F. Reyes Wagnio, «Diseño y desarrollo de aplicación web para generación de planificación semestral académica,» Guayaquil, 2019.
- [5] J. Ferguson Smart, BDD in Action: Behavior-driven development for the whole software lifecycle, Primera ed., Manning, 2014.
- [6] M. Wynne, A. Hellesoy y S. Tooke, The Cucumber Book: Behaviour-Driven Development for Testers and Developers, Segunda ed., Pragmatic Bookshelf, 2017.
- [7] E. Herrera Uribe y L. E. Valencia Ayala, «Del manifiesto ágil sus valores y principios,» vol. XIII, pp. 381-386, 2007.

## **6 ANEXOS**

### **ANEXO I. DOCUMENTO DE REVISIÓN SISTEMÁTICA PARA LA INVESTIGACIÓN DE FUENTES BIBLIOGRÁFICAS**

Enlace:

<https://festive-salary-67b.notion.site/Revisi-n-sistem-tica-01f66b8cfd614d0eb07575942d852be1>

### **ANEXO II. DOCUMENTO DE EXTRACCIÓN DE INFORMACIÓN REALIZADA DURANTE LA INVESTIGACIÓN DE FUENTES BIBLIOGRÁFICAS**

Enlace al documento:

[https://epnecuador.sharepoint.com/:x:/s/Diseodetrabajodeintegracincurricular49/ERzvnSi1kuxMI\\_bQ-annffIBAvh3nWKo\\_w0i3VOVvS1VBg?e=5MWDvG](https://epnecuador.sharepoint.com/:x:/s/Diseodetrabajodeintegracincurricular49/ERzvnSi1kuxMI_bQ-annffIBAvh3nWKo_w0i3VOVvS1VBg?e=5MWDvG)

## ANEXO III. BACKLOG DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Identificador	Título	Descripción	Estimación
1	Parámetros iniciales	Como coordinador deseo establecer los parámetros iniciales como horas disponibles, los días hábiles, edificios y tipos de aulas, para definir los límites en la generación de horario y que cumplan con las limitaciones de la institución.	2
2	Generación de horarios con el software FET	El sistema necesita enviar los datos de entrada (docentes, asignaturas, aulas, carreras, paralelos y sus restricciones) al software FET vía TCP/IP para obtener como salida los horarios de cada nivel académico.	3
3	Difusión de horarios	Como subdecano deseo difundir los horarios aprobados a los respectivos involucrados para comunicar los horarios a los docentes, jefa de departamento, y secretaria para que se registren en el sistema académico de la institución con el fin de ser considerados en el proceso de matriculación de los estudiantes.	3
4	Creación de actividades	Como coordinador deseo poder crear una actividad antes de generar el horario seleccionando un profesor, la asignatura, aulas y sus restricciones para definir los horarios por nivel.	2
5	Registro de docente	Como coordinador deseo cargar la información de cada docente a través de un archivo o individualmente por un formulario y que se envíe una clave de acceso aleatorio para asegurar la integridad de los datos en el proceso de generación de horarios.	3
6	Registro de asignatura	Como coordinador deseo cargar la información de cada asignatura a través de un archivo o individualmente por un formulario para para asegurar la integridad de los datos en el proceso de generación de horarios.	2
7	Registro de carreras	Como coordinador deseo registrar la información de la carrera que me sea designada en la facultad, para asegurar la integridad de los datos en el proceso de generación de horarios.	2
8	Visualización de horarios	Como coordinador y subdecano quiero visualizar los horarios generados por	3

		semestre (de cada carrera) y docente para facilitar el proceso de revisión previo a la validación/carga en el sistema SAEW.	
9	Vinculación de carreras con asignaturas	Cómo coordinador deseo vincular los semestres de cada carrera existente con las asignaturas correspondientes a los mismos, para asegurar que la generación de horarios se enfoque en evitar cruces entre asignaturas del mismo nivel y carrera.	2
10	Aprobación de horarios	Como subdecano deseo aprobar los horarios generados por el software FET para evitar errores, reclamos y modificaciones no autorizadas con respecto a los horarios antes del inicio del semestre subsecuente.	2
11	Registro de espacios físicos	Como gestor de espacios físicos deseo registrar las diferentes aulas y laboratorios, ya sea de forma masiva a partir de un archivo o individualmente a través de un formulario, para distribuir los cursos de manera adecuada así evitando problemas por falta de infraestructura y evitar aglomeraciones.	2
12	Registro de horas no disponibles del docente	Como docente deseo solicitar al jefe de departamento la consideración de mis horas no disponibles, seleccionadas en una matriz, por cada día de la semana para que el horario del periodo subsecuente que me sea asignado pueda acoplarse a mis actividades académicas, de docencia y de investigación.	3
13	Registro de alumnos por asignatura	Como coordinador deseo ingresar a través de un formulario el número de estudiantes que cursarán cada una de las asignaturas del semestre actual, para poder planificar los horarios en base a los cupos que deberán estar disponibles para las asignaturas de dicho semestre.	3
14	Pre-planificación académica de horarios	Como coordinador deseo calcular el número de estudiantes que puedan matricularse en cada asignatura del semestre subsecuente a partir de la información de los estudiantes del semestre actual para contar con datos fiables que ayuden a disminuir la incertidumbre de los cupos que deberán estar disponibles el semestre subsecuente y determinar un número más certero para las asignaturas del siguiente nivel académico.	2

## **ANEXO IV. INFORMACIÓN DEL PROYECTO EN LA HERRAMIENTA NOTION**

Enlace:

<https://festive-salary-67b.notion.site/Proyecto-de-Titulaci-n-69e313d3b42347cbba104e0bdfa3cb29>

## **ANEXO V. REPOSITORIO DEL FRONTEND**

Enlace:

<https://github.com/Trabajo-de-Titulacion/planificacion-academica-fis-frontend/tree/develop>

## **ANEXO VI. REPOSITORIO DEL BACKEND**

Enlace:

<https://github.com/Trabajo-de-Titulacion/planificacion-academica-fis-backend/tree/develop>

## **ANEXO VII. REUNIÓN INFORMATIVA DEL SEMESTRE 2022-A**

Enlace:

<https://www.facebook.com/fisepn/videos/381126487328890/>

## **ANEXO VIII. PRESENTACIÓN CON LA NUEVA SUBDECANA DE LA FIS Y EL COORDINADOR DE LA CARRERA DE INGENIERÍA DE SOFTWARE**

El siguiente enlace requiere acceder mediante una cuenta institucional de la Escuela Politécnica Nacional: <https://tinyurl.com/55jtp96>

## ANEXO IX. FEATURES DE CADA HISTORIA

### Historia M1: Registro de espacios físicos

```
registro_de_espacios_fisicos.feature X
planificacion-academica-fis-backend > features > registro_de_espacios_fisicos.feature
1 # Language: es
2 Característica: Registro de espacios físicos
3 Como gestor de espacios físicos deseo registrar las diferentes aulas y laboratorios, ya sea de forma
4 masiva a partir de un archivo o individualmente a través de un formulario, para distribuir los cursos
5 de manera adecuada así evitando problemas por falta de infraestructura y evitar aglomeraciones.
6
7 @espaciosFisicosPruebaCrearUnEspacioFisico
8 Esquema del escenario: Se agrega un espacio físico
9 Dado que existe un espacio físico llamado "<nombre>"
10 Cuando se agrega un espacio físico llamado "<nuevoEspacioFisico>"
11 Entonces al consultar la base de datos se observan "<numeroRegistros>" registros.
12
13 Ejemplos:
14 | nombre | nuevoEspacioFisico | numeroRegistros |
15 | BetaPrueba | AlfaPrueba | 2 |
16 | BetaPrueba | BetaPrueba | 1 |
17
18
19 @espaciosFisicosPruebaCrearMultiplesEspaciosFisicos
20 Esquema del escenario: Se agrega un archivo con múltiples espacios físicos
21 Dado que existe un espacio físico llamado BetaPrueba
22 Y existe un espacio físico llamado AlfaPrueba
23 Y existe un espacio físico llamado GammaPrueba
24 Y existe un espacio físico llamado SIS502Prueba
25 Cuando se importe el archivo "<nombreArchivo>"
26 Entonces al consultar la base de datos se observan "<numeroRegistros>" espacios físicos.
27
28 Ejemplos:
29 | nombreArchivo | numeroRegistros |
30 # Archivo con 5 espacios físicos y 0 duplicados
31 | espacios_fisicos_prueba1.csv | 9 |
32 # Archivo con 5 espacios físicos y 2 duplicados
33 | espacios_fisicos_prueba2.csv | 7 |
```

### Historia M2: Registro de horas no disponibles del docente. Versión del sprint 2.

```
registro_de_horas_no_disponibles_del_docente.feature M X
planificacion-academica-fis-backend > features > registro_de_horas_no_disponibles_del_docente.feature
1 # Language: es
2 Característica: Registro de horas no disponibles del docente
3 Como docente deseo señalar en una matriz mis horas no disponibles por
4 cada día de la semana para que el horario del periodo subsecuente que
5 me sea asignado pueda acoplarse a mis actividades académicas, de
6 docencia y de investigación.
7
8 @horasNoDisponiblesPrueba1
9 Escenario: Se agregan horas no disponibles de un docente
10 Dado que existe un docente con correo docente.prueba@epn.edu.ec
11 Y un semestre con jornada laboral de lunes a sábado desde las 7h hasta las 20h con almuerzo a las 13h
12 Cuando se agregan como horas no disponibles el día Martes de 14 a 15
13 Y el día Martes de 15 a 16
14 Y también el día Jueves de 7 a 8
15 Entonces al consultar la base de datos de horas no disponibles se observan 3 registros.
```



## Historia M2: Registro de horas no disponibles del docente. Versión del *sprint* 3.

```
registro_de_horas_no_disponibles_del_docente.feature X
planificacion-academica-fis-backend > features > registro_de_horas_no_disponibles_del_docente.feature
1 # Language: es
2 Característica: Registro de horas no disponibles del docente
3 Como docente deseo solicitar al jefe de departamento la consideración de mis horas no disponibles,
4 seleccionadas en una matriz, por cada día de la semana para que el horario del periodo subsecuente
5 que me sea asignado pueda acoplarse a mis actividades académicas, de docencia y de investigación.
6
7 @horasNoDisponiblesPruebaHorasAprobadas
8 Escenario: Se aprueba la solicitud horas no disponibles de un docente
9 Dado que existe un docente con correo docente.prueba@epn.edu.ec
10 Y un semestre con jornada laboral de lunes a sábado desde las 7h hasta las 20h con almuerzo a las 13h
11 Cuando se solicitan como horas no disponibles el día Martes de 14 a 15
12 Y el día Martes de 15 a 16
13 Y también el día Jueves de 7 a 8
14 Y el jefe de departamento aprueba la solicitud
15 Entonces al consultar la base de datos de horas no disponibles se observan 3 registros
16 Y el docente recibe un correo electrónico indicando la aprobación de su solicitud.
17
18 @horasNoDisponiblesPruebaHorasRechazadas
19 Escenario: Se rechaza la solicitud horas no disponibles de un docente
20 Dado que existe un docente con correo docente.prueba@epn.edu.ec
21 Y un semestre con jornada laboral de lunes a sábado desde las 7h hasta las 20h con almuerzo a las 13h
22 Cuando se solicitan como horas no disponibles el día Martes de 14 a 15
23 Y el día Martes de 15 a 16
24 Y también el día Jueves de 7 a 8
25 Y el jefe de departamento rechaza la solicitud
26 Entonces al consultar la base de datos de horas no disponibles se observan 0 registros
27 Y el docente recibe un correo electrónico indicando el rechazo de su solicitud.
```

## Historia M3: Registro de alumnos por asignatura.

```
registro_de_alumnos_por_asignatura.feature U X
planificacion-academica-fis-backend > features > registro_de_alumnos_por_asignatura.feature
1 # Language: es
2 Característica: Registro de alumnos por asignatura
3 Como coordinador deseo ingresar a través de un formulario el número de estudiantes que cursarán cada una
4 de las asignaturas del semestre actual, para poder planificar los horarios en base a los cupos que deberán
5 estar disponibles para las asignaturas de dicho semestre.
6
7 @pruebaNumeroEstudiantes
8 Escenario: Se indica el número de estudiantes de un semestre en curso
9 Dado que existe un semestre cuya planificación está en curso
10 Y existen las asignaturas de AsignaturaPrueba1, AsignaturaPrueba2, AsignaturaPrueba3 y AsignaturaPrueba4
11 Cuando se indica el número de estudiantes 27, 10, 40 y 5, respectivamente,
12 Entonces al consultar la base de datos de número de estudiantes se observan 4 registros.
```

## ANEXO X. DOCUMENTACIÓN DE LA API EN SWAGGER

### Historia M1: Registro de espacios físicos

Espacios físicos		^
GET	/api/espacios_fisicos/obtenerEspaciosFisicos	Método para obtener todos los espacios físicos. <span>↓</span>
GET	/api/espacios_fisicos/obtenerEspaciosFisicos/{id}	Método para obtener un espacio físico mediante su ID. <span>↓</span>
POST	/api/espacios_fisicos/crearUno	Método para crear un espacio físico. <span>↓</span>
POST	/api/espacios_fisicos/crearMultiples	Método para crear múltiples espacios físicos mediante un archivo CSV. <span>↓</span>
PUT	/api/espacios_fisicos/actualizarEspacioFisico/{id}	Método para actualizar un espacio físico mediante su ID. <span>↓</span>
DELETE	/api/espacios_fisicos/eliminarEspacioFisico/{id}	Método para eliminar un espacio físico mediante su ID. <span>↓</span>

### Historia M2: Registro de horas no disponibles del docente. Versión del *sprint 3*.

Horas no disponibles		^
POST	/api/horas_no_disponibles/solicitarHorasNoDisponibles/{id}	Método para solicitar múltiples horas no disponibles de un docente. <span>↓</span>
GET	/api/horas_no_disponibles/obtenerHorasNoDisponiblesSolicitadasPorDocenteId/{id}	Método para obtener todas las horas no disponibles de un docente mediante su ID. <span>↓</span>
GET	/api/horas_no_disponibles/obtenerSolicitudesDelSemestreEnProgreso	Método para obtener todas las solicitudes de horas no disponibles del semestre en progreso. <span>↓</span>
GET	/api/horas_no_disponibles/aprobarSolicitudHorasNoDisponiblesPorDocenteId/{id}	Método para aprobar la última solicitud de horas no disponibles de un docente mediante su ID. <span>↓</span>
GET	/api/horas_no_disponibles/rechazarSolicitudHorasNoDisponiblesPorDocenteId/{id}	Método para rechazar la última solicitud de horas no disponibles de un docente mediante su ID. <span>↓</span>
DELETE	/api/horas_no_disponibles/eliminarHoras/{id}	Método para eliminar horas no disponibles de un docente. <span>↓</span>
GET	/api/horas_no_disponibles/obtenerTodasLasHorasNoDisponiblesAprobadas	Método para obtener todas las horas no disponibles aprobadas de cada docente, para usarlas en la generación de horarios. <span>↓</span>

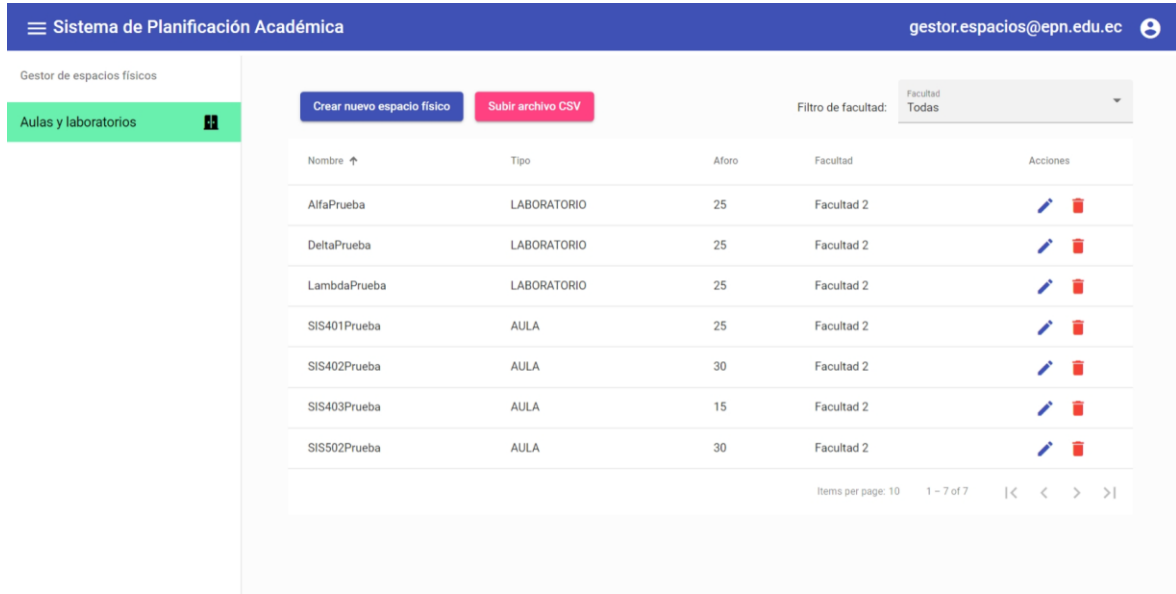
### Historia M3: Registro de alumnos por asignatura.

Número de estudiantes por semestre		^
POST	/api/numero_estudiantes_por_semestre/registrarNumeroEstudiantesPorSemestre	Método para registrar el número de estudiantes de cada asignatura de un semestre en curso. <span>↓</span>
GET	/api/numero_estudiantes_por_semestre/obtenerNumeroEstudiantesPorSemestreId/{id}	Método para obtener el número de estudiantes por cada asignatura de un semestre indicado por su ID. <span>↓</span>

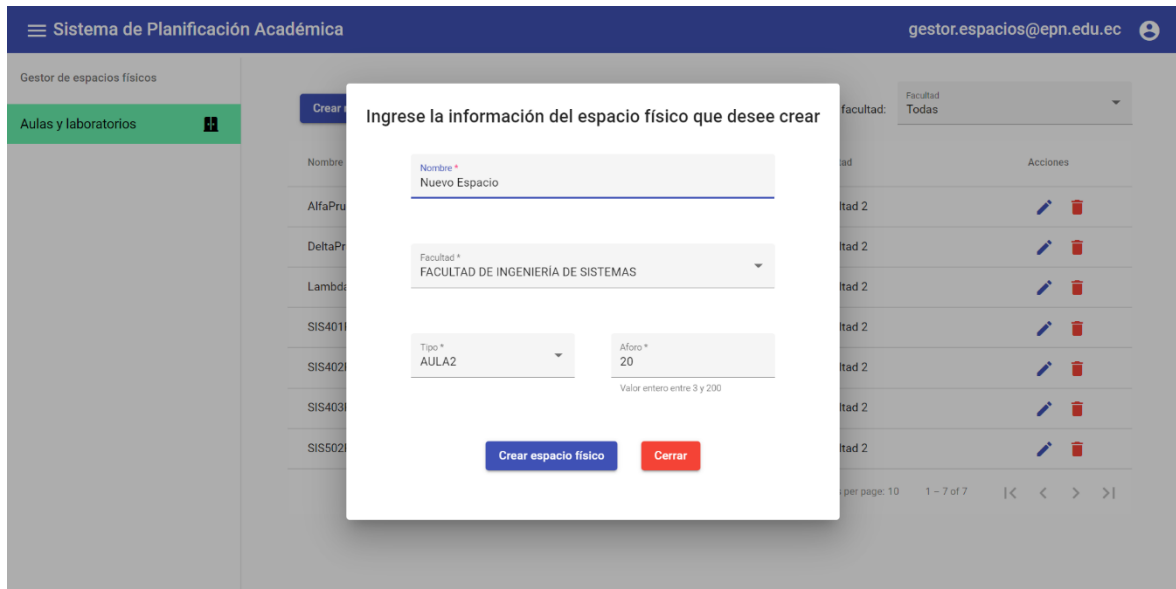
# ANEXO XI. INTERFACES DE USUARIO PARA CADA HISTORIA

## Historia M1: Registro de espacios físicos

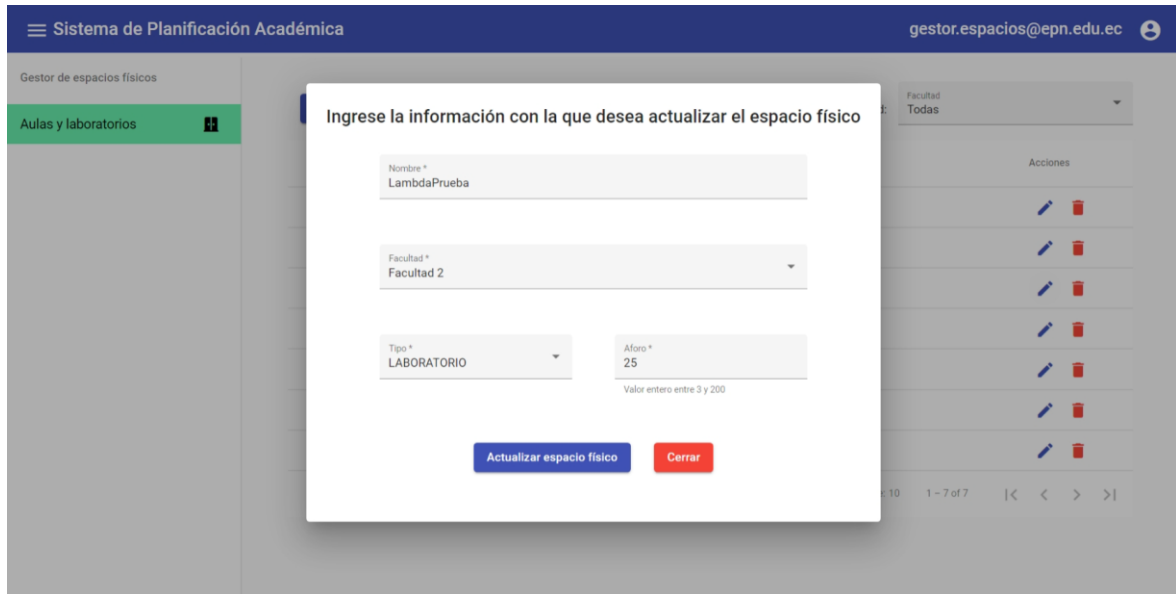
- Pantalla principal



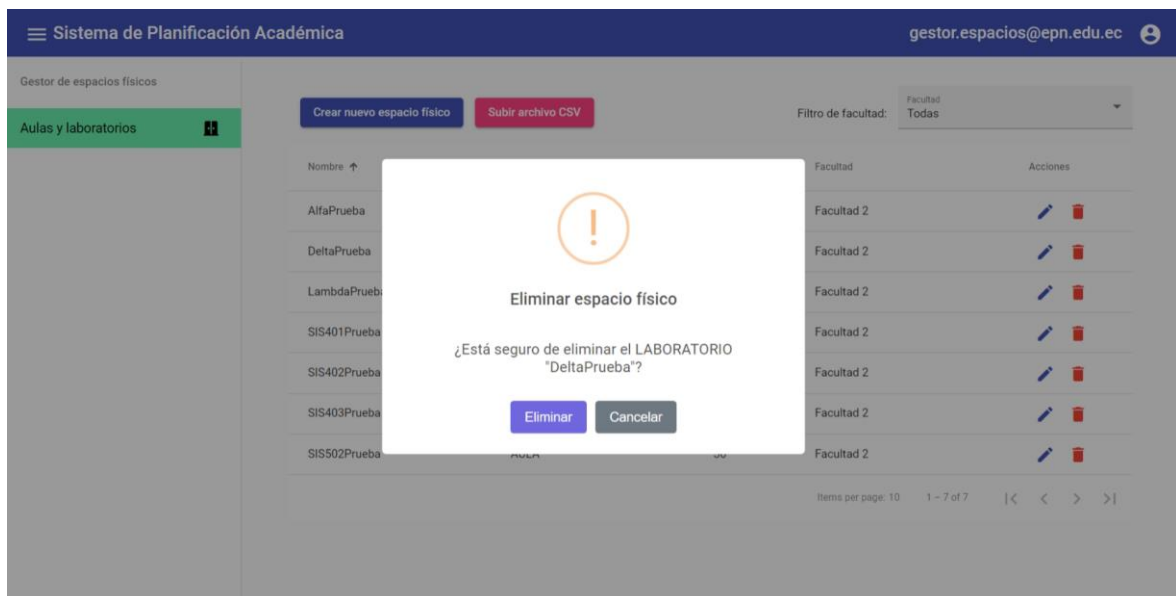
- Creación de un nuevo espacio físico

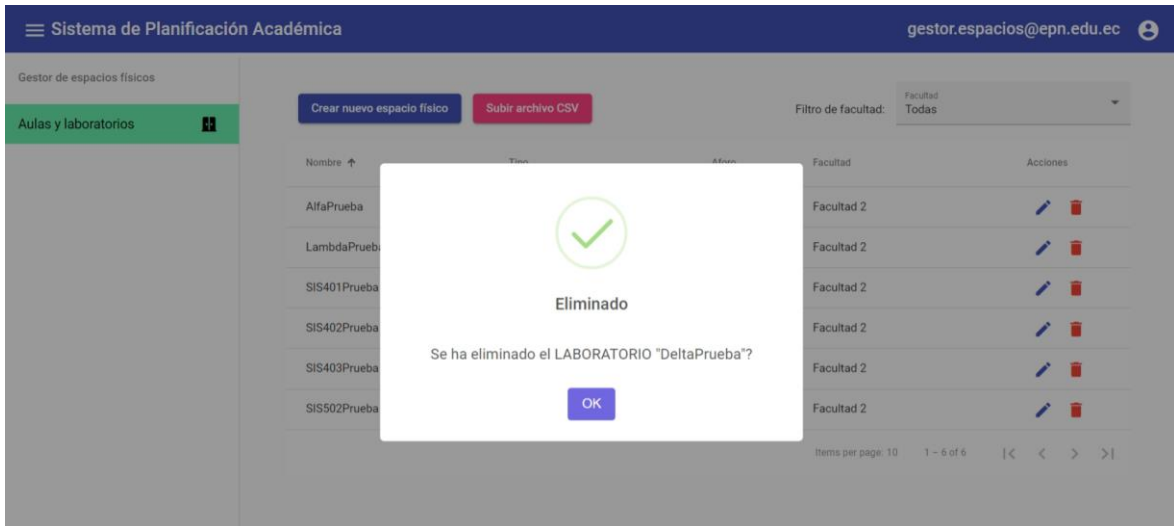


- Actualización de un espacio físico

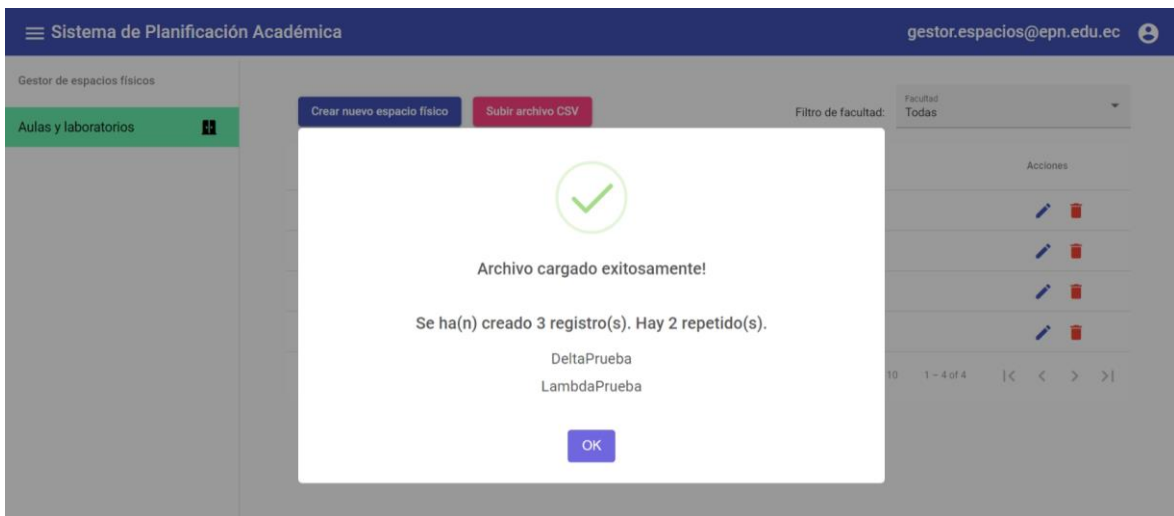


- Eliminación de un espacio físico

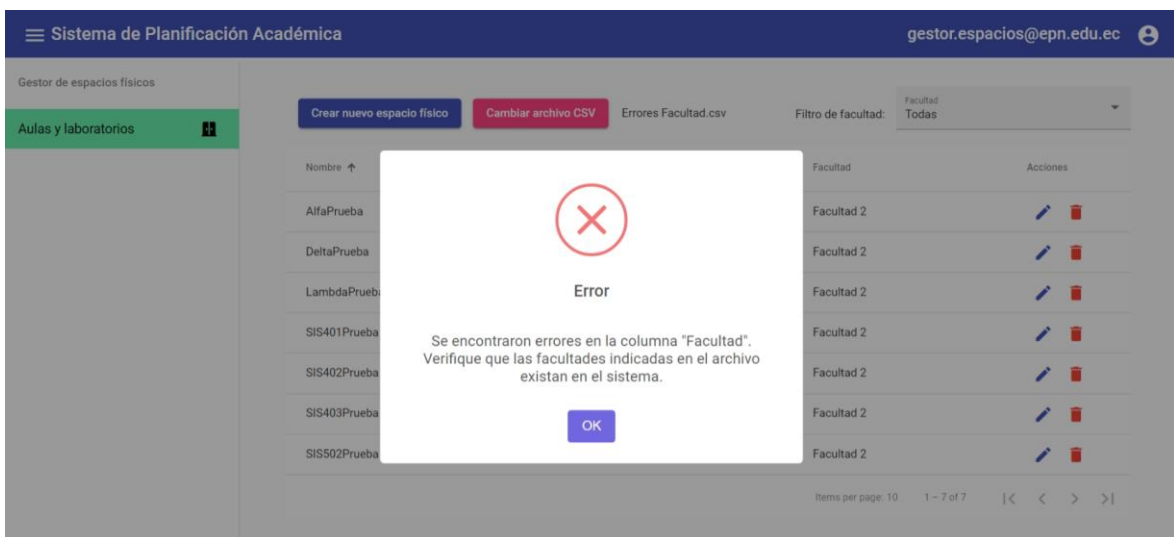




- Creación de múltiples espacios físicos mediante un archivo CSV



- Ejemplo de mensaje de error al subir un archivo con datos incorrectos



## Historia M2: Registro de horas no disponibles del docente. Versión final (sprint 3).

- Pantalla principal de solicitud de horas no disponibles (docente)

Sistema de Planificación Académica mahatma.quijano@epn.edu.ec

Docente

Solicitar horas no disponibles

Seleccione únicamente aquellas horas en las que **NO** pueda llevar a cabo sus actividades de docencia.

Horas totales a la semana: 84 Horas no disponibles: 10

HORA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO
7:00 - 8:00	X	X				
8:00 - 9:00	X	X				
9:00 - 10:00						
10:00 - 11:00						
11:00 - 12:00						
12:00 - 13:00						
13:00 - 14:00	Almuerzo	Almuerzo	Almuerzo	Almuerzo	Almuerzo	Almuerzo
14:00 - 15:00						
15:00 - 16:00						
16:00 - 17:00						
17:00 - 18:00						
18:00 - 19:00						
19:00 - 20:00						
20:00 - 21:00						
21:00 - 22:00	X	X	X	X	X	X

Desmarcar todo Solicitar horas no disponibles

Sistema de Planificación Académica mahatma.quijano@epn.edu.ec

Docente

Solicitar horas no disponibles

Seleccione únicamente aquellas horas en las que **NO** pueda llevar a cabo sus actividades de docencia.

Horas totales a la semana: 84 Horas no disponibles: 10

HORA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO
7:00 - 8:00	X	X				
8:00 - 9:00	X	X				
9:00 - 10:00						
10:00 - 11:00						
11:00 - 12:00						
12:00 - 13:00						
13:00 - 14:00	Almuerzo	Almuerzo	Almuerzo	Almuerzo	Almuerzo	Almuerzo
14:00 - 15:00						
15:00 - 16:00						
16:00 - 17:00						
17:00 - 18:00						
18:00 - 19:00						
19:00 - 20:00						
20:00 - 21:00						
21:00 - 22:00	X	X	X	X	X	X

Horas no disponibles actualizadas exitosamente

Se ha creado una nueva solicitud para el semestre 2022-A, con 10 hora(s) no disponible(s).

OK

Desmarcar todo Solicitar horas no disponibles

- Listado de solicitudes de docentes (jefe de departamento)

Sistema de Planificación Académica jefe.departamento@epn.edu.ec

Jefe de departamento

Solicitudes de horas no disponibles de cada docente para el semestre 2022-A

Filtro:

Docente	Fecha de última modificación ↓	Estado	Acciones
QUIJANO ZANIPATTINI MAHATMA ISRAEL	2022-09-07T22:21:52.217Z	POR_REVISAR	
LLANGANATE VALENCIA LUIS ALEJANDRO	2022-09-04T23:40:38.590Z	POR_REVISAR	
ORDÓÑEZ CALERO HERNÁN DAVID	2022-09-04T23:40:38.590Z	POR_REVISAR	
NAVARRETE RUEDA ROSA DEL CARMEN	2022-09-04T23:40:38.590Z	APROBADA	

Items per page: 10 1 - 4 of 4 |< < > >|

- Visualización de una solicitud (jefe de departamento)

Sistema de Planificación Académica jefe.departamento@epn.edu.ec

Jefe de departamento

Solicitudes de horas no disponibles

Regresar

Solicitud de: QUIJANO ZANIPATTINI MAHATMA ISRAEL

HORA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO
7:00 - 8:00	X	X				
8:00 - 9:00	X	X				
9:00 - 10:00						
10:00 - 11:00						
11:00 - 12:00						
12:00 - 13:00						
13:00 - 14:00						
14:00 - 15:00						
15:00 - 16:00						
16:00 - 17:00						
17:00 - 18:00						
18:00 - 19:00						
19:00 - 20:00						
20:00 - 21:00						
21:00 - 22:00	X	X	X	X	X	X

Rechazar Aprobar

Sistema de Planificación Académica jefe.departamento@epn.edu.ec

Jefe de departamento

Solicitudes de horas no disponibles

Regresar

Solicitud de: QUIJANO ZANIPATTINI MAHATMA ISRAEL

Esta solicitud ya ha sido aprobada.

HORA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO
7:00 - 8:00	X	X				
8:00 - 9:00	X	X				
9:00 - 10:00						
10:00 - 11:00						
11:00 - 12:00						
12:00 - 13:00						
13:00 - 14:00						
14:00 - 15:00						
15:00 - 16:00						
16:00 - 17:00						
17:00 - 18:00						
18:00 - 19:00						
19:00 - 20:00						
20:00 - 21:00						
21:00 - 22:00	X	X	X	X	X	X

**Solicitud aprobada**

Solicitud aprobada. Se ha enviado la notificación respectiva al correo mahatma.quijano@epn.edu.ec.

OK

Rechazar Aprobar

## Historia M3: Registro de alumnos por asignatura.

- Pantalla para cargar número de estudiantes

Sistema de Planificación Académica coordinador@epn.edu.ec

Coordinador

Parámetros iniciales

Docentes

Asignatura

Carrera

**Número de estudiantes**

Horario

### Periodo semestre 2022-A

Código	Nombre	Número de estudiantes
MATD113	ALGEBRA LINEAL	<input type="text" value="30"/>
FISD134	MECANICA NEWTONIANA	<input type="text" value="25"/>
PRLD105	PRACTICAS LABORALES	<input type="text" value="16"/>
CSHD111	COMUNICACION ORAL Y ESCRITA	<input type="text" value="34"/>

Items per page: 10 0 of 0 |< < > >|

**Guardar cambios**

Sistema de Planificación Académica coordinador@epn.edu.ec

Coordinador

Parámetros iniciales

Docentes

Asignatura

Carrera

**Número de estudiantes**


Horario

### Periodo semestre 2022-A

Código	Número de estudiantes
MATD113	<input type="text" value="30"/>
FISD134	<input type="text" value="25"/>
PRLD105	<input type="text" value="16"/>
CSHD111	<input type="text" value="34"/>

Items per page: 10 0 of 0 |< < > >|

**Guardar cambios**



**Datos actualizados**

Se ha registrado el número de estudiantes de 4 asignatura(s).

**OK**



**ANEXO XII. ENTREVISTA A LOS STAKEHOLDERS REALIZADA  
POR CARLOS ANCHUNDIA**

Enlace: <https://youtu.be/c6QVJtO-f4A>

**ANEXO XIII. ENTREVISTA AL SUBDECANO REALIZADA POR EL  
EQUIPO DE DESARROLLO**

Enlace: <https://youtu.be/8Y60S9U4XAq>

**ANEXO XIV. ENTREVISTA A ALEX ULLOA**

Enlace: <https://youtu.be/r8gOMTsNGzQ>