



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

FACULTAD DE INGENIERÍA

IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE  
TICKETS PARA SOPORTE TÉCNICO DE LA DICYG

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

P R E S E N T A

ANGEL BRITO SEGURA

DIRECTORA DE TESIS

M.I. TANYA ITZEL ARTEAGA RICCI



CIUDAD UNIVERSITARIA, Cd.Mx., 2023



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# Agradecimientos

A mis padres, Rosa y José Luis, que en todo momento me han dado todo su apoyo y me motivan a alcanzar cada una de mis metas.

A mi abuelita Sol y a mi querido tío Ale, que siempre me impulsaron para concluir mi carrera y seguirán siendo mi gran inspiración para seguir superándome día con día.

A mis padrinos y a mis primos (hermanos del alma), que me han acompañado en los momentos más importantes de mi vida, constituyendo una gran guía.

A mis amigos, por su entrañable e incondicional amistad.

A mi apreciable y muy querida profesora, la Dra. Martha Angélica Nakayama Cervantes que, desde entonces, tomó mi mano y me ha acompañado en el sendero de mi tan anhelada titulación. Todo mi cariño y agradecimiento por su ardua labor.

A mi directora de tesis, M.I. Tanya Itzel Arteaga Ricci, por su apoyo, consejos, recomendaciones y orientación para la realización del presente trabajo.

A la Unidad de Cómputo de la División de Ingenierías Civil y Geomática, por su cálida bienvenida y retroalimentación que coadyuvaron para sentar las bases del sistema. ¡Muchas gracias por haberme hecho sentir como un miembro más del equipo!

Al Ing. Jorge Alberto Rodríguez Campos, mi admirado profesor que, con sus cátedras magistralmente impartidas pero ante todo por su gran calidad humana, ha dejado una huella indeleble en mi ser.

A todos y cada uno de mis profesores y profesoras de la Facultad de Ingeniería, quienes con su sello distintivo preservan con pundonor la vida académica de la UNAM.

A la DGTIC, por constituir un gran baluarte en mi incipiente formación profesional y cuyas enseñanzas me permitieron sentar los cimientos del sistema.

Especialmente, a mi alma máter, la UNAM, que me ha brindado las mejores herramientas a las que un estudiante pueda aspirar para encauzar su carrera profesional. ¡Gracias, mi querida UNAM; te abrazo con infinito agradecimiento! ¡Siempre gracias, mi casa... mi arraigo!

# Índice general

<b>Introducción</b>	<b>1</b>
Planteamiento del problema . . . . .	2
Hipótesis y objetivo . . . . .	3
Organización de la tesis . . . . .	3
<b>1. Marco teórico</b>	<b>5</b>
1.1. Estado del Arte . . . . .	7
1.1.1. Bugzilla . . . . .	7
1.1.2. Jira Service Management . . . . .	8
1.1.3. Redmine . . . . .	9
1.1.4. Freshdesk . . . . .	10
1.2. Metodologías de desarrollo . . . . .	12
1.2.1. Metodologías tradicionales . . . . .	13
1.2.2. Metodologías ágiles . . . . .	15
1.2.3. Metodología a utilizar . . . . .	17
1.3. Tecnologías para desarrollo web . . . . .	19
1.3.1. Arquitectura cliente-servidor . . . . .	19
1.3.2. Sistemas Gestores de Bases de Datos . . . . .	21
1.3.3. Frameworks de desarrollo . . . . .	22
1.4. Patrones de diseño . . . . .	26
1.5. Ciclo de Vida de Desarrollo de Software . . . . .	27
<b>2. Análisis</b>	<b>30</b>
2.1. Planificación . . . . .	30
2.2. Desarrollo conceptual del sistema . . . . .	32
2.3. Especificación de requerimientos de software . . . . .	36
2.3.1. Propósito . . . . .	36
2.3.2. Alcance . . . . .	36
2.3.3. Perspectiva del producto . . . . .	36
2.3.4. Funciones del producto . . . . .	38
2.3.5. Características de los usuarios . . . . .	39

---

2.3.6.	Restricciones . . . . .	39
2.3.7.	Requerimientos funcionales . . . . .	39
2.3.8.	Requerimientos no funcionales . . . . .	40
<b>3.</b>	<b>Diseño</b>	<b>42</b>
3.1.	Arquitectura propuesta del sistema . . . . .	42
3.1.1.	Web Services Gateway Interface . . . . .	43
3.1.2.	Patrón de diseño en Django . . . . .	44
3.1.3.	Especificaciones de la arquitectura del SGT . . . . .	44
3.2.	Historias de Usuario . . . . .	46
3.2.1.	Administrador Ejecutivo . . . . .	47
3.2.2.	Administrador del Sistema . . . . .	47
3.2.3.	Líder Técnico . . . . .	48
3.2.4.	Supervisor . . . . .	49
3.2.5.	Encargado de Laboratorio . . . . .	49
3.2.6.	Técnicos . . . . .	49
3.2.7.	Solicitante . . . . .	50
3.3.	Diseño de la base de datos . . . . .	51
3.3.1.	Diseño conceptual . . . . .	51
3.3.2.	Diseño lógico . . . . .	51
3.3.3.	Diseño físico . . . . .	52
<b>4.</b>	<b>Desarrollo</b>	<b>54</b>
4.1.	Herramientas de software . . . . .	54
4.2.	Configuraciones iniciales . . . . .	55
4.3.	Módulos del sistema . . . . .	56
4.3.1.	Inicio de Sesión y Gestión de Usuarios . . . . .	57
4.3.2.	Gestión de Catálogos . . . . .	59
4.3.3.	Foros de Comunicación y Asesoría Tecnológica . . . . .	60
4.3.4.	Levantamiento, Asignación y Cierre de Tickets . . . . .	61
4.3.5.	Atención, Verificación y Validación de un Ticket . . . . .	63
4.3.6.	Informes Ejecutivos . . . . .	65
4.4.	Validaciones . . . . .	65
4.4.1.	Información de los usuarios . . . . .	66
4.4.2.	Gestión de contraseña . . . . .	67
4.4.3.	Conexión con el Sistema de Inventarios . . . . .	68
4.4.4.	Rechazo solución propuesta . . . . .	70
4.4.5.	Ticket de Mantenimiento Programado . . . . .	71

<b>5. Liberación del Sistema</b>	<b>73</b>
5.1. Pruebas . . . . .	73
5.1.1. Unitarias . . . . .	74
5.1.2. De integración . . . . .	75
5.1.3. Funcionales . . . . .	75
5.1.4. De extremo a extremo . . . . .	76
5.1.5. De regresión . . . . .	76
5.2. Puesta en producción . . . . .	78
5.3. Documentación para futuros mantenimientos . . . . .	79
<b>6. Resultados</b>	<b>80</b>
6.1. Experiencia de Usuario en el SGT . . . . .	80
6.1.1. Estética . . . . .	81
6.1.2. Funcionalidad . . . . .	81
6.1.3. Accesibilidad . . . . .	81
6.1.4. Credibilidad . . . . .	81
6.1.5. Intuitividad . . . . .	81
6.2. Procesos automatizados creados por el SGT . . . . .	81
6.3. Entrega del SGT a la Unidad de Cómputo . . . . .	82
6.4. Costos del proyecto . . . . .	82
<b>7. Conclusiones y Trabajo Futuro</b>	<b>84</b>
<b>Apéndices</b>	<b>91</b>
<b>A. Reuniones</b>	<b>92</b>
Minuta 1 . . . . .	92
Minuta 2 . . . . .	95
Minuta 3 . . . . .	97
Minuta 4 . . . . .	98
Minuta 5 . . . . .	100
Minuta 6 . . . . .	101
Minuta 7 . . . . .	102
Minuta 8 . . . . .	103
Minuta 9 . . . . .	104
<b>B. Documentación entregada</b>	<b>106</b>
Acta constitutiva . . . . .	106
Carta descriptiva . . . . .	117
Diccionario de datos de la base de datos del SGT . . . . .	120
Acta de cierre del proyecto . . . . .	131

Tríptico Solicitante . . . . .	145
Tríptico Encargado de Laboratorio . . . . .	147
<b>Referencias</b>	<b>149</b>

# Introducción

La inclusión de la tecnología en nuestra vida cotidiana ha venido a revolucionar la manera en que realizamos nuestras tareas y, particularmente, en la atención de soporte técnico ha permeado de manera trascendental, trayendo como consecuencia la imperiosa necesidad de eficientar los procesos. Sirva este corolario para señalar que, en la Unidad de Cómputo (UC) de la División de Ingenierías Civil y Geomática (DICyG) adscrita a la Facultad de Ingeniería de la Universidad Nacional Autónoma de México (UNAM), se brinda soporte técnico al personal que labora en dicha división; no obstante, estos usuarios tienen que realizar sus solicitudes a la UC a través de correo electrónico o llamada telefónica y, por su parte, los empleados de esta unidad administrativa tienen que efectuar el levantamiento de las mismas mediante un proceso manual. El seguimiento de atención en este proceso manual se vuelve inoperante porque en cualquier momento los usuarios van proporcionando información adicional respecto a las fallas que están presentando los equipos; información que, además, se remite innecesariamente a diferentes miembros del staff, lo que se traduce en una atención lenta y poco efectiva de cada una de las solicitudes recibidas porque se dispersa la información, amén del reparto desigual en su asignación.

Crear un sistema central en donde se pueda entablar una comunicación con cada uno de los usuarios que soliciten apoyo a la UC de la DICyG, resolver a cabalidad todos los problemas que se presenten con sus equipos de cómputo, así como el mantener los registros íntegros concernientes a la atención de todas y cada una de las solicitudes (tickets) que se demanden, constituyen una premisa fundamental para la implementación de un sistema de gestión de tickets para soporte técnico.

El empleo de un dispositivo tecnológico por parte de estos usuarios trae consigo la presunción de tener a su alcance un navegador web con el cual están completamente familiarizados. Derivado del uso constante de un navegador, la creación de este sistema web representa una enorme oportunidad para facilitarle al usuario el levantamiento de sus solicitudes y su óptima atención. Además, con un valor agregado que conlleva el poder utilizarlo en cualquier dispositivo; y, por ende, tener literalmente “en sus manos” el seguimiento de su ticket ingresando a la dirección del propio portal web para conocer sus últimas actualizaciones.



Lo anterior fundamenta una de las principales pretensiones de este sistema central a desarrollar: su accesibilidad; esto es, la facilidad con la que cualquier usuario pueda acceder, desde cualquier punto, para levantar sus solicitudes y ser atendidas de forma inmediata.

Para poder coordinar el trabajo de soporte técnico, utilizar un sistema centralizado en la UC de la DICyG será de vital importancia, lo que se sumará a la gran oportunidad de llevar un seguimiento efectivo por cada uno de los tickets levantados, aunado a un puntual registro de lo que se realiza en cada uno de sus equipos de cómputo, que de facto servirá para crear una correcta documentación por cada uno de los tickets atendidos.

La creación de un software a la medida para la UC de la DICyG se traducirá en una solución para mejorar la atención, seguimiento y verificación de los soportes técnicos que realizan en esta unidad administrativa, colocando al usuario como punto toral del ciclo de vida del ticket, optimizando el seguimiento de cada solicitud de soporte técnico que se requiera.

En los capítulos que se desarrollarán a continuación, se detallarán las bases de este software y se mostrará el procedimiento realizado para su análisis, diseño, desarrollo e implementación, que redundarán en una serie de conclusiones que permitirán establecer la relación directa entre los resultados obtenidos y el propósito por el cual fue creado.

## Planteamiento del problema

Actualmente, en la UC de la DICyG se realiza un proceso manual para el levantamiento de un ticket, donde los usuarios se comunican a la unidad y solicitan apoyo para solucionar problemas inherentes a los equipos de cómputo de la división. El ticket es asignado manualmente al staff de la UC, por lo que el reparto se hace de manera aleatoria sin considerar las cargas de trabajo de los miembros del equipo, toda vez que esta asignación lo realiza el propio staff conforme a sus posibilidades, lo que impide un reparto equitativo en los soportes que redunde en un trabajo colaborativo.

En la atención del ticket, los usuarios solicitantes establecen comunicación con los miembros del staff para ir agregando nuevos requerimientos al soporte levantado inicialmente. Esta comunicación se establece vía correo electrónico, redes sociales, telefónicamente, inclusive, de manera personal, lo que se traduce en una dispersión de la información que incide negativamente en la solución del soporte.

Ahora bien, el seguimiento de cada ticket levantado se realiza de manera manual, circunstancia que imposibilita la generación de un reporte final que muestre todo el trabajo llevado a cabo en cada uno de los equipos de cómputo en aras de contar con un histórico de ellos. Esta situación obliga a realizar una serie de gestiones tendientes a la búsqueda, recabación

y análisis de la información que se encuentra dispersa en cada uno de sus miembros para conocer el estado que guardan los equipos que les han sido asignados.

Para atender esta problemática, se propone la implementación de un Sistema de Gestión de Tickets para Soporte Técnico de la DICyG, que en definitiva se traducirá en un solo canal de comunicación entre los usuarios y el propio equipo de soporte técnico de la UC, lo que permitirá focalizar la atención de las solicitudes y, por ende, su pronta, expedita y convincente resolución. Por otra parte, permitirá mantener un histórico de los cambios realizados en cada equipo de cómputo, lo que redundará en soportes técnicos más rápidos y precisos.

## Hipótesis y objetivo

Con la creación de un Sistema de Gestión de Tickets para Soporte Técnico de la DICyG se constituiría una central digital de atención a usuarios, disponible desde cualquier punto, para los solicitantes que requieran de apoyo técnico, focalizando en todo momento la cabal atención de la solicitud y, por ende, la de todos los usuarios; dando paso, inclusive, en un futuro, a la necesidad de analizar las posibilidades de contar con personal de apoyo para un aprovechamiento integral del sistema.

El objetivo del presente proyecto de tesis consiste en crear un sistema web que lleve el control y seguimiento de la gestión de los tickets para soporte técnico realizada por la UC de la DICyG.

## Organización de la tesis

A continuación, se mencionan -a groso modo- los aspectos que contendrán cada una de las partes que componen el presente trabajo.

En el capítulo 1 se explican conceptos teóricos empleados a lo largo de esta tesis, así como un breve estado del arte de sistemas de soporte técnico, un análisis de las tecnologías a implementar en el sistema y patrones de diseño que actualmente se utilizan en la construcción de este tipo de software.

Ahora bien, la fase inicial del ciclo de vida del desarrollo de software se abordará en el capítulo 2. Esta fase recogerá los requerimientos del sistema, analizando el proceso actual que se realiza en la UC para la atención de soporte técnico. Aunado a ello, se detallarán los requerimientos de hardware necesarios para realizar la implementación de este software en cualquier servidor y un bosquejo general de la arquitectura del mismo.

Realizado el levantamiento de requerimientos, se llegará al capítulo 3 con el diseño del sistema, siguiendo desde luego la arquitectura propuesta en el capítulo que antecede. Cabe decir que este diseño se centrará en la creación de la base de datos, así como en diagramas UML (Lenguaje Unificado de Modelado) referentes al manejo del sistema a desarrollar.

Con base al diseño creado, en el capítulo 4 empezará el desarrollo del software con la utilización precisamente de un framework para desarrollo web y la implementación de la base de datos. Importante destacar que también se expondrán los módulos principales del sistema, así como las herramientas de software necesarias para realizar dicha labor.

Concluido el desarrollo, en el capítulo 5 se tratará la liberación del sistema, la cual está constituida por la enunciación de las pruebas realizadas y por el despliegue del mismo; todo ello con el propósito de obtener su desempeño y funcionalidad para distintos casos de uso.

Cerrado el ciclo de vida de desarrollo del sistema, en el capítulo 6 se expondrán los resultados derivados de su liberación al usuario final y de los procesos automatizados que resultasen de su creación.

Finalmente, en el capítulo 7 se formularán las conclusiones que se obtuvieron al implementar el Sistema de Gestión de Tickets y el trabajo futuro que puede desarrollarse para la mejora del sistema que redunde en una excelente atención al usuario.

# Capítulo 1

## Marco teórico

El desarrollo del presente trabajo implica necesariamente conocer un concepto inherente a la propia implementación de un Sistema de Gestión de Tickets para Soporte Técnico. Desde luego se trata del concepto de soporte técnico, el cual comprende una serie de aspectos que resulta importante precisar. Soporte técnico es el servicio que provee una organización a sus empleados con el propósito de que reciban una asistencia a sus problemas de software o hardware en los equipos de cómputo que utilizan para realizar sus actividades laborales, auxiliándoles también en las instalaciones y configuraciones de los mismos. La finalidad de este soporte es ayudar a los colaboradores de la organización a resolver sus problemas de la manera más eficiente posible, sea realizándolo directamente en el espacio del usuario (como su oficina), o bien, a distancia (vía telefónica, correo electrónico o chat) [1].

Las asignaciones más comunes de las que se encarga un equipo de soporte técnico son: manejo, instalación, mantenimiento, realización de diagnósticos de fallas de software o hardware, resolución de problemas y configuración de equipos informáticos, además de la imprescindible comunicación que debe entablarse con los usuarios para resolver sus problemas y/o indicar el estado de resolución de sus peticiones. Más aún, documentar el procedimiento seguido para resolver el problema en cuestión se vuelve fundamental en la optimización del servicio de soporte técnico [2]. Por todo ello, proporcionar este soporte a los trabajadores paralelamente se convierte en un instrumento de crecimiento y éxito para las propias organizaciones, que pueden emplear, inclusive, para la supervisión y correcta solución de los problemas que estrechamente vinculan los recursos humanos y materiales que se reportan [3].

El soporte técnico brinda dos tipos de asistencia [4]:

- 1) ***Intelectual***: Este soporte se centra en la capacitación de tecnologías, productos o servicios, que ocupan cotidianamente los trabajadores para realizar sus labores. El personal de soporte técnico transmite información y conocimiento para que el trabajador pueda

realizar los procedimientos necesarios para la operación de sus equipos de cómputo empleados en las actividades que se les encomiendan.

- II) **Tecnológica:** Este soporte realiza actividades técnicas que involucran un equipo de cómputo, incluyendo los servicios de mantenimiento -preventivo y correctivo- como reparaciones, revisiones y ajustes de hardware o software. El personal de soporte técnico realiza estas acciones en el equipo de cómputo, de acuerdo a lo reportado por el usuario responsable del mismo.

La documentación del soporte técnico es un factor determinante para eficientarlo. Ella se puede recabar a través de una bitácora que almacene cada una de las actividades realizadas en la atención del soporte; no obstante, se recomienda tener un sistema en donde se recoja toda esta información para que esté disponible al usuario del equipo de cómputo, quien podría, basándose en esta fuente de conocimientos, resolver de forma autónoma las dificultades ordinarias que se le presenten [5], lo que redundaría en la mejora en los tiempos de solución de problemas en los equipos de cómputo, reservando a los técnicos la resolución de casos extraordinarios o que conllevan cierta dificultad y que requieren más apoyo por parte de personal especializado para ser resueltos con éxito [1].

Para documentar la atención de estos problemas una buena opción la constituyen los tickets, ya que sería difícil mantener un seguimiento con documentos aislados y no sería escalable cuando hubiese más de un problema a resolver por técnico. El ticket de soporte técnico también constituye el elemento básico de cualquier trabajo relacionado con la experiencia del cliente, lo cual conlleva a que una organización pueda administrar y mantener la lista de problemas o preguntas comunes que sus usuarios pudieran tener. De tal manera que este tipo de documentación ayuda a mantener al usuario del equipo electrónico en el flujo de la atención del ticket, además de dar soporte a los técnicos que atienden el problema, permitiendo auditar los pasos realizados en el equipo en caso de que sea necesario [6].

Un ticket debe contener información sobre el problema que se está atendiendo, los pasos y procedimientos que se realizaron para solucionarlo y la solución a la que se llegó cuando se termina su atención. También es importante registrar las políticas asociadas en la atención del ticket, así como mantener -lo más concisa y actualizada posible- la información que contenga cada uno de ellos, para que cuando se requiera referenciar esta atención, se averigüe fácilmente lo que se tiene que hacer [7].

Esta importante actividad de documentación se puede implementar con una base de conocimientos (biblioteca en línea, que cuenta con varios formatos de contenido [8]: tutoriales, artículos, infografías, videos, guías paso a paso, páginas de preguntas frecuentes, foro de la comunidad, entre otros), donde los usuarios puedan encontrar todas las preguntas frecuentes, la resolución a las mismas y las tareas recurrentes, con todos los pasos a seguir, de una

manera sencilla y clara; así como un software de seguimiento de tickets, con el que todo el equipo de soporte técnico pueda conocer el estado de cada solicitud y los pasos a seguir para la solución del problema reportado, evitando con esta centralización de la información un menoscabo de la misma [1].

A continuación, se mostrarán algunos sistemas de soporte técnico con los que cuenta el mercado actual para un seguimiento de tickets y creación de bases de conocimiento que apoyen a la resolución de los problemas que se le presentan al usuario al realizar sus labores en los equipos de cómputo de la organización a la que pertenecen.

## 1.1. Estado del Arte

La industria de TI (Tecnologías de la Información) utiliza soluciones todo en uno que ayudan a seguir los problemas del usuario, comunicarse con ellos y facilitarles actualizaciones sobre sus solicitudes de soporte. Algunas de las más empleadas son las siguientes [7]:

### 1.1.1. Bugzilla

Es una herramienta de software libre, gratuita y multiplataforma, que permite dar un seguimiento de defectos o fallas de un sistema; además, puede ser utilizada como administrador de pruebas ya que puede conectarse, a su vez, con otras herramientas especializadas de pruebas, tales como Quality Center, Testlink, entre otras [9].

Es posible utilizar este sistema de seguimiento de manera individual o en grupo. Se pueden anexar comentarios, propuestas de solución, designar a los responsables de la solución; todo ello llevando una secuencia de las fechas en las cuales sucede cada evento y, si se configura adecuadamente, enviará correos electrónicos a los interesados del error [10].



Figura 1.1: Características principales de Bugzilla<sup>1</sup>

<sup>1</sup>Logo de Bugzilla recuperado de [JavaTpoint](#) el 12/03/2023

En la figura 1.1 que antecede se muestran las características más importantes de este software, de las cuales se destacan a continuación las que pueden ser empleadas en el seguimiento de tickets de soporte técnico [10]:

- ⇒ Informes programados (por día, hora, semana, mes y año) mediante correo electrónico
- ⇒ Detección automática de errores duplicados
- ⇒ Flujo de trabajo personalizado, así como control de la visibilidad del problema

Entre sus desventajas encontramos las siguientes [11]:

- Interfaz poco intuitiva
- No dispone de la capacidad de inclusión de repositorios adicionales al que por defecto tiene instalado, lo que evita contar con un mejor rendimiento en la búsqueda de bugs y con una optimización en la estructura de los datos que posee
- No permite aplicaciones de soporte
- No tiene capacidad de centralización de mayor información como costos, foros comunitarios ni planificaciones de tareas o funcionalidades añadidas al proceso de desarrollo de software

### 1.1.2. Jira Service Management

Este software de asistencia técnica permite que los técnicos que atienden a los empleados de la organización proporcionen respuestas fáciles y rápidas a quienes soliciten ayuda. Además, contribuye para que los técnicos puedan organizarse, estableciendo la prioridad de las solicitudes y su pronta resolución, proveyendo una interfaz intuitiva (como la que se muestra en la figura 1.2) para técnicos y usuarios con nulo conocimiento de computación [12].

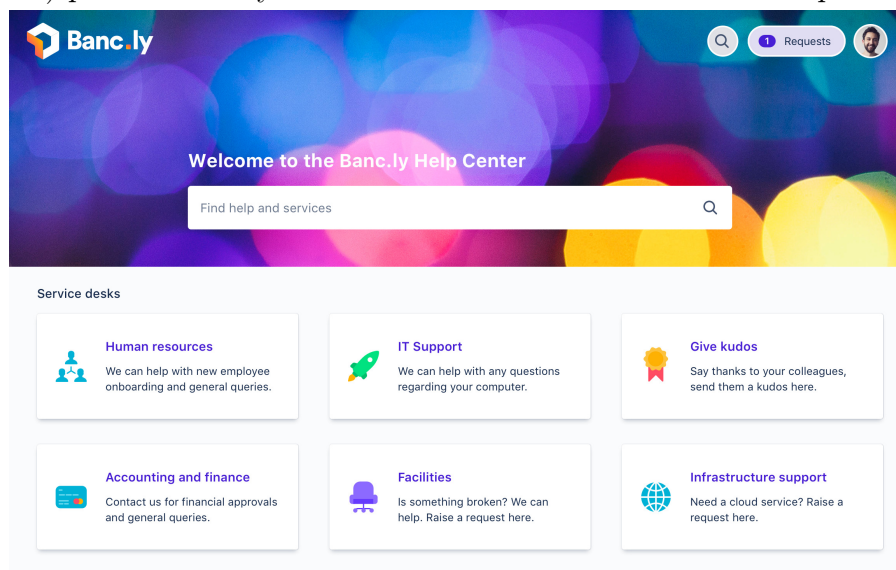


Figura 1.2: Portal de autoservicio de Jira Service Management<sup>2</sup>

<sup>2</sup>Imagen recuperada de [Atlassian](#) el 12/03/2023

Entre sus ventajas destacan su sencillez al utilizarlo porque cuenta con una página web que facilita las solicitudes de asistencia. Esta página web consiste en un portal de autoservicio donde el usuario puede conectarse directamente con los centros de ayuda y supervisar el progreso de solución de su problema. A través del portal, el equipo de soporte técnico puede trabajar en conjunto al conectar los tickets que genera Jira Service Desk con las incidencias de Jira Software. Este software de asistencia finaliza cada tarea de forma inteligente y mediante procesos automatizados en los que los tickets de soporte son fácilmente asignados a un departamento, o a un técnico en específico [13].

Dentro de sus desventajas podemos destacar, en primer término, que se trata de un software propietario costoso -el plan *Standard* cuesta 21 dólares por técnico-; en segundo término, que la versión gratuita solo proporciona hasta 3 técnicos trabajando en los soportes; aunadas al hecho de que esta herramienta viene acompañada de Jira Software, cuya licencia se adquiere de forma independiente. Además, otro inconveniente se encuentra en que su empleo requiere de la integración con Jira Confluence Software para la gestión de la base de conocimientos, en cuyo caso la interfaz puede volverse un poco lenta cuando varios usuarios coinciden en la misma tarea [13].

### 1.1.3. Redmine

Herramienta multiplataforma gratuita y de código abierto para la gestión y planificación de tareas y procesos. Esta planificación se realiza a través de peticiones y las distintas tareas pueden ser asignadas a miembros del equipo de soporte técnico para hacer un seguimiento de su evolución, añadir incidencias, obtener informes, controlar el tiempo, entre otras funcionalidades [14].

Entre sus ventajas destacan que cuenta con una incorporación de una wiki que facilita el acceso a la información a los miembros del equipo de soporte técnico, reduciendo el volumen de correos internos. También destaca la gestión de tareas dado que los distintos procesos pueden ser creados, asignados a usuarios, modificados, monitorizados y evaluados en tiempo real, permitiendo un control total de todo lo que ocurre durante el ciclo de vida del ticket. Además, esta herramienta facilita el control de la dedicación del tiempo de cada usuario a las distintas tareas que tiene asignadas y es una herramienta flexible con gran capacidad de personalización ya que permite la incorporación de una amplia variedad de complementos [14].

La enorme desventaja de esta herramienta es su instalación que puede resultar muy compleja debido a los paquetes requeridos que no están bien documentados y a las notificaciones por correo que están adheridas a todos los proyectos creados, aunado al hecho de que no se pueden personalizar. También se ha encontrado un fallo en el módulo de *Actividad* al no guardar



los cambios realizados sobre el contenido que se quiere visualizar y mostrar la actividad por defecto cuando se vuelve al módulo en uso, como se ilustra en la siguiente figura 1.3 [15].

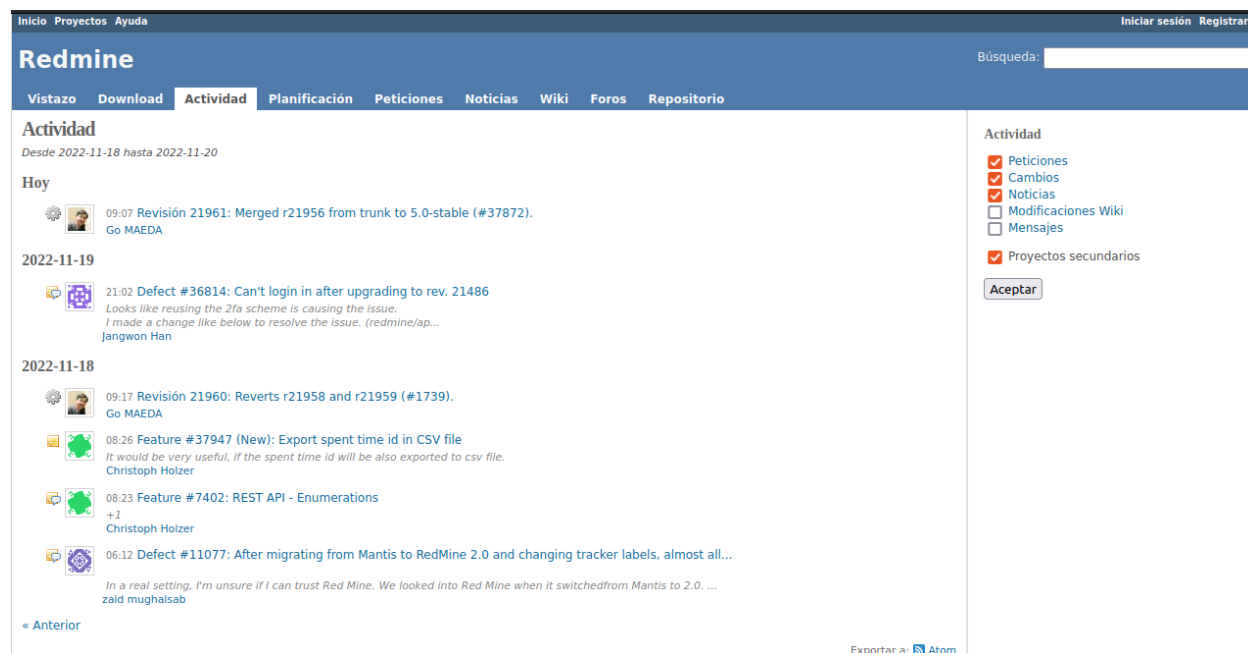


Figura 1.3: Módulo de Actividad en Redmine<sup>3</sup>

### 1.1.4. Freshdesk

El Software de Help Desk (mesa de ayuda) donde los trabajadores de una organización encuentran asistencia sobre productos, servicios y apoyo técnico [4] que se encuentra orientado al rubro de atención al cliente, ofrece la capacidad de unificar los canales de atención para mayor optimización del servicio. Esta central de ayuda unifica los mensajes y la información de los usuarios de equipos de cómputo de todos los canales y les brinda una mejor experiencia de soporte; además, al tratarse de un omnicanal, el equipo de soporte técnico puede responder a los correos electrónicos o llamadas de los usuarios y comunicarse con ellos a través del chat, directamente desde esta herramienta, además de poder acceder a las conversaciones pasadas de cualquier canal desde una única base de datos, permitiendo realizar un seguimiento a los usuarios y controlar las conversaciones convirtiéndolas en tickets de este software [16].

Entre las ventajas de utilizar este software destaca su disponibilidad inmediata al ser una solución SaaS (Software como Servicio en el que el proveedor de la nube desarrolla y mantiene el software, proporcionando actualizaciones automáticas y con un sistema de pago por uso [17]); además de ser de uso fácil debido a que se trata de una herramienta bien estructurada,

<sup>3</sup>Imagen recuperada de [Redmine](#) el 12/03/2023

icónica y con procesos coherentes y definidos. Cuenta con un alto nivel de personalización e integración con otras soluciones del mercado, tal y como se muestra en la figura 1.4, donde se tiene una integración con HubSpot (plataforma CRM -Gestión de las Relaciones con Clientes, por sus siglas en inglés- con herramientas de marketing, ventas, gestión de contenido, servicio de atención y operaciones relacionadas a administrar y analizar las interacciones con los clientes) [18–20].

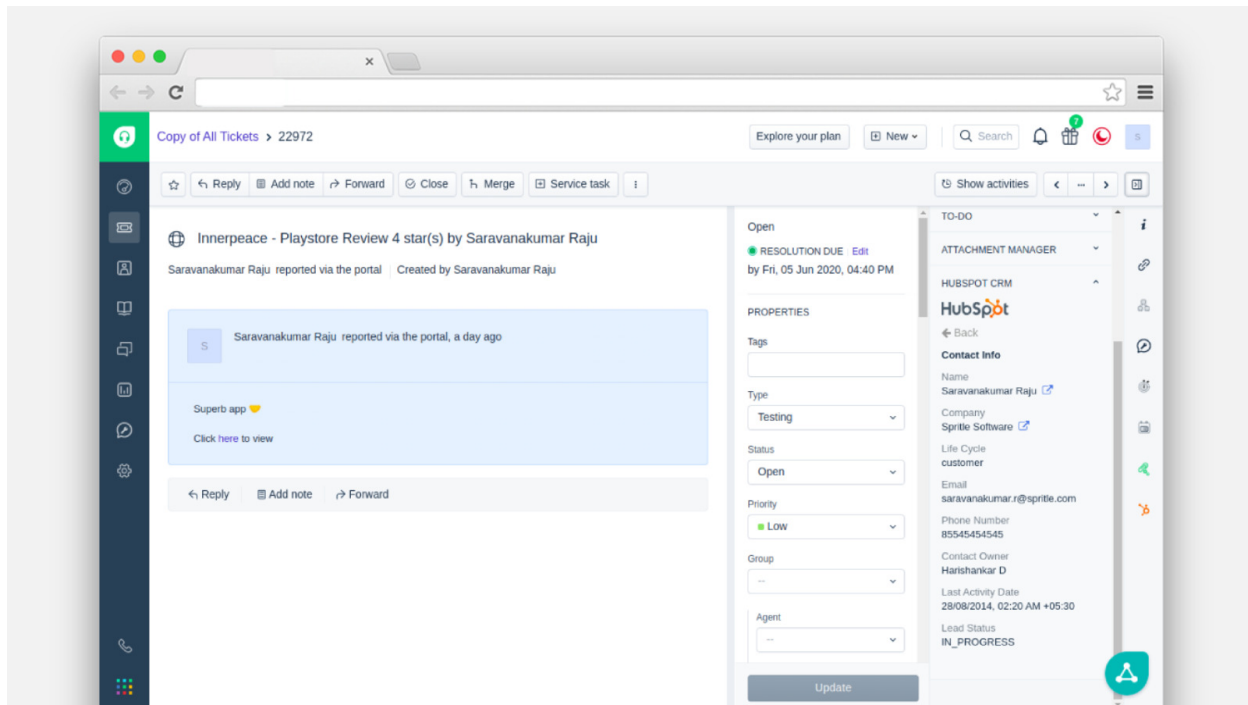


Figura 1.4: Integración de Freshdesk con HubSpot<sup>4</sup>

Entre sus desventajas destacan los tiempos de implementación de este servicio de centralización de canales y atención automatizada, de tal manera que los tickets se pueden ver de forma centralizada y las conversaciones pueden atenderse también de esa manera, lo que implica que todos los equipos pueden acceder a ello. Al ser un software propietario, los costos aumentan debido a que por técnico se tendría que pagar 15 dólares cada mes para poder utilizar el servicio [21].

Como podemos observar en esta breve exposición de los sistemas de soporte técnico con que cuenta el mercado, carecemos de una solución óptima que englobe tanto el seguimiento de tickets como la creación de una base de conocimientos, además de que las versiones gratuitas de las herramientas de software propietario detalladas con anterioridad limitan la

<sup>4</sup>Imagen recuperada de [HubSpot](#) el 12/03/2023

interacción completa con estas soluciones todo en uno. Es por esto que se propone la creación de un software a la medida que comprenda una gestión integral mediante un sistema web, central y accesible, tanto para el equipo de soporte técnico como para los usuarios de los equipos de cómputo de la organización. Al ser un software hecho a la medida, se adaptan los recursos (tanto monetarios como administrativos) que la organización destine para realizar la implementación de este proyecto. Además, permitiría la creación de reportes personalizados, siendo flexible y escalable para futuras necesidades de la organización.

Otra de las ventajas de la creación de este software a la medida es la automatización en el levantamiento de los tickets, la portabilidad del sistema para ser implementado en cualquier ambiente productivo, la mejora en la administración del personal a cargo de la atención de los tickets, la facilidad del usuario para dar seguimiento a su problema, incluyendo el control de los cambios realizados en la atención de cada dispositivo de la DICyG, manteniendo una integridad y consistencia en la información introducida en un solo sistema que comprenda todas y cada una de las actividades referentes al soporte técnico que realiza la UC.

Para realizar dicho software es necesario un conjunto de procedimientos que muestren los pasos a seguir y cómo implementarlos de una manera óptima; es por ello que, a continuación, se mostrarán las principales metodologías de desarrollo que existen y cuál será la que se seguirá a lo largo del presente trabajo.

## 1.2. Metodologías de desarrollo

En el desarrollo de software se utilizan metodologías que permiten organizar el trabajo de una manera que resulte lo más ordenada posible. Estas metodologías se componen de un conjunto de técnicas y métodos que se aplican para diseñar soluciones de software informático, así como para reducir el nivel de dificultad, permitiendo organizar las tareas, agilizar el proceso y mejorar el resultado final de las aplicaciones a desarrollar [22].



Figura 1.5: Diagrama de metodología tradicional contra ágil<sup>5</sup>

<sup>5</sup>Imagen recuperada de [STIC Noticias](#) el 12/03/2023

Como se muestra en la figura 1.5, en la actualidad existen dos grandes grupos de metodologías de desarrollo de software: tradicionales y ágiles. A continuación, se mostrarán las principales metodologías de cada grupo.

### 1.2.1. Metodologías tradicionales

Este grupo de metodologías establecen una disciplina de trabajo sobre el proceso de desarrollo del software, con el propósito de alcanzar un software más eficiente. Se caracterizan por definir y establecer total y rígidamente cada uno de los requisitos al inicio de los proyectos de ingeniería de software [23].

Estas metodologías centran su atención en llevar una documentación exhaustiva de todo el proyecto: planificación, control del proceso y especificaciones de requerimientos. Este control del proceso se realiza mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada [24]. Es así que siguen un enfoque lineal donde las etapas del desarrollo de software deben complementarse secuencialmente una detrás de otra. La captura de requerimientos se realiza por una única vez al principio del proyecto. Precisamente por ello, la estimación tendrá mucha importancia porque de ella dependen todos los recursos que se emplearán en el proyecto [25]. Entre las principales metodologías tradicionales o clásicas, se encuentran las siguientes:

- **Cascada:** Esta metodología desarrolla cada una de sus etapas de arriba a abajo (Requerimientos, Diseño preliminar de software, Análisis, Diseño de proyecto, Codificación, Prueba y Observaciones), llevando a cabo sus funciones en etapas diferenciadas y en estricto orden. Es necesario revisar el producto creado antes de pasar a la siguiente etapa para ver si está listo. Entre sus ventajas se encuentran la planificación sencilla y la mejora en la calidad del software, mientras que sus desventajas son los resultados visibles hasta el final y lo costoso de corregir errores [23].
- **Iterativo Incremental:** En esta metodología se va desarrollando el producto final de manera progresiva, en donde cada etapa añade una nueva funcionalidad y el software se puede empezar a utilizar, incluso, antes de que se complete totalmente. Su principal ventaja es que es la metodología más flexible en este grupo toda vez que descompone un proyecto en una serie de incrementos, los cuales suministran una porción de la funcionalidad respecto de la totalidad de los requerimientos del proyecto, y en donde cada uno de éstos tiene asignada una prioridad, los cuales pueden ser entregados según el orden de prioridad asignado en el incremento [23]. Entre sus desventajas se encuentran que no es recomendable utilizarla para casos de sistemas de tiempo real, alto nivel de seguridad, procesamiento distribuido o de alto índice de riesgos, ya que requiere de mucha planificación administrativa y técnica, además de metas claras para conocer el estado del proyecto [26].

- **Prototipado:** Esta metodología, como su nombre lo indica, se fundamenta en la elaboración de un prototipo que se construye rápidamente para que el usuario lo pruebe y proporcione su retroalimentación sobre el producto de software entregado, lo que permite realizar ajustes pasando de especificaciones de requerimientos incompletas a completas en cada subciclo del prototipo desarrollado [23]. La ventaja de seguir este método de prueba y error es que será útil cuando el usuario conozca los objetivos generales del sistema y no pueda definir los detalles finos del mismo. Esta metodología también favorece la adaptabilidad de un sistema en cuanto su usabilidad e interacción humano-computadora. Dentro de sus desventajas destaca el hecho de que el usuario puede ver rebasadas sus expectativas dado que los prototipos tienen mucha estética pero poca funcionalidad, amén de que los desarrolladores podrían tomar decisiones erróneas en cuanto al diseño al implementar versiones parciales y no centrarse en una versión final para el usuario [26].
- **Espiral:** Esta metodología, como su nombre lo indica, consiste en una serie de ciclos que se repiten en forma de espiral, comenzando desde el centro en donde se conjuga la naturaleza iterativa de la metodología *Prototipado* con los aspectos controlados y sistemáticos de la metodología *Cascada*. Dentro de las ventajas de esta metodología evolutiva se encuentran la minimización de riesgos, incorporación de objetivos de calidad y el hecho de ser considerada una metodología universal que combina otras metodologías (tradicionales y ágiles). Sus desventajas estriban en el tiempo invertido para el desarrollo del sistema, ya que cada etapa del desarrollo de software es mediante espirales que se repiten hasta entregar el producto terminado; además de tratarse de una metodología costosa y que requiere de experiencia en la identificación de riesgos [26].
- **Desarrollo Rápido de Aplicaciones (RAD):** Esta metodología permite desarrollar software de alta calidad en un corto periodo de tiempo, debido a que da prioridad a las entregas, las cuales se realizan con un menor número de repeticiones del ciclo de trabajo; lo anterior para lograr una aplicación completa de forma rápida. Entre sus ventajas destacan que se trata de un desarrollo más flexible y que sus iteraciones rápidas permiten reducir el periodo de desarrollo, agilizando la entrega del producto de software final. Entre sus desventajas encontramos que los costes son mucho más altos, que se requiere una mayor intervención de los usuarios y que las funcionalidades del producto entregado son limitadas debido al poco tiempo del que se dispone para desarrollarlas [22, 23].

Es importante acotar que en estas metodologías los ciclos de desarrollo son poco flexibles y no permiten realizar cambios, por lo que es recomendable adoptarlas en proyectos con un problema conocido y una solución bien definida; es decir, donde ya se cuenta con mucha experiencia con un determinado tipo de producto, lo que permitirá perfectamente hacer esti-

maciones. Es la opción ideal para proyectos donde los requisitos no cambian, las condiciones del entorno son conocidas y estables; así como cuando los proyectos son cortos [25].

### 1.2.2. Metodologías ágiles

Estas metodologías surgen como una alternativa a las tradicionales porque ayudan a reducir la probabilidad de fracaso por subestimación de costos, tiempos y funcionalidades en entornos cambiantes. Se basan en la metodología incremental, en la que en cada ciclo de desarrollo se van agregando nuevas funcionalidades a la aplicación final. Sin embargo, los ciclos son mucho más cortos y rápidos, por lo que se van agregando pequeñas funcionalidades en lugar de grandes cambios [25].

Este tipo de metodologías permiten construir equipos de trabajo autosuficientes e independientes que se reúnen frecuentemente para compartir y comentar novedades. Poco a poco, se va construyendo y puliendo el producto final, a la vez que el cliente puede ir aportando nuevos requerimientos o correcciones, ya que puede comprobar cómo avanza el proyecto en tiempo real [22].

Entre las metodologías ágiles más importantes se encuentran las descritas a continuación:

- **Programación extrema (XP):** Esta metodología de desarrollo de software está basada en las relaciones interpersonales, las cuales se consideran la clave del éxito. Su principal objetivo es crear un buen ambiente de trabajo, en equipo, donde prevalezca una retroalimentación constante del cliente [22]. Entre sus ventajas, se encuentra que todos los requerimientos se expresan como escenarios (historias de usuario), los cuales se implementan directamente como una serie de tareas donde los programadores trabajan en parejas y desarrollan pruebas para cada tarea completada [27]. Dentro de sus desventajas se encuentran que es difícil documentar dado que se trabaja de forma rápida y con cambios constantes; además de que no fomenta el uso de estructuras formales y existe la posibilidad de problemas con el cliente en virtud de que sus propuestas/cambios pueden ser no viables para el proyecto en la cuestión técnica [28].
- **Kanban:** Consiste en dividir las tareas en porciones mínimas y organizarlas a través de un tablero de trabajo dividido en: tareas pendientes, en curso y finalizadas. De esta forma, se crea un flujo de trabajo muy visual basado en tareas prioritarias y va incrementando el valor del producto [22]. Entre sus ventajas destaca que el flujo de trabajo es claro en cada fase del proyecto debido a la organización de tareas por estados, con duración estimada y una correcta distribución, con lo cual es posible detectar cualquier problema que se genere durante el transcurso de la tarea o actividad que se esté desarrollando. Por su parte, las desventajas de utilizar esta metodología de desarrollo es que asume sistemas de producción repetitivos, por lo que las variaciones o eventos

inesperados en las tareas que se realicen pueden afectar negativamente el resultado final, además de que es difícil de manejar cambios provocados por la acumulación de nuevas tareas ya que no es posible anticiparse ante esta demanda de trabajo [29].

- **Lean:** Está configurado para que pequeños equipos de desarrollo muy capacitados elaboren cualquier tarea en poco tiempo. Los activos más importantes son las personas y su compromiso, pasando a un segundo plano el tiempo y los costes. El aprendizaje, las reacciones rápidas y potenciar el equipo son fundamentales para la correcta entrega del sistema [22]. Entre sus ventajas se encuentra la entrega temprana del producto y la creación de un equipo motivado en la que se desarrolla la capacidad de decisión de sus miembros. Las desventajas de esta metodología es la dependencia de la cohesión del equipo y los compromisos individuales de los miembros del mismo y que por lo regular la toma de decisiones es complicada ya que depende de la disciplina de sus integrantes para poder llegar a una decisión segura y a tiempo para resolver el problema presentado [30].
- **SCRUM:** Metodología de adaptación iterativa e incremental para optimizar la predictibilidad y el control del riesgo, rápido, flexible y eficaz, diseñado para ofrecer un valor significativo de forma rápida en todo el proyecto [31]. Entre sus ventajas se encuentra que los usuarios pueden colaborar en cada una de las fases del proceso y proponer soluciones, que los problemas que puedan surgir son solucionados en el momento de su aparición y que permite establecer metas mensurables, con lo que se mantiene a un equipo productivo todo el tiempo. Sus desventajas son que todo el equipo debe conocer muy bien los principios y marco teórico de esta metodología, que el éxito del proyecto depende de la preparación y experiencia del Scrum Master y que si una tarea no se finaliza, el resto de pendientes comenzará a postergarse indefinidamente hasta que se complete y se llegue a la siguiente iteración [32]. En la figura 1.6 se muestra el proceso general que se tiene en esta metodología de desarrollo.



Figura 1.6: Proceso general en SCRUM<sup>6</sup>

<sup>6</sup>Imagen recuperada de Sinnaps el 12/03/2023

### 1.2.3. Metodología a utilizar

Se utilizará una metodología ágil para poder adaptar el software a las necesidades que le vayan surgiendo al cliente, lo que se traducirá en una aplicación web más funcional [22]. Debido a que este tipo de metodologías incorporan mecanismos de gestión del cambio, el usuario final tendrá un papel clave por la generación de nuevos requerimientos.

La metodología de trabajo estará basada en SCRUM, que si bien precisa tener argumentos en contra -los cuales se han enunciado en líneas anteriores-, lo cierto es que para este proyecto de tesis la manera en qué se organizan las tareas resultan muy convenientes, aunado a sus propios atributos, tales como la flexibilidad aplicable a los cambios de requerimientos que se pudiesen generar de las reuniones con el cliente, su fácil implementación al definir de forma clara cada rol, evento y documentación que se estará realizando en cada etapa del desarrollo del sistema, así como la definición de tiempos y entregables iterativos desde el inicio del proyecto que permiten ir mostrando las funcionalidades del sistema con demostraciones en las que se recibe retroalimentación por parte del cliente para continuar o cambiar el desarrollo realizado hasta el momento.

Las razones expuestas en el párrafo que antecede motivan la elección de la siguiente metodología basada en SCRUM para el presente desarrollo:

#### Líder de Proyecto

Este rol será desempeñado exclusivamente por el tesista Angel Brito Segura y realizará las siguientes actividades:

- ⊗ Crear los siguientes documentos:
  - a) **Historias de usuario:** Casos de uso del producto de software
  - b) **Product Backlog (requerimientos):** Listado de todas las características del producto de software
  - c) **Acceptance Criteria:** Acuerdo con los stakeholders (interesados) principales y su contenido versará en todas las condiciones que deban cumplirse por cada entrega
- ⊗ Administrar el tablero de Kanban
- ⊗ Desarrollar el producto de software.
- ⊗ Generar la documentación referente al producto

Si contáramos con un equipo de trabajo aunque fuese pequeño, una propuesta de distribución de estas responsabilidades -atribuidas en este desarrollo exclusivamente al tesista- se basaría



en una asignación de roles que conllevaría, no sólo a la estructuración del propio equipo, sino a algo más importante: a la concientización de responsabilidades.

Así, esta propuesta la constituiría el *Líder de Proyecto* que se encargaría de la gestión del equipo de trabajo, priorizando las entregas oportunas de incrementos de software y manteniendo constante comunicación con los stakeholders en todo lo concerniente a los requerimientos del mismo; el *Desarrollador FrontEnd*, cuyas actividades a su cargo serían las de diseño y creación de las interfaces de usuario (la interacción con el desarrollador de BackEnd en torno a la información necesaria por almacenar y mostrar en el sistema sería una constante, al igual que las presentaciones con los stakeholders para conocer sus experiencias de usuario); y el *Desarrollador BackEnd*, quien tendría la encomienda de la base de datos y del procesamiento de la información recibida a través de las interfaces de usuario.

Desde luego, esta propuesta comprende a un *Documentador* que genere toda la documentación referente al sistema (con un adecuado tratamiento de ella) y que haga las veces de “tester”, recabando las pruebas que realice cada desarrollador en el ámbito de sus competencias.

### Stakeholders (Interesados)

La lista de interesados en el proyecto está integrada por:

- ▷ **Cliente:** Unidad de Cómputo de la DICyG
- ▷ **Usuarios:** Staff de la Unidad de Cómputo y personal de la DICyG (solicitantes)
- ▷ **Patrocinadora:** M.I. Tanya Itzel Arteaga Ricci (Jefa de la Unidad de Cómputo de la DICyG)

### Sprint

Iteración en la que se dividirá el desarrollo del proyecto. Cada una tendrá una duración de 3 a 4 semanas y se entregará un avance del mismo al finalizar cada periodo. Consistirán en lo siguiente:

#### a) **Sprint Planning**

Se plantea el trabajo a realizar durante la iteración de acuerdo al *Prioritized Product Backlog* (requerimientos que se priorizan y se asignan a cada sprint). Se define lo siguiente:

- ✓ **Sprint Goal:** objetivo a alcanzar al terminar esta iteración

✓ **Sprint Backlog:** planificación de la iteración a través del tablero de Kanban, que a su vez contendrá las siguientes secciones: *Product Backlog*, En Proceso y Terminado

b) **Sprint Development**

Se realiza el trabajo planteado para la iteración. Una vez que todos los elementos del tablero Kanban estén en la columna de Terminado y se cumple el *Acceptance Criteria*, se solicita reunión con los Stakeholders principales.

c) **Sprint Review Meeting**

Reunión con los Stakeholders principales donde se presenta el avance del proyecto para su inspección y adaptación correspondientes. Se actualizan el *Product Backlog* y las historias de usuario con las nuevas condiciones que puedan afectar al proyecto. Al finalizar, se define lo siguiente: *Prioritized Product Backlog*, *Acceptance Criteria* y fecha de la próxima reunión.

Cabe precisar que, una vez concluida la iteración, iniciará otra hasta la conclusión del proyecto.

Una vez definida la metodología de trabajo a utilizar, se detallan a continuación las tecnologías con las que se cuenta para la creación del sitio web y la implementación del Sistema de Gestión de Tickets para Soporte Técnico de la DICyG.

## 1.3. Tecnologías para desarrollo web

Este tipo de tecnologías sirven para acceder a todos aquellos recursos que se tienen disponibles a lo largo de la red. Mediante el uso de un explorador web podemos obtener dicha información; éste nos facilitará navegar por Internet: abrir una página, visitarla, abrir una pestaña e ir a otra página web sin ningún tipo de problema, realizar compras en línea, visitas virtuales por museos, entre otras actividades [33]. Todo ello con agilidad, facilidad y comodidad, lo cual se ha logrado gracias a la implementación del patrón arquitectónico Cliente-Servidor. Se explica a continuación en qué consiste este estilo de arquitectura de software:

### 1.3.1. Arquitectura cliente-servidor

De acuerdo al *Software Engineering Institute* [34], la arquitectura de software se refiere a las estructuras de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos [35]. Está basada en modelos, patrones y abstracciones teóricas que se vuelven una guía detallada, que nos permite entender cómo van a encajar cada una de las piezas de software creadas en un producto final para su entrega y

despliegue sin problemas. Consecuentemente, esta planificación nos permite elegir el mejor conjunto de herramientas para llevar a cabo productos de software confiables y determinar los factores económicos y humanos durante el proceso de creación de software [36].

El patrón arquitectónico más utilizado en la creación de sitios web y servicios en línea es el de Cliente-Servidor. Este estilo arquitectónico distribuido está compuesto de dos componentes principales: proveedor (servidor) y consumidor (cliente). El procesamiento de la información y almacenamiento de los datos se lleva a cabo en un servidor, donde múltiples clientes se conectan a él para recuperar todos los recursos a través de la red local o global. Cabe decir que en este patrón el cliente constituye una capa que representa a los datos y es quien detona acciones (peticiones) para modificar el estado del servidor [37].

Dentro de las ventajas de utilizar este patrón arquitectónico encontramos su centralización, donde todos los recursos disponibles se hallan en un único punto. También, dentro de ellas encontramos su escalabilidad, donde cada pieza de software trabaja de manera independiente, además de su fácil administración. Entre sus desventajas destacan que la disponibilidad de los recursos depende de un único servidor que atiende las peticiones de los clientes, produciendo la necesidad de contar con un software y hardware específico para poder atender dicha demanda [36].

Existen dos grandes grupos de tecnologías web que son las más usadas dentro del desarrollo de software: FrontEnd y BackEnd, aunque debe decirse con toda claridad que si bien permiten realizar desarrollos web dentro de este patrón, no son conceptos exclusivos de esta arquitectura cliente-servidor [38]:

- + **FrontEnd:** Se trata de las utilizadas del lado del cliente. Aparecen en todo tipo de dispositivos que utilizamos a diario para conectarnos a los servidores a través de Internet. Como norma general, este tipo de tecnología viene incorporada en los diferentes navegadores web. En ellos encontramos a HTML, CSS y JavaScript como las principales tecnologías que se utilizan.
- + **BackEnd:** Son las empleadas del lado del servidor. En este caso, el servidor las implementa para gestionar todas aquellas peticiones de información que le llegan y, de este modo, gestionar también la base de datos alojada en los mismos. Toda esa información será devuelta a los dispositivos para que sea visualizada a través de tecnologías PHP, Java o Python.

Ahora bien, es preciso acotar que para poder almacenar y acceder a los datos que arroje cualquier desarrollo web, es necesario contar con Sistemas Gestores de Bases de Datos para poder manejar toda la información generada por el sitio web.

### 1.3.2. Sistemas Gestores de Bases de Datos

Un Sistema Gestor de Bases de Datos (DBMS, por sus siglas en inglés) consiste en una colección de datos interrelacionados y en un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos, contiene información relevante para una organización. El objetivo principal de un DBMS es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente [39]. Actualmente en el mercado existen muchos sistemas de software libre y gratuitos, con respecto a los cuales en la siguiente tabla 1.1 se muestran las opciones más viables para este proyecto:

DBMS	Características	Requerimientos mínimos
<b>MariaDB</b>	Se puede usar en cualquier lugar donde se utilizó MySQL con anterioridad ya que es un proyecto derivado de él. Se distribuye bajo la licencia GPL, por lo que se trata de un sistema de software libre que puede utilizarse de manera gratuita en cualquier proyecto [40]	512 MB de memoria RAM, 1024 MB de memoria virtual y 1 GB de espacio en disco duro [41]
<b>MySQL Community Edition</b>	Base de datos esencial, especialmente en los sitios web que tienen la opción de registrar usuarios para iniciar sesión, es de descarga gratuita y el procesamiento de información es lento para tablas grandes, por lo que no se recomienda para desarrollos ágiles [42]	Los mismos que MariaDB, además de una arquitectura 32/64 bit y protocolo de red TCP/IP implementado en el servidor [43]
<b>PostgreSQL</b>	Potente sistema de base de datos relacional, de código abierto, diseñado para soportar un gran volumen de información (aunque es bueno para pequeñas bases de datos). Al ser de código abierto, no hay tarifas de licencia y es de uso totalmente gratuito. Tiene una instalación ilimitada y con gran escalabilidad [44]	1 GHz de procesador, 1 GB de memoria RAM y 512 MB de espacio en disco duro [45]

DBMS	Características	Requerimientos mínimos
<b>Firebird</b>	Base de datos relacional que ofrece muchas características estándar ANSI SQL. Excelente concurrencia, alto rendimiento y un potente soporte de idiomas para procedimientos almacenados y disparadores. Carece de tablas temporales e integración con otros sistemas de bases de datos [45]	Es necesario contar con las librerías glibc 2.2.5 y libstdc++ 5.0 instaladas y para el caso de RedHat 8.0, Mandrake 9.0 o SuSE 8.0 se requiere un procesador Pentium y 64MB de memoria RAM

Tabla 1.1: DBMS de código abierto

Después de analizar estas opciones, se determina que el DBMS óptimo para la creación del sistema será PostgreSQL debido a que es un sistema multiplataforma, que permite instalaciones a gran escala, capaz de ajustarse al número de CPU y a la cantidad de memoria disponible en el servidor que hospeda la base de datos de forma eficiente, por lo que puede procesar una gran cantidad de peticiones simultáneas -y de forma correcta- a la información almacenada. Adicionalmente, cuenta con pgAdmin, herramienta con la que se puede administrar las bases de datos, de manera fácil e intuitiva; e implementa el estándar SQL, resultando sencillo la realización de consultas y la inclusión de scripts de otros DBMS [46].

### 1.3.3. Frameworks de desarrollo

Un framework constituye una serie de herramientas y bibliotecas de código ya escrito, que facilitan a los desarrolladores las tareas cotidianas o de uso común para evitar ejecutarlas desde cero. Estos frameworks dan un marco de trabajo para programar en un lenguaje y, por su parte, las bibliotecas se encargan de solucionar problemas haciendo el código más legible [47]. Para el caso de BackEnd, los frameworks que cobran mayor relevancia para este proyecto se muestran en la siguiente tabla 1.2:

Framework	Características	Requerimientos mínimos
<b>Spring</b>	Estructura popular empresarial de código abierto que permite crear aplicaciones independientes que se ejecutan en la máquina virtual Java (JVM) [48]. Este framework define módulos tanto para el FrontEnd como para el BackEnd, seguridad, cloud, entre otros. <i>Spring MVC</i> (Model-View-Controller) cuenta con soporte para programación asincrónica y reactiva (no es posible para serverless, siendo ideal para microservicios y aplicaciones de red distribuida) [49]	Se requiere alguna de las siguientes soluciones: -Apache Tomcat $\geq 9.0$ -Jetty $\geq 9.4$ -Undertow $\geq 2.0$
<b>Django</b>	Sigue el patrón arquitectónico de <i>MTV</i> ( <i>Model-Template-View</i> ) renderizado del lado del servidor, con soporte adicional para programación reactiva asíncrona [49]. Extensa documentación, que a veces puede llegar a ser confusa y dispersa [50]	Python y Apache
<b>Laravel</b>	Framework web <i>MVC</i> ( <i>Model-View-Controller</i> ) renderizado en el lado del servidor, con soporte adicional para la programación asincrónica [49]. La instalación se hace mediante Composer y está pensado para las últimas versiones de PHP [51]	Es necesario contar con PHP $\geq 7.3$ y las siguientes extensiones de este lenguaje de programación: BCMath, CType, Fileinfo, JSON, Mbstring, OpenSSL, PDO, Tokenizer y XML
<b>Flask</b>	Framework Micro Web que utiliza Jinja2 como motor de plantilla y Werkzeug para el manejo de HTTP. Ideal para el desarrollo de aplicaciones empresariales a baja escala y no sigue el marco “baterías incluidas”, es decir, no cuenta con librerías de paquetes preinstaladas [49]	Python $\geq 3.6$
<b>Ruby on Rails</b>	Framework web que se centra en la convención sobre la configuración, compatible con la arquitectura MVC junto con la programación reactiva asíncrona. Introducción de conceptos nuevos, como DRY (siglas en inglés que se refiere a no repetir código ya creado) y Patrón de registro activo (modelado de tablas/vistas de base de datos en términos de una clase) [49]	Ruby $\geq 2.5$ , SQLite3, Node.js y Yarn

Tabla 1.2: Frameworks de BackEnd

Después de realizar el análisis de estos frameworks, se decide que para la parte de BackEnd se utilizará Django debido a que cuenta con un sistema de autenticación de usuarios (vital para el software a la medida a crear), incorpora una amplia variedad de paquetes de bibliotecas preconfiguradas, incluye opciones de protección para las aplicaciones web, proporciona una estructura de código autogenerado, además de contar con un panel de administración para bases de datos [50]. Y si bien es cierto algunos otros framework pueden ofrecer lo mismo, como lo es el caso de Spring, Django -framework de Python- posee además una comunidad extensa de desarrolladores y su curva de aprendizaje es menor en comparación con los frameworks mencionados.

Para el caso del FrontEnd, en la siguiente tabla 1.3 se muestran los frameworks más utilizados en el mercado:

Framework	Características	Requerimientos mínimos
<b>React</b>	Biblioteca de JavaScript de código abierto para crear componentes de interfaz de usuario. Se puede utilizar como base para desarrollar aplicaciones móviles o de una sola página. Sin embargo, solo se ocupa de la gestión del estado y la presentación de ese estado al DOM (Document Object Model). Esto significa que la creación de aplicaciones con este framework requeriría bibliotecas adicionales para el enrutamiento y cierta funcionalidad del lado del cliente [52]	React 18 es compatible con todos los navegadores modernos como Edge, Firefox, Chrome y Safari. Es necesario contar con Node JS versión 12 o superior y NPM versión 6 o superior [53]
<b>Angular</b>	Basado en TypeScript, con una función única de enlace de datos bidireccional, con lo que existe una sincronización en tiempo real entre la vista y el modelo, es decir, cuando se realiza algún cambio en el modelo, se refleja en la vista instantáneamente y viceversa. Además de ser utilizado en aplicaciones web o móviles, también se puede utilizar para desarrollar aplicaciones web progresivas y de varias páginas [52]	Puede trabajar con servidores de Apache, nginx, Ruby o IIS que tengan Node JS y NPM instalado [54]

Framework	Características	Requerimientos mínimos
<b>VueJS</b>	Framework de JavaScript y de código abierto, diseñado para optimizar el rendimiento de las aplicaciones y abordar distintas complejidades. Utiliza el modelo-vista-vista-modelo (MVVM) para desarrollar aplicaciones e interfaces de usuario de una sola página. Ayuda a los usuarios con múltiples tareas y puede manejar procesos dinámicos y simples con facilidad, incluidas aplicaciones web y móviles, así como aplicaciones web progresivas [52].	Compatible con las siguientes versiones mínimas de cada explorador web: Chrome $\geq$ 87, Firefox $\geq$ 78, Safari $\geq$ 13 y Edge $\geq$ 88 [55]
<b>Bootstrap</b>	Framework CSS de código abierto que combina CSS y JavaScript para estilizar los elementos de una página HTML. Proporciona interactividad en la página, por lo que ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de página, barras de progreso, entre otros. Su principal objetivo es permitir la construcción de sitios web responsivos para dispositivos móviles; es decir, que los sitios web creados estén diseñados para funcionar en computadoras de escritorio, laptops, tablets y smartphones, de una manera muy simple y organizada [56].	Sólo es necesario obtener el compilado de CSS y JavaScript, el código fuente o con algún manejador de paquetes que se cuente en el servidor para que pueda funcionar correctamente [57]

Tabla 1.3: Frameworks de FrontEnd

Después de analizar la información contenida en la tabla anterior, se decide que para la parte de FrontEnd se utilizará Bootstrap por su practicidad de integración de ambas tecnologías y su forma sencilla de implementación al estar directamente controlado desde el BackEnd. Adicionalmente se utilizará el componente del motor de plantillas que viene integrado en el framework de BackEnd desde su versión 1.8 (DTL, por sus siglas en inglés). Este sistema de plantillas que viene por defecto en Django es una alternativa a Jinja2. Con el propósito de que el sitio web sea dinámico con la información dada por el usuario final, se emplearán en conjunto Javascript, JQuery y AJAX para realizar peticiones asíncronas al servidor y mejorar los tiempos de respuesta del sitio web. Finalmente, para enriquecer la visualización de ciertos componentes de Bootstrap, se integrarán estilos personalizados con CSS3 y HTML5.



## 1.4. Patrones de diseño

Cuando se llevan a cabo desarrollos de software nos encontramos con una serie de problemáticas comunes a los mismos; por ello, resulta de suma importancia hablar de los patrones de diseño, los cuales constituyen la base para la búsqueda de soluciones a estos problemas y para otros ámbitos referentes al diseño de interacción o interfaces de usuario. Así, un patrón de diseño resulta ser una solución a un problema de diseño, pero para que una solución sea considerada como un patrón debe poseer ciertas características. Una de ellas es la comprobación de su efectividad resolviendo problemas similares en ocasiones anteriores. Otra, es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias [58].

Los patrones de diseño pretenden [58]:

- ◆ Proporcionar catálogos de elementos reusables en el diseño de sistemas software
- ◆ Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente
- ◆ Formalizar un vocabulario común entre diseñadores
- ◆ Estandarizar el modo en que se realiza el diseño
- ◆ Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente

Es importante señalar que no es obligatorio utilizar los patrones: solo es aconsejable en el caso de tener el mismo problema -o similar- y que éste se solucione con el propio patrón, tomando siempre en cuenta que en un caso particular puede no ser aplicable: abusar o forzar el uso de los patrones puede ser un error [58].

Uno de los patrones de diseño web más utilizados es MVC (Model-View-Controller). Este patrón de diseño se refiere a la separación de la interfaz de usuario (View) de la parte de datos y de negocio (Model), la cual está intermediada por un controlador (Controller). El Controlador realiza las validaciones, limpieza y verificación de datos, manejo de errores, entre otras acciones, permitiendo que los datos que se mandan desde la Vista lleguen de forma más ordenada y limpia a su procesamiento en el BackEnd, que es donde se encuentra el Modelo que accede directamente a la base de datos [59].

Para el sistema a desarrollar, Django implementa el patrón de diseño **MTV** (Model-Template-View), el cual es una interpretación del MVC en el que la vista (*View*) describe qué datos serán mostrados al usuario; el controlador es el propio framework y la presentación de los datos se muestra en la plantilla (*Template*). En cuestiones de seguridad, se utilizará el patrón **Decorator** que añade dinámicamente comportamientos que se han probado para permitir que sólo roles autorizados -con sus permisos relacionados a cada usuario- puedan ingresar a una vista. Tanto este patrón como el de MVT serán los que más se utilizarán en el desarrollo

del sistema a crear para reutilizar funcionalidades creadas por este framework y centrarse en la implementación de la lógica de negocio.

Para el caso del FrontEnd, se seguirá el patrón de diseño responsivo para asegurar la correcta visualización de la información. Sin importar el dispositivo en el que se consulte el sitio web, se redimensiona y se colocan los elementos del propio sitio de forma que se adapten al tamaño de cada dispositivo, trayendo consigo una mejor experiencia de usuario [60].

## 1.5. Ciclo de Vida de Desarrollo de Software

Siguiendo en el mismo tenor de las diversas vicisitudes a las que nos enfrentamos en un desarrollo de software, resulta fundamental señalar que existe un proceso sistemático para la construcción de software que también nos facilita el desarrollo de sistemas: éste es el llamado SDLC (Ciclo de Vida del Desarrollo de Software, por sus siglas en inglés).

El SDLC sirve de soporte para que los líderes de un proyecto puedan planificar el proceso de diseño y la puesta en marcha de cualquier sistema de información que cumpla los requisitos del usuario. Además, sirve de esquema para concretar los tiempos de desarrollo y la inversión del presupuesto. La razón de un SDLC es operar como base sólida en la construcción de un sistema efectivo. En muchos aspectos, el SDLC actuará como red de seguridad, restringiendo el desarrollo, para asegurar que satisface las necesidades del cliente sin sobrepasar el presupuesto o los tiempos acordados [61].

El proceso de SDLC se desglosa en pasos progresivos permitiendo, al mismo tiempo, la flexibilidad necesaria para hacer frente a incidencias y circunstancias imprevisibles, logrando afrontar todas las situaciones que se presenten, a la par de seguir satisfaciendo cualquier requerimiento solicitado por el cliente [62].

El SDLC es el término más general para los métodos de desarrollo de software. Los beneficios de seguir formalmente un método del desarrollo de software en la forma de este ciclo de vida, son [63]:

- ⊙ Instaurar un vocabulario común en cada paso
- ⊙ Definir canales de comunicación entre el equipo de desarrollo y los usuarios finales
- ⊙ Establecer, de manera clara, roles y responsabilidades en los desarrolladores, diseñadores, analistas y administradores de proyecto
- ⊙ Precisar claramente entradas y salidas de un paso para el consecuente
- ⊙ Delimitar si un paso está realmente completado

La importancia del SDLC en el desarrollo de sistemas de información radica en las siguientes razones [61]:

- ⊗ Ofrecer una base en la planeación de proyecto con tiempos y estimaciones
- ⊗ Proveer un marco de trabajo para un conjunto estándar de actividades y entregables
- ⊗ Representar un mecanismo para el seguimiento y control de un proyecto
- ⊗ Incrementar la visibilidad de la planeación del proyecto para todos los usuarios involucrados en el proceso de desarrollo
- ⊗ Aumentar la velocidad de desarrollo sin perder calidad
- ⊗ Mejorar las relaciones con el cliente y usuarios finales
- ⊗ Ayudar para reducir el riesgo del proyecto así como en su manejo

SDLC consiste en las siguientes siete fases [62]:

1. **Planificación:** Se define la necesidad, requisito u oportunidad que llevará a la creación de una propuesta para un sistema que llene ese vacío. Esta fase del ciclo incluye la planificación de los documentos necesarios para gestionar el proyecto. Es aquí donde se identificarán y proporcionarán los recursos necesarios para una solución efectiva dibujando el alcance exacto y los límites definidos del proyecto.
2. **Requerimientos (*Desarrollo Conceptual del Sistema*):** Aterriza los temas y el contenido identificado en la fase anterior. Contempla las necesidades del usuario del sistema y crea la documentación necesaria alrededor de estos requisitos.
3. **Diseño y prototipado:** Esta fase del ciclo toma los requisitos identificados en el paso anterior y los utiliza como base para elaborar un detallado documento del diseño del sistema. Éste, está destinado a analizar profundamente la función que el sistema debe llevar a cabo y a delinear la mejor manera de desarrollarlo.
4. **Desarrollo de Software:** Comprende el trabajo real de convertir el diseño en un sistema de software. El entorno de desarrollo ha de ser instalado: crear y probar la(s) base(s) de dato(s), así como codificar, compilar y mejorar los códigos fuente del sistema es parte de ello.
5. **Pruebas:** Tras completar la fase anterior, el sistema tiene que pasar por una serie de validaciones en torno a la calidad del software desarrollado. El resultado obtenido en estas pruebas se plasmará en un informe que dé cuenta de ello.
6. **Implementación:** El trabajo hecho hasta llegar a esta fase debería llevar, de manera natural, a la implementación del sistema al entorno del mundo real. Cualquier problema identificado en la fase inmediatamente anterior, debió ya haber sido resuelto.
7. **Operación y mantenimiento:** Consiste en compilar los pasos y procesos necesarios para mantener y operar el sistema de manera continua. Esto requiere una revisión del sistema en tiempo real. Tras seguir estos pasos, el sistema estará funcionando. Entonces puede compilarse un documento, detallando el proceso y cualquier observación necesaria mientras se pone el sistema en operación.

Las fases descritas con anterioridad corresponden a un resumen básico del SDLC, aunque debe decirse con toda claridad que, dependiendo del tamaño y complejidad de cada proyecto, fases individuales pueden combinarse, incluso, solaparse.

A continuación, se comenzará a desarrollar el presente software, partiendo de un análisis que englobará las dos primeras fases del SDLC: la planificación y la especificación de los requerimientos.

# Capítulo 2

## Análisis

En este capítulo se abordarán las primeras dos fases del SLDC comenzando con el proceso de planificación que se siguió para el desarrollo del presente proyecto, en donde la categorización de los tickets y la definición de los roles constituyeron piezas fundamentales para la instrumentación de la documentación del Sistema de Gestión de Tickets a implementar, redundando en la *especificación de los requerimientos (ERS)*. Cabe mencionar que para la entrega de sistemas, la Unidad de Cómputo de la DICyG requiere contar con la siguiente documentación: carta descriptiva, manual de usuario, manual técnico de desarrollo y manual administrativo. Además, para formalizar el inicio y finalización del proyecto, se realizó la entrega del acta constitutiva del proyecto y, desde luego, del acta de cierre del proyecto.

### 2.1. Planificación

Para esta etapa del SLDC se llevó a cabo una segunda reunión con el cliente, la cual consta en el apéndice A. En esta reunión se realizó una breve exposición del proceso actual de solicitud de soporte técnico por parte de la UC; este proceso es manual, donde los usuarios solicitantes (personal de la DICyG) hablan a ésta para la resolución de la problemática que se les presenta en torno a sus equipos de cómputo y llenan una solicitud para dar inicio a la atención de su ticket. Acto seguido, la líder técnico en gestión de recursos humanos asigna a un técnico del staff que atenderá el ticket, quien será el que realice las actividades necesarias para resolver el problema. A este respecto se puntualiza que, conforme a la tarea a realizar, la propia UC asigna un puntaje para llevar un seguimiento del desempeño de cada colaborador. Una vez que el problema fue resuelto, se informa de ello al usuario solicitante para verificar si es correcta la atención recibida y se cierra el ticket.

Derivada de la atención de soporte que lleva a cabo la UC, se opta por categorizar los tickets en tres tipos para la implementación del sistema, bajo el siguiente argumento:

1. ***Ticket de Soporte Técnico***: Se realizará el soporte al dispositivo inventariado por la UNAM que se reporta en el levantamiento de la solicitud. El cliente requiere registrar los cambios realizados en el equipo de cómputo para mantener un histórico para futuras atenciones ya que actualmente no se cuenta con ningún registro.
2. ***Ticket de Capacitación***: Se capacitará al trabajador de la DICyG con las tecnologías que requiera, o bien, recibirá asesoría sobre las herramientas que le permitan el desempeño de sus actividades diarias.
3. ***Ticket de Mantenimiento Programado***: Se realizará la atención a los equipos de cómputo asociados a laboratorios de la DICyG en una fecha determinada; es levantado por la Jefatura de la Unidad de Cómputo de esta división. Por cada laboratorio atendido, el cliente requiere que se reporten las tareas realizadas en cada equipo durante el periodo del mantenimiento, así como la verificación del correcto funcionamiento del software instalado.

Cabe destacar que algunos soportes que realiza la UC requieren ser “compartidos” en el caso de que el técnico que en un inicio se asigne no pueda resolver el problema y sea necesario que la atención del ticket continúe con un técnico que le dé su total resolución.

En el caso del *Ticket de Mantenimiento Programado*, donde se generan tickets de soporte técnico por cada equipo de cómputo del laboratorio en que se realiza el mantenimiento, se atienden en conjunto y se reportan como uno solo. En este tipo de ticket destacan cuatro roles: quien asigna, quien realiza, quien supervisa y quien verifica que el mantenimiento al laboratorio se realizó de manera correcta.

Por otra parte, en la introducción ya hemos puntualizado que para dar seguimiento a la atención al ticket, actualmente se abren grupos de WhatsApp en los que se recibe información adicional del usuario solicitante acerca del problema reportado, y que esta comunicación no sólo se realiza en los grupos sino directamente a cada miembro del staff, lo que se traduce en un menoscabo de la información al no llegar directamente al técnico que está atendiendo la solicitud de soporte técnico.

Precisamente, el Sistema de Gestión de Tickets (SGT) a implementar evitará esta serie de problemáticas que se presentan actualmente en la UC, toda vez que al encargarse del levantamiento automático de los tickets de soporte técnico se conservará un histórico de los cambios realizados en los dispositivos inventariados y se mantendrá un foro de comunicación que evitará esta “merma” de información en el seguimiento de la atención de los tickets al establecer una interacción directa entre el usuario solicitante y el técnico o miembro del staff encargado especialmente en la atención de su ticket.

Una vez sentadas las bases del actual proceso manual de soporte técnico, se llevaron a cabo

reuniones posteriores para concretar el levantamiento de requerimientos y así poder efectuar el desarrollo conceptual del SGT, las cuales también constan en el referido apéndice A.

## 2.2. Desarrollo conceptual del sistema

Para poder tener un control de lo que puede realizar y visualizar cada uno de sus usuarios, se utilizará la siguiente definición de roles en el SGT:

- ★ **Administrador Ejecutivo** (*Jefatura*): Tiene el control total del sistema, el acceso a todas las secciones y es el encargado de levantar el ticket de mantenimiento programado.
- ★ **Administrador del Sistema**: Es el responsable de la administración de la información dentro del sistema.
- ★ **Líder Técnico**: Asigna a un Técnico Junior o Máster para atender un ticket.
- ★ **Supervisor**: Verifica técnicamente si el ticket fue atendido de forma correcta.
- ★ **Encargado de Laboratorio**: Realiza la verificación administrativa del mantenimiento a equipos de cómputo en laboratorios a su cargo.
- ★ **Técnico Junior**: Atiende sus tickets asignados. Requiere que un Supervisor verifique si el problema reportado fue resuelto exitosamente.
- ★ **Técnico Máster**: Atiende sus tickets asignados sin supervisión alguna.
- ★ **Solicitante**: Es quien levanta un ticket de soporte técnico o de capacitación y el encargado de validar que su problema fue resuelto.

Estos roles no son excluyentes, ya que un usuario del sistema puede tener uno o más roles asignados. En la siguiente figura 2.1 se muestra la jerarquía que tienen estos roles en el sistema para poder acceder a los distintos módulos que lo conforman.

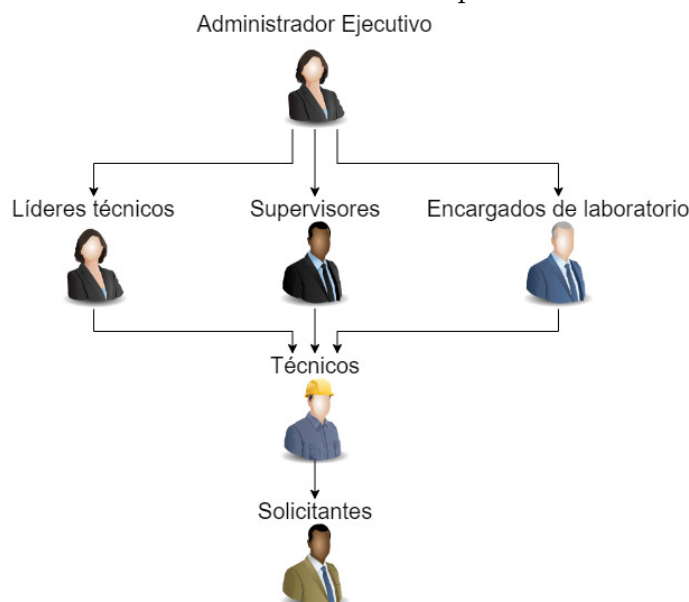


Figura 2.1: Jerarquía de Roles en el Sistema de Gestión de Tickets

Como puede observarse, el Administrador Ejecutivo tendrá acceso a todos los módulos, mientras que el acceso de los usuarios Solicitantes está limitado a los módulos del sistema donde sólo podrán consultar y conocer el avance de las solicitudes que levanten.

Determinados los roles, es importante ahora precisar el ciclo de vida de un ticket; a saber, Levantamiento, Asignación, Atención, Validación y Cierre; fases que nos indican el estatus en que se encuentra la solicitud. Desde luego, cada tipo de ticket cuenta con su propio flujo de desarrollo. Así, al Ticket de Soporte Técnico le corresponde el siguiente flujo:

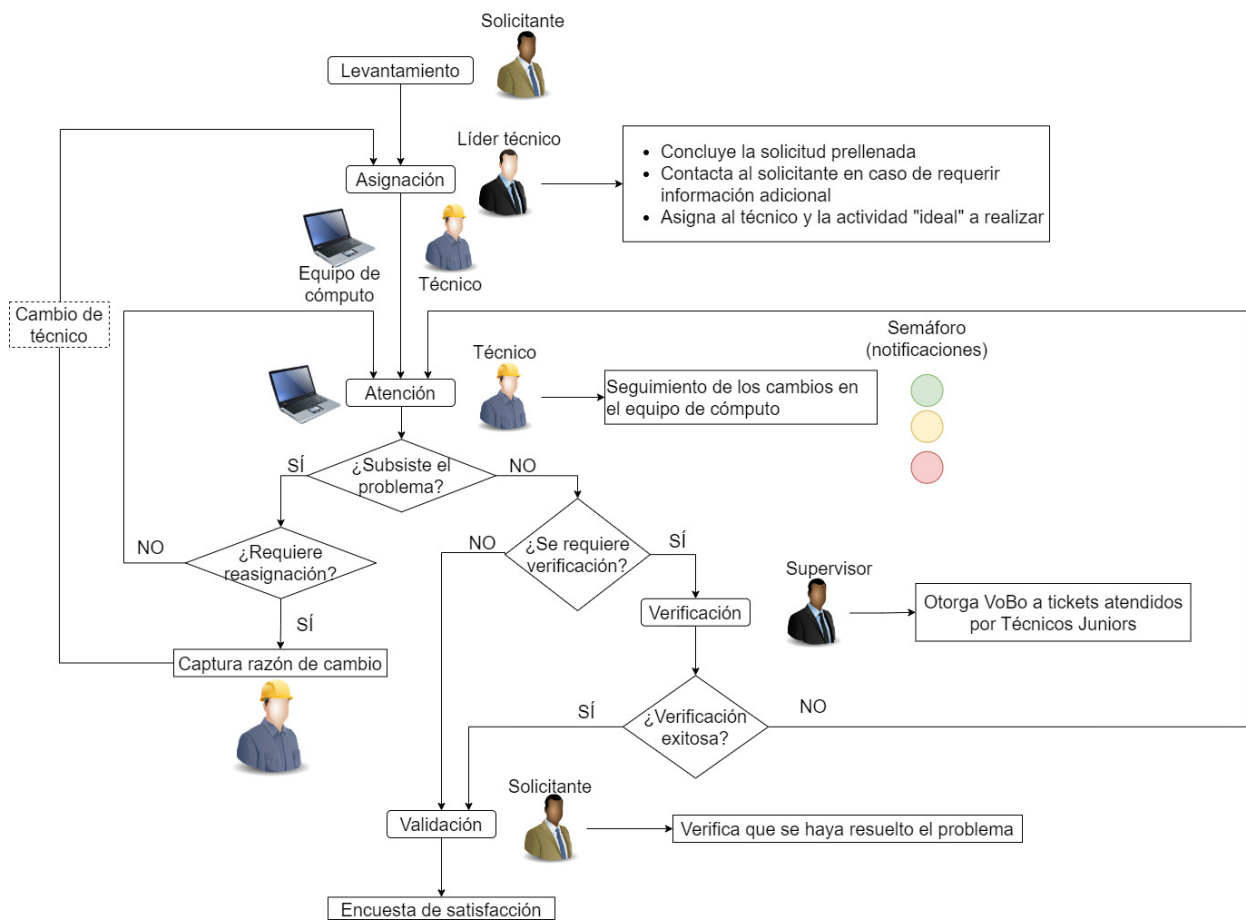


Figura 2.2: Diagrama de flujo para el Ticket de Soporte Técnico

Como se muestra en la anterior figura 2.2, el usuario Solicitante realiza el levantamiento de su solicitud indicando el problema que presenta su equipo de cómputo. Un Líder Técnico realiza la asignación del técnico que atenderá el ticket, poniéndose en contacto con el Solicitante en caso de ser necesario para completar el llenado de la solicitud, señalando la actividad principal a realizar y un tiempo aproximado para su resolución.



En la parte de atención, el Técnico al que le fue asignado el ticket registra los cambios realizados en el equipo de cómputo -si el problema persiste, puede solicitar un cambio de técnico (reasignación del ticket)-. Si el problema es atendido por un Técnico Junior, es necesario que un Supervisor realice la verificación de la atención de su ticket e indique si el problema fue resuelto o no. Una vez verificado (si es necesario), el Solicitante valida que el problema fue resuelto y para cerrar el ticket se muestra una encuesta de satisfacción para la mejora continua del proceso en la atención a usuarios por parte del staff de la Unidad de Cómputo.

Ahora bien, en la siguiente figura 2.3 se muestra el flujo del Ticket de Capacitación, el cual es similar al del anterior, con la salvedad de que en la atención no se registran los cambios en el equipo de cómputo: sólo hay un seguimiento con el usuario Solicitante:

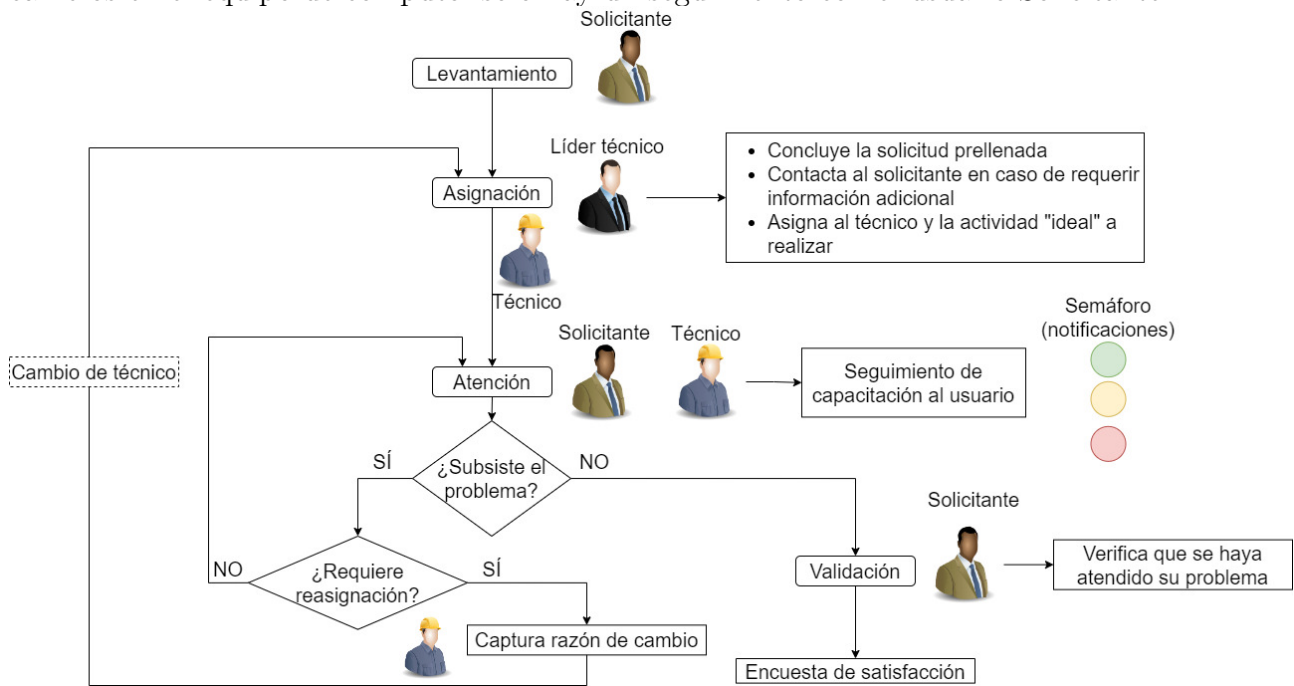


Figura 2.3: Diagrama de flujo para el Ticket de Capacitación

Finalmente, el *Ticket de Mantenimiento Programado*, mostrado en la figura 2.4, está asociado a los laboratorios y es levantado por el Administrador Ejecutivo, quien sólo seleccionará el nombre del laboratorio y la fecha programada para realizar el mantenimiento. Ingresada esta información, se generarán los tickets de soporte técnico asociados a este mantenimiento por cada dispositivo inventariado que se tenga registrado en ese laboratorio, como se muestra en la figura 2.5. Estos tickets generados automáticamente por el SGT son muy similares al *Ticket de Soporte Técnico*, lo único que varía es la obligatoriedad de pasar por dos verificaciones: una, de la parte técnica llevada a cabo por un Supervisor; otra, de la parte administrativa a

cargo del encargado del laboratorio en el cual se realiza el mantenimiento. Adicionalmente, se generará un reporte de lo realizado en dicho mantenimiento para llevar un control de todas las actividades desempeñadas.

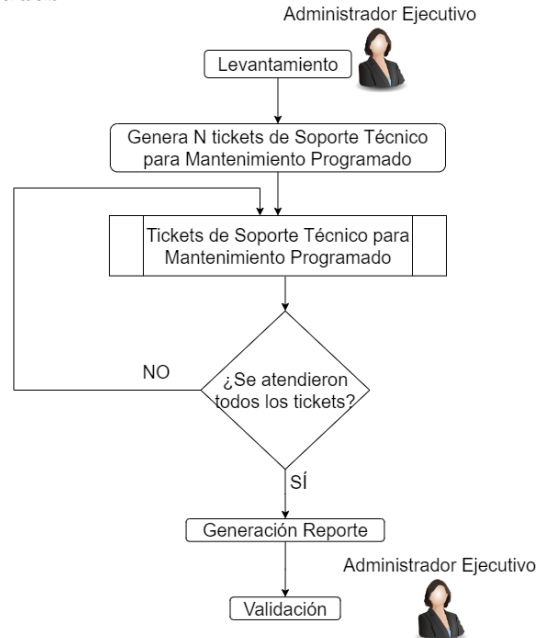


Figura 2.4: Diagrama de flujo para el Ticket de Mantenimiento Programado

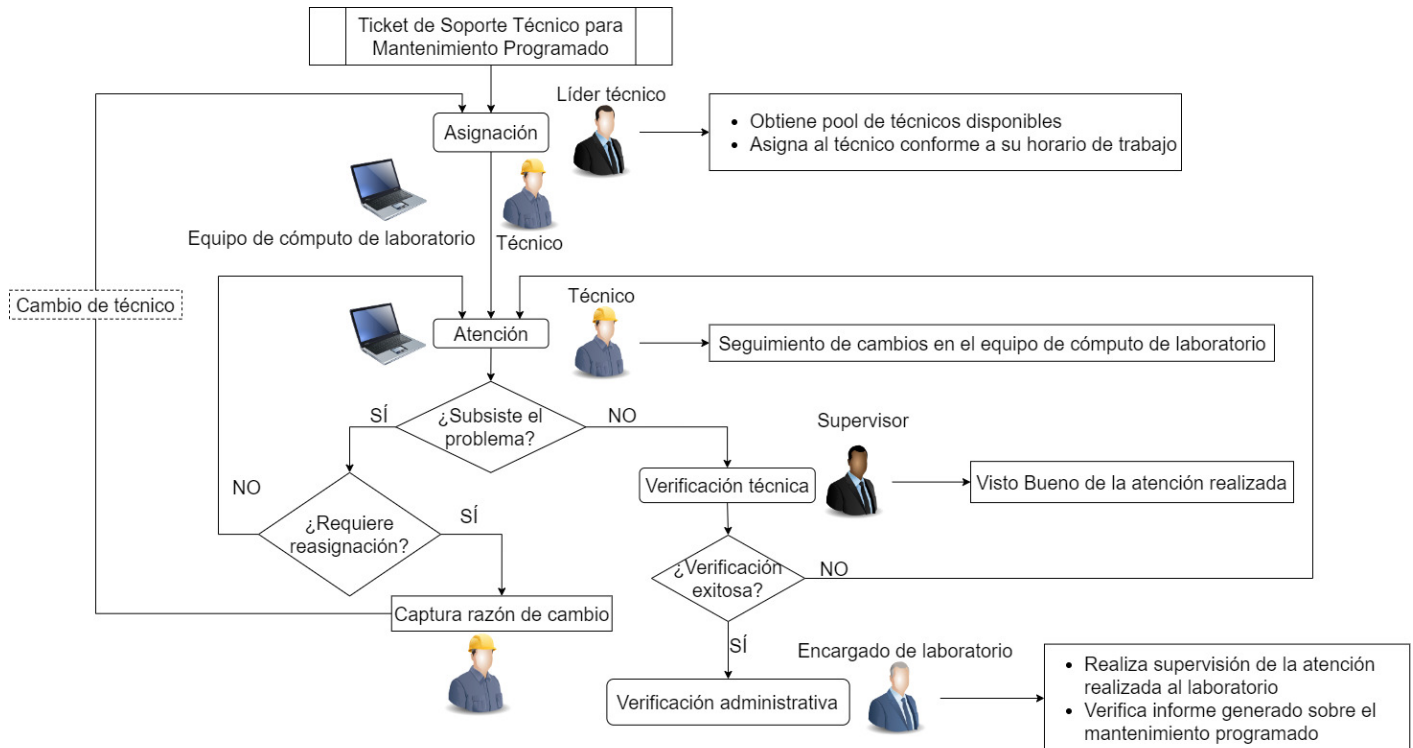


Figura 2.5: Ticket de Soporte Técnico asociado al Mantenimiento Programado

Acotar que los diagramas de flujo que han sido desplegados fueron mostrados al cliente en una tercera reunión (apéndice A) para su validación y visto bueno a fin de iniciar el desarrollo del proyecto. Autorizada que fue la propuesta, se realizó la formalización del proyecto con la creación del acta constitutiva del proyecto y carta descriptiva del mismo, documentos que obran en el apéndice B, definiéndose con ellos los alcances del SGT.

Una vez entregada esta documentación, es posible realizar la especificación de Requerimientos de Software (ERS) definida en el estándar ISO/IEC/IEEE 29148, teniendo como artefactos estas reuniones con el cliente, información dada por él a través de formatos, listados de actividades, planeaciones de mantenimientos realizados con anterioridad y los diagramas de flujo creados en esta etapa de *Planeación* para automatizar el proceso manual que se realiza actualmente en la UC.

## 2.3. Especificación de requerimientos de software

Retomando el estándar ISO/IEC/IEEE 29148 que define la construcción de buenos requerimientos de software, proporcionando atributos y características de éstos y que, además, analiza la aplicación iterativamente y recursivamente de los procesos referentes a la ingeniería de requerimientos a lo largo del SLDC [64], a continuación se muestra la ERS para el Sistema de Gestión de Tickets:

### 2.3.1. Propósito

La especificación de requerimientos tiene como objetivo informar al cliente sobre la creación del Sistema de Gestión de Tickets (SGT), los distintos módulos que lo conformarán, así como la interacción de los distintos roles dentro del sistema.

### 2.3.2. Alcance

El SGT trascenderá como una central de administración del ciclo de vida de los tickets que atiende la UC. Proveerá un histórico de los cambios realizados en cada dispositivo que se encuentre registrado en el Sistema de Inventarios y, a su vez, de las actividades realizadas por cada uno de sus colaboradores de la UC. Dicha información permitirá la generación de reportes personalizados que concentrará lo realizado en la Unidad de Cómputo.

### 2.3.3. Perspectiva del producto

El SGT representará un producto pionero al interconectarse con otro sistema de la Unidad de Cómputo de la DICyG: el Sistema de Inventarios, donde se almacena el registro de

todos los equipos electrónicos que se tienen a cargo. Esta interconexión es de vital importancia dado que tanto el histórico como el levantamiento del ticket de soporte técnico en el SGT dependen del número de inventario para iniciar el proceso; por lo que es necesaria una conexión local de manera directa con la base de datos del Sistema de Inventarios con la creación de un usuario de sólo lectura en su DBMS y empleando una comunicación bidireccional entre los dos servidores en los que se encuentran alojados estos sistemas.

Esta capacidad de comunicación entre distintos DBMS es la que precisamente hará asegurable que esta información pueda ser compartida de forma segura; aún más, que la misma pueda ser accesible inclusive en distintos entornos productivos. Desde luego que la factibilidad de esta comunicación requerirá de cuestiones técnicas de hardware y de software, pero también de comunicación entre los servidores a emplear (cuestiones que se detallarán en los siguientes capítulos del presente trabajo), y será lo que en conjunto logre traducirse en una gran integridad de los datos, donde la accesibilidad a ellos se convierta en una realidad inmersa en un entorno seguro del sitio web.

El SGT será un sistema web provisto de una interfaz intuitiva donde el usuario podrá localizar oportunamente los botones de interacción con el sistema que le permitan realizar las acciones asociadas a cada rol que se encuentre disponible. Esta comunicación entre el usuario-UC se hará directamente desde el portal web a través de foros de comunicación para cada ticket solicitado. Durante todo el proceso, se lanzarán una serie de notificaciones por correo electrónico a los roles involucrados, con lo cual se reducirán los tiempos de atención al estar siendo informado en todo momento de las acciones tomadas para la atención del ticket.

En aras de seguir proveyendo la inmediatez de atención a usuarios, para los tickets que no requieren la intervención de un técnico, se creará un foro de asesoría tecnológica que contendrá una serie de tutoriales (que dicho sea de paso, estarán en constante actualización por parte del staff de la UC) que el usuario podrá seguir cuando así lo decida, mismos que le permitirán resolver problemas comunes y sencillos por sí solo, antes de verse en la necesidad de levantar una solicitud de soporte.

Para poder entrar a cada una de las funcionalidades implementadas en el sistema es necesario contar con un correo electrónico activo y una contraseña (la cual se mandará por correo, una vez dados de alta en el SGT), así como con al menos un rol asignado para poder realizar distintas actividades dentro de los módulos. La gestión de los usuarios la realizarán miembros del staff de la UC para poder mantener un correcto control de los trabajadores que soliciten apoyo a esta unidad.

Una vez autenticado, el SGT mostrará un menú con cada uno de los módulos a los cuales es posible acceder dependiendo del rol(es) asignado(s) y la gestión de la cuenta, con un

cambio de contraseña si el usuario así lo desea. También, se contará con un restablecimiento de la misma desde el propio portal web de manera autogestionada a través de un token enviado al correo registrado para poder realizar el cambio.

### 2.3.4. Funciones del producto

Las funcionalidades principales del SGT se describen en 7 grandes rubros:

1. ***Inicio de Sesión***: Este módulo realiza la gestión de autenticación y autorización del usuario dentro del SGT, así como el control de acceso a través de su contraseña.
2. ***Gestión de Catálogos***: Los usuarios con el rol Administrador del Sistema son los encargados de realizar el alta, activación, actualización de datos o baja de los usuarios registrados en el SGT; cambiar o agregar roles para acceder a las vistas asociadas a cada uno de éstos y administrar los listados mostrados a lo largo del sistema (actividades con su ponderación, software que se maneja, laboratorios registrados y áreas de la Unidad de Cómputo), también forman parte de las tareas encomendadas.
3. ***Levantamiento de Ticket***: En este módulo el usuario Solicitante levanta un ticket de soporte técnico o de capacitación, exponiendo la causa del mismo. Solo el Administrador Ejecutivo tiene habilitada la opción de levantar tickets de mantenimiento programado a laboratorios.
4. ***Asignación de Técnicos***: Los Líderes Técnicos concluyen el llenado de la solicitud correspondiente para comenzar a dar el soporte; asigna al Técnico que se encargará de la solicitud, estableciendo la actividad (ideal) a realizar y una fecha aproximada de término.
5. ***Atención y Seguimiento de Tickets***: Los Técnicos reportan en todo momento los cambios realizados al equipo de cómputo (en caso de ser un ticket de soporte); también agregan nuevas actividades realizadas (si es necesario) y mantiene comunicación con el Solicitante a través del foro correspondiente.
6. ***Foros***: Esta funcionalidad permite mantener una comunicación directa con el usuario Solicitante y el staff de la UC. En ellos es posible encontrar la solución a preguntas y respuestas frecuentes de soportes comunes a través de un tutorial que permita a los usuarios resolver sus propios problemas, sin la necesidad del levantamiento de un ticket.
7. ***Reportes***: Este módulo muestra el informe ejecutivo (graficación de lo hecho a lo largo de un semestre/año) y el listado largo de las actividades realizadas comprendidas en los tickets generados en el sistema.

### 2.3.5. Características de los usuarios

El SGT contempla dos grandes grupos de usuarios: el staff de la UC y el personal de la DICyG. El primer grupo se caracteriza por contar con un perfil más técnico y se encargarán de la gestión del SGT para realizar el correcto desarrollo del ciclo de vida de los distintos tipos de tickets, por lo que deben ser capaces de entender el flujo del proceso en cada módulo implementado en el sistema, comprometiéndose a mantenerlo actualizado y con una correcta integridad de los datos de todos los dispositivos que se encuentran registrados en el Sistema de Inventarios. En tanto, el segundo grupo lo constituye la plantilla de personal de la adscripción, caracterizándose particularmente por ser un solicitante de un servicio.

Ambos grupos de usuarios deberán contar con correo electrónico ya que será el medio por el cual el sistema realizará notificaciones para seguimiento y control de los cambios asociados a la atención de cada ticket.

### 2.3.6. Restricciones

Se deben buscar herramientas Open Source libres o de muy bajo costo para ser implementadas en el SGT. Es importante puntualizar que desde el SGT sólo se harán consultas de los dispositivos registrados en el Sistema de Inventarios, de tal manera que cualquier modificación que se requiera hacer en torno a la información registrada en los dispositivos electrónicos, se realizará directamente en este sistema de la UC.

### 2.3.7. Requerimientos funcionales

Estos requerimientos se refieren a necesidades que el SGT debe de resolver, como lo son:

- El SGT mantendrá un histórico de lo realizado en cada dispositivo a través de su número de inventario.
- El SGT llevará un control de las actividades de soporte técnico realizadas por cada uno de los técnicos al momento de atender los tickets.
- El SGT mostrará un reporte con todas las actividades realizadas por la UC en la atención de soportes técnicos a lo largo del semestre.
- El SGT deberá ser capaz de reasignar a un técnico para soportes difíciles que requieran el intercambio por otro técnico con más experiencia.
- El SGT permitirá que la atención de los tickets se pueda realizar en el momento o en alguna fecha posterior a la asignación del técnico que atenderá la solicitud.
- El SGT se comunicará con los usuarios con un correo electrónico de *no-reply* proporcionado por la UC.

- El SGT asociará una ponderación por cada actividad a realizar; también contará con un listado de tareas a realizar y software a revisar en los mantenimientos a laboratorios.
- El SGT permitirá a los usuarios Solicitantes, Supervisores y Encargados de Laboratorios -de acuerdo a las funciones de cada rol- verificar y/o validar que la atención recibida en el portal web sea la correcta y haya resuelto el problema reportado.
- El SGT permitirá el levantamiento del ticket de mantenimiento programado por cada laboratorio de la DICyG. Este levantamiento se realizará de forma automática para cada máquina registrada y, aunque se trate de varios tickets de soporte técnico, se considerarán como uno solo durante todo su ciclo de vida.
- El SGT permitirá la modificación de cualquier catálogo contenido en el sitio web.

### 2.3.8. Requerimientos no funcionales

Estos requerimientos son limitantes externas al SGT que deben tenerse en cuenta al momento de diseñar y desarrollar el sistema. Constituyen las restricciones impuestas a un sistema que definen sus atributos de calidad. Estos atributos de calidad son las características del sistema propiamente dicho que determina su calidad, en tanto que las restricciones son las limitaciones impuestas al sistema, donde se hayan incluidas el tiempo de respuesta, los recursos y el entorno. Este tipo de requerimientos son importantes porque ayudan a garantizar que el sistema satisfaga las necesidades del usuario [65]:

- El Sistema de Inventarios se encuentra en otro servidor y no está permitido realizarle cambios desde el SGT.
- Una vez puesto en producción, el SGT podrá recibir mantenimiento y modificaciones en cada uno de sus módulos, de manera independiente, a fin de no afectar la funcionalidad del sistema web.
- Homologar la comunicación del usuario a través de un sólo canal en el que el usuario Solicitante sólo se comunique con el staff de la UC y éste le pueda contestar directamente en el sitio web enviando una notificación de correo electrónico.
- El SGT debe contar con un certificado para el cifrado de los datos en la transmisión de la red.
- El SGT evitará problemas de inyección SQL y de código JavaScript.
- El SGT evitará acceso a vistas no autorizadas a través de roles para cada usuario autenticado.

- El SGT tendrá un tiempo de respuesta menor a 200 milisegundos por cada petición que realice un usuario al sitio web.
- El SGT estará disponible en todo momento desde cualquier dispositivo a través de una dirección URL proporcionada y gestionada por el cliente.
- El SGT seguirá funcionando a pesar de que uno o más módulos no estén disponibles (tolerancia a fallas).
- Una vez puesto en producción, se realizarán respaldos programados a la base de datos del SGT de manera periódica para resguardar toda la información registrada en este sistema.

Una vez levantados los requerimientos del SGT, se prosigue con el diseño del mismo, observando en todo momento lo establecido en la ERS y manteniendo un constante contacto con el cliente para la retroalimentación del SGT.



# Capítulo 3

## Diseño

Una vez establecida la ERS, es posible realizar un diseño del SGT a implementar siguiendo cada uno de los requerimientos solicitados por el cliente, por lo que en las siguientes líneas se hará una descripción de la arquitectura propuesta para el desarrollo del sistema.

### 3.1. Arquitectura propuesta del sistema

El SGT estará conformado por los siguientes componentes de software:

- **Django 3.2:** Framework de BackEnd que permitirá la realización de operaciones CRUD (creación, lectura, actualización y eliminación, por sus siglas en inglés) en la base de datos del SGT, así como las de sólo lectura a la base de datos del Sistema de Inventarios para la obtención de información de los dispositivos inventariados.
- **Bootstrap 4:** Framework de FrontEnd que mostrará las vistas dadas por Django de una manera más amigable y con una mejor presentación de la información contenida en el sistema.
- **PostgreSQL 13:** Base de datos en la que se almacenará toda la información generada por el SGT.
- **MySQL 5.5:** Base de datos del Sistema de Inventarios de la DICyG en operación, la cual, por tratarse de un sistema administrado por la UC, sólo tendrá permisos de lectura para poder consultar los dispositivos inventariados.
- **Apache 2.4:** Servidor instalado en la máquina virtual proporcionada por el cliente. Este software, en conjunto con el módulo WSGI (Web Services Gateway Interface), permite transformar el código de Python en peticiones HTTP (Protocolo de Transferencia de Hipertexto, por sus siglas en inglés).

### 3.1.1. Web Services Gateway Interface

El estándar de Python que se detalla en el PEP 3333 [66], es una especificación de interfaz para la comunicación entre aplicaciones web hechas en este lenguaje de programación y cualquier servidor. Describe cómo estas aplicaciones pueden trabajar en conjunto para procesar una solicitud y comunicarse a través del protocolo HTTP [67].

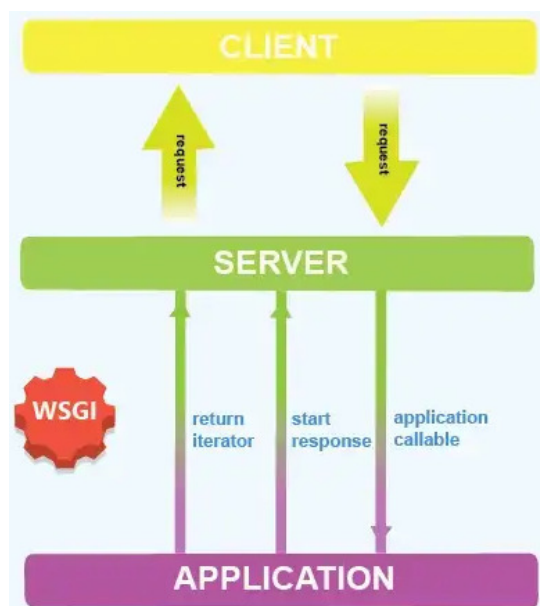


Figura 3.1: Proceso en la petición a un servidor WSGI<sup>7</sup>

En la figura 3.1 se muestra cómo se comunica un servidor web con una aplicación que utiliza la WSGI. Esta comunicación se da cuando un cliente (usuario) realiza una solicitud al sitio web, de tal manera que las aplicaciones escritas en la especificación WSGI esperan a la llamada del servidor web (sólo existen en tiempo de ejecución). Cuando son llamadas, se analiza y procesa la petición solicitada y, una vez ejecutada la petición, se devuelve el resultado al servidor web. Importante señalar que en el caso del servidor web sólo recibe las peticiones que el cliente solicita y responde con lo enviado por la aplicación web [68].

WSGI proporciona flexibilidad permitiendo que en el desarrollo del SGT se puedan reemplazar unos componentes por otros, ya que separa la elección del framework de desarrollo de la elección del servidor web. Aunado a ello, otra de las bondades de esta interfaz es su escalabilidad, toda vez que el servidor WSGI manipula el procesamiento de todas las solicitudes del servidor web y decide cómo comunicar cada una de estas peticiones a los procesos del framework de aplicación [69].

<sup>7</sup>Imagen recuperada de [Medium](#) el 14/03/2023

### 3.1.2. Patrón de diseño en Django

Django implementa el patrón MTV (Modelo/Plantilla/Vista, por sus siglas en inglés), el cual es una interpretación del patrón de diseño más común para el desarrollo web: MVC (Modelo-Vista-Controlador, por sus siglas en inglés). Esta interpretación separa los datos de la presentación y de la lógica de negocio, agrupando el código que gestiona cada parte de la petición HTTP en archivos separados de la siguiente manera [70–72]:

- **Mapeador de URLs:** Redirige las peticiones HTTP a la *vista* apropiada, basándose en la URL (Localizador de Recursos Uniforme, por sus siglas en inglés) de la petición; es decir, en la dirección del recurso único en Internet ingresado en el navegador web por el usuario. También se pueden utilizar patrones de cadenas o dígitos específicos que se añaden a la URL para pasarlos a la correspondiente función de visualización como datos entrantes.
- **Vista:** Función para la gestión de peticiones que recibe las solicitudes HTTP y devuelve respuestas en este mismo protocolo, describiendo cuáles datos serán presentados al usuario. Las vistas acceden a la información que necesitan para satisfacer las peticiones por medio de *modelos*, y delegan la renderización de la respuesta a las *plantillas*.
- **Modelo:** Objeto de Python que define la estructura de los datos de una aplicación y proporciona mecanismos para gestionar operaciones CRUD: añadir, consultar, modificar y borrar registros en la base de datos.
- **Plantilla:** Archivo de texto que define la lógica de la presentación de los datos con una estructura, como una página HTML (Lenguaje de Marcado de Hipertexto), con marcadores de posición que se utilizan para representar el contenido real que se mostrará al usuario, o bien diagrama cualquier otro tipo de archivo.

Como puede observarse en la estructura de este patrón, el controlador es el propio framework en Django, ya que se encarga de enviar la petición a la vista apropiada de acuerdo a la configuración de URL dada por el usuario [72].

En este framework de BackEnd, cada aplicación escrita consiste en un paquete de Python que incluye una colección de archivos de código fuente: modelos, vistas, plantillas y mapeador URL. Esta app (aplicación) puede ser utilizada en múltiples proyectos de Django ya que realiza una función única.

### 3.1.3. Especificaciones de la arquitectura del SGT

Para la creación del Sistema de Gestión de Tickets se implementará el patrón de diseño MTV de acuerdo al framework de BackEnd a utilizar. En la siguiente figura 3.2 se observa

gráficamente el comportamiento de este patrón en el momento que un usuario realiza una petición al sitio web:

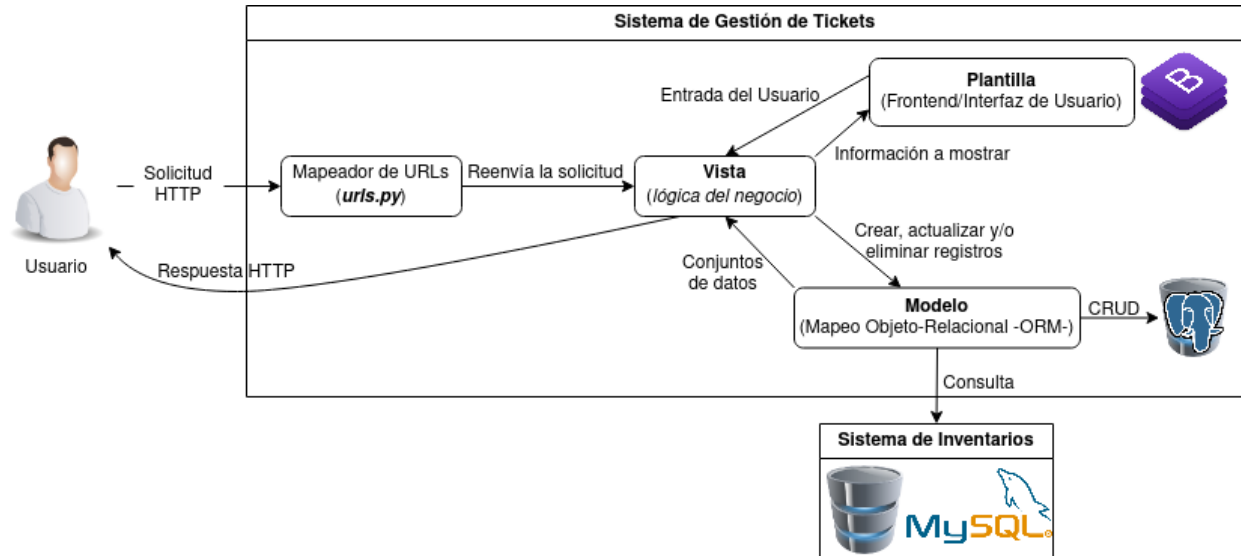


Figura 3.2: Arquitectura del Sistema de Gestión de Tickets

Como se muestra en el diagrama, un usuario entrará a la página web del SGT realizando una solicitud HTTP. Esta solicitud será recibida por el WSGI y enviada al *Mapeador de URLs*, que redirigirá a la *vista* apropiada de acuerdo a la URL ingresada.

En las *vistas* se tendrá todo el control de la lógica del negocio, ya que recibirá la información enviada por el usuario a través de las *plantillas* para ser procesadas en la base de datos mediante *modelos* que responderán con conjuntos de datos. La *vista* enviará esta información a la *plantilla* para su creación.

Las *plantillas* serán la interfaz de usuario del SGT. El framework de FrontEnd *Bootstrap* se utilizará para presentar la información recibida de las *vistas* de una manera responsiva, funcional y usable para cualquiera que ingrese al sistema. También se realizarán validaciones antes de enviar la información al servidor para mejorar la integridad del mismo y evitar que información incompleta o dañina entre al sistema.

Los *modelos* son el mapeo de un objeto de Python a las entidades del modelo relacional de las base de datos de ambos sistemas, quienes serán los encargados de conectarse y realizar las operaciones necesarias enviadas por las *vistas*. Recordar que, el Sistema de Inventarios ya existente, es un sistema externo al SGT que sólo permite operaciones de lectura.

Finalmente, una vez realizado todo el procesamiento de la solicitud en el SGT, al usuario le será enviada una respuesta HTTP correspondiente al recurso de su petición a través del WSGI.

Ahora bien, para realizar la modularización del sistema se construirá una aplicación por cada módulo especificado en la carta descriptiva del SGT (apéndice B). Estas aplicaciones estarán contenidas en un proyecto denominado ***sistema\_tickets***:

1. **usuarios**: Gestionará el alta, la edición y baja de los usuarios del sistema, así como el cambio y reinicio de contraseña para poder acceder al SGT.
2. **catalogos**: Contendrá todo el módulo de Gestión de Catálogos; en consecuencia, esta app realiza la administración de toda la información mostrada en el sistema.
3. **foros**: En esta app se gestionarán los foros de comunicación y de asesoría tecnológica para mantener al usuario informado de la atención de sus tickets.
4. **inventario**: Esta app se encargará del acceso y mapeo de la información del Sistema de Inventarios que se requiera consultar en el Sistema de Gestión de Tickets.
5. **soporte\_tecnico**: Llevará a cabo la gestión de las siguientes etapas del ciclo de vida de un ticket: Levantamiento, Asignación y Cierre.
6. **seguimiento\_atencion**: Llevará a cabo la gestión de las etapas faltantes de la app *soporte\_tecnico*; siendo éstas: Atención, Verificación y Validación.
7. **reportes**: En esta app se mostrarán los informes ejecutivos con graficación y listados largos de las actividades realizadas por la UC.

Esta modularización resulta pertinente para poder realizar las modificaciones que, en un momento determinado, se estimen necesarias en cada módulo del sistema sin que ello derive en afectaciones entre los mismos.

Una vez definida la arquitectura que seguirá la implementación del sistema, se muestran las historias de usuario con diagramas UML que ejemplifican de mejor manera lo que se desarrollará en el SGT.

## 3.2. Historias de Usuario

El Lenguaje de Modelado Unificado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software [73], de los cuales se crean representaciones simplificadas de componentes y procesos. En el caso del Sistema de Gestión de Tickets, este lenguaje se utiliza para dar una representación

visual de los casos de uso para cada rol que accede a él.

El SGT utilizará un acceso basado en el rol o roles con que cuente el usuario y respecto de los cuales se concederán determinados permisos para realizar ciertas funcionalidades en el sistema y poder ingresar a cada uno de los módulos creados. Preciso enfatizar que el manejo de roles permite garantizar los derechos de acceso a los servicios de un sistema.

### 3.2.1. Administrador Ejecutivo

Este usuario tiene acceso a todos los módulos del sistema, por lo que puede ingresar e interactuar en cada uno de ellos. En la figura 3.3 se muestra el diagrama UML para el usuario con el rol de Administrador Ejecutivo, en los casos de uso exclusivo. Como se observa en la figura, este rol es el encargado de levantar los tickets de mantenimiento programado. Cuando el usuario levanta un ticket de mantenimiento programado, el sistema muestra un formulario con la fecha para su atención y el laboratorio en el que se realizará dicha actividad. Levantado el ticket, el sistema envía una notificación por correo electrónico al Líder Técnico y al encargado del laboratorio de la solicitud realizada y genera los tickets de soporte técnico necesarios para cada equipo de cómputo de dicho laboratorio que se encuentra registrado en el Sistema de Inventarios. Una vez que el mantenimiento programado se concluye, el sistema le enviará un correo al Administrador Ejecutivo para validar dicho mantenimiento.

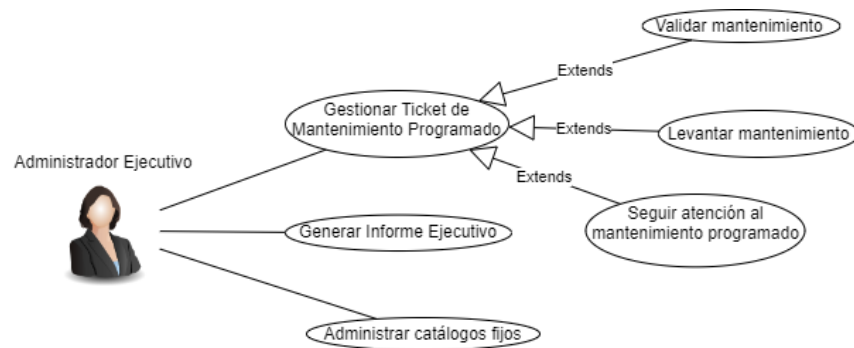


Figura 3.3: Casos de Uso para el Administrador Ejecutivo

El sistema permite al Administrador Ejecutivo observar los avances en el mantenimiento, así como generar el informe ejecutivo (graficación de los tickets atendidos por la UC) y mostrar un listado largo de todas las actividades realizadas en la UC.

### 3.2.2. Administrador del Sistema

En la figura 3.4 se muestra el diagrama UML para el usuario con el rol de Administrador del Sistema. Este usuario se encarga de la administración de todos los catálogos del sistema;

al entrar, el sistema despliega un menú donde se encontrarán los módulos: Foros y Gestión de Catálogos. En el primero, podrá acceder a la sección de Foros de Atención Tecnológica para agregar, modificar o eliminar cualquier tutorial presentado al usuario Solicitante; en el segundo, al darle clic al módulo, se muestran los catálogos que tiene a su disposición para ser modificados (Usuarios, Actividades, Laboratorios y Listas de Tareas y Software a revisar).

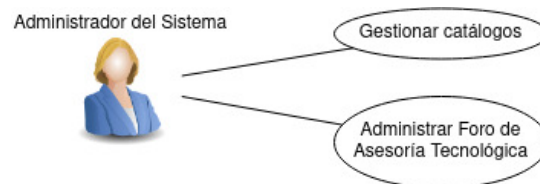


Figura 3.4: Casos de Uso para el Administrador del Sistema

### 3.2.3. Líder Técnico

En la figura 3.5 se muestra el diagrama UML para el usuario con el rol de Líder Técnico. Este usuario realiza la asignación de técnicos. Seleccionado el ticket que corresponda asignar, el sistema arroja los datos que el usuario Solicitante registró en el levantamiento del mismo, con lo cual permite ingresar al técnico que realizará el soporte, la o las actividades propuestas a realizar, fecha propuesta de término y, opcionalmente, una fecha programada de la atención (si el usuario requiere que la atención sea diferente a la que se recibe su solicitud). Un valor agregado del SGT para este usuario es que le permite conocer un listado de los tickets que siguen pendientes de asignar y de los que se han asignado con anterioridad y continúan siendo atendidos. El Líder Técnico puede contactarse con el Solicitante a través del Foro de Comunicación creado para dicha solicitud (una vez que termine de llenar el formulario, el sistema envía un correo electrónico de notificación al técnico asignado para atenderlo). Si el técnico reporta que no puede seguir con la atención del ticket, el sistema envía un correo electrónico al Líder Técnico informándole que es necesario reasignar el ticket a otro técnico.

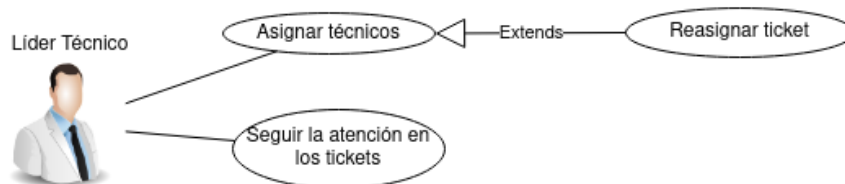


Figura 3.5: Casos de Uso para el Líder Técnico

El Líder Técnico puede seguir, en todo momento, la atención a los tickets abiertos asignados por él, así como acceder a los reportes que el Administrador Ejecutivo tiene acceso para ver el listado de actividades realizadas en la UC y la graficación de toda esta información generada para el reporte ejecutivo.

### 3.2.4. Supervisor

En la figura 3.6 se muestra el diagrama UML para el usuario con el rol de Supervisor. Este usuario se encarga de verificar si la atención de un Técnico Junior fue realizada de manera correcta. Una vez verificada, el sistema envía un correo al técnico notificándole que su atención fue correcta, al mismo tiempo que lo hará para el usuario Solicitante, quien otorgará la validación de la atención de su ticket por haber sido terminado con éxito. Por otra parte, el sistema notifica al Supervisor cuando la atención realizada por este tipo de técnico concluye.

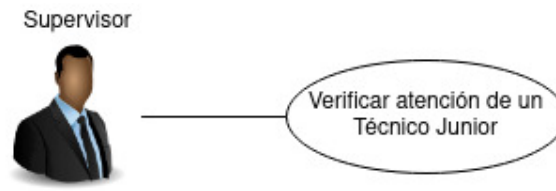


Figura 3.6: Caso de Uso para el Supervisor

### 3.2.5. Encargado de Laboratorio

En la figura 3.7 se muestra el diagrama UML para el usuario con el rol de Encargado de Laboratorio. Este usuario realiza la verificación administrativa del mantenimiento programado; de tal manera que el sistema mostrará las actividades efectuadas en cada equipo de cómputo, pudiendo agregar observaciones generales al mantenimiento de que se trate. Hecha la verificación, el sistema notifica al Administrador Ejecutivo que ha concluido satisfactoriamente el mantenimiento programado solicitado a través de un correo electrónico con el detalle del mismo.

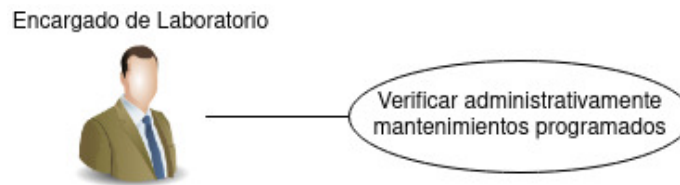


Figura 3.7: Caso de Uso para el Encargado de Laboratorio

### 3.2.6. Técnicos

En la figura 3.8 se muestra el diagrama UML para el usuario con el rol de Técnico Junior o Máster. Estos dos roles difieren exclusivamente en la verificación; es decir, que tratándose de un Junior necesitará de una verificación (por parte de un Supervisor) de la atención prestada, antes de concluir el ticket; mientras que el caso del técnico Máster no estará sujeto a verificación alguna. El sistema muestra al técnico la actividad propuesta



por el líder técnico y lo registrado por el usuario Solicitante. Puede endosar el ticket, en caso de que no pueda atender la solicitud; y registrar los cambios realizados al equipo de cómputo para el caso del Ticket de Soporte Técnico. El sistema le permite comunicarse con el usuario Solicitante a través del Foro de Comunicación creado para cada solicitud. Un aspecto que deberá considerarse, es la existencia de un semáforo que indica el tiempo faltante para resolver la atención de la solicitud.

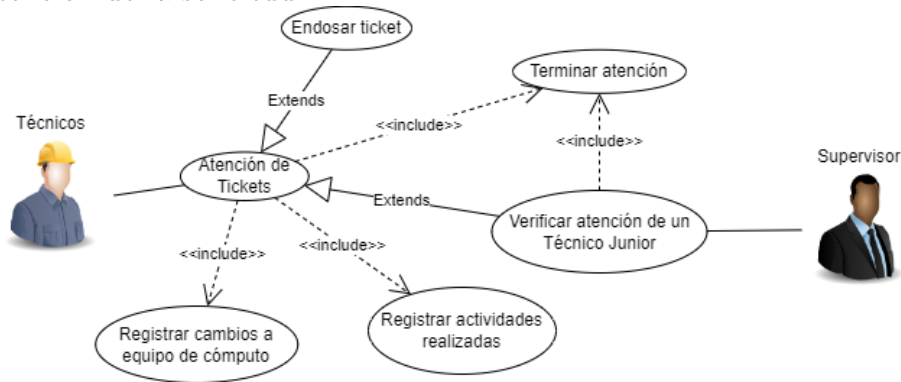


Figura 3.8: Casos de Uso para el Técnico Junior o Máster

### 3.2.7. Solicitante

En la figura 3.9 se muestra el diagrama UML para el usuario con el rol de Solicitante. Este usuario realiza el levantamiento de Tickets de Soporte Técnico y/o Capacitación. El sistema despliega los dispositivos que tiene registrados en el Sistema de Inventarios (a través de su número de cuenta/trabajador), motivo por el cual solicita el soporte, una breve descripción de lo que sucede en el momento de solicitar el apoyo y, si es necesario, accesorios a resguardar. Para el caso del ticket de capacitación, le solicitará el requerimiento y motivo por el cual solicita dicha capacitación. Una vez levantado, se envía correo de confirmación y a los líderes técnicos, el sistema informa que se ha levantado un nuevo ticket para su asignación y pronta atención de un técnico de la UC.

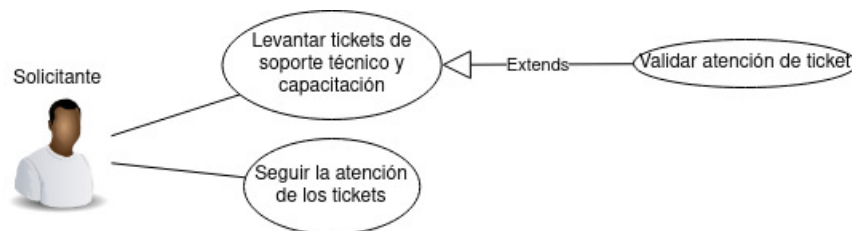


Figura 3.9: Casos de Uso para el Solicitante

El usuario Solicitante se puede comunicar con el staff de la UC en todo momento a través del Foro de Comunicación creado para la atención de su solicitud a fin de darle seguimiento,

estando facultado para consultar el módulo de Seguimiento para observar sus tickets abiertos y en qué estado se encuentra su atención.

### 3.3. Diseño de la base de datos

Una vez realizado el diseño de la arquitectura del sistema y la creación de las historias de usuario para cada rol, es posible continuar con la primera etapa del diseño de la base de datos del SGT; a saber, el diseño conceptual.

#### 3.3.1. Diseño conceptual

Se procede a la construcción del modelo Entidad/Relación extendido, ocupando la notación de Peter Chen como se muestra en la figura 3.10. Este modelo se basa en la percepción del mundo real, circunscritos en un conjunto de objetos básicos llamados entidades y sus consecuentes relaciones entre los mismos.

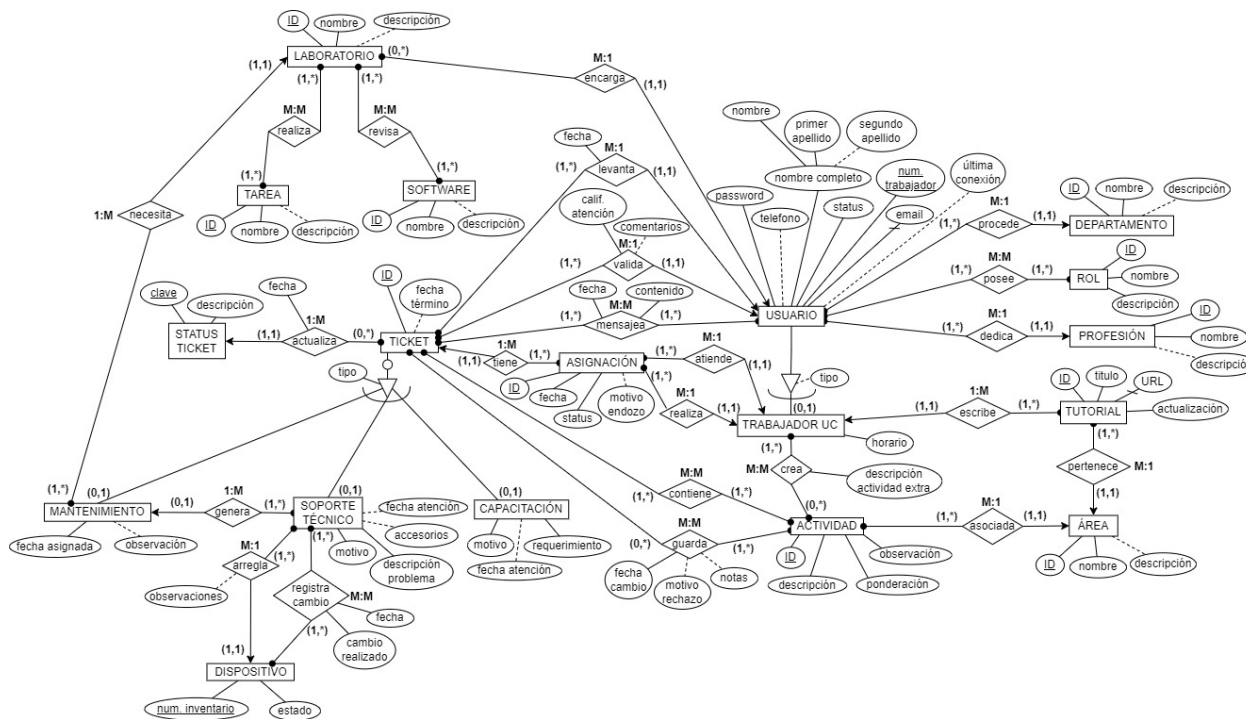


Figura 3.10: Modelo E/R extendido del SGT

#### 3.3.2. Diseño lógico

Acto seguido, se ejecuta la transformación del modelo E/R en el modelo que será empleado por el DBMS. Para el caso del SGT, se utiliza el modelo relacional ocupando la notación

Crow's foot, tal y como se observa en la siguiente figura 3.11. Este esquema lógico global presenta la estructura general de la base de datos del sistema.

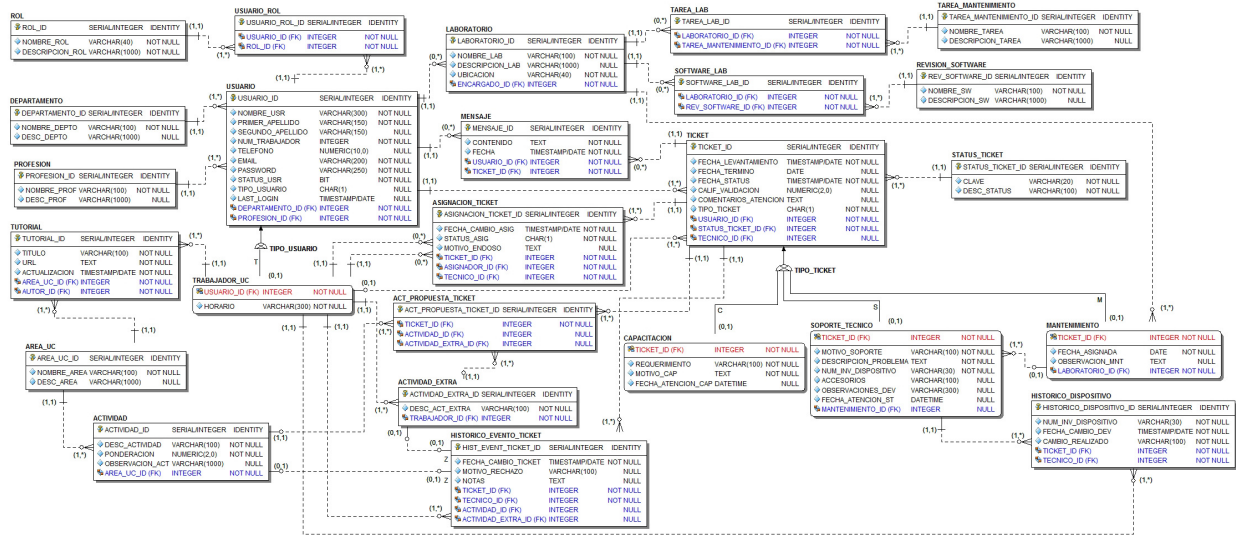


Figura 3.11: Modelo relacional del SGT

### 3.3.3. Diseño físico

A partir del diseño lógico, se implementarán todos los objetos en el esquema public (creado por defecto por el DBMS elegido), mismo en el que se administrará la base de datos del SGT. Para la creación de las restricciones (constraints) a nivel columna, al momento de crear las tablas del modelo relacional en PostgreSQL se seguirá la convención en su nombrado contenida en la tabla 3.1, con la nomenclatura variante de la notación Backus-Naur Form:

TIPO RESTRICCIÓN	Nombre a utilizar
PRIMARY KEY	<nombre-tabla>_pk
FOREIGN KEY	<nombre-tabla-hija>_<nombre-columna>_fk
UNIQUE	<nombre-tabla>_<nombre-columna>_uk
CHECK	<nombre-tabla>_<nombre-columna>_ck

Tabla 3.1: Convención de nombrado para constraints a nivel columna

Cabe precisar que para mejorar el desempeño de la base de datos se utilizaron llaves primarias artificiales en cada una de las tablas del modelo relacional, siguiendo la convención <nombre-tabla>\_id para nombrarlas, respetando en todo momento la creación automática del índice primario para estos campos.

Por otra parte, se hará uso tanto de índices comunes para mejorar el desempeño en consultas que involucren campos utilizados frecuentemente (por ejemplo, en llaves foráneas), así como de índices únicos para verificar duplicidad de valores. Su convención de nombrado será la mostrada en la siguiente tabla 3.2:

TIPO ÍNDICE	Nombre a utilizar
NON UNIQUE	<prefijo-tabla>_<nombre-campo>_ix
UNIQUE	<prefijo-tabla>_<nombre-campo>_iuk

Tabla 3.2: Convención de nombrado para el esquema de indexado

Algunas de las llaves foráneas indexadas fueron el atributo de `usuario_id` en las tablas que se relacionan con los datos del usuario que se almacenan en el sistema, el `tecnico_id` al relacionar tablas con la información de un trabajador de la UC y el `ticket_id` para poder obtener más información de la solicitud en todo el sitio web cuando sea necesario. Esta indexación se realizó con el propósito de mejorar las consultas que utilizan estos atributos como enlaces entre las tablas para obtener información completa en cada página web que las relaciona.

Mientras que algunos de los atributos únicos que se indexaron fueron `num_trabajador` y `email`, por su alto uso en el sistema al momento de emplear estos campos como filtros en las consultas que se realizan a fin de mostrar la información del sitio web acerca del número de trabajador/cuenta y correo electrónico para cada usuario del sistema.

En el apéndice B se detalla el diccionario de datos para la implementación en PostgreSQL del SGT, en el cual se describen los atributos del esquema lógico, especificándose para cada uno de ellos su dominio y valor por defecto -entre otras características- a fin de mejorar la comprensión del diseño físico implementado.

Concluido que fue el modelado de la base de datos, se continúa con el desarrollo del sistema a través de los frameworks de Django y Bootstrap para la parte de FrontEnd y BackEnd del SGT.

# Capítulo 4

## Desarrollo

Creada la base de datos del SGT, se procedió al desarrollo del sistema de manera local, por lo que fue indispensable allegarse de las herramientas de software para el presente proyecto.

### 4.1. Herramientas de software

Para realizar el seguimiento de los cambios hechos sobre el código fuente del sistema, se utilizó *Git* como el sistema de control de versiones distribuido. Lo anterior, para no depender de un servidor central al contar con una copia del repositorio de manera local [74]. Ahora bien, a solicitud del cliente, se ocupó GitHub como servidor en línea del repositorio remoto para subir los cambios realizados localmente.

Por la conexión a las bases de datos que manejará el SGT de manera local, fue necesaria la instalación de *PostgreSQL* (en su versión 13.7 para el servidor local de la base de datos del SGT) y de *MySQL* (en su versión 5.5.47 para el servidor local de la base de datos del Sistema de Inventarios). En ambas bases de datos se implementa una copia de manera local dado que no se cuenta con una conexión a un servidor de pruebas para realizar el desarrollo del sistema.

Un requerimiento indispensable lo constituyó la instalación de *Python 3* (entre las versiones 3.7 y 3.10) para poder ocupar el framework de BackEnd. Ahora bien, para poder emplear los módulos del SGT, la creación de un entorno virtual con la herramienta *pip* y el módulo de *virtualenv*, resultaron altamente recomendados a fin de llevar a cabo distintos desarrollos de software en la misma máquina local y no afectar el desempeño del sistema a crear.

Finalmente, se utilizó *Visual Studio Code* como Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés) que permite combinar las distintas herramientas de las que se vale el desarrollador para programar de una manera más rápida, tales como editor de código fuen-

te, automatizaciones de compilaciones locales y depurador, entre otras, en una sola Interfaz Gráfica de Usuario (GUI, por sus siglas en inglés).

Una vez obtenidas las herramientas de software para realizar el desarrollo del sistema, se ejecutaron las configuraciones locales que posibilitaron el inicio de la creación del proyecto de Django en el que se desarrollará el SGT.

## 4.2. Configuraciones iniciales

De acuerdo a lo propuesto en la sección 3.1.3 del capítulo 3, se creó un proyecto de Django con la herramienta de línea de comandos: `django-admin` que proporciona este framework para realizar tareas administrativas del sistema, como lo fue el inicio del mismo. Con esta herramienta se crea la carpeta del proyecto, plantillas de archivos básicos y el archivo `manage.py` que servirá como script en la gestión de este proyecto.

Dado que, tanto la base de datos del SGT como la del Sistema de Inventarios fueron creadas con antelación a la creación del proyecto de Django, se realizó la generación de modelos por cada tabla del modelo relacional en estos dos sistemas. Para realizar esto de manera automatizada, primero se debe establecer la conexión y manejo de múltiples bases de datos en el proyecto de Django, agregando los parámetros de conexión para cada uno de estos DBMS; es decir, el nombre de la base de datos, el usuario, la contraseña, la IP y el puerto al diccionario de `DATABASES` de las configuraciones encontradas en el archivo `settings.py`. Este diccionario contiene -como su nombre lo indica- las configuraciones de todas las bases de datos que se utilizarán con este framework.

Realizadas dichas configuraciones, es posible llevar a cabo una “introspección” a las bases de datos con el comando `inspectdb` que se encuentra en el script `manage.py` y con el cual se crea un modelo de clase por cada tabla encontrada en la base de datos que se indique. Cabe puntualizar que fue necesaria una limpieza en los modelos generados automáticamente, la cual consistió en reacomodar el orden de los modelos que tienen referencia con otros, en el que primero se declaran los modelos que no tienen llaves foráneas y después los que se relacionan con ellos; además de que se especificaron los atributos que cuentan con valores por defecto en el modelado de la base de datos [75].

Además de esta limpieza, para cada modelo que utilizará Django se agregarán a sus metadatos el nombre de la tabla a la cual tiene que ser mapeada, así como la indicación de que no se realizará ningún cambio en la estructura de la base de datos desde el framework de desarrollo. El propósito de ello estriba en ejecutar los cambios en la base de datos de manera independiente a lo realizado en el código y con previa autorización del líder técnico en desa-

rollo y base de datos de la UC. Para el caso del Sistema de Inventarios, solo se declararon los modelos que serán consultados en el SGT, siendo los mostrados en la siguiente figura 4.1:

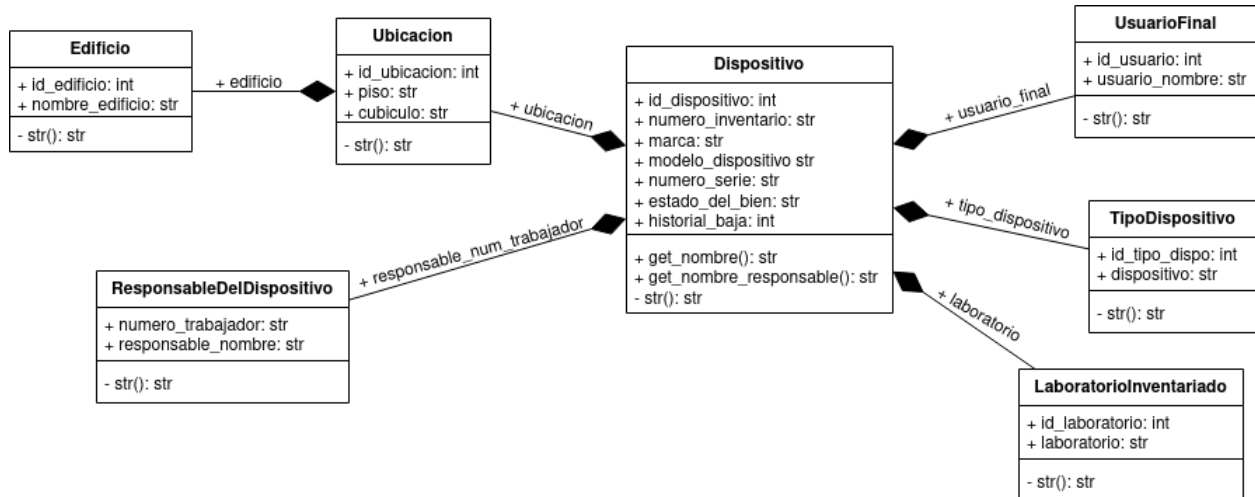


Figura 4.1: Diagrama de clases para el Sistema de Inventarios en el SGT

Con el mapeo de los modelos para cada base de datos, se puede iniciar la creación de cada uno de los módulos que contendrá el sistema con el comando `startapp` del script de **manage.py**. Dicho comando creará una carpeta dentro del proyecto de Django con los archivos iniciales para realizar la creación de cada app. Para cada módulo (app) se agregarán los modelos que principalmente ocupe y los modelos restantes se irán importando e incorporando en cada una de las apps a desarrollar conforme se vayan utilizando en el sistema. También es importante mencionar que, al momento de crear cada app, se debe agregar al diccionario de `INSTALLED_APPS` el nombre del módulo creado e indicar en el archivo **urls.py** el nombre de donde se tomarán las rutas para acceder a cada una de las vistas.

### 4.3. Módulos del sistema

Una vez definidos los modelos que se utilizarán, es necesario realizar las migraciones iniciales que requiere el framework de Django para la gestión de usuarios, la administración de sesiones y demás configuraciones que se requieran sobre la base de datos donde se harán operaciones de escritura. Los comandos `makemigrations` y `migrate` del script **manage.py** permiten el registro de los cambios en los modelos y ser efectuados en el mapeo con la base de datos del SGT. A continuación, se describirá cada módulo del sistema creado con las funcionalidades principales implementadas en el SGT.

### 4.3.1. Inicio de Sesión y Gestión de Usuarios

En la app *usuarios* se realizó la implementación del inicio de sesión en el sistema. Primeramente lo que se abordó es la autenticación -que viene en el framework de BackEnd con el modelo *User*- pero para este software a la medida es necesario personalizar este modelo para que sea adecuado con el usuario que ingresará al sistema.

Para personalizar completamente el modelo *User* dado por Django y no perder las funcionalidades de autenticación dadas por el framework, en el modelo *Usuario* (mapeado de la base de datos del SGT) se realiza la herencia de la subclase *AbstractBaseUser* y se debe indicar, en la variable de instancia `USERNAME_FIELD`, el campo que se utilizará para autenticar al usuario con este nuevo modelo, así como en el archivo de configuraciones del proyecto (`settings.py`) es necesario indicarle a Django la ubicación del modelo personalizado que debe usar para autenticar en la variable `AUTH_USER_MODEL`.

Adicionalmente, debe hacerse particular mención de que se establecieron los permisos a cada módulo del sistema de acuerdo al rol(es) registrado(s) por cada tipo de usuario del SGT. Además resultó indispensable la implementación de un manejador del nuevo modelo *Usuario*, ya que Django desconoce cómo realizar la creación de usuarios a partir de éste, por lo que se crea una clase adicional que hereda de *BaseUserManager* en la que se sobrescriben los métodos de creación de usuario y del superusuario. El superusuario será un Administrador Ejecutivo, por lo que para permitir su creación desde línea de comandos, debe relacionarse con su rol correspondiente en el sistema e indicar que se trata de un trabajador de la UC.

Puntualizar que, establecido el modelo *Usuario* que utilizará Django, para realizar las gestiones de inicio y cierre de sesión del SGT se optó por el sistema de autenticación que proporciona este framework, ya que provee todo el flujo y vistas integradas para el manejo de toda esta funcionalidad. Estas vistas fueron personalizadas por Bootstrap para brindarle al usuario final una mejor visualización y usabilidad como se muestra en la figura 4.2:

Figura 4.2: Inicio de Sesión personalizado para el SGT



Ahora bien, para realizar la gestión de usuarios en el SGT se crearon formularios para ingresar y actualizar la información necesaria en cada modelo y poder así registrar todos los atributos relacionados con un usuario. Tratándose del formulario de registro de un nuevo usuario, se heredó de *UserCreationForm* un nuevo formulario que permitiera registrar si determinado usuario es o no trabajador de la UC, además de los atributos del modelo *Usuario* implementado. Asimismo, en aras de proveer de una mejor visualización y usabilidad del sistema, en todos los formularios se empleó el módulo de *django-crispy-forms* en conjunto con Bootstrap para obtener un formulario como el mostrado en la figura 4.3:

**Sistema de Gestión de Tickets**

[Solicitudes](#)
[Asignación](#)
[Atención](#)
[Seguimiento](#)
[Foros](#)
[Reportes](#)
[Gestión de catálogos](#)
[Tanya Itzel](#)

### Ingrese los datos del nuevo usuario

Nombre\* 
 Primer apellido\* 
 Segundo apellido

Correo electrónico\* 
 Número de trabajador/cuenta\* 
 Teléfono

Departamento\* 
 Ocupación\* 
 ¿Es trabajador de la UC?

Los campos obligatorios se indican con \*

[Agregar](#)
[Regresar](#)

© 2023 División de Ingenierías Civil y Geomática | UNAM, todos los derechos reservados  
 Unidad de Cómputo, Edificio R Conjunto Sur, FI UNAM Tel: 5622-8005 ext. 1037  
 Contacto: [computo.dicyg@gmail.com](mailto:computo.dicyg@gmail.com)

[Créditos](#)

Figura 4.3: Formulario para registrar nuevos usuarios

Cuando se registra a un nuevo usuario, el SGT le notifica sobre esta acción (con su contraseña autogenerada de manera aleatoria) al mismo correo electrónico con el que se realiza su registro. Para poder proceder al envío de dicha notificación, fue necesaria la creación de una plantilla base en la que se estructura un archivo HTML a través de anidamiento de tablas y uso de estilos con CSS para ser responsivo a cualquier plataforma en la que se observe la notificación. Los colores deben ser adecuados tanto para pantallas con fondo blanco como las de en modo oscuro para su óptima visualización, cuya ejemplificación la podemos observar en la siguiente figura 4.4 que muestra la notificación de registro exitoso en el SGT a través de un celular:

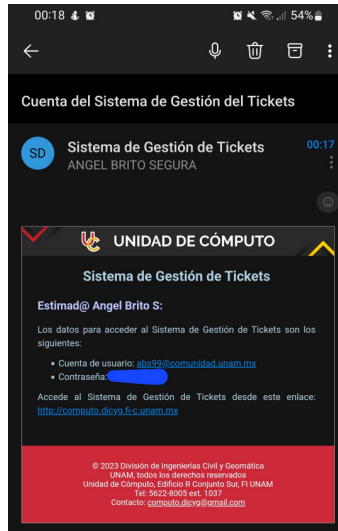


Figura 4.4: Notificación sobre registro exitoso en el SGT

Conocer el estado del sistema en la gestión de nuevos usuarios fue posible gracias al empleo de sesiones que vienen en el framework de Django que fueron ocupadas para permitir el desplazamiento de una vista a otra de manera segura y siguiendo la secuencia indicada en las reglas de negocio del SGT. Estas sesiones se estarán almacenando en la base de datos para mantener la información de forma segura.

### 4.3.2. Gestión de Catálogos

En la app *catalogos* se realiza la gestión de los catálogos que serán utilizados a lo largo del sistema. La modificación, creación y eliminación de registros en cada uno de estos modelos, se harán mediante las vistas genéricas de Django. Para lograrlo fue necesario crear clases que heredan de *CreateView*, *UpdateView* o de *DeleteView* dependiendo de la modificación de datos a realizar. Para listar toda la información registrada en un modelo, se tienen las clases que heredan de *ListView* como se muestra en la figura 4.5:

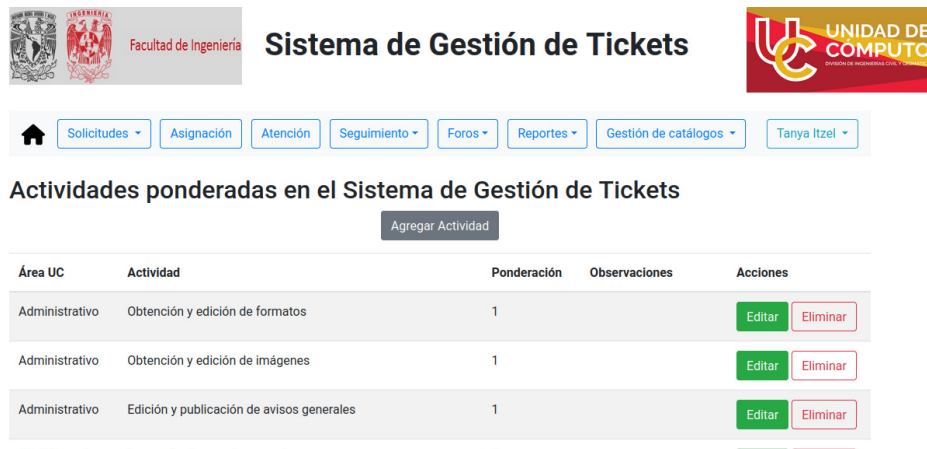


Figura 4.5: Uso de ListView para mostrar listado de registros en un modelo

Por lo que respecta a la gestión de catálogos fijos, como lo son las áreas de la UC, departamentos y profesiones que se asocian a un usuario, debe decirse que el administrador que trae este framework de BackEnd permite agregar, modificar y eliminar cualquier registro de los modelos del archivo `admin.py` en cada app que los administre por esta vía. Los estilos por defecto que utiliza este administrador fueron cambiados para adecuarlos con los empleados en el SGT, como se muestra en la siguiente figura 4.6:



Figura 4.6: Administrador de Django para catálogos fijos del SGT

### 4.3.3. Foros de Comunicación y Asesoría Tecnológica

La app *foros*, como su nombre lo indica, se encarga de la gestión de los foros de asesoría tecnológica, los cuales consisten en una serie de tutoriales creados por el staff de la UC, y donde solo personal de dicha unidad administrativa está autorizado para realizar las modificaciones que estime pertinentes. Como se muestra en la figura 4.7, si el usuario no forma parte del staff de la UC no se le habilitan los botones de edición; solo puede consultar los tutoriales que estén disponibles:

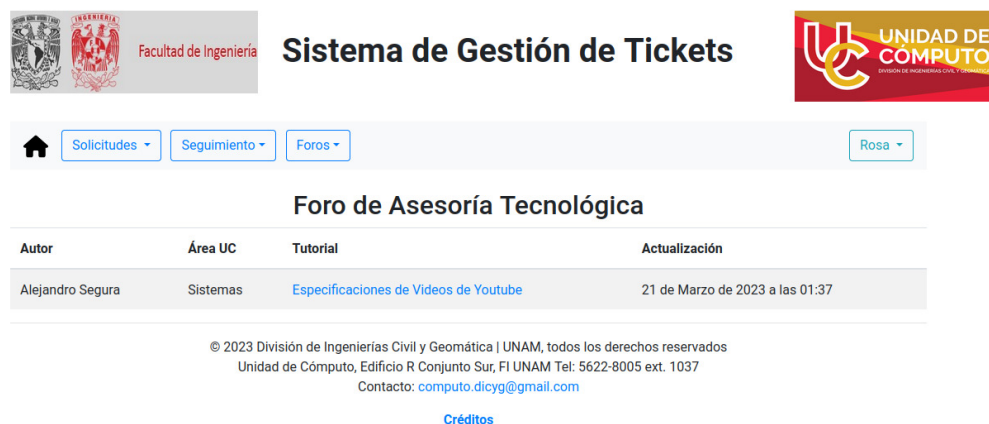


Figura 4.7: Listado de tutoriales disponibles en el Foro de Asesoría Tecnológica

Tratándose del Foro de Comunicación, solo el staff de la UC puede conocer el nombre de los remitentes de cada mensaje enviado, como se observa en la figura 4.8. Para mejorar la vista, se utilizaron avatares para distinguir al usuario solicitante del staff de la UC, así como el uso de colores, posiciones y notificaciones en tiempo real para emular un chat real de cualquier medio de comunicación existente.



Figura 4.8: Foro de comunicación entre el staff de la UC y usuarios Solicitantes

#### 4.3.4. Levantamiento, Asignación y Cierre de Tickets

En la app *soporte\_tecnico* se realiza el levantamiento de cualquier tipo de ticket de soporte técnico, solicitando información relevante para su atención, con formularios creados directamente con Bootstrap para una mejor visibilidad y usabilidad con el usuario Solicitante. Cuando se envía la solicitud, se muestra la siguiente pantalla (figura 4.9) con la cual se confirma que el staff de la UC ha sido notificado para iniciar la atención:

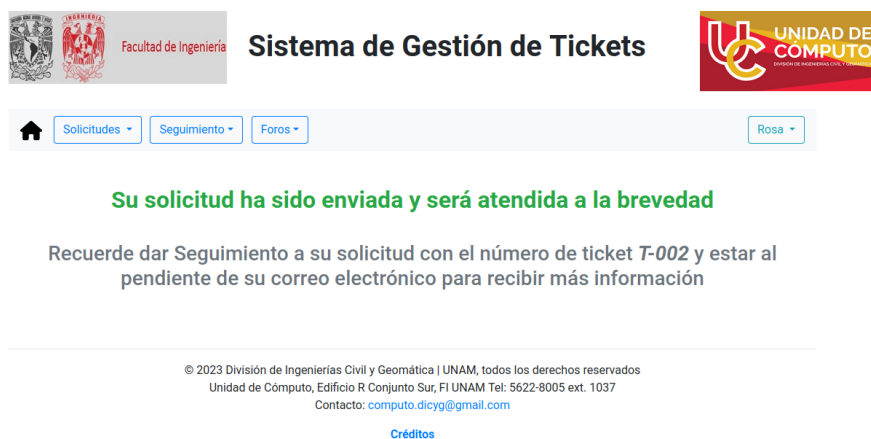
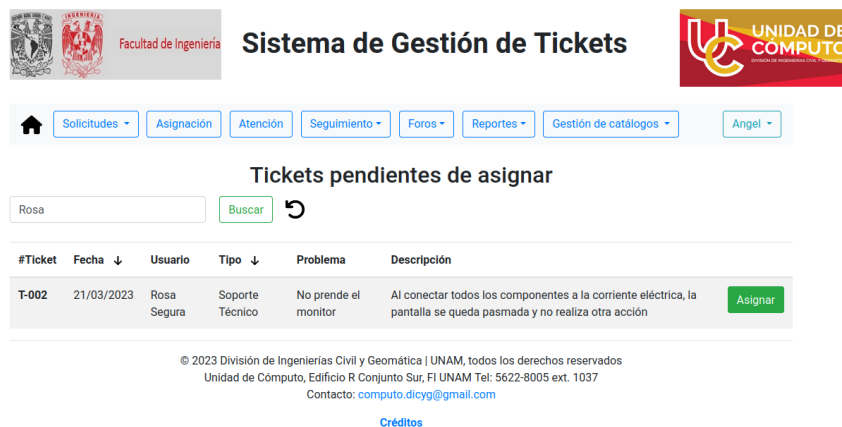


Figura 4.9: Confirmación de levantamiento de ticket exitoso

En la lista de los tickets pendientes de asignar, se agregó una búsqueda por *Full Text Search* para poder filtrar al usuario solicitante faltante de asignación. La ejecución de esta búsqueda se observa en la siguiente figura 4.10:



The screenshot shows the 'Sistema de Gestión de Tickets' interface. At the top, there are logos for 'Facultad de Ingeniería' and 'UNIDAD DE CÓMPUTO'. Below the logos is a navigation bar with buttons for 'Solicitudes', 'Asignación', 'Atención', 'Seguimiento', 'Foros', 'Reportes', 'Gestión de catálogos', and 'Angel'. The main heading is 'Tickets pendientes de asignar'. Below this is a search bar with the text 'Rosa' and a 'Buscar' button. A table below the search bar shows the following data:

#Ticket	Fecha ↓	Usuario	Tipo ↓	Problema	Descripción	
T-002	21/03/2023	Rosa Segura	Soporte Técnico	No prende el monitor	Al conectar todos los componentes a la corriente eléctrica, la pantalla se queda pasmada y no realiza otra acción	Asignar

At the bottom of the page, there is a copyright notice: '© 2023 División de Ingenierías Civil y Geomática | UNAM, todos los derechos reservados. Unidad de Cómputo, Edificio R Conjunto Sur, FI UNAM Tel: 5622-8005 ext. 1037. Contacto: computo.dicyg@gmail.com'. There is also a 'Créditos' link.

Figura 4.10: Búsqueda de usuarios Solicitantes en la asignación de técnicos

En el caso de la asignación de técnicos y actividades propuestas a realizar, se utilizó AJAX para solicitar peticiones asíncronas al servidor y no recargar toda la página en cada una de estas actividades; con ello, se pretenden reducir los tiempos de carga y mejorar la experiencia de usuario.

Finalmente, para concluir el ciclo de vida de este ticket se implementa una pequeña encuesta que es necesario contestar para poder solicitar más tickets. La encuesta mostrada en la figura 4.11 se integra de dos preguntas, siendo obligatorio responder qué tan satisfecho se encuentra con el servicio (opcionalmente se puede responder la segunda de ellas, agregando comentarios para mejorar la atención).



The screenshot shows the 'Sistema de Gestión de Tickets' interface. At the top, there are logos for 'Facultad de Ingeniería' and 'UNIDAD DE CÓMPUTO'. Below the logos is a navigation bar with buttons for 'Solicitudes', 'Seguimiento', 'Foros', and 'Rosa'. A green notification bar at the top says 'Ticket validado con éxito'. The main heading is 'Ticket T-002 concluido con éxito Encuesta de satisfacción'. Below this is the question '¿Cómo fue tu experiencia al realizar esta solicitud?' with three emoji options: a happy face, a neutral face, and a sad face. Below the question is a text input field with the placeholder text 'Ingresar comentarios para mejorar la atención'. At the bottom of the page, there are three buttons: 'Enviar encuesta', 'Limpiar', and 'Regresar'. At the bottom of the page, there is a copyright notice: '© 2023 División de Ingenierías Civil y Geomática | UNAM, todos los derechos reservados. Unidad de Cómputo, Edificio R Conjunto Sur, FI UNAM Tel: 5622-8005 ext. 1037. Contacto: computo.dicyg@gmail.com'. There is also a 'Créditos' link.

Figura 4.11: Encuesta de Satisfacción del Servicio

### 4.3.5. Atención, Verificación y Validación de un Ticket

En la app *seguimiento\_atencion* es posible registrar los cambios realizados en el dispositivo sujeto a soporte técnico, así como las actividades realizadas en la atención del ticket. Al igual que con el módulo de Asignación, se utiliza la función de AJAX de jQuery para solicitar peticiones asíncronas. Si las peticiones son de tipo POST, es necesario obtener el token con el cual se está creando el formulario, a través de una función JavaScript para evitar la vulnerabilidad de CSRF (falsificación de petición en sitios cruzados, por sus siglas en inglés). En la figura 4.12 se muestra una bitácora de las actividades registradas en la atención del ticket.

The screenshot displays the 'Sistema de Gestión de Tickets' interface. At the top, there are logos for 'Facultad de Ingeniería' and 'UNIDAD DE COMPUTO'. The main navigation bar includes buttons for 'Atención', 'Seguimiento', and 'Foros', along with a user profile dropdown for 'José'. The central heading is 'Atención del Ticket T-002'.

**Actividades realizadas anteriormente:**

Fecha cambio	Colaborador	Actividad realizada
21/03/2023 02:33	José Luis	Mantenimiento correctivo sw de equipos (ALERTA)

**Dispositivo 2215714 - CPU COMPAQ Presario 061**  
Registra cada cambio realizado en este dispositivo durante la atención del ticket:

Si es necesario, anota observaciones al dispositivo asociado a esta atención:

**Actividades propuestas a realizar**  
Dar clic a la actividad realizada para su registro:

De ser necesario, registra notas o comentarios a la atención de este ticket:

**Ticket de Soporte Técnico**  
Fecha propuesta de término: 29 de Marzo de 2023  
Semáforo: ●  
Usuario: Rosa Segura  
Dispositivo: 2215714 - CPU COMPAQ Presario 061  
Accesorios:  
Problema reportado: **No prende el monitor**  
Descripción del problema: Al conectar todos los componentes a la corriente eléctrica, la pantalla se queda pasmada y no realiza otra acción

*Si es necesario, contacte con el usuario solicitante en el [Foro correspondiente para solicitar más información](#)*

Figura 4.12: Registro de actividades, notas y cambios en dispositivo durante su atención

Si el ticket de soporte técnico es atendido por un Técnico Junior o es de un Mantenimiento Programado, será necesario pasar por una verificación técnica que realiza el Supervisor. La siguiente figura 4.13 muestra un despliegue de todas las actividades o notas registradas

durante la atención del ticket, incluyendo la verificación o indicación que la solución no es la correcta y, con ello, estar en posibilidad de regresar la atención al técnico asignado.

Figura 4.13: Verificación técnica de atención proporcionada por un Técnico Junior

Finalmente, el usuario Solicitante valida que la atención haya sido la correcta en caso de serlo; o bien, podría rechazar la solución dada por los técnicos, en cuyo supuesto se regresará para su cabal atención al técnico asignado, tal y como se muestra en la siguiente figura 4.14:

Figura 4.14: Validación de la atención realizada

### 4.3.6. Informes Ejecutivos

En la app *reportes* se despliega la graficación de los informes ejecutivos solicitados por el cliente con la librería de JavaScript `Chart.js` para todas las actividades realizadas en la UC durante la atención de tickets. A través de esta librería se visualizará, de manera adecuada, cada una de estas gráficas toda vez que se obtienen de forma dinámica al ser solicitadas por el usuario. En la figura 4.15 se muestra el desempeño de la UC con relación a los levantamientos atendidos en un periodo de tiempo determinado:

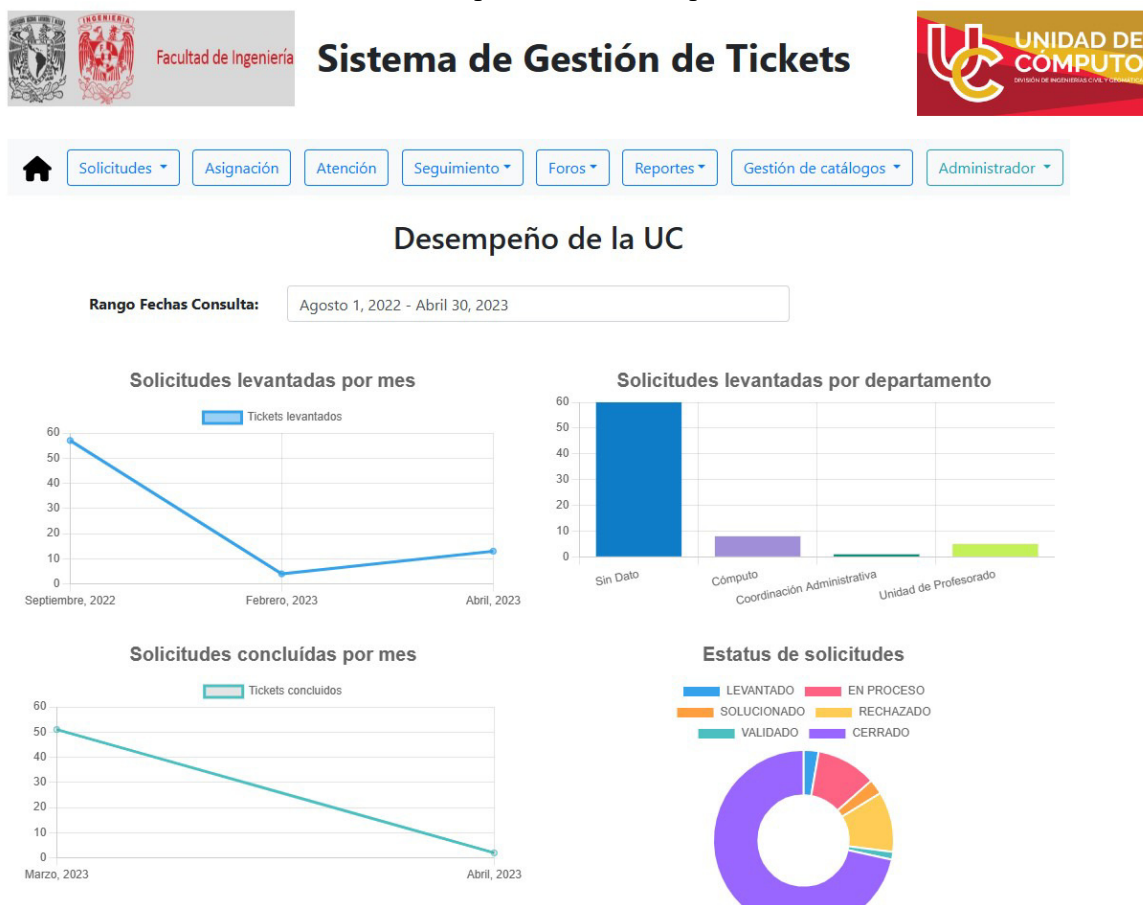


Figura 4.15: Graficación de las solicitudes recibidas por la UC

## 4.4. Validaciones

A continuación se mostrarán las distintas validaciones que se realizaron para comprobar la funcionalidad del SGT en cada uno de los módulos construidos en el desarrollo local, en las que se describen los pasos realizados y el resultado definido por las reglas de negocio de los requerimientos del proyecto.



### 4.4.1. Información de los usuarios

Con el rol Administrador del Sistema o Administrador Ejecutivo, se ingresa al sistema con correo y contraseña proporcionados con anterioridad (al ser registrado por primera vez) seleccionando el módulo del menú **Gestión de Catálogos** en la opción “Usuarios”. En la vista que se despliega (*Usuarios registrados en el Sistema de Gestión de Tickets*), se observa el botón de **Agregar Usuario Nuevo**, centrado en la página web, así como un despliegue de todos los usuarios registrados. Este despliegue, a su vez, trae consigo dos botones para realizar cambio de la información registrada y de los permisos (roles) con los que cuenta el usuario, tal y como lo muestra la figura 4.16:

Nombre completo	Correo	Departamento	Rol(es) en el Sistema	Estatus	Acciones
Angel Brito	antovan99@gmail.com	Sin Dato	Administrador Ejecutivo	✓	Editar Permisos

Figura 4.16: Vista con información de todos los usuarios registrados

Si el correo o número de trabajador/cuenta ya se encuentra registrado en el SGT, en el formulario de creación/edición de datos de un usuario se detiene la subida de información al servidor y se indican los campos que no fueron posible modificar. Ello se muestra en la figura 4.17:

Formulario no válido, checka los campos marcados en rojo y vuelve a intentarlo

**Modifica los datos que deseas cambiar del usuario**

Nombre\*  Primer apellido\*  Segundo apellido

Correo electrónico\*  Número de trabajador/cuenta\*  Teléfono

Ya existe un/a Usuario con este/a Correo electrónico. Ya existe un/a Usuario con este/a Número de trabajador/cuenta.

Departamento\*  Ocupación\*   ¿Usuario activo?  ¿Es trabajador de la UC?

Los campos obligatorios se indican con \*

[Actualizar datos](#) [Regresar](#)

© 2023 División de Ingenierías Civil y Geomática | UNAM, todos los derechos reservados  
 Unidad de Cómputo, Edificio R Conjunto Sur, FI UNAM Tel: 5622-9005 ext. 1037  
 Contacto: [computo.dicoy@gmail.com](mailto:computo.dicoy@gmail.com)  
[Créditos](#)

Figura 4.17: Validación de unicidad para correo o número de cuenta/trabajador

### 4.4.2. Gestión de contraseña

Si al intentar acceder al sistema el usuario no recuerda su contraseña, existe la opción de recuperarla; por lo que en este supuesto se dirige al botón *¿Has olvidado tu contraseña?* dentro de la vista **Inicio de Sesión**. En esta vista de **Recuperación de contraseña**, ingresa el correo electrónico con el cual se encuentra registrado. Si el correo proporcionado no se encuentra en el sistema, se informará al usuario, como se muestra en la figura 4.18:

The screenshot shows the 'Sistema de Gestión de Tickets' interface. At the top, there are logos for 'Facultad de Ingeniería' and 'UNIDAD DE CÓMPUTO'. The main heading is 'Recuperación de contraseña'. A red error message box states: 'No fue posible atender tu solicitud. El correo proporcionado no está dado de alta en el sistema.' Below this, a prompt asks the user to enter their email address. A text input field contains 'ejemplo@dominio.com'. There are two buttons: 'Aceptar' (green) and 'Regresar' (blue). At the bottom, there is a footer with copyright information for 2023, contact details for the 'Unidad de Cómputo' at UNAM, and a 'Créditos' link.

Figura 4.18: Validación para correo no registrado en el sistema

Ingresado que sea un correo registrado en el SGT, se mostrará una vista con el aviso de que las instrucciones han sido enviadas y se recibirá el correo que contiene la figura 4.19 que a continuación se despliega:

The screenshot shows an email template from 'UNIDAD DE CÓMPUTO' for the 'Sistema de Gestión de Tickets'. The recipient is 'Estimad@ Alejandro Segura:'. The message states: 'Estás recibiendo este correo porque solicitaste un reinicio de contraseña en tu cuenta del Sistema de Gestión de Tickets.' It asks the user to click a link to reset their password. A large black button with white text says 'Reiniciar contraseña'. Below the button, it notes that the link is single-use and provides a link to the system site. A footer contains copyright information for 2023 and contact details for the 'Unidad de Cómputo' at UNAM.

Figura 4.19: Correo electrónico con instrucciones para cambio de contraseña

Acto seguido, el usuario deberá dar clic en “Reiniciar contraseña” para ingresar precisamente una nueva contraseña. En caso de que la nueva contraseña cumpla con las validaciones mostradas al momento de ingresarla, se le notificará que el reinicio fue exitoso.

También es posible realizar un *Cambio de Contraseña* desde el SGT dando clic en el nombre del usuario, seleccionando del submenú desplegado la opción “Cambiar contraseña”, a partir de lo cual se mostrará una vista similar a la de restablecimiento de contraseña, como se observa en la figura 4.20, con la salvedad de que en esta ocasión por cuestiones de seguridad se está solicitando la contraseña actual para validar que el usuario sea quien auténticamente está solicitando este cambio de contraseña.

**Sistema de Gestión de Tickets**

[Solicitudes](#)
[Asignación](#)
[Atención](#)
[Seguimiento](#)
[Foros](#)
[Reportes](#)
[Gestión de catálogos](#)
[Angel](#)

### Cambio de contraseña

Ingrese su contraseña anterior, por razones de seguridad, y luego ingrese su nueva contraseña dos veces para que podamos verificar que la ingresó correctamente.

Contraseña antigua\*

Contraseña nueva\*

- Su contraseña no puede ser muy similar a su otra información personal.
- Su contraseña es muy corta. Debe contener al menos 8 caracteres.
- Su contraseña no puede ser una contraseña de uso común.
- Su contraseña no puede ser completamente numérica.

Confirmación de contraseña nueva\*

[Cambiar mi contraseña](#)

© 2023 División de Ingenierías Civil y Geomática | UNAM, todos los derechos reservados  
 Unidad de Cómputo, Edificio R Conjunto Sur, FI UNAM Tel: 5622-8005 ext. 1037  
 Contacto: [computo.dicyg@gmail.com](mailto:computo.dicyg@gmail.com)

[Créditos](#)

Figura 4.20: Vista para cambiar de contraseña en el SGT

#### 4.4.3. Conexión con el Sistema de Inventarios

Para poder visualizar los dispositivos que se encuentran registrados en el Sistema de Inventarios, en el submenú se seleccionará, con el nombre del usuario en sesión, la opción

**Mis Datos.** En esta vista -que se muestra en la figura 4.21-, se obtiene la información con la cual se encuentra registrado el usuario, así como todos los dispositivos activos que se consultan de la base de datos del Sistema de Inventarios al momento de solicitar esta información.

**Sistema de Gestión de Tickets**

Facultad de Ingeniería

UNIDAD DE CÓMPUTO

Soluciones Asignación Atención Seguimiento Foros Reportes Gestión de catálogos Tanya Itzel

### Datos registrados

# Trabajador/Cuenta	Nombre completo	Correo electrónico	Teléfono	Ocupación	Departamento
836917	Tanya Itzel Arteaga	tanya@fi-b.unam.mx		Académico	Cómputo

### Dispositivos registrados en el Sistema de Inventarios

# Inventario	Tipo	Marca	Modelo	# Serie	Ubicación	Estado del equipo
0	Mouse	0	QCPASS	110607380	0	0
0	CPU	DELL	Optiplex 740	H7ZR2D1	0	0
0	CPU	GATEWAY	GT4222m	CSX7AE1000587	R1-36	Bueno
0	CPU	DELL	Optiplex 740	0	0	0
0	CPU	COOLMASTER	0	RC912KKN1117380	0	0
0	CPU	FENIX	HP Compaq	0	0	0
0	Otro	PANDUIT	Cable Management	0	0	0
0	Otro	PANDUIT	Cable Management	0	0	0
0	Servidor	SUN MICROSYSTEMS	SUNFIRE V100	0	0	0
0	Otro	BEIDEN	GIGAFLEXP56+	0	0	0

« 1 2 3 ... 44 »

Figura 4.21: Vista con datos y dispositivos inventariados del usuario en sesión

Esta conexión del SGT con el Sistema de Inventarios se realiza desde el propio Sistema de Gestión de Tickets -de manera directa- en el DBMS del Sistema de Inventarios a través de una habilitación de comunicación bidireccional entre ambos servidores que alojan a los sistemas y con permisos al usuario de sólo lectura para autenticarse externamente desde la IP local del SGT. No obsta lo anterior para precisar que, el nombre del usuario en sesión que posibilita esta vista de todos los dispositivos activos se obtiene de la autorización que el SGT le concede al usuario identificado, autenticación otorgada por los datos previamente registrados en el sistema (correo electrónico y contraseña).

Adicional a la vista mostrada en la anterior figura 4.21, cuando el usuario Solicitante ingresa al menú **Solicitudes** pidiendo Soporte Técnico, podemos observar unicidad de los números de inventario de cada dispositivo disponible de ser asociado a una solicitud, como se muestra en la siguiente figura 4.22:

The screenshot displays the 'Sistema de Gestión de Tickets' interface. At the top, there are logos for 'Facultad de Ingeniería' and 'UNIDAD DE COMPUTO'. Below the logos is a navigation menu with buttons for 'Solicitudes', 'Asignación', 'Atención', 'Seguimiento', 'Foros', 'Reportes', 'Gestión de catálogos', and 'Angel'. A notification bar states: 'Recuerde, antes de solicitar ayuda, revise el Foro de Asesoría Tecnológica para ver si su problema ya fue resuelto'. The main section is titled 'Soporte Técnico' and contains the following form fields:

- Dispositivo\*:** Radio buttons for '1211 - Teclado DELL RT7D50' and '2198329 - Monitor LG L1750S'. Below this is a note: 'Si no encuentra su dispositivo y no está en soporte, contacte a [computo.dicyg@gmail.com](mailto:computo.dicyg@gmail.com) para solicitar su soporte'.
- Motivo del soporte\*:** A text input field with the placeholder 'Razón del soporte'.
- Descripción del problema\*:** A text area with the placeholder 'Explique su problema'.
- Accesorios a resguardo:** A text input field with the placeholder 'Si es necesario, especificar accesorios'.

At the bottom of the form, there are two buttons: 'Solicitar Soporte' (green) and 'Limpiar' (red). A note indicates 'Los campos obligatorios se indican con \*'. The footer contains copyright information: '© 2023 División de Ingenierías Civil y Geomática | UNAM, todos los derechos reservados. Unidad de Cómputo, Edificio R Conjunto Sur, FI UNAM Tel: 5622-8005 ext. 1037. Contacto: [computo.dicyg@gmail.com](mailto:computo.dicyg@gmail.com)' and a 'Créditos' link.

Figura 4.22: Formulario de solicitud de Soporte Técnico

#### 4.4.4. Rechazo solución propuesta

Una vez que se informa al usuario Solicitante que la atención de su ticket fue concluida, en la vista *Seguimiento* existe la posibilidad de “Invalidar Solución”, en cuyo caso se muestra el formulario de la figura 4.23 que deberá requisitarse para realizar este rechazo:

The screenshot displays the 'Sistema de Gestión de Tickets' interface for rejecting a solution. At the top, there are logos for 'Facultad de Ingeniería' and 'UNIDAD DE COMPUTO'. Below the logos is a navigation menu with buttons for 'Solicitudes', 'Seguimiento', 'Foros', and 'Rosa'. The main section is titled 'Rechazo de la solución del Ticket T-003' and contains the following form fields:

- Motivo de rechazo\*:** A text input field with the placeholder 'Razón del rechazo de la solución propuesta'.
- Ingrese notas/comentarios para que el técnico solucione correctamente la atención:** A text area with the placeholder 'Ingresa razones por la cual se rechaza la solución propuesta'.

At the bottom of the form, there are three buttons: 'Rechazar la solución' (green), 'Limpiar' (blue), and 'Regresar' (red). The footer contains copyright information: '© 2023 División de Ingenierías Civil y Geomática | UNAM, todos los derechos reservados. Unidad de Cómputo, Edificio R Conjunto Sur, FI UNAM Tel: 5622-8005 ext. 1037. Contacto: [computo.dicyg@gmail.com](mailto:computo.dicyg@gmail.com)' and a 'Créditos' link.

Figura 4.23: Formulario para rechazar solución propuesta

Con esto se informa al técnico que atendió la solicitud del usuario que su solución fue rechazada, en cuyo caso el estatus de esta solicitud es de “Rechazado”. Para el técnico que realizó esta solicitud, se envía un correo como el mostrado en la figura 4.24 para ser informado sobre este cambio en el estatus de su atención.

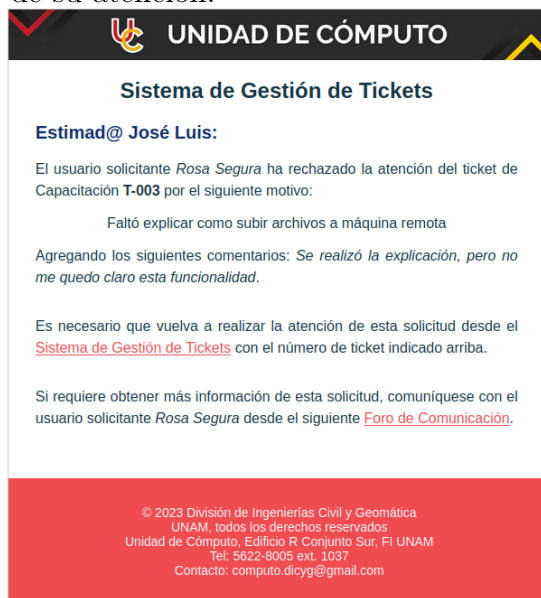


Figura 4.24: Notificación por correo electrónico sobre rechazo a solución propuesta

#### 4.4.5. Ticket de Mantenimiento Programado

El rol Administrador Ejecutivo es quien podrá seleccionar del menú Solicitudes la opción “Solicitar Mantenimiento Programado”. En la vista *Mantenimiento Programado*, si se selecciona un laboratorio que no cuente con dispositivos inventariados, el sistema mostrará la siguiente advertencia de la figura 4.25, indicando que no es posible levantar este tipo de ticket:

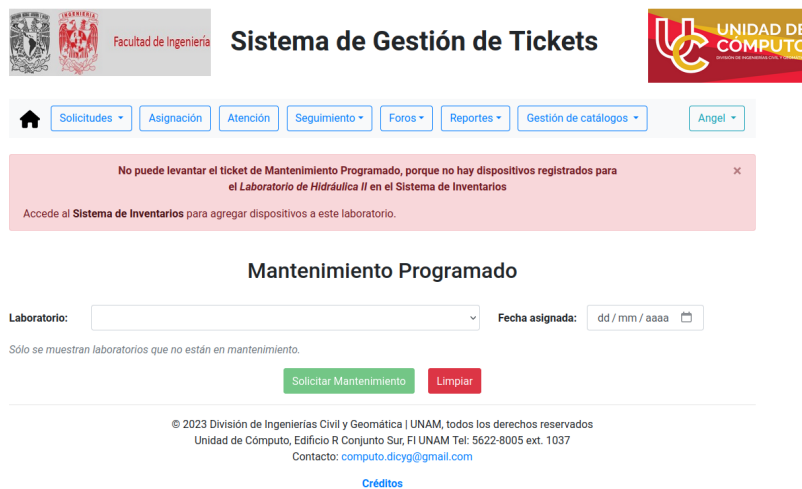


Figura 4.25: Validación para laboratorios sin dispositivos inventariados

Cabe puntualizar que todo el ciclo de vida de este tipo de ticket se tratará como uno solo en los módulos de Asignación, Seguimiento, Verificación (técnica y administrativa) y Validación. Ello podemos observarlo en la figura 4.26, en donde la asignación se realiza para cada soporte hijo, aunque se trate como si fuera uno solo:

**Sistema de Gestión de Tickets**

Facultad de Ingeniería

UNIDAD DE CÓMPUTO  
DIVISION DE INGENIERIAS CIVIL Y GEOMÁTICA

🏠 Solicitudes ▾ Asignación Atención Seguimiento ▾ Foros ▾ Reportes ▾ Gestión de catálogos ▾ Angel ▾

### Asignación del Ticket M-004

**Ingrese los siguientes datos:**

Responsable del mantenimiento:

Por cada dispositivo registrado en este laboratorio, elige a un técnico que realizará su mantenimiento:

Dispositivo 2340994 - CPU HP HP505B Microtower PC: <a href="#">Elegir Técnico</a>
Dispositivo 2340995 - CPU HP HP505B Microtower PC: <a href="#">Elegir Técnico</a>
Dispositivo 2340996 - CPU HP HP505B Microtower PC: <a href="#">Elegir Técnico</a>
Dispositivo 2340997 - CPU HP HP 505B Microtower PC: <a href="#">Elegir Técnico</a>
Dispositivo 2340998 - CPU HP HP505B Microtower PC: <a href="#">Elegir Técnico</a>
Dispositivo 2340999 - CPU HP HP 505B Microtower PC: <a href="#">Elegir Técnico</a>
Dispositivo 2341000 - CPU HP HP 505B Microtower pc: <a href="#">Elegir Técnico</a>
Dispositivo 2341001 - CPU HP HP 505B Microtower PC: <a href="#">Elegir Técnico</a>
Dispositivo 2341002 - CPU HP HP505B Microtower PC: <a href="#">Elegir Técnico</a>
Dispositivo 2341003 - CPU HP HP505B Microtower PC: <a href="#">Elegir Técnico</a>

**Ticket de Mantenimiento**

Fecha de solicitud: 28 de Marzo de 2023 a las 03:34

Usuario: **Angel Brito**

Laboratorio: **Laboratorio de Cómputo Geomático**

Figura 4.26: Asignación de técnicos para Ticket de Mantenimiento Programado

Una vez terminado el desarrollo del sistema, se procede a la liberación del mismo con verificaciones y pruebas necesarias para ponerlo en producción.

# Capítulo 5

## Liberación del Sistema

Una vez concluido el desarrollo local del sistema, se realizaron una serie de pruebas para comprobar que los flujos y procedimientos que se llevan a cabo sean los requeridos por el cliente, a fin de ponerlo en producción, con todas las medidas de seguridad obligadas para poder acceder a él a través de internet.

### 5.1. Pruebas

Probado que fue cada uno de los módulos de manera local, se procedió a implementarlos en el servidor para la realización de pruebas por parte del líder de desarrollo como del propio cliente. Una vez que se concluyeron todos los módulos, se estableció un periodo bimestral de pruebas con la UC, periodo dentro del cual se emitieron comentarios, observaciones y mejoras al sistema, lo que se hace constar en la novena reunión (apéndice A).

El Sistema de Gestión de Tickets se liberó en un ambiente de pruebas, en el cual se crearon los primeros usuarios pertenecientes al staff de la UC, quienes probaron el sistema conforme a sus distintos roles para poder observar el flujo de vida de un ticket en el servidor habilitado para ello. En estas pruebas, los integrantes del staff referenciaron encontrar mensajes imprecisos al momento de ser notificados por correo electrónico; por esta situación se optó por resaltar la información relevante en la notificación que se les envía.

Por otra parte, se encontró un error en la librería `django-crispy-forms`. Este error se debe a que la versión instalada por *pip* es dependiente de otra librería que permite emplear el framework de FrontEnd en este sistema. Para resolver esta problemática se instaló la librería `crispy-bootstrap4`, con la cual fue posible mostrar los formularios que se utilizan en este módulo de Django.

Aunado a ello, al momento de realizar la implementación en el servidor Apache, propor-



cionado por la UC para la puesta en producción del sistema, se produjo la excepción `RuntimeError(populate() isn't reentrant)` provocada por error de configuración con el WSGI. Esta excepción se solucionó chequeando las configuraciones del archivo `wsgi.py` y reiniciando el servicio de Apache.

A continuación se expondrán los tipos básicos de pruebas de software, definiéndolas de manera general [76], seguidas de una explicación de su implementación en el SGT:

### 5.1.1. Unitarias

Estas pruebas consisten en probar métodos y funciones individuales de las clases, componentes o módulos que usa un sistema. Para el caso del SGT, se realizaron en el momento de agregar un nuevo módulo para validar que la entrada y salida de información fuera la correcta; se ejecutaron para modelos, vistas y, en caso de existir, para formularios creados en Python en el archivo `tests.py` de cada carpeta que contiene el módulo a probar.

---

```

1 from django.test import TestCase
2
3 from .models import Usuario
4
5 class UsuarioModelTest(TestCase):
6     @classmethod
7     def setUpTestData(cls):
8         # Configurando objeto que no puede ser modificado en pruebas
9         Usuario.objects.create(usuario_id=0, nombre_usr='Angel',
10            primer_apellido='Brito', num_trabajador=2000,
11            email='ejemplo@dominio.com', password='123456789', )
12
13     def test_nombre_usr_label(self):
14         usuario = Usuario.objects.get(usuario_id=0)
15         field_label = usuario._meta.get_field(
16             'nombre_usr').verbose_name
17         self.assertEqual(field_label, 'nombre')
18
19     def test_primer_apellido_max_length(self):
20         usuario = Usuario.objects.get(usuario_id=0)
21         max_length = usuario._meta.get_field(
22             'primer_apellido').max_length
23         self.assertEqual(max_length, 150)
24
25     def test_nombre_completo(self):
26         usuario = Usuario.objects.get(usuario_id=0)
27         if usuario.segundo_apellido != None:
28             expected_object_name = f'{usuario.nombre_usr} {usuario.
29 primer_apellido} {usuario.segundo_apellido}'
30         else:

```

```

30         expected_object_name = f' {usuario.nombre_usr} {usuario.
    primer_apellido}'
31         self.assertEqual(str(usuario), expected_object_name)

```

---

Extracto de código fuente 5.1: Prueba unitaria para el modelo Usuario

Con el comando `test` que se encuentra en el script `manage.py` se corrían de manera automatizada cada una de este tipo de pruebas realizadas a los módulos del SGT.

### 5.1.2. De integración

En estas pruebas se verifican que los distintos módulos o servicios utilizados por el sistema funcionan bien en conjunto. En el SGT se validó que la interacción de los diferentes módulos no causara ningún tipo de conflicto y que trabajasen de manera conjunta; asimismo, se realizaron pruebas automatizadas para el envío de notificaciones.

---

```

1 from django.test import TestCase
2
3 from django.core import mail
4
5 class EmailTest(TestCase):
6     def test_send_email(self):
7         # Envío de correo electrónico
8         mail.send_mail(
9             "Asunto de Prueba",
10            "Mensaje de prueba",
11            None,
12            ["para@ejemplo.com"],
13            fail_silently=False,
14            html_message = "<h1>Esto es una prueba</h1>"
15        )
16
17        # Probar que un mensaje fue enviado correctamente
18        self.assertEqual(len(mail.outbox), 1)
19
20        # Verificar que el asunto fue enviado al correo de pruebas
21        self.assertEqual(mail.outbox[0].subject, "Asunto de Prueba")

```

---

Extracto de código fuente 5.2: Prueba de integración para el envío de notificaciones por correo electrónico

### 5.1.3. Funcionales

Éstas se centran en los requisitos levantados por el cliente, verificándose el resultado de una acción y no comprueban los estados intermedios del sistema al realizar dicha acción. En el SGT se ocuparon estas pruebas para validar que los requerimientos solicitados fueran

los correctos, relacionándolos con la información que el sistema proporciona a través de las reuniones con el cliente (séptima y octava reunión que constan en el apéndice A) para su aprobación.

#### 5.1.4. De extremo a extremo

Estas pruebas replican el comportamiento de un usuario con el software en un entorno de aplicación completo. Además, verifican que diversos flujos de usuario funcionen según lo previsto. Cargar una página web, iniciar sesión y la verificación de notificaciones de correo electrónico fueron algunas de las pruebas que se realizaron al SGT, validaciones que se muestran en la sección 4.4 del capítulo 4.

#### 5.1.5. De regresión

Estas pruebas se utilizan para constatar que los cambios más recientes -relacionados con el código fuente del sistema- no hayan afectado las funciones existentes de manera adversa. En el SGT se volvieron a ejecutar los casos de prueba que salieron erróneos antes de aplicar las correcciones correspondientes para confirmar que el sistema funcionara correctamente con las mejoras implementadas, como fue el caso del registro de un trabajador de la UC cuando, al indicar en la casilla de verificación esta condición y guardar los datos de registro de usuario nuevo, se mostraba el siguiente error (figura 5.1):



Figura 5.1: Error al registrar trabajador de la UC

Se corrigió el error provocado por las sesiones entre las vistas: implementados los cambios a nivel código del módulo *usuarios*, se corrió de nueva cuenta el caso de prueba y se solucionó el problema al mostrarse la siguiente vista de la figura 5.2:

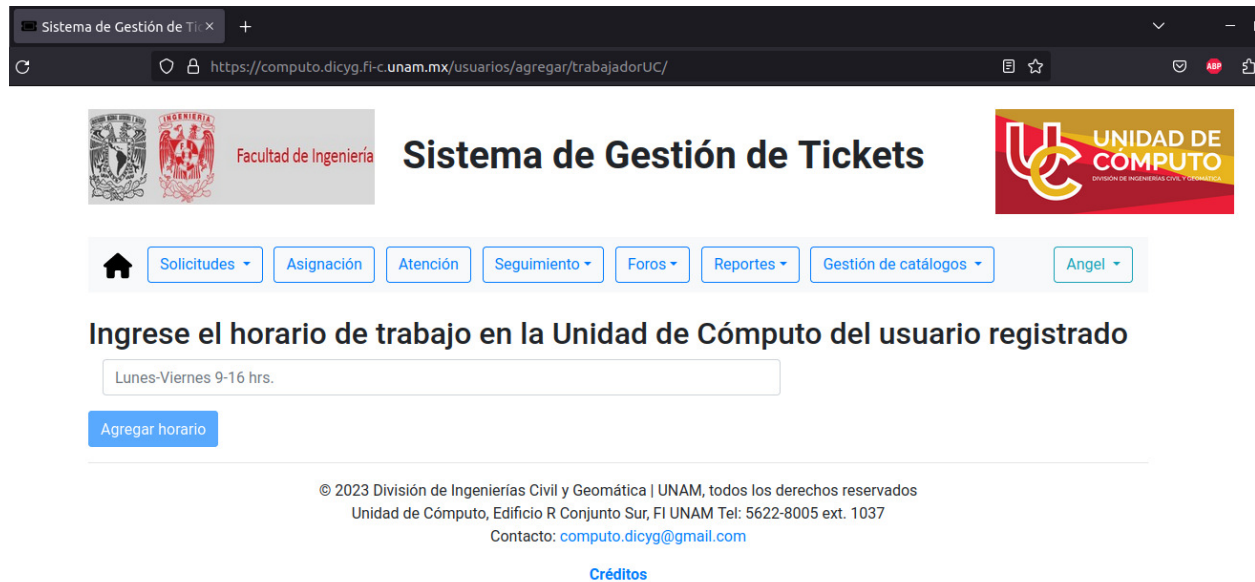


Figura 5.2: Vista para registrar horario de trabajador de la UC

Otras pruebas ejecutadas fueron de rendimiento del SGT para poder medir la fiabilidad, velocidad, escalabilidad y capacidad de respuesta del sistema con lo que se logra una tolerancia a fallas y contar con una alta disponibilidad del sitio web.

Puntualizar que lo ideal es realizar pruebas automatizadas en todos los sistemas para garantizar la calidad de los mismos y una integración continua de nuevas características a implementar sin que se comprometa la disponibilidad del servicio. Sin embargo, si bien es cierto que todos estos tipos de pruebas deben ser automatizadas para que puedan ser ejecutadas  $N$  veces, también lo es el hecho de que debido al tamaño del presente proyecto, al número limitado de desarrolladores (uno) y la falta de recursos asignados, las pruebas implementadas en el SGT se enfocaron más a la propia usabilidad y funcionalidad del mismo. En un futuro, para una nueva versión pueden utilizarse herramientas de pruebas más robustas como *Coverage*, *Selenium*, *CircleCI*, *Semaphore*, solo por mencionar algunas, que facilite a los desarrolladores integrar código nuevo de calidad en el repositorio donde se almacena el código fuente y, una vez que las pruebas sean superadas de forma correcta, al servidor en producción [77].

Así fue que en algunas de las pruebas realizadas al SGT, debe decirse con toda franqueza, no se obtuvo el resultado correcto, situación que condujo a corregir aquellas inconsistencias

o errores, como lo fue el caso de “Recurso no Encontrado” por no especificar que el usuario es un trabajador de la UC; o bien, el caso presentado cuando al momento de registrar los roles de cada usuario no se permitía mostrar un rol único en el sistema; solo por citar alguno de los errores presentados al momento de realizar dichas pruebas cualitativas. Lo que conlleva a la creación de un sistema robusto y con tolerancia a fallas, apto para una puesta en producción por estar constituido por módulos completamente funcionales.

## 5.2. Puesta en producción

Para realizar la implementación de cada uno de los módulos probados en el servidor de la UC, es indispensable contar con una comunicación habilitada por SSH que posibilite remotamente la conexión a esta máquina virtual, así como el protocolo SFTP para subir al servidor remoto los archivos generados en el desarrollo local a fin de ser utilizados y modificados directamente en la interfaz de línea de comandos.

Establecida la comunicación con el servidor, es necesario contar con el software instalado, el cual ha quedado debidamente detallado en el capítulo 4, además de una completa administración con el usuario dado por la UC. Adicionalmente, es necesario instalar el paquete de herramientas de desarrollo para el cliente de MySQL que se está utilizando para realizar las consultas al Sistema de Inventarios<sup>8</sup>, el módulo WSGI para Apache y la herramienta pip para crear un entorno virtual en el que se aislen todos los módulos que requerirá el sistema y, con ello, no afectar el funcionamiento global de Python en la máquina virtual.

Una vez creado el entorno virtual de Python y subidos que han sido los códigos fuente y demás archivos para el correcto funcionamiento del SGT, es necesario editar el archivo de configuración de Apache, indicándole dónde se alojará la ruta absoluta de los archivos estáticos -la que se encargará de servir-, así como la ruta donde se encuentran las configuraciones de Django para el módulo WSGI, la ubicación del entorno virtual y de todas las librerías que utilizarán la aplicación del SGT.

Existen ciertas configuraciones de seguridad que nos sugiere Django realizar en el archivo de configuración **settings.py**, siendo las más relevantes la de habilitar solo los dispositivos permitidos para comunicarse con este sistema; proteger el manejo de datos sensibles, como lo son los accesos para las bases de datos con archivos en otras rutas a fin de mantener resguardadas dichas conexiones; habilitar los registros de seguimiento de errores a través de

---

<sup>8</sup>La conexión directa a la base de datos del Sistema de Inventarios no es la mejor forma de comunicación ya que genera una dependencia fuerte del SGT con él (en una futura versión se emplearán soluciones más adecuadas, tales como manejo de servicios REST, *WebSockets* o replicación de información para contar con mayor accesibilidad y alta disponibilidad de los datos que utiliza el SGT para levantar los tickets de soporte técnico).

un archivo de logs para poder encontrar errores u otros problemas que se generen en un futuro sobre el sistema; la creación de *cron jobs* para procesos que requieren que el servidor notifique o realice tareas periódicas; así como la creación de plantillas para los distintos tipos de errores que pueden surgir al estar utilizando el sistema; lo anterior, con independencia de la implementación del certificado SSL para habilitar que toda la comunicación solo se realice por el protocolo seguro de HTTPS.

Concluidas las configuraciones en los archivos del proyecto de Django, es posible llevar a cabo una verificación automática de estas recomendaciones que da el framework con el comando `check` que se encuentra en el script `manage.py`, especificando -como argumento de la función- que se realizará un despliegue en producción (*deploy*).

### 5.3. Documentación para futuros mantenimientos

Sabido es que todo sistema requiere de mantenimiento para estar a la vanguardia de las tecnologías emergentes, así como de los cambios que en este caso de manera particular se presenten en el propio tratamiento de las solicitudes de soporte técnico, por lo que al finalizar la puesta en producción y el periodo de pruebas se entregaron al cliente los manuales técnico de desarrollo y administrativo para realizar futuras mejoras al sistema.

El primero de ellos describe, a groso modo, como se fueron implementando cada uno de los módulos y el patrón Modelo-Vista-Plantilla con lo que futuros desarrolladores podrán conocer la lógica de negocio implementada en cada aplicación creada por Django.

El segundo se trata de un manual más operativo, dado que describe la serie de pasos a seguir para poder administrar y solucionar problemas que se presenten en el manejo del SGT con nuevas entradas al sistema, o bien, el seguimiento a errores provocados en tiempo real que se generen al momento de realizar cambios en sus módulos.

También se realizó la entrega del manual de usuario para describir cada rol del sistema y conocer el alcance de sus funcionalidades y de dos trípticos para el caso de usuario *Solicitante* y *Encargado de Laboratorio* que contienen las acciones que pueden llevar a cabo estos roles dentro del SGT.

Es así que, tras la liberación del sistema es procedente y, más aún, resulta pertinente presentar los resultados de la implementación del SGT, específicamente, por lo que respecta a la atención de las propias solicitudes de soporte técnico.

# Capítulo 6

## Resultados

Realizado el despliegue del sistema en producción, llegamos prácticamente a la parte final del presente proyecto que implicó una labor de investigación, mediante la cual se fue procesando toda la información recabada que, sin lugar a dudas, se fue recogiendo puntualmente en todas y cada una de las etapas del Ciclo de Vida del Desarrollo de Software (SDLC) hasta derivar en un ambiente con usuarios reales de la DICyG. Por lo que ahora es momento de exponer los resultados obtenidos tras la liberación del SGT que nos lleve a la total comprensión de la problemática por la cual fue creado, pero que también nos permita comprender las alternativas de solución con las que ahora se cuenta, analizando la interacción de los usuarios con el SGT, sus propias funcionalidades y operatividad; lo que constituirá la piedra angular para la futura presentación de conclusiones y así conocer si el sistema cumple con su objetivo de ser accesible a cualquier persona que carece de conocimientos técnicos y que solicitan apoyo para que con los equipos de cómputo que tienen asignados puedan realizar sus actividades sin ningún contratiempo.

### 6.1. Experiencia de Usuario en el SGT

También conocido como UX (por sus siglas en inglés), la Experiencia de Usuario se traduce en un conjunto de factores y elementos relacionados con el proceso de interacción del usuario final respecto a un producto o servicio [78]. Para el caso del SGT, se trata de la interacción de los usuarios con el sitio web, accesible desde cualquier parte del mundo, a través de su dominio registrado por la UC. Si queremos conocer si los flujos de interacción a las funcionalidades del SGT son los más óptimos para el comportamiento del usuario final, se realiza un análisis de sus pilares en UX. Se exponen a continuación:

### **6.1.1. Estética**

En el SGT se tiene una atractiva presentación visual en la que se privilegian los botones para un manejo -sobresaliente y uniforme- de los distintos módulos, donde los usuarios pueden localizar fácilmente el módulo de su interés. Por otra parte, en el envío de las notificaciones por email, al igual que en los módulos, utiliza los colores institucionales de la UC de la DICyG. Cabe destacar que las vistas son las encargadas de “realizar la labor” de un muestreo agradable, coherente y llamativo para atrapar la atención del usuario.

### **6.1.2. Funcionalidad**

Cada uno de los módulos fue probado con antelación y se cuenta con la firme convicción de que en todo momento las nuevas implementaciones agreguen utilidad al SGT para conservar lo altamente funcional de su estatus actual.

### **6.1.3. Accesibilidad**

Es inherente al propio SGT donde el usuario, desde cualquier dispositivo electrónico con conexión a internet, puede ingresar a él con facilidad. La visualización -producto del diseño responsivo del sitio web- le permite navegar sin dificultad para el levantamiento (y un puntual seguimiento) de las solicitudes de soporte técnico.

### **6.1.4. Credibilidad**

Para el caso del SGT, se busca que los módulos entregados al cierre del proyecto no contengan errores y que el área visual provista del sitio web muestre la salida y permita la entrada de datos sin problema alguno en los procesos que realiza el sistema, que trascienda en la generación de confianza en cada uno de los usuarios.

### **6.1.5. Intuitividad**

Se busca que el usuario pueda acceder fácilmente a los módulos implementados y que, cuando le sean mostrados los botones correspondientes a cada sección, también los pueda operar de manera directa e inmediata; es decir, que este fácil entendimiento de la interfaz le genere inclusive la seguridad de que puede cerrar acciones relacionadas con actividades que no puede o no desee realizar.

## **6.2. Procesos automatizados creados por el SGT**

La creación del SGT nos lleva a la total certeza de afirmar que ya no es necesaria la realización “a mano” de una serie de actividades que actualmente se llevan a cabo en la UC,



toda vez que el sistema genera toda la gestión, ejemplo de ello es el ticket de Mantenimiento Programado. Aún más, gracias a este software para este tipo de soporte se generan una serie de notificaciones mediante las cuales se mantienen enterados a todos los implicados en este soporte sobre el seguimiento del estatus de las solicitudes levantadas. También se obtiene una gestión integral en todo el ciclo de vida del ticket que antes no se realizaba y que, incluso, no se conocían las dimensiones de acción que todo el sistema fuera capaz de contener. Lo anterior permite ejecutar cargas masivas a la base de datos con las cuales se obtiene una actualización de los datos mostrados en el sistema en un lapso muy corto de tiempo.

Particularmente los alcances de la liberación del sistema nos llevan a establecer que es posible en un futuro implementar tecnologías emergentes como en este momento lo son las aplicaciones nativas en la nube, microservicios y uso de contenedores para tener un sistema esclable, tolerante a fallos, y con gran adaptabilidad a nuevos frameworks para la mejora de las vistas mostradas al usuario final.

### 6.3. Entrega del SGT a la Unidad de Cómputo

La liberación del SGT se realizó en dos partes: la primera de ellas de manera interna para ser probado por los distintos miembros del staff de la UC; en la segunda, se integró a los usuarios solicitantes y demás personal administrativo de la DICyG que estará involucrado en la gestión de alguna etapa del ciclo de vida del ticket.

El SGT se entrega funcionando en el servidor de la máquina virtual y con la posibilidad de ser implementado en cualquier otro ambiente productivo (portabilidad). También se realiza la entrega del manual técnico de desarrollo y manual administrativo para poder continuar con las mejoras y creación de nuevos módulos, así como del acta de cierre del proyecto para informar a la Unidad de Cómputo de la DICyG las actividades que se llevaron a cabo en la creación SGT y los resultados obtenidos con la puesta en producción de este software.

Por considerarlos muy puntuales y significativos para la experiencia de usuario, se entregaron, adicionalmente al manual de usuario que integra cada rol del sistema, trípticos para los usuarios con rol Solicitante y Encargado de Laboratorio, los cuales consagran los aspectos más importantes a seguir en determinados procesos que integran el SGT.

### 6.4. Costos del proyecto

Una estimación de costos se realiza a continuación para poner en perspectiva la proyección de las erogaciones de recursos -lo gastado- del proyecto, contemplando que la implementación

de este sistema abarcó 6 meses efectivos de trabajo (traducidos en jornadas de 8 horas diarias, 5 días a la semana), atendiendo a la información contenida en la siguiente tabla 6.1:

<b>Concepto</b>	<b>Gasto Unitario</b>	<b>Gasto Total</b>
Pago de servicios del proveedor de electricidad e internet	\$500 por mes	\$3 000
Desarrollador Full Stack <sup>9</sup>	\$27 000 por mes <sup>10</sup>	\$162 000

Tabla 6.1: Gastos del SGT

De acuerdo a lo mostrado, el gasto que generó este sistema fue de \$165 000, lo cual solo refleja lo que el líder de desarrollo reportó como gasto; sin embargo, ello no obsta para contemplar una adicional estimación con respecto al servidor en donde se hospeda el SGT en producción y que, desde luego, genera otro gasto adicional, al igual que el registro y manejo del dominio del sitio web a través del protocolo seguro HTTPS, los cuales corrieron por parte del cliente.

<sup>9</sup>Conocimientos intermedio de los frameworks Django y Bootstrap

<sup>10</sup>De acuerdo al sitio [Talent](#), consultado el día 01/04/2023

# Capítulo 7

## Conclusiones y Trabajo Futuro

Sin lugar a dudas, los procesos informáticos nos acompañan en nuestro día a día en un sinfín de actividades; más aún, si éstas están relacionadas con la academia, la investigación y el desarrollo profesional. Por ello, el hecho de que en la Unidad de Cómputo (UC) de la División de Ingenierías Civil y Geomática (DICyG) de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México (UNAM) se proporcione soporte técnico a su personal sin la eficiencia que hoy en día exige la vorágine tecnológica en la que nos encontramos inmersos, condujo a la necesidad de implementar un sistema automatizado de gestión de tickets para soporte técnico a la altura del público usuario, que optimizara, además, la encomiable labor que realizan los empleados de esta unidad administrativa.

La creación de un sistema central que permitiera entablar una constante comunicación con cada uno de los usuarios que requieren solicitar soporte técnico de la UC de la DICyG, resolver a cabalidad todos y cada uno de los problemas que presentan sus equipos de cómputo a fin de contar con herramientas de trabajo en buen estado que les faciliten el desempeño de sus funciones, así como mantener los registros íntegros -con su correcta documentación- de la atención de todas y cada una de las solicitudes (tickets) que se demandan, sentaron las bases para ello.

El Sistema de Gestión de Tickets representa un producto pionero al interconectarse con otro sistema de la propia Unidad de Cómputo de la DICyG, el Sistema de Inventarios, el cual contiene el registro de todos los equipos electrónicos de esta División y cuyos números de inventario constituyen la piedra angular del propio proceso de levantamiento del ticket de soporte técnico y, por ende, del histórico y demás reportes que documentan todas y cada una de las acciones inherentes al mismo.

Los componentes de software que integran el Sistema de Gestión de Tickets son:

- ⊕ Los Frameworks de BackEnd y FrontEnd, Django 3.2 y Bootstrap 4, respectivamente. El primero de ellos, para operaciones de creación, lectura, actualización y eliminación en la base de datos del propio SGT, y de sólo lectura a la base de datos del Sistema de Inventarios administrado por la UC; en tanto que el segundo, para un muestreo amigable de las vistas dadas por Django y una mejor presentación de la información contenida en el sistema.
- ⊕ Los DBMS, PostgreSQL 13 y MySQL 5.5, bases de datos en las que, en el caso del primero de ellos, almacenará toda la información generada por el SGT; mientras que el segundo es la del Sistema de Inventarios de la UC, la cual será de sólo lectura para poder consultar los dispositivos inventariados.
- ⊕ Servidor Apache 2.4 que, en conjunto con el módulo WSGI, permite transformar el código de Python en peticiones HTTP. A este respecto resulta conveniente reiterar que se seguirá el estándar PEP 3333 que detalla la especificación de interfaz para la comunicación entre aplicaciones web hechas en este lenguaje de programación y cualquier servidor.

La tarea no fue sencilla: crear un software a la medida adaptando estos componentes requirió un estudio a profundidad de cada una de estas herramientas para lograr una correcta integración. La labor de planeación, análisis, diseño, desarrollo y la propia liberación del sistema, amén de las horas-hombre empleadas en la propia implementación del proyecto, representó la suma de esfuerzos, acuerdos y compromisos para su conclusión exitosa.

Así fue que cada una de las etapas del SDLC (Ciclo de Vida del Desarrollo de Software, por sus siglas en inglés) originaron una serie de retos y, por qué no decirlo, de obstáculos que se fueron sorteando y que vale la pena precisarlos, tales como el hecho de que al momento de la planeación aún no se tenía definido el tipo de usuario que se habilitaría para levantar solicitudes de soporte técnico de los dispositivos registrados en el Sistema de Inventarios, lo cual impedía el flujo de desarrollo para esta funcionalidad del sistema. Por otra parte, velar por la facilidad con la que cualquier usuario pudiera acceder al sistema para levantar sus solicitudes de soporte técnico y/o de capacitación desde cualquier navegador web, se convirtió en uno de los principales desafíos al momento de realizar el desarrollo conceptual del sistema.

Ahora bien, en el diseño de la base de datos del SGT se trabajó arduamente en evitar la redundancia de datos y complejidad al momento de crear los modelos entidad-relación para obtener un menor procesamiento al realizar las consultas. Así mismo, otro de los obstáculos presentados -ahora en la etapa del desarrollo del sistema- fue el manejo de las excepciones al ejecutar las consultas al Sistema de Inventarios, lo que conllevaba a errores de unicidad en campos que relacionan al SGT con este último.

Finalmente, en la implementación del sistema en producción se encontró una falta de comunicación con la base de datos del Sistema de Inventarios (también en producción) y la máquina virtual que hospeda el SGT, lo cual se resolvió permitiendo la comunicación bidireccional a través de la red local.

A pesar de los obstáculos, también debe decirse que existieron herramientas y métodos que facilitaron el desarrollo del presente trabajo. El Lenguaje de Modelado Unificado, fue uno de ellos, el cual facilitó una mejor representación visual del SGT. Mención especial merece la *flexibilidad* en la gestión del presente proyecto por su importancia al momento de atender los nuevos requerimientos o cambios solicitados por el cliente derivados de las entregas de cada iteración del ciclo de trabajo. Ahora bien, la misma cobra mayor relevancia si consideramos ya la puesta en producción del SGT por la inexorable necesidad de “ajustarlo” a una realidad de cambios continuos, motivados por la automatización que exige la agilización de los procesos para garantizar la mejor de las eficiencias operativas. Además, tratándose de cuestiones de seguridad y transferencia de datos -dado que se trata de una interconexión entre dos sistemas totalmente diferentes, SGT y el Sistema de Inventarios-, la definición de roles facilita la toma de decisiones para negar accesos o para decidir la cantidad de datos que se transmiten de un software a otro.

Sirva todo lo anterior para establecer que gracias al trabajo conjunto en la planificación del presente proyecto de tesis, a la obtención de los datos requeridos para realizarlo y al análisis de la información proporcionada por la UC de la DICyG, la entonces premisa de que se constituiría una central digital de atención a usuarios disponible desde cualquier punto para los solicitantes que requirieran apoyo técnico para resolver la problemática que presentasen sus equipos de cómputo, una vez que se ha contrastado la experiencia obtenida en el desarrollo de este software con el propósito de creación de un sistema web que llevase el control y seguimiento de los tickets para soporte técnico realizados por la UC de la DICyG, nos lleva a **confirmar la hipótesis** establecida en un inicio.

El Sistema de Gestión de Tickets constituye ya una realidad, al igual que lo es la *accesibilidad* lograda con la instauración de este sistema web, el cual está provisto precisamente de una interfaz intuitiva donde el usuario encuentra oportunamente los botones de interacción que le permiten realizar las acciones asociadas y disponibles para cada rol al ingresar a la dirección del propio portal web en el preciso instante que requiere solicitar soporte técnico; *por lo que el propósito medular de lograr una atención eficaz y eficiente se haya debidamente cumplido.*

El hecho de que el Sistema de Gestión de Tickets esté *integrado por módulos* -con su correspondiente ilación- es atribuible al propósito de mejorar tanto la atención de los soportes técnicos que se realizan en esta unidad de cómputo, como su aparejado seguimiento y verificación. El SGT trascenderá como una central de administración del ciclo de vida de los tickets que atiende la UC. Lo anterior es así porque:

- ✓ Con el SGT se mejorará la atención de los soportes técnicos y se reducirán los tiempos de atención del mismo, toda vez que la definición de roles en el sistema permite una eficiente asignación y coordinación de las tareas de soporte técnico para que cada una de ellas se realicen con la eficacia que demanda la propia problemática que de su solicitud emana. Dado que el ticket se asignará respetando un reparto equitativo de las actividades realizadas por cada técnico en la atención de soportes, el empeño y dedicación de los miembros del staff se focalizará, sí en la resolución de sus propios tickets asignados, pero también esta optimización lograda con el sistema contribuirá al trabajo colaborativo que en todo equipo de trabajo debe imperar.
- ✓ Con el SGT se llevará un seguimiento efectivo de todos los tickets levantados, toda vez que se instrumentaron bases de datos comunicadas entre sí, lo que consecuentemente documentará todas y cada una de las acciones tendientes a la resolución de (también) todos y cada uno los problemas expuestos por el usuario solicitante en relación a los equipos de cómputo de la DICyG, donde prevalecerá la constante comunicación usuario-staff y, desde luego, su injerencia en el flujo de la atención del ticket, con notificaciones en tiempo real de las actualizaciones que al respecto se obtengan hasta su total resolución; lo que permitirá estar informado en todo momento de las acciones realizadas para la atención del ticket y mantener un listado de las preguntas y tareas comunes de sus usuarios. Como valor agregado, esta documentación posibilitará la generación de reportes detallados que muestren todo el trabajo realizado en cada uno de los equipos de cómputo, proveyendo con ello de un histórico de los cambios realizados en cada dispositivo que se encuentre registrado en el Sistema de Inventarios y, a su vez, de las actividades realizadas por cada uno de los colaboradores de la UC sin que se presente “merma” alguna de información, permitiendo -además- la generación de reportes personalizados que concentrarán lo realizado en la Unidad de Cómputo.
- ✓ Con el SGT se instaure una mejora en la verificación de los soportes técnicos, toda vez que la dinámica laboral que se presenta en la UC (por ejemplo, el caso de los estudiantes prestadores de servicio social cuya estancia es temporal; o el caso de los empleados cuyas actividades que realizarán en los soportes técnicos asignados constituyen su primera incursión “al mundo real”), obliga a poner más énfasis en determinados soportes para contribuir a su fiabilidad. Para ello quedaron implementadas en el sistema las funciones del rol “Supervisor”, las que aunadas con la interacción de las demás actividades realizadas en favor del soporte, constituirán un factor determinante para eficientarlo.

Considerando lo narrado en los párrafos precedentes y el binomio de asistencia intelectual-tecnológica que implica todo soporte técnico, es preciso enfatizar que gracias a los módulos completamente funcionales de que consta el SGT, se otorgarán también tutoriales a través de un Foro de Asesoría Tecnológica con base a toda esta gama de conocimientos, información, tecnologías y de las propias actividades técnicas que se destinen para el buen funcionamiento de los equipos de cómputo, para que sean los propios empleados de la DICyG -como usuarios “autosuficientes”- quienes de primera mano cuenten con las herramientas pertinentes que resuelvan “ipso facto” la operatividad de sus ordenadores.

Derivado de todo lo anterior, no debe soslayarse el hecho de que gracias a la manera en que se haya implementado el SGT, su capacidad para manejar una carga creciente de datos se haya garantizada; es decir, la *escalabilidad* presente en este software permitirá satisfacer las necesidades de los usuarios sin afectar su funcionamiento; dicho de otra manera, seguirá manteniendo su rendimiento o eficiencia, aún con el incremento de sus demandas operativas (preciso reiterar que en producción, el SGT podrá recibir mantenimiento y modificaciones en cada uno de sus módulos, de manera independiente, a fin de no afectar la funcionalidad del sistema web).

Además, este sistema cuenta con la capacidad de ser transferido, de forma efectiva y eficiente, a cualquier ambiente productivo proveyendo de una gran adaptabilidad a cualquier servidor. Esta *portabilidad* posibilitaría que el propio sistema pueda ser utilizado en lugar de otro software determinado con el mismo propósito y en el mismo entorno.

Se ha cumplido íntegramente con todos y cada uno de los requerimientos del proyecto. Ello no significa que el sistema no pueda ser susceptible de mejoras o *trabajo futuro* que lo robustezca aún más, realizando los mantenimientos acordes a la realidad tecnológica del momento. En este tenor, se enuncian los siguientes:

- ⇒ Creación de API RESTful del Sistema de Inventarios para realizar las consultas que se utilizan en el Sistema de Gestión de Tickets para eliminar la dependencia directa a la base de datos del Sistema de Inventarios.
- ⇒ Trabajar en el filtrado de los números de inventario repetidos en el Sistema de Inventarios para evitar la duplicidad de los registros de los datos del dispositivo sobre el cual se va a realizar el soporte técnico en el Sistema de Gestión de Tickets.
- ⇒ Instaurar la homologación del usuario final en el Sistema de Inventarios para poder ser consultado desde el Sistema de Gestión de Tickets, de manera más efectiva, con el número de trabajador/cuenta.

- ⇒ Filtrar las actividades ponderadas que se registran para facilitar al staff de la UC la búsqueda inmediata de las actividades que se han realizado en la atención de las solicitudes de soporte técnico.
- ⇒ Agregar procesos asíncronos para que, por un lado, el usuario final obtenga una respuesta más rápida a sus solicitudes y, por el otro, el servidor se encargue de realizar procesos tardados en segundo plano.
- ⇒ Manejar la concurrencia para el caso de alta disponibilidad. Lo anterior, a fin de que sea factible el hecho de que varios usuarios con el mismo rol puedan modificar un recurso sin afectar los cambios realizados por otro usuario en ese mismo instante.
- ⇒ Adicionar a las actividades ponderadas aquellas que se registran en el SGT de manera constante y reiterada en la atención de los tickets.
- ⇒ Poder exportar en formatos CSV, Excel y/o PDF, los reportes generados en el SGT para una mejor manipulación y creación de otro tipo de gráficas necesarias para la administración operativa de la UC.
- ⇒ Programar recordatorios para los usuarios solicitantes que aún no hayan validado la atención de sus tickets, así como al usuario solicitante y técnico en cuestión cuando se indique una fecha programada de atención a la solicitud.
- ⇒ Desconectar al usuario después de minutos de inactividad para evitar problemas con el uso inadecuado de su cuenta.
- ⇒ Añadir, al número de solicitud de levantamiento del ticket, el año en el que se registra con el correspondiente ID incremental que se tiene en el sistema.
- ⇒ Y, desde luego, crear pipelines (canalizaciones/flujos de trabajo) para la integración, entrega y mejora continua (CI/CD por sus siglas en inglés) de este sistema que permitan que los cambios que se suban al repositorio lleven consigo una serie de pruebas, antes de realizar la implementación automática en el servidor y su consecuente puesta en producción.

En suma, el diseño que en un principio fue planteado para el desarrollo del presente software cubre las expectativas de solución para resolver la problemática de soporte técnico que existe en la UC de la DICyG; por lo que con la *Implementación de un Sistema de Gestión de Tickets para Soporte Técnico de la DICyG*, tal y como se planteó en un principio, se abre un solo canal de comunicación entre los usuarios y el propio equipo de trabajo de esta unidad administrativa, lo que redundará en una inmejorable atención de las solicitudes realizadas por los usuarios (quienes serán también los que validarán la calidad de las



mismas), al mismo tiempo que se brindará una resolución óptima de los soportes técnicos levantados, soportes que permitirán eficientar cada una de las actividades que se llevan a cabo, que redundan en un trabajo equitativo-colaborativo, del cual emanarán los reportes que se requieran y su consecuente histórico de los cambios por cada uno de los equipos de cómputo inventariados.

Con todo lo anteriormente expuesto, se puede concluir que:

*El objetivo del presente proyecto de tesis consistente en la factibilidad de creación de un sistema web que llevase el control y seguimiento de la gestión de los tickets para soporte técnico realizada por la UC de la DICyG, ha quedado debidamente cumplimentado.*

# Apéndices

# Apéndice A

## Reuniones

Por cada reunión llevada a cabo con la UC se levantó una minuta, en la cual se realizó puntual registro del objetivo de la misma, de un breve resumen de lo versado en ella, así como de los compromisos y acuerdos logrados, asentándose al calce el nombre de los asistentes y su firma, para dejar constancia secuencial de las actividades comprendidas en las diferentes etapas de desarrollo del proyecto.



---

**Minuta 1**  
**Sistema de Gestión de Tickets**




---



**PROYECTO:** Sistema de Gestión de Tickets  
**ETAPA :** Planeación  
**LUGAR :** A distancia (*Zoom*)

**FECHA:** 24 de agosto de 2021  
**HORA DE INICIO:** 12:00 horas  
**HORA DE TÉRMINO:** 13:10 horas

### **Lista de asistentes:**

-  M.I. Tanya Itzel Arteaga Ricci (*Directora de tesis*)
-  M. en C. Martha Angélica Nakayama Cervantes (*Asesora Técnica*)
-  Angel Brito Segura (*tesista*)

**OBJETIVO**

Conocer los distintos proyectos que se tienen en la División de Ingenierías Civil y Geomática para la elección del proyecto de tesis.

## **ORDEN DEL DÍA**

Explicación -a groso modo- de los siguientes proyectos:

1. Sistema de Finanzas
2. Sistema de Gestión de Tickets
3. Intranet

## **ACTIVIDADES REALIZADAS**

### **1. Sistema de Finanzas**

En la División de Ingenierías Civil y Geomática se realizan pagos a profesores y becas. Se cuenta con un sistema general para la obtención de dichos datos. También, en esta división se tienen ingresos extraordinarios para cuando se realizan proyectos nuevos. Todo está auditado y el control lo realiza Arturo Olvera.

Se requiere un sistema que controle las entradas y salidas de dinero para conocer cómo van los presupuestos de cada unidad de la división y así tener un mejor control en las compras. Este sistema generaría consultas para saber en qué se gasta el dinero.

**NOTA:** Actualmente se cuenta con un sistema incompleto que se puede ocupar o desechar completamente en la implementación del sistema a crear.

### **2. Sistema de Gestión de Tickets**

En la Unidad de Cómputo de la División de Ingenierías Civil y Geomática se tienen equipos de cómputo, en la cual se hace soporte técnico en distintas áreas (actualmente, la solicitud del soporte se hace por llamadas a la M.I. Tanya Itzel Arteaga Ricci).

Se requiere un sistema que comprenda los siguientes parámetros: solicitud del soporte técnico; su reparto equitativo; la creación de un expediente para cada equipo de cómputo registrado en el Sistema de Inventarios; su notificación por correo electrónico; un programa de recompensas acorde a la complejidad del soporte y al desempeño; y la creación de estadísticas para ver la productividad en el área de soporte técnico.



**NOTA:** Acuerdo la selección de este proyecto para el desarrollo de la tesis y propongo implementar una nube híbrida con algún servicio de Azure, generando una página web y una app de Android.

### 3. Intranet

Crear una Intranet en la que se unan varios sistemas de la División de Ingenierías Civil y Geomática. Este sistema se conectará por una VPN y requerirá una base de datos muy robusta para realizar las migraciones correspondientes.

## COMPROMISOS

ANGEL		MAESTRA TANYA	
Actividad	Fecha	Actividad	Fecha
Crear la documentación, tanto del sistema como del desarrollo	24/08/21 y hasta la conclusión del mismo	Agendar reunión con el Director de Sistemas de la División de Ingenierías Civil y Geomática	Próxima reunión
Elegir la metodología ágil del desarrollo	01/09/21		
Crear artefactos para el levantamiento de requerimientos	Próxima reunión		

**PRÓXIMA REUNIÓN: A definir**

FIRMA  
AUTÓGRAFA

Angel Brito Segura

FIRMA  
AUTÓGRAFA

M. en C. Martha Angélica Nakayama Cervantes

FIRMA  
AUTÓGRAFA

M.I. Tanya Itzel Arteaga Ricci



## Minuta 2

### Sistema de Gestión de Tickets



**PROYECTO:** Sistema de Gestión de Tickets  
**ETAPA :** Análisis  
**LUGAR :** A distancia (*Meet*)

**FECHA:** 22 de septiembre de 2021  
**HORA DE INICIO:** 16:30 horas  
**HORA DE TÉRMINO:** 17:45 horas

### Lista de asistentes:

- Angel Brito Segura [*Líder de proyecto – tesista*]
- M.I. Tanya Itzel Arteaga Ricci [*Patrocinadora – directora de tesis*]
- M. en C. Martha Angélica Nakayama Cervantes [*Asesora Técnica*]
- Staff de la Unidad de Cómputo (UC) de la División de Ingenierías Civil y Geomática (DICyG) [*Cliente*]:
  - Alejandra Guzmán Huxley -líder de logística, parte de Front-end (diseñadora)-
  - Jair Camacho -líder técnico del soporte técnico-
  - José Abraham Bonilla Pastor -encargado del soporte técnico-
  - Héctor Martínez Garduño -encargado de soportes de capacitación-
  - Jesús Cuéllar -atención de soportes por whatsapp-
  - Diego Ramírez Romero -atención de soportes por whatsapp-
  - Oscar Alejandro Sánchez Pérez -parte técnica-
  - Miguel Ángel Leyva Bejarano -parte técnica-

#### OBJETIVO

Levantar los primeros requerimientos del proyecto Sistema de Gestión de Tickets (SGT) para su formalización.

## ORDEN DEL DÍA

- Proceso manual actual
- Interacción del staff de la Unidad de Cómputo a cargo del proceso manual
- Ciclo de vida del ticket

## ACTIVIDADES REALIZADAS

### 1. Proceso manual actual

El proceso manual es liderado por Alejandra Guzmán Huxley. Como líder de logística, expuso -a groso modo- cómo los usuarios solicitan el soporte técnico, la asignación que corresponde a cada unidad, así como la ponderación existente en las actividades que lo involucran.

### 2. Interacción del staff de la UC a cargo del proceso manual

Cada área de la Unidad de Cómputo planteó la operación, el manejo y actividades realizadas en el proceso manual en su ámbito de soporte. Además, determinaron y explicaron lo correspondiente a los documentos que se generan en el ticket.

### 3. Ciclo de vida del ticket

Se hizo la especificación de que un ticket está asociado a una solicitud de soporte técnico. Sus etapas son: creación, asignación, resolución, validación y conformidad.



## COMPROMISOS

LÍDER DE PROYECTO		CLIENTE	
Actividad	Fecha	Actividad	Fecha
Crear diagrama del proceso	Próxima reunión	Dar acceso a la documentación referente al proceso manual actual	A la brevedad
Definir los requerimientos iniciales	Próxima reunión	Crear grupo de WhatsApp con Armando y Alejandra para seguimiento del proyecto	A la brevedad

### PRÓXIMA REUNIÓN: A definir

FIRMA  
AUTÓGRAFA

Angel Brito Segura

FIRMA  
AUTÓGRAFA

M. en C. Martha Angélica Nakayama Cervantes

FIRMA  
AUTÓGRAFA

M.I. Tanya Itzel Arteaga Ricci

FIRMA  
AUTÓGRAFA

UC de la DICyG



**Minuta 3**  
Sistema de Gestión de Tickets



**PROYECTO:** Sistema de Gestión de Tickets  
**ETAPA :** Análisis  
**LUGAR :** A distancia (*Meet*)

**FECHA:** 6 de octubre de 2021  
**HORA DE INICIO:** 16:00 horas  
**HORA DE TÉRMINO:** 16:50 horas

**Lista de asistentes:**

- Angel Brito Segura [*Líder de proyecto – tesista*]
- M.I. Tanya Itzel Arteaga Ricci [*Patrocinadora – directora de tesis*]
- Staff de la Unidad de Cómputo (UC) de la División de Ingenierías Civil y Geomática (DICyG) [*Cliente*]

**OBJETIVO** Discutir los diagramas del proceso actual a implementarse en el Sistema de Gestión de Tickets a crear.

**ORDEN DEL DÍA**

- Diagramas del proceso manual actual
- Planteamientos sobre el desarrollo del Sistema de Gestión de Tickets

**ACTIVIDADES REALIZADAS**

**1. Diagramas del proceso manual actual**

Se mostraron los diagramas realizados con la información recibida en la reunión pasada, puntualizando ciertos roles y partes del proceso manual actual a implementar en el Sistema de Gestión de Tickets a crear.

**2. Planteamientos sobre el desarrollo del Sistema de Gestión de Tickets**

Se dio respuesta a las preguntas formuladas para continuar definiendo los requerimientos del proyecto.

**COMPROMISOS**

<b>LÍDER DE PROYECTO</b>		<b>CLIENTE</b>	
Actividad	Fecha	Actividad	Fecha
Adecuar los diagramas con la retroalimentación recibida	11/10/21	Enviar organigrama de la UC de la DICyG	Ala brevedad

**PRÓXIMA REUNIÓN: Por definir**

FIRMA  
AUTÓGRAFA

Angel Brito Segura

FIRMA  
AUTÓGRAFA

M.I. Tanya Itzel Arteaga Ricci

FIRMA  
AUTÓGRAFA

UC de la DICyG





---

**Minuta 4**  
**Sistema de Gestión de Tickets**





---



**PROYECTO:** Sistema de Gestión de Tickets  
**ETAPA :** Diseño  
**LUGAR :** A distancia (*Zoom*)

**FECHA:** 9 de diciembre de 2021  
**HORA DE INICIO:** 15:00 horas  
**HORA DE TÉRMINO:** 16:45 horas

### **Lista de asistentes:**

-  Angel Brito Segura [*Líder de proyecto – tesista*]
-  M.I. Tanya Itzel Arteaga Ricci [*Patrocinadora – directora de tesis*]
-  M. en C. Martha Angélica Nakayama Cervantes [*Asesora Técnica*]
-  Staff de la Unidad de Cómputo (UC) de la División de Ingenierías Civil y Geomática (DICyG) [*Cliente*]

**OBJETIVO**

Mostrar maquetado del Sistema de Gestión de Tickets de próxima creación para exponerle al cliente el flujo a implementar.

## **ORDEN DEL DÍA**

- Sistema de Inventarios
- Maquetado del Sistema de Gestión de Tickets

## **ACTIVIDADES REALIZADAS**

### ***1. Sistema de Inventarios***

Se discutieron algunos aspectos relativos a la información contenida en este sistema: números de inventario y usuarios registrados.

### ***2. Maquetado del Sistema de Gestión de Tickets***

Se mostraron los prototipos correspondientes a los roles de Administrador del Sistema, Solicitante y Líder Técnico\*, en virtud de los cuales se precisaron algunos cambios y se obtuvieron sugerencias, con propuestas de información y acciones a seguir.

\*El maquetado concerniente a los roles restantes se expondrá en una próxima reunión con el staff de la UC de la DICyG.



## COMPROMISOS

LÍDER DE PROYECTO		CLIENTE	
Actividad	Fecha	Actividad	Fecha
Modificar prototipos atento a lo acordado en la reunión	17/12/21	Enviar respaldo de la base de datos del Sistema de Inventarios	11/12/21
		Instalar el software solicitado en el servidor que hospedará al Sistema de Gestión de Tickets	A la brevedad
		Remitir carta descriptiva y acta constitutiva del proyecto firmadas	A la brevedad

### PRÓXIMA REUNIÓN: A definir

FIRMA  
AUTÓGRAFA

Angel Brito Segura

FIRMA  
AUTÓGRAFA

M.I. Tanya Itzel Arteaga Ricci

FIRMA  
AUTÓGRAFA

M. en C. Martha Angélica Nakayama Cervantes

FIRMA  
AUTÓGRAFA

UC de la DICyG



## Minuta 5

Sistema de Gestión de Tickets



**PROYECTO:** Sistema de Gestión de Tickets  
**ETAPA :** Diseño  
**LUGAR :** A distancia (*Meet*)

**FECHA:** 26 de enero de 2022  
**HORA DE INICIO:** 17:15 horas  
**HORA DE TÉRMINO:** 17:55 horas

### Lista de asistentes:

- Angel Brito Segura [*Líder de proyecto – tesista*]
- M.I. Tanya Itzel Arteaga Ricci [*Patrocinadora – directora de tesis*]
- Staff de la Unidad de Cómputo (UC) de la División de Ingenierías Civil y Geomática (DICyG) [*Cliente*]

#### OBJETIVO

Concluir el muestreo del maquetado del Sistema de Gestión de Tickets con especial énfasis en los roles a implementar.

### ORDEN DEL DÍA

Mostrar al cliente el **maquetado del Sistema de Gestión de Tickets** para recibir comentarios y sugerencias que retroalimenten el desarrollo del proyecto.

### ACTIVIDADES REALIZADAS

Se mostraron los prototipos para los roles de Supervisor, Técnico, Administrador Ejecutivo y Encargado del Laboratorio descritos en la carta descriptiva del Sistema de Tickets. La Unidad de Cómputo proporcionó la lista de software y tareas programadas para los laboratorios de la DICyG que realizan mantenimientos; por otra parte, comparte carpeta de Drive para trabajar el manual técnico de desarrollo del Sistema de Gestión de Tickets.

### COMPROMISOS

LÍDER DE PROYECTO		CLIENTE	
Actividad	Fecha	Actividad	Fecha
Entrega del modelo relacional al Ing. Armando Vázquez Díaz para su aprobación y validación	04/02/22	Instalar el software solicitado en el servidor que hospedará al Sistema de Gestión de Tickets	Ala brevedad
Documentar el desarrollo del proyecto en el manual técnico de desarrollo proporcionado por el cliente	Durante todo el proyecto		

#### PRÓXIMA REUNIÓN: A definir

FIRMA  
AUTÓGRAFA

Angel Brito Segura

FIRMA  
AUTÓGRAFA

M.I. Tanya Itzel Arteaga Ricci

FIRMA  
AUTÓGRAFA

UC de la DICyG






**Minuta 6**  
Sistema de Gestión de Tickets



**PROYECTO:** Sistema de Gestión de Tickets  
**ETAPA :** Desarrollo  
**LUGAR :** A distancia (*Meet*)

**FECHA:** 13 de junio de 2022  
**HORA DE INICIO:** 13:30 horas  
**HORA DE TÉRMINO:** 13:50 horas

**Lista de asistentes:**

-  M.I. Tanya Itzel Arteaga Ricci [*Directora de tesis*]
-  Leo Rivera [*Staff Unidad de Cómputo (UC) de la DICyG*]
-  Angel Brito Segura [*Líder de proyecto – tesista*]

**OBJETIVO**

Explicar las herramientas de software que se requieren en el nuevo servidor del Sistema de Gestión de Tickets para que la UC proceda a su instalación.

**ORDEN DEL DÍA**

Solicitar a la Unidad de Cómputo la instalación del software necesario para poder crear el Sistema de Gestión de Tickets en el nuevo servidor.

**ACTIVIDADES REALIZADAS**

Se le explicó el Sistema de Gestión de Tickets, a groso modo, al prestador de servicio social de la UC, Leo Rivera, con el propósito de que conociera los alcances del mismo, para así dar seguimiento a los requerimientos de software indispensables para su instalación en el nuevo servidor.

**COMPROMISOS**

ANGEL	
Actividad	Fecha
Crear la base de datos del Sistema de Gestión de Tickets en el nuevo servidor	Una vez instalado el software requerido

**PRÓXIMA REUNIÓN: A definir**

FIRMA  
AUTÓGRAFA

\_\_\_\_\_  
Angel Brito Segura

FIRMA  
AUTÓGRAFA

\_\_\_\_\_  
M.I. Tanya Itzel Arteaga Ricci

FIRMA  
AUTÓGRAFA

\_\_\_\_\_  
Leo Rivera



**Minuta 7**  
Sistema de Gestión de Tickets



**PROYECTO:** Sistema de Gestión de Tickets  
**ETAPA :** Desarrollo  
**LUGAR :** A distancia (*Meet*)

**FECHA:** 24 de octubre de 2022  
**HORA DE INICIO:** 14:00 horas  
**HORA DE TÉRMINO:** 15:00 horas

**Lista de asistentes:**

- Angel Brito Segura [*Líder de proyecto – tesista*]
- M.I. Tanya Itzel Arteaga Ricci [*Patrocinadora – directora de tesis*]
- Staff de la Unidad de Cómputo (UC) de la División de Ingenierías Civil y Geomática (DICyG) [*Cliente*]

**OBJETIVO** Mostrar al cliente una primera demostración práctica del Sistema de Gestión de Tickets (SGT).

**ORDEN DEL DÍA**

Mostrar al cliente los 4 **módulos añadidos al SGT** para recibir comentarios y sugerencias que retroalimenten el desarrollo del proyecto.

**ACTIVIDADES REALIZADAS**

Se discutieron los módulos desarrollados hasta este momento del SGT y se resolvieron dudas sobre su interconexión con el Sistema de Inventarios.

**COMPROMISOS**

<b>LÍDER DE PROYECTO</b>		<b>CLIENTE</b>	
Actividad	Fecha	Actividad	Fecha
Concluir el módulo de asignación del SGT	04/11/22	Acceso al repositorio de GitHub para mantener un control de versiones en el código	24/10/22
Subir al servidor la página web desarrollada	En cuanto se tenga acceso a MySQL y a la IP pública	Realizar la conexión a la base de datos del Sistema de Inventarios desde el SGT	28/10/22
		Creación de IP pública para el servidor que aloja al SGT	A la brevedad

**PRÓXIMA REUNIÓN: A definir**

FIRMA  
AUTÓGRAFA

Angel Brito Segura

FIRMA  
AUTÓGRAFA

M.I. Tanya Itzel Arteaga Ricci

FIRMA  
AUTÓGRAFA

UC de la DICyG



## Minuta 8




Sistema de Gestión de Tickets



**PROYECTO:** Sistema de Gestión de Tickets  
**ETAPA :** Desarrollo  
**LUGAR :** A distancia (*Meet*)

**FECHA:** 13 de marzo de 2023  
**HORA DE INICIO:** 14:00 horas  
**HORA DE TÉRMINO:** 15:30 horas

### Lista de asistentes:

-  Angel Brito Segura [*Líder de proyecto – tesista*]
-  M.I. Tanya Itzel Arteaga Ricci [*Patrocinadora – directora de tesis*]
-  Staff de la Unidad de Cómputo (UC) de la División de Ingenierías Civil y Geomática (DICyG) [*Cliente*]

#### OBJETIVO

Mostrar al cliente el ciclo de vida de un ticket a través de los módulos implementados en el Sistema de Gestión de Tickets (SGT)

## ORDEN DEL DÍA

Presentar al cliente la conclusión de los **módulos creados en el SGT** para su visto bueno y recibir comentarios/sugerencias que retroalimenten el desarrollo del proyecto.

## ACTIVIDADES REALIZADAS

Se expuso el funcionamiento integral del SGT para la gestión del ciclo de vida de cualquier solicitud de soporte técnico, respecto a lo cual se obtuvo la aprobación del cliente, autorizando el inicio de las pruebas. Adicionalmente, se discutieron los reportes que serán generados por el sistema y la información que se registrará en cada ticket.

## COMPROMISOS

LÍDER DE PROYECTO		PATROCINADORA	
Actividad	Fecha	Actividad	Fecha
Aplicar filtrado en dispositivos inventariados al momento de levantar un ticket	17/03/23	Firmar documentación para el trabajo escrito	13/03/23
Concluir los reportes en producción	21/03/23	Verificar la conexión del cliente con el SGT desde la UC	A la brevedad

### PRÓXIMA REUNIÓN: A definir

FIRMA  
AUTÓGRAFA

Angel Brito Segura

FIRMA  
AUTÓGRAFA

M.I. Tanya Itzel Arteaga Ricci

FIRMA  
AUTÓGRAFA

UC de la DICyG



---

**Minuta 9**  
**Sistema de Gestión de Tickets**





---



**PROYECTO:** Sistema de Gestión de Tickets  
**ETAPA :** Pruebas  
**LUGAR :** A distancia (*Meet*)

**FECHA:** 31 de marzo de 2023  
**HORA DE INICIO:** 14:00 horas  
**HORA DE TÉRMINO:** 15:00 horas

### **Lista de asistentes:**

-  Angel Brito Segura [*Líder de proyecto – tesista*]
-  M.I. Tanya Itzel Arteaga Ricci [*Patrocinadora – directora de tesis*]
-  Dra. Martha Angélica Nakayama Cervantes [*Asesora Técnica*]
-  Staff de la Unidad de Cómputo (UC) de la División de Ingenierías Civil y Geomática (DICyG) [*Cliente*]

**OBJETIVO**

Aclarar dudas y realizar comentarios relacionados con el uso del Sistema de Gestión de Tickets (SGT), así como definir la conclusión del periodo de pruebas en producción.

### **ORDEN DEL DÍA**

Recibir retroalimentación por parte del cliente acerca del ***uso y funcionalidades implementadas en el SGT*** para aclarar cuestiones que no permitan su uso correcto.

### **ACTIVIDADES REALIZADAS**

Se discutieron los diversos mensajes enviados por el SGT y el flujo a seguir para realizar la transición del proceso manual al proceso automatizado a través del sistema para los trabajadores y personal administrativo de la DICyG. También se estableció el término del periodo de pruebas.



## COMPROMISOS

LÍDER DE PROYECTO		CLIENTE	
Actividad	Fecha	Actividad	Fecha
Concluir graficación de reportes en producción	10/04/23	Iniciar las gestiones para incorporar usuarios solicitantes reales al sistema	10/04/23
Elaboración de los manuales: técnico de desarrollo, administrativo y de usuario	10/04/23	Mantener actualizada la base de datos del Sistema de Inventarios	En todo momento
Entrega de guía rápida de usuario Solicitante del SGT	10/04/23	Conclusión de pruebas en producción	17/04/23

### PRÓXIMA REUNIÓN: A definir

FIRMA  
AUTÓGRAFA

Angel Brito Segura

FIRMA  
AUTÓGRAFA

M.I. Tanya Itzel Arteaga Ricci

FIRMA  
AUTÓGRAFA

Dra. Martha Angélica Nakayama Cervantes

FIRMA  
AUTÓGRAFA

UC de la DICyG



# Apéndice B

## Documentación entregada

En este apéndice se integra la documentación solicitada por el cliente y entregada en cada etapa del SDLC de acuerdo a lo establecido en el acta constitutiva, que además consagra la formalización del inicio del proyecto de tesis, así como los requerimientos para el desarrollo del software. Está basado en la guía PMBOK (*Project Management Body of Knowledge*) que establece un criterio de buenas prácticas relacionadas con la gestión, administración y dirección de proyectos.

Definido el alcance del proyecto, se crea la carta descriptiva. Este documento describe, de manera general, los componentes del sistema, haciendo una breve exposición de lo que se pretende abordar en cada uno de ellos.

Por su parte, el documento que describe los atributos del esquema lógico de la base de datos del Sistema de Gestión de Tickets es el denominado diccionario de datos de la base de datos del SGT, donde se especifica para cada uno de ellos su dominio y valor por defecto -entre otras características- para una mejor comprensión del diseño físico implementado en su DB.

Con motivo de la conclusión del sistema, se entrega el informe de resultados y cierre del proyecto, en el cual se describen -entre otros aspectos- los resultados obtenidos y el estatus de entrega del presente proyecto.

Adicionalmente se hizo la entrega de trípticos que hacen patente la transición del proceso manual -actualmente empleado por la UC- al automatizado gracias al SGT. Dichos trípticos describen, por una parte, la serie de pasos que debe realizar un usuario Solicitante para levantar un ticket y dar seguimiento a su solicitud; por otra parte, los pasos para el usuario con rol Encargado de Laboratorio para realizar la Verificación Administrativa de los laboratorios que se tengan registrados en el SGT.

# **ACTA DE CONSTITUCIÓN DEL PROYECTO\***

## ***Sistema de Gestión de Tickets***

*8 de noviembre de 2021*

\*Actualizada a la última versión de la guía PMBOK 6, publicada en 2017.

# Tabla de contenido

- Información del proyecto ..... 3
  - Datos..... 3
  - Patrocinadora..... 3
- Propósito y justificación del proyecto ..... 3
- Descripción del proyecto y entregables..... 4
- Requerimientos de alto nivel ..... 5
  - Requerimientos del producto ..... 5
  - Requerimientos del proyecto ..... 5
- Objetivos ..... 6
- Premisas y restricciones ..... 7
- Riesgos iniciales de alto nivel..... 7
- Cronograma de hitos principales..... 8
- Presupuesto inicial asignado..... 8
- Lista de interesados (*stakeholders*)..... 9
- Requisitos de aprobación del proyecto ..... 9
- Criterio de cierre..... 10
- Asignación del líder de proyecto ..... 10
- Recursos preasignados..... 10
- Aprobación ..... 10

## Información del proyecto

### Datos

Empresa/Organización	Universidad Nacional Autónoma de México
Proyecto	<b>Sistema de Gestión de Tickets</b>
Fecha de preparación	22 de septiembre de 2021
Cliente	Unidad de Cómputo de la División de Ingenierías Civil y Geomática (DICyG)
Patrocinadora	M.I. Tanya Itzel Arteaga Ricci
Líder de proyecto	Angel Brito Segura

### Patrocinadora

Nombre	División	Coordinación	Rama ejecutiva
M.I. Tanya Itzel Arteaga Ricci	Ingenierías Civil y Geomática	Unidad de Cómputo	Jefatura

## Propósito y justificación del proyecto

El Sistema de Gestión de Tickets llevará un control de las solicitudes (tickets) de soporte técnico y de capacitación que realiza la Unidad de Cómputo de la DICyG. Este sistema constituirá una central digital de atención a usuarios, disponible desde cualquier punto, para los solicitantes que requieren de apoyo técnico.

Este proyecto llevará a cabo la graficación de todo lo realizado en la atención de los tickets para poder dimensionar y administrar al personal con el que cuenta la UC. Por otro lado, servirá para mantener un histórico de los cambios realizados en cada equipo de cómputo, lo que redundará en soportes técnicos más rápidos y precisos.

## Descripción del proyecto y entregables

El Sistema de Gestión de Tickets consistirá en un sitio web accesible desde cualquier navegador (requerirá correo electrónico y contraseña para ingresar a él). Estará dividido en 7 secciones principales:

- 🌀 Inicio de Sesión
- 🌀 Gestión de Catálogos
- 🌀 Levantamiento de Ticket
- 🌀 Asignación de Técnicos
- 🌀 Atención y Seguimiento de Tickets
- 🌀 Foros
- 🌀 Reportes

El sistema acepta varios usuarios, con uno o más roles dentro del mismo, y cada uno de ellos cuenta con los permisos necesarios para realizar, en el ámbito de sus atribuciones, las actividades correspondientes. Estos roles son:

- Administrador Ejecutivo (Jefatura)
- Administrador del Sistema (responsable del manejo de la información)
- Líder Técnico (asignador de técnicos)
- Supervisor (verifica la parte técnica de un soporte)
- Encargado de Laboratorio (verifica el mantenimiento de los equipos a su cargo)
- Técnico Junior (atiende ticket; sujeto a supervisión)
- Técnico Máster (atiende ticket; no requiere supervisión)
- Solicitante (levanta ticket)

El sistema se creará a distancia por el líder de proyecto, conectándose por SSH al servidor proporcionado por la Unidad de Cómputo de la DICyG.

Antes de comenzar el desarrollo del proyecto, se entrega la **Carta Descriptiva** del mismo. Al finalizar el proyecto, se entregará el sitio web funcionando en la dirección electrónica dada por el cliente además de la siguiente documentación:

- ✓ Manual de usuario
- ✓ Manual técnico de desarrollo
- ✓ Manual administrativo

## Requerimientos de alto nivel

### Requerimientos del producto

Sitio web para administrar las solicitudes (tickets) de soporte técnico y de capacitación que realiza la Unidad de Cómputo de la DICyG:

- ✓ Las notificaciones del sistema se harán por correo electrónico institucional de *noreply*.
- ✓ El sistema mostrará el historial de cambios y soportes realizados en cada equipo de cómputo con su número de inventario.
- ✓ El usuario Solicitante podrá pedir una solicitud de soporte técnico o de capacitación (levantamiento de ticket) al autenticarse en el sistema.
- ✓ El sistema mantendrá un histórico de quienes hayan atendido cada equipo de cómputo.
- ✓ El sistema generará dos tipos de reportes: un informe ejecutivo y un listado largo de las actividades realizadas comprendidas en los tickets generados.
- ✓ El sistema deberá indicar si un soporte está siendo atendido o no.
- ✓ El sistema ayudará a centralizar la atención de los usuarios Solicitantes.
- ✓ Los tickets pueden ser programados (se establece una fecha de atención) o no (son atendidos en el momento).
- ✓ En el sistema deberá elegirse qué tipo de soporte se requiere de acuerdo al catálogo disponible.
- ✓ El informe ejecutivo deberá graficarse en el sistema para el reporte anual.
- ✓ El sistema contará con un foro de comunicación con el solicitante, así como con un foro de asesoría tecnológica (“presoporte”) en donde se puedan consultar preguntas frecuentes.

### Requerimientos del proyecto

El sistema gestor de la base de datos será PostgreSQL (se requiere mínimo su versión 9.6) y la parte de *backend* será manejada por Django 3.2.9; por lo que se necesita instalar mínimo Python 3.6 en el servidor de la Unidad de Cómputo.

También, se requiere la herramienta *pip* instalada en el servidor para realizar la gestión de paquetes de Python. Los paquetes que deben estar instalados son: Django==3.2.9, psycopg2-binary, mysqlclient y django-bootstrap4; este último para manejar la parte de *frontend* con el framework Bootstrap 4.

## Objetivos

Objetivo	Indicador de éxito
<b>Alcance</b>	
Sitio web de administración de tickets de soporte técnico y de capacitación	Cuenta con las 7 secciones principales en correcto funcionamiento
Foros en sitio web	Dos foros disponibles, abiertos y de fácil manejo
Reportes en el Sistema de Gestión de Tickets	Autogenerados y con graficación acorde al tipo de información
<b>Cronograma</b>	
Base de datos creada y con registros iniciales	Funcionando en servidor (24/01/2022)
Front-End y Back-End funcionando en producción	Sitio web accesible desde cualquier explorador (06/02/2023)
Pruebas en el sitio web con usuarios reales	Exitosas (20/03/2023)
<b>Costo</b>	
Licencias de software	Libres
Horas efectivas de trabajo	1000
<b>Calidad</b>	
Sistema autosustentable	Código documentado
Experiencia de usuario	Amigable: de fácil uso, con buen desempeño y diseño responsivo

## Premisas y restricciones

El Sistema de Gestión de Tickets tiene que conectarse con el Sistema de Inventarios para obtener los datos del equipo de cómputo con su respectivo dueño/usuario, así como para hacer uso del correo institucional para mandar notificaciones a sus usuarios. En las páginas web a crear es necesario respetar la imagen institucional con logos, colores y diseño general. Cuando el grado de avance del desarrollo del proyecto lo permita, se tendrán sesiones con los usuarios solicitantes para recibir sus comentarios acerca de la funcionalidad del mismo.

La administración e instalación de software en el servidor estará a cargo de la Unidad de Cómputo. Por su parte, el control de versiones de código fuente será con Git, siendo el repositorio central GitHub (la Unidad de Cómputo dará las credenciales necesarias). Se seguirá una propia metodología de desarrollo basada en SCRUM, en la cual cada mes se tendrán reuniones con el cliente para mostrar los avances y recibir sus observaciones para agregar nuevos requerimientos o eliminar secciones que estime innecesarias en el sistema.

## Riesgos iniciales de alto nivel

- ✚ Problemas de compatibilidad del software requerido con el servidor de la Unidad de Cómputo
- ✚ Problemas de conexión entre MySQL y PostgreSQL
- ✚ Aún no se cuenta con acceso a un servidor de pruebas



## Cronograma de hitos principales

Hito	Plazo <sup>1</sup>
Muestra de prototipos funcionales	1 mes
Diseño de la base de datos para el Sistema de Gestión de Tickets	2 meses
Diagrama entidad-relación y diccionario de datos de la base de datos del Sistema de Gestión de Tickets	1 semana
Inicio de Sesión y Gestión de Catálogos implementadas en el servidor y puestas en marcha	8 meses
Levantamiento de Ticket de soporte técnico y de capacitación disponible en el sitio web	5 semanas
Creación de foros en el Sistema de Gestión de Tickets	6 semanas
Asignación de Técnicos funcionando en el sitio web	1 mes
Atención y Seguimiento de Ticket (verificación, validación y reasignación) desde el sitio web	3 semanas
Administración de tickets de mantenimientos programados a laboratorios	7 semanas
Muestra de reportes en el sitio web	1 mes
Pruebas de funcionamiento del sitio web en tiempo real	2 semanas

## Presupuesto inicial asignado

En la Unidad de Cómputo no se cuenta con presupuesto para la compra de licencias. El tiempo aproximado para realizar el proyecto es de 2 años.

<sup>1</sup> Tiempo estimado para resolver la actividad descrita, una vez que se concluye el hito precedente

## Lista de interesados (*stakeholders*)

**División:** Ingenierías Civil y Geomática

**Coordinación:** Unidad de Cómputo

Nombre	Rama ejecutiva
Ing. Abraham Bonila Pastor	Líder técnico en arquitectura cliente-servidor
Mtra. Alejandra Guzmán Huxley	Líder técnico en administración de medios sociales, planeación y gestión de recurso humano
Ing. Armando Vázquez Díaz	Líder técnico en desarrollo y bases de datos
Diego Ramírez Romero	Líder técnico en infraestructura y sistemas de gestión de aprendizaje
Héctor Martínez Garduño	Líder técnico en redes y seguridad
Ing. Jair Camacho Alfao	Líder técnico en operación de hardware y software
Jesús Cuellar Sánchez	Líder técnico en operación de hardware y software

## Requisitos de aprobación del proyecto

Sitio web cumple con lo descrito en la carta descriptiva, su código es entendible y puede ser administrado por cualquier otro programador (basándose en su documentación). En este sitio es posible administrar por completo el ciclo de vida de un ticket: Levantamiento, Asignación, Atención, Verificación y Validación; dar seguimiento a los cambios de personal en cada equipo de cómputo y los reportes reflejan la información contenida en el sistema.

## Criterio de cierre

Entrega del sitio web en producción con su documentación completa y aprobada por la Unidad de Cómputo.

## Asignación del líder de proyecto

Nombre	Cargo	División	Firma
Angel Brito Segura	Estudiante de la carrera de Ingeniería en Computación	Ingeniería Eléctrica	FIRMA AUTÓGRAFA

## Recursos preasignados

Recurso	División	Coordinación
Servidor con Debian 10	Ingenierías Civil y Geomática	Unidad de Cómputo
Correo electrónico para notificaciones <i>noreply</i>	Ingenierías Civil y Geomática	Unidad de Cómputo

## Aprobación

Patrocinador	Fecha	Firma
M.I. Tanya Itzel Arteaga Ricci	09/12/2021	FIRMA AUTÓGRAFA

# Carta Descriptiva del Sistema de Gestión de Tickets

8 de noviembre de 2021

El Sistema de Gestión de Tickets administra las solicitudes (*tickets*) de soporte técnico y de capacitación. Está a cargo de la Unidad de Cómputo de la División de Ingenierías Civil y Geomática de la Facultad de Ingeniería de la UNAM. Para acceder a este sistema es necesario ingresar a su [sitio web](https://computo.dicyg.fi-c.unam.mx/) desde cualquier navegador; la dirección es la siguiente: «<https://computo.dicyg.fi-c.unam.mx/>».

Al ingresar, se solicita identificarse para poder entrar y hacer uso de él, por lo que debe proporcionarse el correo electrónico -y contraseña- previamente dado de alta en el sistema.

Este sistema consta de 7 secciones principales que se describen a continuación:

## 1. Inicio de Sesión

Es la página de inicio del sistema: muestra un mensaje de bienvenida y la opción de autenticación para entrar al sistema mediante un correo electrónico y contraseña.

## 2. Gestión de Catálogos

El sistema acepta varios usuarios con uno o más roles dentro del mismo. Estos roles son:

- **Administrador Ejecutivo** (Jefatura): Tiene el control total del sistema, el acceso a todas las secciones y es el encargado de levantar el ticket de mantenimiento programado a laboratorios.
- **Administrador del Sistema**: Es el responsable de la administración de la información dentro del sistema.
- **Líder Técnico**: Asigna a un Técnico Junior o Máster para atender un ticket.

- **Supervisor:** Verifica técnicamente si el ticket fue atendido de forma correcta.
- **Encargado de Laboratorio:** Realiza la verificación administrativa del mantenimiento a equipos de cómputo en laboratorios a su cargo.
- **Técnico Junior:** Atiende sus tickets asignados. Requiere que un Supervisor verifique si el problema reportado fue resuelto exitosamente.
- **Técnico Máster:** Atiende sus tickets asignados sin supervisión alguna.
- **Solicitante:** Es quien levanta un ticket de soporte técnico o de capacitación y el encargado de validar que su problema fue resuelto.

Los usuarios con el rol de Administrador Ejecutivo y Administrador del Sistema son los que tienen acceso a esta sección. Este último realiza alta, activación, actualización de datos o baja de los usuarios; cambia o agrega roles a cada usuario y administra los equipos cómputo registrados. También se pueden modificar los listados mostrados a lo largo del sistema: roles para usuarios, actividades con su ponderación, software que se maneja, laboratorios registrados y áreas de la Unidad de Cómputo.

### 3. Levantamiento de Ticket

Los usuarios con el rol de Administrador Ejecutivo y de Solicitante tienen acceso a esta sección.

Desde esta sección, el Solicitante levanta un ticket de soporte técnico o de capacitación, exponiendo la causa del mismo. Solo el Administrador Ejecutivo tiene habilitada la opción de levantar tickets de mantenimiento programado a laboratorios.

### 4. Asignación de Técnicos

Los usuarios con el rol de Administrador Ejecutivo y de Líder Técnico tienen acceso a esta sección.

Los Líderes Técnicos son los encargados de concluir el llenado de la solicitud correspondiente para comenzar a dar el soporte; asigna al Técnico que se encargará de la solicitud, estableciendo la actividad (ideal) a realizar y una fecha aproximada de término.

### 5. Atención y Seguimiento de Tickets

Los usuarios con el rol de Administrador Ejecutivo, Líder Técnico, Técnico (solo la parte de Atención), Supervisor (si es necesario para el ticket) y Solicitante (solo la parte de Seguimiento) tienen acceso a esta sección.

Los Técnicos son los encargados de atender los tickets que fueron asignados, reportando en todo momento los cambios realizados al equipo de cómputo; también agrega las nuevas actividades realizadas (si es necesario) y mantiene comunicación con el Solicitante a través del foro correspondiente.

Desde esta sección, el Solicitante puede habilitar el seguimiento de su problema para poder observar los cambios que se están haciendo a su equipo de cómputo.

## 6. Foros

El Sistema de Gestión de Tickets tiene dos tipos de foro, ambos disponibles para todos los roles:

### 6.1. Foro de Comunicación

Automáticamente, por ticket levantado (de soporte técnico o de capacitación) se abre un foro, mediante el cual se mantiene contacto directo entre el Solicitante y el staff de la Unidad de Cómputo (Líder Técnico, Técnico asignado y, de ser necesario, Supervisor). Una vez que el Solicitante valida que el problema está resuelto, se cierra este foro.

### 6.2. Foro de Asesoría Tecnológica

En este foro se encuentran preguntas y respuestas frecuentes -en forma de tutoriales- de soportes comunes que pueden ser resueltos por los propios usuarios, sin la necesidad del levantamiento de un ticket.

## 7. Reportes

Los usuarios con el rol de Administrador Ejecutivo y de Líder Técnico tienen acceso a esta sección.

Muestra el informe ejecutivo (graficación) y el listado largo de las actividades realizadas comprendidas en los tickets generados en el sistema.



Unidad de Cómputo

División de Ingenierías Civil y Geomática



# Diccionario de Datos

## Sistema de Gestión de Tickets

Angel Brito Segura

Tabla: USUARIO		Descripción: Tabla en la que se almacenan los datos del usuario				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
usuario_id	BIGSERIAL	PK	NOT NULL	Auto incrementable		Identificador del usuario
nombre_usr	VARCHAR (300)		NOT NULL			Nombre del usuario
primer_apellido	VARCHAR (150)		NOT NULL			Primer apellido del usuario
segundo_apellido	VARCHAR (150)		NULL			Segundo apellido del usuario
num_trabajador	BIGINT	UK	NOT NULL			Número de trabajador/cuenta del usuario
telefono	NUMERIC (10,0)		NULL			Teléfono del usuario
email	VARCHAR (200)	UK	NOT NULL			Correo electrónico del usuario para ingresar al sistema
password	VARCHAR (250)		NOT NULL			Contraseña cifrada del usuario para ingresar al sistema
status_usr	BOOLEAN		NOT NULL	TRUE		Estatus del usuario <ul style="list-style-type: none"> <li>• <b>TRUE</b> - Activo</li> <li>• <b>FALSE</b> - Inactivo</li> </ul>
last_login	TIMESTAMP		NULL			Fecha (año-mes-día) y hora (horas:minutos:segundos) de la última conexión del usuario
tipo_usuario	CHAR(1)		NULL		CK	Tipo de usuario: <ul style="list-style-type: none"> <li>• <b>T</b> – Trabajador de la UC</li> </ul> Si NULL es el valor del campo significa que el usuario no trabaja en la UC de la DICyG
departamento_id	SMALLSERIAL	FK	NOT NULL		Cascada	Llave foránea que hace referencia al identificador del departamento que pertenece el usuario
profesion_id	SMALLSERIAL	FK	NOT NULL		Cascada	Llave foránea que hace referencia al identificador de la ocupación del usuario

Tabla: TRABAJADOR_UC		Descripción: Tabla en la que se almacenan los datos del trabajador de la Unidad de Cómputo (UC) de la División de Ingenierías Civil y Geomática (DICyG)				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
usuario_id	BIGSERIAL	PK, FK	NOT NULL		Restringida	Llave foránea con el identificador del usuario trabajador en la UC de la DICyG en el Sistema de Gestión de Tickets
horario	VARCHAR (300)		NOT NULL			Horario en el que el trabajador se encuentra laborando



Tabla: DEPARTAMENTO		Descripción: Catálogo de los departamentos que integran la DICyG				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
departamento_id	SMALLSERIAL	PK	NOT NULL	Auto incrementable		Identificador del departamento para el Sistema de Gestión de Tickets
nombre_depto	VARCHAR(100)		NOT NULL			Nombre del departamento de la DICyG
desc_depto	VARCHAR(1000)		NULL			Descripción del departamento de la DICyG

Tabla: PROFESION		Descripción: Catálogo de las profesiones que puede tener un usuario				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
profesion_id	SMALLSERIAL	PK	NOT NULL	Auto incrementable		Identificador de la profesión
nombre_prof	VARCHAR(100)		NOT NULL			Nombre de la ocupación del usuario
desc_prof	VARCHAR(1000)		NULL			Descripción de la ocupación del usuario

Tabla: AREA_UC		Descripción: Catálogo de las áreas en la UC de la DICyG				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
area_uc_id	SMALLSERIAL	PK	NOT NULL	Auto incrementable		Identificador del área de la UC en la DICyG
nombre_area	VARCHAR(100)		NOT NULL			Nombre del área en la UC de la DICyG
desc_area	VARCHAR(1000)		NULL			Descripción del área en la UC de la DICyG

Tabla: TUTORIAL		Descripción: Tabla en la que se almacenan los datos de tutoriales que serán mostrados en el Foro de Asesoría Tecnológica				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
tutorial_id	SERIAL	PK	NOT NULL	Auto incrementable		Identificador del tutorial
titulo	VARCHAR(100)		NOT NULL			Nombre del tutorial del Foro de Asesoría Tecnológica
url	TEXT	UK	NOT NULL			URL del tutorial en PDF a mostrar en el Foro de Asesoría Tecnológica
actualizacion	TIMESTAMP		NOT NULL	Fecha y hora actual		Fecha (año-mes-día) y hora (horas:minutos:segundos) de actualización del tutorial
area_uc_id	SMALLSERIAL	FK	NOT NULL		Restringida	Llave foránea que hace referencia al área asociada del tutorial
autor_id	BIGSERIAL	FK	NOT NULL		Restringida	Llave foránea que hace referencia al autor (trabajador de la UC en la DICyG) del tutorial

Tabla: ROL		Descripción: Catálogo de los roles posibles en el Sistema de Gestión de Tickets				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
rol_id	SMALLSERIAL	PK	NOT NULL	Auto incrementable		Identificador del rol en el Sistema de Gestión de Tickets
nombre_rol	VARCHAR(40)		NOT NULL			Nombre del rol en el Sistema de Gestión de Tickets
descripcion_rol	VARCHAR(1000)		NOT NULL			Breve descripción del rol en el Sistema de Gestión de Tickets

Tabla: USUARIO_ROL		Descripción: Tabla en la que se almacena la relación entre el usuario y los roles en el Sistema de Gestión de Tickets				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
usuario_rol_id	BIGSERIAL	PK	NOT NULL	Auto incrementable		Identificador de la relación entre usuario y rol
usuario_id	BIGSERIAL	FK	NOT NULL		Restringida	Llave foránea que hace referencia al usuario con el rol asignado
rol_id	SMALLSERIAL	FK	NOT NULL		Cascada	Llave foránea que hace referencia al rol asignado a un usuario

Tabla: ACTIVIDAD		Descripción: Tabla en la que se almacena el listado de las actividades ponderadas en la UC de la DICyG				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
actividad_id	SERIAL	PK	NOT NULL	Auto incrementable		Identificador de la actividad
desc_actividad	VARCHAR(100)		NOT NULL			Descripción de la actividad ponderada
ponderacion	NUMERIC(2,0)		NOT NULL			Valor numérico de la ponderación de la actividad
observacion_act	VARCHAR(1000)		NULL			Observaciones o notas asociadas a la actividad ponderada
area_uc_id	SMALLSERIAL	FK	NOT NULL		Restringida	Llave foránea que hace referencia al área asociada a la actividad ponderada

Tabla: ACTIVIDAD_EXTRA		Descripción: Tabla en la que se almacenan los datos de actividades que no están listadas en la relación ACTIVIDAD				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
actividad_extra_id	SERIAL	PK	NOT NULL	Auto incrementable		Identificador de la actividad no listada en las actividades ponderadas
desc_act_extra	VARCHAR(100)		NOT NULL		UK	Descripción de la actividad extra no listada en las actividades ponderadas
trabajador_id	BIGSERIAL	FK	NOT NULL		Cascada	Llave foránea que hace referencia al trabajador de la UC en la DICyG que registra la actividad extra

Tabla: LABORATORIO		Descripción: Tabla en la que se almacenan los datos de los laboratorios en los que se realizan mantenimientos programados por parte de la UC de la DICyG				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
laboratorio_id	SERIAL	PK	NOT NULL	Auto incrementable		Identificador del laboratorio para el Sistema de Gestión de Tickets
nombre_lab	VARCHAR(100)		NOT NULL			Nombre del laboratorio
descripcion_lab	VARCHAR(1000)		NULL			Breve descripción del laboratorio
ubicacion	VARCHAR(40)		NOT NULL			Ubicación del laboratorio (utilizar la nomenclatura de claves de la Facultad de Ingeniería)
encargado_id	BIGSERIAL	FK	NOT NULL		Restringida	Llave foránea que hace referencia al encargado del laboratorio

Tabla: TAREA_MANTENIMIENTO		Descripción: Tabla en la que se almacena el listado de tareas programadas para el mantenimiento de los laboratorios				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
tarea_mantenimiento_id	SERIAL	PK	NOT NULL	Auto incrementable		Identificador de la tarea de mantenimiento para los laboratorios
nombre_tarea	VARCHAR(100)		NOT NULL			Nombre de la tarea de mantenimiento
descripcion_tarea	VARCHAR(1000)		NULL			Descripción breve de la tarea de mantenimiento

Tabla: REVISION_SOFTWARE		Descripción: Tabla en la que se almacena el listado de software a revisar en el mantenimiento programado a los laboratorios				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
rev_software_id	SERIAL	PK	NOT NULL	Auto incrementable		Identificador del software a revisar en los laboratorios
nombre_sw	VARCHAR(100)		NOT NULL			Nombre del software a revisar
descripcion_sw	VARCHAR(1000)		NULL			Descripción del software a revisar

Tabla: TAREA_LAB		Descripción: Tabla en la que se almacena la relación entre el laboratorio y las tareas de mantenimiento a realizar				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
tarea_lab_id	BIGSERIAL	PK	NOT NULL	Auto incrementable		Identificador de la relación entre la tarea de mantenimiento y el laboratorio
laboratorio_id	SERIAL	FK	NOT NULL		Restringida	Llave foránea que hace referencia al laboratorio donde se realiza la tarea
tarea_mantenimiento_id	SERIAL	FK	NOT NULL		Cascada	Llave foránea que hace referencia a la tarea de mantenimiento a realizar en el laboratorio

Tabla: TICKET		Descripción: Tabla en la que se almacenan los tickets solicitados a la UC de la DICyG				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
ticket_id	BIGSERIAL	PK	NOT NULL	Auto incrementable		Identificador de la solicitud (ticket)
fecha_levantamiento	TIMESTAMP		NOT NULL	Fecha y hora actual		Fecha (año-mes-día) y hora (horas: minutos: segundos) en la que el usuario levanta el ticket
fecha_termino	DATE		NULL			Fecha propuesta para finalizar la atención del ticket
fecha_status	TIMESTAMP		NOT NULL	Fecha y hora actual		Fecha (año-mes-día) y hora (horas: minutos: segundos) en la que se registra un cambio en la atención del ticket
calif_validacion	NUMERIC(2,0)		NULL		CK	Calificación que el usuario da a la atención general de su solicitud (6, 8 o 10)
comentarios_atencion	TEXT		NULL			Comentarios que el usuario da al técnico que atendió su solicitud
tipo_ticket	CHAR(1)		NOT NULL		CK	Tipo de ticket: <ul style="list-style-type: none"> <li>• <b>C</b> – Ticket de Capacitación</li> <li>• <b>S</b> – Ticket de Soporte Técnico</li> <li>• <b>M</b> – Ticket de Mantenimiento</li> </ul>
usuario_id	BIGSERIAL	FK	NOT NULL		Restringida	Llave foránea que hace referencia al usuario que levanta el ticket
status_ticket_id	SMALLSERIAL	FK	NOT NULL		Restringida	Llave foránea que hace referencia al estatus actual del ticket
tecnico_id	BIGSERIAL	FK	NULL		En Nulo	Llave foránea que hace referencia al técnico actual que está atendiendo el ticket

Tabla: STATUS_TICKET		Descripción: Catálogo en el que se encuentran los distintos estatus en los que se puede hallar un ticket				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
status_ticket_id	SMALLSERIAL	PK	NOT NULL	Auto incrementable		Identificador del estatus del ticket
clave	VARCHAR(20)		NOT NULL			Nombre del estatus del ticket
desc_status	VARCHAR(100)		NOT NULL			Descripción breve del estatus del ticket

Tabla: SOFTWARE LAB		Descripción: Tabla en la que se almacena la relación entre el laboratorio y el software a revisar (instalar/actualizar)				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
software_lab_id	BIGSERIAL	PK	NOT NULL	Auto incrementable		Identificador de la relación entre el software a revisar y su laboratorio
laboratorio_id	SERIAL	FK	NOT NULL		Restringida	Llave foránea que hace referencia al laboratorio donde se realiza la revisión de software
rev_software_id	SERIAL	FK	NOT NULL		Cascada	Llave foránea que hace referencia al software a revisar en el laboratorio

Tabla: MENSAJE		Descripción: Tabla en la que se almacenan los mensajes del Foro de Comunicación				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
mensaje_id	BIGSERIAL	PK	NOT NULL	Auto incrementable		Identificador del mensaje
contenido	TEXT		NOT NULL			Contenido del mensaje
fecha	TIMESTAMP		NOT NULL	Fecha y hora actual		Fecha (año-mes-día) y hora (horas:minutos:segundos) de emisión del mensaje
usuario_id	BIGSERIAL	FK	NOT NULL		Restringida	Llave foránea que hace referencia al usuario que emite el mensaje
ticket_id	BIGSERIAL	FK	NOT NULL		Cascada	Llave foránea que hace referencia al ticket

Tabla: ACT_PROPUESTA_TICKET		Descripción: Tabla en la que se almacena la relación entre el ticket y sus actividades propuestas por el Líder Técnico				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
act_prop_ticket_id	BIGSERIAL	PK	NOT NULL	Auto incrementable		Identificador de la actividad propuesta para el ticket
ticket_id	BIGSERIAL	FK	NOT NULL		Cascada	Llave foránea que hace referencia al identificador del ticket levantado
actividad_id	SERIAL	FK	NULL		En Nulo	Llave foránea que hace referencia al identificador de la actividad listada con su ponderación
actividad_extra_id	SERIAL	FK	NULL		En Nulo	Llave foránea que hace referencia al identificador de la actividad creada por el líder técnico que realiza la asignación

<b>Tabla: ASIGNACION_TICKET</b>		<b>Descripción:</b> Tabla en la que se almacenan los datos de la asignación del ticket realizado por un Líder Técnico				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
<b>asignacion_ticket_id</b>	BIGSERIAL	PK	NOT NULL	Auto incrementable		Identificador de la asignación del ticket
<b>fecha_cambio_asig</b>	TIMESTAMP		NOT NULL	Fecha y hora actual		Fecha (año-mes-día) y hora (horas:minutos:segundos) de cambio en la asignación del ticket
<b>status_asig</b>	CHAR(1)		NOT NULL	A	CK	Estatus de la asignación del ticket: <ul style="list-style-type: none"> <li>● A – asignado</li> <li>● E – endosado</li> </ul>
<b>motivo_endoso</b>	TEXT		NULL			Descripción del motivo del endoso por parte del técnico
<b>ticket_id</b>	BIGSERIAL	FK	NOT NULL		Cascada	Llave foránea que hace referencia al ticket levantado
<b>asignador_id</b>	BIGSERIAL	FK	NOT NULL		Restringida	Llave foránea que hace referencia al Líder Técnico que asignó el ticket
<b>tecnico_id</b>	BIGSERIAL	FK	NOT NULL		Restringida	Llave foránea que hace referencia al técnico que atenderá el ticket

<b>Tabla: HISTORICO_EVENTO_TICKET</b>		<b>Descripción:</b> Tabla en la que se almacena el historial de atención de los tickets levantados				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
<b>hist_event_ticket_id</b>	BIGSERIAL	PK	NOT NULL	Auto incrementable		Identificador del histórico de atención del ticket
<b>fecha_cambio_ticket</b>	TIMESTAMP		NOT NULL	Fecha y hora actual		Fecha (año-mes-día) y hora (horas:minutos:segundos) en la que se realiza un cambio en la atención del ticket
<b>motivo_rechazo</b>	VARCHAR (100)		NULL			Explicación del motivo del rechazo de la solución del ticket por parte del usuario Solicitante
<b>notas</b>	TEXT		NULL			Notas o comentarios por parte del técnico que está atendiendo el ticket
<b>ticket_id</b>	BIGSERIAL	FK	NOT NULL		Cascada	Llave foránea que hace referencia al ticket levantado
<b>tecnico_id</b>	BIGSERIAL	FK	NOT NULL		Restringida	Llave foránea que hace referencia al técnico que atiende al ticket
<b>actividad_id</b>	SERIAL	FK	NULL		En Nulo	Llave foránea que hace referencia a la actividad ponderada realizada
<b>actividad_extra_id</b>	SERIAL	FK	NULL		En Nulo	Llave foránea que hace referencia a la actividad ingresada manualmente por el técnico que atiende el ticket

Tabla: CAPACITACION		Descripción: Tabla en la que se almacenan los datos del ticket de capacitación				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
ticket_id	BIGSERIAL	PK, FK	NOT NULL		Restringida	Llave foránea con el identificador del ticket
requerimiento	VARCHAR (100)		NOT NULL			Descripción de la capacitación a solicitar
motivo_cap	TEXT		NOT NULL			Motivo por el cual se solicita una capacitación
fecha_atencion_cap	TIMESTAMP		NULL			Si no es NULL, indica que se trata de un ticket programado para ser atendido en dicha fecha y hora

Tabla: MANTENIMIENTO		Descripción: Tabla en la que se almacenan los datos del ticket de mantenimiento programado a laboratorios				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
ticket_id	BIGSERIAL	PK, FK	NOT NULL		Restringida	Llave foránea con el identificador del ticket
fecha_asignada	DATE		NOT NULL			Fecha en la que se realizará el mantenimiento programado
observacion_mnt	TEXT		NULL			Observación del mantenimiento programado realizado por el Encargado de Laboratorio
laboratorio_id	SERIAL	FK	NOT NULL		Restringida	Llave foránea que hace referencia al laboratorio en el que se realizará el mantenimiento programado

<b>Tabla: SOPORTE_TECNICO</b>		<b>Descripción:</b> Tabla en la que se almacenan los datos del ticket de soporte técnico				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
<b>ticket_id</b>	BIGSERIAL	PK, FK	NOT NULL		Restringida	Llave foránea con el identificador del ticket
<b>motivo_soporte</b>	VARCHAR (100)		NOT NULL			Razón por la cual se solicita el soporte técnico
<b>descripcion_problema</b>	TEXT		NOT NULL			Descripción del problema a solucionar con el soporte técnico
<b>num_inv_dispositivo</b>	VARCHAR (30)		NOT NULL			Número de inventario del dispositivo (se obtiene de la base de datos del Sistema de Inventarios)
<b>accesorios</b>	VARCHAR (100)		NULL			Accesorios a resguardo entregados al momento del levantamiento de ticket
<b>observaciones_dev</b>	VARCHAR (300)		NULL			Observaciones al dispositivo durante la atención del soporte
<b>fecha_atencion_st</b>	TIMESTAMP		NULL			Si no es NULL, indica que es un ticket programado para ser atendido en dicha fecha y hora
<b>mantenimiento_id</b>	BIGSERIAL	FK	NULL		En Nulo	Si no es NULL, esta llave foránea hace referencia al ticket de mantenimiento programado a laboratorio

<b>Tabla: HISTORICO_DISPOSITIVO</b>		<b>Descripción:</b> Tabla en la que se almacena el historial de cada dispositivo atendido en los tickets de soporte técnico				
Nombre atributo	TIPO	Llave	Obligatoriedad	Omisión	Integridad	Descripción
<b>historico_dispositivo_id</b>	BIGSERIAL	PK	NOT NULL	Auto incrementable		Identificador del historial del dispositivo
<b>num_inv_dispositivo</b>	VARCHAR (30)		NOT NULL			Número de inventario del dispositivo (se obtiene de la base de datos del Sistema de Inventarios)
<b>fecha_cambio_dev</b>	TIMESTAMP		NOT NULL	Fecha y hora actual		Fecha (año-mes-día) y hora (hora:minutos:segundos) en la que se registra un cambio en el dispositivo
<b>cambio_realizado</b>	VARCHAR (100)		NOT NULL			Cambio que se realizó en el dispositivo
<b>ticket_id</b>	BIGSERIAL	FK	NOT NULL		Restringida	Llave foránea que hace referencia al ticket levantado
<b>tecnico_id</b>	BIGSERIAL	FK	NOT NULL		Restringida	Llave foránea que hace referencia al técnico que realizó el cambio



*Historial de cambios*

MOTIVO DEL CAMBIO	AUTOR DEL CAMBIO	DESCRIPCIÓN	FECHA	VERSIÓN
Creación del documento	Angel Brito Segura	Desarrollo de la primera versión del modelo relacional	11/02/2022	1.0
Observaciones al modelo relacional	Angel Brito Segura	Cambios implementados en el modelo relacional debido a las observaciones del Ing. Armando Vázquez Díaz (líder técnico en desarrollo y bases de datos de la UC)	04/03/2022	1.1
Modificación en reglas de integridad	Angel Brito Segura	Cambios realizados en el modelo relacional para subir al servidor de la base de datos en producción	14/07/2022	1.2
Modificación de tabla USUARIO	Angel Brito Segura	Inserción de columna last_login para modelo User de Django	19/08/2022	1.3
Eliminación de constraints	Angel Brito Segura	Eliminación de verificación en URLs de tutoriales y restricción de no nulo en la columna fecha_termino	05/09/2022	1.4
Modificación de tabla ACT_PROPOSTA_TICKET	Angel Brito Segura	Cambio de llave primaria compuesta a subrogada y se agrega relación con la tabla ACTIVIDAD_EXTRA	22/01/2023	1.5
Modificación de tabla ACTIVIDAD_EXTRA	Angel Brito Segura	Restricción de unicidad a la columna desc_act_extra	15/02/2023	1.6

**Sistema de Gestión de Tickets**  
**Informe de Resultados y Cierre de Proyecto**

Versión: 0102

Fecha: 24/04/2023



**Sistema de Gestión de Tickets**  
**Informe de Resultados y Cierre de Proyecto**



## HOJA DE CONTROL

<b>Organismo</b>	Universidad Nacional Autónoma de México		
<b>Proyecto</b>	Sistema de Gestión de Tickets		
<b>Entregable</b>	Informe de Resultados y Cierre de Proyecto		
<b>Autor</b>	Angel Brito Segura		
<b>Versión/Edición</b>	0102	<b>Fecha Versión</b>	24/04/2023
<b>Aprobado por</b>		<b>Fecha Aprobación</b>	
		<b>Nº Total de Páginas</b>	14

## REGISTRO DE CAMBIOS

<b>Versión</b>	<b>Causa del Cambio</b>	<b>Responsable del Cambio</b>	<b>Fecha del Cambio</b>
0100	Versión inicial	Angel Brito Segura	03/04/2023
0101	Ajustes Índice	Angel Brito Segura	06/04/2023
0102	Ajustes diagramas	Angel Brito Segura	24/04/2023

## CONTROL DE DISTRIBUCIÓN

<b>Nombre y Apellidos</b>
Angel Brito Segura



## ÍNDICE

1	INTRODUCCIÓN .....	4
1.1	Objeto.....	4
1.2	Alcance .....	4
2	DESCRIPCIÓN DEL PROYECTO .....	5
3	EVALUACIÓN DEL PROYECTO .....	6
4	ASPECTOS RELEVANTES.....	7
4.1	Factores positivos .....	7
4.2	Factores negativos.....	7
4.3	Problemas superados.....	8
5	NUEVAS INICIATIVAS DETECTADAS .....	9
6	ANEXOS.....	10
6.1	Arquitectura del SGT .....	10
6.2	Ticket de Soporte Técnico .....	10
6.3	Ticket de Capacitación.....	11
6.4	Ticket de Mantenimiento Programado.....	11
7	GLOSARIO.....	13
8	REFERENCIAS .....	14



## Sistema de Gestión de Tickets Informe de Resultados y Cierre de Proyecto



# 1 INTRODUCCIÓN

## 1.1 Objeto

El presente documento tiene como propósito informar a la Unidad de Cómputo (UC) de la División de Ingenierías Civil y Geomática (DICyG) las actividades que se llevaron a cabo para la creación de un software hecho a la medida intitolado *Sistema de Gestión de Tickets (SGT)* y los resultados obtenidos con la puesta en producción en el servidor proporcionado para tal efecto.

## 1.2 Alcance

Este informe de resultados y cierre del proyecto va dirigido a la UC de la DICyG a fin de dar a conocer el estatus de entrega del SGT para su revisión y trabajos futuros, incluyendo inversiones venideras que a este respecto realice la propia institución.



**Sistema de Gestión de Tickets**  
**Informe de Resultados y Cierre de Proyecto**



## **2 DESCRIPCIÓN DEL PROYECTO**

El Sistema de Gestión de Tickets (SGT) es un sitio web, compuesto por módulos, que llevará un control de las solicitudes (tickets) de soporte técnico y de capacitación que realiza la Unidad de Cómputo (UC) de la División de Ingenierías Civil y Geomática (DICyG). Este sitio web, disponible desde cualquier parte del mundo, privilegia en todo momento la atención integral de la solicitud y la entera satisfacción del usuario. Actualmente, el SGT se encuentra en producción y en espera de integrar a los usuarios Solicitantes que forman parte del personal administrativo y académico de la DICyG para empezar a ser utilizado en un ambiente real, con N número de solicitudes de soporte técnico atendiéndose al mismo tiempo.

El SGT tiene como finalidad acelerar y mejorar la atención proporcionada por el staff de la UC, cuyos integrantes, inclusive, se hayan posibilitados para interactuar con el usuario Solicitante a través de foros de comunicación que permiten una atención más personalizada y en tiempo real.

La creación de este proyecto se realizó de manera local con la instalación de herramientas de software relacionadas con la gestión del sistema. El líder de proyecto realizó las actividades de programación, así como el diseño para las funcionalidades que realiza cada rol en el sistema. La metodología utilizada fue basada en SCRUM para favorecer las entregas y, por ende, el total cumplimiento -en tiempo y forma- de cada uno de los requerimientos establecidos por el cliente.

Este sistema constituirá una central digital de atención a usuarios, disponible desde cualquier punto, para los solicitantes que requieran de apoyo técnico, focalizando en todo momento la cabal atención de la solicitud y, por ende, la de todos los usuarios; dando paso, inclusive, en un futuro, a la necesidad de analizar las posibilidades de contar con personal de apoyo para un aprovechamiento integral del sistema.



**Sistema de Gestión de Tickets  
Informe de Resultados y Cierre de Proyecto**



### **3 EVALUACIÓN DEL PROYECTO**

Se realizaron una serie de pruebas para comprobar la calidad del propio software. Lo fueron de distintos tipos: de rendimiento del propio sitio web para garantizar el soporte de carga de usuarios esperada (staff de la Unidad de Cómputo y personal administrativo y académico de la DICyG); las efectuadas para comprobar todas las funcionalidades, incluyendo en ellas las de integración para validar la interacción entre los módulos que componen al SGT y la propia interacción de éste con el Sistema de Inventarios, sin soslayar la importancia de las pruebas unitarias para validar que la entrada y salida de información sean las correctas de acuerdo a las reglas de negocio definidas por el cliente.

Con lo anterior, se puso a prueba cada uno de los módulos implementados al tiempo que también se iban implementando en el servidor. Consecuentemente, con base a lo expuesto, la robustez con que cuenta el sistema de reciente creación queda totalmente garantizada.

Se cumplió con todos y cada uno de los hitos que se detallan en el acta constitutiva del proyecto, así como con los entregables comprometidos al inicio del mismo. Por considerarlos muy puntuales y significativos para la experiencia de usuario, se desarrollaron adicionalmente dos trípticos, los cuales consagran los aspectos más importantes a seguir en determinados procesos que integran el SGT.

En suma, la satisfacción por parte de la UC es alta debido a que se cumplió con los objetivos planteados al inicio del proyecto y dado a que la solución de la problemática que presentaban con la gestión de las solicitudes de soporte técnico se haya resuelta con este nuevo sitio web que se adapta a los procesos manuales que actualmente se realizan en esta unidad administrativa al momento de solicitar soporte técnico.



## 4 ASPECTOS RELEVANTES

La pretensión de la creación de manuales es documentar lo hecho en el proyecto. En tal virtud, se proporcionaron al cliente los siguientes manuales que consagran la información necesaria para tener una visión global del SGT:

- **Manual de usuario:** describe las funcionalidades que tiene cada rol en el SGT.
- **Manual técnico de desarrollo:** detalla el procedimiento y lógica realizada en el desarrollo del SGT.
- **Manual administrativo:** indica la serie de pasos para realizar actualizaciones y resolver problemas que se pudieran presentar al utilizar el SGT.

A continuación se describen los factores, tanto positivos como negativos, y los problemas superados al momento de estar realizando el presente proyecto:

### 4.1 Factores positivos

- ✓ Integración del SGT con el Sistema de Inventarios en producción (solo se permite realizar operaciones de lectura en este sistema desarrollado con anterioridad por la UC).
- ✓ Control del ciclo de vida del ticket a través de solicitudes realizadas en el sitio web, con una serie de notificaciones enviadas a todos los involucrados y que informan cada cambio presentado en el estatus de la solicitud.
- ✓ Manejo del Ticket de Mantenimiento Programado como entidad unitaria en la atención de los soportes técnicos relacionados al mantenimiento de cada laboratorio.
- ✓ Reparto equitativo de actividades realizadas por cada técnico del staff de la UC gracias a la definición de roles.
- ✓ Verificación de los soportes realizados por técnicos incipientes para garantizar la calidad del servicio y la resolución del problema del usuario Solicitante.

### 4.2 Factores negativos

- ✱ Independencia en la paginación cuando se tiene más de una lista de registros en la misma vista del SGT.
- ✱ Falta de unicidad en números de inventario.
- ✱ Múltiples responsables de los dispositivos registrados en el Sistema de Inventarios.
- ✱ Falta de inmediatez en el SGT al momento de enviar más de una notificación de correo electrónico.





### 4.3 Problemas superados

- **Instalación de versión correcta de Python:** el primer servidor proporcionado por la UC contaba con Debian 8 como SO, lo que limitaba actualizar a la versión de Python 3 porque afectaba otros entornos de desarrollo puestos en este servidor. La solución fue la creación de una nueva máquina virtual para el SGT con un SO actualizado.
- **Acceso a documentación del Sistema de Inventarios:** al momento de realizar el diseño del SGT no se contaba con documentación acerca de cómo estaba creada la BD de este sistema, conocimiento que se necesitaba para poder realizar la conexión entre ambos sistemas. En aras de solución, se pidió el acceso al sistema en cuestión con privilegios mínimos, así como una copia de seguridad de la BD para poder trabajar con esta información de manera local.
- **Redundancia de datos en el Sistema de Inventarios:** con la creación de modelos y acceso a la información desde el SGT, se detectaron números de inventario y de trabajador repetidos en el Sistema de Inventarios, por lo que se implementaron mecanismos en el desarrollo del proyecto para evitar que esta duplicidad de información se registrará en el sistema de reciente creación.
- **Comunicación faltante entre la BD del Sistema de Inventarios y el Sistema de Gestión de Tickets:** en el servidor de producción se encontró una falta de comunicación entre la base de datos del Sistema de Inventarios (también en producción) y la máquina virtual que hospeda el SGT. Esto se resolvió permitiendo la comunicación bidireccional a través de la red local en donde se encuentran hospedados ambos servidores.



## **5 NUEVAS INICIATIVAS DETECTADAS**

Se ha cumplido íntegramente con todos y cada uno de los requerimientos del proyecto. Sin embargo, se proponen las siguientes mejoras al SGT:

- ▶ Trabajar en el filtrado de los números de inventario repetidos en el Sistema de Inventarios para evitar la duplicidad de los registros de los datos del dispositivo sobre el cual se va a realizar el soporte técnico en el Sistema de Gestión de Tickets.
- ▶ Instaurar la homologación del usuario final en el Sistema de Inventarios para poder ser consultado desde el Sistema de Gestión de Tickets, de manera más efectiva, con el número de trabajador/cuenta.
- ▶ Agregar procesos asíncronos para que, por un lado, el usuario final obtenga una respuesta más rápida a sus solicitudes y, por el otro, el servidor se encargue de realizar procesos tardados en segundo plano.
- ▶ Manejar la concurrencia para el caso de alta disponibilidad. Lo anterior, a fin de que sea factible el hecho de que varios usuarios con el mismo rol puedan modificar un recurso sin afectar los cambios realizados por otro usuario en ese mismo instante.
- ▶ Adicionar a las actividades ponderadas aquellas que se registran en el SGT de manera constante y reiterada en la atención de los tickets.
- ▶ Poder exportar en formatos CSV, Excel y/o PDF, los reportes generados en el SGT para una mejor manipulación y creación de otro tipo de gráficas necesarias para la administración operativa de la UC.
- ▶ Programar recordatorios para los usuarios solicitantes que aún no hayan validado la atención de sus tickets, así como al usuario solicitante y técnico en cuestión cuando se indique una fecha programada de atención a la solicitud.
- ▶ Desconectar al usuario después de minutos de inactividad para evitar problemas con el uso inadecuado de su cuenta.
- ▶ Manipular el estado del dispositivo registrado en el Sistema de Inventarios desde el SGT para que esté actualizado conforme al estatus que guarda la solicitud de soporte solicitada.

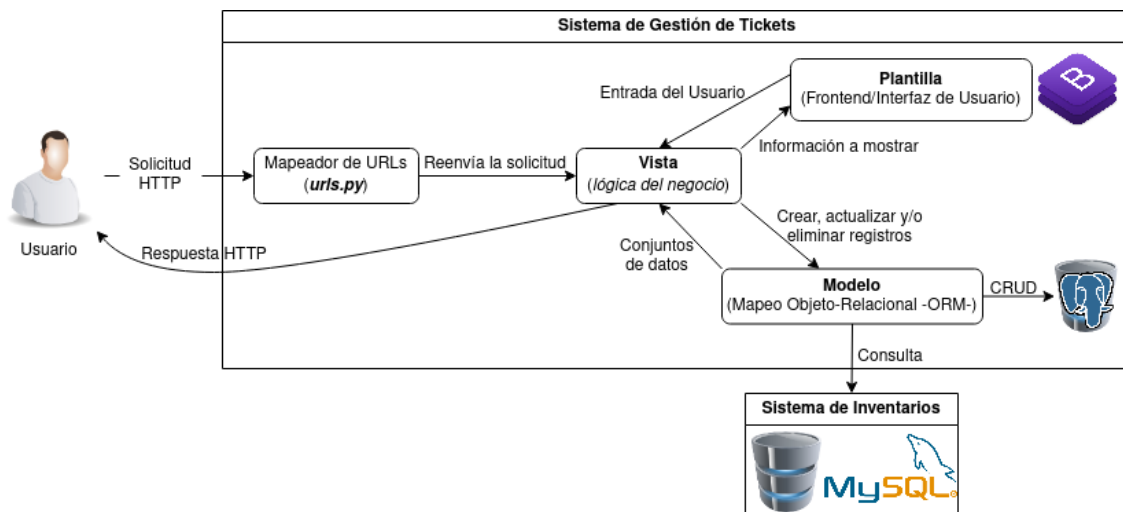


## 6 ANEXOS

### 6.1 Arquitectura del SGT

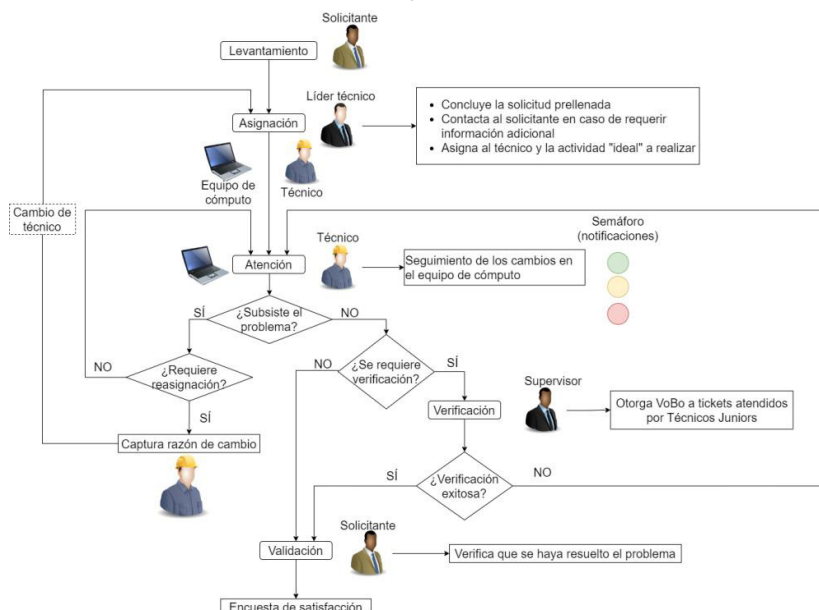
Para la creación del Sistema de Gestión de Tickets se implementó el patrón de diseño MTV utilizado por Django, framework de backend por el que se optó para el desarrollo del presente proyecto.

En la siguiente imagen se describe gráficamente el comportamiento del sistema en el momento de que un usuario realiza una petición al sitio web:



### 6.2 Ticket de Soporte Técnico

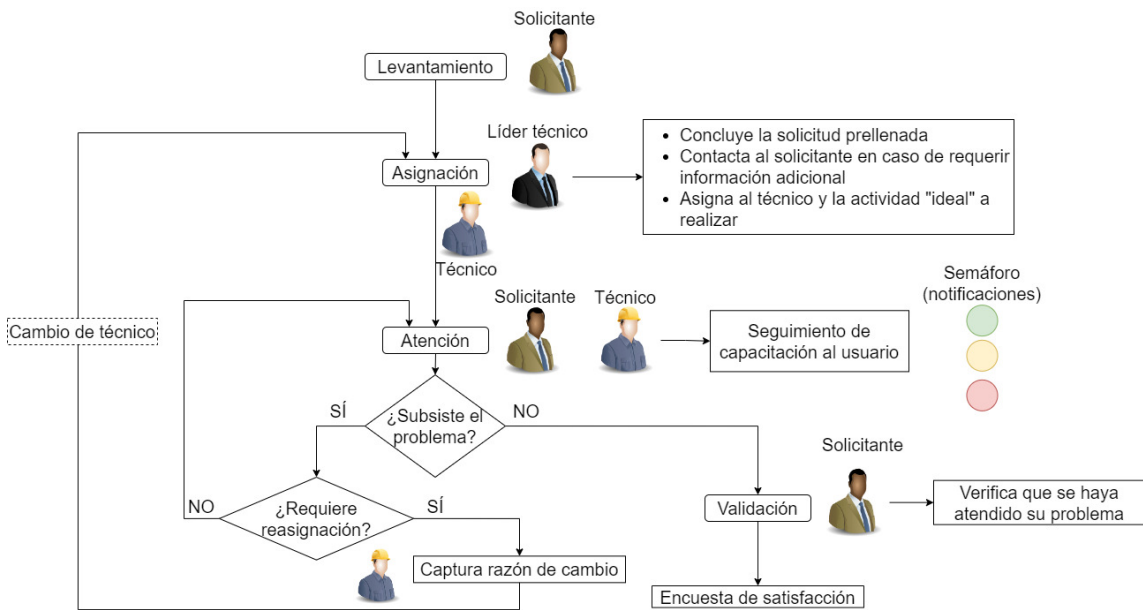
En el diagrama que se muestra a continuación se observa el flujo que se realiza al momento de atender una solicitud de Soporte Técnico en el SGT:





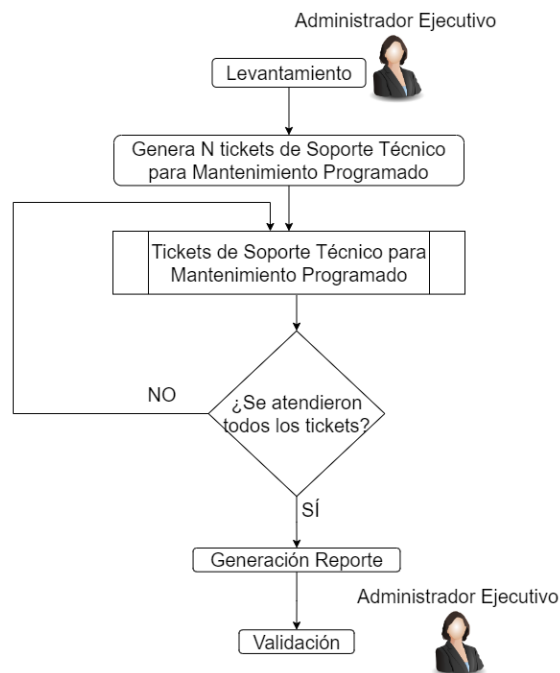
### 6.3 Ticket de Capacitación

En el diagrama que se muestra a continuación se observa el flujo que se realiza al momento de atender una solicitud de Capacitación en el SGT:



### 6.4 Ticket de Mantenimiento Programado

En el diagrama que se muestra a continuación se observa el flujo que se realiza en la atención del Ticket de Mantenimiento Programado en el SGT:

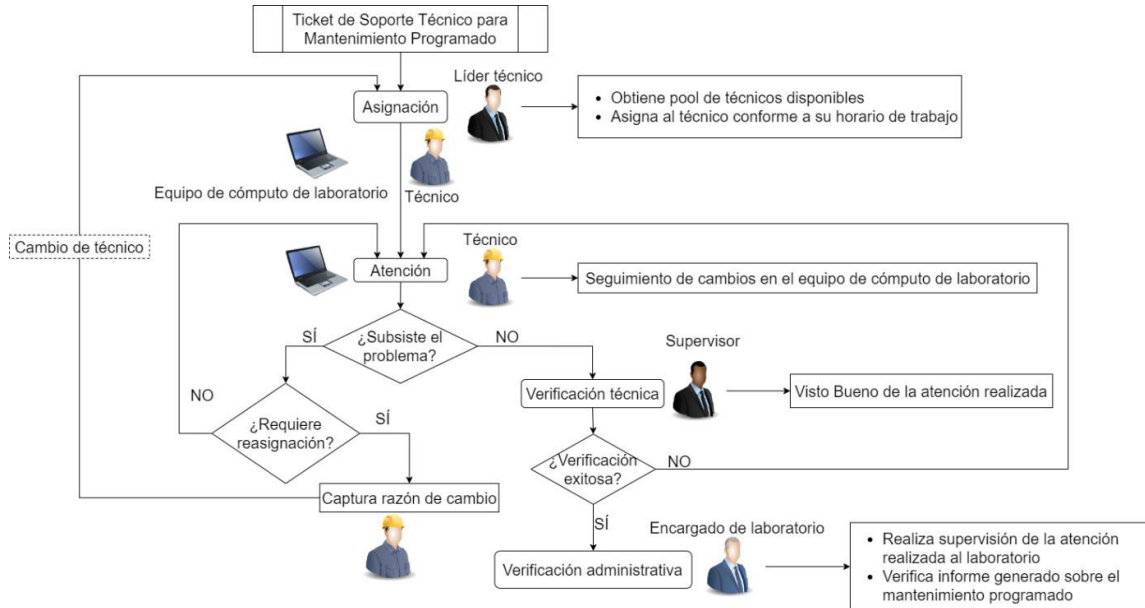




## Sistema de Gestión de Tickets Informe de Resultados y Cierre de Proyecto



Como muestra el diagrama anterior, este tipo de solicitud genera un ticket de soporte técnico por cada dispositivo registrado en el laboratorio donde se realizará el mantenimiento, por lo que el flujo de este ticket autogenerado se detalla en el siguiente diagrama:





**Sistema de Gestión de Tickets**  
**Informe de Resultados y Cierre de Proyecto**



## 7 GLOSARIO

<b>Término</b>	<b>Descripción</b>
backend	Capa en el desarrollo web que se encarga de procesar la información recibida por el usuario y realizar toda la lógica para que funcione el sitio web.
BD	Siglas referentes a Base de Datos. BD es una recopilación organizada de información o datos estructurados, almacenados de forma electrónica en un sistema informático.
conurrencia	Capacidad de un algoritmo o programa para ejecutar más de una tarea a la vez.
CSV	Siglas referentes al formato de archivo de texto en el que los Valores están Separados por Comas, permitiendo guardar los datos en una tabla estructurada con filas y columnas.
Debian	Sistema operativo libre, desarrollado y mantenido por el proyecto con el mismo nombre, siendo una distribución de Linux de código abierto.
framework	Marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos. Se trata de una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software.
hito	Acontecimiento puntual y significativo que marca un momento importante en el desarrollo del proyecto.
MTV	Patrón de diseño que utiliza el framework Django: Modelo, Plantilla, Vista por sus siglas en inglés.
proceso asíncrono	Proceso que ejecuta una tarea "en segundo plano" sin que el usuario tenga que esperar a que ésta finalice.
Python	Lenguaje de programación interpretado, ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el Machine Learning.
SCRUM	Marco de gestión de proyectos de metodología ágil que busca que los equipos aprendan, a través de las experiencias, a autoorganizarse mientras abordan un problema y a reflexionar sobre los resultados obtenidos para mejorar continuamente.
SO	Siglas referentes a Sistema Operativo, el cual es un conjunto de programas responsables de la conexión entre los recursos de hardware en una computadora y las aplicaciones de software que el usuario tiene acceso a través de una interfaz, ya sea gráfica o de línea de comandos.
software	Componente intangible que forma parte de los dispositivos electrónicos, el cual permite su funcionamiento con un conjunto de aplicaciones y programas diseñados para cumplir diversas funciones dentro de un sistema.



**Sistema de Gestión de Tickets**  
**Informe de Resultados y Cierre de Proyecto**

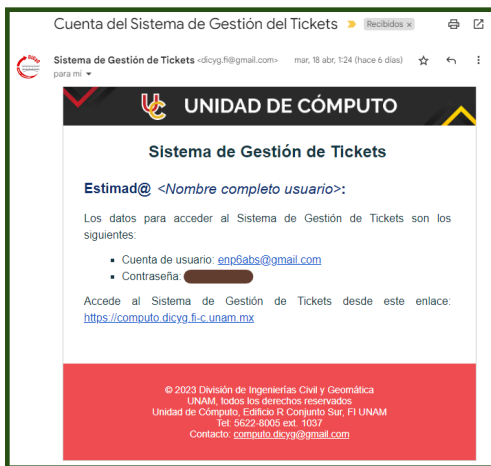


## 8 REFERENCIAS

Referencia	Título
<a href="https://www.loadview-testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/">https://www.loadview-testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/</a>	Tipos de pruebas de software: diferencias y ejemplos [consultado 04/04/23]
<a href="https://www.atlassian.com/es/agile/scrum">https://www.atlassian.com/es/agile/scrum</a>	¿Qué es scrum? [consultado 05/04/23]
<a href="https://www.atinternet.com/es/glosario/os-sistema-operativo/">https://www.atinternet.com/es/glosario/os-sistema-operativo/</a>	OS (Sistema operativo) [consultado 05/04/23]
<a href="https://www.debian.org/intro/index#community">https://www.debian.org/intro/index#community</a>	Introducción a Debian [consultado 05/04/23]
<a href="https://www.oracle.com/mx/database/what-is-database/">https://www.oracle.com/mx/database/what-is-database/</a>	¿Qué es una base de datos? [consultado 05/04/23]
<a href="https://aws.amazon.com/es/what-is/python/">https://aws.amazon.com/es/what-is/python/</a>	¿Qué es Python? [consultado 05/04/23]
<a href="https://www.geeknetic.es/Archivo-CSV/que-es-y-para-que-sirve">https://www.geeknetic.es/Archivo-CSV/que-es-y-para-que-sirve</a>	¿Qué es un archivo CSV y para qué sirve? [consultado 05/04/23]
<a href="https://concepto.de/software/">https://concepto.de/software/</a>	Software [consultado 06/04/23]
<a href="https://trailhead.salesforce.com/es-MX/content/learn/modules/asynchronous_apex/async_apex_introduction">https://trailhead.salesforce.com/es-MX/content/learn/modules/asynchronous_apex/async_apex_introduction</a>	Fundamentos del procesamiento asíncrono [consultado 06/04/23]
<a href="https://uniwebsidad.com/libros/django-1-0/capitulo-5/el-patron-de-diseno-mtv">https://uniwebsidad.com/libros/django-1-0/capitulo-5/el-patron-de-diseno-mtv</a>	El patrón de diseño MTV [consultado 06/04/23]
<a href="https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programacion-de-una-aplicacion-web">https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programacion-de-una-aplicacion-web</a>	Backend y Frontend, ¿Qué es y cómo funcionan en la programación? [consultado 06/04/23]
<a href="https://es.theastrologypage.com/concurrency">https://es.theastrologypage.com/concurrency</a>	¿Qué es la concurrencia? [consultado 06/04/23]
<a href="https://www.edix.com/es/instituto/framework/">https://www.edix.com/es/instituto/framework/</a>	Framework [consultado 06/04/23]

## INGRESO AL SISTEMA DE GESTIÓN DE TICKETS

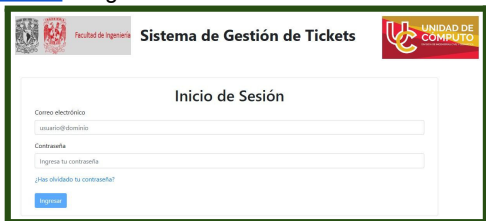
Una vez que el staff de la Unidad de Cómputo registra sus datos y su rol correspondiente en el sistema, recibirá la siguiente notificación en el correo electrónico que proporcionó durante su registro:



Con el correo electrónico y contraseña autogenerada por el sistema, podrá acceder a él desde la siguiente dirección:

<https://computo.dicyg.fi-c.unam.mx/>

Es necesario autenticarse para poder entrar al sistema, por lo que se tiene que dar clic en [Iniciar sesión](#) e ingresar los datos solicitados:



## División de Ingenierías Civil y Geomática

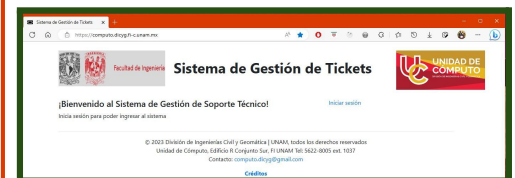
**M.I. Octavio García Domínguez**  
Jefe de la División

**M.I. Tanya Itzel Arteaga Ricci**  
Jefa de la Unidad de Cómputo

**LÍDER DE DESARROLLO**  
Angel Brito Segura

Asesora Técnica  
Dra. Martha Angélica Nakayama Cervantes

## Sistema de Gestión de Tickets de la DICYG



## Usuario Solicitante



Abril, 2023

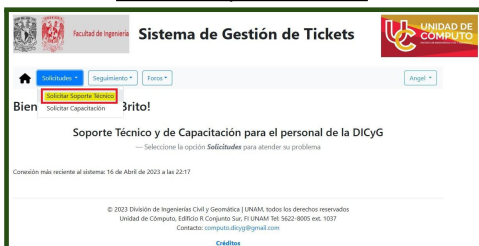


## LEVANTAMIENTO DE SOLICITUDES

Una vez que se ingresa al Sistema de Gestión de Tickets, se pueden requerir de la Unidad de Cómputo de la DICyG dos tipos de solicitudes (tickets): de Soporte Técnico y de Capacitación.

### → Soporte Técnico

En este tipo de solicitud se realiza una petición de soporte técnico para un dispositivo registrado en el [Sistema de Inventarios de la DICyG](#) asociado al usuario autenticado en el sistema. Para ello, dar clic en el módulo **Solicitudes** y seleccionar **Solicitar Soporte Técnico**:



### → Capacitación

En este tipo de solicitud se proporcionará asesoría sobre las herramientas y/o tecnologías que requiera conocer el usuario autenticado en el sistema para el desempeño de sus actividades diarias. Para ello, dar clic en **Solicitudes** y de las opciones mostradas, seleccionar **Solicitar Capacitación**:



**IMPORTANTE.-** Se recomienda ampliamente visitar el [Foro de Asesoría Tecnológica](#) antes de realizar el levantamiento de cualquier solicitud, debido a que en él se encuentran tutoriales que pueden resolver su problema sin necesidad del levantamiento de un ticket.

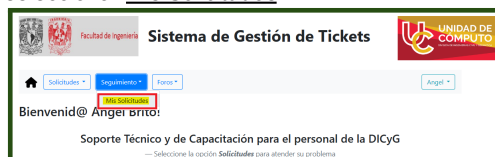
Si no se encuentra un tutorial que resuelva la problemática actual, se procede a realizar el llenado del formulario correspondiente mostrado en pantalla de acuerdo al tipo de solicitud.

El sistema enviará una notificación de confirmación por correo electrónico sobre el levantamiento del ticket y contactará al staff de la UC para informar sobre esta nueva solicitud.

## SEGUIMIENTO DE SOLICITUDES

Es importante mantenerse al tanto del correo electrónico registrado en el sistema debido a que será el medio por cual se notificarán los cambios que se realicen en la atención del ticket.

Para mostrar el seguimiento de las solicitudes levantadas, dar clic en el módulo de **Seguimiento** y, de las opciones desplegadas, seleccionar **Mis Solicitudes**:

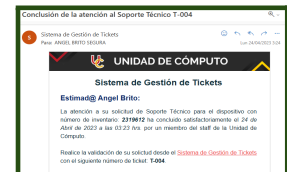


Desde esta vista es posible acceder al [Foro de Comunicación](#) -con la UC- de cualquiera de sus solicitudes dando clic en el botón **Enviar mensaje**:

Seguimiento a Soportes Técnicos				
# Ticket	Dispositivo	Estatus	Actualización	
T-004	2319612 - CPU DELL OptiPlex 780	En proceso	19 de Abril de 2023 a las 22:06	<a href="#">Enviar mensaje</a>
T-005	1489114 - CPU ACER ACER BASIC	Levantado	16 de Abril de 2023 a las 15:19	<a href="#">Enviar mensaje</a>

## VALIDACIÓN Y CIERRE DE ATENCIÓN

Concluida la atención del ticket por parte del staff de la UC, se recibirá una notificación por correo electrónico para realizar la validación correspondiente:



Para validar que la atención fue la correcta, es posible -desde esta misma notificación- dar clic en **Sistema de Gestión de Tickets**; o bien, entrar directamente al sistema donde podrá ingresar al módulo **Seguimiento** en la sección **Mis Solicitudes** y dar clic en el botón **Validar**:

Se desplegarán las actividades realizadas en la atención del ticket; aquí podrá validarse la correcta atención al dar clic en el botón **Cerrar Ticket**, o bien (en caso de no serla) **Invaldar Solución** con el botón de mismo nombre.

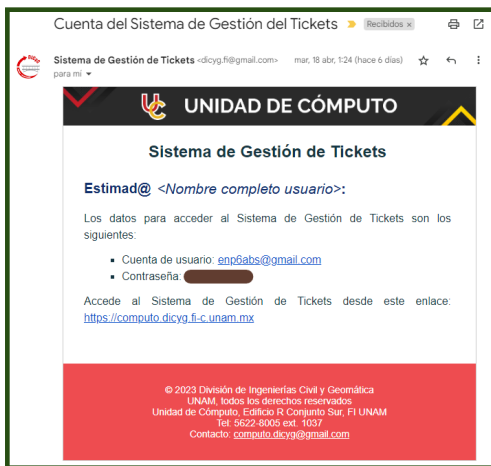
Si se invalida, se solicitará ingresar un motivo del rechazo y, opcionalmente, agregar comentarios al técnico que realizó dicha atención. Esta información le será notificada a éste último, quien de nueva cuenta dará solución a la solicitud.

En caso de validar la atención, se mostrará una breve encuesta de satisfacción para realizar el cierre de la solicitud. Al dar clic al botón **Enviar Encuesta** se concluye la atención del ticket y se envía por correo electrónico una notificación de confirmación, al igual que al técnico que atendió la solicitud, con la calificación y comentarios que le otorgue el usuario Solicitante.



## INGRESO AL SISTEMA DE GESTIÓN DE TICKETS

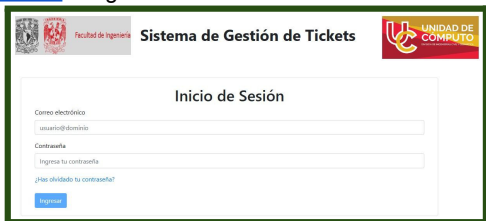
Una vez que el staff de la Unidad de Cómputo registra sus datos y su rol correspondiente en el sistema, recibirá la siguiente notificación en el correo electrónico que proporcionó durante su registro:



Con el correo electrónico y contraseña autogenerada por el sistema, podrá acceder a él desde la siguiente dirección:

<https://computo.dicyg.fi-c.unam.mx/>

Es necesario autenticarse para poder entrar al sistema, por lo que se tiene que dar clic en [Iniciar sesión](#) e ingresar los datos solicitados:



## División de Ingenierías Civil y Geomática

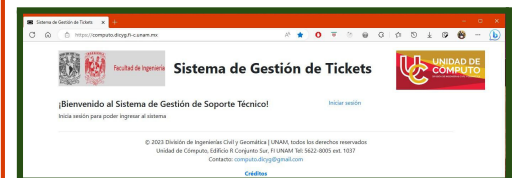
**M.I. Octavio García Domínguez**  
Jefe de la División

**M.I. Tanya Itzel Arteaga Ricci**  
Jefa de la Unidad de Cómputo

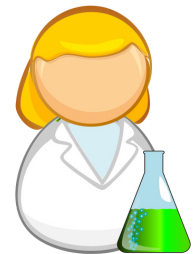
**LÍDER DE DESARROLLO**  
Angel Brito Segura

Asesora Técnica  
Dra. Martha Angélica Nakayama Cervantes

## Sistema de Gestión de Tickets de la DICYG



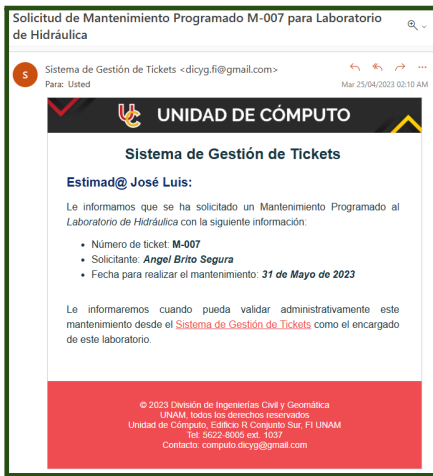
**Encargado de Laboratorio**



*Abril, 2023*

## TICKET DE MANTENIMIENTO PROGRAMADO

Este ticket sólo puede ser levantado por usuarios con el rol **Administrador Ejecutivo** y se notifica precisamente de esta acción al encargado asociado al laboratorio donde se realizará el mantenimiento:



## SEGUIMIENTO DE SOLICITUDES

Es importante estar al tanto del correo electrónico registrado en el sistema debido a que será el medio por el cual se notificará de los cambios que se realicen en la atención del mantenimiento programado.

Para mostrar un seguimiento de las solicitudes levantadas, dar clic en el módulo de **Seguimiento** y, de las opciones desplegadas, seleccionar **Mis Solicitudes**:

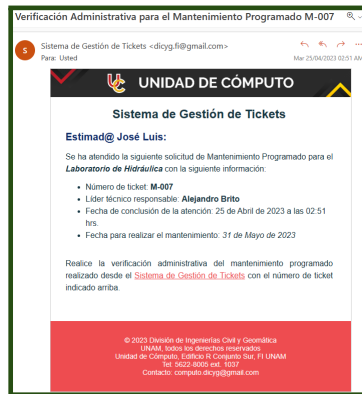


En esta vista puede seguir los cambios en el estatus del ticket, así como sus fechas de actualización:



## VERIFICACIÓN ADMINISTRATIVA

Una vez que se ha concluido la verificación técnica del mantenimiento programado, se notificará que es posible realizar la verificación que le corresponde al encargado del laboratorio donde se realizó el mantenimiento:



Para realizar la verificación administrativa del mantenimiento, dar clic en **Sistema de Gestión de Tickets** de la notificación recibida; o bien, entrar al módulo **Seguimiento** en la sección de **Mis Solicitudes** y dar clic en el botón **Verificar**:



En el reporte generado, es posible **Verificar** o **Rechazar** la atención realizada en cada dispositivo del laboratorio de manera individual al dar clic en el botón que corresponda la acción a realizar:



En caso de rechazar la atención dada a un dispositivo del mantenimiento programado, se solicitará ingresar un motivo del rechazo y, opcionalmente, agregar comentarios al técnico que realizó dicha atención. Esta información será notificada al técnico para que vuelva a realizar la atención al dispositivo en mantenimiento programado.

Adicionalmente, es posible agregar observaciones al mantenimiento programado en el campo correspondiente, dando clic en el botón **Registrar observaciones** para que esta información permanezca en el sistema:



Para concluir la verificación administrativa, dar clic en el botón **Verificar mantenimiento** donde se valida que no haya ninguna atención rechazada. Se notifica al **Administrador Ejecutivo** -que realizó el levantamiento de esta solicitud de mantenimiento programado-, que se ha realizado esta verificación, por lo que es posible cerrar el ticket.

# Referencias

- [1] J. Moreno. (2021, Noviembre) ¿Qué es el soporte técnico? Definición y características. Accedido 10/11/2022. HubSpot, Inc. [En línea]. Disponible en: <https://blog.hubspot.es/service/que-es-soporte-tecnico>
- [2] K. Limehouse. “What does an IT Support Specialist do?” en *Google Technical Support Fundamentals*. Google | Coursera, 2018. [En línea]. Disponible en: <https://es.coursera.org/lecture/technical-support-fundamentals/what-does-an-it-support-specialist-do-cThX5>
- [3] M. Abad Grau, M. J. Hornos Barranco, M. V. Hurtado Torres, & R. Montes Soldado. “HELP DESK: SOPORTE TÉCNICO PARA LA EMPRESA DEL SIGLO XXI” *III Encuentro Iberoamericano de Finanzas y Sistemas de la Información (EFSI’02)*, 2008.
- [4] Zendesk. (2021, Febrero) Definición de soporte técnico y características principales. Accedido 13/11/2022. [En línea]. Disponible en: <https://www.zendesk.com.mx/blog/definicion-de-soporte-tecnico/>
- [5] Y. Oval Riverón, A. Pérez Benítez, & S. Sánchez Ortiz. “Sistema Experto para la gestión de la Base de Conocimientos de Novadesk” Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas, 2011.
- [6] D. da Silva. (2022, Agosto) ¿Qué es un ticket de soporte técnico y para qué sirve? Accedido 15/11/2022. Zendesk. [En línea]. Disponible en: <https://www.zendesk.com.mx/blog/ticket-de-soporte-tecnico/>
- [7] M. Clark. “Ticketing Systems and Documenting Your Work” en *Technical Support Fundamentals*. Google | Coursera, 2018. [En línea]. Disponible en: <https://es.coursera.org/lecture/technical-support-fundamentals/ticketing-systems-and-documenting-your-work-OWWVd>
- [8] D. da Silva. (2021, Junio) ¿Qué es base de conocimiento? Crea la tuya en 3 pasos. Accedido 13/11/2022. Zendesk. [En línea]. Disponible en: <https://www.zendesk.com.mx/blog/que-es-base-de-conocimiento/>

- [9] Bugzilla. Accedido 13/11/2022. [En línea]. Disponible en: <https://www.bugzilla.org/>
- [10] J. W. Camacho Ledezma, Y. E. Pariente Ulloa, & C. A. Quiroga Coronado. “SISTEMA DE SEGUIMIENTO DE ERRORES BUGZILLA” Universidad Mayor de San Simón, Bolivia. Reporte Técnico, 2021. [En línea]. Disponible en: <https://www.studocu.com/bo/document/universidad-mayor-de-san-simon/seguridad-de-sistemas/bugzilla/13666094>
- [11] H. Ekbia & L. Gasser “Common ground: For a sociology of code” *Submitted to Information Technology and People, special issue on Social Theory* : <http://130.203> vol. 133.
- [12] Atlassian. Jira Service Management. Accedido 13/11/2022. [En línea]. Disponible en: <https://www.atlassian.com/es/software/helpdesk-software>
- [13] GB Advisors. (2016, Noviembre) Freshservice vs Jira Service Desk: Pros y contras en Gestión de Servicios. Accedido 13/11/2022. [En línea]. Disponible en: <https://www.gb-advisors.com/es/freshservice-vs-jira-service-desk/>
- [14] Servicios Gestionados. (2020, Octubre) Redmine como herramienta de gestión de proyectos. Accedido 13/11/2022. ILIMIT. [En línea]. Disponible en: <https://www.ilimit.com/blog/redmine-como-herramienta-de-gestion-de-proyectos/>
- [15] Centro de excelencia de software libre de Castilla La Mancha (CESLCAM). (2010, Marzo) Análisis de aplicación: Redmine. Accedido 13/11/2022. Junta de Comunidades de Castilla-La Mancha. [En línea]. Disponible en: <https://www.bilib.es/actualidad/blog/noticia/articulo/analisis-de-aplicacion-redmine/>
- [16] Drew. ¿Qué es Freshdesk? Accedido 13/11/2022. Global Business Consulting. [En línea]. Disponible en: <https://www.wearedrew.co/freshdesk>
- [17] Oracle. ¿Qué es SaaS (Software como servicio)? Accedido 20/11/2022. [En línea]. Disponible en: <https://www.oracle.com/mx/applications/what-is-saas/>
- [18] Optima Solutions. Ventajas de elegir Freshdesk. Accedido 13/11/2022. [En línea]. Disponible en: <https://www.optimasolutions.es/ventajas-de-elegir-freshdesk-como-solucion-de-helpdesk/>
- [19] HubSpot. PLATAFORMA DE CRM DE HUBSPOT. Accedido 20/11/2022. [En línea]. Disponible en: <https://www.hubspot.es/>
- [20] Salesforce. ¿Qué es un CRM? Accedido 20/11/2022. [En línea]. Disponible en: <https://www.salesforce.com/mx/crm/>

- [21] N. Gomez. (2020, Septiembre) ¿Es FRESHDESK una opción completa para integrarse con WHATSAPP? Accedido 13/11/2022. ZENVIA. [En línea]. Disponible en: <https://www.zenvia.com/es/blog/freshdesk-opcion-profesional-para-integrar-whatsapp-2/>
- [22] Santander Universidades. (2020, Diciembre) Metodologías de desarrollo de software: ¿qué son? [En línea]. Disponible en: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>
- [23] Ginzo Technologies. Tipos de Metodología clásica en Desarrollo de Software. Accedido 20/11/2022. [En línea]. Disponible en: <https://ginzo.tech/blog/metodologia-clasica-desarrollo-software/>
- [24] E. G. Maida & J. Pacienza. “Metodologías de desarrollo de software” 2015.
- [25] R. Demera. (2021, Febrero) Metodologías... ¿tradicional vs ágil? Tribalyte Technologies S.L. [En línea]. Disponible en: <https://tech.tribalyte.eu/blog-metodologias-tradicional-vs-agil>
- [26] Tecnológico en Informática. Tabla Comparativa de Modelos en Ingeniería de Software. Accedido 20/11/2022. Facultad de Ingeniería, Universidad de la República Uruguay. [En línea]. Disponible en: <https://www.fing.edu.uy/tecnoinf/mvd/cursos/ingsoft/material/teorico/is02b-Tabla%20Comparativa%20Modelos.pdf>
- [27] I. Sommerville. “Software processes” en *SOFTWARE ENGINEERING*. Addison-Wesley, 2011 pp. 29–50.
- [28] G. Mancuzo. (2020, Agosto) Programación Extrema: Pros y Contras. Accedido 20/11/2022. ComparaSoftware. [En línea]. Disponible en: <https://blog.comparasoftware.com/programacion-extrema-ventajas-desventajas/>
- [29] D. Mas. (2019, Marzo) Metodología Kanban: pros y contras en la gestión de proyectos. Accedido 20/11/2022. Fhios. [En línea]. Disponible en: <https://www.fhios.es/metodologia-kanban-pros-y-contras/>
- [30] J. C. Almanza, C. O. Félix Lachica, F. Quijada Cruz, O. A. Velazquez Barraza, & M. A. Llanez Jiménez. Ventajas y Desventajas Lean Development. Accedido 20/11/2022. [En línea]. Disponible en: <https://iswugleanddevelopment.wordpress.com/ventajas-y-desventajas/>
- [31] Develapps. (2018, Enero) Metodologías ágiles en el desarrollo de software. [En línea]. Disponible en: <http://www.develapps.com/es/noticias/metodologias-agiles-en-el-desarrollo-de-software>

- [32] Academia 3E. ¿Cuáles son las ventajas y desventajas Scrum? Accedido 20/11/2022. [En línea]. Disponible en: <https://academia3e.com/comunidad/ventajas-y-desventajas-de-scrum/>
- [33] Coco Solution. (2022, Noviembre) Las mejores tecnologías de desarrollo web. Accedido 14/11/2022. [En línea]. Disponible en: <https://cocosolution.com/mejores-tecnologias-web/>
- [34] Software Engineering Institute. Accedido 11/02/2023. Carnegie Mellon University. [En línea]. Disponible en: <https://www.sei.cmu.edu/>
- [35] L. Bass, P. Clements, & R. Kazman. *Software Architecture in Practice*. Addison Wesley, 2003.
- [36] P. Huet. (2022, Agosto) Arquitectura de software: Qué es y qué tipos existen. Accedido 20/11/2022. OpenWebinars. [En línea]. Disponible en: <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>
- [37] O. J. B. Iturralde. Arquitectura Cliente-Servidor. Accedido 20/11/2022. Oscar Blancarte Blog. [En línea]. Disponible en: <https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/cliente-servidor>
- [38] J. Fernández. (2021, Abril) ¿Cuáles son las mejores tecnologías para desarrollo web? Accedido 14/11/2022. [En línea]. Disponible en: <https://www.armadilloamarillo.com/blog/cuales-son-las-mejores-tecnologias-para-desarrollo-web/>
- [39] H. F. Korth, A. Silberschatz, & S. Sudarshan. “Introducción” en *Fundamentos de bases de datos*. L. G. Fernández, Ed. McGraw-Hill/Interamericana de España, S.A.U., 2006 pp. 1–24.
- [40] Felipe. (2020, Diciembre) Qué es MariaDB y cuáles son sus características. Accedido 03/09/2021. [En línea]. Disponible en: <https://www.hostingplus.mx/blog/que-es-mariadb-y-cuales-son-sus-caracteristicas/>
- [41] R. Rojo. (2018, Marzo) Requerimientos y procesos de instalación de MariaDB y Oracle. Accedido 03/09/2021. [En línea]. Disponible en: <https://prezi.com/p/avai9rik1agp/requerimientos-y-procesos-de-instalacion-de-mariadb-y-oracle/>
- [42] Ciberninjas. (2021, Marzo) 7 Mejores Software de Bases de Datos de Código Abierto. Accedido 03/09/2021. [En línea]. Disponible en: <https://ciberninjas.com/software-bases-datos-codigo-abierto/>
- [43] V. M. L. Garcia. Requerimientos e instalación del SGBD. Accedido 17/09/2021. [En línea]. Disponible en: <http://lopez-garcia-victor.blogspot.com/2012/08/11-requerimientos-e-instalacion-del-sgbd.html>

- [44] CYBERTEC. Beneficios de PostgreSQL. Accedido 03/09/2021. [En línea]. Disponible en: <https://www.cybertec-postgresql.com/es/postgresql/beneficios-de-postgresql/>
- [45] EnterpriseDB Corporation. (2016, Septiembre) PostgreSQL 9.6 Installation Guide. Accedido 03/09/2021. [En línea]. Disponible en: [http://get.enterprisedb.com/docs/PostgreSQL\\_Installation\\_Guide\\_v9.6.pdf](http://get.enterprisedb.com/docs/PostgreSQL_Installation_Guide_v9.6.pdf)
- [46] J. Segovia. (2018, Agosto) Ventajas y Desventajas de PostgreSQL. Accedido 20/11/2022. Todo PostgreSQL. [En línea]. Disponible en: <https://www.todopostgresql.com/ventajas-y-desventajas-de-postgresql/>
- [47] KeepCoding. (2022, Agosto) 7 tecnologías para desarrollo web. Accedido 14/11/2022. [En línea]. Disponible en: <https://keepcoding.io/blog/7-tecnologias-para-desarrollo-web/>
- [48] IBM. ¿Qué es Java Spring Boot? Accedido 20/04/2023. [En línea]. Disponible en: <https://www.ibm.com/mx-es/topics/java-spring-boot>
- [49] D. Pacheco. Los 10 frameworks de desarrollo web más solicitados en 2021. Accedido 03/09/2021. WordPress. [En línea]. Disponible en: <https://diegooo.com/frameworks-de-desarrollo-web-mas-solicitados-en-2021/>
- [50] ILIMIT. Flask vs. Django. Accedido 20/11/2022. [En línea]. Disponible en: <https://www.ilimit.com/blog/flask-vs-django/>
- [51] C. Herrera. Laravel vs CodeIgniter. Accedido 03/09/2021. [En línea]. Disponible en: <https://carlos-herrera.com/laravel-vs-codeigniter/>
- [52] M. Presta. Los 10 mejores marcos de frontend y backend. Accedido 20/11/2022. Back4App. [En línea]. Disponible en: <https://blog.back4app.com/es/los-10-mejores-marcos-de-frontend-y-backend/>
- [53] Meta Platforms. Requerimientos del entorno de JavaScript. Accedido 20/11/2022. [En línea]. Disponible en: <https://es.reactjs.org/docs/javascript-environment-requirements.html>
- [54] Google. (2022, Febrero) Deployment. Accedido 20/11/2022. [En línea]. Disponible en: <https://angular.io/guide/deployment>
- [55] Evan You & Vite Contributors. Building for production. Accedido 20/11/2022. [En línea]. Disponible en: <https://vitejs.dev/guide/build.html>
- [56] Rock Content. (2020, Abril) Bootstrap: guía para principiantes de qué es, por qué y cómo usarlo. Accedido 20/11/2022. [En línea]. Disponible en: <https://rockcontent.com/es/blog/bootstrap/>



- [57] Bootstrap team. Download. Accedido 20/11/2022. [En línea]. Disponible en: <https://getbootstrap.com/docs/4.4/getting-started/download/>
- [58] C. P. Martinez. (2021) Arquitectura de software y sus beneficios. Accedido 20/11/2022. Devexperto. [En línea]. Disponible en: <https://devexperto.com/arquitectura-del-software/>
- [59] J. R. M. Plascencia. (2022, Septiembre) ¿Qué es FrontEnd? Accedido 14/11/2022. [En línea]. Disponible en: <https://github.com/romarpla/cursoFrontEnd/blob/main/01-INTRO/temario/1.-queEsFrontEnd.md>
- [60] M. Angulo. (2015, Marzo) Patrones de diseño responsivo. Accedido 20/04/2023. Tesseract Space. [En línea]. Disponible en: <https://es.slideshare.net/RevistaSG/patrones-responsivos-mauricio-angulo>
- [61] M. Berg. (2020, Noviembre) Software Development Life Cycle (SDLC). Accedido 20/11/2022. Techopedia. [En línea]. Disponible en: <https://www.techopedia.com/definition/22193/software-development-life-cycle-sdlc>
- [62] M. Martin. (2022, Octubre) Software Development Life Cycle (SDLC) Phases & Models. Accedido 20/11/2022. Guru99. [En línea]. Disponible en: <https://www.guru99.com/software-development-life-cycle-tutorial.html>
- [63] D. Swersky. (2022, Junio) The SDLC: phases, popular models, benefits & more. Accedido 20/11/2022. RAYGUN. [En línea]. Disponible en: <https://raygun.com/blog/software-development-life-cycle/>
- [64] T. C. I. J. S. . Software & systems engineering *ISO/IEC/IEEE 29148:2018 Systems and software engineering - Life cycle processes - Requirements engineering* Second Edition ISO Central Secretariat, Geneva, Switzerland 2018 [En línea]. [En línea]. Disponible en: <https://www.iso.org/standard/72089.html>
- [65] I. Visure Solutions. Qué son los requisitos no funcionales: ejemplos, definición, guía completa. Accedido 21/04/2023. [En línea]. Disponible en: <https://visuresolutions.com/es/blog/non-functional-requirements>
- [66] PEP 3333 - Python Web Server Gateway Interface v1.0.1. Accedido 06/03/2023. Python Enhancement Proposals. [En línea]. Disponible en: <https://peps.python.org/pep-3333/>
- [67] C. P. Neto. (2015) Introduction - WSGI. Accedido 08/01/2023. CodePoint. [En línea]. Disponible en: <http://wsgi.tutorial.codepoint.net/intro>
- [68] N. Alonso. (2019, Noviembre) ¿Qué es un WSGI? Accedido 08/01/2023. Medium. [En línea]. Disponible en: <https://medium.com/@nachoad/que-es-wsgi-be7359c6e001>

- [69] O. Djodji. (2022, Mayo) WSGI in Django. Accedido 08/01/2023. Medium. [En línea]. Disponible en: <https://osirusdjodji.medium.com/wsgi-in-django-f35842ec0285>
- [70] MDN contributors. (2022, Septiembre) Django Introduction. Accedido 08/01/2022. Mozilla Corporation. [En línea]. Disponible en: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction>
- [71] N. George. “Django overview” en *Build a Website with Django 3*. Hamilton NSW, Australia: GNW Independent Publishing, 2016 pp. 15–29.
- [72] A. Bulezyuk. *Django 3 For Beginners* Mayo 2021.
- [73] D. Barajas González & K. Fonseca Márquez “Introducción a UML” en *Programa de Formación en TIC* DGTIC, UNAM, Ed. 2019.
- [74] G. Lionetti. (2012, Febrero) What is version control: centralized vs. DVCS. Accedido 19/03/2023. Atlassian. [En línea]. Disponible en: <https://www.atlassian.com/blog/software-teams/version-control-centralized-dvcs>
- [75] A. Holovaty & J. Kaplan-Moss. Integración con una base de datos existente. Accedido 20/03/2023. Uniwebsidad. [En línea]. Disponible en: <https://uniwebsidad.com/libros/django-1-0/capitulo-16/integracion-con-una-base-de-datos-existente>
- [76] S. Pittet. The different types of software testing. Accedido 20/04/2023. Atlassian. [En línea]. Disponible en: <https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing>
- [77] MDN contributors. (2023, Febrero) Testing a Django web application. Accedido 21/04/2023. Mozilla Corporation’s. [En línea]. Disponible en: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Testing>
- [78] S. Gonzalez. (2020, Abril) ¿Qué es la experiencia de usuario? Accedido 28/03/2023. Cyberclick. [En línea]. Disponible en: <https://www.cyberclick.es/que-es/experiencia-de-usuario>