



**UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR  
(UNTELS)**

**APLICACIÓN DE LA METODOLOGÍA ÁGIL SCRUM EN  
EL DESARROLLO DE UN APLICATIVO PARA LA  
GESTIÓN DE OBSERVADORES ELECTORALES**

TRABAJO DE INVESTIGACIÓN PARA OPTAR EL TÍTULO DE  
INGENIERO DE SISTEMAS

PRESENTADO POR EL BACHILLER:  
**LILIANA LIZETH CUYA ESPINOZA**

LIMA – PERÚ

**2014**



### **Dedicatoria**

El presente trabajo es dedicado a toda mi familia en especial a mis padres y abuelos quienes durante toda mi vida me ofrecieron su cariño incondicional para que yo pueda lograr ser una gran persona y profesional.

### **Agradecimiento**

El agradecimiento a toda mi familia que durante la etapa universitaria me brindaron todo su cariño, apoyo y comprensión. Asimismo agradezco a mis docentes quienes se preocuparon por formar profesionales de

calidad y con valores dentro de nuestra universidad.

## Tabla de contenido

<b>INDICE DE TABLAS .....</b>	<b>6</b>
<b>INDICE DE GRÁFICOS .....</b>	<b>6</b>
<b>1. CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA.....</b>	<b>10</b>
1.1 DESCRIPCION DE LA REALIDAD PROBLEMÁTICA .....	10
1.2 JUSTIFICACION DEL PROBLEMA .....	12
1.3 DELIMITACION DE LA INVESTIGACIÓN .....	13
1.3.1 ESPACIAL.....	13
1.3.2 TEMPORAL.....	13
1.3.3 ALCANCE .....	14
1.4 FORMULACION DEL PROBLEMA.....	14
1.5 OBJETIVOS .....	15
1.5.1 OBJETIVO GENERAL: .....	15
1.5.2 OBJETIVOS ESPECIFICOS.....	15
<b>2. CAPÍTULO II: MARCO TEÓRICO .....</b>	<b>16</b>
2.1 ANTECEDENTES DE LA METODOLOGÍA SCRUM .....	17
2.2 BASES TEORICAS .....	18
2.2.1 HISTORIA DE LOS PROCESOS DE DESARROLLO.....	18
2.2.2 METODOLOGÍAS ÁGILES.....	21
2.2.3 DEL MANIFIESTO ÁGIL SUS VALORES Y PRINCIPIOS .....	22
2.3 MARCO CONCEPTUAL .....	28
2.3.1 SCRUM .....	28
2.3.2 CONSEJO NACIONAL ELECTORAL (CNE) DE ACUADOR.....	46
<b>3. CAPITULO III: DESCRIPCIÓN DEL MODELO.....</b>	<b>48</b>
3.1 ANALISIS DEL APLICATIVO.....	48
3.1.1 HERRAMIENTAS.....	48
3.1.2 SPRINT 0 .....	52
3.2 CONSTRUCCION.....	65
3.2.1 SPRINT 1 .....	66
3.2.2 SPRINT 2 .....	83
3.2.3 SPRINT 3 .....	90
3.3 REVISIÓN Y CONSOLIDACIÓN DE RESULTADOS.....	93

<b>4. CONCLUSIONES.....</b>	<b>99</b>
<b>5. RECOMENDACIONES .....</b>	<b>101</b>
<b>6. BIBLIOGRAFIA.....</b>	<b>103</b>
<b>7. ANEXOS.....</b>	<b>105</b>

## **INDICE DE TABLAS**

Tabla N° 1.- Plantilla de especificación de casos de uso.....	48
Tabla N° 2.- Requisitos Funcionales/No Funcionales Sprint 0.....	59
Tabla N° 3.- Equipo de trabajo.....	61
Tabla N° 4.- Product Backlog – Sprint 0.....	62
Tabla N° 5.- Actores Sprint 0.....	68
Tabla N° 6.- Scrum Backlog Sprint 1.....	79
Tabla N° 7.- Revisión Scrum Backlog Sprint 1.....	79
Tabla N° 8.- Actualización de Product Backlog Sprint 1.....	80
Tabla N° 9.- Sprint Backlog Sprint 2.....	83
Tabla N° 10.- Revisión del Sprint 2.....	86
Tabla N° 11.- Actualización Product Backlog Sprint 2.....	87
Tabla N° 12.- Sprint Backlog. Sprint 3.....	90

## **INDICE DE GRÁFICOS**

Figura N° 1.- Diagrama de Espina de Pescado.....	11
Figura N° 2.- Scrum.....	29
Figura N° 3.- Elementos de Scrum.....	45
Figura N° 4.- Ejemplo de Diagrama de Entidad Relación.....	59
Figura N° 5.- Vista del RedMine.....	50
Figura N° 6.- Herramienta de desarrollo Java.....	51
Figura N° 7.- Forma de Iteración o Sprint.....	66

Figura N° 8.- Diagrama de Casos de Uso.....	71
Figura N° 9.- Prototipo-Gestión Postulación Observador.....	72
Figura N° 10.- Prototipo-Gestión de Registro de Independiente.....	73
Figura N° 11.- Prototipo-Gestión Observador Invitados-Generación.....	73
Figura N° 12.- Prototipo-Gestión de Itinerarios de vuelo.....	74
Figura N° 13.- Prototipo-Gestión de Fichas Técnicas B.....	74
Figura N° 14.- Prototipo-Gestión de Fichas Técnicas A.....	75
Figura N° 15.- Prototipo-Gestión de informes grupales.....	75
Figura N° 16.- Prototipo-Generación de reportes.....	76
Figura N° 17.- Modelo de Base de Datos.....	77
Figura N° 18.- Arquitectura de la aplicación.....	78
Figura N°19.- Incremento en BD- Sprint 2.....	85
Figura N°20.- Incremento en BD- Sprint 3.....	92
Figura N°22.- Interfaz Proceso de Login.....	93
Figura N°23.- Gestión Postulación Observador .....	93
Figura N°24.- Gestión de Registro de Independiente .....	94
Figura N°25.- Gestión Observador Invitados-Generación.....	94
Figura N°26.- Gestión de Itinerarios de vuelo .....	94
Figura N°27.- Gestión de Fichas Técnicas B .....	95
Figura N°28.- Gestión de Fichas Técnicas A .....	96
Figura N°29.- Prototipo-Gestión de informes grupales.....	96
Figura N° 30.- Generación de reportes.....	97

## INTRODUCCIÓN

La tecnología permite realizar procedimientos que permitan abarcar campos tan delicados como los procesos electorales, estos procesos deben ser controlados para verificar que sean transparentes y cumplan con todos los controles que el gobierno necesita.

Un proceso electoral contiene sub procesos que se desarrollan para dar origen a la elección de representantes políticos, uno de estos procesos es el de gestión de observadores electorales, estos observadores son personas externas a la entidad pública que lleva a cabo el proceso de escrutinio, son quienes presencian todo este proceso y velan para que se desarrolle con total transparencia. La implementación de un sistema que permita el registro y el seguimiento de estas personas por las autoridades de la entidad correspondiente, facilita en gran magnitud el proceso electoral ya que de otra manera los procedimientos y trámites para realizar esta labor impedirían que se realice con tal agilidad como se hace con un sistema informático.

Para asegurar el éxito durante el desarrollo de software no es suficiente contar con notaciones de modelado y herramientas, hace falta un elemento importante: la metodología de desarrollo, la cual nos provee de una dirección a seguir para la correcta aplicación de los demás elementos.

El ciclo de vida de desarrollo de un sistema está basado en los pasos comúnmente conocidos como son: la etapa de planificación, donde se procede a conocer el negocio y todos los procesos que intervienen en el problema que se analiza sin necesidad de que estos sean o no automatizados; la etapa de análisis, donde se desarrolla el levantamiento de la información necesaria para gestar los requerimientos y funcionalidades del futuro sistema; la etapa de diseño, en esta fase es donde se describen las especificaciones que tendrá el



sistema para pasar a la etapa de desarrollo o construcción del software; la etapa de desarrollo, donde se construyen las líneas de códigos y las tablas de la base de datos basadas en los requerimientos y en el diseño realizado en el paso anterior; la etapa de pruebas, donde se pone a prueba el sistema desarrollado verificando que se haya cumplido con los requerimientos de diseño planteados, esta etapa retroalimenta a la anterior ya que si se comprueba que hay errores o inconsistencias se debe corregir en la etapa de desarrollo y volver a retornar a esta etapa de pruebas; y finalmente se encuentra la etapa de despliegue o puesta en producción, es aquí donde finalmente termina la construcción del software y comienza a ser usado por los usuarios finales para que soporte el proceso que se solicitó en la etapa de planificación y análisis.

Todas las implementaciones de software se basan en estas etapas, aunque no estrictamente las mismas ya que se pueden utilizar diferentes tipos de metodologías que dirijan dicha implementación. Para un entorno cambiante donde prima el software funcionando en vez de largos procesos de planeación y documentación, se utilizan las metodologías ágiles.

El caso de estudio del presente trabajo propone en base al tipo de organización, la aplicación de la metodología ágil SCRUM para desarrollar un aplicativo para la gestión de Observadores Electorales.

# **1. CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA**

## **1.1 DESCRIPCIÓN DE LA REALIDAD PROBLEMÁTICA**

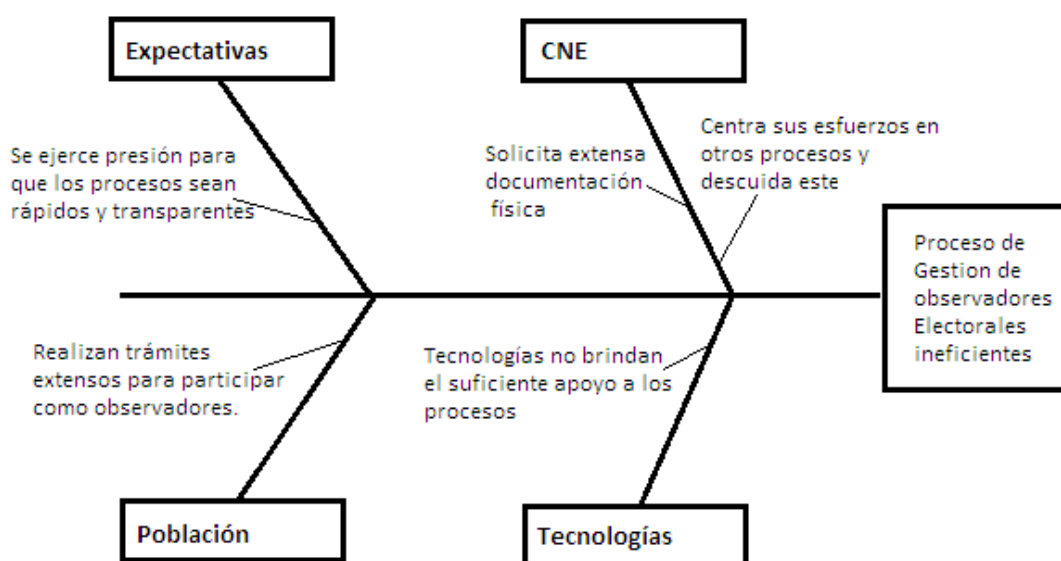
Para cada elección de personalidades políticas, se procede con el registro de observadores electorales, quienes deben registrarse a través de un procedimiento presencial y sumamente burocrático, ya que debe pasar por la revisión del personal encargado de la entidad pública que los convoca. Después del registro estos observadores proceden a hacer acto de presencia en las instalaciones de los locales de votación y finalmente remiten unos documentos donde concluyen todo lo que presenciaron, brindando recomendaciones a la entidad organizadora para mejorar en los siguientes procesos, ya sea en forma de encuestas o formatos de llenado obligatorio, etc. El proceso de registrar y dar seguimiento a la labor de estas personas que participan del proceso electoral de manera voluntaria, es bastante engorroso ya que son muchísimas personas las que participan y a las que hay que darles seguimiento para que sus observaciones puedan ser atendidas y tomadas en consideración.

Los procesos en donde interfiere mucha documentación de por medio, suelen avanzar de manera lenta y poco transparente, estos unos de los principales

problemas que se encuentra al no contar con una herramienta automatizada que realice tareas rutinarias de registro y liberando al personal encargado para poder realizar tareas de supervisión y control de las personalidades a las que se elige para participar del proceso electoral.

Para explicar los factores que justifican la ejecución del presente proyecto, se usará un Diagrama de Causa-Efecto (o Espina de pescado). En este diagrama se determina un problema central que viene a ser el efecto, en este caso definido por el proceso de Registro y Seguimiento de Observadores Electorales; para este efecto o consecuencia, se identifican los factores más importantes que causaron dicho hecho, estos factores están representados por las espinas, tal y como se muestra en la siguiente figura:

Figura N° 1.- Diagrama de Espina de Pescado



Fuente: Elaborado por la autora.

## 1.2 JUSTIFICACIÓN DEL PROBLEMA

Una vez analizado el proceso de Registro y Seguimiento de Observadores Electorales, se determina que el principal objetivo es agilizar los trámites para que no se realicen de manera presencial, y de esta manera reducir costos y tiempos generados en su ejecución.

Se determinó entonces que la solución a este problema, era la automatización de dicho proceso utilizando tecnologías de información que permitan agilizar y mantener un mejor monitoreo de los observadores electorales.

La ejecución de este proyecto permitirá al Consejo Nacional de Elecciones (CNE) del Ecuador contar con una solución que le permita optimizar recursos de tiempo y costos, en un tiempo relativamente corto ya que gracias a la metodología se podrá trabajar en base a un sistema funcionando para revisar si se ajusta a sus necesidades o debe ser modificado según lo requieran.

Los beneficios de la implementación de este módulo son diversos para el CNE ya que además mejora su nivel de credibilidad como institución pública al contar con procesos automatizados que otorgan beneficio a la población.

Para cumplir con la solución mencionada, se plantea el desarrollo de un módulo de Observadores Electorales que sume a las automatizaciones que se vienen desarrollando a nivel electoral para institución encargada (Consejo Nacional Electoral). Para el desarrollo del módulo de Observadores se plantea la utilización del método ágil SCRUM.

Los principales motivos que han llevado a la elección de Scrum son los siguientes:

- Es uno de los métodos de gestión de proyectos, de los denominados ágiles, que permite un manejo apropiado de las expectativas del cliente, basado en resultados tangibles.
- Dadas las condiciones de trabajo con el cliente (CNE), se considera favorable la utilización de esta metodología para lograr una comunicación efectiva entre las partes y poder cumplir con los tiempos de entrega ya que estos procesos no pueden retrasarse por ninguna razón.

Al implementar de manera correcta la metodología respetando los roles, y las actividades que ésta establece, el equipo del proyecto podrá trabajar ordenadamente en base a una dirección común, que permita obtener resultados favorables entregando un producto que esté acorde a lo que el usuario necesite ya que se desarrollara una comunicación efectiva con todos los interesados del proyecto, además se podrá reducir considerablemente las holguras de tiempo puesto que el equipo de desarrollo podrá trabajar en base a especificaciones claras que son constantemente actualizadas y/o verificadas.

### **1.3 DELIMITACIÓN DE LA INVESTIGACIÓN**

#### **1.3.1 ESPACIAL**

La aplicación del presente estudio se desarrollara en el marco de la utilización de la metodología Scrum en el desarrollo del módulo de Observadores Electorales para el Consejo Nacional de Elecciones de Ecuador por parte del equipo de desarrollo de la empresa ScytI Secure Electronic Voting Perú ubicada en Lima.

#### **1.3.2 TEMPORAL**

El desarrollo de esta aplicación debe desarrollarse en un periodo de dos meses.

### **1.3.3 ALCANCE**

En este contexto se establecen los aspectos que debe abarcar la ejecución del presente proyecto, así como también los alcances del módulo estudiado de Observadores Electorales, esto es:

- Estudio y aplicaciones del método ágil Scrum aplicado al desarrollo de un módulo de Observadores Electorales, que permita registrar y dar seguimiento a las personas que participen como observadores electorales en los procesos de escrutinio.
- Establecer Scrum como el marco de trabajo con el que la empresa Scytl desarrolle los demás módulos del Sistema Integrado de Administración Electoral del país de Ecuador, y que posteriormente pueda ser adoptada por los demás proyectos.
- Análisis y desarrollo del sistema informático comprendido por:
  - o Una aplicación web que contenga los siguientes sub módulos:
    - sub módulo de Observadores Independientes
    - sub módulo de Observadores Invitados
    - sub módulo de Reportes Resúmenes
    - sub módulo de Grupos de Observadores

El presente aplicativo web brindará todas las funcionalidades requeridas por el usuario, y optimizara el proceso tradicional que de observadores electorales.

Finalmente el módulo informático será desarrollado en una arquitectura web, que brinde escalabilidad y permita la implementación de nuevos requerimientos no definidos como alcance de este proyecto.

### **1.4 FORMULACIÓN DEL PROBLEMA**

El trabajo de investigación que se presenta, encuentra el siguiente problema:

¿En qué medida mejora el proceso de Gestión de Observadores Electorales tras la implementación de un aplicativo web utilizando la metodología ágil Scrum?

## **1.5 OBJETIVOS**

### **1.5.1 OBJETIVO GENERAL:**

Dado el problema antes mencionado, el objetivo del presente trabajo de investigación es el siguiente:

Desarrollar un aplicativo web que optimice el proceso de Gestión de Observadores Electorales, utilizando la metodología ágil Scrum.

### **1.5.2 OBJETIVOS ESPECIFICOS**

- Revisar y analizar conceptos relacionados a las metodologías de desarrollo de software.
- Analizar el método Scrum y su aporte a la ejecución del presente proyecto.
- Analizar el proceso de registro y seguimiento de Observadores Electorales del Consejo Nacional Electoral de Ecuador.
- Diseñar la arquitectura a emplear en el Sistema Informático para el registro y control de Observadores Electorales.
- Establecer Scrum como el marco de trabajo para el desarrollo de la aplicación propuesta así como de futuras implementación de funcionalidades o de otros módulos.

## 2. CAPÍTULO II: MARCO TEÓRICO

Este capítulo tiene como objetivo, brindar una descripción del marco teórico de referencia al método ágil de desarrollo de software Scrum.

En la ingeniería de software existen diversos tipos de metodologías, las denominadas pesadas y las denominadas ágiles.

Para muchos equipos de desarrollo el uso de metodologías tradicionales resulta muy lejano a su forma de trabajo actual y la realidad de su cartera de proyectos, considerando las dificultades asociadas a la inversión en términos de formación y costos de herramientas utilizadas. Por otro lado están las características de los proyectos para los cuales las metodologías ágiles han sido especialmente pensadas. Aquellos proyectos en los cuales los equipos de desarrollo son pequeños, de corto plazo, con requisitos volátiles y basados en nuevas tecnologías, como es el caso del equipo de desarrollo de este proyecto de la empresa Scytl.

Las metodologías ágiles están orientadas principalmente a proyectos que necesitan soluciones a medida, con una elevada simplificación en términos de tiempo y recursos, sin dejar de lado el aseguramiento de la calidad del producto. Estas metodologías se centran en el factor humano y el producto software, es



decir, le dan mayor importancia al equipo de desarrollo, a la colaboración del cliente al desarrollo incremental del software con iteraciones cortas en tiempo.

## **2.1 ANTECEDENTES DE LA METODOLOGÍA SCRUM**

En 1986 Hirotaka Takeuchi e Ikujiro Nonaka describieron una nueva aproximación holística que incrementa la rapidez y la flexibilidad en el desarrollo de nuevos productos comerciales. Takeuchi y Nonaka comparan esta nueva aproximación holística, en la cual las fases se traslapan de manera intensa y el proceso completo es realizado por un equipo con funciones transversales, como en el rugby, donde el equipo entero actúa como un solo hombre para intentar llegar al otro lado del campo, pasando el balón de uno a otro. Los casos de estudio provienen de las industrias automovilísticas, así como de fabricación de máquinas fotográficas, computadoras e impresoras.

En 1991 Peter DeGrace y Leslie Stahl en su libro *Wicked Problems, Righteous Solutions* (A problemas malvados, soluciones virtuosas), se refirieron a esta aproximación como scrum (melé en inglés), un término propio del rugby mencionado en el artículo por Takeuchi y Nonaka.

A principios de los años 1990 Ken Schwaber empleó una aproximación que lo llevó a poner en práctica el Scrum en su compañía *Advanced Development Methods*. Por aquel tiempo Jeff Sutherland desarrolló una aproximación similar en *Easel Corporation* y fue el primero en denominarla Scrum. En 1995 Schwaber y Sutherland, durante el OOPSLA '95 desarrollado en Austin, presentaron en paralelo una serie de artículos describiendo scrum, siendo ésta la

primera aparición pública de la metodología. Durante los años siguientes, Schwaber y Sutherland, colaboraron para consolidar los artículos antes mencionados, así como sus experiencias y el conjunto de mejores prácticas de la industria que conforman a lo que ahora se le conoce como Scrum.

En 2001, Schwaber y Mike Beedle describieron la metodología en el libro Agile Software Development with Scrum.

Desde 1995 miles de proyectos en todo el mundo han utilizado Scrum para el desarrollo de productos, tanto en empresas pequeñas, “startups” con tan sólo 5 personas desarrollando un producto, como en multinacionales.

En la actualidad, Scrum se está utilizando en diferentes tipos de negocio y, especialmente, en el desarrollo de software. La Scrum Alliance es la organización sin ánimo de lucro que se encarga de difundir Scrum en este ámbito.

## **2.2 BASES TEÓRICAS**

### **2.2.1 HISTORIA DE LOS PROCESOS DE DESARROLLO**

Uno de los grandes pasos dados en la industria del software fue aquel en que se plasmó el denominado modelo de cascada ya que sirvió como base para la formulación Metodologías Ágiles del análisis estructurado, el cual fue uno de los precursores en el camino hacia la aplicación de prácticas estandarizadas dentro de la ingeniería de software. Este modelo surge como respuesta al modelo codificar y probar que era el que predominaba en la década de los sesenta. En esta época ya existían modelos iterativos e incrementales pero no eran disciplinados ni estaban formalizados.

A consecuencia de esta realidad, la idea de tener un modelo que ordenara el proceso de desarrollo y que parecía bastante sencillo de llevar a la práctica hizo que el modelo en cascada tuviese gran promoción. Este modelo se basaba en el desarrollo en forma de cascada ya que se requería de la finalización de la etapa anterior para empezar la siguiente. Esto degeneraba en un “congelamiento” temprano de los requerimientos, los cuales al presentar cambios requerían gran esfuerzo en re-trabajo. Otra opción era no permitir cambio alguno a los

requerimientos una vez que se iniciara el desarrollo lo que traía aparejado que el usuario no veía la aplicación hasta que ya estaba construida y una vez que interactuaba no cubría sus necesidades.

Asimismo, dadas las características inherentes del modelo, la fase de implementación del mismo requería el desarrollo de los módulos en forma independiente con las correspondientes pruebas unitarias, y en la siguiente fase, se realizaba la integración de los mismos. Esto traía grandes inconvenientes debido a que todo estaba probado en forma unitaria sin interacción con los demás módulos. Las sorpresas llegaban cuando se integraban estas piezas para formar la aplicación; lo cual inevitablemente desembocaba en un retraso del proyecto, sacrificando la calidad del mismo.

De esta forma y en forma bastante temprana en algunos casos, fueron surgiendo diversos procesos denominados iterativos que proponían lidiar con la impredecibilidad del software (subsanaando muchas de las falencias del modelo en cascada) mitigando los riesgos en forma temprana. Los procesos iterativos de los cuales se desprenderían diversas instancias, como son el modelo iterativo e incremental, el modelo en espiral, el modelo basado en prototipo, el modelo SLCD, el MBASE, el RUP, etc.

Del modelo en espiral desarrollado por Barry Boehm [1] surgió una de las ideas fundamentales que las metodologías posteriores adoptarían: el temprano análisis de riesgos. El modelo en espiral, de carácter iterativo en sus primeras fases, plantea la necesidad de realizar al principio diversas iteraciones dirigidas a mitigar los riesgos más críticos relevados en el proyecto mediante la realización de prototipos o simulaciones de tipo desechables tendientes a probar algún concepto. Una vez que esos prototipos son validados se suceden iteraciones del tipo: determinar objetivos, evaluar, desarrollar, planear. Una vez que se tenía el diseño detallado y validado por el cliente, se implementaba el software siguiendo las etapas de un modelo en cascada. Esta es una falla importante del modelo ya que no se acomoda a la posibilidad de cambios una vez que se inicia la construcción. Todas las críticas que se le hacían al modelo en cascada se aplican a estas fases del modelo en espiral.

Fue el mismo Barry Boehm, quien en un artículo describe tres hitos críticos a ser utilizados en cualquier proyecto para poder planificar y controlar el progreso del mismo [2], dando visibilidad a los stakeholders. Estos hitos están relacionados con las etapas de avance que se van dando a lo largo de un proyecto de acuerdo a como ocurren las actividades de ingeniería (que componen los espirales del modelo en espiral) a las actividades de producción (que componen la construcción en cascada del software). Su impacto en la industria del software ha sido tan importante que uno de los procesos más utilizados en la actualidad, el RUP, los incorpora. Estos hitos son:

- Objetivos del Ciclo de Vida
- Arquitectura del Ciclo de Vida
- Capacidad de Operación Inicial

El primer hito finaliza con la definición del alcance del software a construir, la identificación de los stakeholders, y el delineamiento del plan de desarrollo del sistema. El mismo ocurre al final de la fase de Concepción según el RUP.

El segundo hito finaliza con el delineamiento de la arquitectura del sistema, la resolución de todos los riesgos críticos del proyecto, y el refinamiento de los objetivos y el alcance del sistema. A partir de este hito, se comienza la construcción en formamasiva del sistema, llegándose a utilizar el máximo de recursos en el proyecto. Asimismo, comienzan las fases más predecibles en cierta medida del desarrollo. El mismo corresponde al hito final de la fase de Elaboración según el RUP.

El último de los hitos corresponde a la entrega del primer release del software, que incorpora la funcionalidad definida en la correspondiente iteración. También se espera el tener material de entrenamiento, como un Manual del Usuario y Manual de Operaciones. Este hito se corresponde con el hito final de la fase de Construcción según el RUP. Lo que resultó interesante de estos hitos propuestos por Boehm es que los mismos son independientes del proceso de desarrollo elegido.

Como se ha mencionado, uno de los procesos con más influencia en la comunidad del software ha sido el RUP, el cual, es uno de los primeros procesos que es vendido como un producto. La idea de los creadores del RUP es que el mismo fuera un repositorio de todas las ideas vigentes y las denominadas buenas prácticas de la Ingeniería de Software. Sin embargo, al intentar abarcar proyectos de envergaduras tan dispares como podrían ser la construcción de un sistema de radares para portaviones versus la construcción de una registración de usuarios para una pequeña consultora, el RUP pierde la granularidad necesaria para describir en detalle uno de los factores más trascendentes de cualquier desarrollo de software: las personas.

### **2.2.2 METODOLOGÍAS ÁGILES**

A principios de la década del '90, surgió un enfoque que fue bastante revolucionario para su momento ya que iba en contra de toda creencia de que mediante procesos altamente definidos se iba a lograr obtener software en tiempo, costo y con la requerida calidad. El enfoque fue planteado por primera vez por Martiny se dio a conocer en la comunidad de Ingeniería de Software con el nombre de RAD o Rapid Application Development [3]. RAD consistía en un entorno de desarrollo altamente productivo, en el que participaban grupos pequeños de programadores utilizando herramientas que generaban código en forma automática tomando como entradas sintaxis de alto nivel. En general, se considera que Metodologías Ágiles este fue uno de los primeros hitos en pos de la agilidad en los procesos de desarrollo.

La historia de las Metodologías Ágiles y su apreciación como tales en la comunidad de la Ingeniería de Software tiene sus inicios en la creación de una de las metodologías utilizada como arquetipo: XP - eXtreme Programming, que nace de la mente de Kent Beck, tomando ideas recopiladas junto a Ward Cunningham.

Durante 1996, Beck es llamado por Chrysler como asesor del proyecto Chrysler Comprehensive Compensation payroll system. Dada la poca calidad del sistema que se estaba desarrollando, Beck decide tirar todo el código y empezar de cero utilizando las prácticas que él había ido definiendo a lo largo del tiempo. El sistema, que administra la liquidación de aproximadamente 10.000 empleados, y

consiste de 2.000 clases y 30.000 métodos, es puesto en operación en Mayo de 1997 casi respetando el calendario propuesto. Como consecuencia del éxito de dicho proyecto, Kent Beck dio origen a XP iniciando el movimiento de metodologías ágiles al que se anexarían otras metodologías surgidas mucho antes que el propio Beck fuera convocado por Chrysler.

Es así como que este tipo de metodologías fueron inicialmente llamadas “metodologías livianas”, sin embargo, aún no contaban con una aprobación pues se le consideraba por muchos desarrolladores como meramente intuitiva. Luego, con el pasar de los años, en febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace formalmente el término “ágil” aplicado al desarrollo de software. En esta misma reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software con el objetivo de esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

Tras esta reunión se creó The Agile Alliance, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida fue el Manifiesto Ágil, un documento que resume la filosofía “ágil”.

### **2.2.3 DEL MANIFIESTO ÁGIL SUS VALORES Y PRINCIPIOS**

#### **2.2.3.1 VALORES**

El manifiesto hace énfasis en cuatro valores principales que deben soportar el desarrollo de software:

- a. Los individuos e interacciones por encima de los procesos y las herramientas: para garantizar una mayor productividad, las metodologías ágiles valoran el recurso humano como el principal factor de éxito. Reconocen que contar con recurso humano calificado con capacidades técnicas adecuadas, facilidades para adaptarse al entorno, trabajar en equipo e interactuar convenientemente con el usuario, da mayor garantía de éxito que contar con herramientas y procesos rigurosos. Las metodologías ágiles reconocen que es más importante construir un buen equipo de trabajo que las herramientas y procesos. Procura primero conformar el equipo y que éste defina el entorno más conveniente de acuerdo con las necesidades y las circunstancias.
  
- b. Software funcionando por encima de la documentación: los profesionales relacionados con el desarrollo de software, aunque no es su fuerte producir documentos, reconocen su importancia, al igual que reconocen el tiempo y costo de mantener una documentación completa y actualizada. En este sentido, las metodologías ágiles respetan la importancia de la documentación como parte del proceso y del resultado de un proyecto de desarrollo de software, sin embargo, con la misma claridad hacen énfasis en que se deben producir los documentos estrictamente necesarios; los documentos deben ser cortos y limitarse a lo fundamental, dando prioridad al contenido sobre la forma de presentación. La documentación, en las metodologías ágiles procura mecanismos más dinámicos y menos costosos como son la comunicación personal, el trabajo en equipo, la auto-documentación y los estándares.
  
- c. La colaboración del cliente por encima de la negociación del contrato: clásicamente el usuario o cliente es quien solicita e indica qué debe hacer el software, y espera los resultados de acuerdo con sus exigencias o expectativas, en los plazos establecidos. Con frecuencia las dos partes, cliente y equipo de desarrollo, asumen posiciones distantes, con ingredientes de rivalidad y prevención al punto de tener que dedicar tiempo valioso a la tarea de redactar, depurar y firmar el contrato. En este sentido, y complementando el valor que se da al trabajo en equipo, las metodologías ágiles incluyen de manera directa y comprometida al cliente o usuario en el

equipo de trabajo. Es un ingrediente más en el camino al éxito en un proyecto de desarrollo de software. Más que un ambiente de enfrentamiento en el cual las partes buscan su beneficio propio, evadiendo responsabilidades y procurando minimizar sus riesgos, bajo la filosofía de las metodologías ágiles se busca el beneficio común, el del equipo de desarrollo y el del cliente. La participación del cliente debe ser constante, desde el comienzo hasta la culminación del proyecto, y su interacción con el equipo de desarrollo, de excelente calidad. Es el cliente quien sabe qué es lo que necesita o desea, el más indicado para corregir o hacer recomendaciones en cualquier momento del proyecto.

- d. La respuesta al cambio por encima del seguimiento de un plan: dada la naturaleza cambiante de la tecnología y la dinámica de la sociedad moderna, un proyecto de desarrollo de software se enfrenta con frecuencia a cambios durante su ejecución. Van desde ajustes sencillos en la personalización del software hasta cambios en las leyes, pasando por la aparición de nuevos productos en el mercado, comportamiento de la competencia, nuevas tendencias tecnológicas, etc. En este sentido, las metodologías pesadas con frecuencia caen en la idea de tener todo completo y correctamente definido desde el comienzo. No se cuenta entre sus fortalezas la habilidad para responder a los cambios. Por el contrario, en las metodologías ágiles la planificación no debe ser estricta, puesto que hay muchas variables en juego, debe ser flexible para poder adaptarse a los cambios que puedan surgir. Una buena estrategia es hacer planificaciones detalladas para unas pocas semanas y planificaciones mucho más abiertas para los siguientes meses.

### **2.2.3.2 PRINCIPIOS DEL MANIFIESTO ÁGIL**

Bajo el concepto de principio se hace referencia a las características que hacen la diferencia entre un proceso ágil y uno tradicional, y constituyen las ideas centrales del desarrollo ágil.

- I. Nuestra mayor prioridad es satisfacer al cliente mediante entregas tempranas y continuas de software con valor. Para que una metodología



puede ser calificada como ágil debe empezar a entregar software funcionando y útil en pocas semanas. Esto acaba con la incertidumbre, desconfianza, insatisfacción y desmotivación producidas en el cliente debido a las largas esperas para ver resultados concretos. Por lo tanto, la participación del cliente se hace más productiva en la medida en que el software está siendo probado, revisado y aprobado constantemente por quien lo requirió y lo va a usar.

- II. Bienvenidos los cambios a los requerimientos, incluso los tardíos. Los procesos ágiles aprovechan los cambios para la ventaja competitiva del cliente. Es ambicioso esperar que el cliente defina de manera definitiva todos sus requerimientos desde el comienzo y peor aún depender de ello para adelantar el proyecto. Los cambios en los requerimientos deben asumirse como parte del proceso de maduración del software, debe entenderse que cuando el cliente describe una necesidad lo hace desde su perspectiva de usuario y que sus conocimientos técnicos lo pueden limitar para hacerse entender completamente. Por lo tanto, las novedades en los requerimientos pueden ser ajenas a la voluntad del cliente. Esta forma de ver los cambios en los requerimientos induce al equipo de desarrollo a preferir los diseños flexibles, lo cual aumenta la satisfacción del cliente y redundando finalmente en beneficio del equipo de desarrollo dada la comodidad en el diagnóstico y ajustes que se requieren en la etapa de mantenimiento.
- III. Liberar frecuentemente software funcionando, desde un par de semanas a un par de meses, con preferencia por los periodos más cortos. El cliente siempre espera ver funcionando el programa, y es eso lo que debe entregársele. Pocas veces resulta conveniente, después de varios meses de trabajo, entregar sólo informes, modelos abstractos y planes. Se deben entregar resultados que incluyan software que el usuario pueda ver trabajando. Si hay una circunstancia que motiva al cliente es poder usar el software que solicitó.

- IV. Las personas del negocio y los desarrolladores deben trabajar juntos diariamente a lo largo del proyecto. Si bien el usuario desconoce lo referente al lenguaje, el diseño de bases de datos, protocolos y demás aspectos técnicos, es él, quien nos puede señalar qué está bien desde el punto de vista de la funcionalidad y resultados entregados por el software. La intervención oportuna del usuario puede resultar decisiva en el éxito de un proyecto y puede reducir el costo o el tiempo. Esta intervención puede ser en cualquier momento, por lo cual el usuario debe estar involucrado todo el tiempo que dure el proyecto.
  
- V. Construir proyectos en torno a individuos motivados. Darles el entorno y apoyo que necesiten, y confiar en ellos para que consigan hacer su trabajo. El ánimo, el sentido de pertenencia y la disposición del equipo de trabajo son fundamentales en un proyecto de software. Parte de la motivación está en la confianza que se muestre en el equipo de trabajo, el respeto por sus aportes y la comodidad que se les conceda en el momento de realizar su trabajo. Todo lo que se pueda hacer por dar ánimo y motivación a las personas participantes en el proyecto debe hacerse.
  
- VI. El método más efectivo y eficiente de compartir información a, y dentro de un equipo de desarrollo, es la conversación cara a cara. El trabajo en equipo debe apoyarse con un buen sistema de comunicación tanto entre los miembros del equipo de desarrollo como entre éstos y el usuario. La mejor forma de hacerlo es hablando personalmente; en la medida en que se evitan los intermediarios en el proceso de comunicación, como son el papel, el teléfono, el sistema de correo, y demás medios de comunicación, se incrementa la posibilidad de que el resultado sea el que se solicitó.
  
- VII. El software funcionando es la medida de progreso. Cuando se trata de establecer el estado de un proyecto, si bien existen diversas formas de medirlo, es la cantidad de requerimientos implementados y funcionando la que más claridad y confiabilidad ofrecen para establecer una medida del

avance del proyecto. Cualquiera otra que se presente será superada por una que involucre el software que ya ha sido probado y aprobado por el usuario.

- VIII. Los procesos ágiles promueven el desarrollo sostenible. Los patrocinadores, desarrolladores y usuarios deberían ser capaces de mantener relaciones cordiales. Se debe trabajar de forma que lo urgente no se imponga sobre lo importante. Desde el inicio del proyecto se debe asignar responsabilidades y tareas de manera que siempre se puedan cumplir.
- IX. La atención continua a la excelencia técnica y al buen diseño incrementan la agilidad. Además de satisfacer los requerimientos del usuario, los aspectos técnicos deben ser excelentes, independientemente de su cantidad y complejidad. La calidad debe ser vista desde dos perspectivas, la del usuario y la del equipo desarrollador. Para el personal técnico resulta evidente que cuanto más calidad tenga el software en cuanto a diseño y estándares de implementación, más rendimiento obtiene en las tareas de pruebas, mantenimiento, y mayor reusabilidad.
- X. La simplicidad –el arte de maximizar la cantidad de trabajo no hecho- es esencial. Se estima que el cliente nunca usará el 90% de la funcionalidad que se implementa sin que está haya sido solicitada. Se deben centrar los esfuerzos en lo que realmente importa, de manera simple, sin excederse en refinamientos y optimizaciones innecesarias. Si funciona así, déjelo así, si se va a perfeccionar u optimizar una rutina o programa se debe evaluar minuciosamente el costo beneficio.
- XI. Las mejores arquitecturas, requerimientos y diseños emergen de los equipos auto-organizados. Los principios que rijan en equipo de trabajo deben surgir de su interior, los ajustes, estructuras administrativas deben formularse con la participación de todo el equipo teniendo siempre presente el bien colectivo, la responsabilidad es de todos.

- XII. En intervalos regulares, el equipo reflexiona sobre cómo volverse más efectivo, entonces afina y ajusta su comportamiento como corresponde. El equipo de trabajo está todo el tiempo dispuesto a cambiar lo que sea necesario para mejorar. En cada tarea siempre existe la posibilidad de hacerlo mejor la próxima vez.

## **2.3 MARCO CONCEPTUAL**

### **2.3.1 SCRUM**

#### **2.3.1.1 DEFINICIÓN**

Scrum es un marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente.

Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que podamos mejorar.

El marco de trabajo Scrum consiste en los Equipos Scrum, roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso.

Las reglas de Scrum relacionan los eventos, roles y artefactos, gobernando las relaciones e interacciones entre ellos.

Scrum se basa en la teoría de control de procesos empírica. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo.

Tres pilares soportan toda la implementación del control de procesos empírico: transparencia, inspección y adaptación

- **Transparencia.**-Los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado. La transparencia requiere que dichos aspectos sean definidos por un estándar común, de tal modo que los observadores compartan un entendimiento común de lo que se está viendo.
- **Inspección.**- Los usuarios de Scrum deben inspeccionar frecuentemente los artefactos de Scrum y el progreso hacia un objetivo, para detectar variaciones. Su inspección no debe ser tan frecuente como para que interfiera en el trabajo. Las inspecciones son más beneficiosas cuando se realizan de forma diligente por inspectores expertos, en el mismo lugar de trabajo.
- **Adaptación.**- Si un inspector determina que uno o más aspectos de un proceso se desvían de límites aceptables, y que el producto resultante no será aceptable, el proceso o el material que está siendo procesado deben ser ajustados. Dicho ajuste debe realizarse cuanto antes para minimizar desviaciones mayores.

Scrum prescribe cuatro eventos formales, contenidos dentro del Sprint, para la inspección y adaptación, tal y como se describen en la sección Eventos de Scrum del presente documento.

- Reunión de Planificación del Sprint (Sprint Planning Meeting)
- Scrum Diario (Daily Scrum)
- Revisión del Sprint (Sprint Review)
- Retrospectiva del Sprint (Sprint Retrospective)

Figura N° 2.- Scrum



Fuente: Adaptado de Epidata Consulting

### 2.3.1.2 **EL EQUIPO SCRUM (SCRUM TEAM)**

El Equipo Scrum consiste en un Dueño de Producto (Product Owner), el Equipo de Desarrollo (Development Team) y un Scrum Master. Los Equipos Scrum son auto-organizados y multifuncionales. Los equipos auto-organizados eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas externas al equipo. Los equipos multifuncionales tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad.

Los Equipos Scrum entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener retroalimentación. Las entregas incrementales de producto “Terminado” aseguran que siempre estará disponible una versión potencialmente útil y funcional del producto.

#### **a. El Dueño de Producto (Product Owner)**

El Dueño de Producto es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. El cómo se lleva a cabo esto podría variar ampliamente entre distintas organizaciones, Equipos Scrum e individuos.

El Dueño de Producto es la única persona responsable de gestionar la Lista del Producto (Product Backlog). La gestión de la Lista del Producto incluye:

- Expresar claramente los elementos de la Lista del Producto;
- Ordenar los elementos en la Lista del Producto para alcanzar los objetivos y misiones de la mejor manera posible;

- Optimizar el valor del trabajo desempeñado por el Equipo de Desarrollo;
- Asegurar que la Lista del Producto es visible, transparente y clara para todos, y que muestra aquello en lo que el equipo trabajará a continuación; y,
- Asegurar que el Equipo de Desarrollo entiende los elementos de la Lista del Producto al nivel necesario.

El Dueño de Producto podría hacer el trabajo anterior, o delegarlo en el Equipo de Desarrollo. Sin embargo, en ambos casos el Dueño de Producto sigue siendo el responsable de dicho trabajo.

El Dueño de Producto es una única persona, no un comité. El Dueño de Producto podría representar los deseos de un comité en la Lista del Producto, pero aquellos que quieran cambiar la prioridad de un elemento de la Lista deben hacerlo a través del Dueño de Producto.

Para que el Dueño de Producto pueda hacer bien su trabajo, toda la organización debe respetar sus decisiones. Las decisiones del Dueño de Producto se reflejan en el contenido y en la priorización de la Lista del Producto. No está permitido que nadie pida al Equipo de Desarrollo que trabaje con base en un conjunto diferente de requerimientos, y el Equipo de Desarrollo no debe actuar con base en lo que diga cualquier otra persona.

#### **b. El Equipo de Desarrollo (Development Team)**

El Equipo de Desarrollo consiste en los profesionales que desempeñan el trabajo de entregar un Incremento de producto “Terminado”, que potencialmente se pueda poner en producción, al final de cada Sprint. Solo los miembros del Equipo de Desarrollo participan en la creación del Incremento.

Los Equipos de Desarrollo son estructurados y empoderados por la organización para organizar y gestionar su propio trabajo. La sinergia resultante optimiza la eficiencia y efectividad del Equipo de Desarrollo.

Los Equipos de Desarrollo tienen las siguientes características:

- Son auto-organizados. Nadie (ni siquiera el Scrum Master) indica al Equipo de Desarrollo cómo convertir elementos de la Lista del Producto en Incrementos de funcionalidad potencialmente desplegados;
- Los Equipos de Desarrollo son multifuncionales, contando como equipo con todas las habilidades necesarias para crear un Incremento de producto;
- Scrum no reconoce títulos para los miembros de un Equipo de Desarrollo, todos son Desarrolladores, independientemente del trabajo que realice cada persona; no hay excepciones a esta regla;
- Scrum no reconoce sub-equipos en los equipos de desarrollo, no importan los dominios particulares que requieran ser tenidos en cuenta, como pruebas o análisis de negocio; no hay excepciones a esta regla; y,
- Los Miembros individuales del Equipo de Desarrollo pueden tener habilidades especializadas y áreas en las que estén más enfocados, pero la responsabilidad recae en el Equipo de Desarrollo como un todo.

El tamaño óptimo del Equipo de Desarrollo es lo suficientemente pequeño como para permanecer ágil y lo suficientemente grande como para completar una cantidad de trabajo significativa. Tener menos de tres miembros en el Equipo de Desarrollo reduce la interacción y resulta en ganancias de productividad más pequeñas. Los Equipos de Desarrollo más pequeños podrían encontrar limitaciones en cuanto a las habilidades necesarias durante un Sprint, haciendo que el Equipo de Desarrollo no pudiese entregar un Incremento que potencialmente se pueda poner en producción. Tener más de nueve miembros en el equipo requiere demasiada coordinación. Los Equipos de Desarrollo grandes generan demasiada complejidad como para que pueda gestionarse mediante un proceso empírico. Los roles de Dueño de Producto y Scrum Master no cuentan en el cálculo del tamaño del equipo a menos que también estén contribuyendo a trabajar en la Lista de Pendientes de Sprint (Sprint Backlog).

### **c. El Scrum Master**



El Scrum Master es el responsable de asegurar que Scrum es entendido y adoptado. Los Scrum Masters hacen esto asegurándose de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum.

El Scrum Master es un líder que está al servicio del Equipo Scrum. El Scrum Master ayuda a las personas externas al Equipo Scrum a entender qué interacciones con el Equipo Scrum pueden ser de ayuda y cuáles no. El Scrum Master ayuda a todos a modificar estas interacciones para maximizar el valor creado por el Equipo Scrum.

El Scrum Master da servicio al Dueño de Producto de varias formas, incluyendo:

- Encontrar técnicas para gestionar la Lista de Producto de manera efectiva;
- Ayudar al Equipo Scrum a entender la necesidad de contar con elementos de Lista de Producto claros y concisos;
- Entender la planificación del producto en un entorno empírico;
- Asegurar que el Dueño de Producto conozca cómo ordenar la Lista de Producto para maximizar el valor;
- Entender y practicar la agilidad; y,
- Facilitar los eventos de Scrum según se requiera o necesite.

El Scrum Master da servicio al Equipo de Desarrollo de varias formas, incluyendo:

- Guiar al Equipo de Desarrollo en ser auto-organizado y multifuncional;
- Ayudar al Equipo de Desarrollo a crear productos de alto valor;
- Eliminar impedimentos para el progreso del Equipo de Desarrollo;
- Facilitar los eventos de Scrum según se requiera o necesite; y,
- Guiar al Equipo de Desarrollo en el entorno de organizaciones en las que Scrum aún no ha sido adoptado y entendido por completo.

### **2.3.1.3 EVENTOS DE SCRUM**

En Scrum existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum. Todos los eventos son bloques de tiempo (time-boxes), de tal modo que todos tienen una duración máxima. Una vez que comienza un Sprint, su duración, es fija y no puede acortarse o alargarse. Los demás eventos pueden terminar siempre que se alcance el objetivo del evento, asegurando que se emplee una cantidad apropiada de tiempo sin permitir desperdicio en el proceso.

Además del propio Sprint, que es un contenedor del resto de eventos, cada uno de los eventos de Scrum constituye una oportunidad formal para la inspección y adaptación de algún aspecto. Estos eventos están diseñados específicamente para habilitar las vitales transparencia e inspección. La falta de alguno de estos eventos da como resultado una reducción de la transparencia y constituye una oportunidad perdida para inspeccionar y adaptarse.

### **a. El Sprint**

El corazón de Scrum es el Sprint, es un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable. Es más conveniente si la duración de los Sprints es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint previo.

Los Sprints contienen y consisten de la Reunión de Planificación del Sprint (Sprint Planning Meeting), los Scrums Diarios (Daily Scrums), el trabajo de desarrollo, la Revisión del Sprint (Sprint Review), y la Retrospectiva del Sprint (Sprint Retrospective).

Durante el Sprint:

- No se realizan cambios que puedan afectar al Objetivo del Sprint (Sprint Goal);
- Los objetivos de calidad no disminuyen; y,
- El alcance puede ser clarificado y renegociado entre el Dueño de Producto y el Equipo de Desarrollo a medida que se va aprendiendo más.

Cada Sprint puede considerarse un proyecto con un horizonte no mayor de un mes. Al igual que los proyectos, los Sprints se usan para lograr algo. Cada Sprint tiene una definición de qué se va a construir, un diseño y un plan flexible que guiará la construcción y el trabajo y el producto resultante.

Los Sprints están limitados a un mes calendario. Cuando el horizonte de un Sprint es demasiado grande la definición de lo que se está construyendo podría cambiar, la complejidad podría elevarse y el riesgo podría aumentar.

Los Sprints habilitan la predictibilidad al asegurar la inspección y adaptación del progreso al menos en cada mes calendario. Los Sprints también limitan el riesgo al costo de un mes calendario.

Un Sprint puede ser cancelado antes de que el bloque de tiempo llegue a su fin. Solo el Dueño de Producto tiene la autoridad para cancelar el Sprint, aunque puede hacerlo bajo la influencia de los interesados, del Equipo de Desarrollo o del Scrum Master.

Un Sprint se cancelaría si el Objetivo del Sprint llega a quedar obsoleto. Esto podría ocurrir si la compañía cambia la dirección o si las condiciones del mercado o de la tecnología cambian. En general, un Sprint debería cancelarse si no tuviese sentido seguir con él dadas las circunstancias.

Cuando se cancela un Sprint, se revisan todos los Elementos de la Lista de Producto que se hayan completado y "Terminado". Si una parte del trabajo es potencialmente entregable, el Dueño de Producto normalmente lo acepta. Todos los Elementos de la Lista de Producto no completados se vuelven a estimar y se vuelven a introducir en la Lista de Producto. El trabajo finalizado en ellos pierde valor con rapidez y frecuentemente debe volverse a estimar.

Las cancelaciones de Sprint consumen recursos, ya que todos deben reagruparse en otra Reunión de Planificación de Sprint para empezar otro Sprint. Las cancelaciones de Sprint son a menudo traumáticas para el Equipo Scrum y son muy poco comunes.

## **b. Reunión de Planificación de Sprint (Sprint Planning Meeting)**

El trabajo a realizar durante el Sprint se planifica en la Reunión de Planificación de Sprint. Este plan se crea mediante el trabajo colaborativo del Equipo Scrum completo.

La Reunión de Planificación de Sprint tiene un máximo de duración de ocho horas para un Sprint de un mes. Para Sprints más cortos, el evento es usualmente más corto. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El Scrum Master enseña al Equipo Scrum a mantenerse dentro del bloque de tiempo.

La Reunión de Planificación de Sprint responde a las siguientes preguntas:

- ¿Qué puede entregarse en el Incremento resultante del Sprint que comienza?
- ¿Cómo se conseguirá hacer el trabajo necesario para entregar el Incremento?

Tema Uno: ¿Qué puede ser terminado en este Sprint?

El Equipo de Desarrollo trabaja para proyectar la funcionalidad que se desarrollará durante el Sprint. El Dueño de Producto discute el objetivo que el Sprint debería lograr y los Elementos de la Lista de Producto que, si se completan en el Sprint, lograrían el Objetivo del Sprint. El Equipo Scrum completo colabora en el entendimiento del trabajo del Sprint. La entrada a esta reunión está constituida por la Lista de Producto, el último Incremento de producto, la capacidad proyectada del Equipo de Desarrollo para el Sprint, y el rendimiento pasado del Equipo de Desarrollo. El número de elementos de la Lista de Producto seleccionados para el Sprint depende únicamente del Equipo de Desarrollo. Solo el Equipo de Desarrollo puede evaluar qué es capaz de lograr durante el Sprint que comienza.

Después de que el Equipo de Desarrollo proyecta qué elementos de la Lista de Producto entregará en el Sprint, el Equipo Scrum elabora un Objetivo del Sprint (Sprint Goal). El Objetivo del Sprint debería lograrse durante el Sprint a través de la implementación de la Lista de Producto, y provee una guía al equipo de desarrollo de por qué se está construyendo el incremento.

Tema Dos: ¿Cómo se conseguirá completar el trabajo seleccionado?

Una vez que se ha establecido el objetivo y seleccionado los elementos de la Lista de Producto para el Sprint, el Equipo de Desarrollo decide cómo construirá esta funcionalidad para formar un Incremento de producto “Terminado”. Los elementos de la Lista de Producto seleccionados para este Sprint, más el plan para terminarlos, recibe el nombre de Lista de Pendientes del Sprint (Sprint Backlog).

El Equipo de Desarrollo por lo general comienza diseñando el sistema y el trabajo necesario para convertir la Lista de Producto en un Incremento de producto funcional. El trabajo podría ser de tamaño o esfuerzo estimado variables. Sin embargo, durante la Reunión de Planificación del Sprint, se planifica suficiente trabajo como para que el Equipo de Desarrollo pueda hacer una proyección de lo que cree que puede completar en el Sprint que comienza. Para el final de esta reunión, el trabajo planificado por el Equipo de Desarrollo para los primeros días del Sprint es descompuesto en unidades de un día o menos. El Equipo de desarrollo se auto-organiza para asumir el trabajo de la Lista de Pendientes de Sprint, tanto durante la reunión de Planificación de Sprint como a lo largo del Sprint.

El Dueño de Producto puede ayudar a clarificar los elementos de la Lista de Producto seleccionados y hacer concesiones. Si el Equipo de Desarrollo determina que tiene demasiado trabajo o que no tiene suficiente trabajo, podría renegociar los elementos de la Lista de Producto seleccionados con el Dueño de Producto. El Equipo de Desarrollo podría también invitar a otras personas a que asistan con el fin de que proporcionen asesoría técnica o relacionada con el dominio.

Al finalizar la Reunión de Planificación de Sprint, el Equipo de Desarrollo debería ser capaz de explicar al Dueño de Producto y al Scrum Master cómo pretende trabajar como un equipo auto-organizado para lograr el Objetivo del Sprint y crear el Incremento esperado.

### **c. *Objetivo del Sprint (Sprint Goal)***

El Objetivo del Sprint es una meta establecida para el Sprint que puede ser alcanzada mediante la implementación de la Lista de Producto. Proporciona una guía al Equipo de Desarrollo acerca de por qué está construyendo el incremento. Es creado durante la reunión de Planificación del Sprint. El objetivo del Sprint ofrece al equipo de desarrollo cierta flexibilidad con respecto a la funcionalidad implementada en el Sprint. Los elementos de la Lista del Producto seleccionados ofrecen una función coherente, que puede ser el objetivo del Sprint. El objetivo del Sprint puede representar otro nexo de unión que haga que el Equipo de Desarrollo trabaje en conjunto y no en iniciativas separadas.

A medida que el equipo de desarrollo trabaja, se mantiene el objetivo del Sprint en mente. Con el fin de satisfacer el objetivo del Sprint se implementa la funcionalidad y la tecnología. Si el trabajo resulta ser diferente de lo que el Equipo de Desarrollo espera, ellos colaboran con el Dueño del Producto para negociar el alcance de la Lista de pendientes del Sprint (Sprint Backlog).

#### **d. Scrum Diario (Daily Scrum)**

El Scrum Diario es una reunión con un bloque de tiempo de 15 minutos para que el Equipo de Desarrollo sincronice sus actividades y cree 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 a la misma hora y en el mismo lugar todos los días para reducir la complejidad. Durante la reunión, cada miembro del Equipo de Desarrollo explica:

- ¿Qué hice ayer que ayudó al Equipo de Desarrollo a lograr el Objetivo del Sprint?
- ¿Qué haré hoy para ayudar al Equipo de Desarrollo a lograr el Objetivo del Sprint?
- ¿Veo algún impedimento que evite que el Equipo de Desarrollo o yo logremos el Objetivo del Sprint?

El Equipo de Desarrollo usa el Scrum Diario para evaluar el progreso hacia el Objetivo del Sprint y para evaluar qué tendencia sigue este progreso hacia la finalización del trabajo contenido en la Lista del Sprint. El Scrum Diario optimiza las posibilidades de que el Equipo de Desarrollo cumpla el Objetivo del Sprint. Cada día, el Equipo de Desarrollo debería entender cómo intenta trabajar en conjunto como un equipo auto-organizado para lograr el Objetivo del Sprint y crear el Incremento esperado hacia el final del Sprint. El Equipo de Desarrollo o los miembros del equipo a menudo se vuelven a reunir inmediatamente después del Scrum Diario, para tener discusiones detalladas, o para adaptar, o re planificar el resto del trabajo del Sprint .

El Scrum Master se asegura de que el Equipo de Desarrollo tenga la reunión, pero el Equipo de Desarrollo es el responsable de dirigir el Scrum Diario. El Scrum Master enseña al Equipo de Desarrollo para que mantenga el Scrum Diario en los límites del bloque de tiempo de 15 minutos. El Scrum Master se asegura de que se cumpla la regla de que solo los miembros del Equipo de Desarrollo participan en el Scrum Diario. Los Scrum Diarios mejoran la comunicación, eliminan la necesidad de mantener otras reuniones, identifican y eliminan impedimentos relativos al desarrollo, resaltan y promueven la toma de decisiones rápida, y mejoran el nivel de conocimiento del Equipo de Desarrollo. El Scrum Diario constituye una reunión clave de inspección y adaptación.

#### **e. Revisión de Sprint (Sprint Review)**

Al final del Sprint se lleva a cabo una Revisión de Sprint para inspeccionar el Incremento y adaptar la Lista de Producto si fuese necesario. Durante la Revisión de Sprint, el Equipo Scrum y los interesados colaboran acerca de lo que se hizo durante el Sprint. Basándose en esto, y en cualquier cambio a la Lista de Producto durante el Sprint, los asistentes colaboran para determinar las siguientes cosas que podrían hacerse para optimizar el valor. Se trata de una reunión informal, no una reunión de seguimiento, y la presentación del Incremento tiene como objetivo facilitar la retroalimentación de información y fomentar la

colaboración. Se trata de una reunión restringida a un bloque de tiempo de cuatro horas para Sprints de un mes. Para Sprints más cortos, se reserva un tiempo proporcionalmente menor. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El Scrum Master enseña a todos a mantener el evento dentro del bloque de tiempo fijado.

La Revisión de Sprint incluye los siguientes elementos:

- Los asistentes son el Equipo Scrum y los interesados clave invitados por el Dueño de Producto;
- El Dueño de Producto explica qué elementos de la Lista de Producto se han “Terminado” y cuáles no se han “Terminado”;
- El Equipo de Desarrollo habla acerca de qué fue bien durante el Sprint, qué problemas aparecieron y cómo fueron resueltos esos problemas;
- El Equipo de Desarrollo demuestra el trabajo que ha “Terminado” y responde preguntas acerca del Incremento;
- El Dueño de Producto habla acerca de la Lista de Producto en el estado actual. Proyecta fechas de finalización probables en el tiempo basándose en el progreso obtenido hasta la fecha (si es necesario);
- El grupo completo colabora acerca de qué hacer a continuación, de modo que la Revisión del Sprint proporcione información de entrada valiosa para Reuniones de Planificación de Sprints subsiguientes.
- Revisión de cómo el mercado o el uso potencial del producto podría haber cambiado lo que es de más valor para hacer a continuación; y,
- Revisión de la línea de tiempo, presupuesto, capacidades potenciales y mercado para la próxima entrega prevista del producto.

El resultado de la Revisión de Sprint es una Lista de Producto revisada, que define los elementos de la Lista de Producto posibles para el siguiente Sprint. Es posible además que la Lista de Producto reciba un ajuste general para enfocarse en nuevas oportunidades.

***f. Retrospectiva de Sprint (Sprint Retrospective)***



La Retrospectiva de Sprint es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo y crear un plan de mejoras que sean abordadas durante el siguiente Sprint.

La Retrospectiva de Sprint tiene lugar después de la Revisión de Sprint y antes de la siguiente Reunión de Planificación de Sprint. Se trata de una reunión restringida a un bloque de tiempo de tres horas para Sprints de un mes. Para Sprints más cortos se reserva un tiempo proporcionalmente menor. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El Scrum Master enseña a todos a mantener el evento dentro del bloque de tiempo fijado. El Scrum Master participa en la reunión como un miembro del equipo ya que la responsabilidad del proceso Scrum recae sobre él. El propósito de la Retrospectiva de Sprint es:

- Inspeccionar cómo 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; y,
- Crear un plan para implementar las mejoras a la forma en la que el Equipo Scrum desempeña su trabajo.

El Scrum Master alienta al equipo para que mejore, dentro del marco de proceso Scrum, su proceso de desarrollo y sus prácticas para hacerlos más efectivos y amenos para el siguiente Sprint. Durante cada Retrospectiva de Sprint, el Equipo Scrum planifica formas de aumentar la calidad del producto mediante la adaptación de la Definición de “Terminado” (Definition of “Done”) según sea conveniente.

Para el final de la Retrospectiva de Sprint, el Equipo Scrum debería haber identificado mejoras que implementará en el próximo Sprint. El hecho de implementar estas mejoras en el siguiente Sprint, constituye la adaptación subsecuente a la inspección del Equipo de Desarrollo a sí mismo. Aunque las mejoras pueden implementarse en cualquier momento, la Retrospectiva de Sprint

ofrece un evento dedicado para este fin, enfocado en la inspección y la adaptación.

#### 2.3.1.4 **ARTEFACTOS DE SCRUM**

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. Lista de Producto (Product Backlog)**

La Lista de Producto es una lista ordenada de todo lo que podría ser necesario en el producto, y es la única fuente de requisitos para cualquier cambio a realizarse en el producto. El Dueño de Producto (Product Owner) es el responsable de la Lista de Producto, incluyendo su contenido, disponibilidad y ordenación.

Una Lista de Producto nunca está completa. El desarrollo más temprano de la misma solo refleja los requisitos conocidos y mejor entendidos al principio. La Lista de Producto evoluciona a medida de que el producto y el entorno en el que se usará también lo hacen. La Lista de Producto es dinámica; cambia constantemente para identificar lo que el producto necesita para ser adecuado, competitivo y útil. Mientras el producto exista, su Lista de Producto también existe. La Lista de Producto enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que constituyen cambios a ser hechos sobre el producto para entregas futuras. Los elementos de la Lista de Producto tienen como atributos la descripción, la ordenación, la estimación y el valor.

A medida que un producto es utilizado y se incrementa su valor, y el mercado proporciona retroalimentación, la Lista de Producto se convierte en una lista más larga y exhaustiva. Los requisitos nunca dejan de cambiar, así que la Lista de Producto es un artefacto vivo. Los cambios en los requisitos de negocio, las

condiciones del mercado o la tecnología podrían causar cambios en la Lista de Producto.

A menudo, varios Equipos Scrum trabajan juntos en el mismo producto. Para describir el trabajo a realizar sobre el producto, se utiliza una única Lista de Producto. En ese caso podría emplearse un atributo de la Lista de Producto para agrupar los elementos. El refinamiento (refinement) de la Lista de Producto es el acto de añadir detalle, estimaciones y orden a los elementos de la Lista de Producto. Se trata de un proceso continuo, en el cual el Dueño de Producto y el Equipo de Desarrollo colaboran acerca de los detalles de los elementos de la Lista de Producto. Durante el refinamiento de la Lista de Producto, se examinan y revisan sus elementos. El Equipo Scrum decide cómo y cuándo se hace el refinamiento. Este usualmente consume no más del 10% de la capacidad del Equipo de Desarrollo. Sin embargo, los elementos de la Lista de Producto pueden actualizarse en cualquier momento por el Dueño de Producto o a criterio suyo.

Los elementos de la Lista de Producto de orden más alto son generalmente más claros y detallados que los de menor orden. Se realizan estimaciones más precisas basándose en la mayor claridad y detalle; cuanto más bajo es el orden, menor es el detalle. Los elementos de la Lista de Producto de los que se ocupará el Equipo de Desarrollo en el siguiente Sprint tienen una granularidad mayor, habiendo sido descompuestos de forma que cualquier elemento puede ser “Terminado” dentro de los límites del bloque de tiempo del Sprint. Los elementos de la Lista de Producto que pueden ser “Terminados” por el Equipo de Desarrollo en un Sprint son considerados “preparados” o “accionables” para ser seleccionados en una reunión de Planificación de Sprint. Los elementos de la Lista de Producto normalmente adquieren este grado de transparencia mediante las actividades de refinamiento descritas anteriormente.

El Equipo de Desarrollo es el responsable de proporcionar todas las estimaciones. El Dueño de Producto podría influenciar al Equipo ayudándoles a entender y seleccionar soluciones de compromiso, pero las personas que harán el trabajo son las que hacen la estimación final.

### **b. Lista de Pendientes del Sprint (Sprint Backlog)**

La Lista de Pendientes del Sprint es el conjunto de elementos de la Lista de Producto seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint. La Lista de Pendientes del Sprint es una predicción hecha por el Equipo de Desarrollo acerca de qué funcionalidad formará parte del próximo Incremento y del trabajo necesario para entregar esa funcionalidad en un Incremento “Terminado”.

La Lista de Pendientes del Sprint hace visible todo el trabajo que el Equipo de Desarrollo identifica como necesario para alcanzar el Objetivo del Sprint.

La Lista de Pendientes del Sprint es un plan con un nivel de detalle suficiente como para que los cambios en el progreso se puedan entender en el Scrum Diario. El Equipo de Desarrollo modifica la Lista de Pendientes del Sprint durante el Sprint y esta Lista de Pendientes del Sprint emerge a lo largo del Sprint. Esto ocurre a medida que el Equipo de Desarrollo trabaja sobre el plan y aprende más acerca del trabajo necesario para conseguir el Objetivo del Sprint.

Según se requiere nuevo trabajo, el Equipo de Desarrollo lo añade a la Lista de Pendientes del Sprint. A medida que el trabajo se ejecuta o se completa, se va actualizando la estimación de trabajo restante. Cuando algún elemento del plan pasa a ser considerado innecesario, es eliminado. Solo el Equipo de Desarrollo puede cambiar su Lista de Pendientes del Sprint durante un Sprint. La Lista de Pendientes del Sprint es una imagen visible en tiempo real del trabajo que el Equipo de Desarrollo planea llevar a cabo durante el Sprint, y pertenece únicamente al Equipo de Desarrollo.

### **c. Seguimiento del Progreso del Sprint (Sprint Review)**

En cualquier momento durante un Sprint, es posible sumar el trabajo restante total en los elementos de la Lista de Pendientes del Sprint. El Equipo de Desarrollo hace seguimiento de este trabajo restante total al menos en cada Scrum Diario para proyectar la posibilidad de conseguir el Objetivo del Sprint. Haciendo

seguimiento del trabajo restante a lo largo del Sprint, el Equipo de Desarrollo puede gestionar su progreso.

#### 2.3.1.5 **DEFINICIÓN DE “TERMINADO” (DEFINITION OF “DONE”)**

Cuando un elemento de la Lista de Producto o un Incremento se describe como “Terminado”, todo el mundo debe entender lo que significa “Terminado”. Aunque esto varía significativamente para cada Equipo Scrum, los miembros del Equipo deben tener un entendimiento compartido de lo que significa que el trabajo esté completado, para asegurar la transparencia.

Esta es la definición de “Terminado” para el Equipo Scrum y se utiliza para evaluar cuándo se ha completado el trabajo sobre el Incremento de producto.

Esta misma definición guía al Equipo de Desarrollo en saber cuántos elementos de la Lista de Producto puede seleccionar durante una reunión de Planificación de Sprint. El propósito de cada Sprint es entregar Incrementos de funcionalidad que potencialmente se puedan poner en producción, y que se ajustan a la Definición de “Terminado” actual del Equipo Scrum.

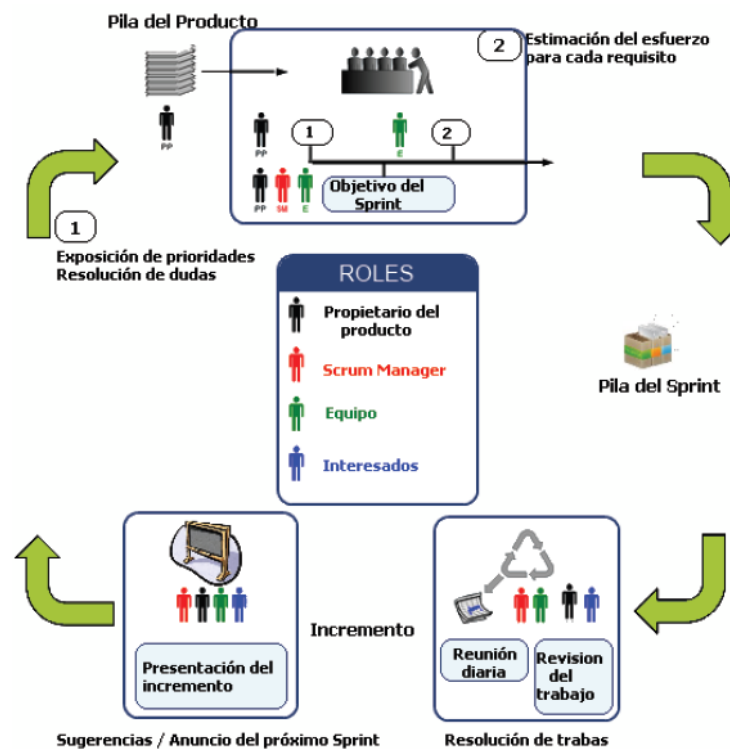
Los Equipos de Desarrollo entregan un Incremento de funcionalidad de producto en cada Sprint. Este Incremento es utilizable, de modo que el Dueño de Producto podría elegir liberarlo inmediatamente. Si la definición de “Terminado” para un incremento es parte de las convenciones, estándares o guías de la organización de desarrollo, al menos todos los Equipos Scrum deben seguirla. Si “Terminado” para un incremento no es una convención de la organización de desarrollo, el Equipo de Desarrollo del Equipo Scrum debe definir una definición de “Terminado” apropiada para el producto. Si hay múltiples Equipos Scrum trabajando en la entrega del sistema o producto, los equipos de desarrolladores en todos los Equipos Scrum deben definir en conjunto la definición de “Terminado”.

Cada Incremento se integra con todos los Incrementos anteriores y es probado exhaustivamente, asegurando que todos los Incrementos funcionan en conjunto.

A medida que los Equipos Scrum maduran, se espera que su definición de “Terminado” se amplíe para incluir criterios más rigurosos para una mayor calidad.

Cualquier producto o sistema debería tener una definición de “Terminado” que es un estándar para cualquier trabajo realizado sobre él.

Figura N° 3.- Elementos de Scrum



Fuente: Adaptado de Gestar.

### 2.3.2 CONSEJO NACIONAL ELECTORAL (CNE) DE ACUADOR

El Consejo Nacional Electoral (CNE) de la República del Ecuador es el máximo organismo de sufragio en dicho país. Tiene su sede en la ciudad de Quito, está constituido por 5 vocales elegidos mediante un concurso de méritos y oposición organizado por el Consejo de Participación Ciudadana y Control Social para un período de 6 años debiendo renovarse una mitad cada 3 años, de entre los

vocales se elige a un Presidente y un Vicepresidente; el CNE goza de completa autonomía financiera y administrativa.

Sus funciones son organizar, controlar las elecciones, puede castigar a partidos y candidatos que infrinjan las normas electorales; y tiene que inscribir y fiscalizar a los partidos y movimientos políticos.

El Consejo Nacional Electoral tiene 24 delegaciones Provinciales en cada una de las provincias para desconcentrar los servicios electorales en todo el país.

Este organismo conforma, junto al Tribunal Contencioso Electoral, la *Función Electoral* el cual es uno de las cinco funciones del estado.

## **3. CAPÍTULO III: DESCRIPCIÓN DEL MODELO**

En este capítulo se describe la adaptación del modelo Scrum en los procesos de ejecución del proyecto que tiene como resultado un sistema funcionando, en este caso un aplicativo para la gestión de Observadores. A continuación se procede a analizar los cuatro Sprints que se manejaron en el desarrollo del aplicativo, cada Sprint contiene sus procedimientos correspondientes; se comienza detallando la adaptación de la metodología con el Sprint 0.

### **3.1 ANÁLISIS**

Se describe la fase inicial del aplicativo utilizando la metodología Scrum, esta metodología comprende esta fase en el Sprint 0. Antes de iniciar describimos las herramientas utilizadas durante todas las iteraciones.

#### **3.1.1 HERRAMIENTAS**

##### **3.1.1.1 TÉCNICAS DE LEVANTAMIENTO DE INFORMACIÓN**

Para la ejecución del presente proyecto se utilizara la observación, las entrevistas con el cliente y los usuarios como principal técnica de relevamiento, a través de los medios que sean posibles.



- **CASOS DE USO**

Los casos de uso ayudan a describir qué es lo que el sistema debe hacer desde el punto de vista del usuario. Por lo tanto se considera que los casos de uso proporcionan un modo claro y preciso de comunicación con el cliente. Los diagramas de casos de uso describen las relaciones y las dependencias entre un grupo de casos de uso y los actores participantes en el proceso.

Una vez realizados los diagramas de casos de uso, que describen gráficamente lo que debe hacer el sistema, se detallan los casos de uso en donde se explica la forma de interactuar entre el sistema y el usuario. Los casos de uso son utilizados en la etapa de análisis.

Para el desarrollo de los casos de uso del presente trabajo, se utilizó el siguiente formato de especificaciones de casos de uso:

Tabla N° 1.- Plantilla de especificación de casos de uso.

Nombre	Nombre del caso de uso
ID	Identificador del caso de uso en el diagrama
Descripción	Breve descripción del caso de uso
Precondición	Condiciones que deben presentarse antes de llegar al caso de uso
Post condición	Condiciones generadas posterior al caso de uso
Flujo Normal	Descripción del flujo del caso de uso
Excepciones	Excepciones que alteren el flujo normal del caso de uso
Notas	Datos adicionales

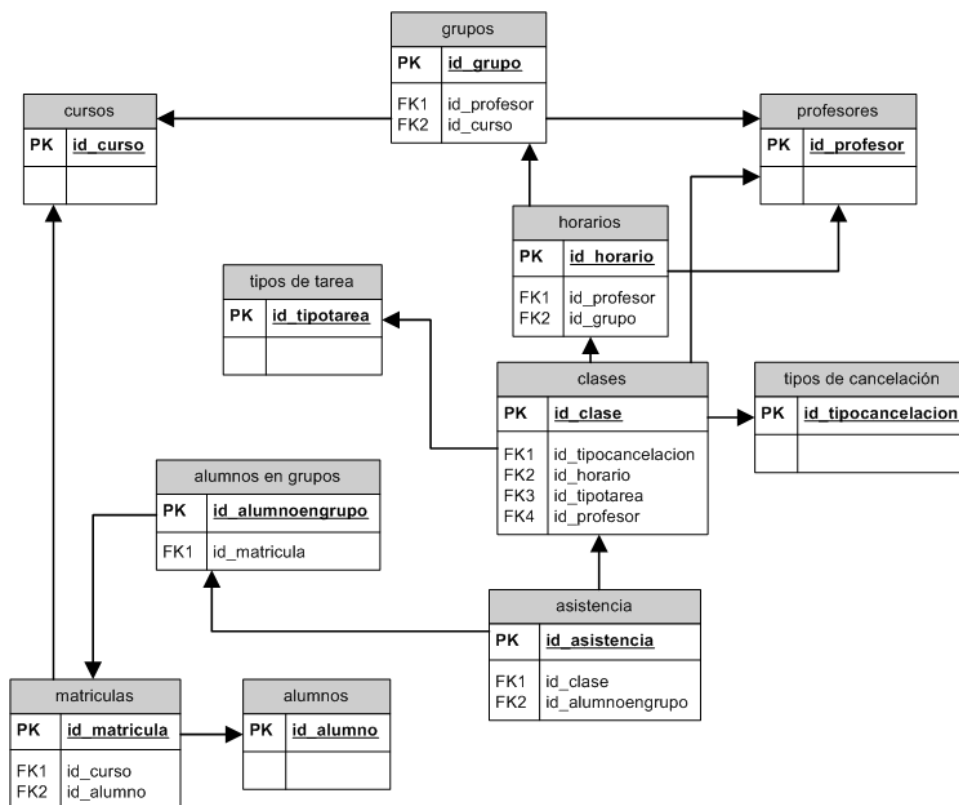
Fuente: ScytI Perú.

- **DIAGRAMA DE ENTIDAD RELACIÓN**

Un modelo de datos describe de forma abstracta como se representan los datos sean en una empresa, en un sistema de información o en un sistema de base de datos.

Como herramienta para el modelado de datos de un sistema de información, los DER expresan entidades relevantes y sus inter-relaciones. Formalmente son un lenguaje gráfico para describir conceptos y describen la información utilizada en un sistema de información.

Figura N° 4.- Ejemplo de Diagrama de Entidad Relación



Fuente: Ender Consulting.

- **REDMINE**

Es un sistema multi-plataforma, programado con Ruby on Rails, open source con licencia GPL, es una herramienta para llevar a cabo el seguimiento del proyecto. Esta herramienta fácil de utilizar, es una aplicación web que permite alojar en la nube de los datos referentes al proyecto y compartir la información del proyecto entre todo el equipo.

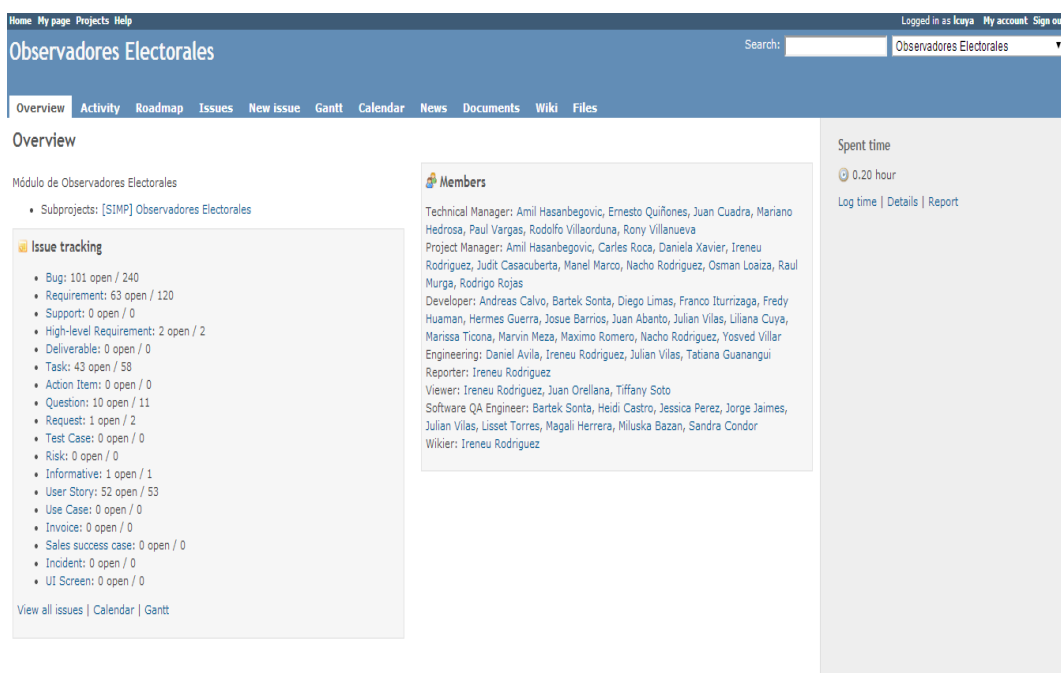
Esta herramienta permite:

- Manejar proyectos manejándolo a través de casos de uso e historias de usuario.

- Actualizar continuamente los avances realizados en las tareas asignadas.
- Reportar bugs durante el proceso de pruebas, y llevar el historial de solución del mismo.
- Crear Task y asignarlos a algún miembro del equipo.
- Sirve para compartir documentación importante entre todos los miembros del equipo.
- Permite asociar a otros miembros interesados del producto para que realicen seguimientos de las tareas o los avances del proyecto.

Esta herramienta servirá para registrar todos los avances de las historias de usuario e ir revisando el avance del proyecto.

Figura N° 5.- Vista del RedMine



Fuente: Redmine01.scytll.net

- **JAVA**

Es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como *WORA*, o "write once,

*run anywhere*"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web.

**Spring.**-Es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java. Proporciona varios módulos los cuales abarcan la mayor parte de las cosas que se debe hacer en cualquiera de las capas de nuestras aplicaciones, desde plantillas para trabajar con JDBC o invocación de Web Services y JMS, pasando por sus propias soluciones, ORM o MVC (web), hasta integración con otros frameworks, como Struts 2, Hibernate, JSF, etc. Todo esto de una forma elegante y haciendo uso de muchos buenos principios de programación.

Spring maneja la infraestructura de la aplicación, por lo que en la programación solo se debemos centrar la atención en la lógica de la misma (y de la configuración de Spring).

La utilización de Spring de forma correcta (lo cual no es difícil) nuestra aplicación quedará dividida en capas bien delimitadas, y con buenas prácticas de programación

El presente proyecto, se trabajó con este framework, en combinación con MVN y Hibernate. Esta herramienta es utilizada en la etapa de construcción.

Figura N° 6.- Herramienta de desarrollo Java



Fuente: [www.java.com](http://www.java.com)

### 3.1.2 SPRINT 0

Para comenzar con la ejecución del proyecto de creación del aplicativo web, se ha tenido que analizar las necesidades del usuario e identificar las funcionalidades que debe presentar el sistema de manera que se pueda desarrollar el Product Backlog que servirá de punto de partida para los demás Sprints.

Este Sprint consiste en concretar las reuniones que el Product Owner mantuvo con el usuario durante reuniones presenciales realizadas en la organización dueña del negocio, es decir en Consejo Nacional Electoral de Ecuador.

En este Sprint, es necesario definir ciertas actividades, una de ellas es el análisis del negocio para luego definir el alcance del proyecto y realizar una estimación de tiempo <sup>(1)</sup>y recursos a utilizar.

### 3.1.2.1 **ANÁLISIS DEL NEGOCIO**

En este punto se analizan las necesidades específicas del usuario a nivel funcional, para implementarlo en requerimientos del aplicativo. Para definir los alcances funcionales, se analizan los procesos realizados internamente por el CNE.

A continuación detallamos el proceso de Gestión de Observadores Electorales sin relacionarse con un sistema informático específicamente para ello.

El macro proceso de observadores electorales consiste en invitar a todas las personas naturales o jurídicas sean nacionales o extranjeras a participar dentro del proceso de escrutinio a través de su postulación ante el Consejo Nacional Electoral (CNE). Las personas interesadas en participar se registran adjuntando toda la documentación que el CNE solicita; esta documentación es evaluada y en base a ello se decide si permiten o no que el postulante forme parte de los observadores para un proceso electoral dado. Cuando una persona se ha acreditado formalmente como observador electoral está en su derecho de presentar informes ante el CNE brindando sus aportes y/o reclamos con respecto a sus experiencias u observaciones del proceso electoral. Su labor fundamental es verificar que todo se

---

- (1) El Product Owner estima un tiempo promedio según su juicio de experto, sin embargo este tiempo es debatido en el Sprint 1.

realice con total transparencia.

El CNE también puede designar por encargo a otras personas para que realicen la labor de observadores, estos observadores son invitados por el CNE, son personalidades que representan algún organismo internacional y si en caso aceptan ser parte del proceso electoral como observadores invitados, entonces deben apersonarse al lugar del escrutinio desde el lugar donde estén residiendo, el CNE se encarga de realizar los trámites correspondientes para encargarse de que los observadores invitados puedan llegar a presenciar el proceso sin ningún problema.

Se definen en base a la explicación del proceso, los siguientes tipos de observadores:

- Observador Independiente.- Debe postular ante el CNE para poder participar
- Observador Invitado.- El CNE lo debe invitar para que pueda participar.

### 3.1.2.2 **OBSERVADORES INDEPENDIENTES**

Un observador independiente puede ser cualquier ciudadano, que se postula, y que debe presentar una serie de requisitos que se deben validar en el CNE. Para este efecto, el postulante a observador independiente deberá rellenar uno de los siguientes formularios, de acuerdo al tipo de persona y nacionalidad.

Existen 4 tipos de formularios:

- Solicitud para Personas pertenecientes a organizaciones Jurídicas Internacionales.
- Solicitud para Personas pertenecientes a organizaciones Jurídicas Nacionales.
- Solicitud Personas Naturales Extranjeras.
- Solicitud Personas Naturales Nacionales.

Estos formularios estarán disponibles en la web del CNE (en períodos determinados), y el aspirante, seleccionará el formulario que le corresponda, y lo rellenará desde el sistema.

La cumplimentación de estos formularios tendrá la inclusión de ficheros adjuntos, que corresponderán a los documentos habilitantes para postularse como observador. Los documentos a adjuntar se describen en párrafos posteriores.

Estos formularios web, han de permitir poder adjuntar fotografías (posterior creación de las acreditaciones).

Las solicitudes se podrán hacer físicamente en la sede del CNE, donde el usuario CNE, únicamente tendrá que acceder a la web y cumplimentar los datos con el ciudadano delante (o a posterior)

Este formulario, se enviará al CNE, y en el backoffice, los usuarios, recibirán “alertas” donde se informará que un nuevo formulario ha sido cumplimentado.

Cada usuario del CNE, podrá acceder a un punto común (listado de solicitudes) y seleccionar el formulario sobre el que va a ser responsable y va a hacer el seguimiento, mediante la validación que se cumplen todos los requisitos solicitados.

Requisitos:

- Cédula de identidad/pasaporte
- Certificado de votación
- Carta de juramento de aceptación
- Fotografía (requisito necesario para la posterior creación de la identificación)
- Certificado de no afiliación política – este documento no debe ser adjuntado por el postulante, es un requisito que se verifica internamente en el CNE.
- No haber sido en los dos años anteriores miembro de Directiva de organización política o candidato a alguna dignidad de elección popular – requisito que se verifica internamente en el CNE.

Se tiene que validar la información del datos del usuario que ha cumplimentado el formulario, con las bases de datos de organizaciones Políticas (adherentes) para verificar si tiene o no afiliación y si no esa sido en los dos años anteriores directivo de alguna organización política o candidato a alguna dignidad de elección popular

El módulo, en cualquier caso, aceptará la carga de un fichero, aunque se estudiará la conexión interna con el módulo de operación electoral, donde consta toda la información de los partidos políticos, así como sus miembros.

Se establecerán validaciones del formulario, que se deberán realizar de manera automática, si el usuario que opta a ser observador está afiliado, o fue directivo o candidato a alguna dignidad de elección popular, el funcionario del CNE, de forma visual por pantalla pueda ver esa validación ejecutada, descartando ya ese candidato, si es usuario autorizador. De no ser usuario autorizador, existirán unos estados para cada candidatura, y se establecerán alertas y avisos a los usuarios autorizadores para que puedan proceder a finalizar el proceso.

Se debe prever que este procedimiento permita la solicitud de participación a personas extranjeras, siendo el procedimiento el mismo descrito.

Una vez verificado por parte del CNE, se aprueba la candidatura y se comunica vía mail que ha sido aceptado, otorgándosele la categoría de observador. A pesar de la comunicación electrónica, debe existir un procedimiento legal de notificación, donde se le entregará al ciudadano un documento firmado con la aceptación/denegación de su candidatura.

Con la información del formulario se crea la acreditación, y sin otra manipulación, el usuario diseño gráfico, podrán imprimir y plastificar las acreditaciones. (El usuario diseño gráfico debe tener la posibilidad de seleccionar los datos que desea que consten en la credencial, según el tipo de observador-nacional, internacional, jurídico, perteneciente a organismos electorales, etc)

Existirá un control de estados para los candidatos, aceptados/rechazados/notificados/ acreditación generada/acreditación enviada



Todo observador independiente ha de ser capacitado, pero esta capacitación, dada su especificidad, queda fuera el ámbito del módulo de capacitación. El observador es notificado que debe asistir a una capacitación, y este debe asistir. El módulo permitirá el registro manual de asistencia a la capacitación, para un monitoreo de las mismas.

Para la presentación del prototipo de todo lo descrito para este apartado, ScytI, aportará una solución segura y factible para aquellos postulantes a observadores no ecuatorianos (personas extranjeras) que no posean cedula de identidad, y puedan presentar una solicitud a postulante observador (se presentará una propuesta de como registrar un observador independiente internacional segura)

Igualmente, ScytI, estudia la viabilidad de que la plataforma sea multi-idioma, y en la presentación del prototipo se dará respuesta a esta petición.

Después del día de elecciones, el observador tiene la obligación de hacer llegar al CNE un informe, que se debe incorporar al sistema. Se evaluará la mejor opción, ya sea mediante la opción de dar una plantilla blanca para que la rellenen en la web, o bien, o también opción de cargar un fichero, por parte del observador, o parte del funcionario del CNE que reciba ese informe por otros canales. Todos los informes deben ser archivados en un repositorio común en el cual constaran los informes de observadores independientes y conducidos.

### 3.1.2.3 **OBSERVADORES INVITADOS**

Los observadores invitados serán cargados de origen en el sistema, en base a la base de datos entregada a ScytI.

A partir de esta carga inicial y única, existirá una pantalla que permitirá el ingreso de nuevos “invitados” en el sistema.

La estructura que deben seguir estos actores invitados es la siguiente:

- Organismos electorales (ej.UNASUR)
- Personas independientes
- Instituciones nacionales

En el momento de ingresar un nuevo sujeto en la base, se deberá indicar, a que ítem de la estructura pertenecen.

En base a los datos de invitados existentes, el sistema ha de permitir, mediante un buscador, localizar a las instituciones o individuos que el usuario necesite, para posteriormente, una vez localizado el individuo, o lista de individuos, el usuario del CNE, pueda seleccionar qué individuos son invitados.

En el momento de indicar que un individuo ha sido seleccionado como miembro invitado, el sistema le notificará a su email, que debe darse de alta en el sistema, para proceder a la cumplimentación del formulario establecido por el CNE. De dicho formulario, para los invitados independientes, el campo “hoja de vida” será obligatorio.

Paralelamente al registro del observador invitado, existe un procedimiento de trabajo propio del CNE, que consiste en la generación (fuera del sistema) de una carta de invitación.

A pesar de ser generada en fuera del sistema, dicha carta, ha de ser adjuntada al “expediente” del observador invitado.

Para cada observador y carta de invitación, existirá un estatus, que el usuario del CNE de forma manual indicará, a saber:

Estatus:

- Carta enviada
- Pendiente de envío.

El usuario del CNE, indicará para cada individuo, el estado de la carta.

Por su parte, cada observador, debe responder por carta, si acepta o no la invitación del CNE.

Nuevamente, en este caso, el sistema ha de permitir el ingreso de dicha documentación.

En esta ocasión, el usuario del CNE, tendrá a su disposición la opción para indicar, no sólo que la carta ha sido enviada (o no), si no, si se está pendiente de respuesta, o respuesta recibida.

En esta ocasión, tendremos disponibles los siguientes estados:

- Pendiente de respuesta
- Respuesta recibida – aceptación invitación
- Respuesta recibida – rechazo invitación.

Una vez el observador invitado acepta la invitación, el CNE procede a preparar los itinerarios de vuelos, y en esta ocasión, el usuario del CNE, tendrá a su disposición, para cada una de las fichas de los observadores, un apartado específico, donde adjuntar el itinerario que ha previsto para él.

En el momento que se adjunta el fichero (Word, xls, pdf), el sistema envía al observador un aviso que le indica que tiene el itinerario a su disposición para que proceda a aceptarlo (o rechazarlo).

El observador, una vez comprobado el itinerario, tendrá a su disposición la funcionalidad de aceptar o rechazar dicho itinerario.

Si **ACEPTA** – el usuario del CNE recibe una aviso que el observador a aceptado el itinerario

Si **RECHAZA** – se activa un campo de observaciones, para que éste pueda indicar el motivo del rechazo del itinerario. El usuario del CNE, recibirá un aviso igualmente, y tendrá opción de leer y responder, si conviene, los comentarios del observador.

El observador tendrá a su disposición en su cuenta del CNE, unas fichas técnicas que debe poder cumplimentar on-line.

El Administrador CNE es quien debe tener la facultad de modificar un formato de Ficha Técnica B. De producirse una modificación en esta ficha, se deben actualizar las mismas en todas las cuentas de observadores que estén activas.

Durante el proceso electoral, se procede a enviar informes por parte de los observadores ante el CNE, los dos tipos de observadores están en su derecho de emitir dichos informes expresando sus observaciones y/o reclamos. De hecho a los observadores invitados se les solicita que llenen ciertas fichas para que evalúen el resultado del proceso (Ficha técnica A y B)

El usuario solicitó también que el sistema permita generar reportes de los observadores según: país, organismo, siglas de organización, sexo.

El proceso finaliza en una fecha determinada donde el CNE establece que no se procederá a la recepción de más informes por parte de los observadores. Permanece el registro de quienes participaron como observadores independientes

e invitados, de estos últimos se mantiene el registro con el fin de invitarlos en próximas ocasiones.

El desarrollo del aplicativo de Gestión Observadores Electorales, introduce la automatización de la gestión de los mismos, además de algunas mejoras propuestas por los usuarios.

Del proceso descrito, se han identificado los requisitos funcionales y no funcionales con los cuales debe cumplir el sistema a desarrollar y se encuentran descritos en la siguiente tabla:

Tabla N° 2.- Requisitos Funcionales/No Funcionales Sprint 0

<b>Requisitos Funcionales</b>	<b>Requisitos no Funcionales</b>
	El sistema brindará seguridad para ser utilizado sólo por usuarios registrados en el sistema, con diferentes niveles de acceso.
El sistema debe realizarse en entorno web.	El sistema debe tener interfaz muy amigable ya que será usada por muchos usuarios no técnicos.
El sistema permitirá registrar solicitudes de observadores independientes	
El sistema permitirá registrar información de observadores independientes	
El sistema permitirá realizar consultas de los candidatos a observadores independientes por parte del CNE, para realizar su gestión.	
El sistema permitirá aprobar o rechazar solicitud de invitación	
El sistema permitirá generar cuentas de usuario para observadores	
El sistema permitirá a los usuarios observadores poder adjuntar informes	
El sistema permitirá asignar observadores independientes a un usuario del CNE para realizar seguimiento	

El sistema permitirá enviar correos electrónicos automáticos a los observadores según su estado.	
El sistema permitirá obtener información de las campañas y partidos políticos desde la base de datos.	
El sistema permite eliminar registro de observadores invitados cualquiera sea su estado.	
El sistema permitirá enviar cartas de invitación y cartas de itinerarios a los observadores invitados.	
El sistema permitirá crear grupos de observadores invitados.	
El sistema permitirá generar reportes.	

Fuente: Elaborado por la autora.

Nota: La lista de requerimientos ha sido socializada con los dueños del producto.

#### 3.1.2.4 **DELIMITACIÓN DEL ALCANCE DEL APLICATIVO**

De acuerdo a las necesidades descritas por el usuario se identificaron cuatro tipos de usuarios:

- Observador Independiente
- Observador Invitado
- Administrador CNE
- Usuario CNE

De acuerdo al análisis de los procesos y las funcionalidades identificadas, el software a desarrollarse tendrá los siguientes alcances.

- Módulo de Observadores Independientes, que permitirá:
  - Para el Observador independiente
    - Postular para ser observador.
    - Recibir notificaciones al correo electrónico sobre estado de postulación.
    - Adjuntar informes previa aceptación de su postulación.
  - Para el Usuario CNE
    - Visualizar listas de observadores aceptados.

- Tomar la responsabilidad de observadores.
- Generar credenciales de los observadores.
- Administrar estado de los observadores.
- Visualizar informes de observadores
- Para el Administrador CNE
  - Visualizar lista de postulaciones de observadores
  - Aprobar o Rechazar postulaciones de observadores.
  - Liberar de responsabilidad a Usuario CNE por algún observador.
  - Visualizar informes de observadores
- Módulo de Observadores Invitados, que permitirá:
  - Para el Observador invitado
    - Recibir correo del CNE invitándolo a participar del proceso
    - Adjuntar informes previa aceptación de invitación.
    - Completar fichas previa aceptación de invitación.
  - Para el Usuario CNE
    - Visualizar lista de observadores invitados disponibles
    - Añadir, editar o eliminar registro de observadores invitados
    - Enviar itinerarios de vuelo al observador invitado
    - Visualizar informes de observadores
    - Crear Grupos de observadores invitados.
    - Generar reportes de observadores.
  - Para el Administrador CNE
    - Visualizar lista de observadores invitados disponibles
    - Editar o eliminar registro de observadores invitados
    - Enviar cartas de invitación a observadores
    - Visualizar informes de observadores
    - Crear Grupos de observadores invitados.
    - Generar reportes de observadores.

### 3.1.2.5 **CONFORMACIÓN DEL EQUIPO DE TRABAJO**

Como en todos los proyectos, es necesario conocer el equipo humano con que se cuenta para trabajar en el proyecto.

El equipo de trabajo tiene el encargo de desarrollar el módulo de Observadores Electorales respetando como marco de trabajo la metodología Scrum.

Tabla N° 3.- Equipo de trabajo.

<b>ROL</b>	<b>PERSONA</b>	<b>AREA</b>
Product Owner	Raúl Murga	Project Owner- ScytI
Scrum Master	Liliana Cuya	Functional Analysis - ScytI
Team	Rony Villanueva	Team Lead - ScytI
	Franco Iturrizaga	Developer - ScytI
	Máximo Romero	Developer - ScytI
	Marissa Ticona	Developer - ScytI
	Fredy Huamán	UI- ScytI
	Marvin Meza	UI- ScytI

Fuente: Elaborado por la autora.

Nota: Raúl Murga es el encargado del proyecto y quien mantiene constante comunicación con el usuario, es miembro del equipo de ScytI Perú, sin embargo, representa las necesidades del usuario ante el equipo.

### 3.1.2.6 **DEFINICIÓN DEL BACKLOG DEL PRODUCTO**

El Backlog del producto o Pila del producto contiene la funcionalidad que el producto final debería tener. Tal como lo dice la metodología, para el presente proyecto se ha elaborado el Backlog del Producto, identificando las funcionalidades, priorizando cada una de ellas y realizando una estimación de tiempo requerido para su implementación.

De acuerdo al análisis que se desarrolló en los capítulos anteriores, el Backlog del Producto para el presente proyecto se encuentra definido en la siguiente tabla:

Tabla N° 4.- Product Backlog – Sprint 0

<b>ID</b>	<b>Módulo</b>	<b>Nombre</b>	<b>Prioridad</b>
1	Común para	Plataforma tecnológica	Muy Alta

2	todos los módulos	Interfaces de usuario	Muy Alta
3		Acceso al sistema de usuarios permitidos	Muy Alta
4		Documentación, análisis de casos de uso, modelo de base de datos.	Alta
5		Desarrollar Prototipos - Maquetas	Muy Alta
6	Módulo de Observadores Independientes	Las personas externas al CNE postulan a ser observador.	Muy Alta
7		Admin CNE aprueban o rechazan solicitud.	Muy Alta
8		Admin CNE/User CNE visualizan lista de observadores independientes	Muy Alta
9		User CNE realizan seguimiento a observadores según su estado	Media
10		User CNE imprime credenciales para observadores independientes	Media
11		Observador Independiente envía informes al CNE	Baja
12		Admin CNE/User CNE visualizan informes de observador independiente	Alta
13		Admin CNE puede eliminar observador	Baja
14	Módulo de Observadores Invitado	Admin CNE y User CNE registran nuevo observador invitado	Alta
15		Admin CNE envía invitación a observador invitado	Muy Alta
16		User CNE/Admin CNE envías itinerario de vuelo a observador	Alta
17		Observador Invitado completa Fichas técnica A y B	Media
18		Admin CNE y User CNE visualizan respuestas de fichas técnicas A y B	Baja
19		Admin CNE edita la Ficha técnica B	Alta
20		Observador invitado modifica su información de registro	Media
21		Observador invitado acepta o rechaza invitación del CNE	Media
22		Observador invitado acepta o rechaza itinerario de vuelo del CNE	Media
23		Admin CNE crea Grupos de observadores invitados	Media
24		Observador CNE adjunta informes al grupo asignado	Baja
25	Módulo de Reportes	User CNE/Admin CNE generan Reportes por países	Media



26	User CNE/Admin CNE generan Reportes por tipos de observadores	Media
27	User CNE/Admin CNE generan Reportes por sexo	Media
28	User CNE/Admin CNE generan Reportes por organización	Media
29	Permite seleccionar tipo de gráfico: lineal, barras o de porciones	Media

Fuente: Elaborado por la autora.

Nota: El Product Backlog inicial está sujeto a cambios que el usuario disponga, además los detalles específicos de cada requerimiento se irán aclarando con el usuario cada cierto tiempo.

La siguiente tabla describe los actores que participan:

Tabla N° 5.- Actores Sprint 0

<b>Actor</b>	<b>Descripción</b>
Usuario Observador Independiente	Usuario que se registra el sistema y si es aceptado por el personal del CNE puede enviar informes de observación.
Usuario Observador Invitado	Usuario que es invitado por el personal del CNE para que forme parte de observadores electorales.
Usuario Administrador CNE	Es quien decide quienes serán observadores, permite liberar de la responsabilidad de un observador a un Usuario CNE.
Usuario CNE	Se responsabiliza por observador y les genera sus credenciales, da seguimiento al estado del observador para que se capacite, reciba sus credenciales, etc.

Fuente: Elaborado por la autora.

### 3.2 CONSTRUCCIÓN

### **3.2.1 SPRINT 1**

En el Sprint número 1 se implementaran las funcionalidades que el Scrum Team Scrum identifique del Product Backlog realizado por el Product Owner.

A continuación se describen las reuniones que se llevaron a cabo para cada fase del Sprint:

#### **3.2.1.1 REUNIÓN DE PLANIFICACIÓN DE SPRINT (SPRINT PLANNING MEETING)**

La reunión de planificación del Sprint 1, se llevó a cabo con todos los integrantes del equipo del proyecto, además de interesados invitados del proyecto por el Product Owner.

Los asistentes a la reunión fueron:

- Product Owner
- Scrum Master
- Scrum Team
- Invitado: Gerente de proyectos de Scytl
- Invitado: Equipo de Calidad (QA)

Las funciones de cada rol dentro de la reunión de planificación del Sprint fueron los siguientes:

Responsabilidades del Product Owner

- Presencia en las reuniones en las que el equipo elabora la pila del sprint. Resolución de dudas sobre las historias de usuario que se descomponen en la pila del sprint.

Responsabilidades del Scrum Manager

- Supervisión y asesoría en la elaboración de la pila de la pila del sprint.

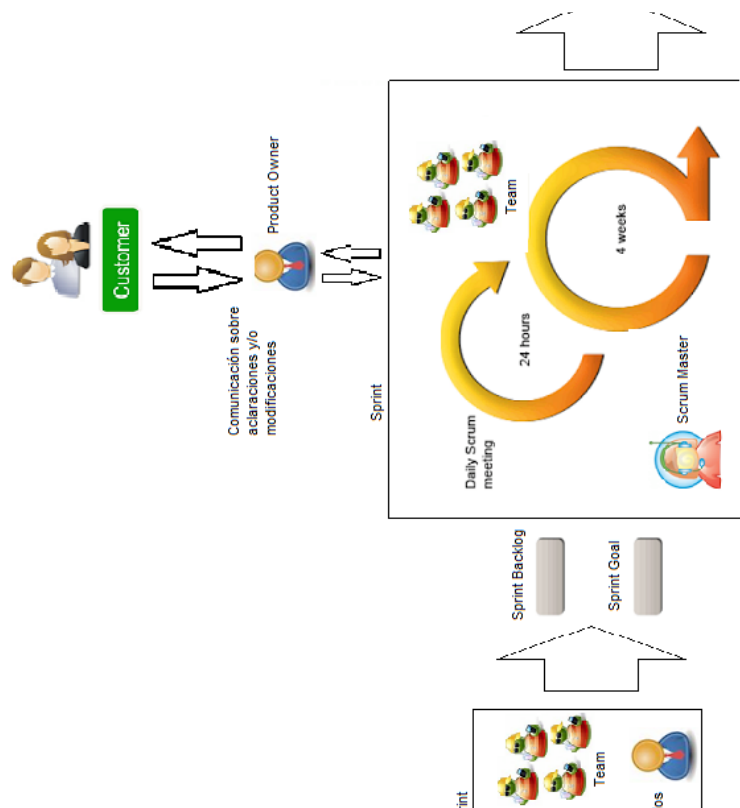
Responsabilidades del Scrum Team

- Elaboración de la pila del sprint.

- Resolución de dudas o comunicación de sugerencias sobre las historias de usuario con el gestor del producto.

La duración de la reunión fue de 7 horas. Durante la primera hora de la reunión se explicó al equipo del proyecto la nueva forma de trabajar ese aplicativo siguiendo la metodología Scrum, además se acordó cómo se realizaría cada iteración. El siguiente gráfico muestra la forma de trabajo:

Figura N° 7.- Forma de Iteración o Sprint



Fuente: Elaborado por la autora

Se explicó también la forma de interpretar los bugs que realizará el área de QA al aplicativo.

Todas las pruebas se realizan en base a las especificaciones que están descritas en la User Story <sup>(2)</sup>.

Cabe mencionar que las User Story se encuentran en el RedMine; por tanto cada bug que se encuentra es designado a la persona que se encarga de desarrollar la User Story dependiendo del nivel del error, es decir, si el error es de Back End entonces se lo designa a la persona encargada, si es de Front End o maquetación entonces se lo designa a quien se esté encargando de esta parte.

Los Bugs han sido tipificados de la siguiente manera:

- Tipo A.- Describen errores que impiden el correcto funcionamiento del flujo básico del aplicativo.
- Tipo B.- Errores que no impiden el flujo pero da al aplicativo un funcionamiento no óptimo.
- Tipo C.- No impiden el flujo, son errores que no influyen en el funcionamiento de algún requerimiento pero no aportan calidad al aplicativo.

Para las pruebas de los requerimientos es necesario que todos estén descritos en la User Story correspondiente, ya que de no ser así no llegará a dar la conformidad de que el sistema está realizado de acuerdo a lo solicitado.

Luego, durante las 3 horas siguientes, el Product Owner dio una explicación de los requerimientos funcionales del aplicativo a desarrollar y se procedió a revisar los ingresos de este Sprint, obtenidos del Product Backlog que se desarrolló en el Sprint 0.

Las entradas que darán origen a la reunión de planificación de este Sprint son:

---

(2) La herramienta RedMine definida anteriormente, contiene las US (User Story) de cada uno de los requerimientos solicitados en el Product Backlog. Ver Anexos sobre ejemplos.

- Product Backlog
- Último Incremento.- por ser el primer Sprint, no se considera esta entrada.
- Experiencias de iteraciones pasadas.- por ser el primer Sprint, no se considera esta entrada.

Al tomar en cuenta estas entradas, esta reunión servirá para identificar qué puntos del Product Backlog se podrán atender en este Sprint y el tiempo estimado de la duración del Sprint, todas estas cuestionamientos son resueltos por el Team Scrum en esa reunión.

### 3.2.1.1.1 DEFINICIÓN DE FUNCIONES DEL EQUIPO DE TRABAJO

El equipo de trabajo para la implementación de las funcionalidades del sub módulo de Observadores Independientes esta descrito en la siguiente tabla:

Tabla N° 5.- Definición del equipo de trabajo Sprint 1

ROL	PERSONA	Descripción de tareas
Product Owner	Raúl Murga	Administración del proyecto desde la perspectiva del negocio.
Scrum Master	Liliana Cuya	Asegurar que el proceso Scrum se lleve a cabo.
	Liliana Cuya	Análisis funcional, documentación.
	Fredy Huaman	Maquetación HTML- Pruebas funcionales
	Marvin Meza	Desarrollo Front End.
	Máximo Romero	Desarrollo Back End
	Marissa Ticona	Desarrollo Back End
	Josue Barrios	Desarrollo Back End
	Rony Villanueva	Asesoría técnica a desarrolladores

Fuente: Elaborado por la autora.

### 3.2.1.1.2 SPRINTBACKLOG

Durante la reunión se coordinó los puntos que se implementarían en la iteración, que según el Scrum Team durará 1 mes, los puntos acordados que se terminaran al finalizar la iteración son los siguientes:

Tabla N° 6.- Scrum Backlog Sprint 1

Backlog ID	Módulo	Tarea	Estado	Responsable	Estimación de tiempo (días)
1	Común	Plataforma	New	Rony	6

	para todos los módulos	tecnológica		Villanueva	
2		Interfaces de usuario	New	Fredy Huaman	8
3		Acceso al sistema de usuarios permitidos	New	Rony Villanueva	4
4		Análisis de casos de uso, modelo de base de datos	New	Liliana Cuya	8
5		Desarrollar Prototipos-Maquetas	New	Liliana Cuya	8
6	Observadores Independientes	Personas externas postulan a ser observador	New	Franco Iturrizaga	5
7		Admin CNE aprueba/rechaza solicitud	New	Franco Iturrizaga	2
8		Admin CNE/User CNE visualizan lista de observadores independientes	New	Marissa Ticona	2
9		User CNE modifica el estado del observador	New	Marvin Meza	3
11		Observador Independiente envía informes al CNE	New	Marissa Ticona	3
14	Observadores Invitados	Admin CNE y User CNE registran nuevo observador invitado	New	Max Romero	3
15		Admin CNE envía invitación a observador invitado	New	Max Romero	3
16		User CNE/Admin CNE envías itinerario de vuelo a observador	New	Max Romero	2
21		Observador	New	Marvin Meza	2

		invitado acepta o rechaza invitación del CNE			
22		Observador invitado acepta o rechaza itinerario de vuelo del CNE	New	Marvin Meza	2

Fuente: Elaborado por la autora.

### **3.2.1.1.3 DEFINICIÓN DEL SPRINT GOAL**

El objetivo que define el Sprint Goal que el equipo técnico desarrollo, consiste en completar todos las funcionalidades descritas en el Sprint Backlog dentro de 1 mes, además el equipo compromete a tener desarrollado un Demo con las que cumpla el flujo básico de estas requerimientos, para que el usuario pueda revisarlo durante la iteración.

### **3.2.1.2 SCRUM DIARIO**

Una vez finalizada la etapa de planificación de la iteración, el equipo se reúne diariamente para ponerse al tanto del avance del proyecto. La reunión se realiza durante 15 a 20 minutos evaluando los porcentajes de avances que se modifican en el Sprint Backlog y en las Historias de usuario que se registraron en RedMine, donde están todas las especificaciones de cada requerimiento.

Se planteó realizar coordinaciones con el usuario directo para aclarar dudas de análisis funcional que no están especificados en las las Historias de Usuario; estas coordinaciones se realizaron a través de la herramienta Skype mediante conversaciones escritas o video conferencias que luego son formalizadas a través de la modificación del Product Backlog.

### **3.2.1.3 INCREMENTO DEL SPRINT**

Los incrementos se desarrollaron en cuando a documentación y en cuanto a funcionalidad del aplicativo:

- **Diagramas de Caso de Uso**

Figura N° 8.- Diagrama de Casos de Uso



Fuente: Elaborado por la autora.

- **Maquetas- Prototipos de análisis**

A continuación se presentan, las maquetas de los casos de uso descritos arriba:

- **Gestión Postulación Observador**

Figura N° 9.- Prototipo-Gestión Postulación Observador



[Inicio](#) > [Sección](#) > Subsección

Bienvenido al inscripción para Observadores Independientes

Texto sobre Observadores electorales e inscripción de observadores independientes

[Icon](#) **Solicitud Personas Jurídicas Interanacionales**  
Lorem ipsum dolor sit amet, consectetur

[Solicita online](#) | [Descarga formulario](#)

[Icon](#) **Solicitud Personas Naturales Extranjeras**  
Importar datos de configuración electoral

[Solicita online](#) | [Descarga formulario](#)

[Icon](#) **Solicitud Personas Jurídicas Nacionales**  
Sed iacinia, massa eget iaculis sollicitudin

[Solicita online](#) | [Descarga formulario](#)

[Icon](#) **Solicitud Personas Naturales Nacionales**  
Lorem ipsum dolor sit amet, consectetur

[Solicita online](#) | [Descarga formulario](#)

[Icon](#) **Envios del informe**  
Lorem ipsum dolor sit amet, consectetur

[Enviar](#)

[Icon](#) **Formularios online**  
Lorem ipsum dolor sit amet, consectetur

[Descargar](#)

[Descargar](#)

[Descargar](#)

[Icon](#) **Asistencia a Capacitación**  
Lorem ipsum dolor sit amet, consectetur

[Empezar](#)

## - **Gestión de Registro de Independiente**

Figura N° 10.- Prototipo-Gestión de Registro de Independiente

## Solicitudes

Mis solicitudes realizadas y pendientes

Título de tabla					+ Solicitar Observación
<a href="#">Importar</a> <a href="#">Descargar</a> <a href="#">Imprimir</a>					Search list...
Solicitud	Pais	Fecha	Estado	Capacitado	
<a href="#">Personas Jurídicas Internacionales</a>	Ecuador	14.09.2013	Pendiente	No	
<a href="#">Personas Jurídicas Internacionales</a>	Ecuador	14.09.2013	Aprobado	Sí	
<a href="#">Personas Jurídicas Internacionales</a>	Ecuador	14.09.2013	Aprobado	Sí	
<a href="#">Personas Jurídicas Internacionales</a>	Ecuador	14.09.2013	Aprobado	Sí	

10 de 128 resultados

[Primero](#)
[Anterior](#)
[1](#)
[2](#)
[3](#)
[4](#)
[Siguiete](#)
[Último](#)

## - Gestión Observador Invitados-Generación

Figura N° 11.- Prototipo-Gestión Observador Invitados-Generación

## Observadores Invitados

Texto sobre observadores

Título de tabla							+ Añadir Observador
<a href="#">Importar</a> <a href="#">Descargar</a> <a href="#">Imprimir</a>							Search list... <a href="#">Búsqueda avanzada</a>
Show: 15							
Solicitud	Itinerario	Estructura	Carta	Estado	Edit	Delete	
<a href="#">Nombre, Apellido</a>	Enviada	UNASUR	Enviada	Invitado			
<a href="#">Nombre, Apellido</a>	Pendiente de Envio	Personas independientes	Pendiente de Envio	Disponible			
<a href="#">Nombre, Apellido</a>	Aceptada	Instituciones nacionales	Aceptada	Invitado			
<a href="#">Nombre, Apellido</a>	Rechazada	UNASUR	Rechazada	Disponible			
<a href="#">Nombre, Apellido</a>	Rechazada	Personas independientes	Rechazada	Disponible			
<a href="#">Nombre, Apellido</a>	Enviada	Instituciones nacionales	Enviada	Invitado			
<a href="#">Nombre, Apellido</a>	Pendiente de Envio	UNASUR	Pendiente de Envio	Disponible			
<a href="#">Nombre, Apellido</a>	Aceptada	Personas independientes	Aceptada	Disponible			
<a href="#">Nombre, Apellido</a>	Rechazada	Instituciones nacionales	Rechazada	Disponible			
<a href="#">Nombre, Apellido</a>	Rechazada	UNASUR	Rechazada	Invitado			

10 de 128 resultados

[Primero](#)
[Anterior](#)
[1](#)
[2](#)
[3](#)
[4](#)
[Siguiete](#)
[Último](#)

## - Gestión de Itinerarios de vuelo


Figura N° 12.- Prototipo-Gestión de Itinerarios de vuelo

## Invitaciones

Texto explicativo sobre observador

Observador Invitado

Título de tabla

Estado de Invitación		Estado de Itinerarios de vuelos	
<input type="checkbox"/> Pendiente de envío <input type="checkbox"/> Carta enviada <input type="checkbox"/> Invitación aceptada <input type="checkbox"/> Invitación rechazada		<input type="checkbox"/> Aceptada <input type="checkbox"/> Rechazada	
Estructura	UNASUR		
Evento Electoral	XXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX		
Nombres	Jose		
Apellidos	Perez		
No. Pasaporte	1234567890		
Tipo de pasaporte	Oficial o de Servicio		
Nombre de organización	Organización X		
Página web	www.organización.com		
No. teléfono	1234567890		
Teléfono celular	1234567890		
Correo electrónico personal	nombre.apellido@mail.com		
Correo electrónico institucional	nombre.apellido@mail.com		
País de residencia	Peru		

## - Gestión de Fichas Técnicas



Figura N° 13.- Prototipo-Gestión de Fichas Técnicas B

### FICHA TÉCNICA (B)


Texto sobre Ficha Técnica B

Comentarios respecto a la Juntas Intermedias

JUNTAS INTERMEDIAS 1/2

7. ¿Ejemplo pregunta añadida?  Sí  No  

---

Pregunta  

Opciones Respuesta  Sí  
 No  
 Campo libre

**+ Añadir pregunta**

< Atras Cancelar Guardar Enviar

Figura N° 14.- Prototipo-Gestión de Fichas Técnicas A

FICHA TÉCNICA (A)

Texto sobre Ficha Técnica A

Título de tabla

Datos Personales    Juntas receptoras del voto    Proceso de Votación    Participación de las electoras    Proceso del escrutinio

GUIA DEL PROCESO DE VOTACIÓN

Nombre del Observador/a

Institución

Fecha

Provincia asignada

Cantón

Circunscripción

Recinto

- **Gestión de informes grupales**

Figura N° 15.- Prototipo-Gestión de informes grupales

Informes grupales

Texto sobre informes grupales

Título de tabla + Crear grupo

[Importar](#)    [Descargar](#)    [Imprimir](#)        [Búsqueda avanzada](#)

Show:

Solicitud	Fecha creación	No. Usuarios	Infomes Adjuntos	Edit	Delete
<a href="#">Grupo 1</a>	01.01.2014	10	SI		
<a href="#">Grupo 2</a>	01.01.2014	6	SI		
<a href="#">Grupo 3</a>	01.01.2014	10	SI		
<a href="#">Grupo 4</a>	01.01.2014	5	NO		
<a href="#">Grupo 5</a>	01.01.2014	8	SI		
<a href="#">Grupo 6</a>	01.01.2014	10	SI		
<a href="#">Grupo 7</a>	01.01.2014	10	NO		
<a href="#">Grupo 8</a>	01.01.2014	7	SI		
<a href="#">Grupo 9</a>	01.01.2014	10	SI		
<a href="#">Grupo 10</a>	01.01.2014	10	SI		

10 de 128 resultados

Primero Anterior 1 2 3 4 Siguiente Último

## - **Generación de Reportes**

Figura N° 16.- Prototipo-Generación de reportes

### Reportes de observadores

Texto sobre reportes

Reportes + Crear reporte

Nombre

Pais

Organismo

Sexo

Tipo de Observador

Asistieron a capacitación

Remitieron informe

Mostrar  entradas

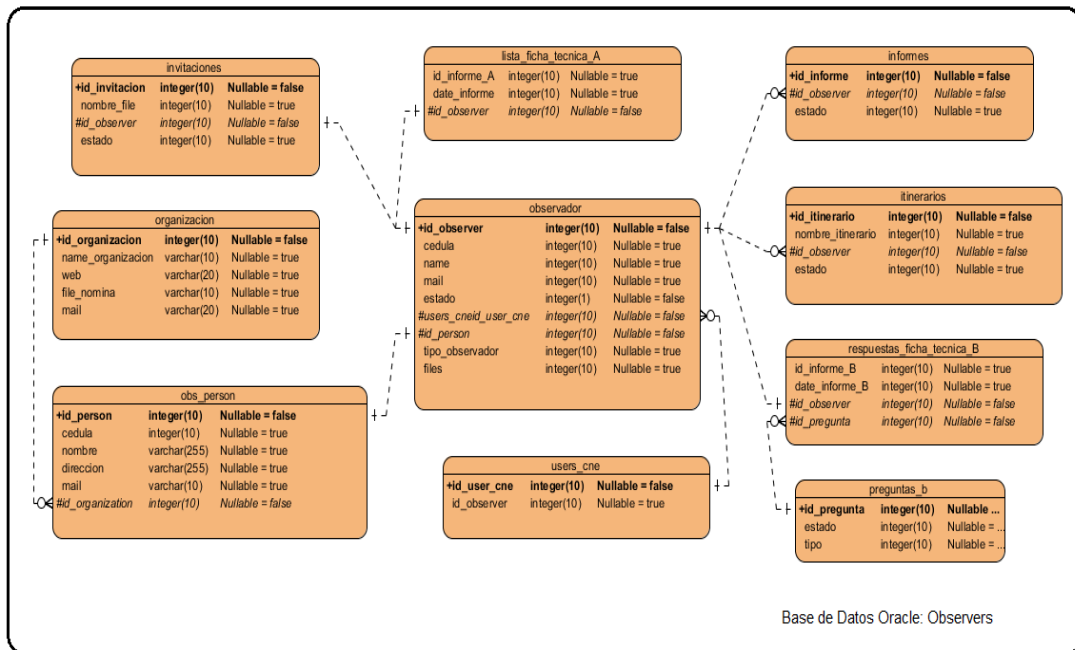
<input type="checkbox"/>	Nombre	Pais	Organismo	Sexo	Tipo de observador	Asistieron a capacitación	Remitieron informe
<input type="checkbox"/>	Nombre Apellido	Chile	ABC	Hombre	Independiente	SI	SI
<input type="checkbox"/>	Nombre Apellido	Argentina	DEF	Hombre	Invitado	SI	SI
<input type="checkbox"/>	Nombre Apellido	Brasil	GHI	Mujer	Natural	NO	NO
<input type="checkbox"/>	Nombre Apellido	Uruguay	JKL	Hombre	Jurídico	NO	NO
<input type="checkbox"/>	Nombre Apellido	Colombia	NMO	Mujer	Independiente	SI	SI
<input type="checkbox"/>	Nombre Apellido	Peru	PQR	Mujer	Invitado	SI	SI
<input type="checkbox"/>	Nombre Apellido	Venezuela	STV	Mujer	Natural	NO	NO
<input type="checkbox"/>	Nombre Apellido	Honduras	WZX	Hombre	Jurídico	NO	NO

10 de 20 resultados

- **Modelado de Base de Datos**

Del análisis de la especificación de los casos de uso, se determina la necesidad de utilizar el modelo de datos descrito en el diagrama de entidad relación mostrado a continuación, las mismas que serán implementados como objetos en la base de datos y como entidades del negocio en la codificación de la aplicación.

Figura N° 17.- Modelo de Base de Datos



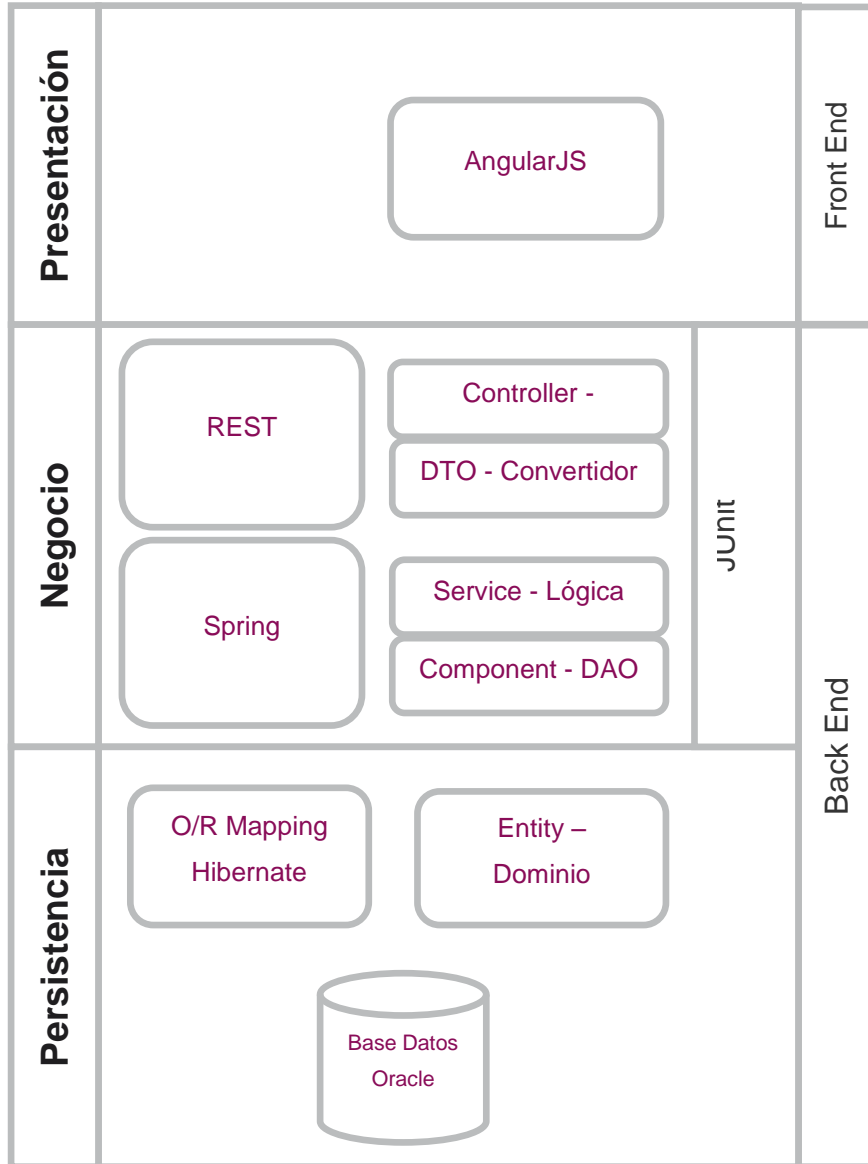
Fuente: Elaborado por la autora.

- **Arquitectura de la Aplicación**

Para la implementación del módulo de Observadores Electorales, se utilizará el modelo:

La capa de Persistencia y Negocio se trabajan en Back End, mientras que la capa de presentación o la interfaz gráfica se trabaja en el Front End con Angular JS.

Figura N° 18.- Arquitectura de la aplicación.



Fuente: Elaborado por la autora.

### 3.2.1.4 REVISIÓN DEL SPRINT (SPRINT REVIEW)

En esta fase se analiza los incrementos obtenidos durante el Sprint que se reflejan en el Sprint Backlog. A continuación se muestra el resultado final de la primera iteración con respecto al Sprint Backlog:

Tabla N° 7.- Revisión Scrum Backlog Sprint 1

Bac klog ID	Módulo	Tarea	Estado	Responsable	Estimació n de tiempo (días)	Avan ce
1	Común para todos los módulos	Plataforma tecnológica	Done	Rony Villanueva	6	
2		Interfaces de usuario	Done	Fredy Huaman	8	
3		Acceso al sistema de usuarios permitidos	Done	Rony Villanueva	4	
4		Análisis de casos de uso, modelo de base de datos	Done	Liliana Cuya	8	
5		Desarrollar Prototipos- Maquetas	Done	Liliana Cuya	8	
6	Observadores Independientes	Personas externas postulan a ser observador	Done	Franco Iturrizaga	5	
7		Admin CNE aprueba/rechaza solicitud	Done	Franco Iturrizaga	2	
8		<b>Admin CN visualiza lista de observadores independientes. Usuario CNE visualiza lista de observadores aceptados.</b>	Done	Marissa Ticona	2	
9		User CNE modifica el estado del observador	New	Marvin Meza	3	80%
11		Observador Independiente envía informes al CNE	New	Marissa Ticona	3	90%
14	Observadores Invitados	<b>User CNE registran nuevo observador invitado</b>	Done	Max Romero	3	
15		Admin CNE envía invitación a observador invitado	Bug	Max Romero	3	
16		<b>User CNE envía</b>	Bug	Max Romero	2	



		<b>itinerario de vuelo a observador</b>				
21		Observador invitado acepta o rechaza invitación del CNE	Done	Marvin Meza		2
22		Observador invitado acepta o rechaza itinerario de vuelo del CNE	Done	Marvin Meza		2

Fuente: Elaborado por la autora.

Nota:

Los requerimientos que tienen estado Done, son los que se terminaron y probaron satisfactoriamente.

Los que tienen estado Bug deben ser atendidos en el siguiente Sprint, y los que aún no se terminan también deben ser atendidos en el siguiente Sprint.

Los ítems que están resaltados en negrita, han sido modificados desde el Sprint Backlog inicial por requerimiento del usuario.

### 3.2.1.5 ACTUALIZACIÓN DE PRODUCT BACKLOG

Tabla N° 8.- Actualización de Product Backlog Sprint 1

ID	Módulo	Nombre	Estado	Prioridad
1	Común para todos los módulos	Plataforma tecnológica	Done	Muy Alta
2		Interfaces de usuario	Done	Muy Alta
3		Acceso al sistema de usuarios permitidos	Done	Muy Alta
4		Documentación, análisis de casos de uso, modelo de base de datos.	Done	Alta
5		Desarrollar Prototipos - Maquetas	Done	Muy Alta
6	Módulo de Observadores Independientes	Las personas externas al CNE postulan a ser observador	Done	Muy Alta
7		Admin CNE aprueban o rechazan solicitud	Done	Muy Alta
8		Admin CNE visualiza lista de observadores independientes. Usuario CNE visualiza observadores aceptados.	Done	Muy Alta
9		User CNE realizan seguimiento a observadores según su estado	80%	Media

10		User CNE imprime credenciales para observadores independientes	-	Media
11		Observador Independiente envía informes al CNE	90%	Baja
12		Admin CNE/User CNE visualizan informes de observador independiente	-	Alta
13		Admin CNE puede eliminar observador	-	Baja
14	Módulo de Observadores Invitado	<b>User CNE registran nuevo observador invitado</b>	Done	Alta
15		Admin CNE envía invitación a observador invitado	-	Muy Alta
16		<b>User CNE envías itinerario de vuelo a observador</b>	-	Alta
17		<b>Observador Invitado completa Fichas técnica A y B. Podrá llenar varias fichas técnicas A.</b>	-	Media
18		Admin CNE y User CNE visualizan respuestas de fichas técnicas A y B	-	Baja
19		Admin CNE edita la Ficha técnica B	-	Alta
20		Observador invitado modifica su información de registro: Acreditación como observador invitado.	-	Media
21		Observador invitado acepta o rechaza invitación del CNE	Done	Media
22		Observador invitado acepta o rechaza itinerario de vuelo del CNE	Done	Media
23		Admin CNE crea Grupos de observadores invitados	-	Media
24		Observador CNE adjunta informes al grupo asignado	-	Baja
25		Módulo de Reportes	User CNE/Admin CNE generan Reportes por países	-
26	User CNE/Admin CNE generan Reportes por tipos de observadores		-	Media
27	User CNE/Admin CNE generan Reportes por sexo		-	Media
28	User CNE/Admin CNE generan Reportes por organización		-	Media
29	Permite seleccionar tipo de gráfico: lineal, barras o de		-	Media

		porciones		
--	--	-----------	--	--

Fuente: Elaborado por la autora.

Nota:

El requerimiento de código 13 de la tabla presentada ha sido eliminado del Product Backlog a pedido del usuario.

Nota:

Los ítems que están resaltados en negrita, han sido modificados desde el Sprint Backlog inicial por requerimiento del usuario.

### **3.2.2 SPRINT 2**

En el Sprint número 2 se implementaran las funcionalidades que el Scrum Team identifique del Product Backlog Actualizado según incremento de la iteración anterior, además de las funcionalidades que no se terminaron de completar en el Sprint anterior y las que fueron reportadas con Bug.

A continuación se describen las reuniones que se llevaron a cabo para cada fase del Sprint:

#### **3.2.2.1 REUNIÓN DE PLANIFICACIÓN DE SPRINT (SPRINT PLANNING MEETING)**

La reunión de planificación del Sprint 2, se llevó a cabo con todos los integrantes del equipo del proyecto, además de los invitados del proyecto:

Los asistentes a la reunión fueron:

- Product Owner
- Scrum Master
- Scrum Team
- Invitado: Gerente de proyectos de ScytI
- Invitado: Equipo de Calidad (QA)

Durante esta reunión de planificación se analizó la situación del Sprint anterior para identificar áreas de mejora para el presente Sprint. De los principales inconvenientes que se encontraron fueron:

- Falta de despliegue en servidor de pruebas, para que el área de Calidad revise el aplicativo a tiempo.
- No se actualizaron las User History del RedMine mientras se iban presentando aclaraciones por parte del usuario.
- Se sugirió que un miembro del equipo que este libre en tiempo, se dedique a probar las funcionalidades antes de mandar a probar a QA.

### 3.2.2.1.1 SPRINT BACKLOG

El Sprint Backlog para la segunda iteración es el siguiente, cabe resaltar que han añadido a la pila las funcionalidades donde se detectaron error para su corrección, y las funcionalidades que no fueron completadas en Sprint anterior.

Tabla N° 9.- Sprint Backlog Sprint 2

Backlog ID	Módulo	Tarea	Estado	Responsable	Estimación de tiempo (días)	Avance
9	Observadores Independientes	User CNE modifica el estado del observador		Marvin Meza	5	80%
10		User CNE imprime credenciales para observadores independientes	New	Franco Iturrizaga	5	
11		Observador Independiente envía informes al CNE		Marissa Ticona	2	90%
12		Admin CNE/User CNE	New	Marissa Ticona	2	

		visualizan informes de observador independiente				
15	Observadores Invitados	Admin CNE envía invitación a observador invitado	Bug	Max Romero	4	
16		User CNE envías itinerario de vuelo a observador	Bug	Max Romero	3	
20		Acreditación como observador invitado	New	Marvin Meza	3	
18		Observador Invitado completa Fichas técnica A y B. Podrá llenar varias fichas técnicas A.	New	Marissa Ticona	4	
22		Admin CNE edita la Ficha técnica B	New	Marvin Meza	4	
23		Admin CNE crea Grupos de observadores invitados	New	Franco Iturrizaga	5	
24		Observador CNE adjunta informes al grupo asignado	New	Franco Iturrizaga	5	

Fuente: Elaborado por la autora.

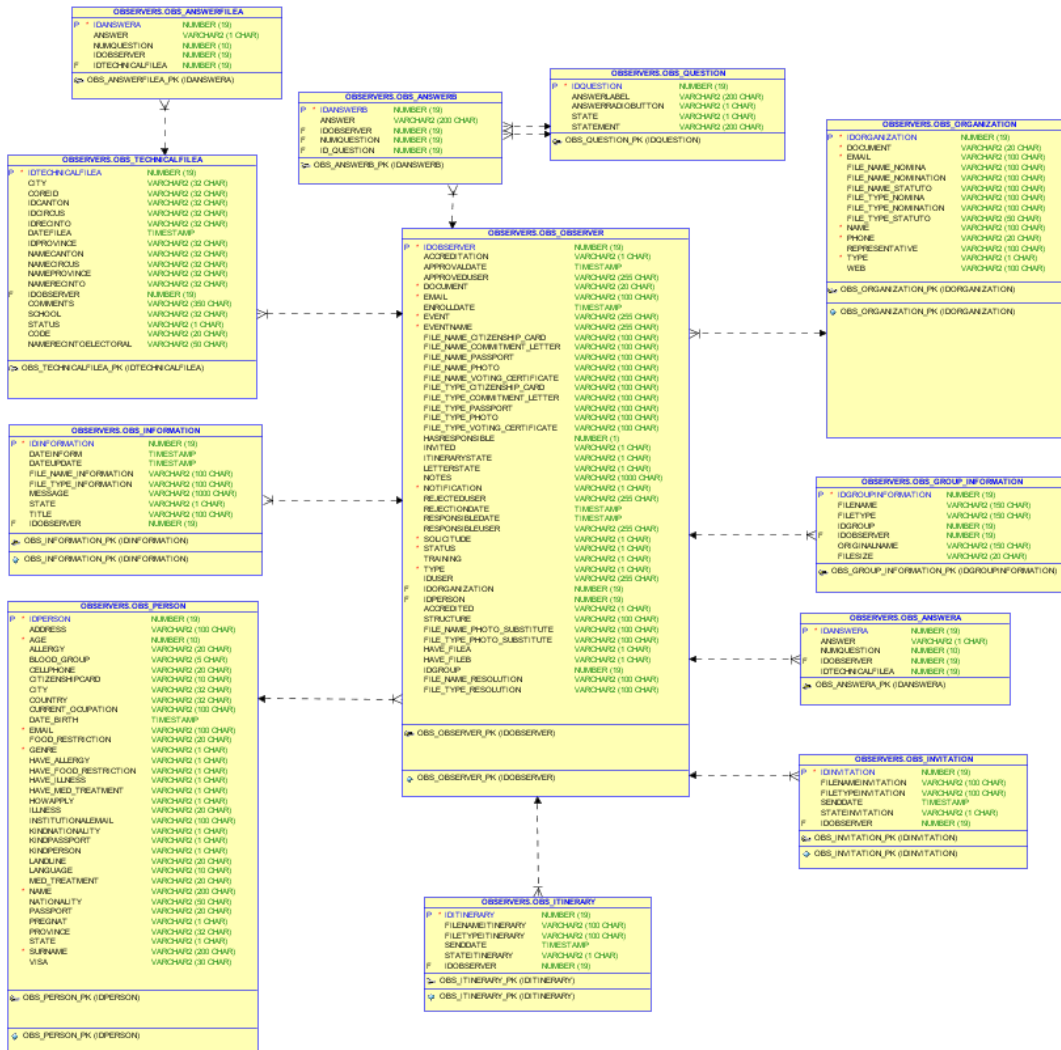
### **3.2.2.1.2 DEFINICIÓN DEL SPRINT GOAL**

El objetivo que define el Sprint Goal que el equipo técnico desarrolló, consiste en completar todas las funcionalidades descritas en el Sprint Backlog dentro de 1 mes, además el equipo compromete a tener listo un despliegue para pruebas en QA durante la mitad del mes solicitado para la duración del Sprint, de esta manera se realizará doble validación al módulo dentro de la misma iteración.

### 3.2.2.2 INCREMENTO DEL SPRINT

- Incremento en Base de Datos

Figura N°19.- Incremento en BD- Sprint 2.



Fuente: ScytI Perú.

- Incremento en funcionalidad

Con cada iteración se van incrementando las entregas para completar el producto, se incrementó la programación en Back End y en Front End, con la creación de nuevas líneas de códigos. Se analizará las funcionalidades atendidas en la revisión del Sprint.

### 3.2.2.3 REVISIÓN DEL SPRINT (SPRINT REVIEW)

Tabla N° 10.- Revisión del Sprint 2

Backlog ID	Módulo	Tarea	Estado	Responsable	Estimación de tiempo (días)	Avance
9	Observadores Independientes	User CNE modifica el estado del observador	Done	Marvin Meza	5	
10		User CNE imprime credenciales para observadores independientes		Franco Iturrizaga	5	60%
11		Observador Independiente envía informes al CNE	Done	Marissa Ticona	2	
12		Admin CNE/User CNE visualizan informes de observador independiente	Done	Marissa Ticona	2	
15	Observadores Invitados	Admin CNE envía invitación a observador invitado	Done	Max Romero	4	
16		User CNE envía itinerario de vuelo a observador	Done	Max Romero	3	
20		Acreditación como observador invitado	Done	Marvin Meza	3	
18		Observador Invitado completa Fichas técnica A y B. Podrá llenar varias fichas técnicas A.		Marissa Ticona	4	80%
22		Admin CNE		Marvin	4	80%

		edita la Ficha técnica B		Meza		
23		Admin CNE crea Grupos de observadores invitados	Bug	Franco Iturrizaga	5	
24		Observador CNE adjunta informes al grupo asignado	Bug	Franco Iturrizaga	5	

Fuente: Elaborado por la autora.

### 3.2.2.3.1 ACTUALIZACIÓN DE PRODUCT BACKLOG

Tabla N° 11.- Actualización Product Backlog Sprint 2

ID	Módulo	Nombre	Estado	Prioridad
1	Común para todos los módulos	Plataforma tecnológica	Done	Muy Alta
2		Interfaces de usuario	Done	Muy Alta
3		Acceso al sistema de usuarios permitidos	Done	Muy Alta
4		Documentación, análisis de casos de uso, modelo de base de datos.	Done	Alta
5		Desarrollar Prototipos - Maquetas	Done	Muy Alta
6	Módulo de Observadores Independientes	Las personas externas al CNE postulan a ser observador	Done	Muy Alta
7		Admin CNE aprueban o rechazan solicitud	Done	Muy Alta
8		Admin CNE visualiza lista de observadores independientes. Usuario CNE visualiza observadores aceptados.	Done	Muy Alta
9		User CNE realizan seguimiento a observadores según su estado	Done	Media
10		User CNE imprime credenciales para observadores independientes	60%	Media
11		Observador Independiente envía informes al CNE	Done	Baja
12		Admin CNE/User CNE visualizan informes de observador independiente	Done	Alta
13		<del>Admin CNE puede eliminar observador</del>	-	<del>Baja</del>
14	Módulo de Observadores	User CNE registran nuevo observador invitado	Done	Alta



15	Invitado	Admin CNE envía invitación a observador invitado	Done	Muy Alta	
16		User CNE envías itinerario de vuelo a observador	Done	Alta	
17		Observador Invitado completa Fichas técnica A y B. Podrá llenar varias fichas técnicas A.	80%	Media	
18		Admin CNE y User CNE visualizan respuestas de fichas técnicas A y B	-	Baja	
19		Admin CNE edita la Ficha técnica B	80%	Alta	
20		Observador invitado modifica su información de registro: Acreditación como observador invitado.	Done	Media	
21		Observador invitado acepta o rechaza invitación del CNE	Done	Media	
22		Observador invitado acepta o rechaza itinerario de vuelo del CNE	Done	Media	
23		Admin CNE crea Grupos de observadores invitados	-	Media	
24		Observador CNE adjunta informes al grupo asignado	-	Baja	
25		Módulo de Reportes	User CNE/Admin CNE generan Reportes por países	-	Media
26			User CNE/Admin CNE generan Reportes por tipos de observadores	-	Media
27			User CNE/Admin CNE generan Reportes por sexo	-	Media
28			User CNE/Admin CNE generan Reportes por organización	-	Media
29	Permite seleccionar tipo de gráfico: lineal, barras o de porciones		-	Media	

Fuente: Elaborado por la autora.

### **3.2.3 SPRINT 3**

En el Sprint número 3 se implementaran las funcionalidades que el Scrum Team identifique del Product Backlog Actualizado según incremento de la iteración anterior.

#### **3.2.3.1 REUNIÓN DE PLANIFICACIÓN DE SPRINT (SPRINT PLANNING MEETING)**

La reunión de planificación del Sprint 3, se llevó a cabo con todos los integrantes del equipo del proyecto, además de un interesado invitado del proyecto por el Product Owner.

Los asistentes a la reunión fueron:

- Product Owner
- Scrum Master
- Scrum Team
- Invitado: Gerente de proyectos de ScytI
- Invitado: Equipo de Calidad (QA)

Durante esta reunión de planificación se confirmó que esta iteración debía ser la última ya que el aplicativo entraría en producción según lo acordado con el Product Owner.

#### **3.2.3.1.1 SPRINT BACKLOG**

El Sprint Backlog para la segunda iteración es el siguiente, cabe resaltar que han añadido a la pila las funcionalidades donde se detectaron error para su corrección, y las funcionalidades que no fueron completadas en Sprint anterior.

Tabla N° 12.- Sprint Backlog. Sprint 3

Backlog ID	Módulo	Tarea	Estado	Responsable	Estimación de tiempo (días)	Porcentaje de Avance
10		User CNE imprime credenciales para observadores independientes		Franco Iturrizaga	5	60%
17	Observadores Invitados	Observador Invitado completa Fichas técnica A y B. Podrá llenar varias fichas técnicas A.		Marissa Ticona	4	80%
18		Admin CNE y User CNE visualizan respuestas de fichas técnicas A y B	New	Marissa Ticona	3	
23		Admin CNE crea Grupos de observadores invitados	Bug	Franco Iturrizaga	5	
24		Observador CNE adjunta informes al grupo asignado	Bug	Franco Iturrizaga	5	
25		User CNE/Admin CNE generan Reportes por países	New	Marvin Meza	3	
26	User CNE/Admin CNE generan Reportes por tipos de observadores	New	Marvin Meza	3		
27	User CNE/Admin CNE generan Reportes por	New	Marvin Meza	3		

		sexo				
28		User CNE/Admin CNE generan Reportes por organización	New	Máximo Romero	3	
29		Permite seleccionar tipo de gráfico: lineal, barras o de porciones	New	Máximo Romero	3	

Fuente: Elaborado por la autora.

### **3.2.3.1.2 DEFINICIÓN DEL SPRINT GOAL**

El objetivo que define el Sprint Goal que el equipo técnico desarrolló, consiste en completar todas las funcionalidades descritas en el Sprint Backlog dentro de 1 mes, además el equipo compromete a tener listo un despliegue para pruebas en QA durante la primera semana del mes ya que son conscientes de que la entrega en producción se aproxima.

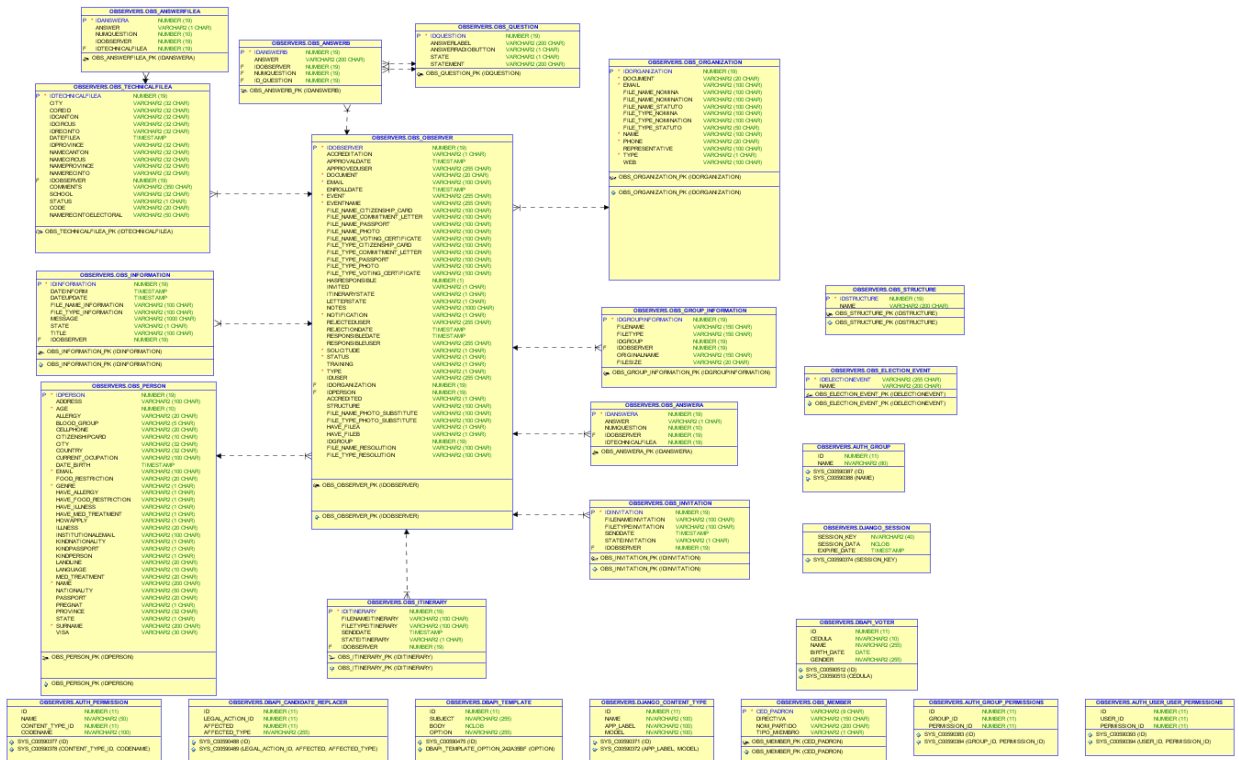
### **3.2.3.2 INCREMENTO DEL SPRINT**

Durante la última iteración, se desarrolló todo el aplicativo completo. Se resolvieron los Bugs de Tipo B y C que no figuraban en la pila del producto, ya que no eran de funcionalidad, pero faltaba realizar algunas validaciones:

- **Incremento en Base de Datos**

Se implementaron todas las entidades necesarias para que el sistema funcione correctamente.

Figura N°20.- Incremento en BD- Sprint 3



Fuente: Scytl Perú.

• **Incremento en funcionalidad**

Con cada iteración se van incrementando las entregas para completar el producto, se incrementó la programación en Back End y en Front End, con la creación de nuevas líneas de códigos. Se analizará las funcionalidades atendidas en la revisión del Sprint.

Se completó la pila del Sprint y también la pila del producto, implementándose todas las funcionalidades requeridas.

**3.3 REVISIÓN Y CONSOLIDACIÓN DE RESULTADOS**

Al finalizar el tercer Sprint, se procedió a entregar el aplicativo a los usuarios para confirmar la entrega con todas las funcionalidades solicitadas. Durante la

implementación del sistema se brindó el soporte correspondiente, asignando a dos personas para subsanas los inconvenientes que pudieran surgir.

La aplicación web inicia con la siguiente interfaz de usuario:

Figura N° 22.- Login



A continuación se presenta la visualización de las pantallas de los procesos representados en los casos de uso:

- **Gestión Postulación Observador**

Figura N° 23 - Gestión Postulación Observador



Fuente: ScytI Perú.

- **Gestión de Registro de Independiente**

Figura N° 24.- Gestión de Registro de Independiente

CONSEJO NACIONAL ELECTORAL SISTEMA ADMINISTRATIVO ELECTORAL							
Observadores Electorales		Español		Rony Villanueva			
<b>Listado de Solicitudes</b>							
<b>Detalles de Solicitudes</b>							
Búsqueda Avanzada							
Mostrar: 5							
Solicitud	Observador	País	Fecha	Capacitación	Estado	Acreditación	Usuario
00025 - PNN	Marisol Aguirre	ECUADOR	10-12-2013		Rechazado		
00026 - PNN	Gerardo Zamora	ECUADOR	10-12-2013	Capacitado	Aprobado	Acreditación Enviada	
00028 - PNE	Vivi Rivasplata	COSTA RICA	10-12-2013		Rechazado		
00031 - PNN	Steve HUYGA	ECUADOR	10-12-2013	Capacitado	Aprobado		
00032 - PNN	Marianela Alayo	ECUADOR	10-12-2013		Aprobado		
11 - 15 de 146 entradas				<input type="button" value="«"/> <input type="button" value="&lt;"/> <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="5"/> <input type="button" value="&gt;"/> <input type="button" value="»"/>			

Fuente: ScytI Perú.

- **Gestión Observador Invitados-Generación**

Figura N° 25.- Gestión Observador Invitados-Generación

Lista de Observadores Invitados									
<a href="#">+ Añadir Observador</a>									
Búsqueda Avanzada									
Mostrar: 5									
Código identificador	Nombre y Apellidos	Evento Electoral	Nombre de Organización	País	Estado	Carta	Itinerario	Editar	Eliminar
00066	Anita Giovana Jirafales Danes	Febrero 2014	Dk Hack Inn	PERU	Invitado	Enviada	Pendiente		
00067	Marianne Bludau	Febrero 2014	Dk Hack Inn	SUDAFRICA	Invitado	Enviada	Pendiente		
00070	Heidi Catherine Castro Avalos	Febrero 2014	Dk Hack Inn	MEXICO	Invitado	Enviada	Pendiente		
00071	Anita Giovana Jirafales Danes	Febrero 2014	Dk Hack Inn	PERU	Invitado	Enviada	Pendiente		
00072	Anita Giovana Jirafales Danes	Febrero 2014	Dk Hack Inn	PERU	Invitado	Enviada	Pendiente		
1 - 5 de 11372 entradas				<input type="button" value="«"/> <input type="button" value="&lt;"/> <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="5"/> <input type="button" value="&gt;"/> <input type="button" value="»"/>					

Fuente: ScytI Perú.

- **Gestión de Itinerarios de vuelo**

Figura N° 26.- Gestión de Itinerarios de vuelo

**Invitar Observador** Itinerarios de Vuelos

---

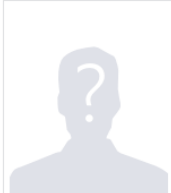
**Formulario de Invitación**

<b>Estado de Invitación</b>	<b>Estado de Itinerarios de Vuelos</b>
<input checked="" type="checkbox"/> Pendiente de Envío <input checked="" type="checkbox"/> Carta Enviada <input type="checkbox"/> Invitación Aceptada <input type="checkbox"/> Invitación Rechazada	<input type="checkbox"/> Aceptada <input type="checkbox"/> Rechazado

Evento Electoral: **Febrero 2014**

---

**Datos Personales**

Nombres:	<b>Anita Giovana</b>	<b>Foto</b> 
Apellidos:	<b>Jirafales Danes</b>	
Nacionalidad:	<b>peruana</b>	
País:	<b>PERU</b>	
Fecha de Nacimiento:	<b>08/05/1989</b>	
Nro. Pasaporte:	<b>4474747858</b>	

Fuente: ScytI Perú.

- **Gestión de Fichas Técnicas**

Figura N° 27.- Gestión de Fichas Técnicas B

**Lista de Fichas Técnicas B**

Mostrar:

Observador/a	Estructura	Evento Electoral	País
<b>someone someone</b>	ORGANIZACIÓN DE ESTADOS AMERICANOS	Febrero 2014	PARAGUAY
<b>Mauricio Matos</b>	Dk Hack Inn	Febrero 2014	ARGENTINA
<b>dd dd</b>	Dk Hack Inn	Febrero 2014	PERU
<b>rr rr</b>	ORGANIZACIÓN DE ESTADOS AMERICANOS	Febrero 2014	COLOMBIA
<b>name name</b>	Dk Hack Inn	Febrero 2014	BOLIVIA

1 - 5 de 20 entradas < < 1 2 3 4 > >



Figura N° 28.- Gestión de Fichas Técnicas A

**Lista de Fichas Técnicas A**

Mostrar: 5 ▼

Solicitud	Observador/a	Estructura	Evento Electoral	País
000001	MARISSA TICONA	Dk Hack Inn	Febrero 2014	PERU
000002	MARISSA TICONA	Dk Hack Inn	Febrero 2014	PERU
000003	MARISSA TICONA	Dk Hack Inn	Febrero 2014	PERU
000001	Denisse Huaman	ORGANIZACIÓN DE ESTADOS AMERICANOS	Febrero 2014	PERU
000003	Jerau Jimenex	ORGANIZACIÓN DE ESTADOS AMERICANOS	Febrero 2014	PARAGUAY

1 - 5 de 16 entradas

Fuente: ScytI Perú.

- **Gestión de informes grupales**

Figura N° 29.- Gestión de informes grupales

**Informes Grupales**

**Lista de Grupos** + Crear Grupo

Búsqueda Avanzada ▼

Mostrar: 5 ▼

Código identificador	Nombre de grupo	Fecha de creación	Nro. Usuarios	Infomes adjuntos	Borrar
121	Los Galacticos	09/12/2013	1	SI	
141	lonely	09/12/2013	1	SI	
101	Q.A.	09/12/2013	1	SI	
321	fasagzdghds	16/12/2013	1	SI	
381	Grupo Individualista	17/12/2013	1	SI	

1 - 5 de 55 entradas

Fuente: ScytI Perú.

- **Generación de Reportes**

Figura N° 30.- Generación de reportes

**Crear Reportes**

**Elegir Datos**

Crear reporte por :

**Customizar del Gráfico**

Tipo de Gráfico

Descripción

Fuente: ScytI Perú.



- No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software.
- Las metodologías tradicionales históricamente han intentado abordar la mayor cantidad de situaciones del contexto del proyecto, exigiendo un esfuerzo considerable para ser adaptadas, sobre todo en proyectos pequeños y de requisitos cambiantes.
- Las metodologías ágiles ofrecen una solución casi a medida para una gran cantidad de proyectos.
- Las metodologías ágiles se caracterizan por su sencillez, tanto en su aprendizaje como en su aplicación; sin embargo, gozan tanto de ventajas como de inconvenientes.
- Las metodologías ágiles permiten a los pequeños grupos de desarrollo concentrarse en la tarea de construir software fomentando prácticas de fácil adopción y en un entorno ordenado que permiten que los proyectos finalicen exitosamente
- La aplicación de la metodología Scrum aplicado a la implementación de este aplicativo facilitó los requerimientos sean atendidos de manera rápida y ajustable, gracias a la constante comunicación entre todas las partes del proyecto.
- El trabajo de investigación presentado, cumplió con realizar en cada iteración las fases correspondientes para finalizar con el desarrollo del aplicativo tal y como lo sugiere la metodología Scrum.
- El desarrollo del aplicativo cumplió con todas las especificaciones solicitadas por el cliente, de esta manera cumplió con el objetivo de la presente investigación.



- Se recomienda antes de aplicar la metodología, capacitar a los integrantes del equipo desde tiempo antes de comenzar a desarrollar el producto ya que los procedimientos deben ser entendidos e internalizados por todos los que participan para que pueda llevarle el proyecto con mayor compromiso y responsabilidad.
- Se recomienda definir lo más detalladamente la pila del producto ya que de eso depende el tiempo que el equipo le dedica al aprendizaje y puede dedicarse al desarrollo del producto cuanto antes.
- Se recomienda establecer canales de comunicación efectivos para la comunicación si no se cuenta con los usuarios de manera presencial durante todo el desarrollo del producto, ya que de no ser así el proyecto se retrasa cuando debe parar para esperar la confirmación del usuario con respecto a alguna duda funcional.
- Se recomienda contar con un equipo especializado para realizar las pruebas respectivas al producto más de una vez por iteración, si la iteración es mayor a 15 días, ya que de esta manera se evita retroceder en funcionalidades que ya se habían completado en iteraciones anteriores.



[1] Boehm, B.: A View of 20<sup>th</sup> and 21<sup>st</sup> Century Software Engineering. In: 28th international conference on Software engineering, pp. 12--29. Shanghai, China (2006)

[2] Dalcher, D., Benediktsson, O., y Thorbergsson, H., “Development Life Cycle Management: A Multiproject Experiment”, Proceedings of the 12th International Conference and Workshops on the Engineering of Computer Based Systems (ECBS’05), 2005

[3] Wellington, A., Briggs, T., y Girard, C.D., “Comparison of Student Experiences with Plan-Driven and Agile Methodologies”, Proceedings of the 35th ASEE/IEEE Frontiers in Education Conference, 2005.

Información de referencia tomada de las siguientes direcciones:

<http://www.proyectosagiles.org/requisitos-de-scrum>

<http://www.dosideas.com/noticias/metodologias/981-scrum-en-1-sola-pagina.html>

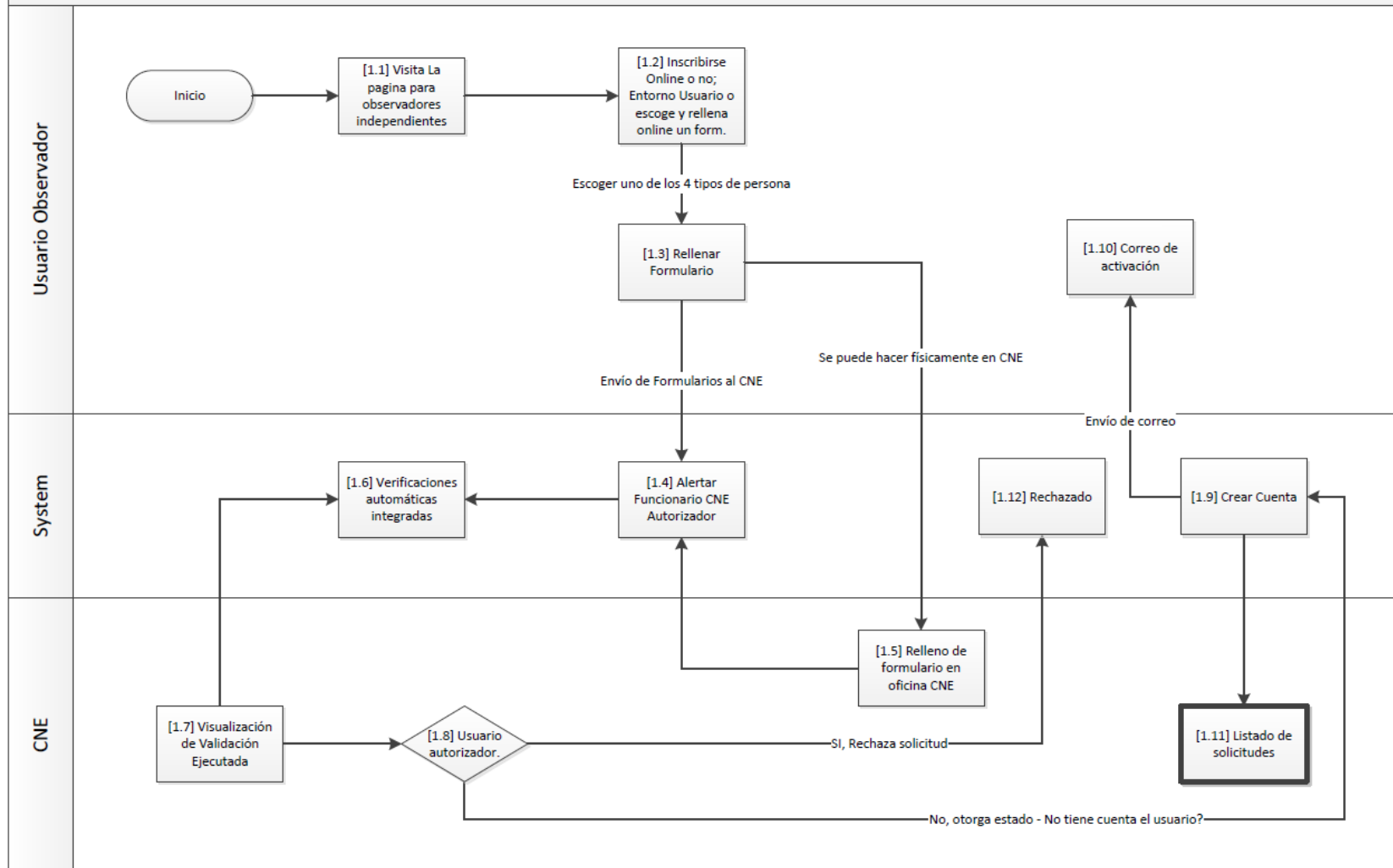
<http://www.pmoinformatica.com/2012/09/scrum-grandes-proyectos-reunion.html>



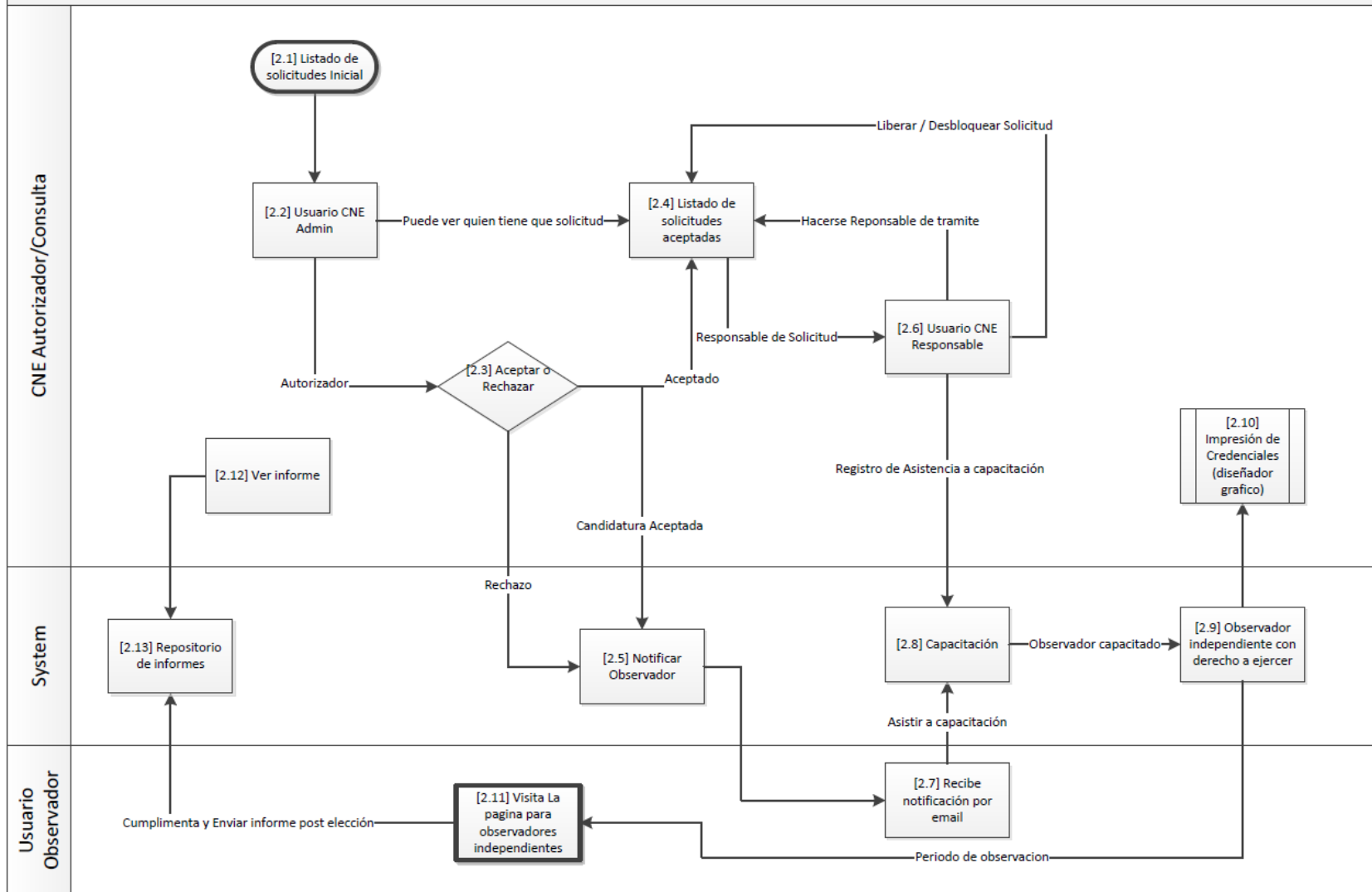
## **7. ANEXOS**

**ANEXO 1.- DIAGRAMA DE ACTIVIDADES DEL PROCESO DE OBSERVADORES ELECTORALES IDENTIFICADOS EN EL DIAGRAMA DE CASOS DE USO**

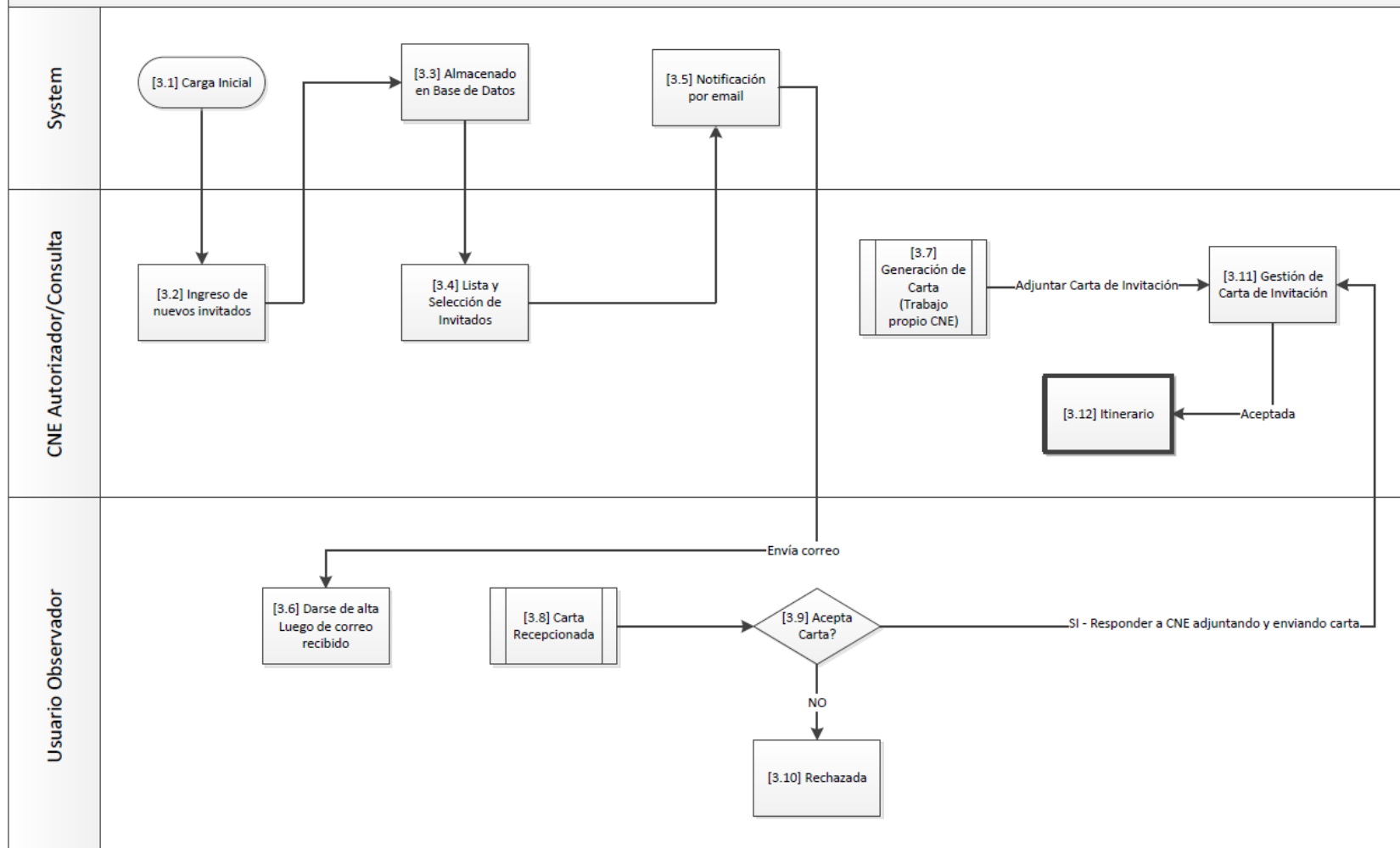
[1.0] Observadores Electorales

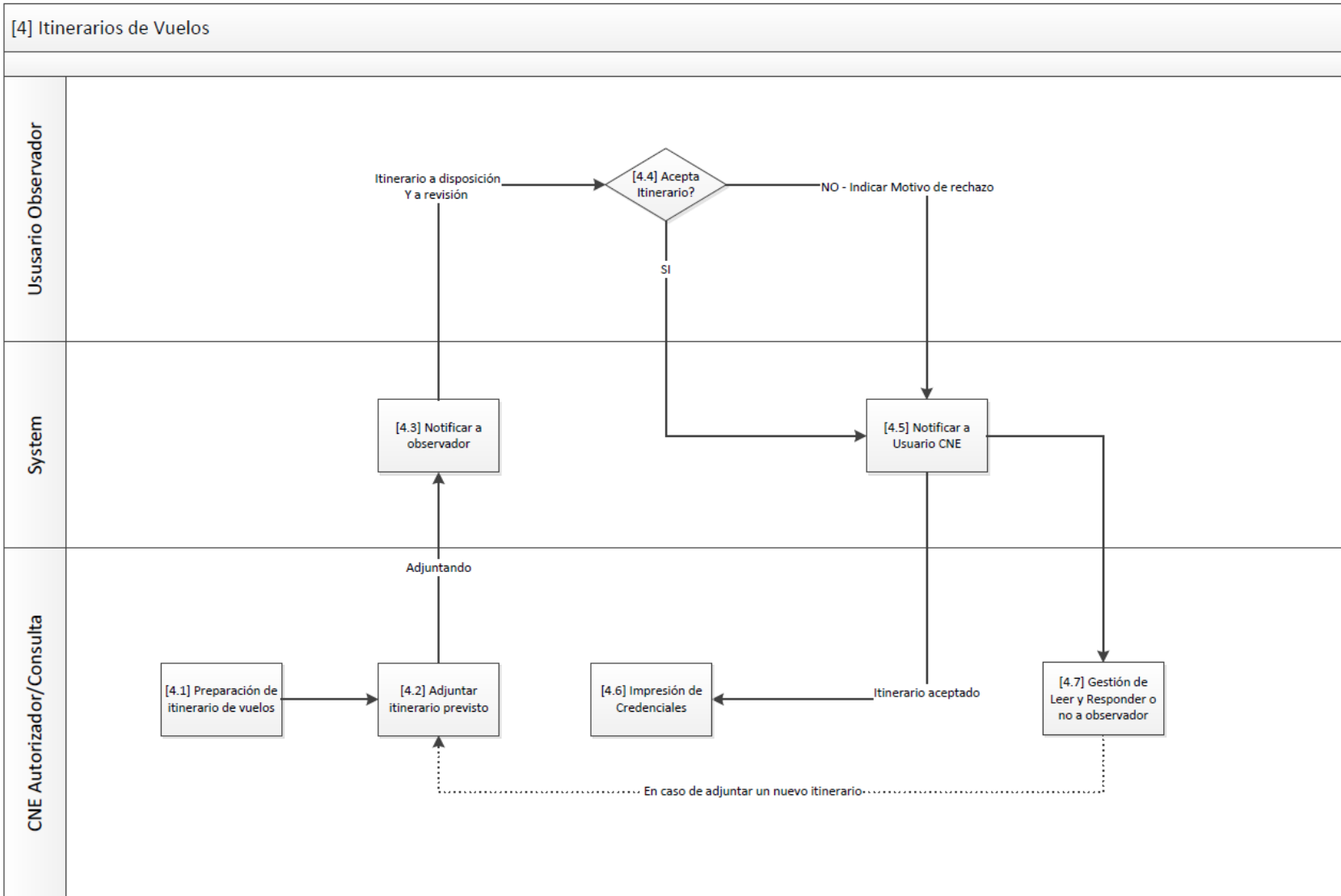


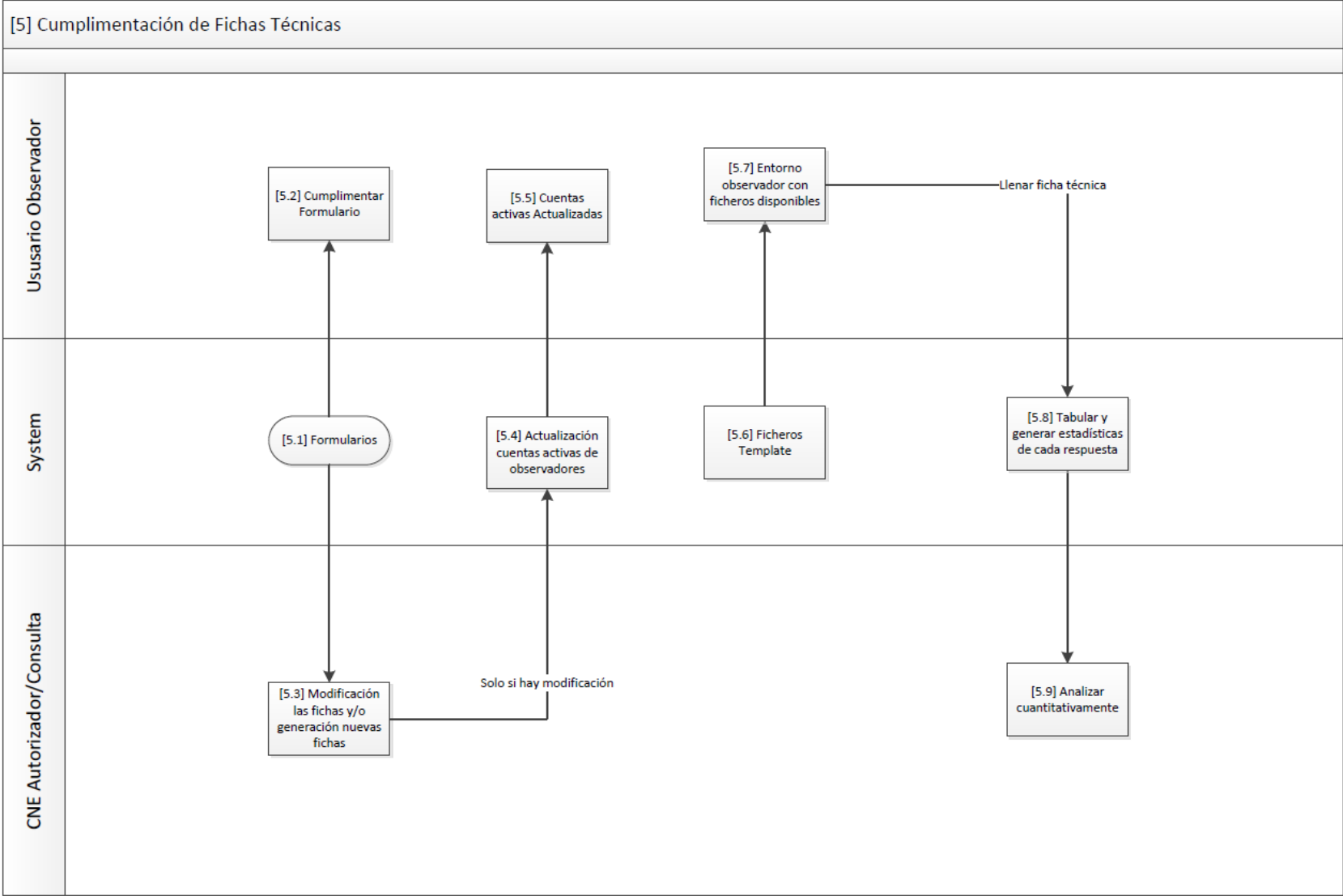
[2] Gestionar Registro de independientes

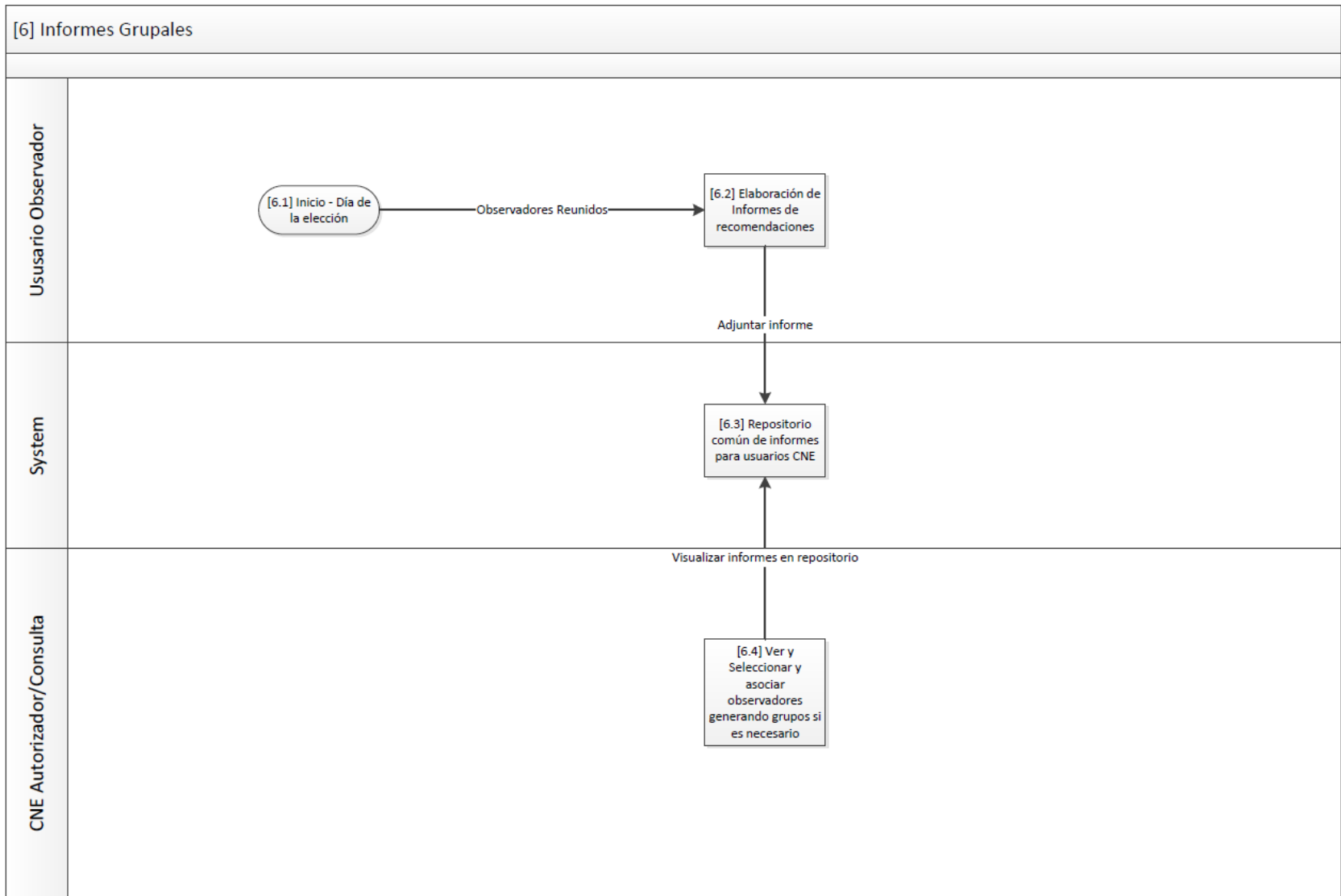


[3] Observadores Invitados - Generación

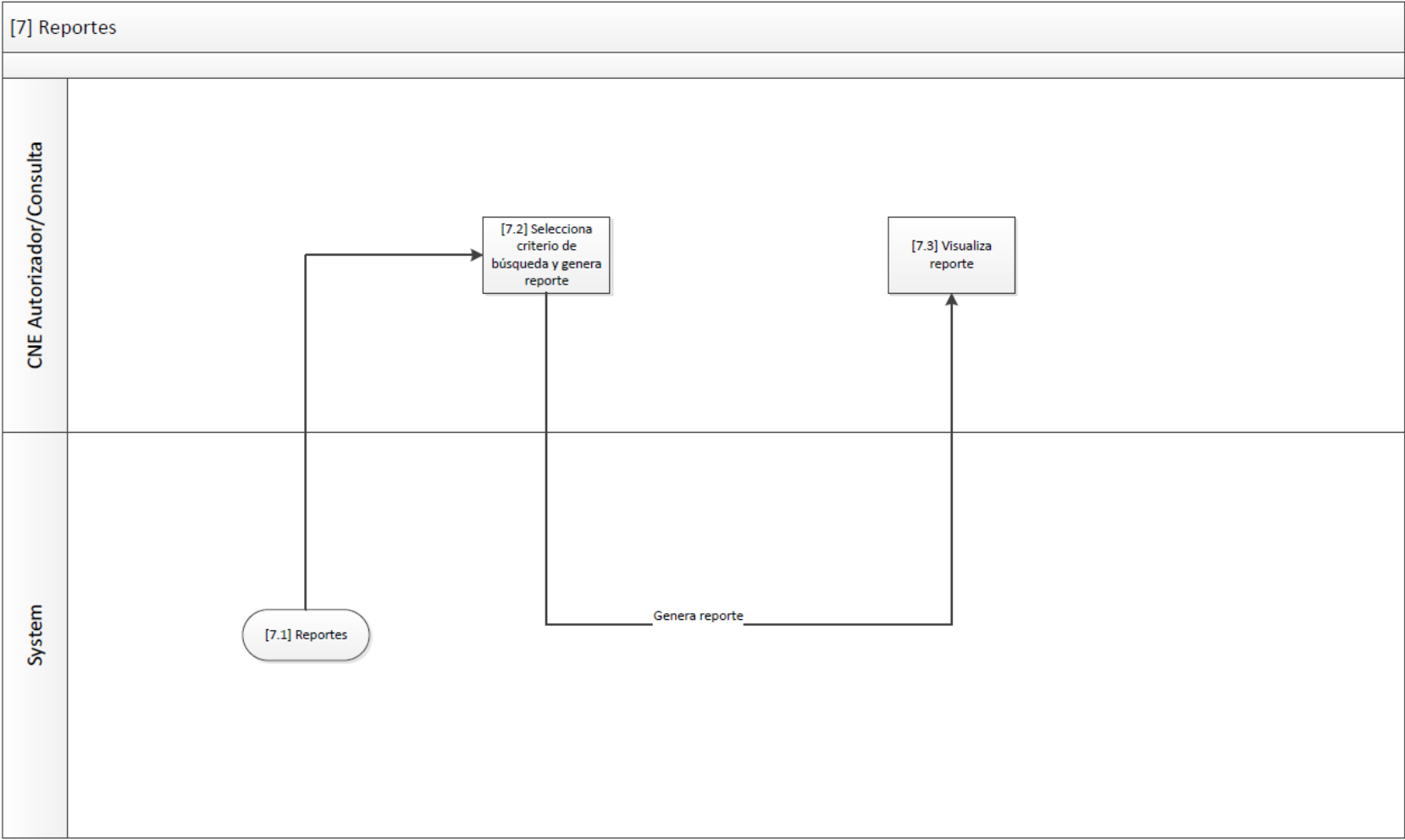












## **ANEXO 2.- EJEMPLOS DE BUGS DOCUMENTADOS EN REDMINE**

## Bug #68693

Update Log time Watch Copy

### [Tipo: A] Nombre de usuario oculto - Observador Invitado

« Previous | 22 of 275 | Next »

Added by Heidi Castro 17 days ago. Updated 15 days ago.

<b>Status:</b>	Assigned	<b>Start date:</b>	29-01-2014
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Rony Villanueva	<b>% Done:</b>	<div style="width: 0%; background-color: #ccc; border: 1px solid #ccc;"></div> 0%
<b>Category:</b>	-	<b>Spent time:</b>	-
<b>Target version:</b>	-		
<b>Importance:</b>		<b>Done in revision:</b>	0
<b>Estimation Points:</b>		<b>Done in build:</b>	0
<b>Error Classification:</b>	A	<b>Verified in build:</b>	
<b>Found in build:</b>	0	<b>Ecuador Team:</b>	

#### Description

Quote

- Al presionar F5 dentro de la pantalla Lista de Fichas Técnicas A, no se muestra el nombre del usuario.

ficha-técnica-A-Observador-Invitado.PNG (32 KB) Heidi Castro, 30-01-2014 10:41 pm

#### Subtasks

Add

#### Related issues

Add

#### History

Updated by Fredy Huaman 16 days ago

#1

## Bug #69034

Update Log time Watch Copy

[Tipo: B] Error al visualizar solicitudes de observadores independientes en cualquier estado - iniciar sesión como Administrador [« Previous](#) | 17 of 275 | [Next »](#)

Added by Heidi Castro 14 days ago.

<b>Status:</b>	New	<b>Start date:</b>	31-01-2014
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Rony Villanueva	<b>% Done:</b>	<div style="width: 0%; background-color: #ccc; border: 1px solid #ccc;"></div> 0%
<b>Category:</b>	-	<b>Spent time:</b>	-
<b>Target version:</b>	-	<b>Done in revision:</b>	
<b>Importance:</b>		<b>Done in build:</b>	
<b>Estimation Points:</b>		<b>Verified in build:</b>	
<b>Error Classification:</b>	A	<b>Ecuador Team:</b>	
<b>Found in build:</b>	0		

### Description

Quote

- Se procede a visualizar una solicitud en cualquier estado (Aprobado, Rechazado o Pendiente), luego se hace clic en la opción Volver y se visualiza que el sistema retorna a el listado de solicitudes y en la parte superior en la ruta se concatena el id de la solicitud visualizada, lo cual es incorrecto como se aprecia en la imagen **id-listado-observadores-independientes.PNG**

id-listado-observadores-independientes.PNG (54.7 KB) Heidi Castro, 31-01-2014 07:10 pm

### Subtasks

Add

### Related issues

Add

Update Log time Watch Copy

### **ANEXO 3.- EJEMPLOS DE USER STORY DOCUMENTADOS EN REDMINE**

## User Story #53925

[Update](#) [Log time](#) [Watch](#) [Copy](#)

High-level Requirement #53511: [SIMP] Observadores Electorales  
Requirement #53905: [2] Gestionar Registro de independientes

[« Previous](#) | 24 of 52 | [Next »](#)

**[UC2] [R2.9] [US1] [FE] Como Usuario CNE quiero disponer de una interfaz de usuario para realizar la impresión de credenciales de observadores electorales independientes.**

Added by Amil Hasanbegovic 5 months ago. Updated 5 months ago.

<b>Status:</b>	Assigned	<b>Start date:</b>	15-08-2013
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Rony Villanueva	<b>% Done:</b>	<div style="width: 0%;"><div style="width: 0%;"></div></div> 0%
<b>Category:</b>	-	<b>Spent time:</b>	-
<b>Target version:</b>	-		
<b>Importance:</b>		<b>Done in revision:</b>	
<b>Estimation Points:</b>		<b>Done in build:</b>	
<b>Test Case Created:</b>	No	<b>Verified in build:</b>	
<b>Test Case Number:</b>			

### Description

[Quote](#)

## User Story Funcional Back-End

### Definición de la User Story:

Como Usuario CNE

quiero disponer de una interfaz de usuario

para realizar la impresión de credenciales de observadores electorales independientes.

### Capa de presentación:

#### Descripción Detallada:

La capa de presentación requiere una interfaz de usuario que permita mostrar los Observadores Electorales que hayan cumplimentado las actividades previas de NOTIFICACION, CAPACITACION, ACREDITACION (es decir todos los checks de actividades marcados en su formulario de solicitud) para realizar la generación de credenciales.

Cada usuario CNE solo tendrá disponible la impresión de las credenciales de los observadores que el sea responsable.

La intefaz presentará los siguientes datos:

#### Filtros:

- Tipo de Credencial (DropDown List, indicando Observador Independiente y Observador Internacional)
- Nombre de Observador (solo mostrará los que han cumplimentado todo el proceso)

Existen 2 tipos de credenciales: Observador Independiente / Observador Internacional

Se tomará una imagen de fondo proporcionada por el CNE, a la cual se agregará la foto ingresada por el observador, el nombre y apellido del observador, si es OBSERVADOR INDEPENDIENTE O INVITADO (INTERNACIONAL) y el número de documento de identidad (PASAPORTE en caso de Internacionales). Y para los observadores que pertenecen a una entidad jurídica se debe imprimir el nombre de la organización.

La credencial se genera en formato PDF. Se usará como base la imagen en JPG enviada por el CNE (archivo Ejemplo de

## User Story #62151

 Update  Log time  Watch  Copy

High-level Requirement #49275: Observadores Electorales  
Requirement #49461: [7] Reportes

### [R7][US1][BE][FE] Como usuario CNE quiero disponer de una interfaz para generar REPORTES ESTADISTICOS

Added by Rodolfo Villaorduna 3 months ago. Updated 2 months ago.

<b>Status:</b>	Assigned	<b>Start date:</b>	22-11-2013
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Marvin Meza	<b>% Done:</b>	<div style="width: 70%;"><div style="background-color: #4CAF50; width: 70%;"></div></div> 70%
<b>Category:</b>	-	<b>Spent time:</b>	-
<b>Target version:</b>	-		
<b>Importance:</b>		<b>Done in revision:</b>	
<b>Estimation Points:</b>		<b>Done in build:</b>	
<b>Test Case Created:</b>	Yes	<b>Verified in build:</b>	
<b>Test Case Number:</b>	C273697, C273698, C273699	<b>Ecuador Team:</b>	

#### Description Quote

### User Story Funcional Back-End

#### Definición de la User Story:

Como Usuario CNE

quiero disponer de una interfaz de usuario

para generar REPORTES ESTADÍSTICOS

#### Capa de Negocio:

##### Descripción Detallada:

a. El usuario CNE da click en el botón CREAR REPORTE

La pantalla mostrará los siguientes datos:

SECCION: Elegir Datos

Permite al usuario CNE seleccionar los datos base del reporte a generar.

Seleccionar una opción para crear reporte:

- País

Organización

-Sexo

-Tipo de Observador

Asistió a capacitación

-Remitieron Informe

--- Validación: Solo se permite escoger uno de los filtros a la vez, es decir la selección es excluyente.

SECCION: Customizar el gráfico

--> Tipo de Gráfico: Barra, Lineal, De Porciones (Pye).

--> Descripción: Texto Libre [alfanumérico][Long. Máxima 150 caracteres]

Botón CREAR REPORTE:

--> Genera el reporte en una venta nueva según los datos seleccionados.