

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**DESARROLLO DE SISTEMA PARA GESTIÓN DE LOS
LABORATORIOS DE INFORMÁTICA - ESFOT**

DESARROLLO DE UN FRONTEND

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN DESARROLLO DE SOFTWARE**

POLETH ALEJANDRA ARIAS PROCEL

DIRECTOR: ING. BYRON LOARTE

DMQ, febrero 2023

CERTIFICACIONES

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



Poleth Alejandra Arias Procel

poleth.arias@epn.edu.ec

Certifico que el presente trabajo de integración curricular fue desarrollado por Poleth Alejandra Arias Procel, bajo mi supervisión.



Ing. BYRON LOARTE, MSc.

DIRECTOR

byron.loarteb@epn.edu.ec

DECLARACIÓN DE AUTORÍA

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

Poleth Alejandra Arias Procel

DEDICATORIA

Dedico este proyecto a mi Dios, por haberme ayudado a lo largo de todo este proceso con sabiduría para seguir hacia adelante en mi vida y en mi educación, a mis padres por haberme apoyado y animado en los momentos más difíciles a lo largo de este proceso, por siempre estar a mi lado brindándome sus consejos para hacer de mi vida una mejor persona, a mi abuelita, a mis tíos y mis primos por sus palabras y compañía, a mis mejores amigos (Christian y Matthew) y compañeros que de una u otra manera han contribuido para el logro de mis objetivos.

Poleth Alejandra Arias Procel

AGRADECIMIENTO

Agradezco a mis padres, abuelitas, primos, tíos por ser parte de mi vida, siempre apoyarme y creer en mí siempre.

Agradezco a la Escuela Politécnica Nacional por haberme permitido formar parte de su familia y estudiar una carrera universitaria, a cada uno de los profesores que forman parte de la Escuela de Formación de Tecnólogos, quienes con su paciencia y sabiduría han compartido sus conocimientos a lo largo de mi etapa universitaria y con ello conseguir este logro y culminar mi proyecto de titulación.

Poleth Alejandra Arias Procel

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco Teórico	4
2 METODOLOGÍA.....	6
2.1 Metodología de Desarrollo	6
Roles	6
Artefactos	7
2.2 Diseño de interfaces.....	11
Herramienta utilizada para el diseño.....	11
2.3 Diseño de la arquitectura	12
Patrón arquitectónico.....	12
2.4 Herramientas de desarrollo	13
3 RESULTADOS	15
3.1 <i>Sprint 0</i> . Configuración del ambiente de desarrollo.....	15
Descripción de requerimientos.....	15
Estructura del proyecto <i>frontend</i>	18
Roles de usuarios	19
3.2 <i>Sprint 1</i> . Interfaces que han sido implementadas para el usuario administrador	20
Consumir <i>endpoints</i> para cambiar contraseña, iniciar y cerrar sesión.....	20
Consumir <i>endpoints</i> para observar y modificar perfil de usuario	22
Consumir <i>endpoints</i> para gestionar pasantes	23
Consumir <i>endpoints</i> para gestionar inventarios	24
Consumir <i>endpoints</i> para observar reporte de inventarios	25
3.3 <i>Sprint 2</i> . Interfaces que han sido implementadas para el usuario administrativo	26
Consumir <i>endpoints</i> para registrarse	26
Consumir <i>endpoints</i> para solicitar <i>tickets</i> de asistencia	27
3.4 <i>Sprint 3</i> . Interfaces que han sido implementadas para el usuario con perfil docente	28

Consumir <i>endpoints</i> para visualizar respuestas y enviar comentarios y/o sugerencias	28
Consumir <i>endpoints</i> para solicitar <i>tickets</i> de asistencia	30
Consumir <i>endpoints</i> para solicitar reservas de laboratorios	31
3.5 <i>Sprint</i> 4. Interfaces que han sido implementadas para el usuario pasante	32
Consumir <i>endpoints</i> para gestionar <i>tickets</i> de asistencia.....	32
Consumir <i>endpoints</i> para gestionar reservas de laboratorios.....	33
3.6 <i>Sprint</i> 5. Ejecución de pruebas y despliegue a un entorno de producción	34
Ejecutar y detallar el resultado de <i>test</i> unitarios.....	35
Ejecutar y detallar el resultado de <i>test</i> de compatibilidad	36
Ejecutar y detallar el resultado de <i>test</i> de aceptación	36
Despliegue del <i>frontend</i>	37
4 CONCLUSIONES	39
5 RECOMENDACIONES.....	40
6 REFERENCIAS BIBLIOGRÁFICAS	¡Error! Marcador no definido.
7 ANEXOS.....	44
ANEXO I	45
ANEXO II	46
ANEXO III	104
ANEXO IV	105

RESUMEN

La Escuela de Formación de Tecnólogos de la Escuela Politécnica Nacional no cuenta con un sistema de gestión y administración para los laboratorios de informática ya que actualmente el jefe de laboratorio y pasantes tienen que acudir a mecanismos tradicionales como uso de medios impresos para realizar inventarios, solicitud de tickets de asistencia, agendamiento de laboratorios por parte de los docentes, entre otros. Lo que ocasionaba que existan cruces de horarios, retraso en la atención de un ticket de asistencia y malestar de por parte de docentes, personal administrativo y estudiantes.

Para dar una solución viable a los innumerables problemas que se presentaban y con el objetivo de ofrecer una mejora en la gestión y administración de laboratorios, en el presente Trabajo de Integración Curricular se ha desarrollado un *frontend* denominado "BOTICS", el cual permite que el jefe de laboratorio y pasantes puedan realizar un inventario de forma adecuada de todos los equipos tecnológicos y que a su vez los docentes puedan agendar un determinado laboratorio o realizar sugerencias o comentarios. Logrando de esta manera que las peticiones por parte del personal docente sean atendidas en el menor tiempo posible y que la información sea gestionada de una forma adecuada y centralizada.

El presente proyecto de Integración Curricular se encuentra organizado de la siguiente manera: primero, se establece la problemática que existía en relación a los laboratorios de informática de la ESFOT, en base a ello se determinan los objetivos, alcance y el marco teórico respectivamente. Después, se especifica la implementación de la metodología utilizada, la cual es *Scrum*, así como la realización de diferentes prototipos, el uso de herramientas, librerías, entre otros. Posteriormente, se detalla cada una de las tareas de los módulos, así como los resultados que se han obtenido en cada *Sprint*. Como último punto, se especifica las conclusiones y recomendaciones que se han obtenido a lo largo del desarrollo del proyecto de Integración Curricular.

PALABRAS CLAVE: *Frontend*, *Scrum*, Inventario, Angular, Laboratorios, Informática

ABSTRACT

The Technological Training School of the Escuela Politécnica Nacional did not have a management and administration system for the computer laboratories since currently the head of the laboratory and interns had to resort to traditional mechanisms such as the use of printed media to carry out inventories, request of assistance tickets, scheduling of laboratories by teachers, among others. What caused there to be time crossings, delay in the attention of an assistance ticket and discomfort on the part by teachers, administrative staff and students.

In order to provide a viable solution to the innumerable problems that arose and with the aim of offering an improvement in the management and administration of laboratories, in this Curricular Integration Work a frontend called "BOTICS" has been developed, which allows the Laboratory chief and interns can make an adequate inventory of all the technological equipment and that, in turn, teachers can schedule a certain laboratory or make suggestions or comments. Achieving in this way that the requests by the teaching staff are attended in the shortest possible time and that the information is managed in an adequate and centralized way.

This Curricular Integration project is organized as follows: first, the problems that existed in relation to the ESFOT computer laboratories are established, based on this, the objectives, scope and theoretical framework are determined respectively. Afterwards, the implementation of the methodology used is specified, which is Scrum, as well as the realization of different prototypes, the use of tools, libraries, among others. Subsequently, each of the module tasks is detailed, as well as the results that have been obtained in each Sprint. As a last point, the conclusions and recommendations that have been obtained throughout the development of the Curricular Integration project are specified.

KEYWORDS: Frontend, Scrum, Inventory, Angular, Laboratories, Computing

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Hoy en día, contar con un control de inventarios es fundamental ya que permite tener un control correcto y ordenado sobre la existencia de los bienes que poseen los negocios y empresas. Además, sin un inventario pertinente, en caso de existir alguna anomalía como la sustracción de algún bien puede ocasionar graves problemas a la empresa ya que no se logra realizar una adecuada toma de decisiones a futuro [1].

En las instituciones de educación es indispensable disponer de un sistema de gestión y administración tanto de laboratorios como de equipos tecnológicos ya que optimiza tiempo y carga para el personal administrativo, es decir, el mecanismo tradicional de realizar los inventarios es reemplazado por un sistema *web* o *móvil* siendo así que se adquiere un mayor control tanto del estado de los equipos como del estado de los laboratorios e incluso se puede adquirir una mejor visión con respecto a las acciones que se realiza en cada laboratorio

En la actualidad, la ESFOT de la Escuela Politécnica Nacional no cuenta con un sistema de gestión y administración de laboratorios, ya que emplean un sistema tradicional para realizar inventarios y agendar reserva de laboratorios, lo cual provoca algunas dificultades en lo que respecta al control de la existencia de equipos y en la reserva de laboratorios cuando son requeridos por algún docente ya que en ocasiones los administradores o pasantes no se percatan de que dicho laboratorio se encuentra ocupado en el horario solicitado por parte de otro docente o en el peor de los casos el docente no está al tanto de todos los programas informáticos que tiene cada laboratorio. Así mismo, cuando existe algún inconveniente en algún laboratorio los problemas no son resueltos de forma inmediata o la persona que solicita la asistencia técnica no tiene una retroalimentación adecuada por parte del técnico que atiende su petición [2].

Por otro lado, los equipos de los laboratorios con el pasar del tiempo necesitan un mantenimiento preventivo y correctivo o de ser el caso ser reemplazados por unos nuevos, ya que al ser utilizados por todas las carreras de la ESFOT semestralmente se deterioran y es fundamental llevar control de dichos mantenimientos y de los equipos.

Considerando la problemática que actualmente presenta los laboratorios de informática de la ESFOT se ha desarrollado un *frontend* como parte de este Trabajo de Integración Curricular para la gestión y administración de laboratorios de informática denominado “BOTICS”, cuyo objetivo es permitir un correcto control de los equipos tecnológicos y agendamientos de laboratorios. Además, el *frontend* dispone de una interfaz amigable para todos los usuarios que haga uso del mismo y 100% adaptable a los distintos dispositivos electrónicos.

1.1 Objetivo general

Desarrollar un *frontend* para la gestión de los laboratorios de informática - ESFOT

1.2 Objetivos específicos

1. Definir los requerimientos para el *frontend* a nivel funcional.
2. Diseñar los prototipos del *frontend* según los requisitos.
3. Codificar los respectivos módulos del *frontend*.
4. Comprobar el correcto funcionamiento del *frontend* a través de una serie de pruebas.
5. Desplegar a producción el *frontend* una vez que se han aprobado todas las pruebas.

1.3 Alcance

Actualmente, los sistemas de control y administración tanto de inventarios como de laboratorios han tenido una gran acogida en la sociedad. Esto se debe a que hoy en día varias empresas e instituciones de educación prefieren utilizar un sistema digital para llevar un adecuado control de la existencia de sus productos. Por tanto, para la ESFOT contar con un sistema de gestión y administración de laboratorios es indispensable para el personal y las actividades que realizan diariamente [2].

La importancia de contar con un *frontend* es vital para que de esta manera los usuarios puedan interactuar con la parte visual haciendo que el diseño sea intuitivo y atractivo, por ejemplo, al momento de que un usuario se dirija a uno de los módulos no se va a perder ya que toda la información, tablas e imágenes se presentan de forma ordenada y de clara comprensión [36]. Por otra parte, para lograr una correcta funcionalidad se ha empleado diferentes tecnologías como HTML, CSS, JavaScript y siendo la principal el *Framework* Angular.

El presente trabajo de Integración Curricular dispone de un componente *frontend*, cuyo objetivo es gestionar y administrar todos los laboratorios de informática de la ESFOT. Utilizando para ello, un conjunto de tecnologías descritas anteriormente, un patrón arquitectónico que facilite la estructuración de código fuente, archivos y directorios, una metodología de desarrollo ágil para el seguimiento y cumplimiento de cada una de las iteraciones y una etapa de pruebas que garanticen la calidad del producto final y la aprobación del dueño del producto. Por último, existen 4 perfiles de

usuario que pueden visualizar los módulos asignados según el nivel de permiso que se ha establecido, como se muestra a continuación:

Perfiles que se establecen:

- Administrador
- Administrativo
- Docente
- Pasante

En el *frontend* el perfil administrador consume varios *endpoints* que le permita:

- Modificar contraseña, iniciar sesión, cerrar sesión.
- Visualizar y editar el perfil de usuario.
- Gestionar pasantes.
- Gestionar inventarios.
- Visualizar reporte de inventarios.

En el *frontend* el perfil administrativo consume varios *endpoints* que le permita:

- Modificar contraseña, iniciar sesión, cerrar sesión.
- Registrarse.
- Visualizar y editar el perfil de usuario.
- Gestionar comentarios y/o sugerencias.
- Solicitar *tickets* de asistencia.
- Enviar comentarios y/o sugerencias.

En el *frontend* el perfil docente consume varios *endpoints* que le permita:

- Modificar contraseña, iniciar sesión, cerrar sesión.
- Registrarse.
- Visualizar y editar el perfil de usuario.
- Solicitar *tickets* de asistencia.
- Enviar comentarios y/o sugerencias.
- Solicitar reservas.

En el *frontend* el perfil pasante consume varios *endpoints* para:

- Modificar contraseña, iniciar sesión, cerrar sesión.
- Visualizar y editar el perfil de usuario.
- Gestionar *tickets* de asistencia.
- Gestionar reservas.

1.4 Marco Teórico

Hoy en día, los sistemas *software* son recursos altamente importantes para las organizaciones, empresas y usuarios en general, ya que son empleados en diversas áreas como, por ejemplo: finanzas, educación, arquitectura, medicina, etc. Es por esta razón que el desarrollo de un *software* es muy necesario ya que ofrece soluciones efectivas y sustentables a los diferentes tipos de usuarios y empresas [3].

Según Roger Pressman en su libro “La Ingeniería del *Software*” hace referencia a la implementación de enfoques sistemáticos, disciplinados y cuantificables al desarrollo de un producto *software* [4].

El *frontend* también conocido como aplicación del lado del cliente, tienen como principal objetivo mostrar al usuario una interfaz amigable para que pueda fácilmente interactuar con la información y contenido multimedia. Además, al momento de codificar el *frontend* también tiene su propia lógica interna al igual que las aplicaciones del lado del servidor [5].

Un *framework* es considerado como el andamiaje para el desarrollo de una aplicación, facilitando así la reutilización de componentes presentes en la estructura, es por esta razón, que actualmente existen una variedad de *frameworks* para diversos lenguajes de programación. Lo único que hay que hacer es seleccionar el que sea más amigable con el desarrollador [6].

Responsive web design, se refiere a la forma de presentar el contenido en diversas pantallas de los dispositivos electrónicos, de tal manera que se mejore la UX al momento de navegar por cualquier sección de la aplicación, página o sitio *web* [3].

Una interfaz de usuario más conocida como UI (*User Interface*) engloba términos de arquitectura de información, patrones y diferentes elementos visuales los mismos que permiten al usuario final interactuar de manera eficiente con el contenido e información dentro de una aplicación, página o sitio *web* [7].

La experiencia de usuario más conocida como UX (*User Experience*) hace referencia a cómo se sienten las personas acerca de un producto o servicio al momento de interactuar y hacer uso del mismo. En otras palabras, la experiencia de usuario trabaja para coordinar todos los elementos de la aplicación con la finalidad de permitir la mejor interacción posible por parte de los usuarios [8].

Angular, es un *Framework* que permite desarrollar aplicaciones *frontend* empleando HTML y JavaScript. Dicho *Framework*, está orientado a objetos ya que trabaja con clases y facilita el uso del patrón MVC (Modelo-Vista-Controlador) [9].

HTML, es un conjunto de etiquetas las cuales ayudan a definir el texto y otros elementos que son parte de una página *web*, como, imágenes, texto, videos, etc [10].

CSS, es el lenguaje que se encarga de manejar el diseño y la presentación de páginas *web*. El nombre de cascada, es debido a que es posible tener varias hojas y una de ellas con las propiedades heredadas. En CSS se puede crear formatos específicos para comunicar fácilmente las ideas y que visualmente sea más agradable para los usuarios [11].

JavaScript (JS), es un lenguaje de programación de *scripts*, es decir, secuencia de comandos con el cual se puede añadir funcionalidades interactivas y contenidos dinámicos en el desarrollo del *frontend*. Por otro lado, es intuitivo y fácil de aprender [12].

Las librerías son conocidas como un conjunto de archivos de código las cuales son utilizadas con el objetivo de facilitar la programación. Además, las librerías proporcionan funcionalidades comunes las cuales ya han sido solucionadas por otros programadores [13].

Interfaz de programación de aplicaciones o más conocida como API, hace referencia al mecanismo que permite que dos componentes de *software* puedan comunicarse entre sí, mediante solicitudes y respuestas. Las API funcionan de manera diferente, dependiendo el motivo de su creación [14].

El protocolo de transferencia de hipertexto o también conocido como HTTP, hace referencia a cómo se envían los mensajes y las acciones que deben tomar los servidores *web* en respuesta a varias peticiones [15].

Cookies, son fragmentos de datos los cuales son creados por el servidor y enviados al cliente con la finalidad de que exista comunicación. Los *tokens*, hacen referencia a credenciales firmadas codificadas en una larga cadena de caracteres creada por el servidor. La principal diferencia entre *cookie* y *token* es que los *tokens* no tienen estado, es decir, el servidor no necesita mantener un registro del *token* ya que cada *token* es autónomo y contiene la información necesaria para la verificación en el servidor [16].

2 METODOLOGÍA

Un estudio de casos es un tipo de indagación y su principal finalidad es realizar una investigación eficaz y conseguir los resultados esperados. En otros términos, es una estrategia la cual se sustenta en la observación y análisis del caso estudiado y así llegar a las conclusiones y buscar el medio para sustentar que dicha conclusión es válida [17].

El actual proyecto implementa un estudio de casos debido a que se enfoca en un análisis definido, en este caso los laboratorios de informática de la ESFOT, ya que no cuentan con un sistema de gestión y administración y en consecuencia a lo citado anteriormente, se ha desarrollado un sistema de gestión y administración, para que el personal administrativo pueda realizar inventarios de los equipos tecnológicos y a su vez gestionar la reserva de laboratorios, entre otras actividades.

2.1 Metodología de Desarrollo

Es una agrupación de técnicas y métodos que son empleadas para diseñar una solución de *software*. Además, trabajar con una metodología de desarrollo es fundamental para una correcta organización, cabe mencionar que los factores tienen que estar ordenados y se debe tener claro lo que se va a realizar. Actualmente, existe varias metodologías de desarrollo como, por ejemplo: cascada, incremental, espiral, etc. [18].

Una metodología ágil tiene como objetivo principal hacer que el proceso de entrega de un producto *software* sea mucho más rápido y a su vez eficiente [19]. Hoy en día, las metodologías ágiles son muy utilizadas ya que permiten acoplarse a las formas de trabajo según las necesidades del cliente [20]. Por tanto, es fundamental que la comunicación sea asertiva entre los integrantes del equipo de trabajo y el cliente ya que los requerimientos del cliente pueden cambiar durante el transcurso del proyecto [21]. Es por esta razón, que al implementar una metodología ágil como es *Scrum*, garantiza un proyecto de calidad y con un nivel alto de satisfacción por parte del dueño del producto.

Por lo citado previamente, se ha desarrollado un *frontend* bajo la metodología *Scrum* aplicando las buenas prácticas, es decir, trabajar colaborativamente, pequeñas entregas funcionales y lograr un producto de calidad [22]. Posteriormente, se presenta la implementación de la metodología de una forma mucho más detallada.

Roles

La metodología ágil *Scrum* está basada en roles, es decir, está compuesta por un equipo de trabajo, más conocido como “Equipo *Scrum*”. Cada integrante del grupo cumple con un rol, por ende, tiene diferentes responsabilidades y a su vez se comprometen a cumplir con las

tareas asignadas con el objetivo de cumplir el proyecto en los tiempos establecidos [22]. Posteriormente, se presentan los distintos roles que han sido necesarios para todo el Trabajo de Integración Curricular.

Product Owner

Es quien se responsabiliza de optimizar el valor del producto y dado que es el propietario del producto, también revisa los resultados y avances del equipo de desarrollo [22]. En este caso, en la **TABLA I** se muestra la persona asignada para cumplir este rol.

Scrum Master

Tiene dos funciones principales a su cargo, por una parte, se encarga de gestionar y eliminar impedimentos que aparecen en el desarrollo del proyecto y por otra siempre trata de motivar al equipo de trabajo, mantiene una buena comunicación con los integrantes de equipo, mantiene reuniones semanales y es el líder del proyecto [22]. En este caso, la respectiva **TABLA I** presenta la persona asignada.

Development Team

Es un equipo de trabajo, por lo general está conformado por un grupo de tres a nueve desarrolladores, los mismos que se encargan de desarrollar el producto y de auto organizarse para así lograr un aumento significativo en cada iteración del producto *software* [22]. En este caso, la **TABLA I** muestra la persona asignada para cumplir este rol.

TABLA I: Equipo de trabajo con sus respectivos roles.

NOMBRE	ROL
Ing. Byron Loarte	<i>Product Owner</i>
Ing. Byron Loarte	<i>Scrum Master</i>
Poleth Alejandra Arias Procel	<i>Development Team</i>

Artefactos

En *Scrum* los artefactos permiten organizar la información de una manera mucho más detallada ya que esto va a permitir ver el avance el proyecto y las acciones que se deben realizar en el caso de existir algún retraso por cualquier eventualidad [23]. En ese sentido, los

tres artefactos más importantes que se han establecido son: Recopilación de requerimientos, *Product Backlog* y *Sprint Backlog* los cuales se detallan en los siguientes puntos.

Recopilación de Requerimientos

La recopilación de requerimientos ayuda a determinar las necesidades del proyecto, estimar el tiempo, alcance del proyecto, poder planificar adecuadamente los entregables y lograr con ello el cumplimiento de los objetivos del proyecto [24]. Mediante varias reuniones con el *product owner* se ha logrado recopilar los requerimientos funcionales del proyecto, los cuales se han colocado en la **TABLA II** mientras que en el **ANEXO II** se encuentra la tabla completa.

TABLA II: Requerimientos recopilados.

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DEL SISTEMA	ID - RR	ENUNCIADO DEL ÍTEM
<i>Frontend</i>	RR002	Como usuario docente y administrativo necesitan consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Registrarse

Historias de Usuario

Es una tarjeta en la cual permite explicar de forma detalla las funcionalidades que debe realizar el *software*; considerada la unidad de trabajo más pequeña en un marco ágil [25]. Además, son sencillas y fáciles de recordar, las mismas siguen un patrón y los elementos utilizados en la elaboración de las historias de usuario son:

- Identificador (ID).
- Nombre Historia.
- Prioridad en negocio.
- Riesgo de desarrollo.
- Iteración.
- Responsable.
- Descripción.
- Observación.

La **TABLA III** ejemplifica una historia de usuario del *frontend* la misma ha sido realizada bajo los requerimientos funcionales que se han recopilado, mientras que en el **ANEXO II** se encuentran las demás historias de usuario.

TABLA III: Historia de usuario 01.

HISTORIA DE USUARIO	
Identificador: HU001	Identificador: HU001
Nombre historia: Visualizar página informativa.	
Prioridad en negocio: Alta	Prioridad en negocio: Alta
Iteración asignada: 1	
Responsable: Poleth Arias	
<p>Descripción: El <i>frontend</i> por medio del perfil administrador, pasante, docente y administrativo necesitan visualizar una página informativa, en la cual se mostrarán las siguientes vistas:</p> <ul style="list-style-type: none"> • Inicio • Inicio de sesión/Registrarse. 	
<p>Observación: Los usuarios administrador, pasante, docente y administrativo, no necesitan iniciar sesión para visualizar la página informativa, es decir, todos los perfiles pueden visualizar la página informativa.</p>	

Product Backlog

Se refiere a una lista ordenada de tareas que se pretenden realizar durante el desarrollo del proyecto [26]. Dentro del *Product Backlog* aparecen las actividades principales y secundarias, es decir, se encuentran de acuerdo a la prioridad en el desarrollo siendo así, que la **TABLA IV** presenta el formato realizado para la elaboración del *Product Backlog*, donde se puede observar la iteración, estado y prioridad de las diferentes tareas las cuales se han obtenido de las historias de usuario, mientras que en el **ANEXO II** está la tabla completa.

TABLA IV: *Product Backlog*.

ELABORACIÓN DEL PRODUCT BACKLOG				
ID – HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU001	Visualizar página informativa	1	Finalizada	Alta

HU002	Registrarse.	1	Finalizada	Media
-------	--------------	---	------------	-------

Sprint Backlog

Es una tabla donde se encuentran las tareas identificadas por el equipo de desarrollo y cada tarea debe ser completada durante cada iteración dependiendo de la prioridad, de tal manera que permita observar el detalle de cada una de las tareas [27]. Muestra de ello, la **TABLA V** presenta la plantilla que se ha realizado para la elaboración del *Sprint Backlog*, mientras que la tabla completa se encuentra en el **ANEXO II**.

TABLA V: *Sprint Backlog*.

ELABORACIÓN DE SPRINT BACKLOG						
ID – SB	NOMBRE	MÓDULO	ID-HU	HISTORIA DE USUARIO	TAREA	TIEMPO ESTIMADO
SB001	Diseño e implementación de la interfaz de los módulos: registro, inicio de sesión y modificación del perfil.	Módulo de registro	HU002	Registrarse	<ul style="list-style-type: none"> • Diseño e implementación del registro para el usuario administrativo y docente. • Validación de los campos requeridos. 	40H
		Módulo inicio de sesión.	HU003	Iniciar sesión, cerrar sesión y modificar.	<ul style="list-style-type: none"> • Diseño e implementación de las interfaces de iniciar sesión, cerrar sesión y modificar contraseña para todos los usuarios. • Validación de los campos requeridos. 	
		Módulo de perfil de usuario.	HU004	Modificar perfil de usuario	<ul style="list-style-type: none"> • Diseño e implementación de la interfaz para visualizar y modificar el perfil de usuario de 	

					todos los perfiles de los usuarios. <ul style="list-style-type: none"> Validación de los campos requeridos. 	
--	--	--	--	--	--	--

2.2 Diseño de interfaces

Se refiere a un bosquejo que se elabora en base a los requisitos del producto inicial y a la vez permite que los usuarios accedan al contenido mediante el uso de elementos gráficos de una forma sencilla [28]. Posteriormente, se muestra la herramienta que se ha seleccionado para el diseño.

Herramienta utilizada para el diseño

Figma se ha empleado para el bosquejo y diseño de las interfaces, ya que es una herramienta colaborativa y ampliamente utilizada para el diseño de interfaces de usuario [29]. Gracias al uso de esta herramienta se ha logrado diseñar el prototipo del primer módulo donde se muestra la información general de los laboratorios de informática, dicho prototipo contiene los siguientes elementos: *header*, *footer*, *buttons*, *components*, entre otros elementos como se ilustra en la **Fig. 1**, los prototipos restantes se encuentran en el **ANEXO II**.



Fig. 1: Prototipo para indicar la información.

2.3 Diseño de la arquitectura

Se refiere a cada uno de los componentes internos de un *software* y su respectiva interacción [30]. Posteriormente, se presenta el patrón arquitectónico implementado en el desarrollo *frontend*.

Patrón arquitectónico

MVC también conocido como Modelo, Vista y Controlador, está enfocado en separar la lógica de la interfaz. De tal manera que, se obtiene una estructura limpia, incrementando así atributos de calidad en el código fuente [31]. Posteriormente, se define el detalle de las capas del modelo MVC:

- **Modelo:** esta capa se encarga de manipular la información.
- **Vista:** esta capa hace referencia a la representación visual, es decir, se encuentra los elementos que el usuario puede visualizar y a la vez interactuar.
- **Controlador:** esta capa hace el papel de intermediario entre las capas del Modelo y Vista. Por otro lado, tiene la responsabilidad de actualizar el Modelo cuando el usuario manipula la Vista.

Posteriormente, en la **Fig. 2** se observa el patrón que se ha empleado para la codificación del *frontend*, lo cual facilita que exista una compatibilidad alta con las demás herramientas y librerías.

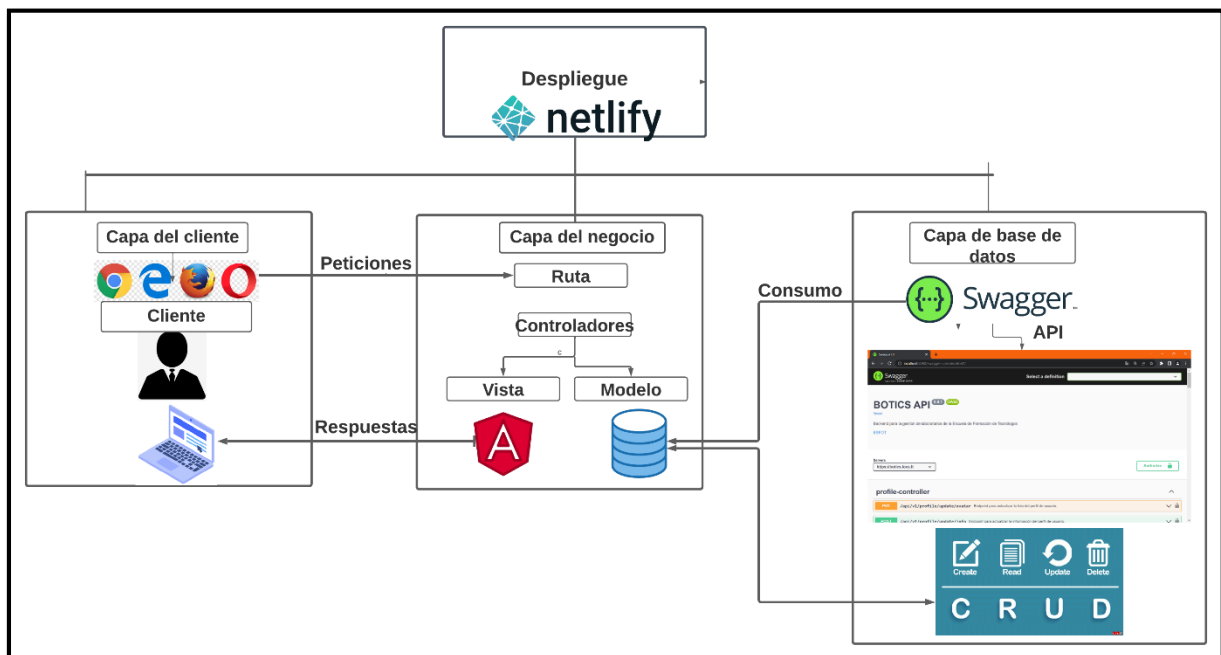


Fig. 2: Patrón que se ha implementado para la codificación del *frontend*.

2.4 Herramientas de desarrollo

Las herramientas son indispensables para el desarrollo del *frontend*, muestra de ello la **TABLA VI** lista a cada una de las principales herramientas que se han empleado en base a la complejidad del proyecto.

TABLA VI: Herramientas para la codificación del *frontend*.

HERRAMIENTA	JUSTIFICACIÓN
<i>Angular</i>	Este <i>framework</i> permite la creación de sistemas <i>web</i> de una sola página a través del uso de <i>TypeScript</i> [32].
<i>Angular CLI</i>	El intérprete de línea de comando (CLI) permite trabajar cómodamente en el desarrollo de un proyecto mediante la ejecución de comandos para los componentes que se requiera [33].
<i>Angular Material</i>	Módulos que son creados principalmente para la parte visual y que fácilmente puedan ser utilizados en Angular [34].

Librerías

Para el desarrollo *frontend* se ha utilizado diferentes librerías las cuales ofrecen varias ventajas para que en el proyecto no existan errores al momento de compilar, ejecutar pruebas y desplegar a producción. Sin embargo, a diferencia de un *framework* una librería no aporta la estructura de cómo se debe realizar, si no es un complemento extra [35]. Posteriormente, la **TABLA VII** presenta las librerías empleadas y una descripción.

TABLA VII: Librerías necesarias para la codificación del *frontend*.

LIBRERÍA	DESCRIPCIÓN
<i>material-app</i>	Librería para el diseño de interfaces del <i>frontend</i> [36].

<i>mat-icon</i>	Esta librería fue empleada para colocar diferentes íconos en la aplicación web por el lado del <i>frontend</i> , ya que dicha librería facilita el uso de íconos [37] .
<i>bootstrap</i>	Esta librería permite añadir a los proyectos una serie de estilos, para lo cual se ha empleado en algunos módulos del <i>frontend</i> del sistema web BOTICS.

3 RESULTADOS

En este punto, se muestran los resultados que se han alcanzado en cada uno de los módulos y elementos visibles como parte del componente *frontend*. También, se muestra las pruebas que se han realizado y el despliegue a un entorno de producción. Los resultados se presentan en *Sprints*, los mismos que han sido determinados con antelación, en el **ANEXO II** se puede encontrar el detalle de los *Sprints*.

3.1 *Sprint 0*. Configuración del ambiente de desarrollo

Posteriormente, se presentan las tareas del *Sprint 0* las cuales han sido definidas en el *Sprint Backlog*:

- Definición de requerimientos.
- Estructura del proyecto *frontend*.
- Roles de usuarios.

Descripción de requerimientos

Visualizar página informativa

Dentro del *frontend* el administrador, administrativo, docente y pasante pueden visualizar una página informativa con tres secciones: Inicio, Inicio de sesión y registrarse. Es importante mencionar que para visualizar la página informativa no hace falta que los usuarios inicien sesión en el *frontend*.

Consumir *endpoints* para el registro de usuarios

Dentro del *frontend* los usuarios de tipo administrativo y docente pueden registrarse para lo cual es necesario que la parte del *frontend* consuma los *endpoints* implementados por parte del componente *backend*.

Consumir *endpoints* para el inicio de sesión, cierre de sesión y modificar contraseña

Dentro del *frontend* los usuarios con perfil administrador, administrativo, docente y pasante pueden iniciar sesión mediante un usuario y contraseña para lo cual es necesario que el *frontend* consuma el *endpoint* generado por parte del *backend*. En ese sentido, para que los usuarios tengan la posibilidad de cerrar sesión es necesario que el *frontend* consuma el *endpoint* implementado por parte del *backend* y para modificar contraseña el *frontend* tiene que consumir varios *endpoints* generados por el componente *backend*.

Consumir *endpoints* para modificar el perfil de usuario

Dentro del *frontend* los usuarios con perfil administrador, administrativo, docente y pasante pueden observar el perfil de usuario para lo cual el *frontend* debe consumir el *endpoint* generado por parte del componente *backend* y para la modificación del perfil de los usuarios mencionados es necesario que el *frontend* consuma el *endpoint* generado por parte del *backend*.

Consumir *endpoints* para gestionar pasantes

Dentro del *frontend* el usuario con perfil administrador puede gestionar pasantes y a su vez visualizar el listado de pasantes que se han registrado, registrar pasantes, habilitar y deshabilitar pasantes, de tal manera que sólo los pasantes que consten como habilitados puedan iniciar sesión y las acciones mencionadas. Para que el usuario de tipo administrador pueda realizar las acciones mencionadas anteriormente, es necesario que el *frontend* consuma los *endpoints* implementados por parte del componente *backend*.

Consumir *endpoints* para gestionar inventarios

El *frontend* consume varios *endpoints* generados por parte del componente *backend*, con el objetivo de que el usuario administrador pueda gestionar inventarios, visualizar el listado de laboratorios con sus respectivas características y registrar computadoras con sus respectivas características según sean necesarias.

Consumir *endpoints* para visualizar reporte de inventarios

Dentro del *frontend* el usuario con perfil administrativo puede visualizar el reporte de inventarios de los equipos tecnológicos y a su vez el historial de movimiento de cada equipo de los laboratorios, para lo cual es necesario que el *frontend* consuma varios *endpoints* generados por parte del componente *backend*.

Consumir *endpoints* para gestionar comentarios y/o sugerencias

Dentro del *frontend* el usuario con perfil administrativo puede gestionar comentarios y/o sugerencias las cuales son generadas por parte del usuario con perfil docente, la finalidad es que el usuario administrativo pueda visualizar y responder a los comentarios y/o sugerencias que han sido realizadas, para realizar las acciones mencionadas anteriormente es necesario que el *frontend* consuma varios *endpoints* implementados por parte del componente *backend*.

Consumir *endpoints* para solicitar *tickets* de asistencia

Dentro del *frontend* el usuario con perfil docente puede solicitar *tickets* de asistencia, en el cual puede redactar su petición con respecto a la problemática que presenta y luego ser

atendido por uno de los pasantes, para lo cual el *frontend* tiene que consumir el *endpoint* generado por parte del componente *backend*. Es importante mencionar que el usuario tiene que iniciar sesión para realizar la acción mencionada anteriormente.

Consumir *endpoints* para enviar comentarios y/o sugerencias

Dentro del *frontend* el usuario con perfil docente puede enviar comentarios y/o sugerencias con la finalidad de que sus observaciones con respecto a los diferentes laboratorios puedan ser atendidas en el menor tiempo posible por parte del usuario con perfil administrativo, para lo cual el *frontend* consume el *endpoint* generado por parte del componente *backend*.

Consumir *endpoints* para solicitar reservas

Dentro del *frontend* el usuario con perfil docente puede solicitar reservas de un laboratorio y a su vez su solicitud es atendida por parte del usuario de tipo pasante, para que el usuario con perfil docente pueda realizar la acción mencionada, es necesario que el *frontend* consuma el *endpoint* generado por parte del componente *backend*.

Consumir *endpoints* para gestionar *tickets* de asistencia

Dentro del *frontend* el usuario con perfil pasante puede gestionar *tickets* de asistencia. Por lo tanto, el usuario con perfil pasante puede visualizar *tickets* de asistencia solicitados por parte del usuario con perfil docente y a su vez atender las solicitudes, para lo cual es necesario que el *frontend* consuma varios *endpoints* generados por parte del componente *backend*.

Consumir *endpoints* para gestionar reservas

Por el lado del *frontend* el usuario con perfil pasante puede gestionar reservas. Por lo tanto, el usuario con perfil pasante puede visualizar las solicitudes de reservas que se han realizado por parte del usuario con perfil docente y a su vez atender las solicitudes, para lo cual es necesario que el *frontend* consuma varios *endpoints* generados por parte del componente *backend*.

Por último, en la **Fig. 3** se presenta un cuadro con todas las funcionalidades de los usuarios con perfil administrador, personal administrativo, docente y pasante para una mejor comprensión y entendimiento.

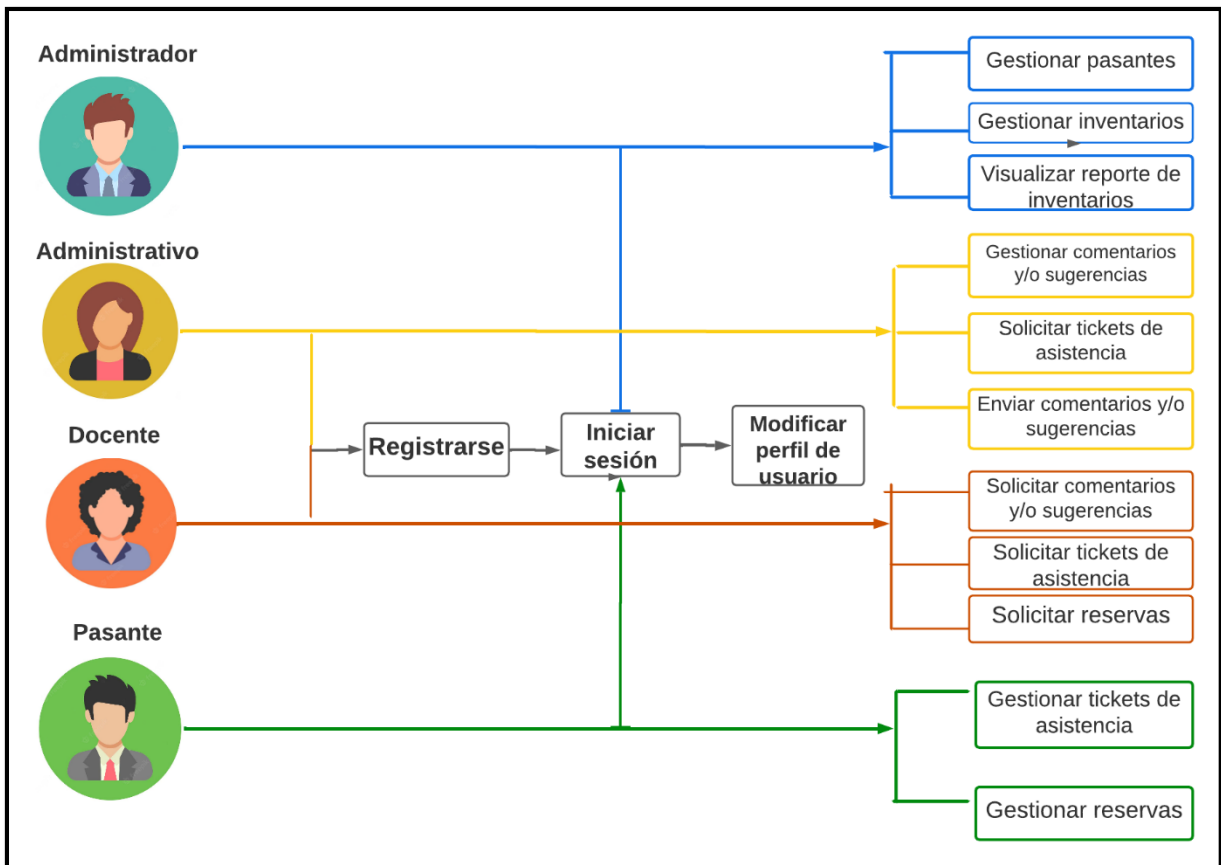


Fig. 3: Funcionalidades que disponen los usuarios dentro del *frontend*.

Estructura del proyecto *frontend*

Para el desarrollo del componente *frontend* se ha empleado *Visual Studio Code*. Este entorno de desarrollo contribuye con la estructura de diferentes módulos, archivos, *scripts*, archivos de configuración y directorios que han sido indispensables en el desarrollo del *frontend*. Posteriormente, en la **Fig. 4** se puede observar la organización del proyecto *frontend*.

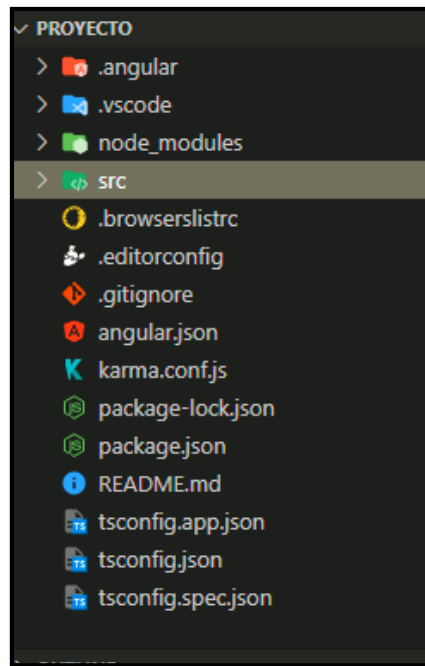


Fig. 4: Estructura de directorios y archivos para el proyecto *frontend*.

Roles de usuarios

Posteriormente, la Fig. 5 presenta los roles del *frontend*, los cuales son: administrador, administrativo, docente y pasante para una mejor comprensión y entendimiento.

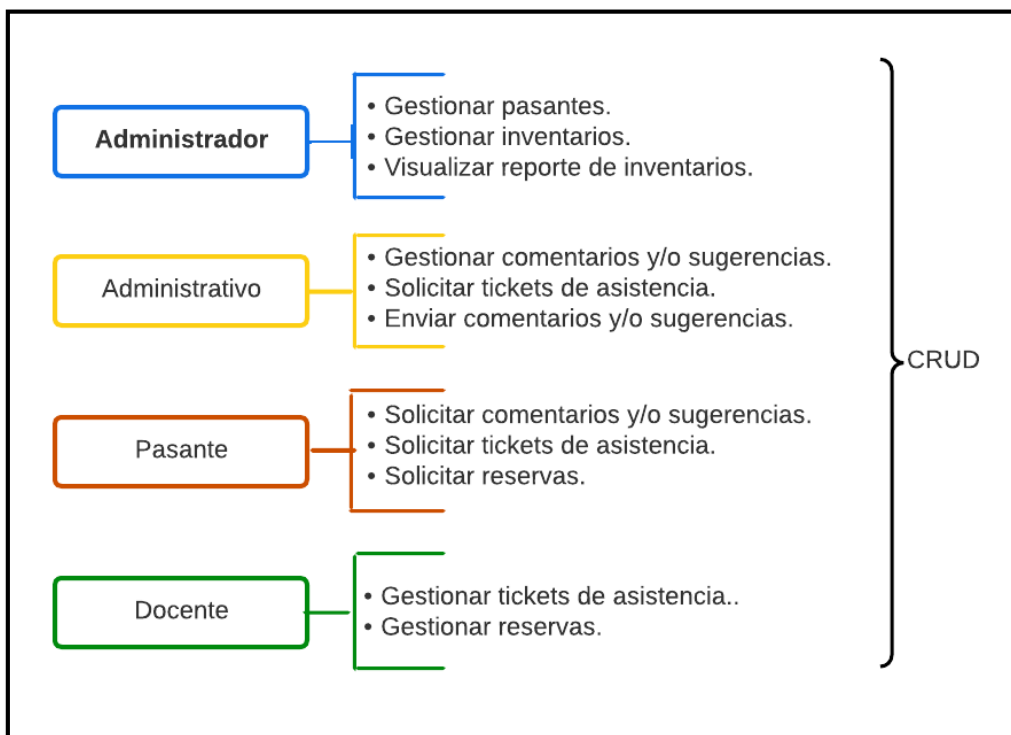


Fig. 5: Roles de usuario del *frontend*.

3.2 *Sprint 1*. Interfaces que han sido implementadas para el usuario administrador

El *Sprint 1* está conformado por las siguientes tareas:

- Consumir *endpoints* para cambiar contraseña, iniciar y cerrar sesión.
- Consumir endpoints para observar y modificar perfil de usuario.
- Consumir *endpoints* para gestionar pasantes
- Consumir *endpoints* para gestionar inventarios
- Consumir *endpoints* para observar reporte de inventarios.

Consumir *endpoints* para cambiar contraseña, iniciar y cerrar sesión.

Dentro del *frontend* el perfil administrador pueden iniciar sesión por medio de un nombre de usuario y la respectiva contraseña, cuyas credenciales son otorgadas por parte del *backend*, en cambio para los usuarios de tipo administrativo y docente pueden iniciar sesión con las credenciales generadas por su autoría. Por otra parte, los usuarios con perfil pasante pueden iniciar sesión con las credenciales proporcionadas por parte del usuario con perfil administrativo. Una vez que los usuarios se encuentren dentro del *frontend* pueden cerrar sesión y modificar o restablecer la contraseña para lo cual necesitan llenar campos como son: correo electrónico, nueva contraseña y verificar contraseña. Todo esto es posible gracias al consumo de los *endpoints* que han sido generados por parte del *backend*. Posteriormente, en las **Fig. 6**, **Fig. 7** y **Fig. 8** se muestra los componentes visuales del *frontend* y en las respectivas **Fig. 9**, **Fig. 10** y **Fig. 11** se presenta el resultado de la ejecución del *test* unitario y a su vez el funcionamiento con más detalle está en el **ANEXO II**.

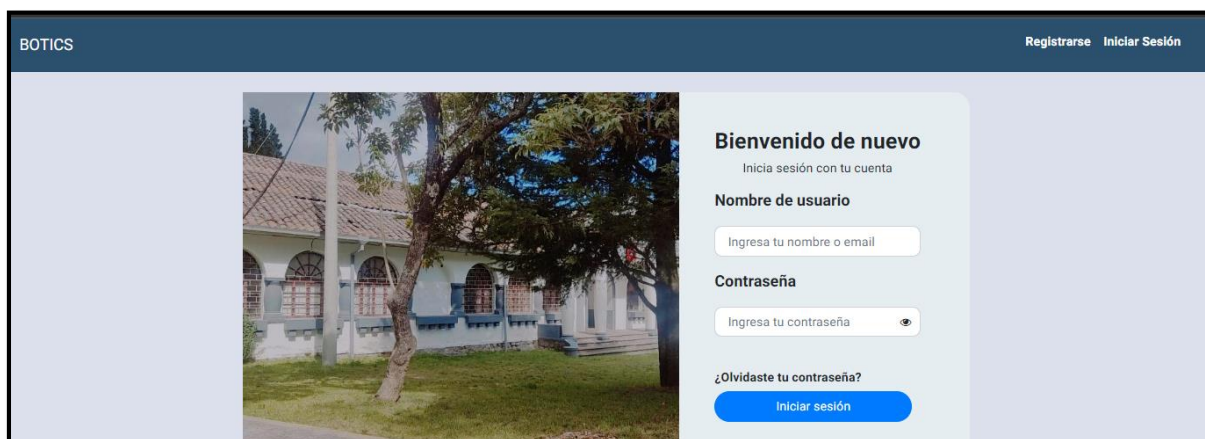


Fig. 6: Inicio de sesión.

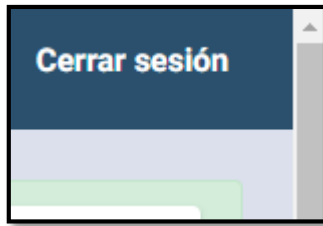


Fig. 7: Cierre de sesión.

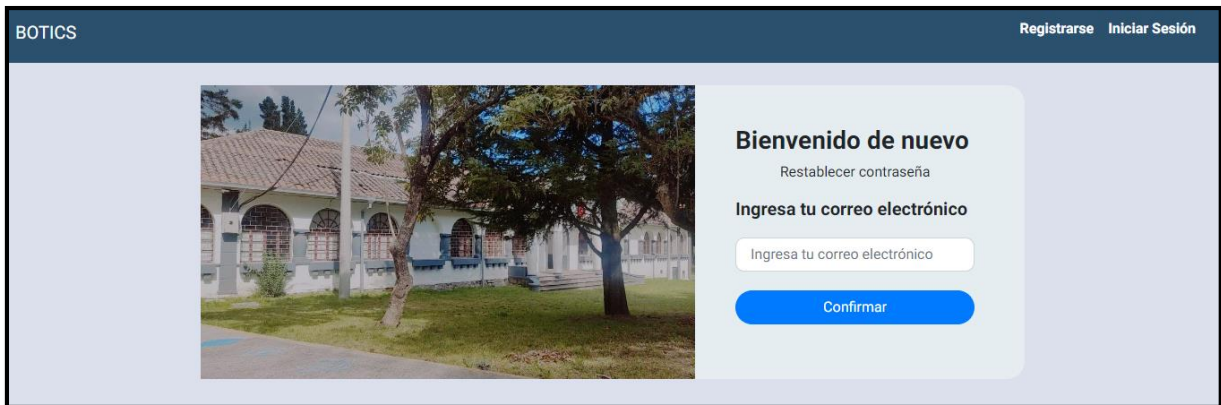


Fig. 8: Modificación/restablecimiento de contraseña.

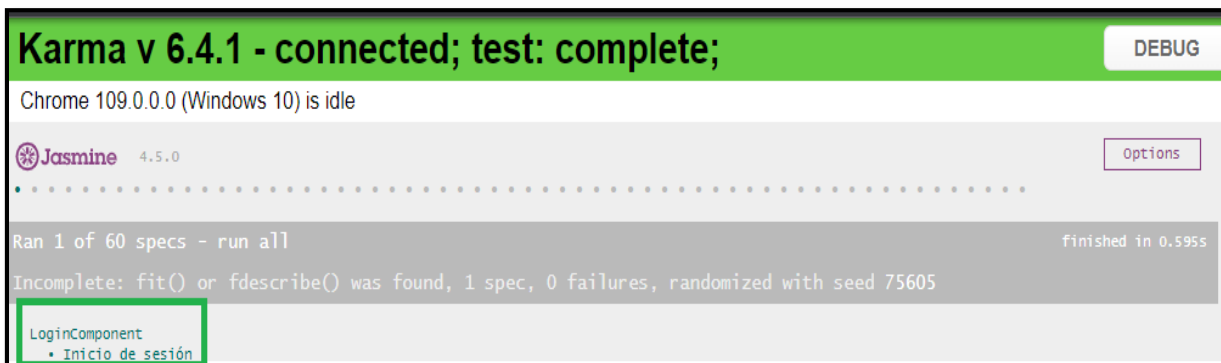


Fig. 9: Resultado obtenido del *test* unitario inicio de sesión.

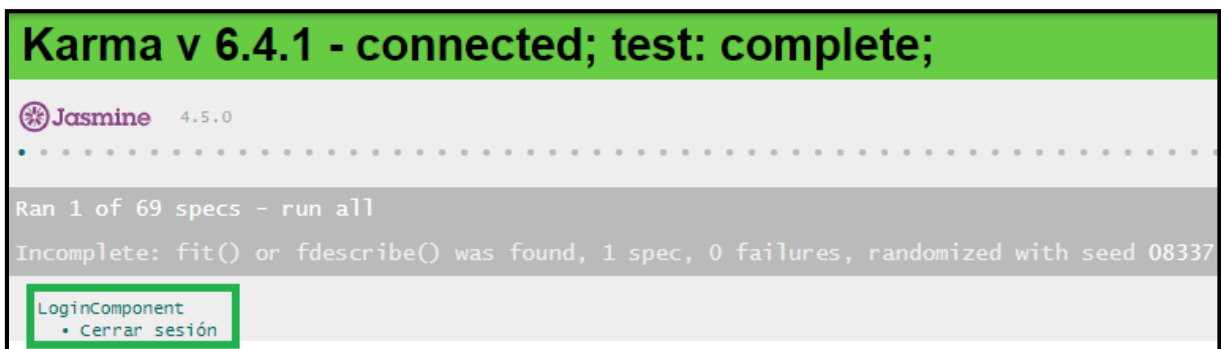


Fig. 10: Resultado obtenido del *test* unitario cerrar sesión.

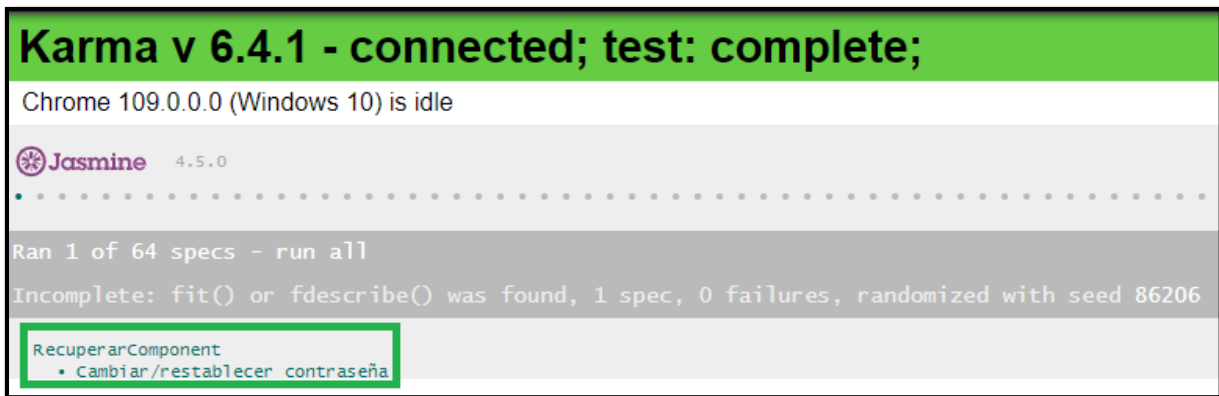


Fig. 11: Resultado obtenido del test unitario modificación/restablecimiento de contraseña.

Consumir *endpoints* para observar y modificar perfil de usuario

Dentro del *frontend* los usuarios de tipo administrador, administrativo, docente y pasante pueden observar su perfil el cual contiene los campos: correo electrónico, nombre de usuario, número celular/extensión y una foto de perfil, y a su vez pueden modificar los campos mencionados anteriormente. Las actividades mencionadas se han llevado a cabo gracias al consumo de los *endpoints* que han sido generados por parte del *backend*. Posteriormente, en la Fig. 12 se muestra los componentes visuales del *frontend* y en la Fig. 13 se presenta el resultado obtenido del *test* unitario y a su vez el funcionamiento con más detalle está en el ANEXO II.



Fig. 12: Observación y modificación del perfil de usuario.



Fig. 13: Resultado del test unitario modificación del perfil de usuario.

Consumir endpoints para gestionar pasantes

En la parte del *frontend* el usuario de tipo administrador puede gestionar pasantes, para lo cual puede registrar pasantes, habilitar o deshabilitar pasantes y a su vez visualizar los pasantes que se han registrado. Por lo tanto, si se ha dado por terminado el periodo del pasante, el usuario con perfil administrador debe inhabilitar a dicho pasante con la finalidad de que el usuario pasante ya no tenga acceso a ninguna información visible desde el *frontend*. Las actividades mencionadas se han llevado a cabo gracias al consumo de los *endpoints* que han sido generados por parte del *backend*. Posteriormente, en las Fig. 14 se muestra los componentes visuales del *frontend* y en la Fig. 15 se presenta el resultado obtenido del test unitario y a su vez el funcionamiento con más detalle está en el ANEXO II.

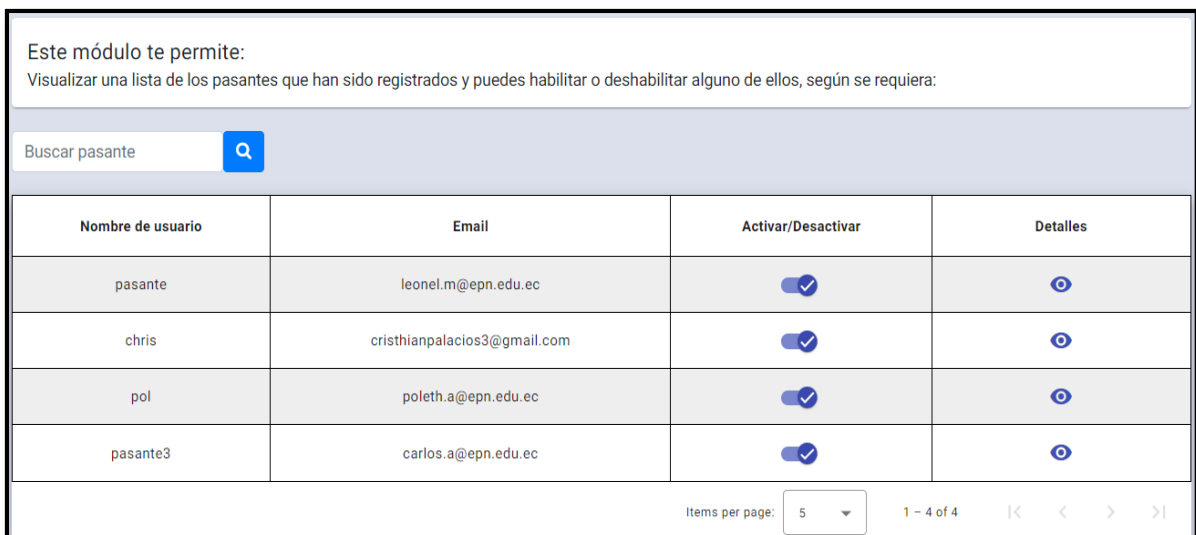


Fig. 14: Gestionar pasantes.


```

Karma v 6.4.1 - connected; test: complete;
Chrome 109.0.0.0 (Windows 10) is idle
Jasmine 4.5.0
Ran 1 of 76 specs - run all
Incomplete: fit() or fdescribe() was found, 1 spec, 0 failures, randomized with seed 30558
ListIComponent
  • Gestionar pasantes

```

Fig. 15: Resultado obtenido del *test* unitario gestionar pasantes.

Consumir *endpoints* para gestionar inventarios

En la parte del *frontend* el perfil administrador puede gestionar inventarios, para el registro de computadoras con sus respectivas características, para lo cual debe llenar un formulario con las características de dicho computador y a su vez puede visualizar el listado de los laboratorios con sus respectivas computadoras. Esto es posible gracias al consumo de los *endpoints* que han sido generados por parte del *backend*. Posteriormente, en la Fig. 16 se muestra los componentes visuales del *frontend* y en la Fig. 17 se presenta el resultado obtenido del *test* unitario y a su vez el funcionamiento con más detalle está en el ANEXO II.

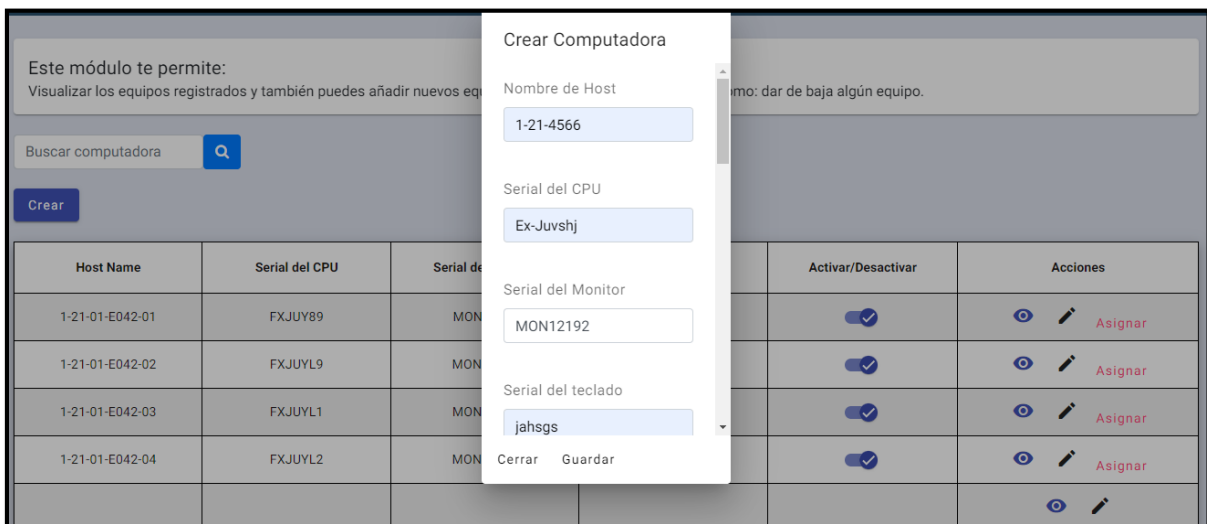


Fig. 16: Registro de equipos para inventario.

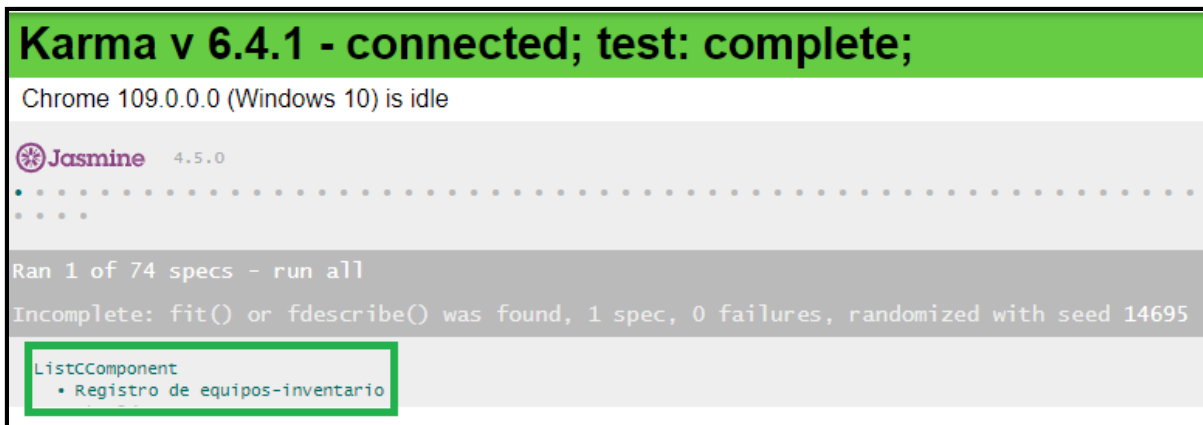


Fig. 17: Resultado obtenido del *test* unitario registro de equipos para inventario.

Consumir *endpoints* para observar reporte de inventarios

En la parte del *frontend* el usuario con perfil administrador puede observar el reporte de inventarios, el cual está conformado por el listado de los inventarios de los equipos tecnológicos de cada laboratorio y también puede visualizar el historial de movimiento de cada equipo tecnológico. Esto es posible gracias al consumo de los *endpoints* que han sido generados por parte del *backend*. Posteriormente, en la **Fig. 18** se muestra los componentes visuales del *frontend* y en la **Fig. 19** se presenta el resultado obtenido del *test* unitario y a su vez el funcionamiento con más detalle está en el **ANEXO II**.

Este módulo te permite:
Visualizar el reporte de las computadoras ingresadas.

Computadoras Historial

Buscar computadora

2/8/23, 11:59 PM

Host Name	Serial del CPU	Serial del Monitor	Modelo	Procesador	RAM	Disco Duro
1-21-01-E042-01	FXJUY89	MON120IL	OPTIPLEX 7070	i7 7700	8 GB	1 TB
1-21-01-E042-02	FXJUYL9	MON120IA	OPTIPLEX 7070	i7 7700	8 GB	1 TB
1-21-01-E042-03	FXJUYL1	MON120IB	OPTIPLEX 7070	i7 7700	8 GB	1 TB

Fig. 18: Observación de reporte de inventarios.

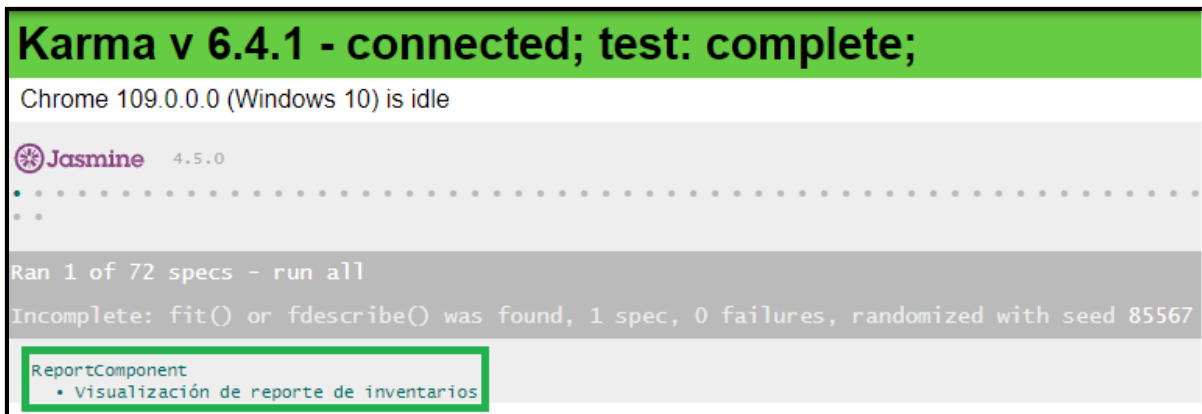


Fig. 19: Resultado obtenido del *test* unitario visualización de reporte de inventarios.

3.3 *Sprint 2*. Interfaces que han sido implementadas para el usuario administrativo

Posteriormente, el *Sprint 2* está conformado por las siguientes tareas:

- Consumir *endpoints* para registrarse.
- Consumir *endpoints* para solicitar *tickets* de asistencia.

Consumir *endpoints* para registrarse

En la parte del *frontend* cada uno de los respectivos usuarios con perfil administrativo y docente pueden registrarse estableciendo un nombre de usuario y contraseña, cuyas credenciales son generadas por su autoría. Las actividades mencionadas se han llevado a cabo gracias al consumo de *endpoints* que han sido generados por parte del *backend*. Posteriormente, en la **Fig. 20** se muestra los componentes visuales del *frontend*, mientras que la **Fig. 21** se evidencias el resultado del *test* unitario y a su vez el funcionamiento con más detalle está en el **ANEXO II**.

BOTICS Registrarse Iniciar Sesión

Bienvenido de nuevo
 Crea una cuenta nueva

Nombre

Apellido

Nombre de usuario

Correo electrónico

Contraseña

Elija el rol
 Administrativo Docente

Fig. 20: Formulario de registro.

```

Karma v 6.4.1 - connected; test: complete;
Chrome 109.0.0.0 (Windows 10) is idle
Jasmine 4.5.0
Ran 1 of 64 specs - run all
Incomplete: fit() or fdescribe() was found, 1 spec, 0 failures, randomized with seed 05015
RegisterComponent
  • Registro
  
```

Fig. 21: Resultado obtenido del *test* unitario registro.

Consumir *endpoints* para solicitar *tickets* de asistencia

Dentro del *frontend* los usuarios con perfil administrativo puede solicitar *tickets* de asistencia, para lo cual debe llenar los siguientes campos del formulario: asunto y descripción. Una vez que los usuarios han redactado su petición acerca de alguna problemática presentada en alguno de los laboratorios, debe presionar en “enviar” para que su petición sea atendida por un usuario de tipo pasante. Las actividades mencionadas se han llevado a cabo gracias al consumo de los *endpoints* que han sido generados por parte del *backend*. Posteriormente, en la Fig. 22 se muestra los componentes visuales del *frontend*, mientras que la Fig. 23 se evidencia el resultado del *test* unitario y a su vez el funcionamiento con más detalle está en el ANEXO II.

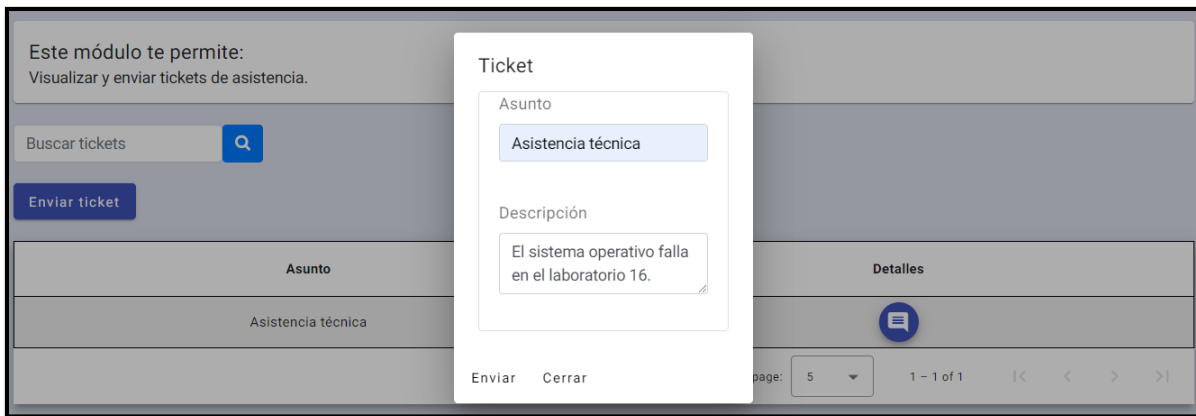


Fig. 22: Formulario de solicitud de *tickets* de asistencia rol administrativo.

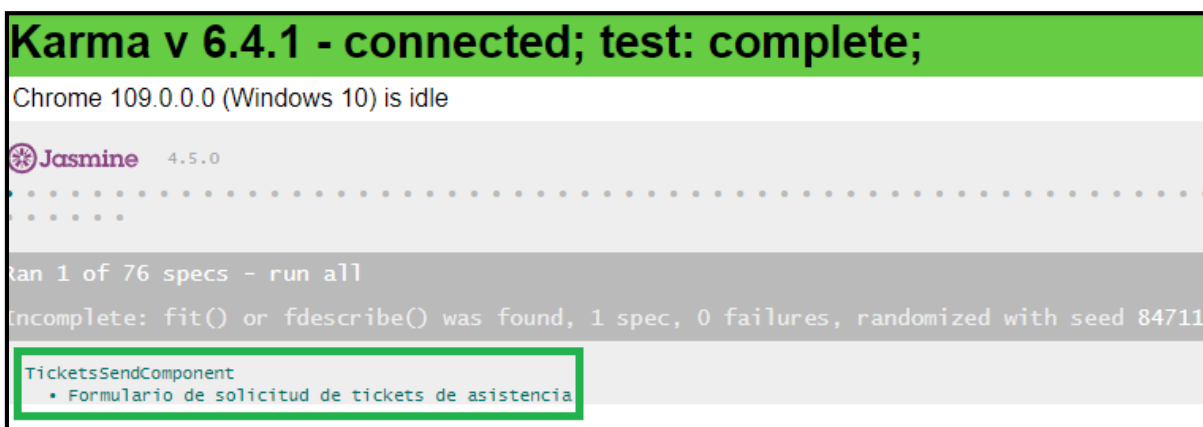


Fig. 23: Resultado obtenido del *test* unitario del formulario de solicitud de *tickets* de asistencia.

3.4 *Sprint* 3. Interfaces que han sido implementadas para el usuario con perfil docente

Posteriormente, el *Sprint* 3 está formado por las siguientes tareas:

- Consumir *endpoints* para visualizar respuestas y enviar comentarios y/o sugerencias.
- Consumir *endpoints* para solicitar *tickets* de asistencia de laboratorios.
- Consumir *endpoints* para solicitar reservas de laboratorios.

Consumir *endpoints* para visualizar respuestas y enviar comentarios y/o sugerencias

Dentro del *frontend* los usuarios con perfil docente y administrativo pueden enviar comentarios y/o sugerencias, para lo cual deben llenar los siguientes campos del formulario: asunto y descripción. Una vez que han sido registrados los comentarios y/o sugerencias, son atendidos por un usuario con perfil pasante. Las actividades mencionadas se han llevado a cabo gracias

al consumo de los *endpoints* que han sido generados por parte del *backend*. Posteriormente, las **Fig. 24** y **Fig. 25** indican los componentes visuales del *frontend* y en la **Fig. 26** y **Fig. 27** se presenta el resultado obtenido del *test* unitario y a su vez el funcionamiento con más detalle está en el **ANEXO II**.

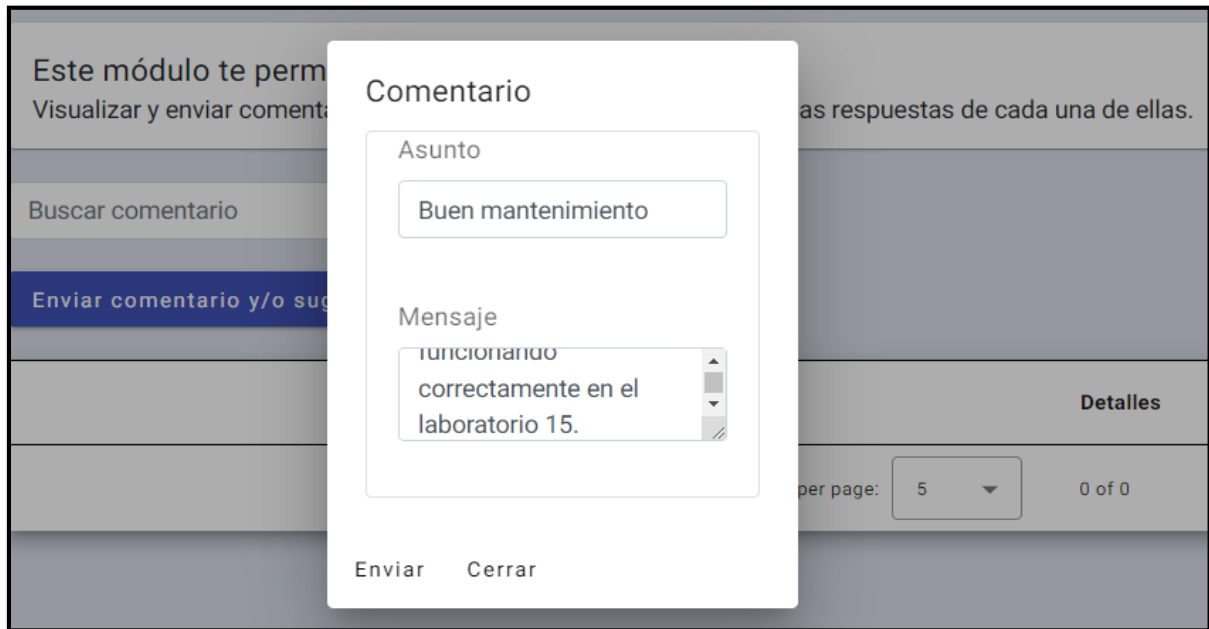


Fig. 24: Formulario para enviar comentarios y/o sugerencias.

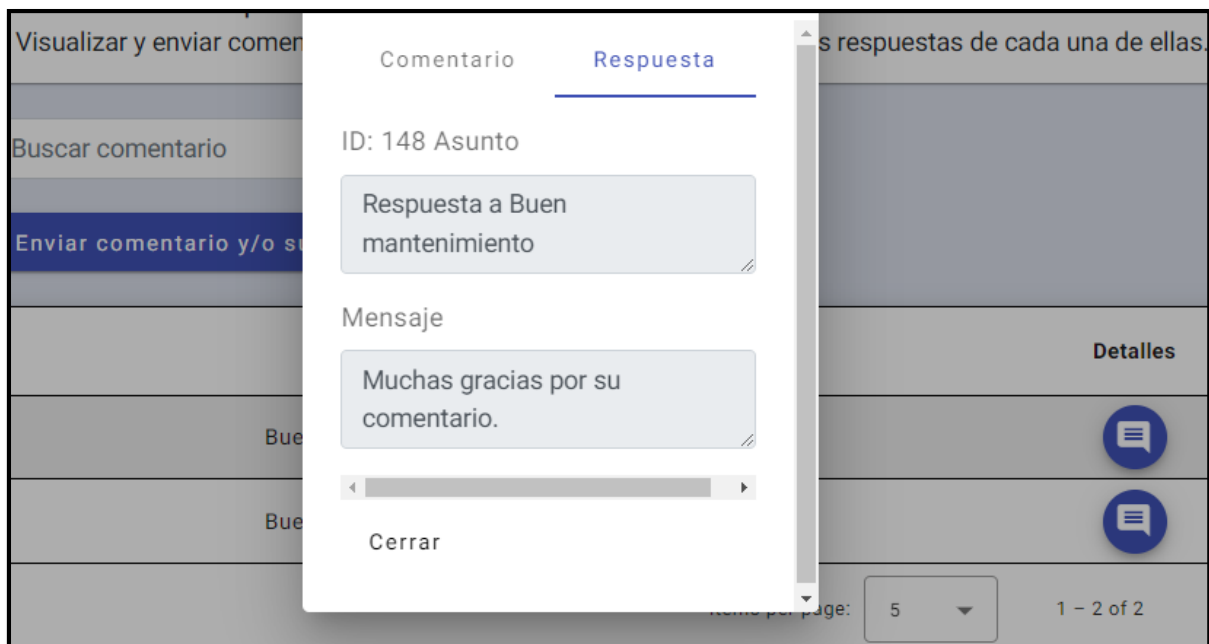


Fig. 25: Visualización de respuestas.

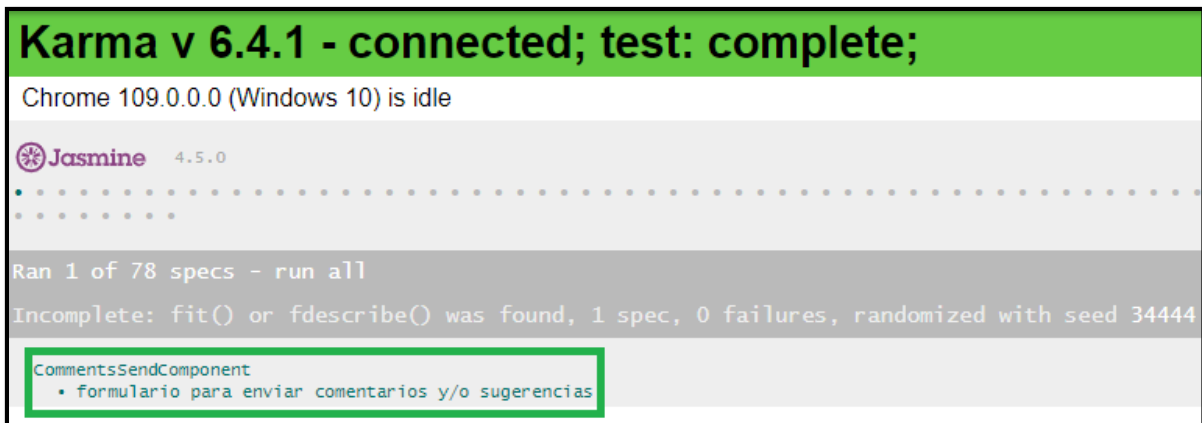


Fig. 26: Resultado obtenido del *test* unitario formulario para enviar comentarios y/o sugerencias

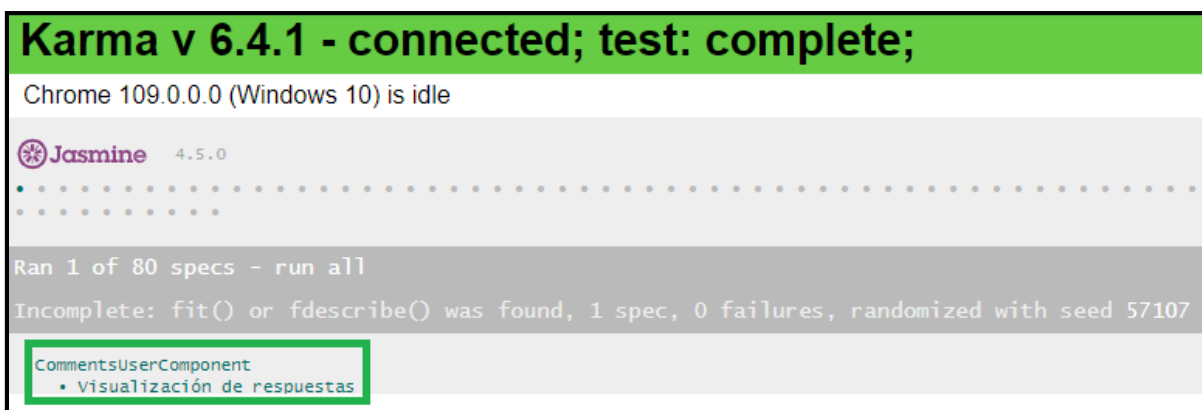


Fig. 27: Resultado obtenido del *test* unitario visualización de respuestas.

Consumir *endpoints* para solicitar *tickets* de asistencia

En la parte del *frontend* el usuario con perfil docente puede solicitar *tickets* de asistencia, para lo cual debe llenar los siguientes campos del formulario: asunto y descripción. Una vez que el usuario docente ha redactado su petición acerca de alguna problemática presentada en alguno de los laboratorios, debe presionar en el botón “enviar” para que su petición sea atendida por un usuario con perfil pasante. Las actividades mencionadas se han llevado a cabo gracias al consumo de los *endpoints* que han sido generados por parte del *backend*. Posteriormente, la Fig. 28 muestra los componentes visuales del *frontend* y en la Fig. 29 se presenta el resultado obtenido del *test* unitario y a su vez el funcionamiento con más detalle está en el ANEXO II.

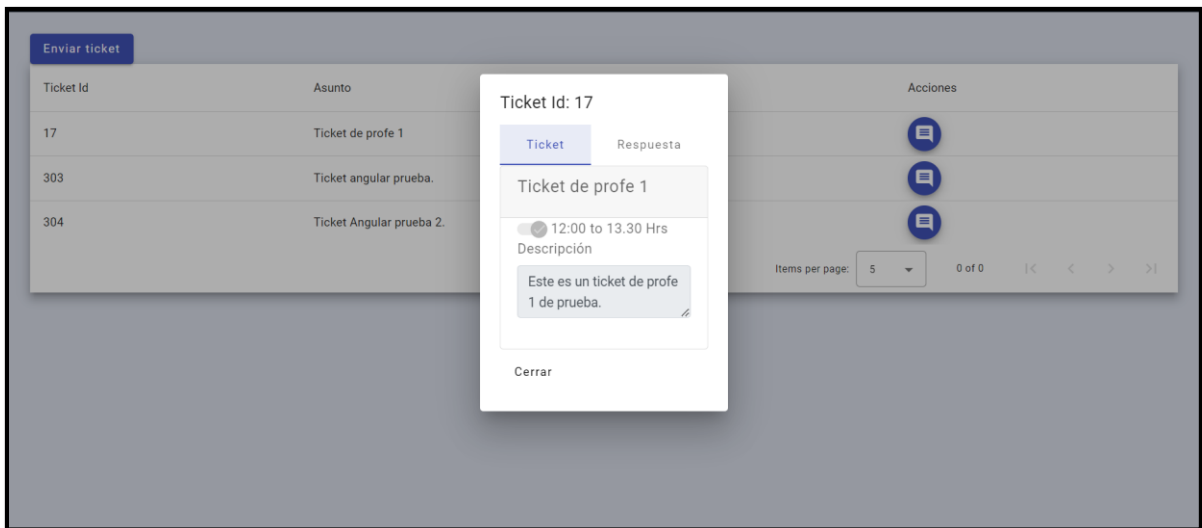


Fig. 28: Formulario para solicitar tickets de asistencia rol docente.

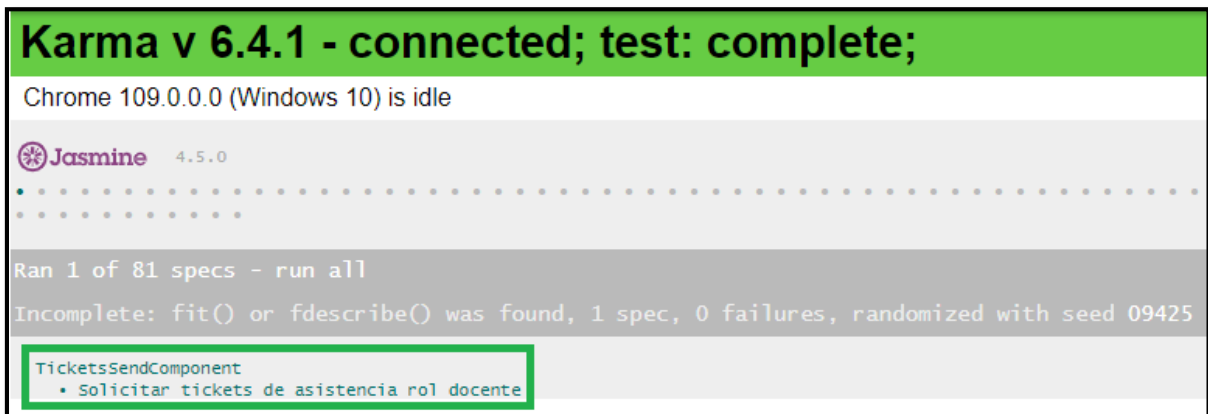


Fig. 29: Resultado obtenido del *test* unitario solicitar *tickets* de asistencia rol docente.

Consumir *endpoints* para solicitar reservas de laboratorios

En la parte del *frontend* el usuario con perfil docente puede solicitar reservas, para lo cual debe llenar los siguientes campos del formulario: nombre del laboratorio, día y hora en el que solicita el laboratorio. Una vez que el usuario docente ha llenado los campos mencionados debe dar clic en enviar para que su solicitud sea atendida por un usuario de tipo pasante. Las actividades mencionadas se han llevado a cabo gracias al consumo de los *endpoints* que han sido generados por parte del *backend*. Posteriormente, en la **Fig. 30**, se muestra los componentes visuales del *frontend* y en la **Fig. 31** se presenta el resultado obtenido del *test* unitario y a su vez el funcionamiento con más detalle está en el **ANEXO II**.

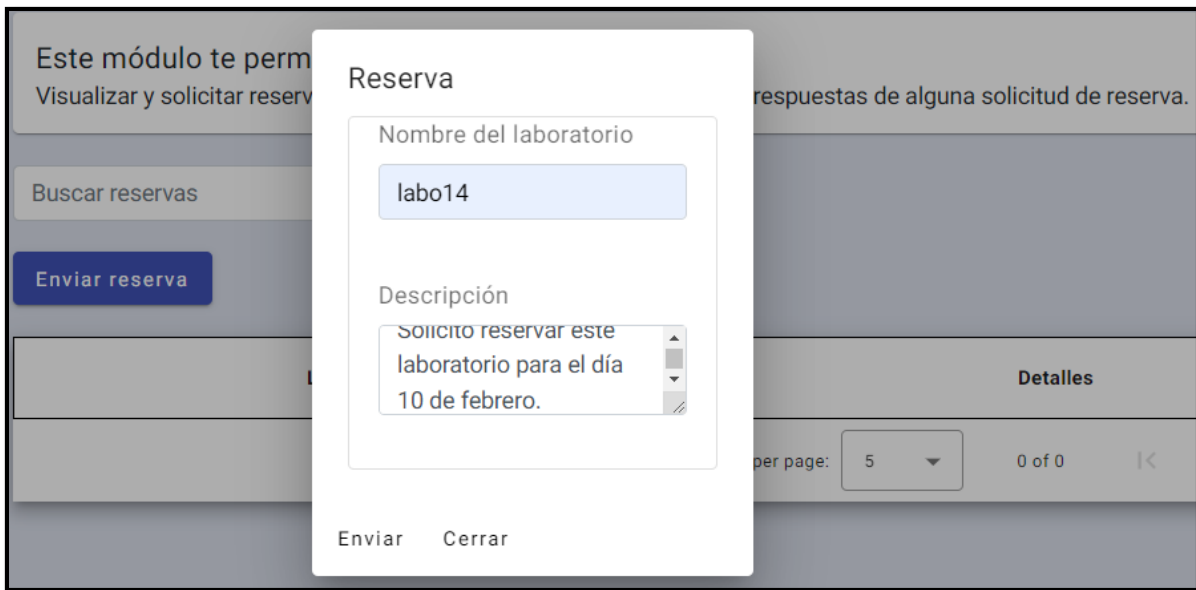


Fig. 30: Formulario para solicitar reservas de laboratorios.



Fig. 31: Resultado obtenido del *test* unitario solicitar reservas de laboratorios.

3.5 *Sprint* 4. Interfaces que han sido implementadas para el usuario pasante

El *Sprint* 4 está conformado por las siguientes tareas:


- Consumir *endpoints* para gestionar *tickets* de asistencia.
- Consumir *endpoints* para gestionar reservas de laboratorios.

Consumir *endpoints* para gestionar *tickets* de asistencia

En la parte del *frontend* el usuario con perfil pasante puede gestionar *tickets* de asistencia, puede visualizar los *tickets* de asistencia que han sido solicitados por parte del usuario docente, los campos que puede visualizar el usuario con perfil pasante son: asunto y

descripción. Las actividades mencionadas se han llevado a cabo gracias al consumo de los *endpoints* que han sido generados por parte del *backend*. Posteriormente, en la **Fig. 32**, se muestra los componentes visuales del *frontend* y en la **Fig. 33** se presenta el resultado obtenido del *test* unitario y a su vez el funcionamiento con más detalle está en el **ANEXO II**.

Este módulo te permite:
 Visualizar los tickets de asistencia solicitados por parte del personal docente y administrativo. También, puedes visualizar las respuestas de los tickets de asistencia y a su vez responder los tickets que no han sido atendidos.

Buscar tickets 







Nombre	Apellido	Email	Rol	Acciones
Byron	Loarte	byron34.loarte@epn.edu.ec	ROLE_PROFESOR	
Byron	Loarte	byron34.loarte@epn.edu.ec	ROLE_PROFESOR	
Wilson	Castillo	wilson@epn.edu.ec	ROLE_ADMINISTRATIVO	 
Byron	Loarte	byron34.loarte@epn.edu.ec	ROLE_PROFESOR	 

Fig. 32: Gestión y observación de *tickets* de asistencia.

```

Karma v 6.4.1 - connected; test: complete;
Chrome 109.0.0.0 (Windows 10) is idle
 Jasmine 4.5.0
.....
Ran 1 of 85 specs - run all
Incomplete: fit() or fdescribe() was found, 1 spec, 0 failures, randomized with seed 93307
TicketsComponent
  • Gestionar tickets de asistencia
  
```


Fig. 33: Resultado obtenido del *test* unitario gestión de *tickets* de asistencia.

Consumir *endpoints* para gestionar reservas de laboratorios

En la parte del *frontend* el usuario con perfil pasante puede gestionar reservas de laboratorios, dichas reservas son realizadas por los usuarios con perfil docente donde detallan alguna solicitud de reserva, una vez que el usuario pasante visualice las solicitudes de reservas envía una respuesta al usuario con perfil docente. Las actividades mencionadas se llevaron a cabo gracias al consumo de los *endpoints* que han sido generados por parte del *backend*. Posteriormente, en la **Fig. 34**, se muestra los componentes visuales del *frontend* y en la **Fig.**

35 se presenta el resultado obtenido del test unitario y a su vez el funcionamiento con más detalle está en el **ANEXO II**.

Este módulo te permite:
Visualizar las reservas de laboratorios enviadas por parte del personal docente y a su vez responder las reservas que no han sido atendidas.

Buscar reservas 







Nombre	Apellido	Email	Rol	Acciones
Byron	Loarte	byron34.loarte@epn.edu.ec	ROLE_PROFESOR	
Byron	Loarte	byron34.loarte@epn.edu.ec	ROLE_PROFESOR	
Alex	Hidalgo	alexhidalgo2019@outlook.com	ROLE_PROFESOR	 
Carlos	Josefo	juanc9384@gmail.com	ROLE_PROFESOR	 

Fig. 34: Gestión de reservas de laboratorios.

```

Karma v 6.4.1 - connected; test: complete;
Chrome 109.0.0.0 (Windows 10) is idle
Jasmine 4.5.0
.....
Ran 1 of 85 specs - run all
Incomplete: fit() or fdescribe() was found, 1 spec, 0 failures, randomized with seed 05174
ReservesComponent
  • Gestionar reservas
  
```

Fig. 35: Resultado obtenido del test unitario gestión de reservas de laboratorios.

3.6 *Sprint 5*. Ejecución de pruebas y despliegue a un entorno de producción

Ahora, se procede a trabajar con el cumplimiento de las tareas del *Sprint 5*, las cuales son:

- Ejecutar y detallar el resultado de *test* unitarios.
- Ejecutar y detallar el resultado de *test* de carga.
- Ejecutar y detallar el resultado de *test* de compatibilidad.
- Ejecutar y detallar el resultado de *test* de aceptación.

- Despliegue del *frontend*.

Ejecutar y detallar el resultado de *test* unitarios.

Con el fin de verificar que existe un correcto funcionamiento del código del *frontend*, se ha realizado pruebas unitarias, las cuales permiten comprobar el correcto funcionamiento de las unidades individuales más pequeñas del código con la finalidad de obtener un producto de calidad ya que reduce el riesgo de que un fragmento de código tenga fallas. Las pruebas unitarias funcionan de esta forma: de uno de los módulos se toma un fragmento de código para posteriormente probar en la herramienta seleccionada y luego visualizar los resultados y en el caso de existir errores, estos se los debe corregir inmediatamente [38].

En la **Fig. 36** se presenta uno de los fragmentos de código que ha sido aplicado previamente en un *test* unitario. Posteriormente, la **Fig. 37** indica el resultado obtenido y su vez en el **ANEXO II** se encuentra el detalle completo de los resultados de los *test* unitarios.

Con la implementación de la prueba mencionada, se puede evidenciar el correcto funcionamiento de los módulos, ya que los fragmentos de código no presentan errores. Por lo tanto, los errores han sido corregidos con éxito.

```
fdescribe('Página informativa', () => {
  let component: HomeComponent;
  let fixture: ComponentFixture<HomeComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ HomeComponent ]
    })
    .compileComponents();

    fixture = TestBed.createComponent(HomeComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  fit('Página informativa', () => {
    expect(component).toBeTruthy();
  });
});
```

Fig. 36: Ejemplar- Página informativa de código del *test* unitario.

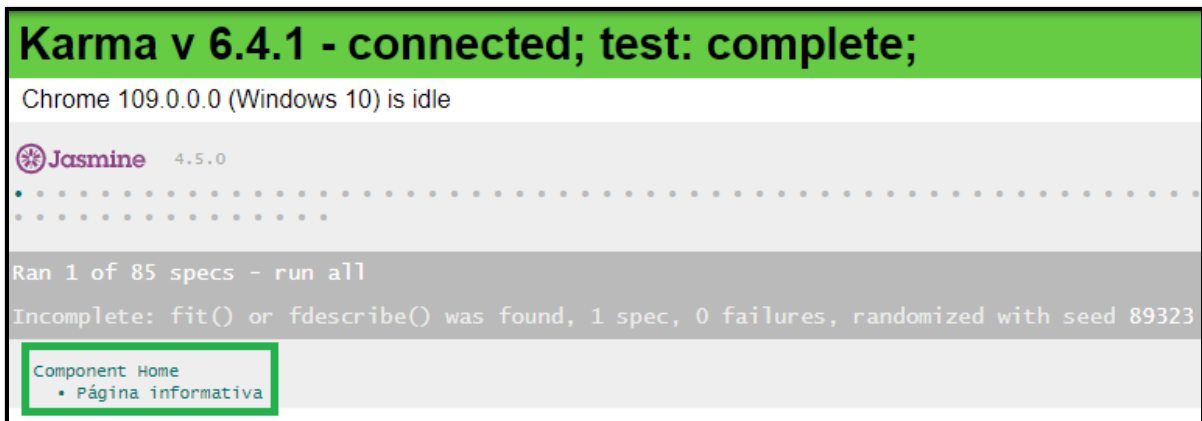


Fig. 37: Ejemplar- Página informativa Resultados obtenidos de la prueba unitaria.

Ejecutar y detallar el resultado de *test* de compatibilidad

Estas pruebas hacen referencia a la verificación de las interfaces de usuario con la finalidad de observar el correcto funcionamiento del producto en distintos dispositivos electrónicos y navegadores *web* [39]. Por tal motivo, la prueba ha sido implementada en los navegadores que se muestran en la **TABLA VIII** y en el **ANEXO II** se encuentra el detalle de las demás pruebas de compatibilidad conjuntamente con los demás resultados que se han obtenido.

Una vez finalizada la prueba mencionada, se determina que el contenido del *frontend* se adapta correctamente en los diferentes navegadores, es decir, los módulos funcionan correctamente y la información se muestra de manera adecuada.

TABLA VIII: Primera prueba de compatibilidad.

NOMBRE	VERSIÓN	OBSERVACIÓN FINAL
Navegador <i>Google Chrome</i>	109.0.5414.122	Operativamente sin problemas
Navegador <i>Microsoft Edge</i>	110.0.1587.50	Operativamente sin problemas
Navegador <i>Opera</i>	95.0.4635.46	Operativamente sin problemas

Ejecutar y detallar el resultado de *test* de aceptación

Dichas pruebas consisten en descubrir si un sistema *software* cumple con los requisitos y expectativas del dueño del producto final y a su vez que certifique que se ha realizado bajo los requerimientos definidos al inicio del proyecto [40]. En **TABLA IX** se puede evidenciar la prueba de aceptación en base a los requisitos que se han determinado por parte del usuario y en el **ANEXO II** se encuentra el detalle de las pruebas de aceptación conjuntamente con los demás resultados.

Una vez finalizado las pruebas de aceptación, se determina que el producto cumple con los requerimientos establecidos al comienzo del proyecto.

TABLA IX: Pruebas de aceptación 1 - Visualizar página informativa.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA001	Identificador de historia de Usuario: HU001
Nombre: Visualizar una página informativa	
<p>Descripción: El <i>frontend</i> por medio del perfil administrador, pasante, docente y administrativo necesitan visualizar una página informativa, en la cual se mostrarán las siguientes vistas:</p> <ul style="list-style-type: none"> • Bienvenida • Laboratorios. • Servicios 	
<p>Pasos de ejecución:</p> <p>Para iniciar sesión usuario administrador, administrativo, pasante y docente:</p> <ul style="list-style-type: none"> • Ubicar la URL del <i>frontend</i> en los navegadores. • Visualizar la página informativa. 	
<p>Resultado deseado:</p> <p>En BOTICS del <i>frontend</i> admite visualizar una página informativa.</p>	
<p>Evaluación de la prueba:</p> <p>Se evidencia el resultado esperado. Conformidad del cliente 100%.</p>	

Despliegue del *frontend*

Ahora, se procede a realizar el despliegue a producción del *frontend* para ello se ha utilizado la herramienta Netlify, la misma provee un *hosting* gratuito en donde se encuentra alojado el *frontend* denominado “BOTICS”, al mismo pueden acceder digitando la siguiente url:

<https://labotics.netlify.app/inicio>

La **Fig. 38** indica el resultado final del despliegue del *frontend* a producción. Por otro lado, en el **ANEXO II** se puede observar el detalle completo del despliegue del *frontend* a producción.

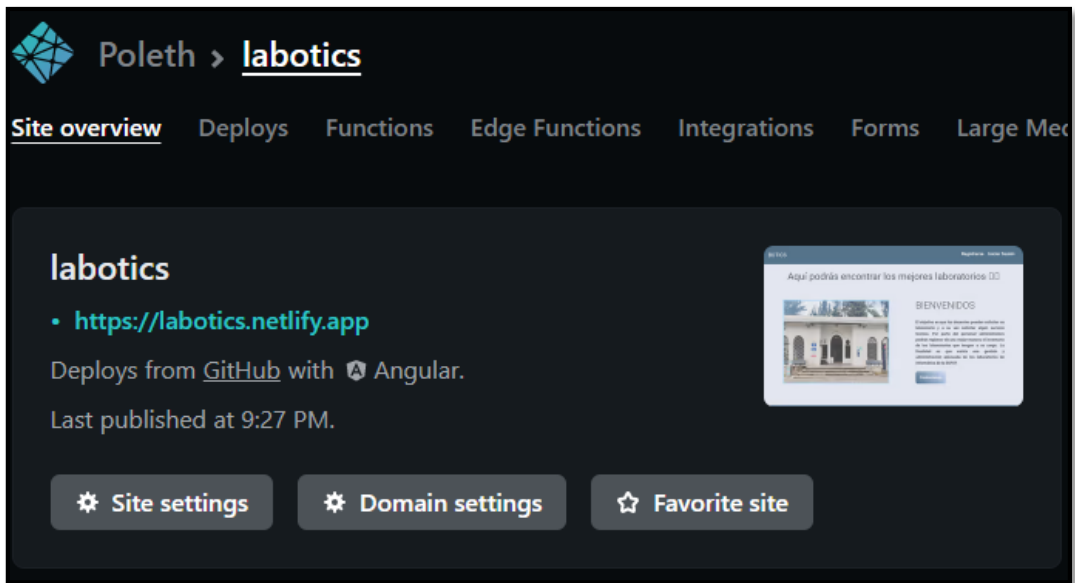


Fig. 38: Despliegue del proyecto a producción.

4 CONCLUSIONES

En esta parte, se exponen las conclusiones obtenidas a lo largo del proyecto de Integración Curricular.

- El actual proyecto de nombre “BOTICS” cumple con todos los objetivos y el alcance, los cuales se establecieron al inicio del proyecto. Logrando así, que el personal administrador, administrativo, docente y pasante puedan interactuar con la aplicación por el lado del cliente (*frontend*), de tal manera que pueden realizar varias acciones dependiendo el rol de usuario.
- La aplicación de la metodología *Scrum* ha hecho posible que el desarrollo del *frontend* se pueda concluir en el tiempo acordado.
- Establecer los requerimientos al inicio del proyecto fue fundamental ya que ha permitido seleccionar de una forma adecuada cada uno de los artefactos, así como cada una de las herramientas para las etapas de codificación, pruebas y despliegue.
- Establecer los prototipos como punto de partida ha facilitado tener en cuenta la UX al inicio del desarrollo del *frontend*.
- La implementación del patrón MVC ha facilitado la organización del proyecto en cuanto a la codificación del *frontend*.
- Gracias a una adecuada implementación de las librerías como herramientas para la compilación del *frontend* ha sido de gran ayuda debido a la facilidad de integración con las demás librerías lo cual ha facilitado el desarrollo del *frontend*.
- En las diferentes pruebas para el *frontend*, se ha obtenido buenos resultados ya que se evidencia el correcto funcionamiento de los módulos, presentación de la información, optimización y aceptación del producto final.
- Una vez establecida la conexión con el *backend*, se puede concluir que BOTICS puede enviar y recibir información de acuerdo a los módulos disponibles según los tipos de usuarios.

5 RECOMENDACIONES

Ahora, se puede observar las recomendaciones obtenidas a lo largo del proyecto de Integración Curricular.

- Se recomienda añadir más funcionalidades para así cumplir con todas las necesidades de los usuarios.
- Se recomienda capacitar a todos los posibles usuarios que pueden acceder a la aplicación por el lado del cliente, para lograr de esta manera generar peticiones de manera correcta.
- Se recomienda implementar políticas de seguridad en la información por parte del *frontend*.
- Es recomendable documentar las API's para lograr consumir la información de forma correcta sin causar problemas a futuro.

6 BIBLIOGRAFÍA

- [1] M. E. S. Vidueira, «La importancia de los inventarios en una empresa. Ekon.,» 29 06 2019. [En línea]. Available: <https://www.ekon.es/blog/importancia-inventarios-empresa/>.
- [2] A. Dgip, « Escuela de Formación de Tecnólogos.,» [En línea]. Available: <https://esfot.epn.edu.ec/>.
- [3] M. Arias Chaves, « La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software.,» *InterSedes*, p. 13, 2007.
- [4] Pressman, *Ingenieria De Software*, 7, 2022.
- [5] N, «¿Qué es Back End, Front End y Back Office y por qué es importante para tu web?,» 07 07 2022. [En línea]. Available: <https://nestrategia.com/desarrollo-web-back-end-front-end/>.
- [6] J. E. Pérez, *Introducción a JavaScript*, 2019.
- [7] J. A. Corrales, «Interfaz de usuario o UI: ¿qué es y cuáles son sus características?,» 12 02 2021. [En línea]. Available: <https://rockcontent.com/es/blog/interfaz-de-usuario/>.
- [8] J. Moreno, «Diseño UX: guía completa sobre la experiencia de usuario.,» 16 05 2022. [En línea]. Available: <https://blog.hubspot.es/marketing/experiencia-de-usuario-ux>.
- [9] M. Boada, «El gran libro de Angular,» 01 2019. [En línea].
- [10] «Qué es HTML,Desarrollo Web,» 01 01 2001. [En línea]. Available: <https://desarrolloweb.com/articulos/que-es-html.html>.
- [11] D. Santos, « Introducción al CSS: qué es, para qué sirve y otras 10 preguntas frecuente,» 09 08 2022. [En línea]. Available: <https://blog.hubspot.es/website/que-es-css>.
- [12] Kinsta, «¿Qué Es el JavaScript? Un Repaso al Lenguaje de Programación de Scripts Más Popular de la Web,» 23 01 2022. [En línea]. Available: <https://kinsta.com/es/base-de-conocimiento/que-es-javascript/>.
- [13] datos.gob.es, «11 librerías para crear visualizaciones de datos,» 03 05 2022. [En línea]. Available: <https://datos.gob.es/es/blog/11-librerias-para-crear-visualizaciones-de-datos>.
- [14] «¿Qué es una API? - Guía sobre las API para principiantes - AWS, Amazon Web Services,» [En línea]. Available: <https://aws.amazon.com/es/what-is/api/>.
- [15] «Generalidades del protocolo HTTP - HTTP,» 03 10 2022. [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>.

- [16] T. Fatunmbi, « A Comparison of Cookies and Tokens for Secure Authentication. Okta Developer,» 08 02 2022. [En línea]. Available: <https://developer.okta.com/blog/2022/02/08/cookies-vs-tokens>.
- [17] «Software DELSOL, Metodología Ágil ¿Qué es?,» 01 04 2019. [En línea]. Available: <https://www.sdelisol.com/glosario/metodologia/>.
- [18] U. Carlemany, «Metodologías de desarrollo de software. Universitat Carlemany,» 06 04 2021. [En línea]. Available: <https://www.universitatcarlemany.com/actualidad/metodologias-de-desarrollo-de-software>.
- [19] B. G. L. Cajamarca, «Desarrollo de un backend para la gestión del sistema penitenciario del Ecuador.,» *Conciencia Digital*, vol. 5, nº 3.2, pp. 47-66, 2022.
- [20] B. G. L. C. y. I. F. M. Soliz, «Desarrollo de una aplicación web y móvil en tiempo real, una evolución de las aplicaciones actuales.,» *Ciencia Digital*, vol. 3, nº 1, pp. 201- 216, 2019.
- [21] Trigas, ««Metodología Scrum,»,» [En línea]. Available: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>.
- [22] «Scrum: roles y responsabilidades.,» 14 01 2019. [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/roles-y-responsabilidades-scrum.html>.
- [23] «Artefactos Scrum: las 3 herramientas clave de gestión.,» 14 01 2019. [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/artefactos-scrum.html>.
- [24] Asana, «Guía de 6 pasos para la recopilación de requisitos para asegurar el éxito de tu proyecto,» [En línea]. Available: <https://asana.com/es/resources/requirements-gathering>. [Último acceso: 2022 11 03].
- [25] Digite, «Historias De Usuarios: Qué Son Y Por Qué Y Cómo Usarlas.,» 25 08 2021. [En línea]. Available: <https://www.digite.com/es/agile/historias-de-usuarios/>.
- [26] J. Ramos, «Scrum: ¿Qué es el Product Backlog? Programación y más,» [En línea]. Available: <https://programacionymas.com/blog/scrum-product-backlog>.
- [27] «Sprint y Sprint Backlog: puntos esenciales de SCRUM, Integra IT Soluciones,» 16 11 2021. [En línea]. Available: <https://integrait.com.mx/blog/sprint-y-sprint-backlog/>.
- [28] «Impress Group, La importancia de la gestión de sistemas informáticos en la empresa.,» 5 04 2020. [En línea]. Available: <https://www.impress-group.com/la-importancia-de-la-gestion-de-sistemas-informaticos-en-la-empresa/>.
- [29] K. Bracey, «¿Qué es Figma? Web Design Envato Tuts+,» 13 05 2020. [En línea]. Available: <https://webdesign.tutsplus.com/es/articles/what-is-figma--cms-32272>.

- [30] «2.1 Arquitectura de las aplicaciones Web. Programacion Web,» 14 11 2013. [En línea]. Available: <https://programacionwebisc.wordpress.com/2-1-arquitectura-de-las-aplicaciones-web/>.
- [31] C. V. H. Otálora, « Patrón arquitectónico MVC (Modelo Vista Controlador).,» 14 06 2018. [En línea]. Available: <https://repositorio.konradlorenz.edu.co/handle/001/138>.
- [32] «“Angular.”,» [En línea]. Available: <https://angular.io/>. [Último acceso: 08 11 2022].
- [33] «Angular - Desarrolle sus aplicaciones web con el framework JavaScript de Google - ¿Qué es Angular CLI?,» [En línea]. Available: <https://www.ediciones-eni.com/open/mediabook.aspx?idR=59febeb081a24f089644d9223202988f>.
- [34] «AngularJS Material,» [En línea]. Available: <https://material.angularjs.org/latest/>.
- [35] L. Bravo, « Framework o librerías: ventajas y desventajas. tiThink Technology,» 05 09 2018. [En línea]. Available: <https://www.tithink.com/es/2018/08/29/framework-o-librerias-ventajas-y-desventajas/>.
- [36] «Introducción a Angular Material, Tribalyte Technologie,» 16 02 2021. [En línea]. Available: <https://tech.tribalyte.eu/blog-introduccion-angular-material>.
- [37] «Angular,» [En línea]. Available: <https://material.angular.io/components/icon/overview>.
- [38] I. Alvarado, « LA IMPORTANCIA DE LAS PRUEBAS UNITARIAS PARA COMPROBAR FRAGMENTOS DE CÓDIGO. Cero Ideas.,» 18 03 2020. [En línea]. Available: <https://ceroideas.es/la-importancia-de-las-pruebas-unitarias-para-comprobar-fragmentos-de-codigo/>.
- [39] A. Puente, «Pruebas de compatibilidad. Globe Testing,» 26 06 2017. [En línea]. Available: <https://ahorasomos.izertis.com/globetesting/2012/07/pruebas-de-compatibilidad/>.
- [40] J. Ramos, « Los diferentes tipos de Pruebas de software. Programación y más,» [En línea]. Available: <https://programacionymas.com/blog/tipos-de-testing-en-desarrollo-de-software>.
- [41] «Aptitud Emprendedora, La importancia del frontend y backend en un proyecto,» [En línea]. Available: https://es.slideshare.net/aptitud_emprendedora/la-importancia-del-frontend-y-backend-en-un-proyecto.

7 ANEXOS

A continuación, se presenta cada uno de los Anexos que se ha utilizado para el desarrollo del *frontend*, los cuales se encuentran detallados de la siguiente manera:

- **ANEXO I.** Resultado del programa antiplagio Turnitin.
- **ANEXO II.** Manual de Usuario.
- **ANEXO III.** Manual de Instalación.
- **ANEXO IV.** Credenciales de acceso y despliegue.

ANEXO I

A continuación, se presenta el certificado que el Director de Tesis ha emitido y en donde se evidencia el resultado que se ha obtenido en la herramienta antiplagio Turnitin.



ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
CAMPUS POLITÉCNICO "ING. JOSÉ RUBÉN ORELLANA"

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 15 de febrero de 2023

De mi consideración:

Yo, Loarte Cajamarca Byron Gustavo, en calidad de Director del Trabajo de Integración Curricular titulado Desarrollo de un frontend asociado al DESARROLLO DE SISTEMA PARA LA GESTIÓN DE LOS LABORATORIOS DE INFORMÁTICA - ESFOT elaborado por la estudiante Poleth Alejandra Arias Procel de la carrera en Tecnología Superior en Desarrollo de Software, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito secciones: Descripción del componente desarrollado, Metodología, Resultados, Conclusiones y Recomendaciones, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 11%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

Atentamente,

Loarte Cajamarca Byron Gustavo
Profesor Ocasional a Tiempo Completo
Escuela de Formación de Tecnólogos

ANEXO II

Recopilación de requerimientos

En la **Tabla X** se visualiza los requerimientos que se han definido al inicio del proyecto en donde concuerda con lo solicitado por el usuario final.

Tabla X: Requerimientos que se han recopilado.

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DEL SISTEMA	ID - RR	ENUNCIADO DEL ÍTEM
Frontend	RR001	Como usuario administrador, pasante, docente y administrativo necesitan visualizar una página informativa.
	RR003	Como usuario administrador, pasante, docente y administrativo necesitan consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Iniciar sesión • Cerrar sesión • Modificar contraseña.
	RR004	Como usuario administrador, pasante, docente y administrativo necesitan consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Modificar perfil de usuario.
	RR005	Como usuario administrador necesita generar varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Gestionar pasantes.
	RR006	Como usuario administrador necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Gestionar inventarios.
	RR007	Como usuario administrador y pasante necesitan consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Visualizar reporte de inventarios
	RR008	Como usuario administrativo necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Gestionar comentarios y/o sugerencias.
	RR009	Como usuario pasante necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Gestionar tickets de asistencia

	RR010	Como usuario pasante necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Gestionar reservas.
	RR011	Como usuario docente y administrativo necesitan consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Solicitar tickets de asistencia.
	RR012	Como usuario docente necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Solicitar reservas.
	RR013	Como usuario docente y administrativo necesitan consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Enviar comentarios y/o sugerencias.

Historias de Usuario

Terminada la etapa de recopilación de requerimientos, se da paso a crear las Historias de usuarios para el *frontend*. Es por ello, que se presentan las 12 Historias de usuario guiadas en base a los requerimientos que comienzan desde **TABLA XI** hasta **TABLA XVII**.

TABLA XI: Registrarse en BOTICS.

HISTORIA DE USUARIO	
Identificador: HU002	Usuario: Docente y Administrativo
Nombre historia: Registrarse	
Prioridad en Negocio: Alta	Riesgo en Negocio: Media
Iteración asignada: 1	
Responsable: Poleth Arias	
Descripción: En el <i>fronted</i> los usuarios docente y administrativo necesitan consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Registrarse. 	
Observación: Los usuarios docente y administrativo para registrarse necesitan llenar los campos e-mail y contraseña de manera obligatoria.	

TABLA XII: Iniciar sesión, cerrar sesión y modificar.

HISTORIA DE USUARIO	
Identificador: HU003	Usuario: Administrador, pasante, docente y administrativo
Nombre historia: Iniciar sesión, cerrar sesión y modificar.	
Prioridad en Negocio: Alta	Riesgo en Negocio: Alta
Iteración asignada: 1	
Responsable: Poleth Arias	
<p>Descripción: En el <i>frontend</i> los usuarios administrador, pasante, docente y administrativo necesitan consumir varios <i>endpoints</i> para:</p> <ul style="list-style-type: none"> • Iniciar Sesión. • Cerrar Sesión. • Modificar contraseña. 	
<p>Observación:</p> <p>Los usuarios administrador, pasante, docente y administrativo pueden iniciar sesión, cerrar sesión y modificar contraseña, para lo cual es necesario que inicien sesión.</p>	

TABLA XIII: Modificar perfil de usuario de rol administrador, administrativo, pasante y docente.

HISTORIA DE USUARIO	
Identificador: HU004	Usuario: Administrador, administrativo, pasante y docente.
Nombre historia: Modificar perfil de usuario.	
Prioridad en Negocio: Media	Riesgo en Negocio: Media
Iteración asignada: 1	
Responsable: Poleth Arias	
<p>Descripción: En el <i>frontend</i> los usuarios administrador, administrativo, pasante y docente necesitan consumir varios <i>endpoints</i> para visualizar y editar su perfil por medio de los siguientes campos:</p> <ul style="list-style-type: none"> • Nombres. • Apellidos. • Avatar. 	

<ul style="list-style-type: none"> • Extensión.
<p>Observación:</p> <p>Los usuarios administrador, administrativo, pasante y docente pueden acceder a las vistas para visualizar y editar el perfil de usuario una vez han iniciado sesión. En la vista para editar el perfil se presentará un formulario con los campos descritos anteriormente.</p>

TABLA XIV: Gestionar pasantes.

HISTORIA DE USUARIO	
Identificador: HU005	Usuario: Administrador
Nombre historia: Gestionar pasantes	
Prioridad en Negocio: Media	Riesgo en Negocio: Media
Iteración asignada: 2	
Responsable: Poleth Arias	
<p>Descripción: En el <i>frontend</i> el usuario administrador necesita consumir varios <i>endpoints</i> para:</p> <ul style="list-style-type: none"> • Registrar pasantes. • Visualizar lista de pasantes. • Correo electrónico. 	
<p>Observación:</p> <p>El usuario administrador es el único que puede acceder a las <i>vistas</i> anteriormente mencionadas, es decir, necesita iniciar sesión para poder gestionar pasantes.</p>	

TABLA XV: Gestionar inventarios.

HISTORIA DE USUARIO	
Identificador: HU006	Usuario: Administrador
Nombre historia: Gestionar inventarios	
Prioridad en Negocio: Alta	Riesgo en Negocio: Media
Iteración asignada: 2	

Responsable: Poleth Arias
<p>Descripción: En el <i>frontend</i> el usuario administrador necesita consumir varios <i>endpoints</i> para:</p> <ul style="list-style-type: none"> • Registrar inventarios. • Modificar inventarios. • Visualizar inventarios. • Eliminar inventarios.
<p>Observación:</p> <p>El usuario administrador podrá acceder a la vista en la cual podrá gestionar inventarios una vez ha iniciado sesión y accedido al módulo de inventarios.</p>

TABLA XVI: Visualizar reporte de inventarios.

HISTORIA DE USUARIO	
Identificador: HU007	Usuario: Administrador
Nombre historia: Visualizar reporte de inventarios	
Prioridad en Negocio: Alta	Riesgo en Negocio: Media
Iteración asignada: 2	
Responsable: Poleth Arias	
<p>Descripción: En el <i>frontend</i> el usuario administrador necesita consumir varios <i>endpoints</i> para:</p> <ul style="list-style-type: none"> • Visualizar reporte de inventario. 	
<p>Observación:</p> <p>Al usuario administrador se le presentará una vista en la cual podrá visualizar el reporte de los inventarios registrados.</p>	

TABLA XVII: Gestionar comentarios y/o sugerencias.

HISTORIA DE USUARIO	
Identificador: HU008	Usuario: Administrativo
Nombre historia: Gestionar comentarios y/o sugerencias	

Prioridad en Negocio: Alta	Riesgo en Negocio: Media
Iteración asignada: 3	
Responsable: Poleth Arias	
<p>Descripción: En el <i>frontend</i> los usuarios administrativo y pasante necesitan consumir varios <i>endpoints</i> para:</p> <ul style="list-style-type: none"> • Visualizar comentarios y/o sugerencias. • Atender comentarios y/o sugerencias. 	
<p>Observación:</p> <p>Los usuarios administrativo y pasante son los únicos que pueden acceder a los <i>endpoints</i> anteriormente mencionados, es decir, necesitan iniciar sesión para poder gestionar comentarios y/o sugerencias.</p>	

TABLA XVIII: Gestionar *tickets* de asistencia.

HISTORIA DE USUARIO	
Identificador: HU009	Usuario: Pasante
Nombre historia: Gestionar <i>tickets</i> de asistencia.	
Prioridad en Negocio: Alta	Riesgo en Negocio: Media
Iteración asignada: 3	
Responsable: Poleth Arias	
<p>Descripción: En el <i>frontend</i> el usuario de perfil pasante necesita consumir varios <i>endpoints</i> para:</p> <ul style="list-style-type: none"> • Visualizar <i>tickets</i>. • Atender <i>tickets</i>. 	
<p>Observación:</p> <p>El usuario pasante es el único que puede acceder a las <i>opciones</i> anteriormente mencionadas, es decir, necesita iniciar sesión para poder gestionar tickets de asistencia.</p>	

TABLA XIX: Gestionar reservas.

HISTORIA DE USUARIO	
Identificador: HU010	Usuario: Pasante
Nombre historia: Gestionar reservas	
Prioridad en Negocio: Baja	Riesgo en Negocio: Baja
Iteración asignada: 3	
Responsable: Poleth Arias	
<p>Descripción: En el <i>frontend</i> el usuario con perfil pasante necesita consumir varios <i>endpoints</i> para:</p> <ul style="list-style-type: none"> • Visualizar reservas. • Tramitar reservas. 	
<p>Observación:</p> <p>El usuario con perfil pasante es el único que puede acceder a las opciones anteriormente mencionadas, es decir, necesita iniciar sesión para poder gestionar reservas.</p>	

TABLA XX: Solicitar *tickets* de asistencia.

HISTORIA DE USUARIO	
Identificador: HU011	Usuario: Docente y administrativo
Nombre historia: Solicitar <i>tickets</i> de asistencia.	
Prioridad en Negocio: Alta	Riesgo en Negocio: Media
Iteración asignada: 4	
Responsable: Poleth Arias	
<p>Descripción: En el <i>frontend</i> los usuarios con perfil docente y administrativo necesitan consumir varios <i>endpoints</i> para:</p> <ul style="list-style-type: none"> • Solicitar un <i>ticket</i> de asistencia. 	
Observación:	

Los usuarios con perfil docente y administrativo son los únicos que pueden acceder a las opciones anteriormente mencionadas, es decir, necesitan iniciar sesión para poder gestionar *tickets* de asistencia.

TABLA XXI: Solicitar reservas.

HISTORIA DE USUARIO	
Identificador: HU012	Usuario: Docente
Nombre historia: Solicitar reservas	
Prioridad en Negocio: Alta	Riesgo en Negocio: Media
Iteración asignada: 4	
Responsable: Poleth Arias	
Descripción: En el <i>frontend</i> el usuario de perfil docente necesita consumir varios <i>endpoints</i> para: Registrar solicitud de reservas.	
Observación: El usuario docente es el único que puede acceder a la vista anteriormente mencionada, es decir, necesita iniciar sesión para poder solicitar reservas.	

TABLA XXII: Enviar comentarios y/o sugerencias.

HISTORIA DE USUARIO	
Identificador: HU013	Usuario: Docente y administrativo
Nombre historia: Enviar comentarios y/o sugerencias	
Prioridad en Negocio: Media	Riesgo en Negocio: Media
Iteración asignada: 4	
Responsable: Poleth Arias	
Descripción: En el <i>frontend</i> los usuarios de perfil docente y administrativo permite implementar varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Registrar comentarios y/o sugerencias. 	

Observación:

Los usuarios docente y administrativo son los únicos que pueden acceder a la vista anteriormente mencionada, es decir, necesitan iniciar sesión para poder solicitar reservas.

Product Backlog

La **TABLA XXIII** muestra el orden de prioridad de cada módulo que se ha implementado en el componente, los cuales son asignados tomando en cuenta la prioridad del usuario y su dificultad.

TABLA XXIII: *Product Backlog.*

ELABORACIÓN DEL <i>PRODUCT BACKLOG</i>				
ID – HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU003	Iniciar sesión, cerrar sesión y modificar contraseña.	1	Finalizada	Media
HU004	Modificar perfil de usuario.	1	Finalizada	Media
HU005	Gestionar pasantes	2	Finalizada	Media
HU006	Gestionar inventarios.	2	Finalizada	Alta
HU007	Visualizar reporte de inventarios.	2	Finalizada	Alta
HU008	Gestionar comentarios y/o sugerencias	3	Finalizada	Alta
HU009	Gestionar <i>tickets</i> de asistencia.	3	Finalizada	Alta
HU010	Gestionar reservas.	3	Finalizada	Media
HU011	Solicitar <i>tickets</i> de asistencia.	4	Finalizada	Alta
HU012	Solicitar reservas.	4	Finalizada	Alta
HU013	Enviar comentarios y/o sugerencias	4	Finalizada	Media

Sprint Backlog

La **TABLA XXIV** muestra todos los *Sprint* que se han planificado en el proyecto los cuales suman un total de 6, en donde cada uno cuenta con el tiempo estimado de cada actividad para culminar con los entregables ya definidos con el usuario.

TABLA XXIV: *Sprint Backlog.*

ELABORACIÓN DEL <i>SPRINT BACKLOG</i>						
ID-SB	NOMBRE	MÓDULO	ID-HU	HISTORIA DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB000	Información principal del <i>Frontend</i>	Módulo informativo	HU001	Visualizar página informativa	<ul style="list-style-type: none">• Diseño de prototipos de la página informativa.• Definir la información de cada módulo.• Diseño de las interfaces de la página informativa.	20h

SB001	Diseño e implementación de la interfaz de los módulos: registro, inicio de sesión y modificación del perfil.	Módulo de registro	HU002	Registrarse	<ul style="list-style-type: none"> • Diseño e implementación del registro para el usuario administrativo y docente. • Validación de los campos requeridos. 	40h
		Módulo inicio de sesión .	HU003	Iniciar sesión, cerrar sesión y modificar.	<ul style="list-style-type: none"> • Diseño e implementación de las interfaces de iniciar sesión, cerrar sesión y modificar contraseña para todos los usuarios. • Validación de los campos requeridos. 	
		Módulo de perfil de usuario.	HU004	Modificar perfil de usuario	<ul style="list-style-type: none"> • Diseño e implementación de la interfaz para visualizar y modificar el perfil de usuario de todos los perfiles de los usuarios. • Validación de los campos requeridos. 	

SB002	Diseño e implementación de la interfaz del usuario administrador con sus respectivos módulos	Módulo de gestión de pasantes	HU005	Gestionar pasantes	<ul style="list-style-type: none"> • Diseño e implementación de las interfaces para registrar y visualizar pasantes. • Validación de los campos requeridos. 	40h
		Módulo de gestión de inventarios	HU006	Gestionar inventarios	<ul style="list-style-type: none"> • Diseño e implementación de las interfaces para registrar, modificar, visualizar y eliminar inventarios. • Validación de los campos requeridos. 	
		Módulo de visualización de reportes de inventarios	HU007	Visualizar reporte de inventarios	<ul style="list-style-type: none"> • Diseño e implementación de métodos para visualizar el reporte de inventario. 	

SB003	Diseño e implementación de la interfaz del usuario administrativo con sus respectivos módulos.	Módulo de gestión de comentarios y/o sugerencias	HU008	Gestionar comentarios y/o sugerencias	<ul style="list-style-type: none"> • Diseño e implementación de la interfaz para visualizar y atender comentarios y sugerencias. 	30h
		Módulo de solicitud de <i>tickets</i> de asistencia	HU009	Solicitar <i>tickets</i> de asistencia	<ul style="list-style-type: none"> • Diseño e implementación de la interfaz para solicitar <i>tickets</i> de asistencia. • Validación de los campos requeridos 	
		Módulo de envío de comentarios y/o sugerencias	HU010	Enviar comentarios y/o sugerencias	<ul style="list-style-type: none"> • Diseño e implementación de la interfaz para registrar comentarios y/o sugerencias. • Validación de los campos requeridos. 	
	Diseño e implementación de la interfaz del usuario docente con sus respectivos	Módulo de solicitud de <i>tickets</i> de asistencia	HU009	Solicitar <i>tickets</i> de asistencia	<ul style="list-style-type: none"> • Diseño e implementación de la interfaz para solicitar <i>tickets</i> de asistencia. • Validación de los campos requeridos. 	

SB004	módulos.	Módulo de envío de comentarios y/o sugerencias	HU010	Enviar comentarios y/o sugerencias	<ul style="list-style-type: none"> • Diseño e implementación de la interfaz para registrar comentarios y/o sugerencias. • Validación de los campos requeridos. 	30h
		Módulo de solicitud de reservas	HU011	Solicitar reservas	<ul style="list-style-type: none"> • Diseño e implementación de la interfaz para registrar solicitudes de reservas. • Validación de los campos requeridos. 	
SB005	Diseño e implementación de la interfaz del usuario pasantes con sus respectivos módulos	Módulo de gestión de <i>tickets</i> de asistencia	HU012	Gestionar <i>tickets</i> de asistencia	<ul style="list-style-type: none"> • Diseño e implementación de la interfaz para visualizar y atender los <i>tickets</i> de asistencia. 	30h
			HU013	Gestionar reservas	<ul style="list-style-type: none"> • Diseño e implementación de la interfaz para visualizar y tramitar las reservas. • Validación de los campos requeridos. 	
SB006	Pruebas y despliegue del <i>frontend</i>	Pruebas en el <i>frontend</i>	<ul style="list-style-type: none"> • Pruebas unitarias. • Pruebas de carga. • Pruebas de compatibilidad. • Pruebas de aceptación. 			20h

		Publicación del <i>frontend</i>	<ul style="list-style-type: none"> Despliegue del <i>frontend</i> en <i>Netlify</i> 	
	Documentación		Trabajo de Integración Curricular. <ul style="list-style-type: none"> Anexos. 	30h
TOTAL				240

Diseño de interfaces

Ahora, se observa los bosquejos de cada módulo del componente *frontend* donde representan cada uno de los diseños de las interfaces. Los cuales van desde la **Fig. 39** hasta la **Fig. 61**.

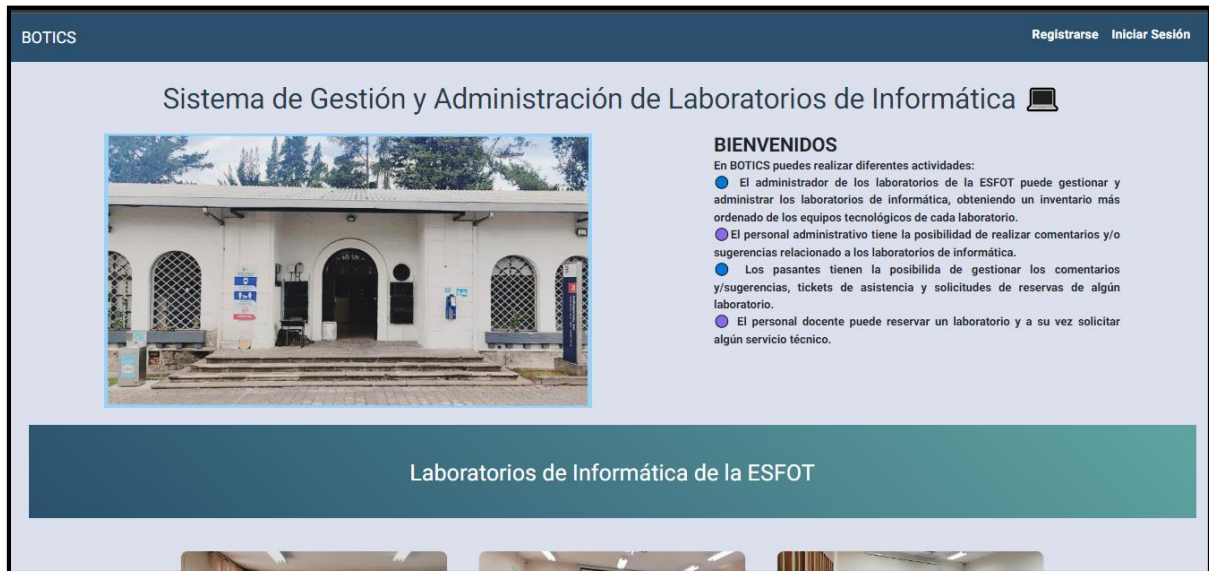


Fig. 39: Visualizar página informativa.



Fig. 40: Registrarse.

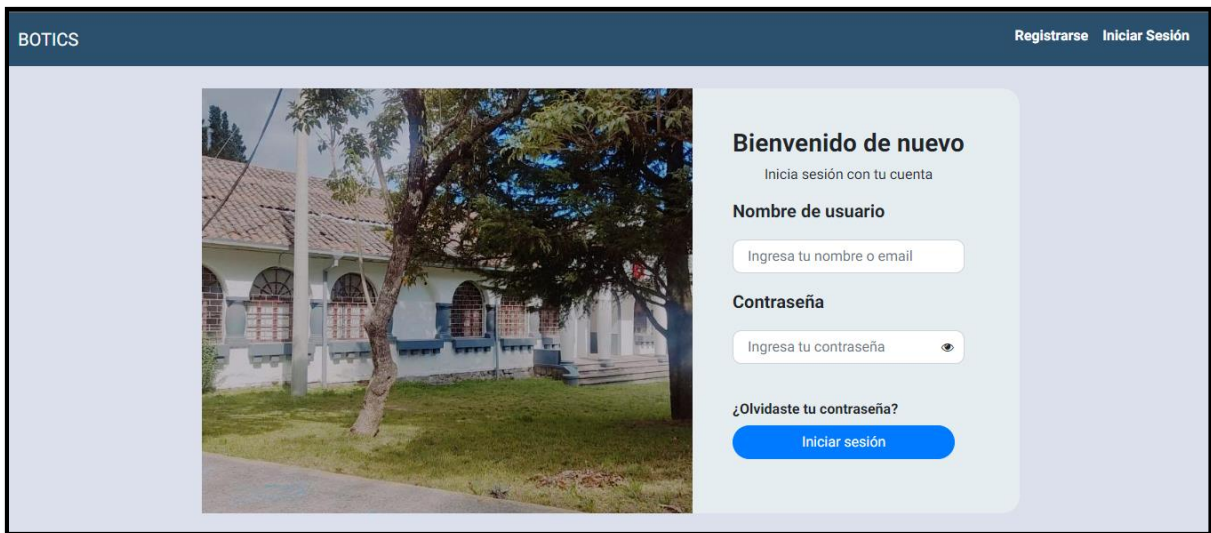


Fig. 41: Inicio de sesión.

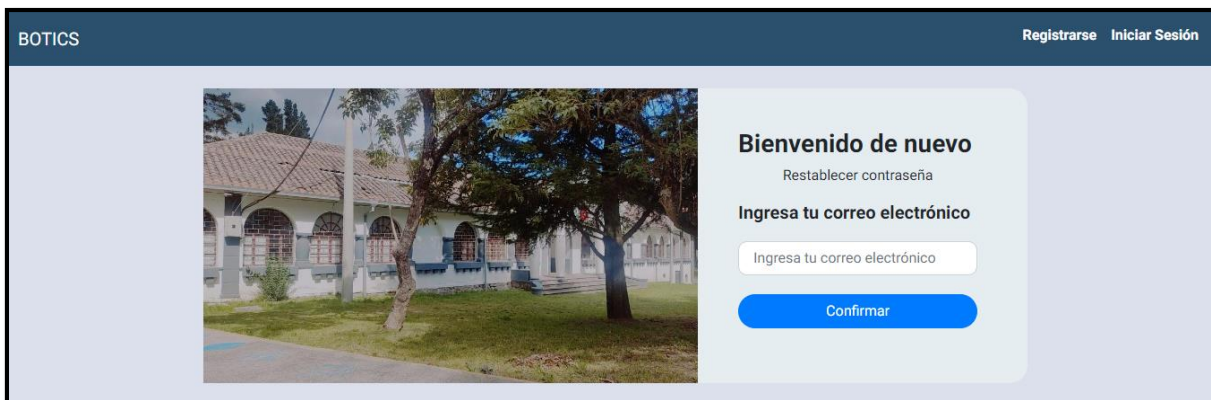


Fig. 42: Recuperar contraseña.

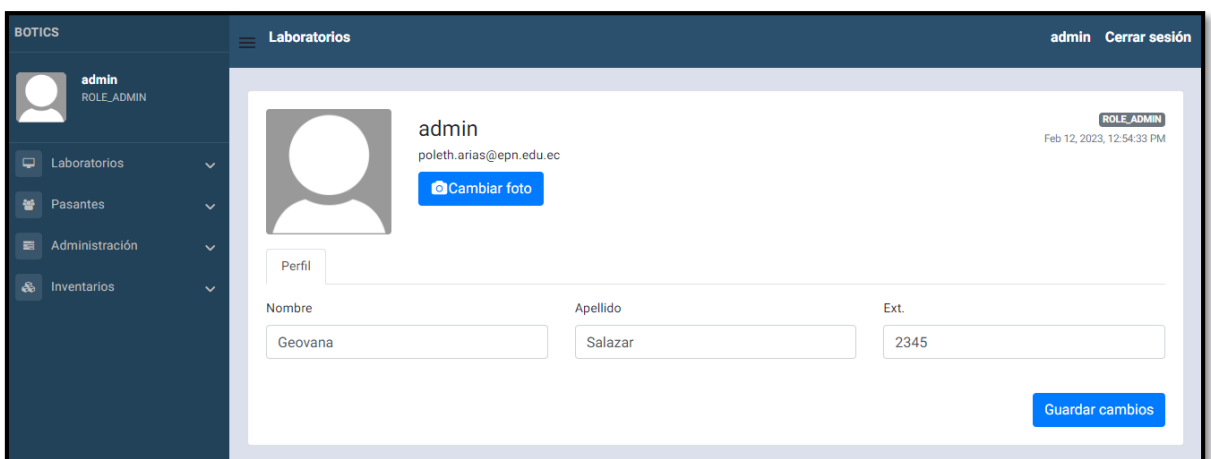


Fig. 43: Módulo perfil administrador.

Este módulo te permite:
Visualizar los equipos asignados a este laboratorio y también puedes registrar nuevos equipos.

Buscar computadora

Host Name	Serial del CPU	Serial del Monitor	Código del bien	Activar/Desactivar	Acciones
1-21-4566	Ex-Juvshj	MON12192	CP539484	<input checked="" type="checkbox"/>	Mantenimiento/Dar de baja Cambio de laboratorio
1-21-8373	Ex-82753	MON-9283	CP-5734	<input checked="" type="checkbox"/>	Mantenimiento/Dar de baja

Fig. 44: Módulo laboratorios-usuario administrador.

Este módulo te permite:
Visualizar una lista de los pasantes que han sido registrados y puedes habilitar o deshabilitar alguno de ellos, según se requiera:

Buscar pasante

Nombre de usuario	Email	Activar/Desactivar	Detalles
juan120	juanescobar12@gmail.com	<input checked="" type="checkbox"/>	<input type="button" value="O"/>
matt34	matiaszabrano23@gmail.com	<input checked="" type="checkbox"/>	<input type="button" value="O"/>
pasante	pasante@epn.edu.ec	<input checked="" type="checkbox"/>	<input type="button" value="O"/>
edu57	eduardoposo45@gmail.com	<input checked="" type="checkbox"/>	<input type="button" value="O"/>

Fig. 45: Módulo gestionar pasantes-usuario administrador.

Comentarios/sugerencias

Este módulo te permite:
Visualizar los comentarios y/o sugerencias realizadas por parte del personal docente y administrativo. También, puedes visualizar las respuestas de los comentarios y/o sugerencias.

Buscar comentario

Nombre	Apellido	Email	Rol	Detalles
Elias	Alcocer	administrativo@epn.edu.ec	ROLE_ADMINISTRATIVO	
Juan	Hidalgo	profesor@epn.edu.ec	ROLE_PROFESOR	
Juan	Hidalgo	profesor@epn.edu.ec	ROLE_PROFESOR	
Josefo	Paz	josepaz568@gmail.com	ROLE_PROFESOR	

Fig. 46: Módulo visualizar comentarios y/u sugerencias-usuario administrador.

Tickets de asistencia

Este módulo te permite:
Visualizar los tickets de asistencia solicitados por parte del personal docente y administrativo. También, puedes visualizar las respuestas de los tickets de asistencia.

Buscar tickets

Nombre	Apellido	Email	Rol	Acciones
Elias	Alcocer	administrativo@epn.edu.ec	ROLE_ADMINISTRATIVO	
Juan	Hidalgo	profesor@epn.edu.ec	ROLE_PROFESOR	
Josefo	Paz	josepaz568@gmail.com	ROLE_PROFESOR	

Fig. 47: Módulo visualizar tickets de asistencia usuario administrador.

Reservas

Este módulo te permite:
Visualizar las reservas de laboratorios enviadas por parte del personal docente.

Buscar reservas

Nombre	Apellido	Email	Rol	Acciones
Juan	Hidalgo	profesor@epn.edu.ec	ROLE_PROFESOR	
Josefo	Paz	josepaz568@gmail.com	ROLE_PROFESOR	
Juan	Hidalgo	profesor@epn.edu.ec	ROLE_PROFESOR	

Fig. 48: Módulo visualizar reservas de laboratorios usuario administrador.

LABOTICS Laboratorios admin [Cerrar sesión](#)

admin
ROLE_ADMIN

- Laboratorios
- Pasantes
- Administración
- Inventarios
 - Computadoras
 - Reporte
 - Historial

Este módulo te permite:
Visualizar los equipos registrados y también puedes añadir nuevos equipos y realizar otras acciones, tales como: dar de baja algún equipo.

Buscar computadora

Host Name	Serial del CPU	Serial del Monitor	Código del bien	Activar/Desactivar	Acciones
1-21-4566	Ex-Juvshj	MON12192	CP539484	<input checked="" type="checkbox"/>	 Mantenimiento/Dar de baja Cambio de laboratorio
1-21-8373	Ex-82753	MON-9283	CP-5734	<input checked="" type="checkbox"/>	 Mantenimiento/Dar

Fig. 49: Modulo gestionar inventarios usuario administrador.

LABOTICS Laboratorios admin [Cerrar sesión](#)

admin
ROLE_ADMIN

- Laboratorios
- Pasantes
- Administración
- Inventarios
 - Computadoras
 - Reporte
 - Historial

Este módulo te permite:
Visualizar el reporte de las computadoras ingresadas.

Computadoras Historial

Buscar computadora

2/12/23, 8:51 PM

Host Name	Serial del CPU	Serial del Monitor	Modelo	Procesador	RAM	Disco Duro
1-21-4566	Ex-Juvshj	MON12192	Lenovo	Intel core I3	8gb	120
1-21-8373	Ex-82753	MON-9283	Lenovo	Intel core I3	8gb	120

Fig. 50: Módulo reporte-usuario administrador.

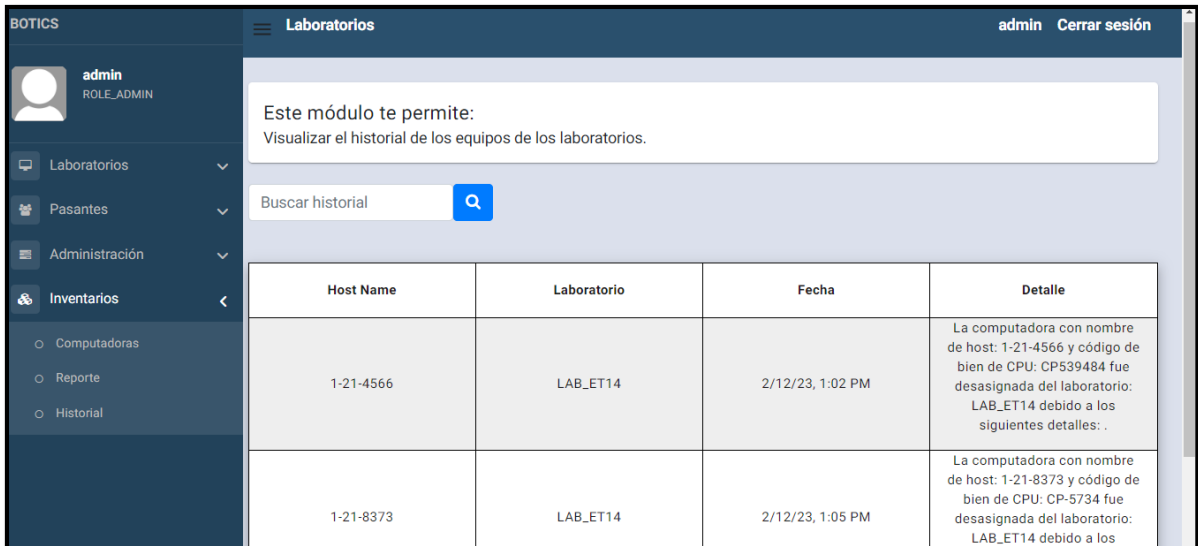


Fig. 51: Módulo historial usuario administrador.

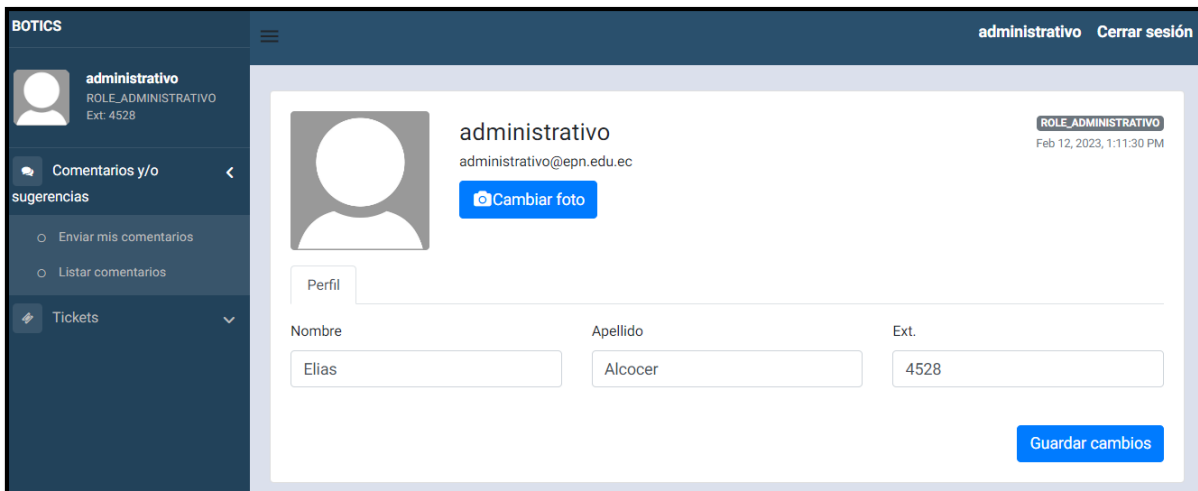


Fig. 52: Módulo usuario administrativo.

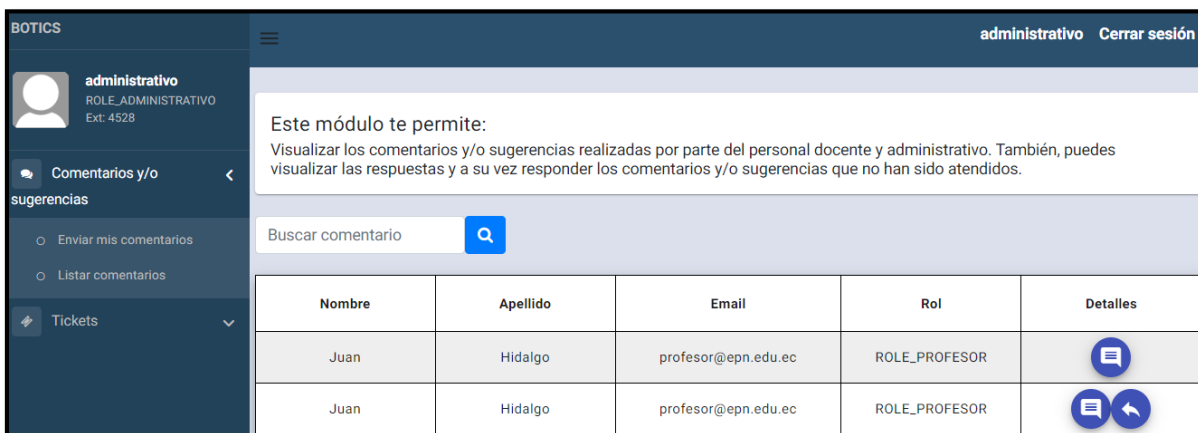


Fig. 53: Módulo gestionar comentarios usuario administrativo.

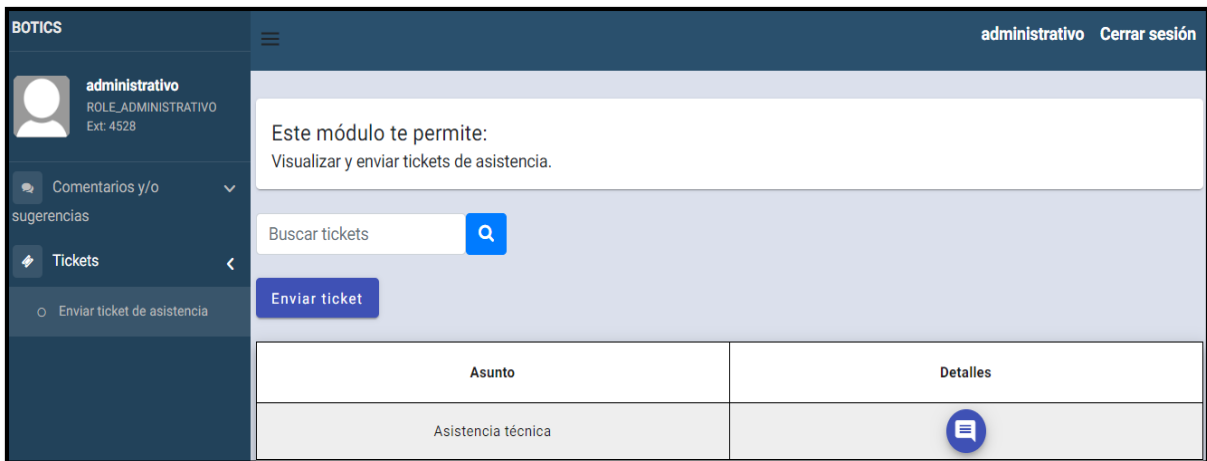


Fig. 54: Modulo enviar ticket de asistencia-usuario administrativo.

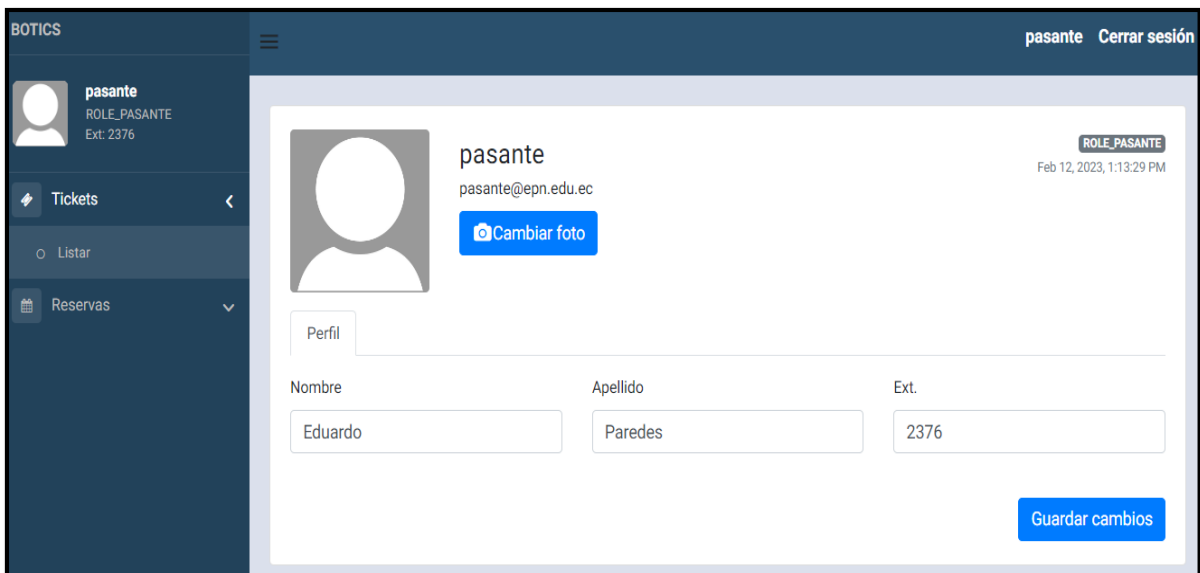


Fig. 55: Módulo usuario pasante.

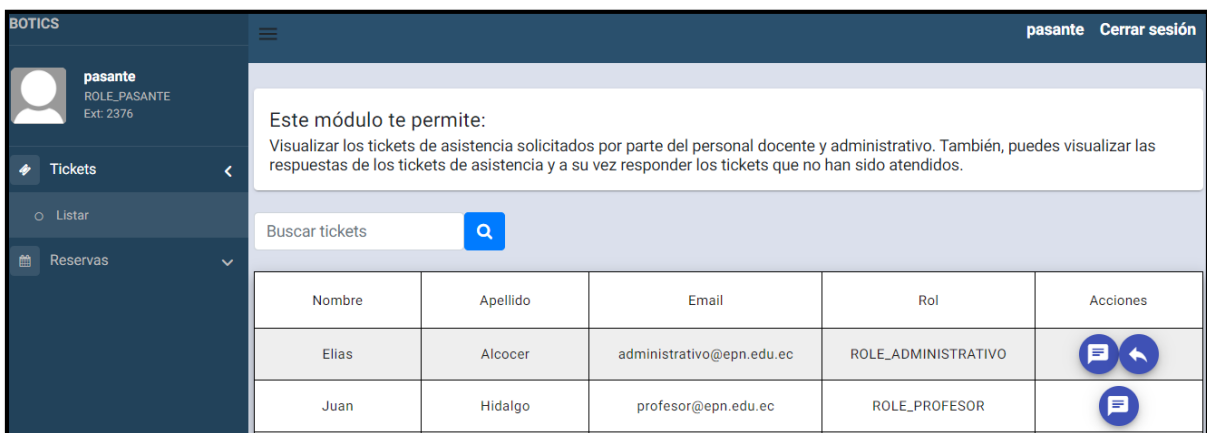


Fig. 56: Módulo gestionar tickets de asistencia usuario pasante.

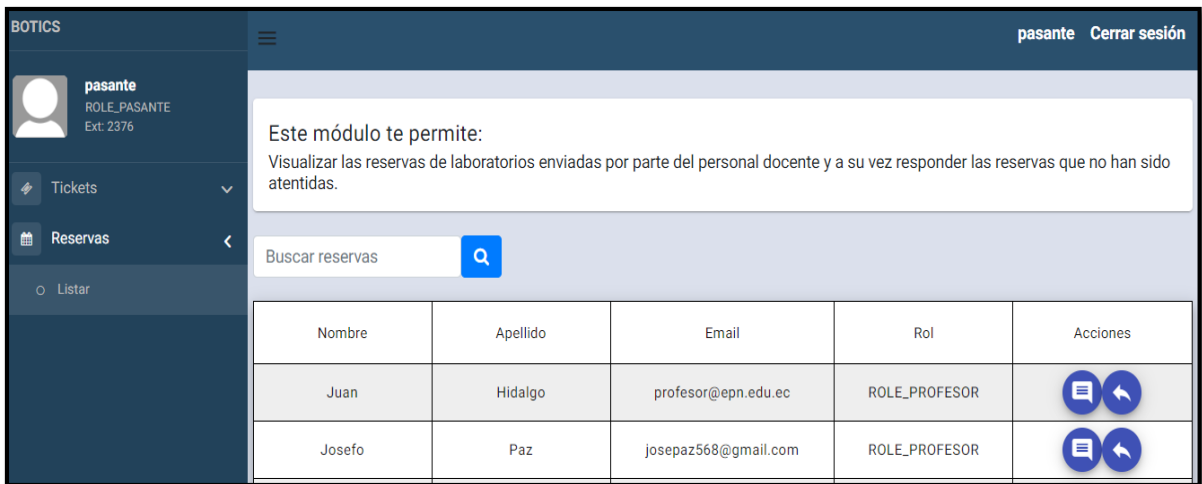


Fig. 57: Módulo gestionar reservas-usuario pasante.

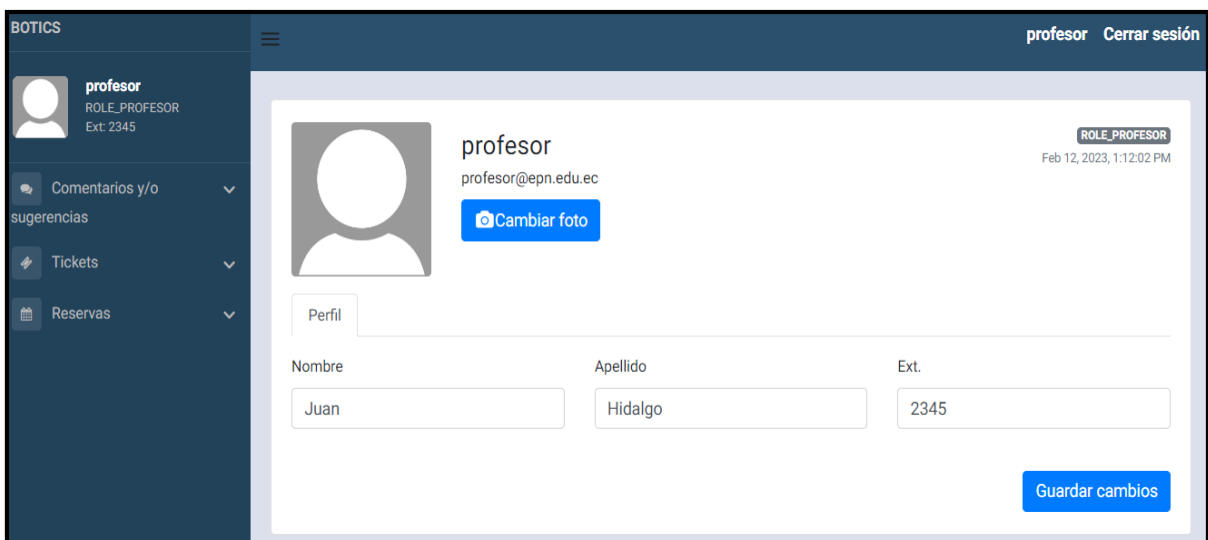


Fig. 58: Módulo usuario docente.

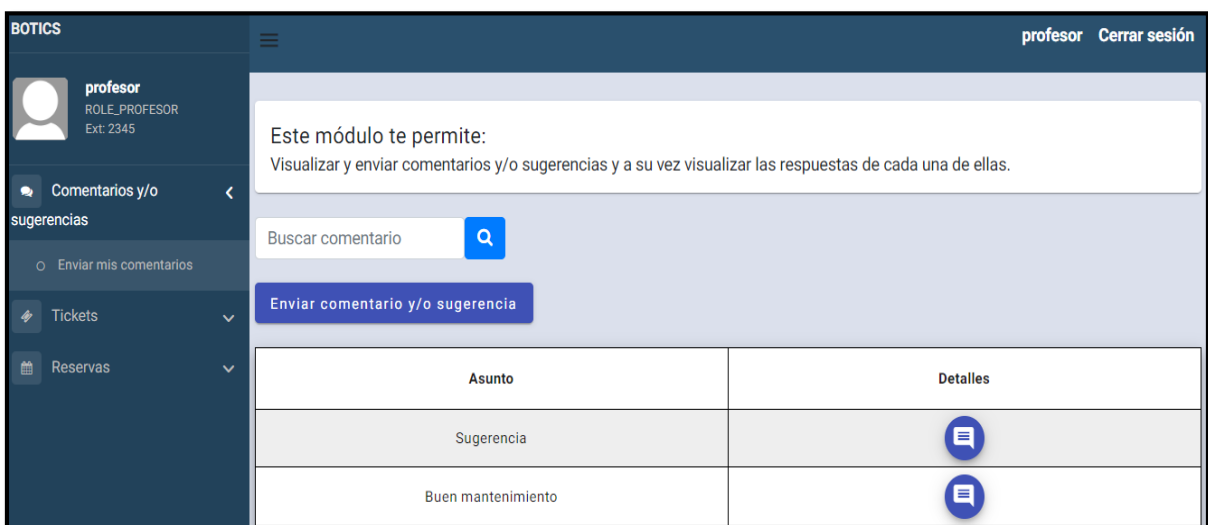


Fig. 59: Módulo enviar comentarios y/o sugerencias-usuario docente.

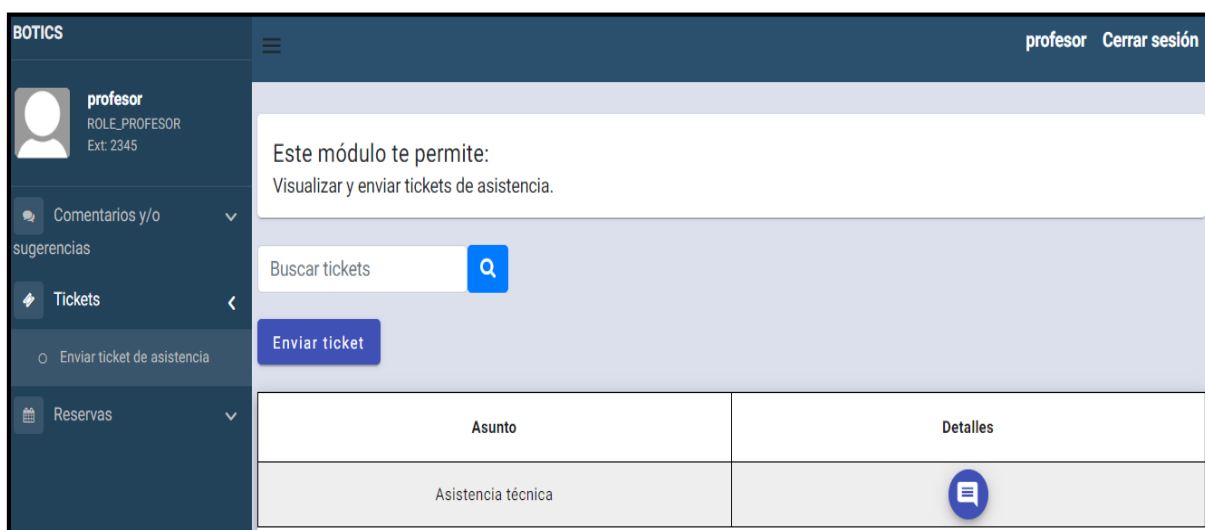


Fig. 60: Módulo enviar tickets de asistencia-usuario docente.

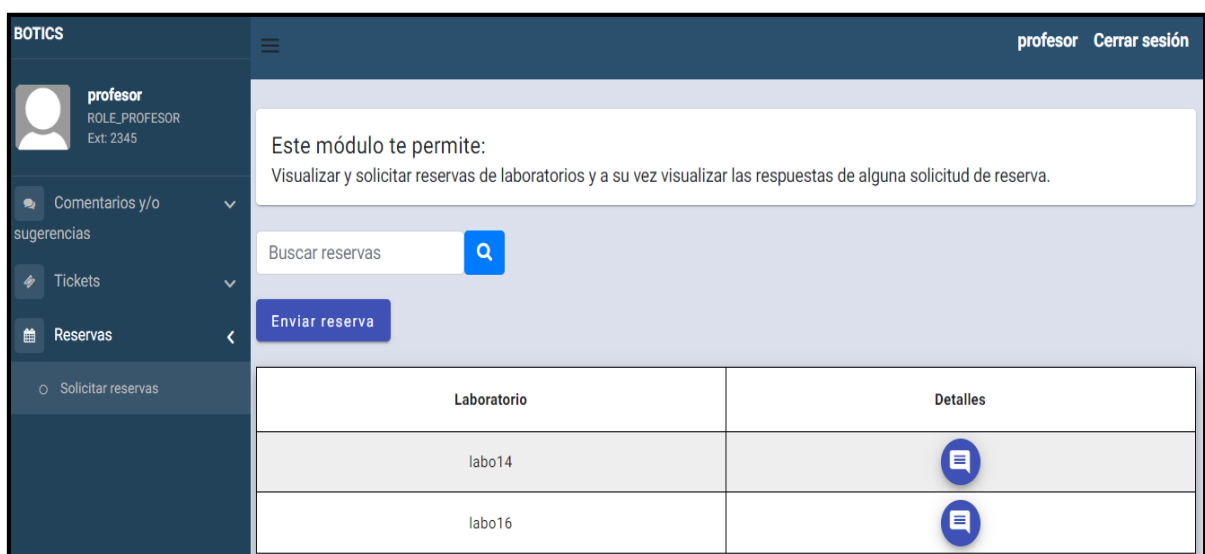


Fig. 61: Módulo solicitar reservas de laboratorios-usuario docente.

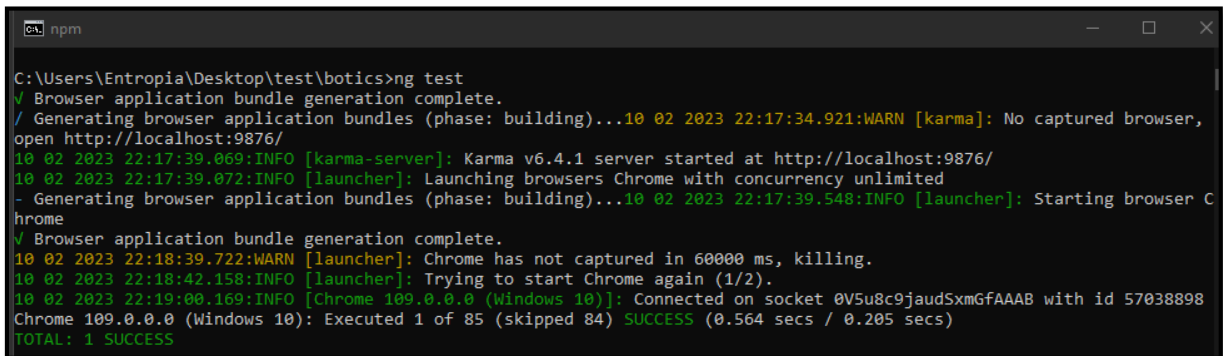
Pruebas

Finalizada la etapa de codificación, se ha realizado la implementación de diversas pruebas, tales como: unitarias, compatibilidad y aceptación con la finalidad de verificar la eficacia del código de cada módulo del componente *frontend*.

Pruebas unitarias

En la **Fig. 62** se observa el comando en karma el cual es de gran utilidad para la realizar pruebas unitarias, en este caso se ha implementado en el código de las principales

funcionalidades del *frontend*, donde se verifica que el módulo pasa el *testing* sin error alguno, como se observa desde la **Fig. 63** hasta la **Fig. 68**.



```
C:\Users\Entropia\Desktop\test\botics>ng test
√ Browser application bundle generation complete.
/ Generating browser application bundles (phase: building)...10 02 2023 22:17:34.921:WARN [karma]: No captured browser,
open http://localhost:9876/
10 02 2023 22:17:39.069:INFO [karma-server]: Karma v6.4.1 server started at http://localhost:9876/
10 02 2023 22:17:39.072:INFO [launcher]: Launching browsers Chrome with concurrency unlimited
- Generating browser application bundles (phase: building)...10 02 2023 22:17:39.548:INFO [launcher]: Starting browser C
hrome
√ Browser application bundle generation complete.
10 02 2023 22:18:39.722:WARN [launcher]: Chrome has not captured in 60000 ms, killing.
10 02 2023 22:18:42.158:INFO [launcher]: Trying to start Chrome again (1/2).
10 02 2023 22:19:00.169:INFO [Chrome 109.0.0.0 (Windows 10)]: Connected on socket 0V5u8c9jaud5xmGfAAAB with id 57038898
Chrome 109.0.0.0 (Windows 10): Executed 1 of 85 (skipped 84) SUCCESS (0.564 secs / 0.205 secs)
TOTAL: 1 SUCCESS
```

Fig. 62: Prueba unitaria Comando para su ejecución.



```
import { TestBed } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { AppComponent } from './app.component';
import { HttpClientModule } from '@angular/common/http';

fdescribe('AppComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [
        RouterTestingModule, HttpClientModule
      ],
      declarations: [
        AppComponent
      ],
    }).compileComponents();
  });

  fit('Funcionalidad entre componentes', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });
});
```

Fig. 63: Prueba unitaria funcionalidad entre componentes.

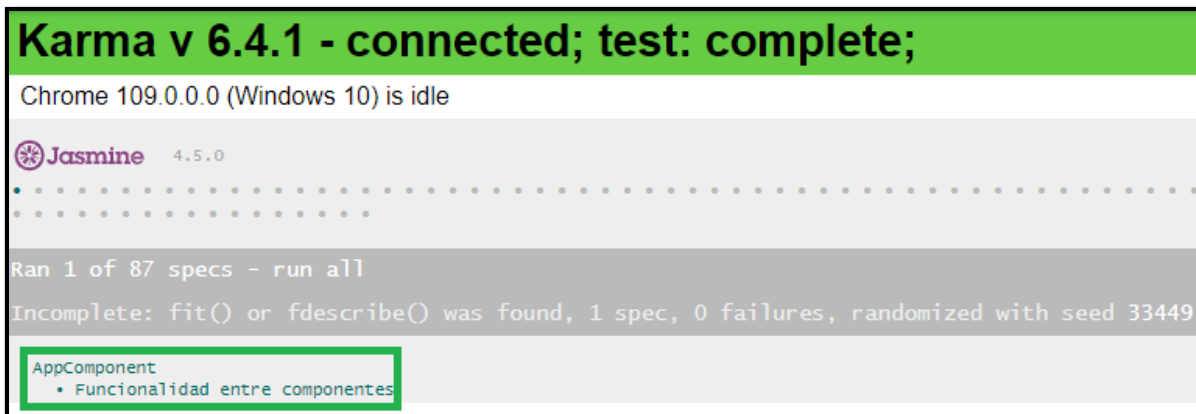


Fig. 64: Resultado de prueba funcionalidad entre Componentes.

```
import { ProfileComponent } from './profile.component';
import { HttpClientModule } from '@angular/common/http';
import { MatDialogModule } from '@angular/material/dialog';
import { FormsModule } from '@angular/forms';
import { ActivatedRoute, RouterModule } from '@angular/router';

fdescribe('ProfileComponent', () => {
  let component: ProfileComponent;
  let fixture: ComponentFixture<ProfileComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [HttpClientModule, MatDialogModule, FormsModule],
      providers: [{provide: ActivatedRoute,
        useValue: { snapshot: {params: {id: '24fkzrw3487943uf358lovd'}}}],
      declarations: [ ProfileComponent ]
    })
    .compileComponents();
    fixture = TestBed.createComponent(ProfileComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  fit('Visualizar perfil', () => {
    expect(component).toBeTruthy();
  });
});
```

Fig. 65: Prueba unitaria visualizar perfil.

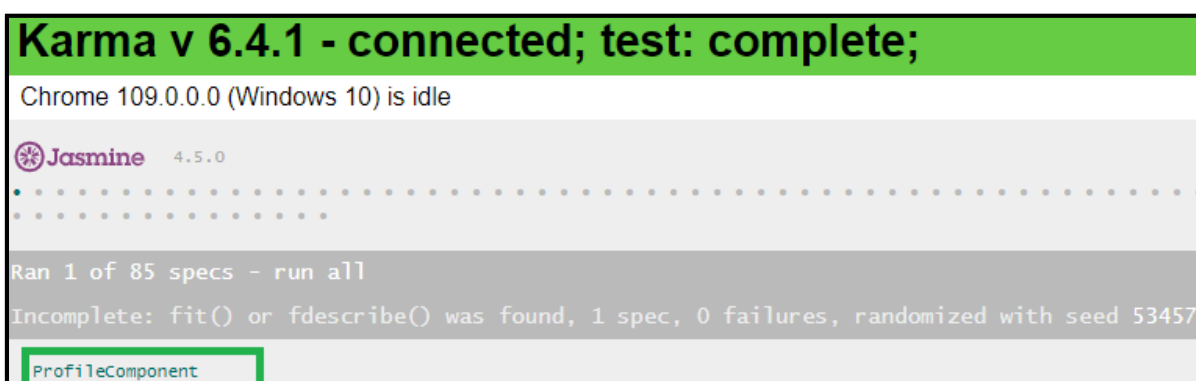


Fig. 66: Resultado de prueba visualizar perfil.


```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { HistoryComponent } from './history.component';
import { MatPaginatorModule } from '@angular/material/paginator';
import { MatSortModule } from '@angular/material/sort';
import { MatTableModule } from '@angular/material/table';
import { HttpClientModule } from '@angular/common/http';
import { MatCardModule } from '@angular/material/card';
import { FormsModule } from '@angular/forms';

fdescribe('HistoryComponent', () => {
  let component: HistoryComponent;
  let fixture: ComponentFixture<HistoryComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports:[MatPaginatorModule, MatSortModule, MatTableModule, HttpClientModule, Mat
      declarations: [ HistoryComponent ]
    })
    .compileComponents();
    fixture = TestBed.createComponent(HistoryComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  fit('Visualizar historial', () => {
    expect(component).toBeTruthy();
  });
});

```

Fig. 67: Prueba unitaria visualizar historial.

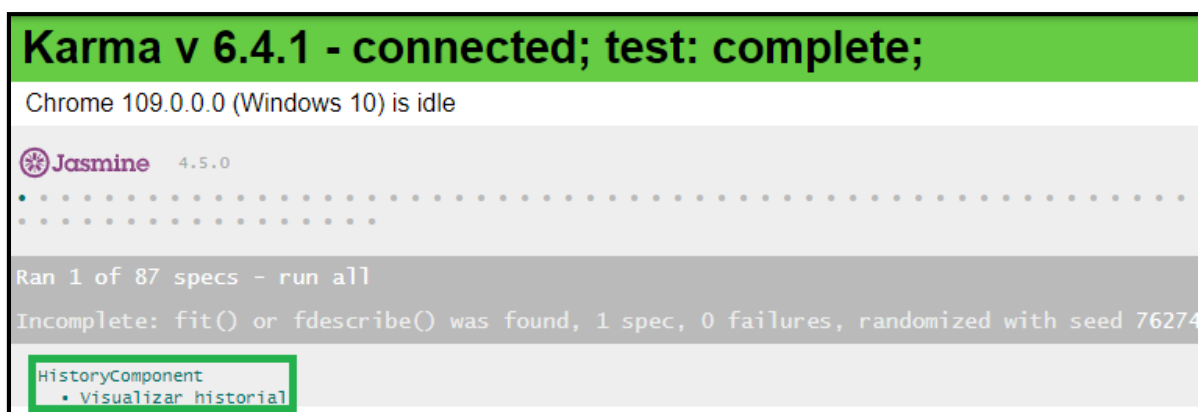


Fig. 68: Resultado de prueba visualizar historial.

Pruebas de compatibilidad

Esta prueba, en el *frontend* se centran en comprobar las diversas funcionalidades en los principales navegadores con el objetivo de visualizar los posibles problemas al momento de presentar el contenido al usuario.

De este modo, desde la **Fig. 69** hasta la **Fig. 110** se visualizan los resultados que se han logrado obtener en los diferentes navegadores.



Fig. 69: Página informativa vista en el navegador Opera.

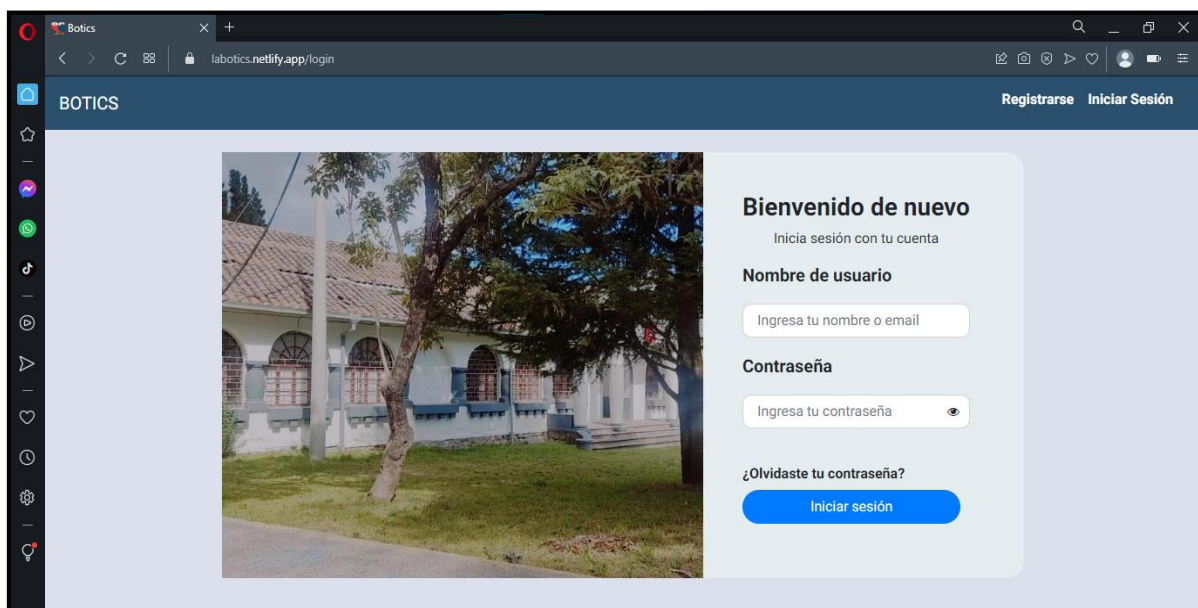


Fig. 70: Página de Inicio de Sesión en el navegador Opera.

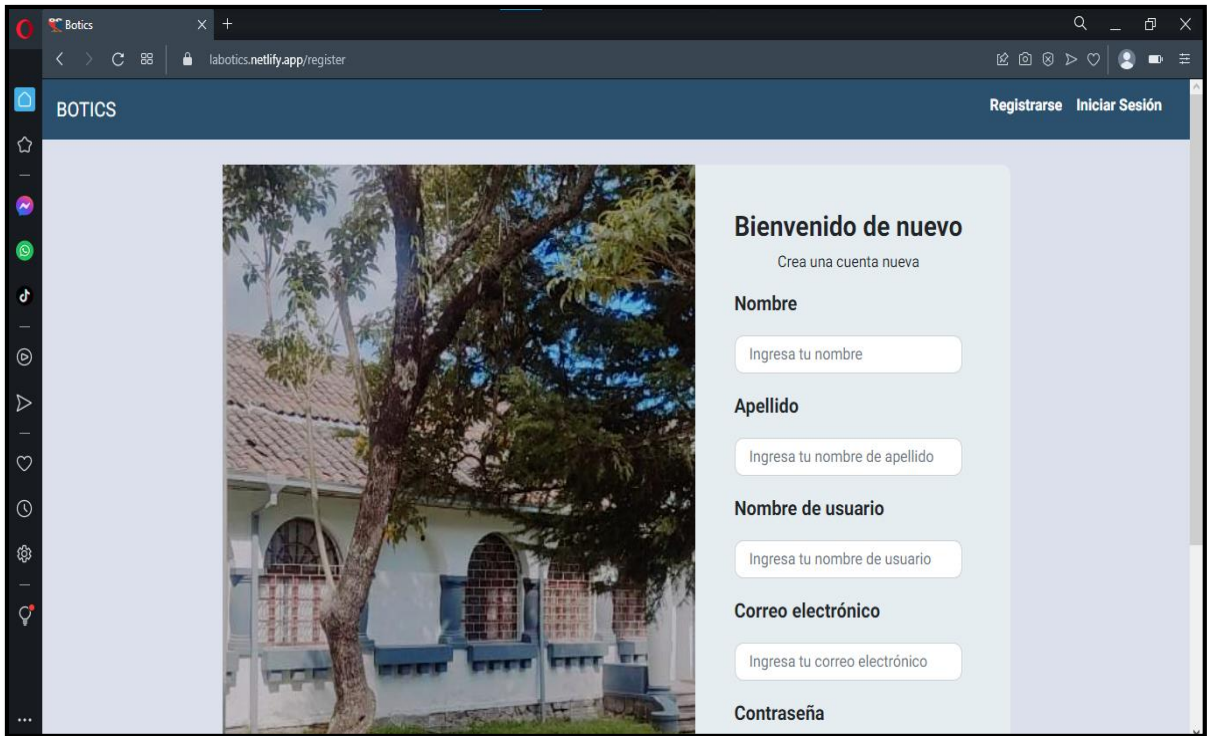


Fig. 71: Página de registro en el navegador Opera.

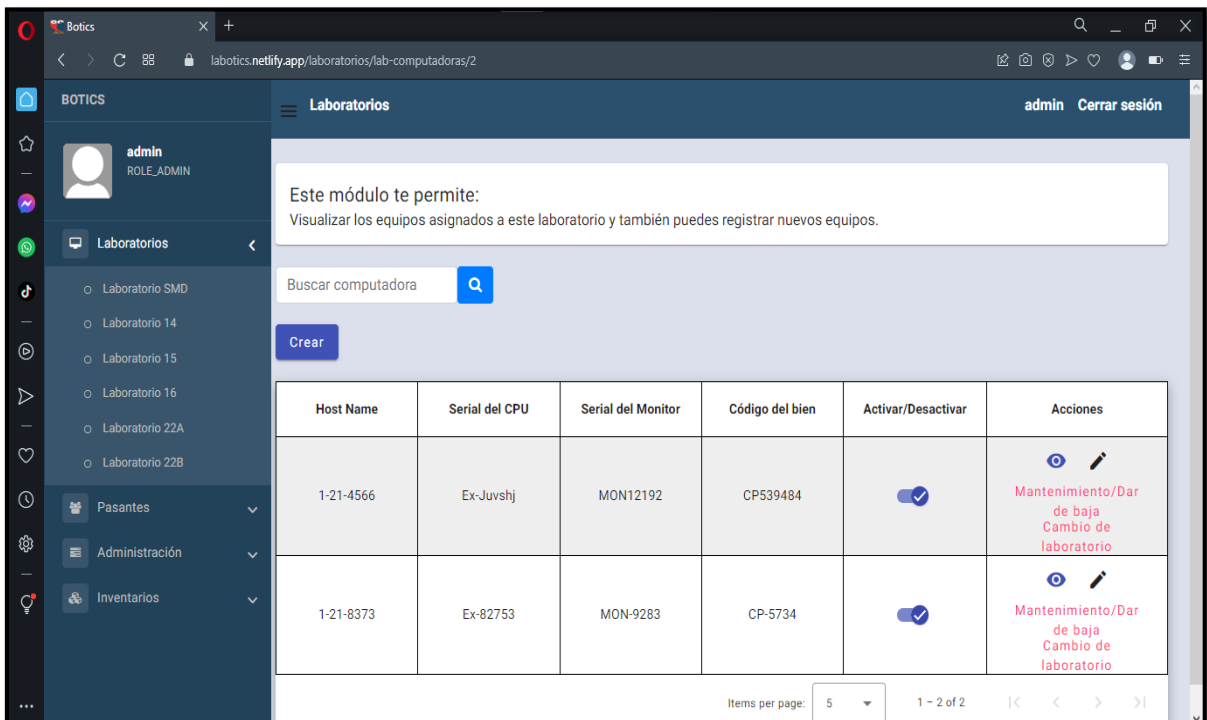


Fig. 72: Página laboratorios (usuario administrador) en el navegador Opera.

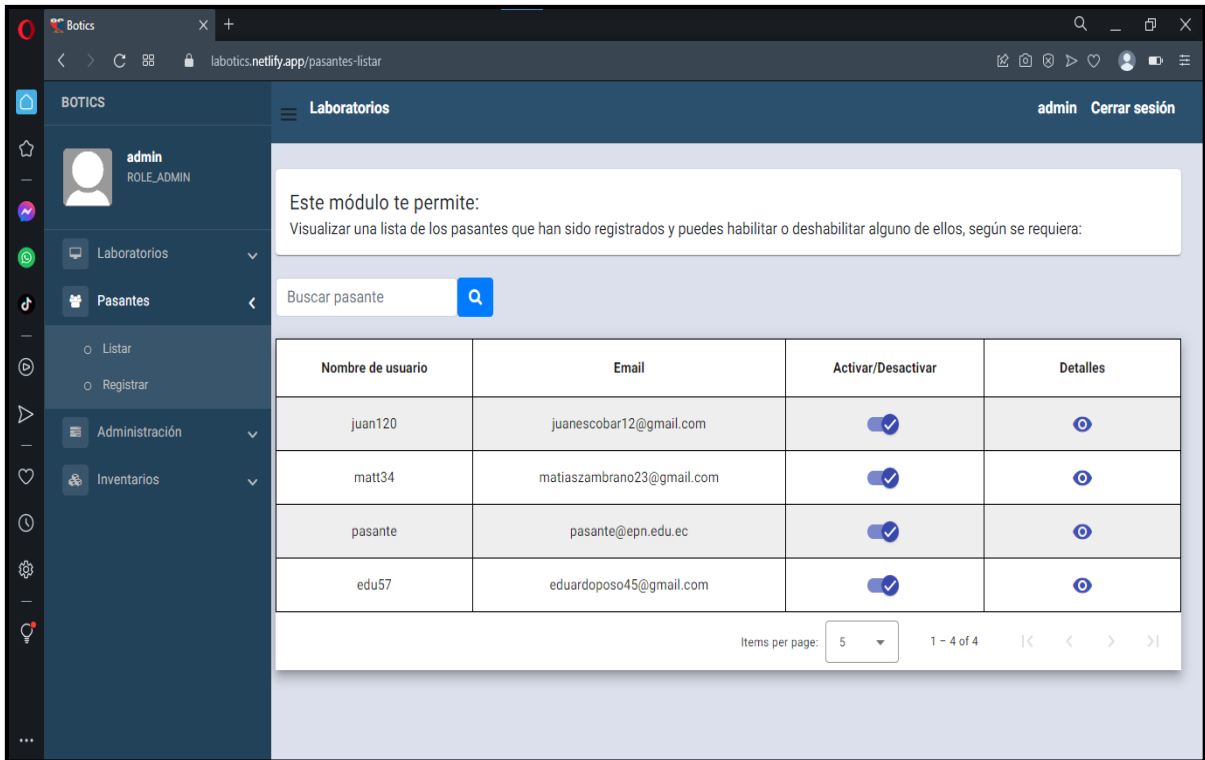


Fig. 73: Página gestionar pasantes (usuario administrador) en el navegador Opera.

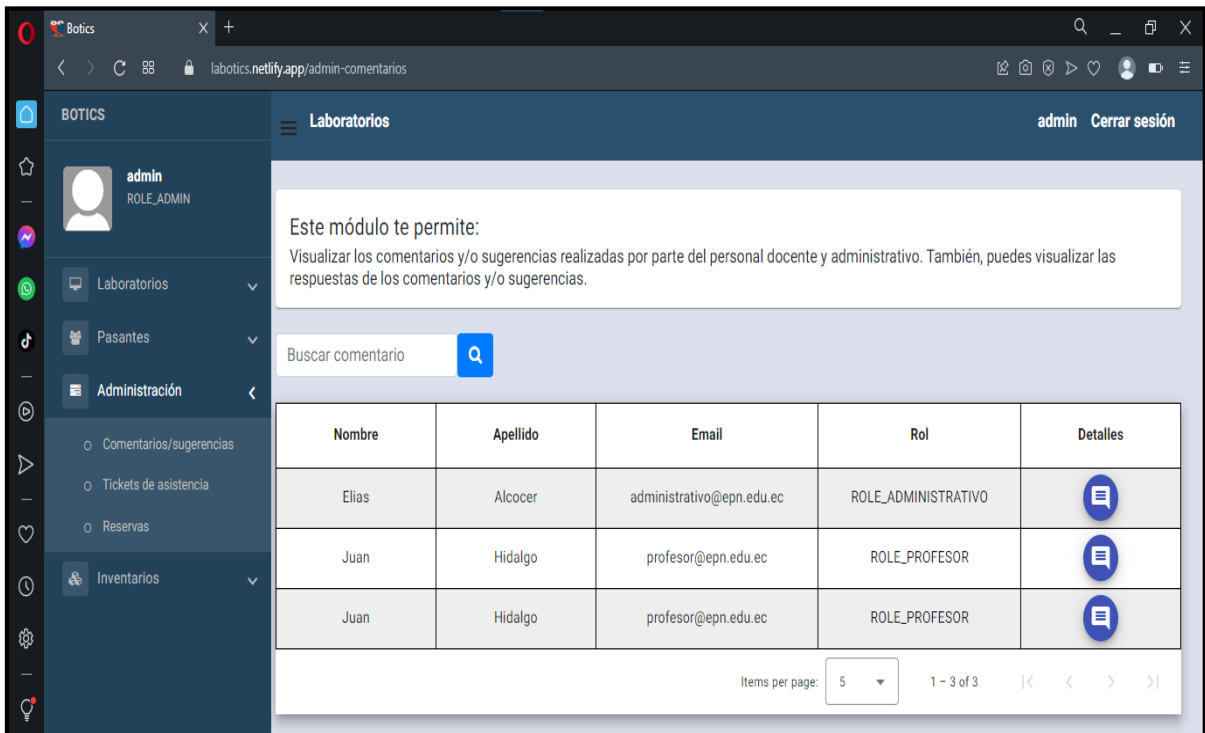


Fig. 74: Página administración (usuario administrador) en el navegador Opera.

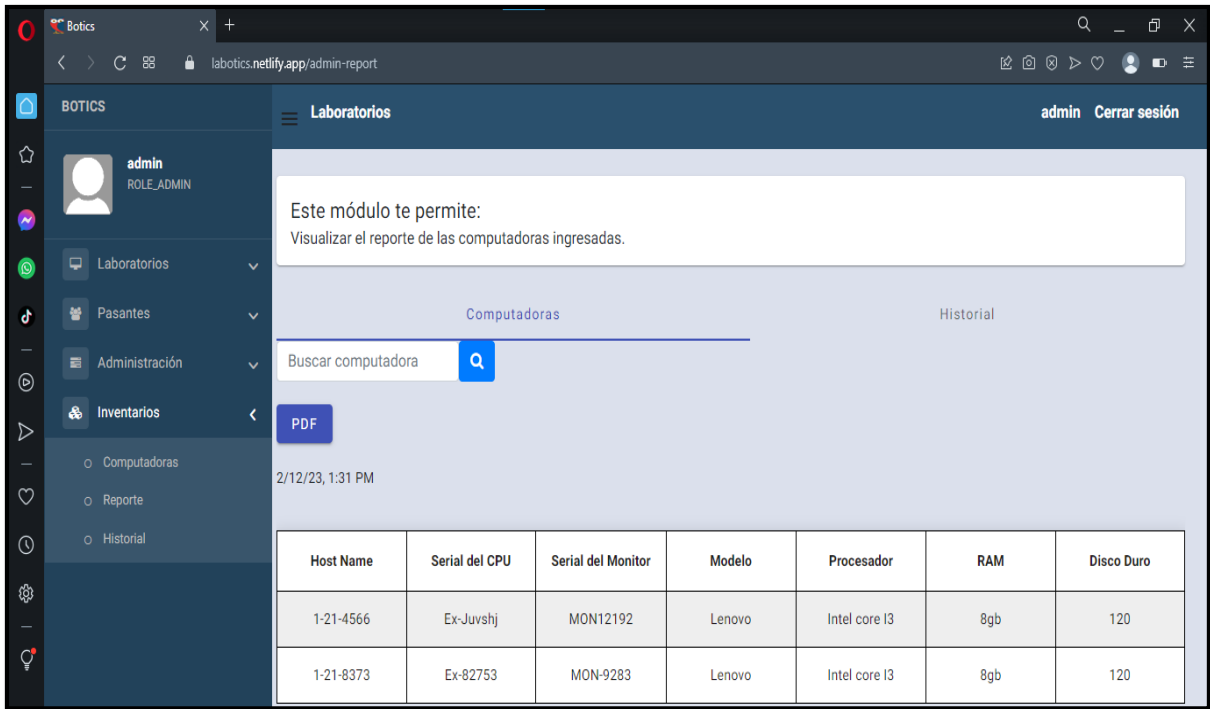


Fig. 75: Página inventarios (usuario administrador) en el navegador Opera.

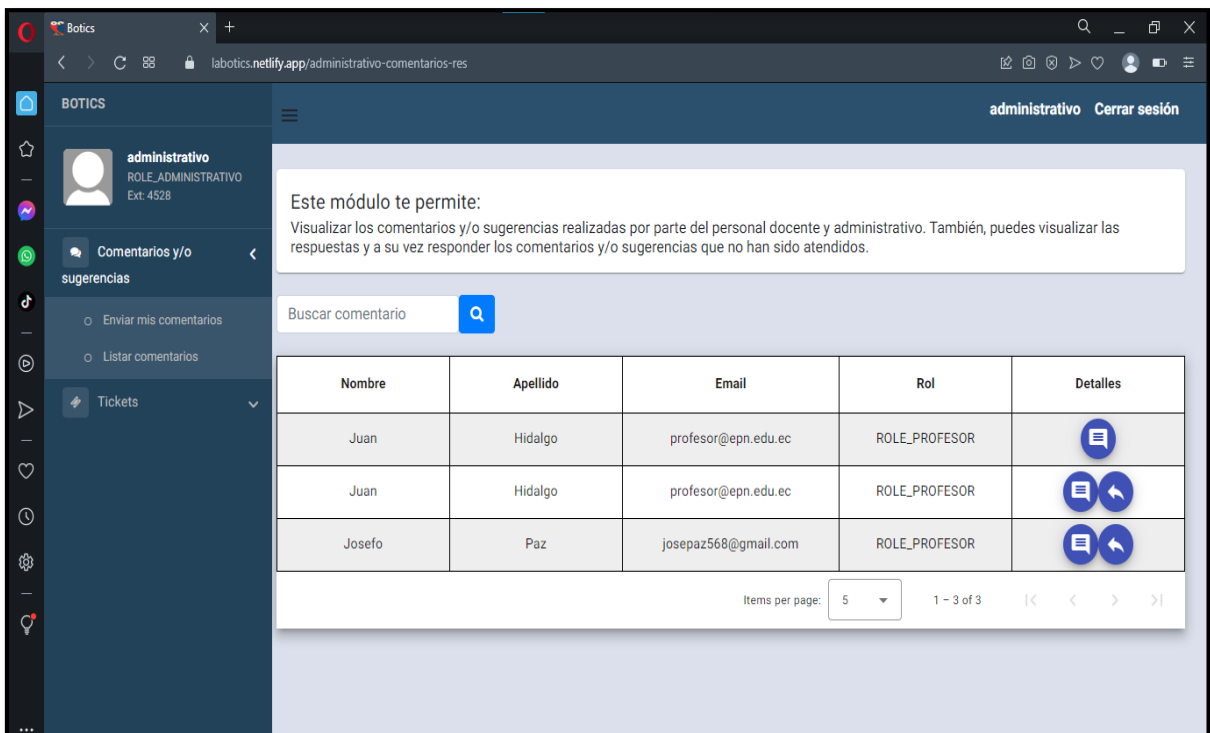


Fig. 76: Página gestionar comentarios y/o sugerencias (usuario administrativo) en el navegador Opera.

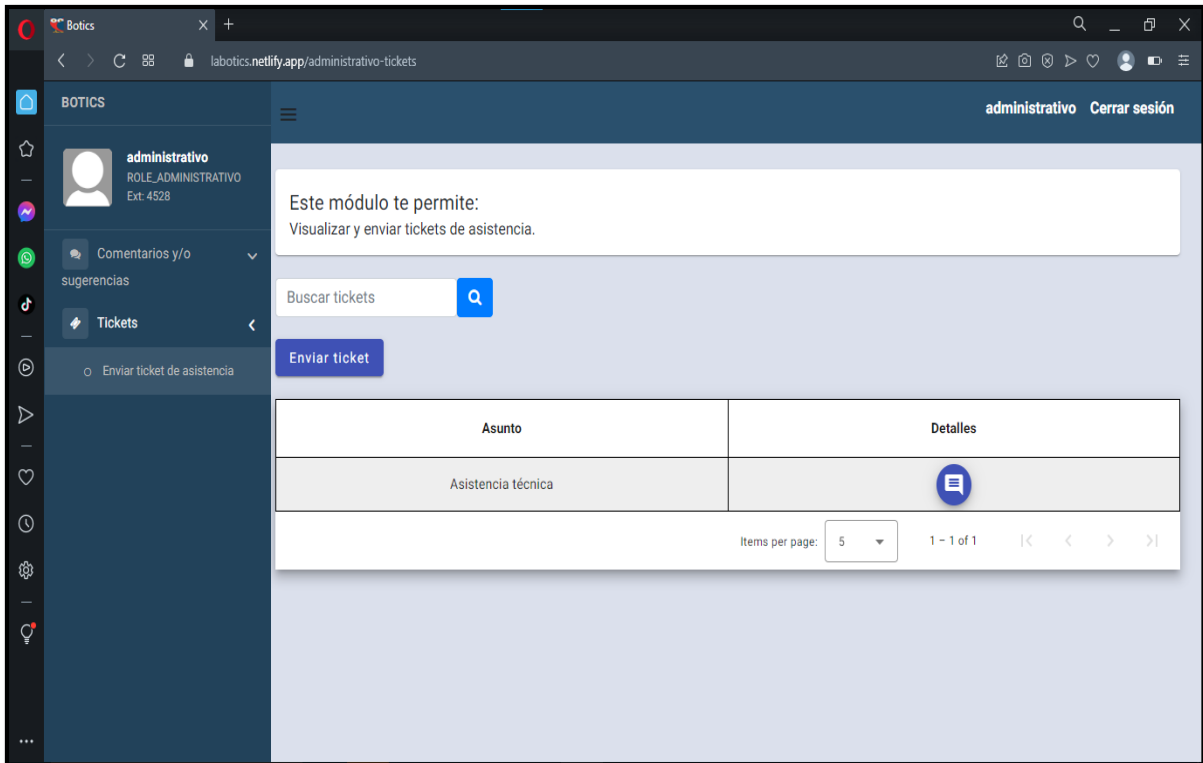


Fig. 77: Página tickets de asistencia (usuario administrativo) en el navegador Opera.

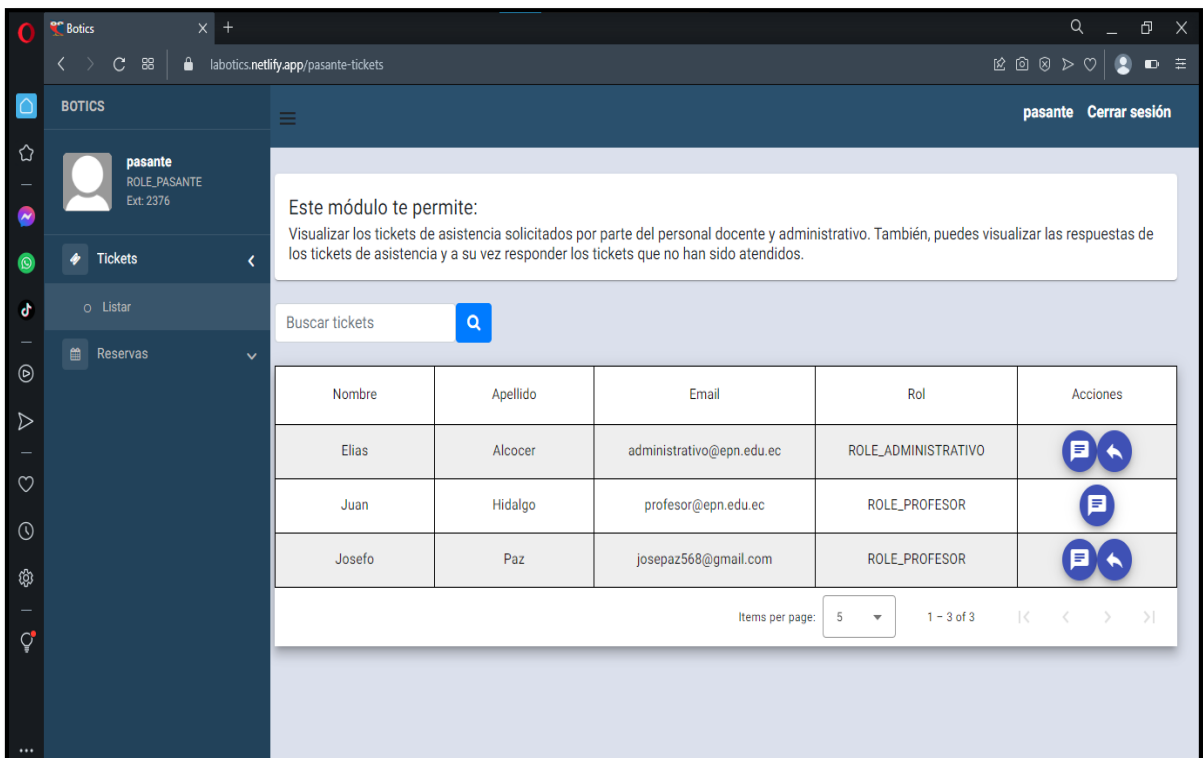


Fig. 78: Página tickets de asistencia (usuario pasante) en el navegador Opera.

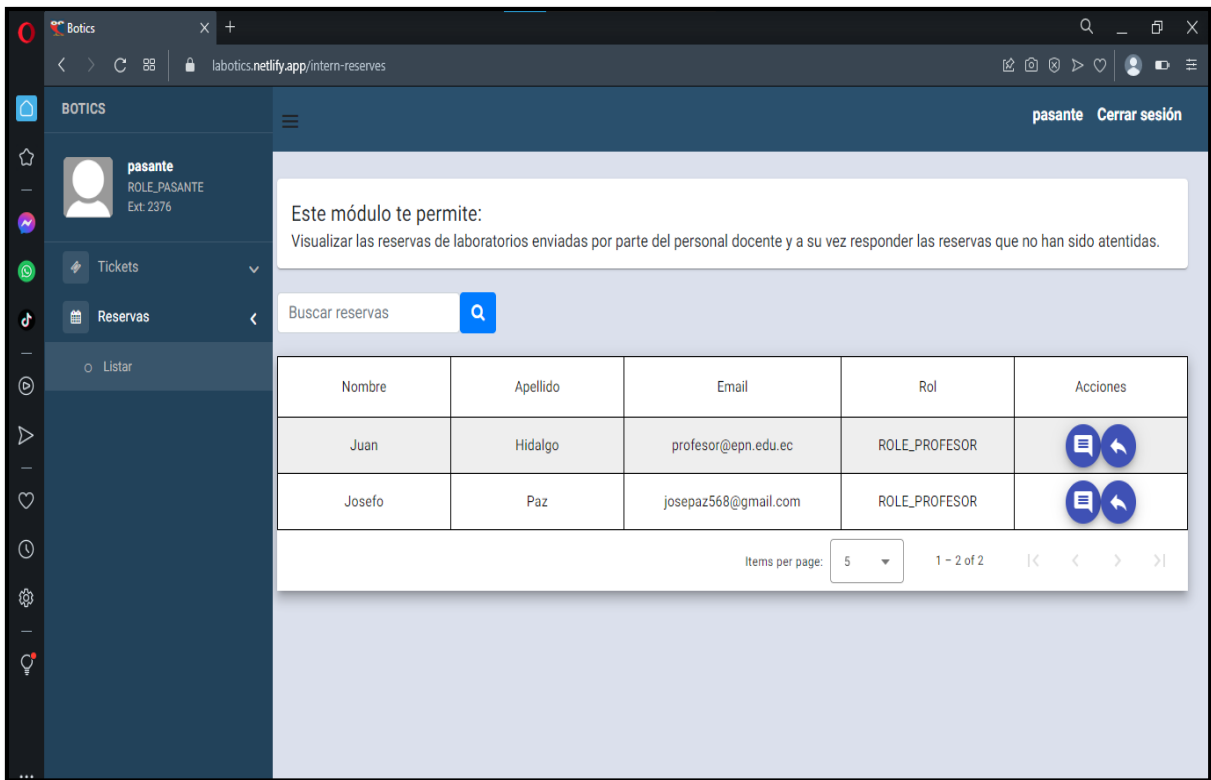


Fig. 79: Página gestionar reservas (usuario pasante) en el navegador Opera.

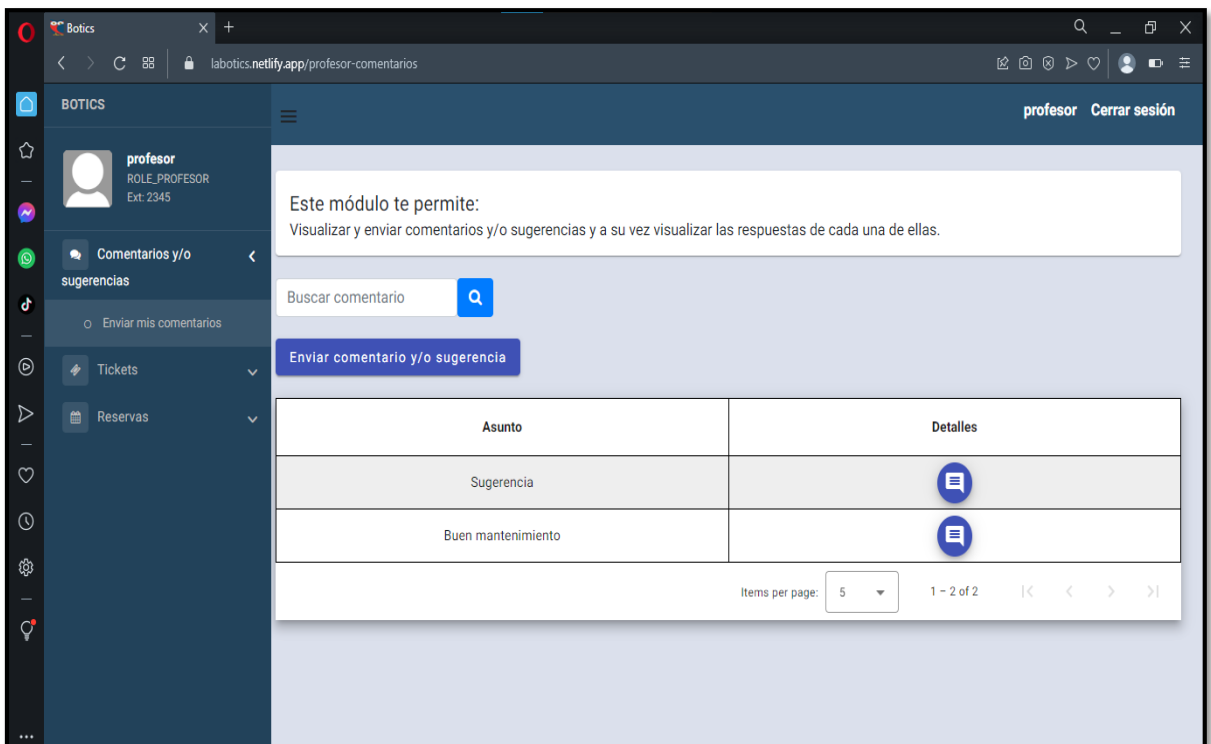


Fig. 80: Página comentarios y/o sugerencias (usuario docente) en el navegador Opera.

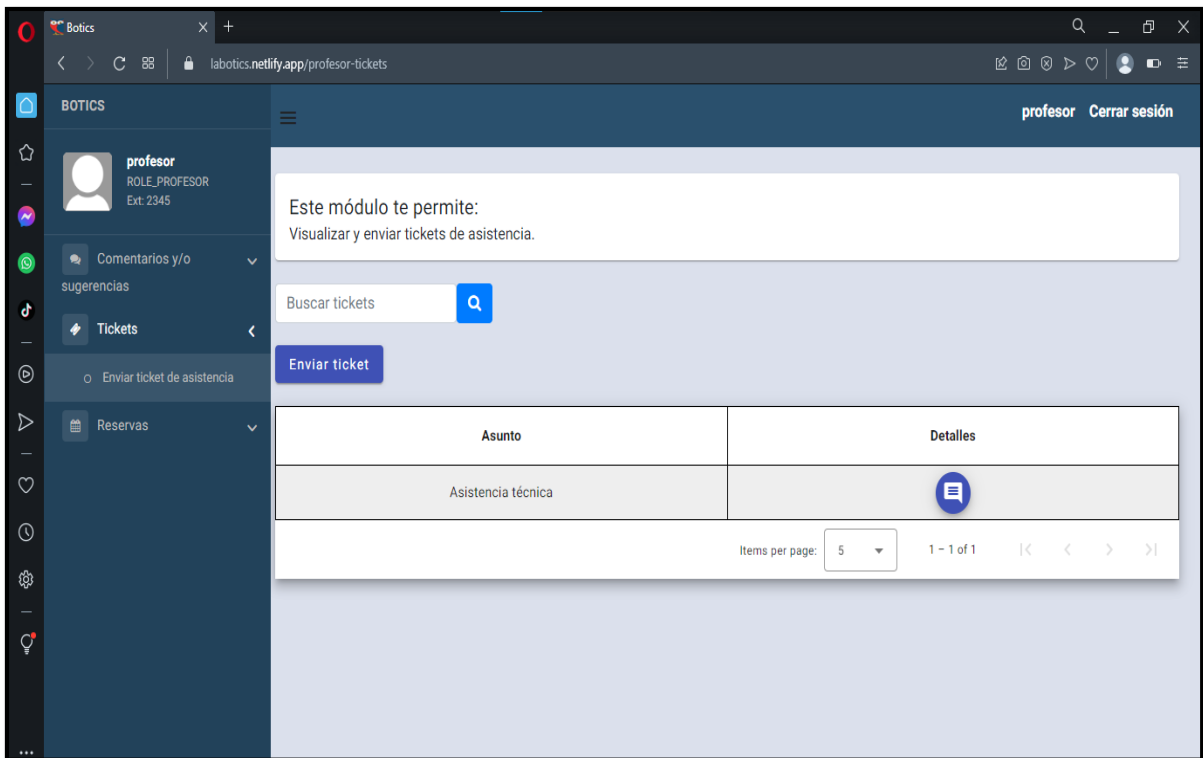


Fig. 81: Página de tickets de asistencia (usuario docente) en el navegador Opera.

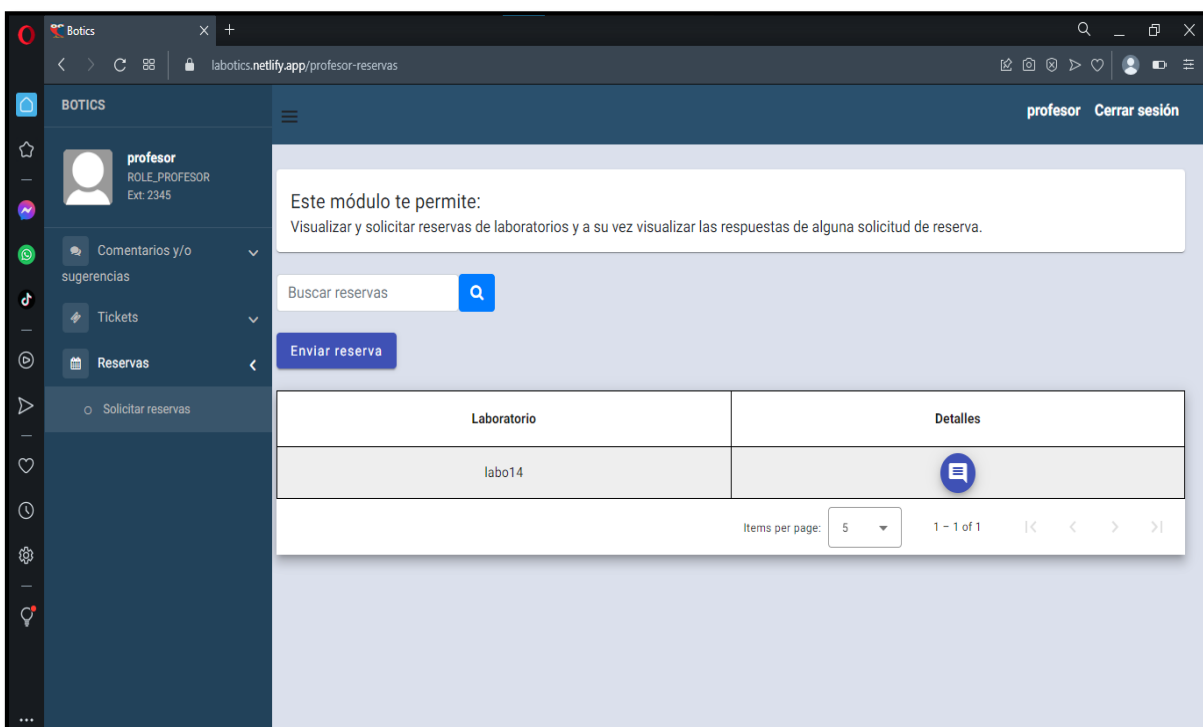


Fig. 82: Página de reservas (usuario docente) en el navegador Opera.

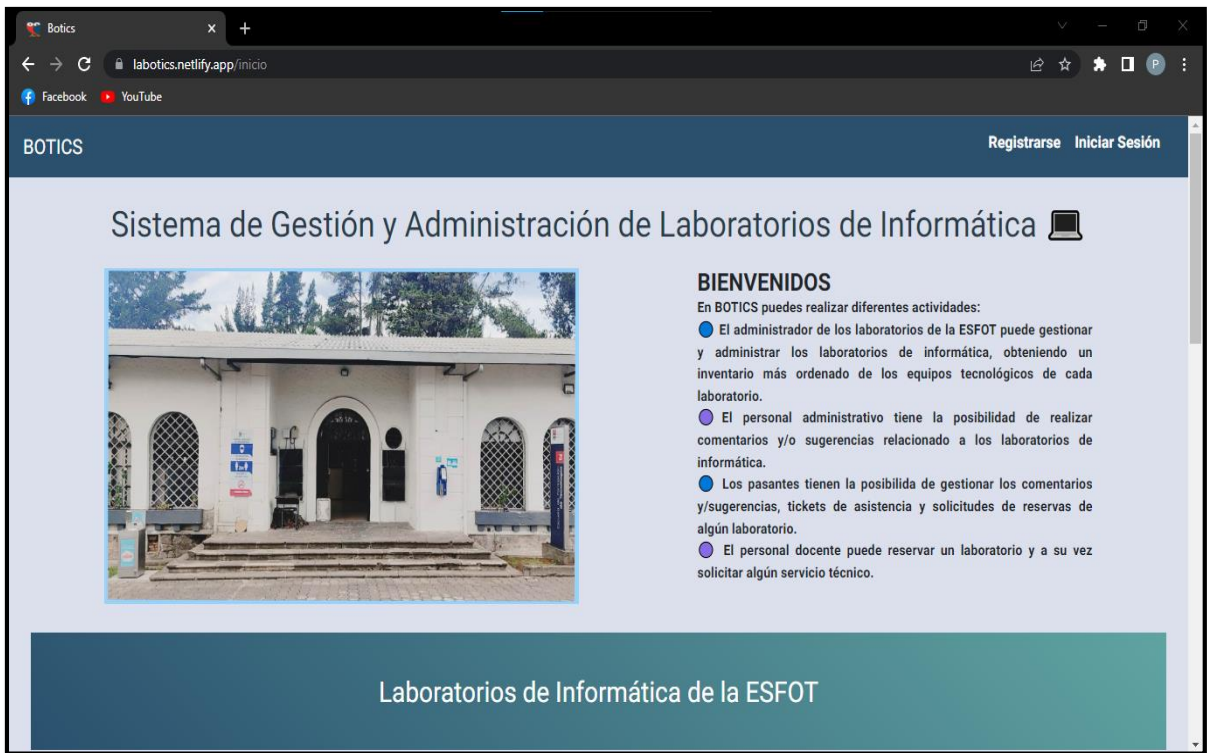


Fig. 83: Página informativa vista en el navegador *Google Chrome*.

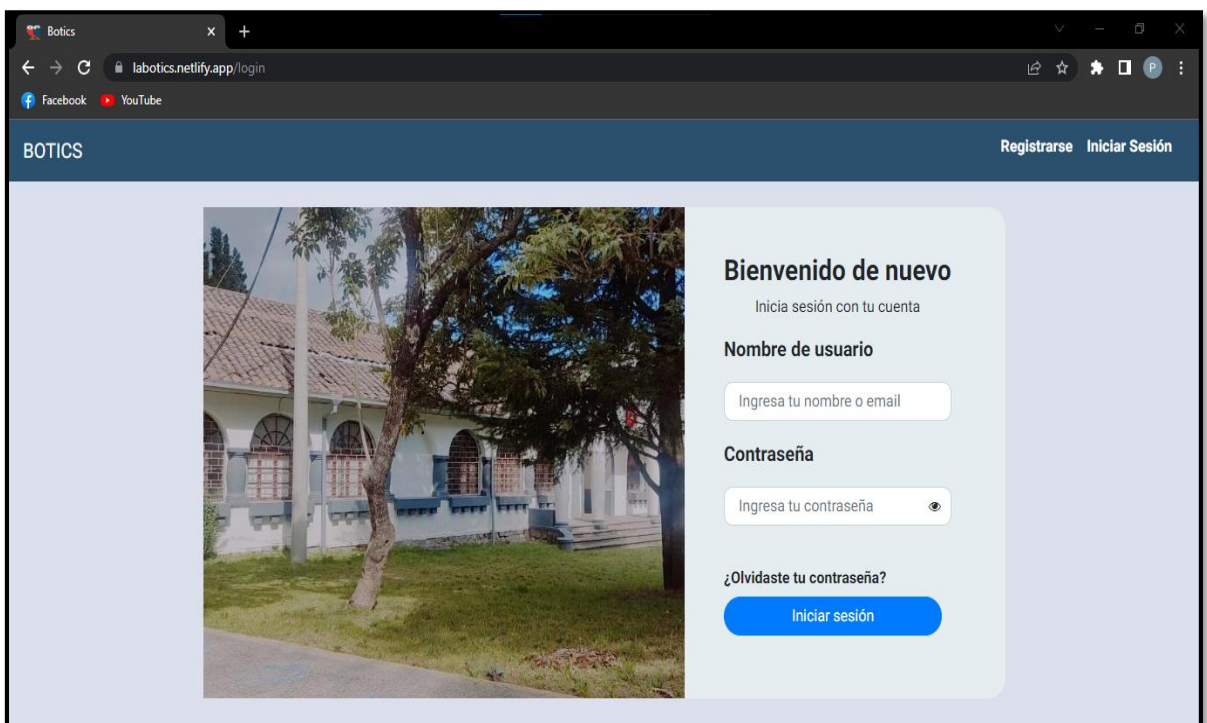


Fig. 84: Página de Inicio de Sesión en el navegador *Google Chrome*.

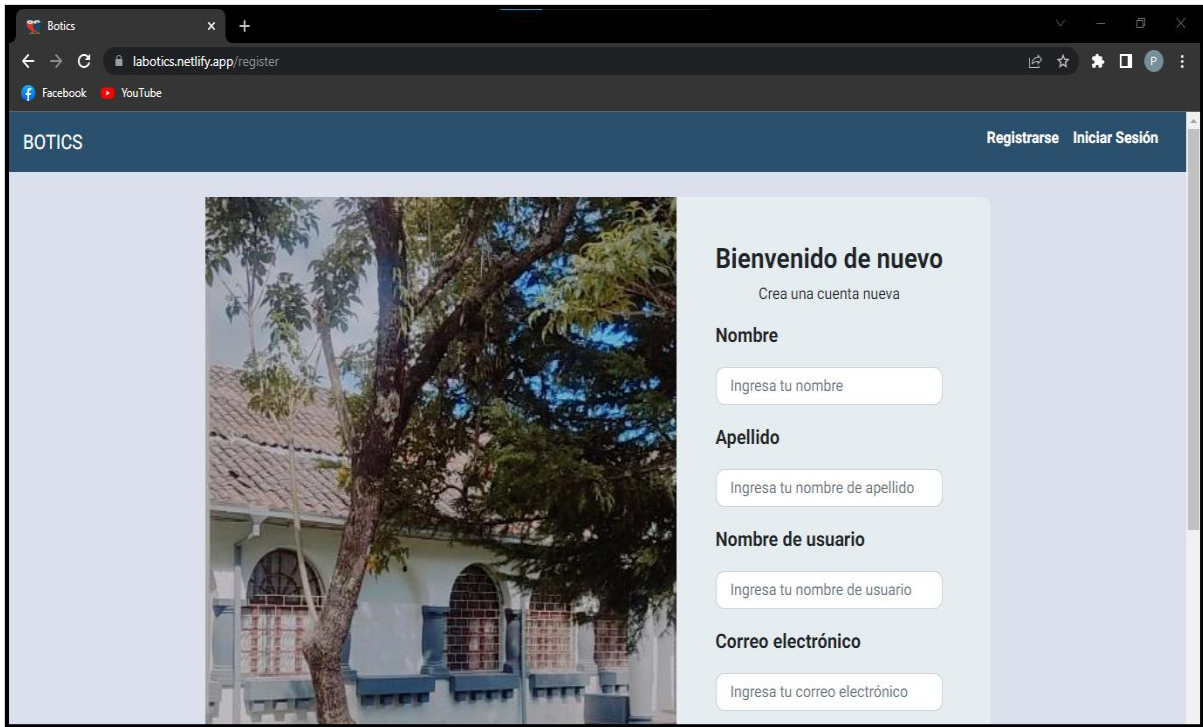


Fig. 85: : Página de registro en el navegador Google Chrome.

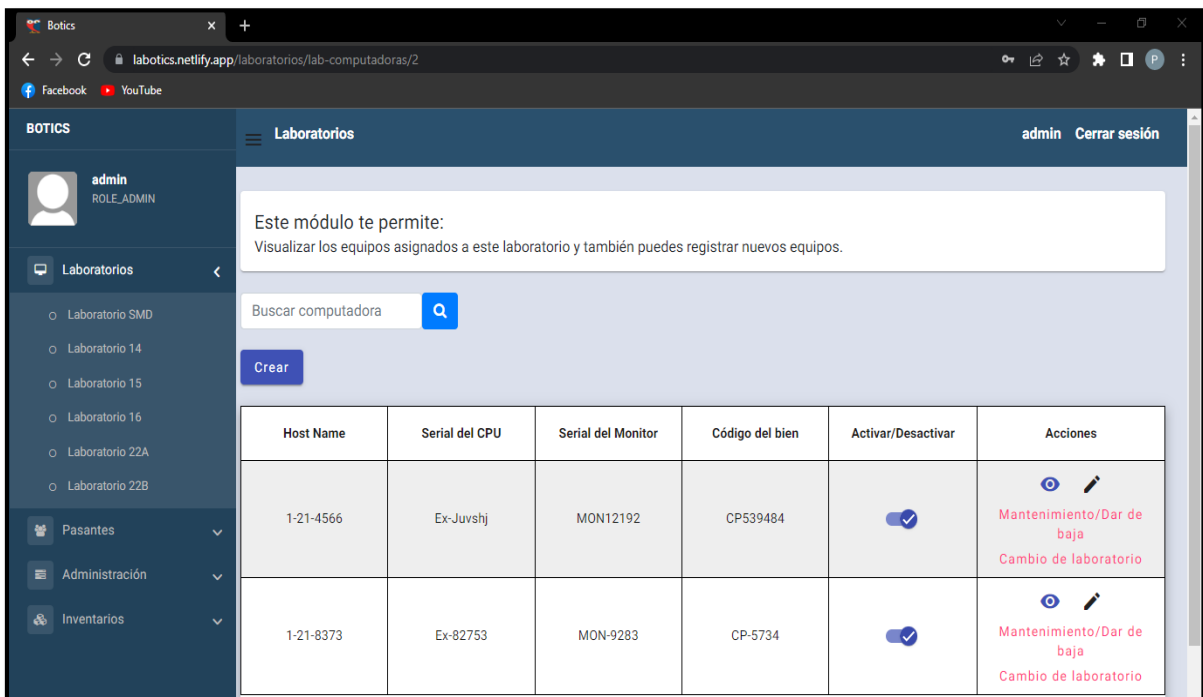


Fig. 86: Página laboratorios (usuario administrador) en el navegador Google Chrome.

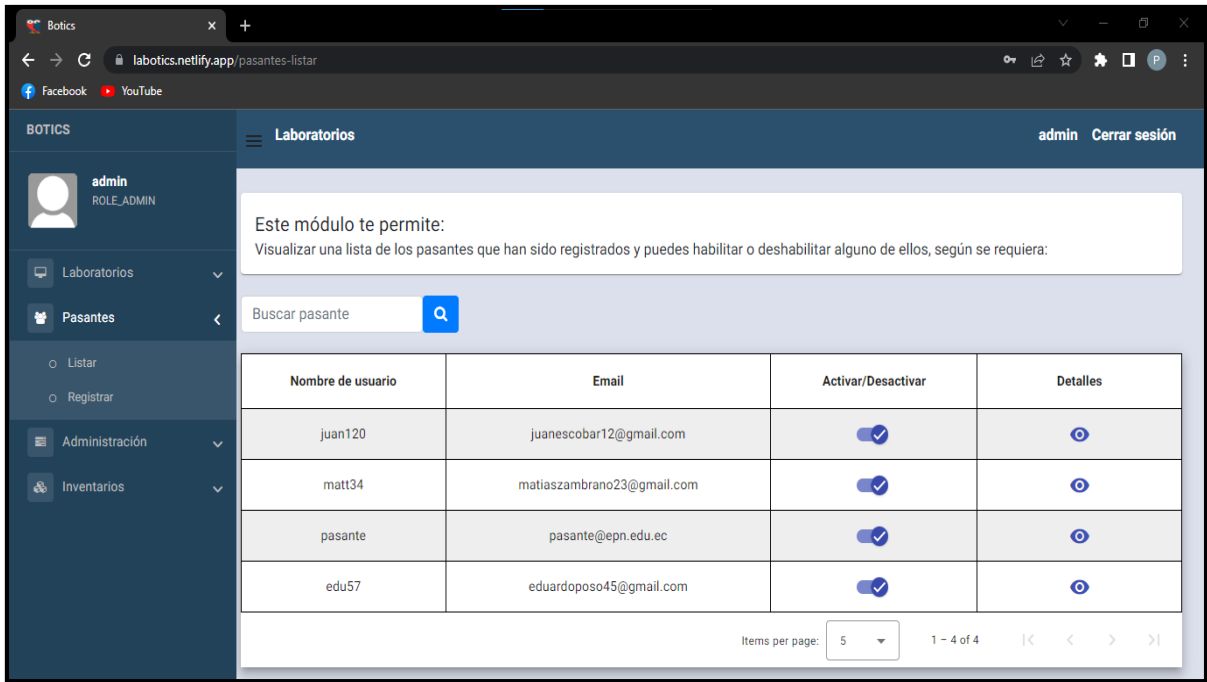


Fig. 87: Página gestionar pasantes (usuario administrador) en el navegador *Google Chrome*.

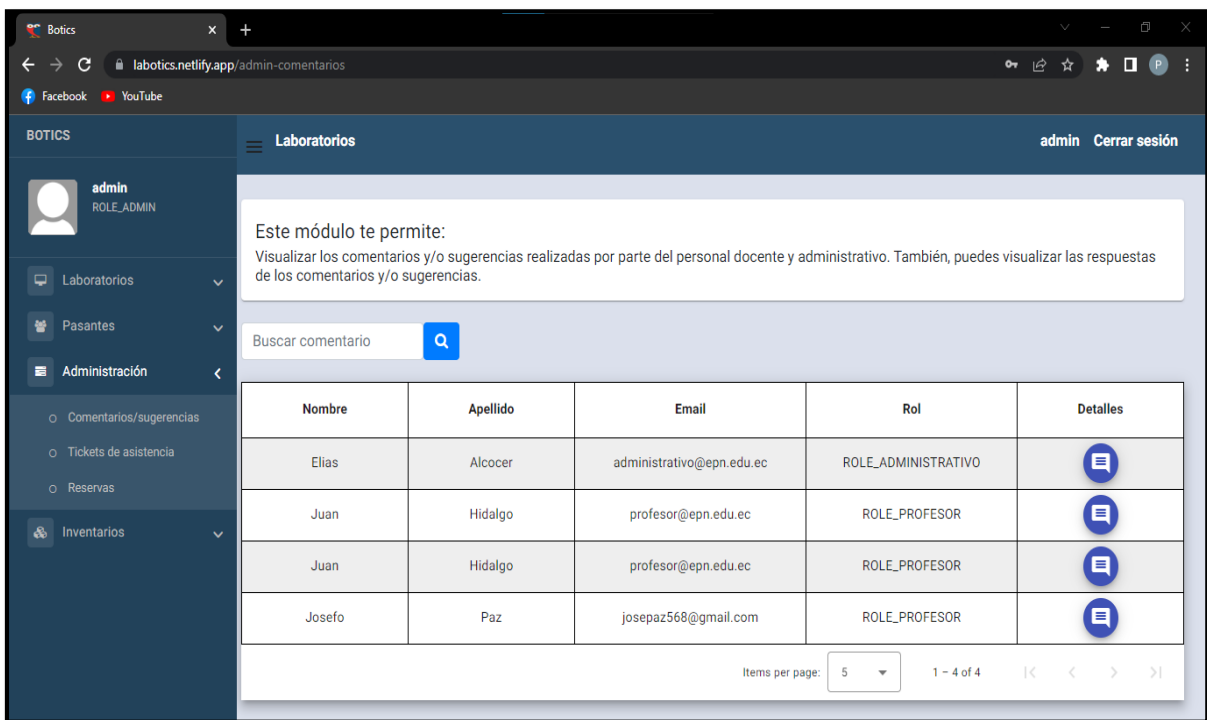


Fig. 88: Página administración (usuario administrador) en el navegador *Google Chrome*.

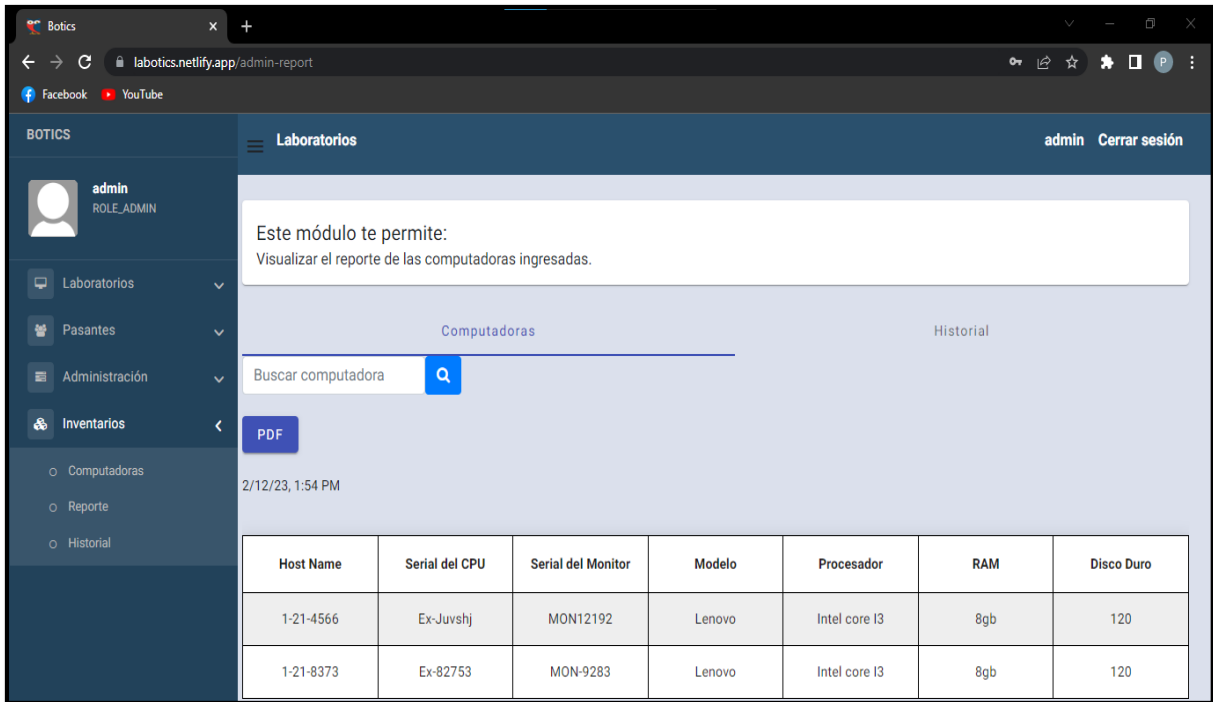


Fig. 89: Página inventarios (usuario administrador) en el navegador *Google Chrome*.

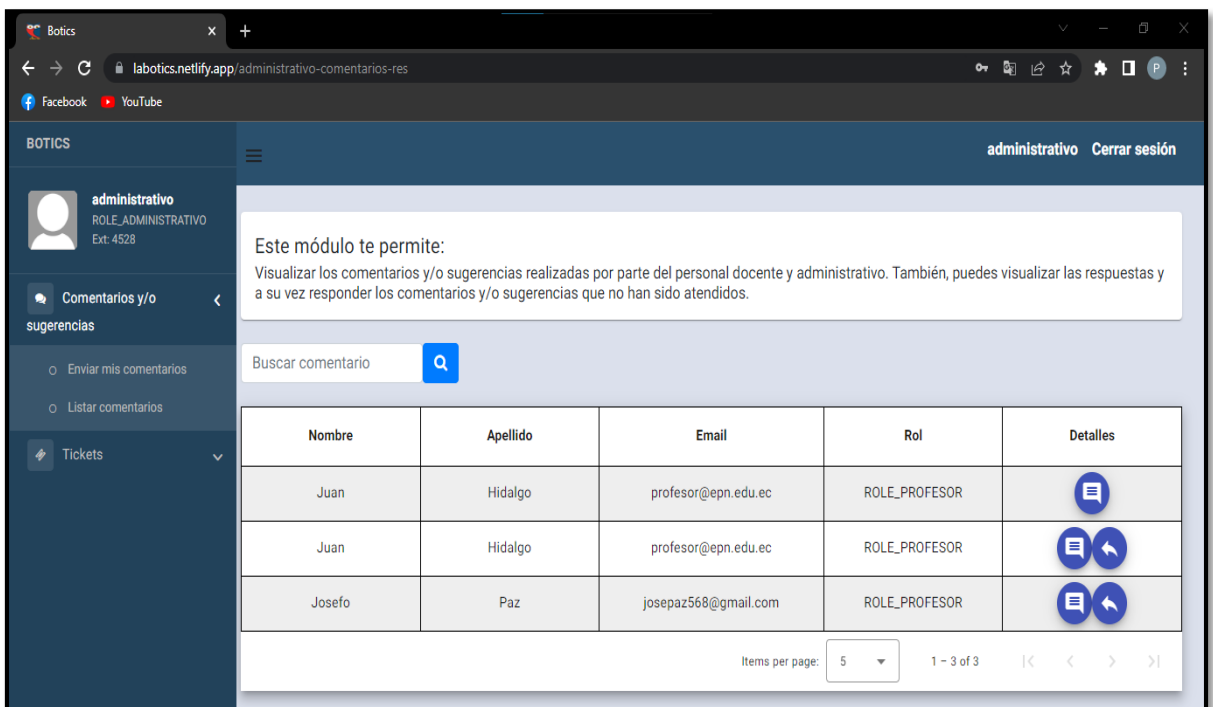


Fig. 90: Página gestionar comentarios y/o sugerencias (usuario administrativo) en el navegador *Google Chrome*.

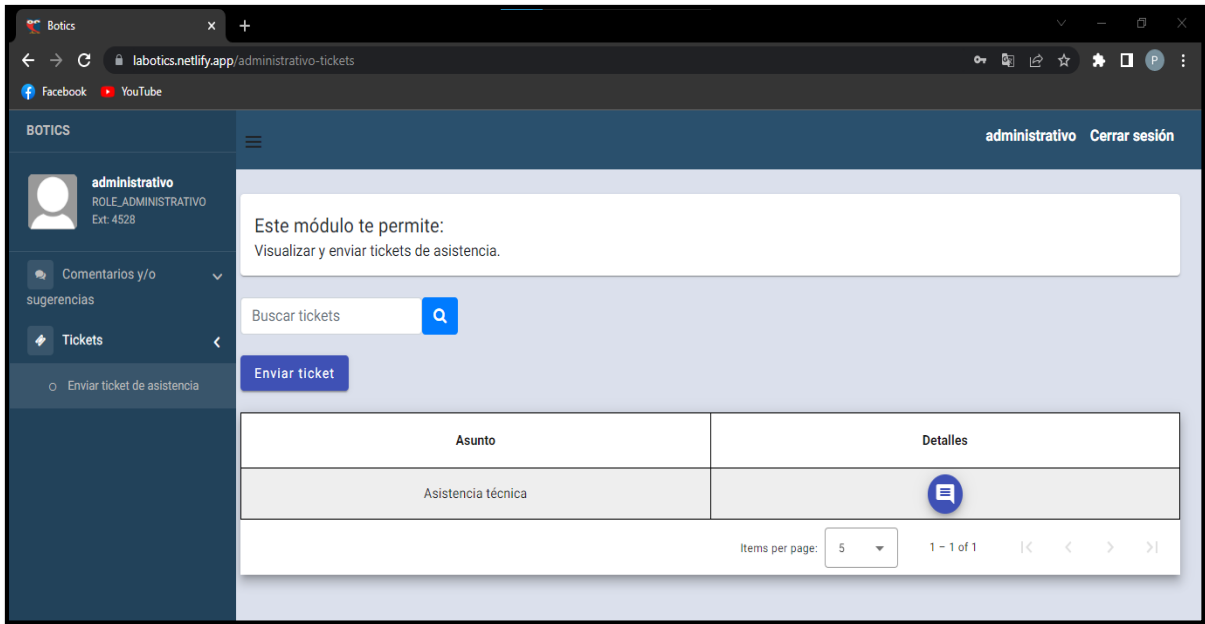


Fig. 91: Página tickets de asistencia (usuario administrativo) en el navegador *Google Chrome*.

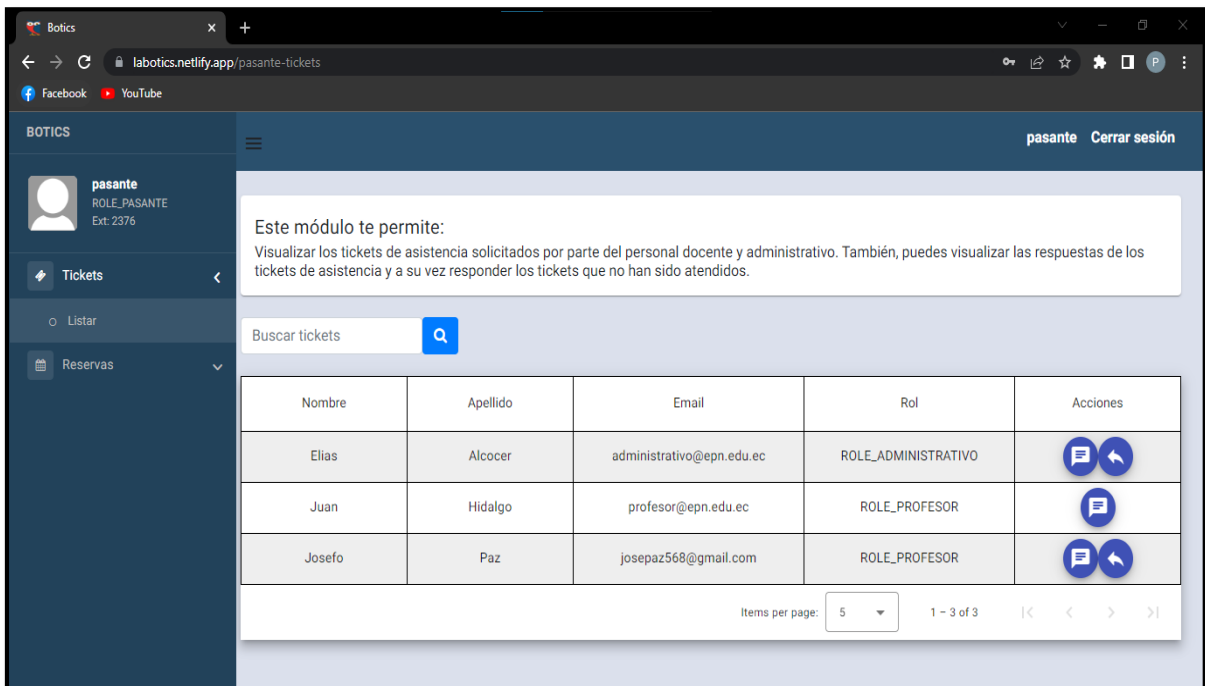


Fig. 92: Página tickets de asistencia (usuario pasante) en el navegador *Google Chrome*.

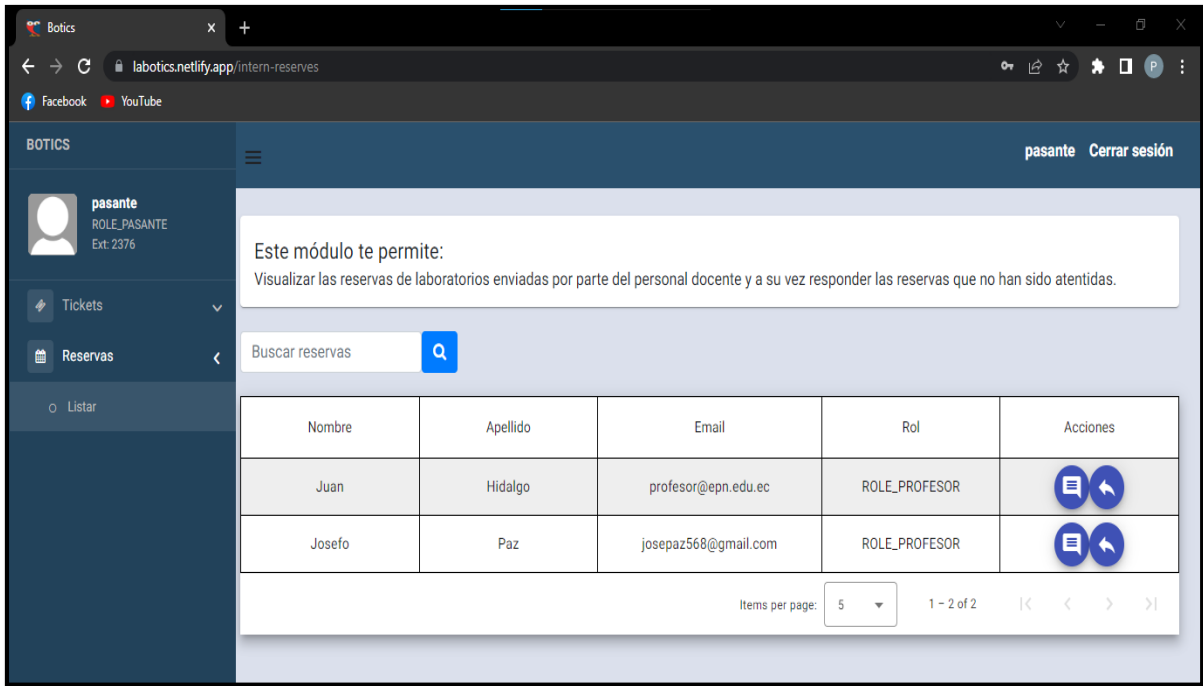


Fig. 93: Página gestionar reservas (usuario pasante) en el navegador Google Chrome.

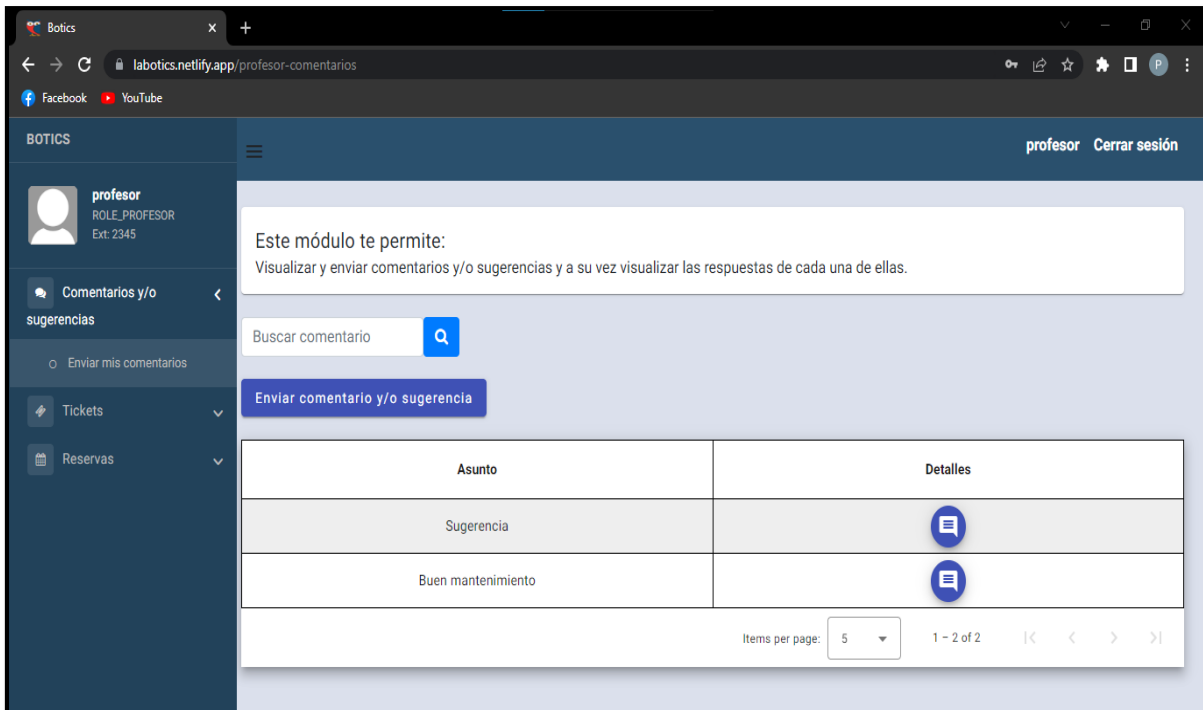


Fig. 94: Página comentarios y/o sugerencias (usuario docente) en el navegador Google Chrome.

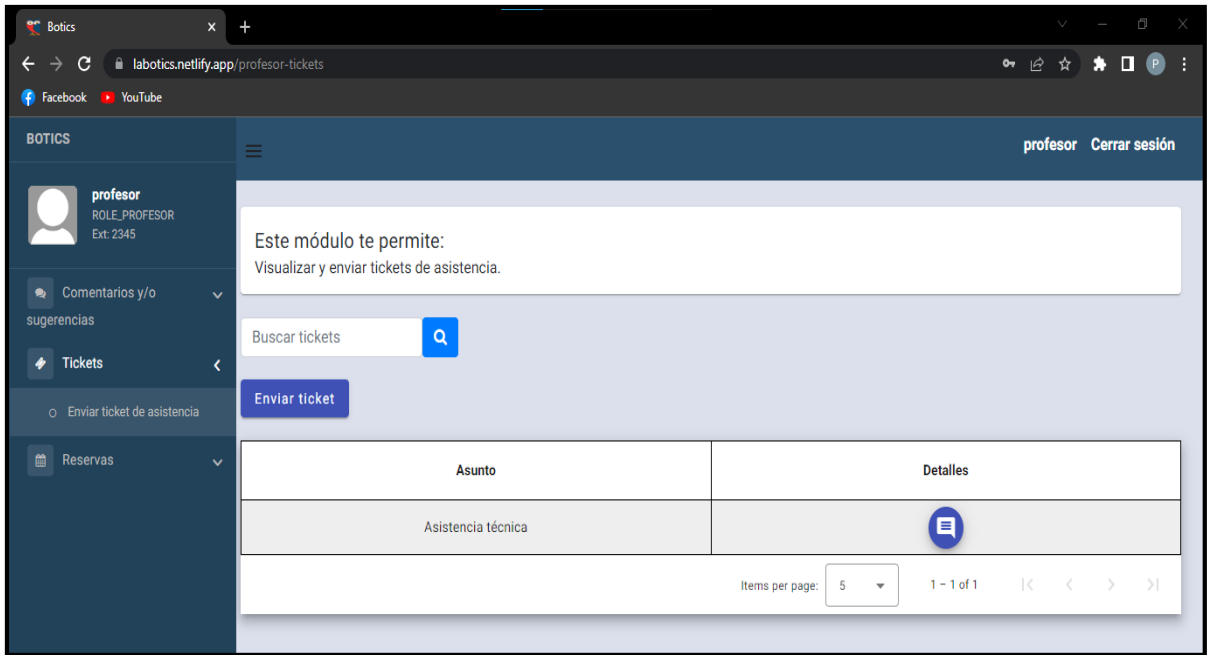


Fig. 95: Página de tickets de asistencia (usuario docente) en el navegador *Google Chrome*.

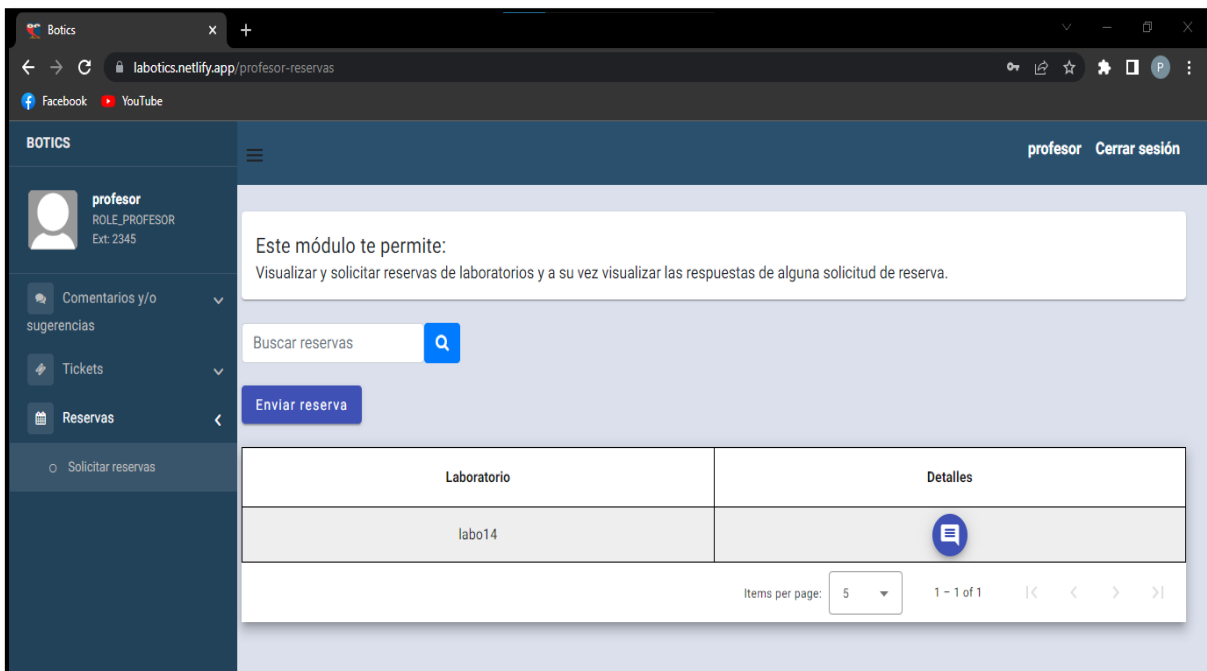


Fig. 96: Página de reservas (usuario docente) en el navegador *Google Chrome*.



Fig. 97: Página informativa vista en el navegador *Microsoft Edge*.

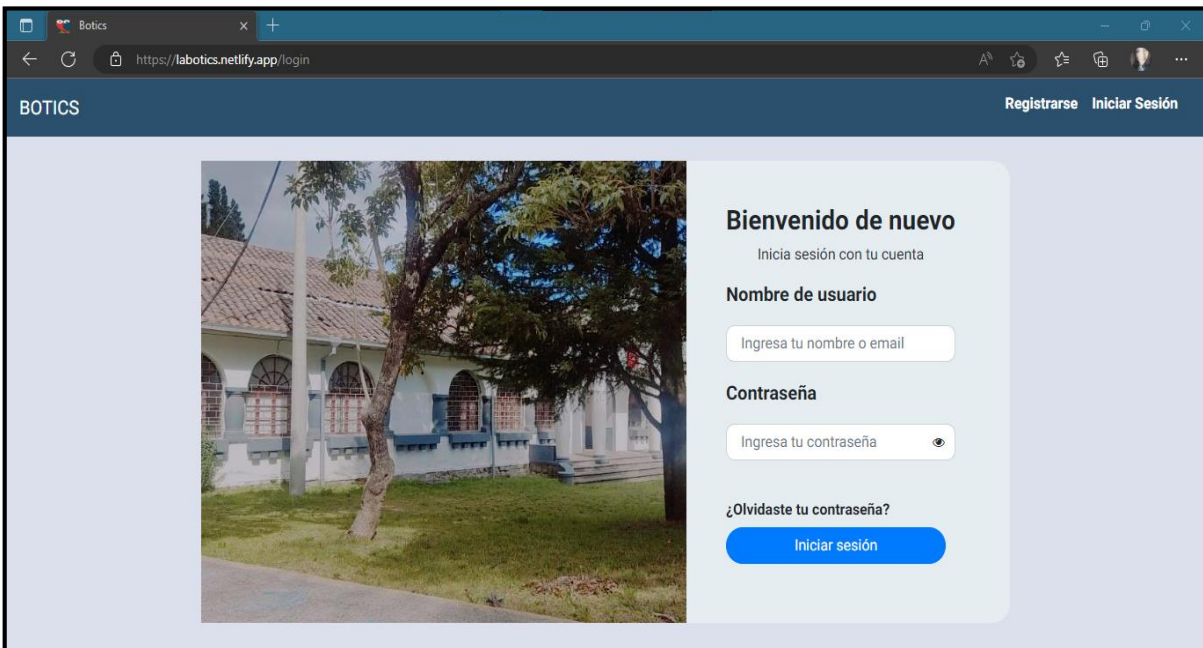


Fig. 98: Página de Inicio de Sesión en el navegador *Microsoft Edge*.

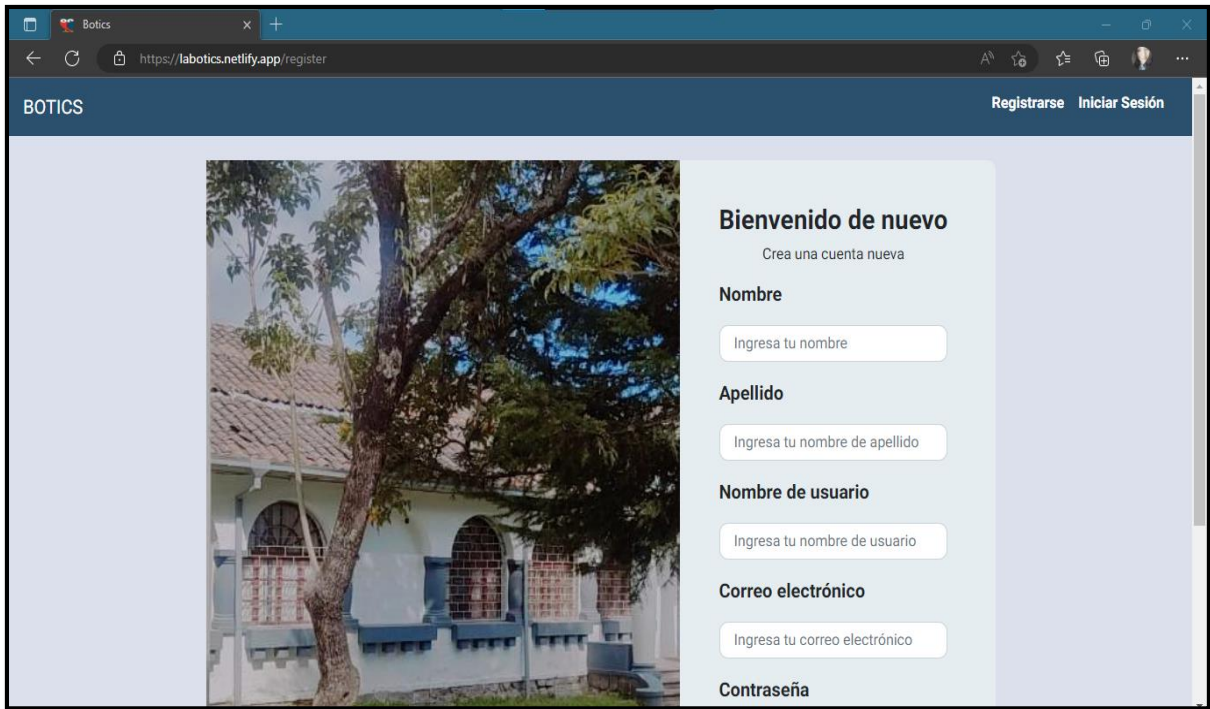


Fig. 99: Página de registro en el navegador *Microsoft Edge*.

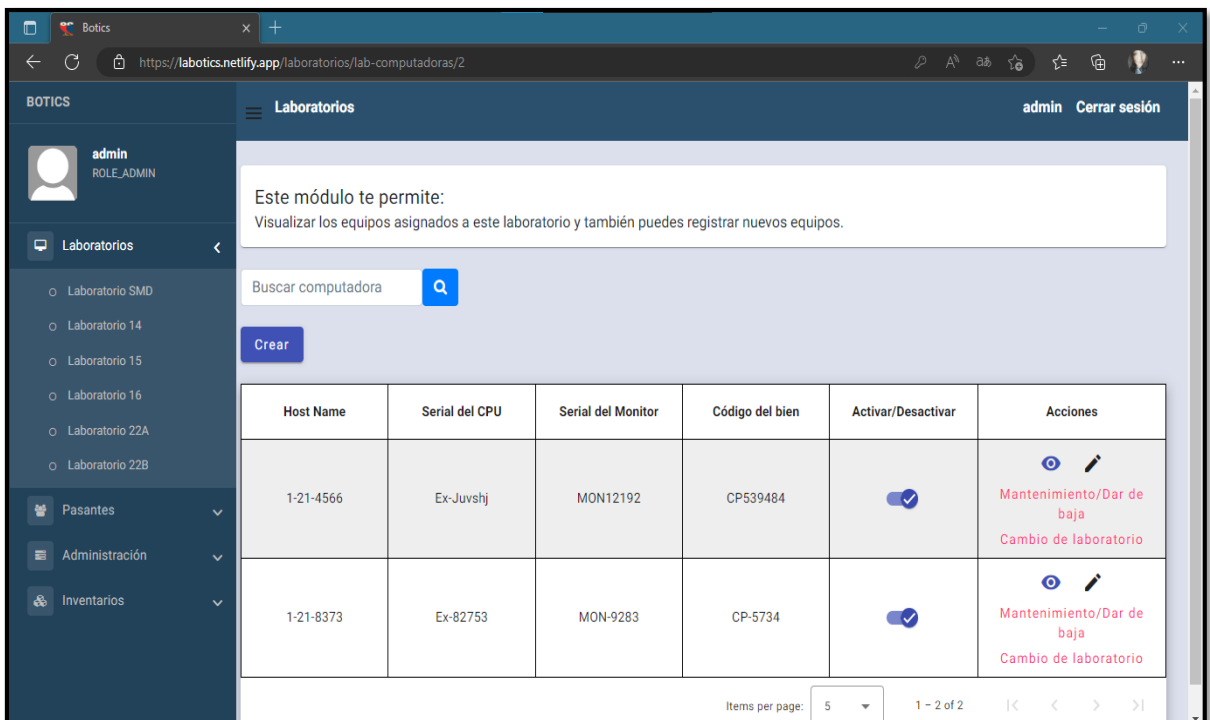


Fig. 100: Página laboratorios (usuario administrador) en el navegador *Microsoft Edge*.

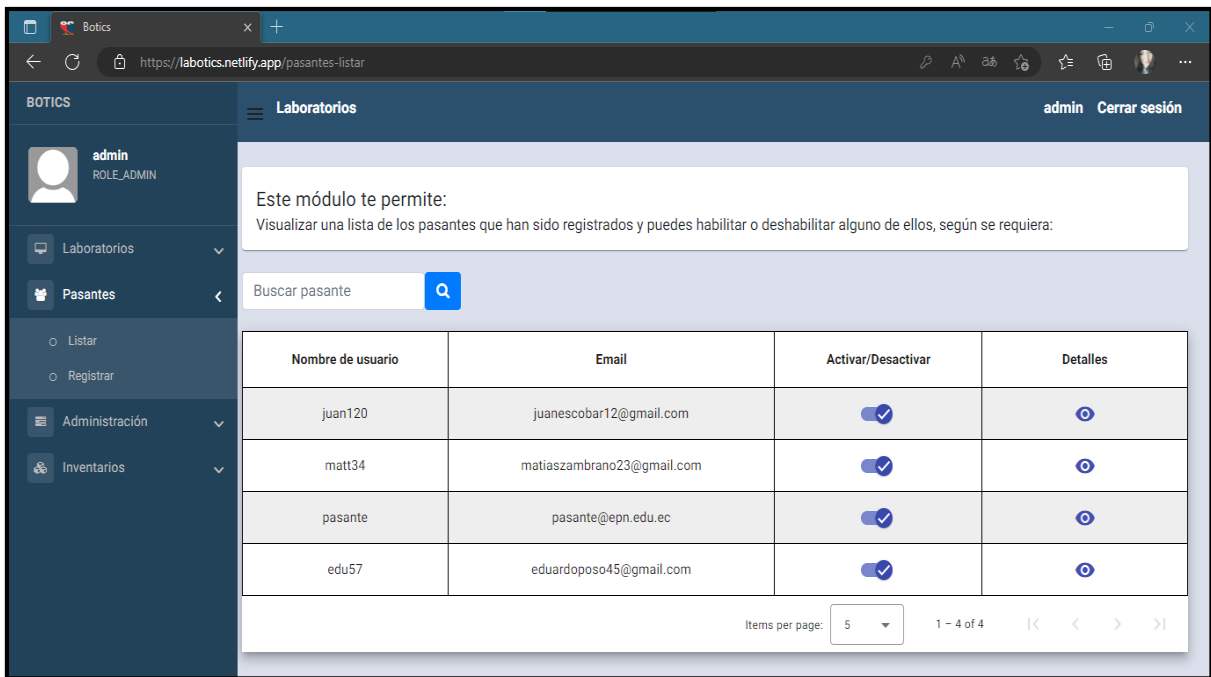


Fig. 101: Página gestionar pasantes (usuario administrador) en el navegador *Microsoft Edge*.

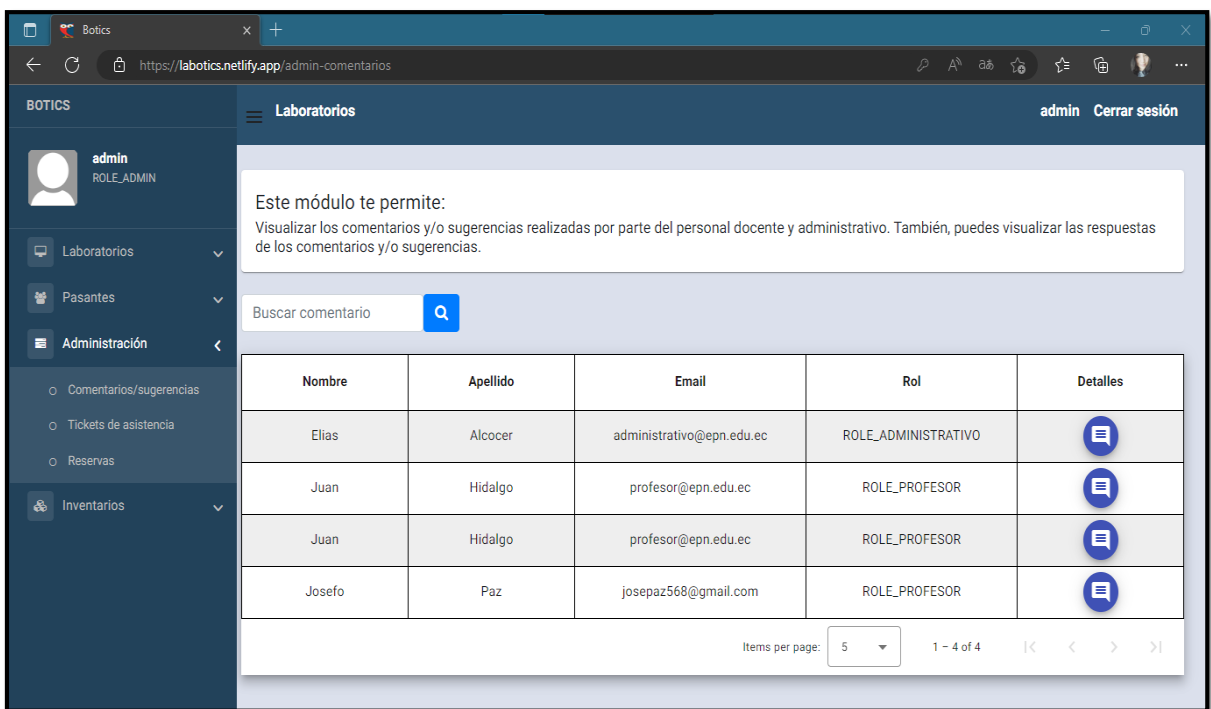


Fig. 102: Página administración (usuario administrador) en el navegador *Microsoft Edge*.

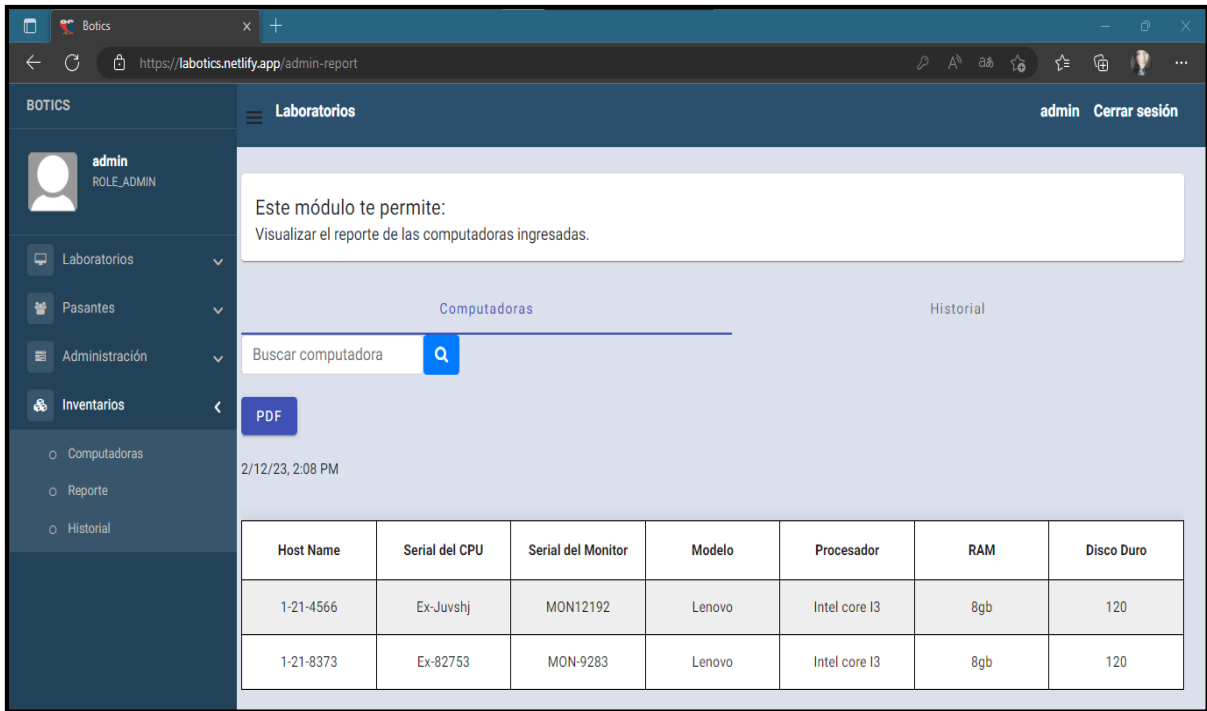


Fig. 103: Página inventarios (usuario administrador) en el navegador *Microsoft Edge*.

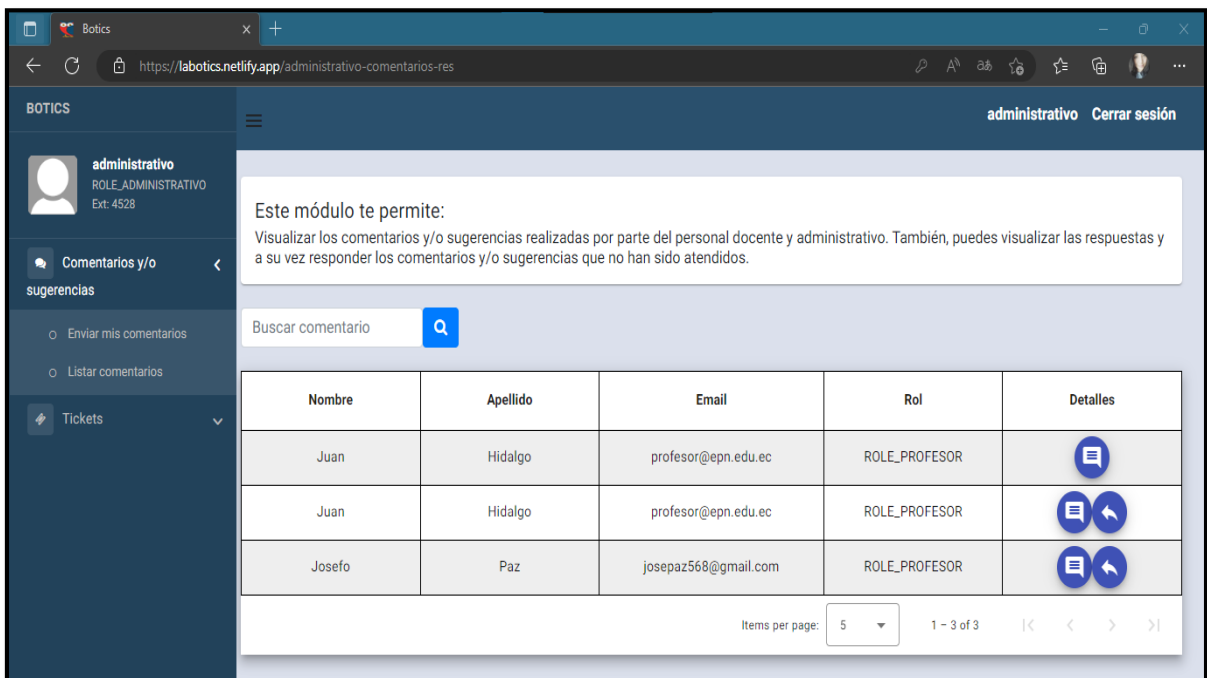


Fig. 104: Página gestionar comentarios y/o sugerencias (usuario administrativo) en el navegador *Microsoft Edge*.

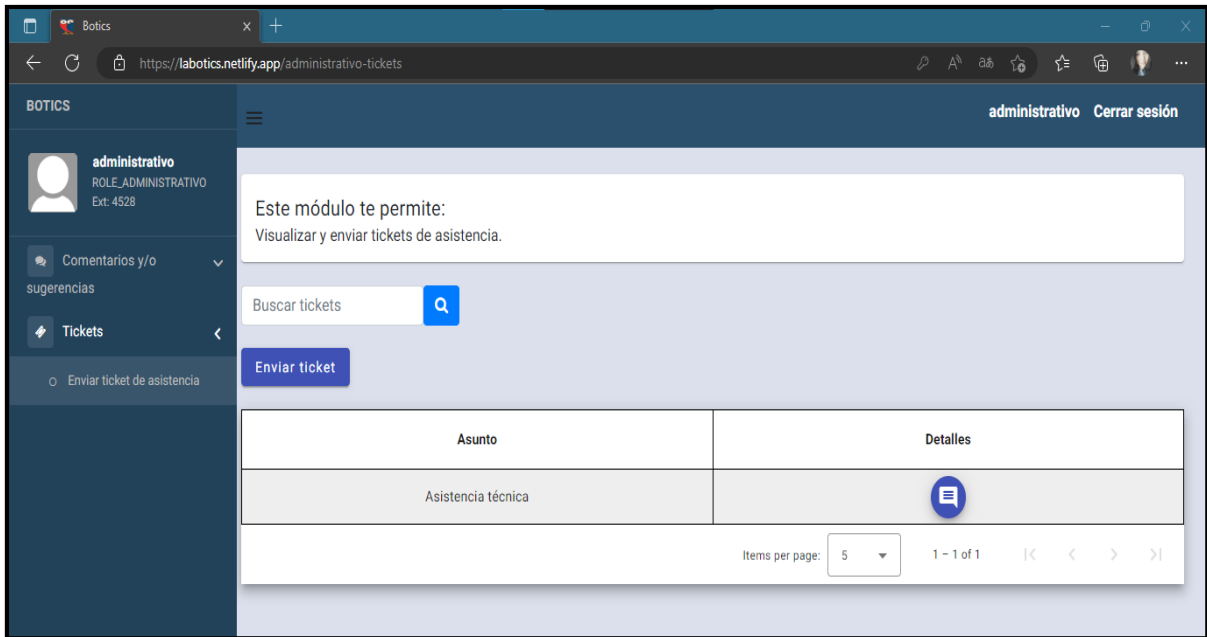


Fig. 105: Página tickets de asistencia (usuario administrativo) en el navegador *Microsoft Edge*.

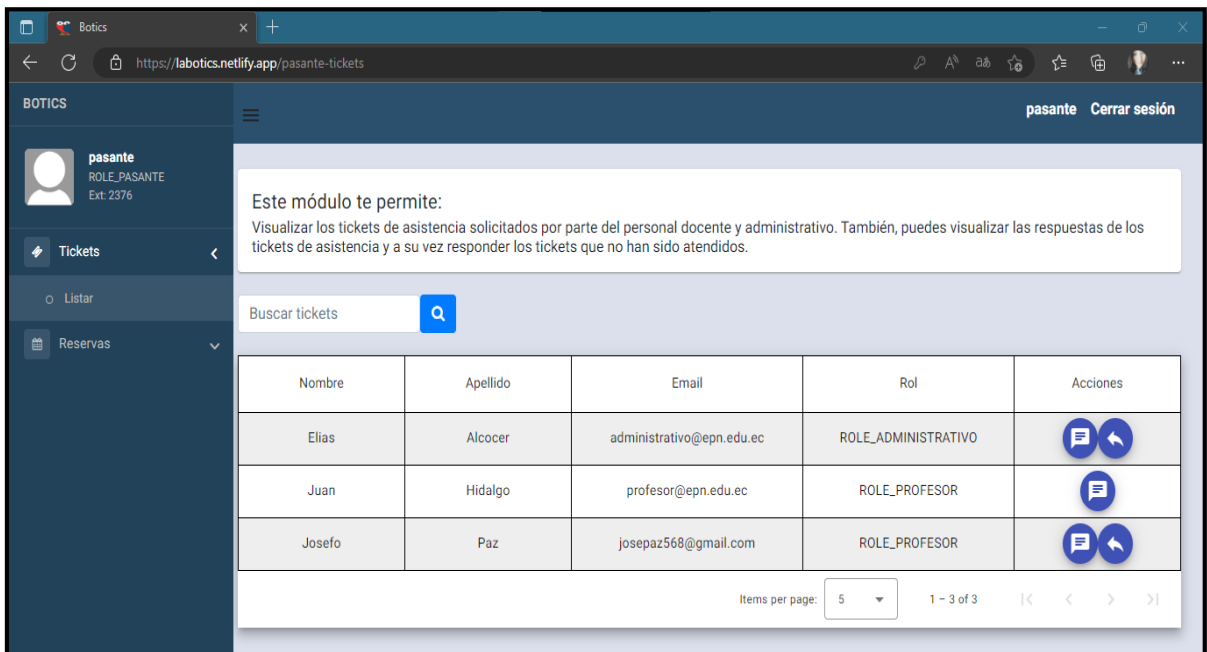


Fig. 106: Página tickets de asistencia (usuario pasante) en el navegador *Microsoft Edge*.

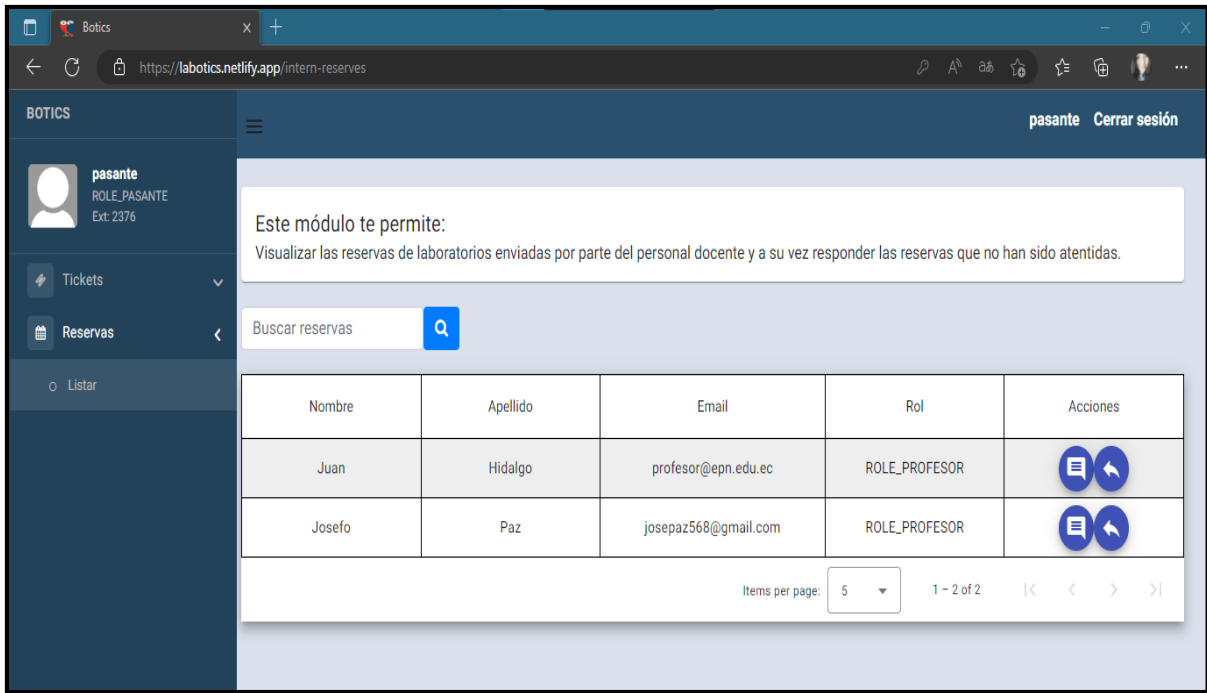


Fig. 107: Página gestionar reservas (usuario pasante) en el navegador *Microsoft Edge*.

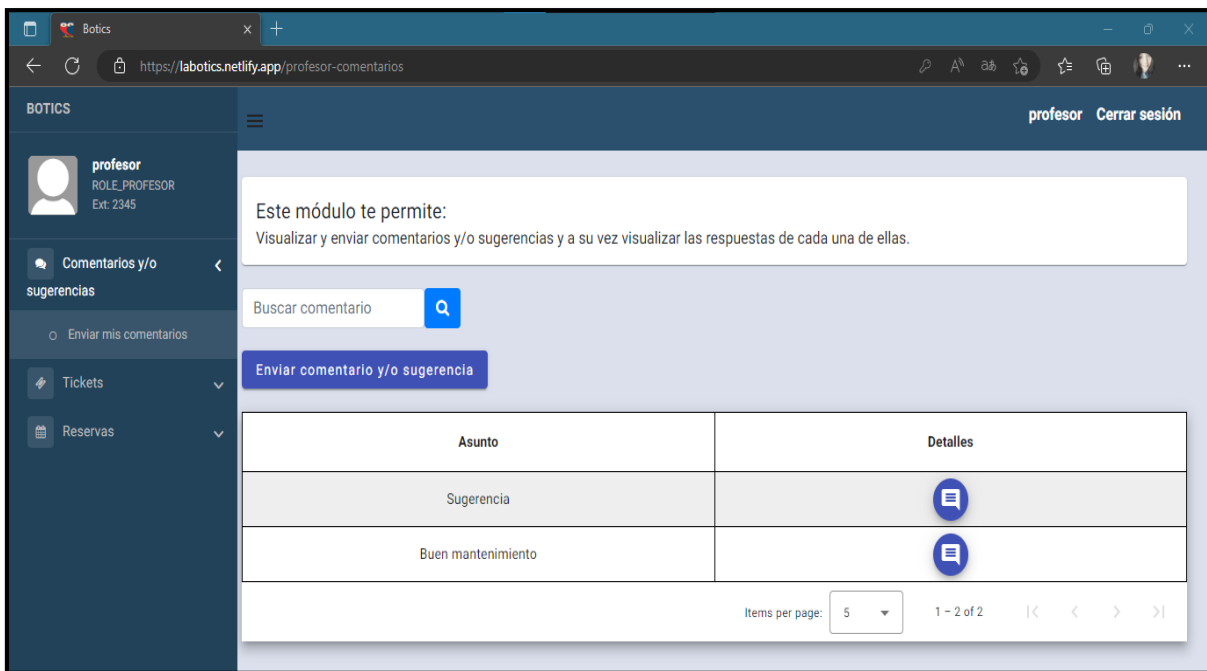


Fig. 108: Página comentarios y/o sugerencias (usuario docente) en el navegador *Microsoft Edge*.

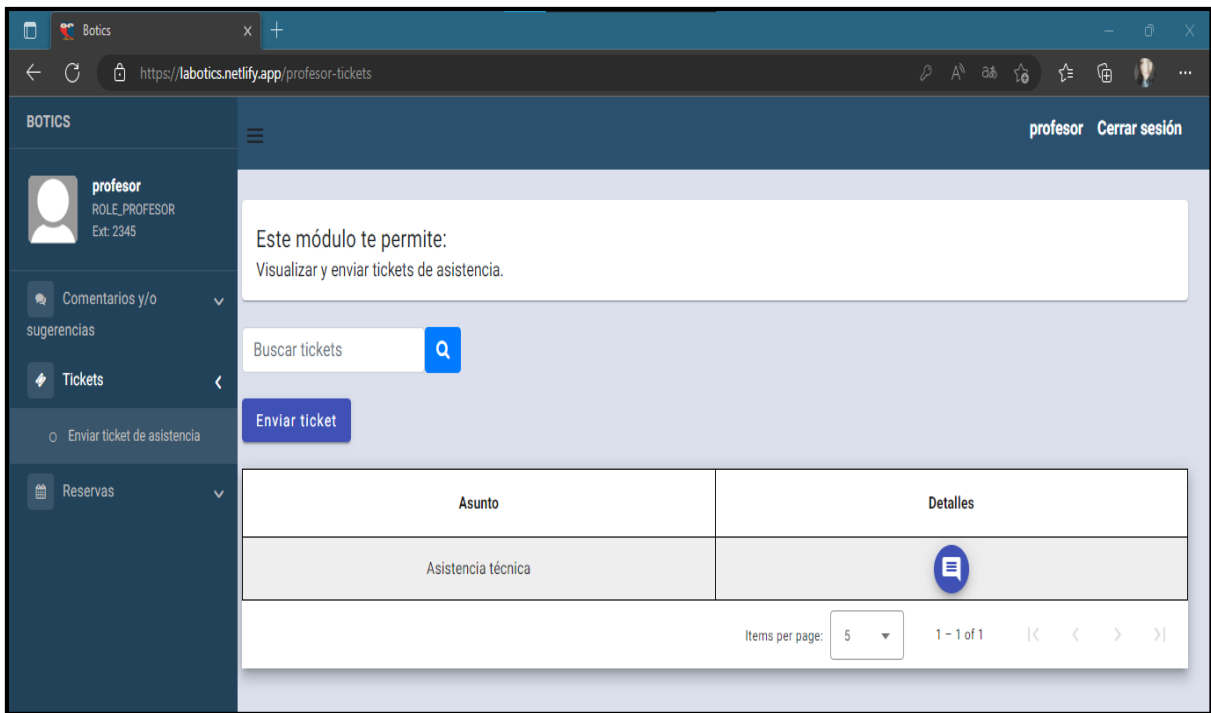


Fig. 109: Página de tickets de asistencia (usuario docente) en el navegador *Microsoft Edge*.

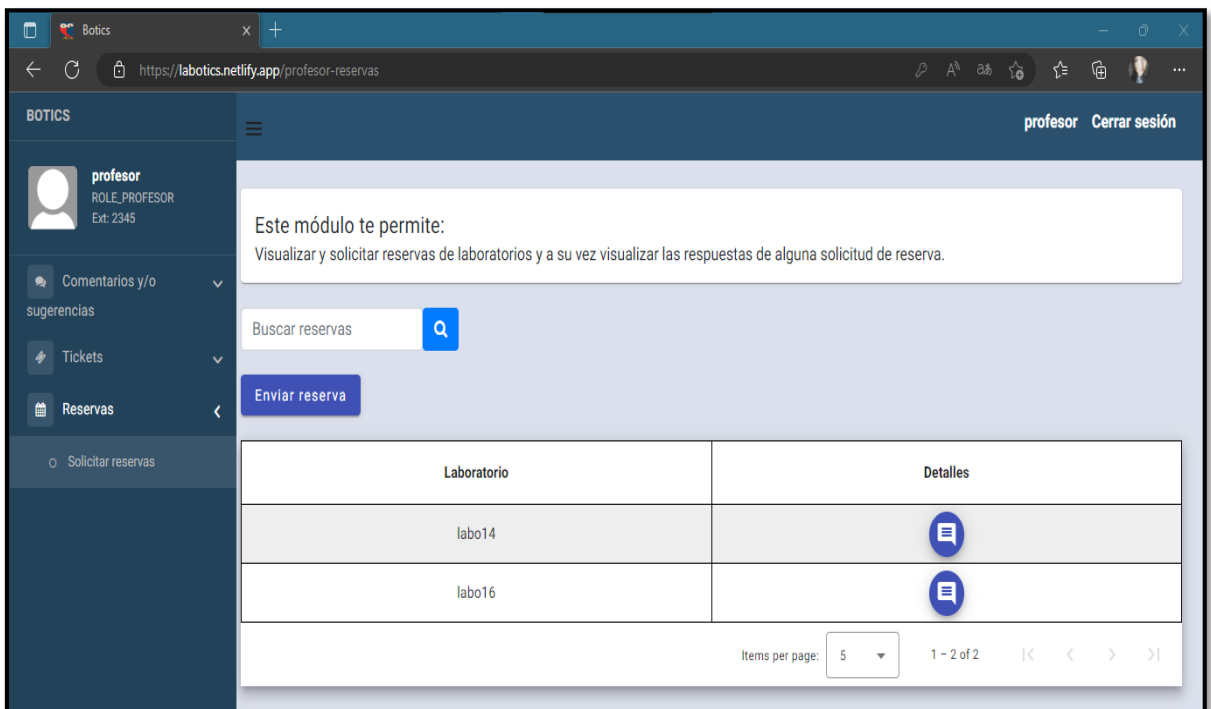


Fig. 110: Página de reservas (usuario docente) en el navegador *Microsoft Edge*.

Prueba de aceptación

Se procede a mostrar las 10 pruebas de aceptación las cuales van desde la **TABLA XXV** hasta la **TABLA XXXVI**. Es importante mencionar que todas las pruebas describen el proceso que corresponde a las diferentes tareas asignadas de los usuarios del *frontend* para su correcta funcionalidad y verificación de cada una de ellas.

TABLA XXV: Prueba de aceptación 2 Consumir *endpoints* para registrarse.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA002	Identificador de historia de Usuario: HU002
Nombre: Registrarse	
Descripción: En el frontend los usuarios docente y administrativo necesitan consumir varios endpoints para registrarse por medio de un formulario, llenando los siguientes campos: <ul style="list-style-type: none">• Nombre de usuario.• Correo electrónico.• Contraseña.• Rol.	
Pasos de ejecución: Para el registro del usuario administrativo y docente: <ul style="list-style-type: none">• Ubicar la URL del <i>frontend</i> en los navegadores.• En el lado derecho en el menú superior dar clic en el botón “Registrarse”.• Ubicar el nombre de usuario, correo electrónico, contraseña y escoger el rol del usuario.• Hacer clic en registrarse.	
Resultado deseado: En BOTICS del <i>frontend</i> admite registrar usuarios con rol administrativo y docente.	
Evaluación de la prueba: Se evidencia el resultado esperado. Conformidad del cliente 100%.	

TABLA XXVI: Prueba de aceptación 3 Consumir endpoints para modificar contraseña

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA003	Identificador de historia de Usuario: HU003
Nombre: Iniciar sesión, cerrar sesión y modificar contraseña.	
<p>Descripción: En el <i>frontend</i> los usuarios administrador, pasante, docente y administrativo necesitan consumir varios endpoints para:</p> <ul style="list-style-type: none"> • Iniciar sesión • Cerrar sesión. • Modificar contraseña. 	
<p>Pasos de ejecución:</p> <p>Para iniciar sesión usuario administrador, pasante, docente y administrativo:</p> <ul style="list-style-type: none"> • Ubicar la URL del <i>frontend</i> en los navegadores. • En lado derecho en el menú superior dar clic en el botón “Iniciar Sesión”. • Poner el correo electrónico y contraseña dependiendo del rol de usuario. • Dar clic en “Ingresar”. <p>Para cerrar sesión:</p> <ul style="list-style-type: none"> • Haber Iniciado sesión en el <i>frontend</i>. • En el lado derecho en el menú superior dar clic en el botón “Salir”. <p>Para modificar contraseña:</p> <ul style="list-style-type: none"> • Ubicar la URL del <i>frontend</i> en los navegadores. • Hacer clic en el menú Iniciar sesión. • Hacer clic en la parte inferior del formulario donde dice modificar contraseña. • Ubicar el correo electrónico en cual se haya registrado en el <i>frontend</i>. • Hacer clic en enviar. • Dirigirse a la cuenta de correo para revisar el correo enviado y hacer clic en el mismo. • Ubicar la nueva contraseña y enviar. • Llenar el formulario respectivo. • Dirigirse nuevamente al <i>frontend</i> y ubicarse en Iniciar sesión y poner la nueva contraseña. 	

<p>Resultado deseado:</p> <p>En BOTICS del <i>frontend</i> admite iniciar sesión y cerrar sesión y recuperar/modificar contraseña del usuario administrador, pasante, docente y administrativo</p>
<p>Evaluación de la prueba:</p> <p>Se evidencia el resultado esperado. Conformidad del cliente 100%.</p>

TABLA XXVII: Prueba de aceptación 4 Consumir varios endpoints para modificar perfil perfil de usuario de rol administrador, administrativo, pasante y docente.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA004	Identificador de historia de Usuario: HU004
Nombre: Modificar perfil de usuario.	
<p>Descripción: En el <i>frontend</i> los usuarios administrador, administrativo, pasante y docente necesitan consumir varios endpoints para visualizar y editar su perfil por medio de los siguientes campos:</p> <ul style="list-style-type: none"> • Nombres. • Apellidos • Avatar • Extensión. 	
<p>Pasos de ejecución:</p> <p>Para editar perfil usuario administrador.</p> <ul style="list-style-type: none"> • Iniciar sesión en el sistema. • Hacer clic en el menú lateral en perfil. • Hacer clic en la parte inferior en editar. • Editar los campos permitidos • Hacer clic en editar. 	

<p>Resultado deseado:</p> <p>En TakiTri del <i>frontend</i> admite visualizar y editar el perfil del usuario administrador, administrativo, pasante y docente.</p>
<p>Evaluación de la prueba:</p> <p>Se evidencia el resultado esperado. Conformidad del cliente 100%.</p>

TABLA XXVIII: Prueba de aceptación 5 Consumir varios *endpoints* para gestionar pasantes

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA005	Identificador de historia de Usuario: HU005
Nombre: Gestionar pasantes.	
<p>Descripción: El usuario administrador en el <i>frontend</i> puede consumir varios métodos para:</p> <ul style="list-style-type: none"> • Registrar pasantes. • Visualizar lista de pasantes. 	
<p>Pasos de ejecución:</p> <p>Para gestionar pasantes por parte del usuario administrador.</p> <ul style="list-style-type: none"> • Iniciar sesión en el sistema como usuario administrador. • Visualizar la lista de pasantes en estado activo e inactivo. • Registrar nuevos pasantes. 	
<p>Resultado deseado:</p> <p>En BOTICS del <i>frontend</i> admite registrar y visualizar la lista de pasantes.</p>	
<p>Evaluación de la prueba:</p> <p>Se evidencia el resultado esperado. Conformidad del cliente 100%.</p>	

TABLA XXIX: Prueba de aceptación 6 Consumir varios endpoints para gestionar inventarios.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA006	Identificador de historia de Usuario: HU006
Nombre: Gestionar inventarios.	
Descripción: En el <i>frontend</i> el usuario administrador necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Registrar inventarios. • Modificar inventarios. • Visualizar inventarios. • Eliminar inventarios. 	
Pasos de ejecución: Para gestionar inventarios por parte del usuario administrador. <ul style="list-style-type: none"> • Iniciar sesión en el sistema como usuario administrador. • Visualizar la lista de inventarios en estado activo e inactivo, y el apartado de acciones, donde el administrador puede crear, modificar y eliminar 	
Resultado deseado: En BOTICS del <i>frontend</i> admite registrar, visualizar, modificar y eliminar inventarios.	
Evaluación de la prueba: Se evidencia el resultado esperado. Conformidad del cliente 100%.	

TABLA XXX: Prueba de aceptación 7 Consumir varios endpoints para visualizar reporte de inventarios.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA007	Identificador de historia de Usuario: HU007
Nombre: Visualizar reporte de inventarios.	

<p>Descripción: En el <i>frontend</i> el usuario administrador necesita consumir varios <i>endpoints</i> para:</p> <ul style="list-style-type: none"> • Visualizar reporte de inventarios.
<p>Pasos de ejecución:</p> <p>Para visualizar reporte de inventarios por parte del usuario administrador.</p> <ul style="list-style-type: none"> • Iniciar sesión en el sistema como usuario administrador. • Visualizar los reportes de inventarios generados.
<p>Resultado deseado:</p> <p>En BOTICS del <i>frontend</i> admite visualizar los reportes de los inventarios.</p>
<p>Evaluación de la prueba:</p> <p>Se evidencia el resultado esperado. Conformidad del cliente 100%.</p>

TABLA XXXI: Prueba de aceptación 8 Consumir varios *endpoints* para gestionar comentarios y/o sugerencias.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA008	Identificador de historia de Usuario: HU008
Nombre: Gestionar comentarios y/o sugerencias.	
<p>Descripción: En el <i>frontend</i> el usuario administrativo y pasante necesita consumir varios <i>endpoints</i> para:</p> <ul style="list-style-type: none"> • Visualizar comentarios y/o sugerencias. • Atender comentarios y/o sugerencias. 	
<p>Pasos de ejecución:</p> <p>Para visualizar reporte de inventarios por parte del usuario administrativo y pasante.</p> <ul style="list-style-type: none"> • Iniciar sesión en el sistema como usuario administrador o pasante. • Visualizar todos los comentarios y/o sugerencias realizadas por parte del usuario docente. 	

<ul style="list-style-type: none"> • Atender las solicitudes de comentarios y/o sugerencias que no han sido respondidas.
<p>Resultado deseado:</p> <p>En BOTICS del <i>frontend</i> admite visualizar y atender los comentarios y/o sugerencias.</p>
<p>Evaluación de la prueba:</p> <p>Se evidencia el resultado esperado. Conformidad del cliente 100%.</p>

TABLA XXXII: Prueba de aceptación 9 Consumir varios *endpoints* para gestionar tickets de asistencia.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA009	Identificador de historia de Usuario: HU009
Nombre: Gestionar <i>tickets</i> de asistencia.	
<p>Descripción: En el <i>frontend</i> el usuario pasante necesita consumir varios <i>endpoints</i> para:</p> <ul style="list-style-type: none"> • Visualizar <i>tickets</i>. • Atender <i>tickets</i>. 	
<p>Pasos de ejecución:</p> <p>Para visualizar reporte de inventarios por parte del usuario pasante.</p> <ul style="list-style-type: none"> • Iniciar sesión en el sistema como usuario pasante. • Visualizar todos los <i>tickets</i> de asistencia solicitados. • Atender las solicitudes de <i>tickets</i> de asistencia que aún no han sido respondidas. 	
<p>Resultado deseado:</p> <p>En BOTICS del <i>frontend</i> admite visualizar y atender los <i>tickets</i> de asistencia..</p>	
<p>Evaluación de la prueba:</p> <p>Se evidencia el resultado esperado.</p>	

Conformidad del cliente 100%.

TABLA XXXIII: Prueba de aceptación 10 Consumir varios endpoints para gestionar reservas.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA010	Identificador de historia de Usuario: HU010
Nombre: Gestionar reservas.	
Descripción: En el <i>frontend</i> el usuario pasante necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none">• Visualizar reservas.• Tramitar reservas.	
Pasos de ejecución: Para visualizar reporte de inventarios por parte del usuario pasante. <ul style="list-style-type: none">• Iniciar sesión en el sistema como usuario pasante.• Visualizar todas las solicitudes de reservas de laboratorios.• Tramitar las reservas, mediante una respuesta a la solicitud de reserva.	
Resultado deseado: En BOTICS del <i>frontend</i> admite visualizar y tramitar reservas.	
Evaluación de la prueba: Se evidencia el resultado esperado. Conformidad del cliente 100%.	

TABLA XXXIV: Prueba de aceptación 11 Consumir varios endpoints para solicitar tickets de asistencia.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA011	Identificador de historia de Usuario: HU011
Nombre: Solicitar <i>tickets</i> de asistencia.	

<p>Descripción: En el <i>frontend</i> el usuario docente y administrativo necesita consumir varios <i>endpoints</i> para:</p> <ul style="list-style-type: none"> • Solicitar un ticket de asistencia.
<p>Pasos de ejecución:</p> <p>Para solicitar <i>tickets</i> de asistencia por parte del usuario docente y administrativo.</p> <ul style="list-style-type: none"> • Iniciar sesión en el sistema como usuario pasante o administrativo según sea el rol correspondiente. • Solicitar un ticket de asistencia, donde debe llenar los campos de asunto y descripción y a su vez puede visualizar todas sus solicitudes realizadas.
<p>Resultado deseado:</p> <p>En BOTICS del <i>frontend</i> admite solicitar <i>tickets</i> de asistencia.</p>
<p>Evaluación de la prueba:</p> <p>Se evidencia el resultado esperado. Conformidad del cliente 100%.</p>

TABLA XXXV: Prueba de aceptación 12 Consumir varios *endpoints* para solicitar reservas.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA012	Identificador de historia de Usuario: HU012
Nombre: Solicitar reservas.	
<p>Descripción: En el <i>frontend</i> el usuario docente necesita consumir varios <i>endpoints</i> para:</p> <ul style="list-style-type: none"> • Registrar solicitud de reservas. 	
<p>Pasos de ejecución:</p> <p>Para solicitar reservas por parte del usuario docente.</p> <ul style="list-style-type: none"> • Iniciar sesión en el sistema como usuario docente. • Registrar la reserva de un laboratorio que desea solicitar, para lo cual debe llenar los campos requeridos. 	

<ul style="list-style-type: none"> • Visualizar las reservas de laboratorios solicitadas.
<p>Resultado deseado:</p> <p>En BOTICS del <i>frontend</i> admite solicitar la reserva de algún laboratorio.</p>
<p>Evaluación de la prueba:</p> <p>Se evidencia el resultado esperado. Conformidad del cliente 100%.</p>

TABLA XXXVI: Prueba de aceptación 13 Consumir varios *endpoints* para enviar comentarios y/o sugerencias.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA013	Identificador de historia de Usuario: HU013
Nombre: Enviar comentarios y/o sugerencias.	
Descripción: En el <i>frontend</i> el usuario docente necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Registrar comentarios y/o sugerencias. 	
Pasos de ejecución: <p>Para enviar comentarios y/o sugerencias por parte del usuario docente.</p> <ul style="list-style-type: none"> • Iniciar sesión en el sistema como usuario docente. • Registrar los comentarios y/o sugerencias con su respectivo asunto y descripción. 	
Resultado deseado: <p>En BOTICS del <i>frontend</i> admite enviar comentarios y/o sugerencias.</p>	
Evaluación de la prueba: <p>Se evidencia el resultado esperado. Conformidad del cliente 100%.</p>	

ANEXO III

A continuación, para visualizar el Manual de Usuario del *frontend* se debe digitar la siguiente URL:

<https://youtu.be/7aKo1Y46tOY>

En donde se explica de forma clara y sencilla las diversas funcionalidades del *frontend*, así como cada uno de los perfiles que forman parte de este componente.

ANEXO IV

A continuación, se presenta las credenciales de acceso del *frontend*, además del repositorio de GitHub, donde se encuentra todo el código de BOTICS.

Credenciales para el acceso del *frontend*.

Para ingresar al *frontend* ya en producción, se ingresa mediante la URL:

<https://labotics.netlify.app/inicio>

Repositorio del *frontend*

El proyecto se encuentra en un repositorio de *GitHub*, que se accede mediante la siguiente URL:

<https://github.com/Poeth120/botics>