

**Desarrollo de una interfaz para la integración en la carga de información  
entre la herramienta TesGestion y el software Ellipse aplicando  
metodologías ágiles de desarrollo de software**

**DIEGO ANDRÉS VANEGAS CORTES**



Universidad Santo Tomas  
Facultad de Ingeniería Electrónica  
Bogotá, Colombia  
2017

**Desarrollo de una interfaz para la integración en la carga de información  
entre la herramienta TesGestion y el software Ellipse aplicando  
metodologías ágiles de desarrollo de software**

**DIEGO ANDRÉS VANEGAS CORTES**



Anteproyecto de grado presentado como requisito para optar el título de  
Ingeniero electrónico

Tutor

Ing. Marco Antonio Vega Torres

Empresa

Tes América Andina S.A.S

Universidad Santo Tomás  
Facultad de Ingeniería Electrónica  
Bogotá, Colombia  
2017

## **ADVERTENCIA**

La Universidad Santo Tomás no se hace responsable de los conceptos emitidos por el alumno de su trabajo de grado. Solo velará porque no se publique nada contrario al dogma ni a la moral católica y porque la tesis no contenga ataques personales y únicamente se vea en ella su anhelo de buscar la verdad y la justicia.

***Artículo 23 Resolución del 13 de julio de 1956***

## TABLA DE CONTENIDO

<b>1. PLANTEAMIENTO DEL PROBLEMA</b> .....	5
<b>2. ANTECEDENTES</b> .....	6
<b>3. JUSTIFICACION</b> .....	7
<b>4. VISION HUMANISTICA</b> .....	8
<b>5. OBJETIVOS</b> .....	9
5.1 Objetivo General.....	9
5.2 Objetivo Específico.....	9
<b>6. FACTIBILIDAD</b> .....	10
6.1 Recursos Humanos.....	10
6.2 Recursos Técnicos.....	10
<b>7. MARCO TEÓRICO</b> .....	11
7.1 Metodologías ágiles de desarrollo de software.....	11
7.1.1 Metodologías tradicionales.....	12
7.1.2 Metodologías ágiles.....	12
7.1.3 Comparación de metodologías.....	12
7.1.4 Scrum.....	13
7.1.4.1 Roles.....	14
7.1.4.2 Reuniones o eventos.....	14
7.1.4.3 Artefactos.....	15
7.2 Suite TesGestion.....	19
7.3 Ellipse.....	19
<b>8. DESARROLLO Y RESULTADOS</b> .....	20
8.1 Desarrollo.....	20
8.1.1 Primer Sprint.....	21
8.1.2 Segundo Sprint.....	22
8.2 Resultados de Sprints.....	23
8.2.1 Primer Sprint.....	24
8.2.2 Segundo Sprint.....	24
8.3 Resultados.....	25
8.3.1 Herramienta de exportación.....	26
8.3.2 Herramienta de importación.....	28
<b>9. CONCLUSIONES</b> .....	33
<b>10. GLOSARIO</b> .....	34
<b>11. BIBLIOGRAFÍA</b> .....	36

## 1. PLANTEAMIENTO DEL PROBLEMA

En la actualidad, en la industria de las telecomunicaciones móviles es indispensable contar con una integración constante entre simuladores de enlaces de microondas y herramientas de gestión de redes, debido a que facilita la documentación y consolidación de la información, sin embargo, es poco usual en Colombia encontrar en un solo instrumento las funciones mencionadas anteriormente, es por esto que el objetivo del presente proyecto, es desarrollar un instrumento capaz de lograr la integración entre uno de los software más usados por los ingenieros en gestión de redes, TesGestion y el simulador de enlaces microondas Ellipse, que hasta el momento es el más completo, ya que cumple con la mayoría de los requerimientos del diseño en enlaces.

Aunque la suite TesGestion tiene como objetivo proveer las funcionalidades necesarias para gestionar una red de telefonía móvil, es necesario realizar una integración con el programa Ellipse para contar mayores prestaciones tecnológicas en el diseño de enlaces microondas y planificación de frecuencias con soporte para estas redes punto a punto y punto a multipunto [1].

Uno de los problemas de este proyecto es que tiene requerimientos cambiantes y se debe desarrollar en el menor tiempo posible, para esto es necesario la utilización de metodologías ágiles para el desarrollo de software y por ello se utiliza el marco de trabajo Scrum, ya que este satisface las necesidades que el cliente pueda tener y se adapta a ellas.

Al unir la Suite TesGestion con el programa Ellipse, podremos generar más funcionalidades y así conseguir una mayor eficiencia y productividad dentro de las diferentes herramientas, además de generar un mayor impacto en la industria, aumentando la gama de posibilidades que puede prestar el producto.

## 2. ANTECEDENTES

Actualmente en la Suite TesGestion existe la integración con el software Pathloss y tiene como funcionalidad exportar un archivo de texto separado por comas (csv), con la información de los enlaces seleccionados para poder cargarlos en la herramienta y así poder simular las diferentes conexiones que sean creadas a partir de TesGestion [2].

Además, esta herramienta cuenta con la capacidad de importar archivos que fueron generados por el software Pathos, permitiendo así la creación de enlaces con las características suministradas durante el diseño en la herramienta de simulación, facilitando la creación de estos a partir de dicho simulador [2].

La herramienta fue diseñada y desarrollada utilizando como eje principal el marco de trabajo Scrum debido a su gran necesidad de implementar metodologías ágiles para el desarrollo de software, puesto que estaba compuesta con requerimientos cambiantes durante su desarrollo, además de entregas de incrementos del producto que fueran aprobadas por los clientes para llenar las necesidades que se planteaban [2].

### 3. JUSTIFICACIÓN

En la actualidad, en el mercado del desarrollo de software, realizar integraciones entre diferentes herramientas para lograr una compatibilidad entre estas es esencial para agilizar el trabajo de las personas, lo que lleva a la realización de una herramienta adicional en la Suite TesGestion que sea capaz de lograr una integración con el simulador de enlaces de microondas utilizado por los clientes en general, el cual recibe el nombre de Elipse.

El desarrollo de esta herramienta tiene como objetivo principal generar comodidad al usuario para la gestión de nuevos enlaces, la cual hoy en día causa gran molestia por la extensa información que se maneja, es por esto que se desarrollará una herramienta capaz de cargar gran cantidad de datos en muy poco tiempo y con la menor cantidad de recursos necesarios, útil en el mercado actual que busca velocidad y agilidad de procesos de documentación.

Conociendo la complejidad que se tiene en los diferentes desarrollos de software es necesario, implementar metodologías ágiles de desarrollo de software, garantizando un enfoque iterativo e incremental para mejorar la previsibilidad y control de riesgos, teniendo como eje principal el marco de trabajo Scrum.

#### **4. VISION HUMANÍSTICA**

En la era de las telecomunicaciones y la tecnología en la que vive el hombre actualmente, es una necesidad lograr la mayor eficiencia en las herramientas que usan para poder disponer de mejor tiempo en el trabajo y librar de estrés los días de trabajo excesivo, es por esto que la idea de integrar las herramientas siempre está presente en el desarrollo de software puesto que se facilita la obtención de datos y el cumplimiento de tareas, generando comodidad a los usuarios que usen constantemente diferentes software y que se ven vinculados en un momento dado.

En el área de desarrollo de software el programador se enfrenta continuamente a momentos donde la sobrecarga de trabajo es un problema continuo, es por esto que hacer uso de la metodologías ágiles de desarrollo se ven casi que obligatorias en este ámbito, ya que es una metodología pensada para que las personas puedan cumplir de manera efectiva el trabajo pero siempre teniendo en cuenta no generar demasiado estrés en los desarrolladores; Scrum busca generar siempre que el espacio de trabajo sea un momento agradable en casos de complejidad en los proyectos, por lo mencionado anteriormente es una de las metodologías más utilizadas y por la que durante este proyecto será la metodología utilizada .

## 5. OBJETIVOS

### Objetivo general

- Diseñar una herramienta en el software TesGestion capaz de importar y exportar enlaces y su información desde y hacia Ellipse.

### Objetivos específicos

- Realizar el desarrollo del proyecto con las prácticas del framework SCRUM.
- Realizar el diseño completo de una interfaz capaz de importar y exportar los documentos necesarios para la integración.
- Implementar software de código abierto con el fin de permitir modificaciones futuras.
- Realizar una validación de software cumpliendo con la norma ISO/IEC 17025

## **6. FACTIBILIDAD**

El proyecto se integra a la línea de investigación de la empresa Tes América andina S.A.S, en su herramienta TesGestion la cual cuenta con trabajo previo del grupo de I+D de la empresa, quienes desarrollan y mantienen continuamente las diferentes herramientas existentes

### **6.1 Recursos Humanos**

Ejecutor: Diego Andrés Vanegas Cortes

Código: 2126484.

Estudiante de Ingeniería Electrónica.

Universidad Santo Tomas.

### **6.2 Recursos Técnicos**

#### **Institucionales**

- Recursos bibliográficos de Tes América andina S.A.S.
- Recursos bibliográficos de la Universidad Santo Tomás.
- Bases de datos virtuales.
- Tutoría prestada por parte de los docentes del programa.

#### **Software**

- Visual Studio 2015.
- Ellipse

## **7. MARCO TEÓRICO**

Dado que el desarrollo de la herramienta tiene una complejidad alta es necesario utilizar metodologías ágiles de desarrollo de software, por lo cual se vio la necesidad de recibir una capacitación de parte de la empresa TES AMERICA para poder entender cómo funcionan, además de identificar que Scrum como metodología ágil, nos brinda los beneficios necesarios para agilizar el trabajo de desarrollo gastando el menor tiempo y logrando cumplir los requerimientos de los clientes, durante este marco teórico se describirán los diferentes puntos a tratar durante la capacitación, además de una explicación de lo que es Scrum y que lo conforma teniendo en cuenta los roles, reuniones y artefactos, agregando un resumen final en base a la Fig.2.

Para la ejecución del proyecto es esencial conocer el funcionamiento de las dos herramientas a integrar, es por eso que al final de este marco teórico, se describe cada una de estas recalando las funcionalidades principales que brindan al usuario.

### **7.1 Metodologías ágiles de desarrollo de software**

En los años noventa surgieron metodologías de desarrollo de software ligeras –luego llamadas ágiles– dirigidas a reducir la probabilidad de fracaso por subestimación de costos, tiempos y funcionalidades en los proyectos de desarrollo de software. Se gestaron como alternativa a las metodologías tradicionales, específicamente para reducir la carga burocrática propia de ellas, en proyectos de pequeña y mediana escala. A diferencia de las tradicionales, las metodologías ágiles son adaptativas –no predictivas–, y están orientadas a las personas –no a los procesos– [3].

Hablar de metodologías ágiles implica hacer referencia a las metodologías de desarrollo de software tradicionales ya que las primeras surgieron como una reacción a las segundas; sus características principales son antagónicas y su uso ideal se aplica en contextos diferentes.

### **7.1.1 Metodologías tradicionales**

Las metodologías tradicionales de desarrollo de software son orientadas por planeación. Inician el desarrollo de un proyecto con un riguroso proceso de levantamiento de requerimientos previos a etapas de análisis y diseño. Con esto tratan de asegurar resultados con alta calidad circunscritos a un calendario.

En las metodologías tradicionales se concibe un solo proyecto, de grandes dimensiones y estructura definida. Se sigue un proceso secuencial en una sola dirección y sin marcha atrás el proceso es rígido y no cambia, los requerimientos son acordados una vez y para todo el proyecto, demandando grandes plazos de planeación previa y poca comunicación con el cliente una vez ha terminado ésta [4].

### **7.1.2 Metodologías Ágiles**

Las metodologías ágiles son flexibles, pueden ser modificadas para que se ajusten a la realidad de cada equipo y proyecto. Los proyectos ágiles se subdividen en proyectos más pequeños mediante una lista ordenada de características.

Cada proyecto es tratado de manera independiente y desarrolla un subconjunto de características durante un periodo de tiempo corto, aproximado de dos a cuatro semanas. La comunicación con el cliente es constante al punto de requerir un representante de él durante el desarrollo. Los proyectos son altamente colaborativos y se adaptan mejor a los cambios; de hecho, el cambio en los requerimientos es una característica esperada y deseada, al igual que las entregas constantes al cliente y la retroalimentación por parte de él. Tanto el producto como el proceso son mejorados frecuentemente [4].

### **7.1.3 Comparación entre metodologías**

La tabla 1 muestra aspectos relevantes de las metodologías de desarrollo tradicional contrastándolas con los aspectos relevantes de las metodologías de desarrollo ágil [4].

Metodologías Tradicionales	Metodologías Ágiles
Predictivos	Adaptativos
Orientado a procesos	Orientado a personas
Proceso rígido	Procesos flexible
Se concibe como un proyecto	Un proyecto es subdividido en varios proyectos más pequeños
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el desarrollo	Entregas constantes de Software
Documentación extensa	Poca documentación

Tabla 1. Metodologías tradiciones vs metodologías Ágiles [4]

#### 7.1.4 Scrum

Su primera referencia en el contexto de desarrollo data de 1986, cuando Takeuchi y Nonaka utilizan el *Rugby Approach* para definir un nuevo enfoque en el desarrollo de productos dirigido a incrementar su flexibilidad y rapidez, a partir de la integración de un equipo interdisciplinario y múltiples fases que se traslapan entre sí [4].

La metodología Scrum para el desarrollo ágil de software es un marco de trabajo diseñado para lograr la colaboración eficaz de equipos en proyectos, que emplea un conjunto de reglas y artefactos y define roles que generan la estructura necesaria para su correcto funcionamiento. Scrum utiliza un enfoque incremental que tiene como fundamento la teoría de control empírico de procesos. Esta teoría se fundamenta en transparencia, inspección y adaptación; la transparencia: Garantiza la visibilidad en el proceso de las cosas que pueden afectar el resultado; la inspección: ayuda a detectar variaciones indeseables en el proceso; y la adaptación: realiza los ajustes pertinentes para minimizar el impacto de las mismas.

#### 7.1.4.1 Roles

Scrum define tres roles: el **Scrum master**, **Product Owner** y **el equipo de desarrollo** [5]. El Scrum master tiene como función asegurar que el equipo está adoptando la metodología, sus prácticas, valores y normas; es el líder del equipo pero no gestiona el desarrollo [4]. El Product Owner, es una sola persona y representa a los interesados, es el responsable de maximizar el valor del producto y el trabajo del equipo de desarrollo; tiene entre sus funciones gestionar la lista ordenada de funcionalidades requeridas o Product Backlog. El equipo de desarrollo, por su parte, tiene como responsabilidad convertir lo que el cliente quiere, el Product Backlog, en iteraciones funcionales del producto; el equipo de desarrollo no tiene jerarquías, todos sus miembros tienen el mismo nivel y a cargo el desarrollador. El tamaño óptimo del equipo está entre tres y nueve personas.

#### 7.1.4.2 Reuniones o Eventos

Scrum define un evento principal o Sprint (Fig.1) que corresponde a una ventana de tiempo donde se crea una versión utilizable del producto (incremento). Cada Sprint, como en el rugby, es considerado como un proyecto independiente. Su duración máxima es de un mes. Un Sprint se compone de los siguientes elementos: reunión de planeación del Sprint, Daily Scrum, trabajo de desarrollo, revisión del Sprint y retrospectiva del Sprint.

En la reunión de **Planeación del Sprint** se define su plan de trabajo: qué se va a entregar y cómo se logrará. Es decir, el diseño del sistema y la estimación de la cantidad de trabajo.

**El Daily Scrum** es un evento del equipo de desarrollo de quince minutos, que se realiza cada día con el fin de explicar lo que se ha alcanzado desde la última reunión; lo que se hará antes de la siguiente; y los obstáculos que se han presentado. Este evento se desarrolla mediante una reunión que normalmente es sostenida de pie con los participantes reunidos formando un círculo, esto, para evitar que la discusión se extienda [4].

**La Revisión del Sprint** ocurre al final del Sprint y su duración es de cuatro horas para un proyecto de un mes (o una proporción de ese tiempo si la duración es menor). En esta etapa: el Product Owner revisa lo que se hizo, identifica lo que no se hizo y discute acerca del Product Backlog; el equipo de desarrollo cuenta los problemas que encontró y la manera en que fueron resueltos, y muestra el producto y su funcionamiento. Esta reunión es de gran importancia para los siguientes Sprints.

**La Retrospectiva del Sprint** es una reunión de tres horas del equipo Scrum en la que se analiza cómo fue la comunicación, el proceso y las herramientas; qué estuvo bien, qué no, y se crea un plan de mejoras para el siguiente Sprint. El tiempo, tal como en los casos anteriores, se debe ajustar proporcionalmente en el caso de proyectos de duración menor a un mes.

#### **7.1.4.3 Artefactos**

Existen también Artefactos de Scrum. Estos son subproductos de las actividades del marco de trabajo que le brindan dirección y transparencia al equipo [5]. Los artefactos de Scrum son: Product Backlog, Sprint Backlog, Monitoreo de Progreso, Incremento y Burndown Chart [5].

**El Product Backlog** es una lista –ordenada por valor, riesgo, prioridad y necesidad– de los requerimientos que el Product Owner define, actualiza y ordena. La lista tiene como característica particular que nunca está terminada, pues evoluciona durante el desarrollo del proyecto.

**El Sprint Backlog** es un subconjunto de ítems del Product Backlog y el plan para realizar en el Incremento del producto. Debido a que el Product backlog está organizado por prioridad, el Sprint backlog es construido con los requerimientos más prioritarios del Product backlog y con aquellos que quedaron por resolver en el Sprint anterior. Una vez construido, el Sprint backlog debe ser aceptado por el equipo de desarrollo, pertenece a éste y solo puede ser modificado por él. Requerimientos adicionales deben ser incluidos en el Product backlog y desarrollados en el siguiente Sprint, si su prioridad así lo indica.

**El Monitoreo de Progreso** consiste en la suma del trabajo que falta por realizar en el Sprint. Tiene como característica que se puede dar en cualquier momento, lo que le permite al Product Owner evaluar el progreso del desarrollo. Para que esto sea posible, los integrantes del equipo actualizan constantemente el estado de los requerimientos que tienen asignados indicando cuánto consideran que les falta por terminar.

**El Incremento** es la suma de todos los ítems terminados en el Sprint backlog. Si hay ítems incompletos deben ser devueltos al Product backlog con una prioridad alta para que sean incluidos en el siguiente Sprint. Se considera que un ítem está terminado si es funcional. La suma de ítems terminados es el producto a entregar.

**El Burndown chart** sirve para saber el tiempo que falta para completar el trabajo. Normalmente se utiliza para saber cuánto falta para terminar las historias comprometidas en un sprint. En la práctica es un diagrama de dos ejes, en el eje X el tiempo en días de duración del sprint en el eje Y la cantidad de trabajo comprometida con el cliente durante el sprint, en las unidades de historias de usuario.

**Historias de usuario:** Representan de manera rápida los requerimientos finales de los clientes, donde se puede evidenciar las horas de trabajo estimadas de cada una y la importancia que conlleva cada historia, además de una breve descripción del problema que va acompañado en la parte trasera de información adicional a tener en cuenta para el cumplimiento de esta, en la Fig.1 se logra evidenciar un ejemplo de esta.

ID	Nombre de Historia	
Descripción general		
Horas	Importancia	
Información adicional sobre la historia		

Fig. 1 Historia de Usuario.

- **ID:** Se asigna una identificación de cada historia de usuario por lo general es un número que ayuda a tener control más fácil de estas.
- **Nombre de historia:** Nombre de historia con el que se describa muy brevemente lo que se quiere lograr.
- **Horas:** Son las horas o puntos de historia que se estiman para cada una, en donde se puede identificar que tanto trabajo conllevaría el cumplimiento de esto.
- **Importancia:** Es un numero donde indica que tanta importancia tiene cada historia entre más urgencia más alto es este, se puede repetir entre historias si las dos no dependen una de la otra.
- **Descripción general:** Es una breve descripción de lo que se debe desarrollar sin muchas especificaciones
- **Información adicional sobre la historia:** Se aclaran de forma más detallada que se quiere lograr durante el cumplimiento de la historia

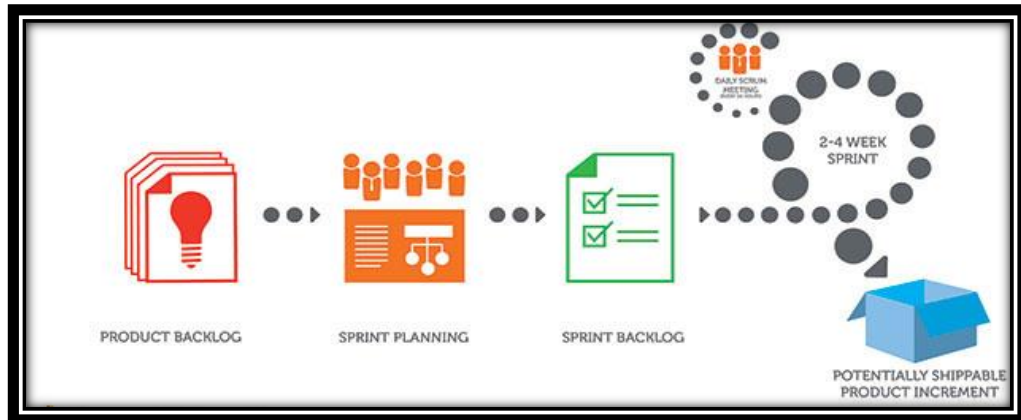


Fig. 2 Modelo de Scrum [5].

En la Fig.2 se puede observar un resumen grafico de la metodologia Scrum, donde se evidencia que para poder empezar el proceso iterativo de desarrollo, es necesario pasar por el Product backlog tomando todas las ideas de los clientes ademas de los requerimientos de este, ya definido esto es necesario organizarlo y darles prioridades a las diferentes historias de usuario que surgan durante el proceso, para ello se celebra la reunion de planeacion del sprint, en esta reunion es necesario la participacion del Scrum Master, Product Owner y el equipo de desarrollo, con las historias de usuario ya realizadas y desglosadas en tareas, se procede a construir un Sprint Backlog el cual sera la guia para trabajar y donde se se llevara la trazabilidad de trabajo que se esté realizando, dando como resultado un Burndown Chart donde se podran evaluar como trabajo todo el equipo. Una vez definido el Sprint Backlog el equipo de desarrollo entra en el proceso iterativo del Sprint planeado, con las Daily Scrum, en donde se hablara de lo que se realiza diariamente y de posibles problemas no muy especificados, como resultado final del sprint se entregaran incremento de producto los cuales representan un adeltanto funcional de desarrollo final el cual sera revisado durante la reunion de revision del sprint validando que el software entregado sea lo que los requerimientos solicitaban.

## **7.2 Suite TesGestion**

La Suite TesGestion es un aplicativo que provee la funcionalidad necesaria para gestionar una red de telefonía móvil celular [2].

La Suite TesGestion tiene dos formas de desplegar la información de la red del operador, una es la red en el estado actual y la otra es el estado planeado de la red. Para cada una de ellas, se tienen descritos de manera gráfica todos los sitios de dicha red con sus respectivos equipos torres y antenas. Además se tiene una relación de los enlaces y servicios que existen entre los diferentes puntos de la red [2].

La Suite TesGestion hace posible la administración y aprovisionamiento de los diferentes equipos existentes en cada sitio (Antenas, radios, BTSs, BSCs, MSCs, DDFs, transcoders, MTX, celdas, compresores, multiplexores), permitiendo la creación y modificación de estos, también permite establecer y configurar rutas sobre la red teniendo en cuenta la capacidad disponible de los enlaces [2].

## **7.3 Elipse**

Elipse (anteriormente MENTUM elipse) ofrece capacidades de ingeniería de microondas sin igual con un amplio soporte para todos los modelos necesarios para el análisis exhaustivo del rendimiento del enlace, incluyendo los modelos de propagación, modelos de intensidad de lluvia, modelos de trayectorias múltiples, modelos de diversidad, MIMO y modelos de rendimiento y disponibilidad [1].

Además, Elipse ofrece una rica interfaz de usuario geográfica con los formatos estándares conocidos, mapa, soporte multi-resolución y potentes opciones de visualización de capa, como la visualización en 3D de áreas urbanas densas. En línea, libre de cargo de datos cartográficos digitales (SRTM) también se pueden utilizar para reducir el tiempo de configuración del proyecto.

## 8. DESARROLLO Y RESULTADOS

### 8.1 Desarrollo

Para lograr el desarrollo de este proyecto se tuvo la necesidad de hablar con el cliente para poder obtener los requerimientos necesarios para crear una herramienta óptima y agradable al usuario, los requerimientos fueron:

1. Se debe crear una herramienta capaz de cargar un archivo de texto con la información de un enlace microondas.
2. Con la información obtenida en el archivo de texto se debe crear un enlace microondas en la Suite TesGestion.
3. Durante la creación el usuario puede verificar y agregar información si es necesario para la creación del enlace.
4. Se debe crear una herramienta capaz de exportar un archivo de Excel con la información necesaria para poderla simular en Ellipse.

Para cumplir con los requerimientos se estimó un tiempo de un mes, el cual se dividiría para en dos Sprints de dos semanas en los cuales se realizarían las dos herramientas solicitadas por el cliente; El resultado de cada Sprint se describe a continuación:

- Primer Sprint: Se diseñó la interfaz gráfica de tal forma que se adaptó a la Suite TesGestion de manera adecuada logrado que sea agradable para el cliente, además, se realizó el desarrollo para la carga de archivos provenientes del software Ellipse , el cual es un archivo de extensión .txt con la información necesaria para crear y evaluar enlaces.

- Segundo Sprint: Se realizó el desarrollo para la exportación de un documento de extensión .xlsx de hojas variables con toda la información necesaria para ser importada en el software Ellipse.

Cada Sprint se realizó teniendo en cuenta las historias de usuario y tareas que se vieron necesarias para completar el sprint de manera óptima, las horas de trabajo se estimaron usando puntos de historia, siendo estos la medida de tiempo generalmente utilizada en la metodología Scrum, donde un punto de historia representa una hora de trabajo por hombre sin distracciones.

### 8.1.1 Primer Sprint

En la Fig.3 se puede evidenciar la historia de usuario utilizada para el primer sprint representada por el ID "1", donde se puede identificar que tiene estimada 25 puntos de historia y una importancia de 100 debido a su gran urgencia en el desarrollo.

1	Cargar archivo de texto
El usuario debe ser capaz de cargar un archivo de texto con la información del enlace simulado en <u>Ellipse</u> .	
25	100
El usuario cargara un archivo de texto con información del enlace simulado, y al cargarlos este debe verificar que toda la información necesaria este cargada, en caso de faltar algún dato el usuario debe ser capaz de cargar la información faltante	

Fig. 3 Historia de usuario primer Sprint.

Esta historia de usuario se dividió en 3 tareas necesarias para cumplirla, estas tareas son:

1. Lograr ejecutar una interfaz adecuada para la carga del archivo de texto, generada luego de hacer clic en el botón asignado en la suite TesGestion **(8 puntos)**.
2. Tomar toda la información del archivo de texto y escoger únicamente la información necesaria para la creación del enlace, además de lograr calcular la faltante con los datos que se tienen **(13 puntos)**.
3. Con la información cargada lograr crear el enlace microondas haciendo pruebas con los archivos suministrados por el cliente, validando los datos **(4 puntos)**.

### **8.1.2 Segundo Sprint**

En la Fig.4 se puede evidenciar la historia de usuario utilizada para el primer sprint representada por el ID "2", donde se puede identificar que tiene estimada 25 puntos de historia y una importancia de 100 igualando la urgencia de desarrollo a la del primer sprint.

2	Exportar Archivo Excel
El usuario podrá generar un archivo de Excel de un enlace seleccionado en la Suite <u>TesGestion</u>	
25	100
El usuario generara un archivo de Excel con toda la información necesaria para ser cargado en el simulador <u>Ellipse</u>	

Fig. 4 Historia de usuario segundo Sprint.

Esta historia de usuario se dividió en 3 tareas necesarias para cumplirla estas tareas son:

1. Lograr ejecutar una interfaz en la cual se seleccionará el enlace a exportar y tomar toda la información de este para asignársela en el documento de Excel final **(13 puntos)**.
2. Asignar toda la información del enlace a un documento de Excel de 31 hojas **(8 puntos)**.
3. Realizar pruebas exportando documentos de Excel de varios enlaces microondas de TesGestion **(4 puntos)**.

## 8.2 Resultados de los Sprints

El resultado del proyecto se evalúa al final con un análisis al Burndown Chart de cada uno de los Sprint, por lo cual estos fueron tabulados y graficados durante el desarrollo.

### 8.2.1 Primer Sprint

Durante el primer sprint se realizaron todas las tareas estimadas en la planeación del sprint teniendo como resultado el siguiente Burndown Chart, donde durante el paso de los días se asignaban los puntos de trabajos que se ocuparon para el desarrollo del primer sprint, tratando de lograr los puntos estimados en las tareas designadas a partir de la historia de usuario:

Tarea	Estimado	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Total Horas
Tarea 1	8	2	2	0	0	2	0	0	2	0	0	8
Tarea 2	13	0	2	1	2	1	2	1	2	1	1	13
Tarea 3	4	0	0	0	0	0	0	2	0	0	2	4
Total	25	2	4	1	2	3	2	3	4	1	3	25

Tabla 2. Trabajo realizado en el primer sprint.

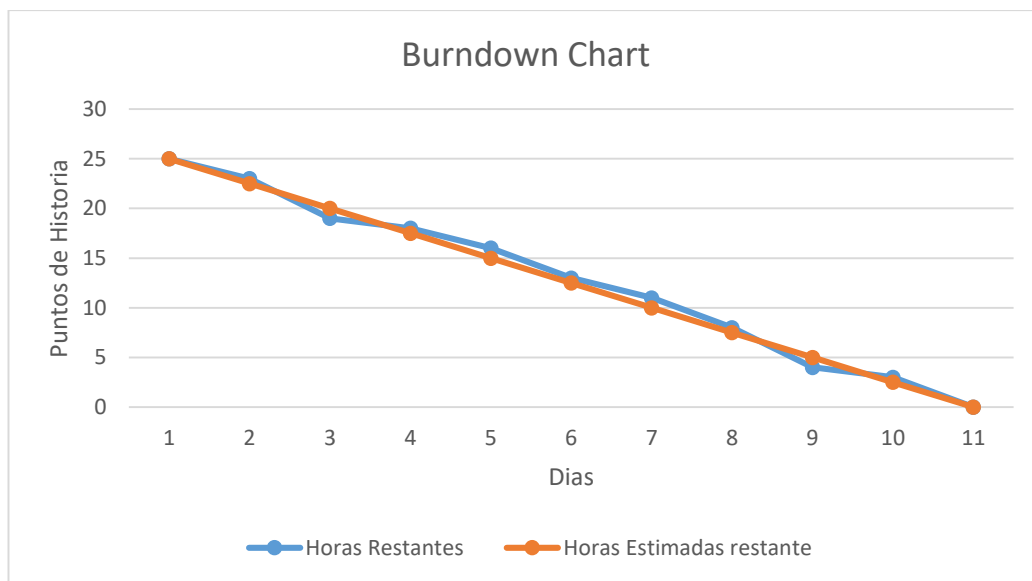


Fig. 5 Grafica del Burndown Chart del primer Sprint.

Con el grafico que se evidencia en la Fig. 5, se puede deducir que el desarrollo del sprint se realizó muy parecido a lo estimado, con un poco de retraso durante los días cinco y siete, pero con un trabajo avanzado en los días tres y nueve,

logrando un cumplimiento efectivo de la meta que se buscó durante esta iteración.

### 8.2.2 Segundo sprint

Durante el segundo sprint se realizaron todas las tareas estimadas en la planeación del sprint teniendo como resultado el siguiente Burndown Chart, donde durante el paso de los días se asignaban los puntos de trabajos que se ocuparon para el desarrollo del primer sprint, tratando de lograr los puntos estimados en las tareas designadas a partir de la historia de usuario:

Tarea	Estimado	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Total Horas
Tarea 1	13	2	2	1	2	1	2	1	2	0	0	13
Tarea 2	8	0	2	1	1	1	1	1	0	1	0	8
Tarea 3	4	0	0	0	0	0	0	2	0	0	2	4
Total	25	2	4	2	3	2	3	4	2	1	2	25

Tabla 3. Propuesta de trabajo para el segundo Sprint.

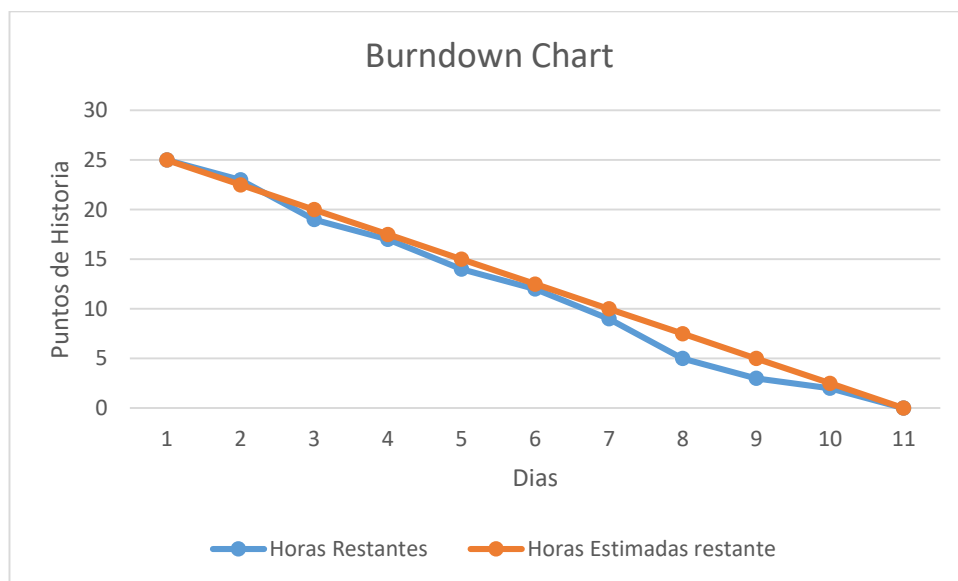


Fig. 6 Grafica del Burndown Chart del segundo Sprint.

Con el grafico que se evidencia en la Fig. 6, se puede deducir que el desarrollo del sprint se realizó muy parecido a lo estimado, donde nunca se evidencia retraso alguno y de lo contrario se evidencia un adelanto de trabajo de gran importancia en los días siete, ocho y nueve del sprint , logrando un cumplimiento efectivo de la meta que se buscó durante esta iteración.

### 8.3 Resultados

En el menú general de la Suite TesGestion [Fig.7], se puede ver el mapa de Colombia con los enlaces que han sido gestionados por los clientes, donde cada uno indica frecuencias, bandas, tecnologías y demás información necesaria para que el ministerio de las tecnologías de la información (MINTIC) sepa sobre la existencia de éstas, además del mapa se logra ver que tienen varias herramientas con las que se puede trabajar esta Suite, sin embargo se vio la necesidad de agregar dos iconos nuevos para poder acceder a los desarrollados realizados para la integración, estos se añadieron en el método de vista de iconos correspondientes a herramientas los iconos de la Fig.8.

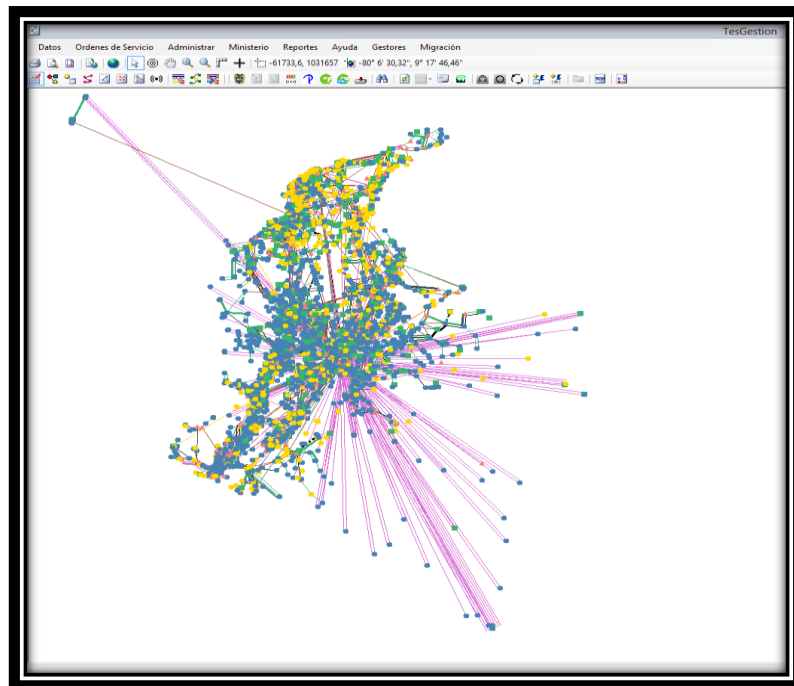


Fig.7 Ventana Principal Suite TesGestion.

Los iconos de las herramientas se agregaron a los recursos del proyecto, además de una breve descripción de cada uno que se desplegara pasando el puntero del mouse por encima.

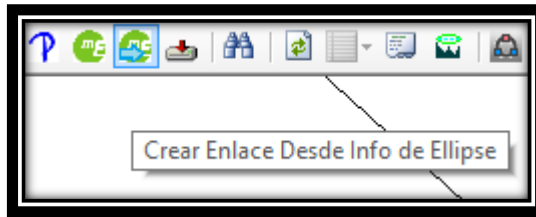


Fig.8 Botones des las herramientas de integración con Ellipse.

### 8.3.1 Herramienta de exportación

Para la herramienta de exportación se realiza un llamado a la herramienta de selección de enlaces microondas en la cual facilita la elección del enlace que se quiera simular en Ellipse.

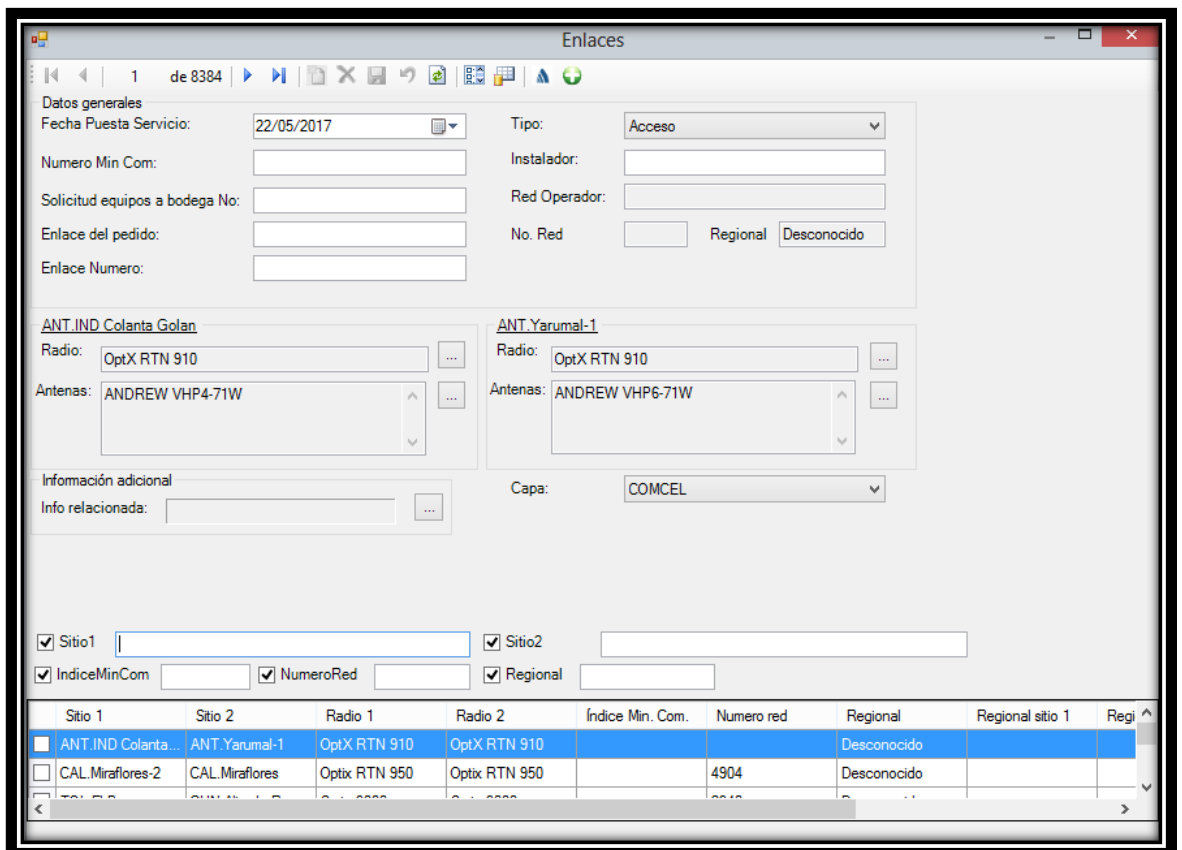


Fig.9 Ventana de selección de enlaces microondas.

En la Fig.9 se puede ver como la ventana da una información básica de cada enlace que se selecciona del listado de abajo, pudiendo identificar si el enlace seleccionado es él quiere exportar. Una vez seleccionado el enlace, se procede

a definir el lugar donde se quiera guardar el archivo de tal manera que sea accesible para el usuario [Fig.10], es necesario recalcar que el archivo solo se podrá guardar en formato .xlsx para que el simulador pueda acceder a él sin problemas.

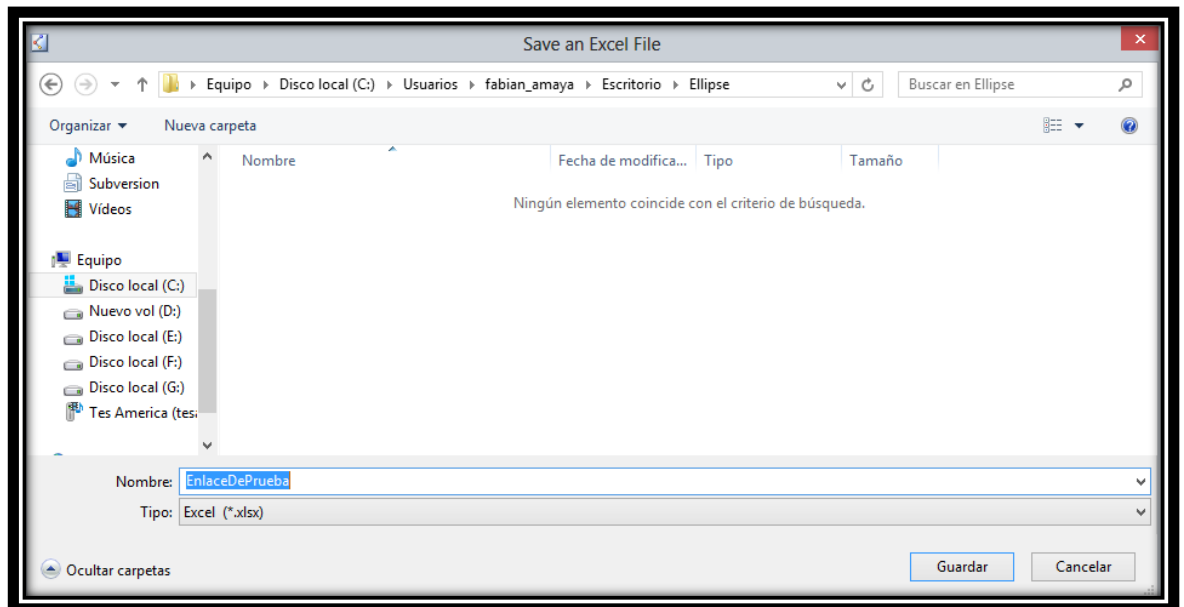


Fig.10 Selección de destino del archivo a exportar.

Por último la herramienta informara al usuario por medio de una caja de mensaje que tu archivo se creó correctamente y ya podrá ser usado para cargarse en Ellipse [Fig.11].

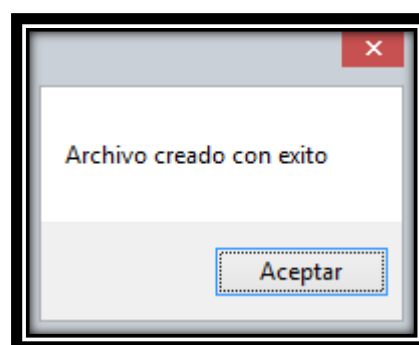


Fig.11 Caja de mensaje avisando la creación del archivo.

### 8.3.2 Herramienta de importación

Para cargar el archivo generado por el simulador Ellipse en la Suite TesGestion es necesario la carga de una ventana capaz de informarle al usuario que

información está cargando y desde que destino se carga para tener revisión constante de la información simulada, para ello se carga la ventana de la Fig.12 la cual facilita al usuario seleccionar el archivo correspondiente y generar alertas de falta de información.

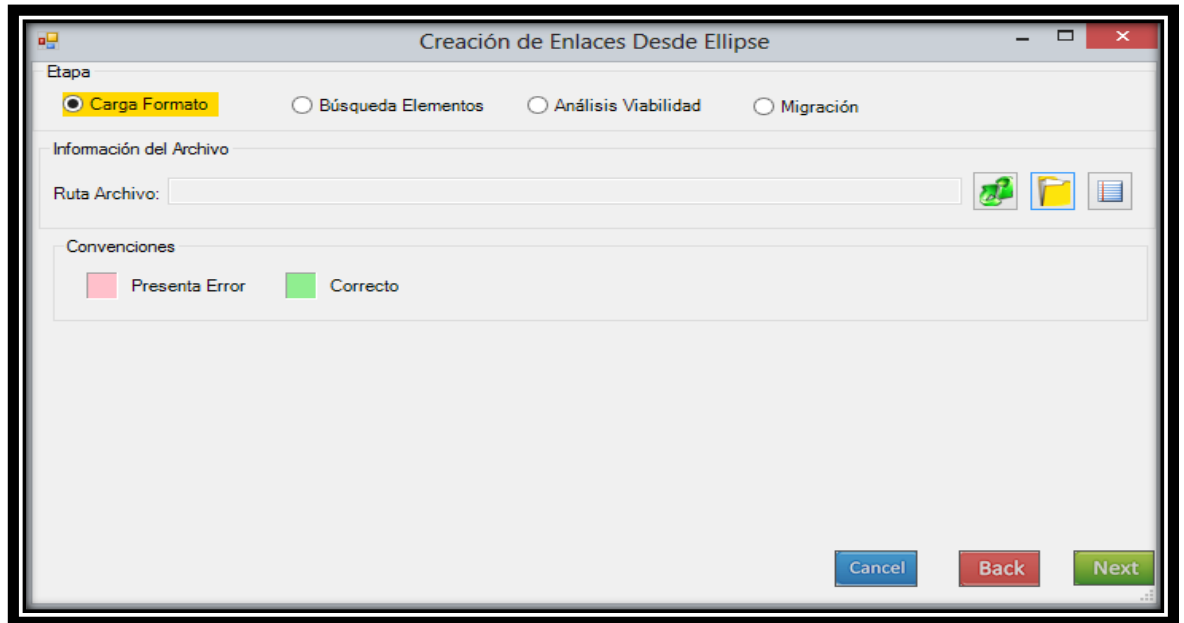


Fig.12 Ventana general para la carga del archivo de texto.

Haciendo clic en el botón con el icono de carpeta se abrirá una ventana de Windows donde se podrá navegar desde los documentos existentes en tu ordenador para poder seleccionar el archivo de texto que se quiere cargar para la creación del enlace microondas, la información que este suministre será revisada y administrada por código para que se verifique.

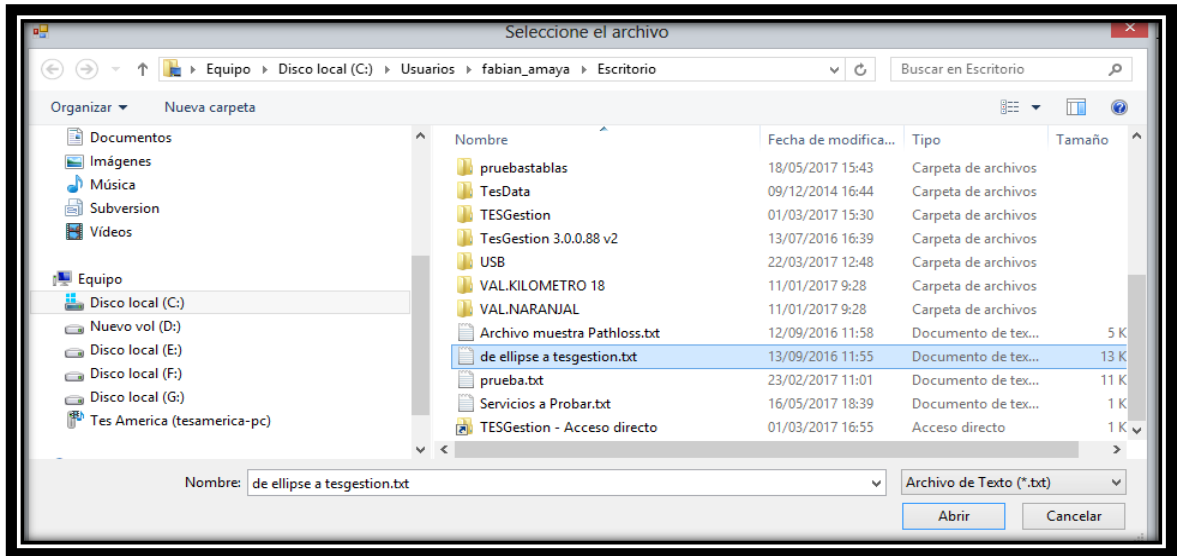


Fig.13 Ventana de Windows para la selección de archivo de texto.

Con el archivo ya seleccionado la ventana de la Fig.13 se modificará indicando con colores en si toda la información se cargó correctamente o si por el contrario falta información en el archivo, en caso de estar correcto se indicaría con color verde todas las casillas en caso opuesto el color se remplazaría por rojo como se ve en la Fig.14.

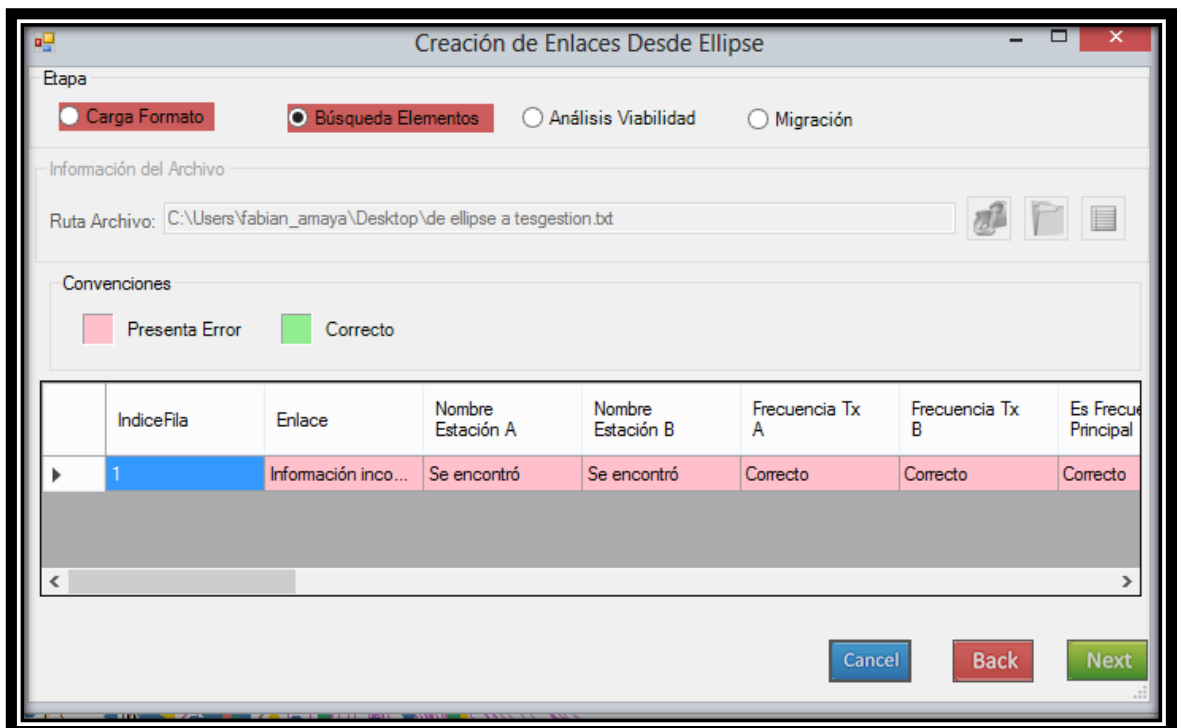


Fig.14 Ventana de comprobación de información cargada.

Ya con la información cargada del archivo se continua con el procedimiento de creación de enlace microondas común, sin embargo tanto en la selección de radios como de antenas se puede apreciar [Fig.15 - Fig.16] como varios datos de ya están llenados y solo se necesita revisión de éstos.

Fig.15 Selección de radio.

En caso de que la información de radio o antena no esté completa el usuario podrá rellenar la información faltante como se haría en un caso común de creación de enlaces microondas, teniendo en cuenta claro que toda la información sea congruente con lo que se simulo en Ellipse.

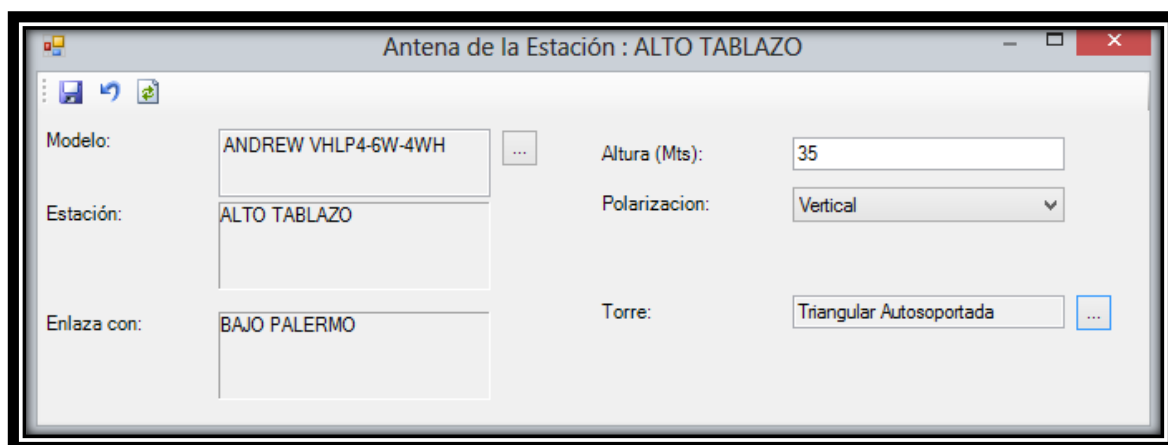


Fig.16 Selección de Antena.

Al final el enlace se creará entre las estaciones seleccionadas y se podrá evidenciar en el mapa general que ofrece TesGestion.

## 9. CONCLUSIONES

- Se logró acelerar el tiempo de ejecución en la creación de enlaces microondas usando la Suite TesGestion luego de que estos fueron simulados en el software Ellipse en un 45%, debido a que en promedio una creación de enlace toma entre unos trece a quince minutos, con la utilización de la herramienta se reduce a un tiempo de ocho minutos aproximadamente, ya que se agiliza el proceso cambiando de manual a semiautomático, como se vio en los resultados solo se necesita una carga de un archivo en formato .txt y unas verificaciones leves para lograr documentar lo requerido.
- El tiempo de lograr simular enlaces microondas en la herramienta Ellipse, que ya están documentados en TesGestion, se vio mejorada reduciéndose en un 75% debido a que en promedio una simulación de este enlace toma unos veinticinco minutos manualmente, ya que es necesario obtener la información de cada enlace y rellenar formularios de gran extensión, el tiempo que toma simular un enlace con el archivo exportado toma tan solo siete minutos aproximadamente, esto gracias a que solo se necesita elegir los enlaces a simular y la Suite generara un archivo de formato .xlsx que podrá ser cargado en la herramienta logrando acelerar un proceso.
- Se logró determinar la fortaleza que tiene Scrum frente a proyectos de gran complejidad , ya que como se observa en los resultados de los Burndown Charts la entrega del producto se realizó con los tiempos estimados debido a una buena planeación durante las reuniones que hacen parte de esta metodología ágil de desarrollo de software.
- Al utilizar patrones de diseño mencionados en el libro de Gamma [9], se logró realizar un código abierto con fácil mantenimiento y de gran calidad para futuras mejoras.

- Debido a la metodología Scrum, el objetivo de realizar una validación de software cumpliendo con la norma ISO/IEC 17025 no se pudo cumplir ya que durante esta, el producto final se valida realizando una reunión donde se revisa cada uno de los desarrollos cumplidos y que estos cumplan a cabalidad con los requerimientos del cliente.
- Aunque Scrum es una metodología útil para la realización de proyectos complejos, no es perfecta pues esta tiene problemas en el enfoque de la documentación y mantenimiento del producto. Esto se debe al concepto emitido por los expertos en agilidad, quienes indican que la documentación debe ser breve, precisa y limitada a las funciones fundamentales del software, porque tener el producto funcionando, es lo más importante.

## 10. GLOSARIO

- **Scrum:** Metodología ágil de desarrollo de software
- **Sprint:** Proceso iterativo que forma parte de la metodología Scrum
- **Burndown Chart:** Grafico con el cual que analiza el desarrollo respecto al tiempo y puntos de usuario
- **Puntos de usuario:** Puntos que hacen referencia a la estimación de trabajo que se tendrá que realizar para una tarea o historia, que hacen equivalencia a horas-hombre.
- **Sprint Backlog:** Es un tablero donde se identifican todos los componentes que se evaluaron para el sprint y se revisa la trazabilidad de trabajo que se lleva durante el proceso
- **Product Backlog :** lista de objetivos y requisitos priorizada donde se entiende lo que quiere el cliente
- **Pathloss:** Software de simulación de enlaces
- **Ellipse:** Software de simulación de enlaces
- **Product Owner:** Dueño del producto o principal contacto entre el cliente y el equipo de desarrollo
- **Scrum Master:** Líder del equipo de Scrum quien se encarga de que siempre se cumplan las metas
- **TesGestion:** Herramienta de gestión y planeación de diferentes enlaces.
- **Daily Scrum:** Reunión diaria donde se habla de lo que se hizo , se va hacer y que inconvenientes se tienen sin ser muy detallados en esto ultimo

## 11. BIBLIOGRAFÍA

- [1]. InfoVista. Artículo "Ellipse Proven enterprise-grade microwave link planning and transport network optimization solution". Courtaboeuf Cedex Les Ulis, Francia. Disponible desde <http://www.infovista.com/products/mobile-backhaul-network-planning-network-optimization> (2016)
- [2]. Tes América Andina S.A.S. Manual "Suite TesGestion". Tes América Andina S.A.S, Bogotá, Colombia. (2014).
- [3]. Patricio Letelier Torres Emilio A. Sánchez López. Artículo "Metodologías Ágiles en el Desarrollo de Software". Grupo ISSI, Alicante, España. Disponibles desde <http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf> (2003)
- [4]. Andrés Navarro Cadavid, Juan Daniel Fernández Martínez, Jonathan Morales Vélez. Artículo "Revisión de metodologías ágiles para el desarrollo de software". Universidad Icesi, Cali, Colombia (2013).
- [5]. Schwaber, K. & Sutherland, J. Libro "The Scrum guide" [Internet], Disponible desde <http://www.scrumguides.org/> (2011).
- [6]. Henrik Kniberg. Libro "Scrum y XP desde las trincheras" [Internet] disponibles desde <http://www.proyectalis.com/wp-content/uploads/2008/02/scrum-y-xp-desde-las-trincheras.pdf> (2007)
- [7]. Sirin Thongsukh, Smitti Darakorn Na Ayuthaya, and Supaporn kiattisin. Artículo "Startup Framework based On Scrum Framework", 2017 International Conference on Digital Arts, Media and Technology (ICDAMT) (2017).
- [8]. Shruti Sharma, Nitasha Hasteer . Artículo "A comprehensive study on state of Scrum development". 2016 International Conference on Computing, Communication and Automation (ICCCA) (2016).
- [9]. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Libro "Patrones de diseño" Addison Wesley, (2003).
- [10]. Eucario Parra Castrillón, Revista "Propuesta de metodología de desarrollo de software para objetos virtuales de aprendizaje -MESOVA-", Revista virtual Universidad católica del Norte" No.34, Colombia (2011)