



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**DISEÑO Y DESARROLLO DE APLICACIÓN WEB SOBRE PROCESOS ADMINISTRATIVOS
DE LOS ESTUDIANTES DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN
CARLOS DE GUATEMALA -ASISTO-**

Nelson Daniel Ramos Gómez

Cristian Emmanuel Sanchez Salvador

Asesorado por el Ing. Edgar Estuardo Santos Sutuj

Guatemala, mayo de 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO Y DESARROLLO DE APLICACIÓN WEB SOBRE PROCESOS ADMINISTRATIVOS
DE LOS ESTUDIANTES DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDA DE SAN
CARLOS DE GUATEMALA -ASISTO-**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

NELSON DANIEL RAMOS GÓMEZ

CRISTIAN EMMANUEL SANCHEZ SALVADOR

ASESORADO POR EL ING. EDGAR ESTUARDO SANTOS SUTUJ

AL CONFERÍRSELES EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, MAYO DE 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Luis Diego Aguilar Ralón
VOCAL V	Br. Christian Daniel Estrada Santizo
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. César Augusto Fernández Cáceres
EXAMINADOR	Ing. Herman Igor Véliz Linares
EXAMINADOR	Ing. Francisco Javier Guevara Castillo
SECRETARIA	Inga. Lesbia Magalí Herrera López

Nelson Daniel Ramos Gómez

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Luis Diego Aguilar Ralón
VOCAL V	Br. Christian Daniel Estrada Santizo
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Herman Igor Véliz Linares
EXAMINADOR	Ing. Pedro Pablo Hernández Ramírez
EXAMINADOR	Ing. Luis Fernando Espino Barrios
SECRETARIA	Inga. Lesbia Magalí Herrera López

Cristian Emmanuel Sanchez Salvador

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO Y DESARROLLO DE APLICACIÓN WEB SOBRE PROCESOS ADMINISTRATIVOS DE LOS ESTUDIANTES DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA -ASISTO-

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas con fecha 28 de junio de 2018.



Nelson Daniel Ramos Gómez



Cristian Emmanuel Sanchez Salvador

Guatemala, 15 de marzo de 2019

Ingeniero
Carlos Alfredo Azurdia
Coordinador de Privados y Revisor de Trabajos de Graduación
Escuela de Ciencias y Sistemas
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Ingeniero Azurdia:

Tengo el agrado de dirigirme a usted para informarle que he revisado el trabajo de graduación **"DISEÑO Y DESARROLLO DE APLICACIÓN WEB SOBRE PROCESOS ADMINISTRATIVOS DE LOS ESTUDIANTES DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA -ASISTO-**", realizado por los estudiantes universitarios **Nelson Daniel Ramos Gómez** con carné **201020550** y **Cristian Emmanuel Sánchez Salvador** con carné **200924554**, quienes contaron con la asesoría del suscrito.

Considero que el trabajo realizado por los estudiantes, cumple con los objetivos bajo los cuales fue planteado y cumple satisfactoriamente cada una de las actividades planificadas, por lo que procedo a aprobarlo.

Agradecimiento la atención dada a la presente

Atentamente



Ing. Edgar Santos
Asesor
Colegiado 5266

Edgar Santos
INGENIERO EN CIENCIAS - SISTEMAS
Colegiado 5266



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 26 de marzo del 2019

Ingeniero
Marlon Antonio Pérez Türk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación de los estudiantes **NELSON DANIEL RAMOS GÓMEZ** con carné **201020550** y CUI **2076 35595 0912**, y **CRISTIAN EMMANUEL SÁNCHEZ SALVADOR** con carné **200924554** y CUI **2112 72868 0101**, titulado: **DISEÑO Y DESARROLLO DE APLICACIÓN WEB SOBRE PROCESOS ADMINISTRATIVOS DE LOS ESTUDIANTES DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA -ASISTO-**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdía
Coordinador de Privados
y Revisión de Trabajos de Graduación



UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN
CIENCIAS Y SISTEMAS
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **“DISEÑO Y DESARROLLO DE APLICACIÓN WEB SOBRE PROCESOS ADMINISTRATIVOS DE LOS ESTUDIANTES DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA -ASISTO-”**, realizado por los estudiantes, NELSON DANIEL RAMOS GÓMEZ y CRISTIAN EMMANUEL SÁNCHEZ SALVADOR aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”


Ing. Marlon Antonio Pérez Turck
Director

Escuela de Ingeniería en Ciencias y Sistemas



Guatemala, 13 de mayo de 2019

Universidad de San Carlos
de Guatemala

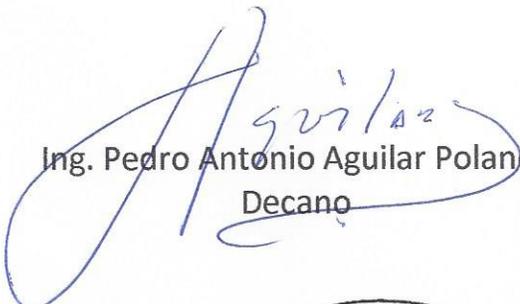


Facultad de Ingeniería
Decanato

DTG. 227-2019

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **DISEÑO Y DESARROLLO DE APLICACIÓN WEB SOBRE PROCESOS ADMINISTRATIVOS DE LOS ESTUDIANTES DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA –ASISTO-**, presentado por los estudiantes universitarios: **Nelson Daniel Ramos Gómez y Cristian Emmanuel Sánchez Salvador**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:


Ing. Pedro Antonio Aguilar Polanco
Decano

Guatemala, mayo de 2019

/gdech



ACTO QUE DEDICO A:

Mi madre

María Gómez, por ser la persona que me ha acompañado durante todo este trayecto pese a las adversidades que se presentaron y que con su amor, esfuerzo y paciencia me han permitido alcanzar hoy este logro.

Mi padre

Francisco Ramos (q. e. p. d.), que a pesar de esta distancia física que nos separa, siempre lo he sentido presente en mi vida y aunque nos faltaron muchas cosas por vivir juntos, sé que este logro hubiera sido muy especial para usted.

Mis hermanos

Enma Cecilia, Thelma Leticia, Lisela Liliana y especialmente a Francisco Leonel y Cecilia Beatriz, por estar conmigo en todo momento y que de una u otra forma me han brindado todo su apoyo a lo largo de mi vida.

Mis sobrinos

Que este logro que obtengo sirva de ejemplo para seguir forjando su futuro.

Mis amigos

Con quienes nos apoyamos mutuamente en nuestra formación profesional y hemos logrado llegar hasta acá.

Nelson Daniel Ramos Gómez

ACTO QUE DEDICO A:

Mi papá Dios

Por darme la sabiduría y las fuerzas para culminar esta meta y siempre estar conmigo, en cada desvelo, entrega de proyecto, calificación, alegrías y decepciones; a Él le debo lo que he logrado.

Mi madre

Olivia Salvador, por su amor incondicional y su incansable lucha para sacarnos adelante, su paciencia y fe de que llegaría este día, espero que te sientas orgullosa.

Mi padre

Israel Sánchez, por inculcarme a seguir esta hermosa carrera y por su ayuda económica.

Mi tío

Alfredo Salvador, por ser fuente de inspiración y parte fundamental en mi desarrollo profesional, por los muchos regaños y palabras de motivación que siguen siendo muy significativas en la toma de mis decisiones.

Mis hermanas

Jennifer y Katherine Sánchez que no importando la poca comunicación que tuvimos por el tiempo invertido en la carrera, yo sé que siempre estarían para mí, su existencia hace que mi vida sea mejor.

Mis amigos de infancia

David López, Samuel Ángel y Bryan Alvarado, por sus consejos y su amistad incondicional, que a pesar de mi situación económica, estaban dispuestos a contribuir para poder acompañarlos.

Mis amigos

Con quienes compartí a lo largo de la carrera, por brindarme su tiempo y disposición para ayudarme; en especial a mi compañero de tesis Nelson Ramos y a mi amiga de batalla Alina Urbina a los que les tengo mucho cariño y admiración.

Cristian Emmanuel Sánchez Salvador

AGRADECIMIENTOS A:

Mis padres

Por ser mi pilar fundamental y haberme apoyado incondicionalmente, que con su esfuerzo, dedicación y sacrificio, ayudaron a culminar esta etapa de mi vida, darles las gracias porque a través de su amor, sus consejos, valores y principios que me han inculcado, ayudaron a trazar mi camino.

Mis hermanos

Por haber sido mi apoyo durante todo este tiempo y haber aportado a mi formación tanto profesional y como ser humano.

Universidad de San de Guatemala

Por contribuir en mi formación académica y brindarme la oportunidad de formarme como profesional.

Facultad de Ingeniería

Por proporcionarme la orientación, los conocimientos y las habilidades necesarias para desempeñarme profesionalmente.

Mis amigos

Con quienes compartí a lo largo de la carrera, por su amistad, confianza y el apoyo que me brindaron, y por todo el tiempo que han compartido conmigo.

Nelson Daniel Ramos Gómez

AGRADECIMIENTOS A:

- Dios** Por darme la oportunidad de estar aquí, por el regalo tan grande que es mi graduación a nivel profesional, por cuidarme en el recorrido diario en bus hacia la universidad, por mi familia, por poner a los amigos correctos, la honra y la gloria sean a Él.
- Mis padres** Por confiar en mí, a que tarde o temprano llegarían este día, por sus muestras de amor y consejos que a pesar de las dificultades, siempre estaban para apoyarme.
- Mi familia** En general por cada consejo y corrección impartida, por más mínima que haya sido, fue de gran ayuda para culminar esta meta.
- Mi novia** Mónica Contreras, que me acompañó en las últimas instancias de mi carrera universitaria para culminarla con éxito.
- Mi iglesia** Por guiarme en mi vida espiritual y compartirme del amor de Dios y la unción del Espíritu Santo.

**Universidad de San
Carlos de Guatemala**

Por brindarme la oportunidad de estudiar en la Facultad de Ingeniería y facilitarme el acceso a la educación superior para ayudar a mi país.

Mis amigos

A todos en general, por formar parte de esta experiencia, les agradezco grandemente por su amistad brindada.

Cristian Emmanuel Sánchez Salvador

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	VII
LISTA DE SÍMBOLOS	XI
GLOSARIO	XIII
RESUMEN	XV
OBJETIVOS	XVII
INTRODUCCIÓN	XIX
1. ESTUDIO DE LA TECNOLOGÍA Y SU IMPACTO	1
1.1. Aplicaciones web	1
1.2. Crecimiento de aplicaciones web	1
1.3. Aplicaciones web adaptables	2
2. IDENTIFICACIÓN DEL PROBLEMA Y SOLUCIÓN PLANTEADA	5
2.1. Antecedentes	5
2.1.1. Acceso a la información	6
2.1.2. Información correcta y completa	6
2.1.3. Disminución de solicitudes de información	7
2.2. Mercado objetivo	7
2.3. <i>Benchmarking</i> aplicado a Asisto	8
2.3.1. Portal de la Facultad de Ingeniería	8
2.3.2. Problemas del portal de la Facultad de Ingeniería	9
2.3.3. Mejores prácticas del portal de la Facultad de Ingeniería	9
2.3.4. Comparación entre sitios	10

3.	DISEÑO DE LA APLICACIÓN BAJO LA NECESIDAD IDENTIFICADA	11
3.1.	Prototipo.....	11
3.1.1.	Aplicación administrativa	11
3.1.1.1.	Usuarios	13
3.1.1.2.	Guías	15
3.1.1.3.	Preguntas frecuentes	16
3.1.2.	Aplicación cliente	17
3.1.2.1.	Sección de guías.....	18
3.1.2.2.	Sección de preguntas frecuentes.....	19
3.2.	Formularios y validaciones	20
3.2.1.	Encuesta.....	20
3.2.2.	Resultados.....	23
3.3.	Diseño intuitivo y usabilidad	34
3.3.1.	Administrador.....	34
3.3.2.	Cliente.....	38
4.	DOCUMENTACIÓN BASE PARA EL DESARROLLO DE LA APLICACIÓN	41
4.1.	Tecnologías utilizadas	41
4.1.1.	NPM.....	41
4.1.2.	Node.js.....	41
4.1.3.	AngularJS	42
4.1.4.	Helmet.....	42
4.1.5.	JWT.....	42
4.1.6.	JSONParser.....	43
4.1.7.	Moment JS.....	43
4.1.8.	Path JS	43
4.1.9.	ExpressJS.....	43

4.1.10.	Mongoose	44
4.1.11.	API RESTful.....	44
4.1.12.	MongoDB	45
4.2.	Arquitectura de la aplicación.....	45
4.2.1.	Modelo vista controlador (MVC)	46
4.2.2.	Modelo	46
4.2.3.	Vista	47
4.2.4.	Controlador	47
4.3.	Modelo de datos	48
4.3.1.	Modelo de base de datos	49
4.3.2.	Descripción de documentos	51
4.3.2.1.	Guía	51
4.3.2.2.	Usuario	52
4.3.2.3.	Faq.....	53
4.3.2.4.	Dependencia	53
4.4.	Herramientas.....	53
4.4.1.	Postman.....	54
4.4.2.	Visual Studio Code	54
4.4.3.	Robomongo	54
4.4.4.	Nodemon	55
4.4.5.	PM2.....	55
4.5.	Hardware.....	55
4.5.1.	Desarrollo.....	55
4.5.2.	Hardware mínimo	56
4.5.3.	Hardware utilizado	56
4.5.4.	Implementación.....	56
4.6.	Tutorial de desarrollo	57
4.6.1.	Instalación y configuración de herramientas	57
4.6.1.1.	Node.js.....	57

4.6.1.2.	MongoDB.....	60
4.6.1.3.	PM2	61
4.6.1.4.	Nodemon.....	62
4.6.1.5.	Robomongo.....	63
4.6.1.6.	Postman	67
4.6.1.7.	Visual Studio Code.....	68
4.6.2.	Desarrollo	71
4.6.2.1.	Configuración del proyecto	72
4.6.2.2.	Conceptos generales para el desarrollo.....	79
4.6.2.2.1.	<i>Blocking</i> vrs <i>non blocking</i> I/O	79
4.6.2.2.2.	<i>Callbacks</i>	80
4.6.2.2.3.	Promesas.....	80
4.6.2.2.4.	<i>Async</i> y <i>await</i>	80
4.6.2.2.5.	Ciclo de vida de un proceso	81
4.6.2.2.6.	Archivo de configuración package.json.....	82
5.	RESULTADOS.....	85
5.1.	Información recabada	85
5.1.1.	Portal de ingeniería.....	85
5.1.2.	Registro y estadística	85
5.1.3.	Control académico.....	86
5.2.	Accesibilidad y adaptabilidad.....	86

CONCLUSIONES 89
RECOMENDACIONES..... 91
BIBLIOGRAFÍA..... 93

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Pantalla de inicio de sesión	12
2.	Pantalla principal, administrador Asisto.....	13
3.	Nuevo usuario.....	14
4.	Administración de usuario	15
5.	Administración de guías	16
6.	Administración de preguntas frecuentes	17
7.	Pantalla principal cliente Asisto	18
8.	Sección de guías	19
9.	Sección de preguntas frecuentes	20
10.	Selección de tipo de estudiante.....	21
11.	Estudiantes de primer ingreso	21
12.	Semestre en el cual se encuentra	22
13.	Menor al séptimo semestre	22
14.	Mayor al octavo semestre.....	23
15.	Gráfica de respuestas a la pregunta 1	24
16.	Respuestas a la pregunta 1, de primer ingreso	25
17.	Respuestas a la pregunta 2, de primer ingreso	26
18.	Pregunta 3 (menor al séptimo semestre)	27
19.	Pregunta 1 (menor al séptimo semestre)	28
20.	Pregunta 2 (menor al séptimo semestre)	29
21.	Pregunta 2 (menor al séptimo semestre)	30
22.	Pregunta 3 (menor al séptimo semestre)	31
23.	Pregunta 1 (mayor al octavo semestre)	32

24.	Pregunta 2 (mayor al octavo semestre)	33
25.	Pregunta 3 (mayor a octavo semestre)	34
26.	Login sitio administrativo	35
27.	Login	35
28.	Creación de usuario.....	36
29.	Creación de dependencias.....	36
30.	Listado de guías	37
31.	Creación de guías.....	37
32.	Creación de preguntas frecuentes	38
33.	Pagina inicial, sitio cliente.....	39
34.	Guías	39
35.	Preguntas frecuentes.....	40
36.	Arquitectura, aplicación Asisto	48
37.	Documento guía	49
38.	Documento administrador.....	50
39.	Documento FAQ.....	50
40.	Documento dependencia.....	51
41.	Asistente de configuración de Node.js	58
42.	Configuración de directorio.....	59
43.	Configuración de instalación de Node.js	59
44.	Verificación de la instalación de Node.js.....	60
45.	Configuración del servicio de MongoDB	61
46.	Instalación de herramienta PM2.....	62
47.	Instalación de herramienta Nodemon	63
48.	Asistente de configuración de Robomongo.....	64
49.	Ventana de configuración de Robomongo.....	65
50.	Configuración de conexión Robomongo	66
51.	Conexión establecida a través de Robomongo.....	67
52.	Herramienta Postman en funcionamiento	68

53.	Asistente de instalación de Visual Studio Code	69
54.	Directorio de instalación de Visual Studio Code	70
55.	Editor de texto Visual Studio Code	71
56.	Configuración inicial del proyecto	72
57.	Verificación de instalación de npm	73
58.	Creación de proyecto Node.js	73
59.	Archivo de configuración, package.json generado.....	74
60.	Configuración package.json sin dependencias	74
61.	Instalación de dependencia helmet	75
62.	Instalación de dependencia path.....	75
63.	Instalación de dependencia jsonwebtoken.....	76
64.	Instalación de dependencia momento.....	76
65.	Instalación de dependencia mongoose	77
66.	Instalación de dependencia body-parser.....	77
67.	Instalación de dependencia express	78
68.	Configuración package.json con dependencias.....	78
69.	Ciclo de vida de un proceso en Node.js	82

TABLAS

I.	Comparación entre sitios	10
II.	Lanzamiento de Asisto	87

LISTA DE SÍMBOLOS

Símbolo	Significado
GB	Gigabyte
GHz	Gigahercio
TB	Terabyte

GLOSARIO

Aplicación web	Herramienta de información a la cual se tiene acceso por medio de internet.
Bootstrap	Herramienta para el desarrollo de aplicaciones web la cual permite que la aplicación se adapte a distintos dispositivos.
Dominio	Nombre con el cual se identifica de forma única a un sitio web en internet.
<i>Framework</i>	Conjunto de herramientas que permite desarrollar de forma ágil una aplicación o proyecto.
GIT	Repositorio virtual para el control de versiones.
GUI	Interfaz gráfica.
Host	Sito en el cual se establece una aplicación web.
HTML	Lenguaje de programación para el desarrollo de páginas en internet.
HTTP	Protocolo utilizado por los sitios en internet.

JavaScript	Lenguaje de programación que extiende las funcionalidades de una página web haciéndolo dinámico.
JSON	Notación para la estructuración de datos.
NoSQL	Sistema de gestión de base de datos no relacionales.
Puerto de red	Interfaz para la comunicación de una aplicación por medio de una red.
RAM	Memoria volátil de acceso aleatorio.
Sistema operativo	Software que gestiona todos los recursos de un sistema.
SSD	Disco duro de estado sólido.
vCPU	Unidad central de procesamiento virtual.

RESUMEN

Asisto es una aplicación web que surge a partir de las necesidades observadas sobre la poca información que tienen los estudiantes sobre los procesos administrativos y académicos que existe actualmente en la Facultad de Ingeniería; esta situación llevó a elaborar este trabajo de graduación: *diseño y desarrollo de aplicación web sobre procesos administrativos de los estudiantes de la facultad de ingeniería de la Universidad de San Carlos de Guatemala –Asisto-*, que consiste en el desarrollo de una aplicación web.

El presente trabajo de tesis consta de cinco capítulos: en el primero se define el impacto de las aplicaciones web informativas. El segundo presenta las tres principales dificultades para el estudiante al obtener información sobre los procesos y de qué forma se les puede dar solución; también, muestra a que población está enfocada la aplicación Asisto. El tercer capítulo presenta una propuesta para la solución que se ha planteado enfocándose en el prototipo de la aplicación administrativa y el cliente. En el cuarto capítulo se describen a detalle todas las tecnologías y herramientas utilizadas en el desarrollo de la aplicación.

El quinto capítulo demuestra los resultados obtenidos al poner en marcha el proyecto y de cómo la aplicación fue alimentada por las diferentes fuentes de información a partir de las cuales se construyeron las guías integradas en Asisto.

OBJETIVOS

General

Brindar al estudiante información verídica y eficaz sobre diversos procesos administrativos o académicos de la Facultad de Ingeniería con la ayuda de una aplicación web amigable que centralice una serie de guías con los pasos para llevar a cabo procesos de la carrera universitaria.

Específicos

1. Centralizar en una aplicación web los procesos académicos y administrativos de al menos tres fuentes distintas de la Facultad de Ingeniería.
2. Construir una aplicación web que se pueda adaptar a los diferentes tipos de dispositivos desde los cuales esta sea consultada.
3. Crear un sistema administrativo para la creación y modificación de guías y preguntas frecuentes bajo la supervisión de personas pertinentes de cada escuela en la Facultad de Ingeniería.

INTRODUCCIÓN

Asisto es una aplicación web de ayuda para el estudiante de la Facultad de Ingeniería, provee información de suma importancia sobre los procesos académicos y administrativos que se visualizan en forma de guías conformadas por una serie de pasos que indican al estudiante a detalle su realización.

Asisto surge a partir de las necesidades observadas sobre la poca información que tienen los estudiantes sobre los procesos que existen actualmente en la Facultad de Ingeniería.

La característica principal de la aplicación web Asisto consta de proveer al estudiante información verídica, de fácil acceso y sobre todo amigable, la cual está conformada de varios procesos tomados de distintos sitios que pertenecen a distintas unidades académicas y escuelas de la Facultad de Ingeniería; lo cual lleva a centralizar y reestructurar la información para que sea ingresada en Asisto, esto con la ayuda de un encargado al que se le asigna un sitio administrativo para su control.

1. ESTUDIO DE LA TECNOLOGÍA Y SU IMPACTO

1.1. Aplicaciones web

Una aplicación web es una herramienta informática cuya ejecución es llevada a cabo por medio de un navegador ya sea a través de internet o bien por medio de una red local.

Funcionan como un tipo de software que se codifica con diferentes tipos de lenguajes los cuales son procesados y almacenados dentro de una red para que puedan ser accedidos por los usuarios y así tener una relación activa entre ellos y la información.

Una aplicación web proporciona diversas facilidades al usuario a través de internet; las aplicaciones web se han convertido en herramientas de suma importancia para todos los usuarios alrededor del mundo, donde ya sea empresas grandes o desarrolladores individuales, muestran sus ideas innovadoras y acercan a los usuarios al universo tecnológico que presentan una gran ayuda en el diario vivir a través de un clic.

1.2. Crecimiento de aplicaciones web

El auge de las aplicaciones web está creando tendencias en el mundo, desde 1990, año en el que fue creado el primer sitio web por el británico Tim Berners-Lee, en el Centro de Investigaciones Nucleares, CERN hasta el año actual, 2019; se ha visto un crecimiento exponencial en los sitios web que circulan a través del internet.

Actualmente, según la encuesta del servidor web de Netcraft realizada en enero de 2018 existen más de 1,8 mil millones de sitios web; de las cuales para agosto de 2016, según un artículo publicado por *Prensa Libre*, se contabilizaron aproximadamente 17 mil 555 dominios con el sufijo .gt; esto sin tomar en cuenta la cantidad de páginas web hechas en Guatemala utilizando otro sufijo; hoy por hoy, esa cifra ha aumentado y más con el surgimiento de nuevos desarrolladores que proporcionan sitios más atractivos en los que plantean soluciones para el diario vivir. Aunque se es una minoría en creación de páginas web, se forma parte de ese vasto conjunto de sitios que se encuentran en internet.

1.3. Aplicaciones web adaptables

Debido a la gran cantidad de distintos tipos de dispositivos por medio de los cuales se acceden a los sitios o aplicaciones web, se debe mantener una correcta visualización de las páginas para cada uno. Es por esta necesidad que surge el concepto de diseño adaptable, adaptativo o responsivo.

Se define el diseño adaptable como una técnica para el diseño de sitios web que posean la capacidad de redimensionar y ubicar todo su contenido de tal forma que estos se puedan visualizar de una manera correcta y brindar una mejor experiencia al usuario, sin importar el tamaño del dispositivo desde el cual se visualiza el sitio, ya sea que este acceda desde una computadora o un *smartphone*.

Una de las características principales de las aplicaciones web adaptables, es que cada uno de los elementos que la conforman son dinámicos, es decir, tienen la capacidad de aumentar o disminuir su tamaño, desaparecer, según el dispositivo. Lo que se busca al diseñar una aplicación web adaptable, es que

esta pueda mantener el mismo aspecto independientemente de donde sea visualizada, sin perder la usabilidad.

Hoy en día el uso de dispositivos móviles tales como *smartphone* o tabletas, ha crecido considerablemente, es por lo mismo que el acceso a los sitios por medio de estos también ha aumentado, por lo cual se requiere más desarrollar aplicaciones web con diseño adaptable. Según una encuesta realizada en el año 2017 por Google Consumer Barometer en 63 países, más del 50 % de los usuarios conectados a internet lo realiza a través de móviles.

Con la tendencia del uso de dispositivos móviles, nace un concepto dentro del diseño responsivo, al cual llaman Mobile-First, que consiste en que el desarrollo de un sitio debe enfocarse en una versión optimizada para dispositivos móviles. Mobile-First no define únicamente el aspecto visual de un sitio, también, la fluidez y el rendimiento que este posee, ya que los dispositivos móviles poseen una limitante en cuanto a memoria y capacidad de procesamiento.

A partir del año 2018 Google ha implementado una nueva técnica con la cual realiza la indexación de los sitios web, en su buscador, llamada Mobile First Index, la cual toma como referencia la versión móvil del sitio, que da prioridad a los sitios que emplean ambas versiones, para que estos sean los primeros en ser presentados al momento que un usuario realice una búsqueda.

2. IDENTIFICACIÓN DEL PROBLEMA Y SOLUCIÓN PLANTEADA

2.1. Antecedentes

En la Facultad de Ingeniería existen varios procesos que los estudiantes deben realizar a lo largo de su periodo universitario; pero en muchas ocasiones el estudiante carece de información para realizar dichos procesos.

La Facultad de Ingeniería cuenta con un reglamento interno con cierta información de algunos procesos; sin embargo, hay procesos que no incluye o se encuentran desactualizados.

Otra opción de ayuda al estudiante es el centro de información de la Facultad de Ingeniería, el cual es otro medio donde el estudiante puede adquirir asesoría ya sea del proceso o también de lugares en donde se pueda avocar para obtener la información necesaria para realizar dicho proceso.

Por lo general, cada escuela o unidad académica brinda información acerca de los procesos estudiantiles y en ocasiones están disponibles en el sitio web de cada uno de estos; sin embargo, la manera más común en obtener dicha información sobre un proceso estudiantil es por medio de compañeros que hayan realizado algún proceso con anterioridad que provoca muchas veces el traslado de información incorrecta.

2.1.1. Acceso a la información

El problema principal que afrontan los estudiantes es la carencia de información para la realización de procesos universitarios, ya que dentro de la facultad no existe un lugar específico o un sitio donde toda la información esté centralizada. Es así como nace la idea de Asisto, un sitio web con el objetivo de proveer al estudiante todos los pasos necesarios de un proceso, por medio de una misma fuente y que además, esta información sea verídica y de fácil acceso.

Los estudiantes no tienen acceso a la información de diversos procesos específicos utilizados a lo largo de la carrera de Ingeniería en Ciencias y Sistemas y a diversos procesos generales de ingeniería.

Se realizó una encuesta en la que se preguntaba al estudiante, si le gustaría una aplicación interactiva que informara acerca de todos los procesos de la facultad, en la cual de 128 estudiantes, el 95,2 % respondió que sí, estudiantes a partir del séptimo semestre; esto indica que en la mayoría de casos, el estudiante está sumamente interesado en un lugar (sitio web Asisto) donde pueda acceder a la información sobre los procesos estudiantiles.

2.1.2. Información correcta y completa

En el recorrido de la etapa universitaria, el estudiante se topa con muchos procesos, muchas ocasiones en la búsqueda de los pasos a seguir de diversos procesos el estudiante se encuentra con varias dudas, las cuales se presentan por la poca información que provee la Facultad de Ingeniería.

Según la encuesta realizada, de los estudiantes a partir del séptimo semestre, el 77,6 % dijo haber sido orientado por otros compañeros para realizar los diversos procesos estudiantiles; esto indica que hay un problema al momento de propagar información, los estudiantes se abocan con sus compañeros y puede que la información recabada no sea del todo correcta o completa.

2.1.3. Disminución de solicitudes de información

En el centro de información de la Facultad de Ingeniería existe gran cantidad de preguntas que se realizan para un mismo proceso, así como se ve afectado el buzón de correo de la persona a cargo de proporcionar información fehaciente de diversos procesos, ya que los estudiantes por medio de correo electrónico solicitan información.

La información existe y muchas veces se puede encontrar en el reglamento interno; pero los estudiantes de primer ingreso los indican que en un 0 % no había leído el reglamento interno, 0 % no se había abocado al centro de información y para la realización de los procesos pertinentes a su grado universitario, le habían consultado a sus compañeros.

Al existir un sitio web en donde se pueda consultar de forma agradable los procesos estudiantiles, se evitará la desinformación de los estudiantes.

2.2. Mercado objetivo

El mercado objetivo de Asisto son todos los estudiantes de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala.

El propósito de esta aplicación web es proveer a los estudiantes información verídica sobre los procesos estudiantiles que se realizan a lo largo de la carrera. Al hablar de los estudiantes, se refiere a los estudiantes de la Facultad de Ingeniería, sin embargo, habrá procesos exclusivamente para la Escuela de Ciencias y Sistemas; posteriormente, se espera que se puedan integrar procesos exclusivos para las demás carreras.

2.3. *Benchmarking* aplicado a Asisto

Benchmarking es una práctica que ayuda a comparar la herramienta con otras del mismo tipo para proveer mejoras y dar nuevas ideas para competir con las existentes en el mercado con la finalidad de exponer limitantes y beneficios que Asisto posee.

Existen sitios web, pertenecientes a las distintas escuelas y unidades académicas, en las que se puede obtener información de los procesos estudiantiles que cada uno de estos cubren; sin embargo, la Facultad de Ingeniería cuenta con un portal, que provee gran información al estudiante.

2.3.1. Portal de la Facultad de Ingeniería

Este sitio cuenta con una sección para que los estudiantes puedan acceder por medio de sus credenciales, desde el cual se tiene información detallada sobre los cursos aprobados, asignados, desasignados, laboratorios vigentes. Además, se pueden generar órdenes de pago para cursos de vacaciones y exámenes de retrasadas, así como la asignación de cursos y exámenes.

Fuera de la sección anterior, el sitio cuenta con accesos por medio de enlaces hacia los sitios web de cada una de las escuelas y unidades académicas; provee información de las redes de estudio de cada carrera además se tiene acceso a los normativos tanto de la facultad como de la universidad, horarios de cursos, calendarización del semestre, información destacada y preguntas frecuentes.

2.3.2. Problemas del portal de la Facultad de Ingeniería

Si bien el sitio de la Facultad de Ingeniería provee gran cantidad de información y servicios, posee carencias en cuanto a la disposición de información sobre los procesos estudiantiles, que sería de utilidad para los estudiantes.

- No posee una sección de información de procesos estudiantiles.
- El sitio no es adaptable a cualquier dispositivo.
- Centraliza únicamente por medio de enlaces, los sitios de las escuelas y unidades académicas, pero no centraliza en una sección del sitio la información de sus procesos.

2.3.3. Mejores prácticas del portal de la Facultad de Ingeniería

El sitio de la facultad posee características que lo hacen destacar, y dan distinción, lo cual le provee usabilidad; cubre de esta manera algunas necesidades de los estudiantes.

- Posee una sección para el inicio de sesión en el cual se tiene toda la información personal y de cursos del estudiante.
- Provee información general sobre las carreras e información de interés para el estudiante.
- Acceso a otros sitios por medio de este.

2.3.4. Comparación entre sitios

La comparación entre Asisto y el portal de la Facultad de Ingeniería, se realiza principalmente para observar ciertos aspectos y características que son cubiertos por ambas aplicaciones, así como identificará carencias para establecer mejoras en la aplicación.

Tabla I. **Comparación entre sitios**

Sitio web	Información del estudiante	Centralización de procesos estudiantiles	Preguntas frecuentes	Sitio adaptable
Asisto		X	X	X
Portal Facultad de Ingeniería	X		X	

Fuente: elaboración propia.

3. DISEÑO DE LA APLICACIÓN BAJO LA NECESIDAD IDENTIFICADA

3.1. Prototipo

Asisto está conformado por un sitio cliente al que tienen acceso todos los estudiantes de la Facultad de Ingeniería, en el cual se muestran las guías de los procesos que se llevan a cabo y un sitio administrativo donde se tiene el control de usuarios, guías y preguntas frecuentes que se proveen a los estudiantes.

A continuación, se muestran los prototipos de ambas aplicaciones web, donde se detallan las funcionalidades de cada una.

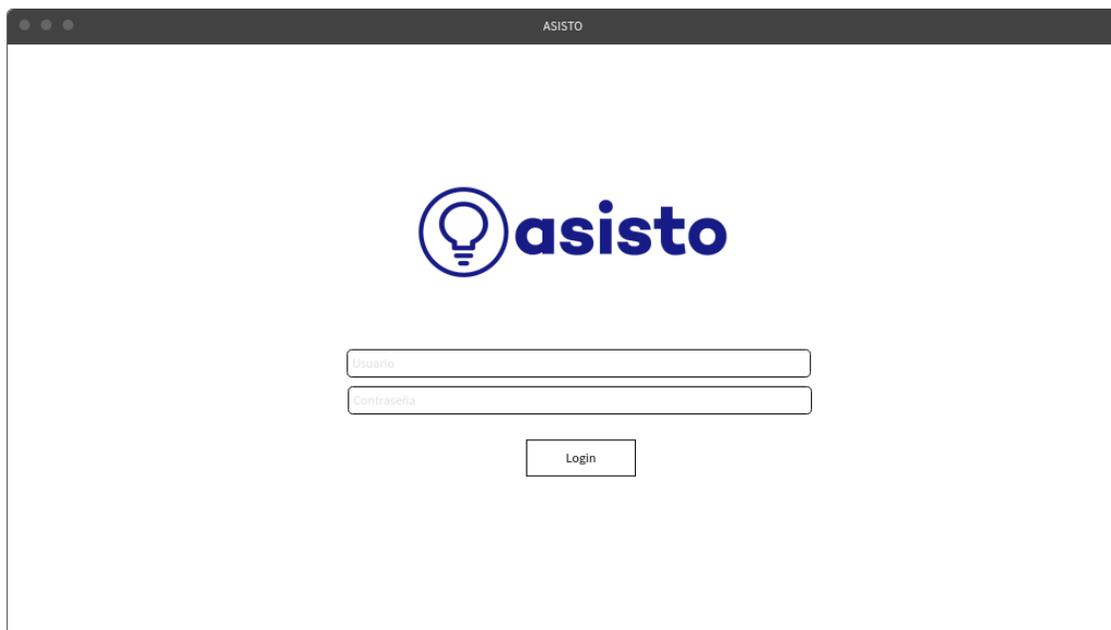
3.1.1. Aplicación administrativa

Esta aplicación cuenta con una página de inicio de sesión como se muestra en la figura 1, en el cual el usuario ingresa sus credenciales, que según sea el rol, este tendrá acceso a distintas secciones del sitio. Se han establecido tres roles, los cuales se describen a continuación:

- **Administrador:** este rol permite al usuario tener el control de todo el sistema, administración de usuarios, guías y preguntas frecuentes.
- **Guía:** este rol permite que el usuario administre las guías y preguntas frecuentes.

- Auxiliar: este rol permite al usuario únicamente administrar las preguntas frecuentes.

Figura 1. **Pantalla de inicio de sesión**



Fuente: elaboración propia, empleando Asisto.

Al ingresar a la aplicación, se tiene la pantalla principal que se muestra en la figura 2, en la que se despliega un menú en la parte inferior, en el cual se tiene acceso a las secciones según sea el rol del usuario que haya iniciado sesión.

Figura 2. **Pantalla principal, administrador Asisto**



Fuente: elaboración propia, empleando Asisto.

3.1.1.1. Usuarios

Esta sección está conformada por dos apartados, en las que por facilidad se establece un menú para agregar nuevos usuario y otro para su administración.

En el primer apartado existe un formulario como se muestra en la figura 3, en el cual se ingresan el nombre del usuario y sus credenciales.

Figura 3. **Nuevo usuario**

The screenshot shows a mobile application interface for creating a new user. At the top, there is a dark header with the word "ASISTO" in the center. Below the header, the screen is divided into two sections. On the left, there is a back arrow icon followed by the text "ATRAS". On the right, the text "NUEVO USUARIO" is displayed. The main content area contains three input fields, each with a label above it: "NOMBRE", "USUARIO", and "CONTRASEÑA". Below these fields is a button labeled "CREAR".

Fuente: elaboración propia, empleando Asisto.

Para la administración de los usuario que se han creado, se ha establecido un apartado específico, donde muestra un listado de todos estos, como se muestra en la figura 4 y al ser seleccionados muestran la información de cada uno de los usuarios, información que puede ser editada, así como realizar un cambio de contraseña y cambio de rol.

Figura 4. **Administración de usuario**

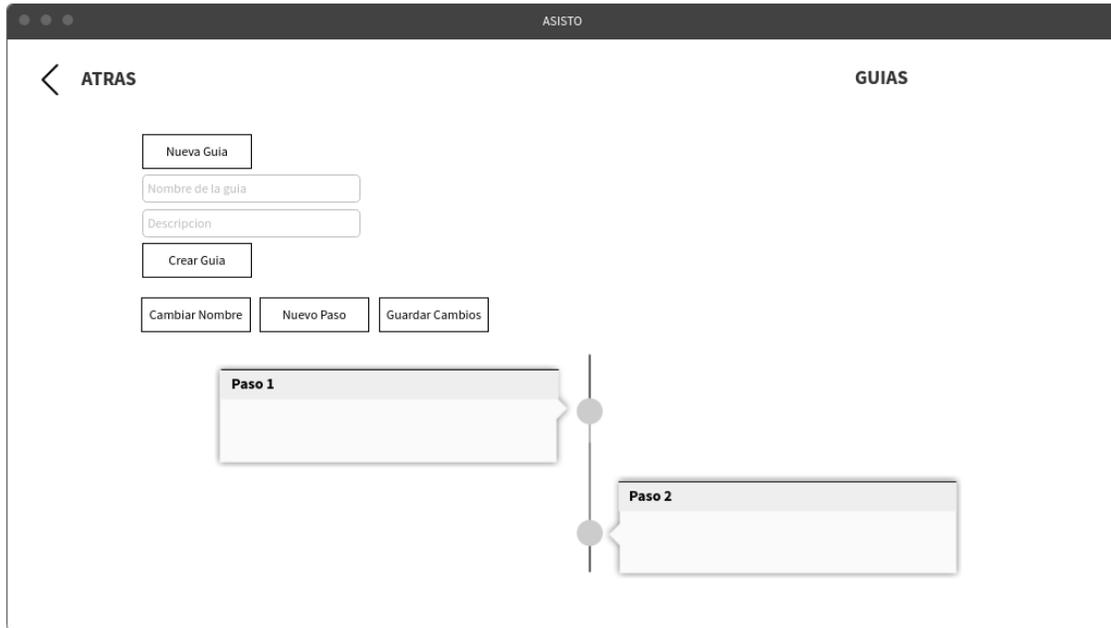
The screenshot shows a mobile application interface for user management. At the top, there is a dark header with the text 'ASISTO'. Below the header, there is a navigation bar with a back arrow and the text 'ATRAS' on the left, and 'USUARIOS' on the right. The main content area is divided into two columns. The left column displays the user's name 'Administrador ASISTO' and role 'Administrador', followed by the user type 'Usuario' and 'Guía'. The right column displays the user's name 'Administrador ASISTO' and a 'Cambiar' button. Below this, there is a section for 'Activo' with the value 'Si'. The 'Fecha de Registro' is '20-06-2018'. The 'Nueva Contraseña' section has an input field and a 'Cambiar' button. The 'Tipo de Administrador' section has two buttons: 'Guía' and 'Auxiliar'.

Fuente: elaboración propia, empleando Asisto.

3.1.1.2. **Guías**

En esta sección se crean y editan las guías y los pasos que la conforman. Cuando la guía es agregada, cada uno de sus pasos son creados y se visualizan, tal y como se ve en la aplicación cliente, esto para que el usuario tenga una vista previa de la guía, como se muestra en la figura 5.

Figura 5. **Administración de guías**



Fuente: elaboración propia, empleando Asisto.

3.1.1.3. Preguntas frecuentes

En esta sección se crean y editan las preguntas frecuentes, que al igual que las guías, estas son agregadas mostrando una vista previa al usuario.

Figura 6. **Administración de preguntas frecuentes**

The screenshot shows a web application interface for managing frequent questions. At the top, there is a dark header with the text 'ASISTO' on the right. Below the header, the page has a white background. On the left side, there is a back arrow icon followed by the text 'ATRAS'. On the right side, the text 'PREGUNTAS FRECUENTES' is displayed. The main content area contains a form with the following elements: a 'Nueva FAQ' button, a text input field labeled 'Escribir Pregunta', another text input field labeled 'Escribir Respuesta', and a 'Guardar' button.

Fuente: elaboración propia, empleando Asisto.

3.1.2. **Aplicación cliente**

Esta sitio web está disponible para todos los estudiantes de la Facultad de Ingeniería, donde se muestran las guías de los procesos más comunes y las preguntas frecuentes que los estudiantes tienen; la página principal tiene un menú inferior en el que se dispone el acceso a guías y a preguntas frecuentes, como se muestra en la figura 6.

Figura 7. **Pantalla principal cliente Asisto**

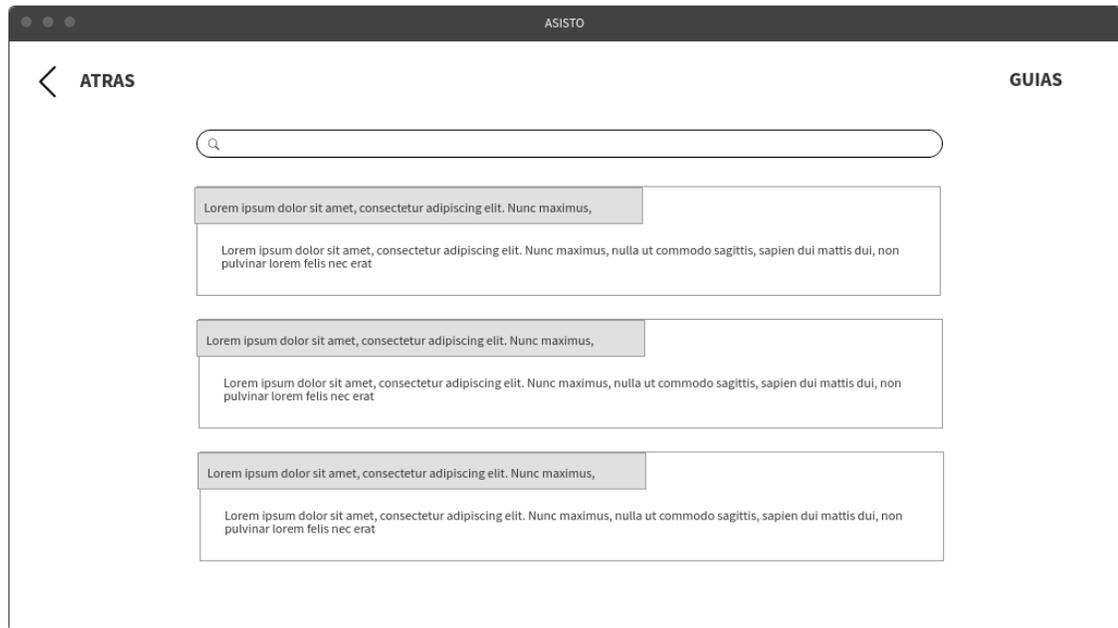


Fuente: elaboración propia, empleando Asisto.

3.1.2.1. Sección de guías

En esta sección se muestran todas las guías, que han sido creadas en el sitio administrativo; se cuenta una barra de búsqueda para tener un acceso rápido a una guía específica; cuando una guía es seleccionada se muestra un diagrama en el cual se detalla sus pasos.

Figura 8. Sección de guías

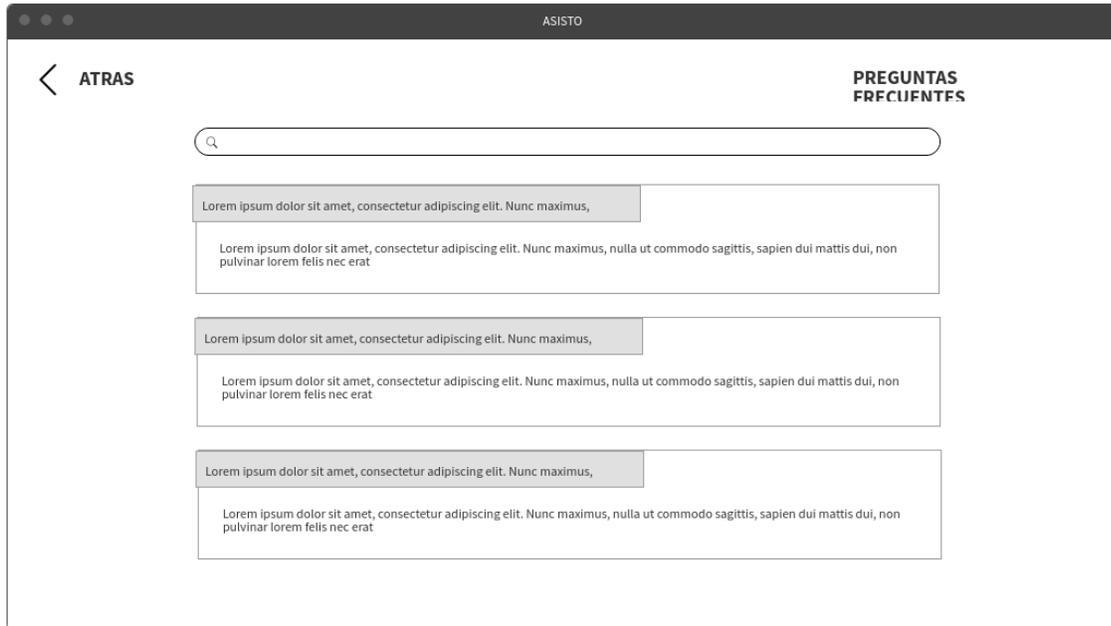


Fuente: elaboración propia, empleando Asisto.

3.1.2.2. Sección de preguntas frecuentes

En esta sección se muestran todas las preguntas frecuentes, se dispone de una barra de búsqueda si el estudiante necesita información sobre una pregunta específica.

Figura 9. Sección de preguntas frecuentes



Fuente: elaboración propia, empleando Asisto.

3.2. Formularios y validaciones

Por medio de una encuesta en línea se obtuvo un acercamiento con los clientes potenciales, en este caso los estudiantes de la Facultad de Ingeniería, a que se le realizó un serie de preguntas sobre la situación actual de la disponibilidad de información de los procesos estudiantiles, según el tipo de estudiante (primer ingreso o reingreso).

3.2.1. Encuesta

La estructura de la encuesta se muestra a continuación:

Figura 10. Selección de tipo de estudiante

Procesos estudiantiles

Esta encuesta fue hecha con el fin de recopilar datos acerca de la información que tienen los estudiantes con respecto a temas de diversos procesos en la facultad de ingeniería.

***Obligatorio**

Usted es estudiante de: *

primer ingreso

Re ingreso

Fuente: elaboración propia, empleando Asisto.

Figura 11. Estudiantes de primer ingreso

Primer Ingreso

¿Cuáles de los siguientes procesos estudiantiles, usted tiene conocimiento de cómo realizarlos?

Inscripción re ingreso

Asignación de cursos

Asignación de cursos vacaciones

Petición certificado de cursos

Otro

Ninguno

¿Cómo se informó sobre los procesos anteriores?

Reglamento interno de la Facultad de Ingeniería

Centro de información

Compañeros

Otro

¿Le gustaría una aplicación interactiva que informara sobre todos los procesos relacionados con la facultad?

Sí

No

Fuente: elaboración propia, empleando Asisto.

Figura 12. **Semestre en el cual se encuentra**

Reingreso
¿En qué rango se encuentra en la carrera?
<input type="radio"/> <= 7mo semestre
<input type="radio"/> >= 8vo semestre

Fuente: elaboración propia, empleando Asisto.

Figura 13. **Menor al séptimo semestre**

Menor al séptimo semestre
¿Cuáles de los siguientes procesos estudiantiles, usted tiene conocimiento de cómo realizarlos?
<input type="checkbox"/> Cursos por suficiencia
<input type="checkbox"/> Retrasada única
<input type="checkbox"/> Desasignación de cursos
<input type="checkbox"/> Asignación curso de vacaciones
<input type="checkbox"/> Otro
<input type="checkbox"/> Ninguno
¿Cómo se informó sobre los procesos anteriores?
<input type="checkbox"/> Reglamento interno de la facultad de Ingeniería
<input type="checkbox"/> Centro de información de Ingeniería
<input type="checkbox"/> Compañeros
<input type="checkbox"/> Otro
¿Le gustaría una aplicación interactiva que informara sobre todos los procesos relacionados con la facultad? *
<input type="radio"/> Sí
<input type="radio"/> No

Fuente: elaboración propia, empleando Asisto.

Figura 14. **Mayor al octavo semestre**

Mayor al octavo semestre

¿Cuáles de los siguientes procesos estudiantiles, usted tiene conocimiento de cómo realizarlos?

- Cierre de pensum
- Curso pre y post
- Auxiliatura
- Eps o tesis

¿Cómo se informó sobre los procesos anteriores?

- Reglamento interno de la Facultad de Ingeniería
- Centro de información de ingeniería
- Compañeros
- Otro

¿Le gustaría una aplicación interactiva que informara sobre todos los procesos relacionados con la facultad?

- Sí
- No

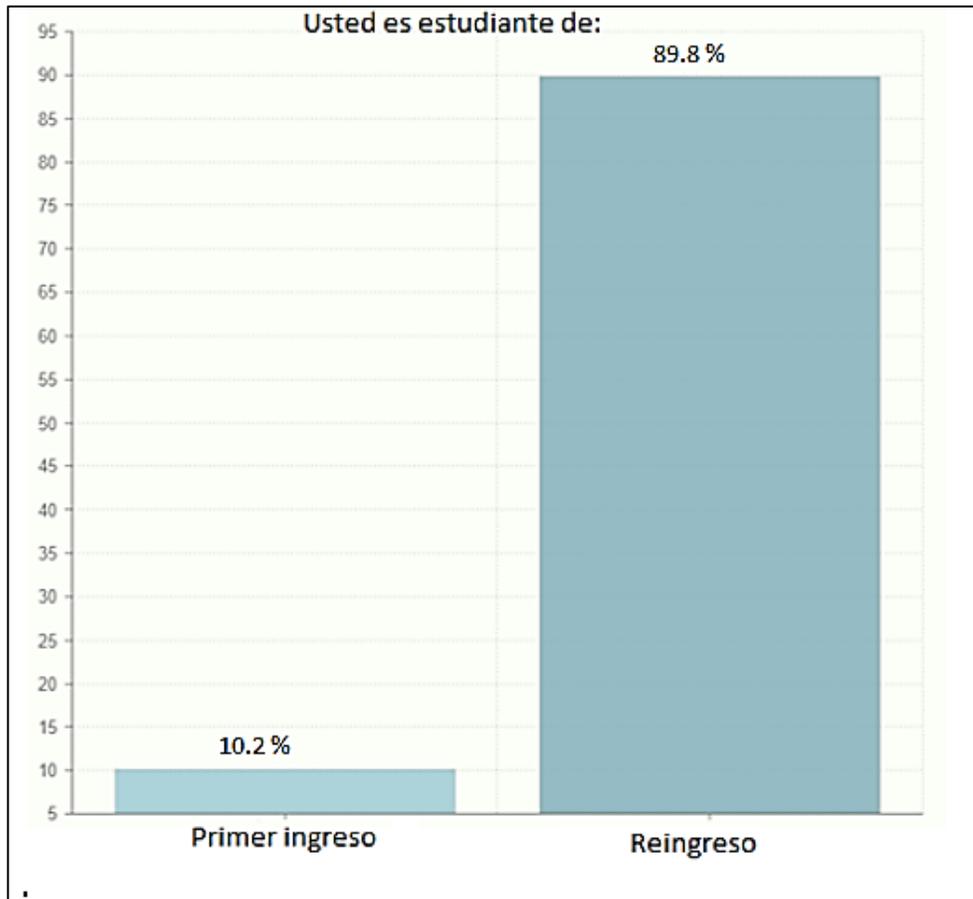
Fuente: elaboración propia, empleando Asisto.

3.2.2. **Resultados**

La encuesta fue realizada con un total de 127 estudiantes; basados en esto se incluye lo siguiente:

Como se muestra en la figura 15, los resultados de la pregunta indican que de los estudiantes encuestados, 13 son de primer ingreso y 114 son de reingreso; es decir, la mayoría de los encuestados son estudiantes de reingreso.

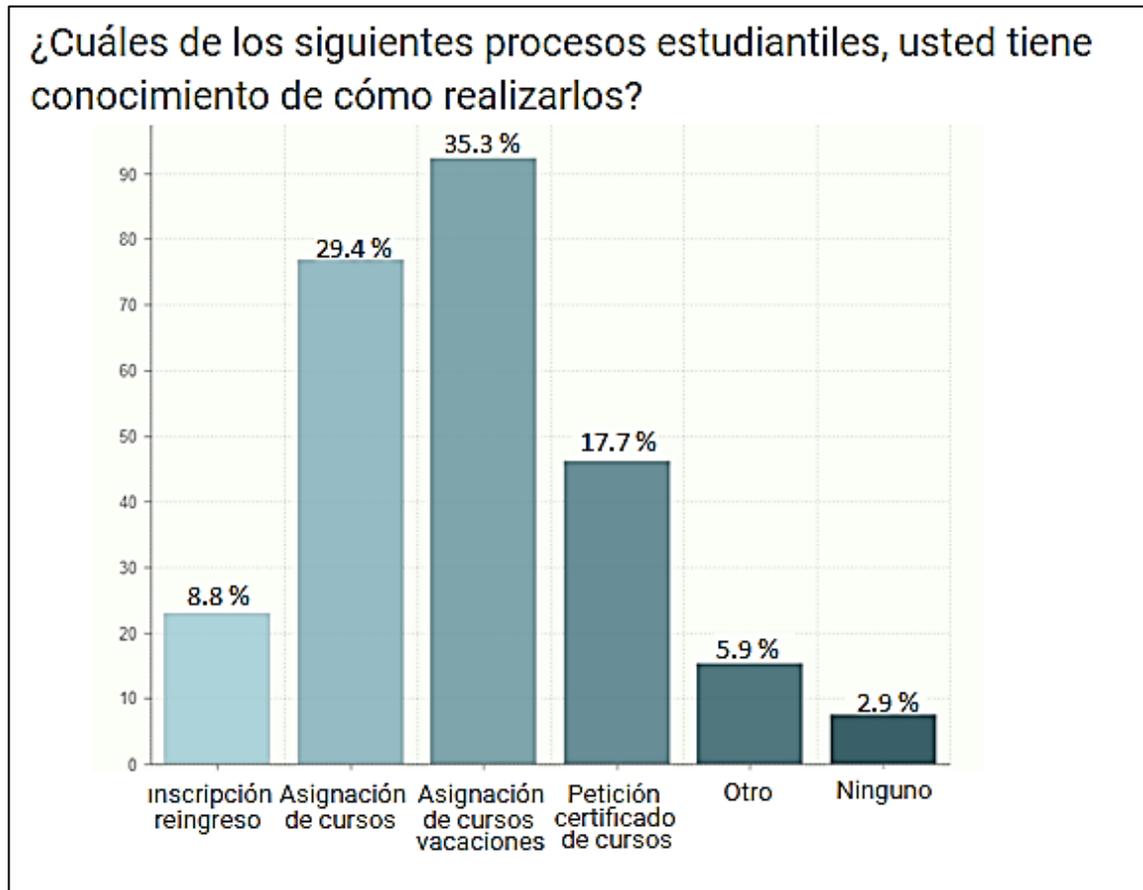
Figura 15. **Gráfica de respuestas a la pregunta 1**



Fuente: elaboración propia, empleando Asisto.

Se describieron cuatro procesos estudiantiles más comunes en la etapa de primer ingreso; dio como resultado que la mayoría de los estudiantes de primer ingreso saben cómo realizar el proceso de asignación de cursos de vacaciones, un proceso bastante común en la facultad; mientras que en menor cantidad tienen el conocimiento del proceso de inscripción de reingreso. Otro pequeño porcentaje tiene el conocimiento de otros procesos adicionales; y en un menor porcentaje no saben cómo realizar ningún proceso estudiantil.

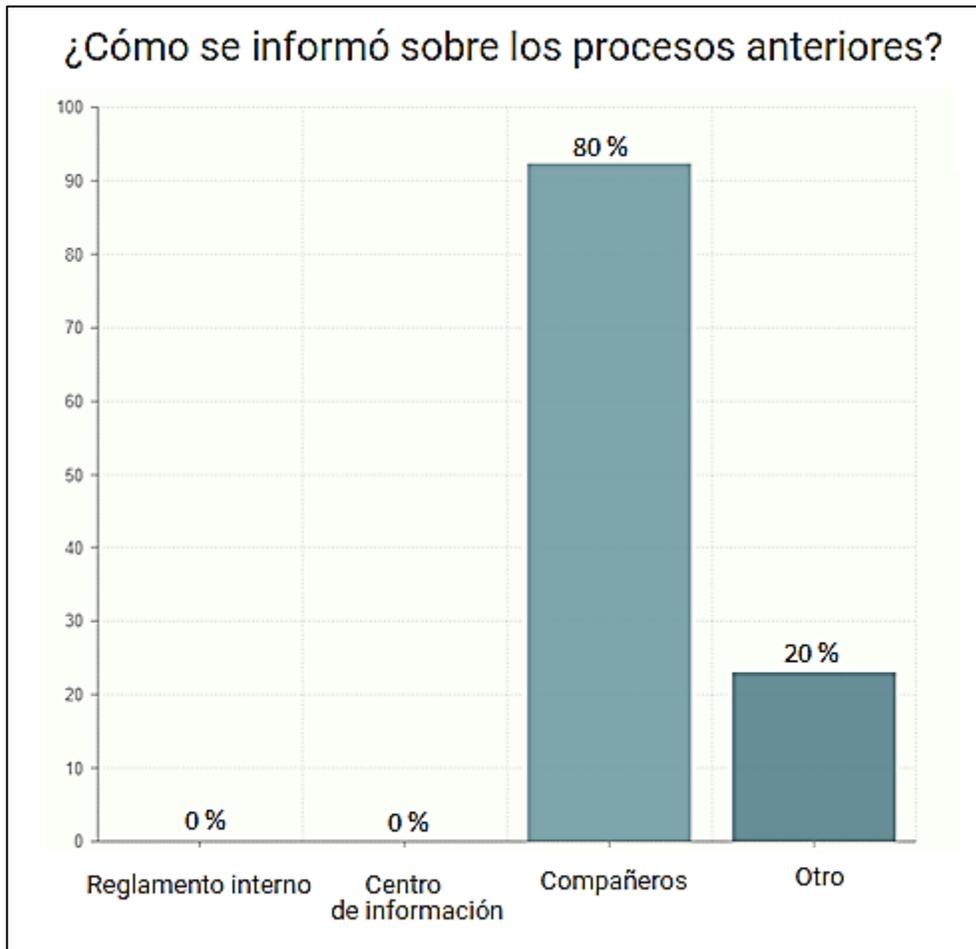
Figura 16. Respuestas a la pregunta 1, de primer ingreso



Fuente: elaboración propia, empleando Asisto.

La mayoría de estudiantes de primer ingreso se ha informado de los procesos estudiantiles por medio de sus compañeros; mientras que un pequeño porcentaje se ha informado por otro medio; se deja plasmado que no han obtenido información ni el reglamento interno, ni por medio del centro de información de la Facultad de Ingeniería.

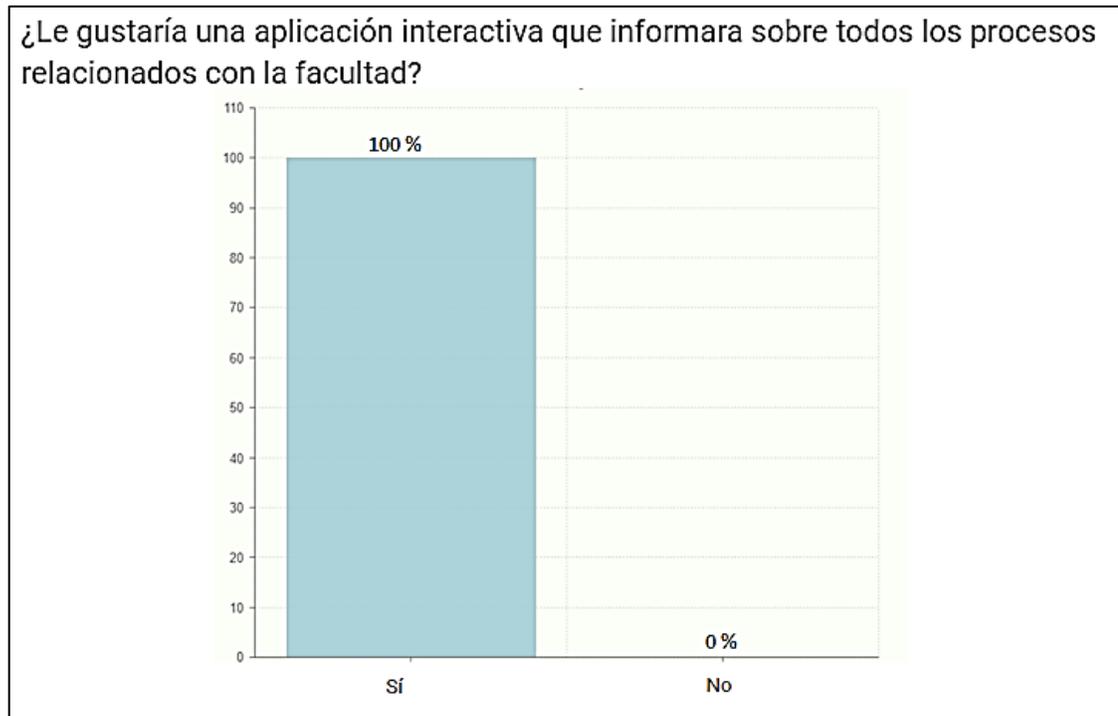
Figura 17. **Respuestas a la pregunta 2, de primer ingreso**



Fuente: elaboración propia, empleando Asisto.

El total de los estudiantes de primer ingreso ve como buena opción tener disponible una aplicación que provea la información de todos los procesos estudiantiles.

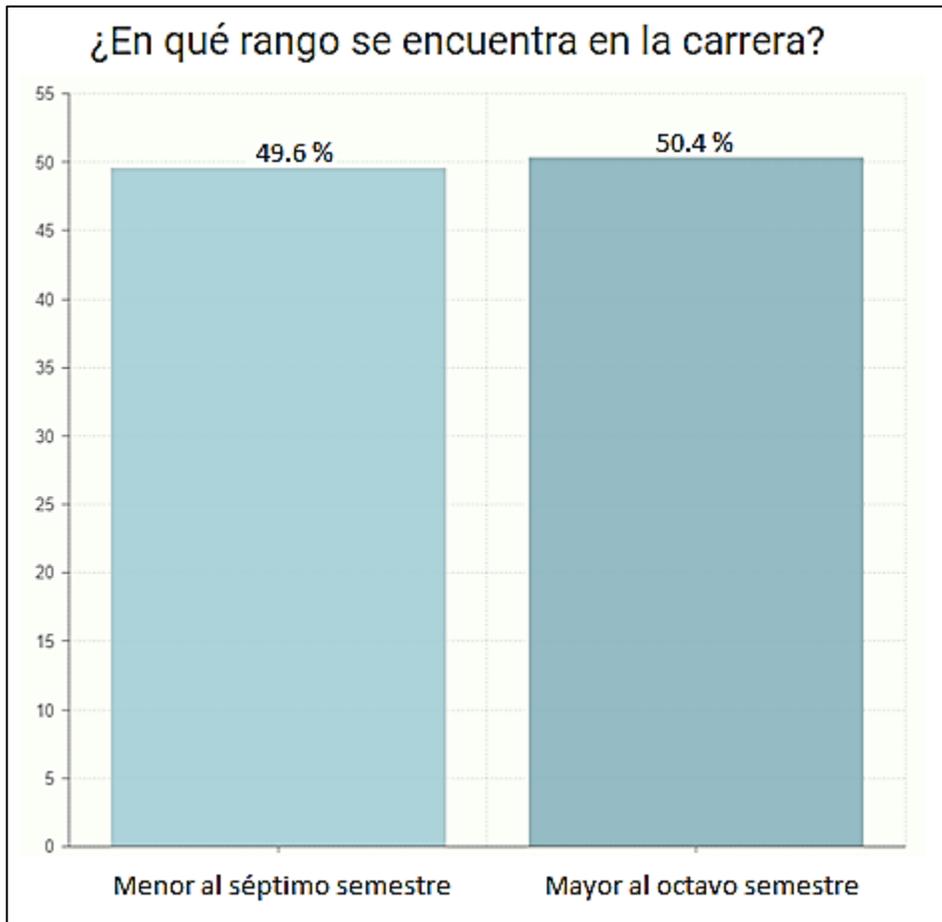
Figura 18. **Pregunta 3 (menor al séptimo semestre)**



Fuente: elaboración propia, empleando Asisto.

De los estudiantes de reingreso, 58 se encuentran en un rango mayor o igual al octavo semestre de la carrera; mientras que 56 estudiantes se encuentran en un rango menor o igual al séptimo semestre de su carrera.

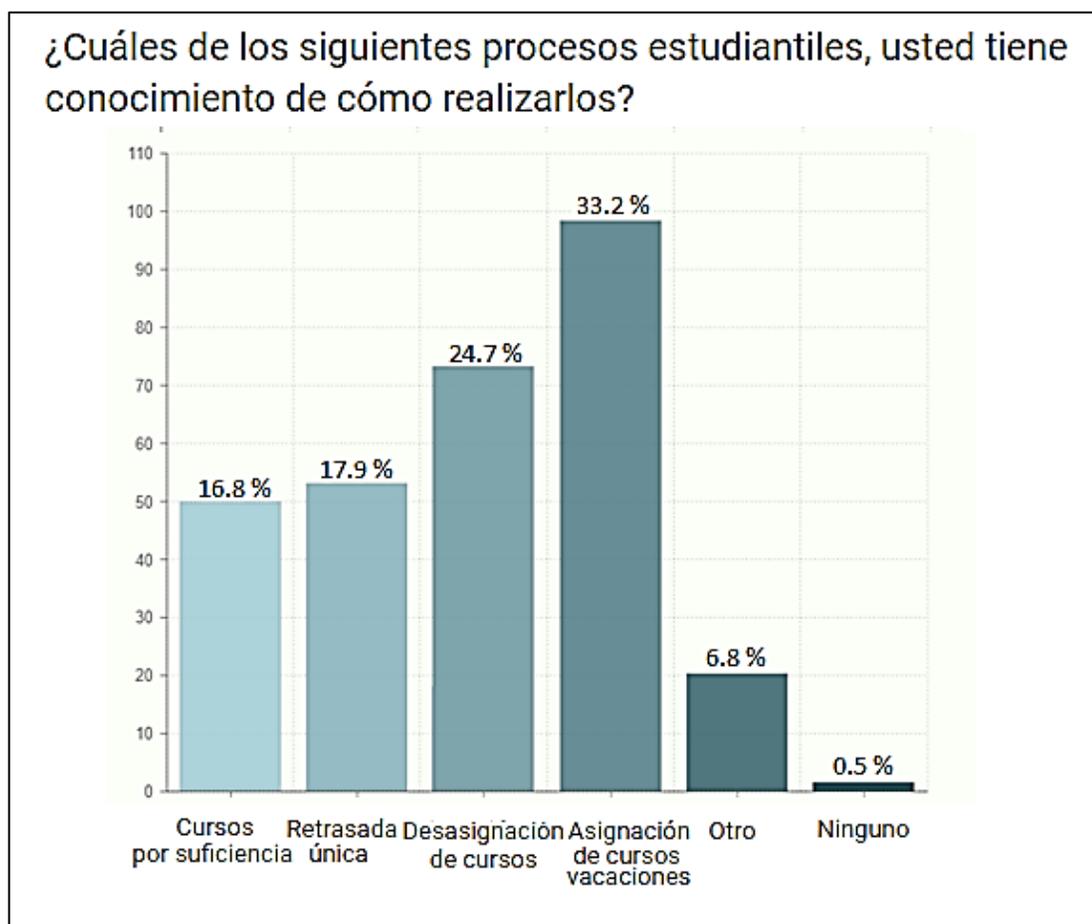
Figura 19. **Pregunta 1 (menor al séptimo semestre)**



Fuente: elaboración propia, empleando Asisto.

La asignación de cursos de vacaciones es el proceso en el cual la mayoría de estudiantes de reingreso se encuentran en un rango menor o igual al séptimo semestre; en cambio, en menor porcentaje, el proceso de suficiencia de cursos es del que tienen poco conocimiento.

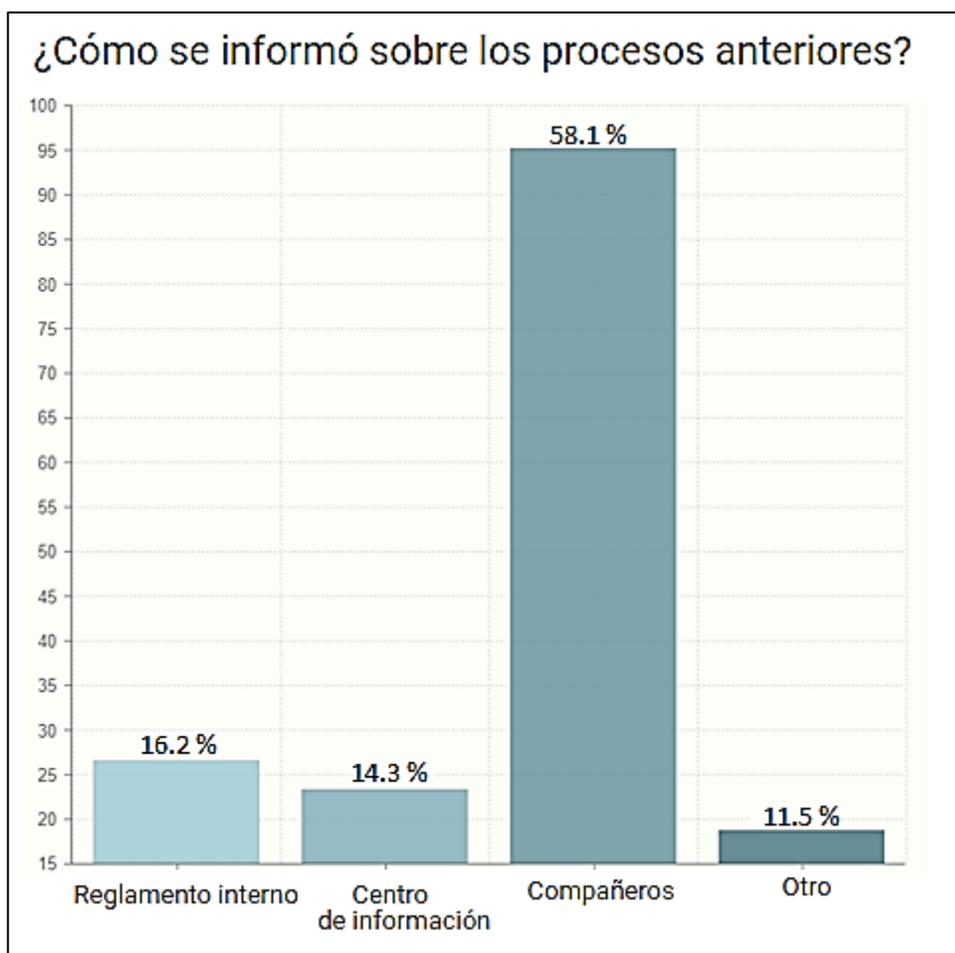
Figura 20. **Pregunta 2 (menor al séptimo semestre)**



Fuente: elaboración propia, empleando Asisto.

La información acerca de cómo realizar un proceso estudiantil, para la mayoría de los estudiantes de reingreso que se encuentran en un rango menor o igual al séptimo semestre es a través de sus compañeros; también, se informan por medio del reglamento interno y el centro de información de la Facultad de Ingeniería, en una menor cantidad los estudiantes se informan por otros medios.

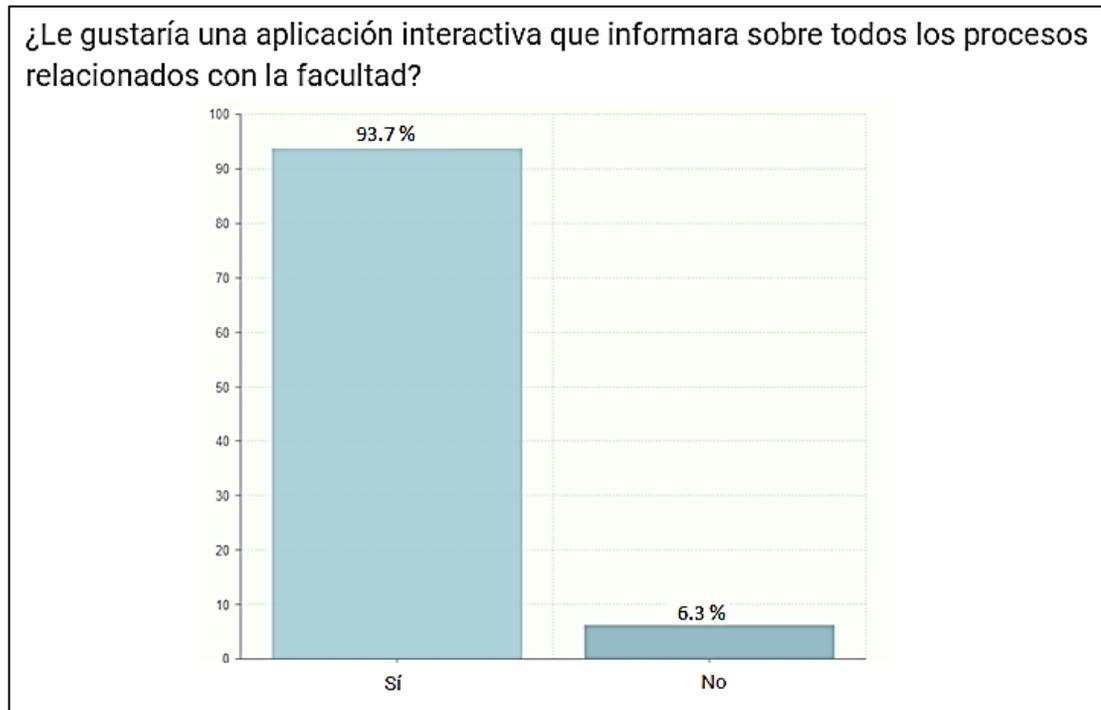
Figura 21. **Pregunta 2 (menor al séptimo semestre)**



Fuente: elaboración propia, empleando Asisto.

La mayoría de estudiantes de reingreso que se encuentran en un rango menor o igual al séptimo semestre indica que sería útil una aplicación en el cual se puedan visualizar los procesos estudiantiles; mientras que un bajo porcentaje indicó que no les sería útil la aplicación.

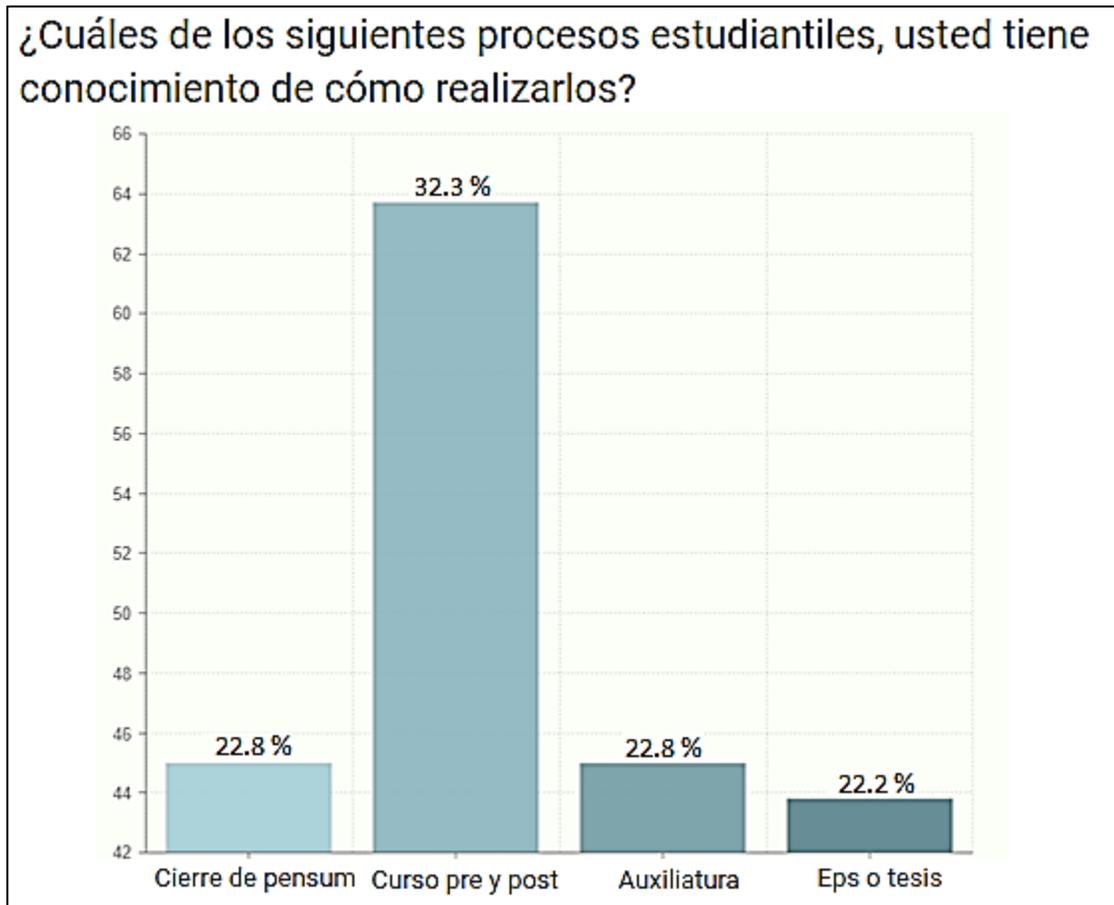
Figura 22. **Pregunta 3 (menor al séptimo semestre)**



Fuente: elaboración propia, empleando Asisto.

El proceso de pre y postrequisito es el que los estudiantes de reingreso que se encuentran en un rango mayor o igual al octavo semestre saben cómo realizarlo; mientras que dejaron reflejado que el proceso que la mayoría desconoce es el de EPS y tesis.

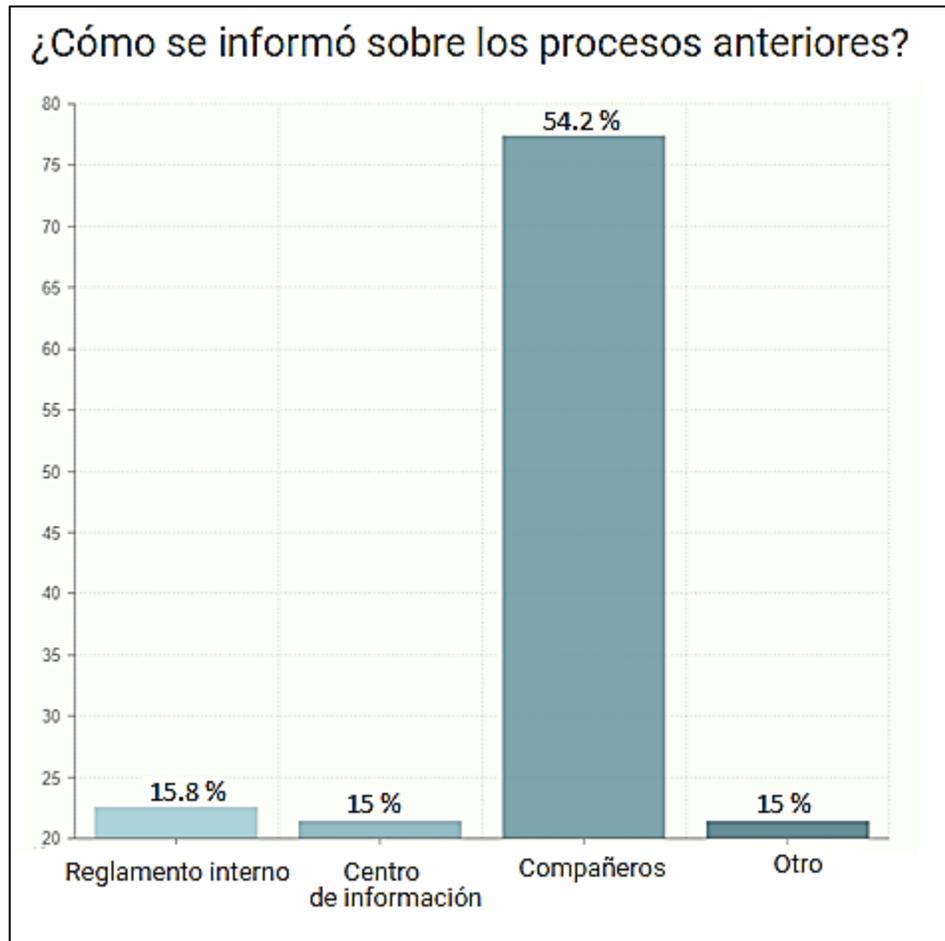
Figura 23. **Pregunta 1 (mayor al octavo semestre)**



Fuente: elaboración propia, empleando Asisto.

El medio más común que los estudiantes de reingreso que se encuentran en un rango mayor o igual al octavo obtienen información de los procesos estudiantiles, es por los compañeros. En un menor porcentaje se informan por medio de reglamento interno y el centro de información de la Facultad de Ingeniería.

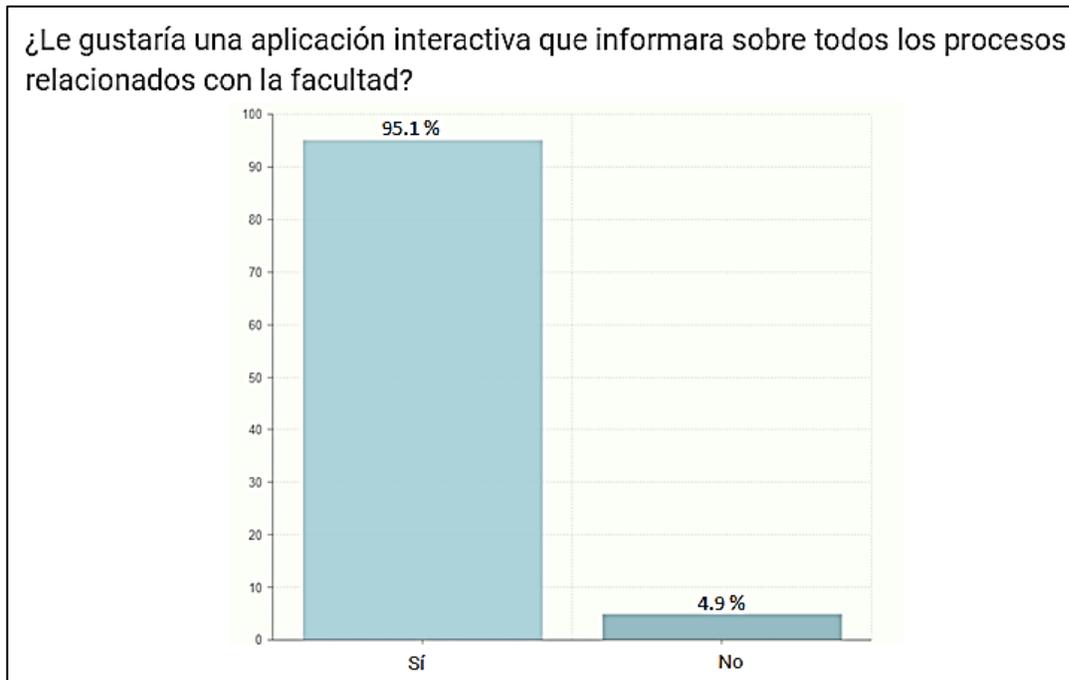
Figura 24. **Pregunta 2 (mayor al octavo semestre)**



Fuente: elaboración propia, empleando Asisto.

La gran mayoría de estudiantes de reingreso que se encuentran en un rango mayor o igual al octavo indican que una aplicación con los procesos estudiantiles sería de utilidad; mientras que un bajo porcentaje de estudiantes indica que no le sería útil.

Figura 25. **Pregunta 3 (mayor a octavo semestre)**



Fuente: elaboración propia, empleando Asisto.

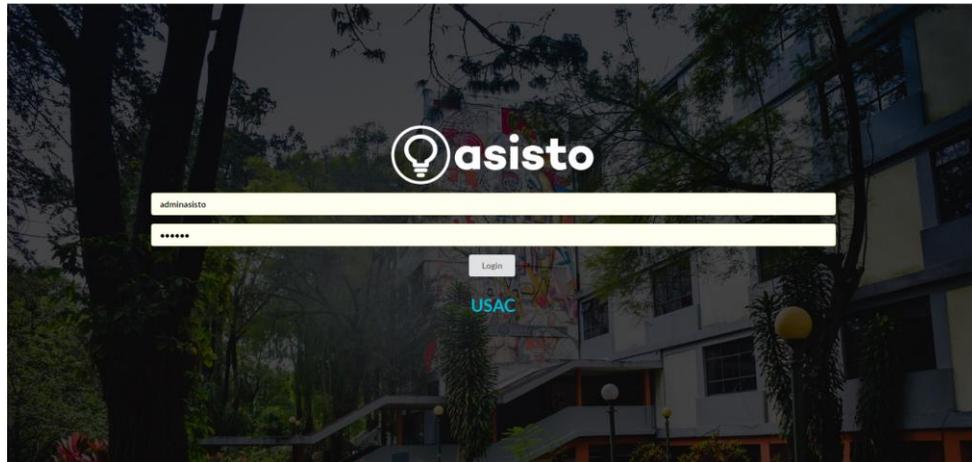
3.3. Diseño intuitivo y usabilidad

La aplicación web Asisto ha sido desarrollada con una interfaz limpia y agradable, esto con el objetivo de que sea fácil de entender para el estudiante y tenga la capacidad de interactuar con todas las funciones que la aplicación web ofrece, mejorando la experiencia del estudiante al utilizarla.

3.3.1. Administrador

Asisto cuenta con un sitio administrativo, en el cual se realizan las gestiones de usuarios, mantenimiento de guías, preguntas frecuentes y dependencias que están disponibles en el sitio cliente.

Figura 26. **Login sitio administrativo**



Fuente: elaboración propia, empleando Asisto.

Figura 27. **Login**



Fuente: elaboración propia, empleando Asisto.

Figura 28. Creación de usuario

The screenshot shows a mobile application interface for creating a new user. At the top left is a back arrow labeled 'Atrás'. At the top right is a blue envelope icon labeled 'Nuevo usuario'. The form contains three input fields: 'Nombre' (Name), 'Usuario' (Username), and 'Contraseña' (Password). Below the 'Contraseña' field is a grey button labeled 'Crear'.

Fuente: elaboración propia, empleando Asisto.

Figura 29. Creación de dependencias

The screenshot shows a mobile application interface for managing dependencies. At the top left is a back arrow labeled 'Atrás'. At the top right is a blue icon of three horizontal lines labeled 'Dependencias'. Below this is a grey button labeled 'Nueva dependencia'. Underneath is a search bar with the placeholder text 'Buscar dependencias'. Below the search bar is a white card with a blue header labeled 'General'. Inside the card, the word 'General' is displayed, and at the bottom are two grey buttons labeled 'Editar' and 'Eliminar'.

Fuente: elaboración propia, empleando Asisto.

Figura 30. Listado de guías



Fuente: elaboración propia, empleando Asisto.

Figura 31. Creación de guías



Fuente: elaboración propia, empleando Asisto.

Figura 32. Creación de preguntas frecuentes

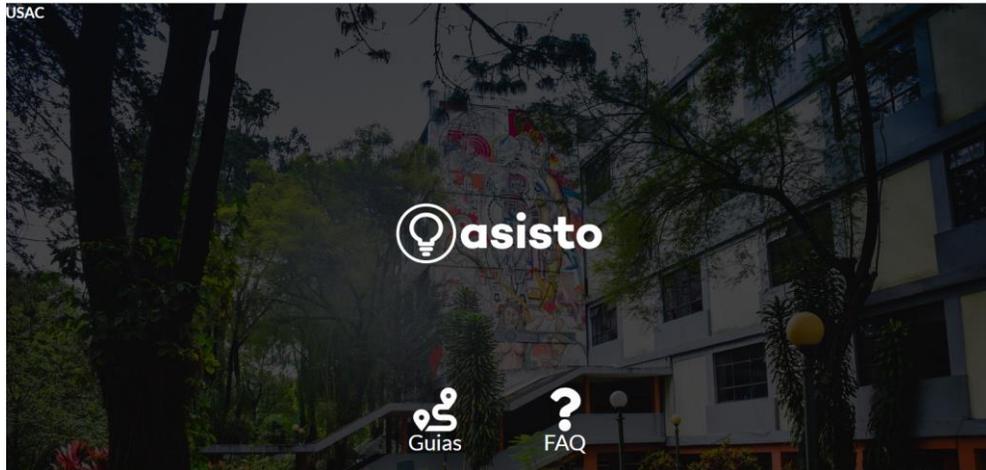


Fuente: elaboración propia, empleando Asisto.

3.3.2. Cliente

Este sitio contiene toda la información de los procesos administrativos, los cuales son mostrados como guías facilitando la visualización de estos, además se presentan una serie de preguntas que surgen de manera frecuente a los estudiantes.

Figura 33. Pagina inicial, sitio cliente



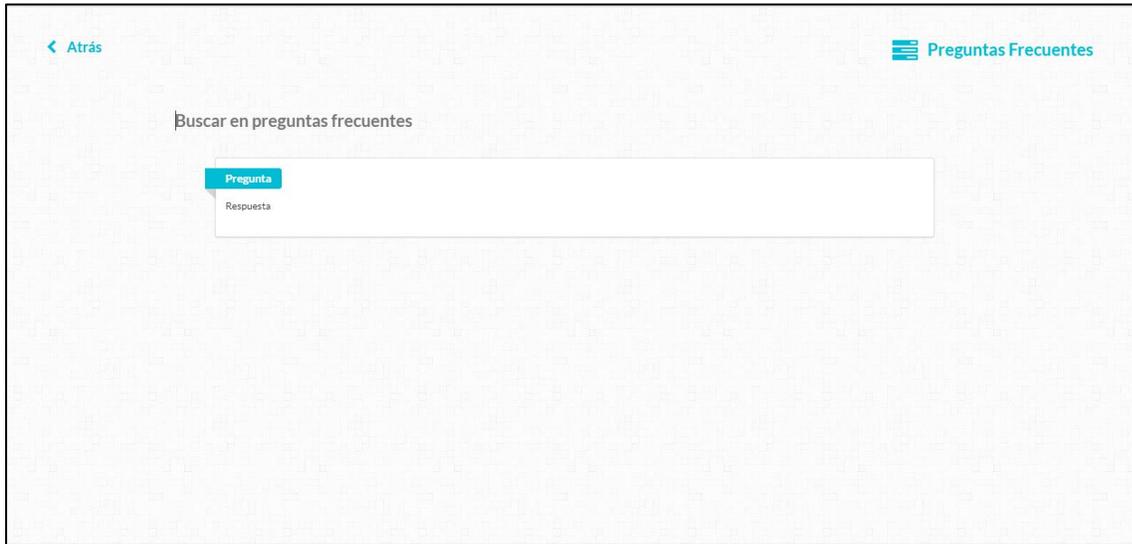
Fuente: elaboración propia, empleando Asisto.

Figura 34. Guías



Fuente: elaboración propia, empleando Asisto.

Figura 35. Preguntas frecuentes



Fuente: elaboración propia, empleando Asisto.

4. DOCUMENTACIÓN BASE PARA EL DESARROLLO DE LA APLICACIÓN

4.1. Tecnologías utilizadas

A continuación, se muestra una serie de tecnologías que fueron utilizadas para el desarrollo de la aplicación web Asisto, estas tecnologías además de proveer una mejor funcionalidad a la aplicación, facilitaron el desarrollo de esta.

4.1.1. NPM

NPM (*node package manager*, por sus siglas en inglés), es un gestor de paquetes JavaScript para Node.js, el cual provee un conjunto de distintas librerías o módulos que buscan brindar flexibilidad y rapidez en el desarrollo, ya que estos paquetes proveen utilidades que únicamente deben ser implementadas y no desarrolladas, así como facilita la administración de paquetes, módulos y dependencias de una aplicación Node.js.

4.1.2. Node.js

Es un entorno para JavaScript el cual se ejecuta de lado del servidor, básicamente es una máquina virtual, la cual se caracteriza por su alta rapidez y desempeño, esto debido a que Node.js utiliza un modelo para I/O (entrada/salida) no bloqueante; es decir, asíncrono y dirigido por eventos, lo cual hace a Node.js liviano y eficiente; provee la capacidad de mantener una gran cantidad de conexiones abiertas y en espera.

4.1.3. AngularJS

Es un marco de trabajo (framework) para JavaScript, utilizado para el desarrollo de aplicaciones web dinámicas; permite el desarrollo de aplicaciones SPA (*single page applications*); este framework promueve un patrón de diseño MVC (*model, view, controller*), lo cual permite delimitar la lógica de la aplicación en capas. AngularJS utiliza HTML como base, extendiendo la sintaxis de los atributos de este, por medio de directivas; además de vincular los datos a HTML a través de expresiones, las cuales facilitan de gran manera el desarrollo.

4.1.4. Helmet

Es un módulo para Node.js el cual provee seguridad a la aplicación, protegiéndola de vulnerabilidades que esta posee por media correcta configuración de encabezados HTTP. Está formado por nueve funciones las cuales evitan que se den ataques a través de ciertas debilidades que tienen algunas cabeceras HTTP; establece reglas para hacer uso de estas.

4.1.5. JWT

JSON Web Tokens, es un medio compacto y seguro de autenticación, el cual codifica como un objeto JSON los datos que son transferidos entre dos partes, el servidor genera un token; a este valor únicamente pueden acceder quienes tengan acceso a la clave secreta con la que los datos fueron encriptados.

4.1.6. JSONParser

Es un módulo para Node.js el cual analiza una cadena de texto como un objeto JSON, esto para entender la información que es enviada al solicitar un recurso al servidor.

4.1.7. Moment JS

Es una librería para JavaScript, la cual provee un fácil manejo de fechas, es decir, ayuda a validar formatos de fechas, convertir a otros formatos, analizar y manipular estas.

4.1.8. Path JS

Es un módulo desarrollado para Node.js que provee utilidades para el desarrollo que involucre acceso a directorios y rutas de archivos que sean útiles en el proyecto.

4.1.9. ExpressJS

Es un framework para el desarrollo de aplicaciones web; sus principales características son la robustez, rapidez, flexibilidad y simplicidad para el desarrollo. El objetivo del framework es establecer una organización en la estructura de la aplicación web que se desarrolla; establece una arquitectura MVC (*model, view, controller*) del lado del servidor; ayuda al manejo de rutas, peticiones y vistas en las aplicaciones, lo cual lo hace útil para el desarrollo de APIs basadas en REST.

4.1.10. Mongoose

Es una librería para Node.js, que tiene como función el modelado de datos en objetos (ODM, *object data modeling*), con el objetivo de establecer una estructura pero al mismo tiempo mantener flexibilidad en esta. Los datos son modelados por medio de esquemas que incluyen opciones de validación de tipos, creación de consultas y lógica que sea necesaria agregar.

4.1.11. API RESTful

API (*application programming interface*) es un conjunto de reglas que que sirve como intermediario entre los sistemas web para que estos se comuniquen entre sí.

REST (*representational state transfer*) es un estilo de arquitectura el cual provee un conjunto de reglas o estándares, los cuales facilitan el desarrollo de un API para la comunicación entre sistemas informáticos en la web. Los sistemas que aplican dichos estándares o reglas se les denominan RESTful; estos se caracterizan por ser sin estados, lo cual significa que ni el servidor ni el cliente necesita saber el estado en que se encuentran cada uno. Otra característica es que la implementación del cliente y el servidor se realizan de forma totalmente independiente; es decir, tanto el cliente como el servidor pueden sufrir cambios sin afectar al otro.

La manera en la que los sistemas se comunican aplicando los estándares REST es por medio del cliente, el cual realiza solicitudes de recursos y los servidores son los encargados de enviar la respuesta a la solicitud. Estas solicitudes consisten un verbo HTTP el cual le indica al servidor que acción debe tomar sobre el recurso, un encabezado en el que se puede enviar

información sobre la solicitud, una dirección hacia donde se encuentra el recurso y el cuerpo del mensaje en el cual se envían todos los datos.

4.1.12. MongoDB

Es una base de datos NoSQL de código abierto, la cual está construida bajo un modelo de datos orientado a documentos, así como un lenguaje no estructurado para realizar consultas; es por ello que no posee una estructura de filas y columnas que son habituales en las bases de datos relaciones; en cambio esta base de datos utiliza una arquitectura de colecciones y documentos; en estos son almacenados los datos en formato JSON mediante un esquema dinámico al cual han nombrado como BSON; dicho documento está estructurado basándose en una unidad básica de datos que consiste en un conjunto de pares clave-valor.

MongoDB se caracteriza por ser una base de datos fiable, de alto rendimiento y escalable, ya que su estructura para indexar los datos, que impacta positivamente en el rendimiento de lectura y mejora las búsquedas. Otra característica es la velocidad y volumen que puede manejar; tiene la capacidad de manejar grandes cantidades de datos realizando muchas operaciones por segundo.

4.2. Arquitectura de la aplicación

En esta sección se especifica la arquitectura de software bajo la cual se construyó la aplicación web Asisto, mostrando los aspectos técnicos de la arquitectura.

El modelo utilizado para el diseño de la aplicación web es la arquitectura MVC (modelo vista controlador) el cual se describe a continuación.

4.2.1. Modelo vista controlador (MVC)

MVC o modelo vista controlador (*model view controller*) es un patrón de diseño de arquitectura de software, el cual establece la estructura que una aplicación debe tener; esta estructura se basa en separar la lógica de la aplicación en tres partes, lo cual permite tener una mejor organización en el código y crear una aplicación mantenible, con modularidad y capacidad de reutilización. MVC se compone de la siguiente forma:

- Modelo
- Vista
- Controlador

4.2.2. Modelo

El modelo es donde se almacenan los objetos de datos que debe contener la aplicación, cuando este cambia notifica a sus observadores; es decir, a la vista o si se tiene una lógica distinta para controlar la vista lo hace al controlador.

En la aplicación Asisto, en el modelo se almacenan todos los esquemas que se han desarrollado con Mongoose, librería para Node.js. En los cuales se establece la estructura de cada documento.

4.2.3. Vista

Es lo que los usuarios pueden ver y el medio por el cual estos se comunican a la aplicación, todos los datos que son almacenados y manejados por el modelo son mostrados de forma entendible al usuario.

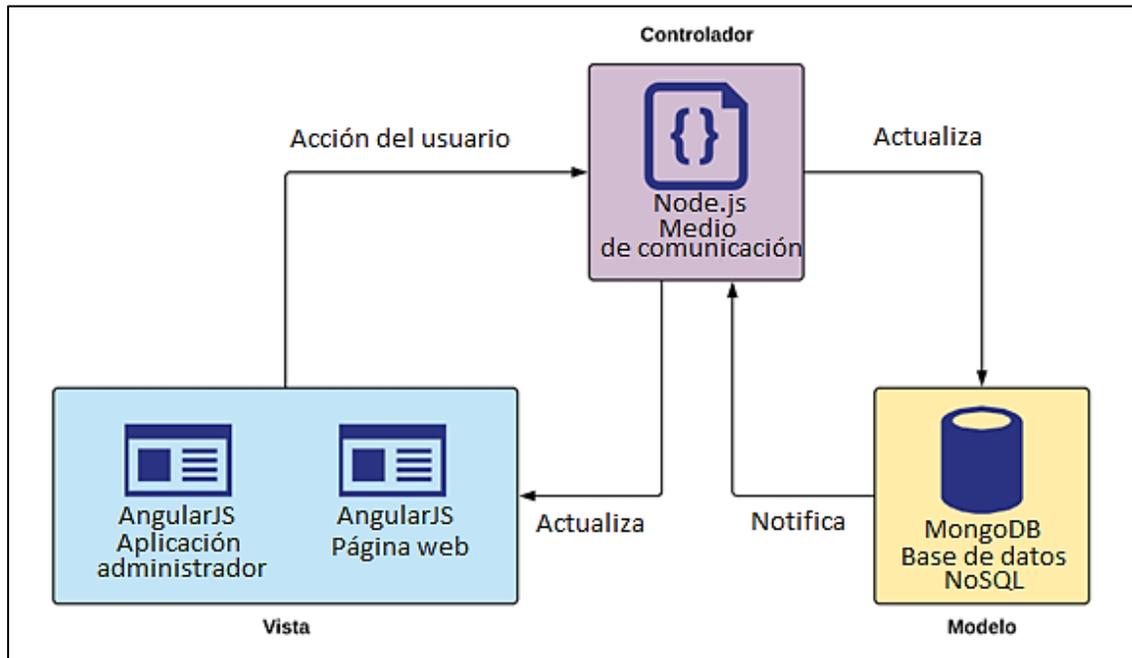
Tanto la aplicación de administración y la página principal de Asisto se desarrollaron con el framework AngularJS, ya que este provee facilidad en el desarrollo y mejora la experiencia del usuario.

4.2.4. Controlador

Tiene la lógica para tomar las decisiones entre el modelo y la vista, el controlador actualiza la vista según los cambios que el modelo vaya teniendo, así como el manejo de eventos que detectan la manipulación de la vista por medio del usuario para actualizar el modelo. Es decir, tanto la vista como el modelo son actualizados con base en las entradas que realicen los usuarios a la aplicación.

La lógica de las aplicaciones que conforman Asisto fue desarrollado en Node.js, los cuales son el medio de comunicación entre la vista y el modelo.

Figura 36. **Arquitectura, aplicación Asisto**



Fuente: elaboración propia, empleando Asisto.

4.3. Modelo de datos

Se realizó un análisis para el diseño de la base de datos no relacional, utilizada por su rapidez en el acceso a los datos; este diseño, además de ser dinámico, se acopla muy bien a las tecnologías que se utilizan tanto en el *backend*, como en el *frontend* de la aplicación, lo cual permite que el usuario tenga una mejor experiencia haciendo uso de Asisto.

4.3.1. Modelo de base de datos

Asisto utiliza como base de datos MongoDB, por lo cual el modelo de datos es no relacional; a continuación, se muestran los documentos JSON que forman parte del modelo, bajo el cual se rige la aplicación.

Figura 37. Documento guía

```
guia {  
  
  nombre:      String  
  desc:        String  
  visible:     Boolean  
  eliminado:   Boolean  
  usuarioAdmin: String  
  fechaCreacion: Date  
  pasos: [  
    {  
      orden:    Number  
      titulo:   String  
      descr:    String  
      anuncio:  String  
    }  
  ]  
  Dependencia: String  
}
```

Fuente: elaboración propia, empleando Asisto.

Figura 38. **Documento administrador**

```
usuario {  
    nombre:      String  
    tipo:        Number  
    pass:        String  
    user:        String  
    flagActivo: Boolean  
    fechaCreacion: Date  
}
```

Fuente: elaboración propia, empleando Asisto.

Figura 39. **Documento FAQ**

```
faq {  
    pregunta:      String  
    leyenda:       String  
    respuesta:     String  
    eliminado:     Boolean  
}
```

Fuente: elaboración propia, empleando Asisto.

Figura 40. **Documento dependencia**

```
dependencia {  
    nombre:          String  
    ubicacion:      String  
    eliminado:      Boolean  
    usuarioAdmin:   String  
    fechaCreado:    Date  
}
```

Fuente: elaboración propia, empleando Asisto.

4.3.2. **Descripción de documentos**

La base de datos utilizada en la aplicación web Asisto, estructuran toda la información en una documentos, en los cuales la información en una representación binaria. A continuación, se muestran los documentos utilizados para el almacenamiento de la información de la aplicación web.

4.3.2.1. **Guía**

En este documento se almacenan la información de las guías, así como los pasos que la conforman; el documento guía está formado por objetos compuesto por pares clave-valor y un arreglo de objetos que son los pasos de las guías; a continuación, se muestra un detalle:

- Nombre: indica el nombre que se le da a la guía

- Desc: contiene información sobre la guía
- Visible: indica si la guía será o no mostrada en la vista
- Eliminado: indica si la guía fue eliminada
- UsuarioAdmin: indica el usuario que crea o indica la guía
- FechaCreacion: contiene la fecha en la cual fue creada la guía

El arreglo paso es una colección de objetos que contienen la siguiente información:

- Orden: orden en qué debe ir un paso
- Título: nombre del paso
- Descr: información acerca del paso
- Anuncio: información adicional que puede ser útil al realizar el paso

4.3.2.2. Usuario

En este documento se modela todos datos sobre los usuarios con acceso al sistema, así como los permisos de navegación sobre la aplicación. A continuación, se muestran un detalle de la estructura del documento:

- Nombre: contiene el nombre del usuario
- Tipo: indica qué rol tiene el usuario
- Pass: contiene la contraseña del usuario
- User: contiene el nombre de usuario
- FlagActivado: indica si el usuario está activo o no
- FechaCreado: contiene la fecha en la cual se creó el usuario

4.3.2.3. Faq

El documento Faq contiene las preguntas y respuestas que son consideradas frecuentes y que se requieren mostrar en la página de consultas de Asisto. A continuación, se muestra un detalle de la estructura del documento:

- Pregunta: contiene la pregunta frecuente
- Leyenda:
- Respuesta: contiene la respuesta a la pregunta frecuente
- Eliminado: indica si la pregunta ha sido eliminada

4.3.2.4. Dependencia

Este documento contiene un catálogo de las escuelas que forman parte de la Facultad de Ingeniería, con el objetivo de asignar una dependencia a cada una de las guías. A continuación, se muestra un detalle de la estructura del documento:

- Nombre: contiene el nombre de la dependencia
- Ubicación: contiene la ubicación de la dependencia
- Eliminado: indica si la dependencia fue eliminada
- UsuarioAdmin: indica el usuario que crea o modifica la guía
- FechaCreado: indica la fecha en que la dependencia fue creada

4.4. Herramientas

Asisto está conformado por dos aplicaciones web, una administrativa y una cliente; para el desarrollo de ambas es necesario cumplir una serie de

requisitos, tales como la arquitectura, el modelo de datos y las herramientas que se emplearon; los cuales se describen a continuación:

4.4.1. Postman

Es un entorno de desarrollo de APIs (ADE, API Development Environment), es decir, una aplicación para la prueba de APIs, por medio del envío de solicitudes o peticiones a un servidor web y mostrando las respuesta que este devuelve al realizar a través de esta herramienta el consumo de servicios. Además, Postman tiene la capacidad de monitorear a detalle todas las solicitudes, así como depurar, automatizar pruebas y documentar un API.

4.4.2. Visual Studio Code

Es un editor de código desarrollado por Microsoft, el cual es compatible con gran cantidad de lenguajes. Se caracteriza por poseer potentes herramientas para desarrollador: autocompletado y depuración de código.

Además de ampliar las funciones de este por medio de *plugins* que aportan más funcionalidades y características al editor. Una característica que destaca es la compatibilidad y agilidad que brinda al desarrollo con el lenguaje JavaScript y Node.js e integración con más herramientas como GIT.

4.4.3. Robomongo

Herramienta que permite la administración de las bases de datos de MongoDB; este provee una interfaz (GUI) la cual facilita la creación de bases de datos y colecciones, así como realizar consultas. Ayuda a mantener una

organización de todas las base de datos creadas, de la misma forma con los datos y sus funciones.

4.4.4. Nodemon

Es una utilidad que provee la funcionalidad de monitoreo de la aplicación en desarrollo; esto significa que para cualquier cambio que se realice en el código fuente de la aplicación; esta utilidad lo detectará de manera automática lo cual reiniciará el servidor para reflejar los cambios que se realizaron; esta herramienta es bastante útil para el desarrollo de aplicaciones Node.js.

4.4.5. PM2

Es una herramienta para la gestión de procesos para Node.js; permite establecer un monitoreo sobre todas las aplicaciones Node.js, manteniendo al tanto del tráfico que existe y logs en tiempo de ejecución, monitoreando la memoria y los procesos existentes en cada aplicación. PM2 tiene la capacidad de mantener siempre activas las aplicaciones evitando que estas entren en un estado de inactividad.

4.5. Hardware

En esta sección se muestra el hardware utilizado durante la fase de desarrollo y el hardware en el cual se implementó la aplicación web Asisto.

4.5.1. Desarrollo

El hardware necesario para el desarrollo de Asisto debe cumplir con ciertos requerimientos, los cuales se especifican a continuación:

4.5.2. Hardware mínimo

Los requerimientos de hardware con los que debe contar el ambiente de desarrollo deben ser al menos los siguientes:

- Procesador 1,6 GHz o superior
- 1 GB memoria RAM
- Sistema operativo Windows 7, OS X Yosemite, Linux (Debian, Ubuntu)

4.5.3. Hardware utilizado

El hardware utilizado para el desarrollo de Asisto es el siguiente:

- Procesador Core i7-7700 2,81 GHz
- 12 GB memoria RAM
- Sistema operativo Windows 10

4.5.4. Implementación

La solución se implementó en una máquina virtual de DigitalOcean, específicamente en un Droplet, el cual cuenta con las siguientes especificaciones:

- 1 vCPU (virtual CPU)
- 1 GB de memoria RAM
- 25 GB disco de estado sólido (SSD)
- 1 TB de transferencia de datos

4.6. Tutorial de desarrollo

A continuación, se muestra una guía para la configuración del ambiente de desarrollo, en la que se especifica la instalación y configuración de cada una de las herramientas utilizadas para el desarrollo de la aplicación web Asisto.

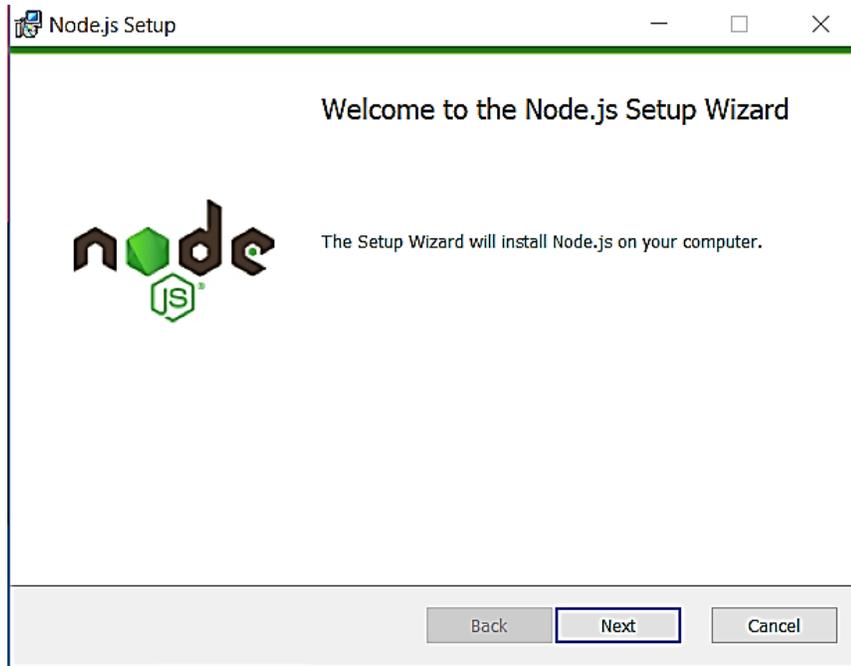
4.6.1. Instalación y configuración de herramientas

Las herramientas utilizadas para el desarrollo de Asisto requieren de cierta configuración para un correcto funcionamiento, desempeño y para obtener todos los beneficios que estas brindan para el desarrollo de la aplicación; esta configuración del ambiente se realizó en el sistema operativo Windows 10.

4.6.1.1. Node.js

Node.js se puede descargar desde su sitio web oficial (<https://nodejs.org/es/download/>), en el cual está disponible para Windows, Mac OS y Linux. Para este caso se recomienda descargar la versión 8.11.4 de Node.js, Una vez descargado, se ejecuta el instalador, que por medio del asistente de configuración (figura 41) proveerá la ayuda necesaria que la instalación requiere.

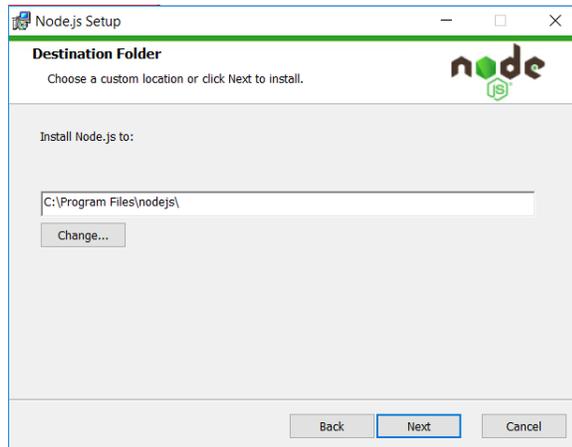
Figura 41. **Asistente de configuración de Node.js**



Fuente: elaboración propia, empleando Asisto.

Se debe establecer un directorio en el cual Node.js será instalado, como se muestra en la figura 42.

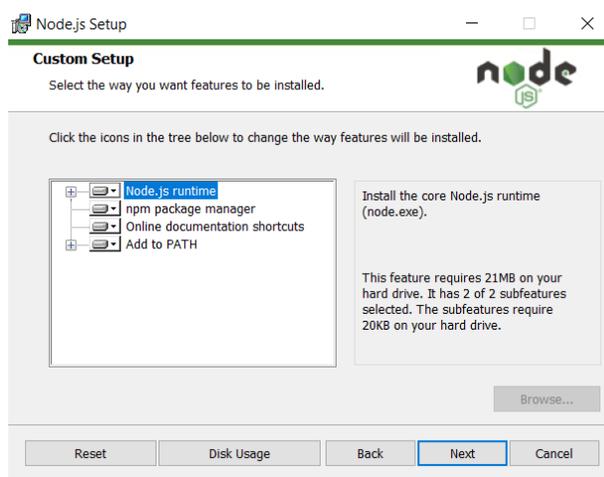
Figura 42. Configuración de directorio



Fuente: elaboración propia, empleando Asisto.

El asistente provee un listado de características que se pueden instalar; para el desarrollo de Asisto, fue necesario instalar todas.

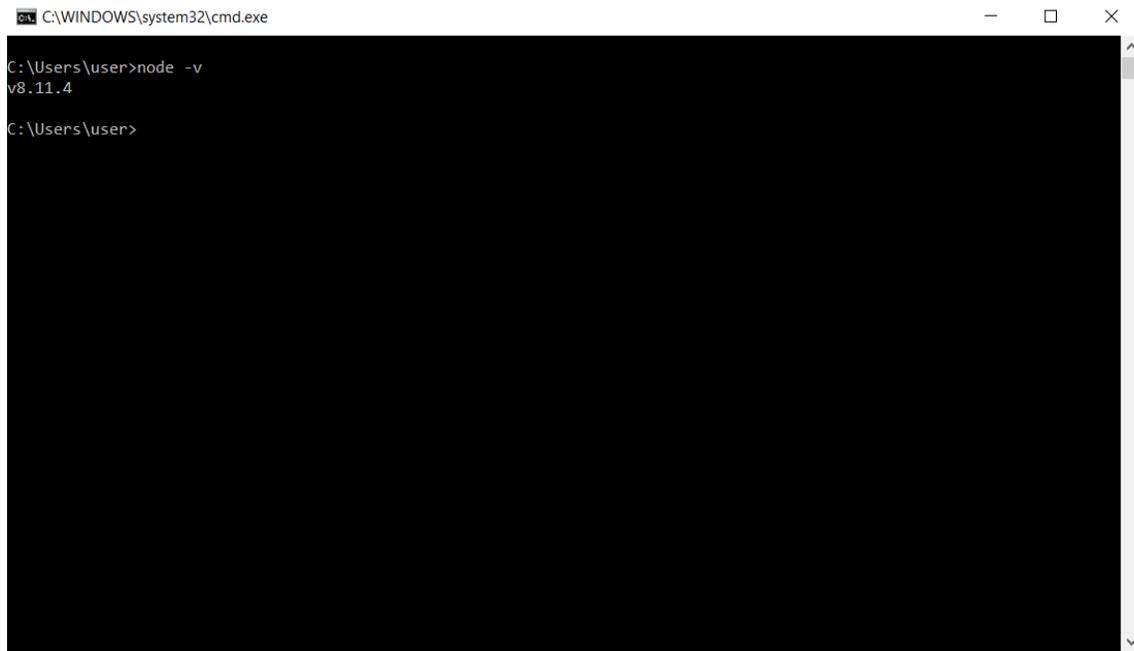
Figura 43. Configuración de instalación de Node.js



Fuente: elaboración propia, empleando Asisto.

Al finalizar la instalación, por medio de la consola del sistema, se verifica por medio de la versión, que Node.js fue instalado.

Figura 44. **Verificación de la instalación de Node.js**



```
C:\WINDOWS\system32\cmd.exe
C:\Users\user>node -v
v8.11.4
C:\Users\user>
```

Fuente: elaboración propia, empleando Asisto.

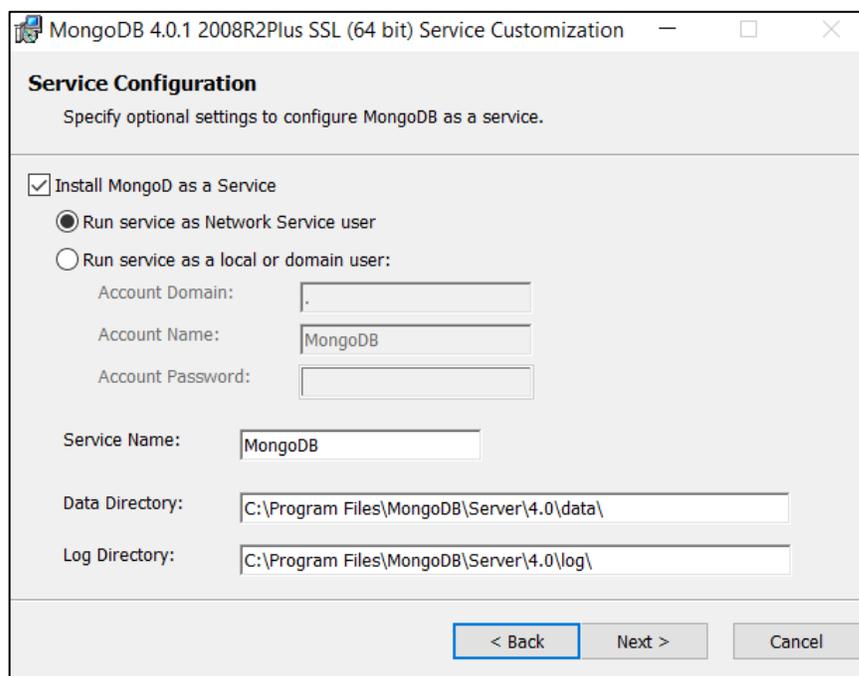
4.6.1.2. MongoDB

Para la instalación y configuración de esta base de datos, es necesario obtener el instalador desde su sitio oficial (<https://www.mongodb.com/download-center#community>), una vez descargado, se procede a instalar por medio de su asistente de configuración.

Durante la instalación se debe realizar la configuración del servicio de MongoDB, en el cual se puede especificar un nombre de dominio y las

credenciales si la instalación será en un dominio. También, se debe especificar el nombre del servicio y los directorios donde se deberán almacenar tanto la data como la bitácora de las bases de datos que se creen, como se muestra en la figura 45.

Figura 45. **Configuración del servicio de MongoDB**



Fuente: elaboración propia, empleando Asisto.

4.6.1.3. **PM2**

Para esta herramienta que tiene la función de monitoreo, fue necesario realizar la instalación por medio de la consola del sistema, a través del gestor de dependencias NPM; utilizando el comando 'npm install pm2 -g' se realizó la instalación de forma global en el sistema, como se muestra en la figura 46.

Figura 46. Instalación de herramienta PM2

```
C:\WINDOWS\system32\cmd.exe
C:\ambiente\proyectos\asisto>npm install pm2 -g
npm WARN registry Unexpected warning for https://registry.npmjs.org/: Miscellaneous Warning EINTEGRITY: sha1-QFUCsAfzGcP0cXXER0UnMA8qta0= integrity checksum failed when using sha1: wanted sha1-QFUCsAfzGcP0cXXER0UnMA8qta0= but got sha512-zr6QQnzL3Ja0t0XI8gws2kn7zV2p01/D3kreNvS6hFZhVU5g+uY/30142jbgT0XGcNBEmBDGJR71J692V92tA==. (260 bytes)
npm WARN registry Using stale package data from https://registry.npmjs.org/ due to a request error during revalidation.
C:\Users\user\AppData\Roaming\npm\pm2-docker -> C:\Users\user\AppData\Roaming\npm\node_modules\pm2\bin\pm2-docker
C:\Users\user\AppData\Roaming\npm\pm2-runtime -> C:\Users\user\AppData\Roaming\npm\node_modules\pm2\bin\pm2-runtime
C:\Users\user\AppData\Roaming\npm\pm2-dev -> C:\Users\user\AppData\Roaming\npm\node_modules\pm2\bin\pm2-dev
C:\Users\user\AppData\Roaming\npm\pm2 -> C:\Users\user\AppData\Roaming\npm\node_modules\pm2\bin\pm2
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.4 (node_modules\pm2\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.4: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ pm2@3.0.3
added 242 packages in 49.875s

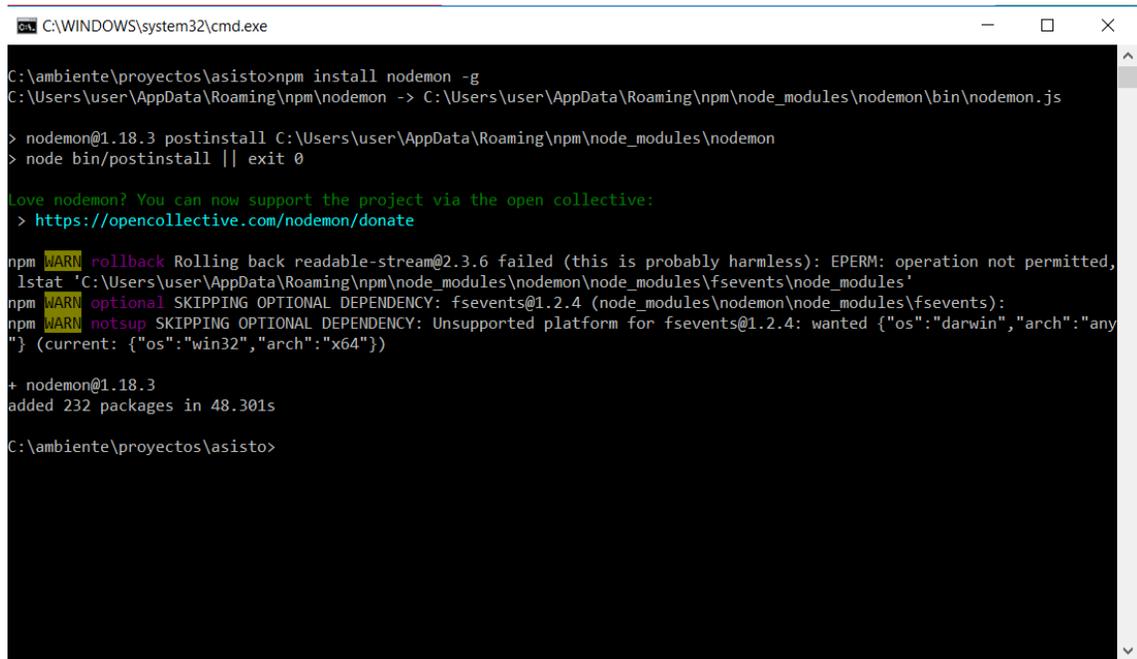
C:\ambiente\proyectos\asisto>
```

Fuente: elaboración propia, empleando Asisto.

4.6.1.4. Nodemon

Por medio del gestor de dependencias NPM, se realizó la instalación de la herramienta para la detección de cambios en el desarrollo y reinicio automático del servidor; con el comando 'npm install nodemon -g' se realizó la instalación de la herramienta de forma global en el sistema, como se muestra en la figura 47.

Figura 47. Instalación de herramienta Nodemon



```
C:\WINDOWS\system32\cmd.exe
C:\ambiente\proyectos\asisto>npm install nodemon -g
C:\Users\user\AppData\Roaming\npm\nodemon -> C:\Users\user\AppData\Roaming\npm\node_modules\nodemon\bin\nodemon.js

> nodemon@1.18.3 postinstall C:\Users\user\AppData\Roaming\npm\node_modules\nodemon
> node bin/postinstall || exit 0

Love nodemon? You can now support the project via the open collective:
> https://opencollective.com/nodemon/donate

npm WARN rollback Rolling back readable-stream@2.3.6 failed (this is probably harmless): EPERM: operation not permitted,
  lstat 'C:\Users\user\AppData\Roaming\npm\node_modules\nodemon\node_modules\fsevents\node_modules'
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.4 (node_modules\nodemon\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.4: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

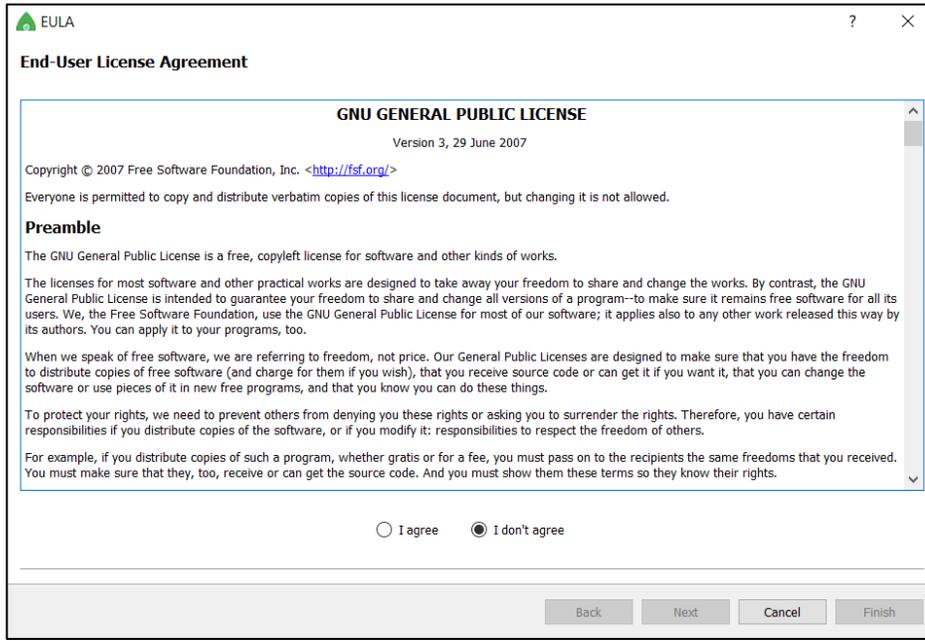
+ nodemon@1.18.3
added 232 packages in 48.301s
C:\ambiente\proyectos\asisto>
```

Fuente: elaboración propia, empleando Asisto.

4.6.1.5. Robomongo

Esta herramienta puede ser descargada desde su sitio oficial (<https://robomongo.org/download>), al finalizar la descarga, se ejecuta el instalador, que con la ayuda del asistente de configuración se realizará la instalación, como se muestra en la figura 48.

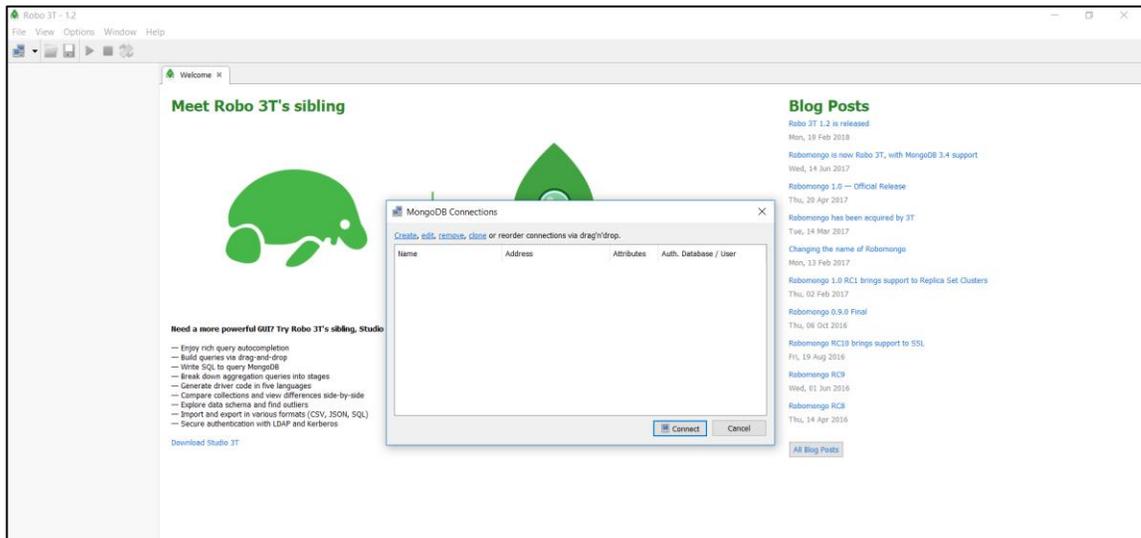
Figura 48. Asistente de configuración de Robomongo



Fuente: elaboración propia, empleando Asisto.

Al finalizar la instalación se debe ejecutar la herramienta, la cual mostrar una venta como se muestra en la figura 49.

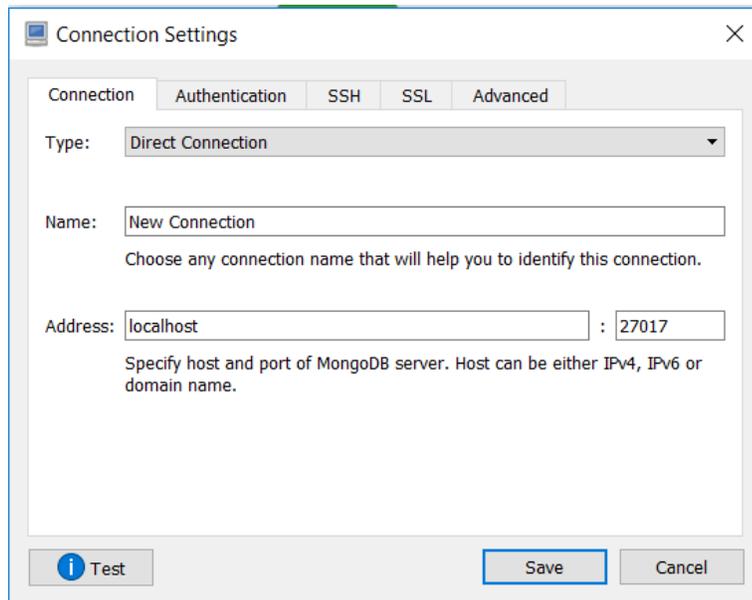
Figura 49. Ventana de configuración de Robomongo



Fuente: elaboración propia, empleando Asisto.

Para que Robomongo se conecte a MongoDB, se debe realizar la configuración de la conexión, en la ventana que se muestra en la figura 50. Se debe especificar un nombre que será asignado a la conexión, la dirección del *host* y el número de puerto, al cual se realizará la conexión.

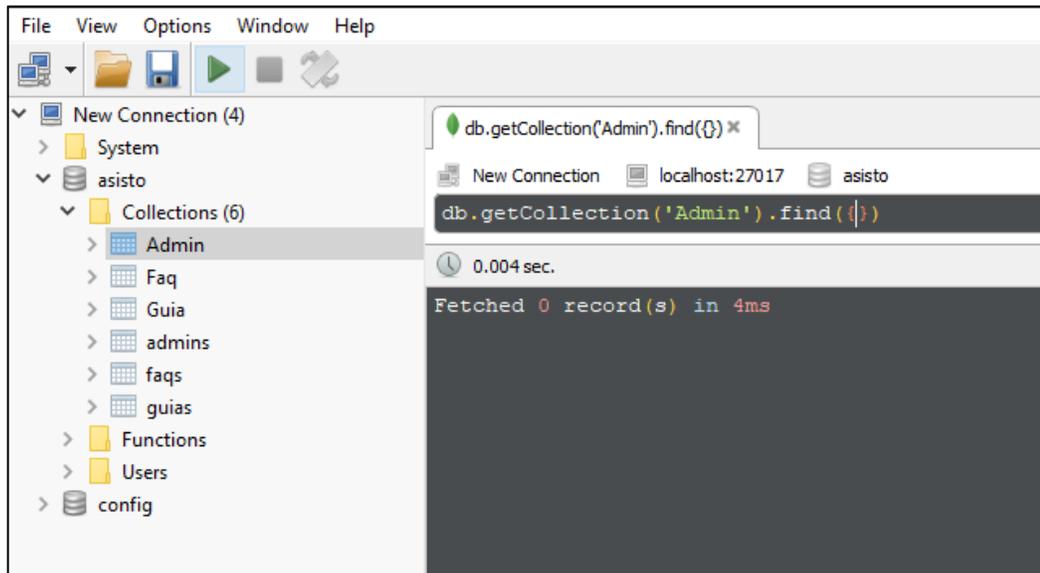
Figura 50. **Configuración de conexión Robomongo**



Fuente: elaboración propia, empleando Asisto.

Luego de la configuración de la conexión, esta se establece con la base de datos de MongoDB, y muestra las bases de datos existentes así como las colecciones de documentos de cada una de estas, en las cuales se pueden realizar las consultas.

Figura 51. **Conexión establecida a través de Robomongo**

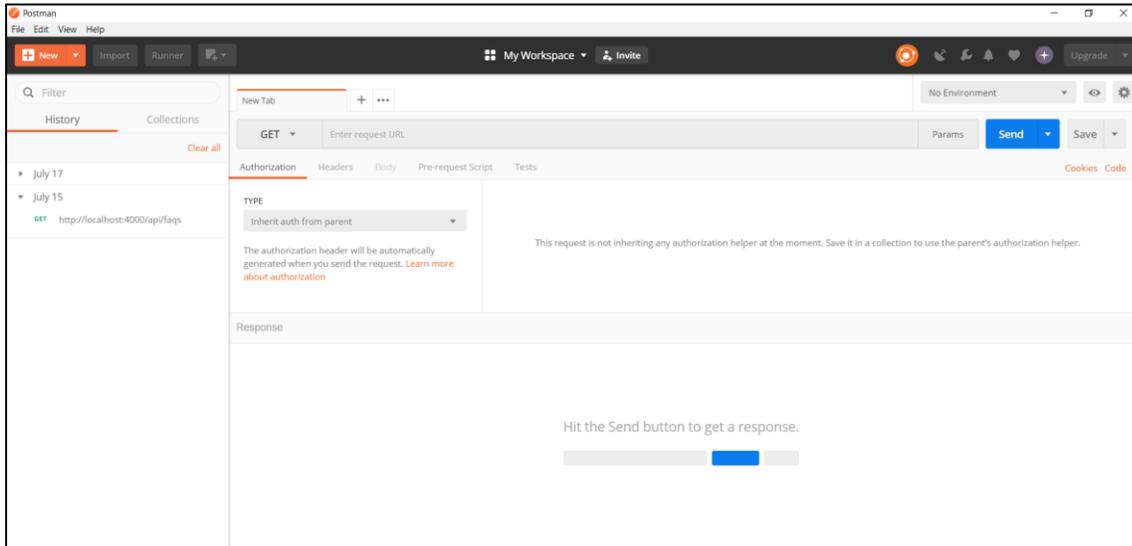


Fuente: elaboración propia, empleando Asisto.

4.6.1.6. **Postman**

El instalador de esta herramienta se puede obtener en su sitio oficial (<https://www.getpostman.com/apps>); al finalizar la descarga, se debe ejecutar el instalador, que provee una guía a través del asistente de configuración; esta herramienta no necesita configuraciones específicas para su funcionamiento; está iniciará y mostrará una interfaz, como que se visualiza en la figura 52.

Figura 52. **Herramienta Postman en funcionamiento**

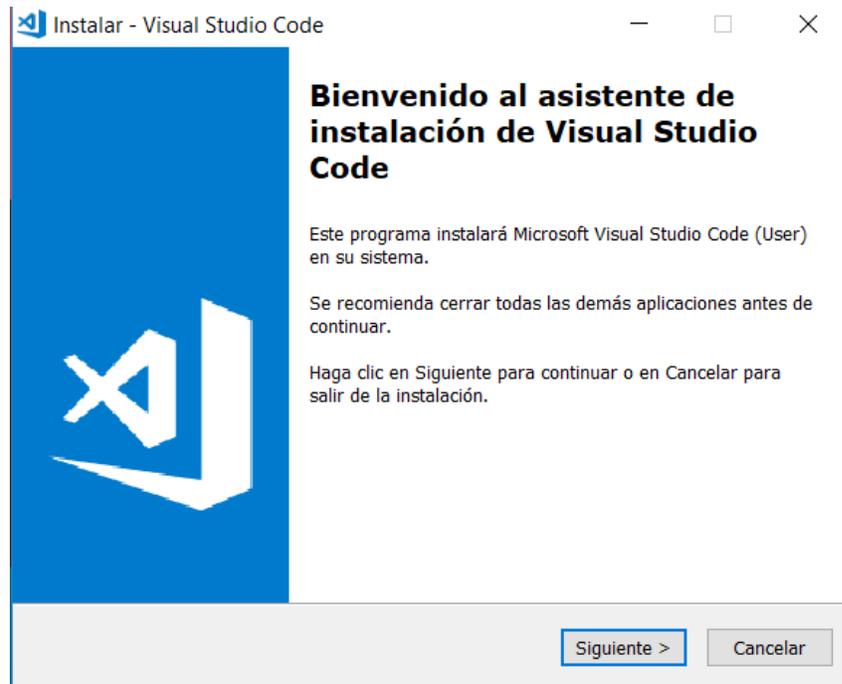


Fuente: elaboración propia, empleando Asisto.

4.6.1.7. **Visual Studio Code**

El editor de código, se descarga de su sitio oficial (<https://code.visualstudio.com/download>); una vez descargado, se procede a ejecutar el instalador que por medio de su asistente de configuración indica cada paso que se debe realizar, como se muestra en la figura 53, para esta herramienta se deben instalar todas las características y configuraciones que tiene por defecto.

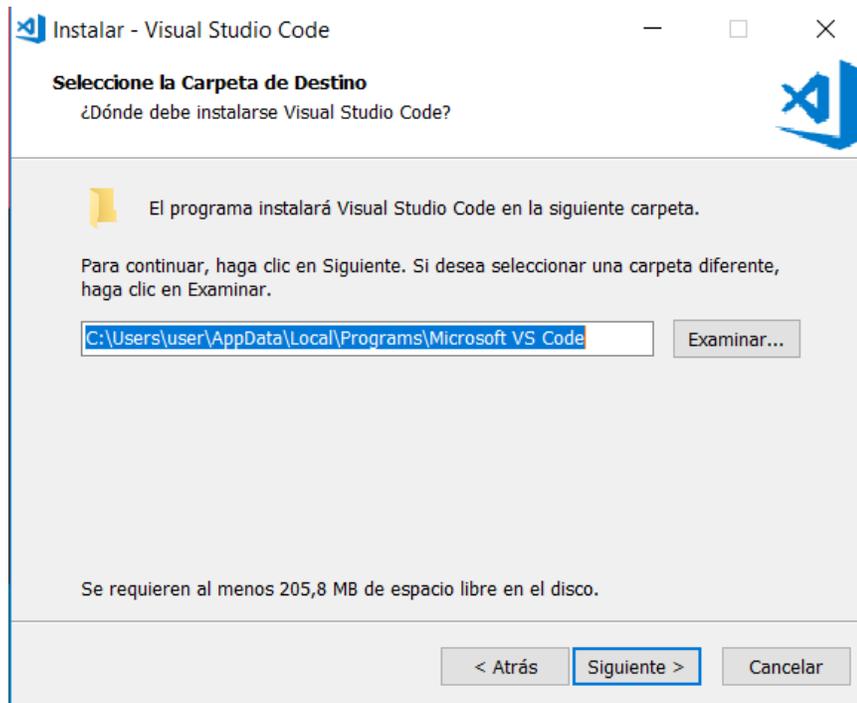
Figura 53. **Asistente de instalación de Visual Studio Code**



Fuente: elaboración propia, empleando Asisto.

La instalación de la herramienta se debe realizar en el directorio que el asistente establece como determinado, figura 54.

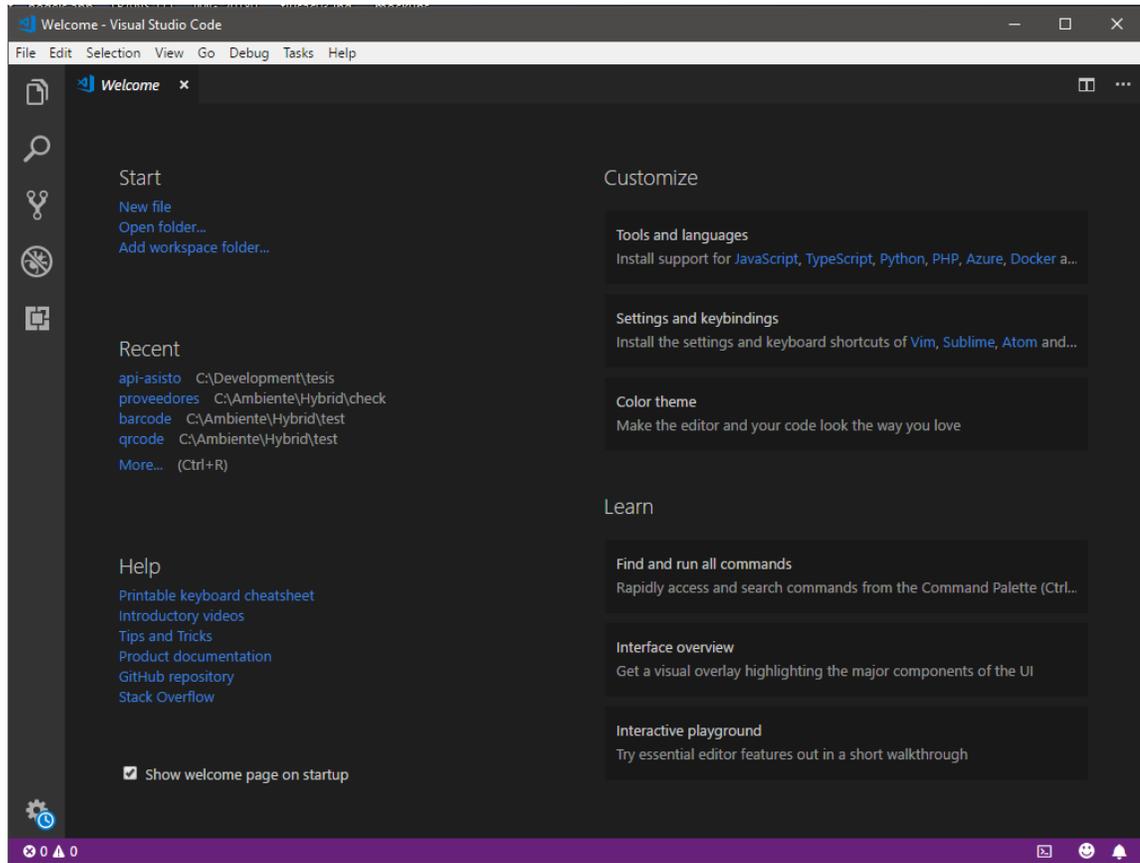
Figura 54. **Directorio de instalación de Visual Studio Code**



Fuente: elaboración propia, empleando Asisto.

Al finalizar la instalación se ejecutará el editor de texto, en la cual se desarrollará el proyecto, como se muestra en la figura 55.

Figura 55. Editor de texto Visual Studio Code



Fuente: elaboración propia, empleando Asisto.

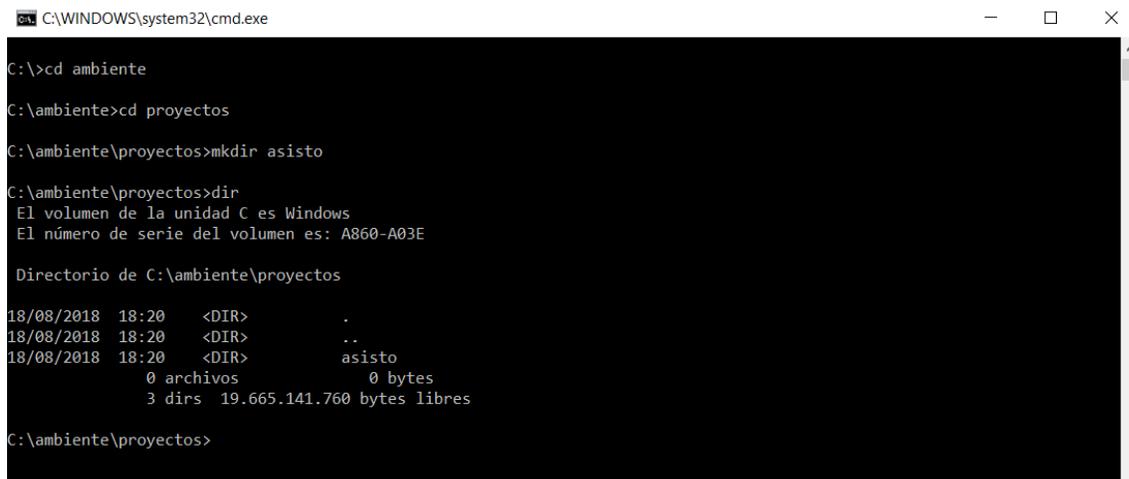
4.6.2. Desarrollo

Para llevar a cabo la correcta ejecución y configuración de la aplicación web es necesario realizar una serie de pasos que enseguida se muestran.

4.6.2.1. Configuración del proyecto

Teniendo todo el ambiente de desarrollo configurado y las herramientas necesarias instaladas, se procede a la creación de un proyecto Node.js; para esto se debe crear un directorio donde estará contenido el proyecto, por facilidad se recomienda crear una carpeta con todos los proyectos Node.js en el directorio raíz del sistema, como se muestra en la figura 56.

Figura 56. Configuración inicial del proyecto



```
C:\WINDOWS\system32\cmd.exe
C:\>cd ambiente
C:\ambiente>cd proyectos
C:\ambiente\proyectos>mkdir asisto
C:\ambiente\proyectos>dir
El volumen de la unidad C es Windows
El número de serie del volumen es: A860-A03E

Directorio de C:\ambiente\proyectos
18/08/2018  18:20    <DIR>        .
18/08/2018  18:20    <DIR>        ..
18/08/2018  18:20    <DIR>        asisto
                0 archivos    0 bytes
                3 dirs  19.665.141.760 bytes libres

C:\ambiente\proyectos>
```

Fuente: elaboración propia, empleando Asisto.

Creado el directorio del proyecto, se debe navegar hacia él, y ejecutar el comando 'npm -v', en la consola del sistema, para verificar que el gestor de dependencias se haya instalado correctamente en conjunto con Node.js.

Figura 57. Verificación de instalación de npm

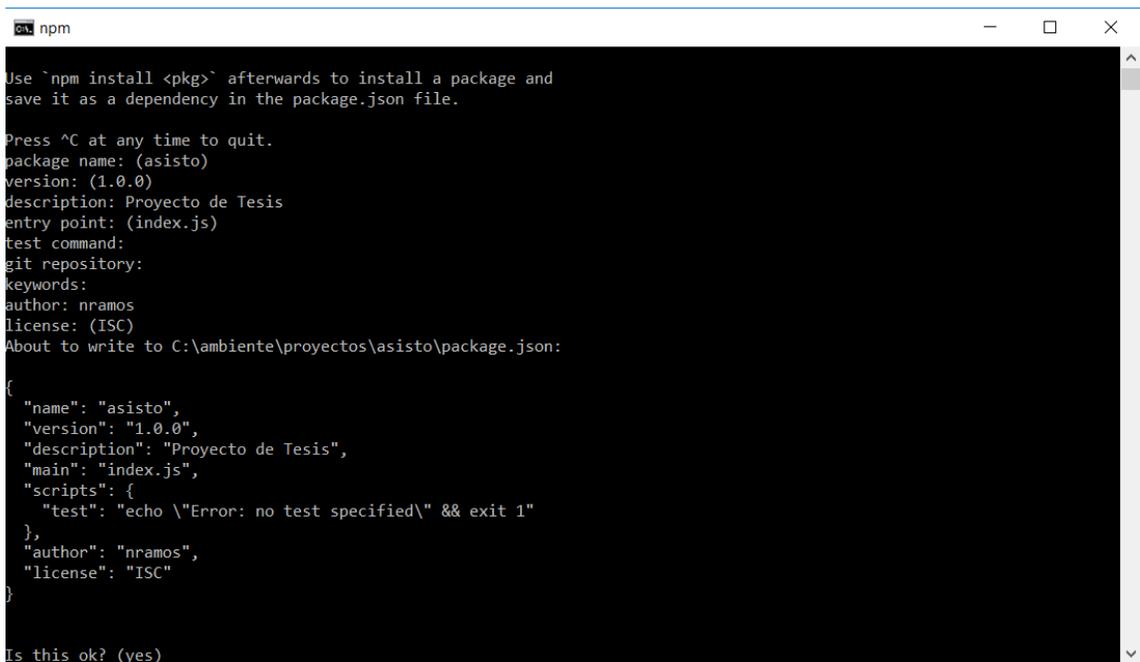


```
C:\WINDOWS\system32\cmd.exe
C:\ambiente\proyectos\asisto>npm --v
5.6.0
C:\ambiente\proyectos\asisto>
```

Fuente: elaboración propia, empleando Asisto.

Para crear el proyecto se debe ejecutar el comando 'npm init' en la consola del sistema, figura 58, al realizar la ejecución este mostrará una serie de preguntas que forman parte de la configuración inicial del proyecto.

Figura 58. Creación de proyecto Node.js



```
npm
Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (asisto)
version: (1.0.0)
description: Proyecto de Tesis
entry point: (index.js)
test command:
git repository:
keywords:
author: nramos
license: (ISC)
About to write to C:\ambiente\proyectos\asisto\package.json:

{
  "name": "asisto",
  "version": "1.0.0",
  "description": "Proyecto de Tesis",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "nramos",
  "license": "ISC"
}

Is this ok? (yes)
```

Fuente: elaboración propia, empleando Asisto.

Luego de responder a las preguntas, mostrará un resumen de la configuración la cual se almacenará en un archivo JSON con el nombre 'package.json', el cual será generado si se confirma la configuración establecida. Cabe mencionar que todas las dependencias que el proyecto contenga serán almacenadas en este archivo de configuración mediante una referencia conformada por el nombre de la dependencia y su versión.

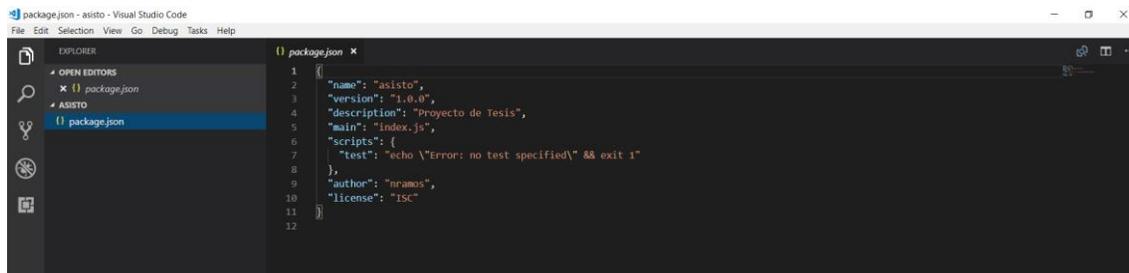
Figura 59. **Archivo de configuración, package.json generado**

Nombre	Fecha de modifica...	Tipo	Tamaño
 package	18/08/2018 18:38	Archivo JSON	1 KB

Fuente: elaboración propia, empleando Asisto.

El archivo tendrá la misma configuración que se estableció desde la consola del sistema, como se muestra en la figura 60.

Figura 60. **Configuración package.json sin dependencias**



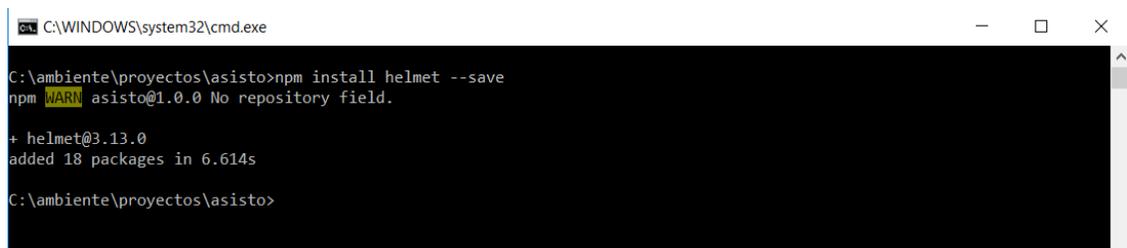
```
1 {
2   "name": "asisto",
3   "version": "1.0.0",
4   "description": "Proyecto de Tesis",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \\\"Error: no test specified\\\" && exit 1"
8   },
9   "author": "nramos",
10  "license": "ISC"
11 }
12
```

Fuente: elaboración propia, empleando Asisto.

Al finalizar la configuración del proyecto, se procede a instalar las dependencia que se utilizaran para el desarrollo, estas se instalan mediante el gestor de dependencias NPM de la siguiente forma por medio del 'npm install NOMBRE_DEPENDENCIA --save', se utiliza el comando '--save' para indicar a NPM que esta dependencia será instalada de forma local, es decir, únicamente el proyecto que se está configurando la tendrá.

```
'npm install helmet --save'
```

Figura 61. **Instalación de dependencia helmet**



```
C:\WINDOWS\system32\cmd.exe
C:\ambiente\proyectos\asisto>npm install helmet --save
npm WARN asisto@1.0.0 No repository field.
+ helmet@3.13.0
added 18 packages in 6.614s
C:\ambiente\proyectos\asisto>
```

Fuente: elaboración propia, empleando Asisto.

```
'npm install path --save'
```

Figura 62. **Instalación de dependencia path**

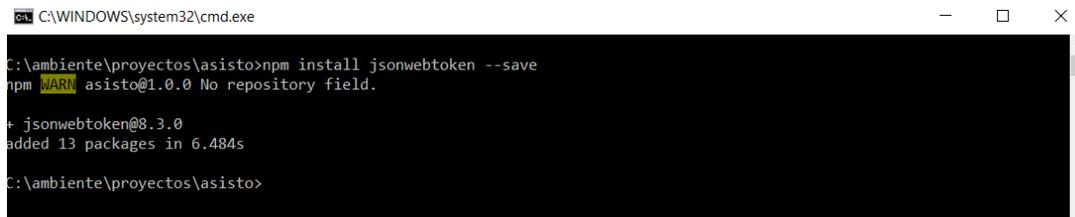


```
C:\WINDOWS\system32\cmd.exe
C:\ambiente\proyectos\asisto>npm install path --save
npm WARN asisto@1.0.0 No repository field.
+ path@0.12.7
added 3 packages in 7.342s
C:\ambiente\proyectos\asisto>
```

Fuente: elaboración propia, empleando Asisto.

'npm install jsonwebtoken --save'

Figura 63. **Instalación de dependencia jsonwebtoken**



```
C:\WINDOWS\system32\cmd.exe
C:\ambiente\proyectos\asisto>npm install jsonwebtoken --save
npm WARN asisto@1.0.0 No repository field.

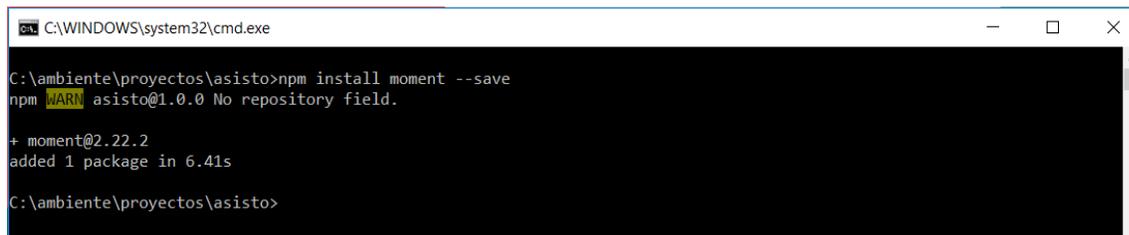
+ jsonwebtoken@8.3.0
added 13 packages in 6.484s

C:\ambiente\proyectos\asisto>
```

Fuente: elaboración propia, empleando Asisto.

'npm install moment --save'

Figura 64. **Instalación de dependencia momento**



```
C:\WINDOWS\system32\cmd.exe
C:\ambiente\proyectos\asisto>npm install moment --save
npm WARN asisto@1.0.0 No repository field.

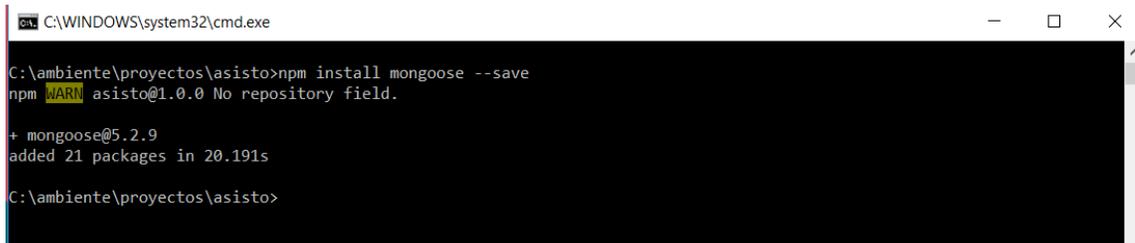
+ moment@2.22.2
added 1 package in 6.41s

C:\ambiente\proyectos\asisto>
```

Fuente: elaboración propia, empleando Asisto.

'npm install mongoose --save'

Figura 65. **Instalación de dependencia mongoose**

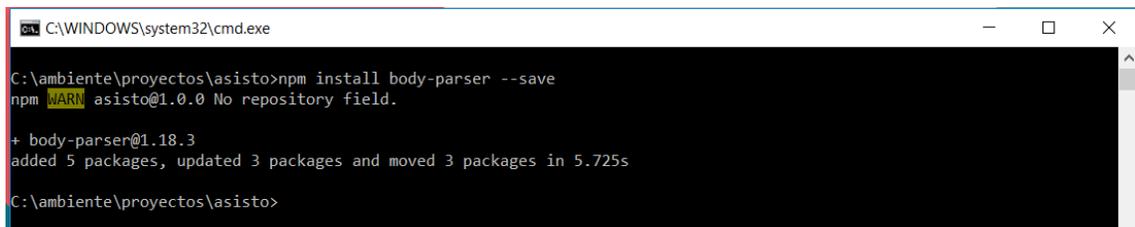


```
C:\WINDOWS\system32\cmd.exe
C:\ambiente\proyectos\asisto>npm install mongoose --save
npm WARN asisto@1.0.0 No repository field.
+ mongoose@5.2.9
added 21 packages in 20.191s
C:\ambiente\proyectos\asisto>
```

Fuente: elaboración propia, empleando Asisto.

'npm install body-parser --save'

Figura 66. **Instalación de dependencia body-parser**

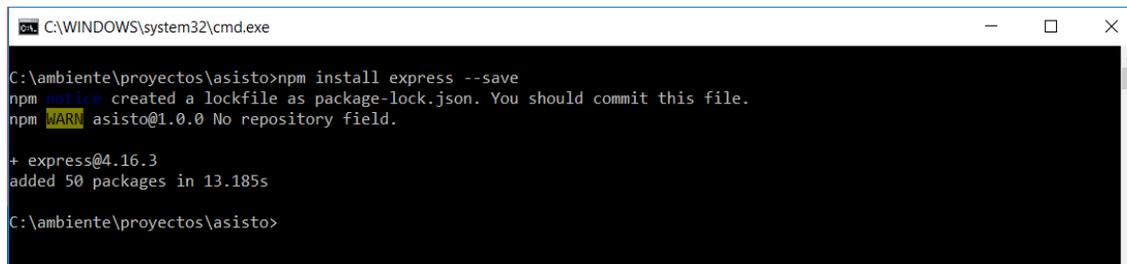


```
C:\WINDOWS\system32\cmd.exe
C:\ambiente\proyectos\asisto>npm install body-parser --save
npm WARN asisto@1.0.0 No repository field.
+ body-parser@1.18.3
added 5 packages, updated 3 packages and moved 3 packages in 5.725s
C:\ambiente\proyectos\asisto>
```

Fuente: elaboración propia, empleando Asisto.

'npm install express --save'

Figura 67. Instalación de dependencia express

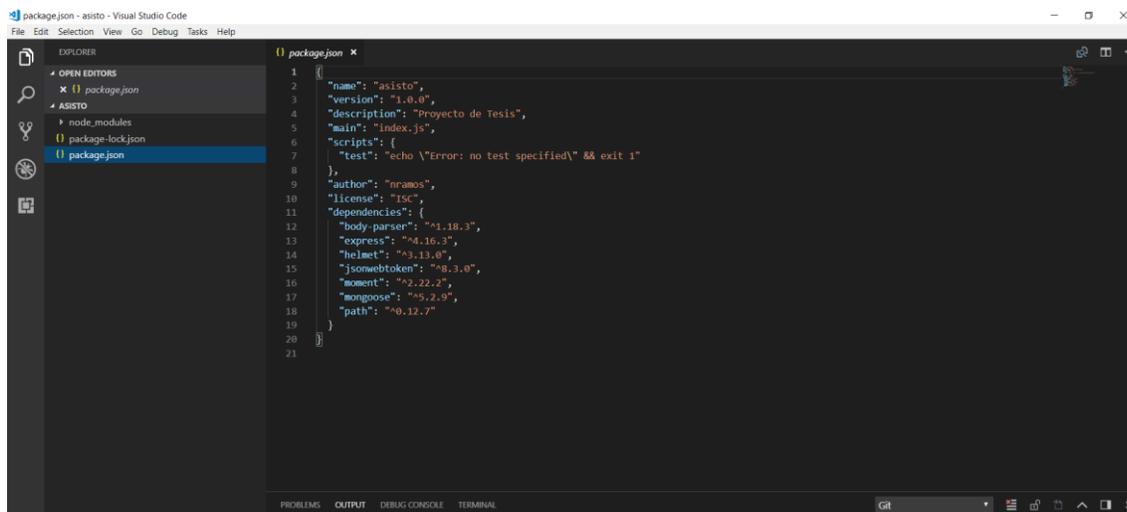


```
C:\WINDOWS\system32\cmd.exe
C:\ambiente\proyectos\asisto>npm install express --save
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN asisto@1.0.0 No repository field.
+ express@4.16.3
added 50 packages in 13.185s
C:\ambiente\proyectos\asisto>
```

Fuente: elaboración propia, empleando Asisto.

Finalizado el proceso de instalación de dependencias se debe verificar que esta se haya agregado al archivo de configuración 'package.json', como se muestra en la figura 68. De esta forma se finaliza la configuración del proyecto e inicia, el desarrollo de la aplicación.

Figura 68. Configuración package.json con dependencias



```
1 {
2   "name": "asisto",
3   "version": "1.0.0",
4   "description": "Proyecto de Tesis",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \\\"Error: no test specified\\\" && exit 1"
8   },
9   "author": "hramos",
10  "license": "ISC",
11  "dependencies": {
12    "body-parser": "^1.18.3",
13    "express": "^4.16.3",
14    "helmet": "^3.13.0",
15    "jsonwebtoken": "^8.3.0",
16    "moment": "^2.22.2",
17    "mongoose": "^5.2.9",
18    "path": "^0.12.7"
19  }
20 }
21
```

Fuente: elaboración propia, empleando Asisto.

4.6.2.2. Conceptos generales para el desarrollo

Para la creación de la aplicación web Asisto es necesario tener un conocimiento básico de los conceptos que se muestran a continuación.

4.6.2.2.1. *Blocking vrs non blocking I/O*

En Node.js existen dos tipos de llamadas a instrucciones: llamadas de bloqueo (*blocking*) y llamadas sin bloqueo (*non blocking*); I/O (entrada/salida) se refiere a la interacción con el disco y red del sistema.

Las llamadas de bloqueo se ejecutan de forma secuencial, es decir, en un bloque de código cada instrucción se ejecuta hasta que la anterior haya finalizado, lo que hace que ciertos procesos sean lentos y disminuya el rendimiento del sistema. El uso de este tipo de llamadas dependerá de la lógica del sistema, ya que en algunos procesos es necesario esperar que finalice una instrucción antes de continuar.

Las llamadas sin bloqueo se ejecutan de forma paralela o simultánea, esto significa que en un bloque de código cada una de las instrucciones se ejecuta de manera independiente. Al igual que las anteriores, el uso de este tipo de llamadas depende de la lógica que se desee implementar, ya que ciertos procesos que contienen un gran número de instrucciones se deben ejecutar al mismo tiempo.

La diferencia entre las llamadas de bloqueo y sin bloqueo es que las segundas utilizan *callbacks*, la cual es una forma de realizar una petición. Esto

permite ejecutar instrucciones sin esperar respuesta de una anterior, continuando de esta manera con la ejecución de las siguientes.

4.6.2.2.2. *Callbacks*

Callback es una función que se ejecuta después de una acción, básicamente es una función que se envía como parámetro a otra función, junto con una condición, que al cumplirse dicha condición la función que se envió como parámetro será ejecutada, es decir, el *callback*.

4.6.2.2.3. *Promesas*

Una promesa permite ejecutar una tarea, que luego de que esta finalice permite realizar una acción en particular, una promesa no es más que una función con la diferencia que esta recibe tres *callbacks* como parámetros; el primer *callback* llamado *resolve*, se ejecuta si la promesa tiene éxito, de lo contrario, la promesa llama al segundo *callback* llamado *reject*, indicando que ocurrió un error o esta no tuvo éxito. Y el tercer *callback* contiene todas las instrucciones que serán ejecutadas por la promesa.

4.6.2.2.4. *Async y await*

La expresión *async* es utilizada en las funciones para que el resultado de estas sea transformado en una promesa, la cual devuelve un valor si esta es exitosa y genera la excepción si hubo un error, simplificando así el uso de las promesas y facilitando el desarrollo de una aplicación.

La expresión *await* pausa la ejecución de una función *async* hasta que la promesa se finalice o se rechace. El valor de la expresión *await* obtendrá el

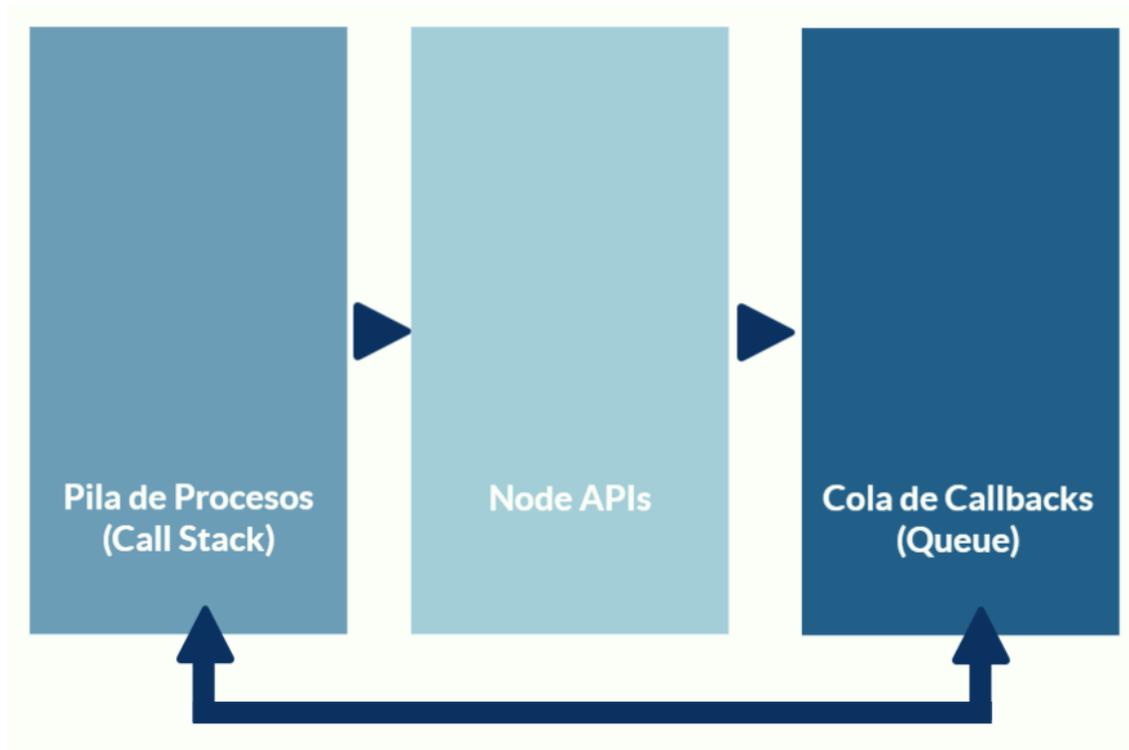
resultado de la promesa; es decir, el *resolve*, si esta tiene éxito; de lo contrario, tendrá el valor del *reject*, es decir, del rechazo, simplificando el uso de las promesas. Para poder hacer uso de *await* es necesario que esté contenido en una función *async* de lo contrario no funcionara.

4.6.2.2.5. Ciclo de vida de un proceso

Al ejecutar una aplicación, Node crea un proceso principal llamado *main*, el cual toma cada instrucción de código, y las coloca en una pila de procesos (*call stack*) para ejecutarlo y al finalizar el proceso, lo elimina de la pila. En el caso de las funciones, Node las registra en la pila, cuando una instrucción dispara dicha función; esta se agrega a la pila de procesos, por lo que Node comenzará a trabajar con el bloque de código que esta contiene, que son eliminadas al finalizar su ejecución.

Cuando se ejecutan tareas asíncronas, estas se registran en la pila de procesos, y luego son trasladadas a el Node APIs en donde Node está pendiente de esos procesos; es decir en espera de ejecutarlos, ya que está esperando respuestas o que tareas asíncronas finalicen. Cuando estas tareas finalizan Node las traslada a una cola de *callbacks*, que son todos los procesos que están listos para ser ejecutados, en la espera de que la pila de procesos que está ejecutando actualmente Node termine; mientras esto ocurre Node continúa con la ejecución de las demás instrucciones. En la figura 69, se muestra el ciclo de vida de un proceso.

Figura 69. **Ciclo de vida de un proceso en Node.js**



Fuente: elaboración propia, empleando Asisto.

4.6.2.2.6. Archivo de configuración package.json

Este es un archivo donde se encuentran todas las configuraciones de un proyecto Node.js, así como las dependencias que forman parte del proyecto. A continuación se describen las propiedades con las que este cuenta:

- **Package name:** contiene el nombre del proyecto, por defecto establece el nombre del directorio en el cual está contenido.

- Version: indica la versión del proyecto, por defecto cuando un proyecto es creado define la versión 1.0.0.
- Description: contiene la descripción del proyecto que será creado.
- Entry Point: indica el archivo JavaScript que será iniciado cuando se ejecute el proyecto.
- Test Command: se establecen comandos personalizados para la ejecución del proyecto.
- Git Repository: indica la dirección del repositorio GIT en el cual se versiona el proyecto.
- Keywords: se establecen palabras claves del proyecto.
- Author: indica quien desarrolla el proyecto.
- License: indica el tipo de licencia que tiene el proyecto, por defecto establece una licencia ISC (*internet systems consortium*), la cual es una licencia permisiva de software libre, lo cual indica que el proyecto puede ser distribuido como libre o privativo.
- Dependencies: contiene un listado de todas las dependencias con las que cuenta el proyecto, cada uno posee un nombre que identifica a la dependencia y la versión de esa, con la cual se está trabajando en el proyecto.

Estas configuraciones son muy útiles, cuando un proyecto se desea ejecutar en un nuevo ambiente, ya que todas las dependencias son instaladas

de forma local. Por lo tanto, solamente se debe ejecutar el comando 'npm install' el cual realiza la búsqueda de este archivo, para verificar la configuración del proyecto y descargar e instalar las dependencias que este utiliza.

5. RESULTADOS

5.1. Información recabada

Se obtuvo información para alimentar la aplicación Asisto, la cual centraliza los procesos administrativos y académicos de los siguientes sitios:

5.1.1. Portal de ingeniería

El portal de ingeniería cuenta con información de suma utilidad al estudiante; sin embargo, para llegar a obtener información puntual, se requiere de una extensa búsqueda.

Esto fue un punto muy importante a tomar en cuenta en el desarrollo de la aplicación y por lo cual se tomó como referencia la información que este sitio presenta, en especial las preguntas frecuentes.

5.1.2. Registro y estadística

El sitio de registro y estadística proporciona una imagen con la información de los pasos a seguir de un proceso muy importante para el estudiante el cual es la inscripción al ciclo académico de ingreso y reingreso.

Se recabó dicha información para agregarla entre los procesos que conforman la aplicación Asisto.

5.1.3. Control académico

El sitio de control académico proporciona información de varios procesos presentados en documentos que se pueden descargar.

Debido a que muchos de estos procesos son muy utilizados por los estudiantes, se tomó la decisión de enfocarse en la mayoría de ellos para ser integrados en la aplicación Asisto.

5.2. Accesibilidad y adaptabilidad

En la actualidad existen dispositivos con distintas características para mostrar la información, lo cual obliga a desarrollar aplicaciones que se adapten al entorno desde el cual son visualizadas, es por ello que la aplicación web Asisto cuentas se basó en estos requerimientos para dar una mejor experiencia al estudiante.

Con la información obtenida a través de la herramienta de Google Analytics la cual fue integrada en la aplicación Asisto, se mostró que en un inicio, al realizar el lanzamiento de la aplicación, la mayor cantidad de visitas se realizó por medio de dispositivos móviles.

A continuación, se muestra una tabla del porcentaje de visitas de distintos dispositivos, realizadas en el primer día de lanzamiento de Asisto con una cantidad de 496 usuarios.

Tabla II. **Lanzamiento de Asisto**

Categoría de dispositivo	Cantidad de usuarios	Porcentaje %
Móviles	415	83,50
Ordenadores	79	15,90
Tablets	3	0,60

Fuente: elaboración propia.

CONCLUSIONES

1. Se logró alimentar la aplicación web Asisto de información capturada de tres sitios distintos: control académico, portal de ingeniería y registro y estadística.
2. Asisto ha sido diseñado utilizando el conjunto de herramientas de Bootstrap para dotar a la aplicación de accesibilidad y adaptabilidad para que pueda acoplarse a distintos dispositivos.
3. Se desarrolló un sitio administrativo desde el cual se lleva el control de guías de procesos y preguntas frecuentes para ser presentadas en el sitio cliente Asisto; se estableció comunicación con personas encargadas del control académico para la administración de dicho sitio, sin embargo, el trámite aún está en proceso.

RECOMENDACIONES

Luego de desarrollar la aplicación web Asisto y ver diversos resultados proporcionados por la herramienta de Google Analytics se logró percibir el alcance de dicha aplicación.

Como Asisto proporciona datos de sumo interés para la comunidad estudiantil; se enumeran las siguientes recomendaciones:

1. Al no existir una entidad que proporcione la información de todos los procesos académicos y administrativos, se recomienda crear usuarios para cada una de las unidades académicas y escuelas para que estos lleven el control de su información.
2. Con el fin de proveer credibilidad al estudiante sobre la información que Asisto presenta, se sugiere que la aplicación sea puesta en servidores de la Facultad de Ingeniería con el uso del mismo dominio.
3. Por el impacto que Asisto puede tener, se considera necesario crear un acceso desde el portal de ingeniería a dicha aplicación.
4. Con el objetivo de mejorar y darle mantenimiento a la aplicación, se sugiere a la Escuela de Ingeniería en Ciencias y Sistemas, que este proyecto se establezca como una forma de realizar prácticas finales.
5. Con el fin de que la comunidad estudiantil se beneficie de la información que Asisto provee, se recomienda que la facultad promueva dicha

aplicación por medio de las diversas fuentes de publicación que esta posea.

BIBLIOGRAFÍA

1. Deutschland.de. *La primera página web del mundo*. [en línea]. <[https://www.deutschland.de / es / topic / economía / innovacion-tecnica/la-primer-pagina-web-del-mundo](https://www.deutschland.de/es/topic/economia/innovacion-tecnica/la-primer-pagina-web-del-mundo)>. [Consulta: 2 de septiembre de 2018].
2. Efectovisual. *Diseño responsivo ¿Qué es y en qué consiste?* [en línea]. <<https://www.efectovisual.cl/blog/70-diseno-responsivo-que-es-y-en-que-cosiste.html>>. [Consulta: 2 de septiembre de 2018].
3. FOWLER, Henry. *¿How many websites are there in the world? | Tek eye*. [en línea]. <<https://tekeye.uk/computing/how-many-websites-are-there>>. [Consulta: 2 de septiembre de 2018].
4. iLifebelt™. *Usuarios y uso de internet en Guatemala*. [en línea]. <<https://ilifebelt.com/usuarios-uso-internet-guatemala/2016/09/>>. [Consulta: 2 de septiembre de 2018].
5. Think with Google. *Estudio de consumer barometer 2017: el año de los móviles*. [en línea]. <<https://www.thinkwithgoogle.com/intl/es-419/recursos-y-herramientas/m%C3%B3vil/consumer-barometer-study-2017-year-mobile-majority/>>. [Consulta: 4 de septiembre de 2018].
6. TORNE, Marta. *Mobile First Index: nuevo cambio de Google. ¿Cómo te afecta?* [en línea]. <<https://www.inboundcycle.com/blog-de>>

inbound-marketing/mobile-first-index>. [Consulta: 4 de septiembre de 2018].

7. Wiboo Media. *¿Qué son las Aplicaciones Web? Ventajas y tipos de desarrollo web.* [en línea]. <<https://wiboomedia.com/que-son-las-aplicaciones-web-ventajas-y-tipos-de-desarrollo-web/>>. [Consulta: 2 de septiembre de 2018].