VISIÓN ELECTRÓNICA

Preparación de Artículos revista VISIÓN ELECTRÓNICA: algo más que un estado sólido
Fecha de envío:
Fecha de recepción: 12 de Agosto de 2016
Fecha de aceptación:

Implementación de un Estado de radicación y generación de un código de certificados de espectro para autenticación única en la plataforma SGE (Sistema de Gestión del Espectro Radioeléctrico) del MINTIC aplicando metodologías ágiles de desarrollo y criterios de TES América Andina

Implementation of a State of establishment and code generation spectrum certificates for unique authentication on the SGE platform (System Spectrum Management) of MINTIC applying agile development methodologies and TES America Andina criteria

Marlon Patiño Bernal[1]
Fabián Rodrigo Amaya Colorado[2]

Resumen

En este trabajo se muestra como se implementó la generación de un código único para los certificados de espectro en la plataforma Sistema de Gestión del Espectro Radioeléctrico SGE del Ministerio de Tecnologías de la información y las telecomunicaciones MINTIC de Colombia. La herramienta facilita que se pueda autenticar únicamente con ese código y permite la implementación de un nuevo estado de radicación para solicitudes referentes al espectro electromagnético.

Para realizar este proyecto se requirió capacitación por parte de la empresa TesAmerica Andina Ltda. Se presentan los temas de formación que requirió el proyecto y los pasos requeridos para el desarrollo de la herramienta entre los que se incluyen metodologías ágiles, continuando con programación orientada a objetos, la planeación del proyecto y los pasos para el desarrollo. Todo el proyecto fue orientado por los ingenieros líderes y expertos del área de Investigación y Desarrollo (I+D) de la empresa Tes América Andina, la cual es catalogada por Colciencias como una empresa altamente innovadora

La financiación del proyecto fue asumido por la empresa Tes América Andina con un costo aproximado de $4.060.000 y el tiempo de duración del proyecto fue de 5 meses y la cantidad de horas trabajadas por la persona de 480 horas.

[1] Ingeniero Electrónico- Universidad Distrital FJC, Magister telecomunicaciones Universidad Nacional. E-mail: marlonpb@udistrital.edu.co
[2] Aspirante al título de Ingeniero en Telecomunicaciones de la Universidad Distrital Francisco José de Caldas. E-mail: amayafabian08@hotmail.com

Palabras clave

Ministerio de Tecnologías y las Comunicaciones (MinTIC), Espectro Radioeléctrico, Metodologías ágiles, programación orientada a objetos (POO), SCRUM, autenticación y Estado.

Abstract

In this paper shows how was implemented the generation of a unique code for certificates of spectrum in the platform System of Spectrum management SGE from Ministry of Information Technology and Telecommunications MINTIC of Colombia. The tool helps that can authenticate only with that code and allows the implementation of a new status of filing for applications concerning the electromagnetic spectrum.

To realize this project was required training by the Company TesAmerica Andina Ltda. Topics of training required the project and the steps required for Development It presents training topics that required the project and the steps required for the development of the tool including agile methodologies, continuing with oriented programming Objects, the planning of the and Steps for development. The whole project was guided by leaders and experts engineers of Research and Development (R & D) of the Company Tes America Andina, which is listed by Colciencias as a as a highly innovative Company.

Funding for the project was taken over by the company Tes America Andina with an approximate cost of $ 4.060.000 and the project duration was 5 months and the number of hours worked per person of 480 hours.

Key Words

Ministry of Technology and Communications (MinTIC), Radio Spectrum Agile Methodologies, object-oriented programming (OOP), SCRUM, Authentication and State.

1. Introducción

TES América Andina SAS es una compañía que provee servicios especializados de ingeniería, consultoría y soluciones personalizadas a distintas entidades en la rama de las telecomunicaciones, entre ellas algunos operadores de telefonía móvil, su factor más importante es la Gestión del Espectro Radioeléctrico la cual es la combinación de los procedimientos administrativos y técnicos necesarios para garantizar la utilización eficaz del espectro de frecuencias radioeléctricas por todos los servicios de radiocomunicaciones definidos en el Reglamento de Radiocomunicaciones de la UIT, y la explotación de los sistemas radioeléctricos sin causar interferencia perjudicial. Para ello, el Ministerio TIC ha puesto a disposición de los interesados una herramienta web "SGE" que permite tramitar las solicitudes correspondientes a asignación, renovación, modificación y cancelación de sus respectivos permisos otorgados para el uso del espectro radioeléctrico, mediante la Ley 671 de 2001, el cual en materia de administración del espectro radioeléctrico establece que "La Asignación y utilización de recursos escasos Todo procedimiento para la asignación y utilización de recursos como las frecuencias los números y los derechos de paso se llevarán a la práctica de manera objetiva, oportuna, transparente no discriminatoria. Se pondrá a disposición del público el estado actual de las bandas de frecuencia asignadas, pero no es preciso identificar detalladamente las frecuencias asignadas a usos oficiales específicos. Para esto se debe registrar que la información que se

vaya a tramitar debe ser autentica por el MinTic con respecto a los usuarios registrados y que puedan acceder a esta información. Para esto se necesita saber que la información que se genere sea única y sea propia del usuario, también se debe generar un estado de radicación que quede la solicitud en un estado de aceptable para poder radicar, si no se puede radicar se va a un estado documental lo importante es la fecha de radicación, se debe tener en cuenta la fecha en la que se radico, dependiendo del estado del expediente. Para esto se realizara con ayuda de las metodologías ágiles y conocimientos de programación orientada a objetos y realizar la respectiva planeación del proyecto y llevarlo a cabo satisfactoriamente por lo tanto se desea superar cada uno de los objetivos propuestos gracias a una metodología.

Primero realizando la respectiva capacitación acerca de las metodologías ágiles y realizando un ejercicio aplicando todas los conceptos básicos de las metodologías ágiles para afianzar los términos y aplicarlos en el proyecto satisfactoriamente de igual forma se realiza la respectiva investigación y capacitación de los conceptos de programación orientada a objetos y realizar una presentación de cada uno de los temas de POO con lo cual se cumpliría el primer objetivo propuesto.

Ya teniendo los conceptos de las metodologías ágiles y la programación orientada a objetos se procede a realizar el desarrollo del proyecto con SCRUM se procede a realizarlo bajo esta metodología.

Luego de realizar la planeación del desarrollo se deben realizar las tareas correspondientes para la culminación del proyecto apropiadamente siguiendo con el protocolo de SCRUM.

Aprovechando el desarrollo que han tenido las metodologías ágiles y siendo un tema relativamente novedoso cuyos orígenes datan de mediados de la década de los 90 (Cadavid Martínez 2013) y teniendo en cuenta que es apto para un entorno con requerimientos inestables como lo son muchos proyectos de software, se opta por emplear SCRUM como framework para desarrollar el proyecto, con el ideal de mostrar algunos resultados importantes sobre la implementación de la metodología de manera que sirva como referencia para trabajos futuros.

La modalidad de grado es pasantía en la Empresa TesAmerica Andina Ltda., para la empresa se plantea nuevos retos para el crecimiento sostenido de la compañía, siendo uno de ellos la expansión regional, considerando la nueva dinámica que adquiere la administración del espectro radioeléctrico en las diferentes regiones del mundo.

2. OBJETIVO GENERAL

Implementar un Estado de radicación y generación de un código de certificados de espectro para autenticación única en la plataforma SGE (Sistema de Gestión del Espectro Radioeléctrico) del MINTIC aplicando metodologías de ágiles de desarrollo y criterios de TES América Andina LTDA

2.1 OBJETIVOS ESPECÍFICOS

- Realizar una capacitación acerca de metodologías ágiles de desarrollo y programación orientada a objetos.
- Presentar un informe de la capacitación realizada.
- Realizar el desarrollo del proyecto con las prácticas de las metodologías SCRUM.
- Generar un método para la autenticación de certificados de espectro radioeléctrico.

- Generar el nuevo proceso de radicación en un estado intermedio para los expedientes.

3. Investigación Propuesta de la Pasantía

3.1 Gestion del Espectro

De forma simplificada puede decirse que el espectro radioeléctrico hace referencia al medio por el cual se transmiten las frecuencias de ondas electromagnéticas que permiten las telecomunicaciones (radio, televisión, Internet, móviles, TDT, etc.), y que son administradas y reguladas por los gobiernos de cada país.

La definición del espectro radioeléctrico, tal y como la ha establecido la ITU, organismo especializado en telecomunicaciones de la ONU, encargado de regular las telecomunicaciones a nivel internacional entre las distintas administraciones y empresas operadoras, es: "Las frecuencias del espectro electromagnético usadas para los servicios de difusión y servicios móviles, de policía, bomberos, radioastronomía, meteoróloga y fijos" (Foster, 2007). Este concepto es dinámico, pues a medida que evolucionan las tecnologías cambian los rangos de frecuencia utilizados en comunicaciones en función del estado de estas.

En el caso particular de Colombia el organismo al cual le corresponde la gestión, administración, vigilancia y control del espectro radioeléctrico es la ANE y lo define como un recurso natural conformado por el conjunto de ondas electromagnéticas cuya frecuencia se fija convencionalmente por debajo de 3000 GHz, que se propagan por el espacio sin guía artificial. Es propiedad exclusiva del Estado y como tal constituye un bien de dominio público, inajenable e imprescriptible.

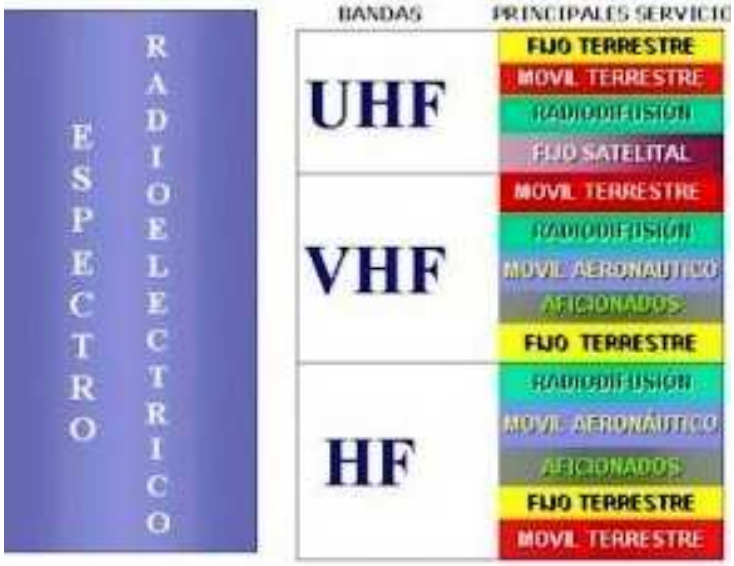


Figura 1. Bandas de frecuencias ERE
Fuente: www.ane.gov.co

La Figura 1 muestra la división del espectro radioeléctrico en bandas de frecuencia, estas bandas competen a cada servicio que las ondas electromagnéticas están en capacidad de prestar para las distintas compañías de telecomunicaciones avaladas y protegidas por las instituciones creadas para tal fin de los estados soberanos.

3.2 Ministerio de Tecnologías de la Información y las Comunicaciones

La primer institución estatal encargada de las comunicaciones a nivel nacional se creó en el año 1923 con el nombre de Ministerio de Correos y Telégrafos en el gobierno de Pedro Nel Ospina, hasta Julio del año 2009 cuando por medio de la Ley 1341 del mismo año se cambia su nombre a Ministerio de Tecnologías de la Información y las Comunicaciones, además de redefinir su estructura y sus funciones todo esto en el marco de la creación del sector TIC (Tovar F. Asociados Abogados Ltda., 2012b). Así pues que la Ley TIC (Ley 1341) define al MinTIC como la entidad que se encarga de diseñar, adoptar y promover las políticas, planes, programas y proyectos del sector de las Tecnologías de la Información y las Comunicaciones.

En adición a las funciones generales de cada ministerio definidas en el artículo 59 de la Ley 489 de 1998, el artículo 17 de la Ley 1341 de 2009 dispone los objetivos del MinTIC, mientras que a partir del artículo 18 de la misma Ley se establecen las funciones específicas de dicha entidad para cumplir con los objetivos definidos y que se listan a continuación:

- Diseñar, formular, adoptar y promover las políticas, planes, programas y proyectos del Sector de Tecnologías de la Información y las Comunicaciones, en correspondencia con la Constitución Política y la ley, con el fin de contribuir al desarrollo económico, social y político de la Nación y elevar el bienestar de los colombianos.
- Promover el uso y apropiación de las Tecnologías de la Información y las Comunicaciones entre los ciudadanos, las empresas, el Gobierno y demás instancias nacionales como soporte del desarrollo social, económico y político de la Nación.
- Impulsar el desarrollo y fortalecimiento del Sector de Tecnologías de la Información y las Comunicaciones, promover la investigación e innovación, buscando su competitividad y avance tecnológico conforme al entorno nacional e internacional.
- Definir la política y ejercer la gestión, planeación y administración del espectro radioeléctrico y de los servicios postales y relacionados, con excepción de lo que expresamente determine la ley.

En (Tovar F. Asociados Abogados Ltda., 2012b) se hace un extenso detalle de esta entidad, su organización interna y administrativa, sus funciones específicas, las responsabilidades, su historia, etc.

3.2 Metodologías Ágiles

El ambiente de trabajo de la gran mayoría de las empresas de desarrollo de software, y en general dedicadas al desarrollo del conocimiento, presenta grandes diferencias respecto al que origino la gestión de proyectos predictiva (tradicional), debido a la rápida y constante evolución de los mercados se requieren estrategias enfocadas a la entrega temprana y continua de productos con resultados tangibles, y a la respuesta ágil y flexible. Ahora se construye el producto mientras se modifican y aparecen nuevos requisitos, pues el cliente parte de una visión medianamente clara de su producto, sin embargo, el grado de innovación que requiere, su evolución y adaptación a los cambios generados por la dinámica del sector de su negocio, le impiden predecir con detalle cómo ser el resultado final. Podría llegar a afirmarse que ya no se

generan “productos finales”, en vez de estos se entregan productos en una dinámica de continua evolución y mejora.
Dado que las metodologías tradicionales son muy rígidas y “pesadas”, pues se dedican tiempo y recursos en el desarrollo de una planificación exhaustiva y en las que se busca minimizar los cambios debido al costo eleva que implica solucionar los errores detectados o introducir modificaciones cuanto más avance el proyecto (Boehm et al., 1981), se hacen inapropiadas en un mercado dinámico. Las metodologías ágiles nacen de la necesidad de adaptarse a un entorno variante e inestable, que inherentemente implica el cambio y la continua evolución rápida y en el que se requieren entregas tempranas que satisfagan las necesidades cambiantes de los usuarios y lógicamente del cliente. Es así que las metodologías ágiles surgen para dar soluciones a los principales inconvenientes que presentan las metodologías tradicionales:

Se invierten cantidades considerables de tiempo y esfuerzo en la fase inicial que es la planificación, cuyo resultado debe ser una “fotografía exacta” de lo que desea el cliente (quien generalmente no tiene claro lo que desea), además de minimizar al máximo los cambios de requerimientos por el costo que involucra lo que la hace inflexible, además de ser el cliente quien debe acarrear con los costos del nuevo requerimiento. En esta fase el cliente no ve más que papeles.
Se requiere de una documentación excesiva, que no siempre es útil y que inherentemente genera un gasto de tiempo y esfuerzo que no genera valor al producto y el desarrollo se hace más lento.

Si tiende a acumular los riesgos y dificultades a medida que se avanza en el desarrollo del producto, lo cual puede ocasionar retrasos en la entrega.
Es una metodología burócrata donde no hay “contacto” cercano y continuo con el cliente.

Las dificultades o nuevos requerimientos aparecen en las fases finales que son donde el producto es testeable y en donde es “visible” para el cliente, generando costos elevados y retrasos en el tiempo de entrega.

En la Tabla 1 se presenta un cuadro comparativo entre las metodologías ágiles y las tradicionales, mientras que la figura 3 muestra los riesgos y costos de cambios en las fases de cada una. Finalmente la figura 2 muestra el ciclo de vida de las dos metodologías.



Figura 2. Ciclos de vida ágil y tradicional
Fuente: http://juanbsandoval.blogspot.es/tags/postagilismo/

Ágiles	Tradicionales
Se basan en heurísticas provenientes de prácticas de producción de código	Se basan en normas procenientes de estándares seguidos por el entorno de desarrollo
Cierta resistencia a cambios	Preparados para cambios durante el proyecto
Reglas impuestas externamente	Reglas impuestas internamente por el equipo
Proceso muy controlado, numerosas normas	Proceso poco controlado con mínimas normas
Contrato prefijado	Contrato flexible, indeterminado o incluso inexistente
El cliente es ajeno al desarrollo, solo interactúa mediante reuniones	El cliente es parte del desarrollo

Tabla 1: Comparación de metodologías.

Como se puede observar en la imagen siguiente el costo de cambios en los requerimientos empleando metodologías tradicionales es muy elevado finalizando el proyecto, este comportamiento se genera debido a que es necesario replantearse muchas cosas, se pierde una cantidad importante de los recursos invertidos en las fases anteriores y se debe volver a hacer el análisis y la planeación. Mientras que con las metodologías ágiles el costo de cambio no es tan elevado y es casi constante durante el ciclo de vida del proyecto debido a la adaptación y preparación para el cambio al no dedicar demasiados recursos en la planeación exhaustiva.

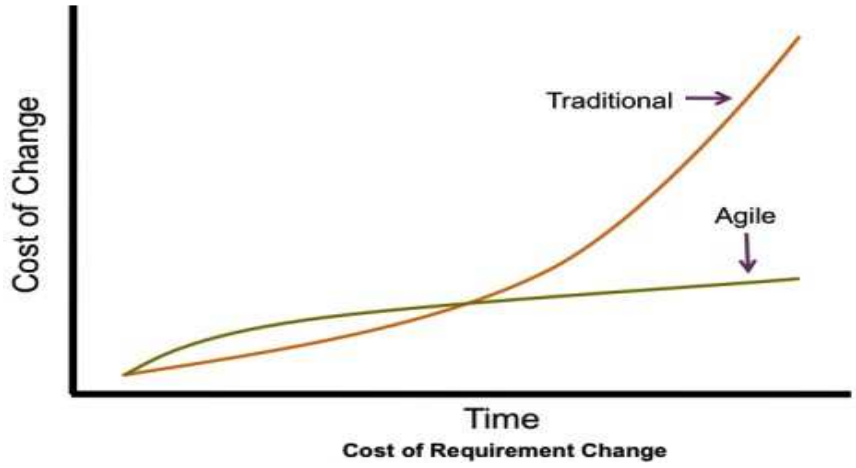


Figura 3. Costo de cambios en ambas metodologías
Fuente: http://coachingsscrum.blogsapot.com

Como se observa, se valoran más las personas que los procesos sin indicar que estos no sean necesarios, se valora más la adaptación y la agilidad que la documentación y la planeación rigurosa sin dejarlas de lado por completo. Además de los valores se acordaron 12 principios que se listan a continuación:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity–the art of maximizing the amount of work not done–is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

3.3 SCRUM

Es un marco de trabajo que se basa en el empirismo y la experticia, su origen se dio gracias al estudio desarrollado por los profesores Horotaka Takeuchi e Ikujiru Nonaka a inicios de 1980 quienes analizaron el proceso de desarrollo de productos tecnológicos de las principales industrias de sector: Canon, Honda, Nec, Hewlett-Packard, entre otras, y cuyos resultados se publicaron en 1986 (Takeuchi and Nonaka, 1986). El resultado de este análisis Takeuchi y Nonaka lograron identificar la similitud entre la forma de trabajo de los equipos de estas organizaciones y el avance en formación SCRUM propia del Rugby, de ahí que ese nuevo framework adopte este nombre.

Este framework es apropiado en entornos de desarrollo con requerimientos inestables, no muy bien definidos y en los que se requieres rapidez y flexibilidad. Entre los años 1993 y 1995 se conforma el primer equipo SCRUM y se formaliza este framework gracias a Jeff Sutherland y Ken Schwaber (Deemer et al., 2010).

3.3.2 El Modelo

La Figura 4 muestra gráficamente el marco de Scrum con sus elementos: Roles, artefactos y eventos. El modelo es iterativo incremental y se desarrolla en ciclos de trabajo denominados Sprints, los cuales son limitados en tiempo (Time boxing), tienen una duración fija y se realizan uno tras otro.

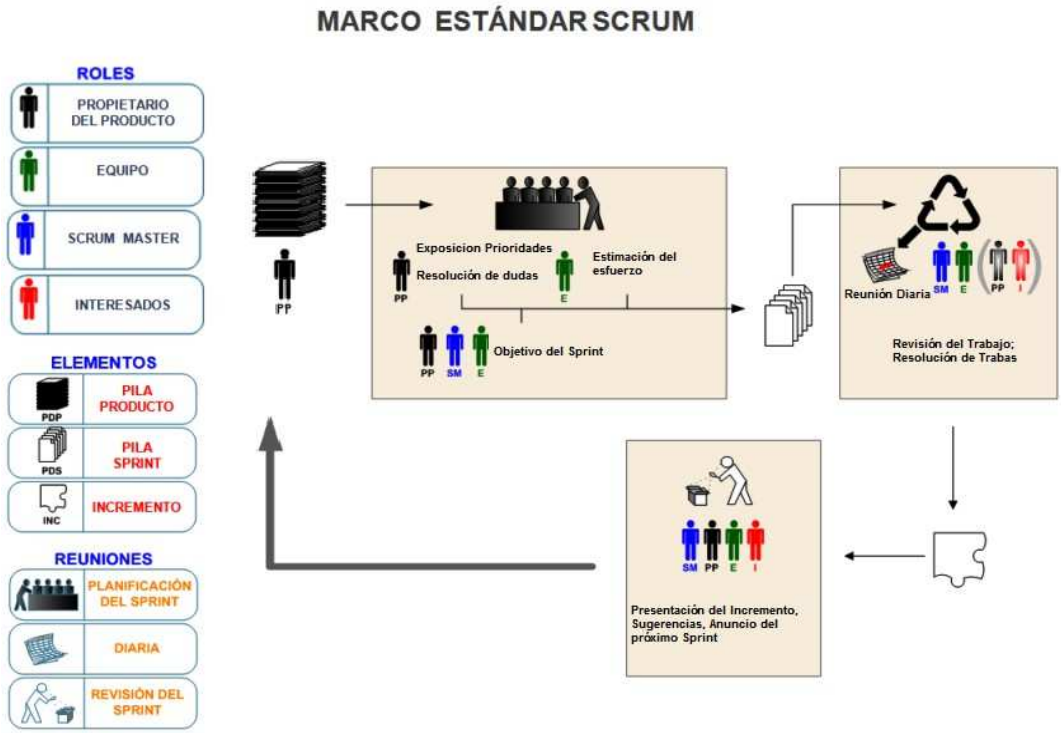


Figura 4. Marco de SCRUM
Fuente: http://www.scrummanager.net/

Los tres pilares sobre los que se soporta la gestión y el control del proceso son (Schwaber et al., 2013).

• Transparencia: Todos aquellos comprometidos con el resultado del desarrollo deben poder visualizar los elementos importantes de proceso. Estos elementos deben estar bajo un estándar común, de tal modo que quienes lo visualicen entiendan la misma idea o concepto.

• Inspección: Se deben inspeccionar los artefactos y el progreso frecuentemente a fin de detectar variaciones, esta inspección no debe interferir con el trabajo por esto sería ideal que las realizaran inspectores expertos.

• Adaptación: Si como resultado de la inspección se detecta que existen elementos que se desvían de la aceptación y que el producto no estará en concordancia con la idea inicial Y/o que se corre el riesgo de ser rechazado, se deben hacer los ajustes necesarios para minimizar mayores desviaciones. En Scrum se emplean cuatro reuniones para realizar la inspección y adaptación: Sprint Planning Meeting, Daily Scrum Meeting, Sprint Review, Sprint Retrospective Meeting.

Para desarrollar el producto se parte de una visión general de lo que el cliente desea obtener y a partir de ella se especifican y detallan los ítems más prioritarios, los cuales se incluirán en el Sprint y a los cuales el equipo se compromete a entregar al finalizar el Sprint. Durante el transcurso del sprint se desarrollan reuniones diarias de corta duración con el objeto de examinar el trabajo desarrollado el día anterior, el trabajo para el día presente y los posibles inconvenientes que se tengan y actualizar las gráficas empleadas para visualizar el “estado” del sprint.

La gestión de la evolución del proyecto se realiza en base al trabajo pendiente por realizar (Palacio, 2014), para lo cual se llevan a cabo reuniones de revisión al finalizar cada sprint en las que se hace la presentación de la nueva funcionalidad (incremento) ante el cliente y los interesados de quienes se recibe la retroalimentación que se tiene en cuenta en los próximos sprints.

La Tabla.2 presenta los elementos agrupados en roles, artefactos y eventos que serán descritos con detalle a continuación:

Elementos de Scrum	
Roles	Product Owner Scrum Team Scrum Master stakeholders
Artefactos	Product Backlog Sprint Backlog User Stories Sprint Tasks Sprint Burndown Chart Sprint
Eventos	Sprint Planning Meeting Backlog Refinement Meeting Daily Scrum Meeting Sprint Review Meeting Sprint Restospective Meeting

Tabla 2. Elementos de Scrum

3.2.3 Roles

Los roles se clasifican entre comprometidos e involucrados de acuerdo a la fábula del cerdo y la gallina (Keith, 2010). Los comprometidos (cerdos) son: el Scrum Team, el Scrum Master y el Product Owner aunque en ciertos aspectos puede ser considerado como gallina (involucrado), entre estos roles no existe ningún nivel de jerarquía; mientras que los involucrados (gallinas) son los Stakeholders.

A. Scrum Team (ST): El equipo scrum ST está compuesto por los individuos que desarrollan el producto trabajando en conjunto. Es importante para su eficiencia que cada miembro siga un objetivo en común, se adhiera a las mismas normas y que muestre respeto a los demás. Los equipos desde su inicio pasan por diferentes etapas que se describen el modelo Tuckman: Formación, Tormenta, Normalización y Ejecución (Tuckman, 1965).
Los Scrum Teams siempre tienen las siguientes características (Kniberg, 2007):

- El Equipo Scrum está facultado.
- Está trabajando tan autónomo, como le sea posible.
- El Equipo Scrum es auto-organizado.
- Las habilidades dentro del equipo Scrum se equilibran.
- Un equipo Scrum es pequeño y no tiene sub-equipos.
- Las personas dentro del equipo Scrum trabajan a tiempo completo en el equipo.
- Las personas están localizadas en un mismo sitio.

Reglas y Normas: El entorno define algunas de las normas que cada miembro del Scrum Team debe seguir, algunas de estas se desarrollan durante la fase de Normalización, y son muy importantes pues de no existir los miembros del equipo tendrán que perder constantemente valioso tiempo para cambiar entre los diferentes sistemas de valores y conjuntos de reglas. Ejemplos de tales normas y reglas son: hora y lugar de la reunión diaria de Scrum, la definición de Hecho (DoD) utilizada para decidir si el trabajo está terminado o no, directrices de codificación, herramientas a utilizar, etc.

Responsabilidad: El Scrum Team en conjunto es responsable de la entrega comprometida en el tiempo y con la calidad definida. Un buen resultado o un fracaso nunca se atribuyen a un solo miembro, siempre el resultado es del Scrum Team.

Empoderamiento y Organización Propia: El Scrum Team tiene que estar facultado para definir lo que se compromete a entregar al final del sprint, como los resultados esperados han sido desglosarse en tareas, que tarea llevara a cabo y en qué orden se realizan. Solo si el Scrum Team está facultado para decidir estas cosas va a trabajar con la más alta motivación y rendimiento posible.

Conjunto equilibrado de habilidad: Las personas dentro del Scrum Team con toda seguridad, tienen las habilidades y enfoques especializados. Sin embargo, para lograr el mejor rendimiento posible será optimo tener un conjunto equilibrado de habilidades, solo entonces el Scrum Team será capaz de hacer frente a los retos siempre cambiantes y pueden actuar de manera tan autónoma, como sea posible. Por un lado, esto significa que un Scrum Team debe ser multidisciplinario (desarrolladores, probador, arquitectos, etc.) desde el principio. Por otro lado, esto también significa que cada miembro del equipo debe aprender un poco de la especialización de cada uno, por ejemplo, si es necesario para naturalmente llegar a la meta comprometida, un desarrollador también debe realizar o escribir pruebas. Una norma muy importante es que dentro del marco de Scrum no se clásica por habilidades, por ejemplo tester y arquitecto, todos comparten el mismo título “Scrum Team Member", incluso si la habilidad principal no es el desarrollo de código de producción.

Tamaño: Los Scrum Teams son pequeños. El tamaño ideal es de 7 +/- 2 personas. Si hay más gente la sobrecarga de la comunicación se hace muy grande y el equipo debe dividirse en varios Scrum Teams. Estos Scrum Teams deben coordinarse y comunicarse entre sí, pero de trabajar de forma independiente.

Locación: Para minimizar la sobrecarga de comunicación innecesaria cada Scrum Team debe ubicarse. Si el trabajo tiene que ser repartidos en varias ubicaciones, deben crearse Scrum Teams independientes.

A. Responsabilidades del Equipo Scrum: El Scrum Team y cada uno de los miembros tienen ciertas responsabilidades que deben cumplirse: Tienen que desglosar los requisitos (historias de usuario) en elementos más pequeños (tareas), estimarlas y distribuirlas, en otras palabras, esto significa que tienen que crear el Sprint Back log; además deben participar en la reunión diaria (Scrum Daily Meeting), actualizar el estado y los esfuerzos que quedan para sus tareas para permitir la creación del Burndown Chart y desarrollar el incremento.

B. Scrum Master (SM): Es el entrenador del equipo, ayuda a que se consiga el máximo rendimiento del Scrum Team. Es el responsable de asegurar que Scrum es entendido y adoptado, su principal función es asegurar que el Scrum Team trabaja ajustándose a la teoría, prácticas y reglas de Scrum.

El Scrum Master debe ser un líder que está al servicio del Scrum Team, ayuda a las personas externas al equipo Scrum a entender que interacciones pueden ser de ayuda y cuáles no, busca que se modifiquen estas interacciones de tal forma que el Scrum Team pueda concentrar su esfuerzo en el desarrollo del producto. Además es el encargado de hacer la logística y liderar las diferentes reuniones (eventos) de Scrum.

El SM se relaciona con los demás roles de scrum y con la organización como se muestra en la Tabla 3 (Schwaber et al., 2013).

Relación del SM con	
Product Owner	Le ayuda a encontrar técnicas para gestionar la lista de producto de manera efectiva Busca que el PO conozca como ordenar la lista de producto para maximizar el valor Facilitar los eventos de Scrum según se requiera o necesite. Entender y practicar la agilidad
Scrum team	Guia al equipo de desarrollo en ser auto-organizado y multifuncional Eliminar impedimentos para el progreso del Equipo de desarrollo Facilitar los eventos Scrum Guiar al equipo de desarrollo en el entorno de organizaciones en las que scrum aún no ha sido adoptado o entendido Ayuda a al equipo a crear productos de alto valor.
Organización	Liderar y guiar a la organización en la adopción de scrum Planificar las implementaciones de scrum en la organización Motivar cambios que incrementen la productividad del equipo Trabajar con otros scrum masters para incrementar la efectividad de la aplicación de scrum en la organización

Tabla 3. Relación del SM con los demás roles

C. Product Owner (PO): El dueño del producto Product Owner (PO) es un individuo miembro de la organización que requiere el software y es el responsable de conducir el proyecto desde la perspectiva del negocio. Es quien debe comunicar una visión clara del producto y definir sus características principales, priorizando los requerimientos del cliente para que el proceso de desarrollo se centre en aquellos requerimientos necesarios para la organización. En todo momento el PO debe contribuir con el equipo para remover todas las dudas que surgen acerca de los requerimientos. Por esta razón, es necesario que el lugar de trabajo del PO este en el mismo espacio físico que el equipo.

Las responsabilidades del PO (que puede ser interno o externo a la organización) son:

* Gestionar el Product Back log: Crear las historias de usuario, priorizar el PB y reorganizarla (antes de cada sprint).
* Ser el representante de todas las personas interesadas (stakeholders) en los resultados del proyecto (internas o externas a la organización, promotores del proyecto y usuarios o consumidores finales del producto) y actuar como interlocutor único ante el equipo, con autoridad para tomar decisiones.
  * Definir los objetivos del producto o proyecto.
  * Dirigir los resultados del proyecto y maximizar su ROI:
    - Es el propietario de la planificación del proyecto y por tanto se deben respetar sus decisiones: crea y mantiene la lista priorizada con los requisitos necesarios para cubrir

los objetivos del producto o proyecto, conoce el valor que aportara cada requisito y calcula el ROI a partir del coste de cada requisito que le proporciona el equipo.
- Reparte los objetivos/requisitos en iteraciones y establece un calendario de entregas.
- Antes de iniciar cada iteración re planifica el proyecto en función de los requisitos que aportan más valor en ese momento, de los requisitos completados en la iteración anterior y del contexto del proyecto en ese momento (demandas del mercado, movimientos de la competencia, etc.).

* Colaborar con el equipo para planificar, revisar y dar detalle a los objetivos de cada iteración.
* Participar en la reunión de planificación de iteración, proponiendo los requisitos más prioritarios a desarrollar, respondiendo a las dudas del equipo y detallando los requisitos que el equipo se comprometer a hacer.
* Estar disponible durante el curso de la iteración para responder a las preguntas que puedan aparecer.
* Participar en la reunión de demostración de la iteración, revisando los requisitos completados.

D. Stakeholders SH: Los Stakeholders (SH) son las personas interesadas en el producto, como lo son: clientes, usuarios, patrocinadores, etc. Su rol no es obligatorio pero tiene una gran importancia ya que aportan al Product Owner muchas ideas a partir de las cuales el crea su visión del producto. Se consideran "gallinas" junto con el PO pues están involucrados en el desarrollo mas no comprometidos. No es conveniente que interactúen directamente con el SM ni mucho menos con el ST ya que pueden generar "ruido" y distracciones.

3.2.4 Artefactos

Los artefactos de Scrum representan trabajo o valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación. Los artefactos definidos por Scrum están diseñados específicamente para maximizar la transparencia de la información clave, que es necesaria para asegurar que todos tengan el mismo entendimiento del artefacto.

A. Product Back log: La pila de producto Product Back log (PB) es la lista de requerimientos que le dan valor de negocio al producto, contiene las funcionalidades y/o requisitos que el producto ira adquiriendo con cada iteración. Es gestionada por el Product Owner quien debe crearla y ordenarla por prioridad. Las características principales del PB son:

- Contendrá los requerimientos del producto expresados en historias de usuario (User Stories).
- Se gestiona entre cada sprint exclusivamente por parte del PO y nunca durante el transcurso de un Sprint.
- Visible a todos, incluidos los stakeholders.
- Evoluciona constantemente, es un artefacto vivo.
- Antes de iniciar cualquier Sprint es necesario decir los objetivos y tener la lista de requerimientos no tan extenso los detalle pero si con la información suficiente para que el equipo pueda desarrollarla. Los elementos de las listas se llaman historias de usuario (User Stories) y se expresan en terminología del cliente (Kniberg, 2007).

B. User Stories: Las historias de usuario User Stories (US) son los ítems del Product Back log y contienen la información de cada funcionalidad a desarrollar. Estas historias tiene los siguientes campos como mínimo.

- ID: Un identificador único para facilitar el seguimiento.
- Nombre: Título descriptivo de la historia.
- Descripción: Una descripción corta, no más de 15 palabras, lo suficientemente clara para que el PO comprenda de que se trata y que se distinga de las demás. Ejemplo: \Como usuario quiero consultar el reporte de notas en una página web".
- Prioridad: La importancia que le da el PO a la historia de acuerdo al valor de negocio que representa para él. Entre más alto el valor, más importante es la historia.
- Estimación: La valoración que el Scrum Team da en cuanto el esfuerzo que se requiere para desarrollar la historia. Sus unidades son puntos de esfuerzo, la estimación absoluta no es importante, es más importante la relatividad de la estimación: que una historia de 2 puntos dure la mitad que una de 4(Kniberg, 2007).
- Como Probarlo: Una descripción a alto nivel de cómo se demostrara la historia desarrollada. Es una breve descripción de un test.

Notas: Se agrega información adicional como fuentes de información, clarificación, etc. Normalmente muy breve
La Figura 5 muestra un ejemplo del formato de las User Stories.



Figura 5. Formato Historias de Usuario
Fuente: Autor

C. Sprint Back log (SB): la Pila de Sprint SB está compuesta por las User Stories que el Product Owner y el Scrum Team negocian durante la Sprint Planning Meeting y este último se compromete a entregar al finalizar el sprint. Se expresa en una lista de tareas (Tareas del Sprint (TS)). Su alcance idealmente es fijo y es visible a todo el equipo. Es gestionado por el Scrum Team durante la Daily Scrum Meeting.

D. Sprint Tasks (TS): Las TS son la descomposición de las historias de usuario en elementos más pequeños que especifican como alcanzar el que de las User Stories y son identificadas por el Scrum Team durante la Sprint Planning Meeting. Son las actividades deben realizarse para desarrollar la historia de usuario a la cual pertenecen. Se expresan en lenguaje técnico del equipo y cada persona se compromete a trabajar sobre la tarea que el voluntariamente escoja y/o que sea asignada.

E. Sprint Burndown Chart (SBC): EL grafico de trabajo realizado Sprint Burndown Chart (SBC) es una herramienta que ayuda a visualizar y hacer seguimiento al trabajo realizado y el trabajo pendiente por hacer. Su gestión y actualización se realiza en la DSM por

parte del ST. Es importante saber interpretar la gráfica y actuar ante los resultados y variaciones que está presente. La Figura. 6 muestra un ejemplo del Sprint Burndown Chart.



Figura 6. Sprint Burndown Chart
Fuente: Autor

3.2.5 Eventos

A. Product Back log Refinement Meeting (PBRM): La mayoría de las US deben re- finarse, ya que son grandes y poco comprendidos. En la reunión de refinamiento del PB, el equipo estima la cantidad de esfuerzo que se debe invertir para completar los ítems del PB proporcionando información técnica para ayudar al PO a priorizarlos. Las historias que requieren esfuerzos demasiado grandes se dividen y se clarifican, teniendo en cuenta temas tanto de negocio como técnicos. El PO ayuda al equipo resolviendo dudas e inquietudes con el fin de que todos los miembros del ST tengan la misma idea de cada historia de usuario. Las estimaciones de esfuerzo se realizan por diversos métodos, el más popular y más empleado es el Planning Póker.
   A veces, un subconjunto del equipo, junto con el PO y los SH, descompone y dividen los ítems del PB antes de involucrar a todo el equipo en la estimación. Un SM experimentado puede ayudar al equipo a identificar finas rebanadas verticales de trabajo que tienen valor comercial, mientras promueve una definición rigurosa de “terminado” que incluye las pruebas adecuadas y refactorización.

B. Sprint Planning Meeting (SPM): Antes de empezar un Sprint, el PO, el SM y el ST tienen una Reunión de Planificación del Sprint SPM que se divide en dos partes (Deemer et al.,2010). En la primera parte se definen los objetivos del Sprint, la duración de este y se negocian que ítems del PB intentaran convertir en producto funcional durante el Sprint, en resumen se responde ¿qué puede ser terminado en este Sprint? El PO es el responsable de declarar cuales son los ítems más importantes para el negocio. El equipo es responsable de seleccionar la cantidad de trabajo que se comprometen a realizar sin acumular deuda técnica, lo cual es una práctica clave en Scrum: decidir qué cantidad de trabajo se compromete a realizar basándose en su experiencia y conocimiento tanto técnico como funcional del equipo, en vez de que sea un trabajo asignado por alguien más; el SM es el moderador y el encargado de hacer que la reunión fluya y no se estanque pues es una reunión limitada en tiempo (timeboxing). El equipo “toma” el trabajo desde el PB hacia el SB. En la segunda parte se responde a la pregunta ¿Cómo se conseguirá hacer el trabajo necesario para entregar el incremento?; para ello se dividen las US en las Sprint

Task en base a las necesidades más detalladas del PO y de los conocimientos técnicos del ST.

C. Daily Scrum Meeting (DSM): La reunión diaria de Scrum DSM es una reunión con un bloque de tiempo de 15 minutos en la cual el Scrum Team sincroniza sus actividades y crea un plan para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Scrum Diario y haciendo una proyección acerca del trabajo que podría completarse antes del siguiente:

El Scrum Diario se realiza todos los días en el mismo lugar y a la misma hora, con la misma duración. Durante la reunión, cada miembro del Scrum Team responde a las siguientes preguntas planteadas por el SM:

• ¿Que hizo ayer que ayudo al ST a lograr el Objetivo del Sprint?

• ¿Que hará hoy para ayudar al ST a lograr el Objetivo del Sprint?

• ¿Ve algún impedimento que evite que el ST o yo logremos el Objetivo del Sprint?

Otro propósito que tiene el DSM es evaluar el progreso hacia el Objetivo del Sprint y la tendencia que tiene este progreso de cara a la culminación del trabajo pendiente en el Sprint Back log. Además se debe actualizar el SBC.

D. Sprint Review Meeting (SR): En Scrum la inspección y adaptación son pilares para ver y saber lo que está pasando y luego evolucionar en base a la retroalimentación, en la repetición de ciclos. La Revisión del Sprint es una actividad para este propósito, es un tiempo para que el PO conozca lo que está sucediendo con el producto y con el equipo (es decir, una revisión de la Sprint); y para que el equipo se entere lo que sucede con el acroshort PO y el mercado.
Así pues, un elemento vital es una profunda conversación entre el ST y el PO para conocer la situación y obtener retroalimentación. La revisión incluye una demostración de lo que el equipo construido durante el Sprint, pero no es el objetivo principal.

E. Sprint Retrospective Meeting (SRM): La reunión de retrospectiva del Sprint es muy importante pues es donde se analiza lo que ocurrió durante el Sprint. Es la reunión siguiente a la del Sprint Review y antes de la siguiente planeación de Sprint, durante la reunión se discuten los siguientes puntos:
    - Inspeccionar como fue el último Sprint en cuanto a personas, relaciones, procesos y herramientas.
    - Identificar y ordenar los elementos más importantes que salieron bien y las posibles mejoras.
    - Crear un plan para implementar las mejoras a la forma en la que el Equipo Scrum desempeña su trabajo.
    - Esta reunión es fundamental para ayudar al equipo a mejorar, pues del examen interno se detectan los aspectos a mejorar y se crean compromisos que adquiere cada miembro para con el resto del equipo y la organización. Esta reunión es manejada por el Scrum Master quien debe buscar que cada miembro comprenda su importancia y además sobrepasar las barreras psicológicas mencionadas por (James, 2010).

3.3 Conceptos de Programación

En esta sección se expondrán los conceptos y fundamentos de programación empleados para desarrollar el proyecto, como lo son la programación orientada a objetos (Object Oriented Programming (OOP)) que es el paradigma empleado, las bases de datos y programación multihilos.

3.3.1 Programación Orientada a Objetos

La programación orientada a objetos nace como la mayoría de la tecnología a partir de una necesidad. En este caso surge como solución a las limitaciones de su predecesora la programación estructural y a la crisis del software, aunque no ha sido la solución definitiva a esta última. En la programación estructural se diseñan y manejan los programas como grupos de elementos funcionales de código reutilizable llamados procedimientos que definen parámetros de entrada y salida y que además se invocan (llaman) unos a otros modificando datos almacenados en determinadas posiciones de memoria. (Velarde et al., 2006)

El inconveniente de la programación estructural radica en la dificultad y limitación en la reutilización de código, pues únicamente los procedimientos se pueden reutilizar y es muy complicado haceros genéricos y flexibles al ser estructuras de datos abstractas y que generalmente requieren distintas implementaciones. Esto no quiere decir que no sea útil, de hecho en aplicaciones sencillas es más fácil emplear la programación estructural. (Nakov, 2013)

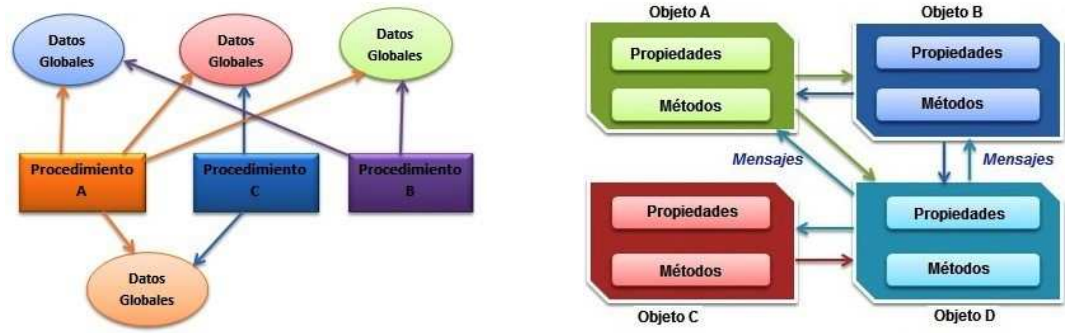
Otro problema que se tiene al emplear la programación estructural es que debe existir una fuerte comunicación, entendimiento y sincronización entre los desarrolladores, si se tiene en cuenta el hecho que cada uno de ellos estará manipulando funciones separadas que pueden trabajar sobre tipos de datos mutuamente compartidos, luego los cambios de un desarrollador deben reflejarse en el trabajo del resto del equipo. Además no modela muy bien el mundo real al separar datos de funciones.

Desde la década de 1960 se vena trabajando en las ideas y principios bases de la OOP con Simula, luego en la década de los 70's aparece Small talk, no fue hasta la década de los 80's que este paradigma iba tomando fuerza entre los programadores debido a la aparición de C, C++ y se extendió a BASIC y Pascal. Seguidamente en Eiel (ahora JAVA) y a partir del año 98 se desarrolla la arquitectura de objetos distribuidos. La Figura.8 muestra gráficamente la comparación de la programación estructurada y la Object Oriented Programming.

La idea en la OOP es pasar de ajustar el mundo al lenguaje, a ajustar el lenguaje a lo que es el mundo real, a modelarlo. Se deja de ver todo como cajas negras (procedimientos) a las que entran y salen datos manejados “aisladamente” para adoptar un modelo de entidades (objetos) que poseen atributos y comportamientos, de manera tal que ya no se manejan datos y funciones por separado, sino que se integran y son parte de cada objeto (encapsulamiento) y por lo tanto cada objeto es responsable de sus datos y funciones. Los objetos interactúan entre sí mediante mensajes para producir un resultado, lo cual modela mejor el comportamiento de las cosas en el mundo real y más fácil modelar los problemas al emplear un paradigma más intuitivo. La Figura.6 ilustra el mapeo de las entidades del mundo real a la OOP.



Figura 7. Modelamiento del mundo real a OOP
Fuente: Autor



(a) Programación Estructurada (b) OOP

Figura 8. OOP vs Estructurada
Fuente: Autor

3.3.2 Conceptos de POO

A continuación se presenta una descripción de los principales elementos y características más importantes que se manejan en OOP:

Objeto: Conceptualmente describe una entidad del mundo real que posee atributos y comportamientos; los primeros son sus características, por ejemplo, en un coche su marca, modelo, velocidad etc.; mientras que los segundos son las acciones que ejecutan ante algún estimulo, como un coche aumenta su velocidad al pisar el pedal del acelerador. Además de estos componentes, un objeto tiene una identidad que lo diferencia de los demás objetos. Formalmente se define un objeto es una instancia de una clase que tiene un ciclo de vida dentro del programa.
Clase: Una clase es una plantilla o modelo que se usa para definir los atributos y métodos que tendría un objeto. De manera más formal se define como la abstracción de las características y comportamientos de alguna entidad que se está modelando en el programa.

Abstracción: En la OOP, no es esencial reproducir todas las características ni comportamientos de una entidad en el programa, pues la cantidad de información sería inmanejable y la gran

mayoría sería información superflua. Es la abstracción la propiedad por la cual se tiene en cuenta solo aquellas características más relevante y se deja de lado aquellas que no lo son desde el punto de vista del problema para así obtener una visión clara del mismo.

Encapsulamiento: Significa reunir en el mismo nivel de abstracción los elementos pertenecientes a una entidad y así tener una buena estructuración. Se emplea conjuntamente con la ocultación que consiste en proteger las propiedades de un objeto para que no sean modificadas por quien no deba, de modo tal que el estado del objeto debe ser responsabilidad de este y lo modifica por medio de sus procedimientos.

Herencia: Con la herencia se generan nuevas clases a partir de clases base existentes, lo cual facilita el rehusó de código. Booch la define como: "una clasificación u ordenamiento de la abstracción". Por ejemplo: a partir de la abstracción animal, se generan las abstracciones aves y mamíferos las cuales además de tener los atributos y comportamientos de animal, especifican los propios.

Poliformismo: Está relacionado con el hecho de que existen comportamientos similares (en resultados) pero con implementaciones distintas. Por ejemplo: para ir de una cuidad A a una ciudad B los aviones se desplazan por aire, mientras que los automóviles van por las carreteras. Avión y automóvil son abstracciones de medios de transporte y ejecutan la misma función desplazarse, pero de distintas maneras.

4. Sprint Estado Intermedio y Código Certificado

Después de la investigación de los temas y los debidos procesos a evaluar con reuniones semanales exponiendo los temas designados para el correcto manejo de las metodologías se procede a realizar todo el proceso de planeación del sprint con los conocimientos adquiridos en la capacitación se realizan las tareas designadas y se procede a la generación de las tareas y el esfuerzo que representan

Tareas Sprint SO Estado intermedio Código Certificado	Esfuerzo
Base de Datos Descarga PDF	2
Almacenar Información Técnica	5
Actualizar Estados Proceso	3
Crear estado y Manejar la Lógica	3
Nuevo Proceso de Radicado	5
Lógica de Radicación Extremo	2
Método Almacenamiento PDF	3
Actualizar Servicio SO	3
Actualizar Servicio Alpha	3
Modificación Proceso de Radicado	3
Generar Código	3
Diseño de Base de Datos ENTITI	2

Tabla 4. Tareas de Sprint Estado Intermedio Cód. Certificado

Después de la generación de tareas se procede a realizar el debido sprint subiendo los cambios diarios en el programa TortoiseSVN para que no se pierda el proceso llevado al estar trabajando

en computador local y que los otros programadores sepan que cambios se están realizando esto es una medida que toda empresa que está especializada en Software usa.

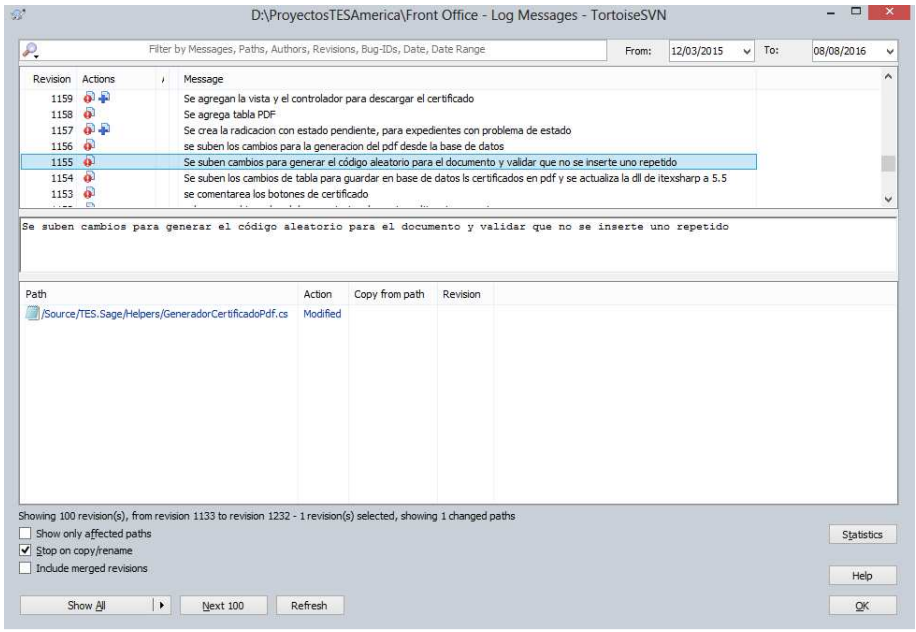


Figura 9. Cambios de generación de código
Fuente: Autor

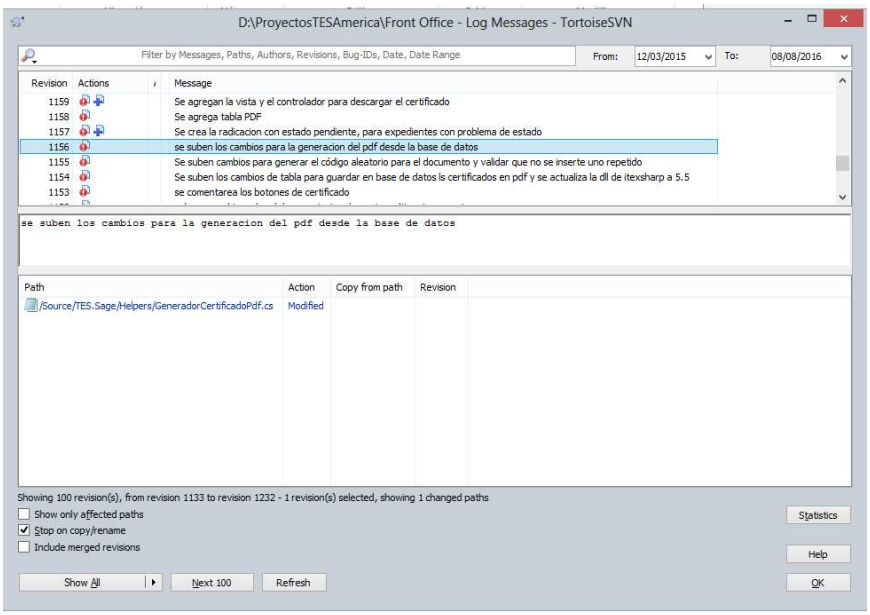


Figura 10. Generación del PDF desde la base de datos
Fuente: Autor

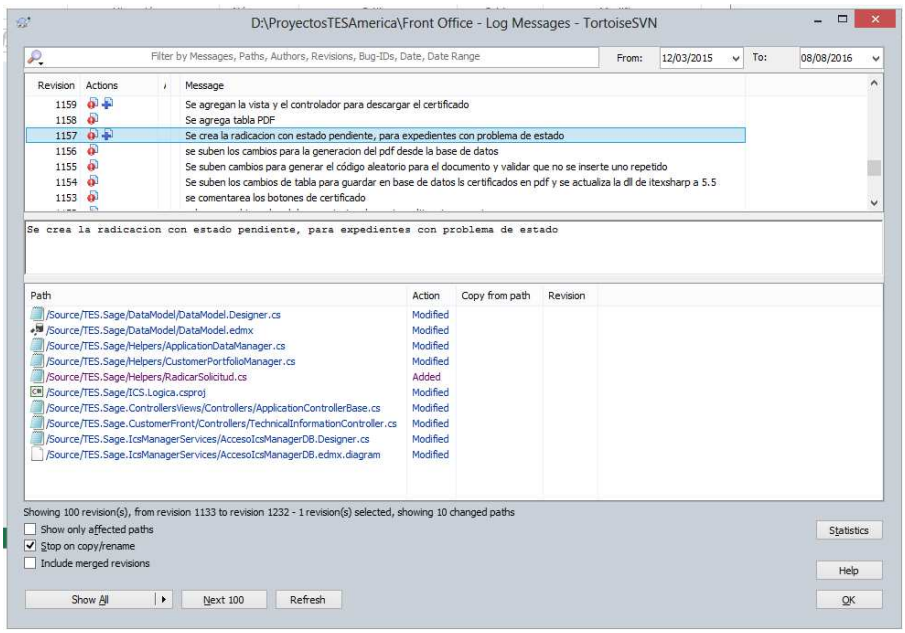


Figura 11. Creación de estado de radicado en estado pendiente
Fuente: Autor

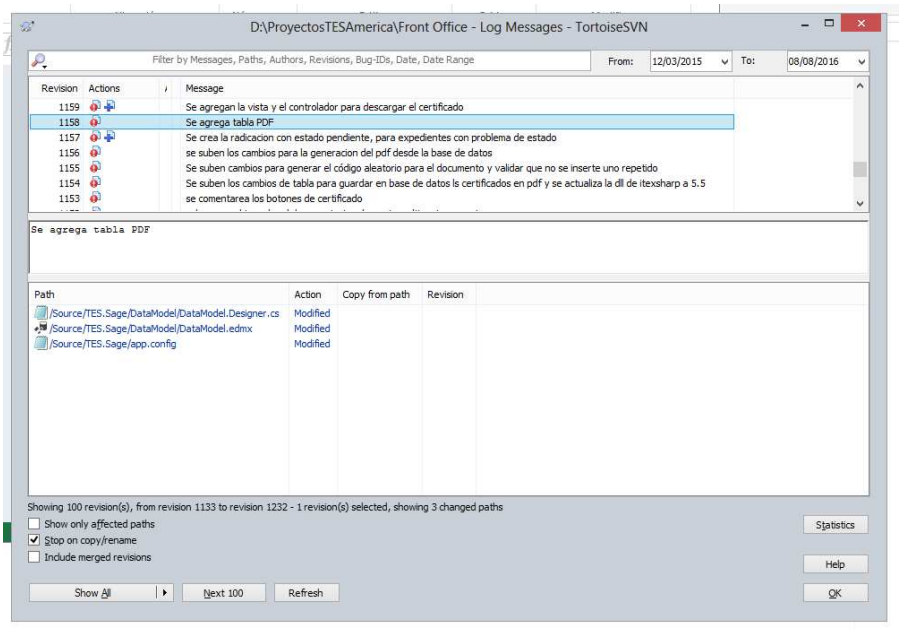


Figura 12. Agrega tabla de PDF
Fuente: Autor

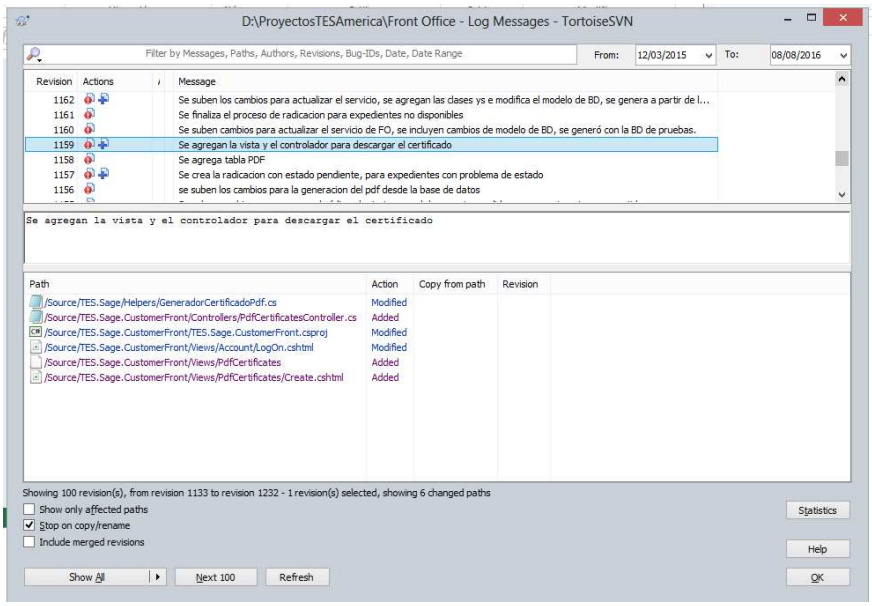


Figura 13. Se agrega la vista y el controlador para el certificado.
Fuente: Autor

5. Resultados Finales

Para los resultados finales luego de realizar las pruebas y que todo esté en funcionamiento se procede a realizar las pruebas de que todo esté en correcto funcionamiento. De acuerdo con el burdown chart se estipula todo el proceso que se llevó a cabo en la realización del proyecto

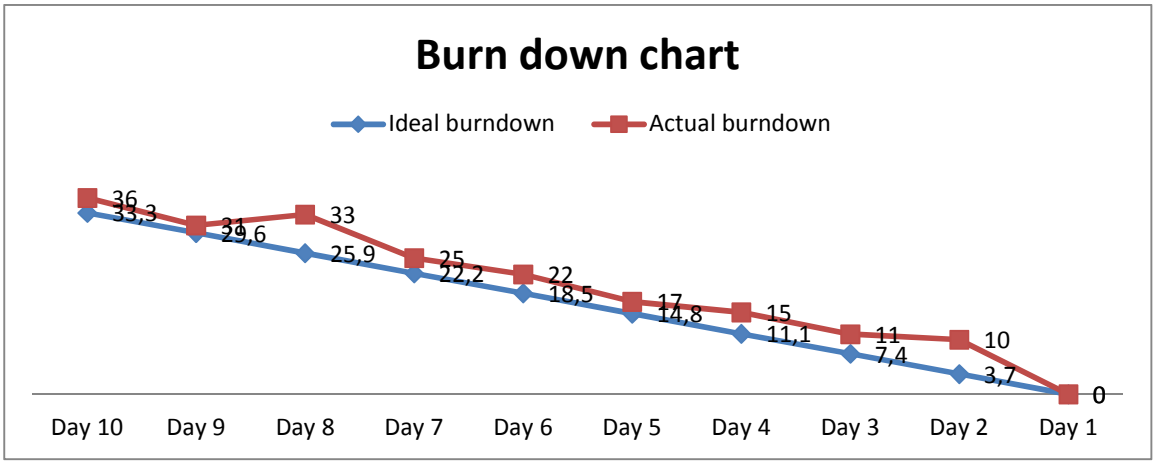


Figura 14. Burdown chart
Fuente: Autor

5.1 Generación de Código de certificado

Se empieza a realizar las pruebas de código único para la generación de certificados, esto se realiza en un ambiente de pruebas para que se le presente al cliente sin ninguna modificación de la información del MInTIc.

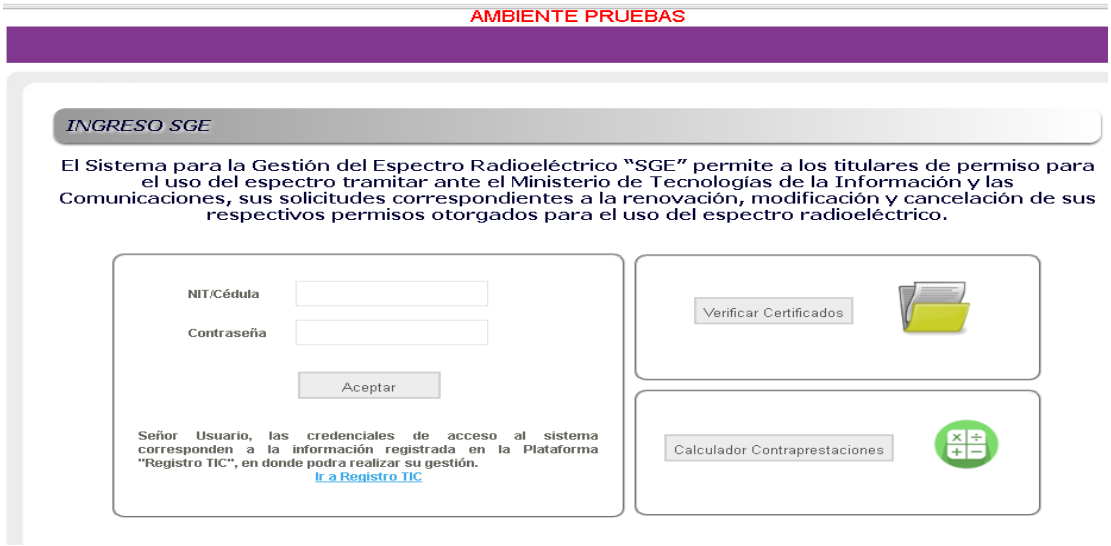


Figura 15. Vista previa
Fuente: Autor

En la vista previa el usuario sin loguearse puede descargar el pdf sin ningún inconveniente solo necesita el código del pdf para descargarlo.

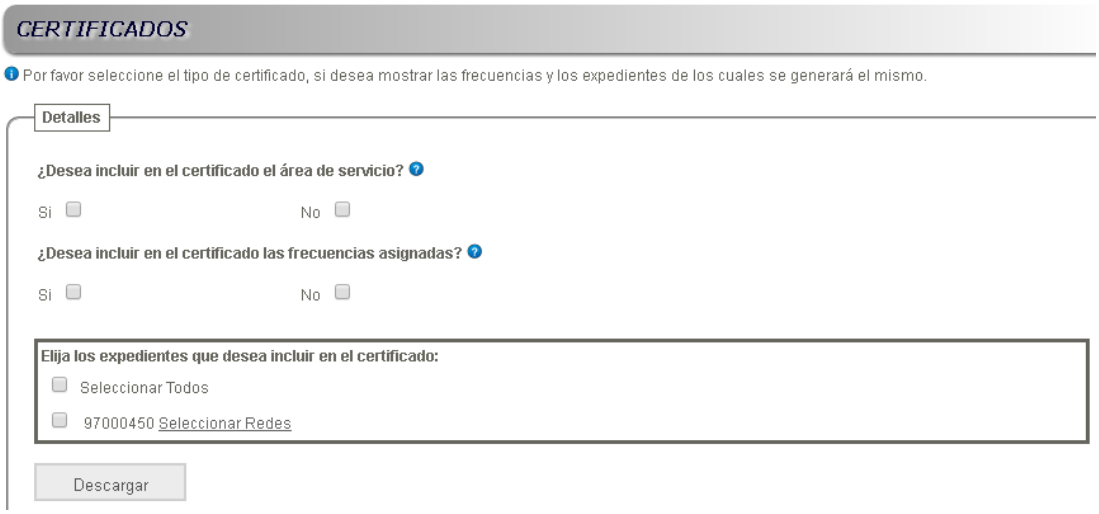


Figura 16. Ventana de Certificados
Fuente: Autor

En la ventana se escoge que tipo de certificado desea descargar si es de área de servicio de frecuencias asignadas.



12 de agosto del 2016 - 03:21:39 p.m.

TODOS POR UN NUEVO PAÍS

MINTIC

LA SUSCRITA SUBDIRECTORA
PARA LA INDUSTRIA DE COMUNICACIONES

CERTIFICADO No. 00000033

HACE CONSTAR QUE:

La empresa **PruebasDev - 360 GRADOS SEGURIDAD LTDA**, con **NIT 860.065.464-3**, tiene habilitación general para la provisión de redes y/o servicios de telecomunicaciones de acuerdo a la ley 1341 de 2009. Se relacionan a continuación los expedientes registrados ante el Ministerio de Tecnologías de la Información y las Comunicaciones:

Áreas de servicio por expediente

La presente se expide a solicitud del interesado el 12-08-2016

Este documento puede ser verificado empleando el siguiente código: 494570d13cc64b87 en la página del Ministerio en el link http://gestion-espectro.mintic.gov.co/

GLORIA PATRICIA PERDOMO RANGEL
Subdirectora para la Industria de las Comunicaciones

Ministerio de Tecnologías de la Información y las Comunicaciones
Edificio Murillo Toro Cra. 8a entre calles 12 y 13
Código Postal: 111711, Bogotá, Colombia
T: +57 (1) 344 3460 Fax: +57 (1) 344 2248
www.mintic.gov.co - www.vivedigital.gov.co

vive digital Colombia

Página 1 - NIT 860.065.464-3

Figura 17. Certificado de Área de servicio por expediente
Fuente: Autor

Por último se descarga el certificado impreso con el código único en el simulador de pruebas.

5.2 Implementación Estado de Radicación

Esta es la Página principal después de ingresar el usuario y contraseña que tenemos por defecto para realizar las diferentes pruebas, para el usuario el nuevo estado de radicación es transparente ya que no debería afectar el correcto uso de donde se está utilizando la herramienta ya cuando la solicitud queda radicado el usuario ve que cambia automáticamente su solicitud.

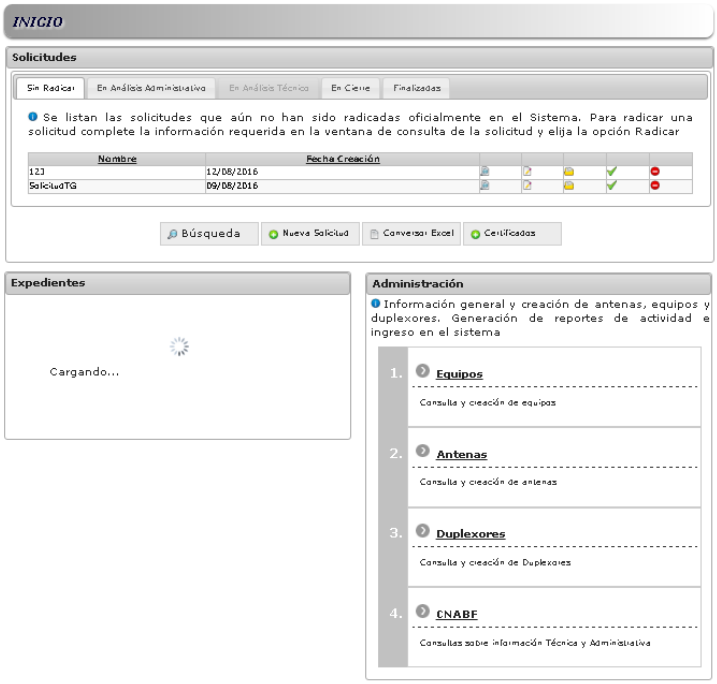


Figura 18. Creación y radicación de solicitudes
Fuente: Autor

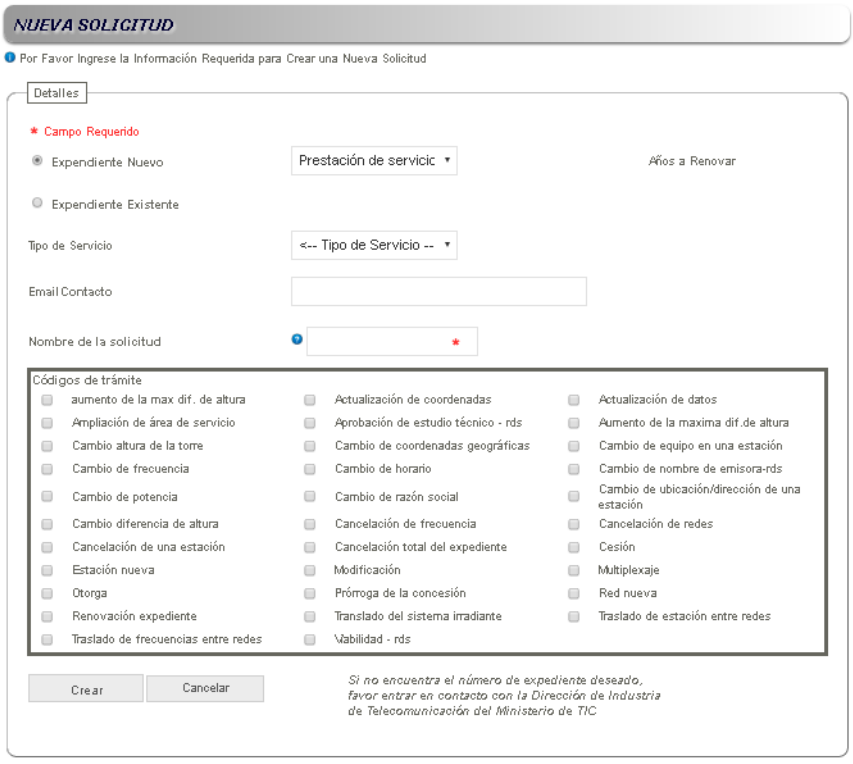


Figura 19. Creación de solicitudes
Fuente: Autor



Figura 20. Radicación de Solicitudes
Fuente: Autor

Aquí se determina si la solicitud está completa o si falta información para ser radicada.



Figura 21. Información de la solicitud
Fuente: Autor

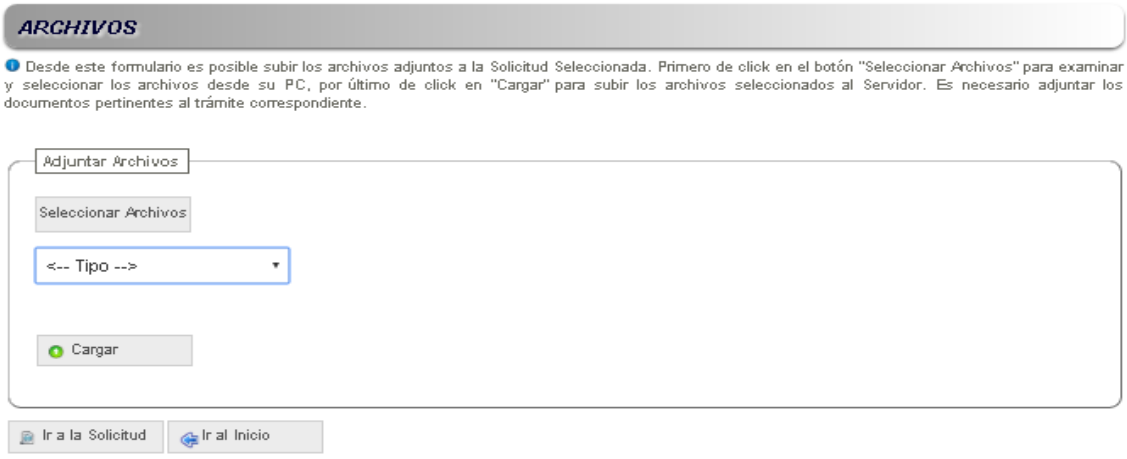


Figura 22. Ventana para cargar adjuntar archivos
Fuente: Autor

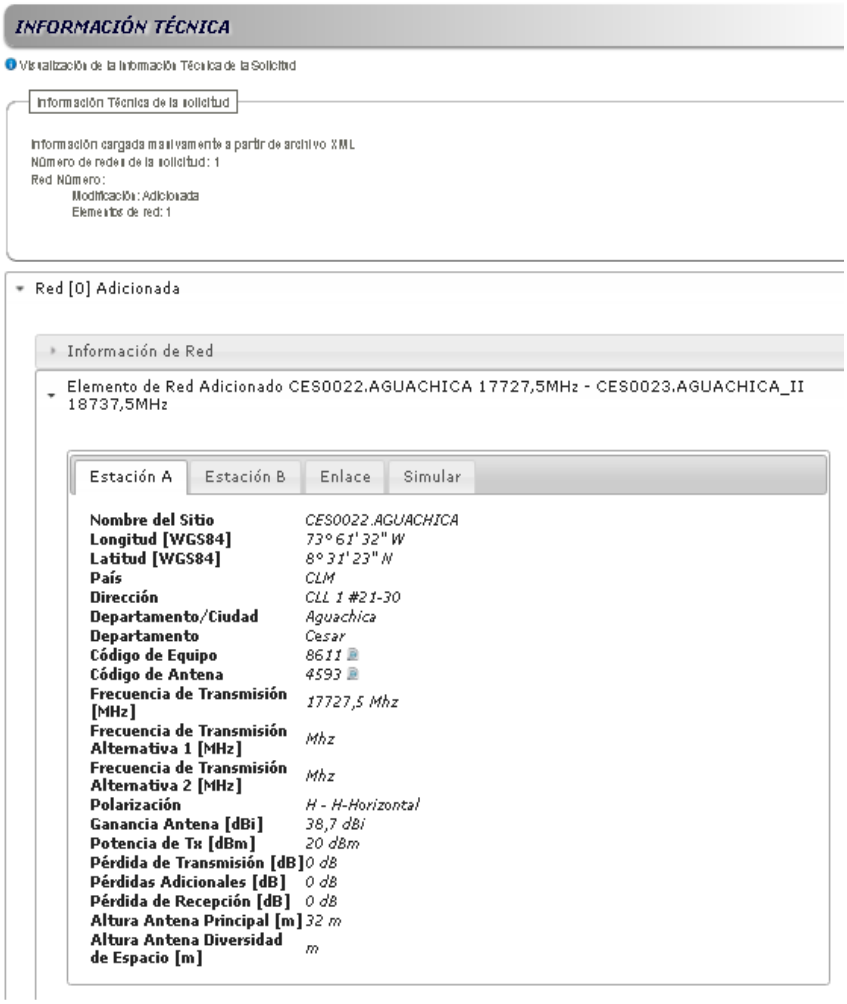


Figura 23. Información Técnica para ser Radicada
Fuente: Autor

6. **Conclusiones**

Después de completar la capacitación a Scrum los conceptos adquiridos fueron fundamentales para realizar este proyecto y son la base para proyectos futuros, es una metodología que es muy fuerte por la facilidad de implantación y por su agilidad en cuanto a cambios ya que los avances se ven desde el principio para que se vayan viendo los avances y sepa cómo se está llevando a cabo el proyecto y si existe cambios que afecte lo menos posible a los costos, los tiempos y el equipo de trabajo.

Nuestro objetivo es encontrar flexibilidad y esta se consigue conociendo todas nuestras posibilidades y emplearlas de la mejor manera a las circunstancias de cada proyecto y/o cada empresa, De cara al futuro y en base a nuestra experiencia, esta metodología comenzará a usarse en un mayor número de empresas tecnológicas, y más concretamente en las dedicadas al mundo de la informática, por sus características tan favorecedoras en este tipo de proyectos.

De la gestión del proyecto también hay que destacar la buena estimación inicial que se realizó y la escasa desviación que finalmente ha resultado, unido a la obtención de casi la totalidad de los beneficios presupuestados inicialmente.

Es importante por tanto, que las pruebas realizadas cubran todos los casos, para así asegurar que la funcionalidad está completada y evitar así futuros errores que conlleven un mayor gasto de tiempo y mayores costes para el proyecto. De todas las pruebas que se deben realizar, hay que hacer especial hincapié en las pruebas unitarias, que permiten verificar el correcto funcionamiento de todos los métodos que contiene nuestra aplicación. Estas pruebas son muy fáciles de implementar y ver donde se encuentra el error del método que están probando.

7. **Referencias Bibliográficas**

Cadavid, A. N., Martinez, J. D. F., and Vélez, J. M. (2013). Revisión de metodologías ágiles para el desarrollo de software. Prospectiva, 11(2):30{39.

Deemer, P., Beneeld, G., Larman, C., and Vodde, B. (2010). The scrum primer. Scrum Primer is an in-depth introduction to the theory and practice of Scrum, albeit primarily from a software development perspective, available at: http://assets. scrumtraininginstitute.com/downloads/1/scrumprimer121.pdf, 1285931497:15.

James, M. (2010). Scrum reference card. CollabNet Inc.

Keith, C. (2010). Agile Game Development with Scrum. Pearson Education.

Kniberg, H. (2007). Scrum y xp desde las trincheras. C4Media Inc. InfoQ.

Palacio, J. (2014). Gestion de proyectos scrum manager. Scrum Manager.

Schwaber, K., Sutherland, J., and Beedle, M. (2013). The denitive guide to scrum: The rules of the game. Recuperado de: http://www. scrumguides. org/docs/scrumguide/v1/scrum-guideus.pdf.

Takeuchi, H. and Nonaka, I. (1986). The new new product development game. Harvard business review, 64(1):137{146.

Tovar F. Asociados Abogados Ltda., M.-A. (2012b). Manual de Gestión Nacional del Espectro Radioeléctrico, Título I, volumen 1 - Fundamentos, chapter 2 - Fundamentos de la Gestión Nacional del Espectro en Colombia. MinTic-ANE.

Tuckman, B. W. (1965). Developmental sequence in small groups. Psychological bulletin, 63(6):384.

Velarde, O., Murillo, M., Gómez, L., and Felcita, C. (2006). Introducción a la programación orientada a objetos. Pearson Education, 1 edition.