

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



TESIS DE GRADO

**"WORKFLOW COLABORATIVO BASADO EN SCRUM
DISTRIBUIDO PARA TELETRABAJO IMPLEMENTADO EN
CLOUD COMPUTING"**

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA

MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

POSTULANTE: JORGE RENATO FERNÁNDEZ ROJAS

TUTORA METODOLÓGICA: LIC. MENFY MORALES RÍOS

ASESORA: LIC. CELIA ELENA TARQUINO PERALTA

LA PAZ – BOLIVIA

2015



UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA

A mi padre, por apoyarme incondicionalmente.

A mis tíos, por aconsejarme apoyarme y cuidarme como a un hijo.

A mis docentes, por enseñar lecciones de vida además de enseñarme la profesión.

A esa amiga incondicional, por apoyarme y compartir lo bueno y lo malo, gracias.

A todos mis compañeros, por sus sonrisas y la alegría que supieron transmitirme.

AGRADECIMIENTOS

Agradecer ante todo a Dios, por hacer posible que mi vida tomara un nuevo rumbo hacia una vida académica inspirada en la superación.

Agradecer a mi padre, por apoyarme con mis metas y en especial con la de concluir con esta etapa especial de preparación en mi vida.

Agradecer a mi docente tutora M.SC. Menfy Morales Ríos, por dedicar su tiempo en orientarme hacia la conclusión de mi tesis de manera exitosa.

Gracias a mi docente revisora M.SC. Celia Tarquino Peralta, por tener paciencia en los momentos en que perdí el norte en la realización de mi tesis, y aconsejarme siempre en un sentido positivo de la manera de hacer las cosas de manera correcta.

Agradecer a los docentes de informática por brindarme sus conocimientos y en especial al docente M.SC. Aldo Ramiro Valdez Alvarado por su motivación, ideas y consejos que fueron el combustible para entender que es lo que quería para realizar mi tesis de manera correcta.

RESUMEN

La presente tesis trata de la implementación de un sistema colaborativo del tipo workflow, cuyo objetivo es el de modelar los procesos y flujos de trabajo de la metodología ágil scrum para su fase de análisis.

Los workflow son modelos de sistemas, cuyo objetivo es el de facilitar el trabajo en una organización implementando el teletrabajo entre los miembros de la misma, ahorrando tiempos y por ende costes, lo que hace que sea un tema de interés y explotación para los informáticos.

Para que una herramienta sea considerada adecuada para teletrabajar, debe ser fácil de adoptar por la organización en el sentido de acceso, por lo que los expertos de implementación del teletrabajo como la organización de telecomunicaciones CISCO aconsejan el uso de la computación en la nube.

En este trabajo se implementa al sistema colaborativo en la plataforma de desarrollo en la nube, de Google, denominada Google App Engine.

Los sistemas que se alojan en la nube de Google, tienen la particularidad de utilizar la misma infraestructura de las aplicaciones de google, además del mismo sistema de almacenamiento que es el Bigtable enmascarado por una base de datos llamada datastore, con una capacidad de almacenamiento considerable.

Con la implementación de la aplicación del presente trabajo se pretende aportar conocimiento, sobre el cómo se debe implementar software como servicio en la nube, además de entender las ventajas de implementar sistemas para teletrabajar, tanto para el informático como para la sociedad en general.

Palabras clave: Teletrabajo, Scrum, Computación en la nube, Google App Engine, Workflow Colaborativo

ABSTRACT

The present thesis is about the implementation of a collaborative system of the type workflow whose objective is the one of modeling the processes and flows of work of the methodology agile scrum for its analysis phase.

The workflow is model of systems whose objective is the one of facilitating the work in an organization implementing the teletrabajo among the members of the same one, saving times and for ende costs, that makes him to be a topic of interest and exploitation for the computer specialist.

So that a tool is considered appropriate for teletrabajos, it should be easy to adopt for the organization in the access sense, for what the experts of implementation of the teletrabajo like the organization of telecommunications CISCO advise the use of the computation in the cloud.

In this work it is implemented to the collaborative system in the development platform in the cloud, of Google, denominated Google App Engine.

The systems that lodge in the cloud of Google, have the particularity of using the same infrastructure of the google applications, besides the same storage system that is Bigtable masked by a database called datastore, with a capacity of considerable storage.

With the implementation of the application of the present work it is sought to contribute knowledge, on the how it should be implemented software like service in the cloud, besides understanding the advantages of implementing systems for teletrabajos, as much for the computer specialist as for the society in general.

Keywords: Teletrabajo, Scrum, Computation in the cloud, Google App Engine, Collaborative Workflow

ÍNDICE

DEDICATORIA.....	2
RESUMEN.....	4
ABSTRACT.....	5
ÍNDICE DE TABLAS	10
CAPITULO 1.....	1
MARCO INTRODUCTORIO.....	1
1.1 INTRODUCCIÓN	1
1.2 ANTECEDENTES.....	2
1.2.1 ANTECEDENTES DE LA TEMÁTICA	2
1.3 ANTECEDENTES DE TRABAJOS SIMILARES	6
1.4 PLANTEAMIENTO DEL PROBLEMA.....	7
1.4.1 ANTECEDENTES DE LA PROBLEMÁTICA.....	7
1.4.2 PROBLEMA CENTRAL	8
1.4.3 PROBLEMAS SECUNDARIOS	8
1.5 DEFINICIÓN DE OBJETIVOS	8
1.5.1 OBJETIVO GENERAL	8
1.5.2 OBJETIVOS ESPECÍFICOS	8
1.6 HIPÓTESIS.....	9
1.7 OPERACIONALIZACIÓN DE VARIABLES	9
1.7.2 VARIABLE INDEPENDIENTE:.....	9
1.7.3 VARIABLE DEPENDIENTE:	9
1.7.4 VARIABLES INTERVINIENTES:.....	9
1.8. JUSTIFICACIÓN	10
1.8.1 JUSTIFICACIÓN ECONÓMICA.....	10
1.8.2 JUSTIFICACIÓN SOCIAL	10
1.8.3 JUSTIFICACIÓN CIENTÍFICA.....	10
1.8.4 JUSTIFICACIÓN TÉCNICA.....	11
1.8.4.1 REQUERIMIENTOS DE HARDWARE.....	11
1.8.4.2 REQUERIMIENTOS DE SOFTWARE	11
1.9 ALCANCES Y LÍMITES	11
1.9.1 ALCANCES	11
1.9.2 LIMITES	12
1.10 APORTE.....	12
1.10.1 PRÁCTICO.....	12
1.10.2 TEÓRICO	13
1.11 METODOLOGÍA.....	13
1.11.1 MÉTODO CIENTÍFICO	13
1.11.1.1 MÉTODO DESCRIPTIVO	13
1.11.1.2 METODOLOGÍA PARA PROBAR LA HIPÓTESIS	14
1.11.2 METODOLOGÍAS DE INGENIERÍA	14
1.11.2.1 METODOLOGÍA DE DESARROLLO ÁGIL AUP.....	14
1.11.2.2 HERRAMIENTAS DE DESARROLLO GOOGLE APP ENGINE.....	15

CAPITULO 2	16
MARCO TEÓRICO.....	16
2.1 INTRODUCCIÓN	16
2.2 METODOLOGÍAS ÁGILES	17
2.2.1 DEFINICIÓN	17
2.2.2 IMPORTANCIA.....	17
2.2 LA METODOLOGÍA ÁGIL SCRUM.....	18
2.2.1 DEFINICIÓN.....	18
2.2.1.1 ROLES.....	18
a) SCRUM MÁSTER	19
i) DEFINICIÓN.....	19
ii) RESPONSABILIDADES	19
b) EQUIPO DE DESARROLLO	19
i) DEFINICIÓN.....	19
ii) RESPONSABILIDADES	19
c) PROPIETARIO DEL PRODUCTO	20
i) DEFINICIÓN.....	20
ii) RESPONSABILIDADES	20
d) OTROS INTERESADOS.....	20
2.2.1.2 ARTEFACTOS	20
a) PILA DEL PRODUCTO.....	20
i) CONSTRUCCIÓN	20
ii) VISIÓN GENERAL DEL PROYECTO	21
iii) CONSTRUCCIÓN Y PREPARACIÓN.....	21
iv) FORMATO.....	21
b) SPRINT.....	22
i) PILA DEL SPRINT	22
ii) FORMATO DE LA PILA DEL SPRINT	23
iii) INCREMENTO.....	24
c) MÉTRICAS AGILES	24
i) GRAFICO DEL PRODUCTO (BURN-UP).....	24
ii) GRAFICO DEL SPRINT (BURN-DOWN).....	24
iii) JUEGO DE DECISIÓN "ESTIMACIÓN DE PÓKER"	25
2.2.1.3 EVENTOS.....	26
a) REUNIÓN DE PLANIFICACIÓN DEL SPRINT	26
i) DESCRIPCIÓN GENERAL	26
ii) PRECONDICIONES	26
iii) ENTRADAS	26
iv) RESULTADOS	26
v) FORMATO	27
vi) PARTICIPANTES	27
vii) SOPORTE	27
b) REUNIÓN DIARIA DE SEGUIMIENTO DEL SPRINT.....	28
i) DESCRIPCIÓN GENERAL	28
ii) PRECONDICIONES	28

iii) ENTRADAS	28
iv) RESULTADOS	28
v) FORMATO	28
vi) PARTICIPANTES	29
vii) SOPORTE	29
c) REUNIÓN DE RETROSPECTIVA	29
i) DESCRIPCIÓN GENERAL	29
ii) PRECONDICIONES	29
iii) ENTRADAS	29
iv) RESULTADOS	29
v) FORMATO	29
vi) PARTICIPANTES	30
vii) SOPORTE	30
2.3 HERRAMIENTAS COLABORATIVAS PARA SCRUM	30
2.3.1 KUNAGI.....	30
2.3.2 SCRUMDO	31
2.3.3 SPRINTOMETER.....	31
2.4 CRITERIOS DE EVALUACIÓN PARA LAS HERRAMIENTAS COLABORATIVAS COMO BASE DE COMPARACIÓN PARA EL SISTEMA A IMPLEMENTAR	31
2.4.1 CRITERIOS DE EVALUACIÓN PARA HERRAMIENTAS DE TELETRABAJO	32
2.4.1.1 CRITERIO DE ACCESIBILIDAD.....	32
2.4.1.2 CRITERIO DE SEGURIDAD	32
2.4.1.3 CRITERIOS DE COMUNICACIÓN	32
2.4.2 CRITERIOS DE EVALUACIÓN PARA HERRAMIENTAS COLABORATIVAS	32
2.5 SCRUM DISTRIBUIDO.....	34
2.5.1 DEFINICIÓN.....	34
2.5.2 TIPOLOGÍA DE DISTRIBUCIÓN	35
2.5.2.1 EQUIPOS DISTRIBUIDOS PARCIALMENTE SIMPLES	35
2.5.2.2 EQUIPOS DISTRIBUIDOS PARCIALMENTE COMPLEJOS	35
2.5.2.3 EQUIPOS DISTRIBUIDOS TEMPORALMENTE SIMPLES	35
2.5.2.4 EQUIPOS DISTRIBUIDOS TEMPORALMENTE COMPLEJOS	35
2.5.3 ROLES DISTRIBUIDOS	35
2.5.3.2 SCRUM MANAGER	35
a) DEFINICIÓN.....	35
b) RESPONSABILIDADES	35
2.5.3.2 EQUIPO DE DESARROLLO	36
a) DEFINICIÓN.....	36
b) RESPONSABILIDADES	36
2.5.3.3 PROPIETARIO DEL PRODUCTO	36
a) DEFINICIÓN	36
b) RESPONSABILIDADES	36
2.5.3.4 STAKEHOLDERS	37

2.5.4 EVENTOS DISTRIBUIDOS	37
2.5.5 ARTEFACTOS EN AMBIENTES DISTRIBUIDOS	37
2.5.6 HERRAMIENTAS DE COMUNICACIÓN COMO ESTRATEGIA	37
2.6 METODOLOGÍA ÁGIL DE DESARROLLO DEL SISTEMA AUP	38
2.6.1 DEFINICIÓN.....	38
2.6.2 FASES DE AUP.....	38
2.6.2.1 INICIO	39
2.6.2.2 ELABORACIÓN	40
2.6.2.3 CONSTRUCCIÓN	40
2.6.2.4 TRANSICIÓN	40
2.6.3 ARTEFACTOS	40
2.7 SISTEMAS WORKFLOW.....	41
2.7.1 DEFINICIÓN.....	41
2.7.1.1 FUNCIONES	41
2.7.2 CLASIFICACIÓN	41
2.7.2.1 WORKFLOW DE PRODUCCIÓN	42
2.7.2.2 WORKFLOW DE COLABORACIÓN	42
2.7.2.3 WORKFLOW DE ADMINISTRACIÓN.....	42
2.8 MODELADO DE PROCESOS DE UN WORKFLOW CON BPMN.....	42
2.8.1 MODELOS DE PROCESOS EN BPMN	44
2.8.1.1 PROCESOS DE NEGOCIO PRIVADOS	44
2.8.1.2 PROCESOS DEL NEGOCIO ABSTRACTOS.....	44
2.8.1.3 PROCESOS DE COLABORACIÓN.....	45
2.8.2 DIAGRAMAS BPMN.....	45
2.8.2.1 ELEMENTOS BÁSICOS DE LOS DIAGRAMAS BPMN	45
a) OBJETOS DE FLUJO (FLOW OBJECTS).....	45
i) EVENTOS (EVENTS)	45
ii) ACTIVIDADES (ACTIVITY)	47
iii) PASARELAS (GATEWAY).....	48
b) CONECTORES (CONNECTING OBJECTS)	49
i) FLUJO DE SECUENCIA	49
ii) FLUJO DE MENSAJE	49
iii) ASOCIACIÓN	49
c) CALLES (SWIMLANES)	49
i) LANE ("EN LÍNEA")	50
ii) POOL ("AGLUTINADOR")	50
d) ARTEFACTOS (ARTIFACTS)	50
2.9 TELETRABAJO	50
2.9.1 DEFINICIÓN.....	51
2.9.1.1 DEFINICIÓN SEGÚN LA OIT	51
2.9.1.2 DEFINICIÓN SEGÚN FUNDETIC	51
2.9.1.3 DEFINICIÓN SEGÚN EL LIBRO BLANCO	51
2.9.2 ESTADO DE LA CIENCIA.....	51
2.9.2.1 DESDE EL ASPECTO SOCIAL	51
2.9.2.2 DESDE EL ASPECTO LEGAL	52

2.9.2.3 DESDE EL ASPECTO TECNOLÓGICO	52
2.9.3 TELETRABAJO EN BOLIVIA	52
2.9.4 MODALIDADES DE TELETRABAJO.....	52
2.9.4.1 TELETRABAJO AUTÓNOMO	52
2.9.4.2 TELETRABAJO SUPLEMENTARIO	53
2.9.4.3 TELETRABAJO MÓVIL	53
2.9.5 IMPLEMENTACIÓN DEL TELETRABAJO EN UNA ORGANIZACIÓN	53
2.9.5.1 REQUERIMIENTOS PARA LA IMPLEMENTACIÓN.....	53
a) COMPROMISO INSTITUCIONAL.....	54
b) PLANEACIÓN DEL PROYECTO	54
c) AUTO EVALUACIÓN	55
d) PRUEBA PILOTO	55
e) APROPIACIÓN	55
2.9.6 COMPONENTE TECNOLÓGICO.....	56
2.9.7 IDENTIFICACIÓN DE SOLUCIONES TECNOLÓGICAS	57
2.10 EL TELETRABAJO BAJO LA METODOLOGÍA SCRUM DISTRIBUIDO	57
2.10.1 MODALIDADES DE TELETRABAJO	57
2.10.1.1 TELETRABAJO SUPLEMENTARIO.....	57
2.10.1.2 TELETRABAJO PERMANENTE.....	58
2.10.2 LA IMPLEMENTACIÓN DEL TELETRABAJO.....	58
2.10.3 IDENTIFICACIÓN DE LAS SOLUCIONES TECNOLÓGICAS.....	58
2.11 CLOUD COMPUTING.....	59
2.11.1 CARACTERÍSTICAS DE CLOUD COMPUTING	60
2.11.2 CLASIFICACIÓN.....	61
2.11.2.1 SOFTWARE COMO SERVICIO	61
2.11.2.2 PLATAFORMA COMO SERVICIO	61
2.11.2.3 INFRAESTRUCTURA COMO SERVICIO	62
2.12 PLATAFORMAS DE DESARROLLO EN LA NUBE	62
2.12.1 APP FOG	63
2.12.2 OPEN SHIFT	63
2.12.3 ZEND CLOUD.....	64
2.13 GOOGLE APP ENGINE (GAE)	65
2.13.1 SOBRE LA PLATAFORMA.....	65
2.13.2 CONTENIDO DEL SDK	65
2.13.3 COMPILEACIÓN Y DESPLIEGUE.....	66
2.13.4 ALMACENAJE DE DATOS	66
2.13.5 CAPACIDAD DE ALMACENAMIENTO	68
2.13.6 RECURSOS ADICIONALES DE LA PLATAFORMA.....	68
2.14 APLICACIONES JAVA WEB	69
2.14.1 SERVLETS	69
2.14.2 PAGINAS JSP	69
2.15 IDE DE DESARROLLO NETBEANS	70
2.16 HERRAMIENTA UTILIZADA PARA MODELADO DE FLUJOS CON BPMN	70

2.17 OTRAS HERRAMIENTAS UTILIZADAS.....	71
CAPITULO 3	72
MARCO APlicATIVO	72
3.1 INTRODUCCIÓN	72
3.2 FASE DE INICIO	73
3.2.1 DEFINICIÓN DEL ALCANCE DEL PROYECTO	73
3.2.2 DEFINICIÓN DE RIESGOS MEDIANTE EL MODELADO DE PROCESOS DEL SISTEMA CON BPMN	73
3.2.2.1 MAPA DE PROCESOS GENERALES PARA EL SISTEMA.....	73
3.2.2.2 SUB PROCESO DE TRABAJO BAJO SCRUM.....	74
3.2.2.3 DESCRIPCIÓN DE LOS SUB PROCESOS DE SCRUM Y TAREAS.....	75
a) DEFINICIÓN DE LA VISIÓN GENERAL DEL PROYECTO.....	76
b) REUNIÓN DE TORMENTA DE IDEAS.....	76
c) PILA DEL PRODUCTO	77
i) PROCESO DE ESTIMACIÓN DE PÓKER	78
d) PLANIFICACIÓN DEL SPRINT	78
i) ELECCIÓN DE LAS FUNCIONALIDADES	79
3.2.3 DETERMINACIÓN DE FACTIBILIDAD DEL PROYECTO MEDIANTE LOS REQUISITOS DEL SISTEMA A ALTO NIVEL	80
3.2.3.1 IDENTIFICACIÓN DE LOS ACTORES	80
a) SCRUM MASTER	81
b) DUEÑO DEL PRODUCTO.....	81
c) EQUIPO DE DESARROLLO	81
3.2.3.2 IDENTIFICACIÓN DE LOS PROCESOS	81
a) INGRESAR AL SISTEMA	81
b) ACCESO A LOS PROYECTOS YA SEA COMO ADMINISTRADOR O INVITADO	82
c) CREAR UN PROYECTO.....	82
d) INGRESAR A UN PROYECTO	82
e) TELETRABAJAR BAJO LA METODOLOGÍA SCRUM	82
3.2.3.3 ANÁLISIS DE LOS REQUISITOS A NIVEL DEL SISTEMA MEDIANTE CASOS DE USO	83
i) DESCRIPCIÓN DE CASOS DE USO.....	83
3.3.4 PREPARACIÓN DEL ENTORNO DE TRABAJO.....	85
3.3.5 PRIMERA ITERACIÓN.....	85
3.3 FASE DE ELABORACIÓN.....	90
3.3.1 SEGUNDA ITERACIÓN	91
3.3.2 PROCESO DE DEFINICIÓN DE LA VISIÓN DEL PRODUCTO POR PARTE DEL ROL DE PROPIETARIO DEL PRODUCTO.	91
3.3.3 PROCESO DE REUNIÓN DE TORMENTA DE IDEAS	91
3.4 FASE DE CONSTRUCCIÓN	92
3.4.1 MODELO DEL DOMINIO	92
3.4.1.1 DIAGRAMA DE CLASES.....	92
3.4.2 IMPLEMENTACIÓN DE ENTIDADES DEL MODELO DEL DOMINIO.....	93
3.4.3 TERCERA ITERACIÓN	94

3.4.3.1 DEFINICIÓN DE LA VISIÓN DEL PRODUCTO	94
3.4.3.2 CREACIÓN DE LA PILA DEL PRODUCTO	95
3.4.3.2 PRIORIZACIÓN DE LA PILA DEL PRODUCTO Y SU ESTIMACIÓN	96
3.4.3.3 CREACIÓN DE LA PILA DEL SPRINT.....	97
3.5 LA FASE DE TRANSICIÓN.....	98
CAPITULO 4	99
MARCO DEMOSTRATIVO	99
4.1 INTRODUCCIÓN	99
4.2 DESCRIPCIÓN DE LA PRUEBA	99
4.3 POBLACIÓN Y MUESTRA	100
4.4 LA PRUEBA PILOTO	100
4.4 CONTROL DEL TELETRABAJO CON EL PANEL DE CONTROL	101
4.5 ANÁLISIS E INTERPRETACIÓN DE DATOS RECOGIDOS	104
4.5.1 EVALUACIÓN DE LA ENCUESTA CON LA DISTRIBUCIÓN ESTADÍSTICA CHI-CUADRADO	108
4.5 CONCLUSIONES SOBRE EL ESTUDIO.....	109
CAPITULO 5	110
CONCLUSIONES Y RECOMENDACIONES	110
5.1 CONCLUSIONES	110
5.2 RECOMENDACIONES	111
BIBLIOGRAFÍA.....	113
GLOSARIO	116
ANEXO-A MARCO LÓGICO	120
ANEXO-B ÁRBOL DE PROBLEMAS	121
ANEXO-C ÁRBOL DE OBJETIVOS.....	122
ANEXO-D METODOLOGÍA DE DESARROLLO.....	123
PROCESO UNIFICADO ÁGIL “AUP”	123
INVESTIGACIÓN CUALITATIVA.....	126
DISTRIBUCIÓN ESTADÍSTICA CHI-CUADRADO	126

ÍNDICE DE FIGURAS

Figura 2.1: Incremento Iterativo de Scrum.....	18
Figura 2.2: Formato de la pila del Producto	22
Figura 2.3: Formato de la pila del Sprint.....	23
Figura 2.4: Grafico de Burn-Up	24
Figura 2.5: Grafico de Burn-Down.....	25
Figura 2.6: Estimación de Póker	25
Figura 2.7: Reuniones De Scrum.....	30
Figura 2.8: Equipos Distribuidos Geográficamente	34
Figura 2.9: Ciclo de vida de AUP.....	39
Figura 2. 10Tareas que se Realizan en cada Iteración.....	40
Figura 2. 11Artefactos de la Metodología AUP	41
Figura 2.12: Grafico de un diagrama de modelado de Procesos con BPMN	43
Figura 2.13: Modelo de Procesos del negocio privados	44
Figura 2.14: Modelo de Procesos del negocio abstractos.....	44
Figura 2.15: Modelo de Procesos de colaboración.....	45
Figura 2. 16: Elemento grafico de modelado con BPMN, las pasarelas	49
Figura 2. 17: Elemento grafico de modelado con BPMN, las piscinas	50
Figura 2. 18: Elemento grafico de modelado con BPMN, agrupadores	50
Figura 2. 19: Visión general de los pasos a seguir y los requisitos para implementar el teletrabajo en una organización.	53
Figura 2. 20: Requisito de compromiso institucional.....	54
Figura 2. 21: Requisito de planificación del proyecto.....	54
Figura 2. 22: Requisito de autoevaluación	55
Figura 2. 23: Requisito de prueba piloto	55
Figura 2. 24: La Etapa de Apropiación.....	56
Figura 2. 25: El Componente Tecnológico.....	57
Figura 2. 26: Identificación de las necesidades tecnológicas	59
Figura 2. 27 Formas de pago por los Recursos y Servicios prestados por los Proveedores en la Nube	60
Figura 2. 28: Plataforma de desarrollo en la nube AppFog.....	63
Figura 2. 29: Plataforma de desarrollo en la nube Open Shift	64
Figura 2. 30: Plataforma de desarrollo en la nube Zend Cloud	64
Figura 2. 31 Tecnologías a utilizar en el desarrollo	65
Figura 2. 32 Modelado de Procesos con BPMN en la herramienta Microsoft Visio	71
Figura 2. 33 Traductor de Escritorio LEC Power Translator	71
Figura 3. 1: Vista de procesos con BPMN de alto nivel.....	74
Figura 3. 2: Vista a nivel del subprocesso de teletrabajo bajo Scrum en el sistema.....	75
Figura 3. 3: Subproceso de definición de la visión del producto, por parte del rol de propietario del producto.....	76
Figura 3. 4: Subproceso de la reunión de tormenta ideas, donde participan los roles de propietario del producto, el equipo de desarrollo y el Scrum master	77

Figura 3. 5: Subproceso para la elaboración de la pila del producto, donde participan los roles de propietario del producto, equipo de desarrollo y el scrum master	77
Figura 3. 6: Subproceso de estimación de la pila del producto por miembros del equipo de desarrollo, mediante el juego de estimación de póker	78
Figura 3. 7: Subproceso de planificación del sprint, donde participan los roles de propietario del producto, equipo de desarrollo, y el scrum master.....	79
Figura 3. 8: Subproceso de elección de funcionalidades por el propietario del producto, con la participación del scrum master y el equipo de desarrollo.....	80
Figura 3. 9: Requisitos a nivel de sistema, basados en casos de uso.....	83
Figura 3. 10: Panel de control en la nube, de versiones subidas	86
Figura 3. 11: Primera versión de la entrada al sistema.....	86
Figura 3. 12: Entrada al sistema en la nube	87
Figura 3. 13: Entrada al sistema mediante un usuario de Google	87
Figura 3. 14: Bienvenida al sistema.....	88
Figura 3. 15: Creación de un proyecto bajo el rol de Scrum Master	88
Figura 3. 16: El rol de Scrum Master invita mediante email a los actores del sistema con un mínimo de tres miembros del equipo de desarrollo.....	89
Figura 3. 17: Pantalla de aviso de creación del proyecto	89
Figura 3. 18: Pantalla de entrada a un proyecto creado.....	90
Figura 3. 19: Listado de proyectos de la cuenta de usuario google, diferenciado por el rol en cada proyecto.....	90
Figura 3. 20: Entre las opciones que se habilitan al rol de propietario está la de subir un documento que represente la visión del producto	91
Figura 3. 21: Primera implementación para realizar la tormenta de ideas mediante el uso de un servicio IRC gratuito.....	92
Figura 3. 22: Modelo del dominio para diseño de entidades objeto en el Datastore de Google App Engine	93
Figura 3. 23: Entidades implementadas y gestionadas desde el panel de control de GAE	94
Figura 3. 24: Elección de un documento subido como el oficial	94
Figura 3. 25: Acceso al documento oficial elegido por el rol de propietario del producto	95
Figura 3. 26: Acceso opcional a una sala de chat que elija uno de los miembros.....	95
Figura 3. 27: Creación de la Pila del Producto por el rol de propietario del producto...	96
Figura 3. 28: Priorización de la Pila del Producto por el rol de propietario del producto	97
Figura 3. 29: Estimación de la Pila del Producto por los miembros del equipo con la estimación de póker	97
Figura 3. 30: Creación de la Pila del Sprint bajo las condiciones dadas	98

ÍNDICE DE TABLAS

Tabla 2. 1 Tabla comparativa para herramientas colaborativas	33
Tabla 2. 2 Tabla comparativa de criterios para una herramienta colaborativa de teletrabajo	34
Tabla 2. 3: Actividades a realizar en la fase de Inicio	39
Tabla 2. 4: Elementos de tipo evento del estándar de modelado BPMN	47
Tabla 2. 5: Elementos de tipo actividad del estándar de modelado BPMN	48
Tabla 2. 6: Ejemplos concretos de plataforma como servicio	62
Tabla 2. 7: Ejemplos concretos de infraestructura como servicio	62
Tabla 2. 8: Similitudes entre una base de datos relacional con el modelo de base de datos de GAE el Datastore.....	67
Tabla 2. 9: Capacidades de los servicios que ofrece la plataforma de desarrollo en la nube Google App Engine	68

CAPITULO 1

MARCO INTRODUCTORIO

1.1 INTRODUCCIÓN

Los Workflow son sistemas que ayudan a administrar y automatizar procesos de negocios, los procesos del negocio son uno o más tareas o actividades que están ligados de manera directa para lograr un objetivo común. Por definición estos procesos deben ser realizados bajo una estructura que asigne roles y funciones entre los miembros. Como ejemplo de estos procesos se puede mencionar: procesamiento de órdenes, reportes de gastos, entre otros. Los componentes comunes de los procesos del negocio son: personas, información, rutas y las decisiones. Para poder identificar estos componentes se realizan las siguientes preguntas: ¿Qué rutas se siguen? ¿Qué gente participa? ¿Cuál es el rol que juega cada participante? ¿Qué decisiones son tomadas? ¿Qué información es requerida por cada participante? (Geronimo, 2002).

Los procesos y flujos de la información no son exclusivos de los negocios pues están implícitos en los trabajos de las personas, así en el campo de la informática por ejemplo podemos mencionar: las metodologías agiles del software tales como Scrum, puesto que depende de la organización de sus procesos y un flujo de la información. Así desde el punto de vista de Scrum se ve al equipo de trabajo como una entidad autónoma, con sus propias reglas. Scrum tiene una versión para practicar el teletrabajo identificado como scrum distribuido. El teletrabajo significa realizar trabajo a distancia con ayuda de las TIC, en la actualidad se implementa en países del primer mundo y más aún es países latino americanos debido a que da soluciones a problemas de tráfico, inclusión social, ahorro de energéticos entre otros revolucionado así el modo de trabajar, en Bolivia desde mayo de 2014 ya hay proyectos de ley al respecto.

A pesar del esfuerzo de la propuesta de Scrum distribuido por llevar a cabo la construcción del software de manera eficiente en ambientes distribuidos mediante teletrabajo, son propuestas teóricas de procesos a seguir y sus flujos, acudiendo así en sus referencias a distintas herramientas colaborativas no integradas, en la actualidad este problema de falta de integración se podría solucionar con la tecnología de la computación en la nube.

La computación en la nube “Cloud Computing” es una nueva forma de denominar al internet, donde los recursos y servicios informáticos son ofrecidos y consumidos como servicios solicitados por el usuario, un ejemplo de una plataforma de desarrollo en la

nube es: GOOGLE APP ENGINE que de ahora en adelante se abreviara como GAE, utiliza la infraestructura de google, para hospedaje y desarrollo de aplicaciones web¹.GAE ofrece un plan gratuito de desarrollo, que es la que se usara en este trabajo. El presente trabajo trata de la implementación de un workflow colaborativo para la metodología ágil Scrum en su propuesta distribuida, implementada como una aplicación web en java alojada en la nube de GOOGLE.

1.2 ANTECEDENTES

1.2.1 ANTECEDENTES DE LA TEMÁTICA

No hace mucho que las distintas organizaciones con la globalización, tuvieron la necesidad de agilidad, precisión, y automatización en los procesos implicados a sus trabajos, además del almacenamiento de información que conlleva este control, una de las respuestas a este problema es la innovación herramientas de software sobre internet que permiten el intercambio de información, y el proceso implícito de colaboración que esto significa. A estas herramientas se les denomina software colaborativo o Groupware, y entre sus funcionalidades están las del intercambio de documentos su corrección y publicación, también se les integran funcionalidades de comunicación entre los usuarios tales como: video conferencias, chats, correo electrónico, entre otros (Rondo Montes, 2008).

Sin embargo a pesar de la importancia y utilidad que traen el intercambio de información a través de sistemas colaborativos groupware la complejidad de planificar y gestionar los procesos de las organizaciones es una tarea que requiere un tratamiento más complejo así por ejemplo enseñar desde cero cierto proceso administrativo a un nuevo empleado, una de las respuestas a este tipo de problemas es los sistemas de software llamados Workflow que son un tipo de Groupware que en especial intenta automatizar los flujos de la información de los procesos y flujos de información de los negocios.

Existe una filosofía que engloba los conceptos de colaboración de los Groupware y la sistematización de los flujos del trabajo de los workflow, esta es BPM², que tiene por objetivo coordinar los recursos del proyecto tales como las personas, la información, las maquinas, y subsistemas de coordinación de los procesos, en resumen los groupware y por tanto los workflow son parte de los BPM. La filosofía BPM tiene como objetivo mejorar la eficiencia de los procesos del negocio, es así que un sistema workflow se

¹ Una aplicación web es aquellas que residen en una red, generalmente se acceden desde un navegador.

² BPM por sus siglas en inglés que significa Gestión de procesos del negocio y hace referencia a los sistemas de software que ayudan en una gestión integral de los procesos de un negocio.

centra en encaminar estos procesos. En este trabajo se automatizara los flujos de los procesos de la metodología Scrum en su versión distribuida con un sistema Groupware de tipo workflow.

Los objetivos de un workflow al interior de una organización son:

- Reflejar, mecanizar y automatizar los métodos.
- Independizar el método y flujo de trabajo de las personas.
- Facilitar la movilidad del personal.
- Agilizar el intercambio de la información y la toma de decisiones.
- Optimizar los objetivos de la metodología sobre la que se implemente.

[CANSECO, 2002].

Debido a la variedad de procesos del negocio, se tiene una clasificación de los sistemas workflow según la funcionalidad así se tiene: de producción, de colaboración, y de administración. Los workflow de producción son aquellos que están ligados a transacciones con bases de datos, estructuran trabajos repetitivos con gran manejo de datos. Los workflow de colaboración son aquellos que estructuran procesos que involucran personas, las cuales quieren llevar un objetivo a su fin, se sigue la ruta de estos documentos y sobre estos se decide las acciones a tomar. Los workflow de administración son aquellos que solucionan los procesos administrativos de un negocio [RONDO, 2008]. En este trabajo se utilizara las bases de un workflow colaborativo ya que lo que se quiere es estructurar los procesos de la metodología Scrum en su propuesta distribuida, ya que dichos procesos involucran a personas cuyo objetivo es construir un proyecto de software, para lo que se sigue rutas definidas de trabajos e información.

Las herramientas colaborativas brindan una forma de trabajar de forma remota entre los miembros de un equipo más aún hay herramientas de trabajo colaborativo que se han especializado para ciertos grupos de trabajo, en lo que compete a este trabajo se puede mencionar a herramientas colaborativas para equipos de desarrollo de software basados en las prácticas ágiles y sus procesos así se menciona: kunagi que es herramienta web que proporciona la gestión de proyectos basados en Scrum que para su uso se hace necesario realizar una descarga de su producto que es una aplicación web implementada en java además de que por ahora dicho software esta en inglés, otro software en habla inglesa es ScrumDo que es una herramienta web con la posibilidad de trabajar en su versión gratuita, con opciones de realizar suscripciones de pago según el uso que se hace de la herramienta, SprintoMeter es otra herramienta que gestiona y hace seguimiento de

proyectos Scrum y XP³ implementada en inglés y para su uso se hace necesario una descarga del producto, otra opción es IceScrum que es una herramienta en inglés y para su uso es necesario una descarga. En resumen existen herramientas de colaboración basados en la gestión ágil sin embargo estas requieren un conocimiento de inglés técnico, además el proceso de construcción del software se basa solamente en las reglas propuestas por Scrum clásico, sin tomar en cuenta los detalles del teletrabajo en equipo, como lo hace la metodología Scrum en su versión distribuida. El teletrabajo y su implementación están sustentadas por métodos probados en países de América latina, estos métodos serán vistos con detalle en este trabajo.

Este trabajo trata de la implementación de un sistema perteneciente al conjunto de los sistemas groupware del tipo workflow, tomando como procesos del negocio o entidad, a los procesos de la metodología ágil de desarrollo del software scrum en su versión distribuida, lo que es válido puesto que como se mencionó en la introducción scrum contempla procesos, tareas, y actividades que son susceptibles de ser modeladas en su fase de análisis, además que la implementación tomara en cuenta las buenas prácticas identificadas por corporaciones especialistas en networking y telecomunicaciones como CISCO, que se resumen en la utilización de la virtualización en sistemas de almacenamiento y los entornos de trabajo, lo que quiere decir que se debería utilizar los sistemas bajo Cloud Computing. (TIC, 2013)

El Teletrabajo es un concepto que define el trabajo a distancia, no necesariamente desde el hogar, este concepto en la actualidad es un factor de importancia en países del primer mundo, ya que es una solución a problemas de tráfico, ahorro a costos de operación, y problemas de inclusión social. Esta modalidad de trabajo también se ha adoptado en países latino americanos como Chile y Colombia en este último país la modalidad de teletrabajo se ha llevado a la práctica por iniciativa de los políticos ya que se ha cubierto el aspecto legal por completo tanto así que en la actualidad es el estado quien da las pautas y pasos detallados que las instituciones deben seguir en la implementación del teletrabajo dado que el teletrabajo es una temática jurídica, y lo más importante es que estos detalles están basados en prácticas internacionales lo que quiere que su modelo de teletrabajo se basa en procesos ya probados por países más avanzados en el tema como lo es la unión europea o Estados Unidos. En Bolivia las primeras iniciativas políticas de sentar las bases de un gobierno digital datan del 2011 con la promulgación de leyes que

³ XP por sus siglas en inglés que significan programación extrema y hace referencia a una metodología de ingeniería del software ágil.

reglamentan el uso de las TIC, que dicho de paso son parte propia de las herramientas de las que se vale el teletrabajo, posteriormente en el año 2014 con motivo del día mundial del internet el 17 de mayo se presenta el proyecto de ley a nivel nacional del teletrabajo a cargo de la institución gubernamental FUNDETIC que es una organización sin ánimo de lucro cuyo objetivo es el de promover y estimular actividades relacionadas con el desarrollo de las TIC. Según la organización se define al teletrabajo como una forma de organización laboral en la que hay relación de dependencia laboral entre empleador y empleado, esta relación trata del desempeño de actividades remuneradas utilizando para ello como soporte de comunicación y evaluación las TIC. Según las prácticas de la implementación del teletrabajo probadas internacionalmente, uno de los aspectos importantes e intrínsecos⁴ es la tecnología, y que entre estas tecnologías se tiene los sistemas colaborativos, que como se justificó en la definición de sistema colaborativo Groupware incluyen a los sistemas workflow.

Los Workflow son una manera de modelar el teletrabajo para su implementación posterior en sistemas informáticos, esto es de importancia en este trabajo puesto que Scrum en su versión distribuida aborda el teletrabajo como eje central en el desarrollo de software bajo las prácticas de Scrum. En la implementación del teletrabajo en una organización se debe tomar en cuenta las tecnologías que son recomendadas por los expertos, que recomiendan a la computación en la nube como la mejor opción por la virtualización que bridan sobre los recursos computacionales, ya sea hardware o software. La computación en la nube es un modelo que permite tener acceso a un conjunto de recursos computacionales compartidos, de una forma rápida y sencilla, sin necesidad de esfuerzo administrativo y con poca interacción con el proveedor de servicios, este modelo promueve la disponibilidad, ahorro en licencias, administración de equipos necesarios para la administración de servicios, actualización permanente sin cambios traumáticos en los equipos de los usuarios. Todo esto gracias a su arquitectura formada por capas también llamadas modelos de implementación, entre sus características esenciales se tiene cinco, y tres modelos de implementación:

Características:

- Servicio auto gestionado bajo demanda
- Acceso global

⁴ Término utilizado en filosofía para designar lo que corresponde a un objeto por naturaleza y no por su relación.

- Conjunto de recursos físicos virtualizados disponibles en línea.
- Software como servicio.
- Servicios medibles y tarifables según uso del recurso u servicio.

Modelos de implementación:

- Cloud software as a services
- Cloud platform as a services
- Cloud infrastructure as a services

[Perez, 2012]

1.3 ANTECEDENTES DE TRABAJOS SIMILARES

En la Carrera de informática de la UMSA los trabajos que se relacionan con la presente investigación son:

- Sistema de gestión, control y monitoreo de procesos utilizando tecnologías workflow centro de multi servicios educativos CEMSE, realizado por Laura Quisbert Bustamante para la institución de multi servicios educativos CEMSE bajo la modalidad de proyecto de grado en el año 2005. Es un sistema web diseñado para el registro de la correspondencia y el tratamiento de esta.
- Sistema de información para administración y control de inventarios basado en SCRUM, realizado por Adrian Leonardo Ramirez Villegas en el año 2007 para la institución Bolivia Educa bajo la modalidad de proyecto. Es un trabajo que implementa un sistema de control de inventarios con planificación bajo SCRUM.
- Sistema de información para el control y seguimiento de actividades de proyectos, realizado por Ariel Milton Quiroz el año 2008 para la institución CAEM bajo la modalidad de proyecto. Se trata de un sistema web para facilitar el manejo de la información y asignar los procesos requeridos por el usuario.
- Workflow Aplicado al Sistema Integrado de Control y Seguimiento de la Comercialización de Certificados de Rosetas Soat, realizado por Elizabeth Patricia Rondo Montes en el año 2008, para la compañía de seguros y reaseguros La Boliviana Ciacruz bajo la modalidad de proyecto de grado. Es un sistema web modelado con la metodología UWE, implementado en ASP.NET y cuya función principal es corregir los problemas de asignación, devolución, y ventas de los lotes de los certificados y rosetas SOAT.

En otras universidades del exterior:

- Aplicación para gestión de proyectos con Scrum, realizado por Jesús María Eraso Lerena, Univ. De La Rioja 2012, en la modalidad de Trabajo fin de grado. Sistema web para gestión de proyectos de un departamento de informática, cuya implementación se realiza bajo la metodología SCRUM.
- Computación en las Nubes como estrategia competitiva para las PYMES, realizado por Julio Cesar Méndez Landa, Univ. Veracruzana 2010, en la modalidad de Tesis. Trabajo que aborda el del porque en México los empresarios desconocen el paradigma del Cloud Computing.
- Metodología para desarrollo de software en ambientes virtuales realizado por Alejandro Mitaritonna, Univ. Tecnológica Nacional de Buenos Aires 2010 en la modalidad tesis de maestría. Trabajo que propone una metodología de desarrollo en ambientes distribuidos, haciendo énfasis en tres capas: gestión, metodología y herramientas.
- Implementación de tecnología de Cloud Computing para ofrecer servicios de infraestructura en la Univ. de Colima facultad de telemática 2012. Se trata de la implementación de servicios de infraestructura mediante la virtualización de los recursos físicos mediante programas como Xen Citrix, que virtualiza servidores, maquinas, y los agentes necesarios para ofrecer infraestructura en la nube.

1.4 PLANTEAMIENTO DEL PROBLEMA

1.4.1 ANTECEDENTES DE LA PROBLEMÁTICA

La metodología Scrum en su versión distribuida, es una versión de la metodología scrum que proporciona las reglas y recomendaciones a seguir en un equipo de trabajo de desarrollo de software que implementa el teletrabajo, para esto propone el uso de herramientas colaborativas en línea, sin embargo estas propuestas de software son excluyentes, es decir se propone distintas herramientas para un solo trabajo, lo que hace que la práctica de esta metodología ágil sea incomoda de implementar. Además está el problema de las herramientas colaborativas dirigidas a proyectos que trabajan bajo Scrum que están ya implementadas son exclusivamente para el público anglosajón; estas herramientas no consideran el escenario del teletrabajo, puesto que no hacen análisis particular de la organización o entidad ni las evalúan según una herramienta reglamentada exclusivamente para el teletrabajo.

1.4.2 PROBLEMA CENTRAL

¿Cómo implementar el teletrabajo, al interior de un equipo de desarrollo de software, que utiliza como metodología de desarrollo ágil a Scrum, de tal manera que la solución adoptada sea viable de implementar?

1.4.3 PROBLEMAS SECUNDARIOS

- Barreras geográficas y temporales entre los miembros de un proyecto de desarrollo de software
- Densidad de la población de automotores y el tráfico que causa esto lo que con lleva un estrés para los miembros del proyecto de software y provoca un bajo rendimiento.
- Falta de herramientas colaborativas que implementen el teletrabajo bajo prácticas y métodos probados, dirigido a equipos de desarrollo de software en general, que trabajen bajo la lógica de la metodología ágil Scrum.
- Necesidad insatisfecha de contar con herramientas implementadas para un escenario de teletrabajo, implementada en español.
- Necesidad de contar con herramientas para teletrabajo sencillas de manejar, accesibles a todo público.
- Falta de una herramienta colaborativa para el teletrabajo propuesto por la metodología Scrum en su versión distribuida.

1.5 DEFINICIÓN DE OBJETIVOS

1.5.1 OBJETIVO GENERAL

Desarrollar un sistema workflow colaborativo para teletrabajo, sencillo de gestionar, que sea accesible desde la nube publica de google con solo el requisito de tener una cuenta google, que modele los procesos, y flujos de trabajo de la metodología ágil Scrum en su versión distribuida para mejorar la colaboración, comunicación, coordinación al interior de un equipo de desarrollo de software de habla hispana que implemente el teletrabajo.

1.5.2 OBJETIVOS ESPECÍFICOS

- Implementar una herramienta de teletrabajo en una plataforma de desarrollo en la nube, accesible desde internet.
- Escoger uno de los escenarios de teletrabajo que expone Scrum en su versión distribuida y tomarlo como la entidad u organización que requiere del teletrabajo.

- Modelar los flujos y procesos de información de la metodología scrum con el lenguaje de especificación estándar BPMN, según los roles que define esta metodología.
- Modelar los artefactos que define scrum para el proceso de construcción de software, en particular la pila del producto y del sprint.
- Poner a disposición la herramienta colaborativa en la nube de google.
- Valorar la implementación de software bajo el concepto de Cloud Computing, en su modelo de software como servicio indicando las ventajas, desventajas.

1.6 HIPÓTESIS

Si se implementa software como servicio en la nube, en una plataforma de desarrollo de aplicaciones web pública como Google App Engine, que modele y automatice los procesos y flujos de la información de la metodología ágil scrum, entonces se contara con una herramienta colaborativa para teletrabajar mejorando así las actividades de comunicación coordinación y colaboración al interior de un equipo de desarrollo de software de habla hispana que trabaja con la metodología Scrum en un escenario donde los miembros tienen la posibilidad de reunirse personalmente.

1.7 OPERACIONALIZACIÓN DE VARIABLES

1.7.2 VARIABLE INDEPENDIENTE:

- Conocimiento del equipo de desarrollo sobre las prácticas de desarrollo que propone la metodología ágil scrum.

1.7.3 VARIABLE DEPENDIENTE:

- Sistema de workflow colaborativo basado en Scrum distribuido e implementado en la nube de google, en su modelo de software como servicio.
- La colaboración, comunicación, coordinación entre los miembros del equipo de software que utilizaran la herramienta colaborativa para teletrabajo.

1.7.4 VARIABLES INTERVINIENTES:

- Definición de un workflow bajo la arquitectura estándar de especificación de workflow industriales: Workflow Management Coalition.
- Modelo de implementación en la nube: Software como servicio.
- Conceptos de teletrabajo propuestos por Scrum Distribuido.
- Conceptos para la implementación de una aplicación web en java.

1.8. JUSTIFICACIÓN

1.8.1 JUSTIFICACIÓN ECONÓMICA

Este trabajo es justificable económicamente por tratarse del estudio e implementación de software para teletrabajo para la entidad scrum en su versión distribuida, lo que quiere decir que los beneficios económicos se desprenden de los que se obtiene al implementar el teletrabajo, los cuales están probados en los países que lo han adoptado y cuya justificación están dadas por estadísticas concretas que serán mencionadas en el capítulo siguiente. Los ámbitos de la justificación económica son:

- Para el negocio: reducción de costos fijos en la planta física, mantenimiento, servicios públicos, entre otros.
- Para el área tecnológica: reducción de costos en adquisición de hardware y software, reducción de costos de mantenimiento de los equipos.
- Para los trabajadores: ahorro de dinero derivados de la disminución de desplazamientos.

1.8.2 JUSTIFICACIÓN SOCIAL

El trabajo es justificable socialmente, debido a que el teletrabajo tiene entre sus propósitos soluciones a problemas actuales que sufre la sociedad, así se puede mencionar:

- Reducción de la huella de carbono al evitar desplazamientos de los trabajadores.
- Inclusión socio laboral de la población vulnerable gracias a las TIC, entre las cuales se menciona: personas en situación de discapacidad física, personas en gestación, personas con viviendas alejadas del centro urbano, entre otros.
- Mejoramiento de la calidad de vida de los trabajadores, ya que tienen más tiempo para convivir con su entorno familiar.

1.8.3 JUSTIFICACIÓN CIENTÍFICA

Científicamente el trabajo es justificable porque el conocimiento que se generara será de interés científico a la comunidad de la carrera de Informática de la Universidad Mayor de San Andrés pues aborda e investiga el concepto de que el software debe ser vendido como un servicio, también que es el teletrabajo y como los informáticos aportan a su desarrollo, lo que es un tema actual en países del primer mundo. Otro beneficio científico implícito del estudio es aportar con conocimientos sobre las maneras de implementar sistemas colaborativos para teletrabajo según las normas y buenas prácticas internacionales.

1.8.4 JUSTIFICACIÓN TÉCNICA

El trabajo es justificable técnicamente dado los requerimientos para su implementación y para el posterior uso del sistema terminado, están al alcance de la mayor parte de los desarrolladores y usuarios que requieren teletrabajar. A continuación se muestra los requerimientos de software y hardware tanto para el desarrollo, como para el acceso al sistema una vez terminado.

1.8.4.1 REQUERIMIENTOS DE HARDWARE

- PARA LA IMPLEMENTACIÓN
 - Procesador de 32 bits (x86) o 64 bits (x64).
 - Memoria RAM de 1 gigabyte (GB) (32 bits) o memoria RAM de 2 GB (64 bits).
 - Espacio disponible en disco rígido de 16 GB (32 bits) o 20 GB (64 bits).
 - Dispositivo gráfico DirectX 9 con controlador WDDM 1.0 o superior.
- PARA ACCEDER A LA APLICACIÓN
 - Procesador de 32 bits (x86) o 64 bits (x64).
 - Acceso a Internet

1.8.4.2 REQUERIMIENTOS DE SOFTWARE

- PARA LA IMPLEMENTACIÓN:
 - IDE DE DESARROLLO: Netbeans 8.0
 - JDK 7U10 para Windows X86
 - SDK 1.9.18 de la plataforma de desarrollo Google App Engine
 - Cualquier navegador Web Actual.
- PARA ACCEDER A LA APLICACIÓN
 - Conexión a internet
 - Navegador Web

1.9 ALCANCES Y LÍMITES

1.9.1 ALCANCES

- Exponer los escenarios de teletrabajo que propone la metodología scrum en su distribuida.
- Escoger un escenario de teletrabajo de scrum en su versión distribuida, para luego tomarlo como una entidad u organización a modelar.
- Modelar mediante la definición workflow los flujos y procesos de trabajo de uno de los escenarios de teletrabajo la metodología Scrum en su versión distribuida apoyándose en estándares definidos recientemente, como lo es BPMN, estándar dirigido exclusivamente al desarrollo de sistemas workflow.
- Implementar una interfaz sencilla y amigable que permita escribir, estimar y priorizar la pila del producto.
- Implementar una interfaz sencilla y amigable que permita crear y gestionar las iteraciones de los Sprint.

- Poner en comunicación a los miembros del proyecto de desarrollo software con medios telemáticos, bajo los conceptos de teletrabajo de scrum en su versión distribuida, así por ejemplo el uso de chat.
- Opción de que el dueño del producto pueda dar a conocer su visión del producto, en un formato pdf.
- Acceso a los datos de los artefactos de scrum como pila del producto, pila del sprint, gracias a una base de datos orientada a objetos implantada en la nube.
- Acceso a la información y su modificación según los roles definidos en Scrum.

1.9.2 LIMITES

- El sistema a implementar está sujeto a prueba bajo los límites y cuotas de una versión de prueba gratuita de la plataforma Google App Engine.
- El tipo de teletrabajo para el que se pretende modelar el sistema workflow, es el definido para un escenario parcialmente distribuido, lo que significa que se asume que en cierto momento del trabajo grupal, los miembros se encuentran para complementar sus labores.
- Los flujos y procesos que se modelan de la metodología Scrum, corresponde a los de la fase de análisis además de registro de las iteraciones, y no así las fases de diseño e implementación.
- El modelado del sistema bajo un sistema workflow, se limita a facilitar el flujo de información en un proyecto de software, y no así a un procesamiento de esta información de una manera sintáctica que sería la tarea de un sistema experto por ejemplo.
- El sistema a desarrollar se lo realiza con fines de estudio y aporte a la carrera de informática y no comerciales.

1.10 APORTES

1.10.1 PRÁCTICO

Se pretende mejorar el proceso de comunicación, colaboración, comunicación mediante el teletrabajo, al interior de un equipo de desarrollo del software que sigue los procesos de su construcción con la metodología ágil Scrum en su versión distribuida, con la implementación de software de tipo workflow colaborativo alojado en una plataforma pública Cloud Computing, respetando el paradigma de estos sistemas y los requisitos técnicos particulares de la plataforma.

1.10.2 TEÓRICO

Un sistema workflow colaborativo tiene la intención de centralizar los procesos y flujos de trabajo de negocios donde participan personas su principal funcionalidad es la del intercambio de información entre las personas de una manera coherente. En este trabajo se pretende centralizar los procesos y flujos de trabajo que conllevan el desarrollo de software bajo la metodología ágil Scrum Distribuido cuyas soluciones se combinan con conceptos de teletrabajo, que por otro lado entre las buenas prácticas de su implementación sugiere al Cloud Computing como tecnología ideal.

1.11 METODOLOGÍA

Para la elaboración del presente trabajo se seguirá el método de investigación científico con un enfoque descriptivo.

1.11.1 MÉTODO CIENTÍFICO

El método científico es un método que se aplica por todo investigador con intenciones formales de llevar a cabo una investigación. Para ello propone una serie de etapas:

- Elección de tema.
- Planteamiento del problema.
- Justificación del problema de investigación.
- Objetivos de investigación.
- Estructuración del esquema de investigación.
- Marco teórico.
- Elaboración de la hipótesis.
- Metodología.

[RUIZ, 2007]

1.11.1.1 MÉTODO DESCRIPTIVO

El método o enfoque descriptivo tiene como objeto la descripción de los datos, los cuales tienen un impacto de importancia en la investigación, para ello describe objetos, costumbres y actividades predominantes en el tema de interés, además de un análisis e identificación de las relaciones que existen entre dos o más variables de la investigación.

[SAMPIERI, 2006].

1.11.1.2 METODOLOGÍA PARA PROBAR LA HIPÓTESIS

La hipótesis del presente trabajo tiene las connotaciones de ser una hipótesis cualitativa, lo que significa que la variable dependiente identificada para esta no tiene un carácter de cuantificación específico dado que en esta se involucran hechos de tipo subjetivo los que dependen de cada persona: “La colaboración, comunicación, y coordinación entre los miembros de un equipo de desarrollo que trabajan con la metodología Scrum”. Por esto para la demostración de la hipótesis del trabajo se utiliza la investigación cualitativa la que para demostrar un argumento trata de comprender la perspectiva de los sujetos de investigación acerca de los fenómenos que les rodean. Para esto la investigación cualitativa propone una serie de pasos que terminan con la demostración o negación de un argumento.

- Concebir la idea de lo que se quiere investigar
- Planteamiento del problema
- Definición de la muestra de estudio
- Recolección de datos
- Análisis de los datos
- Interpretación de resultados
- Elaboración de informe con la conclusión de los estudios.

Adicionalmente dentro de la investigación cualitativa se puede encontrar diversas técnicas de recolección de la información entre las que se menciona:

- La observación
- La entrevista
- El cuestionario
- Grupos de discusión

Y como apunte adicional sobre la investigación cualitativa se puede mencionar que en resumen lo que hace para demostrar una hipótesis es analizar mediante una serie de pasos las variables dependientes de la propuesta de hipótesis con una serie de herramientas aptas para la muestra de estudio. [SAMPIERI, 2006].

1.11.2 METODOLOGÍAS DE INGENIERÍA

El presente trabajo a de utilizar el método, técnicas y siguientes herramientas que se describen a continuación en el desarrollo.

1.11.2.1 METODOLOGÍA DE DESARROLLO ÁGIL AUP

AUP es un refinamiento del proceso unificado UP que es un marco de desarrollo de software iterativo, entre otros refinamientos se tiene: Rational Proces Unified “RUP”, el refinamiento AUP simplifica algunas disciplinas de UP, conservando la escancia del desarrollo incremental. La metodología está dirigida por casos de uso y se centra en la arquitectura. Su ciclo de vida consiste en cuatro fases: inicio, elaboración, construcción y transición. Las disciplinas que AUP considera: modelado, implementación, prueba, despliegue, y la gestión de la configuración. Opcionalmente la metodología se apoya en

los casos de uso para recoger los requisitos, y por tanto la utilización de los modelos derivados de este, como modelo del dominio, de navegación, entre otros dirigidos a objetos.

1.11.2.2 HERRAMIENTAS DE DESARROLLO GOOGLE APP ENGINE.

Google App Engine es una plataforma de desarrollo de aplicaciones web en la nube, es decir que ofrece un servicio de software como servicio. En actualidad para la implementación de las aplicaciones ofrece como opción de desarrollo los lenguajes Java y Phyton. Su forma de tarificación está al alcance de todos desde las versiones de prueba que son gratuitas hasta proyectos de gran envergadura y cuyo costo está asociado al concepto del cobro solo por la cantidad usada. En este trabajo se trabajara una aplicación web con java en su estándar disponible para aplicaciones web los Servlets con las paginas JSP⁵ que es el soportado por GAE por el momento.



⁵ JSP que por sus siglas en ingles significan Java Server Pages, y son paginas dinámicas que sirven el contenido de un servidor java, para aplicaciones java en la web.

CAPITULO 2

MARCO TEÓRICO

2.1 INTRODUCCIÓN

El ritmo acelerado que se lleva hoy en día en las sociedades modernas, trae desventajas en distintos contextos, uno de estos es en el trabajo en equipo que es de vital importancia en las organizaciones y sus negocios. Entre las consecuencias de este ritmo de vida en los equipos de trabajo tiene distancias geográficas insalvables ya sea por falta de tiempo o distancia física lo que conlleva desventajas en distintas ámbitos en los que desenvuelven las organizaciones, así por ejemplo: para la productividad, para las operaciones, los costos, cumplimiento de programas de responsabilidad social como inclusión de personas con capacidades distintas, calidad de vida de los empleados. Una solución adoptada en países desarrollados es el teletrabajo.

El área de desarrollo, mantenimiento, y corrección del software no es la excepción a este tipo de problemas puesto que al ser el equipo de desarrollo una estructura independiente con sus reglas internas, auto-organización, procesos y flujos de trabajo propios, adolece de cierto grado de grado de desorganización fruto del poco tiempo que se dedica a la planeación del cómo lidiar con las distancias geográficas entre los miembros en el caso de los equipos distribuidos vale decir cuando se trabaja de forma remota.

Los miembros de Scrum⁶ manager ha sacado una versión distribuida de sus prácticas de desarrollo, en la que se trata precisamente de la manera de trabajar con equipos distribuidos de manera geográfica poniendo como solución a el teletrabajo. El teletrabajo es una forma de organización laboral en la que las personas realizan sus labores fuera del recinto físico de la organización, y que como plataforma de comunicación utilizan las TIC⁷.

En este tema se trata los fundamentos teóricos para la implementación de software colaborativo para teletrabajar bajo las prácticas agiles de scrum. Entre los fundamentos teóricos a tratar se tiene: Las metodologías ágiles de desarrollo, Scrum y las herramientas colaborativas disponibles en línea para su práctica, la versión distribuida de Scrum, metodología de desarrollo AUP, Workflow su arquitectura, las herramientas de modelado del workflow UML⁸ y BPMN⁹ ,el teletrabajo y la forma correcta de implementarlo según

⁶ Asociación especializada en la formación de la gestión de proyectos con Scrum, sin ánimo de lucro
<http://www.scrummanager.net>

⁷ TIC Abreviación

⁸ UML por sus siglas en inglés que significa, Lenguaje para el Modelado Unificado.

normas y experiencias internacionales probadas, escenarios de teletrabajo en scrum distribuido, computación en la nube, plataformas de desarrollo en la nube y sus ofertas, la plataforma de desarrollo Google App Engine y las aplicaciones java en la web.

2.2 METODOLOGÍAS ÁGILES

La competitividad actual hace que los sistemas de información se tengan que desarrollar de forma rápida para adaptarse a la organización. Las prisas hacen que lo primero que se deseche es un análisis exhaustivo y se sustituye por uno superficial o simplemente se elimina, esto es un error pues al querer un sistema desarrollado rápidamente nos topamos con un sistema lleno de errores. Es por eso que las metodologías de análisis y desarrollo se han adaptado a las reglas del mercado mundial que es un ambiente lleno de presiones y cambiante, en este ambiente. Se necesita que los resultados se puedan ver mostrar y sobretodo utilizar, por tanto las metodologías debe ser ágiles, de tal manera de incrementar las funcionalidades en cada iteración, donde se mostrara el resultado que deberá preservarse según lo que requieren las organizaciones (González, 2004).

2.2.1 DEFINICIÓN

Las metodologías agiles son métodos de ingeniería del software basados en el desarrollo iterativo e incremental donde los requerimientos y soluciones evolucionan mediante la colaboración de grupos auto organizados y multidisciplinarios. (González, 2004)

El objetivo principal de las metodologías agiles es el desarrollo de software de forma que sea adaptable a los cambios, esto se hace posible porque las metodologías adoptan un desarrollo de forma incremental en la que se entrega software funcional de forma iterativa permitiendo al cliente corregir errores o introducir requerimientos en etapas del proyecto que son viables de llevar a cabo, además de que el producto final se entrega al cliente en un lapso de tiempo comprometido. (Presman, 2006)

2.2.2 IMPORTANCIA

El desarrollo de forma ágil es de importancia en ciertos proyectos de software que están influenciados por su medio ambiente el cual exige adaptabilidad a cambios repentinos, así por ejemplo los sistemas web cuyo uso principal es el marketing. Las metodologías ágiles son importantes en este tipo de ambientes porque el software se ve sometido a cambios imprevistos ante los cuales se debe responder de forma adaptable y ágil, sin perder cierto grado de rigurosidad que garantice una entrega eficiente. (González, 2004)

⁹ BPMN por sus siglas en inglés que significa notación para el modelado de procesos del negocio

2.2 LA METODOLOGÍA ÁGIL SCRUM

En este apartado se da los fundamentos necesarios para entender en todo su detalle a la metodología scrum, ya que el tema central del trabajo trata sobre el modelado de una herramienta colaborativa fundamentada en los procesos y flujos de la metodología scrum.

2.2.1 DEFINICIÓN

Scrum es una metodología ágil de gestión de proyectos, que se basa en la adaptación continua a las circunstancias evolutivas del proyecto apoyándose en iteraciones cortas conocidas como Sprints. En resumen lo que Scrum realiza es la definición de los requisitos del cliente de estos se escoge unos a priorizar y se realizan en ciclos o iteraciones denominados sprint con una duración máxima de seis semanas, si surgiera alguna observación por parte del dueño del producto entonces se prioriza y se realiza la iteración hasta que el dueño del producto este conforme. Para el monitoreo de los sprint se realiza reuniones diarias de quince minutos máximo de duración en que se da cuenta de los trabajos realizados y a realizar. Para que Scrum funcione se define: Roles/Responsabilidades, Artefactos, Eventos. (Palacio, Gestión de Proyectos Scrum Manager, 2014). Scrum es aplicable a todo tipo de proyectos que tengan necesidad de agilidad y cuyo entorno sea impredecible, Scrum en los proyectos de software requiere necesariamente de la gestión de las iteraciones de forma diaria a través de las reuniones diarias. (Palacio, Flexibilidad con Scrum, 2007). En la figura 2.1 se observa la aplicación de scrum en las organizaciones, según el grado de conocimiento de esta metodología, el grado más avanzado es la aplicación de un incremento continuo donde el scrum master sabe cómo romper reglas sin afectar de manera negativa. En este trabajo se implementa una herramienta colaborativa suponiendo un incremento iterativo, donde las reglas están bien definidas.

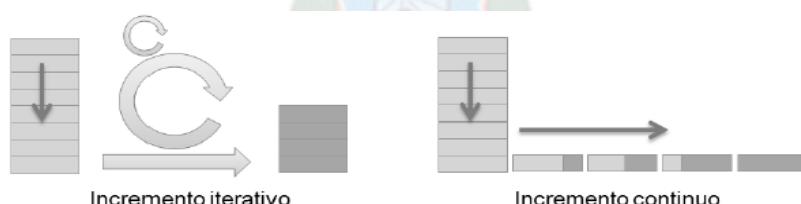


Figura 2.1: Incremento iterativo de Scrum

Fuente: (Palacio, Gestión de Proyectos Scrum Manager, 2014)

2.2.1.1 ROLES

Todas las personas que intervienen de manera directa o indirecta en el proyecto se clasifican en dos grupos: comprometidos e implicados. Esto se debe entender desde un punto de textual es decir que los comprometidos son los que arriesgan más en la situación

en cuestión, los implicados por el contrario no arriesgan más bien son una especie de colaboradores, o testigos. (Palacio, Flexibilidad con Scrum, 2007)

a) SCRUM MÁSTER

i) DEFINICIÓN

Responsable del funcionamiento de las reglas en un marco de scrum técnico, asegurando que todos los miembros del proyecto comprendan scrum, además proporciona ayuda y formación técnica al propietario del producto y el equipo de desarrollo. (Palacio, Gestión de Proyectos Scrum Manager, 2014)

ii) RESPONSABILIDADES

- Asesoría y formación al equipo para trabajar de forma auto organizada o con responsabilidad de equipo.
- Revisión y validación de la pila del producto.
- Moderación de reuniones.
- Resolución de los impedimentos que pueden entorpecer la ejecución de las tareas de cada sprint.
- Gestión de las dinámicas de grupo en el equipo de desarrollo.
- Hacer respetar las formas de scrum, hacia un diseño y mejora continua de las prácticas de scrum.

b) EQUIPO DE DESARROLLO

i) DEFINICIÓN

Conjunto de profesionales encargados de desarrollar el producto de forma iterativa en cada sprint, se recomienda que este equipo debiera estar formado entre 4 y 8 personas debido a que más de 8 personas significan la aparición de los roces habituales de la dinámica de grupos. El trabajo al interior de este grupo debe ser multifuncional, es decir que más allá de las especialidades de cada miembro, todos deben estar dispuestos a colaborar a los demás puesto que la responsabilidad laboral es del grupo (Palacio, Gestión de Proyectos Scrum Manager, 2014)

ii) RESPONSABILIDADES

- Todos conocen y comprenden la visión general del proyecto
- Aportan en la preparación de los elementos de la pila del producto
- Comparten el objetivo del sprint y la responsabilidad del incremento
- Todos aportan con opiniones para la toma de decisiones

- Se respeta los aportes de cada uno de los miembros
- Todos conocen el modelo de trabajo Scrum.

c) PROPIETARIO DEL PRODUCTO

i) DEFINICIÓN

El propietario del producto es aquella persona responsable de velar por los intereses del cliente para el que se construye el sistema, este cliente puede ser una organización, un departamento de una organización, una persona particular entre otros. El propietario de producto puede ser una persona que represente a un conjunto de interesados directos en el producto, como también puede ser un cliente directo. Para ejercer este rol es necesario conocer perfectamente el entorno del negocio, además de manejar a la perfección la visión general del proyecto. (Palacio, Gestión de Proyectos Scrum Manager, 2014).

ii) RESPONSABILIDADES

- Definir cómo será el producto final ya sea en base a sus requisitos personales o una fase de recolección y licitación de requisitos con el conjunto de interesados.
- Responsable de mantener las funcionalidades de la pila del producto “requisitos del sistema” de forma ordenada desde los más prioritarios para el cliente y aportan más valor desde su punto de vista.
- Conoce que es lo que quiere el cliente final y esto lo comunica de forma metódica bajo las formas de scrum a los miembros del proyecto.

d) OTROS INTERESADOS

Tienen intereses en el proyecto, e intervienen de manera indirecta.

2.2.1.2 ARTEFACTOS

Los artefactos son sub procesos del proceso general de desarrollo scrum, es decir que los artefactos son sub conjuntos que a su vez están formados por tareas específicas.

a) PILA DEL PRODUCTO

Lista de requisitos del sistema, mejoras, corrección de errores que evoluciona a largo del desarrollo por lo que se lo considera un documento vivo. La accesibilidad de este artefacto es para todo los miembros del proyecto pudiendo así aportar con sugerencias. (Palacio, Flexibilidad con Scrum, 2007).

i) CONSTRUCCIÓN

Se construye de manera usual a partir de una tormenta de ideas donde participan los miembros del equipo de desarrollo a partir de una visión general del proyecto dada por el dueño del producto o cliente.

ii) VISIÓN GENERAL DEL PROYECTO

La visión general del proyecto, representa lo que esperar el cliente del producto de una forma general, el formato de este documento es irrelevante pudiendo ser una presentación informal en forma de lista de funcionalidades deseadas, o una redacción, sin embargo la importancia es de alto nivel, porque esta visión debe ser plenamente comprendida por todos los miembros del proyecto, para una puesta en práctica de scrum dentro de un contexto comprendido y coherente.

iii) CONSTRUCCIÓN Y PREPARACIÓN

La segunda fase luego de la construcción de la pila del producto a partir de la visión general del proyecto dada por el dueño del producto mediante la reunión de tormenta de ideas donde participan el equipo de desarrollo, con la organización del scrum máster, y la guía del propio dueño del producto, donde se asume que el equipo ya se ha familiarizado con la visión general, y por tanto la tormenta de ideas gira en torno a traducir la visión general en funcionalidades en términos de requisitos de software, haciendo estas funcionalidades listas para la incorporación a un ciclo de sprint, en la construcción de esta lista de funcionalidades se debe definir los siguientes tres puntos por cada funcionalidad, a lo que se denomina preparación de la pila del producto.

- Priorización: es dar importancia a las funcionalidades, según los objetivos del proyecto, y es el dueño del producto o cliente quien da esta priorización ya que es él para quien se construye el sistema.
- Detallado: el detallado es la descripción de forma verbal de lo que trata la funcionalidad, esto lo realiza el dueño del producto con ayuda de todos los miembros del equipo del proyecto.
- Estimación: es el esfuerzo estimado por cada funcionalidad, y se realiza por el equipo de desarrollo, ya que son quienes realizaran el trabajo.

iv) FORMATO

De manera preferencial la comunicación debe de ser de forma visual y llamativa, por los que se prefiere el uso de pizarras ya sean físicas o digitales, las cuales deben contener como mínimo la información que se muestra en la figura 2.2:

Id	Orden	Est.	Descripción	Criterio validación	Obs.
1	10	30	Plataforma tecnológica	Se tiene el diagrama de la arquitectura, validado por xxx	La arquitectura debe permitir escalabilidad por clusterización de
2	20	40	Prototipos interfaz usuario	Todas las pantallas de interfaz están dibujadas y se puede recorrer toda la func	Debe estar interfaz para las funcionalidades de la pila a fecha
3	30	40	Diseño de datos	Diagrama BB.DD. Realizado, validado por xxx	
4	40	60	El operador define el flujo y textos de un expediente	Definir completamente un expediente con la funcionalidad programada	
5	50	999.	Etc...	Etc...	

Figura 2.2: Formato de la Pila del Producto

Fuente: (Palacio, Flexibilidad con Scrum, 2007)

En la figura 2.2 se puede observar los siguientes campos para la pila del producto en orden ascendente:

- Un ID único de funcionalidad o trabajo, para identificar de manera única a cada funcionalidad.
- Orden o Priorización de las funcionalidades, dadas por el cliente o propietario del producto.
- Estimación de la duración de las funcionalidades dadas en número de días, esta estimación de manera ideal se debería dar por el equipo de desarrollo ya que ellos son los que traducirán dichas funcionalidades en producto terminado.
- Descripción de las funcionalidades de forma textual.
- Criterio de validación para saber cuál es el parámetro que se considere a una funcionalidad como realizada, como ejemplo la funcionalidad de diseño de base de datos el criterio de validación podría ser que exista un diagrama físico realizado.
- Observaciones se refiere puntualizaciones y apuntes de cada una de las funcionalidades.

b) SPRINT

Un Sprint es el nombre que se da a una iteración de desarrollo en la metodología scrum, se considera el núcleo central de avance por tiempos prefijados en la fase de planificación de este que como se menciona más adelante se realiza mediante un evento en forma de reunión donde se define el objetivo del sprint y las tareas a desarrollar. (Palacio, Flexibilidad con Scrum, 2007)

i) PILA DEL SPRINT

Lista de tareas a desarrollar por el equipo de desarrollo durante un sprint, esta lista es el producto de la planificación del sprint en la reunión de planificación. La base de esta lista es la pila del producto con sus funcionalidades a desarrollar y que están preparadas, vale

decir que las tareas de esta pila descomponen las funcionalidades de la pila del producto.
(Palacio, Flexibilidad con Scrum, 2007)

ii) FORMATO DE LA PILA DEL SPRINT

El formato de la pila de sprint de forma preferencial debe ser de manera gráfica, y además que englobe todos los elementos necesarios para un buen entendimiento del equipo de desarrollo de lo que se está realizando en cada sprint, lo que debería contener de manera mínima una pila de sprint como información es:

- Numero de Sprint
- Objetivo del sprint
- Lista de tareas que se llevan a cabo en el sprint
 - Descripción
 - Responsable
 - Tiempo estimado
 - Estado
 - Campo de revisión y seguimiento de la tarea
 - Fecha de revisión
 - Descripción de lo realizado el día anterior
 - Descripción de los que se planea hacer el presente día
 - Observaciones
 - Tiempo estimado que queda para acabar la tarea

Categoría	Tarea	Responsable	Estimado en horas	Estado	OBJETIVO DEL SPRINT				
					Crear y publicar versión básica del sitio web público				
Diseño	Crear diseño de base de datos	Juan	24	Completo	24	16	8	4	
Diseño	Validar diseño de base de datos	Pedro	4	Completo	4	4	4	4	
Desarrollo	Contratar servicio de hosting	Pedro	4	Completo	4	2			
Desarrollo	Crear layout y estilos de sitio web	Maria	16	Activo	8	8	4	2	
Desarrollo	Crear página principal	Maria	24	Pendiente	24	24	24	24	24
Desarrollo	Mostrar resúmenes de noticias por sección	Juan	16	Pendiente	16	16	16	16	8
Desarrollo	Crear banners de publicidad	Luis	24	Pendiente	24	24	24	24	24
Desarrollo	Visualizar un Artículo	Luis	8	Pendiente	8	8	8	8	8
Desarrollo	Imprimir un Artículo	Luis	4	Pendiente	4	4	4	4	4

Figura 2.3: Formato de la Pila del Sprint

Fuente: (Palacio, Flexibilidad con Scrum, 2007)

En la figura 2.3 se puede observar de forma gráfica una pila del sprint con sus componentes.

iii) INCREMENTO

Un incremento es el resultado de cada sprint en forma de producto listo para ser entregado.

c) MÉTRICAS AGILES

Permite ver el progreso de la ejecución de las tareas de la pila del Sprint o de la pila del producto, de manera gráfica estadística similar a los gráficos de aplicaciones como Excel. Entre estas métricas ágiles se puede mencionar: grafico de Burn-Up, Burn-Down y los juegos o protocolos de decisión. (Palacio, Gestión de Proyectos Scrum Manager, 2014)

i) GRAFICO DEL PRODUCTO (BURN-UP)

Este grafico representa la velocidad con que se desarrolla la pila del producto y su evolución, se confecciona en un eje de coordenadas cartesianas, donde el eje “x” representa las fechas de culminación de los sprint, y el eje “y” representa el esfuerzo estimado por cada sprint, es decir cuánto tiempo se estima que dure un sprint, esta estimación se la puede realizar en la pila del producto, estimando el tiempo en días de cada funcionalidad y sumando estas según pertenezcan a una versión es decir un sprint. En la figura 2.4 se observa un ejemplo del grafico para la pila del producto.

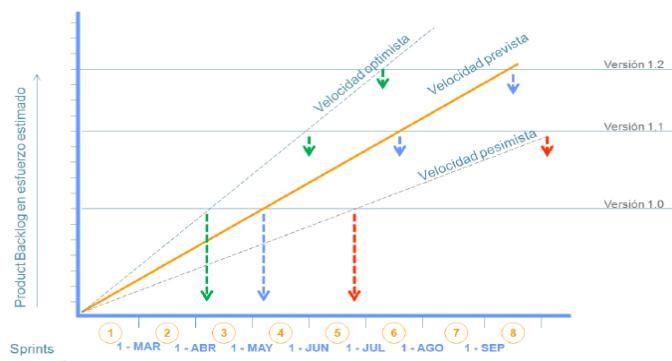


Figura 2.4: Grafico de Burn-Up

Fuente: (Palacio, Flexibilidad con Scrum, 2007)

ii) GRAFICO DEL SPRINT (BURN-DOWN)

Este grafico es de utilidad para el equipo de desarrollo, ya que muestra el avance diario de un sprint, y revela de forma temprana desviaciones en el avance. Se representa por un gráfico cartesiano, donde el eje “x” representa los días laborables en los que se elabora el sprint, y en el eje “y” el trabajo pendiente en el contexto del sprint, medido en días, así por ejemplo si se tiene 3 tareas con duración estimada de 3, 7, y 10 días de duración entonces la estimación total de días que faltan para culminar el sprint es la suma de estos tres es decir 20, los miembros de equipo en cada día que pasa actualizan esta

información, aclarando cuanto les falta de tiempo de su tarea de forma estimada. De forma ideal la recta con pendiente negativa acaba en la fecha prevista, sin embargo en la realidad esta no es una recta si no hay variaciones.

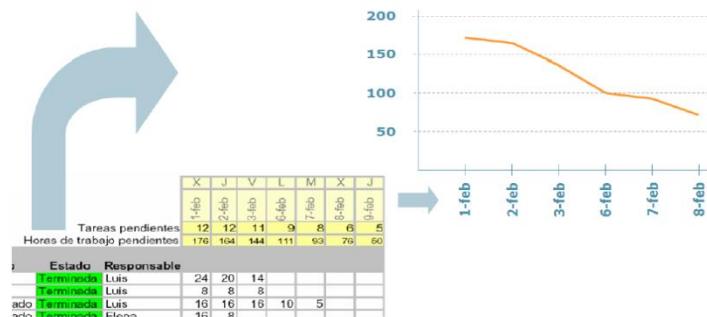


Figura 2.5: Grafico de Burn-Down

Fuente: (Palacio, Flexibilidad con Scrum, 2007)

La figura 2.5 muestra el grafico de avance para los sprint, se observa como los datos de la pila de sprint se traducen en un gráfico estadístico del avance.

iii) JUEGO DE DECISIÓN “ESTIMACIÓN DE PÓKER”

En los proyectos de software al igual que en los de otro tipo se necesita estimar o planificar las duraciones asignadas a las tareas. En particular las metodologías ágiles necesitan métodos que de estimación que sean consecuentes con la necesidad de agilidad, así por ejemplo en la programación extrema “XP” se sugiere la programación en pareja dejando así las planificaciones a una coordinación implícita entre las parejas de programadores, en otras metodologías ágiles como AUP la planificación o estimación se deja a criterio de los desarrolladores dejando simplemente claro que se debe delimitar donde comienzan sus fases y la entrega de producto de manera iterativa. En scrum se ha diseñado un juego didáctico para la estimación de los tiempos de duración de los sprint, y tareas asignadas, siendo único en su clase, este es denominado estimación de poker.

La estimación de póker se basa en un juego donde todos los miembros del equipo cuentan con un juego de cartas similar al de la figura 2.6.



Figura 2.6: Estimación de Póker

Fuente: (Palacio, Gestión de Proyectos Scrum Manager, 2014)

Los números que se observan representan las unidades a estimar por ejemplo en días, así la carta 7 se entiende como 7 días estimados en la duración de la implementación de la tarea. Cada miembro luego de la presentación de la funcionalidad a estimar por parte del dueño del producto y absueltas sus dudas, escoge una carta que se aproxime a su estimación, hecho esto por parte de todos los miembros del equipo de desarrollo, el scrum master realiza un cálculo de la media de la estimación más escogida. Se debe entender que la carta con infinito se levanta cuando la estimación supera un máximo de duración estimada de ante mano, y si la mayoría coincide con que la estimación es infinito entonces la funcionalidad se debe descomponer en sub tareas. Esta estimación se incorpora como una opción en el sistema, sin embargo no es la única, ya que por razones de tiempo la herramienta de teletrabajo debe brindar velocidad es así que también el dueño del producto podrá estimar de manera directa todas las funcionalidades.

2.2.1.3 EVENTOS

Un evento de la metodología scrum, es un suceso con la intención de llevar a cabo los procesos del software definidos por la metodología, los eventos a diferencia de los artefactos son actividades que realizan los miembros mientras que los artefactos son herramientas con un formato determinado.

a) REUNIÓN DE PLANIFICACIÓN DEL SPRINT

i) DESCRIPCIÓN GENERAL

Reunión donde el equipo define la pila del Sprint a partir de la explicación del producto por parte del propietario del producto. Esta reunión se divide en dos, en la primera se decide qué funcionalidades de la pila del producto se desarrollaran, en la segunda se desglosan las funcionalidades en tareas a realizar a nivel de software y se las encomienda a los miembros del equipo según sus capacidades.

ii) PRECONDICIONES

- Se tiene un número aceptable de funcionalidades preparadas de la pila del producto a incorporar a un sprint.
- Se asume conocimiento técnico sobre las tecnologías empleadas, y la visión general de proyecto por parte del equipo de desarrollo.
- Duración de la reunión como máximo de un día.

iii) ENTRADAS

- Funcionalidades preparadas de la pila del producto

iv) RESULTADOS

- Objetivo del sprint

- Pila del sprint

v) FORMATO

- Primera parte: la duración de la primera parte es de una a cuatro horas, y es donde se presenta las funcionalidades con mayor prioridad, esto se realiza en una secuencia de tres pasos:
 - El propietario del producto presenta las funcionalidades preparadas que prevea que se puedan realizar en el sprint, para que todos tengan conocimiento de los que se desea obtener.
 - El equipo realiza preguntas para absolver dudas sobre la explicación dada por el propietario del producto.
 - El equipo define en una frase el objetivo del sprint que se llevara a cabo.
- Segunda parte: esta segunda parte puede alargarse hasta el final de la jornada, y es donde se desglosa las funcionalidades que se han elegido en la primera parte, además de la designación de las tareas a los miembros del equipo, en el siguiente orden:
 - Se desglosa las funcionalidades por parte del equipo de desarrollo, en tareas estimando el tiempo que llevaría realizarse sin interrupciones.
 - Se asigna las tareas a los miembros de equipo.

vi) PARTICIPANTES

- Scrum Master: tiene un papel de garante y responsable de la reunión, garantiza que el dialogo entre el equipo y el propietario sea fluida, además de que es garante que la comunicación se enmarque en un ambiente de respeto entre los interlocutores.
- Equipo de desarrollo: se encargan de descomponer las funcionalidades a desarrollar en tareas, la auto asignación de las tareas en un ambiente democrático y concertado.

vii) SOPORTE

El soporte de la reunión hace referencia a las herramientas de ayuda para visualizar los recursos que están en juego como objetivo del sprint definido, duración estimada del sprint, hora fijada para el seguimiento diario del sprint, visión de las funcionalidades de la pila del producto y por cada funcionalidad un pila de sprint con las tareas que lograran la funcionalidad.

b) REUNIÓN DIARIA DE SEGUIMIENTO DEL SPRINT

i) DESCRIPCIÓN GENERAL

Reunión de mejora continua donde el equipo analiza los diferentes problemas encontrados y los aspectos mejorables de la aplicación de Scrum en el proyecto. También se revisa el incremento generado de una forma informal, con objeto de verificar los resultados.

ii) PRECONDICIONES

- Disponibilidad de lugar para realizar las reuniones de manera diaria
- Pila de sprint actualizada
- Asiste todo el equipo de desarrollo
- Asistencia del scrum master
- Un miembro de equipo de desarrollo controla que los tiempos de intervención se cumplan.

iii) ENTRADAS

- Pila del sprint actualizada
- Gráfico de avance Burn Down

iv) RESULTADOS

- Identificación de impedimentos
- Pila de sprint actualizada
- Gráfico de avance Burn Down actualizado

v) FORMATO

La reunión se reduce a un saludo de protocolo, y de manera inmediata los miembros del equipo responden a tres cuestiones de manera breve:

- Tarea que realizó el día anterior
- Tarea que realizará el día de hoy
- Informar si tienen algún impedimento o necesitan algo especial para realizar la tarea asignada.

Estas cuestiones se deben anotar en un medio para tener un seguimiento claro de lo que se realiza por cada miembro, además el scrum master es el encargado de gestionar el tercer punto.

vi) PARTICIPANTES

- Scrum Master: gestiona las necesidades identificadas por cada miembro del equipo.
- Equipo de desarrollo: control del tiempo de intervención de cada miembro, y actualizar la información de su trabajo.

vii) SOPORTE

La herramienta que hace de soporte para esta reunión debe ser capaz de registrar y visualizar la siguiente información: todas las tareas del sprint presente con el nivel de detalle como que miembro es el responsable de cada tarea, duración estimada del sprint, además de la información que irán registrando cada uno de los miembros.

c) REUNIÓN DE RETROSPECTIVA

i) DESCRIPCIÓN GENERAL

Esta reunión se realiza al final de cada sprint con el objetivo de evaluar cómo se lleva a cabo el desarrollo del proyecto mediante el análisis de gráficos de avance, y el punto de vista del incremento por parte del dueño del producto, con el fin de realizar los ajustes necesarios entre los miembros del proyecto.

ii) PRECONDICIONES

- Se ha concluido el sprint
- Asisten: todo el equipo de desarrollo, propietario del producto, y personas implicadas en el proyecto
- El propietario del producto ha recibido el incremento y lo ha probado.

iii) ENTRADAS

- Evaluación del incremento por parte del propietario del producto.

iv) RESULTADOS

- Retroalimentación para los miembros del proyecto en cuanto a su trabajo en la última iteración.
- Definición de la fecha de la próxima reunión de planificación del siguiente sprint.
- Actualización de la pila del producto con las correcciones necesarias.

v) FORMATO

Esta reunión informal se puede reducir a la definición de los siguientes puntos:

- Exposición y clarificación del incremento por parte de equipo al dueño del producto.
- Decisión de actualización o no de la pila del producto en base a la decisión del dueño del producto.
- El Scrum Master fija la fecha de la próxima reunión de planificación del siguiente sprint.

vi) PARTICIPANTES

- Equipo de desarrollo: expone al dueño del producto lo que se ha logrado en el incremento desarrollado
- Dueño del producto: expone sus dudas respecto del incremento que se le ha entregado y decide posibles cambios en la pila del producto en base a los resultados que se van obteniendo
- Scrum Master: se pone al día de en qué nivel se logra los objetivos del presente sprint, ayuda al dueño del producto en la posible actualización de la pila del producto, y fija fecha de la próxima reunión de planificación del siguiente sprint.

vii) SOPORTE

La herramienta de soporte que se ha de utilizar en esta reunión ha de permitir registrar los puntos de exposición del equipo de desarrollo, también las dudas del dueño del producto, actualización de la pila del producto por parte del dueño del producto y por último fijar la fecha de la próxima reunión de planificación del siguiente sprint. En la figura 2.7 se muestra un gráfico sobre las reuniones o eventos de Scrum.



Figura 2.7: Reuniones de Scrum

Fuente: (Palacio, Flexibilidad con Scrum, 2007)

2.3 HERRAMIENTAS COLABORATIVAS PARA SCRUM

2.3.1 KUNAGI

Kunagi es una herramienta web colaborativa en inglés implementada bajo java con la librería Google Web Toolkit. Ofrece la colaboración de proyectos, gestión de proyectos,

gestión archivos y documentos. Se puede utilizar esta herramienta para apoyar y complementar las prácticas de la metodología ágil scrum para lo cual implementa panel de control de proyectos, pizarra interactiva para el sprint en uso, planificación póker para la estimación, comentarios para cada entidad, y un panel de discusión. Para utilizar esta aplicación se debe descargar la aplicación de forma gratuita de su sitio, para posteriormente instalarlo en un servidor que soporte java. Esto porque la aplicación está dirigida a trabajar en el escritorio para luego actualizar los datos en línea.

2.3.2 SCRUMDO

Es una herramienta colaborativa web, dirigida a mejorar los proyectos de desarrollo web que se desarrollan bajo la metodología ágil scrum, es de pago sin embargo hay una versión gratuita de prueba accesible para todo público, tiene soporte para todos los elementos de scrum como: pila del producto, pila del sprint, y métricas basadas en los datos del proyecto con fines estadísticos. Se caracteriza por usada por los desarrolladores de la corporación Apple. Esta implementada en el idioma inglés. Además ofrece entrenamiento en línea sobre las buenas prácticas de scrum, para lo cual se debe pagar como un servicio común en internet.

2.3.3 SPRINTOMETER

Es una aplicación de escritorio, esta requiere para su funcionamiento un servidor para centralizar la función de esta. La descarga se la realiza desde página oficial: <http://sprintometer.com/>, esta aplicación esta implementada en su versión original en inglés aunque ofrece un plugin para el idioma español disponible para descargar. Se define a sí misma como una herramienta de gestión de proyectos que desarrollan bajo scrum y XP, y por ultimo para poder instalar la aplicación se hace necesario cumplir los requerimientos de software que indica en su manual de uso que está disponible para su descarga.

2.4 CRITERIOS DE EVALUACIÓN PARA LAS HERRAMIENTAS COLABORATIVAS COMO BASE DE COMPARACIÓN PARA EL SISTEMA A IMPLEMENTAR

El presente trabajo se trata de la implementación de una herramienta para teletrabajar entre miembros de un equipo de desarrollo software que implementan los procesos de desarrollo de la metodología ágil scrum, en el mercado actual de aplicaciones colaborativas existen herramientas diseñadas para complementar el trabajo de proyectos que implementan la metodología scrum, sin embargo no se cubren todas las necesidades requeridas por los usuarios, ya sea por el público a que está dirigido o costo de uso. Es por esto que con el objetivo de contrastar las diferencias entre las herramientas expuestas

y la que se pretende desarrollar en el trabajo se realiza una comparación en base a criterios de evaluación dirigidos tanto a herramientas colaborativas como a herramientas de teletrabajo. Los criterios de evaluación son útiles para diferenciar las cualidades de una y otra herramienta que se pretende utilizar.

2.4.1 CRITERIOS DE EVALUACIÓN PARA HERRAMIENTAS DE TELETRABAJO

Los criterios de evaluación de las herramientas de teletrabajo son los que se expone en libro base de teletrabajo del presente trabajo titulado “El libro Blanco” que es un de autoría del ministerio de tecnología de Colombia lanzado en el 2014, el teletrabajo se trata en detalle en el apartado 2.8.

2.4.1.1 CRITERIO DE ACCESIBILIDAD

Este criterio de evaluación hace referencia la facilidad de acceso a las herramientas. Para la facilidad de acceso se evalúa la infraestructura informática utilizada para el acceso a la herramienta ya sea los centros de datos propios o compartidos o la computación en la nube.

2.4.1.2 CRITERIO DE SEGURIDAD

Políticas de seguridad de las herramientas, protección de acceso a datos de las organizaciones que utilizan las herramientas.

2.4.1.3 CRITERIOS DE COMUNICACIÓN

Las herramientas de teletrabajo deben resolver el problema suplir en el mayor grado posible la necesidad de la comunicación cara a cara, por lo que un criterio de selección de una herramienta de teletrabajo es evaluar si se cuenta con aplicaciones tales como: soporte o instrucciones adicionales de manejo, chat, email, foros entre otros. A continuación

2.4.2 CRITERIOS DE EVALUACIÓN PARA HERRAMIENTAS COLABORATIVAS

Las herramientas colaborativas son una de las herramientas para teletrabajar, ya que además de estas se requiere de herramientas que lleven un registro de las actividades de los teletrabajadores, herramientas de comunicación, aplicaciones de soporte ya que un teletrabajador durante un tiempo necesitará el soporte necesario para aprender a utilizar las herramientas en un ambiente de teletrabajo, y aplicaciones de gestión de documentos. Las herramientas colaborativas para scrum son las mencionadas en los acápite 2.3.1, 2.3.2 y 2.3.3 en las páginas 30 y 31 por lo que se da a entender que las herramientas colaborativas por si solas no cubren todos los ámbitos para llegar a ser herramientas de teletrabajo, sin embargo para un análisis más detallado a continuación se muestra una tabla comparativa basada en criterios de evaluación tanto para las herramientas colaborativas (Graván, 2011) como para las de teletrabajo (Colombia, 2014), con el objeto de resaltar la diferencia entre la propuesta dada en el presente trabajo y las herramientas colaborativas para scrum existentes.

	Criterio Económico	Criterio Lingüístico	Servicios de Comunicación	Seguridad ofrecida	Requisitos de software	Dirección oficial
KUNAGI	Versión	Implementada	Foros de	Acceso según	Descarga de	kunagi.org

	Gratuita limitada	solo en inglés	discusión	una contraseña de proyecto	software para la instalación en un servidor	
SCRUMDO	Versión Gratuita limitada	Implementada solo en inglés	-----	Acceso según contraseña a los proyectos	Solo contar con internet	scrumdo.com
SPRINTOMETER	Versión Gratuita	Implementada en inglés y plugin para español	-----	Otorgada por claves de acceso además de la implementada por el servidor del usuario	Descarga de software para la instalación en un servidor	sprintometer.com
<i>El sistema del presente trabajo</i>	Gratis	En español	Chat opcional incorporado	Clave acceso proyecto, seguridad de infraestructura GAE	Solo contar con internet	-----

Tabla 2. 1 Tabla comparativa para herramientas colaborativas

De la tabla comparativa se saca la conclusión para el presente trabajo que ninguna cumple al 100% los criterios para ser una herramienta colaborativa enfocada en el teletrabajo, las que deben ser de fácil acceso además de que sean accesibles por usuarios hispano hablantes. La herramienta colaborativa que se pretende implementar en el presente trabajo se diferencia de las existentes en:

- Criterio lingüístico: desarrollada para usuarios que hablen español.
- Requisitos de software: se basa en los requisitos para una herramienta de teletrabajo es decir que sea de fácil acceso y en cualquier lugar.
- Servicios de comunicación: incorpora al menos un medio de comunicación entre los usuarios, que es uno de los criterios que las herramientas de teletrabajo deben cumplir.

Para contrastar más aun la herramienta que se implementara en el presente trabajo de las herramientas colaborativas para Scrum que existen se realiza una tabla comparativa de los criterios que debe tener una herramienta colaborativa para teletrabajo, dando a resaltar así cuales criterios son los que no toman en cuenta las herramientas colaborativas expuestas.

	Criterio de accesibilidad	Criterio de seguridad	Criterio de comunicación
KUNAGI	No es accesible a todo público, debido a que se requiere un servidor para correr el programa	Brinda una seguridad que depende del servidor donde se instale	Implementa foros de discusión
SCRUMDO	Es accesible desde la web, en su versión gratuita	Implementa seguridad de acceso a los proyectos creados	No implementa medios de comunicación

SPRINTOMETER	No es accesible a todo público, debido a que se requiere un servidor para correr el programa	Brinda una seguridad que depende del servidor donde se instale	No implementa medios de comunicación
SISTEMA A DESARROLLAR	Será accesible desde la web con solo contar con una cuenta google	Dos niveles, uno a nivel de usuario registrado en google y otro a nivel de proyecto creado en el sistema	Implementa la opción de hacer uso de un chat gratuito

Tabla 2. 2 Tabla comparativa de criterios para una herramienta colaborativa de teletrabajo

De la tabla 2.2 comparativa se concluye que el sistema se centra en los criterios que una herramienta colaborativa de teletrabajo debe cumplir de manera mínima.

2.5 SCRUM DISTRIBUIDO

2.5.1 DEFINICIÓN

Scrum distribuido es la variante de Scrum adaptada a ambientes distribuidos, donde los diferentes participantes del proyecto no comparten una misma ubicación física y/o temporal. Scrum distribuido atiende la problemáticas asociadas a las distintas variantes de distribución, además de matizar los distintos aspectos de Scrum (roles, artefactos y reuniones) poniendo énfasis espacial en las relaciones entre los miembros del equipo: particularidades de las reuniones, herramientas colaborativas, impacto en los roles y responsabilidades (Raúl Herranz, 2011).

El modo en que scrum distribuido atiende mediante recomendaciones a las problemáticas de un equipos distribuido se asienta como eje central en el uso de las telecomunicaciones y las TIC, esto es importante de resaltar ya que es parte de las definiciones de teletrabajo que se darán más adelante en el capítulo. En la figura 2.8 se muestra como los usuarios ubicados en distintos lugares están conectados gracias a la red de telecomunicaciones.



Figura 2.8: Equipos distribuidos geográficamente

Fuente: (Raúl Herranz, 2011)

2.5.2 TIPOLOGÍA DE DISTRIBUCIÓN

Dentro de un ambiente distribuido se dan distintas configuraciones entre los miembros del equipo, en base a distintos parámetros, los cuales pueden solaparse a continuación se menciona una clasificación de los distintos escenarios que pueden presentarse. (Raúl Herranz, 2011).

2.5.2.1 EQUIPOS DISTRIBUIDOS PARCIALMENTE SIMPLES

Es el escenario más favorable porque favorece los principios de desarrollo ágil, los miembros se encuentran en la misma situación física aunque en diversas ocasiones, los integrantes teletrabajan. (Raúl Herranz, 2011)

2.5.2.2 EQUIPOS DISTRIBUIDOS PARCIALMENTE COMPLEJOS

Este escenario aún conserva condiciones para llevar a cabo un trabajo colaborativo de forma remota favorable puesto que se tiene un mismo franja horaria y los miembros pueden trabajar desde casa de forma permanente. (Raúl Herranz, 2011)

2.5.2.3 EQUIPOS DISTRIBUIDOS TEMPORALMENTE SIMPLES

Los miembros no comparten una franja horaria en común y por tanto ubicación geográfica, sus franjas horarias tienen unas pocas horas de diferencia, esto último hace posible que se pueda planificar la comunicación en ventanas de tiempo acordadas. (Raúl Herranz, 2011)

2.5.2.4 EQUIPOS DISTRIBUIDOS TEMPORALMENTE COMPLEJOS

En este escenario la dificultad es total, puesto que la comunicación es un obstáculo casi insalvable debido a que las franjas horarias varían de doce horas en adelante, aun con las tecnologías de colaboración como foros, correos electrónicos entre otras

2.5.3 ROLES DISTRIBUIDOS

Scrum Distribuido da definiciones de los roles desde su ámbito distribuido, puesto que este ambiente particular afecta en las responsabilidades de cada rol, sin embargo se mantiene todas las responsabilidades que se dan en scrum técnico, ya definido.

2.5.3.2 SCRUM MANAGER

a) DEFINICIÓN

En un ambiente distribuido debe encargarse de la red de comunicaciones formada por las relaciones humanas y herramientas de comunicación seleccionadas para el teletrabajo.

b) RESPONSABILIDADES

- Asesorar y formar sobre las herramientas que se usan para trabajar a distancia

- Acceso a las conversaciones de los miembros de equipo del proyecto ya que es el encargado de moderar el buen desenvolvimiento de estas.
- Gestión de dudas con respecto a la práctica de scrum.
- Acceso visual a la pila del producto mediante herramientas colaborativas de tal manera de revisarla y validarla.
- Hacer cumplir las reglas de scrum mediante una continua inspección de las comunicaciones y las prácticas que se lleva a cabo.

2.5.3.2 EQUIPO DE DESARROLLO

a) DEFINICIÓN

En el ambiente distribuido el equipo de desarrollo debe mostrar un mayor compromiso con sus quehaceres personales y mayor disciplina, además de mayor responsabilidad de interacción.

b) RESPONSABILIDADES

- Revisión de la visión general del proyecto, que el dueño del producto difunde a través de la red de comunicación o herramientas colaborativas.
- Participación en la preparación de la pila del producto usando las herramientas de comunicación seleccionadas.
- Estar enterados de los avances tanto a nivel de pila del producto como a nivel de sprint, mediante la revisión de métricas del proyecto puestas en línea.
- Conocen las prácticas de la metodología ágil scrum, y en caso de dudas recurren al scrum máster.

2.5.3.3 PROPIETARIO DEL PRODUCTO

a) DEFINICIÓN

Al igual que en el entorno tradicional de Scrum, este tiene como una de sus funciones facilitar la comunicación, la fluidez de la información entre los integrantes del equipo, definir un objetivo concreto y realizable, participar activamente en las reuniones con los equipos, involucrarse e involucrar a los miembros en el proyecto.

b) RESPONSABILIDADES

- Poner a disposición de los miembros del proyecto, la visión general de proyecto, a través de las herramientas disponibles.
- Atender a las dudas que surjan en relación con la visión general

- Se encarga de la construcción de la pila del producto mediante las herramientas colaborativas disponibles con ayuda de los miembros del proyecto.

2.5.3.4 STAKEHOLDERS

Son aquellas personas o grupos que forman del entorno global y que tienen intereses en el proyecto además de influenciarlos, por tanto estos no tiene nada que cumplir en el proyecto, al contrario lo que se debe hacer con estos es: identificarlos, comprometerlos en el proyecto, comunicarse con ellos.

2.5.4 EVENTOS DISTRIBUIDOS

Las reuniones que se han definido en scrum técnico se mantienen en ambientes distribuidos con la inclusión de las siguientes consideraciones para el trabajo a distancia:

- Establecer límites temporales de manera explícita
- Realizar registros de asistencia de los participantes
- Control por parte del rol que corresponda de que las reuniones no se salgan de los objetivos iniciales.

2.5.5 ARTEFACTOS EN AMBIENTES DISTRIBUIDOS

Los artefactos tradicionales como la pila del producto, pila del sprint se mantienen, con la diferencia de que su elaboración, visualización, gestión, y actualización, estarán sujetas a la tecnología de comunicación que se utilice para el trabajo a distancia, así estas pueden ir desde simples hojas de cálculo en google docs hasta sistemas colaborativos disponibles en línea.

2.5.6 HERRAMIENTAS DE COMUNICACIÓN COMO ESTRATEGIA

En los ambientes distribuidos existen problemas de comunicación, entre estos se puede mencionar:

- Exceso de comunicación formal.
- Inadecuado soporte en ambientes distribuidos
- Carencia de reuniones cara a cara
- Desconocimiento de aspectos culturales de ambas partes
- Problemas de lenguaje

No obstante se recomienda herramientas de software, para conseguir la comunicación de la mejor manera posible en un equipo:

- Email: comunicación asíncrona
- Mailing List: comunicación de broadcasting asíncrona

- Mensajería Instantánea: comunicación en tiempo real, basada en texto.
- Conference Calls: conferencias telefónicas entre diversos miembros
- Skype: alternativa a Conference Calls, con opciones gratuitas.
- Video conferencias: conferencias audio/video entre diferentes sedes
- Escritorios Compartidos: interacción persona – maquina distribuida.
- Presentaciones: para consolidar ideas y metas

2.6 METODOLOGÍA ÁGIL DE DESARROLLO DEL SISTEMA AUP

En este apartado se explica y estudia los fundamentos de la metodología de desarrollo que se utilizara en el marco aplicativo, AUP cuya abreviación significa “Procesos Unificado Ágil” la cual es una metodología ágil de desarrollo lo que la hace ideal dado que el prototipo a desarrollar es una aplicación web y por ende requiere adaptabilidad a su entorno y agilidad en su implementación.

2.6.1 DEFINICIÓN

AUP describe una manera simple de desarrollar software, aplicando técnicas ágiles y conceptos de UP. Como una de sus prioridades se ocupa en primer lugar de los riesgos y cómo gestionarlos, mediante la elaboración de prototipos durante el desarrollo del producto. AUP se dirige por casos de uso, los mismos que se utilizan en UML, esto con el fin de centrarse en la funcionalidad que los requisitos del usuario demanda además el hecho de usar los diagramas de casos de uso es porque tiene el trasfondo de orientar las actividades de desarrollo. (AMBLER, 2005)

2.6.2 FASES DE AUP

El ciclo de vida de AUP, se describe como el cumplimiento de cuatro fases consecutivas. Las cuales a su vez están divididas en iteraciones que cumplen con la práctica de una serie de disciplinas de desarrollo. En cada iteración se entrega un avance del producto, de manera usual la primera entrega tarda más que el resto debido a que en la fase inicial se realiza el acopio de requisitos, además del modelado y diseño a implementar. En la figura 2.9 se observa las fases de construcción en las columnas para AUP, además de que en las filas están las prácticas que se realizan por cada fase. Debe entenderse el grafico partiendo de las columnas, por ejemplo la fase de inicio englobara todas las filas o practicas pero en un grado mayor o menor representado por las curvas internas, así inicio hace énfasis en la práctica de modelado. (AMBLER, 2005)

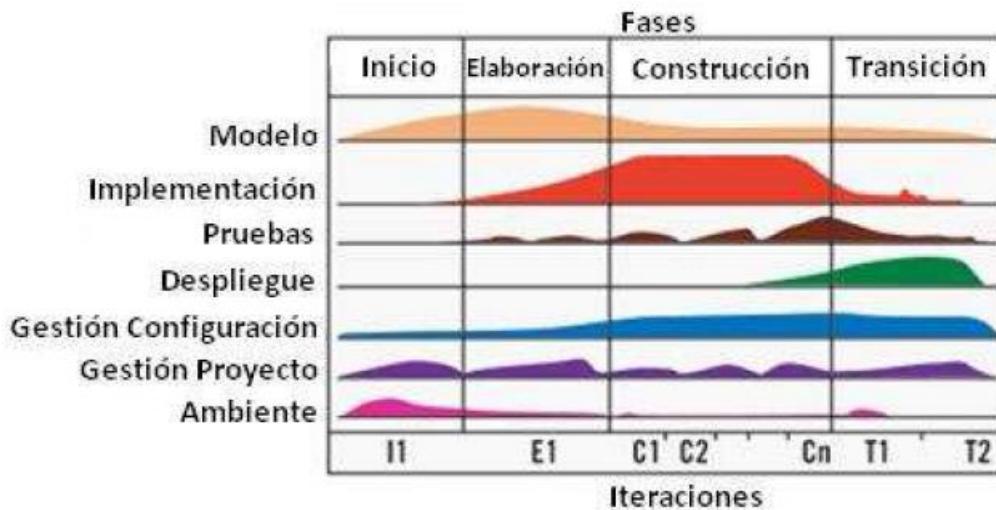


Figura 2.9: Ciclo de vida de AUP

Fuente: (Flores, 2009)

2.6.2.1 INICIO

El objetivo de esta fase es obtener una comprensión común entre el cliente y el equipo de desarrollo. Las actividades que se destacan en esta fase según el autor de la metodología Scott Ambler, se detallan en la tabla 2.3.

ACTIVIDAD	TAREAS A REALIZAR
Definir el alcance del Proyecto	En esta actividad se define los alcances del sistema a implementar, indicando lo que realizará este a un alto nivel.
Estimación de Costos y Calendario	Se realiza una estimación a alto nivel de los costes y calendario, pudiendo modificarse luego en el desarrollo del proyecto.
Definición de Riesgos	La metodología AUP se caracteriza por realizar una gestión de riesgos que puedan atrasar el proyecto, lo que se realiza en esta actividad es la identificación de estos riesgos y las acciones a realizar para neutralizarlos.
Determinar Factibilidad del Proyecto	Se analiza las posibilidades reales de llevar a cabo el proyecto, y se desechará las cosas que son imposibles de realizar.
Preparar el Entorno del Proyecto	En esta actividad se especifica personal, área física y requerimientos de software y hardware

Tabla 2. 3: Actividades a realizar en la fase de Inicio

2.6.2.2 ELABORACIÓN

En esta fase el equipo debe profundizar y comprender los requisitos del sistema, además de la validación de la arquitectura. Las actividades que se realizan en esta fase solo son de una especificación mediante modelados de los riesgos identificados en la fase anterior, además de una preparación para la fase de construcción.

2.6.2.3 CONSTRUCCIÓN

En esta fase el sistema es desarrollado y probado en el ambiente de desarrollo. La forma de desarrollarlo es de forma incremental, lo que quiere decir que se entregan prototipos funcionales en cada iteración al interior de esta fase. En la construcción de los prototipos en cada iteración se repite una serie de pasos que se asemejan a mini proyectos al interior de cada iteración (AMBLER, 2005). En la figura 2.10 se observa esta serie de pasos:

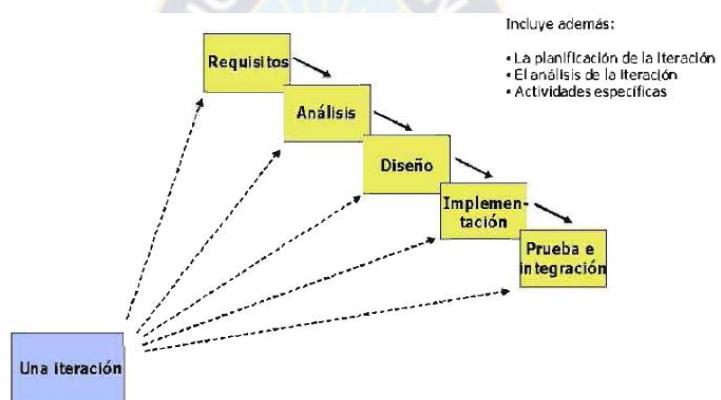


Figura 2. 10: Tareas que se realizan en cada iteración

2.6.2.4 TRANSICIÓN

En esta fase el sistema se lleva a los entornos de producción, donde se somete a pruebas de validación, para el despliegue en los sistemas de producción. Las tareas que se llevan a cabo son: pruebas beta, modificaciones significativas, y corrección de errores.

2.6.3 ARTEFACTOS

Los únicos artefactos en los que se apoya AUP, son los casos de uso y los modelos que de este se pueden derivar en un modelado con el lenguaje UML, la razón del porque no se especifica más artefactos como el proceso unificado UP, es porque AUP se ha diseñado con el objeto de tener procesos ágiles centrados en el producto y no así en la documentación. En la figura 2.11 se muestra los modelos usuales al desarrollar con el lenguaje UML.

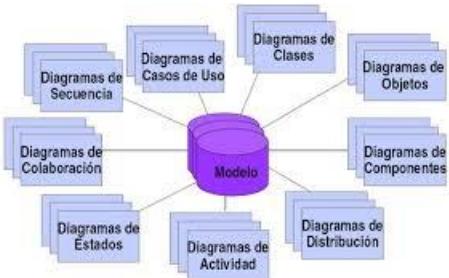


Figura 2. 11: Artefactos de la metodología AUP

2.7 SISTEMAS WORKFLOW

2.7.1 DEFINICIÓN

Los workflow son un tipo especial de groupware La palabra workflow en inglés significa traducida al castellano flujo del trabajo, y es esto precisamente en resumen lo que realizan los sistemas workflow, se define a un workflow como una secuencia de tareas desempeñadas en serie o en paralelo de manera lógica a esto se le llama proceso de trabajo, los procesos del trabajo son realizados por dos o más miembros de un equipo para lograr una meta en común. Como ejemplo de procesos del trabajo se tiene: procesamiento de órdenes, reportes de gastos, procedimientos de producción. (Gerónimo, 2002)

2.7.1.1 FUNCIONES

Las funciones comunes que proporcionan los workflow son: (Gerónimo, 2002)

- Asignación de tareas al personal
- Aviso de tareas pendientes
- Permitir la colaboración en tareas comunes.
- Optimización de recursos humanos y técnicos alineándolos a los principios de la empresa.
- Automatización de las secuencias de procesos del negocio.
- Agilización de procesos del negocio.
- Control y seguimiento de los procesos del negocio.

2.7.2 CLASIFICACIÓN

Debido a la diversidad de procesos del negocio, que existen dentro de las organizaciones, se tiene la siguiente clasificación de workflow: de producción, de colaboración, y de administración. (Gerónimo, 2002)

2.7.2.1 WORKFLOW DE PRODUCCIÓN

Son los workflow más complicados puesto que requieren de una mayor estructuración puesto que los usuarios suelen tener funciones bien definidas, con respecto a su actividad dentro de la organización, un ejemplo es cuando atiende a las transacciones que se efectúan en una base de datos, es decir que su función es la de automatizar los procesos de negocios que tienden a ser repetitivos, estructurados, con gran manejo de datos. (Gerónimo, 2002)

2.7.2.2 WORKFLOW DE COLABORACIÓN

Son aquellos que resuelven procesos del negocio donde participa la gente para lograr un objetivo común, es decir que estructuran o semi estructuran los procesos del negocio donde participan las personas, en su forma tradicional se trabaja sobre documentos que son los contenedores de la información, y la forma de trabajarlos es siguiendo la ruta de estos paso a paso, con objeto de mantener su integridad. Se caracterizan por que permiten la creatividad de los usuarios es decir que en su diseño son flexibles y tolerantes al cambio se aplican en: instituciones financieras, empresas consultoras entre otros. (Gerónimo, 2002)

2.7.2.3 WORKFLOW DE ADMINISTRACIÓN

Son aquellos que involucran procesos de administración en una empresa tales como órdenes de compra, reportes de ventas, entre otros, estos workflow son empleados donde existe una gran cantidad de procesos administrativos. Se caracterizan por trabajar sobre procesos repetitivos y estructurados. (Gerónimo, 2002)

2.8 MODELADO DE PROCESOS DE UN WORKFLOW CON BPMN

BPMN¹⁰ es un lenguaje de especificación de procesos del negocio y servicios web, este lenguaje es considerado un estándar en la definición de procesos porque está a cargo de la OMG, se caracteriza por ser fácilmente comprensible por analistas, desarrolladores, e incluso el cliente puesto que su notación gráfica es sencilla de entender. Este lenguaje de especificación se ocupa únicamente de los procesos del negocio lo que significa que otro tipo de modelos como la organización, recursos, modelos de datos, entre otros quedan fuera de la especificación. (Jimenez, 2008)

Por otro lado la OMG también define como herramienta de definición de procesos del negocio a los diagramas de actividades de UML en su versión 2.X, sin embargo la OMG da más prioridad a BPMN debido a su simplicidad y nivel de expresividad, además de

¹⁰ BPMN por sus siglas en inglés que significa notación para el modelado de procesos del negocio.

que BPMN tiene apoyo de la organización WFMC que es en con la que se guía este trabajo en la definición y posterior modelado del workflow, que dicho sea de paso la WFMC es miembro de la OMG. (Jimenez, 2008)

Desde el punto de vista de BPMN el modelado de procesos del negocio es la captura de la secuencia de actividades del negocio, es decir la manera como una empresa alcanza sus objetivos, por lo que la lógica de cómo se logran las tareas queda fuera del detallado de las secuencias a seguir. Cuando se modela los procesos existen diferentes niveles de modelado, lo que permite un modelado de tipo jerárquico. (Analitica, 2007)

- Mapas de proceso: son diagramas simples del flujo de actividades, diagramas sin más detalle que el nombre de las actividades y tal vez las condiciones de decisión más generales.
- Descripción de proceso: conforman una extensión de los anteriores y manejan información adicional como las personas involucradas en llevar a cabo los procesos “roles”, los datos, la información, entre otros
- Modelos de proceso: diagramas de flujo extendidos con suficiente información para que el proceso pueda ser analizado, simulado, y/o ejecutado, por herramientas de software.

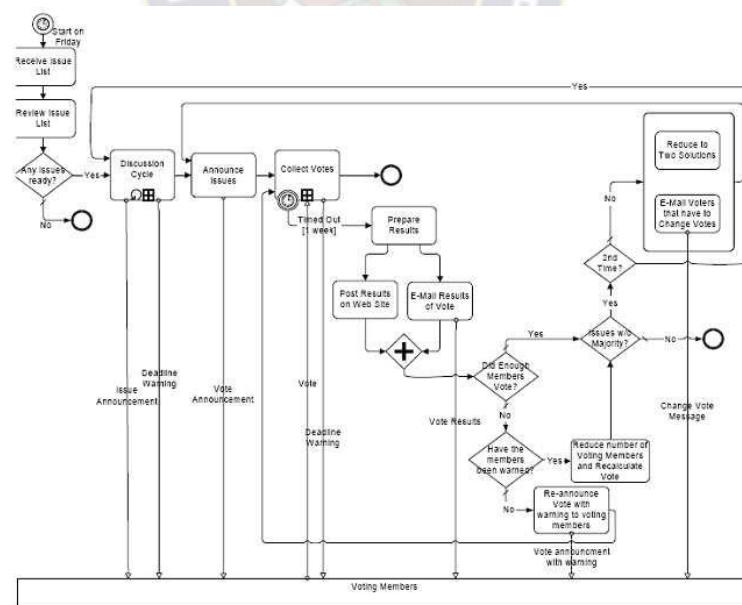


Figura 2.12: Gráfico de un diagrama de modelado de procesos con BPMN

Fuente: (Jimenez, 2008)

En la figura 2.12 se muestra un ejemplo de un modelo de procesos con el lenguaje de especificación BPMN, se puede apreciar que cuenta con una cantidad suficiente de

elementos gráficos para representar los procesos de decisión, tareas, inicio de procesos, agrupaciones entre otros que se desarrollan en los siguientes apartados.

2.8.1 MODELOS DE PROCESOS EN BPMN

Los modelos de BPMN expresan y sirven para diferenciar los tres tipos de procesos que se consideran en este lenguaje. Los modelos se expresan mediante diagramas y son 3. (Jimenez, 2008)

2.8.1.1 PROCESOS DE NEGOCIO PRIVADOS

Estos procesos privados del negocio son los que al interior de las organizaciones son conocidos por los colaboradores como parte de su trabajo diario, así por ejemplo el proceso de aprobación de una solicitud la que debe pasar por distintas etapas de aprobación, se muestra este ejemplo en la figura 2.13, estos tipo de procesos pueden interactuar con los otros tipos de procesos definidos.



Figura 2.13: Modelo de proceso del negocio privado

Fuente: (Jimenez, 2008)

2.8.1.2 PROCESOS DEL NEGOCIO ABSTRACTOS

Los procesos abstractos representan las iteraciones que tienen un proceso privado con otro proceso privado, u otro participante del proceso. En este tipo de procesos solo se muestran las actividades del proceso privado que comunican algo al exterior del proceso.

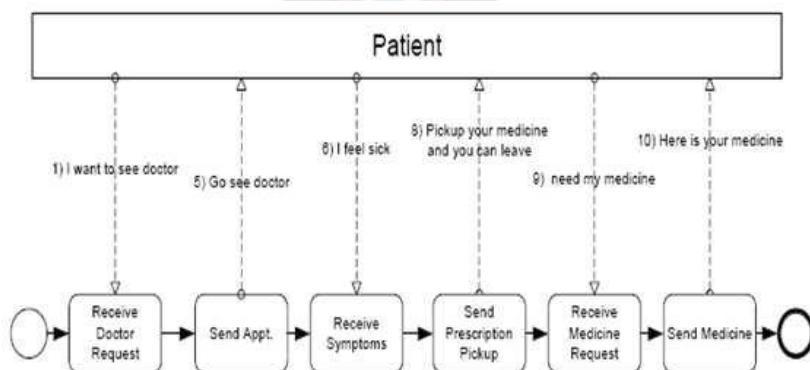


Figura 2.14: Modelo de procesos del negocio abstracto

Fuente: (Jimenez, 2008)

La figura 2.14 muestra un proceso abstracto que representa como un paciente interactúa con un proceso privado de una clínica que aprueba las citas con el médico.

2.8.1.3 PROCESOS DE COLABORACIÓN

La colaboración en BPMN se entiende como la comunicación entre dos o más procesos del negocio, por tanto un proceso de colaboración sirve para mostrar la interacción y comunicación mediante mensajes entre los procesos privados del negocio.

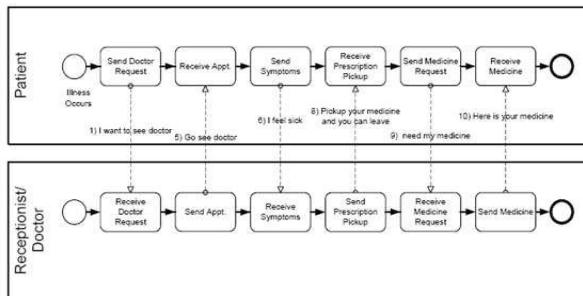


Figura 2.15: Modelo de procesos de colaboración

Fuente: (Jimenez, 2008)

En la figura 2.15 se observa un proceso de colaboración entre los procesos personales que llevan a cabo un paciente y los correspondientes a los de una institución de sanidad.

2.8.2 DIAGRAMAS BPMN

Como se mencionó BPMN es un lenguaje estándar para la especificación de procesos del negocio de manera gráfica, para esto se vale de diagramas los cuales tienen una clasificación según los modelos de procesos ya mencionados a continuación se hace una presentación de los elementos básicos de estos diagramas.

2.8.2.1 ELEMENTOS BÁSICOS DE LOS DIAGRAMAS BPMN

Los elementos básicos para la construcción de diagramas en BPMN se clasifican en cuatro categorías según la funcionalidad que prestan.

a) OBJETOS DE FLUJO (FLOW OBJECTS)

Los objetos de flujo son aquellos que permiten representar las etapas por las que pasa la información durante el flujo de la información. En BPMN se tiene tres tipos de estos objetos para evitar que los modeladores de flujos no deban memorizar una gran cantidad de objetos.

i) EVENTOS (EVENTS)

Este tipo de objetos representan acciones que pueden suceder durante el transcurso del proceso del negocio, se representan con círculos, de acuerdo al momento en que afectan al flujo se dividen en tres tipos: inicio, intermedio, y fin. En la tabla 2.4 los elementos de tipo evento.

	INICIO	INTERMEDIO	FINAL	
	Envío		Recepción	
Plan	○	○		○
Mensaje	✉	✉	✉	✉
Tiempo	⌚	⌚		
Error		⚡		⚡
Cancelar		✖		✖
Compensación		◀	◀	◀
Condicional	☰	☰		
Señal	△	△	▲	▲

Múltiple					Maneja la ocurrencia de uno o varios eventos, ya sean mensajes, señales entre otros.
----------	--	--	--	--	--

Tabla 2. 4: Elementos de tipo evento del estándar de modelado BPMN

Fuente: (Institut, 2012)

Los eventos de inicio están presentes en todo proceso o subprocesso, estos eventos de inicio pueden ser de los tipos mostrados arriba en la tabla, así por ejemplo un evento de inicio de mensaje podría ser por ejemplo cuando un proceso o aplicativo se manda un

mensaje para iniciar un proceso específico y se representaría por el símbolo:



Los eventos intermedios son los que están a la mitad del flujo o comienzo, por ejemplo el

símbolo que representa un retraso de tiempo en unidades de hrs, días entre otros.

ii) ACTIVIDADES (ACTIVITY)

Las actividades representan los trabajos que llevan a cabo los miembros de la organización es decir que el nombre genérico que se da a las tareas es el de actividades, estas pueden ser atómicas o compuestas, a continuación en la tabla 2.5 se resumen las actividades del modelo BPMN.

TIPO ACTIVIDAD	GRÁFICOS	DESCRIPCIÓN
Tarea	 	Tarea es una actividad atómica que está presente dentro de un proceso los hay de siete tipos: tareas de usuario donde interviene el mismo, tareas de servicio son las que realiza el sistema sin intervención humana y se ejecutan automáticamente (es decir ni si quiera un click debería realizar el usuario) en cuanto llega el flujo. Las tareas abstractas no tienen

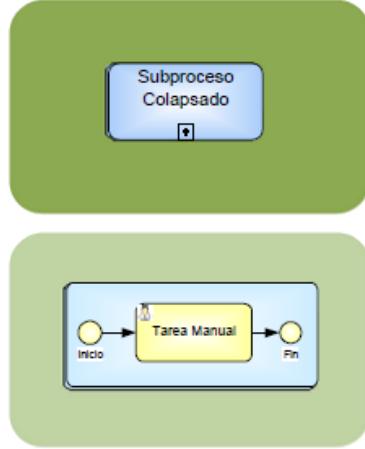
		tipo definido. Tareas de script representa una función automatizada al interior del motor de procesos. Tareas de servicio son aquellas que las realizan sistemas externas al motor de los procesos.
Sub proceso		Es un conjunto de actividades incluidas dentro de un proceso relacionadas entre sí, se presentan de dos modos el modo colapsado es cuando no se puede ver detalles internos del sub proceso debido al nivel que implica, y el expandido que muestra detalles porque esta al mismo nivel de detalle del proceso.

Tabla 2. 5: Elementos de tipo actividad del estándar de modelado BPMN

Fuente: (Institut, 2012)

iii) PASARELAS (GATEWAY)

Las pasarelas tienen el objeto de direccionar el flujo de la información, con el objeto de controlar la convergencia, divergencia, ramificaciones, bifurcaciones, combinaciones, y fusiones al interior del proceso esta variedad hace que sean ideales para la representación de procesos del negocio. En la figura 2.16 muestra las distintas variaciones de los elementos gráficos pasarelas.



Figura 2. 16: Elemento gráfico de modelado con BPMN, las pasarelas

Fuente: (Jimenez, 2008)

b) CONECTORES (CONNECTING OBJECTS)

Son gráficos que representan la interacción con entre los objetos de flujos de trabajo, con el objeto de crear el esqueleto estructural de los procesos del negocio existen tres tipos de conectores.

i) FLUJO DE SECUENCIA

Elemento utilizado para indicar el orden en el cual son ejecutadas las actividades del proceso del negocio.



ii) FLUJO DE MENSAJE

Para mostrar el intercambio de mensajes entre dos participantes ya sean entidades de negocio o roles.



iii) ASOCIACIÓN

Para asociar artefactos con flujos de trabajo



c) CALLES (SWIMLANES)

Son un mecanismo que nos permite clasificar las actividades o tareas de manera visual, para ilustrar diferentes capacidades funcionales o responsabilidades. BPMN soporta dos tipos de calles principalmente:

i) LANE (“EN LÍNEA”)

Representa un participante dentro de un proceso, el cual contiene un conjunto de actividades asociadas a este rol.

ii) POOL (“AGLUTINADOR”)

Representa los actores externos con los que interactúa el proceso o aplicativo. En la figura 2.17 se muestra los dos tipos de aglutinadores especificados a la fecha.

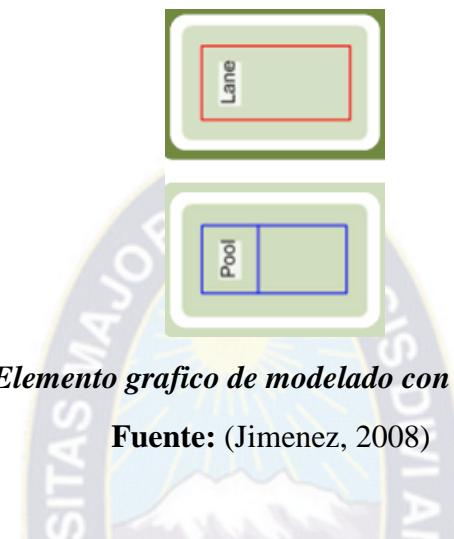


Figura 2. 17: Elemento grafico de modelado con BPMN, las piscinas

Fuente: (Jimenez, 2008)

d) ARTEFACTOS (ARTIFACTS)

Los artefactos son un tipo de elementos diseñados con el fin de representar conceptos de uso común como: documentos, información o comentarios adicionales. Estos artefactos no afectan el flujo de la información. La figura 2.18 muestra los artefactos especificados a la fecha.

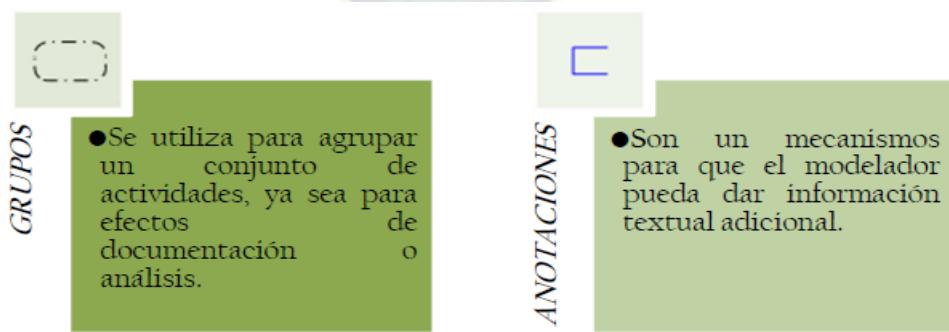


Figura 2. 18: Elemento gráfico de modelado con BPMN, agrupadores

Fuente: (Jimenez, 2008)

2.9 TELETRABAJO

En este apartado se desarrolla el fundamento teórico necesario para entender que es el teletrabajo desde los distintos puntos de vista de los estados que han adoptado esta modalidad de trabajo, hasta los proyectos al respecto en nuestro país.

2.9.1 DEFINICIÓN

Las definiciones de esta modalidad de trabajo son diversas, ya sean por distintos autores, investigadores, leyes estatales, entre otros. Se tomara en cuenta las definiciones de la Organización internacional del trabajo OIT por ser la entidad especializada en asuntos relativos al trabajo y relaciones laborales, también se toma en cuenta la definición que da la organización FUNDETIC, que es una institución boliviana fundada en el año 2010 con el fin de promover y estimular actividades relacionadas con el desarrollo de las TIC, además de participar en el desarrollo del gobierno electrónico mediante el uso y difusión de las TIC, y por último se toma en cuenta la definición dada en el libro blanco que es una guía de implementación metodológica del teletrabajo editado en Colombia en 2012 debido a que se basa en experiencias internacionales además de que ya se puso en práctica por organizaciones bancarias de ese país. Este libro se toma como base en el desarrollo de este trabajo.

2.9.1.1 DEFINICIÓN SEGÚN LA OIT

El teletrabajo es la forma de trabajo efectuado en un lugar distante a la oficina y centro de producción que permita la separación física del trabajador, que implique el uso de una nueva tecnología facilitadora de la comunicación.

2.9.1.2 DEFINICIÓN SEGÚN FUNDETIC

El teletrabajo es una nueva forma de organizar el trabajo realizado a distancia, mediante el uso de las TIC, este trabajo se puede realizar en el hogar o en otro lugar distante a la organización.

2.9.1.3 DEFINICIÓN SEGÚN EL LIBRO BLANCO

Es una forma de organización laboral que consiste en el desempeño de actividades remuneradas y prestación de servicios a terceros utilizando como soporte las TIC, para el contacto con el trabajador y la empresa, sin requerir la presencia física del trabajador en un sitio específico de trabajo.

2.9.2 ESTADO DE LA CIENCIA

El estado de desarrollo del teletrabajo puede ser visto desde distintos aspectos, así por ejemplo: aspecto legal, aspecto político, aspecto social, aspecto tecnológico.

2.9.2.1 DESDE EL ASPECTO SOCIAL

Desde esta visión, el teletrabajo al igual que en sus inicio tiene entre sus prioridades el ahorro energético, además se han unido otras más como las de inclusión e igualdad social, mejor calidad de vida de los trabajadores, empoderamiento de las nuevas

tecnologías de la información y comunicación entre otras que los poderes políticos van proponiendo según sus necesidades.

2.9.2.2 DESDE EL ASPECTO LEGAL

El aspecto legal es un elemento crucial a la hora de adoptar el teletrabajo dado que al tratarse de una forma laboral, se dan una serie de aspectos particulares relacionados a los derechos de teletrabajadores, sus deberes, formas en las que se debe llevar a cabo, horarios, igualdad de derechos con los trabajadores en planta entre otros. Prueba de ello son las políticas y proyectos de ley en los distintos países de la región latinoamericana como chile y Colombia.

2.9.2.3 DESDE EL ASPECTO TECNOLÓGICO

El aspecto tecnológico es un factor que es ineludible dadas las definiciones ampliamente aceptadas en las que se basa el teletrabajo. En la actualidad las TIC son parte de este aspecto de la implementación del teletrabajo, por tanto la base tecnológica de teletrabajo está asentada en las redes de internet y las tecnologías que estas soportan, en especial la computación en la nube dado que soporta la virtualización que es una de sus propiedades, esta propiedad permite simular hardware especializado mediante programas informáticos como pueden ser los servidores además de poner a disposición software especializado al público con el único requisito de tener acceso a internet.

2.9.3 TELETRABAJO EN BOLIVIA

En Bolivia el teletrabajo está en el estado de proyecto de ley desde el 12 de mayo de 2014 que es el día internacional del internet, este proyecto de ley está a cargo de la organización FUNDETIC. Sin embargo ya existen normativas dirigidas a la adopción de las TIC, así en año 2011 se establece el régimen general para: telecomunicaciones, TIC, servicio postal y aduanas. Por tanto en el país falta la voluntad política como inicio para el desarrollo adecuado de esta forma de trabajo.

2.9.4 MODALIDADES DE TELETRABAJO

Según la definición del libro blanco las modalidades de teletrabajo responden a los espacios de ejecución del teletrabajo, las tareas a ejecutar, y el perfil del trabajador. Atendiendo al perfil del trabajador se tiene las siguientes modalidades.

2.9.4.1 TELETRABAJO AUTÓNOMO

Son trabajadores independientes o empleados que para desarrollar sus tareas utilizan las TIC, ejecutándolas desde cualquier lugar elegido por ellos.

2.9.4.2 TELETRABAJO SUPLEMENTARIO

Son los trabajadores que cuentan con un contrato laboral, que alternan su trabajo en distintos días de la semana entre la empresa y un lugar elegido por ellos.

2.9.4.3 TELETRABAJO MÓVIL

Trabajadores que utilizan teléfonos móviles u otros aparatos portables para ejecutar sus tareas, sus actividades les permiten ausentarse con frecuencia de la oficina.

2.9.5 IMPLEMENTACIÓN DEL TELETRABAJO EN UNA ORGANIZACIÓN

La implementación del teletrabajo en una organización debe tomar en cuenta 3 factores: gestión del cambio organizacional, definición de la infraestructura y plataformas tecnológicas que soportaran el teletrabajo, atención a la legislación vigente en materia jurídica de riesgos laborales y relaciones con los sindicatos.

2.9.5.1 REQUERIMIENTOS PARA LA IMPLEMENTACIÓN

Producto de la revisión de los modelos empleados a nivel mundial, uno de los modelos que se sugiere a la hora de la implementación de este modelo es tomar en cuenta el compromiso institucional, desarrollo de una planeación, y el proceso de prueba con una validación correspondiente. El proceso de implementación según el modelo colombiano debería seguirse en los siguientes cinco pasos, mostrados en la figura 2.19:

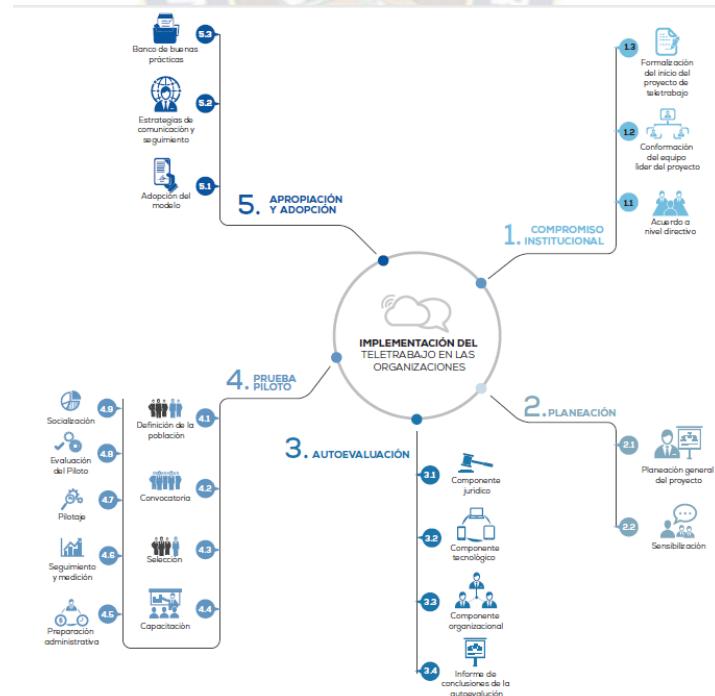


Figura 2. 19: Visión general de los pasos a seguir y los requisitos para implementar el teletrabajo en una organización.

Fuente: (Colombia, 2014)

a) COMPROMISO INSTITUCIONAL

Para la adopción del teletrabajo es indispensable el apoyo de la alta dirección. La iniciativa puede venir de cualquiera de las áreas en que se divide la organización, entre las actividades que se deben desarrollar en esta etapa se tiene:

- Acuerdo a nivel directivo
- Conformación de un equipo líder del proyecto
- Formalización del inicio del proyecto



Figura 2. 20: Requisito de compromiso institucional

Fuente: (Colombia, 2014)

En la figura 2.20 se observa las actividades mínimas para llevar a cabo el compromiso institucional.

b) PLANEACIÓN DEL PROYECTO

Como todo proyecto la implementación del teletrabajo requiere una planificación que oriente los logros de los objetivos, garantice los recursos necesarios y mitigue los riesgos posibles para la organización. Para esto se necesita una planeación general de proyecto y su sensibilización. En la figura 2.21 se muestra las actividades mínimas para la planeación de un proyecto de teletrabajo.



Figura 2. 21: Requisito de planificación del proyecto

Fuente: (Colombia, 2014)

c) AUTO EVALUACIÓN

Con objeto de identificar la viabilidad de la adopción del teletrabajo, la organización debe realizar un diagnóstico sobre sus condiciones y requisitos iniciales, esta evaluación se recomienda que se enmarque en los tres componentes que propone el modelo: jurídico, tecnológico, y organizacionales. Los pasos para la auto evaluación de las organizaciones para implementar el teletrabajo se muestran en la figura 2.22.



Figura 2. 22: Requisito de autoevaluación

Fuente: (Colombia, 2014)

d) PRUEBA PILOTO

Es recomendable probar como se adapta el modelo de teletrabajo a la organización, esta etapa de prueba debería tener una duración de cuatro a seis meses, para analizar: acciones de mejora, costos, control de riesgos, y la aplicabilidad del modelo. Esta etapa de la implementación es la más trabajosa y comprende los siguientes nueve pasos que se muestran en la figura. Los pasos mínimos para la prueba piloto se muestran en la figura 2.23



Figura 2. 23: Requisito de prueba piloto

Fuente: (Colombia, 2014)

e) APROPIACIÓN

Una vez pasada la etapa de la prueba lo que queda es la adopción definitiva del modelo definitiva del modelo, para lo que se hará un ajuste definitivo al plan general de teletrabajo y los documentos que sustentaran esta modalidad laboral al interior de la organización, además de un seguimiento permanente y se conformara un banco de

preguntas de buenas prácticas. Los pasos de la apropiación del modelo de teletrabajo se muestran en la figura 2.24



Figura 2. 24: La etapa de apropiación

Fuente: (Colombia, 2014)

2.9.6 COMPONENTE TECNOLÓGICO

En el contexto del teletrabajo, la tecnología hace referencia a la combinación de: la infraestructura, dispositivos. En las etapas de implementación del teletrabajo se vio que en la parte de la autoevaluación uno de los pasos era ver el componente tecnológico con que cuenta la organización, se hace una analiza este factor en especial puesto que el presente trabajo ésta orientado a desarrollar parte de la tecnología para un equipo de desarrollo de software que teletrabaja bajo las reglas y procesos de la metodología scrum distribuido. La evaluación de este componente deberá realizarse tomando en cuenta dos pasos que se realizan de manera consecutiva: primero se debe evaluar los puestos que se quiere que sean teletrabajables, para esto se debe identificar qué es lo que requiere el trabajador en su escritorio físico y tomar estos requerimientos como los que se debe dotar al trabajador en su lugar de trabajo a distancia. Como segundo paso se debe evaluar los elementos tecnológicos con que se cuenta en la empresa, para evaluar el nivel de preparación tecnológica de la organización, entre los elementos tecnológicos a evaluar se recomienda tomar en cuenta: infraestructura de red, telefonía, acceso a internet, seguridad de la información, una mesa de ayuda, y los procesos de administración para el uso de las herramientas. En la figura 2.25 se muestra los componentes tecnológicos a tomar en cuenta para el modelo de teletrabajo.

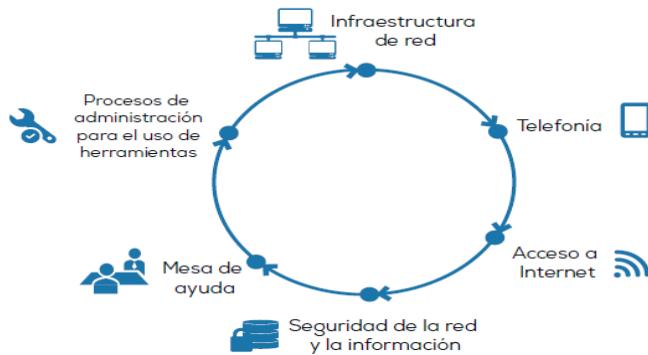


Figura 2. 25: El componente tecnológico

Fuente: (Colombia, 2014)

2.9.7 IDENTIFICACIÓN DE SOLUCIONES TECNOLÓGICAS

La identificación de las herramientas tecnológicas necesarias para teletrabajar depende de los recursos disponibles de cada organización, las necesidades particulares de cada perfil de trabajador, estas incógnitas pueden responderse mediante las siguientes preguntas:

- ¿Qué tipo de labor realizará remotamente el teletrabajador?
- ¿Qué sistemas de la entidad deberán ser utilizados por el teletrabajador para efectuar estas labores?
- ¿Cómo puede el teletrabajador acceder a los sistemas requeridos?
- ¿Qué nivel de seguridad se requiere para cada sistema accedido remotamente?
- ¿Cómo se puede cumplir con los niveles de seguridad de la información definidos en las Políticas de la organización?
- ¿Desde dónde y con qué dispositivo accederá el teletrabajador a los sistemas?
- ¿Cómo se mantendrá contacto con el teletrabajador?
- ¿Cómo podrá el teletrabajador interactuar con el equipo de trabajo?

2.10 EL TELETRABAJO BAJO LA METODOLOGÍA SCRUM DISTRIBUIDO

2.10.1 MODALIDADES DE TELETRABAJO

La modalidad de teletrabajo hace referencia a los distintos modos de teletrabajar que se pueden presentar, dependiendo de distintos factores como los horarios, y grado de implementación de este modo de trabajo (Colombia, 2014). A continuación se presenta las modalidades de teletrabajo en función del grado de implementación de este modo de trabajo.

2.10.1.1 TELETRABAJO SUPLEMENTARIO

Esta modalidad se caracteriza por que los actores turnan su trabajo entre las oficinas centrales y otros puntos geográficos fuera de estas, y por otro lado en la versión

distribuida de scrum el escenario de equipos distribuidos parcialmente simples tiene como modalidad de trabajo la corresponde al teletrabajo suplementario, es decir que en ocasiones el equipo de desarrollo software trabaja desde lugares fuera de sus oficinas. (Colombia, 2014)

2.10.1.2 TELETRABAJO PERMANENTE

Los escenarios que se corresponden con esta modalidad de teletrabajo son: equipos distribuidos parcialmente complejos, distribuidos temporalmente simples y complejos. La característica de esta modalidad es que los teletrabajadores no pueden estar presentes en la oficina en ninguna ocasión debido a la distancia geográfica. (Colombia, 2014)

2.10.2 LA IMPLEMENTACIÓN DEL TELETRABAJO

La implementación del teletrabajo desde su propia definición es un proceso metodológico, en el que se siguen una serie de pasos ya mencionados, por otro lado desde la perspectiva de scrum en su versión distribuida corre por cuenta propia la organización haciendo más énfasis en lo que es la práctica scrum en sí. Por estas dos razones y los objetivos de este trabajo en lo que se enmarcara el trabajo es en escoger una modalidad de teletrabajo de scrum distribuido para así adecuar la herramienta colaborativa a este escenario. Sin embargo en la práctica real una equipo de desarrollo de software debería tomar enserio los pasos de implementación del teletrabajo ya mencionados. (Colombia, 2014)

2.10.3 IDENTIFICACIÓN DE LAS SOLUCIONES TECNOLÓGICAS

Las soluciones tecnológicas a implementar por el equipo de software dependen del escenario en que se realice, así en escenarios complejos que no tienen la posibilidad de reunirse de forma física en ninguna ocasión, necesitaran variedad de herramientas más complejas de las que se necesitara en otro escenario en el que el teletrabajo tiene carácter suplementario. (Colombia, 2014)

Estos son algunos ejemplos de soluciones tecnológicas requeridas por las entidades, dependiendo de sus necesidades y recursos:

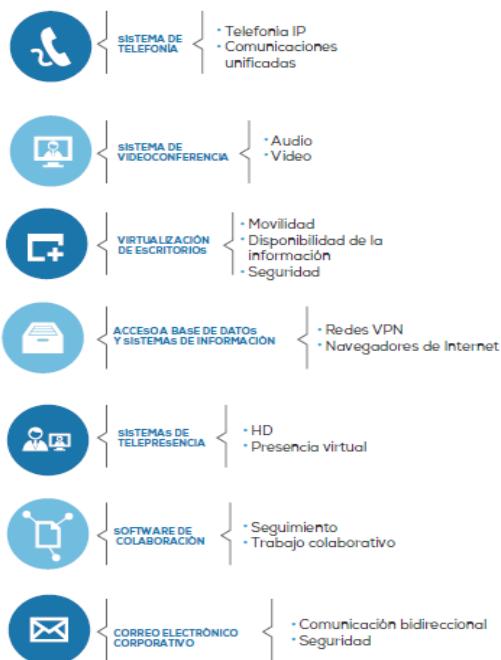


Figura 2. 26: Identificación de las necesidades tecnológicas

Fuente: (Colombia, 2014)

La clasificación de necesidades tecnológicas en la figura 2.26 sirve para contextualizar que tecnología se necesita para teletrabajar, y por tanto es una de las guías a tomar para el presente trabajo. En específico se pretende implementar una herramienta de colaboración que implemente los procesos de la metodología scrum con el objeto de teletrabajar para lo que realiza consultas a una base de datos en la nube.

2.11 CLOUD COMPUTING

El Cloud Computing, es un concepto o una forma nueva de dar servicios desde internet. En español significa computación en la nube y su función se resume como los servicios que nos da una computadora como las aplicaciones entre otras funciones, se ponen a disposición del usuario en internet, el usuario puede ser una persona particular hasta una organización puesto que esta al igual que una persona tiene necesidades, metas, planes, entre otras. Por tanto el Cloud Computing es un modo de implementación desde el punto de vista del desarrollador que conlleva sus respectivos modos de hacer las cosas como en olvidarse en los equipos físicos y concentrarse en la virtualización, en segundo lugar desde el punto de vista del usuario es una forma nueva de acceder a servicios informáticos con el único requisito de tener acceso a internet desde cualquier dispositivo que así lo permita. (Sectorial, 2010). La manera en que los desarrolladores cobren a los usuarios depende de las características del proveedor de recursos o servicios, por ejemplo

las grandes corporaciones como Google ponen a disposición software como servicio de una manera gratuita llegando a cobrar solo si un usuario requiere más prestaciones, cosa que en los casos comunes no ocurre, se puede mencionar como ejemplo de estos servicios al grupo de aplicaciones denominadas Google Apps entre las que están Gmail, Google Drive, Google Calender, entre otras que son conocidas, él como el proveedor en este caso Google solventa los gastos de infraestructura son retos que asume de manera creativa con tal de no perder a los usuarios, como el márquetin explotado en forma de anuncios. Otros pequeños emprendedores de negocios en la nube aparte de la propaganda optan por una modo de pago por parte de los usuarios, dándose así el alquiler del software en la nube, en la figura 2.27 se puede observar los distintos modos de cobro por los servicios en la nube, incluyendo el auto solventado por sus proveedores, como el caso de Google Apps.



Figura 2. 27 Formas de pago por los recursos y servicios prestados por los proveedores en la Nube

2.11.1 CARACTERÍSTICAS DE CLOUD COMPUTING

Según la definición que da el NIST¹¹ de computación en la nube, esta tiene cinco características esenciales: (Sectorial, 2010)

- Auto servicio por demanda: lo que significa que los servicios se dan bajo el pedido del cliente, este solo paga por el tiempo de uso.
- Acceso Ubicuo a la red: los servicios son accesibles desde cualquier medio con acceso a la red.

¹¹ NIST por sus siglas en inglés, que traducido significa Instituto Nacional de Estándares y tecnología de los EEUU.

- Fondo común de recursos: significa que los recursos son accesibles desde distintos puntos geográficos, proporcionando independencia de la ubicación
- Rápida Elasticidad: la cantidad de servicios ofrecidos en la nube pueden aumentar o disminuir sin problemas técnicos, gracias a la infraestructura que soporta la nube.
- Servicio medido: los servicios que son ofrecidos, son facturables.

2.11.2 CLASIFICACIÓN

La clasificación de los distintos tipos de nubes se realiza en función del grado de acceso al público, también se realiza según el modo de implementación y finalmente según el tipo de servicios que preste. (Sectorial, 2010)Según el modelo de implementación de las aplicaciones en la nube se pueden clasificar en:

2.11.2.1 SOFTWARE COMO SERVICIO

Consiste en la distribución de software en la nube como un servicio para usuarios específicos, la empresa proporciona el mantenimiento, soporte y operación que usara el cliente, durante el tiempo que el usuario contrate el servicio, así por ejemplo se tiene: Gmail, Google Docs, Redes sociales, entre otros. Para la implementación del software como servicio se necesita una infraestructura de software y plataforma de desarrollo de manera tal que el software como servicio tenga las características de distribución y acceso vía internet. (Sectorial, 2010)

En este trabajo se implementara un software como servicio dirigido a los equipos de desarrollo de software que precisen agilidad en sus proyectos y que además necesiten implementar el teletrabajo con la metodología de desarrollo scrum.

2.11.2.2 PLATAFORMA COMO SERVICIO

Se trata de ofrecer una plataforma de implementación de aplicaciones en la nube, lo que quiere decir que se provee servidores y una base de datos para poner en marcha estas aplicaciones. Algunos ejemplos son Google App Engine, Amazon, entre otros. La tabla 2.5 muestra una clasificación de proveedores según el tipo de servicio como plataforma que ofrecen.

TIPO DE SERVICIO	EJEMPLO
Plataformas de desarrollo	Amazon Simple Queue Service (Amazon SQS) (Amazon Web Services, Amazon Simple Queue Service (Amazon SQS)), Amazon Simple Storage Service (Amazon S3) (Amazon Web Services, LLC), Google App Engine (Google), GRIDS Lab Aneka (Vecchiol, Chu, & Buyya, 2009).
Bases de datos	Amazon SimpleDB (Amazon Web Services, Amazon SimpleDB), Big Table (Chang, y otros, noviembre de 2006), Microsoft SQL Azure Database (Microsoft).
Cola de mensajes	Amazon Simple Queue Service (Amazon SQS) (Amazon Web Services, Amazon Simple Queue Service (Amazon SQS)).
Servidores de aplicaciones	NetSuite Business Operating System (NS-BOS) (NetSuite, Inc.).

Tabla 2. 6: Ejemplos concretos de plataforma como servicio

Fuente: (Sectorial, 2010)

2.11.2.3 INFRAESTRUCTURA COMO SERVICIO

Se trata de la virtualización de infraestructura de computación, así por ejemplo computadoras, tarjetas procesadoras, servidores, entre otros. Como ejemplos se tiene Amazon EC2, Azure de Microsoft, entre otros. La tabla 2.7 muestra los recursos como servicio que ofrecen las infraestructuras como servicio en la nube.

TIPO DE SERVICIO	EJEMPLO
Procesamiento	Amazon Elastic Compute Cloud (Amazon EC2) (Amazon Web Services, LLC), Sun Network.com (Sun Grid) (SUN Microsystems, Inc.), ElasticHost (ElasticHosts Ltd.), Eucalyptus (Nurmi, y otros, 2009), Nimbus (Alliance), OpenNebula (Grupo de Arquitectura Distribuida), Enomaly (Enomaly, Inc.).
Distribución de contenido a través de servidores virtuales	Akamai (Technologies), Amazon CloudFront Beta (Amazon Web Services, LLC).
Almacenamiento	Amazon Simple Storage Service (Amazon S3) (Amazon Web Services, LLC), Amazon SimpleDB (Amazon Web Services, Amazon SimpleDB), Amazon Elastic Block Store (Amazon Web Services, Amazon Elastic Block Store (EBS)), Microsoft SkyDrive (Microsoft Corporation), Flickr (Flickr, LLC), Youtube (YouTube, LLC), Nirvanix Storage Delivery Network (Nirvanix), Microsoft Live Mesh Beta (Microsoft Corporation, 2009), Flickr (Flickr, LLC).
Administración de sistemas	Elastra (Elastra Corporation), Engine Yard (Engine Yard, Inc.), FlexiScalable (XCalibre Communications), Grid Layer (Layered Technologies, Inc.), Joyent (Joyent, Inc.), Mosso (Rackspace, US Inc.), Savvis Virtual Intelligent Hosting (Savvis, Inc.).
Administración de alojamiento	Digital Realty Trust (Digital Realty Trust, Inc.), GoDaddy.com (GoDaddy.com, Inc.), Layered Technology (Layered Technologies, Inc.).

Tabla 2. 7: Ejemplos concretos de infraestructura como servicio

Fuente: (Sectorial, 2010)

2.12 PLATAFORMAS DE DESARROLLO EN LA NUBE

En la actualidad las opciones de plataforma en la nube para el desarrollo y alojamiento de aplicaciones como software de servicio en la nube son diversos y escoger una plataforma se basa en la disponibilidad de recursos de la empresa, el lenguaje y tecnologías que se desea utilizar en el desarrollo. En este trabajo se hace una análisis a modo de justificación de porque se escoge la plataforma de desarrollo en la nube de Google App Engine. Se

analizara cuatro plataformas de desarrollo en la nube: App Fog, Open Shift, Zend Cloud y Google App Engine.

2.12.1 APP FOG

Es una plataforma que permite centrarse en el código, lo que quiere decir que el escalado de las aplicaciones y gestión de los servidores es automático, entre sus características es que hace un uso efectivo de la memoria cache para acceder a las aplicaciones de forma repetitiva. Tiene la capacidad de soportar muchos lenguajes de desarrollo y programas de gestión de contenidos, entre ellos: java, drupal, django, wordpress, por otro lado soporta también diversas bases de datos: mysql, mongoldb, postgresSql, entre otros. La dirección web de esta plataforma es: www.appfog.com.

La desventaja que tiene el uso de esta plataforma es que el desarrollo libre está limitado a un periodo de 30 días, por lo que es un periodo corto de análisis de cómo funciona en la nube un prototipo con las características de este trabajo como es la inexperiencia en desarrollo de aplicaciones en la nube y el deseo de actualizar varias versiones del prototipo sin importar el lapso de tiempo que lleve esto. En la figura 2.28 se observa el index del sitio web.



Figura 2. 28: Plataforma de desarrollo en la nube AppFog

Fuente: (AppFog, 2015)

2.12.2 OPEN SHIFT

Es una plataforma de desarrollo en la nube perteneciente a la empresa Red Head, y al igual que la anterior cumple con la definición de plataforma como servicio, que dice que este forma de implementación en la nube debería permitir centrarse en el código y la lógica de las aplicaciones dejando la gestión de recursos y escalado característico de las aplicaciones en la nube al proveedor de la plataforma de desarrollo. Para la implementación de aplicaciones se da tres modalidades: plataforma como servicio público, plataforma empresarial con nube privada, comunidad de código abierto de plataforma de desarrollo libre. Soporta varios lenguajes de programación, entre ellos: java, Php, Ruby, Python, y Perl. La dirección web de acceso a la información de la plataforma es: www.openshift.com

La desventaja por la que no se escoge esta plataforma es que en sus dos modos de implementación libre a saber: plataforma como servicio público, comunidad de código abierto para desarrollo en la plataforma. Es necesario desarrollar mediante un API en línea que proporciona esta plataforma de desarrollo, lo que para un trabajo de investigación como este resulta costoso dado el requisito indispensable de tener acceso continuo a internet para desarrollar y realizar las pruebas. En la figura 2.29 se observa la página principal del su sitio.

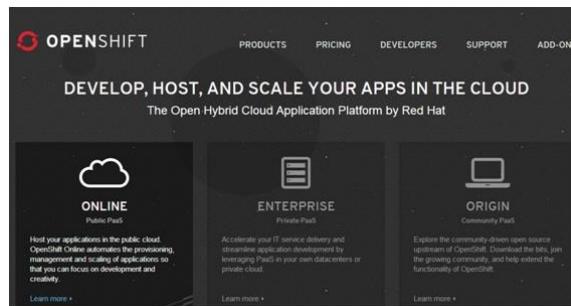


Figura 2. 29: Plataforma de desarrollo en la nube Open Shift

Fuente: (Shift, 2015)

2.12.3 ZEND CLOUD

Es una plataforma de desarrollo pensada en especial para los desarrolladores de aplicaciones web, esta plataforma tiene una visión pensada para las empresas que cuentan con su propia infraestructura física de servidores, brindando el software necesario para poner en la nube las aplicaciones de este tipo de organizaciones. La dirección web de esta plataforma es: www zend.com

Es notable la desventaja el hecho de escoger esta plataforma para el desarrollo de un trabajo de investigación como lo el presente, dada la necesidad de descargar software para un servidor físico con el que no se cuenta. En la figura 2.30 se observa la página principal de Zend Cloud.



Figura 2. 30: Plataforma de desarrollo en la nube Zend Cloud

Fuente: (Cloud, 2015)

A continuación se presenta las características de GAE, que lo hacen ideal para la implementación de prototipos académicos como lo es el presente.

2.13 GOOGLE APP ENGINE (GAE)

En este apartado se expone la plataforma de desarrollo GAE para aplicaciones web, haciendo énfasis en él porque se ha escogido como plataforma para el proyecto.

2.13.1 SOBRE LA PLATAFORMA

GAE es una plataforma de desarrollo en la nube perteneciente a la corporación GOOGLE, trata de un SDK¹² para desarrollo en la nube, disponible para IDs comerciales como Netbeans, y Eclipse además de proporcionar IDs para Python, y Go. En este trabajo de investigación se hará uso de plugin para java en Netbeans. La figura 2.31 es un resumen de las tecnologías y herramientas para la implementación del trabajo.



Figura 2. 31: Tecnologías a utilizar en el desarrollo

La plataforma de Google App Engine hereda muchas de las características y ventajas de las tecnologías propietarias de Google. Una de las características es el escalado automático lo que significa que se agrega más servidores para la ejecución de la aplicación según la demanda de los usuarios de forma automática o el aumento de recursos de hardware como CPU, memoria entre otros. Otras ventajas aparte del escalado automático son: tolerancia a fallos, replicación o redundancia de datos. En concreto para los desarrolladores de java soporta el JDK¹³ cinco y seis, el servidor java soportado es Jetty que dicho sea de paso contiene al contenedor de servlets que son la clase que interactúa entre servidor y cliente mediante el protocolo HTTP para el intercambio de hipertexto que es el que usa la red internet en cada transacción.

2.13.2 CONTENIDO DEL SDK

Para desarrollar una aplicación a alojarse en GAE, se debe tener en cuenta que se realiza fuera de línea es decir no es necesario estar conectado a la nube GAE, para este cometido

¹² SDK por sus siglas en inglés que significa kit de desarrollo de software.

¹³ JDK por sus siglas en inglés que significa kit de desarrollo java.

GAE proporciona el SDK, el contenido del SDK varía según el lenguaje escogido de programación escogido para desarrollar, así por ejemplo el SDK de Python contiene un IDE de desarrollo. En lo que respecta a este trabajo el SDK es para el lenguaje Java, este SDK contiene un servidor de desarrollo en local, los paquetes Jars necesarios para trabajar con java lo que significa que el usuario solo debe concentrarse en desarrollar la aplicación. La simulación en un entorno local tiene ventajas como desventajas, así entre las ventajas se tiene que actúa de manera similar al entorno real de ejecución en la nube lo que es importante a la hora de pensar como estará conformada la aplicación entre estas similitudes son los servicios que brinda el entorno real de alojamiento como lo son: simulación de acceso a la aplicación a través de las cuentas de google, acceso a un almacén de datos mediante una interfaz web independiente de la aplicación con el fin de editar los datos almacenados. Entre las desventajas de la simulación en local se tiene que no se respeta la lista blanca de clases soportadas en el entorno de ejecución real sin embargo en el sitio oficial de GAE en la sección de java se proporciona esta lista para que la aplicación subida a la nube funcione de manera correcta. En este trabajo se toma en cuenta la lista blanca de clases permitidas en un entorno de ejecución java. (java, 2012)

2.13.3 COMPILACIÓN Y DESPLIEGUE

Para poder compilar y desplegar las aplicaciones web en GAE, se hace necesario un simple cuenta de google ya sea de Gmail o de cualquier Google App, con este requisito uno se dirige al sitio oficial de GAE, en donde se manda un SMS de conformación a su número de celular, al igual que cuando se olvida la contraseña de cuenta Gmail. Luego de registrarse como usuario de GAE lo que queda es la compilación y despliegue de su aplicación para lo que se pide realizar una reserva de ID de aplicación, el IDE de una aplicación en GAE es de vital importancia ya que es a través de este ID de aplicación que se crea una dirección URL de acceso a la aplicación para los usuarios de la misma, sin embargo el proceso de reserva de ID es un poco tedioso ya que antes de aceptar un ID el motor de GAE comprueba que esté disponible, ya que GAE nunca borra los ID de las aplicaciones aunque hayan sido eliminadas. Las cuentas libres tienen el límite de subida de aplicaciones a diez como máximo. La forma que toma la dirección URL de las aplicaciones subidas a GAE es: <http://<AppID>.appspot.com> (java, 2012)

2.13.4 ALMACENAJE DE DATOS

Para hablar del almacén de datos de GAE se debe entender que es Bigtable. Bigtable es un almacén de datos desarrollado y probado por GOOGLE, se caracteriza por ser no-

relacional, su capacidad de almacenaje es de proporciones de Peta bytes, las aplicaciones estrellas de google los utilizan, así como el buscador de google, la aplicación google heart entre otros usos de nivel de investigación importante de la corporación. En GAE para acceder al Bigtable de una forma amigable a los desarrolladores se desarrolló el DataStore, esta abstracción tiene similitud pequeña con las bases de datos relacionales, y alta con las orientadas a objetos, lo que se puede entender en la tabla 2.8 de términos parecidos con una base de datos relacional. (java, 2012)

Datastore Vocabulary Map	
Relational DataBase	Google DataStore Equivalent
Database	Datastore
Table	Kind
Row	Entity
Row ID	Key
Column	Property

Tabla 2. 8: Similitudes entre una base de datos relacional con el modelo de base de datos de GAE el Datastore

En el SDK para java, el API del datastore proporciona una abstracción mediante la clase Entity la cual en palabras simples permite la creación de datos persistentes es decir objetos, además de su manipulación básica como lo son las altas, bajas y modificaciones tradicionales, esto es algo particular en el desarrollo de java en la web ya que de forma habitual la forma de trabajar de java en la web es a través mapeados de las bases de datos relacionales mediante componentes como: JDBC¹⁴, ORM¹⁵, JPA¹⁶, JDO¹⁷ dado el uso proliferado de las bases de datos relacionales. Sin embargo dada la necesidad de manejo de datos de forma detallada, GAE soporta el uso de JPA Y JDO, lo que es de importancia para aquellos desarrolladores que están familiarizados con el manejo de Beans java que no son más que clases que representan a los datos en un almacén de datos. Por otro lado el lenguaje de consulta a la base de datos varía según el componente de interacción con la base de datos, así por ejemplo con JDBC se realiza las consultas a la base de datos de manera directa con el lenguaje de consulta de bases de datos relacionales SQL, en GAE al soportar a un base de datos no relacional el lenguaje especial de consulta es GQL que maneja de forma amigable a los índices que tiene bigtable. (java, 2012).

¹⁴ JDBC por sus siglas en inglés que significa conexión a bases de datos con java.

¹⁵ ORM por sus siglas en inglés que significa mapeo de objetos relacionales.

¹⁶ JPA por sus siglas en inglés que significa Api de persistencia java.

¹⁷ JDO por sus siglas en inglés que significa Objetos de datos java.

2.13.5 CAPACIDAD DE ALMACENAMIENTO

El datastore destinado al almacenamiento de datos de las aplicaciones web tiene una capacidad de almacenamiento en función del plan con que un desarrollador trabaja. Como se puede apreciar la tabla 2.8 tomada del sitio oficial de GAE, la facturación para el datastore es de acuerdo al uso que se hará de los servicios de la plataforma en este trabajo se destaca por ejemplo que el datastore o almacén de datos tiene un uso gratuito limitado por día de manera gratuita de 1 GB de almacenamiento lo que es suficiente para probar el sistema de teletrabajo. En la tabla 2.9 se observa los detalles de facturación para el plan gratuito y los de pago tomado de la página oficial de GAE.

Servicios		
Todas las aplicaciones son gratis dentro de una cuota de uso que se reinicia diariamente. Aplicaciones de pago pueden utilizar más en los precios por debajo.		
Servicio	Cuota gratuito por aplicación por día	Precio Si usted excede su cuota gratuita
Instancias	28 horas de instancia	\$ 0.05 / instance / hora
Nube de almacén de datos (base de datos NoSQL)	50k de lectura / escritura / pequeña de almacenamiento de 1 GB	\$ 0.06 / 100k leer ni escribir ops operaciones pequeñas gratis * \$ 0.18 / GB / mes
Saliente del tráfico de red	1 GB	\$ 0.12 / GB
El tráfico de red entrante	1 GB	Libre
Cloud Storage	5 GB	\$ 0.026 / GB / mes
Memcache	Uso gratuito de la piscina compartida contingente n libre para piscina Dedicado	Uso gratuito de la piscina compartida piscina Dedicado: \$ 0.06 / GB / hora
Búsqueda	1.000 operaciones básicas 0,01 documentos de indexación GB de almacenamiento de documentos 0.25 GB 100 búsquedas	\$ 0,50 / 10k búsquedas \$ 2.00 / documentos de indexación GB \$ 0.18 / GB / mes de almacenamiento
Email API	100 beneficiarios	Contacto de Ventas
Registros API	100 MB	\$ 0.12 por GB
Tarea Cola y Registros de Almacenamiento	5 GB 1 GB	\$ 0.026 / GB / mes
SSL IPs virtuales	Sin cuotas	\$ 39 / IP virtual / mes
Servicios incluidos		
Cron, Archivos API, Manipulación de imágenes, PageSpeed, SNI certificados SSL, Socket API, API cola de tareas, URLFETCH, API Usuarios		

Tabla 2. 9: Capacidades de los servicios que ofrece la plataforma de desarrollo en la nube Google App Engine

2.13.6 RECURSOS ADICIONALES DE LA PLATAFORMA

Entre los recursos adicionales de la plataforma GAE, se puede reportar Bugs, acceso a un canal IRC de chat de forma gratuita donde los desarrolladores hablan de sus experiencias de desarrollo en la plataforma, para aplicaciones desarrolladas en java, se cuenta con una lista de clases soportadas por la plataforma denominada la lista blanca. A continuación se lista las direcciones para estos recursos:

- Canal IRC: <http://webchat.freenode.net/?channels=appengine>

- Reporte de errores: <http://code.google.com/p/googleappengine/issues/list>
- Lista Blanca: <http://code.google.com/appengine/docs/java/jrewhitelist.html>

2.14 APLICACIONES JAVA WEB

La plataforma java de GAE, utiliza la J2EE que es parte de la plataforma de java para la creación de aplicaciones web, esta permite crear aplicaciones web portables entre plataformas, escalables, y de fácil distribución, destaca por su seguridad ya que evita accesos no autorizados a recursos tanto del servidor como de la PC del usuario que utiliza la aplicación. En este trabajo se ha de usar este entorno debido a que existen frameworks y componentes compatibles con J2EE, que permiten simplificar tareas. (Sierra, 2010).

2.14.1 SERVLETS

Los servlets son las interfaces de comunicación entre el usuario y el servidor, haciendo una analogía un servlet es lo que es una archivo php, este se comunica con el servidor para generar contenido dinámico alimentándose de alguna base de datos. En lenguaje técnico para la tecnología java se le describe como un “componente” que hace posible la comunicación mediante el protocolo http en el ambiente de las aplicaciones, en la implementación se traduce a una clase que tiene propiedades especiales de las clases a diferencia de las clases usuales con las que se aprende el lenguaje. En la actualidad este componente se usa aunque enmascarado con una serie de frameworks, con distintos propósitos, así por ejemplo el framework GWT¹⁸ que facilita el manejo de tecnología Ajax. En este trabajo se hace énfasis en el estándar de tecnologías web para java es decir no se utiliza frameworks, esto debido a que la plataforma de desarrollo Google App Engine lo tiene como requisito.

2.14.2 PAGINAS JSP¹⁹

Las páginas JSP son parte del estándar de desarrollo de aplicaciones web para java, se trata de páginas dinámicas basadas en HTML, se les denomina dinámicas porque permiten generar contenido dinámico gracias a que permiten embeder código java o también a su soporte de etiquetas especiales que interactúan con el servidor de desarrollo. Como toda tecnología web en la actualidad también permite trabajar con los estándares de diseño como hojas de estilo, java script, librerías jquery, html5, todo en el marco de ver las restricciones de los distintos escenarios así por ejemplo no tendrá sentido diseñar

¹⁸ GWT por sus siglas en inglés que significa Google Web Toolkit

¹⁹ JSP por sus siglas en inglés que significan Páginas de Servidor Java

funciones de validación en local al a ver ya librerías de uso múltiple, el cuidado a tener es ver las compatibilidades.

2.15 IDE DE DESARROLLO NETBEANS

El entorno de desarrollo a utilizar es Netbeans en su versión 8, que a la fecha actual es la reciente versión de este entorno, el motivo de su elección es la familiaridad con su espacio de trabajo y la forma de organización de los archivos, sin embargo no es la única opción para la que da soporte la nube Google App Engine. En este trabajo los pasos que se ha realizado para poder tener IDE listo para subir aplicaciones a la nube son:

- Descarga del IDE de desarrollo Netbeans versión 8.0, en su versión JavaEE esto último es importante ya que permitirá el desarrollo de aplicaciones web.
- El kit de desarrollo para java “JDK” 7u10, esto porque es el que soporta la plataforma de desarrollo GAE, sin embargo a la fecha Oracle ya ha lanzado la versión 8.0, es por eso que cualquiera que vaya a trabajar con java y aplicaciones web, debe echar buen ojo a la cuestión de compatibilidades para no desperdiciar tiempo en bajar cosas que no le vayan a servir.
- Descarga del complemento de GAE para Netbeans que incluye el servidor en local de la dirección: <https://code.google.com/p/nb-gae/>

2.16 HERRAMIENTA UTILIZADA PARA MODELADO DE FLUJOS CON BPMN

Para el modelado de los procesos y flujos del workflow colaborativo para scrum, se hace uso de la herramienta Microsoft Visio 2013, esta herramienta se caracteriza por ser un medio de diagramación que incluye diversas plantillas desde arquitectura hasta el ramo de la informática. En particular el modelado de procesos interesa más a los administradores que son los que analizan los flujos en una organización con la intención de aplicar sus conocimientos administrativos, sin embargo a la hora del desarrollo de nuevas soluciones tecnológicas como lo son los sistemas BPM o los workflow es una buena práctica el modelado de los procesos y flujos con redes de Petri, y otros modelos estándar en los que se encuentra BPMN. (Jimenez, 2008). La figura 2.32 muestra como el programa Visio de Microsoft, en su versión 2013 incorpora las plantillas de modelado para el estándar BPMN.

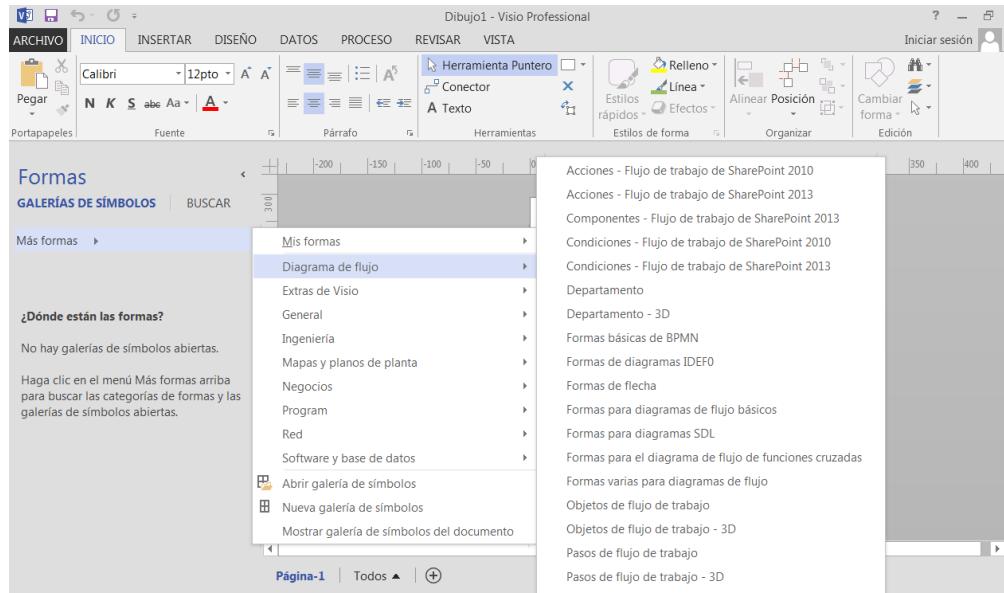


Figura 2. 32: Modelado de procesos con BPMN en la herramienta Microsoft Visio

2.17 OTRAS HERRAMIENTAS UTILIZADAS

Entre otras herramientas que se han utilizado para poder realizar el presente trabajo, se hizo necesario de un traductor, dado que el libro base que se utiliza para el estudio de la plataforma GAE es el oficial: “*Programming Google App Engine*” de Dan Sanderson de la editorial O’REILLY, esta editorial se caracteriza por sacar libros para informática y ramas afines en su versión ultima, con el inconveniente de estar disponibles solo en inglés. Los costos de internet para acceder de manera permanente al traductor de google se hicieron altos, por lo que en un momento se determinó el uso de traductores en escritorio, el programa utilizado fue “LEC Power Translator” que brinda varias fuentes de origen y destino en cuanto a lenguajes. La figura 2.33 muestra el funcionamiento del traductor de escritorio “LEC Power Translator”.

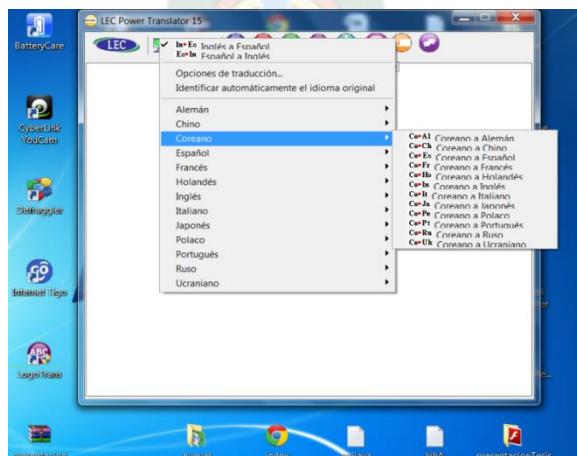


Figura 2. 33: Traductor de escritorio LEC Power Translator

CAPITULO 3

MARCO APlicativo

3.1 INTRODUCCIÓN

El objetivo de este capítulo es el de formalizar el análisis y diseño del sistema “Workflow Colaborativo Basado en Scrum Distribuido para Teletrabajo Implementado en Cloud Computing” utilizando para este fin la metodología de desarrollo AUP, que como se mencionó en el capítulo anterior está basado en casos de uso para recoger los requisitos. En este trabajo se tiene la necesidad del modelado de los procesos del negocio de un workflow por lo que también se utiliza el lenguaje de especificación BPMN. Al ser AUP una metodología que se centra en la arquitectura, se modelara estos aspectos con la ayuda de los modelos de UML que soporta de manera completa el modelado de aplicaciones web, como lo es el sistema de este trabajo.

- Modelado de los procesos del workflow: BPMN
- Ciclo de vida del desarrollo del software: según la metodología ágil AUP.

Según la metodología de desarrollo AUP, el ciclo de vida del desarrollo consta de cuatro fases: inicio, elaboración, construcción y transición. Las tareas a realizar según las fases de AUP en este trabajo y bajo las definiciones de contenido de la misma en el capítulo 2, son:

- Fase de inicio: escenario de scrum distribuido a automatizar, elaboración del modelado de procesos para scrum con BPMN, recopilación de las funcionalidades o requisitos del sistema mediante el modelo de casos de uso a nivel de sistema, delimitación del dominio mediante el diseño conceptual.
- Fase de elaboración: Especificación de requisitos del sistema mediante casos de uso a nivel de funcionalidades, y por ultimo elaboración del diseño navegacional.
- Fase de construcción: implementación de la arquitectura del workflow con las funcionalidades definidas, en iteraciones que entregaran distintas versiones del sistema.
- Fase de Transición: Esta fase es de entrega del producto a los usuarios, por lo que su desarrollo se deja para el capítulo 4, donde se realiza un análisis de lo logrado en un marco demostrativo para la hipótesis.

3.2 FASE DE INICIO

3.2.1 DEFINICIÓN DEL ALCANCE DEL PROYECTO

En el capítulo 2 del presente trabajo se expuso los escenarios que considera la metodología de desarrollo scrum en su versión distribuida, y la conclusión a la que se llega es que la viabilidad de implementación de una herramienta colaborativa para teletrabajo depende del tiempo, y el nivel de conocimiento, es por esto que a se explica a continuación el principal motivo por el que no se modelan los escenarios que requieren un teletrabajo fuera de la oficina de forma permanente, para la herramienta colaborativa a implementar en este trabajo:

- La dificultad del diseño de herramientas de teletrabajo fuera de la oficina de forma permanente, está en que se tendría que considerar un método que permita compartir el código y probarlo en línea, esto no es imposible ya que se podría virtualizar un servidor en la nube e implementar software base que permita la funcionalidad mencionada, esto demandaría un tiempo de investigación más extenso por lo que se lo deja como un tema de investigación para futuros trabajos.

Dada esta justificación, el escenario que se modela y automatiza de la metodología scrum es la del trabajo suplementario, en el que se supone que al menos son dos veces a la semana en la que los miembros del equipo de desarrollo de software se reúnen de forma física para coordinar sus actividades, esto es necesario puesto que pone una solución al problema de unir el software y mostrar los incrementos al dueño del producto

3.2.2 DEFINICIÓN DE RIESGOS MEDIANTE EL MODELADO DE PROCESOS DEL SISTEMA CON BPMN

El sistema que se implementa en este trabajo, tiene la naturaleza de un workflow colaborativo y por motivos que se expuso en el capítulo 2 es el que se escoge para el modelado. Se ha hecho recuerdo al lector que este modelado de procesos es equivalente a los diagramas de actividades del estándar UML, que sin embargo no son los recomendados para el diseño de workflow debido a la complejidad para lectores no informáticos.

3.2.2.1 MAPA DE PROCESOS GENERALES PARA EL SISTEMA

La entrada al sistema tiene la misma idea de identificación que las aplicaciones de google, es decir que con una cuenta google se tiene acceso a la aplicación y sus funcionalidades, de esta manera este modelado abarca desde cuando un usuario entra a la

herramienta hasta que sale, visto de una perspectiva general, en esta etapa es donde se define el rol con el que se ingresa a un proyecto. El administrador por defecto de la instancia de un proyecto es el usuario al que se le asignara el rol de scrum máster en el proyecto, debido a su papel de experto en reglas de desarrollo de proyectos bajo la metodología scrum, a nivel de sistema este estará encargado de invitar mediante un correo electrónico a los miembros del proyecto como el propietario del producto y el equipo de desarrollo, los que deberán ingresar a la aplicación la que identificara en los proyectos en que trabaja mediante su correo electrónico. El único requisito obligatorio para cualquier usuario será tener un correo en gmail, este servirá de entrada al sistema. El modelado de este proceso de alto nivel con el estándar BPMN, se observa en la figura 3.1

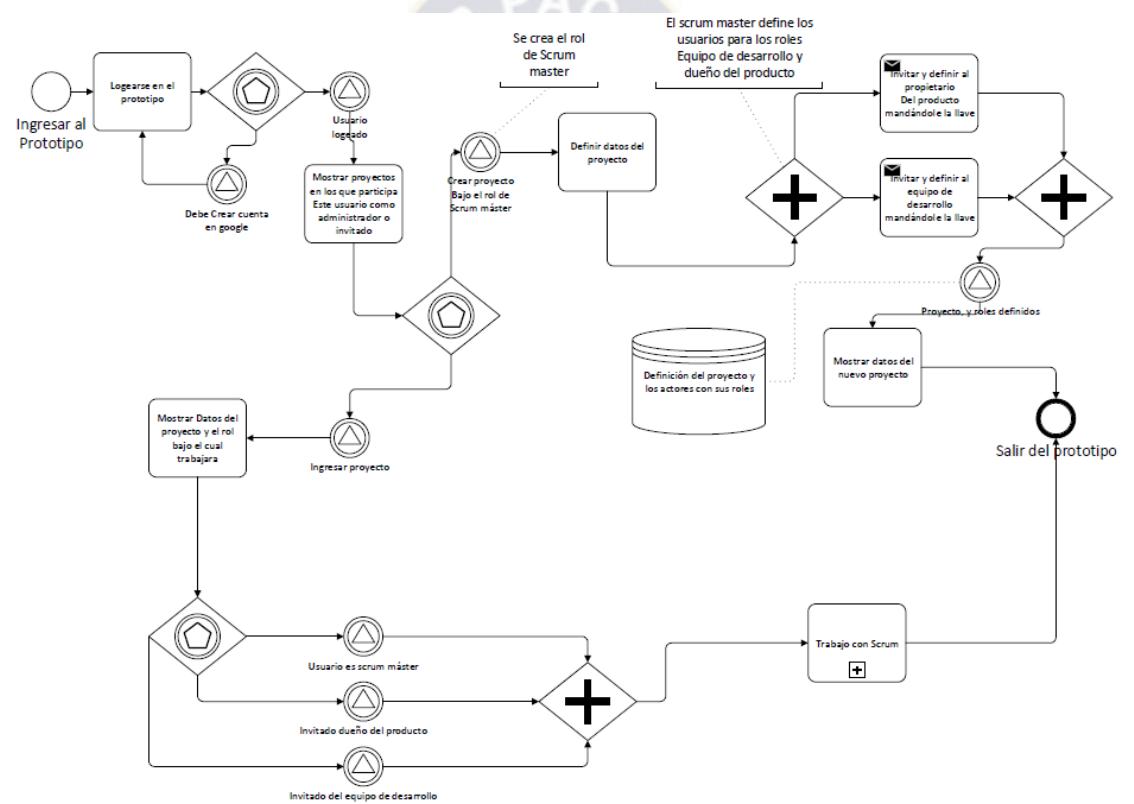


Figura 3. 1: Vista de procesos con BPMN de alto nivel

3.2.2.2 SUB PROCESO DE TRABAJO BAJO SCRUM

El mapa de sub proceso de trabajo bajo scrum es una vista de las sub procesos que comprenden el desarrollo de software bajo esta metodología, además de ser la guía del flujo de trabajo que deberá llevar a cabo la herramienta colaborativa de este trabajo. Debe tenerse en cuenta que este es un subproceso que tiene como entrada clara y definida el rol del usuario, como se aprecia el subproceso de trabajo scrum cuenta con subprocesos los cuales son realizados según el rol del usuario, cuando se pase detallar a estos subprocesos con su debido mapa de proceso como define el estándar BPMN se pondrá de manera

explícita que roles son los que están involucrados, sin embargo se deja anotaciones en el subproceso scrum respecto a los roles. Además note que la fase de implementación, pruebas y armado de las distintas partes asignadas a los miembros a nivel de código se describe como un proceso manual debido a que el alcance del trabajo es un escenario parcialmente distribuido y este proceso se lo realiza en un encuentro personal entre los miembros del equipo de desarrollo. La figura 3.2 muestra el flujo de información y muestra los procesos de la metodología scrum propuestos en el presente trabajo.

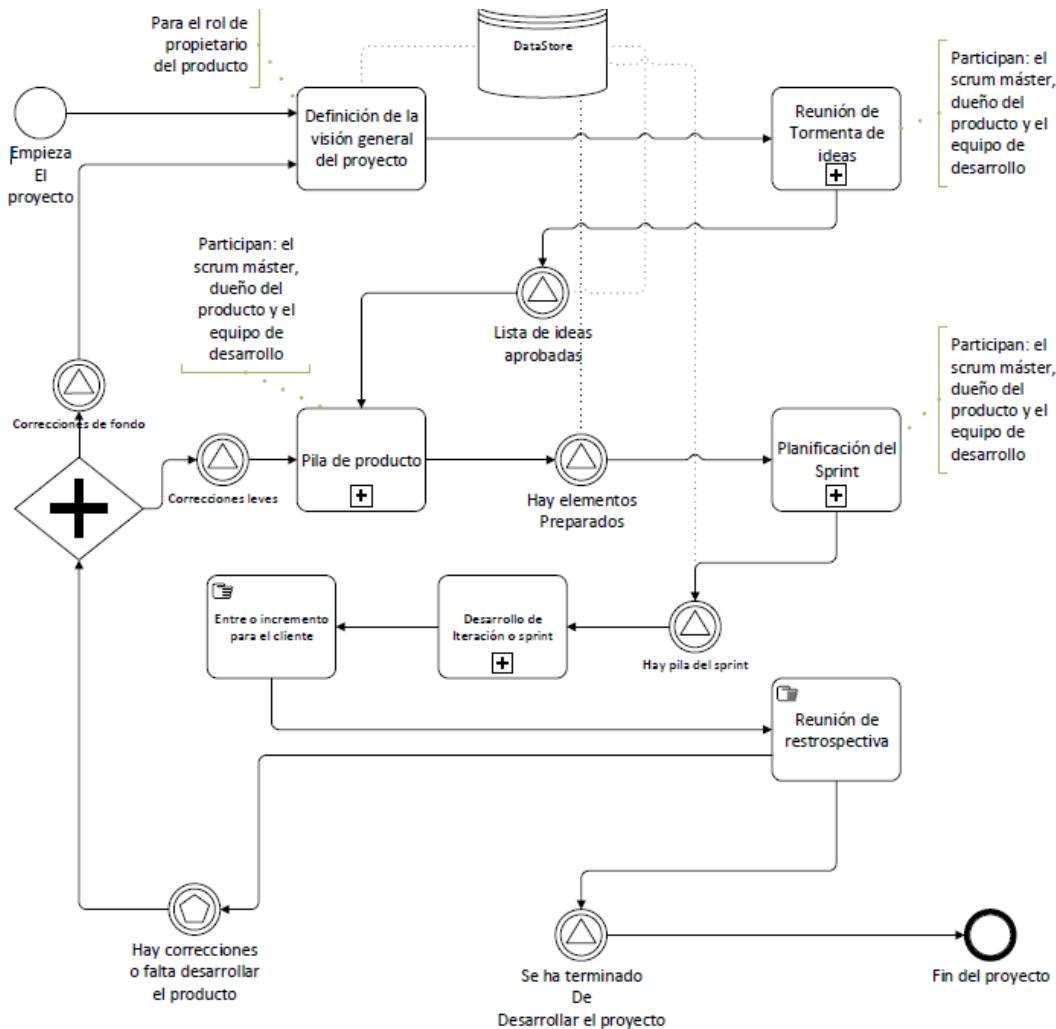


Figura 3. 2: Vista a nivel del subproceso de teletrabajo bajo Scrum en el sistema

3.2.2.3 DESCRIPCIÓN DE LOS SUB PROCESOS DE SCRUM Y TAREAS

Como se puede apreciar en el mapa de procesos de scrum hay una tarea simple y cinco sub procesos que tienen más detalles a nivel de proceso interno, a continuación una descripción de estos elementos.

a) DEFINICIÓN DE LA VISIÓN GENERAL DEL PROYECTO

La definición de la visión general del proyecto es una tarea simple y está a cargo del dueño del producto. Ver figura 3.3

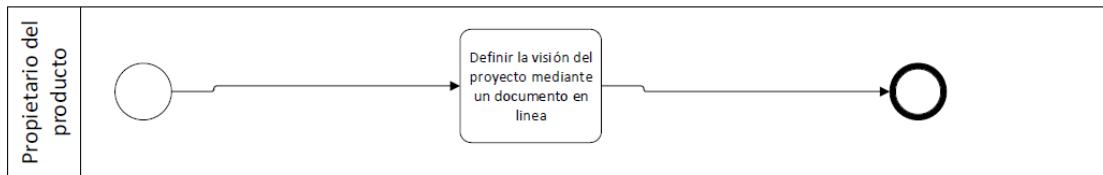


Figura 3. 3: Subproceso de definición de la visión del producto, por parte del rol de propietario del producto

b) REUNIÓN DE TORMENTA DE IDEAS

La reunión de tormentas de ideas es un sub proceso que comprende la intervención del equipo de desarrollo, el scrum master, y el dueño del producto. Este proceso empieza cuando el scrum master en su papel de coordinador del proyecto fija una fecha en la que se ha de realizar la tormenta de ideas, se asume la disponibilidad de los miembros del equipo puesto que se está en un entorno de laboral donde están dedicando todo su esfuerzo. Una vez llega la fecha es el dueño del producto quien debe crear una sala de chat que está incorporada en el sistema, es así que luego de esto los miembros del equipo y el mismo scrum master ya pueden participar en la tormenta de ideas. El dueño del producto es el encargado de registrar las ideas que cree que deben ser parte del producto a desarrollar, todo esto en el marco de trabajo que siguiere la metodología Scrum. En este trabajo a razones de flexibilidad para el usuario se permite que el rol de dueño producto defina de una vez sin necesidad de la reunión las funcionalidades, sin embargo se incorpora un chat de uso libre en el que un usuario determinado crea una sala de conversación y los demás acceden con solo saber el nombre de la sala. Ver figura 3.4

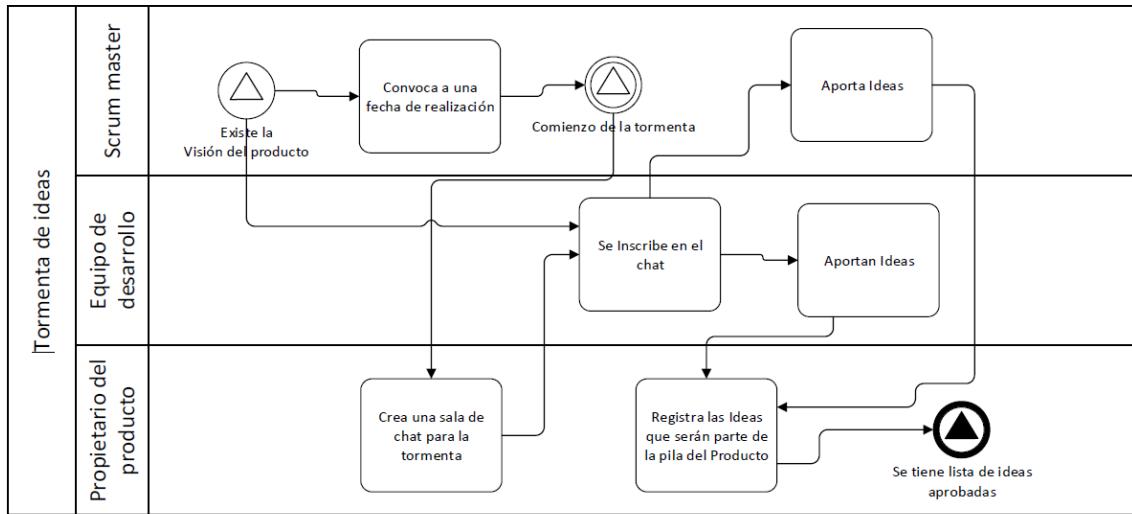


Figura 3. 4: Subproceso de la reunión de tormenta ideas, donde participan los roles de propietario del producto, el equipo de desarrollo y el Scrum master

c) PILA DEL PRODUCTO

Una vez que se han definido las funcionalidades que se incorporaran a la pila del producto en la reunión de tormenta de ideas, se procede a preparar las funcionalidades es decir que se deben: DETALLAR, PRIORIZAR, y ESTIMAR, para luego pasar al siguiente sub proceso de planificación. El proceso empieza con la convocatoria a los invitados del proyecto, por parte del scrum master, mediante la definición de la fecha y hora en que se realizará esta actividad, lo que se realiza mediante un email enviado a los miembros, se supone la disponibilidad de los miembros, por estar teletrabajando lo que significa la disponibilidad de su tiempo.

Ver figura 3.5

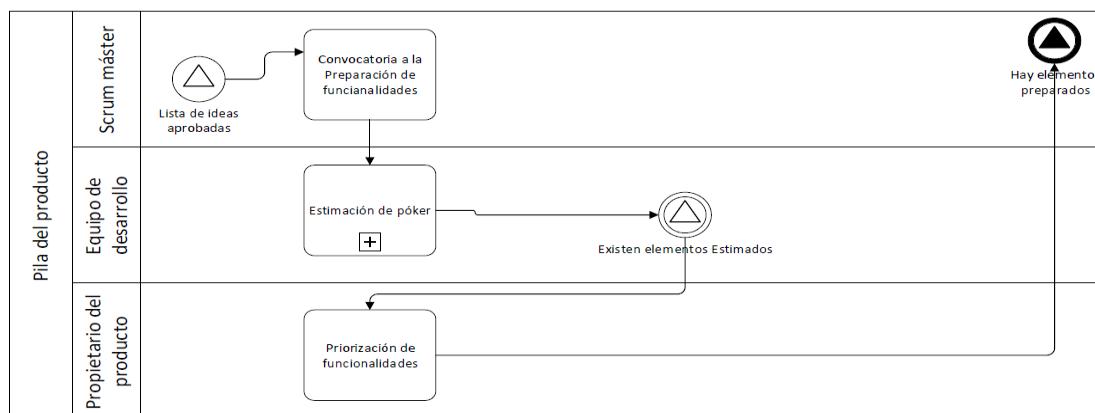


Figura 3. 5: Subproceso para la elaboración de la pila del producto, donde participan los roles de propietario del producto, equipo de desarrollo y el scrum master

i) PROCESO DE ESTIMACIÓN DE PÓKER

El proceso de estimación mediante el juego de póker, se realizara funcionalidad a funcionalidad, la secuencia es: el propietario del producto expone una explicación por funcionalidad, detallando que es lo que quiere, luego el equipo de desarrollo encargado de la estimación realiza preguntas al respecto a modo de retroalimentación, y por último el equipo realiza la estimación de forma individual es decir sin que vean los demás, una estimación personal de lo que duraría la funcionalidad mediante el registro de una de las cartas de la baraja estándar (ver el capítulo 2), un proceso interno de la aplicación realiza un cálculo de la media de las estimaciones para marcar a la funcionalidad como estimadas o a descomponer en más funcionalidades en función del resultado obtenido, esto se repite por cada una de las funcionalidades. Ver la figura 3.6

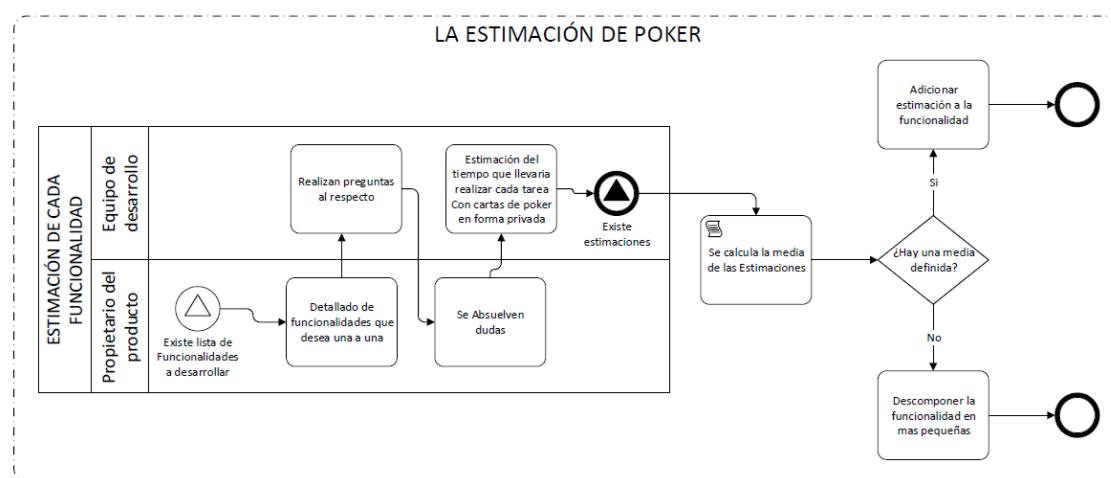


Figura 3. 6: Subproceso de estimación de la pila del producto por miembros del equipo de desarrollo, mediante el juego de estimación de póker

d) PLANIFICACIÓN DEL SPRINT

La actividad de planificar un sprint de scrum, es una de las más complejas y largas debido a que el centro del proyecto es el desarrollo de software el cual se da por cada sprint que se desarrolla. Es por esto que la planificación tiene hasta tres niveles de procesos. Esta actividad tiene como inicio la detección de un evento de señal que indique que existen funcionalidades preparadas, que son el resultado del proceso anterior. En este trabajo este proceso se lo puede realizar de manera directa sin la planificación o planificando mediante una interacción de chat. Para más detalle de este proceso vea la figura 3.7

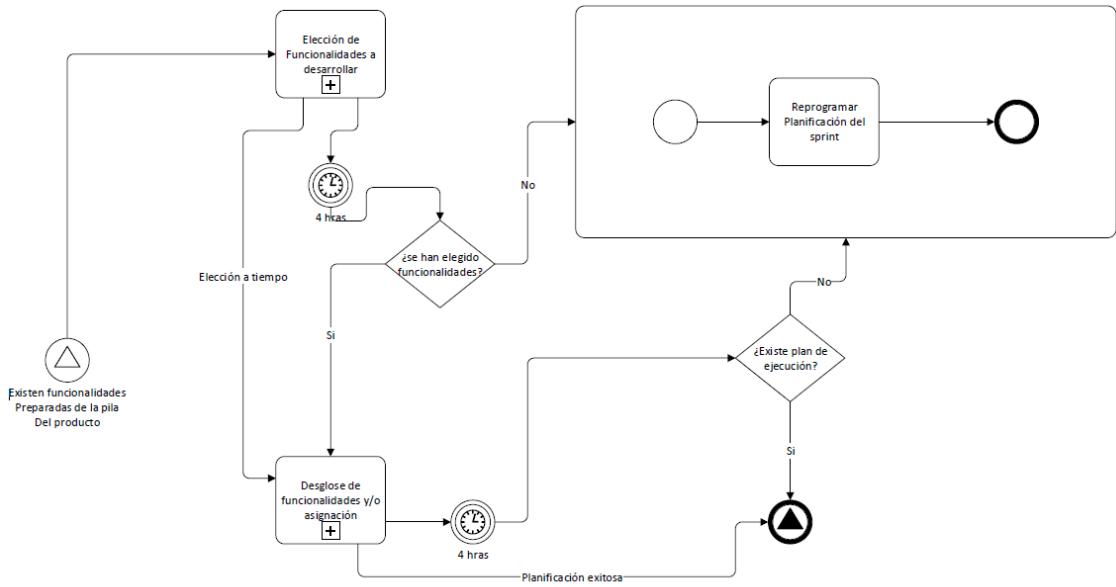
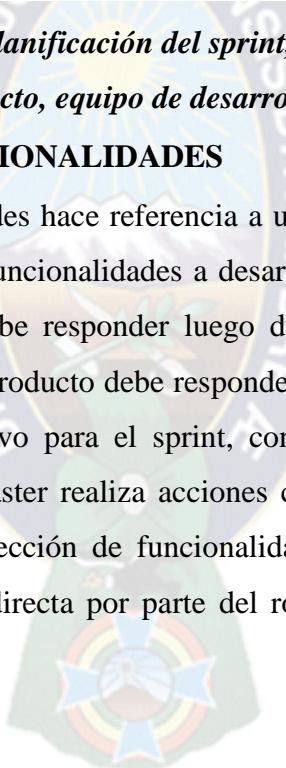


Figura 3. 7: Subproceso de planificación del sprint, donde participan los roles de propietario del producto, equipo de desarrollo, y el scrum master

i) ELECCIÓN DE LAS FUNCIONALIDADES

La elección de las funcionalidades hace referencia a un proceso de exposición por parte del dueño del producto de las funcionalidades a desarrollar durante el próximo sprint al equipo de desarrollo el que debe responder luego de la exposición si tiene dudas al respecto, si es así el dueño del producto debe responder y si no hay más dudas, el equipo de desarrollo redacta un objetivo para el sprint, con lo que se acaba esta etapa. La constante es como el scrum máster realiza acciones correctivas a la comunicación. En este trabajo este proceso de elección de funcionalidades se puede hacer siguiendo el formato descrito o de manera directa por parte del rol del dueño del producto. Ver la figura 3.8



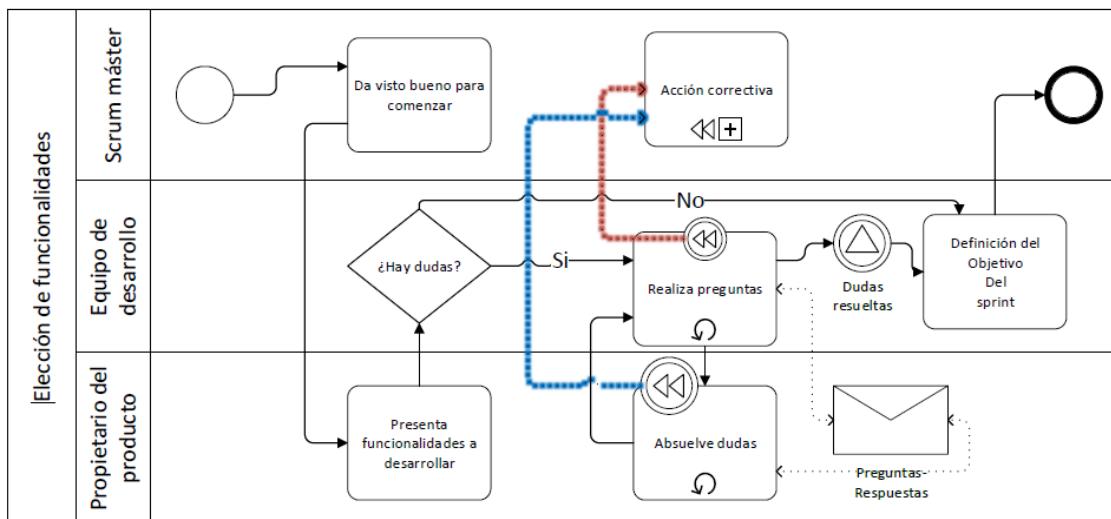


Figura 3. 8: Subproceso de elección de funcionalidades por el propietario del producto, con la participación del scrum master y el equipo de desarrollo

El proceso de elección de funcionalidades, implica la asignación del responsable y la formación de la pila del sprint correspondiente a una iteración.

3.2.3 DETERMINACIÓN DE FACTIBILIDAD DEL PROYECTO MEDIANTE LOS REQUISITOS DEL SISTEMA A ALTO NIVEL

Los requisitos del sistema bajo la metodología AUP se deben realizar dirigidos por los casos de uso del lenguaje de modelado UML, en este trabajo se ha iniciado el modelado de procesos del workflow mediante el estándar BPMN, que es su definición aclara que BPMN solo es un lenguaje de especificación y modelado de procesos de un negocio con vista a su automatización, dejando otras necesidades de modelado como las bases de datos, especificación de requisitos, entre otros. El modelado de estas necesidades se puede llevar a cabo con los modelos del lenguaje de especificación UML que para el diseño de base de datos se apoya en un diseño de diagrama de clases o diagrama físico y los requisitos del sistema con el modelo de casos de uso.

3.2.3.1 IDENTIFICACIÓN DE LOS ACTORES

La característica de este trabajo es que desde un inicio ya se tiene identificados a los actores del sistema y sus funciones, sin embargo antes de pasar a modelar un sistema de herramienta colaborativa para estos, hay que hacer algunos ajustes en sus roles para definir como accederán a la herramienta, además la descripción de quiénes son y sus tareas en el sistema es un requisito de modelado de AUP, por lo que se detalla a continuación los roles de scrum en un ambiente de trabajo a distancia.

a) SCRUM MASTER

El scrum master será el encargado de crear una instancia de proyecto de trabajo bajo la metodología scrum, en el sistema se le da el papel de administrador de proyectos, por lo que es el encargado de decidir y definir quiénes serán los invitados para trabajar con la herramienta colaborativa para una instancia de proyecto. Sus funciones al interior de un proyecto cuando ya se empieza a trabajar colaborativamente se resumen a las responsabilidades de un scrum manager en scrum, esto se detalló en los procesos del negocio modelados con BPMN en el apartado anterior, y se detallaran en forma de casos de uso en la descripción a nivel de funcionalidades.

b) DUEÑO DEL PRODUCTO

El dueño del producto como actor de la herramienta colaborativa es quien es invitado por el administrador del sistema es decir el scrum máster, el dueño del producto tiene la responsabilidad de aceptar la invitación, con lo que se hace miembro de una instancia de proyecto, a partir de aquí sus funciones quedan en el marco de trabajo bajo las prácticas de scrum a través de la herramienta colaborativa, estos detalles se los dará en el nivel de descripción de funcionalidades.

c) EQUIPO DE DESARROLLO

Los miembros del equipo de desarrollo son usuarios invitados por administrador, estos al igual que el dueño del producto deben aceptar la invitación para poder ser miembros del proyecto, y sus funcionalidades con la herramienta bajo las reglas de scrum quedan pendientes para el siguiente nivel de especificación de requisitos.

3.2.3.2 IDENTIFICACIÓN DE LOS PROCESOS

Como segundo paso se realiza la recolección de requisitos mediante los casos de uso de la notación estándar UML, se lo realiza como una definición de los procesos especificados en modelado de procesos con BPMN del workflow con el objeto de comprender las relaciones de los procesos con los actores, en primer lugar se da una definición global de los procesos que llevara a cabo el sistema, también denominado casos de uso a nivel sistema.

a) INGRESAR AL SISTEMA

Para ingresar al sistema para teletrabajar bajo la metodología scrum, se necesita del ingreso o logeo tradicional, esto en casos comunes donde la aplicación web depende de un servidor rentado significa realizar un módulo que sea capaz de dar soporte en el proceso de autentificación de usuarios de una aplicación, sin embargo en este una de las

ventajas de trabajar con una plataforma en la nube es que esta tarea es fácil de salvar gracias a los servicios que brida dicha nube. En concreto la plataforma Google App Engine brinda el servicio de soporte para administrar a los usuarios, para lo que permite el ingreso de los usuarios a las aplicaciones mediante una cuenta de google.

b) ACCESO A LOS PROYECTOS YA SEA COMO ADMINISTRADOR O INVITADO

Una vez el usuario ha ingresado a la aplicación mediante su cuenta de google, el sistema lanza un listado de proyectos en los que participa el usuario actual, ya sea como administrador o invitado. Esto quiere decir que los usuarios del sistema pueden ser administradores en más de un proyecto, y también participar en múltiples proyectos a los que haya sido invitado.

c) CREAR UN PROYECTO

Cuando un usuario de la aplicación decide crear un proyecto se le asigna de manera automática el rol de scrum manager, esto quiere decir que al igual que en los proyectos presenciales de desarrollo del software, el scrum manager es el encargado principal de velar y evaluar las buenas prácticas de desarrollo con el fin de obtener un resultado acorde a los objetivos del proyecto. Ya en el sistema este usuario con rol de scrum manager es el encargado de enviar las invitaciones de teletrabajo mediante la herramienta a los otros roles que define scrum, es decir a los miembros del equipo de desarrollo y al propietario del producto. También en este proceso de creación del proyecto el administrador es decir el scrum manager define los datos más relevantes del proyecto como ser: nombre del proyecto, plataforma de desarrollo y la fecha de inicio.

d) INGRESAR A UN PROYECTO

El ingreso a un proyecto se ha de realizar de manera directa desde el listado de proyectos en los que participa el usuario que está autenticado en el momento.

e) TELETRABAJAR BAJO LA METODOLOGÍA SCRUM

Cuando un usuario ha ingresado a uno de los proyectos en los que participa, el sistema identifica su rol en el proyecto actual, dándole así acceso a la información del proyecto y las respectivas tareas que realizará bajo su rol en el ámbito de scrum. Así por ejemplo el dueño del producto es el único rol capaz de modificar la visión general del producto, y el resto solo tiene acceso de lectura.

3.2.3.3 ANÁLISIS DE LOS REQUISITOS A NIVEL DEL SISTEMA MEDIANTE CASOS DE USO

Se pasa a la elaboración de los casos de uso, que en este apartado es a nivel del sistema, tomando en cuenta los actores identificados y las funciones descritas, lo que se puede observar en la figura 3.9 y su descripción correspondiente.

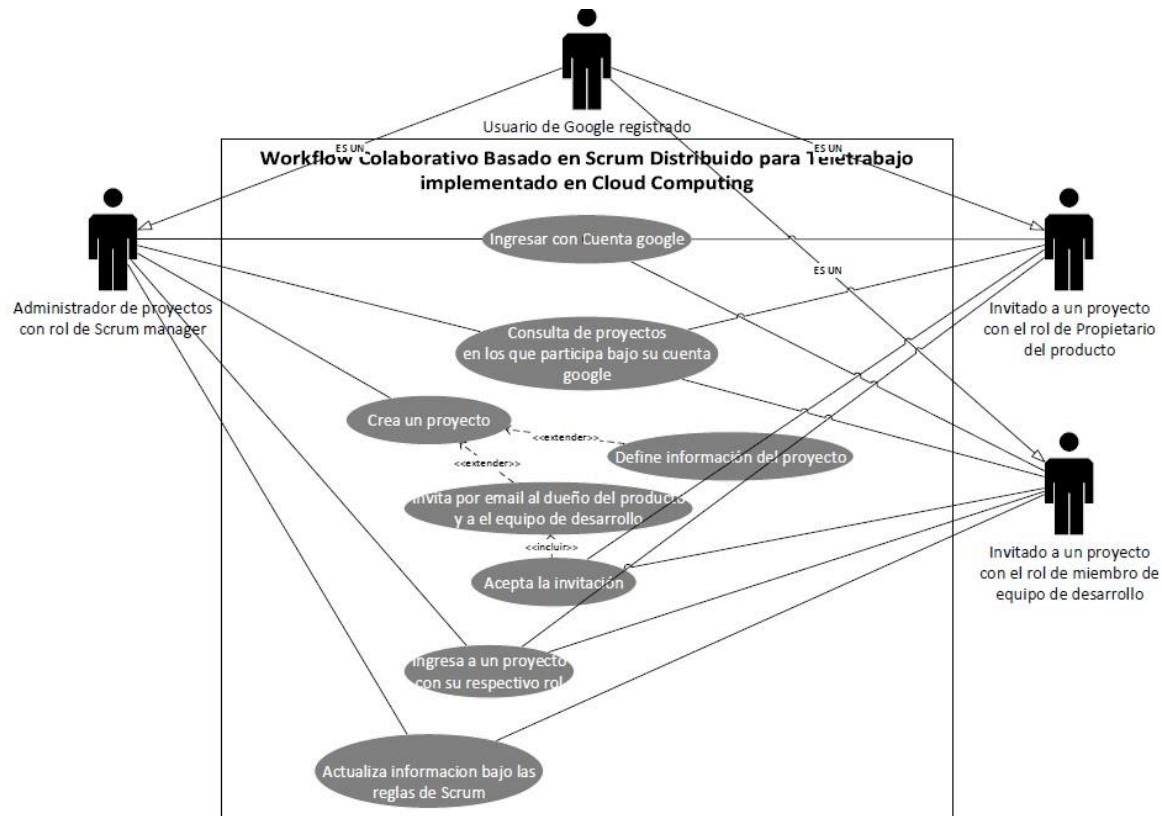


Figura 3. 9: Requisitos a nivel de sistema, basados en casos de uso

i) DESCRIPCIÓN DE CASOS DE USO

NOMBRE DE ESCENARIO	Ingresar con cuenta google
NOMBRE DE ACTOR	Administrador del proyecto scrum master, invitados al proyecto propietario del producto y el equipo de desarrollo
DESCRIPCIÓN DEL ESCENARIO	La plataforma de desarrollo proporciona una instancia de usuarios para la entrada, mediante un usuario de google

NOMBRE DE ESCENARIO	Ingresar con cuenta google
CONSULTA DE PROYECTOS EN LOS QUE PARTICIPA BAJO SU CUENTA GOOGLE	Administrador del proyecto scrum master, invitados al proyecto propietario del producto y el equipo de desarrollo
DESCRIPCIÓN DEL ESCENARIO	Al igual que una de las aplicaciones de google, Google App Engine da la opción de acceso a las aplicaciones alojadas en su nube mediante una cuenta google.

NOMBRE DE ESCENARIO	Crear un proyecto para teletrabajar con Scrum
NOMBRE DE ACTOR	Administrador del proyecto scrum master
DESCRIPCIÓN DEL ESCENARIO	Los usuarios que ingresan al sistema pueden o acceder a un proyecto o crear uno, los usuarios que crean un proyecto se les asigna el rol de scrum master

NOMBRE DE ESCENARIO	Invitar a un usuario para teletrabajar
NOMBRE DE ACTOR	Administrador del proyecto scrum master
DESCRIPCIÓN DEL ESCENARIO	Al momento de la creación de un proyecto, es cuando el sistema permite invitar a los miembros de proyecto,

	mediante un correo electrónico de invitación.
--	---

NOMBRE DE ESCENARIO	Actualización de la información de los proyectos según el rol y reglas que define scrum
NOMBRE DE ACTOR	Administrador del proyecto scrum master, propietario del producto, y equipo de desarrollo
DESCRIPCIÓN DEL ESCENARIO	La modificación de los datos de un proyecto esta validada según los roles, así por ejemplo el scrum master define o cambia los datos generales del proyecto, el equipo de desarrollo registra los avances diarios.

3.3.4 PREPARACIÓN DEL ENTORNO DE TRABAJO

Para empezar con la implementación del prototipo del presente trabajo, se realizó la instalación de los componentes de software y hardware requeridos, lo que esta explicado en el capítulo I, en la sección 1.8.4 en la página 11.

3.3.5 PRIMERA ITERACIÓN

La metodología AUP se caracteriza por la entrega de incrementos por cada una de sus fases, la teoría explica que en la primera fase que es la de inicio no es necesario la entrega de software funcional, sin embargo se debe entregar avances significativos para que en la segunda fase todo gire en torno al desarrollo de incrementos funcionales. En el caso de este trabajo se entrega un prototipo al comienzo de la materia que consiste en la entrada al sistema, o lo que se conoce la parte administrativa, para lo que se ha seguido el modelo de procesos diseñado con BPMN de manera general en los apartados anteriores. Se muestra a continuación la captura de pantallas según las tareas que se han modelado para el proceso general del sistema, y basadas en los casos de uso a nivel de sistema que muestra de manera general como se entra y administra al sistema las capturas de pantallas

correspondientes a esta primera iteración se observan en las figuras de la 3.11 a 3.19. En la figura 3.10 se puede observar el panel de control de la plataforma en la nube, donde se puede acceder a las versiones de los sistemas subidos, tráfico, y administración de una base de datos en la nube.

Version	Default	Deployed	Delete
1.	Yes	3 days, 14:07:12 ago by jrfrenato.7@gmail.com	Cannot delete default version.
2.	No	3 days, 14:20:49 ago by jrfrenato.7@gmail.com	Delete
3.	No	03/26 ago by jrfrenato.7@gmail.com	Delete

[Make Default](#)

[Traffic Splitting](#)

Learn more about uploading [Python.versions](#), [Java.versions](#), [Go.versions](#) or [PHP.versions](#).

Use the [appcfg download_app](#) command to download your application's code. If you do not want any admin to be able to download code, you can [permanently prohibit code downloads](#).

Figura 3. 10: Panel de control en la nube, de versiones subidas

En la figura 3.10 se observa la entrada al sistema, que ya está corriendo en la nube de google.

Figura 3. 11: Primera versión de la entrada al sistema.

La figura 3.12 muestra una segunda versión del sistema en la que se incorpora la parte administrativa del sistema con la implementación del servicio de entidades usuarios de google de la plataforma.



Figura 3. 12: Entrada al sistema en la nube

La figura 3.13 muestra cómo se accede a la aplicación mediante una cuenta de google.



Figura 3. 13: Entrada al sistema mediante un usuario de Google

En la figura 3.14 se puede observar que el sistema indica con que cuenta de google se ha accedido.



Figura 3. 14: Bienvenida al sistema

En la figura 3.15 se observa el primer formulario de creación de un proyecto que trabajara bajo las reglas de scrum.

A screenshot of a web browser showing the 'Crear Proyecto' form. The title bar says 'INICIO. Crear Proyecto'. The URL is 'mystic-rigging-86914.appspot.com/tutoriales?accion=prueba'. The page has a green header with the title 'PROTOPO PARA TELETRABAJO' and a sub-header 'Bienvenid@ renatito.fernandez@gmail.com'. It features a sidebar on the left with 'Documentacion' and several red buttons: 'Como Empezar', 'Los roles de los usuarios', 'Artefactos de teletrabajo', and 'Metricas de medición'. The main content area has a section titled 'DATOS DEL PROYECTO' with fields: 'Nombre Proyecto' (filled with 'planillas2'), 'Plataforma de Desarrollo' (filled with 'java'), 'Password(será enviado a sus invitados)' (filled with '.....'), 'Repita su Password' (filled with '.....'), and 'Fecha de inicio' (set to '28/03/2015'). At the bottom are 'Limpiar' and 'Siguiente' buttons. The background has a decorative sunburst graphic.

Figura 3. 15: Creación de un proyecto bajo el rol de Scrum Master

La figura 3.16 muestra como en un segundo paso de creación de un proyecto, se procede a indicar los correos electrónicos de los invitados a teletrabajar.



Figura 3. 16: El rol de Scrum Master invita mediante email a los actores del sistema con un mínimo de tres miembros del equipo de desarrollo

La figura 3.17 muestra la pantalla de bienvenida a un proyecto creado.



Figura 3. 17: Pantalla de aviso de creación del proyecto

La figura 3.18 muestra el acceso a un proyecto del sistema, mediante una contraseña del proyecto.



Figura 3. 18: Pantalla de entrada a un proyecto creado

La figura 3.19 muestra cómo se accede a distintos proyectos en los que participa una determinada cuenta. Además que en el listado se indica el rol de la cuenta en cada uno de los proyectos

Nombre del proyecto	Plataforma del Proyecto	Su rol	Ingreso
planillas	java	SCRUM MANAGER, ADMINISTRADOR DEL PROYECTO	Entrar
planillas2	java	SCRUM MANAGER, ADMINISTRADOR DEL PROYECTO	Entrar

Figura 3. 19: Listado de proyectos de la cuenta de usuario google, diferenciado por el rol en cada proyecto.

3.3 FASE DE ELABORACIÓN

La fase de elaboración, bajo la metodología AUP, trata de la implementación de las primeras implementaciones de lo modelado en la fase de inicio, es así que a continuación se realiza el despliegue de pantallas que representan los procesos especificados en la fase de inicio, de una manera inicial, es decir sin acabar del todo la implementación del proceso, centrándose así en entregas continuas.

3.3.1 SEGUNDA ITERACIÓN

En esta fase de elaboración como se indica en el apartado anterior la meta es la entrega de avances significativos pero sin la necesidad de terminar el producto, es así que los siguientes corresponden a la segunda iteración de entrega de la metodología.

3.3.2 PROCESO DE DEFINICIÓN DE LA VISIÓN DEL PRODUCTO POR PARTE DEL ROL DE PROPIETARIO DEL PRODUCTO.

El proceso de definición de la visión del producto, se manifiesta en la subida de un archivo en cualquier formato, por parte del rol dueño del producto, cumpliendo así que todos deben saber qué es lo que quiere el propietario del producto. Vea la figura 3.20



Figura 3. 20: Entre las opciones que se habilitan al rol de propietario está la de subir un documento que represente la visión del producto

3.3.3 PROCESO DE REUNIÓN DE TORMENTA DE IDEAS

La elaboración de la pila del producto, nace gracias a una reunión de tormenta de ideas donde participan todos los miembros del proyecto, en esta primera implementación se considera a esta reunión como un requisito para la elaboración de la pila del producto, sin embargo en la siguiente iteración la reunión será una opción a elegir pudiendo el rol del propietario del producto definir la pila del producto de manera autónoma, esto por razones de flexibilidad al usuario. En la figura 3.21 se muestra como se realiza la reunión con la ayuda del servicio de chat IRC, de frenode que es de libre acceso.

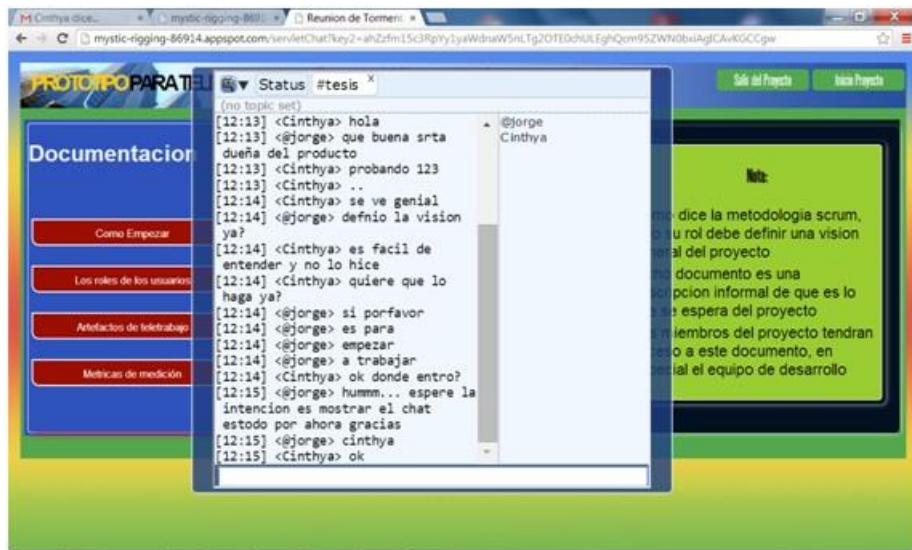


Figura 3. 21: Primera implementación para realizar la tormenta de ideas mediante el uso de un servicio IRC gratuito.

3.4 FASE DE CONSTRUCCIÓN

En la fase de construcción de la metodología AUP, se pasa a la fase final de la implementación, complementando los procesos que se empezaron a implementar en la fase de implementación.

3.4.1 MODELO DEL DOMINIO

Para cumplir con una implementación coherente y completa a continuación se hace una descripción del modelo del dominio con que se implementa el resto de la aplicación.

3.4.1.1 DIAGRAMA DE CLASES

El diagrama de clases es una referencia de cómo se organiza la base de datos, que en este trabajo se trata de una base orientada a objetos alojada en la nube. Debe tomarse que en este diagrama se representa las relaciones entre los objetos. En una base de datos relacional esto es un formalismo, sin embargo en una orientada a objetos solo representa como un objeto está asociado a otro. Ver figura 3.22

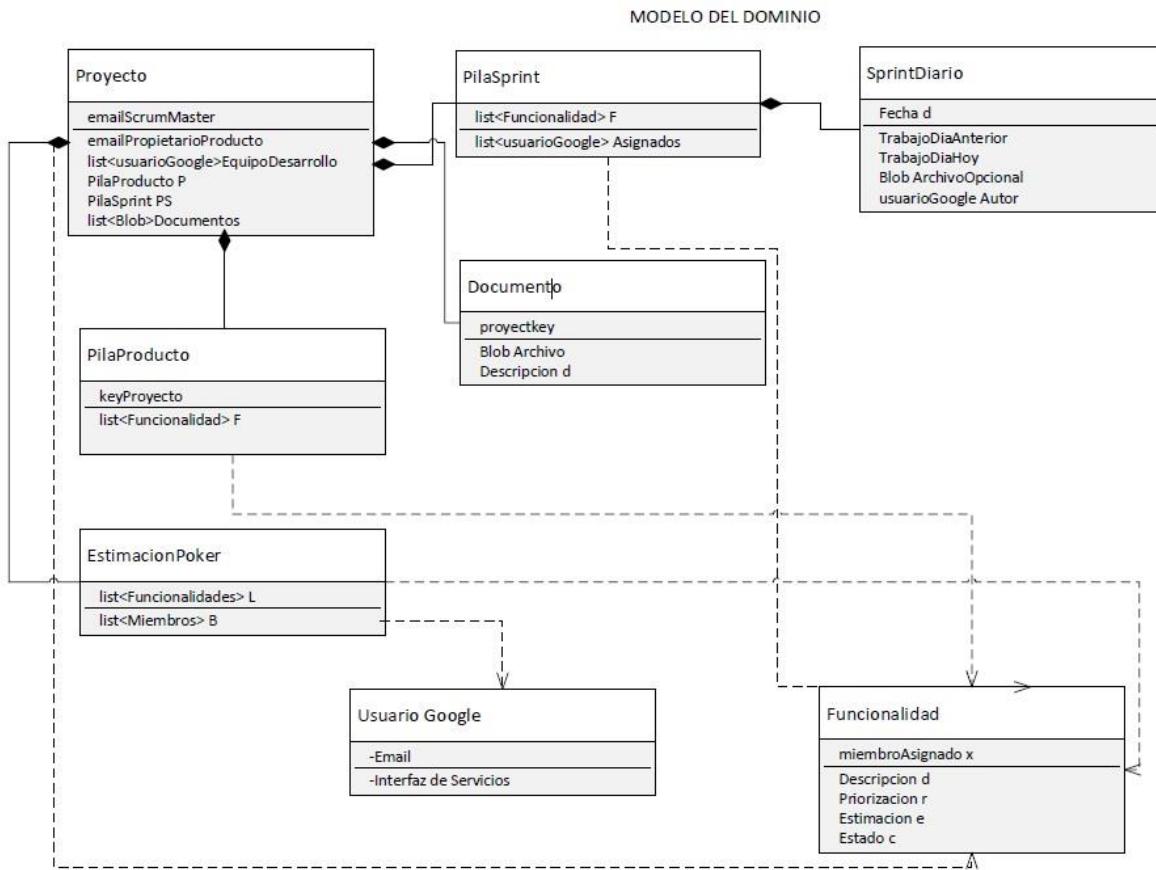


Figura 3. 22: Modelo del dominio para diseño de entidades objeto en el Datastore de Google App Engine

3.4.2 IMPLEMENTACIÓN DE ENTIDADES DEL MODELO DEL DOMINIO

Una de las ventajas de implementar una aplicación en la nube de google, es que provee soporte para la construcción de las entidades de manera directa, esto quiere decir que en lugar de la manera tradicional de construir clases que representen a las entidades de la base de datos, se hace uso de una cómoda manera de la construcción de los objetos llamando al servicio de entidades para el lenguaje java, estos detalles se los muestra en los anexos, a continuación se presenta una captura de pantalla, en la que se muestra el panel administrativo con las entidades ya construidas. Ver figura 3.23

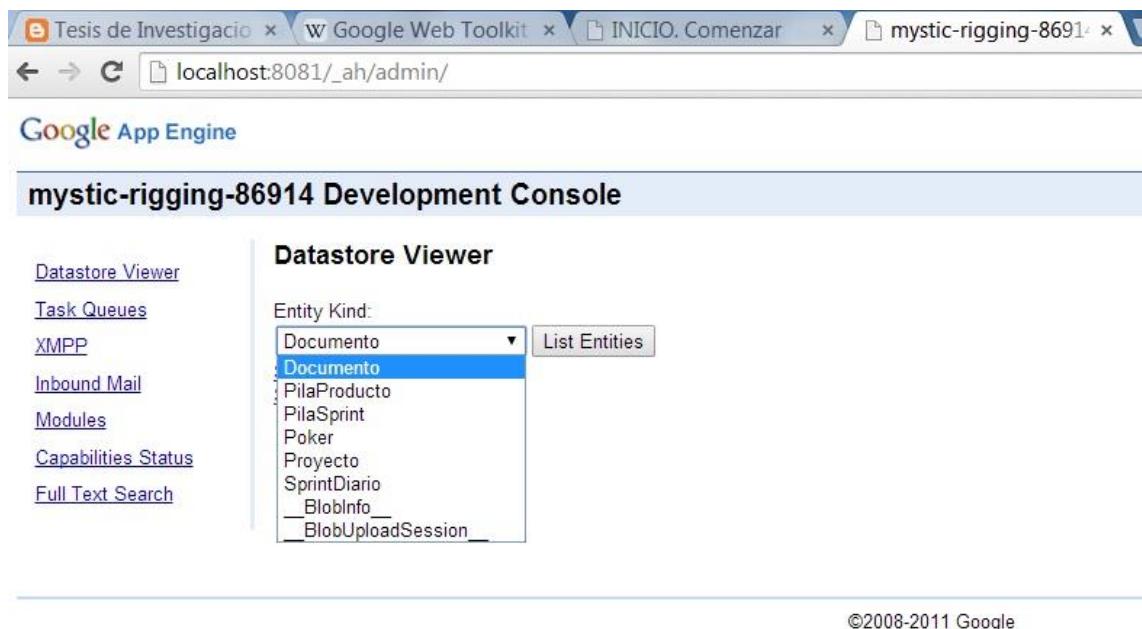


Figura 3. 23: Entidades implementadas y gestionadas desde el panel de control de GAE

3.4.3 TERCERA ITERACIÓN

A continuación se presenta la captura de pantallas para una tercera iteración de implementación, en la que se pone en práctica el modelo del dominio propuesto.

3.4.3.1 DEFINICIÓN DE LA VISIÓN DEL PRODUCTO

La definición de la visión del producto por el rol del propietario del producto, debe ser flexible en el sentido de poder subir muchos archivos y ser él quien elija cual será el documento oficial al que accederá el resto de los miembros como documento oficial, las figuras 3.24 y 3.25 son las que muestran como el propietario elige un documento oficial de entre muchos y como pueden acceder el resto al documento oficial, también se tiene la opción de eliminación de estos archivos.

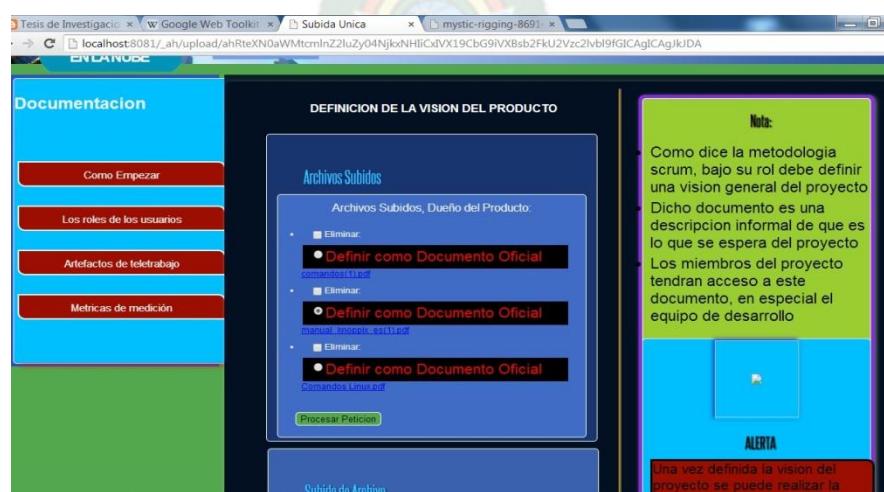


Figura 3. 24: Elección de un documento subido como el oficial

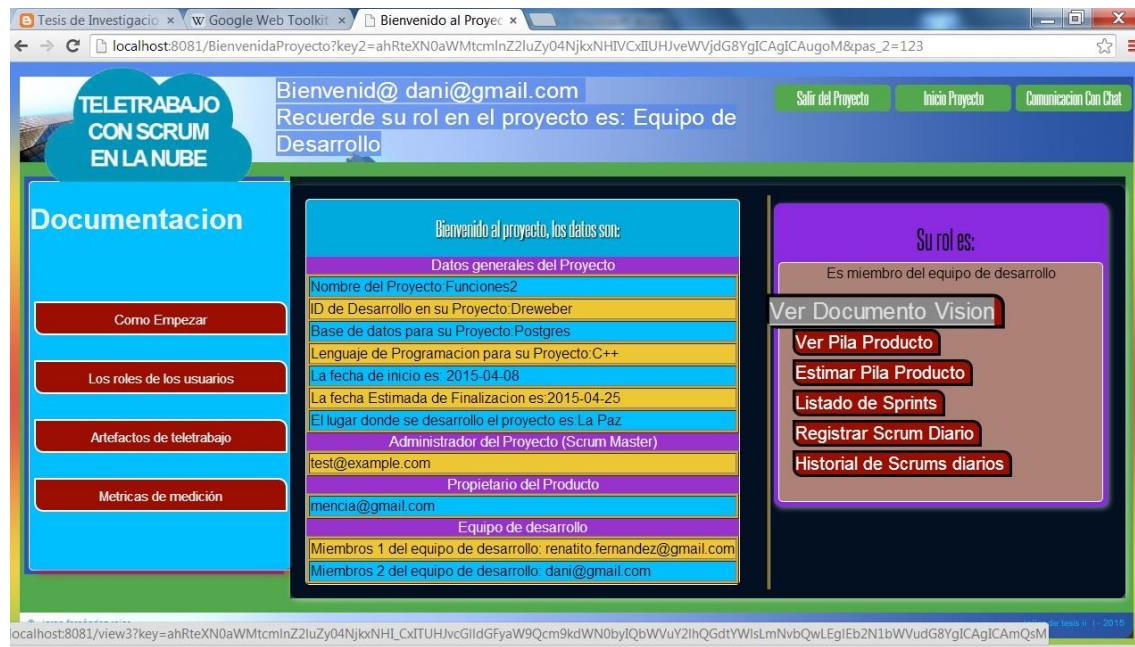


Figura 3. 25: Acceso al documento oficial elegido por el rol de propietario del producto

3.4.3.2 CREACIÓN DE LA PILA DEL PRODUCTO

Por motivos de accesibilidad al usuario la creación de la pila del producto, se realiza de manera directa, dejando la decisión de realizar una reunión de tormenta de ideas con el chat incrustado como opcional en el menú comunicación con chat, que para esta iteración se vuelve opcional. Las figuras 3.26, 3.27 muestran las pantallas de acceso al chat opcional y la creación de la pila del producto por parte del rol del propietario del producto.

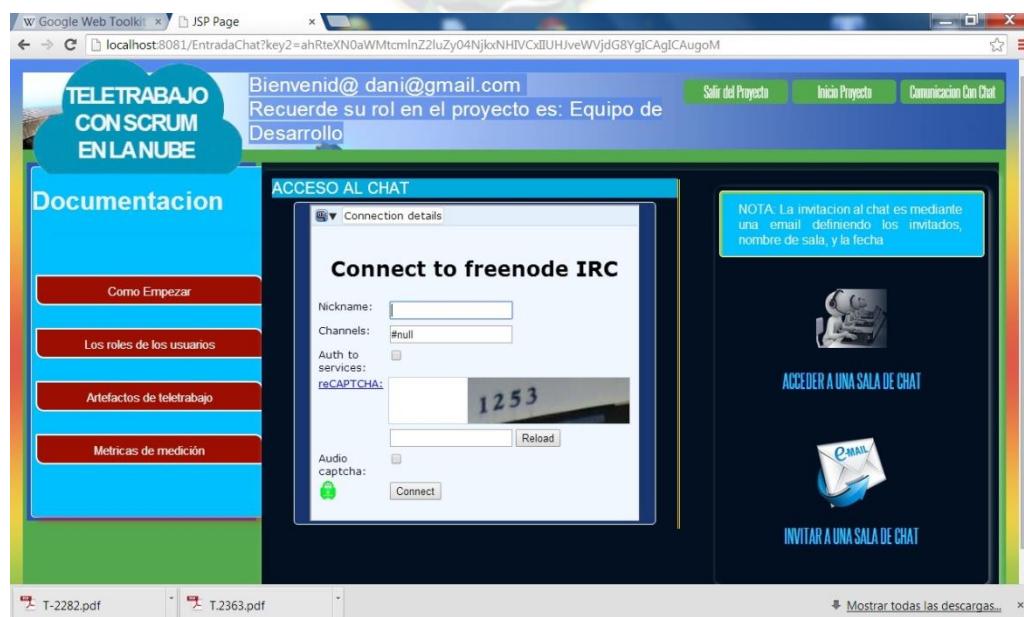


Figura 3. 26: Acceso opcional a una sala de chat que elija uno de los miembros

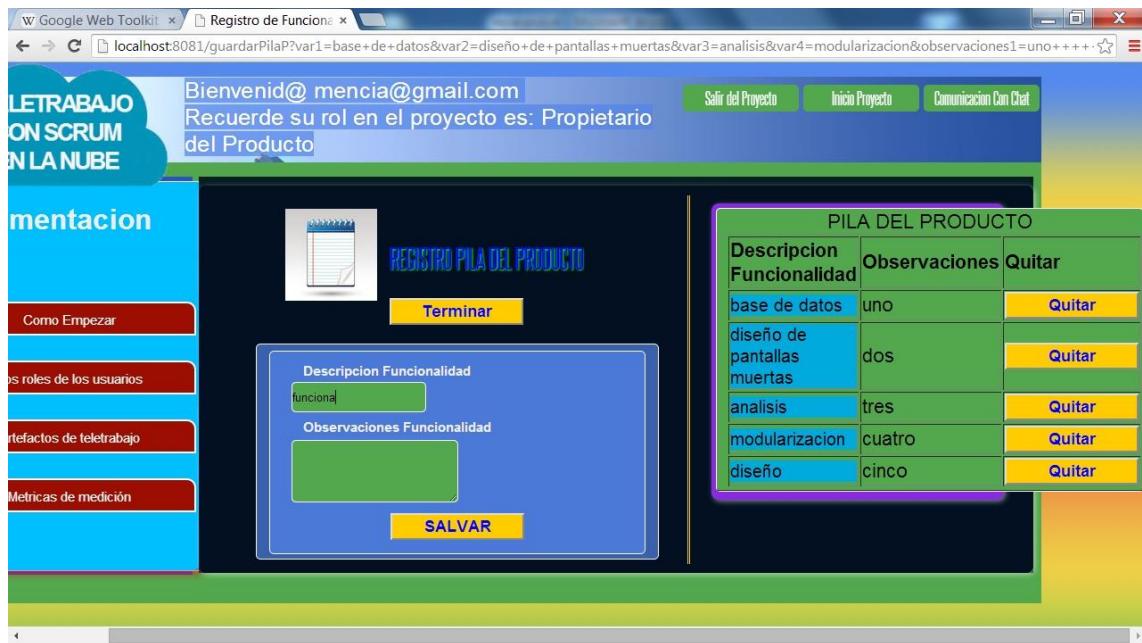


Figura 3. 27: Creación de la Pila del Producto por el rol de propietario del producto

3.4.3.2 PRIORIZACIÓN DE LA PILA DEL PRODUCTO Y SU ESTIMACIÓN

Como se ha visto en el capítulo dos las funcionalidades de la pila del producto deben cumplir que estén priorizadas y estimadas para poder ser incorporadas a un sprint, es así que se implementa estos módulos, la priorización de las funcionalidades corresponde al rol del propietario del producto y su estimación al equipo de desarrollo, en una última iteración se dará la opción de que el propietario del producto sea quien estime la pila, esto por razones de necesidad de velocidad. En las figuras 3.28 y 3.29 se muestra como el propietario del producto prioriza las funcionalidades de una sola vez, y como el equipo de desarrollo estima del tiempo de realización de las funcionalidades, mediante la estimación de póker.

Bienvenid@ mencia@gmail.com

Recuerde su rol en el proyecto es: Propietario del Producto

PRIORIZACION DE FUNCIONALIDADES DE LA PILA DEL PRODUCTO

DETALLES	FUNCIONALIDAD
Alta	FUNCIONALIDAD UNO
Alta	FUNCIONALIDAD DOS
Alta	FUNCIONALIDAD TRES
Alta	FUNCIONALIDAD CUATRO
Alta	FUNCIONALIDAD CINCO

DETALLES

VALOR

Descripción: diseño

Observación: cinco

Estimación: null

Priorización: null

Nro De Sprint en que se desarrolla: null

PRIORIZAR

Figura 3. 28: Priorización de la Pila del Producto por el rol de propietario del producto

Bienvenid@ mencia@gmail.com

Recuerde su rol en el proyecto es: Equipo de Desarrollo

ESTIMACION CON EL JUEGO DE POKER

ESTIMACION

Escoger Funcionalidad

base de datos

0 1 2 3 5

Estimación en Días

• 0 días • 1 días • 2 días • 3 días • 5 días

6 7 8 9

T-2282.pdf T.2363.pdf

Mostrar todas las descargas...

Figura 3. 29: Estimación de la Pila del Producto por los miembros del equipo con la estimación de póker

3.4.3.3 CREACIÓN DE LA PILA DEL SPRINT

A pesar de que el modelo del dominio indica que una entidad proyecto está compuesta de las pilas del producto y del sprint, en la teoría de los procesos de la metodología scrum, se sabe que la pila del sprint no puede existir si no hay una pila del producto, este hecho esta validado de tal manera que no se puede crear la pila de un sprint si no se tiene la pila del producto. Además como mínimo para que una funcionalidad se incorpore a la pila del sprint debe estar estimado, esta condición también se toma en cuenta. En la figura 3.30 se ve la creación de una pila del sprint.

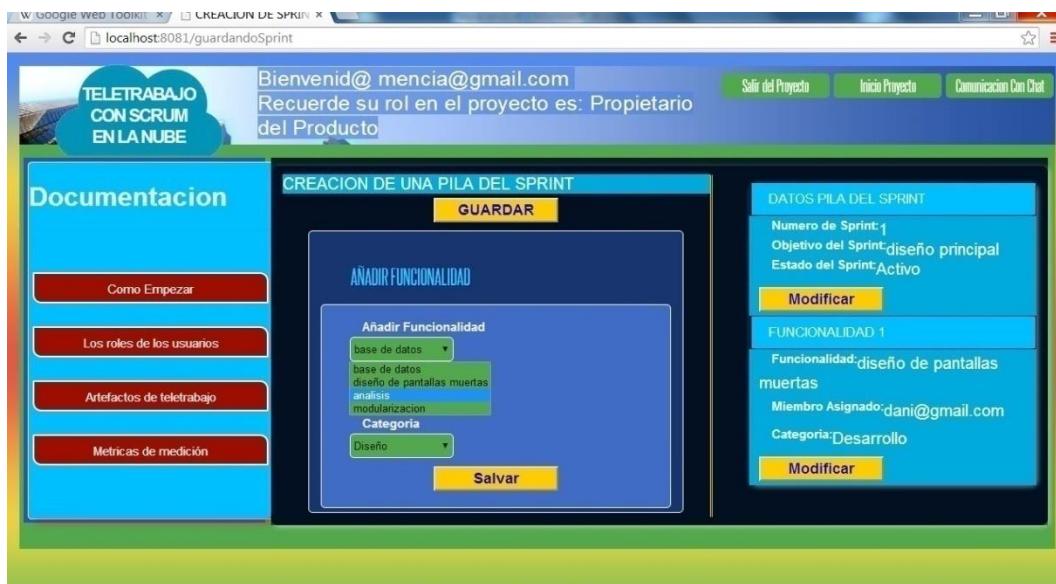


Figura 3. 30: Creación de la Pila del Sprint bajo las condiciones dadas

3.5 LA FASE DE TRANSICIÓN

En esta fase se debe realizar las pruebas funcionales ya sea por parte de los desarrolladores o también por el cliente, es así que las pruebas realizadas para el presente trabajo, corresponden tanto al desarrollador del trabajo como a una prueba piloto realizada con alumnos de la materia de taller de sistemas de información cuya sigla es INF-281. El objeto de esta fase es comprobar que la herramienta se ajusta a los requisitos del cliente, que en este contexto son los objetivos e hipótesis planteados en el marco introductorio del trabajo. Explicado para qué sirve la fase de transición se procede a desarrollar el Capítulo 4 el marco demostrativo, en el que se explica los medios utilizados para realizar la prueba piloto con alumnos de la carrera de informática. La explicación del porque se realiza la prueba con los alumnos de la carrera es simple, y es que la herramienta colaborativa para teletrabajo desarrollada en este trabajo está diseñada para desarrolladores de software que trabajan bajo la metodología scrum.



CAPITULO 4

MARCO DEMOSTRATIVO

4.1 INTRODUCCIÓN

En este capítulo se demostrará la hipótesis del trabajo. Para esto se plantea realizar los siguientes pasos en el marco del estudio cualitativo, y en las pruebas piloto que una herramienta de teletrabajo debe cumplir en sus fases de pruebas finales, como se describió en el capítulo II apartado 2.8.6.1 incisos A, página 55.

- Descripción de lo que se quiere probar
- Definición de la población y muestra
- Prueba piloto de la herramienta de teletrabajo
- Recolección de datos de estudio mediante encuestas y el panel de control de la plataforma de desarrollo Google App Engine.
- Análisis e interpretación de los datos
- Conclusión del estudio.

A continuación se desarrollan los pasos anteriores con el objetivo de probar la hipótesis del trabajo.

4.2 DESCRIPCIÓN DE LA PRUEBA

Para probar una hipótesis del tipo cualitativo, como es el caso del presente trabajo, se identifica a las variables dependientes que son las que hacen que la hipótesis tome un valor de verdadero o falso, en base a un estudio de las condiciones que perciben los sujetos de estudio, además de la recolección de datos sobre el uso de recursos en la nube con el fin de ver el teletrabajo que se lleva a cabo. Es así que para el trabajo se pretende ver si los usuarios que probaron la herramienta, perciben una mejoría en su teletrabajo con respecto a la comunicación, colaboración y coordinación bajo la metodología scrum con la utilización del sistema colaborativo implementado en la nube, acompañado de una recolección de datos sobre los recursos utilizados en la plataforma de desarrollo en la nube. La hipótesis del trabajo y la variable independiente de esta son:

- **HIPÓTESIS:** “Si se implementa software como servicio en la nube, en una plataforma de desarrollo de aplicaciones web pública como Google App Engine, que modele y automatice los procesos y flujos de información de la metodología ágil Scrum, entonces se contará con una herramienta colaborativa para teletrabajar mejorando así las actividades de comunicación, coordinación y colaboración al interior de un equipo de desarrollo de software de habla hispana que trabaja con la metodología scrum en un escenario donde los miembros tienen la posibilidad de reunirse personalmente”.
- **VARIABLE DEPENDIENTE:** “comunicación, coordinación y colaboración entre los miembros del equipo de desarrollo software que implementen la herramienta colaborativa para teletrabajo”

4.3 POBLACIÓN Y MUESTRA

La identificación de la población y muestra de manera explícita es una parte importante de las pruebas piloto para las herramientas de teletrabajo, por los que se detalla estos.

- **POBLACIÓN:** La población para el presente trabajo son los equipos de desarrollo software, que en su trabajo utilizan los procesos de desarrollo definidos en la metodología ágil scrum.
- **MUESTRA:** La muestra de estudio que se escogió para el trabajo, son los equipos de trabajo que se forman en la materia de “Taller de Sistemas de Información-INF281” de la carrera de Informática de la UMSA.

4.4 LA PRUEBA PILOTO

La herramienta colaborativa para teletrabajo se prueba con los estudiantes de informática de la carrera de informática de la UMSA, que desarrollan sus proyectos para la materia de INF-281 “Taller de Sistemas de información”. Se asistió al aula a fin de dar a conocer de los que se trataba el presente trabajo, para luego fueran ellos los que evalúen mediante una encuesta, en qué medida se satisface la variable independiente del trabajo y los objetivos planteados en el marco introductorio.

La encuesta es realizada a una muestra de 20 alumnos, correspondiente a 5 equipos de trabajo. A continuación se muestra las preguntas que se han incluido en la encuesta, y la correspondencia con los objetivos “O_i” planteados para el trabajo.

Pregunta	Objetivo al que Corresponde
1.1 El sistema colaborativo cumple con los objetivos de una herramienta colaborativa para teletrabajo bajo la metodología Scrum	O1.- Implementar una herramienta colaborativa, accesible desde internet.
1.2 El programa cumple con las metas generales de mejora de la comunicación, colaboración y coordinación en un ambiente donde el equipo de desarrollo implementa el teletrabajo. 4.1 Las diferentes interfaces de trabajo de la herramienta, tienen un título claro que identifica el objetivo de cada una de las interfaces. 4.2 Se proporciona la información necesaria para trabajar con la herramienta y para entender el trabajo bajo la metodología Scrum 4.3 El acceso a la herramienta colaborativa es posible desde los distintos navegadores y dispositivos de uso masivo actuales. 5.1 Cada proyecto definido al interior de la herramienta colaborativa, guardan su propósito y lógica en los sucesivos accesos a la herramienta. 5.2 Los responsables de los proyectos definidos con la herramienta colaborativa tienen los permisos suficientes sobre cada proyecto. 6.1 Se proporciona la posibilidad de reportar errores o consultas sobre la herramienta colaborativa.	OG.- Desarrollar un sistema workflow colaborativo para teletrabajo, sencillo de gestionar, que sea accesible desde la nube publica de google con solo el requisito de tener una cuenta google, que modele los procesos, y flujos de trabajo de la metodología ágil Scrum en su versión distribuida para mejorar la colaboración, comunicación, coordinación al interior de un equipo de desarrollo de software de habla hispana que implemente el teletrabajo.
2.1 El diseño del programa identifica	O2.- Modelar los flujos y procesos de

adecuadamente los perfiles o roles y las características del participante.	información de la metodología scrum con el lenguaje de especificación estándar BPMN, según los roles que define esta metodología.
<p>3.4 Se da la opción del intercambio de información, de forma contextualizada entre los miembros de un proyecto al interior de la herramienta colaborativa.</p> <p>3.5 La herramienta colaborativa implementa los artefactos con los que scrum se apoya para el desarrollo de software.</p>	O3.-Modelar los artefactos que define scrum para el proceso de construcción de software, en particular la pila del producto y del sprint.

4.4 CONTROL DEL TELETRABAJO CON EL PANEL DE CONTROL

En la prueba piloto con los alumnos de “INF-281”, se verifico adicionalmente que los equipos tengan acceso a la herramienta y utilicen los recursos de la plataforma en la nube, para lo que se accedió al panel de control de GAE. La figura 4.1 muestra la entrada al panel de control en la nube, se puede apreciar que se puede alojar y desarrollar más de una aplicación por una cuenta de usuario google.



Figura 4. 1: Entrada al panel de administración de GAE

En la figura 4.2, muestra cómo se puede acceder la información sobre los recursos utilizados por las distintas aplicaciones.

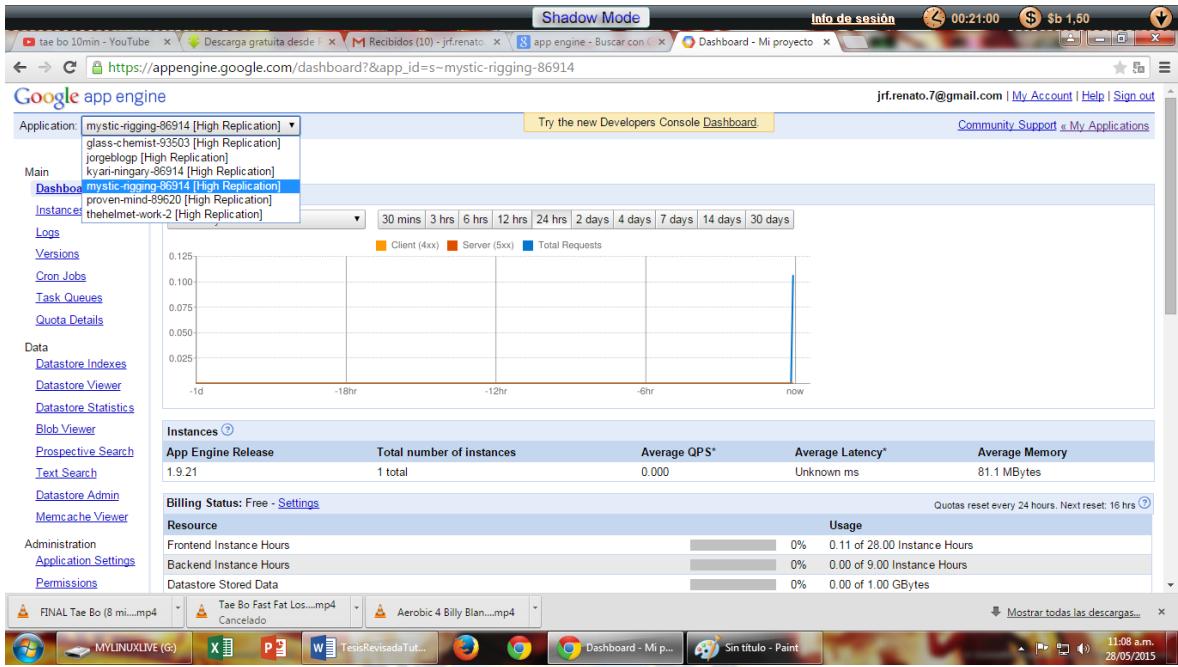


Figura 4. 2: Información de consumo de los recursos y servicios en la plataforma por cada aplicación

La figura 4.3, muestra la administración de cada aplicación de forma individual, se puede observar cómo se puede tener acceso a las distintas versiones que se haya subido a la plataforma.

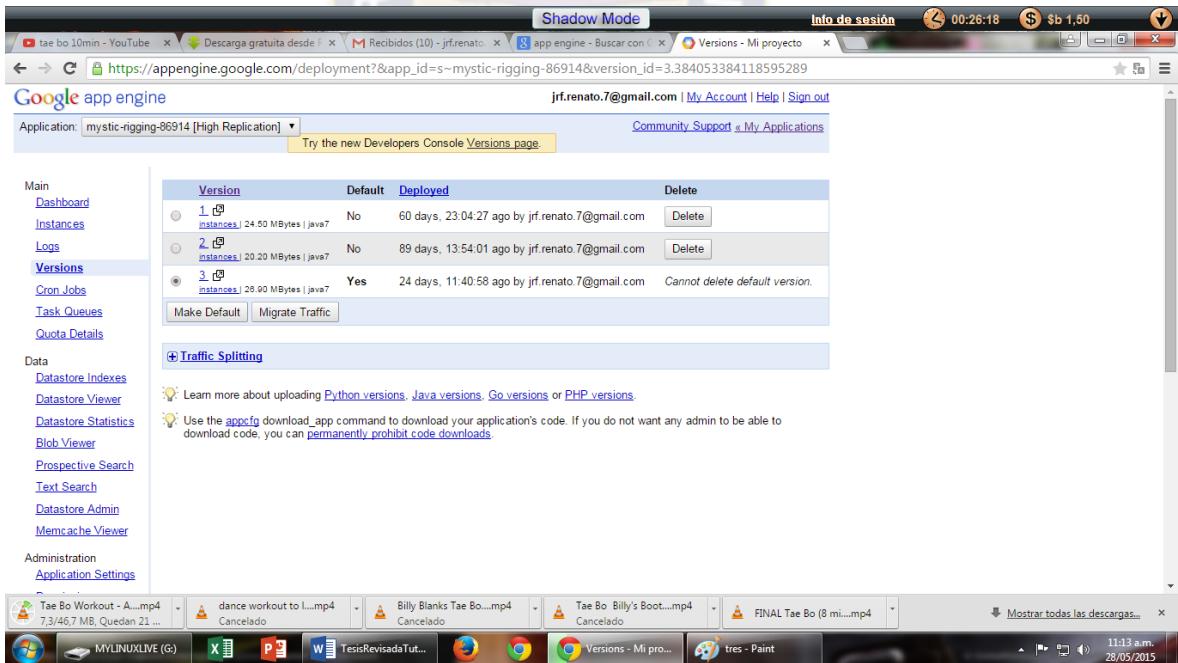


Figura 4. 3: Versionado de las instancias subidas de una aplicación

En la figura 4.4 se muestra el acceso al almacén de datos al que se denomina Datastore, en particular se observa las entidades correspondientes a la Kind (que es equivalente a una tabla en el modelo relacional) proyecto, estas entidades corresponden a las que se crearon por los estudiantes de informática en la prueba piloto.

The screenshot shows the Google App Engine Datastore Explorer interface. The main title bar includes 'Shadow Mode', 'Info de sesión', '00:28:45', '\$ \$b 1,50', and a user icon. The URL is https://appengine.google.com/datastore/explorer?submitted=1&app_id=s~mystic-rigging-86914&show_options=no&version_id=3.384053384118595289&kind=. The left sidebar has links for Dashboard, Instances, Logs, Versions, Cron Jobs, Task Queues, Quota Details, Data, Datastore Indexes, Datastore Viewer, Datastore Statistics, Blob Viewer, Prospective Search, Text Search, Datastore Admin, Memcache Viewer, Administration, Application Settings, Permissions, Blacklist, Admin Logs, Billing, Billing Status, Usage History, and Resources. The main content area is titled 'Proyecto Entities' with a table showing columns: ID/Name, BaseDatos, Contraseña, Departamento, EmailCincoEquipo, EmailCuatroEquipo, EmailDosEquipo, EmailIP, EmailSM, EmailSeisEquipo, EmailTresEquipo, and EmailUnoEquipo. The table lists 12 entities with various details like MySQL, Oracle, cooperativa, La Paz, etc., and their corresponding email addresses. At the bottom of the table are 'Delete' and 'Flush Memcache' buttons.

Figura 4. 4: Entidades de la kind Proyecto creadas en la prueba piloto

En la figura 4.5 se muestra las estadísticas correspondientes al uso de espacio utilizado por la aplicación, a lo que se denomina archivos estáticos y que la plataforma GAE maneja de forma independiente a los datos dinámicos del Datastore, lo que asegura que la performance de la aplicación no se vea afectada por archivos estáticos.

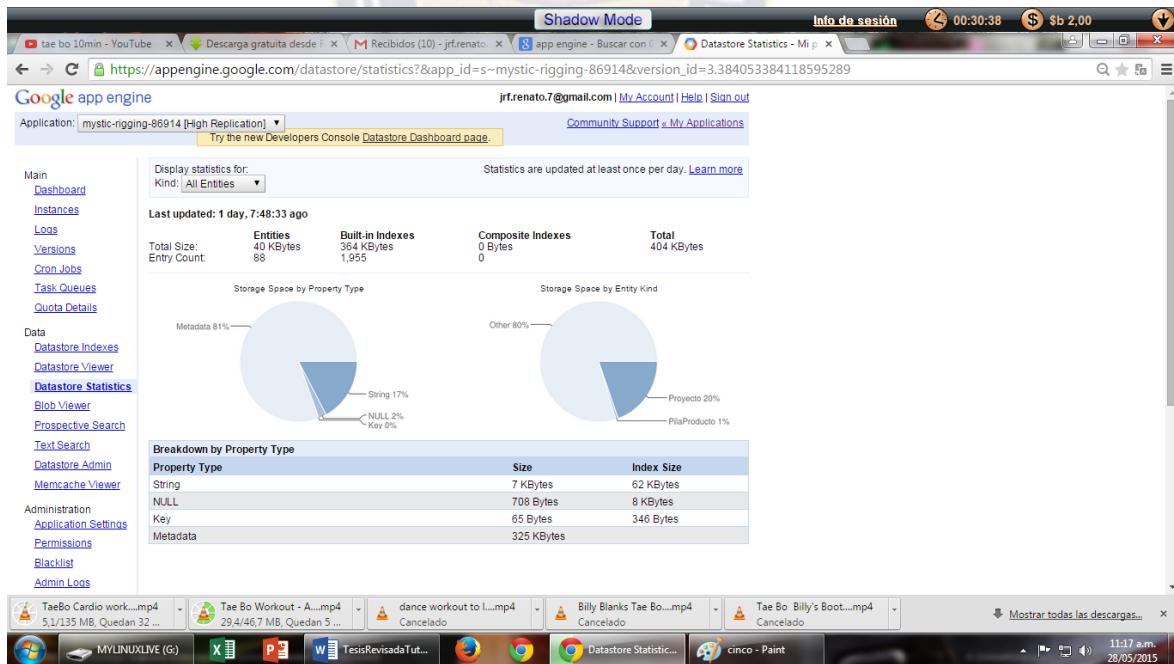


Figura 4. 5: Gráficos de información sobre los archivos estáticos de la aplicación

La figura 4.6 muestra como un administrador puede las veces que ha modificado la aplicación, esto es de importancia en los casos en la aplicación tenga más de una administrador.

Figura 4. 6: Información correspondiente a las modificaciones de la aplicación por el administrador

4.5 ANÁLISIS E INTERPRETACIÓN DE DATOS RECOGIDOS

Para el análisis de datos, se procesa la encuesta con la distribución Chi-cuadrado, a fin de ver si se la variable dependiente de la hipótesis se cumple.

- Funcionalidad: esta propiedad evalúa si el software se adecua a la funcionalidad esperada, que en la encuesta está representada por las preguntas 1.1, 1.2 y 1.3.

Los resultados obtenidos son:

Respuestas en porcentajes correspondientes a 20 encuestas:

Nro	Funcionalidad		
	1.1	1.2	1.3
1	A	M	M
2	A	A	A
3	M	M	B
4	B	M	M
5	M	A	A
6	M	A	M
7	M	M	M
8	M	A	A
9	B	M	M
10	B	M	A
11	B	M	M
12	M	B	M
13	B	M	M
14	M	B	M
15	B	M	M
16	M	B	A
17	B	M	M
18	B	M	A

19	A	M	A
20	M	A	M

El detalle en porcentajes es:

Alto (A) = 15 = 25% de respuestas Alto para la sección del “contexto”

Medio (M)= 33= 55% de respuestas Medio para la sección del “contexto”

Bajo (B)= 12= 20% de respuestas Bajo para la sección del “contexto”

Total Respuestas = 60

- b) Confiabilidad: Evalúa las herramientas que incluye el software según la funcionalidad para la que ha sido diseñado. En la encuesta las preguntas correspondientes son: 2.1, y 2.2

Respuestas en porcentajes correspondientes a 20 encuestas:

Nro	Confiabilidad	
	2.1	2.2
1	M	A
2	B	M
3	A	B
4	A	M
5	M	B
6	M	M
7	A	M
8	B	A
9	B	A
10	A	M
11	M	M
12	M	M
13	B	M
14	M	M
15	M	A
16	B	M
17	M	M
18	B	M
19	M	M
20	M	M

El detalle en porcentajes es:

Alto = 8 = 20% de respuestas Alto para la sección del “recursos”

Medio = 24 = 60% de respuestas Medio para la sección del “recursos”

Bajo = 8 = 20% de respuestas Bajo para la sección del “recursos”

Total Respuestas = 40

- c) Eficiencia: Evalúa el cumplimiento de las necesidades técnicas para la puesta en práctica del software, las preguntas correspondientes en la encuesta son: de la 3.1 a la 3.9

Respuestas en porcentajes correspondientes a 20 encuestas:

Nro	Eficiencia								
	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9
1	A	A	A	M	A	A	A	A	M
2	M	A	A	A	M	M	M	A	M
3	M	A	A	A	A	A	M	A	A
4	A	M	M	M	A	M	A	B	A
5	A	A	A	B	A	A	A	B	A
6	B	A	A	A	B	B	A	A	A
7	B	A	A	B	A	B	A	A	A
8	M	A	A	M	A	B	A	B	A
9	A	A	A	M	A	B	A	A	B
10	A	A	B	A	B	M	A	A	A
11	A	B	B	A	A	A	B	A	A
12	M	A	A	B	A	A	M	A	A
13	A	A	A	B	B	A	B	A	A
14	B	B	B	A	B	M	A	A	B
15	B	A	A	A	A	A	M	A	A
16	A	A	M	B	B	A	B	A	A
17	A	A	A	A	A	M	A	A	B
18	M	A	M	B	A	B	A	A	A
19	A	A	A	B	A	B	B	A	A
20	A	M	A	A	A	A	A	A	A

El detalle en porcentajes es:

Alto = 117 = 65% de respuestas Alto para la sección del “requerimientos Plataforma”

Medio = 27 = 15% de respuestas Medio para la sección del “requerimientos Plataforma”

Bajo = 36 = 20% de respuestas Bajo para la sección del “requerimientos Plataforma”

Total Respuestas = 180

- d) Accesibilidad: Evalúa si el software es accesible para los usuarios, en cuanto a la funcionalidad, las preguntas correspondientes son de la 4.1 a la 4.5

Respuestas en porcentajes correspondientes a 20 encuestas:

Nro	Accesibilidad				
	4.1	4.2	4.3	4.4	4.5
1	A	M	B	B	B
2	A	M	B	B	M
3	M	A	M	A	B
4	M	A	A	M	M
5	M	A	M	A	B
6	A	M	A	A	M
7	M	M	A	A	M
8	B	B	B	A	M

9	A	B	M	M	B
10	M	A	A	A	M
11	M	A	A	M	M
12	A	A	M	M	B
13	A	A	B	A	M
14	A	A	A	M	B
15	M	B	A	A	B
16	M	M	A	M	A
17	M	M	A	A	B
18	M	A	M	A	A
19	A	A	A	M	B
20	M	M	B	M	M

El detalle en porcentajes es:

Alto = 40 = 40% de respuestas Alto para la sección del “Accesibilidad”

Medio = 40 = 40% de respuestas Medio para la sección del “Accesibilidad”

Bajo = 20 = 20% de respuestas Bajo para la sección del “Accesibilidad”

Total Respuestas = 100

- e) Usabilidad: Evalúa si la herramienta cumple con el objetivo de su diseño, las preguntas correspondientes son de la 5.1 a la 5.4

Respuestas en porcentajes correspondientes a 20 encuestas:

Nro	Usabilidad			
	5.1	5.2	5.3	5.4
1	A	M	B	B
2	M	A	M	B
3	M	A	B	M
4	A	A	A	M
5	M	B	M	A
6	B	M	M	A
7	B	M	A	A
8	M	A	A	A
9	B	A	A	A
10	A	M	M	B
11	B	A	M	B
12	A	A	B	M
13	B	A	M	B
14	A	B	A	M
15	M	A	A	A
16	B	M	B	A
17	M	B	A	M
18	M	M	M	M
19	M	B	M	M
20	B	M	M	M

El detalle en porcentajes es:

Alto = 28 = 35% de respuestas Alto para la sección del “Usabilidad”

Medio = 32 = 40% de respuestas Medio para la sección del “Usabilidad”

Bajo = 20 = 25% de respuestas Bajo para la sección del “Usabilidad”

Total Respuestas = 80

- f) Mantenibilidad: Evalúa si la herramienta hace un seguimiento de sus objetivos y brinda un asesoramiento continuado. Las preguntas correspondientes a esta sección son de la 6.1 a la 6.7

Respuestas en porcentajes correspondientes a 20 encuestas:

Nro	Mantenibilidad						
	6.1	6.2	6.3	6.4	6.5	6.6	6.7
1	A	M	M	A	B	A	M
2	A	A	M	A	A	A	A
3	A	A	A	B	M	A	B
4	M	A	M	M	A	A	A
5	M	A	A	A	A	A	A
6	A	B	A	A	M	A	B
7	A	M	A	A	B	M	A
8	M	A	A	M	B	A	A
9	A	M	A	M	A	B	A
10	M	M	A	A	B	A	A
11	A	A	B	A	A	A	B
12	A	B	A	A	M	B	A
13	M	A	A	M	B	A	B
14	A	A	B	A	A	M	A
15	A	A	A	A	B	A	A
16	M	A	A	M	A	A	A
17	A	A	B	A	A	M	M
18	A	M	A	A	B	M	M
19	B	A	M	A	A	M	B
20	M	A	M	A	A	M	M

El detalle en porcentajes es:

Alto = 84 = 60% de respuestas Alto para la sección del “Seguimiento”

Medio = 35 = 25% de respuestas Medio para la sección del “Seguimiento”

Bajo = 21 = 15% de respuestas Bajo para la sección del “Seguimiento”

Total Respuestas = 140

4.5.1 EVALUACIÓN DE LA ENCUESTA CON LA DISTRIBUCIÓN ESTADÍSTICA CHI-CUADRADO

Como segundo paso del análisis de resultados se trabaja con la distribución Chi-cuadrado en base a las secciones propuestas en la encuesta y las cantidades estadísticas analizadas en la sección anterior, para lo que se toma el total de las 600 respuestas de la encuesta correspondientes a un total de 20 encuestas. Además las respuestas correspondientes a Alto y Medio de cada una de las seis secciones se consideran como el éxito traducido a

que existe satisfacción con la herramienta de teletrabajo, el hecho de que se mida la satisfacción como éxito está en relación con lo que el objetivo de la investigación cualitativa, que mide la percepción de los sujetos de estudio.

Las variables que considera la distribución Chi-cuadrado son: F_o que es la frecuencia observada del éxito que se mide, y F_e representa la frecuencia esperada de éxito.

	Contexto	Recursos	Requerimientos	Accesibilidad	Usabilidad	Seguimiento
F_o (frecuencia observada)	48	32	144	80	60	119
F_e (frecuencia esperada)	60	40	180	100	80	140
Valor Chi Cuadrado	2.4	1.6	7.2	4	5	3.15

El valor correspondiente para un grado de libertad de 5 (número de variables) y una confiabilidad de 0.05 es de: 4,4 y el valor Chi-cuadrado calculado para las encuestas es de 23.35 con lo que se acepta que la variable dependiente: “comunicación, coordinación y colaboración entre los miembros del equipo de desarrollo software que implementen la herramienta colaborativa para teletrabajo” es verdadera y por tanto la hipótesis planteada es verdadera.

4.5 CONCLUSIONES SOBRE EL ESTUDIO

En base al análisis elaborado de la encuesta se evidencia que los objetivos del trabajo se han cumplido, por lo que la hipótesis del trabajo queda demostrada.

CAPITULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- Se ha logrado contrastar las plataformas de desarrollo en la nube, en el capítulo 2 apartado x página x, explicando el porqué se ha escogido la plataforma GAE como de desarrollo para el presente trabajo.
- Se expuso y diferenció los escenarios de teletrabajo posibles cuando se pone en práctica la metodología scrum en su versión distribuida, para luego escoger un escenario de teletrabajo parcialmente distribuido, para el diseño de la herramienta colaborativa de teletrabajo.
- Ejemplificación del modelado e implementación de un sistema colaborativo de tipo workflow, con el uso de estándares aceptados por las organizaciones que actualmente se especializan en el desarrollo de sistemas workflow aglutinadas en la “Workflow Management Coalition”, las cuales son BPMN y UML.
- Se logró la identificación de procesos y flujos de información de la metodología scrum, para el modelado del workflow colaborativo.
- Se alojó la aplicación web colaborativa de tipo workflow, en la nube de google, dando a esta la característica de una herramienta de teletrabajo, que es la de ser accesible desde cualquier dispositivo que esté conectado a la red.
- Se llega a la conclusión de que la implementación de software como servicio en un para teletrabajo es beneficioso tanto para los teletrabajadores como para los proveedores de servicio, ya que se abstraerá a los usuarios de la administración de los recursos además de asegurarle un soporte permanente.

Además de los objetivos anteriores que se logró cumplir en el presente trabajo, también se demostró la hipótesis del trabajo con la investigación cualitativa, a continuación se resalta la conclusión con respecto a la hipótesis del trabajo.

- Se ha logrado mejorar la comunicación, colaboración y coordinación al interior de un equipo de desarrollo de software que trabaja con Scrum como metodología de desarrollo, mediante la implementación de una herramienta colaborativa alojada en la nube de Google, “Google App Engine” diseñada para teletrabajo ya que cumple con los requerimientos de esta según el libro blanco para implementación de esta modalidad de trabajo, ya que incorpora artefactos de colaboración como la pila producto, pila del sprint, definición de la visión

de proyecto, además de la inclusión de una sala de chat para la comunicación entre los miembros de los equipos de trabajo.

5.2 RECOMENDACIONES

- Para teletrabajar con la herramienta colaborativa, se debe tener en cuenta que los miembros del equipo de desarrollo debe tener la posibilidad de encontrarse de manera personal al menos una vez por semana, para completar su trabajo.
- La herramienta de teletrabajo implementada solo considera un medio de comunicación síncrono que es el chat, sin embargo los usuarios pueden usar otros medios de comunicación disponibles en internet más sofisticados como Skype.
- Realizar herramientas colaborativas para teletrabajo para distintas áreas del conocimiento, considero alertas del sistema de tal manera de tener una herramienta lo más interactiva posible.
- Considerar el desarrollo de sistemas en la nube en las modalidades de plataformas como servicio o hardware como servicio, puesto que en la carrera a la fecha no existe ningún trabajo hecho en estas modalidades.
- Realizar una herramienta colaborativa de teletrabajo para scrum, que lance estadísticas de avance según los artefactos pila del producto y del sprint.
- Proponer como proyecto de grado, la elaboración de una aplicación móvil en una plataforma en la nube como Google Play, para una empresa, con el objeto de que sea comercial.
- Proponer implementar una herramienta de teletrabajo para un escenario totalmente distribuido.



BIBLIOGRAFÍA

Bibliografía

- Analitica. (2007). *Manual de diagramación de Procesos.*
- AppFog, C. (12 de Enero de 2015). www.appfog.com . Recuperado el 12 de Enero de 2015, de www.appfog.com : www.appfog.com
- Cloud, Z. (21 de Enero de 2015). *Zend Cloud.* Recuperado el 22 de Enero de 2015, de Zend Cloud: www zend com
- Colombia, G. y. (2014). *Libro Blanco el ABC del teletrabajo en Colombia -Version3-*. Medellin: www.colombiadigital.net.
- Condori, L. F. (2010). Casos de Uso Cloud Computing. *Información Tecnología e Infromación* , 4.
- David, C. (1997). *Comunidad Virtual Worflow y Trabajo Cooperativo.*
- Flores, L. E. (2009). *PROCESO UNIFICADO ÁGIL* . 7.
- Gerónimo, G. (2002). *Breve Introducción a los Sitemas Colaborativos Groupware y Workflow.*
- Geronimo, G. (2002). Sistemas Colaborativos: Groupware & Workflow. Leon, Oaxaca , Mexico.
- González, J. F. (2004). *Introducción a las Metodologías Ágiles .*
- Graván, P. R. (2011). Posibilidades Formativas de las Herramientas Groupware. Sevilla, Sevilla, España.
- Institut, H. P. (Agosto de 2012). *Business Process Technology.*
- java, D. R. (17 de Diciembre de 2012). refecardaz.com. Recuperado el 20 de Diciembre de 2015, de refecardaz.com: refecardaz.com
- Jimenez, J. D. (2008). *Notaciones y Lenguajes de Procesos.* Madrid.
- Koch, N. (2005). *UML-Based Web Engineering.*
- Moreno, D. (2005). Extenciones de Lenguaje de Workflow para la Generacion dinámica de Vistas . Madrid, Madrid, España.
- Palacio, J. (2007). *Flexibilidad con Scrum.* Mexico: safeCreative.
- Palacio, J. (2014). *Gestión de Proyectos Scrum Manager.* Mexico: safeCreative.
- Pereira, B. (30 de enero de 2015). Métricas de calidad Software. La Paz, La Paz, Bolivia.
- Presman, R. (2006). *Ingenieria del software.*

- Raúl Herranz, N. M. (2011). *Scrum Distribuido*.
- Rondo Montes, E. P. (2008). *Workflow Aplicado a Sistema de Comercializacion del Soat*. La Paz - Bolivia.
- Sanderson, D. (2012). *Programming Google App Engine Edición 2* . EEUU: O'REILLY.
- Sectorial, M. (2010). *Cloud Computing Una Perspectiva Para Colombia*.
- Shift, O. (20 de Enero de 2015). *open Shift*. Recuperado el 20 de Enero de 2015, de open Shift: www.openshift.com
- Sierra, A. F. (2010). *Un prototipo en Google App Engine* .
- TIC, M. d. (2013). *El ABC del Teletrabajo en Colombia*. Bógota: Colombia Digital.
- Tola, J. L. (2013). *Sistema Web Usando Agentes Inteligentes*.
- Workflow Management Coalition*. (1995).





GLOSARIO

GLOSARIO

C

Cloud Computing.

Concepto dado a una nueva forma de implementación de servicios a través de internet, estos servicios se enfocan en la virtualización de equipos físicos lo que significa que se crean equipos físicos de forma simulada en un solo equipo físico, lo que deriva en tres modelos de prestación de los servicios: Infraestructura como servicio, donde el proveedor suministra los medios para la virtualización de equipos como computadoras, servidores, tarjetas, procesadores, entre otros. Plataforma como servicio, donde el proveedor brinda una plataforma de desarrollo de aplicaciones. Software como servicio donde el proveedor pone a disposición una aplicación.

D

Dinámicas de Grupo.

Las dinámicas grupales intentan explicar los cambios internos que se producen al interior de un equipo y como los integrantes reaccionan a estos cambios, logrando así definir y delimitar de forma científica los fenómenos grupales. Por tanto una persona encargada de moderar las dinámicas de grupos tiene por trabajo hacer que los cambios e interacción al interior de un equipo tengan un buen desarrollo en beneficio de un ambiente de trabajo.

E

Equipos Multidisciplinarios.

Un equipo multidisciplinario es aquel donde sus integrantes no tienen tareas ni deberes fijos, siendo sus obligaciones colaborar con lo que se necesite y por tanto cada miembro se encuentra preparado de manera adecuada a responder a la mayoría de las tareas que se llevan a cabo en la institución.

G

Google App Engine.

Plataforma de desarrollo de aplicaciones web alojada en la nube, con una versión gratuita para los desarrolladores que desean realizar una prueba. Los lenguajes de desarrollo soportados por la plataforma son: PHP, PYTON, JAVA, y GO.

Globalización.

Proceso económico, tecnológico, social y cultural que consiste en la interdependencia entre los países debido al capitalismo democrático y que han abierto sus puertas a la revolución informática.

GroupWare.

Groupware hace referencia a un conjunto de programas informáticos que integran el trabajo en un solo proyecto con muchos usuarios concurrentes que se encuentran en diversas estaciones de trabajo conectadas a través de una red.

K

Kit de Desarrollo de Software.

Se abrevia como SDK, y es un grupo de herramientas informáticas destinadas a trabajar en grupo para la consecución de un proyecto informático, de tal manera que el desarrollador solo se ocupe de la lógica de desarrollo.

L

Lenguaje Unificado de Modelado.

Se abrevia como UML, y es una notación estándar para el modelado de aplicaciones web. Aporta diagramas para la visualización, construcción, y documentación de distintos artefactos del sistema. Su versión actual es 2.0

N

Notación para el Modelado de Procesos.

Se abrevia como BPMN por sus siglas en inglés, y sirve para modelar, estudiar y especificar los procesos al interior de una organización, esta notación es un estándar de modelado reconocido en la actualidad.

P

Países del Primer Mundo.

Denominación o clasificación social surgida durante la segunda guerra mundial, a los países alineados con los países aliados se los denomina países del primer mundo y se caracterizan el día de hoy por su acceso equitativo a las riquezas y nivel de vida de sus ciudadanos en un nivel aceptable. Los países que optaron por los alinearse con los países socialistas se los denominó del segundo mundo que el día de hoy se los considera de primer mundo con la culminación de la guerra fría y caída del muro de Berlín, los países que no tomaron parte por ninguna opción se consideran de tercer mundo y se caracterizan por su subdesarrollo y pobreza.

Proceso Unificado Ágil.

Se abrevia como AUP, es una metodología ágil e iterativa centrada en los casos de uso para el diseño. Surge en el año 2005 como una refinación del proceso unificado “UP”, que es una metodología prescriptiva basada en el modelo en cascada lo que significa que define una serie de disciplinas que se realizan en ciclos hasta acabar el producto. AUP define cuatro fases: inicio, elaboración, construcción, y transición.

Procesos de un Negocio.

Los procesos de un negocio son las actividades o tareas realizadas de manera conjunta con un objetivo en común. Los procesos del negocio planificados y administrados de manera correcta son una de las prácticas de empresas exitosas y son el nuevo campo de explotación del área informática como ejemplo están los sistemas BPM “Management Process Modeling”

R

Rol.

Es el papel que juega un actor en un determinado escenario, estando definidas las acciones de cada rol de una manera formal.

S

Scrum.

Metodología de desarrollo software perteneciente al grupo de metodologías agiles que se caracterizan por adaptables a distintos escenarios, tener un ciclo de vida iterativo, y responder a los modernos escenarios de desarrollo caracterizados por los tiempos cortos asignados a la planeación.

Scrum Distribuido.

Es una investigación llevada a cabo por investigadores de la metodología scrum, publicada en el sitio www.scrummaster.com y tiene como característica las consideraciones que se deben tomar en cuenta en un escenario de desarrollo donde los miembros del proyecto software se encuentran distribuidos geográficamente.

T

Teletrabajo.

Es una modalidad de trabajo en la que los trabajadores pueden realizar sus actividades des lugares distintos al recinto de trabajo de la empresa, para lo que se vale de herramientas de comunicación como internet y tecnologías de comunicación en la misma como herramientas colaborativas, de asistencia, teleconferencia, entre otras.



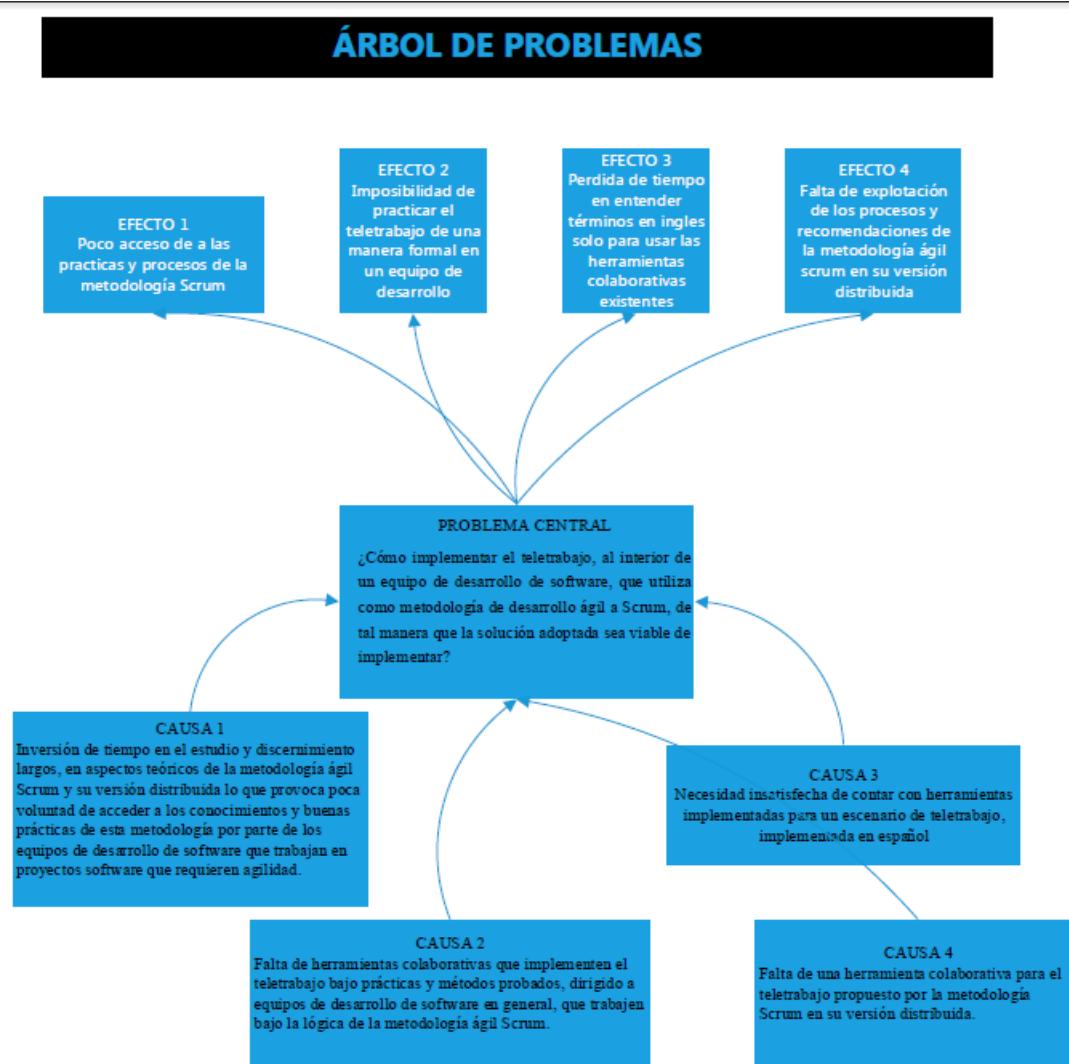


ANEXOS

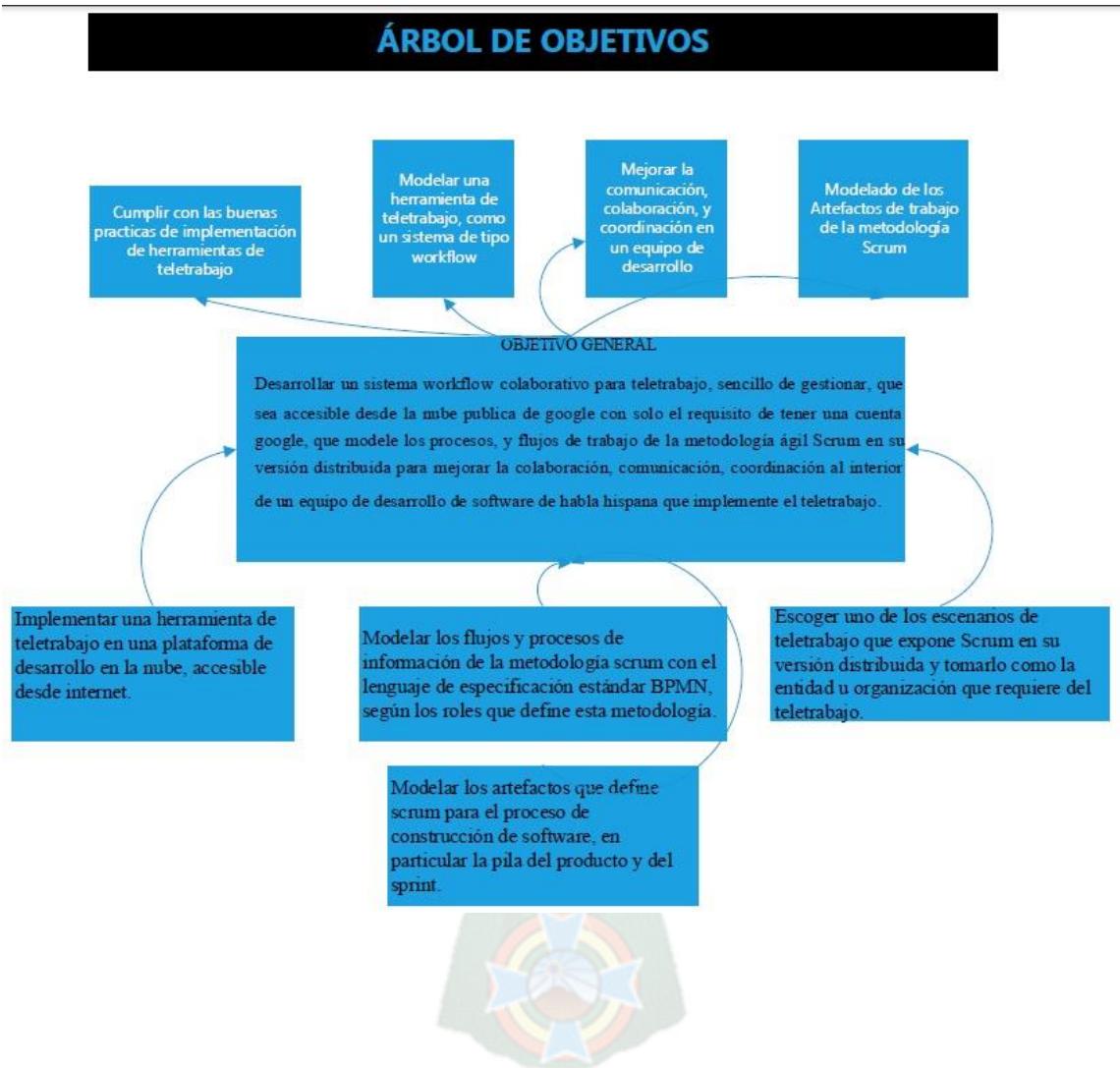
ANEXO-A MARCO LÓGICO

	RESUMEN NARRATIVO	INDICADORES OBJETIVAMENTE VERIFICABLES	MEDIOS DE VERIFICACION	SUPUESTOS
FIN	Contribuir al conocimiento y desarrollo de sistemas de teletrabajo como software de servicio	Evaluación del sistema workflow colaborativo en la nube	Pruebas e informes realizados	
PROPOSITO	Desarrollar un prototipo sistema workflow colaborativo en la nube para desarrolladores de software, como equipo distribuido de Teletrabajo	Prototipo de workflow colaborativo	* Presentación y aprobación de la documentación	Existencia de sistemas colaborativos en el mercado para la Version oficial de Scrum
PRODUCTOS	* Sistema Workflow para Scrum Distribuido * Implementacion en Google App Engine	* Nivel de Abstraccion respecto de la Version teorica de Scrum Distribuido	* Documentación del trabajo de investigación	* Existe un sitio oficial sobre Scrum Distribuido * Existe Programas colaborativos para Scrum pero no en Workflow
ACTIVIDADES O INSUMOS	* Investigar sobre las plataformas de desarrollo en cloud computing * Identificar los flujos de trabajo y los procesos de la metodología Scrum distribuido * Investigar el modelado de procesos y flujos de trabajo mediante sistemas Workflow * Investigar conceptos de Teletrabajo bajo Scrum Distribuido	* Lectura de libros de, internet y articulos de Cloud Computing sobre Google App Engine * Identificación de los flujos de trabajo y procesos de Scrum distribuido * Estudio del lenguaje de desarrollo bajo la plataforma Google App Engine	* Documento final del trabajo * Documentos sobre datos estadisticos del Teletrabajo	* Disposición de bibliografía necesaria * Disponibilidad de internet * Disponibilidad de información en internet

ANEXO-B ÁRBOL DE PROBLEMAS



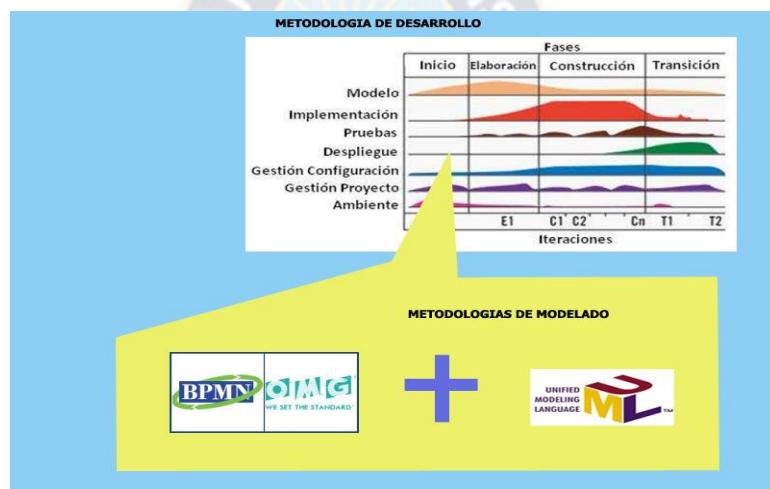
ANEXO-C ÁRBOL DE OBJETIVOS



ANEXO-D METODOLOGÍA DE DESARROLLO

PROCESO UNIFICADO ÁGIL “AUP”

El marco del desarrollo del software debe estar guiado por una metodología de ingeniería del software, en este trabajo se utiliza una metodología ágil denominada “AUP”²⁰, que define cuatro fases del ciclo de desarrollo: inicio, elaboración, construcción y transición. Sin embargo para la fase de análisis se necesita el modelado y especificación de requisitos del sistema, para lo que la metodología AUP sugiere los casos de uso y los modelos con los que cuenta UML²¹. En el trabajo para la fase de inicio que representa el análisis y diseño se hace uso de los casos de uso, sin embargo como paso previo a esto se realiza el modelado de los procesos de la metodología scrum mediante el estándar BPMN, lo que se encontró valido puesto que los diagramas de BPMN²² tienen la misma función de modelado que el diagrama de secuencia del estándar UML, es más en la actualidad los estándares BPMN y UML están a cargo de la misma organización que es la Object Management Group “OMG”²³ y define a ambos estándares como de modelado de procesos.



A continuación se explica de manera detallada las tareas realizadas en cada una de las fases de la metodología en el presente trabajo. Cabe resaltar que estas tareas se enmarcan en las que define la metodología por cada fase, para más referencia ver el capítulo 2, apartado 2.6 pagina 38.

1. LA FASE DE INICIO.

- Definición del alcance del Proyecto: Se ha definido que el alcance del prototipo será para un escenario de teletrabajo parcial, donde los miembros del equipo tienen la posibilidad de encontrarse de manera personal.
- Definición de riesgos: se realiza la definición de riesgos a través del modelado de los procesos con el lenguaje de especificación BPMN y casos de uso de alto nivel, visualizando así que se debe implementar primero y evitando así un desorden que crearía retrasos.

²⁰ AUP por sus siglas en inglés que significan Processes Unit Ágil

²¹ UML por sus siglas en inglés que significan Unified Modeling Language

²² BPMN por sus siglas en inglés que significan Notation for Modeling Business Processes

²³ OMG por sus siglas en inglés que significan Object Management Group.

- c. Determinación de la factibilidad del proyecto: Esta tarea se traduce en ver los procesos que son posibles modelar y sus flujos, así por ejemplo se vio que cumplir todos los flujos de manera estricta por razones de usabilidad ya que cumplirlos de manera estricta haría que la herramienta fuera muy lenta para utilizar.
 - d. Preparar el entorno de Trabajo: En esta tarea de la fase de inicio, se ha hecho pruebas de compatibilidad entre Kit de desarrollo para GAE y el JDK java necesario, dejando así listas las herramientas para el desarrollo.
2. LA FASE DE ELABORACIÓN.
- a. Definición de la Arquitectura del Sistema: En el trabajo esta tarea se ha llevado a cabo con el diseño del diagrama de clases, que son la base para la construcción de las entidades de almacenamiento en el datastore y por tanto son la base para la construcción de las distintas interfaces del sistema colaborativo para teletrabajo.
3. LA FASE DE CONSTRUCCIÓN.
- a. Construcción del Sistema por Iteraciones: En el trabajo se ha practicado los conceptos que explica la metodología AUP, en concreto en esta tarea para la fase de construcción se ha construido el sistema en 5 iteraciones, concentrándose así en conseguir el producto funcional, y teniendo en cuenta que cada iteración es una paso y por tanto evitar el estrés de conseguir todo el producto en una sola iteración.
 - b. Primera Iteración: Se construyó la parte administrativa del sistema, es decir la entrada con la definición de usuarios que puedan acceder a la aplicación, que en este trabajo son los usuarios que tengan una cuenta google.
 - c. Segunda Iteración: Para la segunda iteración se define las entidades y se las implementa para el respectivo almacenamiento de datos. Además se crea una primera versión de las interfaces para trabajar con las entidades creadas.
 - d. Tercera Iteración: Se llegó a la conclusión de la imposibilidad de la implementación de una sala de chat con los servicios que provee la plataforma GAE, ya que este servicio cayó en desuso en el año 2013 con la salida del servicio Google Talk. Además se empieza con la búsqueda de una nueva opción para que la herramienta incluya un servicio de comunicación entre los usuarios.
 - e. Cuarta Iteración: Se decide la incorporación de servicio IRC libre, denominado Freenode, el que en su sitio oficial da la opción de copiar código para empotrar en las aplicaciones web la sala de chat correspondiente. La interfaz para la sala de chat se la implementa de tal manera que sea opcional además de poder definir la fecha de charla y los usuarios que participaran en dicha conversación.
 - f. Quinta Iteración: Se realiza pruebas con las interfaces para los artefactos pila del producto y del sprint, encontrando errores de lógica y se realiza las correcciones correspondientes.

- g. Sexta Iteración: Se implementa una entidad para poder realizar el incremento de los sprint, considerando solamente el intercambio de archivos, y dejando pendiente al equipo de trabajo la coordinación de unir sus partes en las reuniones cara a cara.

4. LA FASE DE TRANSICIÓN

- a. Pruebas Beta: Las pruebas beta se realizaron una semana antes de la prueba piloto con los estudiantes de Informática, mejorando la accesibilidad en cuanto a diseño, por recomendaciones de los coordinadores y colaboradores del trabajo que en el caso de una Tesis son tanto el Revisor y Tutor de la misma.
- b. Pruebas en el Entorno de Trabajo: Las pruebas piloto realizadas con los estudiante de Informática, son las que se clasifican como las pruebas en el entorno de producción, y más aún la herramienta incorpora un soporte para los que decidieron usar la herramienta de forma permanente, mediante una interface que permite la consulta y observaciones entre los usuarios y el administrador del sistema colaborativo.



ANEXO-E METODOLOGÍA DE PRUEBA DE HIPÓTESIS

INVESTIGACIÓN CUALITATIVA

La investigación cualitativa se utiliza en mayor grado en investigaciones del área social, debido a que los estudios se centran en las percepciones subjetivas de los sujetos de estudio. En el presente trabajo se la toma como metodología de prueba de la hipótesis del trabajo, debido a que la naturaleza de la hipótesis involucra la percepción subjetiva de los usuarios, en concreto la percepción de los mismos hacia la mejora de la comunicación, colaboración y coordinación gracias al uso de la herramienta colaborativa para teletrabajo objeto del presente trabajo. Sin embargo dado que el área de investigación es de ciencias exactas, se necesita adicionalmente el procesamiento de datos cuantificables, por lo que para sustentar la percepción de los usuarios de manera cuantificable se elaboró una encuesta la que se procesó con métodos estadísticos en concreto la distribución Chi-cuadrado, definiendo así un éxito y fracaso para la investigación dando la posibilidad de afirmar el cumplimiento de la hipótesis con la distribución mencionada.

DISTRIBUCIÓN ESTADÍSTICA CHI-CUADRADO

La distribución Chi-cuadrado es una distribución de probabilidad continua con parámetro de grados de libertad k . Esta distribución se utiliza cuando se quiere probar una hipótesis dado un grado de confiabilidad por el investigador entre 1% y 100%. La confiabilidad significa que tan confiable es la investigación en una escala numérica en porcentajes. La prueba de hipótesis mediante esta distribución, se resume en los siguientes pasos:

- ✓ Escoger la muestra, en un número menor a 30 sujetos y mayor a 15
- ✓ Observar las frecuencias reales de éxito de investigación
- ✓ Cálculo del valor Chi-cuadrado, mediante la fórmula:

$$\chi^2 = \frac{\sum (f_o - f_e)^2}{f_e}$$

- ✓ Decisión sobre la verdad o falsedad de la hipótesis en base a la comparación del valor calculado Chi-cuadrado y el mismo correspondiente para los grados de libertad en la tabla Chi-cuadrado.

ANEXO-F ENCUESTA UTILIZADA PARA LA PRUEBA PILOTO

EVALUACIÓN DE LA CALIDAD PARA WORKFLOW COLABORATIVO BASADO EN SCRUM DISTRIBUIDO PARA TELETRABAJO IMPLEMENTADO EN CLOUD COMPUTING.

Sexo: M: _____ F: _____

Edad: _____

PARA RESPONDER LAS SIGUIENTES PREGUNTAS POR FAVOR MARQUE CON UNA "X" LA OPCIÓN QUE CORRESPONDA

1. Funcionalidad: A través de los ítems siguientes y de los indicadores asociados tratamos de valorar si el programa se adecúa a las características, condiciones y necesidades funcionales para los procesos y flujos de información de la metodología Scrum y en qué medida lo hace.

	Alto	Medio	Bajo
1.1 El programa está adecuado equipos de desarrollo y personas interesadas			
1.2 El programa cumple con las metas generales de análisis y mejora la calidad de la fase de análisis de Scrum en un ambiente distribuido			
1.3 El programa promueve la colaboración entre grupos en el análisis de Scrum, en ambientes en que los miembros no pueden encontrarse FISICAMENTE			

2. Confabilidad: En este apartado evaluaremos si el programa, en su diseño, toma en cuenta los roles en la fase de análisis de Scrum, lo que se traduce en una confiabilidad hacia el prototipo en cuanto a su objetivo central que es el de teletrabajar bajo la metodología scrum mediante la presente herramienta colaborativa.

	Alto	Medio	Bajo
2.1 El diseño del programa identifica adecuadamente los perfiles y las características del participante.			
2.2 El diseño especifica y detalla adecuadamente rol de los diferentes participantes.			

3. Eficiencia: Se quiere evaluar la eficiencia de la herramienta para los que se verá si la plataforma tecnológica (prototipo) dispone de los requerimientos técnicos necesarios para hacer viable las metas y objetivos del análisis de Scrum en un ambiente distribuido geográficamente entre los miembros.

	Alto	Medio	Bajo
3.1 El acceso del participante a la plataforma tecnológica es fácil de entender			
3.2 La plataforma tecnológica permite el acceso en diferentes horarios sin poner restricciones.			
3.3 La plataforma tecnológica pone a disposición del participante las herramientas estándares de comunicación en ese momento, tanto síncronas como asíncronas.			
3.4 La plataforma tecnológica pone a disposición del participante herramientas de gestión y seguimiento de su propio progreso.			
3.5 La plataforma tecnológica dispone de recursos para la creación de un repositorio que permita compartir documentos y materiales elaborados por el participante.			
3.6 La plataforma tecnológica pone a disposición del participante herramientas de trabajo en grupo y de colaboración.			
3.7 La plataforma tecnológica incorpora diferentes herramientas de apoyo de tareas individuales y grupales			
3.8 La plataforma tecnológica permite el establecimiento de contactos online entre los participantes			
3.9 La plataforma tecnológica permite el acceso abierto a los recursos internos (elaborados por los participantes), que se utilicen en el programa			

4. Accesibilidad: Evaluaremos si el programa es sensible a la accesibilidad en su diseño.

	Alto	Medio	Bajo
4.1 En las tablas, los datos se presentan siguiendo una estructura lógica y ordenada.			
4.2 Se proporciona adecuada información de contexto al participante			
4.3 Todas las partes del programa pueden ser identificadas a través de su título.			
4.4 Se proporciona información al participante sobre los procesos de desarrollo con scrum			
4.5 El acceso al ambiente se puede realizar desde cualquier navegador: mozilla firefox, chrome.			

5. Usabilidad: En este apartado veremos si la gestión del programa asegura el adecuado funcionamiento y coordinación de los elementos que hacen posible el cumplimiento de los objetivos de análisis de desarrollo de software con Scrum.

	Alto	Medio	Bajo
5.1 Permite la definición de las funciones y responsabilidades de cada una de las personas implicadas.			
5.2 Se establecen canales de comunicación estables entre las personas encargadas de la implementación del software.			

5.3 Los errores de acceso a páginas o recursos del programa se corrigen con diligencia.			
5.4 Los responsables de la gestión administrativa realizan el registro del seguimiento del participante.			

6. Mantenibilidad: Esta propiedad del modelo de evaluación de calidad (Dromey) trata de responder si el prototipo considera la mantenibilidad después de su entrega, para lo que se verá si el programa promueve el asesoramiento continuado, así como si crea espacios virtuales de comunicación e intercambio entre el participante y el administrador del sistema

	Alto	Medio	Bajo
6.1 El programa proporciona al participante acceso permanente para apoyar o guiar la implantación de los contenidos del software cuando éste ha finalizado.			
6.2 El programa de la opción de compartir recursos, promoviendo así una colaboración grupal			
6.3 Se ofrece al participante un servicio de información en relación con el contenido del programa.			
6.4 El programa da una solución a la forma de trabajar obligatoriamente en un lugar físico con la presencia de todos los miembros del equipo			
6.5 El programa ayuda a entender mejor los roles dentro de scrum			
6.6 La herramienta es una forma didáctica para aprender a usar scrum, dando la facilidad de realizarlo de manera separada al lugar de trabajo			
6.7 El programa permite definir de manera correcta la misión y visión del software a desarrollar			

