ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

App integrada con dispositivos de ambientes inteligentes para el control y mejora en la productividad y eficiencia de los recursos tecnológicos del laboratorio de sistemas telemáticos

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero en Telemática

Presentado por:

Ronny Alexander Martínez Flores

Joel Alexis Jordán Freire

GUAYAQUIL - ECUADOR

Año: 2022

DEDICATORIA

Ronny Alexander Martínez Flores

Dedico este trabajo a aquellos que han sido una fuente constante de motivación y apoyo en mi vida, especialmente a mi madre, padre y hermanos, quienes han sido una presencia constante y han contribuido a mi crecimiento y desarrollo personal.

Joel Alexis Jordán Freire

El presente proyecto se lo dedico a mi familia, porque ellos han sido la razón de mi vida, por sus consejos, su paciencia, su apoyo incondicional, porque todo lo que soy es gracias a ellos, me han formado con reglas y ciertas libertades, pero siempre me motivaron a ser constante para alcanzar mis anhelos y cumplir mis sueños.

AGRADECIMIENTOS

Ronny Alexander Martínez Flores

Expreso mi profundo agradecimiento a Dios y a mi familia por su inestimable apoyo y colaboración en la realización de este importante proyecto. Gracias por su constante motivación y brindarme las ideas necesarias para superar los desafíos que se presentaron durante el desarrollo.

Joel Alexis Jordán Freire

Agradezco a Dios por haberme otorgado una familia maravillosa, agradezco a mi familia por apoyarme en cada decisión y proyecto de vida, por confiar en mi, por su aporte y amor, gracias a la vida porque cada día me demuestra que siempre se puede seguir adelante a pesar de los diferentes obstáculos que se presenten, agradezco a mis amigos por haber hecho este desafío universitario más llevadero y agradezco también a Andreina Toala por haberme acompañado en gran parte de este camino y haber sido un soporte en muchos aspectos.

DECLARACIÓN EXPRESA

”Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; RONNY ALEXANDER MARTÍNEZ FLORES y JOEL ALEXIS JORDÁN FREIRE damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

Ronny Alexander Martínez Flores

Joel Alexis Jordán Freire

EVALUADORES

Firmado electrónicamente por:
IGNACIO MARIN GARCIA

Msc. Ignacio Marín García

PROFESOR DE LA MATERIA

Firmado electrónicamente por:
CHRISTOPHER JAVIER VACCARO CEDILLO

Ing. Christopher Vaccaro

PROFESOR TUTOR

RESUMEN

Este proyecto busca mejorar la gestión de la asistencia y el inventario de los recursos del laboratorio de sistemas telemáticos mediante la integración de dispositivos y aplicaciones. Para este fin, se ha desarrollado una aplicación web y una API web que sirve para registrar la asistencia de los estudiantes y permite que ellos mismos ingresen su asistencia, mientras que el profesor como usuario administrador, puede ver y gestionar toda la información pertinente. La aplicación también incluye una funcionalidad de inventario que permite monitorear, registrar y gestionar los objetos RFID detectados mediante un lector RFID que trabaja en una frecuencia de 902 a 928 MHz. Para el desarrollo de la aplicación se utilizaron tecnologías como HTML, CSS, JavaScript, .NET, ASP.NET Core MVC, ASP.NET Core Web API, IIS y SQL Server. Además, se implementaron medidas de seguridad para proteger la información registrada.

Los resultados demostraron que la aplicación es eficiente en la gestión de la asistencia y el inventario, es fácil de usar y proporciona información precisa y actualizada.

En conclusión, este proyecto muestra la viabilidad de crear una aplicación para simplificar el proceso de toma de asistencia y poder conocer las actividades que se desarrollan en el laboratorio de sistemas telemáticos, incluyendo a las personas y la utilización de los recursos del mismo mediante el control de inventario.

Palabras clave: Asistencia, Inventario, Aplicación Web, API Web, RFID.

ABSTRACT

This project sought to improve the management of attendance and inventory of telematics laboratory resources through the integration of devices and applications. For this purpose, a web application and a web API was developed that serves to record the attendance of the students and allows them to enter their attendance themselves, while the teacher as administrator user, can view and manage all the relevant information. The application also includes an inventory functionality that allows to monitor, record and manage detected RFID objects using an RFID reader that works at a frequency of 902 to 928 MHz. For the development of the application technologies such as HTML, CSS, JavaScript, .NET, ASP.NET CORE MVC, ASP.NET Core Web API, IIS, and SQL Server. In addition, security measures were implemented to protect the recorded information.

The results showed that the application is efficient in managing attendance and inventory, is easy to use and provides accurate and up-to-date information.

In conclusion, this project shows the feasibility of creating an application to simplify the process of taking attendance of students and to track the activities that are developed in the laboratory of telematic systems, including people and the use of the resources of the same through inventory control.

Keywords: Attendance, Inventory, Web Application, Web API, RFID.

ÍNDICE GENERAL

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
FIEC	Facultad de Ingeniería Eléctrica y Computación
RFID	Identificación por radiofrecuencia (del inglés, Radio Frequency Identification)
LST	Laboratorio de Sistemas Telemáticos
IDAAI	Integración de Dispositivos y Aplicaciones en Ambientes Inteligentes
API	Interfaz de programación de aplicaciones (del inglés, Application programming interface)
APP	Aplicación (del inglés, Application)
IIS	Servicios de información de Internet (del inglés, Internet Information Services)
CSV	Valores separados por comas (del ingĺes, Comma-Separated Values)
HTTP	El Protocolo de transferencia de hipertexto (del inglés, Hypertext Transfer Protocol)
UHF	Frecuencia ultraalta (del inglés, Ultra high Frequency)
SNMP	Protocolo simple de administración de red (del inglés, Simple Network Management Protocol)
SQL	Lenguaje de consulta estructurada (del inglés, Structured Query Language)
VSWR	Relación de onda estacionaria de voltaje (del inglés, Voltage Standing Wave Ratio)
HVAC	Calefacción, ventilación y aire acondicionado (del inglés, Heating, Ventilation, and Air Conditioning)
IoT	Internet de las cosas (del inglés, Internef of things)
JSON	Notación de objetos de JavaScript (del inglés, JavaScript Object Notation)

SIMBOLOGÍA

KHz Kilohercios

MHz Megahercios

dBm decibelio-milivatio

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO 1

1. INTRODUCCIÓN

El Laboratorio de Sistemas Telemáticos (LST) es el primer laboratorio a nivel nacional, que ha sido diseñado específicamente para la disciplina de telemática. Fue inaugurado en el año 2014 por el Rector Sergio Flores Macías y la coordinadora de Ingeniería en telemática Dra. Patricia Chávez Burbano, y desde entonces ha sido el nido de una basta cantidad de desarrollo de proyectos de gran importancia, tanto a nivel nacional, como internacional, siendo dichos proyectos merecedores de diversos tipos de reconocimientos. Por ejemplo, el concurso interuniversitario Siemens 2013, el premio nacional Invoca TIC 2013, y el concurso Innovation 2013, en el que quedó entre los 25 mejores a nivel mundial. El LST es utilizado para diversas materias de la carrera de pregrado de Ingeniería en Telemática, sin mencionar que sus equipos sirven como soporte para la Maestría en Telecomunicaciones que brinda la Escuela Superior Politécnica del Litoral (ESPOL), así como proyectos de investigación tanto de grado como de postgrado.[1]

Debido al espacio y los recursos que brinda este laboratorio para llevar la teoría aprendida en los salones de clases a la práctica, se ha vuelto cada vez más utilizado. Por lo cual, se pretenden implementar ciertos de cambios en la forma en que se llevan a cabo dos procesos muy importantes y al mismo tiempo, muy repetitivos, estos son: La toma de asistencia y el control de inventario.

El presente trabajo dará comienzo a la exposición de una serie de implementaciones que tendrán como fin: Facilitar, simplificar y automatizar la forma en que los procesos anteriormente mencionados son llevados a cabo.

1.1 Definición de la problemática

La forma en que se lleva a cabo la toma de asistencia y el control de inventario de los recursos del laboratorio cuando se realizan los préstamos de estos, corresponde actualmente a un proceso manual, donde docente y estudiante interactúan entre sí para realizar dichas acciones. Aunque este procedimiento por sí mismo funciona, puede conllevar algunos problemas que podrían ser causados debido a errores humanos, tales como: realizar la anotación de asistencia o falta de un estudiante que no correspondía, olvidarse de iniciar el proceso de préstamo de un dispositivo a un estudiante por causa de alguna novedad y en sí, cualquier tipo de error de carácter humano que pudiese llegar a acontecer. Por esta razón, se plantea evitar recurrir a la necesidad de un proceso manual en la realización de estas operaciones, recurriendo en su lugar, a un método mas automático.

Este nuevo método de recolección plantea automatizar dichos procesos a través de un registro de asistencia por parte de los estudiantes en la aplicación web, mientras que para el control de préstamo de dispositivos del laboratorio se plantea implementar un sistema mediante etiquetas de identificación por radiofrecuencia, que diferenciará cada dispositivo como único para poder detectar los dispositivos registrados que ya no se encuentren dentro del laboratorio.

1.2 Objetivos

El objetivo general del presente proyecto es **simplificar el proceso de toma de asistencia y conocer sobre las actividades que se desarrollan en el LST, incluyendo a las personas y la utilización de los recursos del mismo mediante el control de inventario, con ayuda de una aplicación web.**

Objetivos específicos

- Crear un sistema informático que permita gestionar la toma de asistencia de estudiantes en el laboratorio a través de una página web, para que se automatice este proceso.
- Crear un sistema informático para controlar el préstamo de los dispositivos del laboratorio a estudiantes, por medio de etiquetas por identificación de radiofrecuencia (RFID) que servirán para diferenciar dichos dispositivos sin necesidad de la intervención de terceros.
- Crear un sistema informático mediante el desarrollo de una aplicación web que tomará los datos correspondientes a los registros de asistencia y el inventario de objetos RFID, para después mostrar dicha información en diferentes paneles.

1.3 Descripción de escenarios para la propuesta

Se necesita gestionar la mejora del laboratorio de Sistemas Telemáticos de la Facultad de Ingeniería Eléctrica y Computación (FIEC), perteneciente a ESPOL, convirtiéndolo en un ambiente inteligente para conocer sobre las actividades que se desarrollan en dicha área, incluyendo a las personas y la utilización de los recursos del mismo. Con la finalidad de lograr lo propuesto previamente se presenta la creación de una aplicación web que constaría de dos modos: Laboratorio de clases y Laboratorio abierto.

En el modo laboratorio de clases se permitirá tomar lista y gestionar la presencialidad de los estudiantes, integrándose con la red WiFi, para verificar la ocupancia del laboratorio se tomará en cuenta la siguiente métrica: número de personas registradas (lista de asistencia). Para detectar el número de personas registradas, es decir la toma de asistencia, se tiene contemplada la creación de una página web, donde los estudiantes tendrán que acceder a la misma para confirmar su asistencia mediante el registro de su número de matricula.

En el modo laboratorio abierto se permitirá ver la disponibilidad y los recursos del mismo, integrándose con un lector de identificación por radiofrecuencia (RFID) que detecta los recursos/dispositivos etiquetados para el control del inventario.[2]

1.4 Estado del arte

Los ambientes, entornos o espacios inteligentes hacen referencia a espacios físicos en los cuales se unen diversas tecnologías para mejorar la experiencia del usuario. Con el avance tecnológico en los últimos años ha crecido la demanda de implementación de estos espacios inteligentes, sin embargo, estos ambientes tienen un precedente en los años 60, en donde ya existían edificios inteligentes que se caracterizaban porque muchas de sus funciones se encontraban automatizadas.

El concepto de ambientes inteligentes se lo atribuye a Michael Cohen en 1998, quien lo definió como aquello que conjugaba la computación ubicua con una interacción natural entre la persona y el computador, quien pretendía que en un futuro el mundo digital se integre con el mundo real.[3]

La inteligencia de estos ambientes se puede encontrar de forma distribuida en de diferentes maneras, estas pueden ser en: sistemas empotrados o de forma centralizada en grandes servidores. Los dispositivos que se usan para la integración de estos espacios se organizan en redes de comunicaciones que pueden ser: inalámbricas, cableadas o híbridas, las entradas y salidas son sensores y actuadores, que pueden ser simples transductores o dispositivos inteligentes que constituyen cierta capacidad de proceso y/o comunicación.

Hace aproximadamente más de una década ya se conocía que la inteligencia de un edificio era el resultado de tres condiciones básicas: comunicaciones avanzadas con grandes capacidades, automatización y un completo y gran equipamiento ofimático. A partir de la década de los 60, se pueden establecer cuatro generaciones de edificios inteligentes:

Edificios inteligentes de primera generación: Los sistemas de calefacción, ventilación y aire acondicionado, conocidos como sistemas HVAC, fueron los primeros sistemas de construcción controlados electrónicamente. Los chips de computadora permitían que sean controlados por sensores localizados, lo que permitía respuestas más rápidas y precisas a los cambios en las condiciones climáticas.

El uso y abuso del término edificio inteligente provocó una reacción violenta por parte de los propietarios de edificios, patrocinadores y grupos financieros que señalan que estos edificios no cumplen con sus verdaderos requisitos.[4]

Edificios inteligentes de segunda generación: En respuesta a lo ocurrido en la primera fase, grandes equipos de desarrollo tecnológico diseñaron e implementaron nuevos sistemas y tecnologías de infraestructura para dar paso a la segunda generación de edificios inteligentes, las cuales son tecnologías que hacen posible la integración y separación de sistemas utilizando tecnologías informáticas y de telecomunicaciones avanzadas.[5]

Edificios inteligentes de tercera generación: Los edificios de tercera generación se crearon alrededor de los años 90. Los países desarrollados han dado grandes pasos en el desarrollo de edificios inteligentes, pero también han tenido notorias consecuencias negativas que empezaron poco a poco a ser superadas. En general, los problemas eran las consecuencias negativas de la optimización unilateral de subsistemas. Actualmente, en varios campos de trabajo, existe una tendencia a combinar la llamada inteligencia distribuida centralizada utilizando sistemas autónomos inteligentes que se comunican con redes de comunicación, donde se puede crear un punto de control a partir de diferentes puntos de red e interactuar con diferentes sistemas interconectados. Siendo así, en el caso de ocurrir una falla en el equipo no solo dejaría al equipo sin sistema, sino que una falla general en la red permitiría al equipo autónomo operar con un rendimiento mínimo debido a la inteligencia compartida entre estos sistemas.[6]

Edificios inteligentes de cuarta generación: Incorporan el concepto de inteligencia avanzada, que implica la toma de decisiones inteligentes e integradas basadas en el conocimiento actual e histórico, también considera conceptos de arquitectura protegida y diseño de vida que son relevantes para nuestra realidad.[6]

Los ambientes inteligentes se expandieron para incluir edificios y entornos de trabajo, con la introducción de sistemas de automatización de edificios y tecnologías de comunicación inalámbrica. Debido a la creciente popularidad de Internet los ambientes inteligentes se volvieron más conectados y se empezó a hablar de la ”Internet de las cosas” (IoT), el cual es el proceso que conecta elementos físicos cotidianos a Internet. [7]

Hoy en día, los ambientes inteligentes incluyen una amplia gama de tecnologías, desde sensores y dispositivos móviles hasta sistemas de inteligencia artificial y realidad virtual. Se utilizan en una variedad de entornos, como hogares, oficinas, hospitales, fábricas y ciudades inteligentes. Los ambientes inteligentes tienen como objetivo mejorar la eficiencia, la comodidad, la seguridad y la sostenibilidad, y se espera que desempeñen un papel cada vez más importante en el futuro de la tecnología y la sociedad.[8]

CAPÍTULO 2

2. METODOLOGÍA

En este capítulo se explica la estrategia y herramientas a utilizar para asegurar el éxito de la investigación y lograr los objetivos del proyecto.

Diseño de la investigación: Se ha propuesto en el presente trabajo utilizar el diseño descriptivo, debido al escenario que envuelve las necesidades de este trabajo, puesto que se busca explicar cada proceso interno (toma de asistencia y control de inventarios de dispositivos del laboratorio) en su debida fase, antes y después de la aplicación del presente proyecto y como se fue introduciendo el cambio.[9]

Enfoque de la investigación: Se busca utilizar el enfoque cualitativo, puesto que se trata de una investigación en donde se busca cambiar la forma en que se procesan actividades específicas dentro del laboratorio, mas no determinar un hipotético nivel de satisfacción entre estudiantes o docentes.[10]

Población: La población seleccionada para este proyecto son tanto estudiantes como docentes, debido a que son los protagonistas en el sistema de toma de asistencia y control de inventario de dispositivos, siendo un proceso en el que ambos interactúan entre sí, como se comentó con anterioridad.

Tamaño de la muestra: Estará constituido por un docente y 10 estudiantes, que corresponde a la cantidad en promedio de estudiantes que se registran en el laboratorio de sistemas telemáticos, y la cantidad de estudiantes que normalmente están dentro del laboratorio cuando este se encuentra en modo abierto, es decir, cuando no hay clases y los estudiantes entran a utilizar sus equipos e instalaciones.

Tipo de muestreo: Para la recopilación de datos en el desarrollo de este proyecto se busca utilizar una técnica de muestreo basada en la probabilidad, conocida como el muestreo deliberado, es aquel que se selecciona en base al propósito de la investigación, por ello se ha seleccionado al laboratorio de sistemas telemáticos para realizar el proyecto.

Técnicas de investigación: Para este proyecto se busca utilizar la técnica de investigación cuantitativa debido que es un técnica objetiva y empírica de la cual se logrará llegar a conclusiones específicas de acuerdo a lo observado. Por ello mediante el uso de sensores previamente instalados en el laboratorio recolectaremos datos necesarios para el desarrollo del proyecto.

2.1 Instrumentos de recolección de datos

Para la recolección de datos se hará uso de instrumentos dependiendo del proceso en el que se utilicen y las capacidades de los usuarios. Estos tres instrumentos de recolección de datos son: :

Aplicación web: Con la finalidad de convertir el laboratorio en un ambiente inteligente, se creará una aplicación la cual se integrará con los diferentes sensores previamente instalados, por ende, este instrumento también permitirá el análisis de los datos recolectados, tanto la cantidad de estudiantes (con su respectiva asistencia) que se encuentran en el laboratorio como la cantidad de recursos del mismo, mediante el etiquetado RFID.

Portal para registro de asistencia: Para la recaudación de datos de los estudiantes que asistieron a clases se hará uso de una sección de la aplicación web llamada ”Registro de Asistencia”, se trata de un submenú de la aplicación web al que accederán los estudiantes una vez se conecten a la red requerida. En esta página cada estudiante podrá ingresar su respectivo número de matrícula y asi registrar la asistencia de ese día para ese módulo, que corresponde al módulo actual seleccionado por el usuario (profesor) en la aplicación.

Sistema RFID: Con el fin de mantener un control de inventario se usó un sistema RFID, el cual mediante un lector RFID y una antena permite obtener la identificación de una etiqueta, que está adherida a un objeto, en este caso serán los recursos/dispostivos, de forma inalámbrica mediante las ondas de radio para poder determinar si se encuentran dentro del laboratorio.

Técnicas de análisis de datos: Implica la recopilación de información detallada sobre las características y funcionalidades de la aplicación web y del API desarrollados para la gestión de inventario y asistencia de los estudiantes en el laboratorio.

2.2 Metodología para el desarrollo

Para la desarrollo del proyecto se busca implementar una metodología ágil que lleva el nombre de Scrum.[11] Éste es un marco de trabajo que permitirá avanzar en el proyecto de manera rápida y segura, pudiendo cambiar entre estrategias para el desarrollo del proyecto según se vaya dando el caso a través del tiempo. La metodología Scrum permite una adaptación ágil a cambios y/o reestructuraciones, sobretodo en el ámbito del desarrollo informático, por esto, es la principal razón por la cual se plantea desarrollar el proyecto bajo este marco de trabajo.[12]

Proceso de desarrollo del software: Se ha procedido a determinar una serie de pasos, los cuales siguen un orden lógico estratégicamente estructurado, que serán los escalones que en conjunto permitirán el desarrollo de la aplicación web y el API. Cada paso seguirá un orden claro y bien definido para alcanzar los objetivos propuestos y garantizar un resultado efectivo.

Levantamiento de información: Se investiga sobre distintas fuentes que tengan relación con los temas comprendidos en el proyecto en cuestión, para afianzar el conocimiento que será requerido en el desarrollo del trabajo, además de idear la estructura lógica en la cual se procederá a desarrollar el trabajo. Esto comprende los puntos a tener en cuenta para el enfoque que tendrá el desarrollo del software, haciendo uso del hardware específico que servirá para la recolección de los datos que la aplicación consumirá.

Adquisición de materiales, herramientas y hardware: Proceder con la adquisición los materiales, herramientas y todo el hardware necesario para la recolección de la información que se utilizará para almacenarse en la base de datos a través del API, que corresponde a la antena con el lector rfid y las etiquetas rfid, ambos recursos debiendo trabajar en la misma frecuencia (rango 902 - 928 MHz).

Instalación y configuración de recolectadores de datos (hardware): Instalación de etiquetas RFID para la distinción de cada dispositivo del laboratorio que pertenece al inventario de herramientas y hardware que se prestarán a estudiantes. Configuración de lector RFID para distinguir estos dispositivos que reenviará de forma estructurada a la base de datos.

Creación y configuración de la base de datos: Corresponde a la instalación y configuración de la base de datos en SQL Server para el almacenaje de toda la información que utilizará la aplicación web a través de la API web para mostrarla al usuario en las diferentes pantallas. Esta información corresponde principalmente a los datos que guardan las tablas para la funcionalidad que involucra el proyecto.

Creación de API REST: Desarrollo de servicio API REST con ASP.NET Core Web API cuyo despliegue se realizó con IIS, por sus siglás en inglés: ”Internet Information Services”. La presente web API se conectará a la base de datos configurada y traerá la información pertinente en formato JSON, utilizando los procedimientos almacenados desarrollados dependiendo del endpoint que esté siendo utilizado. Dichos endpoints internos de cada ruta del API corresponden a cada parte de la lógica empleada, a continuación se lista los mismos (para mayor información sobre la funcionalidad y ejemplos de uso de los mismos revisar la sección de documentación al final del documento):

Creación de aplicación Web: Desarrollo de aplicación Web, que tendrá como fin la monitorización y gestión por el usuario (profesor) de la información correspondiente a los estudiantes y sus registros de asistencia, además del inventario y toda la gestión de sus componentes (ítems y registros de préstamo de ítems). Para este fin, se utilizaron las siguientes tecnologías: HTML, CSS, Javascript, el Framework de .NET (ASP.NET Core

MVC) y librerías como Boostrap y Jquery. La aplicación presenta una serie de páginas a las que el usuario podrá ingresar para la gestión correspondiente, y una página para el registro de asistencia de los estudiantes por medio de su número de matrícula (a esta pantalla los estudiantes podrán ingresar sin necesidad de registrarse o iniciar sesión, y registrar dicha su respectiva matrícula a través de sus dispositivos móviles). En cuanto a las demás páginas, que serán las que utilice el usuario, tenemos las siguientes:

Despliegue del API y aplicación web: El proceso de despliegue del API se realizó en un servidor local, utilizando la plataforma IIS para garantizar su correcta ejecución y disponibilidad para los usuarios (tanto la aplicación web como la antena RFID que se conecta a un endpoint para enviar los ítems detectados). Este enfoque permitió una gestión más eficiente de la infraestructura y un mejor control de las funcionalidades ofrecidas por el API.

Por otro lado, para la aplicación web se implementó Conveyer, lo que brindó la posibilidad de acceder a ella desde cualquier dispositivo externo conectado a la red mediante su dirección IP. Ampliando significativamente el alcance de la aplicación y permitiendo a los usuarios y sobretodo a los estudiantes poder conectarse desde sus dispositivos móviles para el fin de registrar su asistencia.

En general, la combinación de IIS y Conveyer permitió un despliegue óptimo y eficiente del API y la aplicación web, lo que influyó positivamente en la mejora de la experiencia de los usuarios y garantizó un correcto funcionamiento de las funcionalidades implementadas.

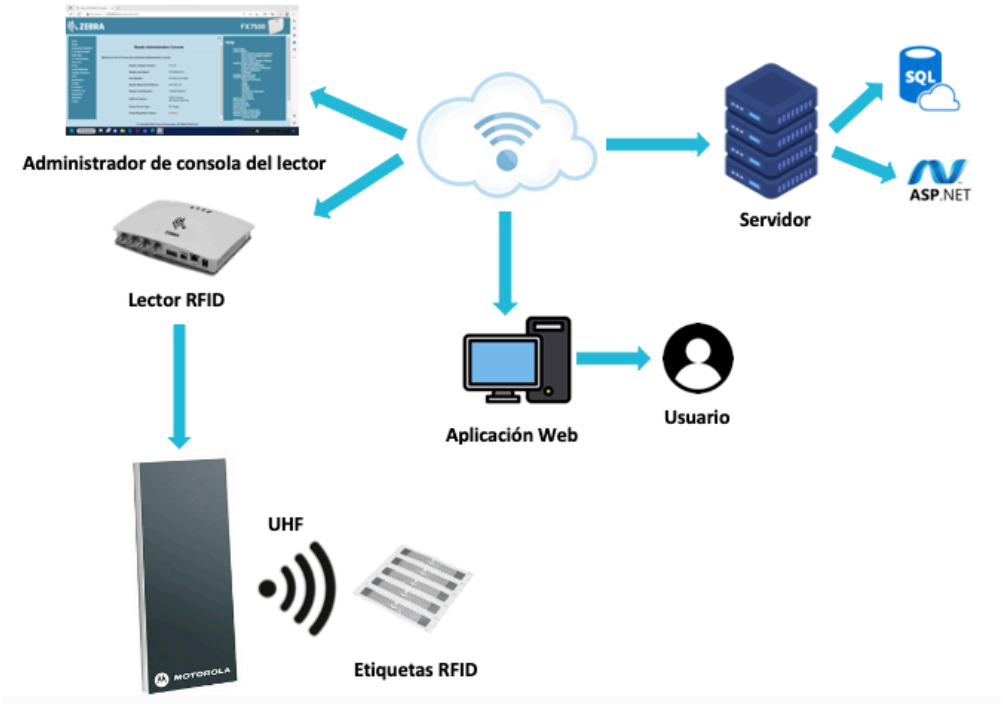

Figura 2.1: Diagrama de arquitectura del sistema

Proceso de Implementación del hardware: Tal como se muestra en la Figura 2.1, el diagrama de arquitectura del sistema detalla de forma esquematizada lo que se ha desarrollado en este proyecto. El sistema está compuesto por aproximadamente tres fases: la primera fase se basa en la captura de datos de identificación de los recursos a los que se encuentran adheridas las etiquetas RFID, después estos datos son enviados a través de la red a la segunda fase, que consiste en la gestión y almacenamiento de datos, donde se tiene un servidor que contiene la base de datos(SQL) y los servicios web(Asp.net) que están a la espera de las solicitudes ya sean para el almacenamiento de los datos o para que sean enviados a la última fase, que radica en mostrar esos datos al usuario a través de la aplicación web que también muestra el registro de asistencia y el modo en el que se encuentre el laboratorio, sea éste, modo clases o modo laboratorio abierto.

Lectura de datos: El sistema RFID implementado en este proyecto consta de los siguientes componentes: Lector fijo RFID FX7500, Antena Motorola AN400 RFID y Etiquetas RFID.

Para poder identificar cada uno de los recursos del laboratorio se colocó una etiqueta RFID sobre ellos, por tanto, cuando se estos se encuentren dentro del rango de radiofrecuencia que produce la antena del lector se enviarán de forma inmediata sus IDs hacia el lector, el cual los exportará al servidor web puesto que se configuraron los respectivos endpoints en el modulo Zebra IoT Connector que se encuentra en el administrador de consola de lector.

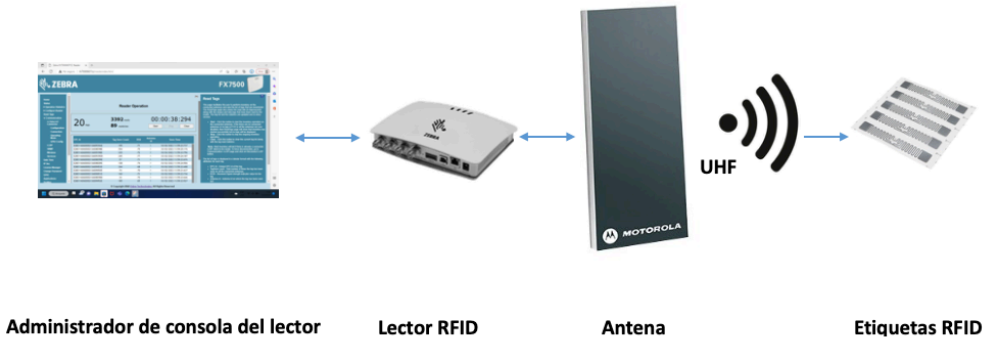

Figura 2.2: Diagrama de lectura de datos

Como se puede observar en el diagrama de lectura de datos de la Figura 2.2, el sistema RFID de nuestro proyecto está compuesto básicamente por tres fases:

1. En el administrador de consola de lector se configuran los endpoints para se poder iniciar la lectura.
2. La antena comienza a emitir un campo de radiofrecuencia, a través de ondas que harán que las etiquetas que se encuentren dentro de dicho rango se activen y envíen su ID.
3. Los datos son recibidos por el lector y los envía al servidor web para que se puedan visualizar de forma rápida en la aplicación web y ser almacenados en la base de datos.[13]

En conclusión, el seguimiento de un orden lógico de procesos durante cada etapa del desarrollo del sistema permitió una fluidez y agilidad para la construcción del mismo, que finalmente sirvió para que el producto final, en este caso la aplicación web y el API, sean viables y eficientes, funcionando con un rendimiento alto, integrándose con los sensores y cumpliendo a su vez con lo planteado, como se podrá observar en en sus respectivos resultados en el siguiente capítulo.

CAPÍTULO 3

3. PRUEBAS Y RESULTADOS

En este capítulo se presentará con una gran cantidad de detalles el entorno que se ha configurado para las pruebas correspondientes tanto para el hardware como para el software. Luego de ello, se analizarán todos los resultados obtenidos de dichas pruebas de manera exhaustiva.

3.1 Pruebas

Prueba de estrés - API web Esta prueba fue pensada para evaluar la respuesta en disponibilidad y estabilidad del API a una base de 100 usuarios a 100 peticiones simultáneamente. Para ello se utilizó el software JMeter de Apache. En primer lugar se configuró la cantidad de hilos (usuarios) que ejecutaron las peticiones HTTP, junto con el número de peticiones por cada uno de ellos. Después se configuraron cada uno de los endpoints que serían ejecutados en las pruebas. En las siguientes figuras se puede apreciar dicha configuración:

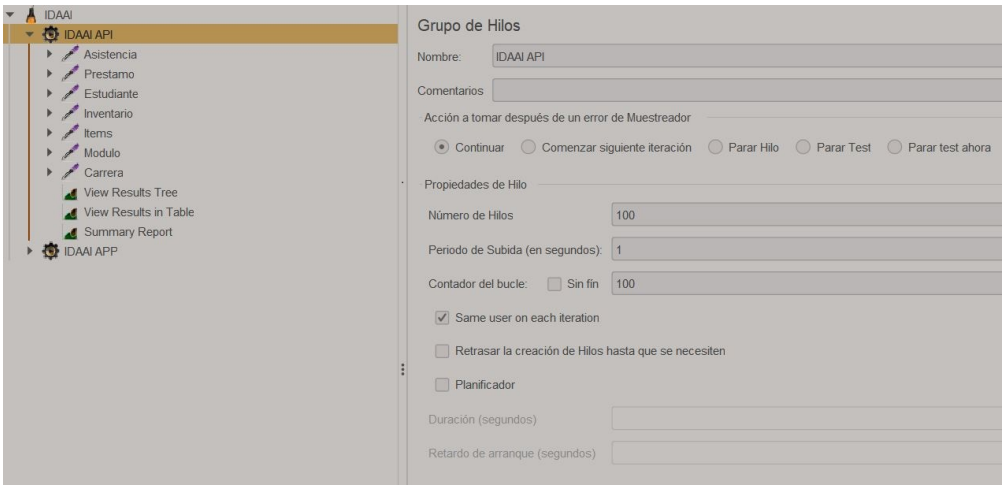

Figura 3.1: Pruebas de estrés - API: Configuración de hilos y llamadas (carga de usuarios)

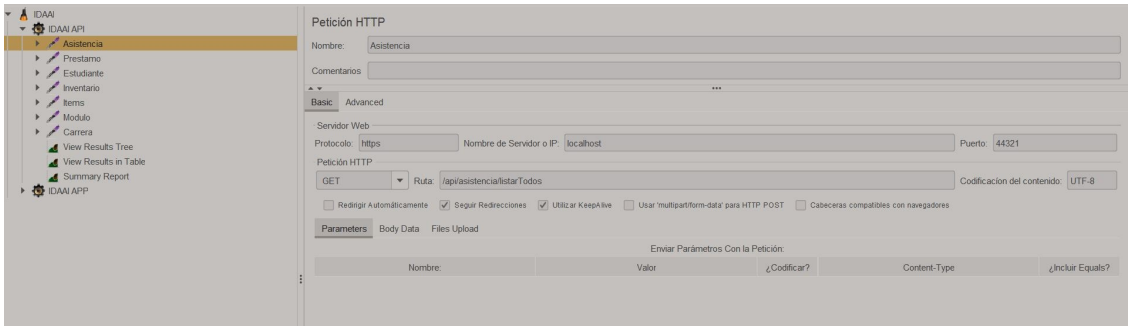

Figura 3.2: Pruebas de estrés - API: Configuración de endpoints

En la figura 3.1 podemos visualizar la configuración del grupo de hilos establecido para las pruebas de los endpoints del API, donde el número de hilos corresponde una simulación del número de usuarios. El periodo de subida se refiere al tiempo que tarda en aumentar el número de usuarios simulados en un grupo de hilos. En otras palabras, el periodo de subida es el tiempo que se establece para que los usuarios simulados vayan aumentando gradualmente en número, en lugar de iniciar todos al mismo tiempo. Por ejemplo, si se tiene un grupo de hilos con 100 usuarios y un periodo de subida de 20 segundos, JMeter agregará cinco usuarios por segundo hasta llegar a la carga máxima de 100 usuarios. El periodo de subida se utiliza para simular un crecimiento más realista de la carga de trabajo en una aplicación o sitio web.[14] Por ultimo, el contador de bucle se refiere a la cantidad de ejecuciones de los endpoints que efectuara cada hilo.

En la figura 3.2 se muestra la configuración específica de un endpoint del API utilizado en las pruebas, en la pestaña ”Basic” se encuentra el protocolo al que pertenece el endpoint, la dirección del host del API se indica en los campos ”Nombre de servidor o IP” y ”Puerto”. La sección ”Petición HTTP” indica el tipo de petición, como en el caso del ejemplo en la imagen, que es ”GET” porque se trata de un endpoint de consulta de datos, mientras que la dirección específica del endpoint se encuentra en ”Ruta”.

Prueba de estrés - Aplicación web Esta prueba se llevó a cabo para evaluar el rendimiento y la estabilidad de la aplicación web que se conecta a la API desarrollada, bajo una carga simulada de 100 usuarios realizando 100 peticiones simultáneamente. Se utilizó también para este caso el software JMeter de Apache para configurar el número de usuarios y peticiones que se ejecutarían en las pruebas. La configuración incluyó el número de hilos de usuarios y el número de peticiones por cada hilo. También se configuraron las peticiones que se ejecutarían en las pruebas. En las siguientes imágenes se puede visualizar dicha configuración realizada:

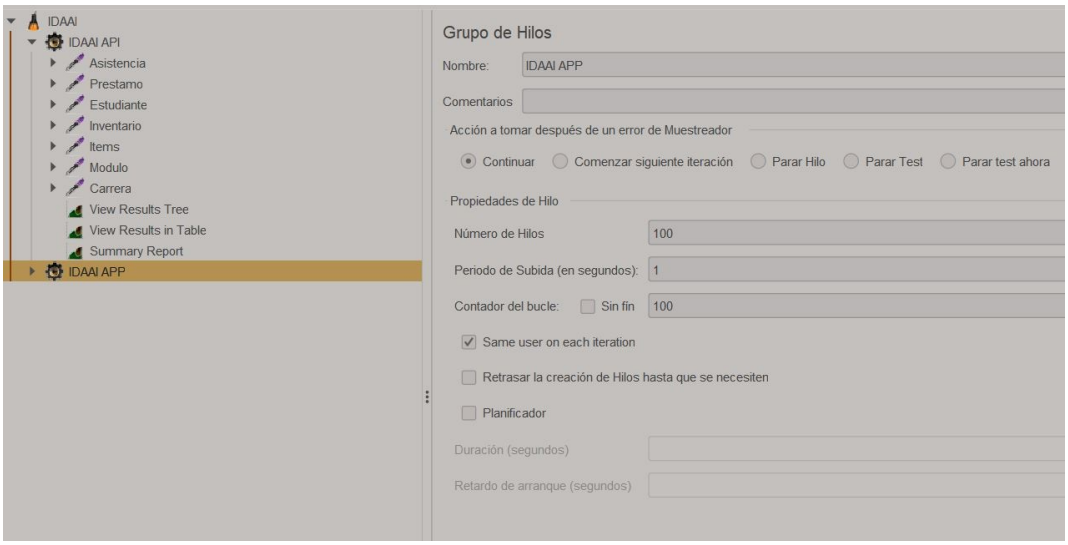

Figura 3.3: Pruebas de estrés - Aplicaicón web: Configuración de hilos y llamadas (carga de usuarios)

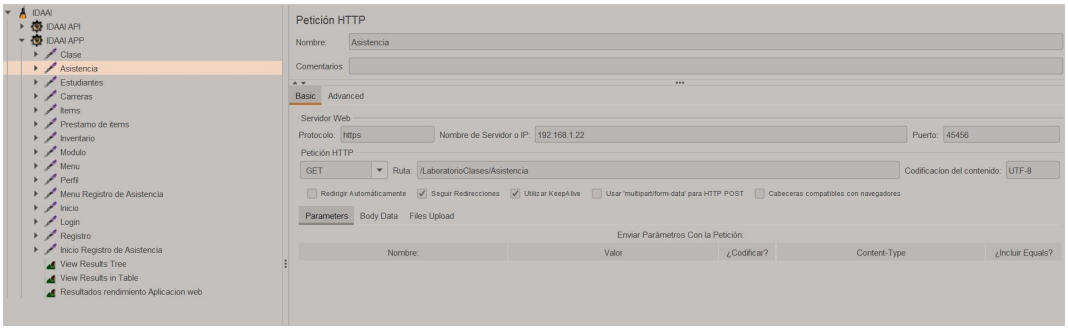

Figura 3.4: Pruebas de estrés - Aplicaicón we: Configuración de endpoints

En la figura 3.3 podemos visualizar la configuración del grupo de hilos establecido para las pruebas de las pantallas de la aplicación web que se conectan al API desarrollado, donde podemos visualizar una configuración similar a la del API, que son ”Número de Hilos”: 100, ”Período de subida (en segundos)”: 1, y ”Contador del bucle”: 100.

En la figura 3.4 se muestra la configuración específica de una pantalla de la aplicación web que se conecta a un endpoint del API, en la pestaña ”Basic” se encuentra el protocolo al que pertenece el endpoint, la dirección del host del API se indica en los campos ”Nombre de servidor o IP” y ”Puerto”. La sección ”Petición HTTP” indica el tipo de petición, como en el caso del ejemplo en la imagen, que es ”GET” porque se trata de un endpoint de consulta de datos, mientras que la dirección específica de la ruta de la aplicación web se encuentra en ”Ruta”.

Prueba de lectura de etiquetas RFID Antes de realizar las configuraciones respectivas para exportar los datos leídos de las etiquetas RFID al servidor web, se hicieron pruebas de lecturas de manera local para demostrar la funcionalidad del lector y de las etiquetas como se puede observar en la Figura 3.5. Para la prueba inicial se conectó la antena al lector y el lector al computador por cable de red para poder acceder localmente al administrador de consola y verificar que las etiquetas estaban en buen funcionamiento.

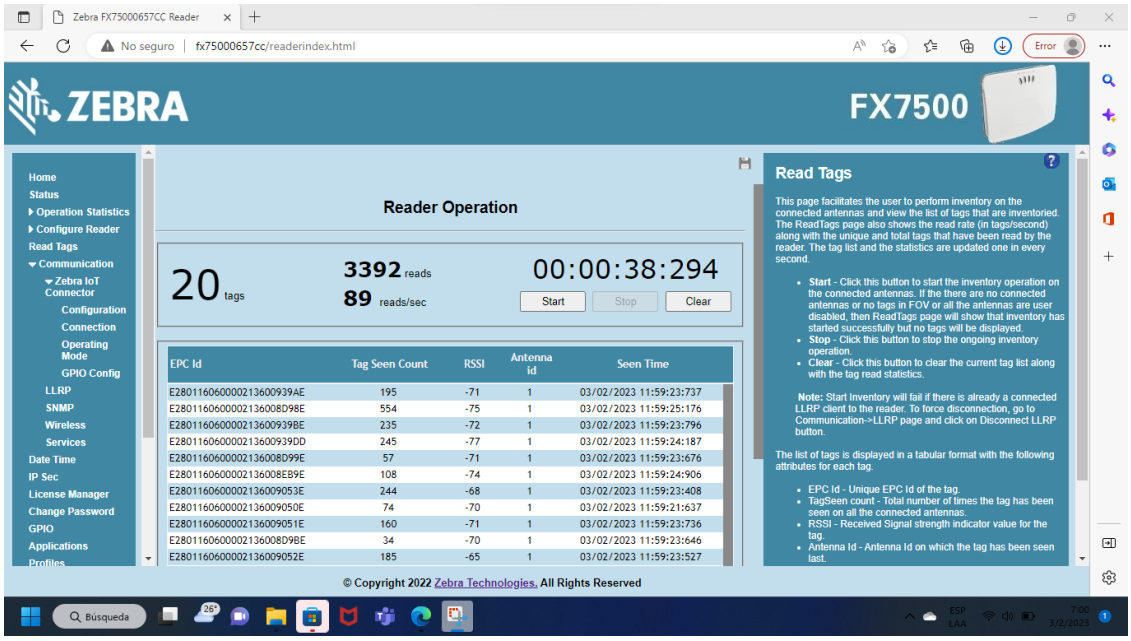

Figura 3.5: Resultado de lectura de las etiquetas RFID

3.2 Análisis de resultados

Análisis de resultados de prueba de estrés - API web

Tabla 3.1: Resultados pruebas de estrés - API (1)

Endpoint	# Muestras	Media	Min	Máx	Desv. Est.
Asistencia	10000	580	11	50333	70062
Prestamo	10000	567	10	62602	91936
Estudiante	10000	559	11	66875	101381
Inventario	10000	555	9	65244	107050
Items	10000	550	10	58970	82281
Modulo	10000	536	10	10917	21177
Carrera	10000	540	9	47887	51670
Total	**70000**	**555**	**9**	**66875**	**80179**

Tabla 3.2: Resultados pruebas de estrés - API (2)

Endpoint	% Error	Rendimiento	KB/s Recibidos	KB/s enviados	Media de bytes
Asistencia	0	2490052	6335	428	26050
Prestamo	0	2492529	4496	482	18470
Estudiante	0	2498876	4937	429	20230
Inventario	0	2503110	1748	440	7150
Items	0	2506360	1339	448	5470
Modulo	0	2509990	2052	422	8370
Carrera	0	2512841	1335	425	5440
Total	**0**	**17427458**	**22168**	**3059**	**13026**

De los resultados mostrados en las tablas 3.1 y 3.2 se puede observar que para una base de 100 usuarios a 100 peticiones HTTP al API desarrollado, el API está funcionando con un desempeño satisfactorio. Los tiempos de respuesta promedio de las peticiones simultáneas oscilan entre 536 y 580 milisegundos, lo que indica un tiempo de respuesta rápido y eficiente. Este resultado muestra que la API es capaz de manejar una gran cantidad de solicitudes de manera eficiente y de proporcionar respuestas rápidas a los usuarios. Además, este resultado también sugiere que la infraestructura y la implementación de la API están optimizadas para brindar un alto rendimiento y escalabilidad.

Los resultados de rendimiento muestran tiempos de respuesta bastante estables y eficientes. Al tener una cantidad significativa de 100 usuarios realizando 100 peticiones simultáneamente, el API está respondiendo en un promedio de 24,9 a 25,1 peticiones por segundo. Indicando un rendimiento sólido y una capacidad de manejar una carga significativa sin una disminución significativa en los tiempos de respuesta.

Por otra parte, existe una variabilidad en los datos de desviación estándar, lo que se refleja en los valores altos que van desde 211,77 hasta 1070,50. Esto puede indicar que la respuesta del API es algo inconsistente, ya que la diferencia entre los tiempos de respuesta más lentos y ás rápidos puede ser significativa. Se recomienda investigar las causas de esta variabilidad, como la carga en el servidor o la complejidad de las solicitudes, y tomar medidas para mejorar la estabilidad de la respuesta de la API.

En los resultados de los porcentajes de error, podemos observar un excelente desempeño en términos de fiabilidad y consistencia. Al observar que el porcentaje de error se sitúa en 0 en todas las pruebas, se puede afirmar que la API está proporcionando una respuesta eficiente y confiable para cada petición. Esto es una indicación clara de un buen diseño y ejecución de la API y su capacidad de manejar una alta cantidad de peticiones sin fallar. Este resultado es muy positivo y refleja un buen rendimiento en términos de escalabilidad y fiabilidad del API desarrollada.

En cuanto a la cantidad de datos recibidos y enviados, se observa un rango sólido desde 13,35 hasta 63,35 KB/s para los datos recibidos y un rango estrecho desde 4,22 hasta 4,82 KB/s para los datos enviados. Esto sugiere un buen equilibrio entre la cantidad de información recibida y enviada, permitiendo un procesamiento eficiente de los datos.

Los valores registrados en el promedio de bytes indican una eficiencia moderada en la transferencia de datos y pueden ser un indicador de la cantidad de información que se está enviando y recibiendo en cada solicitud. Es importante considerar también que el tamaño promedio de los datos puede tener un impacto en el tiempo de respuesta total y en la carga en el servidor.

En conclusión, los resultados de los análisis de rendimiento del API web indican que la aplicación cumplió con las expectativas de funcionamiento. Los tiempos de respuesta, el porcentaje de error, la cantidad de datos transmitidos y otros indicadores mostraron resultados aceptables y dentro de lo esperado. Esto sugiere que la aplicación es viable y funciona adecuadamente en el contexto actual que es el proyecto del trabajo de tesis. Además, esto también puede ser una indicación de que el API podría tener potencial para ser utilizada en otros contextos similares, si se requiere.

Análisis de resultados de prueba de estrés - Aplicación web

Tabla 3.3: Resultados pruebas de estrés - Aplicación web (1)

Pantalla	**# Muestras**	**Media**	**Min**	**Máx**	**Desv. Est.**
Clase	10000	279	6	4657	190,3
Asistencia	10000	271	7	4431	167,15
Estudiantes	10000	273	8	2721	167,63
Carreras	10000	270	10	3204	167,45
Items	10000	272	8	2402	163,17
Prestamo de items	10000	272	10	3042	174,25
Inventario	10000	273	7	3190	172,31
Modulo	10000	273	6	2882	168,92
Menu	10000	271	5	3165	166,41
Perfil	10000	269	6	2418	161,85
Menu Asistencia	10000	154	4	2335	123,6
Inicio	10000	144	3	1818	116,38
Login	10000	149	4	1903	117,43
Registro	10000	150	4	2302	116,71
Inicio Asistencia	10000	151	3	4308	126,12
Total	**150000**	**231**	**3**	**4657**	**165,69**

Tabla 3.4: Resultados pruebas de estrés - Aplicación web (2)

Pantalla	**% Error**	**Rendimiento**	**KB/s Recibidos**	**KB/s enviados**	**Media de bytes**
Clase	0,025	28,47105	326,07	10,3	11727,5
Asistencia	0,0275	28,51521	326.07	10,57	11709,4
Estudiantes	0,0281	28,56172	326,41	10,57	11702,4
Carreras	0,0283	28,56661	326,36	10,42	11698,5
Items	0,0271	28,57223	326,66	10,26	11707,1
Prestamo de items	0,0255	28,58204	327,29	10,5	11725,8
Inventario	0,0272	28,58727	326,97	10,6	11712,2
Modulo	0,0237	28,59161	327,83	10,41	11741,1
Menu	0,0246	28,59905	327,07	9,25	11710,8
Perfil	0,029	28,60461	326,66	10,38	11694
Menu Asistencia	0,0152	28,61263	285,64	4,9	10222,6
Inicio	0,0135	28,61632	211,69	4,33	7575
Login	0,0112	28,61935	324,96	4,5	11627
Registro	0,0121	28,62516	329,81	4,58	11798,2
Inicio Asistencia	0,0146	28,62795	288,08	4,9	10304,4
Total	**0,02217**	**426,90771**	**4687,53**	**125,96**	**11243,7**

Tal como se ve en las tablas 3.3 y 3.4 se puede concluir que en el análisis de los resultados de la aplicación web, se pueden identificar varios aspectos positivos que demuestran su eficacia y utilidad para ser empleados en el laboratorio de sistemas telemáticos. Con los resultados que hemos obtenido, podemos apreciar lo siguiente:

Tiempo de respuesta: El tiempo de respuesta en general de la aplicación web es corta, por lo que si sitúa como rápida y eficiente, además de indicar un buen rendimiento y por ende una buena experiencia de usuario.

Eficiencia en el uso de recursos: La aplicación web utiliza de manera eficiente los recursos del sistema, lo que permite un buen rendimiento durante la utilización de la misma.

Usabilidad: La aplicación web desarollada, a su vez, posee una interfaz intuitiva, lo que la hace fácil de comprender, utilizar y permite al usuario (el profesor), realizar las respectivas tareas de monitorización o gestión de manera rápida y eficiente.

Estabilidad: La aplicación web es estable y se mantiene funcionando de manera adecuada, lo que permite una buena experiencia de usuario y un buen rendimiento en el uso de la aplicación.

Los resultados obtenidos muestran que la aplicación web es una herramienta eficiente para el contexto mencionado del laboratorio de sistemas telemáticos. Además, estos resultados indican que la aplicación cumple con las metas originales establecidas para el proyecto, lo que demuestra su viabilidad.

En cuanto a las posibles limitaciones y mejoras, se puede considerar la posibilidad de realizar un seguimiento continuo de los resultados para identificar posibles problemas y mejorar la aplicación en el futuro. También se pueden considerar sugerencias del/los usuario/s y evaluar si sería útil implementar nuevas funcionalidades para mejorar la experiencia de usuario.

En conclusión, la aplicación web desarrollada se presenta como una herramienta útil y de buen rendimiento, cumpliendo con las objetivos establecidos y ofreciendo además una buena experiencia de usuario.

Análisis de resultados de la prueba lectura de etiquetas RFID Una vez iniciado el proceso de lectura, la antena de manera inmediata comienza a emitir un campo de radiofrecuencia activando a las etiquetas que se encuentre dentro del rango y permitiendo así enviar el ID al lector e inmediatamente reenviarlo a la aplicación web. Al realizar los cambios en las configuraciones del lector agregando los nuevos endpoints, sea este el servidor web al que se envián los datos, también se puede configurar la forma en la que se envían los mismos al servidor, permitiendo que se pueda enviar lo estrictamente necesario para mostrar en la aplicación web, visualizando el ID y el estado del dispositivo que posee ese etiqueta RFID, sea disponible o no disponible en caso que no se detecte dentro del área de lectura, tal como se puede observar en al Figura 3.6.

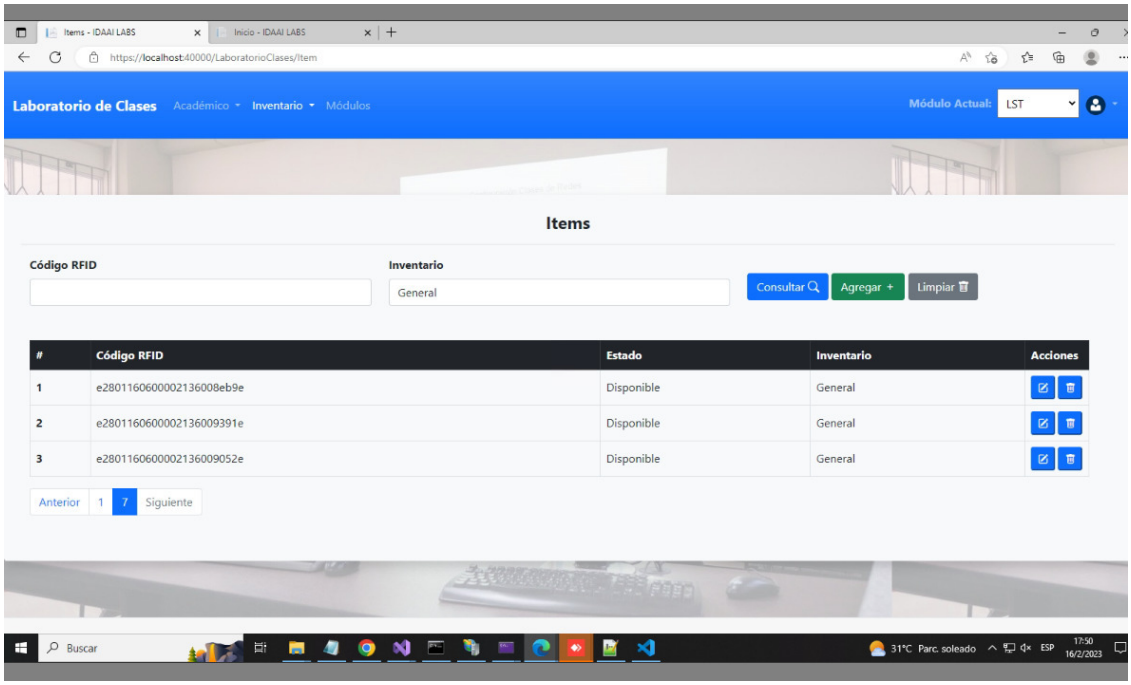

Figura 3.6: Resultados lectura visualizada en aplicación web

CAPÍTULO 4

4. CONCLUSIONES Y LINEAS FUTURAS

En función de los resultados alcanzados y el proceso de desarrollo que abarcó cada fase del proyecto, se puede concluir lo siguiente:

4.1 Conclusiones

- El uso de un servidor web para que los estudiantes registren su asistencia personalmente es una solución conveniente que previene la necesidad de que el profesor tenga que dedicar tiempo a esta tarea. Gracias a este sistema, no es necesaria la intervención del profesor, cumpliendo con el primer objetivo del presente trabajo. La implementación de este sistema demuestra la capacidad y utilidad del mismo, asegurando una gestión organizada y adecuada de la asistencia de los estudiantes.

- El uso de la aplicación web que consulta la disponibilidad de los recursos del laboratorio por medio de las etiquetas RFID, brinda una solución competente para la gestión de los recursos y dispositivos. La información se puede obtener en tiempo real desde cualquier lugar, lo que optimiza el tiempo de búsqueda y facilita el acceso a los mismos, en comparación con la búsqueda manual de los registros de inventario. La tecnología RFID y la aplicación web trabajan juntas para mantener actualizada la información y lograr una gestión eficiente de los recursos del laboratorio.

- En comparación a un proceso manual para realizar inventarios, el sistema RFID ha demostrado ser una solución de mucha utilidad para la gestión de recursos y dispositivos, debido a que no se necesita mano de obra humana para la

actualización de los mismos, demostrando una reducción en costos, una mayor rápidez y facilidad en la identificación automática de los dispositivos, sin costo adicional por lectura. Esto permite llevar a cabo proyectos relacionados a estos sistemas de forma efectiva y económica.

4.2 Recomendaciones

- Se recomienda utilizar un marco de trabajo especializado para la construcción de la interfaz de usuario, como React, debido a la presencia de una gran cantidad de elementos similares o repetidos en la aplicación web. Este framework se destaca como la mejor opción actual, ya que agiliza el proceso de desarrollo y ayuda a reducir la cantidad de código duplicado.

- Se recomienda crear los endpoints del API con la lógica de negocio implementada en los procedimientos almacenados de SQL Server y agruparlos de tal manera que haya un solo endpoint (por ejemplo, ”listar”) para cada tipo de método de petición HTTP (en lugar de tener varios puntos finales como ”listarPorNombres”, ”listarPorModulo”, etc.). Esto facilitará el uso de los mismos, tanto directamente como a través del consumo del servicio.

- Se recomienda utilizar una frecuencia de 13.56 MHz o 125 kHz para el lector y las etiquetas RFID, ya que son mucho más comunes, por lo tanto, serían más accesibles.

- Cuando se trabaja con este tipo de equipos descontinuados es recomendable siempre actualizar el firmware, debido que hay módulos en versiones anteriores a los que solo se podían acceder con una licencia actualizada, pero con las actualizaciones estos módulos han emigrado a otros, permitiendo hacer configuraciones que antes estaban bloqueadas sin necesidad de tener una licencia.

4.3 Líneas futuras

- Se sugiere explorar la posibilidad de expandir el software existente en una aplicación móvil, con el objetivo de brindar una experiencia de usuario más accesible y conveniente a través de dispositivos móviles. La demanda por soluciones tecnológicas accesibles en cualquier momento y lugar ha aumentado significativamente en los últimos años, y desarrollar una aplicación móvil podría ser una oportunidad valiosa para ampliar la alcance e impacto del software.

- Se sugiere investigar la posibilidad de integrar los resultados de la cámara de conteo de estudiantes dentro del aplicativo web y la aplicación móvil. Esta integración podría brindar una mayor eficiencia y eficacia en la gestión y monitoreo de la asistencia de los estudiantes. Al tener acceso a esta información de manera oportuna y en tiempo real, los usuarios (profesores) pueden tomar decisiones informadas y mejorar la calidad de la educación.

- Se recomienda ampliar la funcionalidad de la aplicación de registro de asistencia, incluyendo un sistema de reconocimiento biométrico basado en huellas digitales. Esto se lograría mediante la integración de un hardware especializado para el reconocimiento de huellas dactilares en la entrada del laboratorio, permitiendo una toma de asistencia más precisa y segura para los estudiantes. Con esta mejora, se puede aumentar la eficiencia y la efectividad en la gestión de la asistencia, y al mismo tiempo proporcionar una experiencia más moderna y conveniente para los mismos estudiantes.

BIBLIOGRAFÍA

[1] A. Verdezoto, “Nuevo laboratorio de telemática en la espol,” *Noticias ESPOL*, Nov 2014. Fuente: Noticias ESPOL, disponible en url http://noticias.espol.edu.ec/article/nuevo-laboratorio-de-telematica-en-la-espol, consultado el 17 de febrero de 2023.

[2] Zebra, “Lector fijo de rfid fx7500,” *Zebra*, 2022. Fuente: Zebra, disponible en url https://www.zebra.com/la/es/products/rfid/rfid-readers/fx7500.html, consultado el 17 de febrero de 2023.

[3] U. de Málaga, “Entornos inteligentes,” *Máster Oficial en Sistemas Electrónicos para Entornos Inteligentes*, 2015. Fuente: Master SEEI, disponible en url https://www.masterseeiuma.es/entornos-inteligentes-1/, consultado el 17 de febrero de 2023.

[4] A. Decoración, “El edificio inteligente,” *Portal de arquitectura Arqhys.com*, 2011. Fuente: Portal de arquitectura Arqhys, disponible en url https://www.arqhys.com/decoracion/el-edificio-inteligente.html, consultado el 17 de febrero de 2023.

[5] M. A. A. Pinedo, “Edificios inteligentes,” *Filadd*, 2020. Fuente: Filadd, disponible en url https://filadd.com/doc/i2-ft2-2004-edificiosinteligentes-pdf, consultado el 17 de febrero de 2023.

[6] U. N. de la Plata, “Edificios inteligentes con integraciÓn de variables y diseÑo energÉtico. revisiÓn del estado del arte para un modelo local,” *Máster Oficial en Sistemas Electrónicos para Entornos Inteligentes*, 2001. Fuente: Universidad Nacional de la Plata, disponible en url http://sedici.unlp.edu.ar/bitstream/handle/10915/79738/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y, consultado el 17 de febrero de 2023.

[7] R. Hat, “¿qué es el internet de las cosas (iot)?,” *Read Hat*, 2019. Fuente: Read Hat, disponible en url https://www.redhat.com/es/topics/internet-of-things/what-is-iot, consultado el 17 de febrero de 2023.

[8] C. M.-G. osé Guillermo Hernández-Calderón, Edgard Benítez-Guerrero, “Ambientes inteligentes en contextos educativos: Modelo y arquitectura,” *Universidad Veracruzana*, 2014. Fuente: Danilopy Blog, disponible en url https://www.rcs.cic.ipn.mx/2014_77/Ambientes%20inteligentes%20en%20contextos%20educativos_%20modelo%20y%20arquitectura.pdf, consultado el 17 de febrero de 2023.

[9] O. H. Saldaño, “Tesis de grado. metodología de la investigación,” Fecha desconocida. Archivo PDF, disponible en url http://imagenes.mailxmail.com/cursos/pdf/6/tesis-grado-metodologia-investigacion-23736.pdf, consultado el 17 de febrero de 2023.

[10] G. Arteaga, “Cómo escribir una metodología,” 11 2021. Fuente en línea, disponible en url https://www.testsiteforme.com/como-escribir-una-metodologia-de-investigacion/, consultado el 17 de febrero de 2023.

[11] IEBSchool, “Metodología scrum: ¿qué es agile scrum y para qué sirve?,” *IEBSchool Blog*, 2019. Fuente: IEBSchool Blog, disponible en url https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/, consultado el 17 de febrero de 2023.

[12] P. Deemer, G. Benefield, C. Larman, and B. Vodde, “Información básica de scrum,” 2009. Fuente: Scrum Training Institute, disponible en url https://www.scrumtraininginstitute.com/basic-scrum-knowledge/, consultado el 17 de febrero de 2023.

[13] Z. Technologies, “Ziot connector web interface,” *Zebra Devs*, 2023. Fuente: Zebra, disponible en url https://zebradevs.github.io/rfid-ziotc-docs/setupziotc/index.html#install-mqtt-client, consultado el 17 de febrero de 2023.

[14] PNT, “Load testing con jmeter,” 2021. Fuente: Somos PNT Blog, disponible en url https://somospnt.com/blog/91-load-testing-con-jmeter, consultado el 17 de febrero de 2023.

[15] DANILOPY, “Ambiente inteligente / inteligencia ambiental,” *Wordpress*, 2019. Fuente: Danilopy Blog, disponible en url https://danilopy.wordpress.com/2019/09/18/ambiente-inteligente-inteligencia-ambiental/, consultado el 17 de febrero de 2023.

[16] G. M. Smith, “¿qué es un sensor y qué hace?,” *Dewesoft*, 2020. Fuente: Dewesoft Blog, disponible en url https://dewesoft.com/es/daq/que-es-un-sensor/, consultado el 17 de febrero de 2023.

[17] MuleSoft, “What is an api?,” *MuleSoft Resources*, 2021. Source: MuleSoft Resources, available at url https://www.mulesoft.com/resources/api/what-is-an-api, accessed on February 17, 2023.

[18] A. Taylor, “Cómo crear puntos finales y por qué los necesita,” *Blog de AppMaster*, 2022. Fuente: Blog de AppMaster, disponible en url https://appmaster.io/es/blog/como-crear-puntos-finales-y-por-que-los-necesita, consultado el 17 de febrero de 2023.

[19] D. RFID, “Etiquetas rfid: qué son y qué aplicaciones tienen,” *Dipole RFID*, sin fecha. Fuente: Dipole RFID Blog, disponible en url https://www.dipolerfid.es/blog-rfid/etiquetas-rfid-y-aplicaciones, consultado el 17 de febrero de 2023.

[20] Motorola, “Rfid antenna family,” *Motorola*, 2008. Fuente: Motorola, disponible en url http://rfidctp.com/wp-content/uploads/pdf/Moto-AN200-400-480.pdf, consultado el 17 de febrero de 2023.

[21] Zebra, “Fx7500 fixed rfid reader,” *Zebra*, 2022. Fuente: Zebra, disponible en url https://www.zebra.com/gb/en/products/spec-sheets/rfid/rfid-readers/fx7500.html, consultado el 17 de febrero de 2023.

[22] YARONGTECH, “Etiqueta rfid uhf,” *Amazon*, 2023. Fuente: Amazon, disponible en url https://www.amazon.com/-/es/dp/B01LYBKMYM?ref=ppx_yo2ov_dt_b_product_details&th=1, consultado el 17 de febrero de 2023.

APÉNDICES

Apéndice A: Documentación IDAAI API

A continuación, se presenta la sección correspondiente a la API web desarrollada con ASP.NET Core Web API, con los ejemplos de pruebas y resultados pertinentes para cada endpoint dentro del servicio.

Consultas en API web: Se obtuvo como resultado las respuestas del servicio a través de las consultas en los diferentes End devices que corresponden a cada sección del proyecto, de los que se obtuvieron los siguientes resultados:

Servicio Estudiante: Este servicio contiene los End devices que corresponden a todas las peticiones relacionadas a los estudiantes. Comprendiendo las consultas por diferentes campos, y peticiones de creación, actualización y eliminación de estudiantes. Para los endpoints de consulta se tendrán dos campos adicionales: ”RecordsPorPagina” y ”Pagina”, que sirven para la paginación de la lista de resultados.

- **Consultar por Email [GET]:** End point que recibe dos campos: ”Email” y ”Modulo” del estudiante que se consulta y da como resultado un JSON con la información del estudiante encontrado, conteniendo los siguientes campos como respuesta: ”Nombres”, ”Apellidos”, ”Matricula”, ”Email”, ”Direccion”, ”Carrera”, y ”Modulo”.
- **Consultar por matrícula [GET]:** End point que recibe dos campos: ”Matricula” y ”Modulo” del estudiante que se consulta y da como resultado un JSON con la información del estudiante encontrado, conteniendo los siguientes campos como respuesta: ”Nombres”, ”Apellidos”, ”Matricula”, ”Email”, ”Direccion”, ”Carrera”, y ”Modulo”.

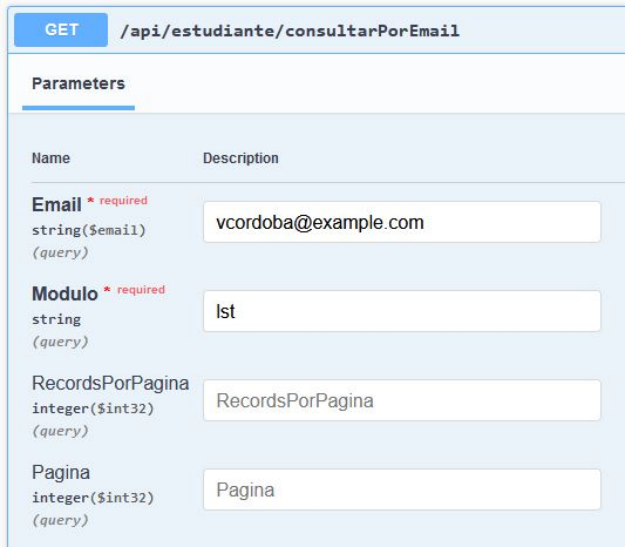

Figura A1: Servicio Estudiante: Consulta end device ”consultarPorEmail”

Figura A2: Servicio Estudiante: Respuesta end device ”consultarPorEmail”

- **Listar por nombres [GET]:** End point que recibe tres campos: ”Nombres”, ”Apellidos”, y ”Modulo” del estudiante que se consulta y da como resultado un JSON con una lista de la información de los estudiantes encontrados, conteniendo los siguientes campos como respuesta: ”Nombres”, ”Apellidos”, ”Matricula”, ”Email”, ”Direccion”, ”Carrera”, y ”Modulo”.

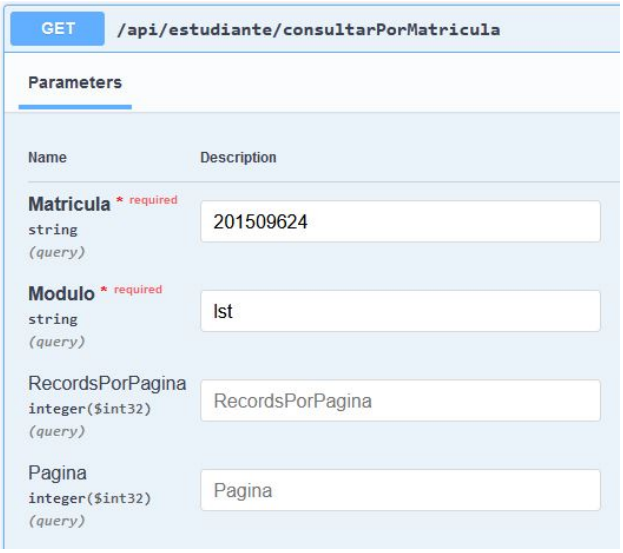

Figura A3: Servicio Estudiante: Consulta end device ”consultarPorMatricula”

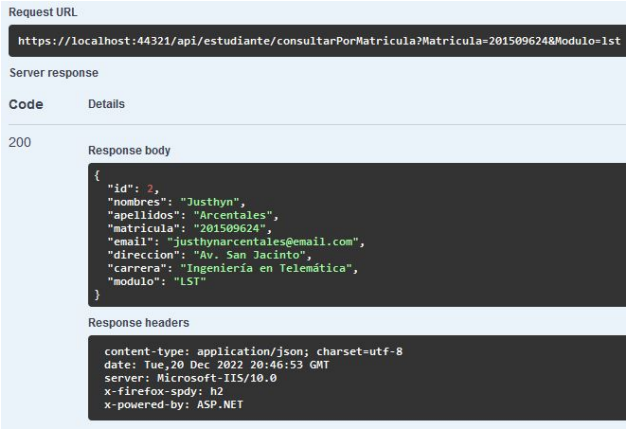

Figura A4: Servicio Estudiante: Respuesta end device ”consultarPorMatricula”

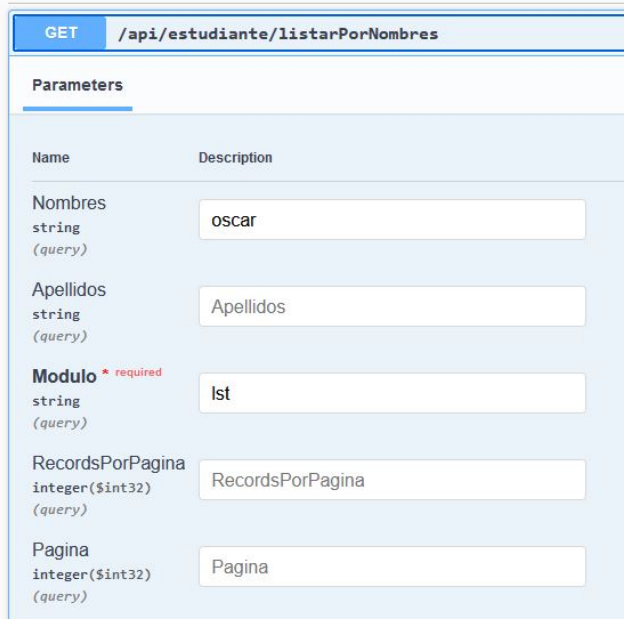

Figura A5: Servicio Estudiante: Consulta end device ”listarPorNombres”

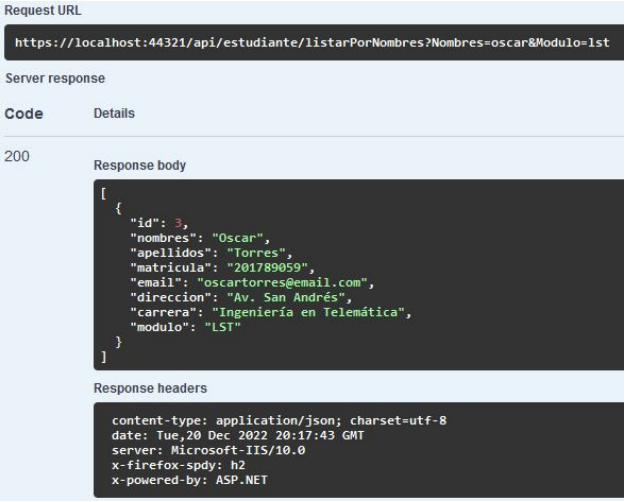

Figura A6: Servicio Estudiante: Respuesta end device ”listarPorNombres”

- **Listar por carrera [GET]:** End point que recibe dos campos: ”Carrera” y ”Modulo” del estudiante que se consulta y da como resultado un JSON con una lista de la información de los estudiantes encontrados, conteniendo los siguientes campos como respuesta: ”Nombres”, ”Apellidos”, ”Matricula”, ”Email”, ”Direccion”, ”Carrera”, y ”Modulo”.

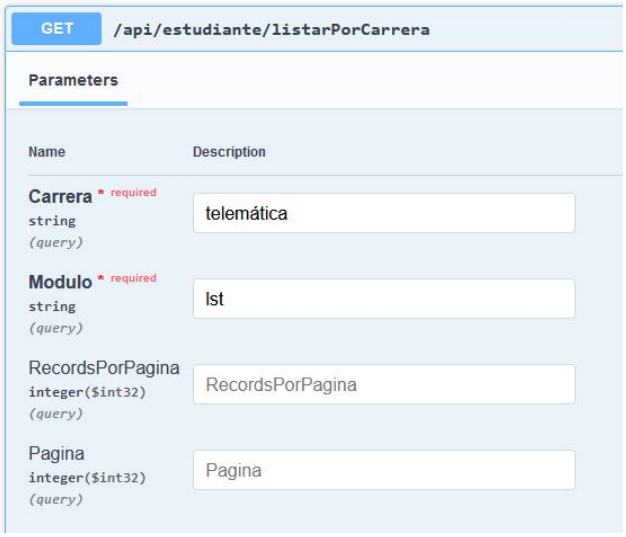

Figura A7: Servicio Estudiante: Consulta end device ”listarPorCarrera”

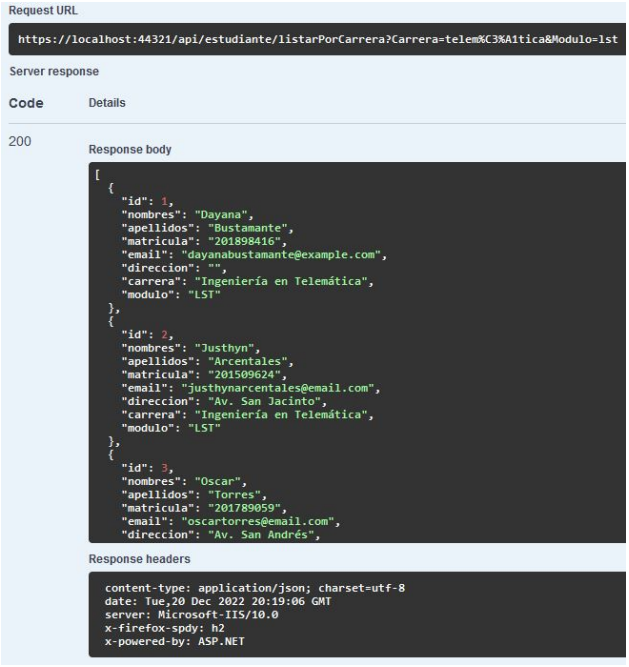

Figura A8: Servicio Estudiante: Respuesta end device ”listarPorCarrera”

- **Listar por módulo [GET]:** End point que recibe el campo ”Modulo” del estudiante que se consulta y da como resultado un JSON con una lista de la información de los estudiantes encontrados, conteniendo los siguientes campos como respuesta: ”Nombres”, ”Apellidos”, ”Matricula”, ”Email”, ”Direccion”, ”Carrera”, y ”Modulo”.

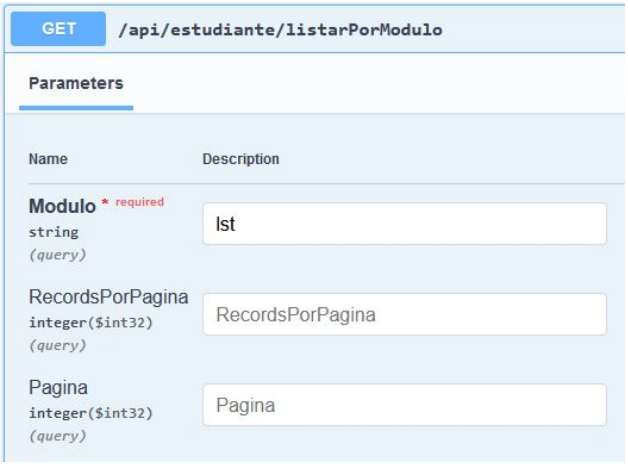

Figura A9: Servicio Estudiante: Consulta end device ”listarPorModulo”

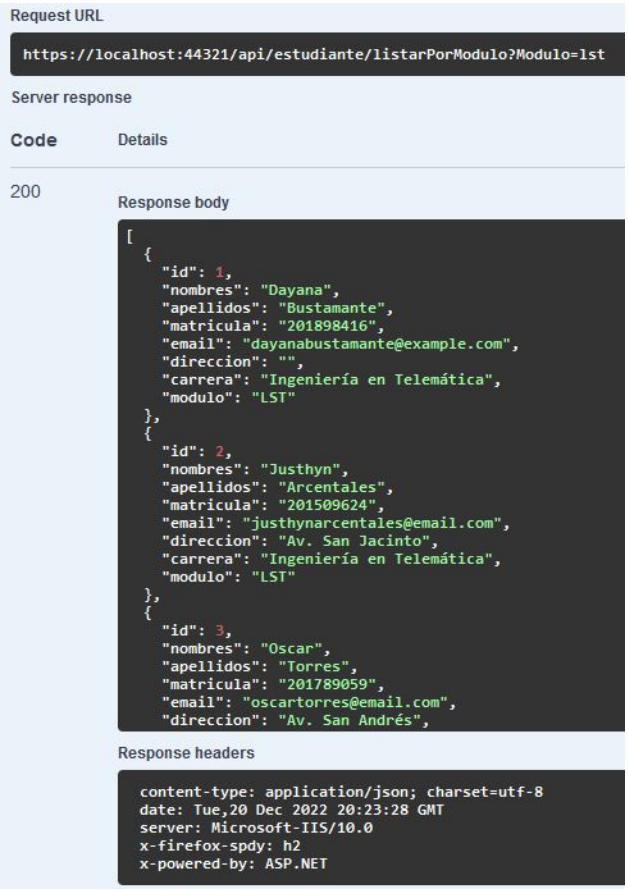

Figura A10: Servicio Estudiante: Respuesta end evice ”listarPorModulo”

- **Listar todos [GET]:** End point que no recibe campos. Da como resultado un JSON con una lista de la información de todos estudiantes registrados, conteniendo los siguientes campos como respuesta: ”Nombres”, ”Apellidos”, ”Matricula”, ”Email”, ”Direccion”, ”Carrera”, y ”Modulo”.

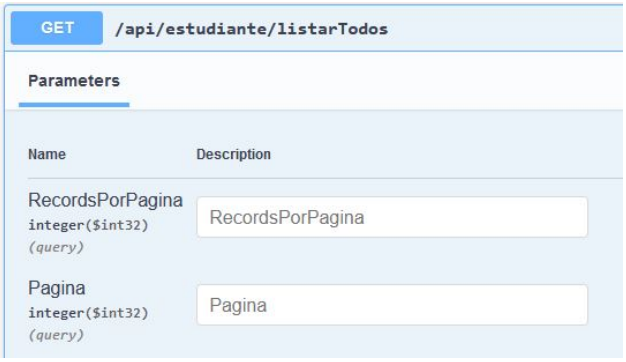

Figura A11: Servicio Estudiante: Consulta end device ”listarTodos”

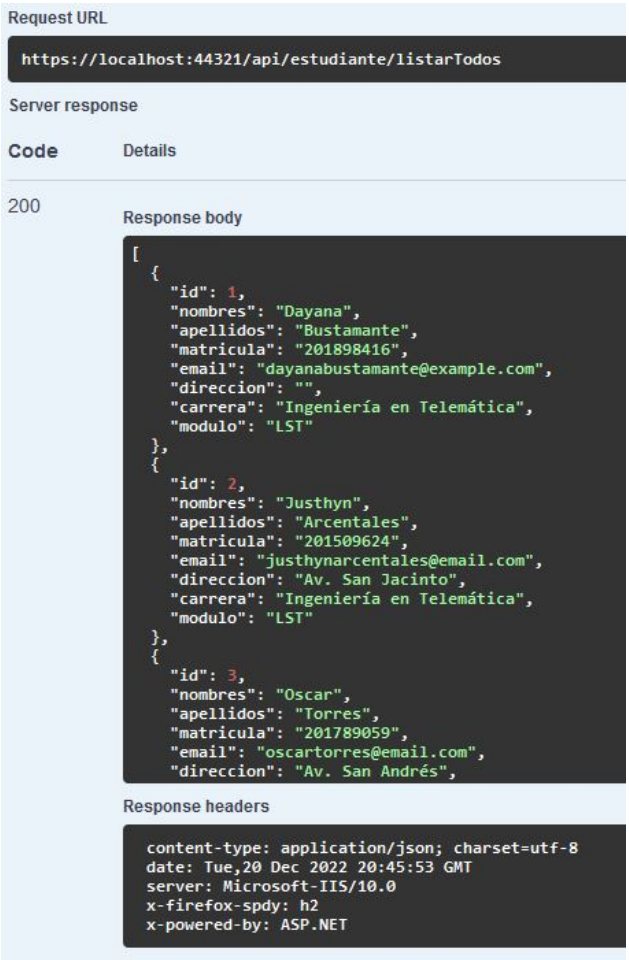

Figura A12: Servicio Estudiante: Respuesta end device ”listarTodos”

- **Registrar estudiante [POST]:** End point que envía como request un objeto con los campos: ”nombres”, ”apellidos”, ”matricula”, ”email”, ”direccion”, ”carrera”, y ”modulo”. Para luego registrar el estudiante en cuestión con la informacón enviada si pasa correctamente las validaciones pertinentes.

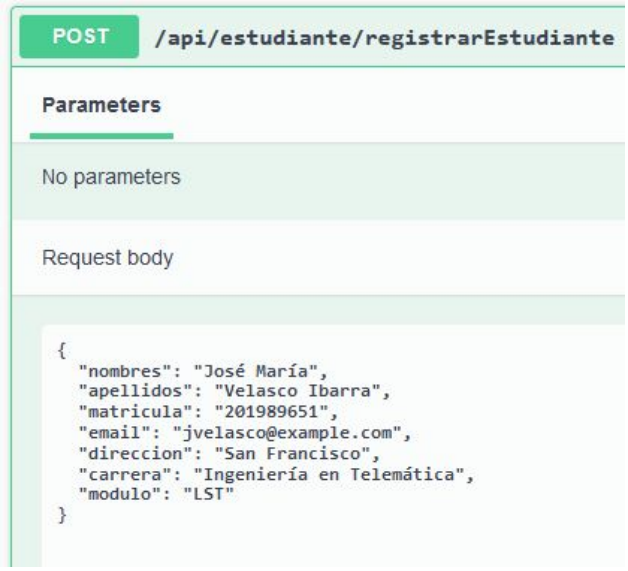

Figura A13: Servicio Estudiante: Solicitud end device ”registrarEstudiante”

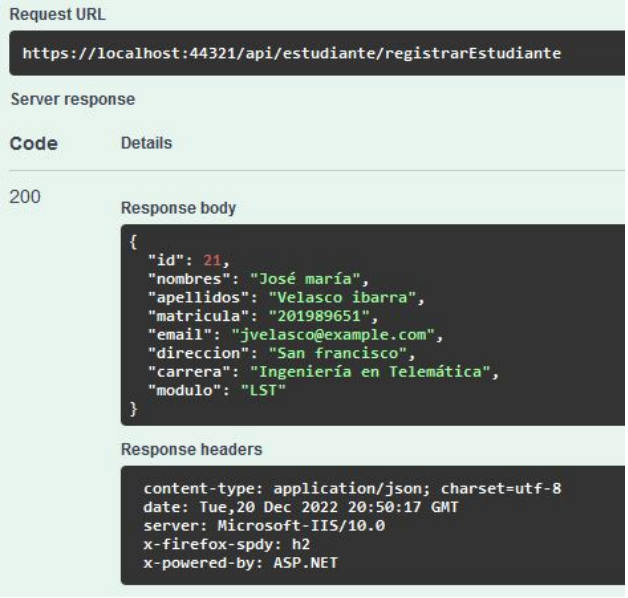

Figura A14: Servicio Estudiante: Respuesta end device ”registrarEstudiante”

- **Registrar grupo estudiante [POST]:** End point que envía como request un objeto con el campo ”modulo” y en el request un archivo .csv donde se encuentre la información de los estudiantes. Para luego registrar el/los estudiante en cuestión con la informacón enviada si pasa correctamente las validaciones pertinentes.

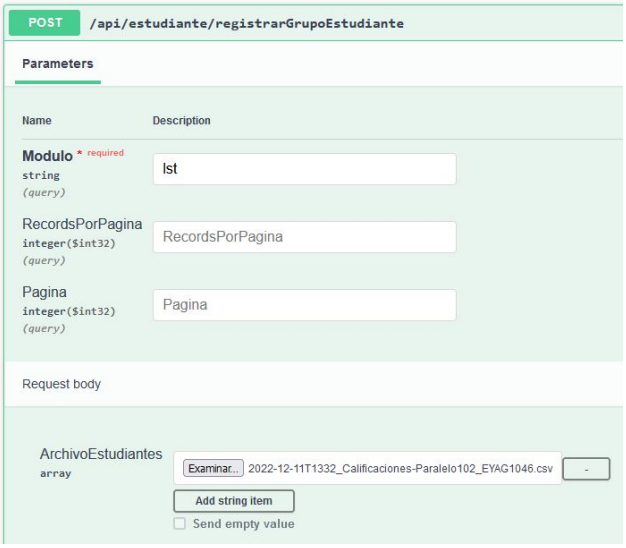

Figura A15: Servicio Estudiante: Solicitud end device ”registrarGrupoEstudiante”

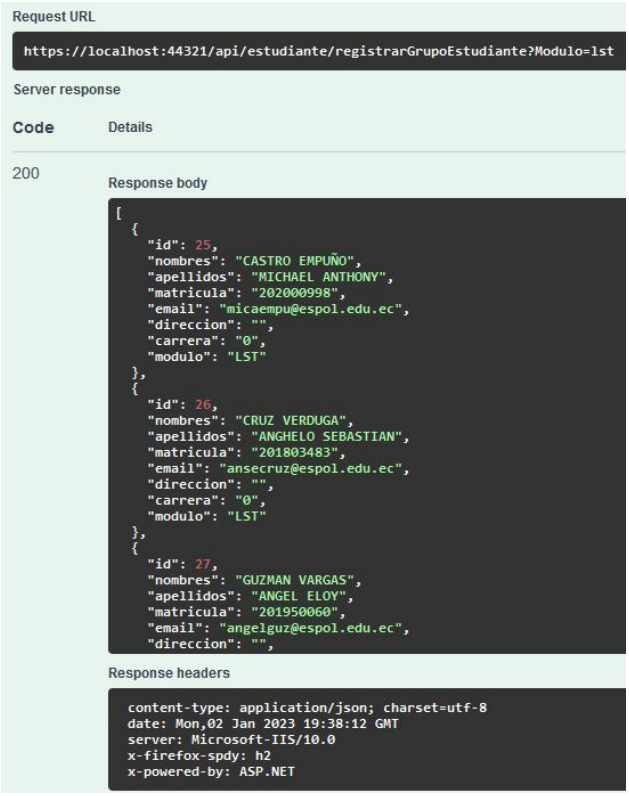

Figura A16: Servicio Estudiante: Respuesta end device ”registrarGrupoEstudiante”

- **Editar estudiante [PUT]:** End point que envía como request un objeto con los campos: ”id”, ”nombres”, ”apellidos”, ”matricula”, ”email”, ”direccion”, ”carrera”, y ”modulo”. Para luego editar el estudiante en cuestión con el ”id” enviado con los campos en los cuales se envió información.

Figura A17: Servicio Estudiante: Solicitud end device ”editarEstudiante”

Figura A18: Servicio Estudiante: Respuesta end evice ”editarEstudiante”

- **Eliminar estudiante [DELETE]:** End point que envía como request un objeto con el campo ”id”, para eliminar (de forma lógica) al estudiante con el id enviado, haciendo las validaciones pertinentes.

Figura A19: Servicio Estudiante: Solicitud end device ”eliminarEstudiante”

Figura A20: Servicio Estudiante: Respuesta end device ”eliminarEstudiante”

Servicio Asistencia: Este servicio contiene los End devices que corresponden a todas las peticiones relacionadas a los registros de asistencia. Comprendiendo las consultas por diferentes campos, y peticiones de creación, actualización y eliminación de los registros de asistencia que corresponden a cada estudiante. Para los endpoints de consulta se tendrán dos campos adicionales: ”RecordsPorPagina” y ”Pagina”, que sirven para la paginación de la lista de resultados.

- **Consultar por matrícula [GET]:** End point que recibe dos campos: ”Matricula” y ”Modulo” del estudiante que se consulta y da como resultado un JSON con la información del registro de asistencia encontrado del estudiante, conteniendo los siguientes campos como respuesta: ”Nombres”, ”Apellidos”, ”Matricula”, ”Email”, ”Fecha”, ”EstadoAsistencia”, ”Carrera”, y ”Modulo”.

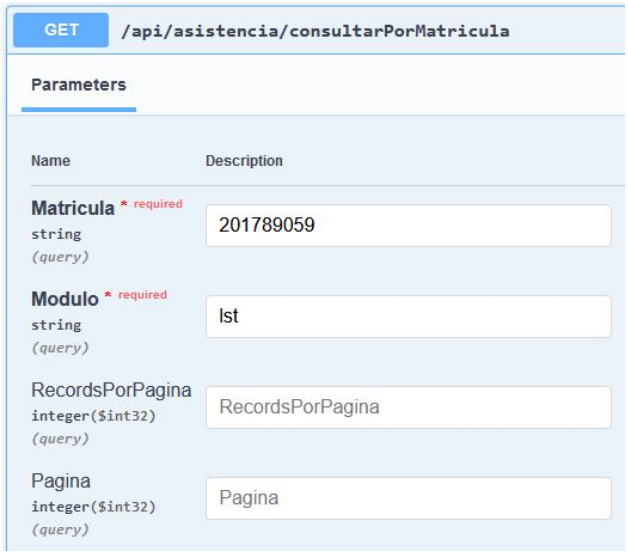

Figura A21: Servicio Asistencia: Consulta end device ”consultarPorMatricula”

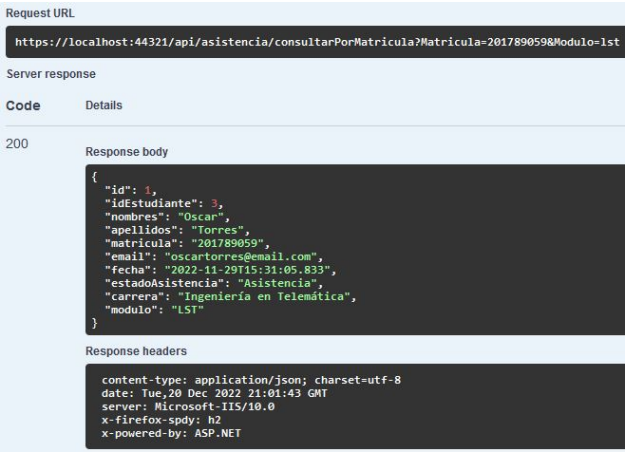

Figura A22: Servicio Asistencia: Respuesta end device ”consultarPorMatricula”

- **Listar por nombres [GET]:** End point que recibe tres campos: ”Nombres”, ”Apellidos”, y ”Modulo” del estudiante que se consulta y da como resultado un JSON con una lista de la información del registro de asistencia encontrado del estudiante, conteniendo los siguientes campos como respuesta: ”Nombres”, ”Apellidos”, ”Matricula”, ”Email”, ”Fecha”, ”EstadoAsistencia”, ”Carrera”, y ”Modulo”.

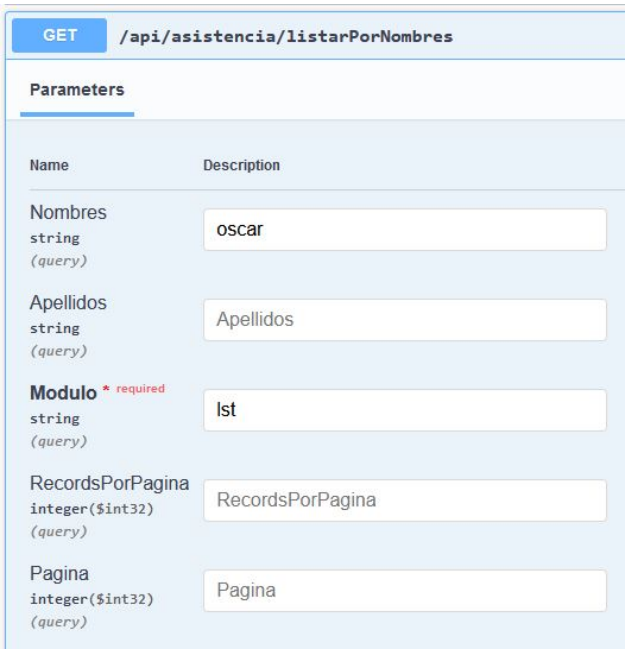

Figura A23: Servicio Asistencia: Consulta end device ”listarPorNombres”

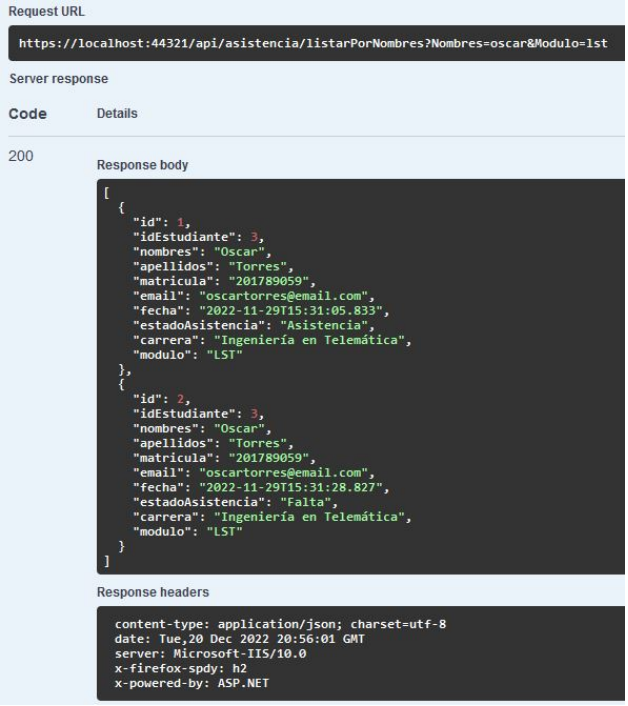

Figura A24: Servicio Asistencia: Respuesta end device ”listarPorNombres”

- **Listar por carrera [GET]:** End point que recibe dos campos: ”Carrera” y ”Modulo” del estudiante que se consulta y da como resultado un JSON con una lista de la información del registro de asistencia encontrado del estudiante, conteniendo los siguientes campos como respuesta: ”Nombres”, ”Apellidos”, ”Matricula”, ”Email”, ”Fecha”, ”EstadoAsistencia”, ”Carrera”, y ”Modulo”.

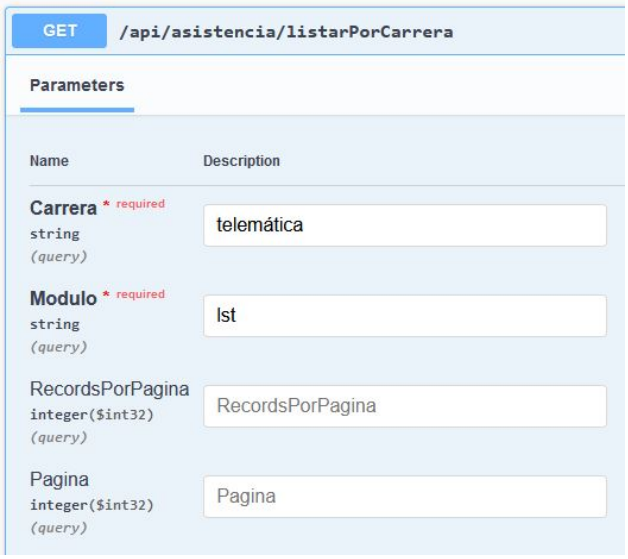

Figura A25: Servicio Asistencia: Consulta end device ”listarPorCarrera”

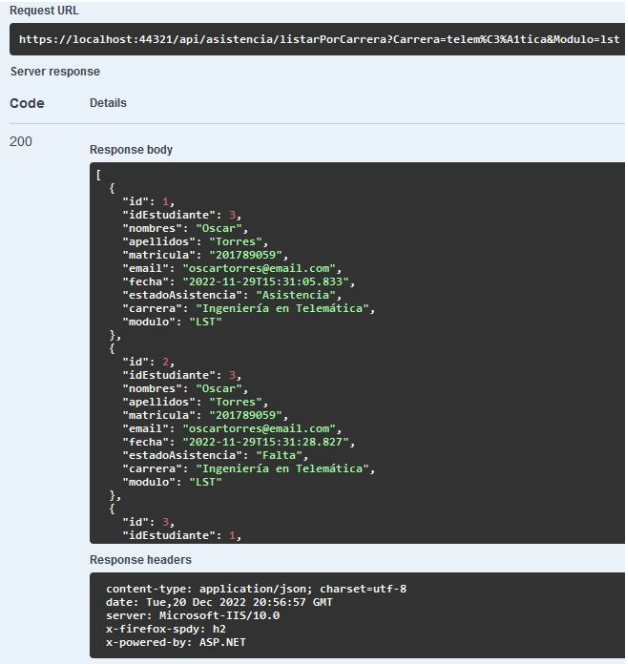

Figura A26: Servicio Asistencia: Respuesta end device ”listarPorCarrera”

- **Listar por módulo [GET]:** End point que recibe el campo ”Modulo” del estudiante que se consulta y da como resultado un JSON con una lista de la información del registro de asistencia encontrado del estudiante, conteniendo los siguientes campos como respuesta: ”Nombres”, ”Apellidos”, ”Matricula”, ”Email”, ”Fecha”, ”EstadoAsistencia”, ”Carrera”, y ”Modulo”.

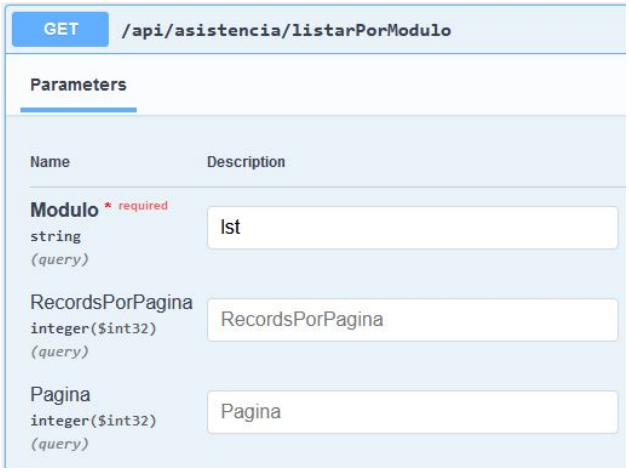

Figura A27: Servicio Asistencia: Consulta end device ”listarPorModulo”

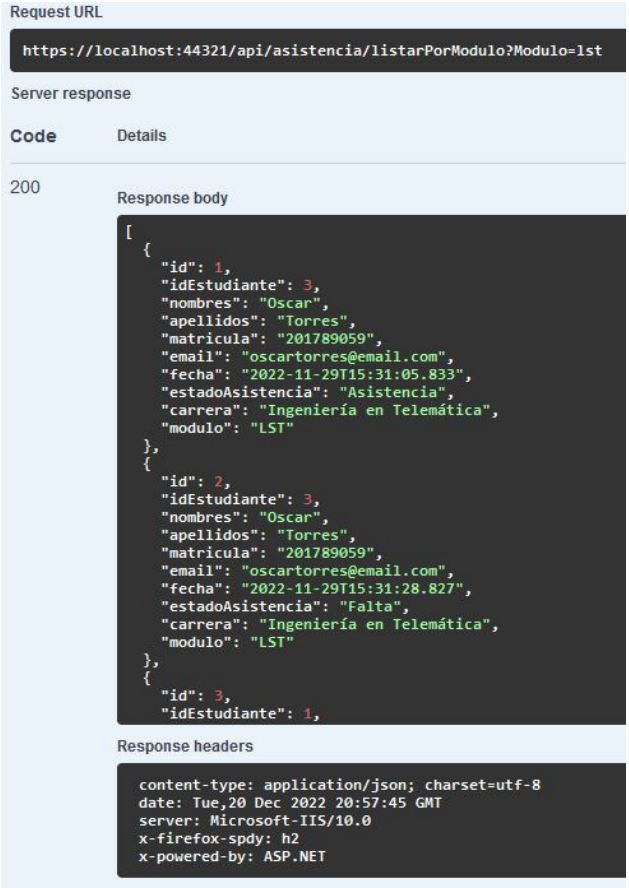

Figura A28: Servicio Asistencia: Respuesta end device ”listarPorModulo”

- **Listar todos [GET]:** End point que no recibe campos. Da como resultado un JSON con una lista de la información del registro de asistencia encontrado del estudiante, conteniendo los siguientes campos como respuesta: ”Nombres”, ”Apellidos”, ”Matricula”, ”Email”, ”Fecha”, ”EstadoAsistencia”, ”Carrera”, y ”Modulo”.

Figura A29: Servicio Asistencia: Consulta end device ”listarTodos”

Figura A30: Servicio Asistencia: Respuesta end device ”listarTodos”

- **Registrar registro de asistencia [POST]:** End point que envía como request un objeto con los campos: ”matricula”, ”fecha”, ”esAsistencia”, y ”modulo”. Para luego crear el registro de asistencia en cuestión con la informacón enviada si pasa correctamente las validaciones pertinentes.

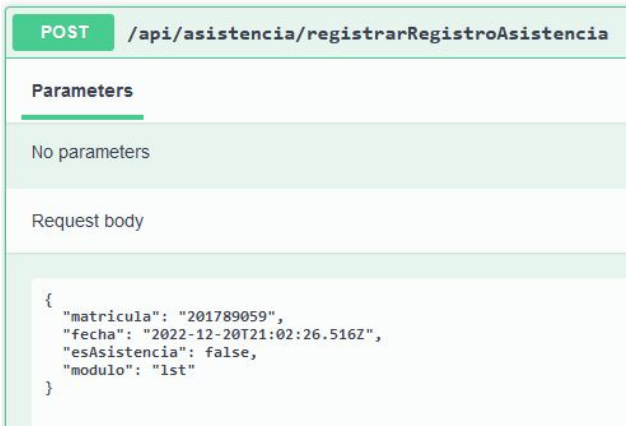

Figura A31: Servicio Asistencia: Solicitud end device ”registrarRegistroAsistencia”

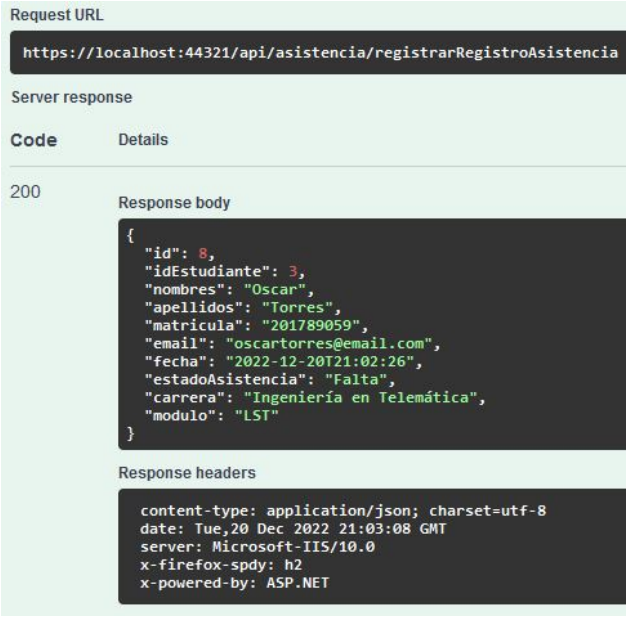

Figura A32: Servicio Asistencia: Respuesta end device ”registrarRegistroAsistencia”

- **Editar registro de asistencia [PUT]:** End point que envía como request un objeto con los campos: ”id” y ”esAsistencia”. Para luego editar el registro de asistencia en cuestión con el ”id” enviado, para cambiar entre asistencia y falta, según corresponda.

Figura A33: Servicio Asistencia: Solicitud end device ”editarRegistroAsistencia”

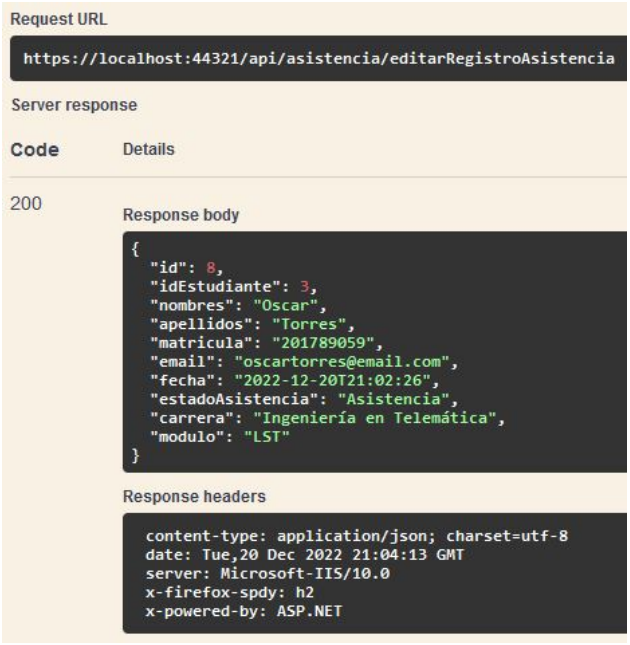

Figura A34: Servicio Asistencia: Respuesta end device ”editarRegistroAsistencia”

- **Eliminar registro de asistencia [DELETE]:** End point que envía como request un objeto con el campo ”id”, para eliminar (de forma lógica) al registro de asistencia con el id enviado, haciendo las validaciones pertinentes.

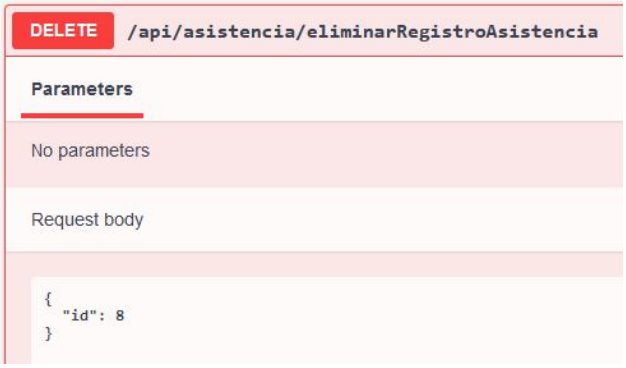

Figura A35: Servicio Asistencia: Solicitud end device ”eliminarRegistroAsistencia”

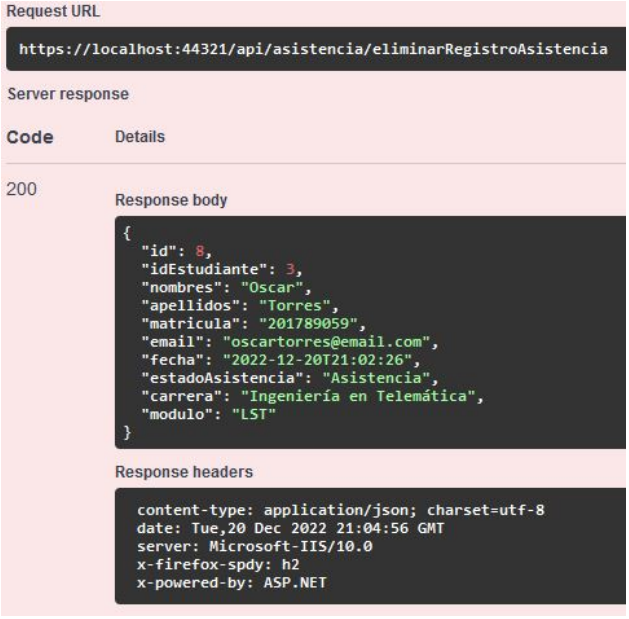

Figura A36: Servicio Asistencia: Respuesta end device ”eliminarRegistroAsistencia”

- **Establecer Asistencia [PUT]:** End point que envía como request un objeto con el campo ”matricula”. Que sirve para que el estudiante registre su asistencia, editando el estado de asistencia de ””Falta” a ”Asistencia” del estudiante con la matrícula ingresada que exista en el módulo actual elegido por el usuario.

Figura A37: Servicio Asistencia: Solicitud end device ”establecerAsistencia”

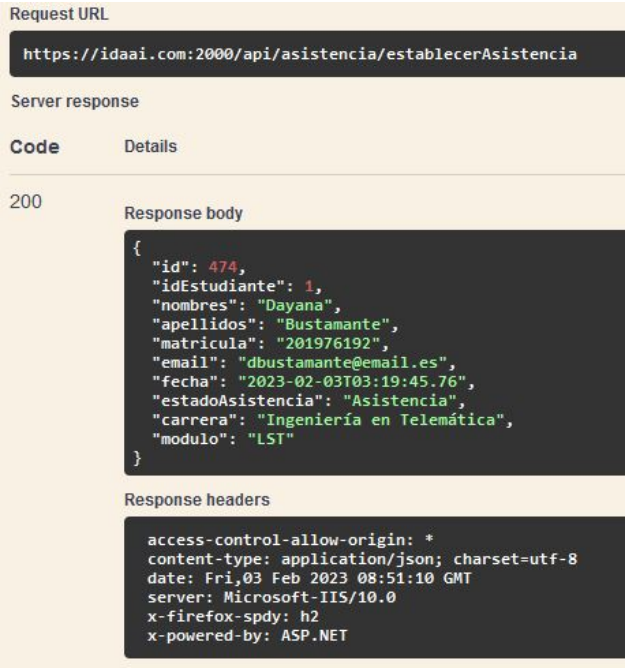

Figura A38: Servicio Asistencia: Respuesta end device ”establecerAsistencia”

Servicio Carrera: Este servicio contiene los End devices que corresponden a todas las peticiones relacionadas a las carreras. Comprendiendo las consultas por diferentes campos, y peticiones de creación, actualización y eliminación de los registros en la tabla Carreras. Para los endpoints de consulta se tendrán dos campos adicionales: ”RecordsPorPagina” y ”Pagina”, que sirven para la paginación de la lista de resultados.

- **Listar por nombre [GET]:** End point que recibe dos campos: ”Nombre” y ”Modulo” de la carrera que se consulta, y da como resultado un JSON con una lista de la información de las carreras encontradas, conteniendo los siguientes campos como respuesta: ”id”, ”nombre”, y ”modulo”.

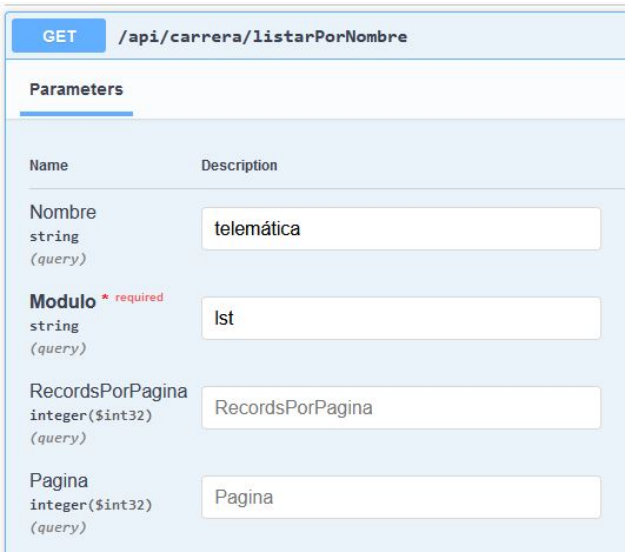

Figura A39: Servicio Carrera: Consulta end device ”listarPorNombre”

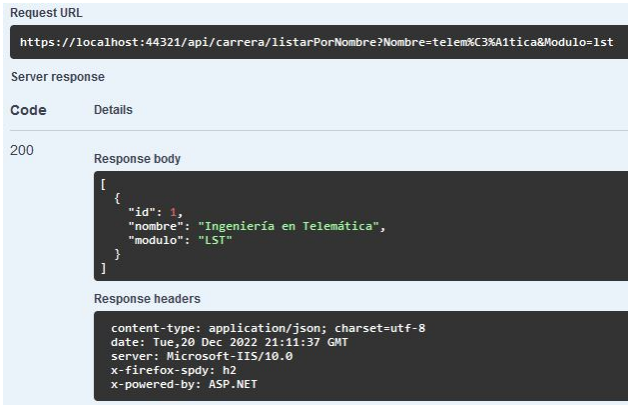

Figura A40: Servicio Carrera: Respuesta end device ”listarPorNombre”

- **Listar por módulo [GET]:** End point que recibe el campo ”Modulo” de la carrera que se consulta, y da como resultado un JSON con una lista de la información de las carreras encontradas, conteniendo los siguientes campos como respuesta: ”id”, ”nombre”, y ”modulo”.

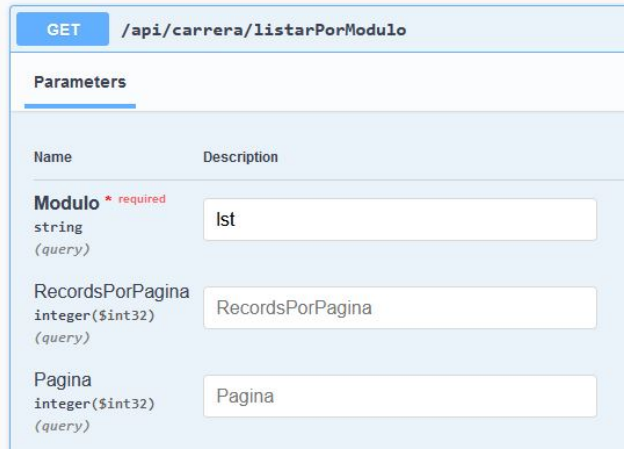

Figura A41: Servicio Carrera: Consulta end device ”listarPorModulo”

Figura A42: Servicio Carrera: Respuesta end device ”listarPorModulo”

- **Listar todos [GET]:** End point que no recibe campos. Da como resultado un JSON con una lista de la información de todas las carreras registradas, conteniendo los siguientes campos como respuesta: ”id”, ”nombre”, y ”modulo”.

Figura A43: Servicio Carrera: Consulta end device ”listarTodos”

Figura A44: Servicio Carrera: Respuesta end device ”listarTodos”

- **Registrar carrera [POST]:** End point que envía como request un objeto con los campos: ”nombre” y ”modulo”. Para luego, si pasa correctamente las validaciones pertinentes, registrar la carrera en cuestión con la informacón enviada.

Figura A45: Servicio Carrera: Solicitud end device ”registrarCarrera”

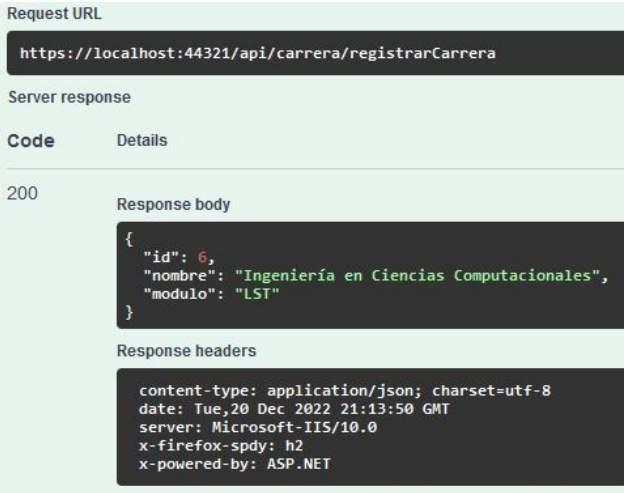

Figura A46: Servicio Carrera: Respuesta end device ”registrarCarrera”

- **Editar carrera [PUT]:** End point que envía como request un objeto con los campos: ”id”, ”nombre”, y ”modulo”. Para luego editar la carrera en cuestión con el ”id” enviado.

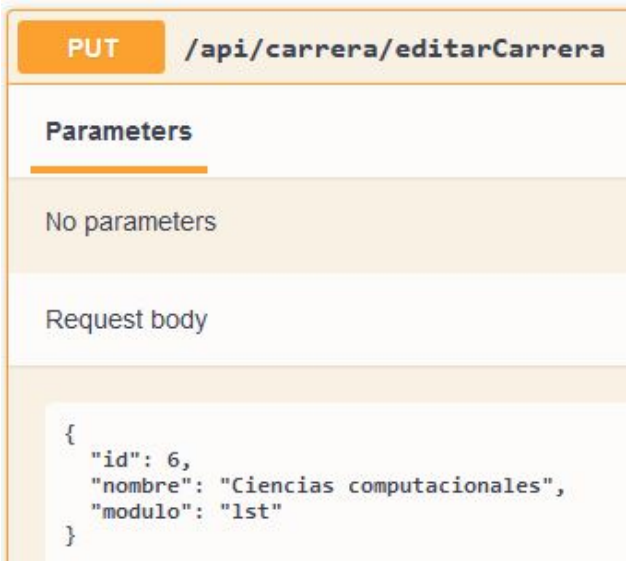

Figura A47: Servicio Carrera: Solicitud end device ”editarCarrera”

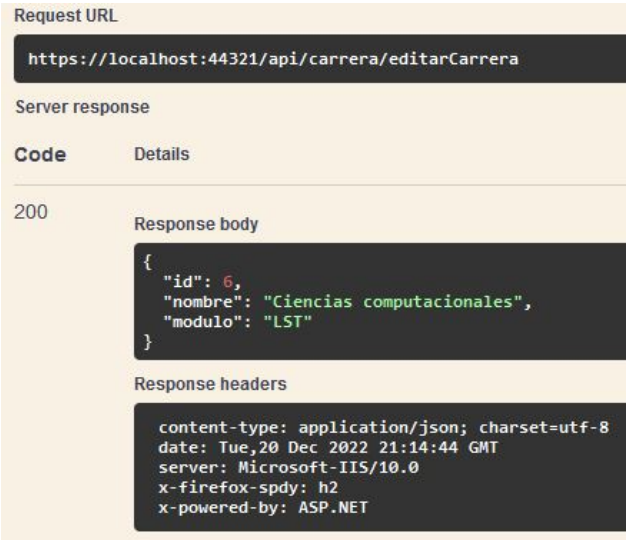

Figura A48: Servicio Carrera: Respuesta end device ”editarCarrera”

- **Eliminar carrera [DELETE]:** End point que envía como request un objeto con el campo ”id”, para eliminar (de forma lógica) la carrera con el id enviado, haciendo las validaciones pertinentes.

Figura A49: Servicio Carrera: Solicitud end device ”eliminarCarrera”

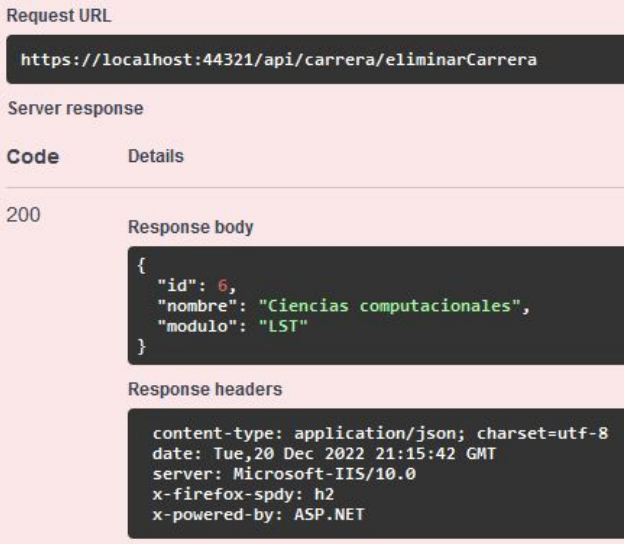

Figura A50: Servicio Carrera: Respuesta end device ”eliminarCarrera”

Servicio Módulo: Este servicio contiene los End devices que corresponden a todas las peticiones relacionadas a los Módulos. Comprendiendo las consultas por diferentes campos, y peticiones de creación, actualización y eliminación de los registros en la tabla Módulos. Para los endpoints de consulta se tendrán dos campos adicionales: ”RecordsPorPagina” y ”Pagina”, que sirven para la paginación de la lista de resultados.

- **Listar por nombre [GET]:** End point que recibe el campo ”Nombre” del módulo que se consulta, y da como resultado un JSON con una lista de la información de los módulos encontrados, conteniendo los siguientes campos como respuesta: ”id”, ”nombre”, ”descripcion”, y ”periodoAcademico”.

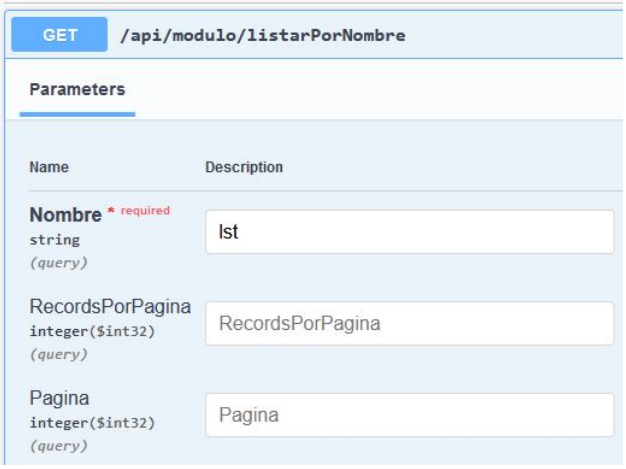

Figura A51: Servicio Módulo: Consulta end device ”listarPorNombre”

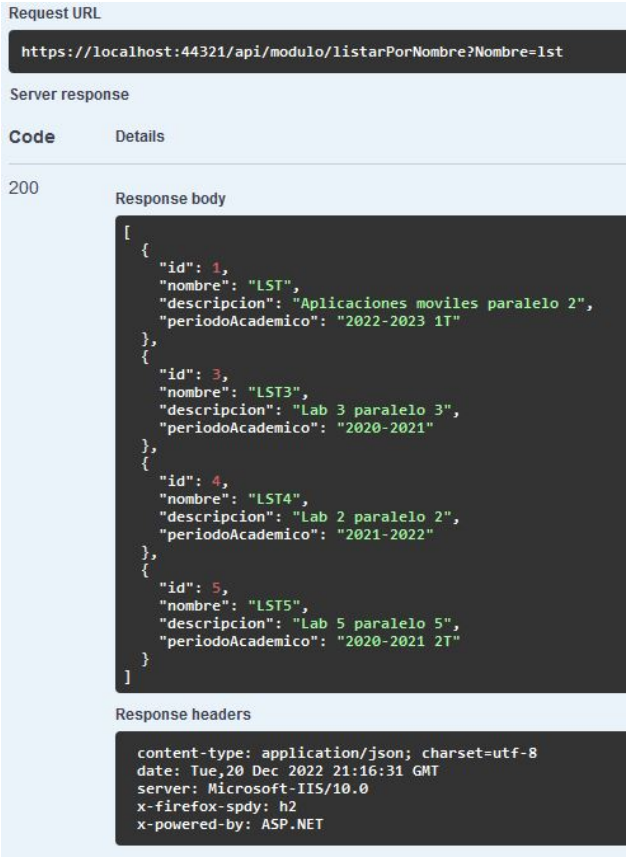

Figura A52: Servicio Módulo: Respuesta end device ”listarPorNombre”

- **Listar por período académico [GET]:** End point que recibe el campo ”PeriodoAcademico” del módulo que se consulta, y da como resultado un JSON con una lista de la información de los módulos encontrados, conteniendo los siguientes campos como respuesta: ”id”, ”nombre”, ”descripcion”, y ”periodoAcademico”.

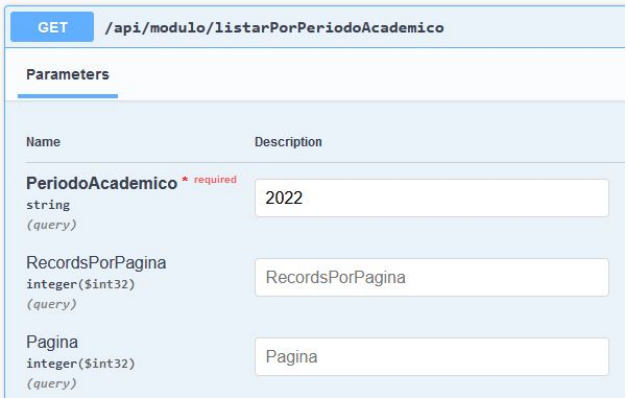

Figura A53: Servicio Módulo: Consulta end device ”listarPorPeriodoAcademico”

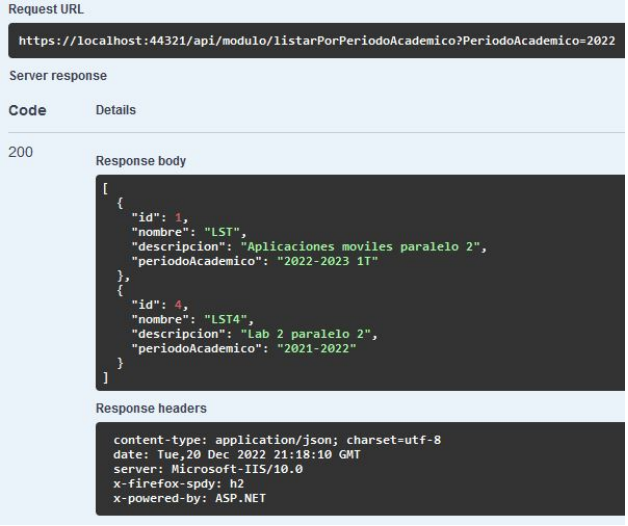

Figura A54: Servicio Módulo: Respuesta end device ”listarPorPeriodoAcademico”

- **Listar todos [GET]:** End point que no recibe campos. Da como resultado un JSON con una lista de la información de todos los módulos registrados, conteniendo los siguientes campos como respuesta: ”id”, ”nombre”, ”descripcion”, y ”periodoAcademico”.

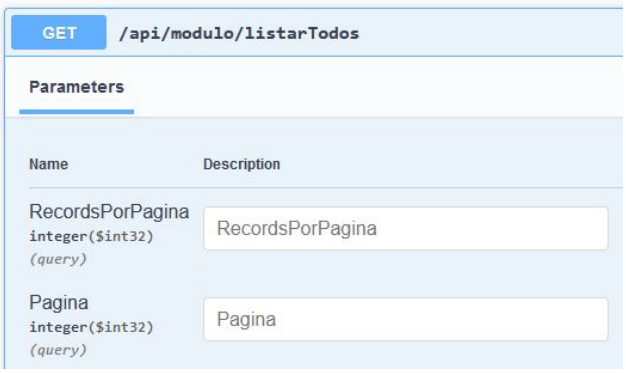

Figura A55: Servicio Módulo: Consulta end device ”listarTodos”

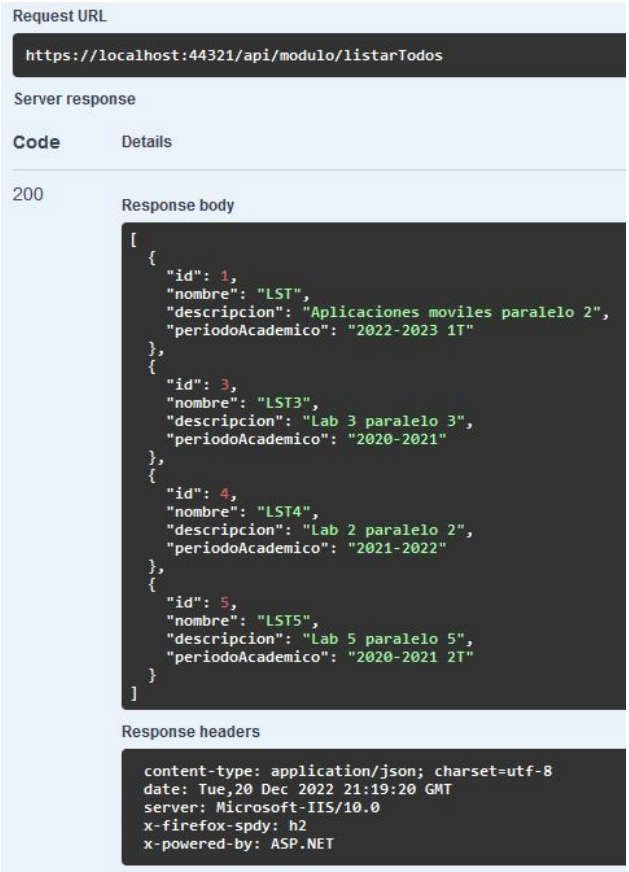

Figura A56: Servicio Módulo: Respuesta end device ”listarTodos”

- **Registrar Módulo [POST]:** End point que envía como request un objeto con los campos: ”nombre”, ”descripcion”, y ”periodoAcademico”. Para luego, si pasa correctamente las validaciones pertinentes, registrar el módulo en cuestión con la informacón enviada.

Figura A57: Servicio Módulo: Solicitud end device ”registrarModulo”

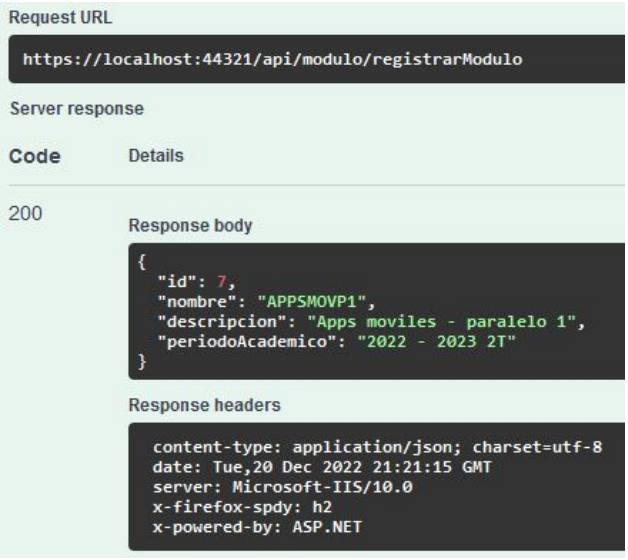

Figura A58: Servicio Módulo: Respuesta end device ”registrarModulo”

- **Editar Módulo [PUT]:** End point que envía como request un objeto con los campos: ”id”, ”nombre”, ”descripcion”, y ”periodoAcademico”. Para luego editar el módulo en cuestión con el ”id” enviado.

Figura A59: Servicio Módulo: Solicitud end device ”editarModulo”

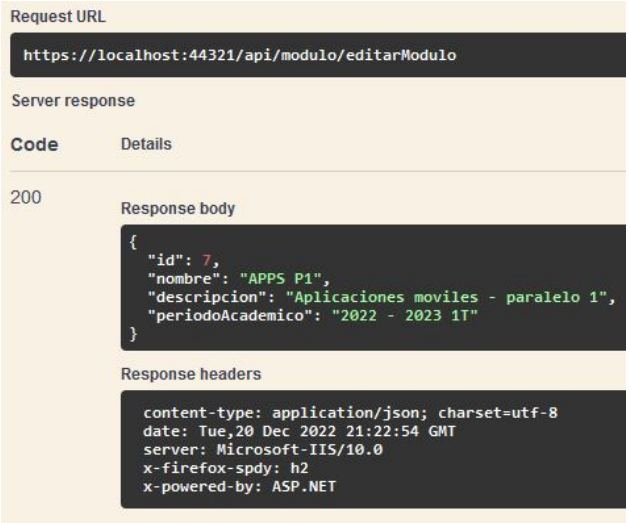

Figura A60: Servicio Módulo: Respuesta end device ”editarModulo”

- **Eliminar Módulo [DELETE]:** End point que envía como request un objeto con el campo ”id”, para eliminar (de forma lógica) el módulo con el id enviado, haciendo las validaciones pertinentes.

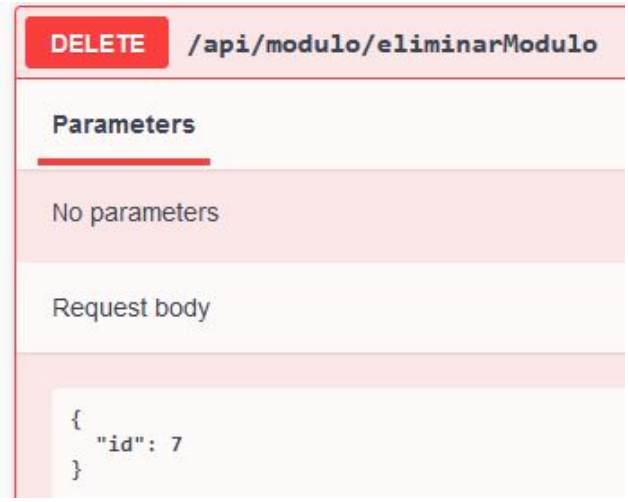

Figura A61: Servicio Módulo: Solicitud end device ”eliminarModulo”

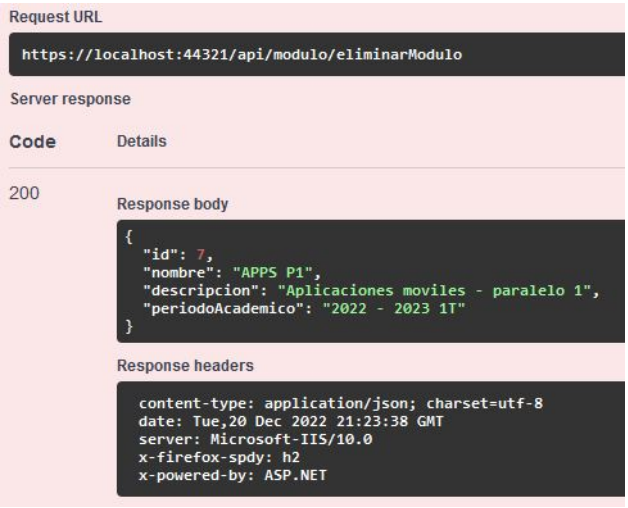

Figura A62: Servicio Módulo: Respuesta end device ”eliminarModulo”

Servicio Usuario: Este servicio contiene los End devices que corresponden a todas las peticiones relacionadas a los Usuarios. Comprendiendo las funcionalidades de inicio de sesión y registro, renovación de token (para mantener al usuario con la sesión iniciada), además de peticiones de actualización y eliminación de usuarios.

- **Login usuario [POST]:** End point que envía un objeto request con los campos: ”usuario” y ”password”, para autenticar al usuario. La respuesta es un objeto con los campos: ”token” y ”expiracion”, los cuales tendrán valores cuando el usuario sea correctamente autenticado, es decir, exista en la base de datos.

Figura A63: Servicio Usuario: Solicitud end device ”loginUsuario”

Figura A64: Servicio Usuario: Respuesta end device ”loginUsuario”

- **Registrar usuario [POST]:** End point que envía un objeto request con los campos: ”usuario”, ”password”, y ”email” para registrar al usuario correspondiente. La respuesta es un objeto con los campos: ”token” y ”expiracion”.

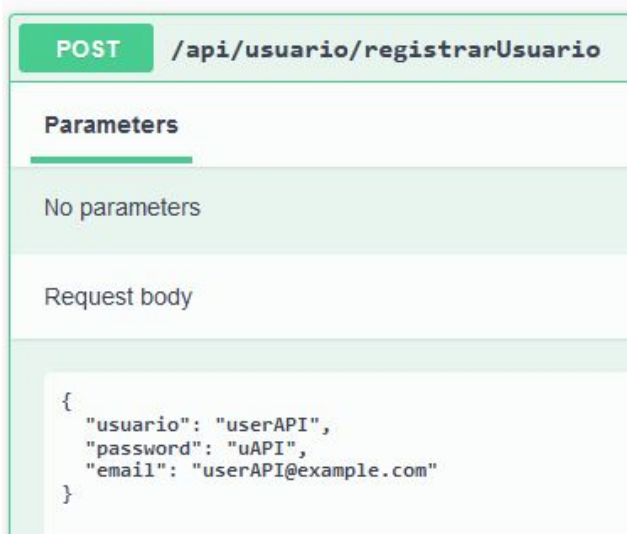

Figura A65: Servicio Usuario: Solicitud end device ”registrarUsuario”

Figura A66: Servicio Usuario: Respuesta end device ”registrarUsuario”

- **Editar usuario [PUT]:** End point que envía un objeto request con los campos: ”id”, ”usuario”, ”password”, y ”email” para editar los campos enviados del usuario que corresponde a la id enviada. La respuesta es un objeto con los campos: ”id”, ”usuario”, y ”email”.

Figura A67: Servicio Usuario: Solicitud end device ”editarUsuario”

Figura A68: Servicio Usuario: Respuesta end device ”editarUsuario”

- **Eliminar usuario [DELETE]:** End point que envía como request un objeto con el campo ”id”, para eliminar (de forma lógica) el usuario con el id enviado, haciendo las validaciones pertinentes.

Figura A69: Servicio Usuario: Solicitud end device ”eliminarUsuario”

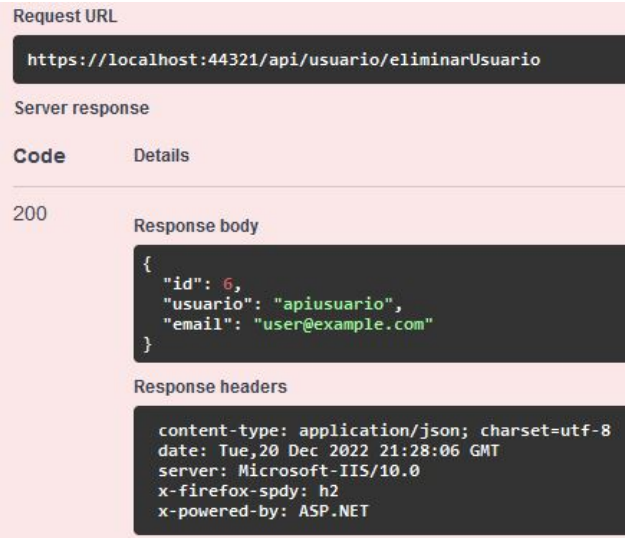

Figura A70: Servicio Usuario: Respuesta end device ”eliminarUsuario”

- **Renovar token [GET]:** End point que recibe de entrada un parámetro Usuario. Su funcionamiento radica en extender la duración de tiempo en 4 horas del token autenticado en la sesión actual. La respuesta es un objeto con los campos: ”token” y ”expiracion”.

Figura A71: Servicio Usuario: Respuesta end device ”renovarToken”

- **Obtener Usuario [GET]:** End point que recibe el campo ”Usuario” que corresponde

al nombre de usuario al que se consulta los datos, y da como resultado un JSON con una lista de la información de la información del usuario ingresado, conteniendo los siguientes campos como respuesta: ”id”, ”usuario”, ”email”, y ”moduloActual”.

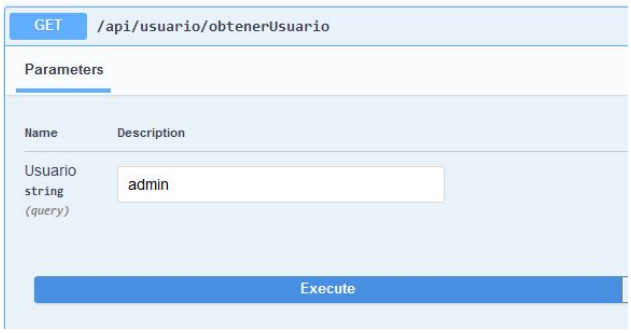

Figura A72: Servicio Usuario: Consulta end device ”obtenerUsuario”

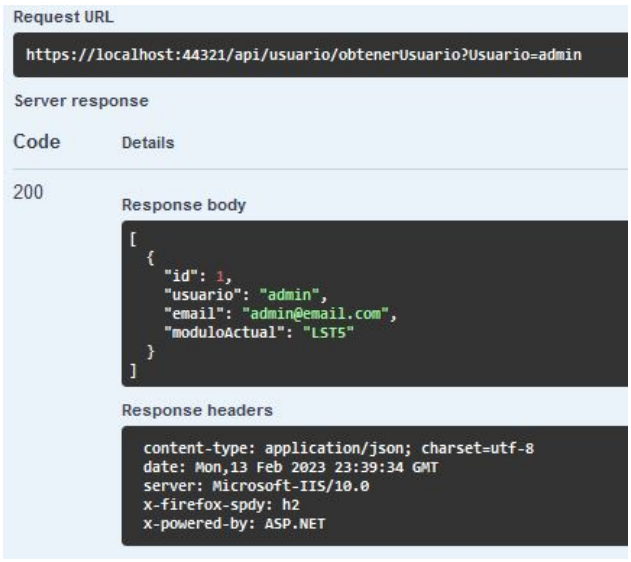

Figura A73: Servicio Usuario: Respuesta end device ”obtenerUsuario”

- **Actualizar Fecha [PUT]:** End point que envía un objeto request con el campo ”usuario”, para actualizar la fecha de último inicio de sesión del usuario ingresado a fecha actualizada de cuando se ejecuta dicho endpoint.

Figura A74: Servicio Usuario: Solicitud end device ”actualizarFecha”

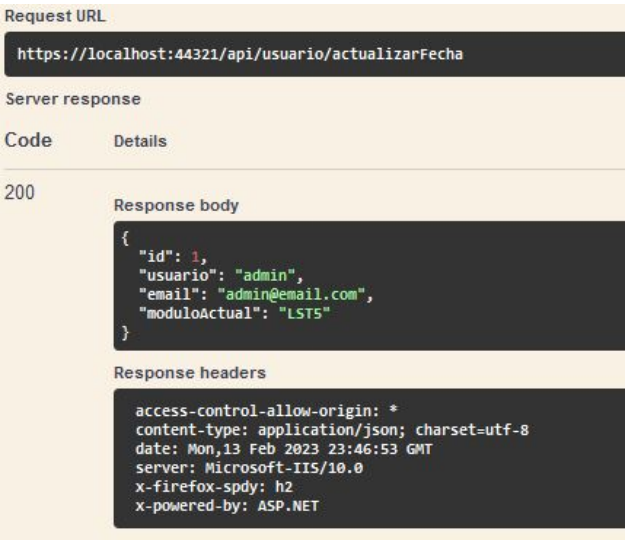

Figura A75: Servicio Usuario: Respuesta end device ”actualizarFecha”

- **Actualizar Módulo Actual [PUT]:** End point que envía un objeto request con los campos: ”usuario” y ”modulo”, para actualizar el módulo actual seleccionado por el usuario correspondiente, y así establecer el módulo al que solo los estudiantes pertenecientes al mismo podrán enviar su asistencia a través del endpoint ”establecerAsistencia” del servicio de Asistencia.

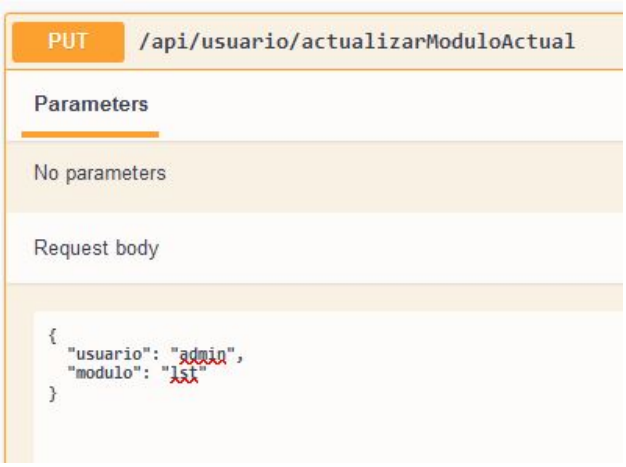

Figura A76: Servicio Usuario: Solicitud end device ”actualizarModuloActual”

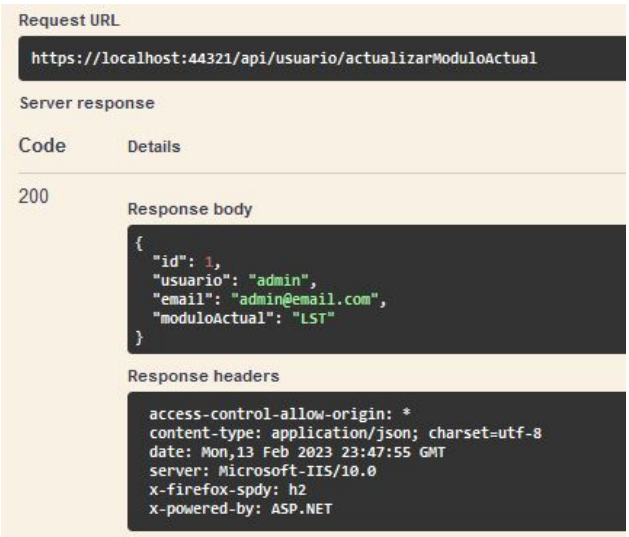

Figura A77: Servicio Usuario: Respuesta end device ”actualizarModuloActual”

Servicio Inventario: Este servicio contiene los End devices que corresponden a todas las peticiones relacionadas al inventario de objetos. Comprendiendo las consultas por diferentes campos, y peticiones de creación, actualización y eliminación de los registros en la tabla Inventario. Para los endpoints de consulta se tendrán dos campos adicionales: ”RecordsPorPagina” y ”Pagina”, que sirven para la paginación de la lista de resultados.

- **Listar por nombre [GET]:** End point que recibe el campo ”Nombre” del registro de inventario que se consulta, y da como resultado un JSON con una lista de la información de los registros encontrados, conteniendo los siguientes campos como respuesta: ”id”, ”nombre”, ”descripcion”, ”cantidadDisponible”, y ”cantidadTotal”.

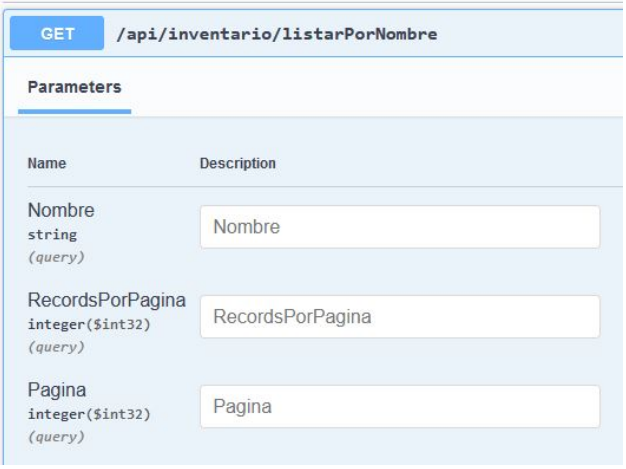

Figura A78: Servicio Inventario: Consulta end device ”listarPorNombre”

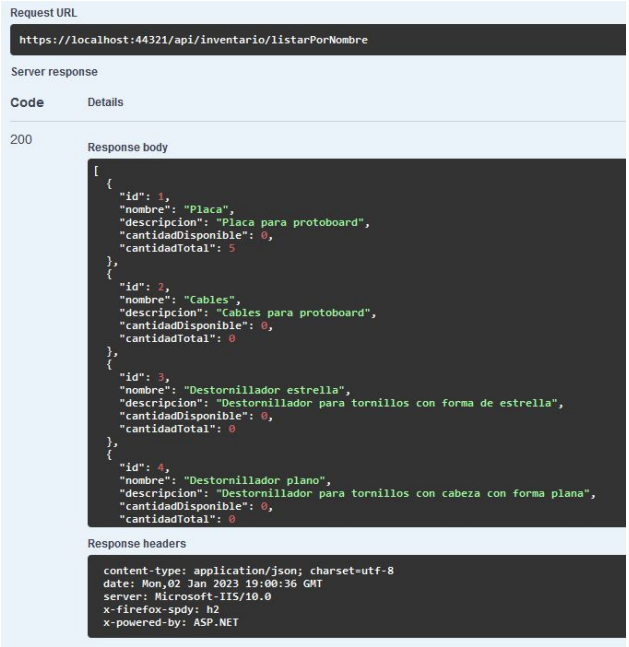

Figura A79: Servicio Inventario: Respuesta end device ”listarPorNombre”

- **Registrar Inventario [POST]:** End point que envía como request un objeto con los campos: ”nombre”, ”descripcion”, y ”cantidadTotal”. Para luego, si pasa correctamente las validaciones pertinentes, crear el registro en cuestión en la tabla Inventario con la informacón enviada.

Figura A80: Servicio Inventario: Solicitud end device ”registrarInventario”

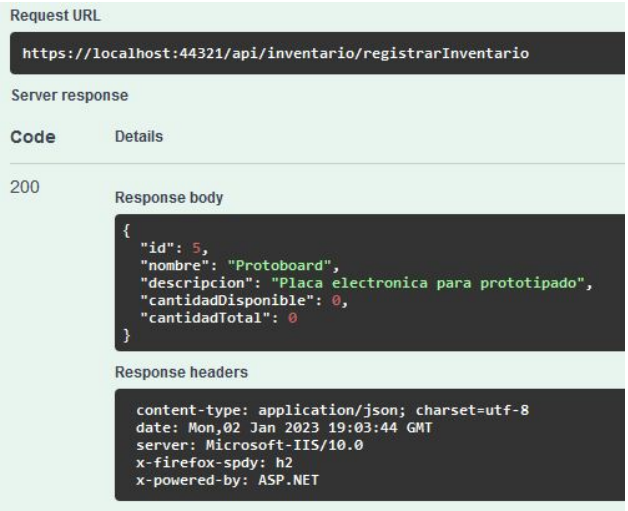

Figura A81: Servicio Inventario: Respuesta end device ”registrarInventario”

- **Editar Inventario [PUT]:** End point que envía como request un objeto con los campos: ”id”y ”nombre””. Para luego editar el registro en cuestión con el ”id” enviado.

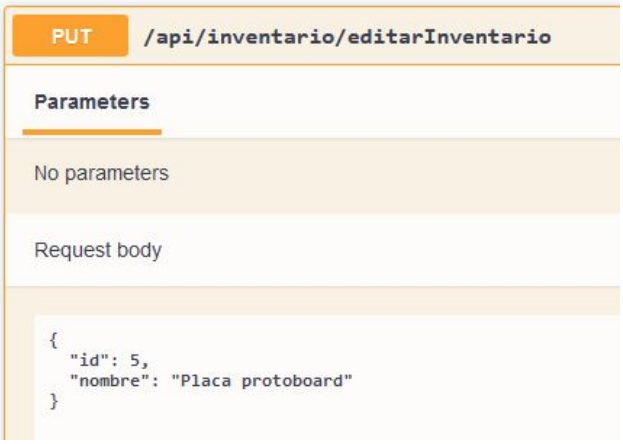

Figura A82: Servicio Inventario: Solicitud end device ”editarInventario”

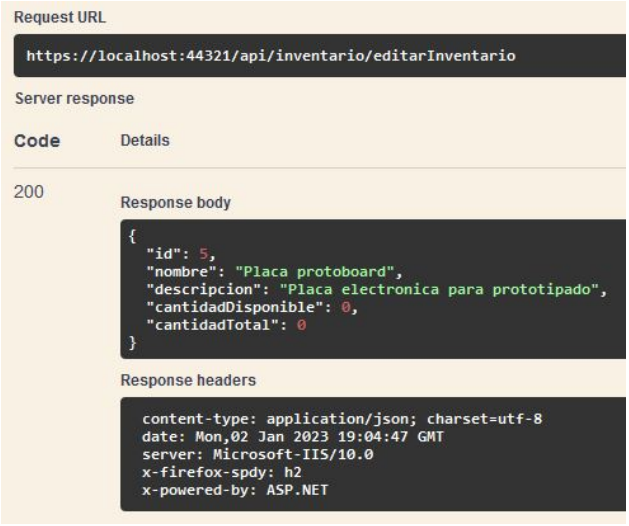

Figura A83: Servicio Inventario: Respuesta end device ”editarInventario”

- **Eliminar Inventario [DELETE]:** End point que envía como request un objeto con el campo ”id”, para eliminar (de forma lógica) el registro de inventario con el id enviado, haciendo las validaciones pertinentes.

Figura A84: Servicio Inventario: Solicitud end device ”eliminarInventario”

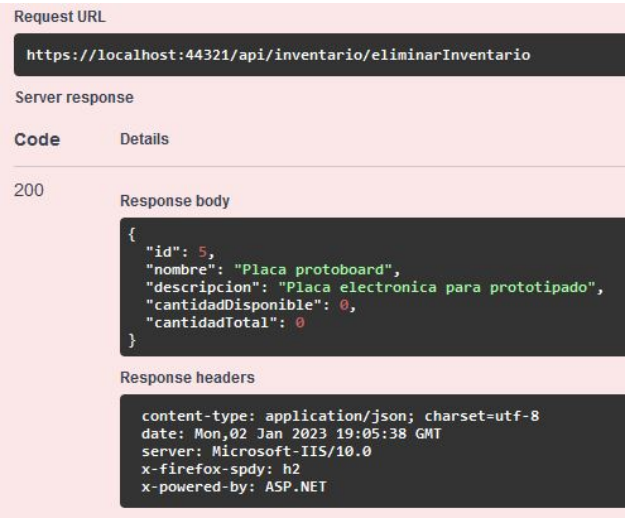

Figura A85: Servicio Inventario: Respuesta end device ”eliminarInventario”

- **Actualizar cantidades del inventario [POST]:** End point que no envía campos en su ejecución. Da como respuesta un mensaje indicando que las cantidades de objetos en el inventario se actualizaron correctamente, pues sirve para actualizar las cantidades disponibles de los objetos en el inventario de forma inteligente. Ésto lo realiza detectando la cantidad de Items que pertenecen a un registro de inventario que se encuentran en estado disponible.

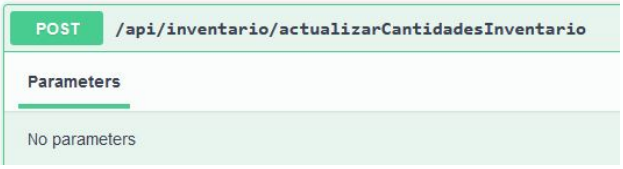

Figura A86: Servicio Inventario: Solicitud end device ”actualizarCantidadesInventario”

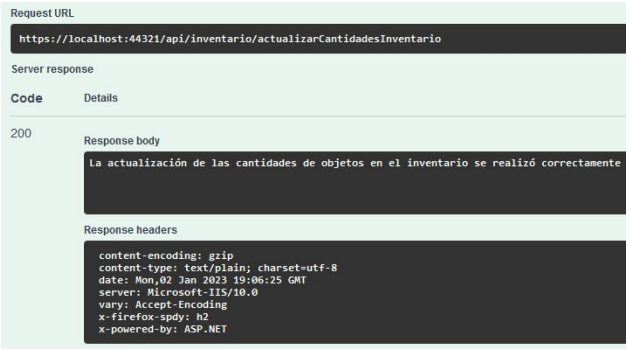

Figura A87: Servicio Inventario: Respuesta end device ”actualizarCantidadesInventario”

Servicio Item: Este servicio contiene los End devices que corresponden a todas las peticiones relacionadas a los Items. Comprendiendo las consultas por diferentes campos, y peticiones de creación, actualización y eliminación de los registros en la tabla Items. Para los endpoints de consulta se tendrán dos campos adicionales: ”RecordsPorPagina” y ”Pagina”, que sirven para la paginación de la lista de resultados.

- **Listar items [GET]:** End point que recibe los campos ”Rfid” y ”Inventario” del item que se consulta, y da como resultado un JSON con una lista de la información de los items encontrados, conteniendo los siguientes campos como respuesta: ”id”, ”rfid”, ”estadoItem”, e ”inventario”.

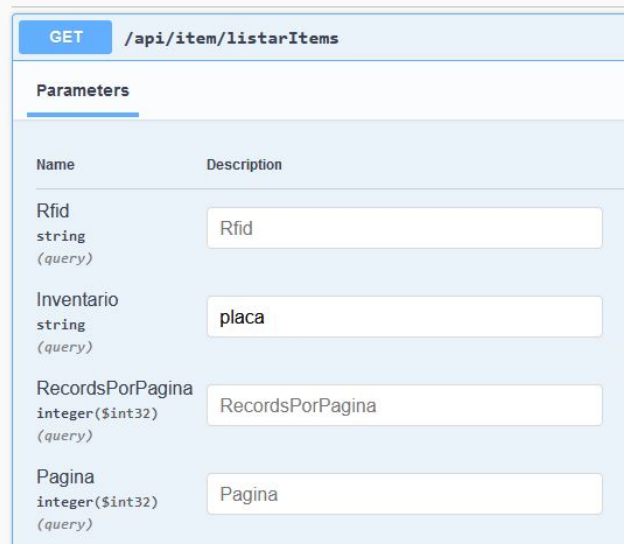

Figura A88: Servicio Item: Consulta end device ”listarItems”

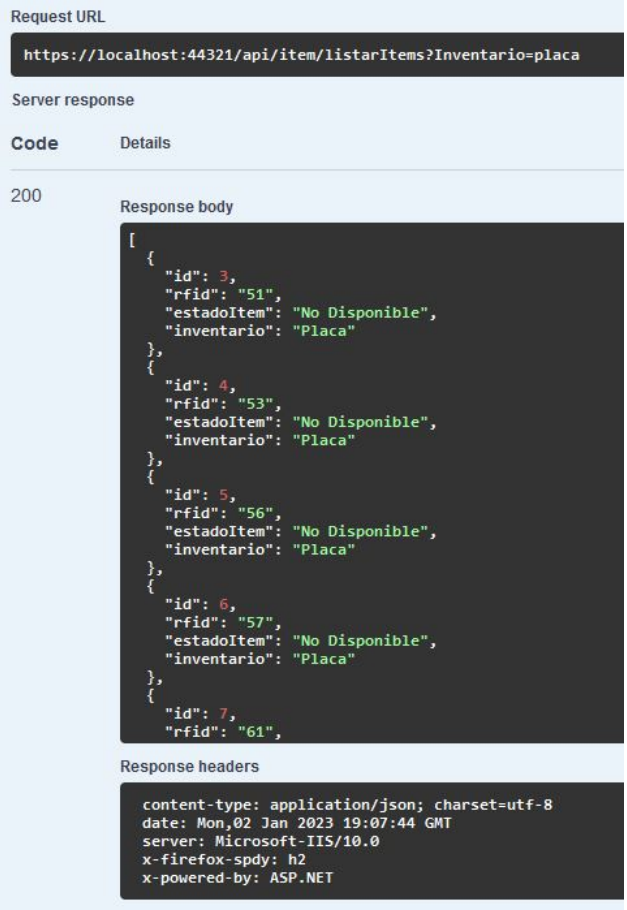

Figura A89: Servicio Item: Respuesta end device ”listarItems”

- **Registrar Item [POST]:** End point que envía como request un objeto con los campos ”rfid” e ”inventario”. Para luego, si pasa correctamente las validaciones pertinentes, registrar el item en cuestión con la informacón enviada, correspondiendo el item presente al inventario ingresado.

Figura A90: Servicio Item: Solicitud end device ”registrarItem”

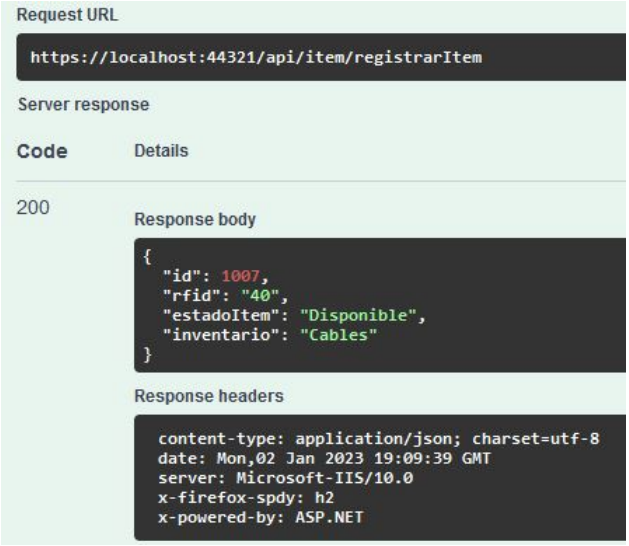

Figura A91: Servicio Item: Respuesta end device ”registrarItem”

- **Editar Item [PUT]:** End point que envía como request un objeto con los campos: ”id”, ”rfid”, ”estaDisponible”, e ”inventario”. Para luego editar el item en cuestión con el ”id” enviado según los campos ingresados.

Figura A92: Servicio Item: Solicitud end device ”editarItem”

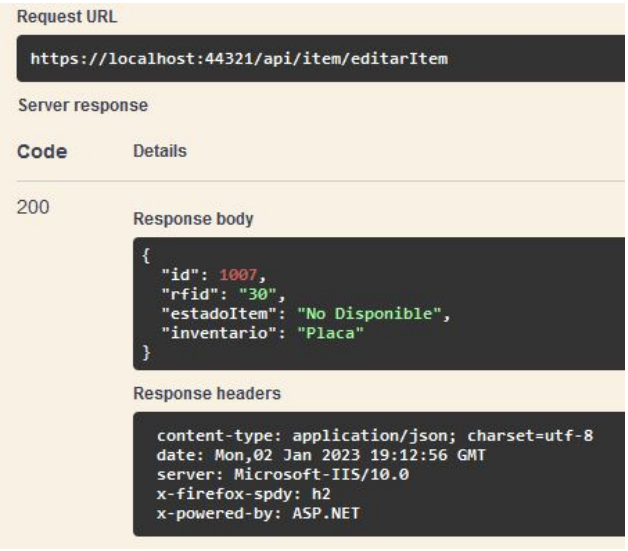

Figura A93: Servicio Item: Respuesta end device ”editarItem”

- **Eliminar Item [DELETE]:** End point que envía como request un objeto con el campo ”id”, para eliminar (de forma lógica) el item con el id enviado, haciendo las validaciones pertinentes.

Figura A94: Servicio Item: Solicitud end device ”eliminarItem”

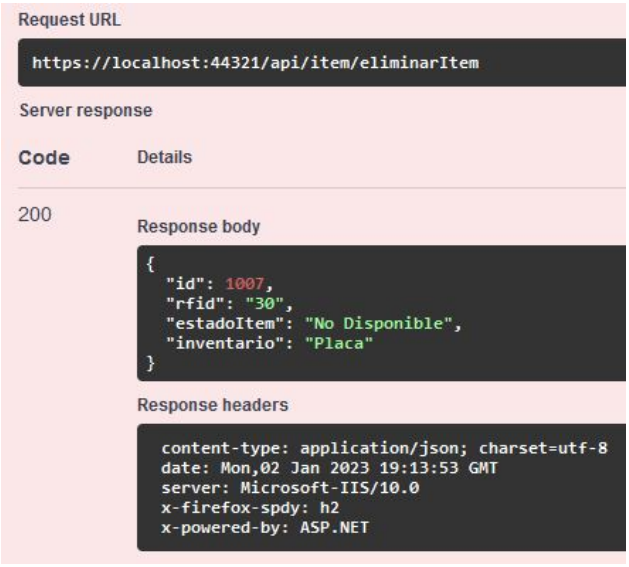

Figura A95: Servicio Item: Respuesta end device ”eliminarItem”

- **Enviar Items Detectados Zebra [POST]:** End point que envía como request un objeto con la siguiente estructura: { ”listaDatos”: [”data”: ”channel”: 0, ”eventNum”: 0, ”format”: ”string”, ”idHex”: ”string” , ”timestamp”: ”2023-02-03T08:57:15.912Z”, ”type”: ”string”] } La cual corresponde a la forma de petición que envía la antena RFID Zebra que detecta y envía los objetos RFID de forma periódica. Para luego registrar con estado ”Disponible” a todos los ítems con RFID que detecta, y si ya existían les actualiza el estado al mencionado, y en el caso de los ítems ya registrados que no sean detectados se les asigna el estado a ”No Disponible”, todos estos ítems en específico se almacenan en el Inventario con nombre ”General”.

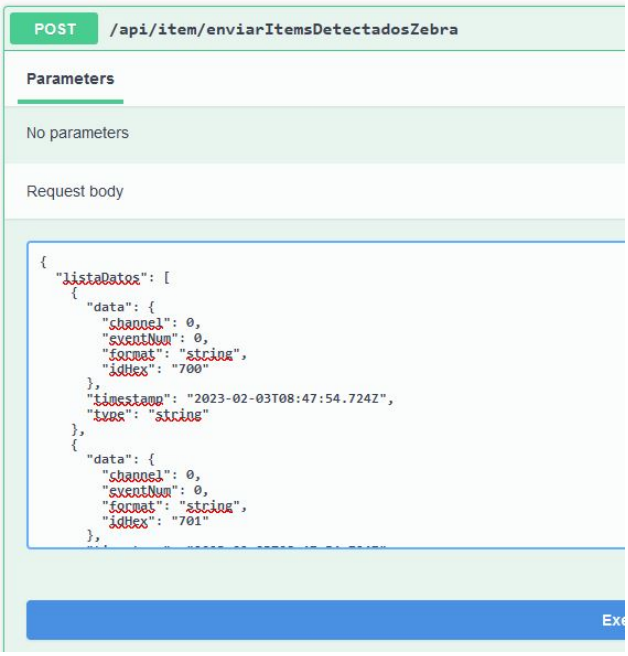

Figura A96: Servicio Item: Solicitud end device ”enviarItemsDetectadosZebra”

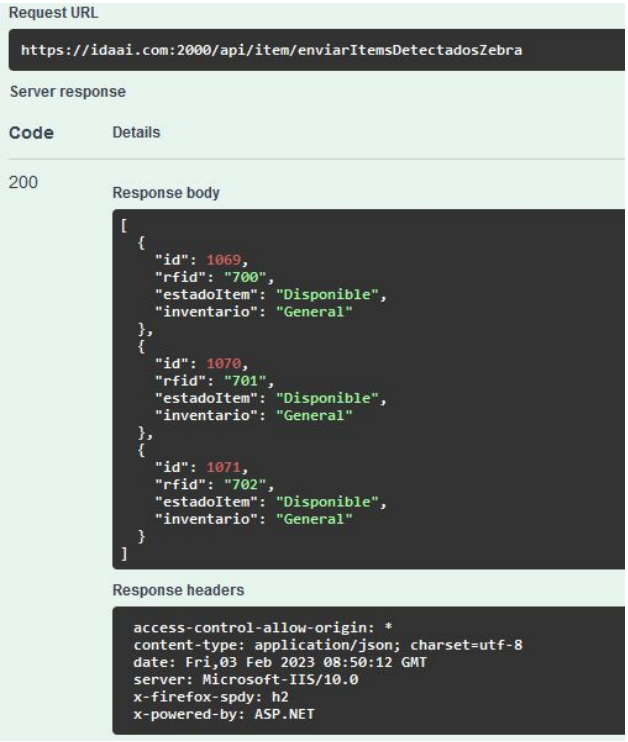

Figura A97: Servicio Item: Respuesta end device ”enviarItemsDetectadosZebra”

Servicio Préstamo: Este servicio contiene los End devices que corresponden a todas las peticiones relacionadas a los préstamos de items. Comprendiendo las consultas por diferentes campos, y peticiones de creación, actualización y eliminación de los registros en la tabla RegistroPrestamoItem. Para los endpoints de consulta se tendrán dos campos adicionales: ”RecordsPorPagina” y ”Pagina”, que sirven para la paginación de la lista de resultados.

- **Listar préstamos por estudiante [GET]:** End point que recibe el campo ”Matricula” del estudiante al que se consulta su registro de préstamos de items, y da como resultado un JSON con una lista de la información de los registros encontrados, conteniendo los siguientes campos como respuesta: ”id”, ”fechaPrestado”, ”fechaDevuelto”, ”inventario”, ”item”, ”nombres”, ”apellidos”, ”matricula”, ”email”, y ”estadoDevolucion”.

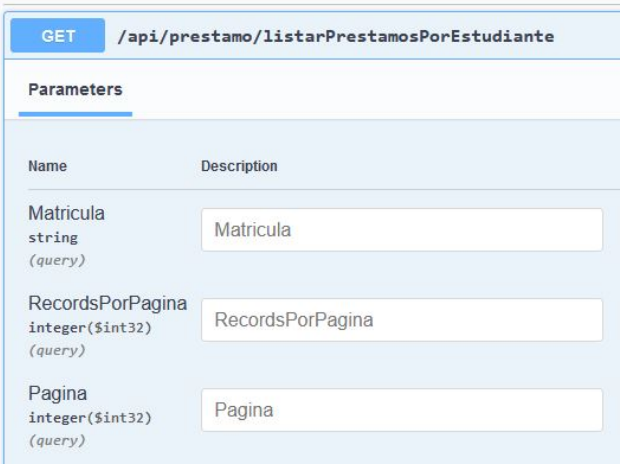

Figura A98: Servicio Préstamo: Consulta end device ”listarPrestamosPorEstudiante”

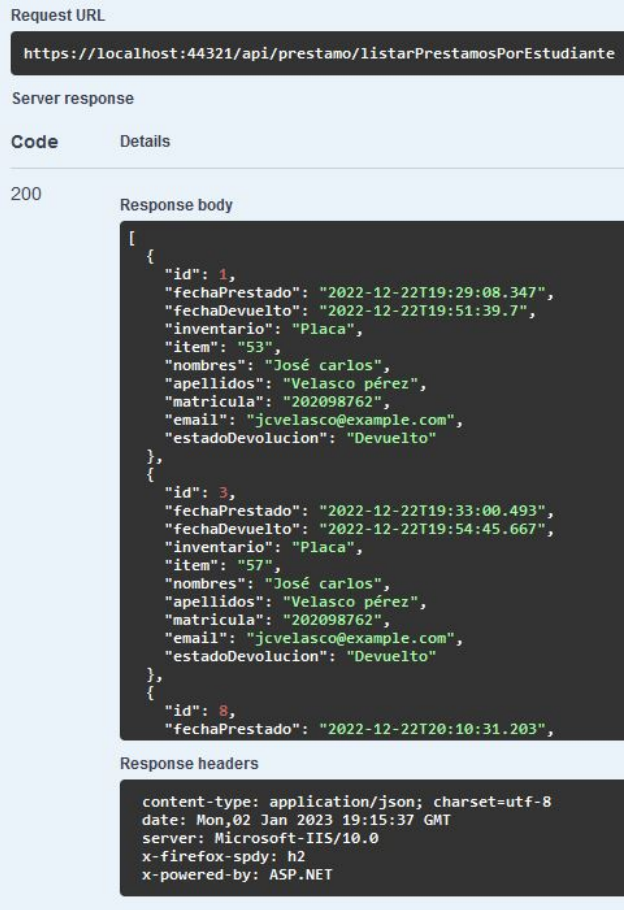

Figura A99: Servicio Préstamo: Respuesta end device ”listarPrestamosPorEstudiante”

- **Listar préstamos por módulo [GET]:** End point que recibe el campo ”Modulo” al que se consulta su registro de préstamos de items, y da como resultado un JSON con una lista de la información de los registros encontrados, conteniendo los siguientes campos como respuesta: ”id”, ”fechaPrestado”, ”fechaDevuelto”, ”inventario”, ”item”, ”modulo”, y ”estadoDevolucion”.

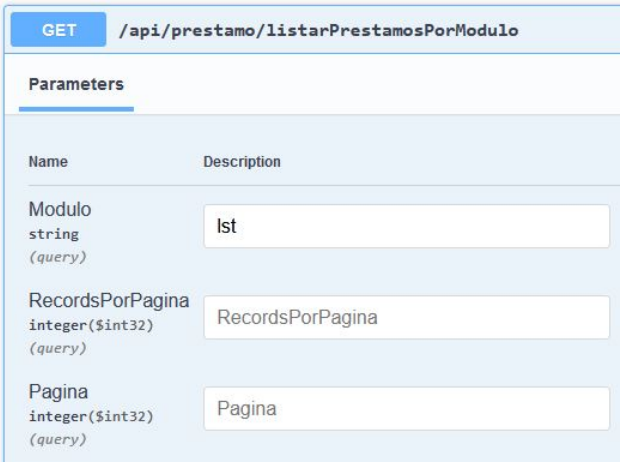

Figura A100: Servicio Préstamo: Consulta end device ”listarPrestamosPorModulo”

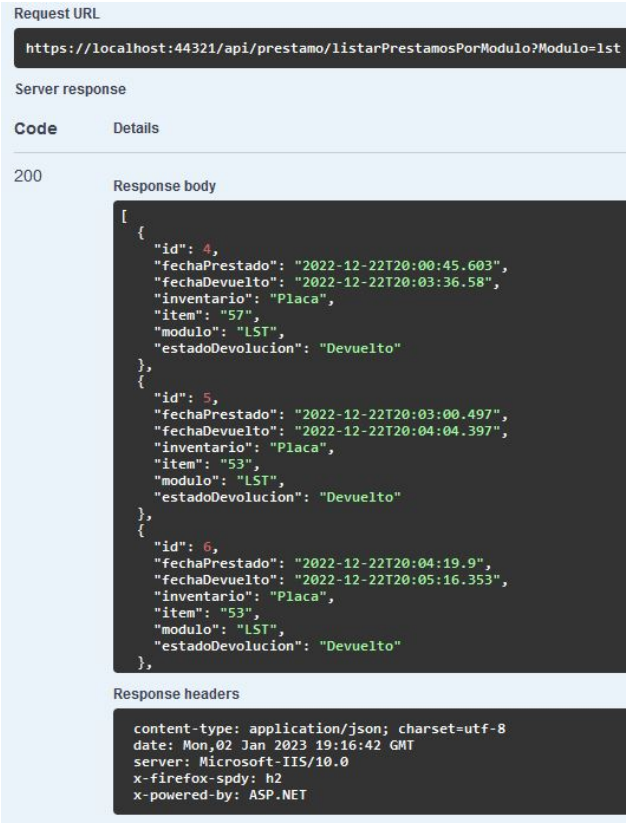

Figura A101: Servicio Préstamo: Respuesta end device ”listarPrestamosPorModulo”

- **Registrar préstamo por estudiante [POST]:** End point que envía como request un objeto con los campos ”rfid” y ”matricula”. Para luego, si pasa correctamente las validaciones pertinentes, crear el registro de prŕestamo en cuestión con el item y estudiante que corresponden al rfid y matricula ingresados respectivamente.

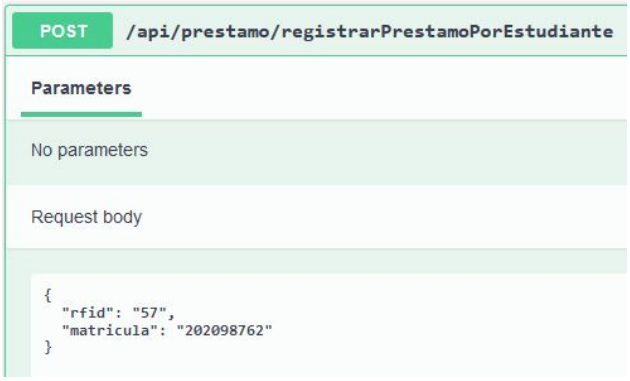

Figura A102: Servicio Préstamo: Solicitud end device ”registrarPrestamoPorEstudiante”

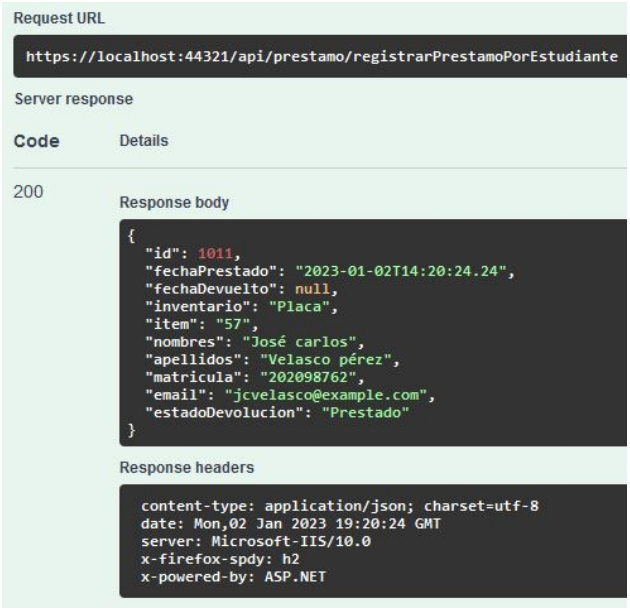

Figura A103: Servicio Préstamo: Respuesta end device ”registrarPrestamoPorEstudiante”

- **Registrar préstamo por módulo [POST]:** End point que envía como request un objeto con los campos ”rfid” y ”modulo”. Para luego, si pasa correctamente las validaciones pertinentes, crear el registro de prŕestamo en cuestión con el item y modulo que corresponden a los campos ingresados.

Figura A104: Servicio Préstamo: Solicitud end device ”registrarPrestamoPorModulo”

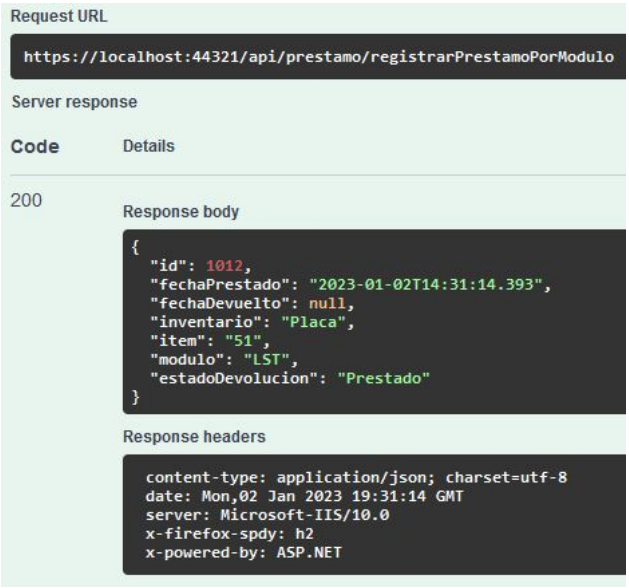

Figura A105: Servicio Préstamo: Respuesta end device ”registrarPrestamoPorModulo”

- **Registrar grupo préstamo por módulo [POST]:** End point que envía como request un objeto con los campos: ”recordsPorPagina”, ”pagina” ”modulo”, y una lista con nombre ”rfids”, que contendrá los rfids que se desea registrar. Para luego, si pasa correctamente las validaciones pertinentes, crear el/los registro/s en cuestión con el módulo e item/s, que corresponden al nombre de módulo y el/los rfis ingresados.

Figura A106: Servicio Préstamo: Solicitud end device ”registrarGrupoPrestamoPorModulo”

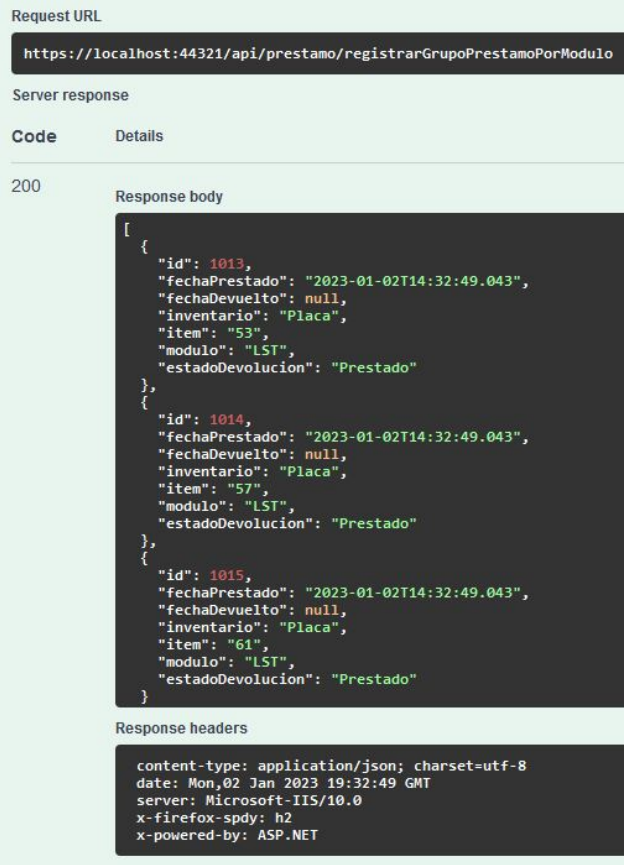

Figura A107: Servicio Préstamo: Respuesta end device ”registrarGrupoPrestamoPorModulo”

- **Devolver préstamo por estudiante [PUT]:** End point que envía como request un objeto con el campo ”id”. Sirve para cambiar el estado de devolucińde ”prestado” a ”devuelto”, del registro de prŕestamo de item a estudiante, con el id correspondiente.

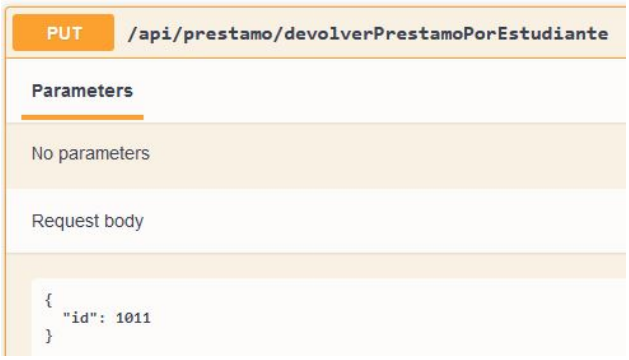

Figura A108: Servicio Préstamo: Solicitud end device ”devolverPrestamoPorEstudiante”

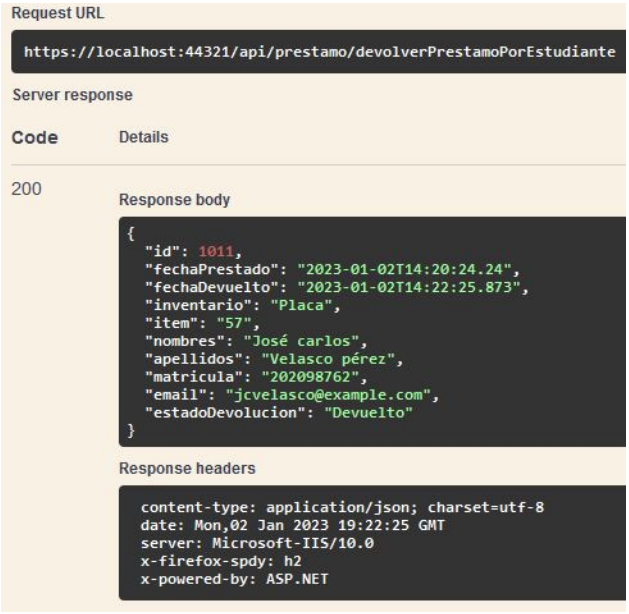

Figura A109: Servicio Préstamo: Respuesta end device ”devolverPrestamoPorEstudiante”

- **Devolver préstamo por módulo [PUT]:** End point que envía como request un objeto con el campo ”id”. Sirve para cambiar el estado de devolucińde ”prestado” a ”devuelto”, del registro de prŕestamo de item a módulo, con el id correspondiente.

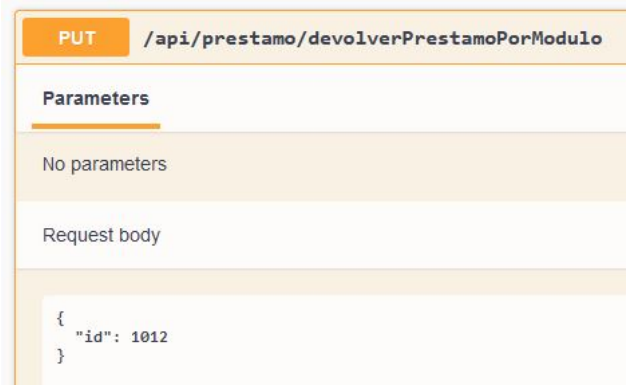

Figura A110: Servicio Préstamo: Solicitud end device ”devolverPrestamoPorModulo”

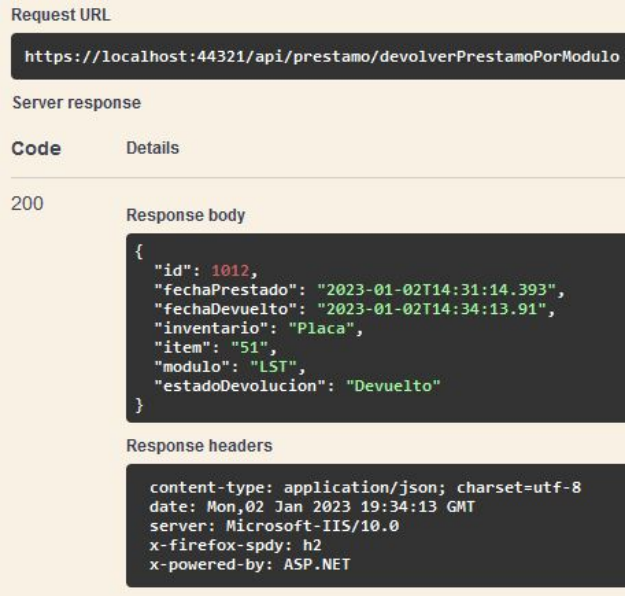

Figura A111: Servicio Préstamo: Respuesta end device ”devolverPrestamoPorModulo”

- **Eliminar préstamo por estudiante [DELETE]:** End point que envía como request un objeto con el campo ”id”, para eliminar (de forma lógica) el registro de prŕestamo de item a estudiante con el id enviado, haciendo las validaciones pertinentes.

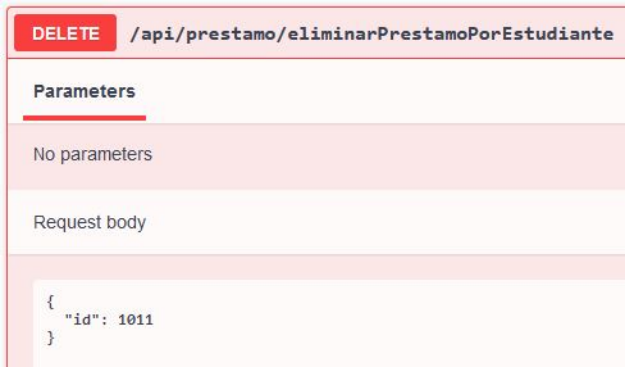

Figura A112: Servicio Prŕestamo: Solicitud end device ”eliminarPrestamoPorEstudiante”

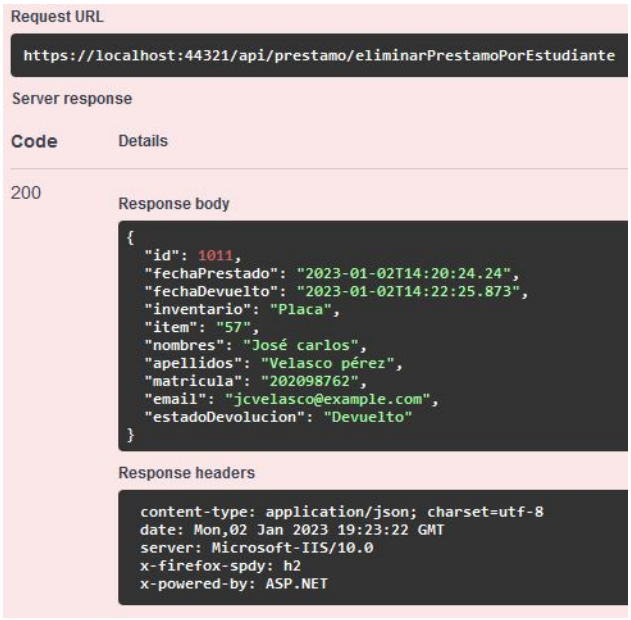

Figura A113: Servicio Préstamo: Respuesta end device ”eliminarPrestamoPorEstudiante”

- **Eliminar préstamo por módulo [DELETE]:** End point que envía como request un objeto con el campo ”id”, para eliminar (de forma lógica) el registro de prŕestamo de item a módulo con el id enviado, haciendo las validaciones pertinentes.

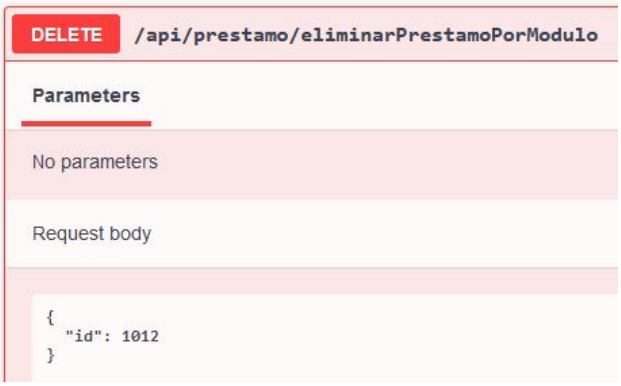

Figura A114: Servicio Prŕestamo: Solicitud end device ”eliminarPrestamoPorModulo”

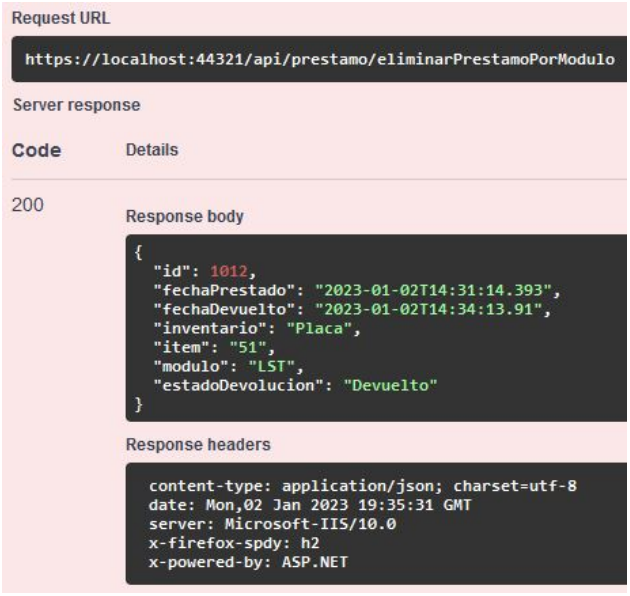

Figura A115: Servicio Préstamo: Respuesta end device ”eliminarPrestamoPorModulo”

Apéndice B: Documentación IDAAI APP

A continuación se presenta la sección correspondiente a la aplicación web desarrollada con ASP.NET Core MVC, con los ejemplos de pruebas y resultados pertinentes para cada página dentro de la aplicación.

Pantallas WEB APP:

A continuación se procederá a mostrar las pantallas de la aplicación web (IDAAI APP) correspondientes a cada sección del funcionamiento del presente proyecto, en conjunto con las consultas y respuestas a través del API con el que se conecta para mostrar dicha información (IDAAI API).

Inicio: Esta pantalla corresponde a la primera que aparecerá en cuanto abramos ingresemos a la aplicación web, nos brinda una pantalla de inicio y tres pantallas más, una para el inicio de sesión de usuario, otra para el registro de un nuevo usuario, y otra para el registro de asistencia de los estudiantes, la cual será mostrada al final de la presente sección (a diferencia de las demás, ésta será utilizada por el propio estudiante para enviar su número de matrícula y así marcar su asistencia).

- **Inicio** Pantalla de bienvenida al usuario, cuenta con una barra de navegación para el ingreso a las diferentes opciones mencionadas anteriormente.

Figura B1: Pantalla de inicio: Sección Home

- **Login** Pantalla que cumple la funcionalidad del inicio de sesión para el usuario con todas las validaciones pertinentes. Si el usuario no es válido, aparecerá un modal con dicho mensaje en la pantalla. Si el usuario es válido entonces será redirigido a la pantalla de Menú.

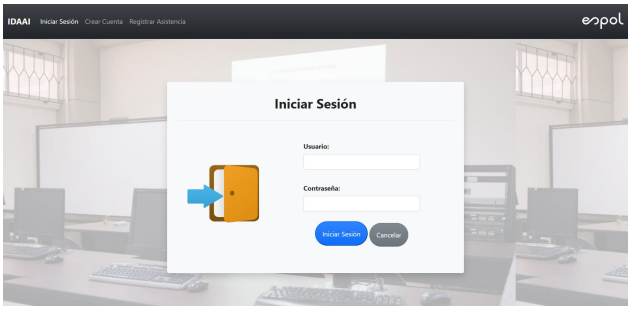

Figura B2: Pantalla de inicio: Sección Login

- **Registro** Pantalla que cumple la funcionalidad del registro de nuevo usuario, con todas las validaciones pertinentes. Si el usuario ya existe, aparecerá un modal con dicho mensaje en la pantalla. Si el usuario no existe, entonces se creará y será redirigido a la pantalla de Menú.

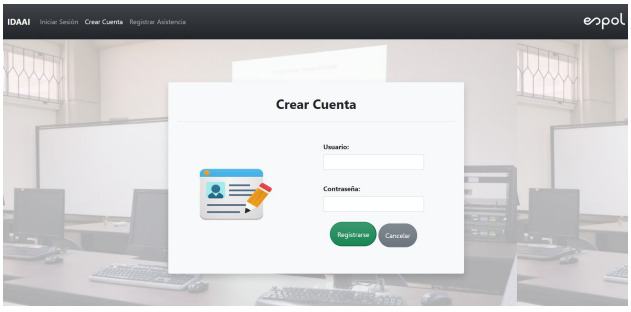

Figura B3: Pantalla de inicio: Sección Registro

Menú: Esta pantalla sirve para la selección de laboratorio e información de usuario. Cuenta con una barra superior de navegación con tres opciones: Menú, usuario y cerrar sesión. Cuya funcionalidad radica en: Redirigir a la pantalla de menú, a la pantalla de información del perfil de usuario, y cerrar sesión, respectivamente.

- **Menú** Pantalla que cumple con la funcionalidad de brindar al usuario la opción de elegir ingresar a uno de los dos modos con las que cuenta la aplicación. Los modos son: Laboratorio Abierto y Laboratorio de clases. Siendo el primero utilizado para toda la gestión mencionada en el proyecto respecto a estudiantes fuera de clases, en otras palabras, cuando el laboratorio esta en modo fuera de clases. Por otra parte, el segundo es utilizado para toda la gestión mencionada en el proyecto respecto a estudiantes dentro de clases, dicho de otra forma, cuando el laboratorio se encuentra en modo de clases. Cuando el usuario selecciona uno d elos dos modos, ingresa a la pantalla del modo correspondiente.

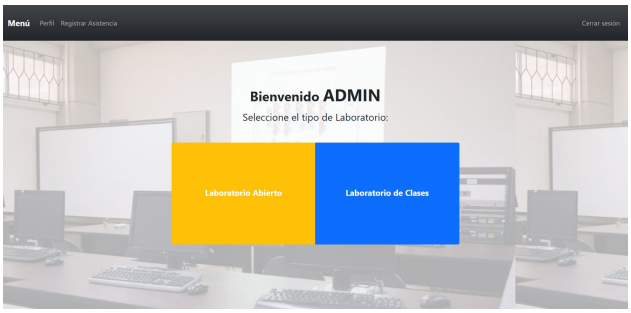

Figura B4: Pantalla de Menú: Sección Laboratorios

- **Perfil** Pantalla que cumple la funcionalidad de mostrar la información de usuario, además de permitir editar dicha información.

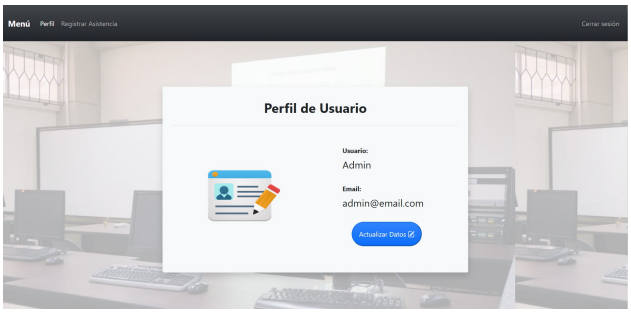

Figura B5: Pantalla de Menú: Sección Perfil de usuario

Laboratorio de Clases: Posee un conjunto de pantallas que servirán para la gestión de asistencia e inventario, y toda la información que ameriten dichas pantallas (Estudiantes, carreras, módulos, items y préstamo de items) para el laboratorio en modo en clases.

- **Asistencia** Pantalla que cumple con la funcionalidad de consultar, registrar, editar (estado de la asistencia), y eliminar los registros de asistencia.

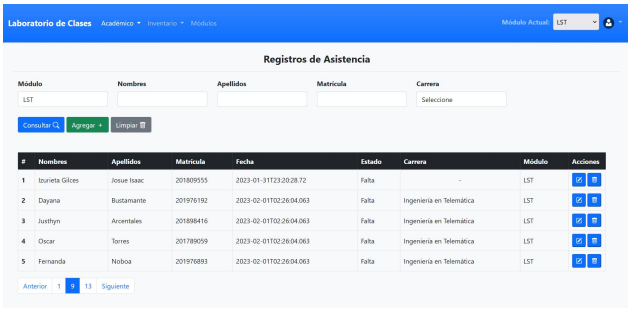

Figura B6: Pantalla de Laboratorio de Clases: Sección Asistencia

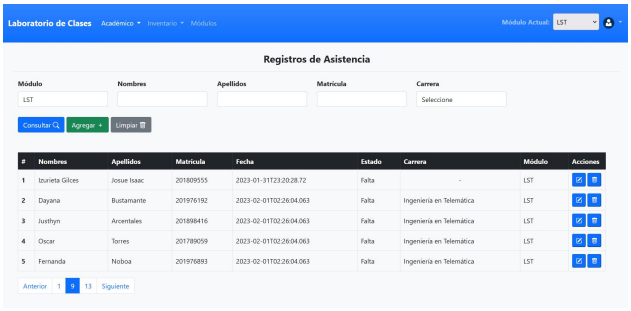

Figura B7: Pantalla de Laboratorio de Clases: Sección Asistencia - Consulta

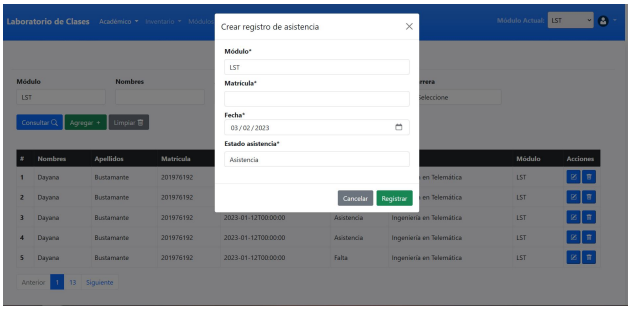

Figura B8: Pantalla de Laboratorio de Clases: Sección Asistencia - Registro

- **Inventario** Pantalla que cumple con la funcionalidad de consultar, registrar, editar, y eliminar los registros de inventario.

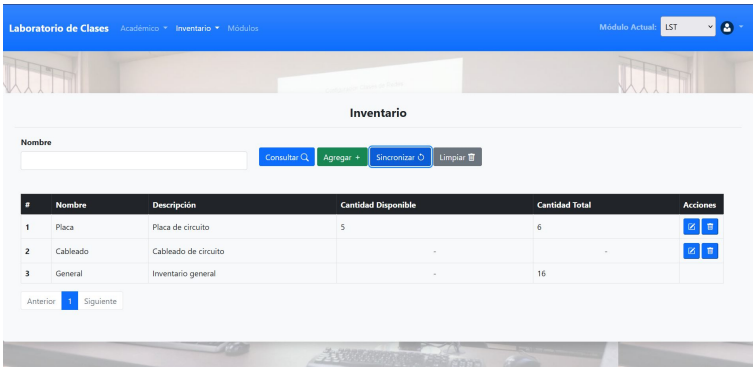

Figura B9: Pantalla de Laboratorio de Clases: Sección Inventario

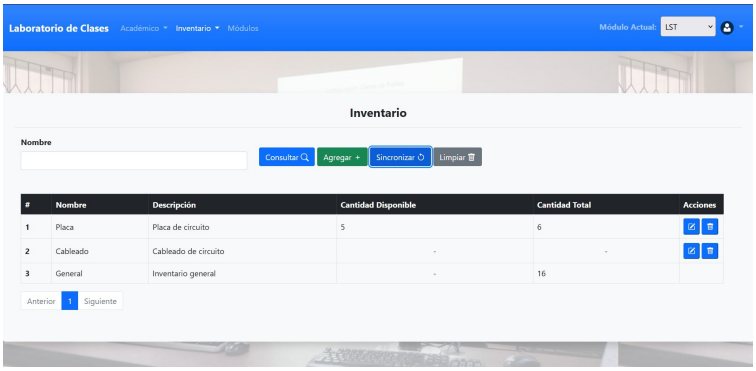

Figura B10: Pantalla de Laboratorio de Clases: Sección Inventario - Consulta

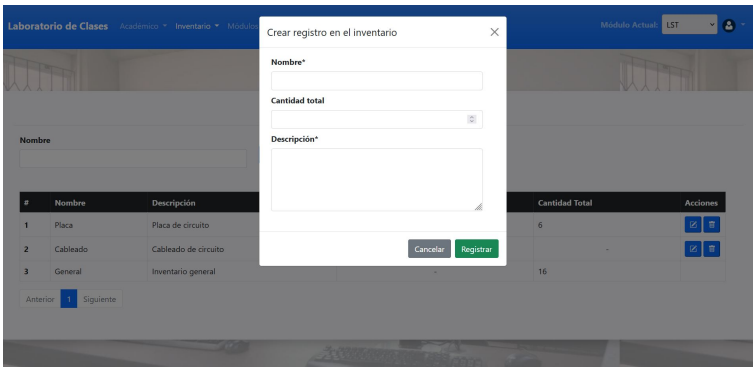

Figura B11: Pantalla de Laboratorio de Clases: Sección Inventario - Registro

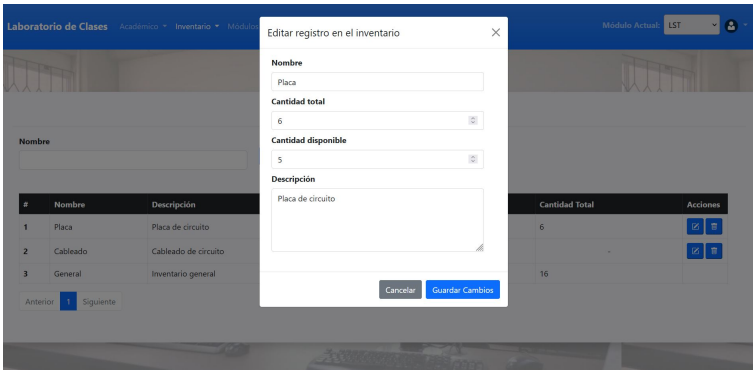

Figura B12: Pantalla de Laboratorio de Clases: Sección Inventario - Edición

- **Items** Pantalla que cumple con la funcionalidad de consultar, registrar, editar, y eliminar los items (objetos/herramientas que presta el laboratorio a los estudiantes).

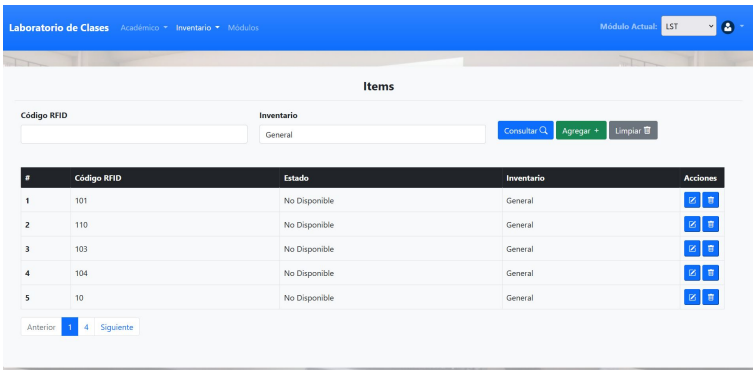

Figura B13: Pantalla de Laboratorio de Clases: Sección Items

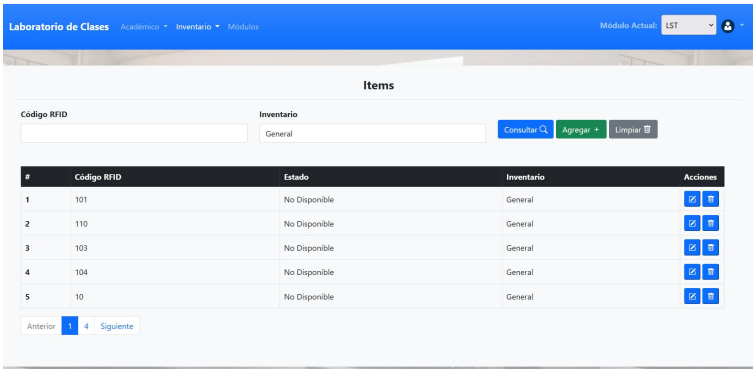

Figura B14: Pantalla de Laboratorio de Clases: Sección Items - Consulta

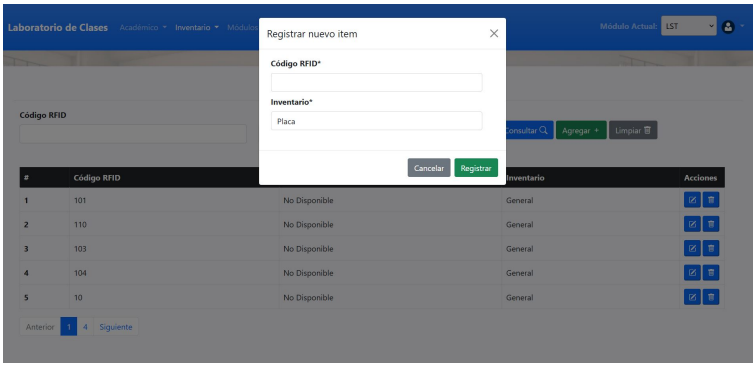

Figura B15: Pantalla de Laboratorio de Clases: Sección Items - Registro

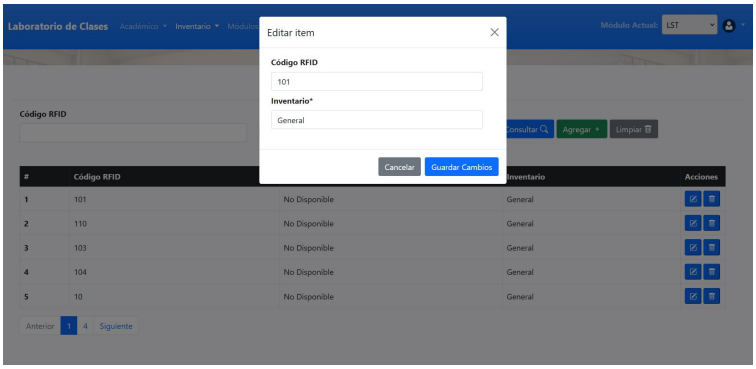

Figura B16: Pantalla de Laboratorio de Clases: Sección Items - Edición

- **Préstamo de items** Pantalla que cumple con la funcionalidad de consultar, registrar, editar (estado de devolución), y eliminar los registros de prŕestamo de items.

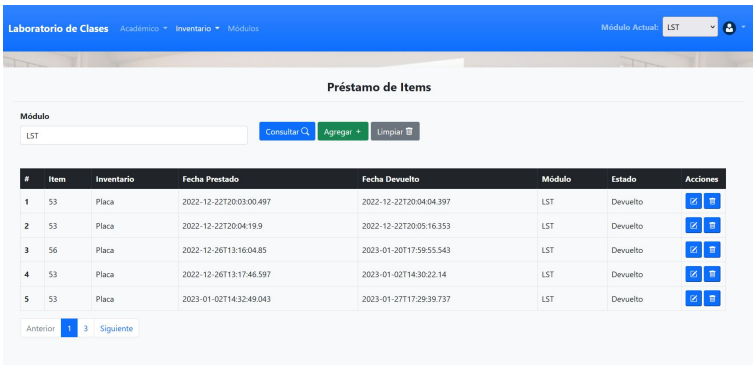

Figura B17: Pantalla de Laboratorio de Clases: Sección Préstamo de items

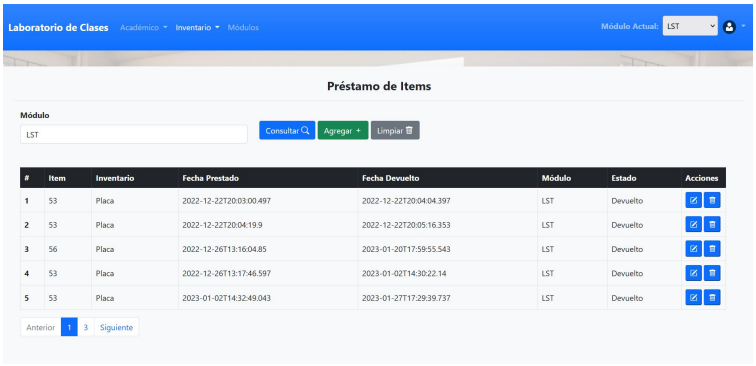

Figura B18: Pantalla de Laboratorio de Clases: Sección Préstamo de items - Consulta

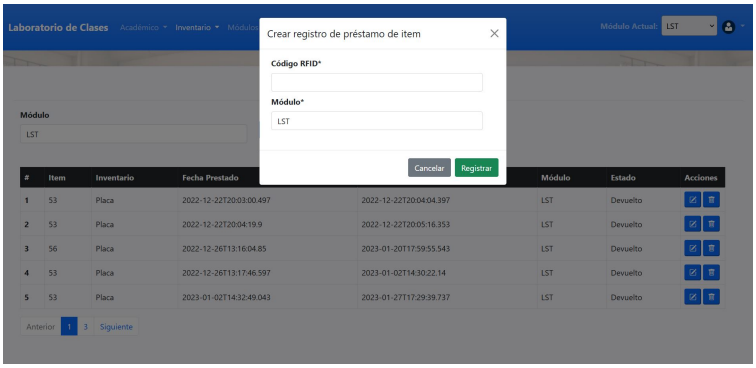

Figura B19: Pantalla de Laboratorio de Clases: Sección Préstamo de items - Registro

- **Estudiantes** Pantalla que cumple con la funcionalidad de consultar, registrar, editar, y eliminar los estudiantes.

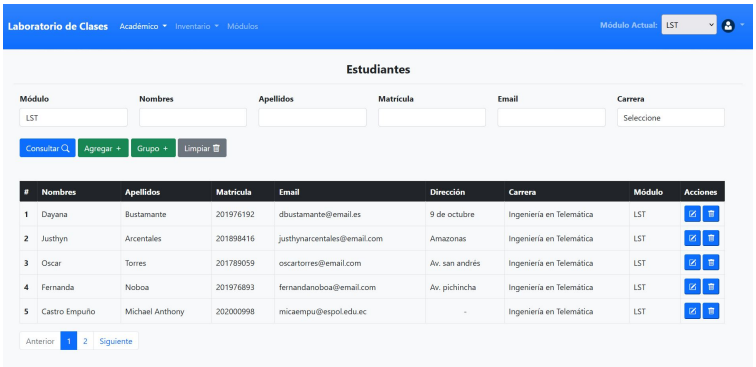

Figura B20: Pantalla de Laboratorio de Clases: Sección Estudiantes

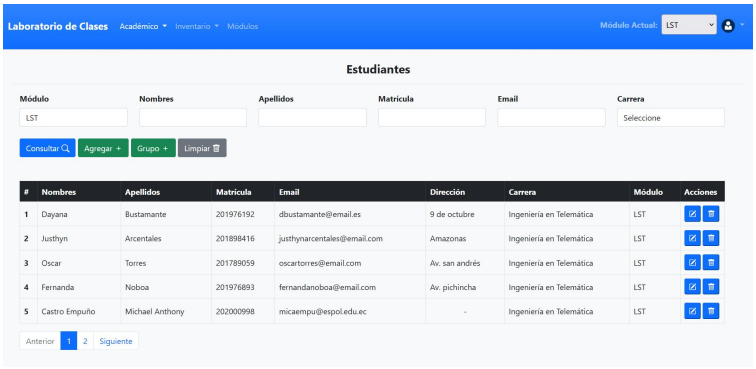

Figura B21: Pantalla de Laboratorio de Clases: Sección Estudiantes - Consulta

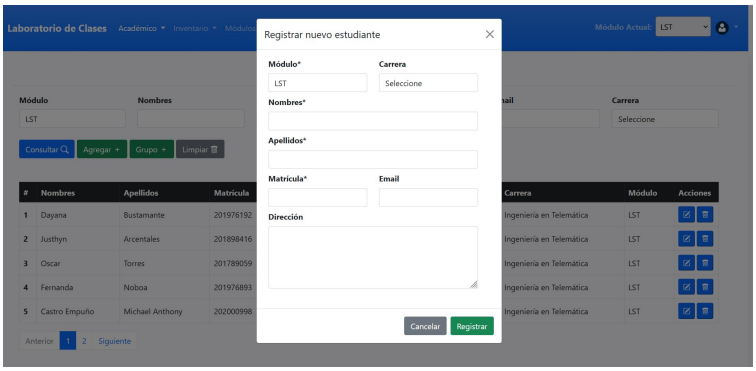

Figura B22: Pantalla de Laboratorio de Clases: Sección Estudiantes - Registro

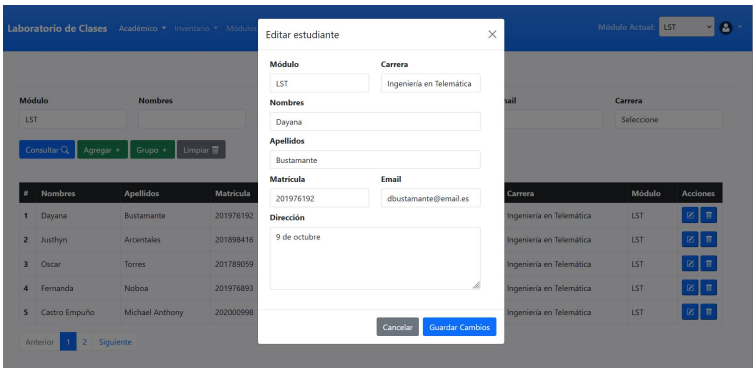

Figura B23: Pantalla de Laboratorio de Clases: Sección Estudiantes - Edición

- **Carreras** Pantalla que cumple con la funcionalidad de consultar, registrar, editar, y eliminar las carreras de los estudiantes.

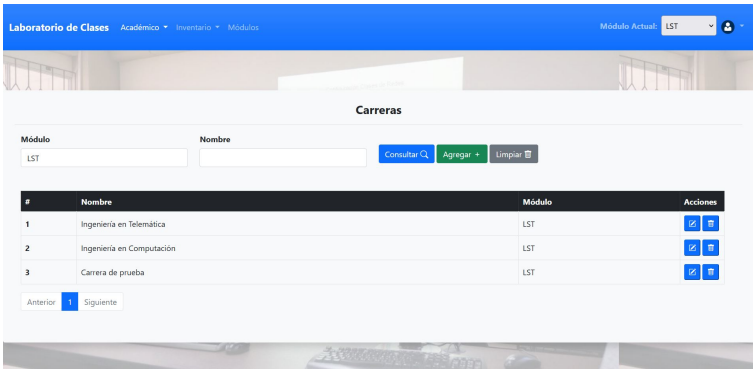

Figura B24: Pantalla de Laboratorio de Clases: Sección Carreras

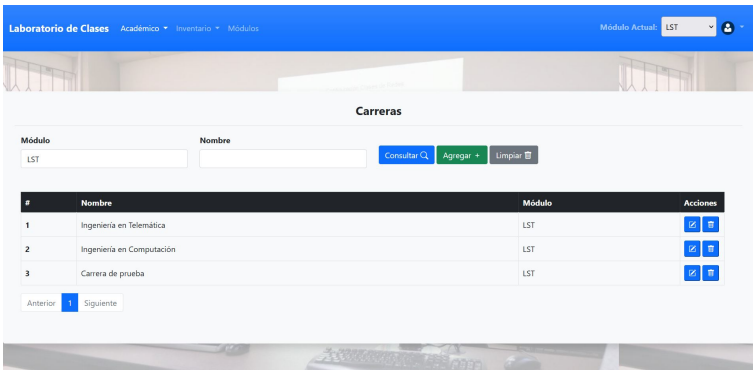

Figura B25: Pantalla de Laboratorio de Clases: Sección Carreras - Consulta

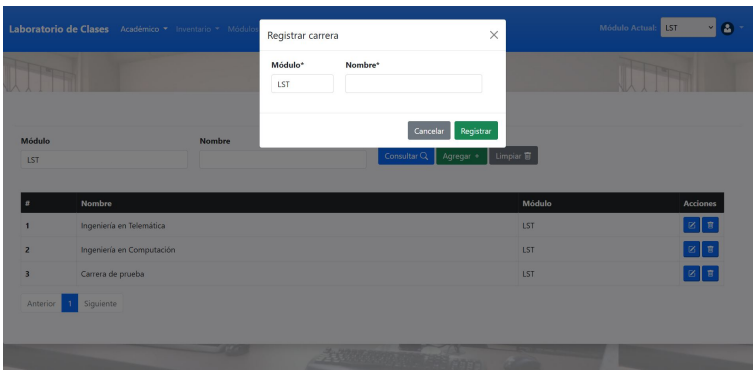

Figura B26: Pantalla de Laboratorio de Clases: Sección Carreras - Registro

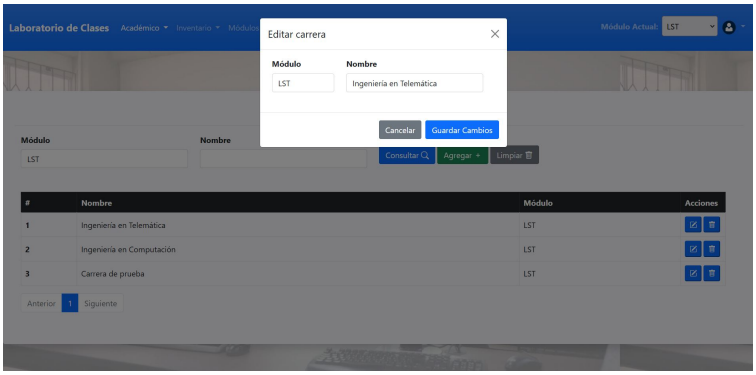

Figura B27: Pantalla de Laboratorio de Clases: Sección Carreras - Edición

- **Módulos** Pantalla que cumple con la funcionalidad de consultar, registrar, editar, y eliminar los módulos (cursos) de los estudiantes.

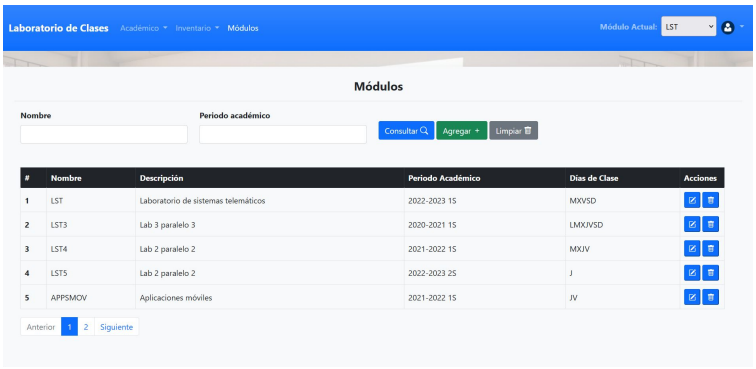

Figura B28: Pantalla de Laboratorio de Clases: Sección Módulos

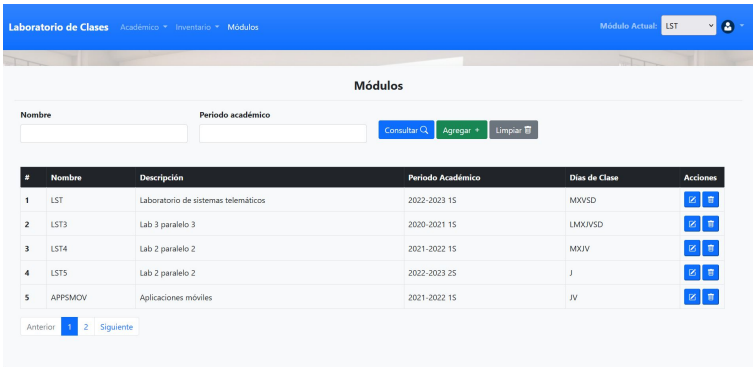

Figura B29: Pantalla de Laboratorio de Clases: Sección Módulos - Consulta

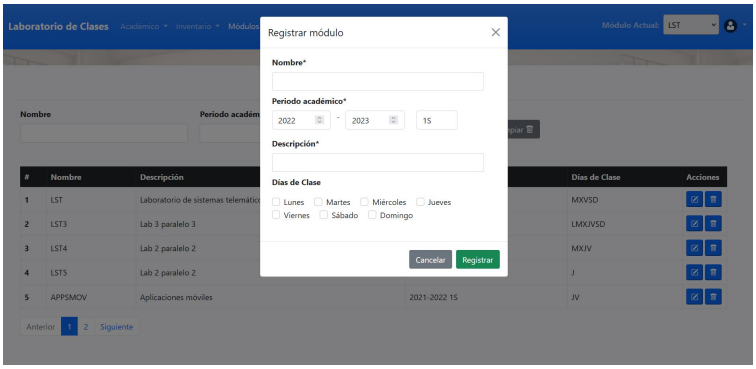

Figura B30: Pantalla de Laboratorio de Clases: Sección Módulos - Registro

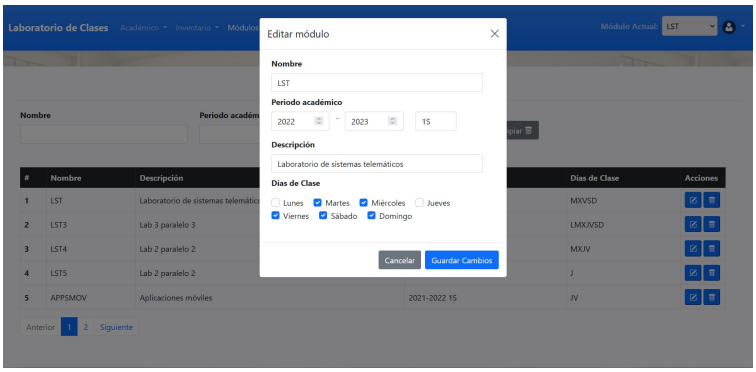

Figura B31: Pantalla de Laboratorio de Clases: Sección Módulos - Edición

Barra de navegación: Posee un conjunto de pestañas mediante las cuales podemos navegar a las diferentes pantallas de la aplicación, en la parte izquierda tiene un apartado que nos indica el tipo de laboratorio en el que nos encontramos, y que al dar click nos devuelve a la pantalla de menú. A la derecha tenemos dos pestañas desplegables:

- **Módulo actual** Permite seleccionar el módulo actual que tendrá seleccionado el usuario, dicho módulo actual será el módulo al que deberán pertenecer los estudiantes que ingresan su asistencia desde la pantalla Registro Asistencia en Inicio.

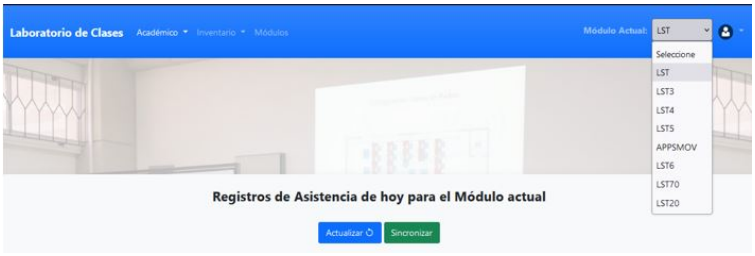

Figura B32: Pantalla de Laboratorio Cerrado: Menú desplegable Módulo actual

- **Cuenta** Permite seleccionar la opción de perfil, que llevará al usuario a la pantalla de Perfil del usuario, y cerrar sesión.

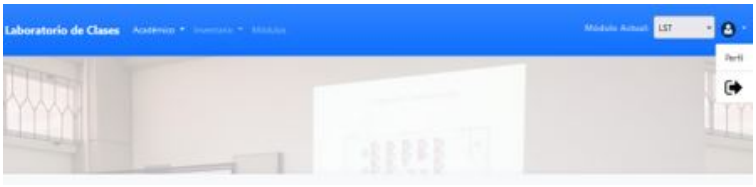

Figura B33: Pantalla de Laboratorio Abierto: Menú desplegable Cuenta

Laboratorio Abierto: Posee un conjunto de pantallas que servirán para la gestión de asistencia e inventario, y toda la información que ameriten dichas pantallas (Estudiantes, Asistencia, carreras, Inventario, items y préstamo de items) para el laboratorio en modo abierto (para estudiantes que utilizan el laboratorio fuera de horarios de clases).

- **Pantalla de laboratorio abierto** Las pantallas a mostrar aquí poseen una interfaz gráfica idéntica a las vistas con anterioridad en el laboratorio abierto, a excepción de la barra superior, y su funcionalidad propia para laboratorio abierto (en cuanto al inventario los registros que se gestionan son los mismos, pero en los estudiantes y sus pantallas relacionadas muestra datos que no tienen relación con los del laboratorio en modo clases).

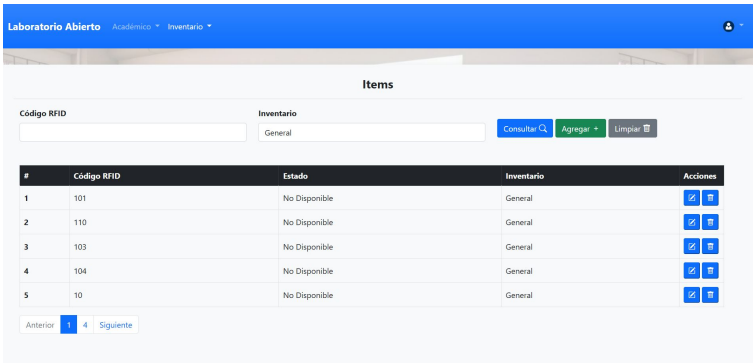

Figura B34: Pantalla de Laboratorio Abierto: Interfaz general de pantallas pertenecientes al laboratorio abierto

Registro Asistencia: Pantalla perteneciente a la página de Inicio (y de menú para uso opcional desde usuario), que tiene la función de registrar la asistencia de los estudiantes, en la que cada estudiante desde su dispositivo celular entrará a esta sección y mediante el envío de su número de matrícula, el cual, si es estudiante activo del módulo actual seleccionado por el usuario (profesor), tendrá su asistencia agregada, pero en caso de no ser así, tendrá un mensaje de respuesta dependiendo del caso que se valida.

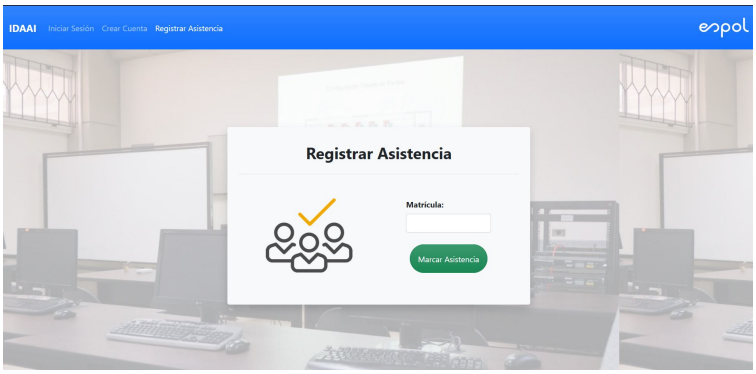

Figura B35: Pantalla de Registrar Asistencia (Inicio): Envío de asistencia por estudiantes

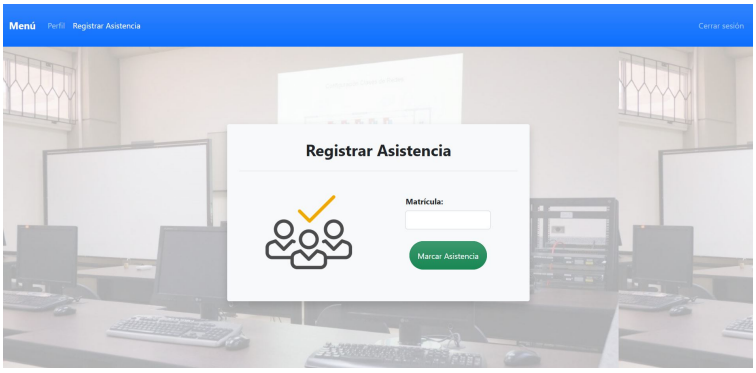

Figura B36: Pantalla de Registrar Asistencia (menu): Envío de asistencia de estudiantes por usuario

Apéndice C: Configuración lector RFID

La configuración de Zebra IoT Connector es un proceso básicamente de tres pasos:

- Agregar configuración de los endpoints
- Configurar interfaces
- Iniciar el servicio IoT Connector

Configuración del lector En esta sección se agregan las configuraciones de los endpoints MQTT y HTTP Post.

Configuración de endpoints

- **Agregar endpoint MQTT** Para esta configuración inicial se utiliza un MQTT Broker de código abierto en test.mosquitto.org. Se deben configurar los campos de nombre de endpoint, descripción, conexión, temas y certificados.

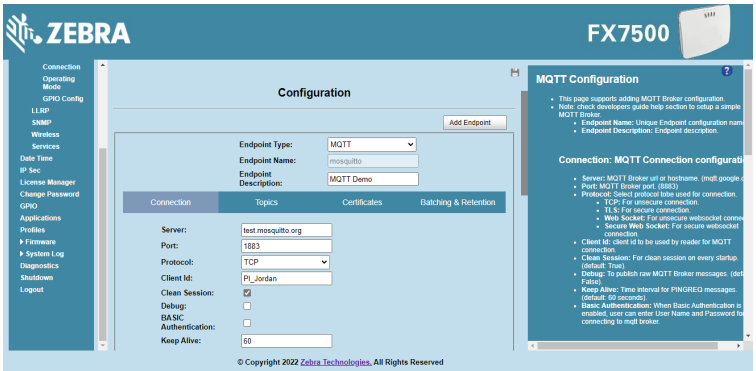

Figura C1: Configuración MQQT

- **Agregar endpoint HTTP Post** Para esta configuración inicial se utiliza un servidor HTTP de código abierto en webhook.site. Se deben configurar los campos de nombre de endpoint, descripción, conexión y certificados.

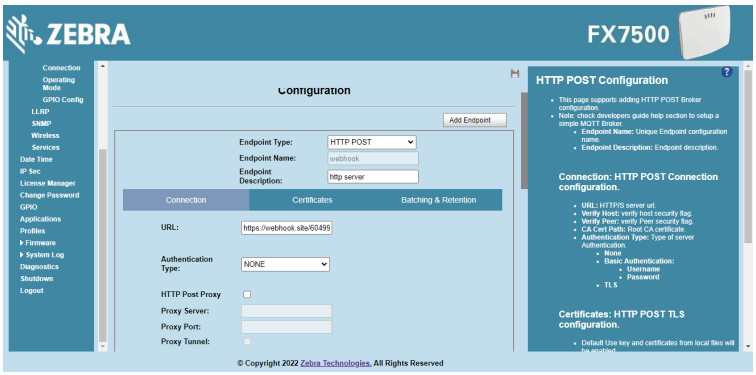

Figura C2: Configuración HTTP Post

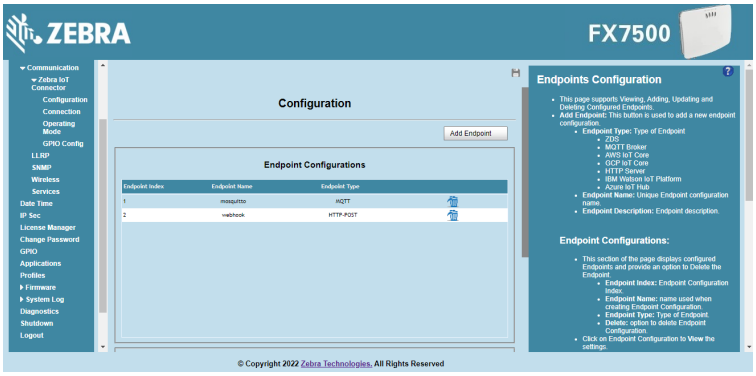

Figura C3: Endpoints configurados

Configuración de la interfaz

Interfaz Seleccione el endpoint MQTT para la interfaz de administración, interfaz de control e interfaz de eventos de administración y seleccione el endpoint HTTP Post para la interfaz de datos de etiquetas.

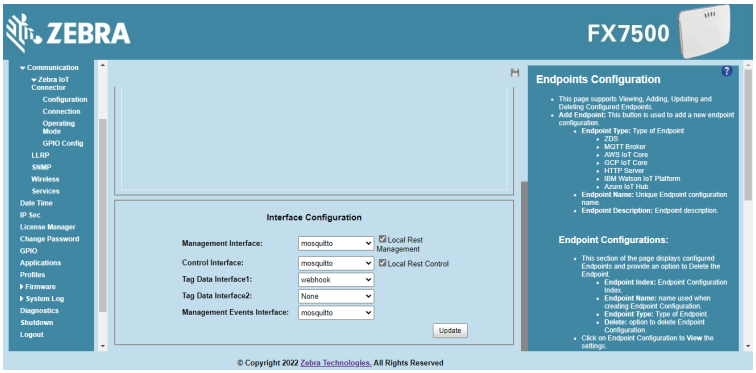

Figura C4: Configuración de las interfaces

Incio del servicio del conector Zebra IoT

Conección Una vez conectado el estado actual de la conexión de las interfaz se podrá observar en la sección Estado de conexión.

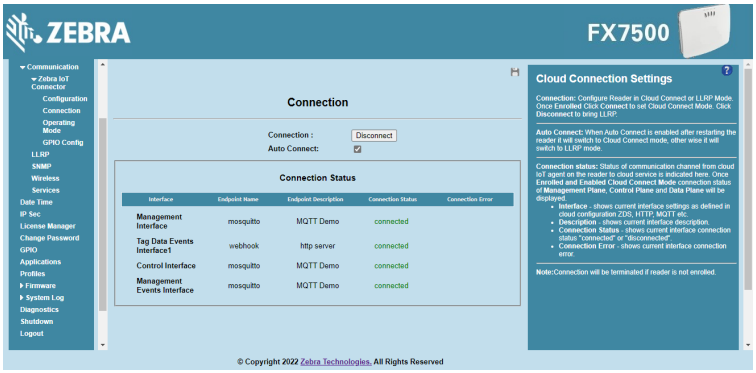

Figura C5: Conexiones

Empezar a leer las etiquetas

- **Enviar solicitud GET** Se envía una solicitud GET a la interfaz local de administración de descansos con /cloud/localREstLogin. Se debe seleccionar el tipo de autenteticación, nombre de usuario y contraseña deben ser los mismos que se utilizan para iniciar sesión en el adminstrador de consola del lector.

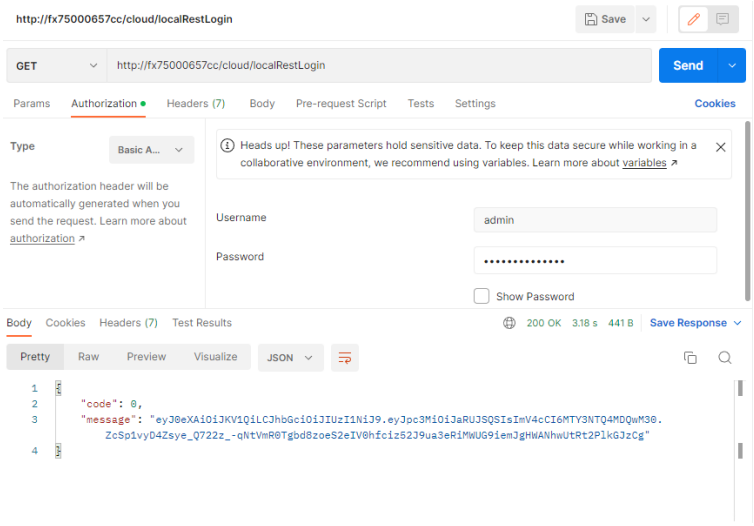

Figura C6: Solicitud GET

- **Enviar solicitud PUT** Se envía una solicitud PUT a la interfaz local de administración de descansos con /cloud/start. Se debe seleccionar el tipo de autentitación y se debe usar el token devuelto por la API localRestLogin.

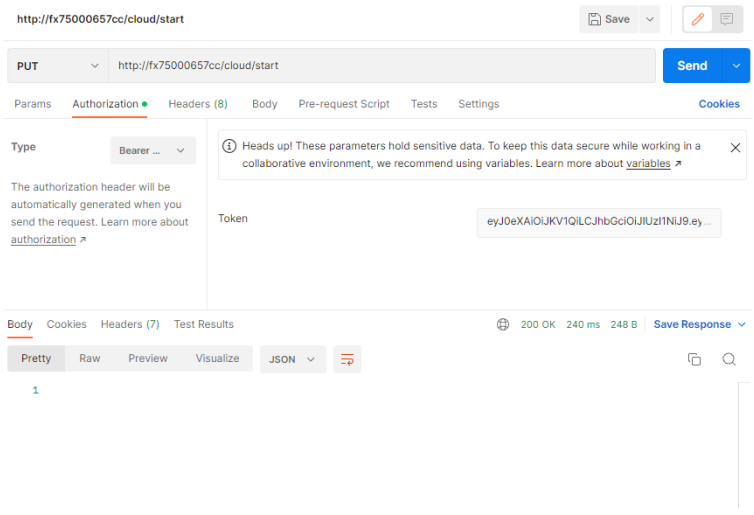

Figura C7: Solicitud PUT

Configurar cliente MQTT

Una vez instalado el cliente MQTT, debe iniciarse. Se configura MQTT X Client con todos los detalles del broker MQTT configurados en el lector y conectar.

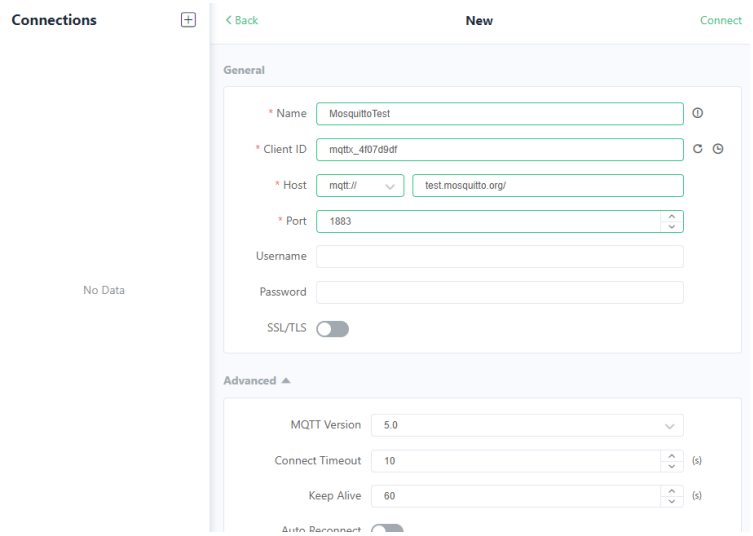

Figura C8: Configuración cliente MQTT

Interactuar con el lector

- **Iniciar lecturas** Las lecturas de las etiquetas se inician enviando el comando start al lector.

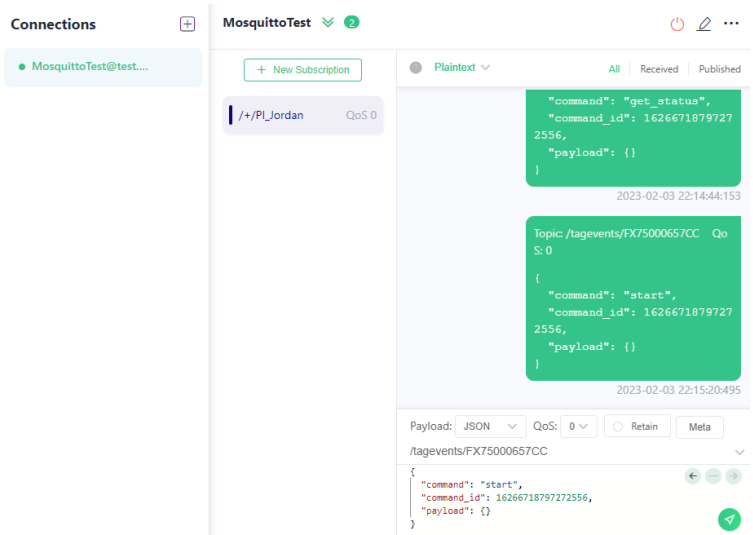

Figura C9: Iniciar lecturas

- **Modo de configuración** Para recibir los datos en el servidor se puede configurar la forma, el tiempo de lectura o seleccionar que datos se quieren recibir en el mismo.

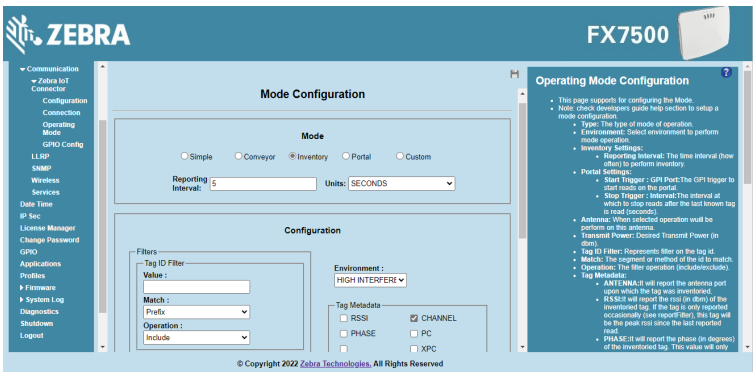

Figura C10: Modo de configuración

- **Servidor HTTP** Una vez se hayan realizado todas las configuraciones, los datos de las etiquetas se enviarán al servidor HTTP configurado en el lector.

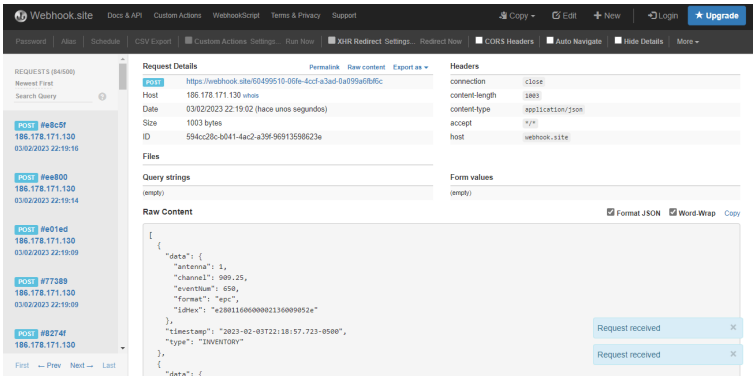

Figura C11: Datos de las etiquetas recibidos en el servidor

Apéndice D: Análisis de costos

En esta sección se presentará un análisis detallado de los costos asociados al desarrollo del proyecto. Se incluirán los costos de hardware y software, servicios profesionales, y cualquier otro costo relevante para el proyecto. Además, se explicará cómo se determinaron los costos y se proporcionará una descripción detallada de cada uno de ellos, lo que permitirá una mejor comprensión del presupuesto total del proyecto.

Sistema RFID Para el desarrollo de este sistema se describen los dispositivos a usar en la Tabla 1.

Tabla D1: Dispositivos RFID

Dispositivo	Descripción
Lector RFID	FX7500 de 2 puertos de antena
Antena RFID	AN400 de 2 conectores N hembra
Etiquetas RFID	Adhesivas UHF 860-960 Mhz

El número de antenas y lectores dependerá del tamaño del área donde se hará el inventario, el lector RFID que se está utilizando contiene 2 puertos para conectar cada uno de ellos a una antena RFID, para el área de implementación del laboratorio de sistemas telemáticos es suficiente un lector y una antena RFID. El número de etiquetas RFID dependerá de la cantidad de elementos a inventariar.

Se procede a detallar la cantidad de dispositivos y el costo total de la implementación del sistema RFID en la Tabla 2.

Tabla D2: Costos de implementación del sistema RFID

Dispositivos	**Cantidad**	**Precio unitario**	**Precio total**
Lector RFID	1	$1030	$1030
Antena RFID	1	$393.26	$393.26
Etiquetas RFID	>100	$0.20	$20
Tiempo profesional	1	$1600	$6400
Total			$7843.26

Sistema de Software (Aplicación web y API) Para el desarrollo de este sistema se describen los recursos a usar en la Tabla 3.

Tabla D3: Recursos de software

Recurso	**Descripción**
SQL Server Standard	Licencia de Gestor de base de datos de nivel profesional
Visual Studio 2022 Professional	Licencia de IDE para desarrollo y despliegue de aplicaciones
Servidor de alojamiento	Servidor utilizado para alojar en internet las aplicaciones desarrolladas

SQL Server Standard: Este costo incluye la compra del software SQL Server Standard: Servidor y la licencia para un gestor de base de datos de nivel profesional.

Visual Studio 2022 Professional: Este costo incluye la compra de la herramienta de desarrollo Visual Studio 2022 Professional y la licencia correspondiente.

Servidor de alojamiento: Este costo incluye el alquiler de un servidor que se utilizará para alojar las aplicaciones desarrolladas y hacerlas accesibles en línea.

Tiempo profesional: Este costo incluye el tiempo laboral que el profesional a cargo dedique a desarrollar el sistema de software, incluyendo la investigación, el diseño y la implementación.

Se procede a detallar las características de los recursos y el costo total de la implementación del sistema de software en la Tabla 4.

Tabla D4: Costos de implementación del sistema de software

Recursos	**Cantidad**	**Precio mensual**	**Precio Total (4 meses)**
SQL Server Standard	1	$899	$899
Visual Studio 2022 Professional	1	$499	$499
Servidor de alojamiento	2	$12	$48
Tiempo profesional	1	$1600	$6400
Total			$7846

Apéndice E: Glosario

Ambientes inteligentes: Corresponden a espacios que utilizan tecnologías de sistemas integrados y otras tecnologías de la información y la comunicación para crear entornos interactivos que acercan las computadoras al mundo físico y los problemas cotidianos. El objetivo de estos sistemas es permitir que las computadoras y los sensores participen en actividades y resultados con los que nunca antes se habían asociado, lo que permite a las personas (usuarios) interactuar con varias entidades a través de gestos, habla, movimientos o información contextual simple.[15]

Sensor: Es un tipo de dispositivo que tiene la facultad de medir alguna característica física del entorno que los rodea. La forma en que lo mide depende del tipo de sensor, convirtiéndola en un voltaje analógico o a veces en una señal digital, para que después otro dispositivo o sistema pueda utilizar dicha información para realizar alguna acción en consecuencia. Los sensores se dividen muchos tipos, por ejemplo, tenemos: Sensores de temperatura, sensores de galgas extensiométricas, sensores de celda de carga, sensores transformadores diferenciales de variable lineal (LVTD), sensores de vibración - acelerómetros, sensores de sonido, sensores de corriente, sensores ópticos, sensores de cámara, sensores digitales, etc.[16]

API: Un API (por sus siglas en inglés ”Application Programming Interface”: Interfaz de Programación de Aplicaciones) es un intermediario que permite que diferentes aplicaciones se comuniquen entre sí. A través de un API, una aplicación puede enviar solicitudes de información a otra aplicación y recibir respuestas en retorno. Esto permite que las aplicaciones interactúen y compartan datos entre sí de manera más eficiente y organizada.[17]

Endpoint: Un endpoint de API es un punto de acceso específico en un sistema de API que permite la transmisión de información entre aplicaciones y servicios en línea, funcionando como una dirección específica dentro de esa API a la cual son enviadas las peticiones.[18]

Identificación por radiofrecuencia (RFID): La identificación por radiofrecuencia o autoidentificación, es una técnica de transmisión y recepción de datos sin contacto entre el dispositivo de transporte de datos, llamado transpondedor o etiqueta RFID y un interrogador, que también se conoce como lector RFID.

Existen diversos tipos de sistemas RFID, pero todos tienen en común los siguientes elementos funcionales:

- **Lector RFID:** recibe la señal de radio y la transmite de forma comprensible para un ordenador
- **Software RFID:** gestiona la información recibida
- **Antena RFID:** actúa como un enlace entre las etiquetas y el lector
- **Etiqueta RFID:** se adhieren a los objetos que deseamos identificar

En resumen, la tecnología RFID se basa en ondas de radio para transmitir datos del lector a la etiqueta y a cambio, recibe ecos de retorno modulados de la etiqueta a través del lector.[19]

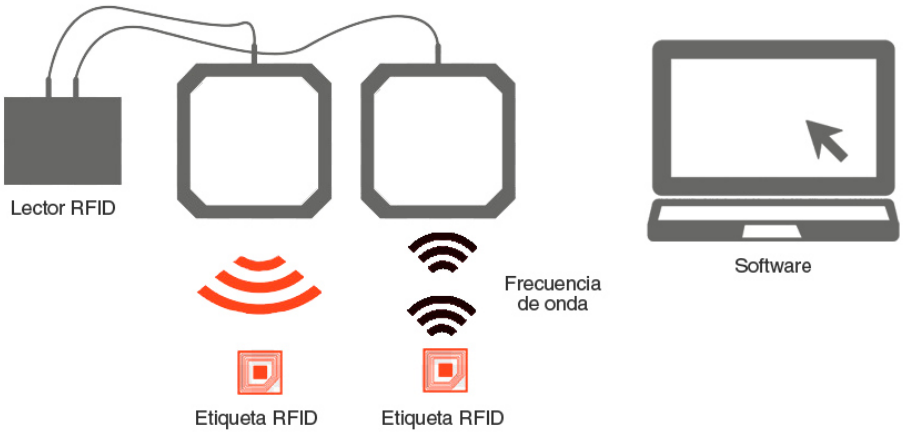

Figura E1: Sistema básico RFID

Apéndice F: Componentes de Hardware

- Antena Motorola AN400 RFID que posee las características mostradas en la Tabla 5.[20]

Tabla F1: Características de la antena AN400

Dimensiones	71.7 cm L x 31.7 cm W x 3.8 cm D
Peso	3.6 kg
Polarización	2 conjuntos de conexiones polarizadas circulares
Temperatura de funcionamiento	0° a +50° C
Conectores	2 tipo “N” hembra
Aislamiento	-37 db
VSWR	1.25
Rango de lectura	Campo lejano

• Lector fijo RFID FX7500 que posee las características mostradas en la Tabla 6.[21]

Tabla F2: Características del lector FX7500

Dimensiones	19.56 cm L x 14.99 cm W x 4.32 cm D
Peso	0.86 kg ± 0.05 kg
Temperatura de funcionamiento	-20° a +55° C
Seguridad	UL 60950-01, UL 2043, IEC 60950-1, EN 60950-1
RF/EMI/EMC	FCC Parte 15, RSS 210, EN 302 208, ICES-003 Clase B, EN 301 489-1/3
Comunicaciones	10/100 BaseT Ethernet (RJ45) con soporte POE; Cliente USB (USB Tipo B)*, Puerto Host USB (Tipo A)*
E/S	2 entradas, 3 salidas, ópticamente aisladas (Bloque de terminales)
Fuente de alimentación	POE, POE+ o +24 V CC (aprobado por UL) La operación de 12 V-48 VCC puede ser compatible
Puertos de antena	2 puertos monoestáticos (Voltity TNC)
CPU	Texas Instruments AM3505 (600 Mhz)
Memoria	Flash 512 MB; DRAM 256 MB
Sistema operativo	Linux
Protocolos de gestión	RM 1.0.1 (con XML sobre HTTP/HTTPS y Encuadernación SNMP); RDMP
Firmware	BSP 3.21.23
Servicios de red	DHCP, HTTPS, FTPS, SFPT, SSH, HTTP, FTP, SNMP y NTP
Frecuencia (banda UHF)	Lector global: 902 MHz-928 MHz
Potencia de transmisión de salida	10 dBm a +31,5 dBm
Max. sensibilidad de recepción	-82 dBm
Dirección IP	Estática y dinámica
Protocolo de interfaz de host	LLRP

- Etiquetas RFID que poseen las características mostradas en la Tabla 7.[22]

Tabla F3: Características de las etiquetas

Dimensiones	7.3 cm L x 1.7 cm W
Peso	0.634 onzas
Frecuencia	UHF 860-960MHZ
Chip	ALIEN Higgs-3
Protocolo	EPC Clase 1 Gen2, ISO18000-6C
Distancia de lectura	1 a 26.2 ft (depende del lector)
Aplicaciones	Gestión de la cadena de suministro, logística de distribución, autenticación de productos, inventario y seguimiento de activos, manejo y seguimiento de equipaje

Apéndice G: Detalles de base de datos, API y aplicación web

Creación y configuración de la base de datos: Corresponde a la instalación y configuración de la base de datos en SQL Server para el almacenaje de toda la información que utilizará la aplicación web a través de la API web para mostrarla al usuario en las diferentes pantallas. Esta información corresponde principalmente a los datos que guardan las tablas para la funcionalidad que involucra el proyecto, habiendo construido para este fin las siguientes tablas:

Usuarios: Posee datos de la información de usuario que es quien, luego de iniciar sesión, hará uso tanto del web API como de la aplicación web. Contiene las siguientes columnas: Id, Usuario, Password, Email, ModuloActualId (Id de tabla Modulo que servirá para que el usuario pueda seleccionar el módulo habilitándolo para que los estudiantes que pertenecen a dicho módulo puedan ingresar su respectiva asistencia), UltimaEntrada y Estado.

Estudiantes: Posee datos de la información de los estudiantes. Contiene las siguientes columnas: Id, Nombres, Apellidos, Matricula, Email, Direccion, CarreraId (Id de tabla Carreras), ModuloId (Id de tabla Modulos), UsuarioId (Id de tabla Usuario) y Estado.

RegistroAsistencia: Posee datos de la información de los registros de asistencia de los estudiantes. Contiene las siguientes columnas: Id, Fecha, EstudianteId (Id de tabla Estudiantes), EstadoAsistenciaId (Id de tabla EstadoAsistencia), UsuarioId (Id de tabla usuarios) y Estado.

EstadoAsistencia: Posee datos de la información de los estados de asistencia (EstadoAsistenciaId) para la tabla RegistroAsistencia. Contiene las siguientes columnas:

Id, Estado (en esta tabla estado corresponde a la descripción del estado de asistencia, existiendo los siguientes campos: Id=1 => Estado=”Asistencia”, Id=2 => Estado=”Falta”).

Carreras: Posee datos de la información de los registros de las carreras de los estudiantes. Contiene las siguientes columnas: Id, Nombre, ModuloId (Id de la tabla Modulo), UsuarioId (Id de tabla usuarios) y Estado.

Modulos: Posee datos de la información de los registros de los módulos, que corresponde en sí a los cursos. Contiene las siguientes columnas: Id, Nombre, Descripcion, PeriodoAcademico, DiasClase, UsuarioId (Id de tabla usuarios) y Estado.

Inventario: Posee datos de la información de los registros del inventario (clasificación de objetos). Contiene las siguientes columnas: Id, Nombre, Descripcion, CantidadDisponible, CantidadTotal, UsuarioId (Id de tabla usuarios) y Estado.

Items: Posee datos de la información de los ítems, que son los objetos únicos registrado con su respectivo código RFID, cada uno de estos objetos (ítems) pertenece a un registro en específico del inventario. Contiene las siguientes columnas: Id, Rfid, EstadoItemId (Id de la tabla EstadoItem), InventarioId (Id de la tabla Inventario), UsuarioId (Id de tabla usuarios) y Estado.

EstadoItem: Posee datos de la información de los estados de los ítems (EstadoItemId) para la tabla Items. Contiene las siguientes columnas: Id, Descripcion (corresponde a la descripción del estado de asistencia, existiendo los siguientes campos: Id=1 => Descripcion=”Disponible”, Id=2 => Descripcion=”No Disponible”).

RegistroPrestamoItem: Posee datos de la información de los registros de préstamos de ítems a estudiantes o módulos (según corresponda para laboratorio abierto y laboratorio de clases respectivamente). Contiene las siguientes columnas: Id, FechaPrestado, FechaDevuelto, ItemId (Id de la tabla Items), EstadoDevolucionId (Id de la tabla EstadoDevolucion), EstudianteId (Id de la tabla Estudiante), ModuloId (Id de la tabla Modulo), ModoClase (indicará al sp-prestamo en su funcionalidad pertinente si el registro

que esta procesando pertenece a laboratorio clases o laboratorio abierto), UsuarioId (Id de tabla usuarios) y Estado.

EstadoDevolucion: Posee datos de la información de los estados de la devolución de los ítems a estudiantes o módulos (EstadoDevolucionId) para la tabla RegistroPrestamoItem. Contiene las siguientes columnas: Id, Descripcion (corresponde a la descripción del estado de devolución del ítem en cuestión, existiendo los siguientes campos: Id=1 => Descripcion="Prestado", Id=2 => Descripcion="Devuelto", Id=3 => Descripcion="Prestado con atraso", Id=4 => Descripcion="Devuelto con atraso").

En todas las tablas en las que no se especifique lo contrario, la columna "Estado" sirve para la funcionalidad de borrado lógico que se implementó durante el desarrollo del proyecto, siendo Estado=1 cuando el registro "existe", y Estado=2 cuando el registro "no existe". Tambien se crearon una variedad de procedimientos almacenados de los cuales hará uso el servicio WEB API, que es donde se encuentra en mayor parte la lógica de negocio. Siendo dichos procesos almacenados los siguientes:

sp-usuario: Funcionalidades de usuario, correspondiendo éstas al login, registro, actualización de información de usuario, obtención de datos de usuario, actualización de módulo actual seleccionado por el usuario y renovación de token.

sp-estudiante: Funcionalidades de consulta, creación, actualización y eliminación para los registros de información de los estudiantes.

sp-registroasistencia: Funcionalidades de consulta, creación, actualización y eliminación para los registros de asistencia de los estudiantes.

sp-carrera: Funcionalidades de consulta, creación, actualización y eliminación para los registros de información de las carreras de los estudiantes.

sp-modulo: Funcionalidades de consulta, creación, actualización y eliminación para los registros de información de los módulos.

sp-inventario: Funcionalidades de consulta, creación, actualización y eliminación para los registros de información del inventario.

sp-item: Funcionalidades de consulta, creación, actualización y eliminación para los registros de información de los ítems únicos que corresponden al inventario con el que se registra.

sp-prestamoitem: Funcionalidades de consulta, creación, actualización y eliminación para los registros de préstamo de los ítems a módulos en laboratorio de clases y estudiantes en laboratorio abierto.

Creación de API REST: Desarrollo de servicio API REST con ASP.NET Core Web API cuyo despliegue se realizó con IIS, por sus siglás en inglés: ”Internet Information Services”. La presente web API se conectará a la base de datos configurada y traerá la información pertinente en formato JSON, utilizando los procedimientos almacenados desarrollados dependiendo del endpoint que esté siendo utilizado. Dichos endpoints internos de cada ruta del API corresponden a cada parte de la lógica empleada, a continuación se lista los mismos (para mayor información sobre la funcionalidad y ejemplos de uso de los mismos revisar la sección de documentación al final del documento):

Asistencia: Servicio cuya ruta es ”/api/asistencia” y que encierra los siguientes endpoints: GET ”listarPorNombres”, GET ”listarPorCarrera”, GET ”listarPorModulo”, GET ”listarTodos”, GET ”consultarPorMatricula”, POST ”registrarRegistroAsistencia”, PUT ”editarRegistroAsistencia”, GET ”consultarClase”, GET ”sincronizarAsistenccia”, PUT ”establecerAsistencia”, DELETE ”eliminarRegistroAsistencia”.

Estudiante: Servicio cuya ruta es ”/api/estudiante” y que encierra los siguientes endpoints: GET ”listarPorNombres”, GET ”listarPorCarrera”, GET ”listarPorModulo”, GET ”listarTodos”, GET ”consultarPorMatricula”, GET ”consultarPorEmail”, POST ”registrarEstudiante”, POST ”registrarGrupoEstudiante”, POST ”procesarRegistroGrupoEstudiante”, PUT ”editarEstudiante”, DELETE ”eliminarEstudiante”.

Carrera: Servicio cuya ruta es ”/api/carrera” y que encierra los siguientes endpoints: GET ”listarPorNombre”, GET ”listarPorModulo”, GET ”listarTodos”, POST ”registrarCarrera”, PUT ”editarCarrera”, DELETE ”eliminarCarrera”.

Modulo: Servicio cuya ruta es ”/api/modulo” y que encierra los siguientes endpoints: GET ”listarPorNombre”, GET ”listarPorPeriodoAcademico”, GET ”listarTodos”, POST ”registrarModulo”, PUT ”editarModulo”, DELETE ”eliminarModulo”.

Inventario: Servicio cuya ruta es ”/api/inventario” y que encierra los siguientes endpoints: GET ”listarPorNombre”, POST ”registrarInventario”, PUT ”editarInventario”, DELETE ”eliminarInventario”, POST ”actualizarCantidadesInventario”.

Item: Servicio cuya ruta es ”/api/item” y que encierra los siguientes endpoints: GET ”listarItems”, POST ”registrarItem”, PUT ”editarItem”, DELETE ”eliminarItem”, POST ”enviarItemsDetectados”, POST ”enviarItemsDetectadosRfid”, POST ”enviarItemsDetectadosZebra”.

Prestamo: Servicio cuya ruta es ”/api/prestamo” y que encierra los siguientes endpoints: GET ”listarPrestamosPorEstudiante”, GET ”listarPrestamosPorModulo”, POST ”registrarPrestamoPorEstudiante”, POST ”registrarGrupoPrestamoPorModulo”, PUT ”devolverPrestamoPorEstudiante”, PUT ”devolverPrestamoPorModulo”, DELETE ”eliminarPrestamoPorEstudiante”, DELETE ”eliminarPrestamoPorModulo”.

Usuario: Servicio cuya ruta es ”/api/usuario” y que encierra los siguientes endpoints: POST ”loginUsuario”, POST ”registrarUsuario”, PUT ”editarUsuario”, DELETE ”eliminarUsuario”, GET ”obtenerUsuario”, PUT ”actualizarFecha”, PUT ”actualizarModuloActual”, GET ”renoverToken”.

Creación de aplicación Web: Desarrollo de aplicación Web, que tendrá como fin la monitorización y gestión por el usuario (profesor) de la información correspondiente a los estudiantes y sus registros de asistencia, además del inventario y toda la gestión de sus componentes (ítems y registros de préstamo de ítems). Para este fin, se utilizaron las siguientes tecnologías: HTML, CSS, Javascript, el Framework de .NET (ASP.NET Core

MVC) y librerías como Boostrap y Jquery. La aplicación presenta una serie de páginas a las que el usuario podrá ingresar para la gestión correspondiente, y una página para el registro de asistencia de los estudiantes por medio de su número de matrícula (a esta pantalla los estudiantes podrán ingresar sin necesidad de registrarse o iniciar sesión, y registrar dicha su respectiva matrícula a través de sus dispositivos móviles). En cuanto a las demás páginas, que serán las que utilice el usuario, tenemos las siguientes:

Inicio: Presenta una pantalla de inicio con bienvenida para el usuario, indicando de forma breve y en forma de título el funcionamiento de la aplicación.

Iniciar Sesión: Presenta la funcionalidad para el inicio de sesión del usuario, una vez ingrese se guardará su información de inicio de sesión para que no tenga que volver a ingresar los datos.

Registro de nueva cuenta de usuario: Presenta la funcionalidad para el registro de nueva cuenta de usuario, una vez ingrese se guardará su información de inicio de sesión para que no tenga que volver a ingresar los datos.

Menú: Presenta una pantalla de menú donde podrá seleccionar el tipo de laboratorio al que desea ingresar para su debida gestión, siendo éstos el laboratorio de clases y laboratorio abierto.

Perfil: Presenta la información del usuario actual, con la opción de poder editar dichos datos.

Cerrar sesión: Cierra la sesión del usuario actual y regresa a la pantalla de Inicio.

Laboratorio de clases - Clases: Página que presenta una tabla con la información de la asistencia de los estudiantes que se encuentran en clases (muestra la información a fecha del día actual y del módulo actual seleccionado), pudiendo actualizar dichos registros o sincronizar (forzar la preparación de los registros de asistencia con estado ”Falta” para los respectivos estudiantes, los cuales al registrar su asistencia cambiarán dicho estado

a ”Asistencia”), permite además al usuario actualizar los estados de asistencia y también la eliminación de dichos registros, facilitando así la gestión correspondiente.

Laboratorio de clases - Asistencia: Página que presenta la consulta, registro, actualización y eliminación de los registros de asistencia de los estudiantes para laboratorio de clases, según los parámetros ingresados, mostrando dicha información en una tabla.

Laboratorio de clases - Estudiantes: Página que presenta la consulta, registro, registro grupal (por medio de archivo CSV donde se encuentre la información de los estudiante para laboratorio de clases,s según el formato que utiliza espol para este fin), actualización y eliminación de los registros de los estudiantes según los parámetros ingresados, mostrando dicha información en una tabla.

Laboratorio de clases - Carreras: Página que presenta la consulta, registro, actualización y eliminación de los registros de las carreras de los estudiantes para laboratorio de clases, según los parámetros ingresados (es dependiente del módulo elegido), mostrando dicha información en una tabla.

Laboratorio de clases - Módulos: Página que presenta la consulta, registro, actualización y eliminación de los módulos según los parámetros ingresados, mostrando dicha información en una tabla.

Laboratorio de clases - Inventario: Página que presenta la consulta, registro, actualización y eliminación de los registros de inventario según los parámetros ingresados (cada usuario tendrá un registro de inventario con nombre ”General” donde se registrarán específicamente los ítems detectados periódicamente por la antena RFID), mostrando dicha información en una tabla.

Laboratorio de clases - Items: Página que presenta la consulta, registro, actualización y eliminación de los ítems según los parámetros ingresados (los ítems correspondientes al inventario ”General” variarán su estado dependiendo de si fueron o no detectados por la antena RFID), mostrando dicha información en una tabla.

Laboratorio de clases - Préstamo de ítems: Página que presenta la consulta, registro, actualización y eliminación de los registros de préstamo de ítems según los parámetros ingresados (los ítems correspondientes al inventario ”General” no podrán ser utilizados al momento de registrar un préstamo, pues su lógica se basa en la detección periódica antena y no en el ingreso por parte del usuario), mostrando dicha información en una tabla.

Laboratorio de clases - Módulo actual: Menú desplegable donde el usuario podrá elegir, de los módulos que registra, uno al que quiera tener seleccionado para el registro de asistencia por parte de los propios estudiantes, habilitando así el módulo (curso) que se encuentra teniendo clases en ese momento y permitiendo asi el respectivo registro de asistencia sólo de los estudiantes que corresponden a ese módulo elegido.

Laboratorio abierto - Asistencia: Página que presenta la consulta, registro, actualización y eliminación de los registros de asistencia de los estudiantes para laboratorio abierto, según los parámetros ingresados, mostrando dicha información en una tabla.

Laboratorio abierto - Estudiantes: Página que presenta la consulta, registro, registro grupal (por medio de archivo CSV donde se encuentre la información de los estudiantes según el formato que utiliza espol para este fin), actualización y eliminación de los registros de los estudiantes para laboratorio abierto, según los parámetros ingresados, mostrando dicha información en una tabla.

Laboratorio abierto - Carreras: Página que presenta la consulta, registro, actualización y eliminación de los registros de las carreras de los estudiantes para laboratorio abierto, según los parámetros ingresados (es dependiente del módulo elegido), mostrando dicha información en una tabla.

Laboratorio abierto - Inventario: Página que presenta la consulta, registro, actualización y eliminación de los registros de inventario según los parámetros ingresados (cada usuario tendrá un registro de inventario con nombre ”General” donde se registrarán específicamente los ítems detectados periódicamente por la antena RFID), mostrando dicha información en una tabla.

Laboratorio abierto - Items: Página que presenta la consulta, registro, actualización y eliminación de los ítems según los parámetros ingresados (los ítems correspondientes al inventario ”General” variarán su estado dependiendo de si fueron o no detectados por la antena RFID), mostrando dicha información en una tabla.

Laboratorio abierto - Préstamo de ítems: Página que presenta la consulta, registro, actualización y eliminación de los registros de préstamo de ítems según los parámetros ingresados (los ítems correspondientes al inventario ”General” no podrán ser utilizados al momento de registrar un préstamo, pues su lógica se basa en la detección periódica antena y no en el ingreso por parte del usuario), mostrando dicha información en una tabla.

Laboratorio de clases/abierto - Cuenta: Menú desplegable donde el usuario podrá elegir entre dos opciones: editar su perfil, lo que lo llevará a la página ”Perfil”, y cerrar sesión.