

Universidad de las Ciencias Informáticas

Facultad #5



**Propuesta de guía de desarrollo de software con
ajuste a Sistemas de Realidad Virtual**

**Trabajo de Diploma para optar por el título de
Ingeniero en ciencias Informáticas**

Autores:

Yanisleis Pérez Díaz

Grenny Negret Rodriguez

Tutor:

Ing. Yaima Fiallo Valle

Ciudad de la Habana

Julio 2009

MERVI



REALIDAD VIRTUAL

Autores.

Nombre: Yanisleis Pérez Díaz.

Correo electrónico: yanisleis@estudiantes.uci.cu

Nombre: Grenny Negret Rodriguez.

Correo electrónico: gnegret@estudiantes.uci.cu

Tutor.

Nombre: Yaima Fiallo Valle.

Correo electrónico: yfiallo@uci.cu

Consultante.

Nombre: Karel Pérez Ramírez.

Correo electrónico: kperez@uci.cu

AGRADECIMIENTOS

Agradezco desde lo más profundo de mi corazón a mis padres y hermana, por estar siempre a mi lado, brindarme su cariño y confiar siempre en mí. Le doy las gracias a mi compañero de tesis por ayudarme siempre que lo he necesitado y dar lo mejor de sí en este trabajo de diploma. Muchas gracias los tutores por brindarnos su ayuda, a Kenia por ayudarme tanto a lo largo de estos cinco años. Agradezco a mis abuelos porque sé que siempre me han ayudado y aun continúan asíéndolo, a mis tíos por brindarme su apoyo incondicional, en fin a todos los compañeros que han permanecido a mi lado en estos cinco años y que de alguna manera forman parte de lo que soy hoy.

Yanisleis

Agradezco a mi familia, por el apoyo y el amor que me han dado. A Manolito mi primo por ser mi amigo y mi hermano, le agradezco el haber cuidado de mi madre cuando he estado lejos. A mis tías maternas por sus consejos y el apoyo que me han brindado. A mis primas por el cariño que me han ofrecido, especialmente Yádira por haber estado presente siempre, ella ha sido y es, una hermanita para mí. También le agradezco a Madelyn (esposa de mi padre) por haberme apoyado tantos años. Gracias a Sonia (madre de Yanisleis), por su preocupación y por haberme ayudado cuando lo necesité. Le agradezco además, a mi grupo por soportarme estos 5 años de carrera. A mis compañeros de apartamento, por la paciencia que han tenido conmigo; pues solo ellos y yo sabemos cuántos tiempos de máquina utilicé para lograr esto. Les doy gracias a todos mis compañeros y amigos, que de alguna manera han contribuido al logro de este sueño hecho realidad. De igual manera a todos los instructores que han sido parte de mi formación, y a todas las personas que han aportado su granito de arena en la realización de este trabajo.

Por último, le quiero agradecer también, a los obstáculos que han pasado por mi vida. Ellos son los que me han hecho fuerte, perseverante y finalmente, triunfador.

Grenny

A mis padres y hermana que siempre me han brindaron su apoyo para que viera realizados mis sueños, quiero regalarles este momento y honrarlos por tanto amor y dedicación. Los quiero mucho.

Yanisleis

Quiero dedicar el logro de este trabajo y todo lo que esto representa, a las personas que son la esencia de lo que soy. A mi madre, quien es la expresión más sublime de mis sentimientos, que no se alcanza con la palabra AMOR en grado superlativo. A mi abuela Lidia, a quien adoro y que siempre está presente en mis pensamientos. A Pablo, por ser mi amigo, mi hermano y mi padre.

Para mi madre, mi abuela materna y mi papá, dedico este triunfo. El cual es el fruto y la garantía, de que eternamente estaré retribuyendo con amor; todo el cariño, el respeto, la confianza y el apoyo que me han ofrecido.

Grenny

Partiendo de la necesidad de la utilización de una guía que se ajuste al desarrollo de los Sistemas de Realidad virtual (SRV), en la Facultad 5 de la Universidad de las Ciencias Informáticas (UCI), se realiza esta propuesta, la cual constituye una solución a la problemática planteada. Para la elaboración de esta guía, fue preciso realizar un estudio detallado de las características, y el modo de producción de los proyectos del Polo de Realidad Virtual (PRV), así como el análisis de varias metodologías que han sido (o están siendo actualmente) utilizadas por el Polo para el desarrollo de sus productos, dentro de las que se encuentran OpenUP, XP, MSF, RUP, UPEDU, y Scrum. Además, se realizaron encuestas a los Jefes de proyectos, con el objetivo de conocer, y tener en cuenta las necesidades, e intereses de cada uno de los proyectos del PRV. También, se analizaron todos los expedientes de proyectos con el propósito de determinar los roles, y artefactos que no incluyen las metodologías estudiadas, y que son de gran utilidad en el Polo. La propuesta realizada en el presente trabajo, toma características de modelos, metodologías, y métodos de trabajo analizados. Como parte de la solución, se describe un conjunto de elementos capaces de orientar el desarrollo de aplicaciones de Realidad Virtual (RV), y que se adaptan al modo de producción del PRV. Finalmente, la propuesta es validada aplicando encuestas a especialistas en temas de Ingeniería de Software (ISW), y RV. Resultando evidente, la necesidad de poner en práctica esta guía en el desarrollo de aplicaciones de RV.

PALABRAS CLAVE

Guía de desarrollo de software, Sistemas de Realidad Virtual, Polo de Realidad Virtual

TABLA DE CONTENIDOS

AGRADECIMIENTOS.....	I
DEDICATORIA	II
RESUMEN	III
INTRODUCCIÓN	1
CAPÍTULO 1: MARCO TEÓRICO	4
INTRODUCCIÓN	4
1.1 VISIÓN GENERAL DE LA INGENIERÍA DE SOFTWARE.....	4
1.2 MODELOS DE PROCESO DE SOFTWARE.....	5
1.2.1 MODELO INCREMENTAL.....	5
1.2.2 MODELO ESPIRAL.....	6
1.2.3 MODELO DE DESARROLLO BASADO EN COMPONENTES.....	7
1.2.4 MODELO INTEGRADO DE MADUREZ Y CAPACIDAD (CMMI).....	8
1.3 ESTUDIO DE METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	9
1.3.1 RUP.....	9
1.3.2 UPEDU.....	16
1.3.3 OPENUP.....	17
1.3.4 PROGRAMACION EXTREMA (XP).....	19
1.3.5 MSF ÁGIL.....	23
1.3.6 SCRUM.....	27
1.4 ESTUDIO DEL DESARROLLO DE SISTEMAS DE REALIDAD VIRTUAL	30
1.4.1 SECUENCIA DE ACTIVIDADES PROPUESTAS POR JUAN C. PARRA M.....	31
CONCLUSIONES PARCIALES	33
CAPÍTULO 2: DISEÑO METODOLÓGICO.....	34
INTRODUCCIÓN	34
2.1 PECULIARIDADES DEL POLO DE REALIDAD VIRTUAL	34
2.2 ESTUDIO DEL POLO DE REALIDAD VIRTUAL	35
2.2.1 PROPUESTA DEL SISTEMA DE GESTIÓN DE LA CALIDAD	40
2.3 APLICACIÓN DE ENCUESTAS Y ENTREVISTAS	41
2.3.1 FORMULACIÓN DE OBJETIVOS.....	42
2.3.2 DISEÑO DE LA MUESTRA.....	43
2.3.3 CONFECCIÓN DEL CUESTIONARIO.....	44
2.3.4 RECOLECCIÓN DE LOS DATOS	45
2.3.5 ANÁLISIS Y PRESENTACIÓN DE LOS RESULTADOS	45
CONCLUSIONES PARCIALES	55
CAPÍTULO 3: PROPUESTA DE LA SOLUCIÓN	56
INTRODUCCIÓN	56
3.1. DESCRIPCIÓN DE LA PROPUESTA	56
3.1.1 PRINCIPIOS Y PRÁCTICAS	57
3.1.2 MODELO DE PROCESO.....	58
3.1.3 MODELO DE ROLES.....	62

3.1.4 DISCIPLINAS	70
3.1.4.1 FLUJOS DE PROTECCIÓN.....	70
3.1.4.2 FLUJOS DE TRABAJO.....	73
3.1.5 FASES.....	75
3.1.6 ARTEFACTOS	79
3.2 APORTES	85
3.3 VALIDACIÓN	88
3.3.1 MÉTODO.....	88
3.3.2 EVALUACIÓN.....	90
CONCLUSIONES PARCIALES	91
CONCLUSIONES	92
RECOMENDACIONES.....	93
REFERENCIAS BIBLIOGRÁFICAS	94
BIBLIOGRAFÍA.....	96
WEBGRAFÍA.....	98
GLOSARIO DE TÉRMINOS.....	99
GLOSARIO DE ABREVIATURAS	101
ANEXOS	103

El avance de la Informática en los últimos tiempos ha dado lugar a nuevas tecnologías cada vez más complejas, donde los Sistemas de Realidad Virtual (SRV) han ocupado un espacio importante dentro de este desarrollo. Esta rama de la Informática fue concebida primeramente con objetivos militares, investigativos y académicos, hoy en día su aplicación se ha ampliado hacia el campo profesional, doméstico e industrial.

El desarrollo de la RV presenta una fuerte dependencia del hardware, que unido a su componente software es perfeccionado constantemente, lo que hace que las aplicaciones de la RV en el futuro sean aún inimaginables.

A través de juegos electrónicos o viajes virtuales se puede disfrutar de sensaciones de la realidad. Además, los accidentes y las pérdidas de recursos que se pueden evitar en un ambiente aparentemente real son innumerables. Esto puede aplicarse en el entrenamiento aeronáutico con instalaciones virtuales, en la preparación de un médico antes de enfrentarse a una operación de alto riesgo anatómico, y en la simulación de proyectos por citar algunos ejemplos.

Cuba no está ajena a todas estas transformaciones que produce la ciencia. Con el interés de contribuir al desarrollo social y económico, se crea la UCI; la misma, con la idea de construir una institución educacional y productiva, ha ido fortaleciendo la organización en Polos Productivos (PP) que tributan a la integración de los procesos de formación, investigación – desarrollo, producción - comercialización.

El impacto de los resultados de los proyectos vinculados a la RV ha merecido que se hayan incluidos en los niveles de incubación de los Polos Productivos dentro de la UCI. De esta manera surge el Polo Productivo de RV, que desarrolla sus actividades fundamentales en la Facultad 5 de nuestra institución. Este polo garantiza un elevado nivel científico y categoría de sus miembros a través de las acciones que desarrolla el Grupo de Investigación de RV (GIRV). En este marco se han obtenido resultados aplicables a productos terminados.

En este proceso la Facultad 5 de la UCI ha concebido la necesidad del uso de una guía ajustada al desarrollo de SRV, debido a que las metodologías utilizadas en el proceso de desarrollo de aplicaciones

en el PRV, no se adaptan a las características de los posibles equipos de trabajo, y roles que se requiere para desarrollar un SRV en la Universidad. Con la utilización de algunas metodologías se generan demasiados artefactos, lo que provoca pérdida de tiempo en el desarrollo del producto. El presente trabajo surge a partir de que muchas de las aplicaciones que usan técnicas de RV, generalmente requieren de largos tiempos de construcción, por lo que se hace necesaria la utilización de alguna guía de desarrollo de software que mejore el trabajo de los proyectos productivos en el PRV. Dicha guía incluye el ajuste de plantillas, roles, artefactos, y disciplinas con el fin de disminuir estos tiempos de desarrollo, ser competitivos, rentables; disminuir costos, pérdidas de clientes y recursos.

Algunos proyectos en el PRV trabajan de forma empírica, sin utilizar una guía que oriente el desarrollo de SRV; por lo que el trabajo no cuenta con la organización requerida para elevar la calidad de las aplicaciones y la productividad. En la actualidad toda industria productora de software, necesita de una guía para el proceso de desarrollo, debido a su complejidad y magnitud. Está demostrado que la empresa que ponga en práctica todos los elementos necesarios para mejorar el proceso de desarrollo de software tendrá una mayor ventaja competitiva frente a las demás. El Modelo de Madurez de la Capacidad (CMM)¹ del Software (SW) también nos indica que los mejores informáticos necesitan un entorno disciplinado y estructurado en el cual puedan realizar un trabajo en equipo, para lograr productos con alta calidad.

Basado en lo antes expuesto, el presente trabajo va encaminado a solucionar el **problema científico**: ¿Cómo elaborar una guía de desarrollo de SW con ajuste a SRV?

Por lo que el **objeto de estudio** son las metodologías de desarrollo de SW existentes.

Luego, el **objetivo general** que se persigue en este estudio radica en proponer una guía para el desarrollo de aplicaciones, según las necesidades y los intereses de trabajo del PRV.

Por tanto, el **campo de acción** de la investigación estará dirigido fundamentalmente a las metodologías de desarrollo de SW aplicables a los SRV.

Continuamente, para la realización de la investigación se han identificado las siguientes **tareas investigativas**:

¹ Conjunto de modelos que permiten obtener un diagnóstico preciso de la madurez de los procesos relacionados con las

- Identificación de los modelos existentes, para ajustarlos al desarrollo de los SRV.
- Identificación de las metodologías y procesos existentes, para ajustarlas al desarrollo de los SRV.
- Búsqueda de modos de producción de los SRV, tanto en la facultad 5 como a nivel nacional e internacional; para identificar necesidades e intereses del PRV.
- Identificación de disciplinas, roles, y artefactos necesarios para el desarrollo de aplicaciones de RV.
- Análisis del postmortem de desarrollo de SRV, para realizar mejoras al proceso de desarrollo de SW.

Con el presente estudio se aspira obtener como **posible resultado** una guía de desarrollo de SW con ajuste a SRV.

Para la elaboración del presente trabajo los investigadores harán uso de varios **métodos científicos** con el objetivo de lograr una mayor organización y profundidad en la investigación. Para abordar el problema científico que aquí se expone, construir y desarrollar la teoría científica; se trabajará a partir de Métodos Teóricos, principalmente el Analítico-Sintético, la Modelación y la Inducción-Deducción. También se utilizarán Métodos Empíricos como la observación, y las diferentes técnicas de recolección de información (encuestas, entrevistas, revisión de documentación, etc.); con esto se pretende lograr una caracterización del fenómeno a investigar mediante la obtención de datos y conocimientos sobre los hechos fundamentales.

INTRODUCCIÓN

La RV definitivamente es una tecnología que ha causado gran impacto en el mundo actual, y aunque su desarrollo es aún innovador, ha surtido grandes efectos en la adaptación de mucha gente a los cambios constantes que la ciencia de la Informática impone. La RV permite percibir sensaciones a través de los medios electrónicos; pues, mediante un ordenador las personas se pueden situar en un ambiente casi real e interactuar con lo que les rodea. En el presente capítulo se hará un estudio de los antecedentes, donde se identificarán los precedentes fundamentales para ampliar el trabajo. Además, se analizará un estado del arte para exponer los elementos que fundamentan la investigación. Se ofrecerá un panorama general de la ISW, y se hará referencia a algunos modelos, y metodologías de SW conocidos en la actualidad. También serán analizadas las características de los SRV, identificando algunos métodos de trabajos utilizados en el desarrollo de aplicaciones de RV.

1.1 VISIÓN GENERAL DE LA INGENIERÍA DE SOFTWARE

La ISW no es más que la realización del análisis, diseño, construcción, verificación, y gestión de una entidad informática. El trabajo asociado a estas actividades, se puede dividir en 3 fases o etapas genéricas independientemente del área de aplicación, tamaño o complejidad del proyecto. Estas fases son:

Definición: Se centra sobre el qué. Es decir, durante la definición, el que desarrolla el software intenta identificar qué información ha de ser procesada, qué función y rendimiento se desea, qué comportamiento del sistema, qué interfaces van a ser establecidas, qué restricciones de diseño existen, y qué criterios de validación se necesitan para definir un sistema correcto. Por tanto, han de identificarse los requisitos claves del sistema y del software. Aunque los métodos aplicados durante la fase de definición variarán dependiendo del paradigma de ingeniería del software (o combinación de paradigmas) que se aplique, de alguna manera tendrán lugar tres tareas principales: ingeniería de sistemas o de información, planificación del proyecto de software y análisis de los requisitos. [Pressman, 2001]

Desarrollo: Se centra en el cómo. Es decir, durante el desarrollo un ingeniero del software intenta definir cómo han de diseñarse las estructuras de datos, como ha de implementarse la función dentro de una

arquitectura de software, cómo han de implementarse los detalles procedimentales, como han de caracterizarse interfaces, como ha de traducirse el diseño en un lenguaje de programación (o lenguaje no procedimental) y cómo ha de realizarse la prueba. Los métodos aplicados durante la fase de desarrollo variarán, aunque las tres tareas específicas técnicas deberían ocurrir siempre: diseño del software, generación de código y prueba del software. [Pressman, 2001]

Mantenimiento: Se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software y a cambios debidos a las mejoras producidas por los requisitos cambiantes del cliente. [Pressman, 2001]

1.2 MODELOS DE PROCESO DE SOFTWARE

Un modelo o paradigma de proceso de SW es una estrategia de desarrollo que utiliza una persona u organización productora de SW para resolver un problema real. Esta estrategia acompaña al proceso, a los métodos, y a las herramientas de la ISW para representar de manera simplificada al proceso real. Por tanto, un modelo de procesos de SW es una abstracción que puede ser utilizada para explicar diferentes enfoques del desarrollo de SW. Por la aplicación que tienen en el PRV, algunos de los modelos que existen se describen a continuación:

1.2.1 MODELO INCREMENTAL

El modelo Incremental combina elementos de desarrollo lineal secuencial (aplicados repetidamente) con la filosofía interactiva de construcción de prototipos. El modelo Incremental aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento del software. [MDE, 93] La figura siguiente muestra la interacción entre las fases de este proceso.

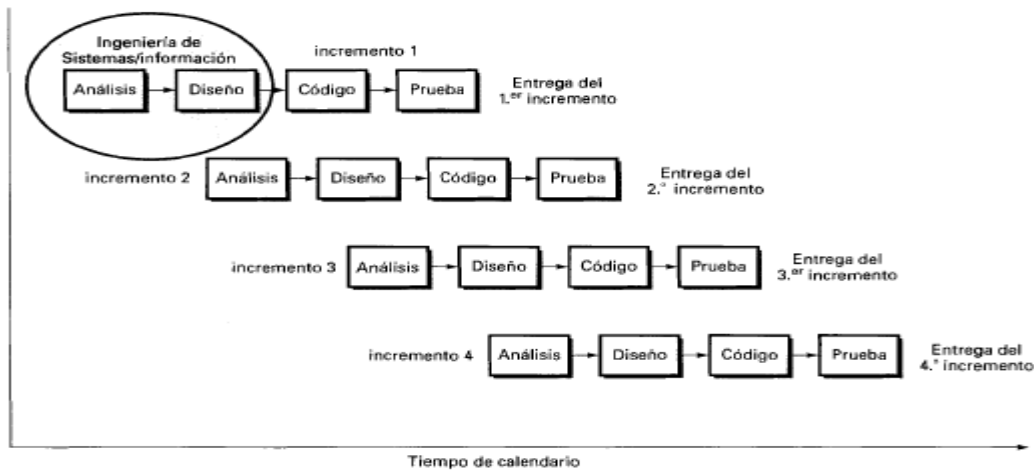


Figura 1: Modelo incremental. [Pressman, 2001]

1.2.2 MODELO ESPIRAL

Este modelo toma en consideración explícitamente el riesgo, esta es una actividad importante en la administración del proyecto.

Es un modelo de proceso evolutivo que conjuga la naturaleza iterativa de construcción de prototipos con los aspectos controlados y sistemáticos del modelo Lineal Secuencial. Proporciona el potencial para el desarrollo rápido de versiones incrementales del software. En el modelo Espiral, el software se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas del sistema diseñado. [Pressman, 2001]

Cada ciclo de desarrollo de este modelo se divide en seis fases (Ver Figura 2):

1. Comunicación con el cliente: Se establece la comunicación entre el desarrollador y el cliente.
2. Planificación: Se determina si continuar con otro ciclo. Se planea la siguiente fase del proyecto.
3. Análisis de riesgos: Se realiza un análisis detallado de cada riesgo identificado. Pueden desarrollarse prototipos para disminuir el riesgo de requisitos dudosos. Se llevan a cabo los pasos para reducir los riesgos.
4. Ingeniería: Se construye una o más representaciones de la aplicación.

5. Construcción y acción: Se realiza la construcción, prueba, e instalación del SW. Se proporciona soporte al usuario.
6. Evaluación del cliente: el cliente evalúa las representaciones e implementación del software, y se obtiene una reacción de él.



Figura 2: Modelo espiral. [Pressman, 2001]

1.2.3 MODELO DE DESARROLLO BASADO EN COMPONENTES

El modelo Desarrollo Basado en Componentes (DBC) incorpora muchas de las características del modelo Espiral. Exige un enfoque evolutivo e iterativo para la creación del software. Sin embargo, el modelo DBC configura aplicaciones desde componentes (clases) preparados de software. [Pressman, 2001]

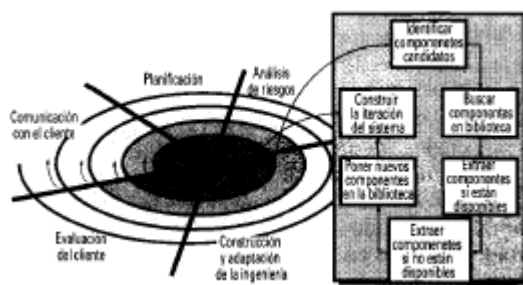


Figura 3: Desarrollo basado en componentes. [Pressman, 2001]

La actividad de la ingeniería comienza con la identificación de clases candidatas. Las clases creadas en los proyectos de ISW anteriores, se almacenan en una biblioteca de clases o diccionario de datos (repositorio). Una vez identificadas las clases candidatas, la biblioteca de clases se examina para determinar si estas clases ya existen. En caso de que así fuera, se extraen de la biblioteca y se vuelven a utilizar. Si una clase candidata no reside en la biblioteca, se aplica el análisis, diseño y prueba orientados a objetos. Se compone así la primera iteración de la aplicación a construirse, mediante las clases

extraídas de la biblioteca y las clases nuevas construidas para cumplir las necesidades únicas de la aplicación. El flujo del proceso vuelve a la espiral y volverá a introducir por último la iteración ensambladora de componentes a través de la actividad de ingeniería.

1.2.4 MODELO INTEGRADO DE MADUREZ Y CAPACIDAD (CMMI) [CMMI, 2008]

El Modelo Integrado de Madurez y Capacidad (CMMI) es:

- Un marco de referencia para el crecimiento de capacidades y madurez. Enfocado tanto en Procesos de Administración, como en los de Ingeniería de Sistemas y Software.
- Un modelo evolutivo para medir/evaluar la madurez y capacidades de la organización. Enfocado en la mejora continua a través de una cultura de procesos.

Este modelo tiene dos representaciones:

- **Representación continua** (Una categoría para una sola área o conjunto de áreas procesos).

Categoría	Áreas de proceso
Administración de proceso.	Enfoque Organizativo de Proceso. Definición Organizativa de Proceso +IPPD. Entrenamiento Organizativo. Función Organizativa de Proceso. Implementación e Innovación Organizativa.
Administración de proyecto.	Planificación de Proyecto. Monitoreo y Control de Proyecto. Administración de Acuerdos del Proveedor. Administración de Proyecto Integrado +IPPD. Administración de Riesgo. Administración Cuantitativa de Proyecto.
Ingeniería.	Administración de Requisitos. Desarrollo de Requisitos. Solución Técnica. Integración del Producto. Verificación. Validación.
Soporte.	Administración de Configuración. Aseguramiento de la Calidad del Proceso y el Producto. Análisis y Medición. Análisis y Resolución de la Decisión. Análisis y Resolución Causal.

- **Representación por etapas** (Un nivel de madurez para un conjunto definido de áreas de proceso de la organización)

Nivel	Enfoque	Áreas de proceso
5 En Optimización.	Mejoramiento Continuo de Procesos.	Implementación e Innovación Organizativa. Análisis y Resolución Causal.
4 Administrado Cuantitativamente.	Administración Cuantitativa.	Administración Cuantitativa de Proyecto. Función Organizativa de Proceso.
3 Definido.	Estandarización de procesos.	Desarrollo de Requisitos. Solución Técnica. Integración del Producto. Verificación. Validación. Enfoque Organizativo de Proceso. Definición Organizativa de Proceso +IPPD. Entrenamiento organizativo. Administración de Proyecto Integrado +IPPD. Administración de Riesgo. Análisis y Resolución de la Decisión.
2 Administrado.	Administración de proyecto.	Administración de Requisitos. Planificación de Proyecto. Monitoreo y Control de Proyecto. Administración de Acuerdos del Proveedor. Análisis y Medición. Aseguramiento de la Calidad del Proceso y el Producto. Administración de Configuración.
1 Inicial.		

1.3 ESTUDIO DE METODOLOGÍAS DE DESARROLLO DE SOFTWARE

1.3.1 RUP

RUP es un proceso de desarrollo de SW y a la vez, es un conjunto de actividades dirigidas a transformar los requerimientos del cliente en un sistema de SW. Aumenta la productividad del equipo de desarrollo, permitiendo a cada miembro compartir un lenguaje común (UML), un proceso y una vista de cómo revelar el software. RUP es un proceso configurable y es soportado por herramientas que automatizan partes grandes del proceso. Son usadas para crear y mantener los diversos artefactos de la ingeniería de software.

Principios de RUP [RUP, 2005]

- Adaptar el proceso.
- Equilibrar las prioridades de los interesados que están enfrentadas.
- Colaborar con los otros equipos.
- Demostrar el valor de forma iterativa.
- Elevar el nivel de abstracción.
- Centrarse continuamente en la calidad.

Roles de RUP [RUP, 2005]

Analistas: Se encarga fundamentalmente de obtener e investigar los requisitos.

- Analista del negocio.
- Analista del sistema.
- Especificador de requisitos.

Desarrolladores: Se encarga fundamentalmente de diseñar e implementar software.

- Diseñador del negocio.
- Arquitecto del sistema.
- Arquitecto de software.
- Diseñador.
- Diseñador de interfaz de usuario.
- Diseñador de base de datos.
- Diseñador de sistemas.
- Implementador.
- Integrador.

Gestores: Se encarga de gestionar y configurar el proceso de ingeniería de software.

- Gestor de configuración.
- Gestor de control de cambios.

- Gestor de proyectos.
- Gestor de pruebas.
- Revisor de gestión.

Producción y Soporte: No están directamente relacionados con la definición, gestión, desarrollo y prueba de software, pero que se necesitan para dar soporte al proceso de desarrollo de software, o para producir materiales adicionales que necesita el producto final.

- Administrador.
- Ingeniero de proceso.

Verificadores: Se encarga fundamentalmente de verificar el software.

- Analista de pruebas.
- Diseñador de pruebas.
- Verificador.

Roles Generales: No encajan en ningún otro conjunto de roles pero son necesarios definirlos en un momento determinado.

- Revisor Técnico.
- Coordinador de revisión.
- Cualquier rol.
- Revisor.
- Interesado.

Artefactos de RUP

RUP es un proceso que genera una cantidad muy grande de artefactos. Con el afán de resumirlos, sólo se enumeran a continuación los que más se han utilizado en los proyectos de la UCI:

- | | |
|-----------------------------------|-------------------------------|
| 1. Modelo de CU del negocio. | 25. Modelo de Implementación. |
| 2. Modelo de objetos del negocio. | 26. Plan de prueba. |

3. Especificaciones complementarias del negocio.
4. Glosario de términos del negocio.
5. Caso de uso (CU).
6. Modelo de CU.
7. Glosario de términos.
8. Especificación de requisitos.
9. Plan de administración de requisitos.
10. Documento visión.
11. Paquetes de CU.
12. Modelo de análisis.
13. Modelo de diseño.
14. Modelo de despliegue.
15. Documento de Arquitectura de Software.
16. Clases.
17. Realización de CU.
18. Subsistema.
19. Paquetes.
20. Modelo de datos.
21. Mapa de Navegación.
22. Prototipo de Interfaz de Usuario.
23. Cápsula.
24. Diseño de Pruebas.
27. Estrategia de prueba.
28. Arquitectura de automatización de la prueba.
29. Configuración del entorno de la prueba.
30. Script de prueba.
31. Especificación de interfaz de la prueba.
32. Casos de prueba.
33. Resumen de evaluación de la prueba.
34. Resultado de la prueba.
35. Suite de Pruebas.
36. Arquitectura de automatización de las pruebas.
37. Datos de Pruebas.
38. Plan de proyecto.
39. Plan de desarrollo de software.
40. Plan de Gestión de Configuración.
41. Proceso de desarrollo.
42. Infraestructura de desarrollo.
43. Caso de desarrollo.
44. Directrices específicas del proyecto.
45. Plantillas específicas del proyecto.
46. Herramientas.
47. Guía de elaboración del manual.

El ciclo de vida de RUP [RUP, 2005]

La vida de un sistema transcurre a través de ciclos de desarrollo, desde que comienza hasta que termina, en cada ciclo se repite el proceso unificado de desarrollo. Cada uno consta de cuatro fases (Inicio, elaboración, construcción, transición) y concluye con una versión del producto. Cada fase se subdivide en iteraciones. Una iteración es una secuencia de actividades con un plan establecido y criterios de evaluación, cuyo resultado es una versión del software.

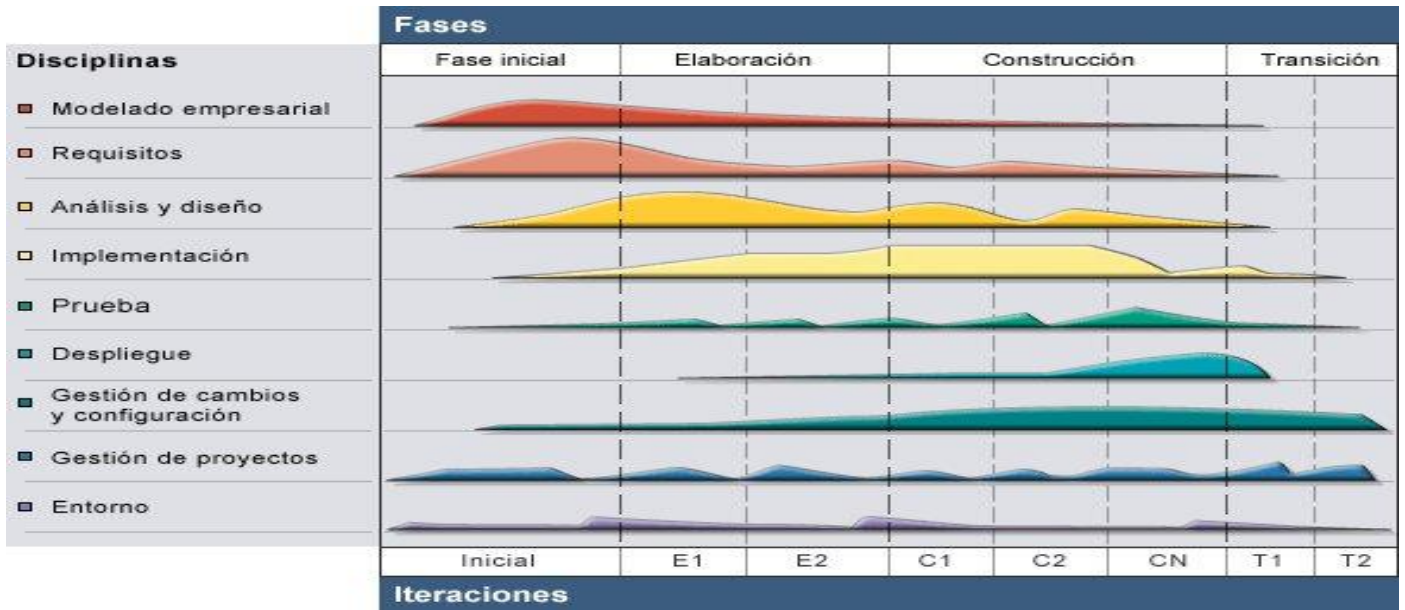


Figura 4. Arquitectura global de RUP [RUP, 2005]

El ciclo de vida de RUP se caracteriza por ser: [RUP, 2000]

1. Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
2. Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.
3. Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada mini proyecto es el resultado de una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

Fases de RUP [RUP, 2005]

Una fase es un período de tiempo entre dos objetivos importantes. Al final de cada fase se lleva a cabo una valoración para determinar si los objetivos de la fase se han alcanzado. Una valoración satisfactoria permite que el proyecto continúe a la fase siguiente.

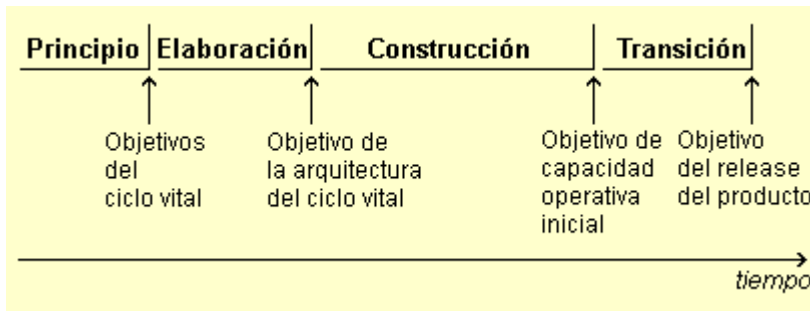


Figura 5. Fases e hitos. [RUP, 2005]

- Inicio: Establece un acuerdo entre todos los interesados respecto a los objetivos del ciclo de vida para el proyecto. La fase inicio es muy significativa fundamentalmente en los esfuerzos de desarrollo nuevos, pues son más arriesgados para los requisitos y para la actividad comercial. Deben abordarse antes de que el proyecto pueda continuar.
- Elaboración: Establece una línea base para la arquitectura del sistema para proporcionar una base estable para el grueso del diseño y del esfuerzo de implementación en la fase de construcción. La arquitectura evoluciona a partir de una consideración sobre los requisitos más significativos (los que tienen un gran impacto en la arquitectura del sistema) y una valoración de los riesgos. La estabilidad de la arquitectura se evalúa mediante uno o más prototipos arquitectónicos.
- Construcción: Se clarifican los requisitos restantes y se completa el desarrollo del sistema basándose en la arquitectura de línea base. La fase de construcción es, de alguna manera, un proceso de fabricación, en el que se pone el énfasis en la gestión de los recursos y el control de las operaciones para optimizar los costes, la planificación y la calidad.
- Transición: Se garantiza que el software esté disponible para los usuarios. La fase de transición puede acarrear varias iteraciones e incluye las pruebas del producto en preparación para la liberación, así como ajustes menores basados en la información de retorno de los usuarios. En este momento del ciclo vital, la información de retorno de los usuarios debe centrarse especialmente en el ajuste del producto, las cuestiones de configuración, instalación y utilización.

Flujos de trabajo de RUP [RUP, 2005]

La enumeración de roles, artefactos y actividades no constituye por sí solo un proceso. Necesitamos además una forma de describir secuencias significativas de actividades que producen algún resultado valioso. Y un flujo de trabajo no es más que esto, una secuencia de actividades que produce un resultado de valor observable. Los flujos de trabajos de RUP están divididos en 6 de ISW, más 3 de soporte. En total son 9 flujos que se describen a continuación.

- Modelado del negocio: Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- Requerimientos: Explica cómo obtener las solicitudes de los interesados y transformarlas en un conjunto de productos de trabajo de los requisitos que cubran el ámbito del sistema que va a crearse y proporcionen requisitos detallados sobre lo que el sistema debe hacer.
- Análisis y diseño: Explica cómo transformar los productos de trabajo de los requisitos en los productos de trabajo que especifiquen el diseño del software que el proyecto va a desarrollar.
- Implementación: Explica cómo desarrollar, organizar, realizar pruebas de unidad e integrar los componentes implementados basándose en las especificaciones de diseño.
- Prueba: Proporciona orientación sobre cómo evaluar y valorar la calidad del producto.
- Despliegue: Describe las actividades asociadas al garantizar que el producto de software esté disponible para los usuarios. Su principal objetivo es garantizar con éxito distribuciones del producto para los usuarios.
- Gestión de cambios y configuración: Explica cómo controlar y sincronizar la evolución del conjunto de productos de trabajo que componen un sistema de software. Su principal finalidad es mantener la integridad de todos los artefactos que se crean en el proceso.
- Gestión de proyectos: Se centra en la planificación del proyecto, la gestión del riesgo, la supervisión del progreso y la métrica.
- Entorno: Organiza aquellos elementos de método que proporcionan el entorno de desarrollo de software que da soporte al equipo de desarrollo, incluidos los procesos y las herramientas. La finalidad de este flujo es proporcionar a la empresa de desarrollo de software un entorno de desarrollo de software (los procesos y las herramientas) que den soporte al equipo de desarrollo.

1.3.2 UPEDU [9]

UPEDU es un proceso de ISW personalizado de RUP, para el desarrollo de aplicaciones educativas. Proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es garantizar la producción de software de alta calidad que satisfaga las necesidades de sus usuarios finales dentro de un calendario y presupuesto previsible.

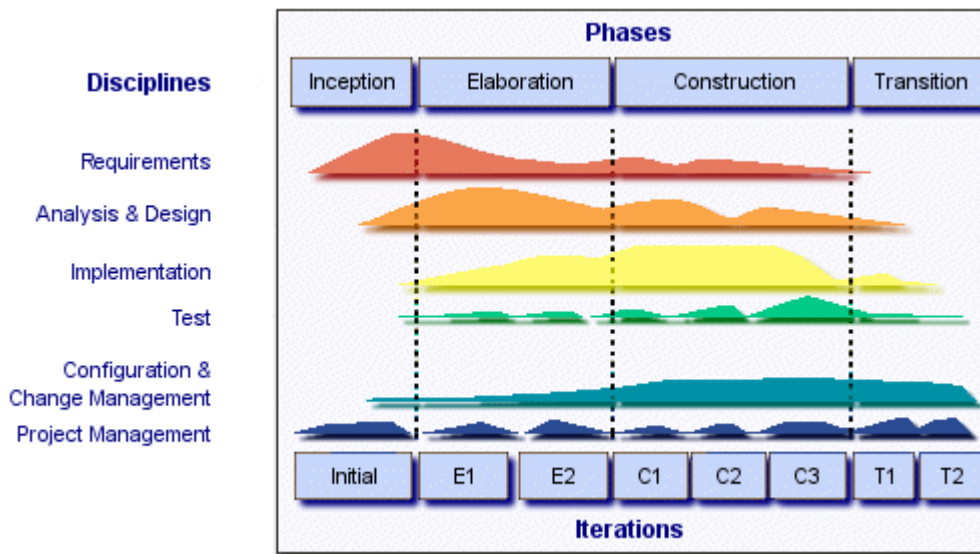


Figura 6. UPEDU. [9]

Roles de UPEDU

- Analista.
- Diseñador.
- Integrador.
- Probador.
- Administrador de control y cambio.
- Administrador de Configuración.
- Líder de Proyecto.
- Cualquier otro.
- Stakeholder.

Artefactos de UPEDU

1. Visión.
3. Glosario.
5. Especificación de requerimientos de software.
7. Especificaciones suplementarias.
9. Caso de Uso (CU).
11. Modelo de CU.
13. Clase de análisis.
15. Realización de CU.
17. Clase de diseño.
19. Modelo diseño.
21. Compilación.
23. Modelo de implementación.
25. Subsistema.
27. Plan de prueba.
2. Reporte de evaluación de prueba.
4. Caso de prueba.
6. Resultados de prueba.
8. Defecto.
10. Clase de prueba.
12. Componente de prueba.
14. Plan de desarrollo de software.
16. Plan de medición.
18. Plan de iteración.
20. Lista de riesgos.
22. Orden de trabajo.
24. Registro de revisiones.
26. Plan de administración de configuración.
28. Solicitud de cambio.

Flujos de trabajo de UPEDU

- Requerimientos.
- Análisis y diseño.
- Implementación.
- Prueba.
- Gestión de cambios y configuración.
- Gestión de proyectos.

1.3.3 OPENUP [10]

OpenUP es un proceso unificado iterativo e incremental. Se centra en la naturaleza de colaboración del desarrollo de software. Se trata de una metodología, que puede ser ampliada para hacer frente a una amplia variedad de tipos de proyectos.

Principios de OpenUP

- Balance.
- Colaboración.
- Evolución continua.
- Enfoque en articular la arquitectura.

Roles de OpenUP

- Analista.
- Cualquier rol.
- Arquitecto.
- Desarrollador.
- Líder de Proyecto.
- Stakeholder.
- Probador.

Artefactos de OpenUP

1. Cuaderno de apuntes de arquitectura.
2. Glosario.
3. Aplicación.
4. Visión.
5. Compilación.
6. Requerimientos no funcionales.
7. Diseño.
8. Modelo de Casos de Uso.
9. Desarrollador de prueba.
10. Caso de Uso.
11. Riesgo.
12. Caso de prueba.
13. Lista de productos de trabajo.
14. Prueba de Script.
15. Plan de Iteración.
16. Registro de prueba.
17. Plan de Proyecto.

Ciclo de vida de OpenUP

El ciclo de vida para OpenUP está estructurado en las mismas fases e hitos de RUP. Este ciclo de vida les crea a los interesados y miembros del equipo la visibilidad, y puntos de decisión a lo largo del proyecto. Esto permite una supervisión eficaz, y le permite tomar las decisiones en el momento oportuno. Un plan de proyecto define el ciclo de vida, y el resultado final es una aplicación liberada. OpenUP, es también iterativo e incremental, dirigido por Casos de Uso, y centrado en la arquitectura.

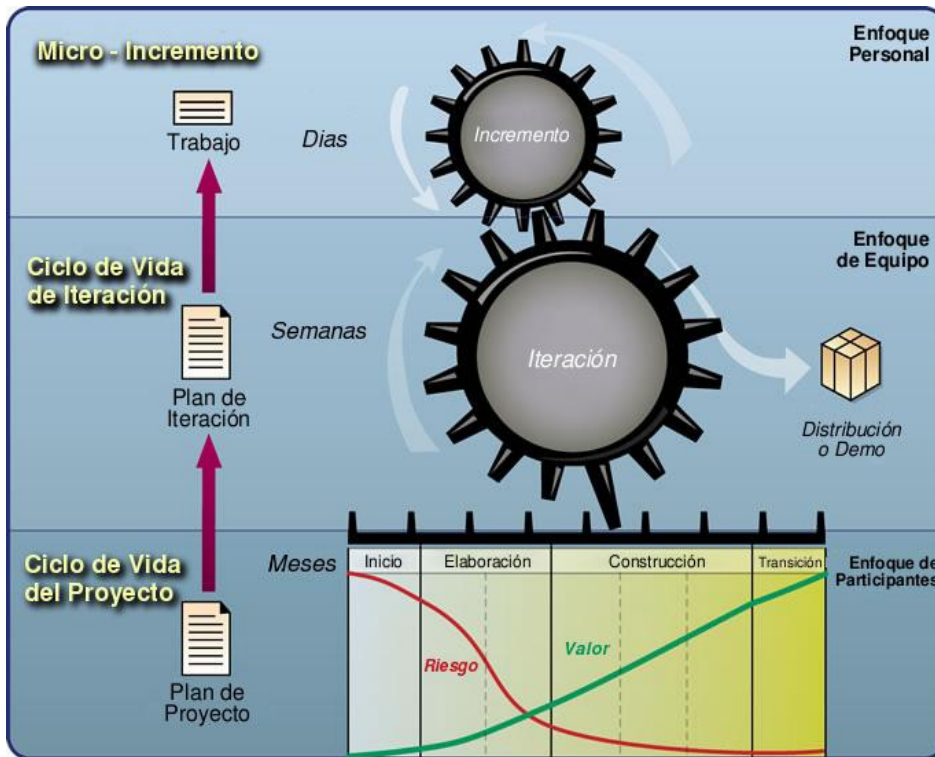


Figura 7. Proceso OpenUP. [10]

Flujos de trabajo de OpenUP

- Requisitos.
- Arquitectura.
- Desarrollo.
- Prueba.
- Gestión de proyectos.

1.3.4 PROGRAMACION EXTREMA (XP)

XP es una metodología de desarrollo ágil de SW, que puede ser usado por equipos pequeños y medianos. Para la construcción de aplicaciones con elevada calidad, con un mínimo gasto fijo, dentro de un presupuesto y calendario previsible. Con XP, son priorizados los resultados de trabajo inmediatos, haciendo el proceso de desarrollo más sencillo y aplicando el sentido común. También aumenta la productividad en el desarrollo, potenciando al máximo el trabajo en equipo.

Principios y prácticas de XP [11]

- Retroalimentación a escala fina: Desarrollo guiado por pruebas. Juego de planificación. Cliente presente. Programación en pares.
- Proceso continuo en lugar de por lotes: Integración continua. Refactorizar. Liberación pequeña.
- Entendimiento compartido: Diseño simple. Metáfora del sistema. Propiedad colectiva del código. Convenciones del código.
- Bienestar del programador: Paso sostenible.

Roles de XP [3]

- Programador: Escribe el sistema y mantiene el código lo más simple y definitivo posible.
- Cliente: Debe escribir buenas historias que ayuden a entender que es lo que quiere, tomar decisiones y escribir las pruebas que serán aplicadas al software.
- Probador: Ayuda a los clientes a escoger y escribir las pruebas funcionales del sistema. Corre dichas pruebas periódicamente, envía los informes con los resultados y se asegura que las herramientas de prueba funcionan bien.
- Rastreador: Realiza las mediciones y la recolección de métricas. Mide el progreso del proyecto y lo compara con lo estimado.
- Entrenador: Asegura el cumplimiento de todas las prácticas, transmitiendo sus conocimientos y experiencia al resto del equipo.
- Asesor: Es una persona que posee el conocimiento técnico necesario para poder ayudar al equipo con determinados problemas.
- Jefe: Toma las decisiones más importantes del proyecto. Es el encargado de comunicarse con el equipo para determinar cuál es la situación actual y distinguir cualquier dificultad o deficiencia en el proceso y poder solucionarlo.

Artefactos XP [12]

- Visión.
- Historias de Usuario.
- Prueba del cliente.

- Plan de liberación.
- Sistema de nombres.
- Plan de iteración.
- Estándar de código.
- Prueba de unidad.
- Código.
- Compilación.

Ciclo de vida de XP

El ciclo de vida de XP se enfoca en el carácter iterativo e incremental del desarrollo. Una iteración de desarrollo es un período de tiempo en el que se realiza un conjunto de funcionalidades determinadas; y en el caso de XP, corresponden a un conjunto de historias de usuarios. Las iteraciones son relativamente cortas, pues XP plantea que entre más rápido se le entreguen desarrollos al cliente, más retroalimentación se va a obtener y esto va a representar una mejor calidad del producto a largo plazo. Existe una fase de análisis inicial orientada a programar las iteraciones de desarrollo y cada iteración incluye diseño, codificación y pruebas, fases superpuestas de tal manera que no se separen en el tiempo. (Ver Figura 8)

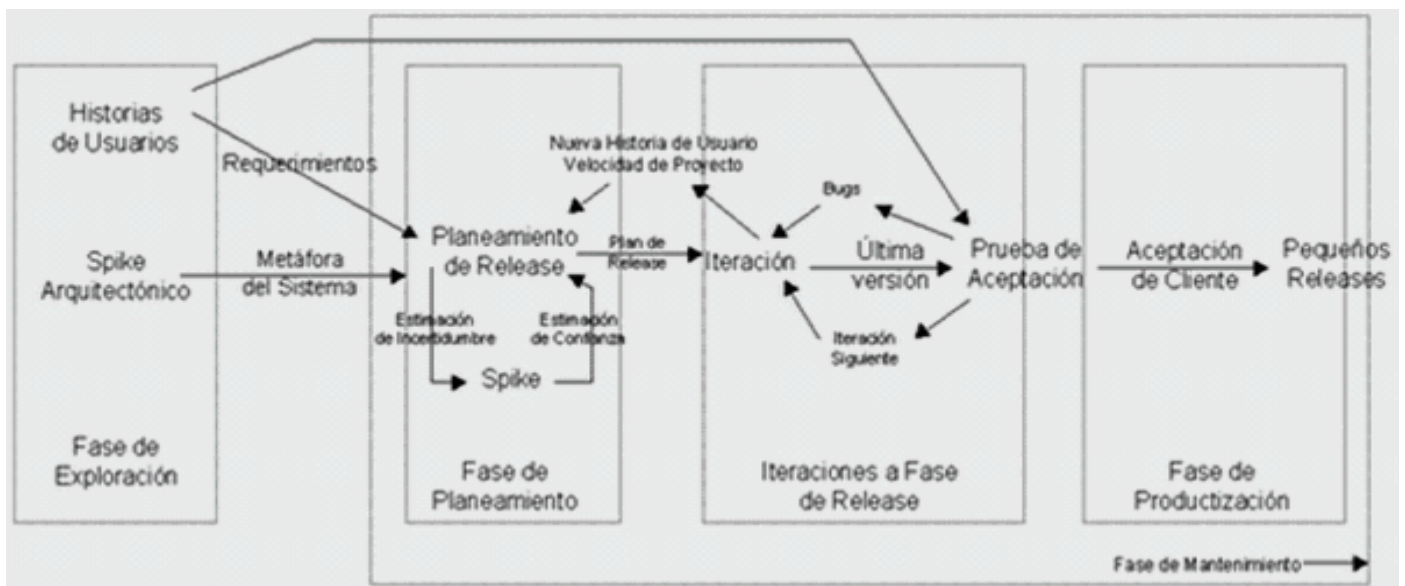


Figura 8. Proceso XP [3]

Fases de XP [Valencia, 2006]

- **Exploración:** Los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. El equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto.
- **Planeación:** Se fija una fecha en la que estará terminado el grupo de historias más pequeño y más importante.
- **Iteraciones:** Esta fase incluye varias iteraciones del sistema antes de la primera entrega. El cliente decide que historias van a ser implementadas para cada iteración.
- **Producción:** En esta fase aparecen nuevos cambios y se tiene que decidir si serán incorporados o no en dicha entrega.
- **Mantenimiento:** En esta fase por lo general se necesita un esfuerzo extra de los programadores para satisfacer los requerimientos del cliente.
- **Muerte:** Llegamos a esta fase cuando el cliente está satisfecho con el sistema, o cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

Actividades básicas de XP [3]

- **Codificar:** Es la única actividad de la que no podremos prescindir. Sin código fuente no hay programa. Por tanto se necesita codificar y plasmar las ideas a través del código. En una programación en pareja el código expresa la interpretación del problema, así se puede utilizar el código como un medio de comunicación entre los miembros del equipo, compartir las ideas, y por tanto para aprender y mejorar.
- **Hacer pruebas:** Las características del software que no pueden ser demostradas mediante pruebas simplemente no existen. Las pruebas indican si el trabajo está bien hecho o no. Las pruebas deben ser tan profundas como sea posible. No se debe escribir tan solo una prueba para ver que funciona y terminar, se debe de pensar en todas las posibles pruebas para el código. Cuando no se puede pensar en ninguna prueba que pudiera originar un fallo en el sistema, se ha acabado la actividad. Programar y probar es más rápido que sólo programar. Con esta técnica disminuyen los errores, se minimiza el tiempo de localización y revisión del código, además de que se obtendrán menos insatisfacciones del cliente.
- **Escuchar:** Es muy importante escuchar a los clientes. El cliente es quien sabe qué es lo que desea que haga el producto y cuáles son los problemas de su negocio. Se deben formular preguntas para los

clientes con el objetivo de captar la información que se necesita y explicar lo que es fácil o difícil de obtener. De la misma manera debe haber una escucha activa y retroalimentación entre todos los miembros del equipo de desarrollo.

- Diseñar: El diseño crea una estructura que organiza la lógica del sistema, un buen diseño permite que el sistema crezca con cambios en un solo lugar. Los diseños deben de ser sencillos, si alguna parte del sistema es de desarrollo complejo, es aconsejable dividirla en otras partes. Si hay fallos en el diseño o malos diseños, estos deben de ser corregidos cuanto antes.

1.3.5 MSF ÁGIL [GUIDANCE, 2005]

MSF Agile ofrece una guía sobre cómo organizar personas y proyectos para planificar, desarrollar y desplegar soluciones tecnológicas de software de una forma orientada al cambio. Constituye un disciplinado proceso ágil de desarrollo de SW, y un marco de trabajo extensible y adaptable.

Principios de MSF Agile

- Asociación con los clientes.
- Trabajo hacia una visión compartida.
- Entrega de resultados incrementales.
- Inversión en la calidad.
- Fortalecimiento de los miembros del equipo.
- Establecimiento de responsabilidades claras.
- Aprendizaje de todas las experiencias.
- Promoción de la comunicación abierta.
- Mantenimiento ágil, adaptación al cambio.

Roles de MSF Agile

- Analista del negocio.
- Administrador de proyecto.
- Arquitecto.
- Desarrollador.
- Probador.

- Administrador de entrega.
- Administrador de base de datos.
- Desarrollador de base de datos.

En MSF ágil cada miembro del equipo es responsable de representar las necesidades específicas de sus circunscripciones, y ninguno es más importante que otro. El conjunto de estos puntos de vista permiten controles y equilibrios para garantizar que el equipo produzca la solución correcta. Esto es representado en un Modelo de Equipo de Trabajo, el cual se muestra en la figura siguiente.



Figura 9. Modelo de equipo. [GUIDANCE, 2005].

Artefactos de MSF Agile

- | | |
|---|--|
| <ol style="list-style-type: none"> 1. Visión. 3. Requerimientos funcionales. 5. Acta de aprobación de visión. 7. Acta de aprobación de cronograma. 9. Manuales técnicos, de usuarios y de instalación. 11. Registro de pruebas. 13. Conjunto de archivos. 15. Registro de reportes. | <ol style="list-style-type: none"> 2. Detalle de la visión. 4. Matriz de riesgos. 6. Cronograma. 8. Código. 10. Acta de finalización de Desarrollo. 12. Acta de aprobación de Versión Aprobada. 14. Acta de Entrega y Finalización de Proyecto. |
|---|--|

Ciclo de vida de MSF Agile

El proceso de MSF Ágil combina las mejores prácticas del desarrollo Cascada y Espiral, con puntos de fijación para cada iteración.

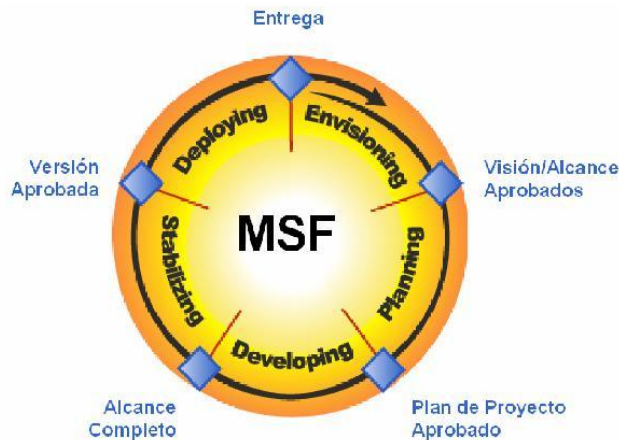


Figura 10. Modelo de proceso de desarrollo MSF. [15].

La integración sin problemas de MSF Agile en Visual Studio Team System soporta el desarrollo iterativo con el aprendizaje continuo y el refinamiento. La definición del producto, el desarrollo, y las pruebas se producen en la superposición de iteraciones incrementales resultantes de la finalización del proyecto. Las iteraciones pequeñas permiten reducir el margen de error en sus estimaciones y proporcionar información rápida acerca de la exactitud de los planes de su proyecto. Cada iteración debería traducirse en una parte estable de todo el sistema.

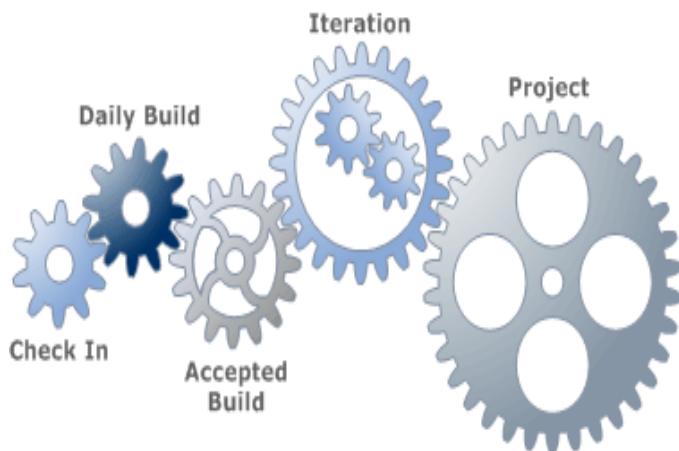


Figura 11. Ciclos de MSF. [GUIDANCE, 2005]

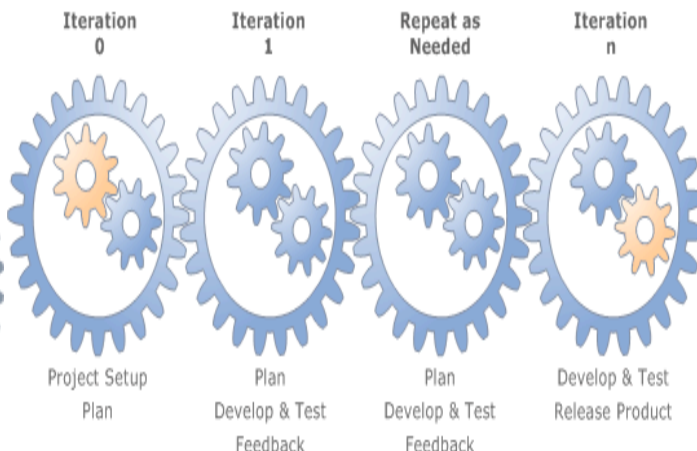


Figura 12. Iteraciones de MSF. [GUIDANCE, 2005]

Fases de MSF [13]

- **Visión:** Obtener una visión del proyecto compartida, comunicada, entendida y alineada con los objetivos del negocio. Además, identificar los beneficios, requerimientos funcionales, sus alcances, restricciones; y los riesgos inherentes al proceso.
- **Planeación:** Obtener un cronograma de trabajo que cumpla con lo especificado en la fase de Visión dentro del presupuesto, tiempo y recursos acordados. Este cronograma debe identificar puntos de control específicos que permitan generar entregas funcionales y cortas en el tiempo.
- **Desarrollo:** Obtener iterativamente de la mano de la fase anterior y la siguiente, versiones del producto entregables y medibles que permitan de cara al cliente probar características nuevas sucesivamente. Esto incluye ajustes de cronograma necesarios.
- **Estabilización:** Obtener una versión final del producto probada, ajustada y aprobada en su totalidad.
- **Instalación:** Entregar (instalar) al cliente el producto finalizado en su totalidad. Como garantía se han superado con éxito las etapas anteriores.

Disciplinas de MSF Ágil [13]

- **Gestión de Proyectos:** Planificar sobre entregas cortas. Incorporar nuevas características sucesivamente. Identificar cambios ajustando el cronograma.
- **Gestión de Riesgos:** Diseñada para ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan surgir. Este modelo proporciona un

entorno estructurado para la toma de decisiones y acciones valorando los riesgos que puedan provocar.

1.3.6 SCRUM

Esta es otra de las metodologías de desarrollo ágil utilizado en la producción de diferentes productos de software. Y como tal, es un modo de desarrollo de carácter adaptable más que predictivo. Orientado a las personas más que a los procesos. Emplea la estructura de desarrollo incremental basada en iteraciones y revisiones. El proceso Scrum en particular requiere trabajo duro porque no se basa en el seguimiento de un plan, ni prácticas de ISW, sino en la adaptación continua a las circunstancias de la evolución del proyecto.

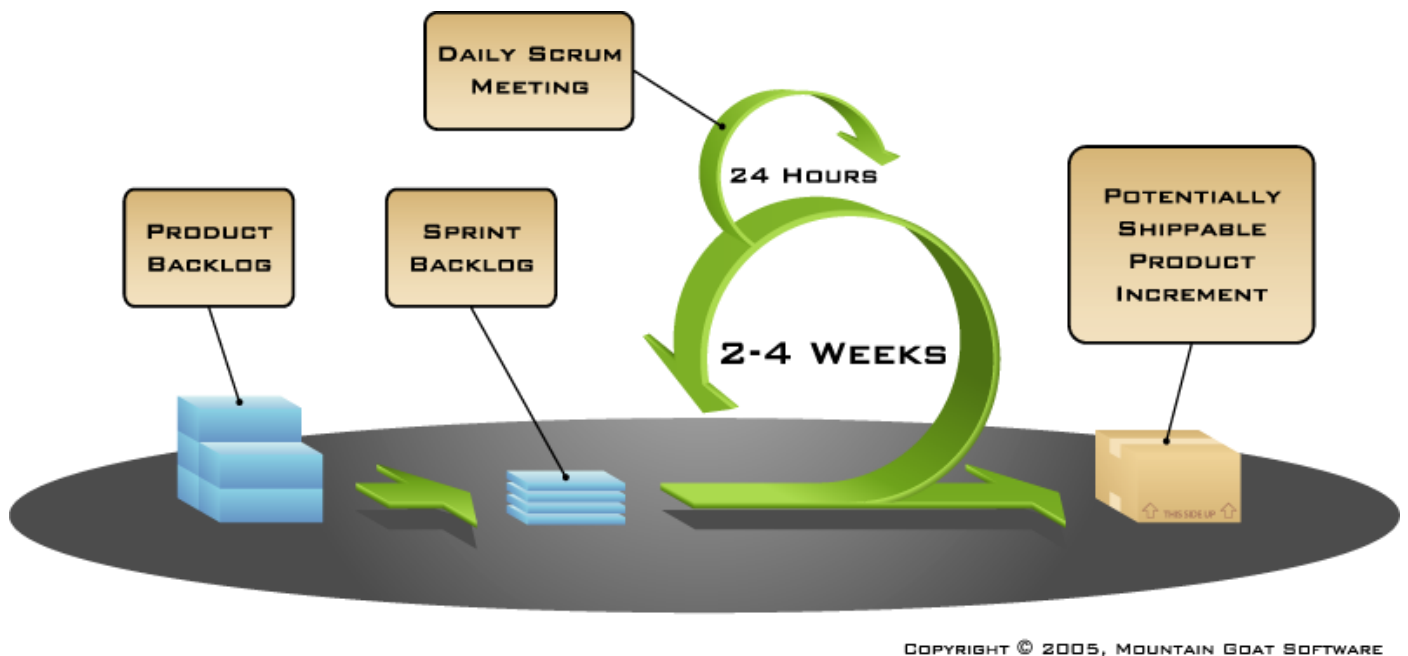


Figura 13. Vista general de Scrum. [14]

Ciclo de vida y desarrollo de Scrum [4]

Se comienza con la visión general del producto, especificando y detallando las funcionalidades o partes que tienen mayor prioridad de desarrollo y que pueden llevarse a cabo en un período de tiempo breve (normalmente de 30 días). Cada uno de estos períodos de desarrollo es una iteración que en Scrum se

denomina “Sprint”, y se finaliza con la producción de un incremento operativo del producto. (Ver figura 14) El Sprint es por tanto el núcleo central que proporciona la base del desarrollo iterativo e incremental del proceso ágil. Scrum gestiona su evolución a través de reuniones breves diarias en las que todo el equipo revisa el trabajo realizado el día anterior y el previsto para el día siguiente.

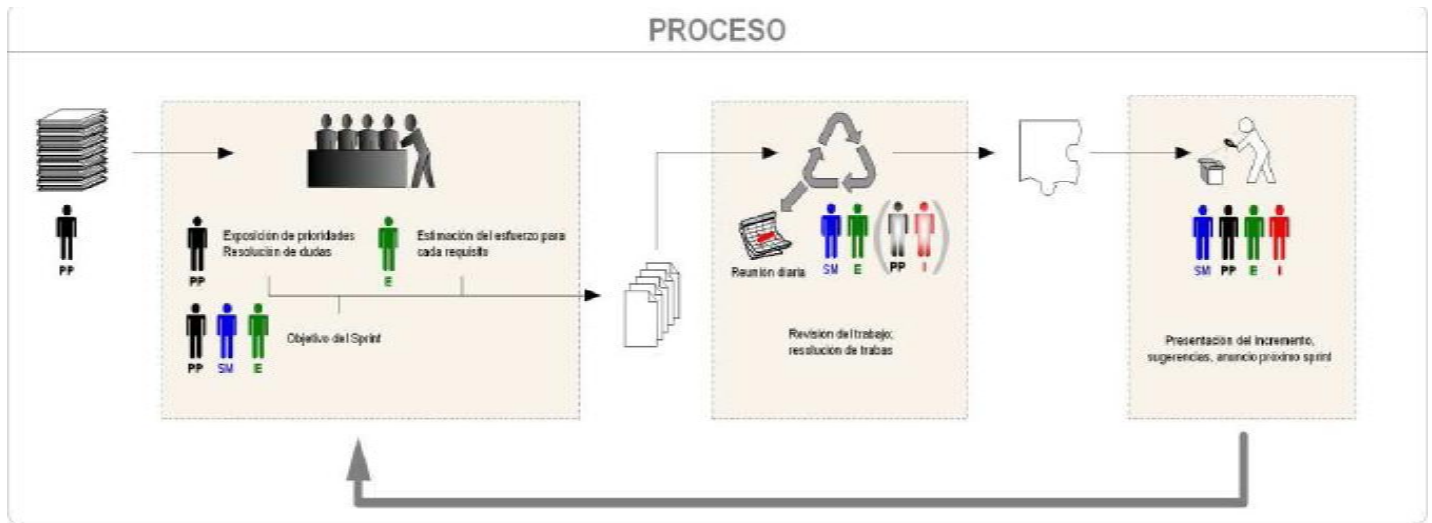


Figura 14. Proceso Scrum [4]

Roles de Scrum [4]

Scrum hace una clara diferencia entre dos grupos (comprometidos e implicados):

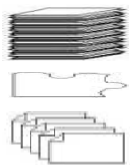
Comprometidos: garantiza que quienes tienen la responsabilidad tienen también la autoridad necesaria para poder lograr el éxito del proceso.

- Product Owner (Propietario del Producto): Representa a todos los interesados en el producto final. Marca las prioridades del producto. Lleva el control de las estimaciones y de las inversiones.
- Scrum Master: Es el responsable del proceso de Scrum. Se encarga de incorporar a Scrum, en la cultura de la organización. Asegura el cumplimiento de los roles y responsabilidades, la formación y el entrenamiento en el proceso.
- Scrum Team (Equipo): Debe transformar las tareas del Sprint Backlog en un incremento de funcionalidad en el software. Desarrolla el producto con calidad. Debe auto gestionarse y auto organizarse para facilitar la ejecución del proceso. Debe ser multifuncional y no debe tener más de ocho miembros.

Implicados: Garantiza que quienes no tienen la responsabilidad no puedan producir interferencias innecesarias.

- Usuarios finales
- Marketing
- Áreas comerciales
- Áreas contables
- Otros interesados.

Artefactos de Scrum [14]



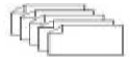
Product Backlog (Generador de producto).

Cuaderno de tareas



Gráfica de progreso del Sprint

Gráfica de progreso de liberación



Sprint Backlog (Generador de Sprint)

Incremento del Producto

Fases de Scrum [Schwaber, 2001]

1. Pre juego: Esta fase incluye las subfases de Planeamiento y de Montaje. Consisten en la definición del sistema que será construido y en la planificación del diseño de alto nivel del sistema respectivamente.
2. Juego: Es la parte ágil de Scrum y controla los cambios constantemente a través de los Sprints.
3. Post Juego: El sistema está listo para ser liberado. Se realiza la integración, pruebas del sistema y documentación.

Flujos de trabajo de Scrum [4]

- Estimación del Product Backlog: Periódicamente el Team Scrum, estimará el tamaño de cada producto en el Product Backlog.
- Priorización de los Backlog: Los productos en el Product Backlog, son priorizados por el Product Owner antes de planificar y entregar el Sprint.
- Planificación de la liberación: Al principio del proyecto el equipo creará un plan de liberación de alto nivel.

- Desarrollo del Sprint: Después de la Reunión de revisión del Sprint, el equipo comienza a desarrollar el Sprint.



Reunión de planificación de Sprint: Es donde el Team Scrum y el Product Owner deciden cuáles funciones y tareas serán desarrolladas en el Sprint próximo.



Reunión diaria: El Scrum Master y el Scrum Team, se reúnen diariamente (alrededor de 15 min).



Reunión de revisión del Sprint: Análisis y revisión del incremento generado.

1.4 ESTUDIO DEL DESARROLLO DE SISTEMAS DE REALIDAD VIRTUAL

La RV ofrece simulaciones en un ambiente tridimensional, que permiten percibir e interactuar con el contenido de dicho ambiente. Por tanto, un SRV es aquel que está compuesto por la simulación, la interacción y la percepción. Es decir, un SRV debe tener la capacidad de convencer a los usuarios con objetos o ambientes, de estar en una situación paralela a la realidad. Debe permitir el control del sistema creado y facilitar la interacción con los sentidos del usuario (vista, oído y tacto). Según la complejidad del sistema, los elementos externos utilizados para producir estas sensaciones serán más o menos simples. Para construir un mundo virtual se debe tener en cuenta fundamentalmente los elementos de gráficos 3D (modelo geométrico del objeto, iluminación y textura), los movimientos, los elementos de la “vida artificial”, y las características del SW para el desarrollo del SRV.

Descritos los elementos necesarios para conocer sobre la RV, se puede entonces hacer un análisis acerca de los diferentes métodos de trabajo que se conocen en el mundo actual. Existen muchos en RV, aquí sólo se mencionará una de las propuestas que mejor define sus actividades.

1.4.1 SECUENCIA DE ACTIVIDADES PROPUESTAS POR JUAN C. PARRA M²

Esta propuesta se basa primeramente en una estructuración de pasos a seguir para la construcción del ambiente virtual, en un análisis top-down³ en la etapa de diseño y en un proceso bottom-up⁴ para construcción de los objetos. [Concepción, 2001]

Esta propuesta de desarrollo contempla las siguientes etapas de desarrollo, las cuales incluyen algunas consideraciones que describimos a continuación:

1. Definir ambiente problema: Conceptualizar el problema, identificar los componentes que interactúan (externos e internos al ambiente problema) y el nivel de detalle del problema (los objetos que participan), por último generar diagrama jerárquico con objetos válidos.
2. Definir los objetos del sistema por construir:
 - 2.1. Describir los objetos del ambiente problema.
 - 2.2. Definir detalles de los objetos (incluir escala).
 - 2.2.1. Definir componentes por objetos (Diagrama jerárquico de componentes).
 - 2.2.2. Definir interacción entre objetos.
 - 2.2.3. Definir texturas, sonido, entre otros en objeto global.
 - 2.3. Definir detalle de componentes de los objetos.
 - 2.3.1. Identificar partes que interesan en comportamiento.
 - 2.3.2. Definir interacción entre componentes.
 - 2.3.3. Definir comportamiento de los componentes.
 - 2.3.4. Definir reglas adicionales (mensajes, alertas, etc.).

² Juan Carlos Parra; Ingeniero Civil Informático por la Universidad de Concepción (1990). Se desempeña como docente del Dpto. de Sistemas de Información de la Universidad del Bío-Bío desde 1994 e Ingeniero consultor en Informática. Ha sido Director del Laboratorio de Realidad Virtual y Director del Programa de Investigación en Realidad Virtual de la misma Universidad. Ha presentado ponencias y publicado artículos en Chile, Bolivia, Argentina, Brasil, Cuba, Estados Unidos, Bélgica y España.

³ Donde lo principal es el problema. Se estructura el sistema sobre la base de preguntas básicas, estableciendo los atributos de los objetos, el comportamiento, la iluminación y otros atributos. Establecer las propiedades del espacio virtual que contendrá los objetos para luego ir desarrollando en detalle las especificaciones, es decir, mirar la aplicación desde arriba, en forma general para luego ir a los niveles de detalle.

⁴ Corresponde a definir primeramente los detalles de la aplicación hasta llegar al sistema global. Aquí se toman en cuenta primeramente los sistemas coordenados y los objetos con sus características de rotación, traslación y escalamiento, las especificaciones de color, sombra y textura para cada objeto y finalmente se tratan los límites, la iluminación y las características del ambiente simulado.

- 2.3.5. Definir texturas, sonido, etc. por cada componente.
3. Construir los objetos a partir de componentes:
 - 3.1. Descomponer objetos en partes constituyentes válidas para el problema.
 - 3.1.1. Chequear definición de componentes.
 - 3.1.2. Identificar forma válida de construcción para cada componente.
 - 3.1.3. Definir escalas y medida de componentes y objetos.
 - 3.2. Construir forma para cada componente (Asociar texturas necesarias).
 - 3.3. Construir objeto final a partir de la forma.
 - 3.3.1. Distribuir componentes.
 - 3.3.2. Agrupar componentes según necesidad.
 - 3.3.3. Agrupar objetos.
 - 3.4. Asociar sonido, atributos, texturas, entre otros a objeto total.
4. Incorporación del programa de comportamiento: Definir algoritmos de comportamiento para cada componente y objeto que lo requiera, construir programa de comportamiento a partir de algoritmo y realizar prueba de comportamiento (para componentes y objetos).
5. Construir el ambiente virtual total a partir de los objetos: Incorporar atributos a objetos, chequear bosquejo con definición del problema, ajustar programas de comportamiento mediante prueba de interacción, agrupar objetos, construir programas adicionales, y por último, definir las reglas adicionales (mensajes, diseño, etc.).
6. Optimización de diseño: En esta etapa se prueban los objetos con comportamiento, se corrigen detalles estéticos tales como colores, luces y otros.
7. Incorporación de periféricos: Se instalan los dispositivos según sea la necesidad de la aplicación, tales como ascos, guantes, etc.
8. Validar diseño final: Es la validación final de comportamiento y aspecto globales del diseño. Es la prueba final del sistema propiamente tal. Se incluye aquí la validación de la configuración del equipamiento en el cual trabajará la aplicación.

CONCLUSIONES PARCIALES

Tras el estudio detallado en este capítulo, se llegan a las siguientes conclusiones:

- Se logra profundizar en algunos métodos de trabajo existentes, en busca de elementos a tener en cuenta, para elaborar una guía de acuerdo a las exigencias de los proyectos de RV.
- Los métodos, y procesos de trabajo deben ser adaptados a las necesidades del PRV y no al revés.

INTRODUCCIÓN

Los Polos Productivos (PP) son organizaciones dentro de la UCI que conforman el conjunto de materias y proyectos productivos dedicados a una especialidad en particular en términos de Informática. El PRV impulsado por la Facultad 5 de la UCI, tiene un trabajo bastante amplio, aunque no tiene bien definida una guía ajustada al desarrollo de sus sistemas. El objetivo fundamental del PRV es generar conocimientos y soluciones tecnológicas de punta asociadas a la RV, acorde a los modelos internacionales de calidad, con un alto impacto social, económico y medioambiental, que tributen a la informatización del país, y a la comercialización de productos con un alto valor agregado. En el presente Capítulo se analizará profundamente el modo de producción que actualmente presenta el PRV, mediante la revisión de documentación para recopilar la información necesaria. Igualmente, se aplicarán encuestas, así como entrevistas dirigidas a los líderes, y miembros de proyectos. Continuamente se valorarán las metodologías, procesos y técnicas estudiadas, identificando de esta manera, los aspectos favorables en la producción de los SRV dentro del PRV.

2.1 PECULIARIDADES DEL POLO DE REALIDAD VIRTUAL

Las metodologías existentes no se adecuan completamente al modo de producción del PRV, debido a que los productos que se desarrollan en el Polo tienen varias características específicas, que han propiciado la necesidad de elaborar una guía centrada en las peculiaridades de las aplicaciones de RV. Algunas de estas particularidades, se mencionan a continuación:

- El desarrollo de aplicaciones de RV, genera algunos productos de trabajo específicos como las “media”, dentro de los cuales se encuentran las texturas, modelos, sonidos, videos, animaciones y efectos especiales.
- Los requisitos funcionales son muy cambiantes, y no se obtienen siempre desde una etapa inicial.
- Se requieren de nuevos roles, y actividades para el desarrollo de productos de RV.
- Se utilizan guiones para describir el ambiente del problema, y de la aplicación.
- La reutilización de soluciones, es un factor primordial en el desarrollo de productos de RV.

De manera general, la estructura del software de RV presenta mayor complejidad que lo normal en un software computacional. Esto se debe a que un SRV constituye un ambiente 3D que se extiende hasta capacidades multimediales, también dispone de una programación específica, así como el control de combinados dispositivos externos funcionando como un todo, y modificándose en tiempo real. De esta manera, se establecen un conjunto de características especiales, que conllevan a los autores de este trabajo a elaborar una guía para el desarrollo de aplicaciones de RV, teniendo en cuenta que el proceso debe dividirse en dos aspectos fundamentales: El desarrollo del dominio, y la aplicación.

Durante el desarrollo del dominio se utiliza el guión para diseñar el escenario 3D, audio, comportamientos, y configurar dispositivos. El desarrollo de la aplicación, debe definir actividades para la navegación interactiva, donde se puede visualizar, recorrer, y manipular el ambiente virtual, de acuerdo con lo preparado en el desarrollo del dominio.

2.2 ESTUDIO DEL POLO DE REALIDAD VIRTUAL

Actualmente el PRV agrupa sus proyectos en Áreas Temáticas (AT), según las características específicas o la aplicación de cada producto que se está desarrollando. (Ver Figura 15)

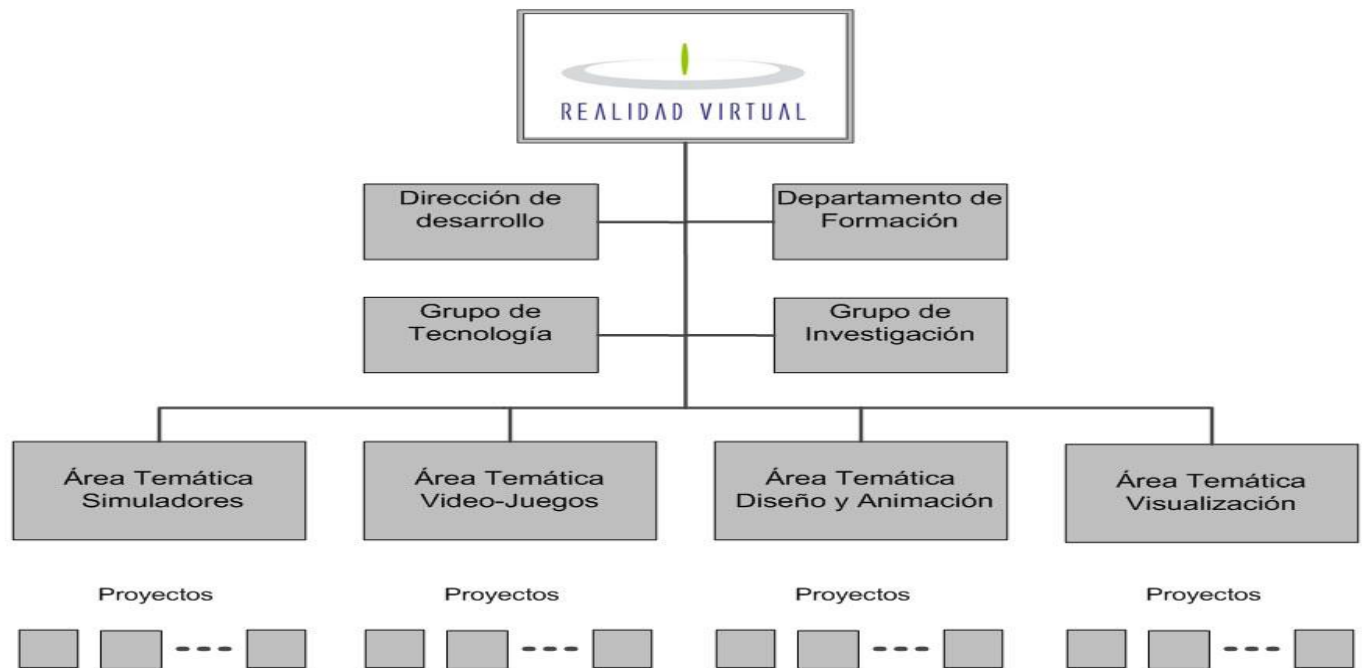


Figura 15. Organigrama del Polo de Realidad Virtual.

El AT de Simuladores se encarga de la investigación, y el desarrollo de herramientas y simuladores de propósito general. Además, regula una arquitectura base única para los productos del Polo, así como las metodologías, y flujos de trabajo para el desarrollo de simuladores profesionales. Esta área oferta servicios de actualizaciones, y asesoramiento en el uso de esta herramienta para otras áreas temáticas. El AT de Diseño y Animación, tiene como propósitos fundamentales promover la reutilización de soluciones entre los proyectos, definir flujos de trabajo para el desarrollo y la interacción de los proyectos. Esta área oferta servicios de videos en 3D de bajo y alto nivel de detalle, modelado y animación. Así como texturas, entornos y recorridos virtuales, maquetas digitales interactivas, videos compuestos y vistas panorámicas a 360°. El objetivo principal del AT de Video-Juegos es agrupar proyectos afines a la construcción de videojuegos. Esta área promueve la reutilización de soluciones entre los proyectos del Polo, y regula la uniformidad del trabajo en cuanto a arquitectura, metodologías, entre otros. El AT de Visualización, se dedica a la producción de sistemas, y soluciones 3D con un nivel alto de fiabilidad. Los sistemas desarrollados en esta área deberán tener un grado de interactividad adecuado, y permitir cierta sensación de continuidad visual.

De manera general, para el proceso de maduración de la calidad en la formación de los profesionales, el PRV pone a su disposición el PSP (Personal Software Process), y el CMM (Capability Maturity Model) como modelo de calidad a un nivel personal, y organizacional.

Los proyectos del PRV deben garantizar aplicaciones multirenderer (que se puedan representar las escenas inicialmente con OpenGL, DirectX, y extensible a otras bibliotecas gráficas). Así como garantizar aplicaciones multiplataforma (que corran igualmente sobre Sistemas Operativos (SO) Windows y Linux, y extensible a otros SO). Se debe trabajar en código C++ estándar, y seguir los estándares de codificación y documentación de código definidas para el Polo.

Según estudios realizados en el proceso de desarrollo, actualmente el Polo distribuye las responsabilidades por grupos de roles según la categoría docente del personal. Entre los grupos de roles se identifican:

Grupo	Roles	Categoría docente
Dirección del proyecto	Líder de proyecto	Profesor
	Jefe de equipo	Profesor o Estudiante de 5to año
Investigación	Investigador	Estudiante de 4to, 5to año
Producción	Arquitecto de software	Estudiante de 4to, 5to año
	Analista de software	Estudiante de 4to, 5to año
	Integrador	Estudiante de 4to, 5to año
	Desarrollador (programador)	Estudiante de 3er, 4to año
	Diseñador de interfaz de usuario	Estudiante de 2do, 3er año
	Escritor técnico	Estudiante de 2do año
Control de calidad	Probador	Estudiante de 1er, 2do año
	Diseñador de casos de prueba	Estudiante de 3er año
	Auditor de la calidad	Estudiante de 3er año
	Revisor técnico formal	Estudiante de 4to año
	Planificador de la calidad	Estudiante de 4to, 5to año
	Asuntos legales	Profesor
Administración	Arquitectura y tecnologías	Estudiante de 4to, 5to año
	Administrador de configuración	Estudiante de 3er, 4to año

Tabla #1. Roles definido para el Polo de Realidad Virtual. [6]

La documentación y los artefactos generados en cada proyecto, durante el proceso de desarrollo de SW en el PRV, se encuentra organizada en un Expediente de Proyecto⁵. El cual está basado en la propuesta

⁵ Es un sistema de carpetas y archivos, que contiene toda la información referente al proyecto productivo durante todo su ciclo de vida.

de expediente en su versión 2.01, realizada por la Dirección de Calidad de Software [5] en la Universidad, y tiene la estructura siguiente:

Expediente de proyecto

1. Ingeniería

1.1 Requisitos

- 1.1.1 Diagrama de Proceso Nombre del Proceso
- 1.1.2 Especificación de requisitos de software
- 1.1.3 Evaluación de Áreas de la Organización
- 1.1.4 Historias de Usuario
- 1.1.5 Modelo del Dominio
- 1.1.6 Modelo del Negocio
- 1.1.7 Modelo del Negocio con BPM
- 1.1.8 Modelo del sistema
- 1.1.9 Plan de gestión de requisitos

1.2 Arquitectura y diseño

- 1.2.1 Arquitectura de Información
- 1.2.2 Arquitectura de Software
- 1.2.3 Informe del Levantamiento de Información para la Arquitectura de Información
- 1.2.4 Modelo de Diseño

1.3 Implementación y pruebas

- 1.3.1 **Código fuente**
- 1.3.2 **Manual de usuario**
- 1.3.3 Casos de prueba
- 1.3.4 Diseño de Caso de Pruebas funcionales basados en CU
- 1.3.5 Diseño de Casos de Prueba HU
- 1.3.6 No Conformidades
- 1.3.7 Plan de pruebas
- 1.3.8 Registro de Prueba Unitaria o Integración

1.4 Despliegue e instalación

1.4.1 Modelo de Despliegue

2. Gestión de proyecto

2.1 Plan del proyecto

- 2.1.1 Diario de actividades
- 2.1.2 Pila de Sprint (Sprint Backlog)
- 2.1.3 Plan Desarrollo de Software
- 2.1.4 Presupuesto
- 2.1.5 Registro Tarea de Ingeniería

2.2 Riesgos

- 2.2.1 Plan Mitigación de Riesgos

2.3 Recursos

- 2.3.1 Ambiente de desarrollo
- 2.3.2 Plan de capacitación
- 2.3.3 Roles y responsabilidades

2.4 Acuerdos de trabajo

- 2.4.1 Documento Visión
- 2.4.2 Proyecto técnico

2.5 Información de clientes

2.6 Informes

- 2.6.1 Diagnóstico

2.7 Reuniones

- 2.7.1 Minuta de reunión

2.8 Investigaciones

- 2.8.1 Diseño de Investigación
- 2.8.2 Estado del arte del producto a desarrollar
- 2.8.3 Reporte de Investigación

3. Soporte

3.1 Aseguramiento de la calidad

- 3.1.1 Glosario de Términos
- 3.1.2 Listas de chequeo
- 3.1.3 Plan aseguramiento de la calidad

3.1.4 Plan de mediciones

3.2 Gestión de configuración

3.2.1 Pedido de cambio

3.2.2 Plan Gestión de Configuración

3.2.3 Solicitud de cambio

2.2.1 PROPUESTA DEL SISTEMA DE GESTIÓN DE LA CALIDAD

En el PRV se han realizado varias propuestas para orientar el desarrollo de aplicaciones de RV, una de ellas es “Propuesta del Sistema de Gestión de la Calidad” [Domínguez, 2007] basada en la ISO⁶ 9001:2000. Donde se identifican algunos procesos claves para la producción de SRV dentro del PRV. Uno de estos procesos es el de diseño y desarrollo, el cual se resume en la figura 16 que a continuación se presenta.

⁶ Esta Norma Internacional proporciona una guía a las organizaciones para la aplicación de la Norma ISO 9001:2000 a la adquisición, suministro, desarrollo, operación y mantenimiento de software y los servicios de soporte técnico asociados. La misma no añade requisitos ni cambia los de la ISO 9001:2000. [ISO, 2006]

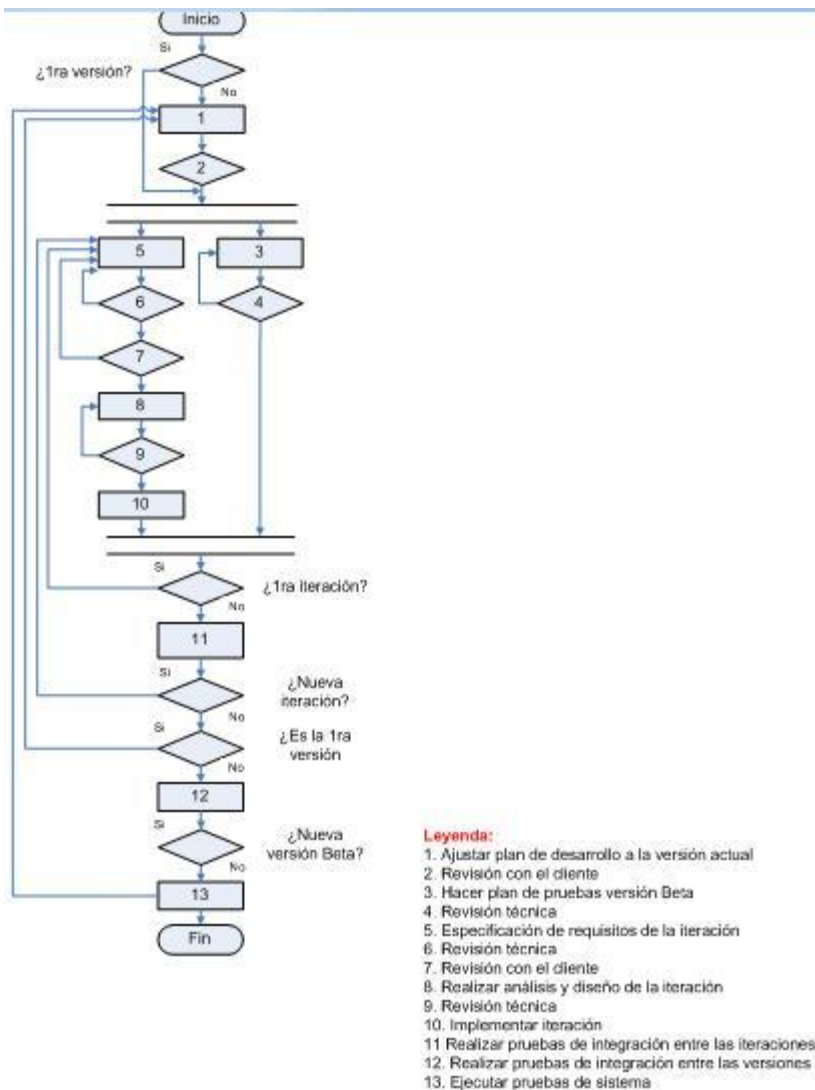


Figura 16. Proceso “Diseño y desarrollo”. [Domínguez, 2007]

2.3 APLICACIÓN DE ENCUESTAS Y ENTREVISTAS

La encuesta es un diseño o estrategia general no experimental, constituye una de las formas más utilizadas para indagar sobre las características principales de un segmento poblacional y/o para conocer las opiniones, experiencias y expectativas de grupos de personas. Esta técnica permite recabar una serie de datos en un amplio número de casos. [GUIDO, 2007]

La entrevista tiene características similares al cuestionario, siendo la principal diferencia el hecho de que es el encuestador u observador (los autores) quien anota las respuestas a las preguntas. Y que además, se le añade el uso de una grabadora para la posterior transcripción de los diálogos. Este instrumento conlleva una mayor habilidad por parte de los autores en conducir el tema de la entrevista, debido a que las respuestas son por lo general abiertas y permiten implementar nuevas preguntas no contempladas por el encuestador inicialmente. Esto adiciona la conveniencia de explotar temas que al principio no se tenían en cuenta o profundizar en algunos de los contemplados.

La aplicación de encuestas y entrevistas para la recopilación de la información proporciona un conocimiento de primera mano de la realidad, y contribuye a realizar mediciones precisas. Además de que permite obtener datos de grandes poblaciones, y recurre al lenguaje sobre temas complejos para explicar la información.

De acuerdo con la información que se quiere consultar en el presente trabajo, las encuestas aplicadas pueden ser de:

- Expectativas: Indagan sobre las necesidades y los intereses fundamentalmente del PRV para ajustar su proceso de desarrollo a los SRV.
- Satisfacción: Miden la opinión sobre la guía propuesta y relevan la experiencia de los proyectos durante su adaptación.
- Índice de satisfacción: Se indaga sobre una serie de temas que posteriormente, se traducen en un puntaje que permite calificar el desempeño de los proyectos principalmente dentro del PRV, desde sus experiencias.

2.3.1 FORMULACIÓN DE OBJETIVOS

Con este estudio investigativo, mediante la aplicación de las encuestas y entrevistas se desea lograr:

- Conocer cuáles son las metodologías usadas en la Facultad 5 para SW de RV.
- Conocer con que precisión se maneja cada metodología de forma independiente en los proyectos de RV.
- Conocer que documentación se lleva a cabo en cada proyecto según el tipo de metodología.
- Tener una idea de la efectividad y cumplimiento de los proyectos según el tipo de metodología.

- Establecer segmentos de población según la experiencia y la calificación de los jefes de proyectos.

De manera general, se desea fundamentalmente profundizar en los temas sobre los cuales se desea indagar, y lograr un mayor acercamiento al modo de producción dentro del PRV.

2.3.2 DISEÑO DE LA MUESTRA

Para definir la manera de acceder al grupo de personas que será encuestada, primeramente se ha realizado un censo dentro de la Facultad 5. Dirigido a conocer la población de proyectos que se desarrollan y quiénes son sus líderes. A través del censo se pudo identificar que la Facultad 5 cuenta con 3 proyectos de Hardware (HW) y Automática (SCADA, Supervisión Energética e Instrumentación Virtual), 3 horizontales (Calidad, Arquitectura y Operación Verdad), 2 de Innovación Pedagógica (Superación del Claustro y ProDiMat) y 14 de Realidad Virtual. Para una totalidad de universo de 22 proyectos.

Continuamente se procedió a escoger una muestra de la población. La misma está compuesta por todos los proyectos de RV; los cuales, pertenecen al PRV y están divididos en las diferentes AT que se mencionan a continuación:

Visualización

- Simulador Quirúrgico.
- Realidad Aumentada.
- Laboratorios Virtuales.

Video Juegos

- Compilación de juegos.
- Juegos CNEURO.
- Juegos ICAIC.

Simuladores

- Juegos de Consola.
- Simulador de Auto.

- Entrenadores Aduana
- Herramientas de Desarrollo para Sistemas de Realidad Virtual (HDSRV)
- Simulador de Tiro

Diseño y Animación

- Meñique
- Paseos Virtuales
- Escenarios Virtuales

Es importante señalar que este muestreo tiene un carácter probabilístico. Pues otorga una probabilidad conocida, no nula, de integrar la muestra a cada uno de los proyectos de análisis de la Facultad 5.

2.3.3 CONFECCIÓN DEL CUESTIONARIO

El cuestionario presentado en el anexo 1, fue elaborado por los autores de la presente investigación, unido a los asesores. Estará dirigido fundamentalmente a los líderes de los proyectos productivos del PRV. Cuenta con 7 preguntas entre cerradas (donde ya están dadas de antemano las posibles respuestas) y abiertas (en las cuales los encuestados tienen la posibilidad de ofrecer cualquier respuesta que consideren necesaria). Dichas preguntas se exponen claramente y sin ambigüedades, para que las personas encuestadas puedan responder con toda la sinceridad y coherencia posible. Se ha calculado que el cuestionario puede ser llenado satisfactoriamente en un tiempo de 30 min.

Con el cuestionario se desea identificar junto con los líderes de proyectos, las oportunidades sustantivas de mejora que pueda haber dentro del proceso de desarrollo de software del PRV en la Facultad 5. De ser implementadas estas mejoras, se influenciaría de manera muy positiva a todos los demás procesos de negocio.

En la encuesta (presentada en el anexo 1), se realizaron preguntas para conocer qué metodologías usan generalmente los proyectos, y cómo se lleva a cabo el proceso de producción en ellos. Así como la documentación u otros artefactos generados, los roles, las etapas y actividades principales que intervienen para tener una idea del manejo de las mismas.

Cabe destacar que la confección de las encuestas y entrevistas aquí realizadas, está estructurada en cuanto a nivel de información, grado de satisfacción, expectativa global, posición de la guía propuesta en comparación a otras, sugerencias de mejora y datos de clasificación.

2.3.4 RECOLECCIÓN DE LOS DATOS

Para realizar la recolección de la información se aplicaron fundamentalmente los siguientes métodos de encuestas:

- Personales: Implican formular preguntas directas cara a cara a algunos grupos de desarrollo de proyectos y líderes dentro del PRV.
- Telefónicas: Implican el contacto telefónico con algunos líderes o miembros de proyectos seleccionados, así como a otros jefes del PRV.
- Auto-administradas: Supone el llenado del cuestionario por parte de los encuestados sin asistencia de encuestadores.
- Vía e-mail: Aprovechando la tecnología con que cuenta la UCI, lo cual resulta de bajo costo y fácil implementación.
- Ad-hoc: Supone el llenado de manera voluntaria por parte de los encuestados.

Además de las encuestas y entrevistas efectuadas, se realizó también un estudio cuidadoso de los proyectos mediante los expedientes de proyectos, y algunos datos adicionales que se ofrecen en la Infraestructura Productiva [6] de la UCI.

2.3.5 ANÁLISIS Y PRESENTACIÓN DE LOS RESULTADOS

A continuación se realiza un análisis de los resultados obtenidos de la revisión de los expedientes de proyecto, también de las encuestas y entrevistas realizadas a los líderes o miembros importantes de los proyectos productivos del PRV, acerca de los elementos fundamentales que se deben tener en cuenta para elaborar una guía de desarrollo de SW ajustado a SRV.

Metodologías.

Actualmente, el 64% de los proyectos del PRV trabajan sobre la base de RUP. El 21% de las aplicaciones del Polo, pertenece al AT de Diseño y Animación. Donde solo el proyecto Paseos Virtuales utiliza RUP para la realización del SW Multimedia, las aplicaciones restantes en esta área no emplean una metodología definida para su desarrollo. El resto son los proyectos Juegos ICAIC y Laboratorios Virtuales, que utilizan como proceso de desarrollo OpenUP y UPEDU respectivamente. La gráfica siguiente, muestra las metodologías empleadas en las diferentes AT del Polo, a partir de la información presentada en la tabla 1 del anexo 3.

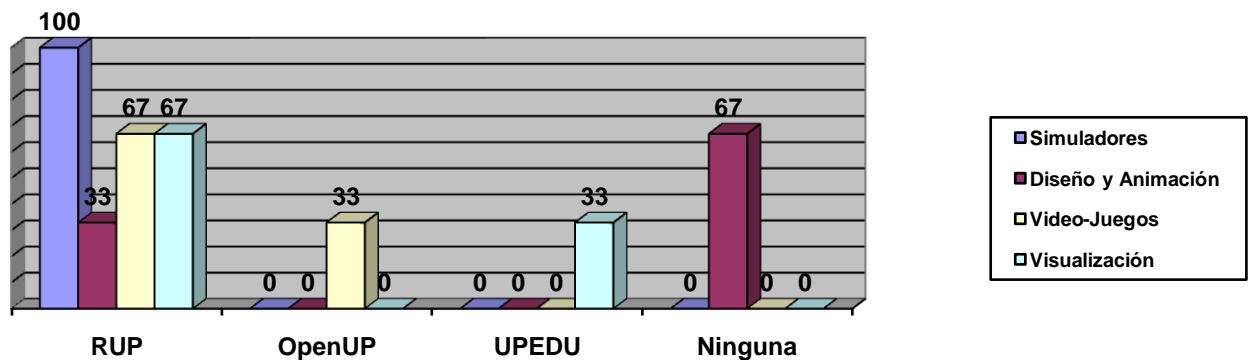


Figura 17. Empleo de las metodologías en las AT del Polo.

Según estudios realizados, se ha empleado MSF Ágil (Microsoft Solutions Framework Agile) para el desarrollo de SW en el proyecto Juegos de Consola. También el proyecto Paseos Virtuales ha guiado su proceso de desarrollo con prácticas de XP, utilizando UML para la comprensión de determinada funcionalidad. Cabe destacar que existe una gran tendencia al empleo de OpenUP en la Facultad 5, pues, este Proceso Unificado ágil es capaz de adaptarse muy bien, y de integrarse fácilmente con otras metodologías como XP o Scrum. Debido a que fusiona las prácticas y principios de un Proceso Unificado conocido como RUP, con la filosofía del desarrollo ágil.

Etapas y disciplinas.

Las aplicaciones del Polo dividen sus etapas en las que define la metodología que usan, pero modifican las iteraciones y los hitos de acuerdo a sus necesidades. Con el objetivo principal de obtener un producto entregable con calidad, en un período corto de tiempo. Se ha observado también, que de todos los

proyectos que tienen definida una metodología de desarrollo para sus soluciones, solo algunos utilizan Modelamiento del Negocio, más de la mitad realizan Levantamiento de requisitos, y Despliegue. Además todos realizan el Análisis-Diseño (valorando si es necesario o no, realizar el análisis), Implementación y Prueba. Por otra parte, las actividades de protección se reducen a 2 fundamentales: Administración de cambios/configuración, y Administración de proyecto, las cuales se realizan en gran medida en los proyectos del PRV, y se efectúan durante casi todo el ciclo de vida de una aplicación. A partir de la tabla 2 del anexo 3, se pudo calcular el porcentaje (mostrado en la tabla 2) en que se emplean estas disciplinas en el PRV.

Disciplinas	% Empleado	% No empleado
Implementación	100%	0%
Prueba	100%	0%
Análisis-Diseño	100%	0%
Administración de proyecto	100%	0%
Administración de cambios/configuración	86%	14%
Levantamiento de requisitos	79%	21%
Despliegue	50%	50%
Modelamiento del negocio	36%	64%

Tabla #2. Presentación de resultados en cuanto a disciplinas empleada en el PRV.

En el caso del proyecto Meñique, debido a que sus entregables son artefactos de diseño 3D que se realizan a pedido del cliente (que puede ser también cualquier proyecto del PRV), las actividades a realizar están enfocadas específicamente al desarrollo de 3 líneas principales: Realización de Setup de personajes, programación Script, realización de Texturas y Materiales. El proyecto Escenarios Virtuales es un caso similar a Meñique, pues sus entregables son modelos 3D y animaciones para el PRV. Por lo que sus actividades estarán dirigidas a la realización de estos artefactos. Los estudios realizados indican que las tareas a realizar en cada uno de los flujos de trabajo, no necesariamente son las mismas para todos los proyectos del Polo. Se ajustan en dependencia del tipo de sistema que se quiere desarrollar, o de las características del proyecto específico.

Roles.

Otro de los problemas que atentan contra las aplicaciones en su proceso de desarrollo, es la asignación de roles, y la determinación de responsabilidades. Pues aún en algunos proyectos del Polo, existen roles solapados y personalizados. No se hace pleno uso de los roles definidos para el Polo, lo que influye en la

desorientación del proceso de selección del personal, en la calidad del software y en su tiempo de entrega. En este caso, se realizó un análisis de los roles en cada uno de los proyectos de RV, mediante la comparación de las funciones que estos realizan con las de la metodología utilizada en el proyecto. Verificando principalmente si coincide cada responsabilidad del rol, con las planteadas por la metodología. Además, si queda alguna tarea por realizar, o si es realizada por algún rol que no es el más indicado para desempeñar la labor debido a que tiene muchas responsabilidades. También se tuvo en cuenta las experiencias adquiridas por los líderes de proyectos, mediante las entrevistas realizadas.

Con este detallado análisis se han determinado algunos roles que son significativos en la producción de SW en el PRV, y que también podrían servir de guía para organizar de manera más eficiente la selección de personal en los proyectos. Además se ha descubierto que en un mismo proyecto, o en proyectos diferentes, se asignan muchos roles con las mismas responsabilidades, y con nombres diferentes. Esto constituye formas de personalización, y solapamiento. Éste es el caso de los roles: Administrador de Proyecto, Analista del Sistema, Administrador de control de cambios, Diseñador de Pruebas, Revisor, y Analista de Pruebas. Los cuales están propuestos por RUP, y son utilizados en varios proyectos del PRV para representar las responsabilidades de los roles: Líder de Proyecto, Arquitecto de software, Administrador de Configuración, Diseñador de Casos de Pruebas, Auditor de la Calidad, y Revisor Técnico Formal respectivamente. Todos estos roles ya están definidos por el Polo.

Otro rol que utiliza muchas acepciones es el de Diseñador, definido por RUP. Éste es asignado en diferentes proyectos del PRV como modelador, Realizador 3D, Diseñador de escena, Desarrollador de Texturas y Materiales, así como Diseñador gráfico. Lo mismo sucede con los roles Arquitectura y Tecnologías, Escritor Técnico, y Planificador de la calidad, los cuales ya están definidos por el Polo. El primero de ellos es asignado en algunos proyectos como Arquitecto de Información, el segundo como Documentador, y el tercero como Ingeniero de Pruebas, y Administrador de la calidad.

Se ha observado además, que el modo de producción en el PRV adquiere algunas diferencias entre las AT, debido a las características y necesidades especiales que posee el desarrollo de los sistemas y servicios en cada una de ellas. Esto influye considerablemente en la creación de equipos de trabajo, y en la determinación de roles, responsabilidades y conocimientos necesarios para el proceso de producción de SRV en el Polo. Muchos proyectos han evidenciado la necesidad de definir un rol que represente a la(s) persona(s) especializada(s), o con autoridad para guiar o llevar a cabo tareas exclusivas. El rol

Especialista Funcional, es un asesor especializado capaz de proveer información valiosa para el proyecto. Puede ser asignado, por ejemplo, a la(s) persona(s) especialista(s) en psicología (A través del rol Psicólogo), o al Grafista⁷. De la misma manera, algunos proyectos han percibido que sería útil la utilización de un Rol Adicional. El cual se encargaría de las tareas que puede efectuar cualquier miembro del equipo, por ejemplo, este rol puede ser asignado a la(s) persona(s) responsables de velar por los resultados docentes-productivos de los estudiantes del proyecto, a través del Trabajo Educativo.

Además, con la necesidad de crear nuevos grupos de trabajo, encargados de la producción de otras líneas como la textura, setup de personajes, video, sonido, animación, y efectos especiales. En algunos proyectos, principalmente en el AT de Diseño y Animación, se han definido los roles de Texturizador, Desarrollador Setup de Personajes, Editor, Sonidista, Animador, Desarrollador de Efectos Especiales; que ayudarían a la orientación y realización de estos productos. Por otra parte, existen roles que aunque básicamente son asignados para desempeñar las actividades para las que están destinados, en algunos proyectos o AT, adquieren más o menos responsabilidades que en otros(as), o requieren de más o menos conocimientos para poder realizar sus tareas. Éste es el caso del rol Analista de Software, el cual se puede desempeñar como Guionista en determinados proyectos que utilicen guiones (técnico, y de Contenido) en vez de descripción de Caso de Uso (CU), para especificar los requerimientos, y definir las tareas o funcionalidades del sistema. Lo mismo sucede con el Programador, que toma las responsabilidades del Programador Script en el proyecto Meñique. Además del rol Integrador, que en el caso específico del AT de Video-Juegos, es utilizado como Productor⁸.

La existencia de roles genéricos se ha observado sobre todo en el AT de Diseño y Animación, pues para cada grupo encargado de una línea de producción específica, aparece un Jefe de Equipo. Por ejemplo, para los grupos de desarrollo de texturas, diseños (modelos), animaciones, y videos; algunos proyectos utilizan el rol Jefe de Texturizado, Jefe de Modelación, Jefe de Animación, y Director respectivamente. Muchas veces estos roles son personalizados, ya que representan a solo una persona encargada de guiar y producir en el desarrollo de algunos artefactos en un SRV. En la figura 18 se muestra el rol Jefe de Diseño y Animación, el cual puede asumir las responsabilidades de un Jefe de línea de desarrollo en un

⁷ Suele ser un licenciado en Bellas Artes, trabajan básicamente con dibujo en plano, con tecnología tradicional o en tecnología digital.

⁸ Encargado de integrar todas las partes del videojuego. En este caso, no es necesario que sean programadores experimentados, pues sólo se requiere que tengan los conocimientos básicos de todas las áreas para poder integrar todo el trabajo.

proyecto determinado. También el rol Diseñador puede tomar las labores del Jefe de Texturas, o Jefe de Modelado según lo requiera el proyecto que vaya a utilizarlo.

El análisis anterior se representa a continuación en una estructura organizativa de roles (presentado en la figura 18), que sirva de base en la orientación del proceso de asignación de roles y responsabilidades a la propuesta de guía de desarrollo para los SRV del Polo. También puede servir de apoyo para definir la estructura organizativa de cualquier proyecto del PRV.

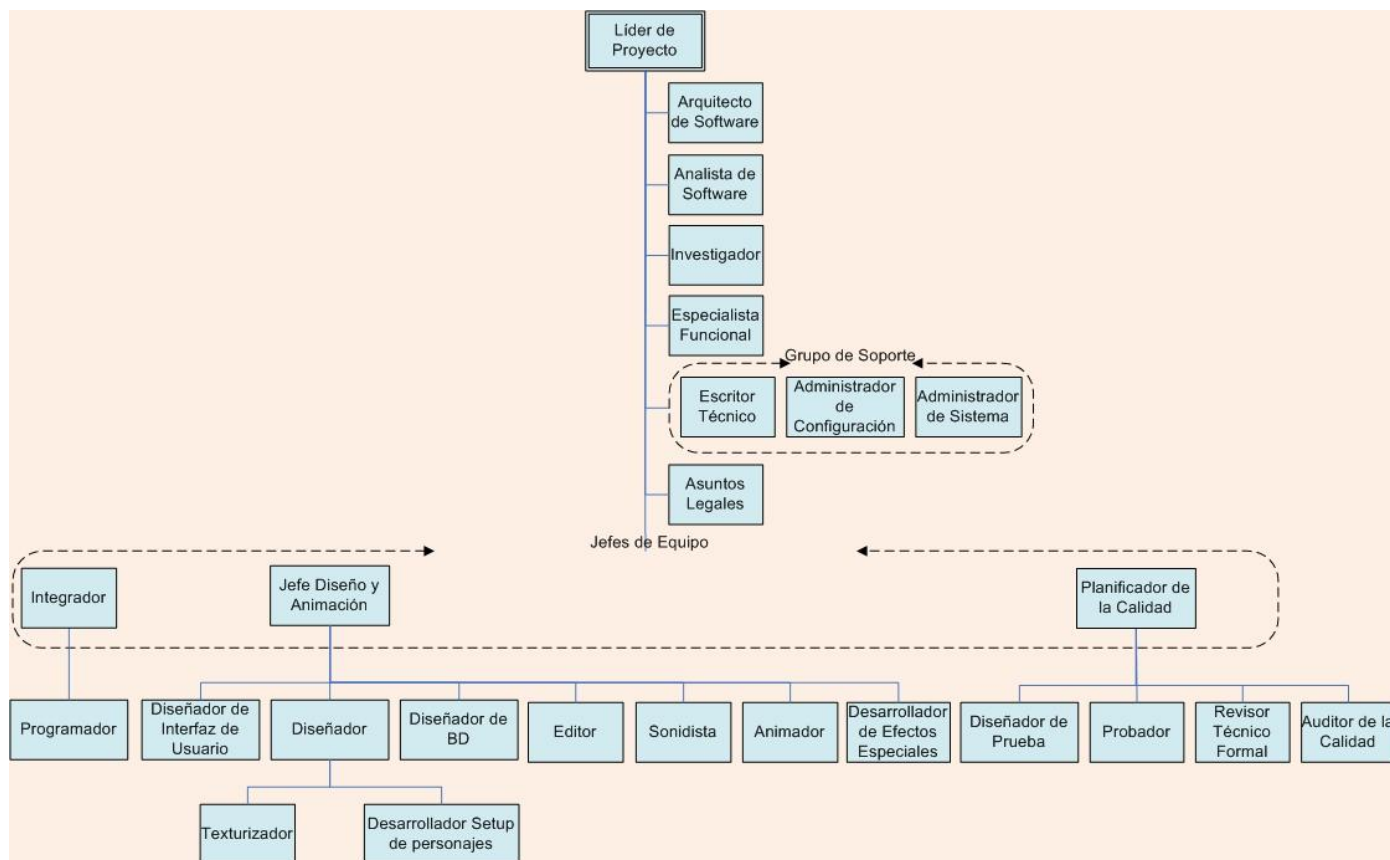


Figura 18. Primera versión de Estructura Organizativa de Roles.

A partir de la tabla 3 del anexo 3, se construye la tabla 3 con los roles representados, y analizados anteriormente en la figura 18; en la cual se observa el porcentaje empleado en los proyectos del PRV. Entre los roles que más han sido empleados en el Polo se pueden encontrar: Líder de proyecto, Arquitecto de software, Analista de software, Investigador, Administrador de Configuración, Programador, Planificador de la Calidad, y Diseñador, así como también son usados en menor medida el Escritor técnico,

Administrador de Sistema, Diseñador de Prueba, Probador, Revisor Técnico Formal, Auditor de la Calidad, el Especialista Funcional, Integrador, y Diseñador de Interfaz de Usuario. Otros roles no dejan de ser importantes, por ser utilizados en pocos proyectos del PRV, éste es el caso del Rol Adicional, Sonidista, y Diseñador de BD. Así como el Texturizador, y Desarrollador Setup de personajes; que sólo son definidos en el proyecto Escenarios Virtuales, y Meñique individualmente. El rol Jefe de Diseño y Animación, representado en la figura 18, fue creado con la intención principal de eliminar la personalización de roles que puede causar el rol Jefe de Equipo en los grupos de esta área. Aunque actualmente los roles Editor, Animador, y Desarrollador de Efectos Especiales; nos son concretados como tal en los proyectos analizados en la muestra. No obstante, sí han sido definidos por el AT de Diseño y Animación, utilizados en el desarrollo de sistemas o productos pasados, y tenidos en cuenta para proyectos futuros.

Roles	% Empleado	% No empleado
Líder de proyecto	100%	0%
Programador	86%	14%
Arquitecto de software	79%	21%
Analista de software	71%	29%
Administrador de Configuración	64%	36%
Planificador de la Calidad	57%	43%
Investigador	50%	50%
Diseñador	50%	50%
Escritor técnico	43%	57%
Administrador de Sistema	43%	57%
Diseñador de Prueba	43%	57%
Probador	43%	57%
Revisor Técnico Formal	43%	57%
Auditor de la Calidad	43%	57%
Especialista Funcional	29%	71%
Integrador	29%	71%
Diseñador de Interfaz de Usuario	29%	71%
Diseñador de BD	14%	86%
Sonidista	14%	86%
Rol Adicional	14%	86%

Tabla #3. Presentación de resultados en cuanto a roles empleados en el PRV.

Puede ser muy productivo el hecho de que los proyectos del PRV determinen sus roles basados en los definidos para el Polo, así como la determinación de responsabilidades y conocimientos con una correcta justificación, de acuerdo a las características, necesidades, e intereses de cada sistema o producto que se desarrolle. Es cierto que no existe una metodología estricta a seguir, sino un proceso donde se persiguen

las mejores prácticas que se necesitan para el proyecto; pero si se utiliza una metodología, las responsabilidades de los roles que se definen en el proyecto deben estar conforme a las que ella plantea.

Artefactos y documentación.

En el análisis de los artefactos y la documentación generada en las soluciones del PRV, además de los resultados obtenidos en las encuestas y entrevistas realizadas, se tuvo en cuenta fundamentalmente las plantillas organizadas en la parte de ingeniería del expediente de proyecto. En la tabla 4 del anexo 3, se puede observar cómo se emplean dichas plantillas para la generación de la documentación en cada uno de los proyectos del Polo; las mismas son representadas en porcentaje de empleadas en la tabla 4. En la captura de los requisitos, se utiliza la Especificación de requisitos, y el Plan de Gestión de requisitos; mientras que el Modelo del Dominio, y el Modelo de Negocio se efectúan con menor medida. El Diagrama de Proceso – Nombre de proceso, sólo se realiza en el proyecto HDSRV. En otros pocos proyectos del PRV, se efectúa la Evaluación de Áreas de la Organización. Para la arquitectura y diseño, el Documento de Arquitectura de Software, y Modelo de Diseño es esencial; teniendo un gran empleo en los proyectos del Polo. La Arquitectura de Información (AI) sólo se tiene en cuenta en los proyectos Simulador de Auto y Compilación de Juegos. Durante la implementación y prueba, se genera Código fuente, Manual de Usuario, Diseño casos de prueba, y Plan de pruebas en más de la mitad de los proyectos.

En el Polo, algunos proyectos generan artefactos o productos de trabajo como consecuencia de sus necesidades, intereses, y características propias del proceso de desarrollo. Tal es el caso de Escenario Virtuales y Meñique, los cuales no se rigen por alguna metodología para generar productos de trabajo. Además, la documentación de Guión de contenidos, y Guión técnico son utilizados en pocos proyectos del PRV, la mayor parte de ellos pertenece al AT de Videojuegos. Muchas de las plantillas del expediente de proyecto analizadas, son manipuladas insuficientemente en los proyectos del Polo. Debido a que no están ajustadas al Polo, y no se corresponden con los procesos básicos para la construcción de los SRV.

Elementos de trabajo	% Empleado	% No empleado
Plantilla Plan de Gestión de requisitos	86%	14%
Plantilla Especificación de requisitos	86%	14%
Plantilla Modelo de Diseño	86%	14%
Plantilla Documento de Arquitectura de Software	71%	29%
Plantilla Diseño casos de prueba	71%	29%
Plantilla Plan de pruebas	57%	43%

Código fuente	57%	43%
Manual de Usuario	50%	50%
Manual de Usuario	50%	50%
Plantilla Modelo de Negocio	50%	50%
Plantilla Modelo del Dominio	43%	57%
Guión técnico	43%	57%
Guión de contenidos	43%	57%
Plantilla Evaluación de Áreas de la Organización	29%	71%

Tabla #4. Presentación de resultados en cuanto a los elementos de trabajo empleados en el PRV.

Valoración general.

La mayoría de los proyectos productivos del PRV, utilizan RUP como metodología básica en sus procesos. Por ser la metodología que más conocen los líderes de proyectos, también por su versatilidad y flexibilidad. Por otra parte, la Dirección de la Calidad del Software (DCS) en la UCI realiza la auditoría basada en el proceso de RUP, el cual es considerado por esta Dirección como el más idóneo para la producción de SW en la Universidad. Esto ha traído consigo algunas dificultades, debido a que la manera de llevar a cabo el proceso en el PRV no se ajusta adecuadamente a su modo producción en general. Los líderes de algunos proyectos han tenido que adoptar prácticas de otras metodologías como XP, OpenUP, UPEDU y otras; buscando alternativas para desarrollar sus productos con calidad.

En la mayoría de las aplicaciones del Polo, el uso de una metodología fuerte como RUP implica evadir o modificar algunas actividades, artefactos, y documentación que sólo traen consigo pérdida de tiempo y recursos. Por otro lado, uno de los rasgos que inclinan al uso de metodologías ágiles, es la existencia de equipos de trabajo pequeños. Además, algunos proyectos del PRV se caracterizan por ser breves, con fechas de entregas cortas del producto que cumpla con todos los requerimientos del cliente. Esto trae como consecuencia un aceleramiento del proceso en el modo de producción del Polo, sin que se vea afectada la productividad, eficiencia y calidad del producto.

El cumplimiento estricto de algunas metodologías ágiles como XP, tampoco satisfacen todas las expectativas del marco de trabajo del PRV. Esto es debido a que no se cuenta con las condiciones máximas para su implementación. En muchos casos cuesta esfuerzo hacer que el cliente forme parte del equipo de desarrollo, y reunir a los miembros del proyecto como lo exige el Desarrollo Rápido de Aplicaciones (DRA). Esto se debe a que existen equipos de trabajo dispersos, en proyectos del Polo. Además, los miembros del equipo de desarrollo en su mayoría son estudiantes. No cuentan con la

experiencia necesaria, y no están especialmente preparados para cambios durante el proyecto. A pesar de que en los proyectos del Polo no se despliega ni se almacena toda la documentación que RUP propone, y que además existe un mal manejo de los artefactos definidos en cada flujo de trabajo. En el PRV se continúa trabajando bajo la base teórica de RUP, adoptando prácticas de las metodologías ágiles según sean las necesidades y los cambios variantes que se presenten en el desarrollo de los proyectos en el Polo.

El empleo de UML ha sido un factor muy importante para el modelado de las aplicaciones de RV. UML es el lenguaje estándar de modelado que le ha servido al PRV para la especificación, documentación, construcción, y visualización de los artefactos de los SRV en los que el SW juega un papel importante. Este lenguaje ha permitido a los desarrolladores, visualizar los resultados de su trabajo en esquemas o diagramas estandarizados. Esto le ha propiciado al PRV comunicar sus modelos, no solo entre los miembros de los proyectos, sino a todas las personas involucradas en él, y a los desarrolladores de las futuras generaciones. Por lo que en el Polo se necesita mucho de este lenguaje no solo para comunicarse, sino también para proporcionar un marco en el que los desarrolladores individuales pueden analizar y pensar. Independientemente de lo planteado, también UML necesita ser ajustado a los SRV desarrollados por el Polo. Pues el uso de este lenguaje ha sido indiscriminado e incorrecto, debido a la falta de experiencia en su utilización para la producción de las aplicaciones del PRV; las cuales tienen exigencias un tanto particular. Además de la falta de conocimiento práctico por parte de los desarrolladores.

De manera general, en dependencia del AT al que pertenece, el desarrollo de los proyectos del PRV tiende a poseer algunas diferencias entre sí. Las investigaciones realizadas indican que las etapas, actividades, documentación, los roles y artefactos utilizados para llevar a cabo un producto en cada uno de estos proyectos, deben ser ajustados a sus características específicas.

CONCLUSIONES PARCIALES

Después del análisis detallado en este capítulo, se llegan a las siguientes conclusiones:

- El PRV, carece de profundidad en la comprensión de las metodologías usadas por los proyectos, así como de una guía para el proceso de desarrollo de los SRV.
- Las etapas, actividades, herramientas, documentación, los roles, y artefactos utilizados para llevar a cabo un producto en el PRV, deben ser ajustados a las características, necesidades, e intereses del proyecto que lo desarrolla.
- El modo de producción del PRV, da evidencias suficientes para un desarrollo basado en reutilización, y sugiere la definición de nuevos roles, artefactos, actividades u otros elementos para su proceso de desarrollo de SRV.
- Se pudo conocer las principales deficiencias, e inquietudes que existen en el PRV.
- Se pudo observar más de cerca el proceso actualmente llevado a cabo, en cada uno de los proyectos productivos del PRV.

INTRODUCCIÓN

El uso de un proceso ajustado a los SRV, es un tema que alcanza mucha importancia dentro del PRV. El actual Capítulo se dedicará al ajuste de una “Guía de desarrollo de SW para los SRV” (MERVI) del PRV en la Facultad 5 de la UCI, tomando como antecedentes el análisis y descripción de los capítulos anteriores. Se identifican los elementos de las metodologías, modelos, y métodos de trabajo examinados; que pueden ser aplicables al desarrollo de los proyectos en el PRV, y que contribuyan al mejoramiento del proceso que actualmente se lleva a cabo en el Polo. Dicha guía describirá elementos adaptables al proceso que debe llevarse a cabo en el PRV, para el desarrollo de sus productos. Teniendo en cuenta la existencia de principios que promuevan prácticas adaptativas. Que sea cubierto todo el ciclo de desarrollo, además de la eficiente verificación, planificación, y control. Así como la comunicación efectiva, la puesta en práctica sobre los proyectos del Polo, la formación y documentación apropiada. También la utilización conveniente de las actividades, el mantenimiento, y la reutilización de SW.

3.1. DESCRIPCIÓN DE LA PROPUESTA

MERVI describe quién hace qué y cuándo en un proceso de ISW, que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro del PRV. El objetivo esencial es garantizar la producción de aplicaciones de RV de alta calidad, que satisfaga las necesidades de sus usuarios finales, dentro de un calendario, y presupuesto previsible.

La idea base para esta guía, parte de las características del modo de producción actual del PRV. Ya que está dividido por AT, y que cada área se dedica a la producción de productos con afinidades comunes. Sería propicio pensar en un desarrollo de aplicaciones para el Polo basado en Líneas de Productos de Software (LPS).

Una LPS es un conjunto de sistemas de software que comparten aspectos comunes y gestionados, que satisfacen las necesidades específicas de un segmento de mercado o misión, y que son desarrollados a partir de un conjunto común de activos fundamentales de software de una manera prescrita. [Clements and Northrop, 2002]

De modo que el PRV, es una LPS de RV; ya que se gestionan, y desarrollan familias de aplicaciones de RV; con características, funcionalidades, necesidades, e intereses comunes, así como algunas que otras diferencias. Cada AT genera una familia de productos de RV a partir de activos⁹ de SW, y asociadas a un dominio¹⁰ determinado. Lo que significa que un proyecto perteneciente a una LPS específica, se puede dedicar a la producción de otras líneas más concretas, donde la solución a un problema común, está dada por el uso repetido de un conjunto común de activos reutilizables. Por lo que el desarrollo de SW basado en reutilización, es una parte imprescindible de la guía que a continuación se describe.

3.1.1 PRINCIPIOS Y PRÁCTICAS

La propuesta se basa en los principios siguientes:

1. Cooperación: Para colaborar entre los equipos, alineando los intereses, y logrando una visión compartida. Mediante la promoción de una comunicación abierta, metáfora del sistema, diseño simple, propiedad colectiva, trabajo en conjunto, y las convenciones establecidas.
2. Estabilidad: Para lograr el equilibrio y fortalecimiento de los miembros en el equipo, confrontar las prioridades, maximizar el valor para los interesados, mantenerse dinámico y adaptarse al cambio. Mediante la asociación de los miembros del equipo, clientes, e interesados. Con el establecimiento de responsabilidades claras, y compartidas. Además de la creación de oportunidades para la obtención de conocimientos, y el aprendizaje de todas las experiencias.
3. Perfeccionamiento: Para continuamente ajustar el proceso, demostrar resultados de forma iterativa e incremental, obtener retroalimentación oportuna de los clientes, reducir riesgo, y centrarse en la calidad. Mediante la evolución continua, elevando el nivel de abstracción, y llevando a cabo un juego de planificación, refactorizaciones, integraciones continuas, desarrollo guiado por pruebas, y pequeñas liberaciones. La calidad, también debe ser llevada a los planes y programas del proyecto.
4. Articulación: Para llevar a cabo un enfoque modular de la arquitectura del sistema, facilitando la colaboración técnica. Además de reducir los riesgos, y adaptarse al cambio, así como minimizar excesos, y trabajo extra. Tratando de lograr el bienestar de los trabajadores, a un paso sostenible.

⁹ Un conjunto de elementos de SW (requisitos, diseños, componentes, casos de prueba, etc.) que se configuran y componen de una manera determinada para obtener los productos de la LPS.

¹⁰ Es un área del conocimiento enfocado en la aplicación a productos de SW. Los dominios pueden agrupar otros dominios, y ocurren cuando se produce más beneficios reutilizar un activo en múltiples productos, que crear el activo para cada producto.

Estos principios se soportan entre sí, los cuales promueven las mejores prácticas de algunas metodologías examinadas en el primer capítulo.

3.1.2 MODELO DE PROCESO

MERVI se fundamenta sobre el desarrollo de LPS, tiene una rápida evolución iterativa; combinando el desarrollo incremental, espiral, y basado en reutilización. Con refinamiento, y aprendizaje continuo. La figura 19 es una representación del proceso referido en esta guía. El cual consta de cuatro fases (Pre-Producción, Ingeniería de Dominio, Ingeniería de Aplicación, y Post-Producción) a través de todo el ciclo. Las fases Ingeniería de Dominio (ID), e Ingeniería de Aplicación (IA), están enfocadas al desarrollo de SW basado en reutilización; y son procesos de negocio de una LPS. Cada fase contribuye al cumplimiento de un hito o meta, y radica en una o más iteraciones intercaladas (presentadas en la figura 20).

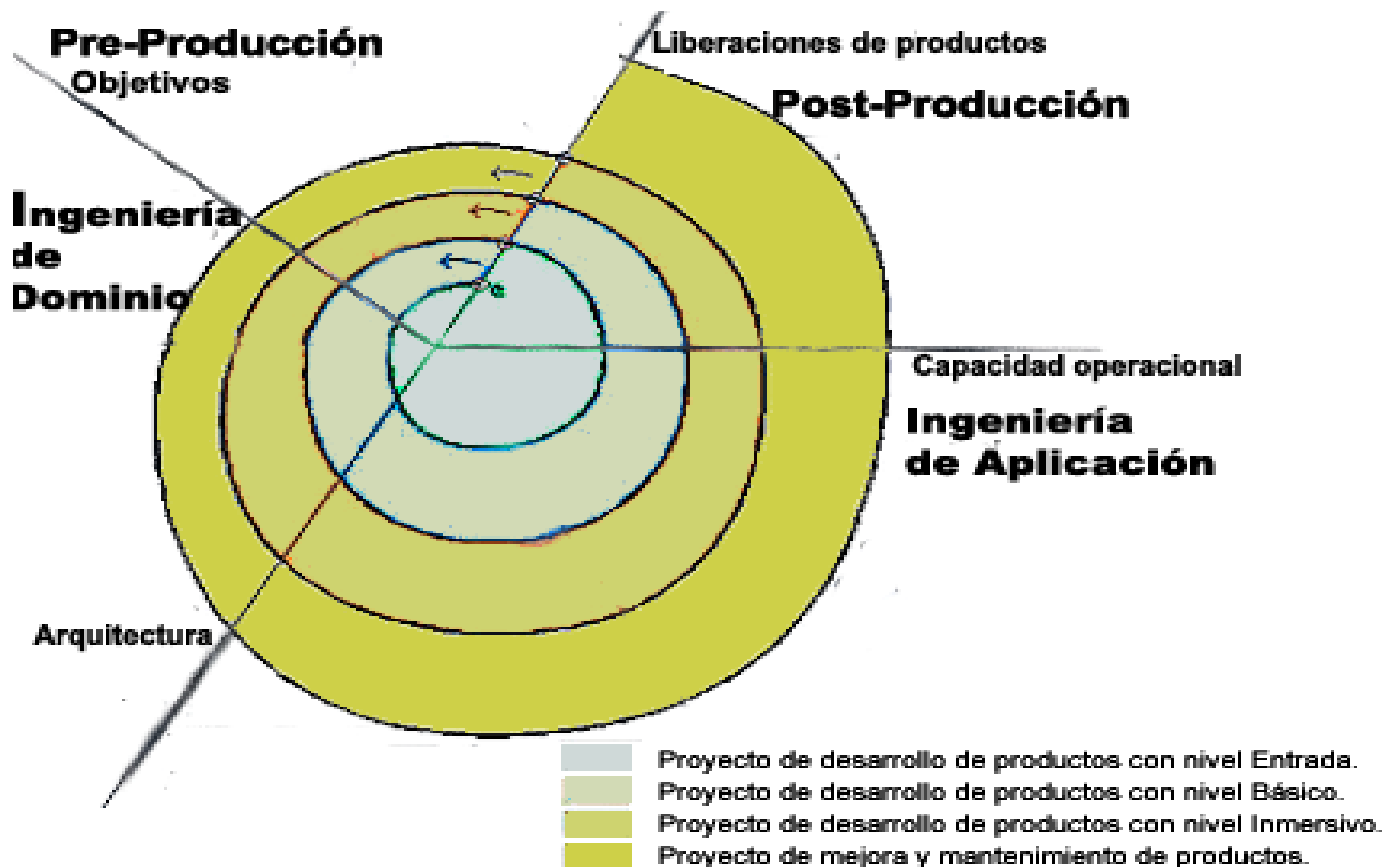


Figura 19. Modelo espiral de MERVI.

Cada iteración realiza micro-Incrementos del producto; dando como resultado, la evolución incremental de un prototipo del proyecto en cada fase. Las iteraciones tienen diferentes enfoques para abordar la liberación del producto, no obstante, estarán dirigidas al cumplimiento del hito de la fase que se esté realizando. Entre más pequeñas sean las iteraciones, más pequeño es el margen de error en sus estimaciones, y se proporciona una rápida retroalimentación acerca de la exactitud de los planes del proyecto. Al inicio del espiral, el equipo de trabajo gira a través de ciclos comenzando por el centro (marcado con ω en la figura 19), y en el sentido indicado. Un proyecto de desarrollo de producto, puede concluirse con el completamiento de un ciclo. La transición de un ciclo a otro, al igual que entre las fases, estará determinada por tres factores principales: Las necesidades del cliente o usuario final, las decisiones de gestión, y los resultados del proceso de aseguramiento de la calidad.



Figura 20. Iteraciones.

En la figura 21 se encuentran representados los responsables, y las posibles salidas más significativas de cada actividad en un diagrama de actividades. En el diagrama, algunos elementos aparecen con color amarillo. Estos representan algunos de los ajustes más significativos al proceso inherente de MERVI.

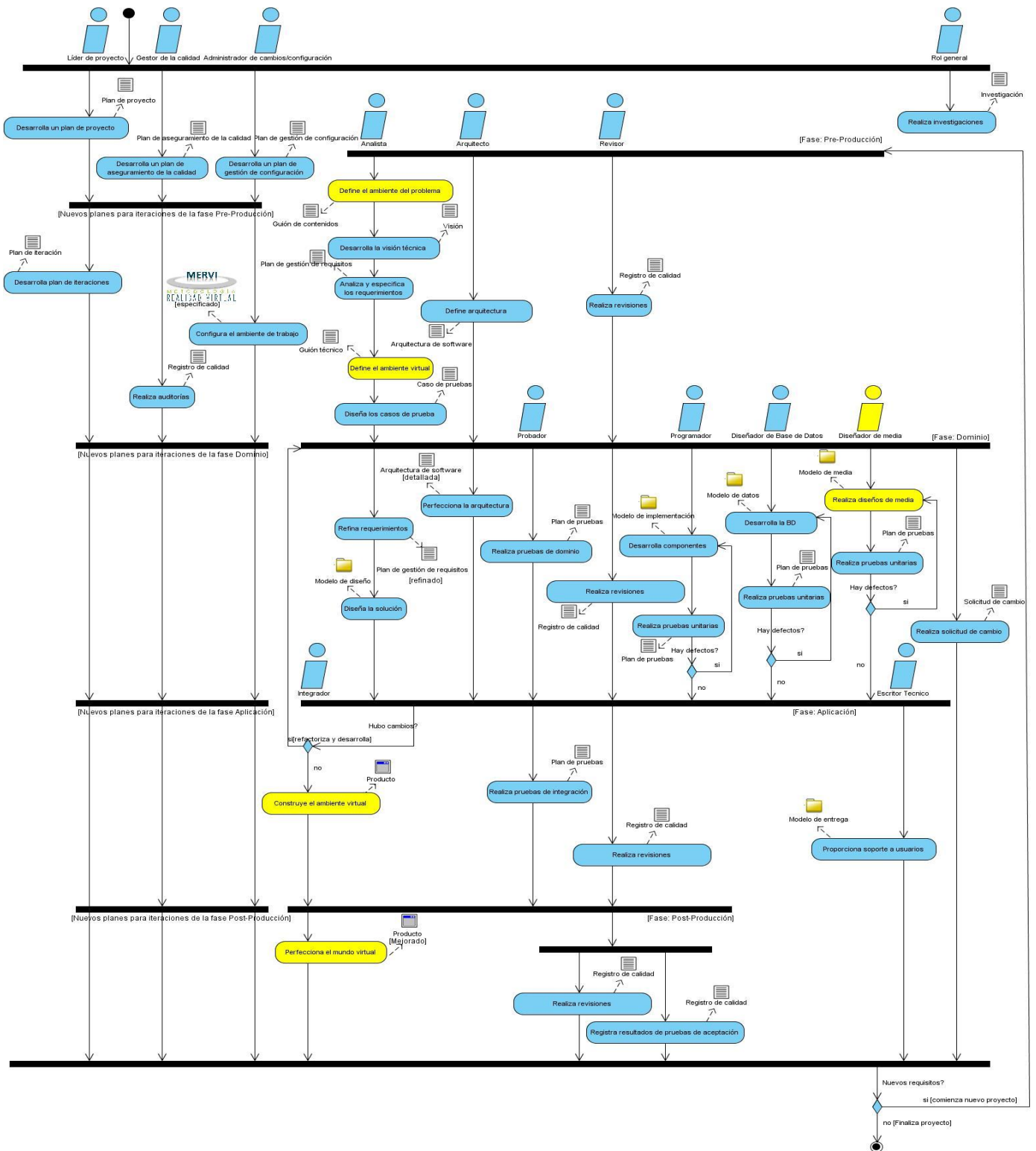


Figura 21. Diagrama de actividades del proceso referido por MERVI.

Durante todas las fases del ciclo, se realizan actividades del marco de trabajo (Ingeniería de requisitos, diseño y codificación, prueba, entrega). Estas actividades pueden tener mayor o menor influencia, en dependencia de la iteración en que se realice. En cambio, los flujos de apoyo (Gestión de proyecto, Administración de cambios/configuración, Aseguramiento de la Calidad) ejercen un gran peso en toda la evolución del proyecto. Ya que se mantienen efectuándose en todas las fases definidas, e iteraciones concebidas, con eficaz dinamismo. El correcto funcionamiento de estas disciplinas, proporciona una variación favorable a los proyectos de desarrollo de productos, durante todas sus fases en cuanto a riesgos (curva roja en el ciclo de vida del proyecto), y valor (curva verde en el ciclo de vida del proyecto). La figura 22, ofrece una vista general de todo el ciclo de vida incremental del proceso.

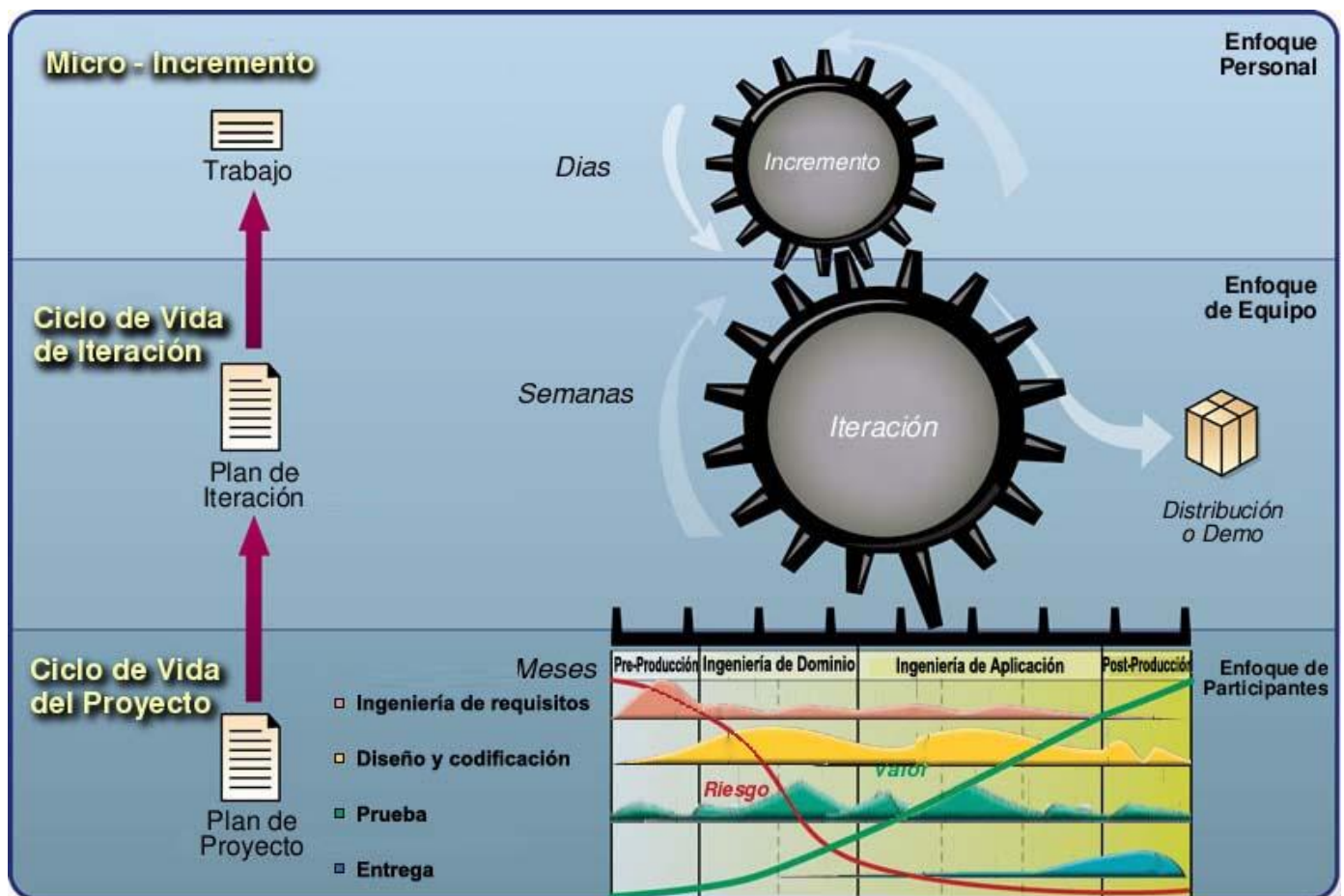


Figura 22. Ciclo de vida incremental de MERVI.

3.1.3 MODELO DE ROLES

Los roles no son individuos, sino una descripción de cómo un individuo o varios, trabajando juntos como un equipo, se comportan en el marco de trabajo, y qué responsabilidades tienen esas personas. De manera que un individuo puede ejercer varios roles, así como un rol puede ser representado por muchos individuos. Las responsabilidades se identifican como un conjunto de actividades que se lleva a cabo, o como el encargarse de la realización de un conjunto de productos de trabajo llamados artefactos.

Teniendo en cuenta las necesidades del PRV, así como los equipos de trabajo, roles, responsabilidades, y conocimientos de otras metodologías; MERVI propone un modelo abierto de equipos de trabajo (mostrado en la figura 23) con una visión común, responsabilidades claras, y comunicación abierta. Donde predomine el fortalecimiento de los integrantes de equipo en cuanto a conocimientos, y la colaboración entre ellos. Cada miembro es responsable de una parte de la calidad de la solución completa, representa las necesidades de sus compañeros, y ninguno es más importante que otro. Con estos puntos de vista se puede asegurar la cooperación, y estabilidad durante todo el ciclo del SW, y que el equipo produce la solución correcta.



Figura 23. Modelo de equipos de MERVI.

Como se puede observar en la figura anterior, los equipos de trabajo (Grupo funcional, Analistas, Calidad, Producción, Arquitectura, Dirección - Administración) están organizados en cuatro capas fundamentales de contenido:

1. Cooperación y Estabilidad: Constituye la columna vertebral de MERVI, y soporta al resto de las capas. Contiene a todos los miembros del proyecto, expresa la naturaleza colaborativa, comunicativa, de visión común y aprendizaje entre ellos. Esta área se basa principalmente sobre los principios nombrados de igual manera, los cuales soportan prácticas que ayudan al equipo a cooperar efectivamente, y a mantener el equilibrio de los proyectos del PRV en un ambiente educativo de producción.
2. Intención: Esta área representa a los equipos de trabajo cuyos miembros tienen la responsabilidad de conducir, así como traducir los propósitos del Cliente, y demás interesados, al resto de los integrantes del proyecto. También, para asegurar que todos los involucrados en la solución, comprendan correctamente los objetivos del proyecto. Asimismo, garantizar que los productos desarrollados, son un reflejo de los requerimientos del cliente.
3. Solución: En esta área se encuentran los equipos de desarrollo, encargados de realizar las actividades de ISW necesarias para llevar a cabo soluciones con calidad en el proyecto.
4. Gestión: Constituye los equipos encargados de las actividades de dirección, y administración del proyecto de SW. Esta capa garantiza que los conocimientos de todos los involucrados, se reflejen en un plan. Además de que todos los miembros estiman su trabajo, que cada uno de ellos asume una tarea específica, y es responsable de una parte de la calidad del producto.

Los equipos de trabajo y/o conjuntos de roles¹¹ que define MERVI, son los siguientes:

- Grupo funcional: Agrupa roles para personas que podrían ser de gran ayuda profesional, o cuyas necesidades deben ser satisfechas por el proyecto, y que podrían ejercerse por cualquiera que esté (o eventualmente estará) materialmente afectado por el resultado del proyecto. Tienen participación parcial o permanente en el proyecto, de acuerdo a las necesidades, y los intereses que existan. En

¹¹ Los conjuntos de roles se utilizan para agrupar roles que tienen habilidades comunes, y funcionan con técnicas similares, pero deben diferenciarse por el método de ISW. En cambio, un equipo de trabajo es un conjunto de personas, destinadas a lograr un mismo objetivo, con roles definidos. Por lo que en un equipo de trabajo, puede estar representado uno, o varios conjuntos de roles.

este grupo se encuentra el rol Interesado, el cual representa a los clientes, y usuarios finales implicados. También el rol Especialista funcional, que representa a otros roles, o persona(s) especializada(s), o con autoridad para guiar y llevar a cabo tareas exclusivas. Además, podrían ser asesores especializados que proporcionen información valiosa para el proyecto. Ejemplos de especialistas funcionales, es el Director artístico¹² y el Guionista¹³.

- Analistas: Obtienen información de los interesados para comprender el problema a ser solucionado, capturar, y ajustar las prioridades para los requerimientos. En el PRV, a este conjunto de roles se asocia el Analista.
- Calidad: Este grupo de roles es responsable de las tareas principales para garantizar la calidad del producto. Esto incluye identificar, definir, implementar, y dirigir las pruebas necesarias. También verificar, analizar, así como registrar los resultados de las pruebas, revisiones, y auditorías que se realicen en el proyecto. En este equipo de roles, se identifican los siguientes: Probador, y Revisor.
- Producción: En este conjunto de roles están representados aquellos que se encargan de analizar, y diseñar el dominio de la línea del producto. Así como diseñar, implementar, hacer pruebas unitarias e integrar los componentes de la aplicación. También son capaces de producir materiales adicionales que necesita el producto. Los roles identificados en este conjunto son: Escritor técnico, Integrador, Programador, Diseñador de Base de Datos (BD), y Diseñador de media. Este último, es un caso especial que se explica más adelante.
- Arquitectura: En este conjunto se encuentra laborando el rol de Arquitecto, el cual es responsable de diseñar la arquitectura del SW. Abarca asumir las fundamentales determinaciones técnicas que establecen generalmente el diseño, y la implementación del producto.
- Dirección – Administración: Constituye el equipo gestor, los roles que laboran en este conjunto se encargan de administrar, y configurar el proyecto (incluyendo el proceso de ISW empleado). Esto significa que conducen la gestión del proyecto, y el fortalecimiento de la calidad del mismo. Coordinan la comunicación con los interesados, y reúnen al equipo del proyecto para mantenerlo enfocado en los objetivos. También son capaces de proporcionar soporte a los miembros del proyecto, en cuanto a

¹² Un Director artístico (también Director de arte) es una persona con mucha experiencia en el diseño para el mundo de la realidad virtual, y conoce perfectamente todos los aspectos necesarios para el trabajo gráfico de estas aplicaciones, es la persona que controla y dirige toda la parte artística de un proyecto, con el propósito de que se cumplan correctamente los planes definidos.

¹³ El guionista es una persona que no es necesario que tenga conocimientos informáticos, más bien debe poseer dominio de literatura, debe ser muy creativo para que sus ideas tengan resultado, es un rol muy importante debido a que de su trabajo depende que el producto tenga el impacto deseado.

herramientas y tecnologías. Entre los roles que trabajan en este equipo están: Líder de proyecto, Jefe de equipo¹⁴, Administrador de cambios/configuración, Gestor de la calidad.

Los proyectos del PRV, pueden apoyarse en la estructura organizativa de roles (presentada en la figura 24) que MERVI propone. La cual facilitará, y servirá de guía para el proceso de formación de roles, y organización de equipos. Esta estructura puede ser adaptada incluso, a la escala de los proyectos del Polo. Pues, los equipos de trabajo están combinados en pequeños grupos de roles, para proporcionar responsabilidades claras, compartidas, y comunicación abierta. De manera que cada grupo, puede ajustarse según lo requiera el proyecto.

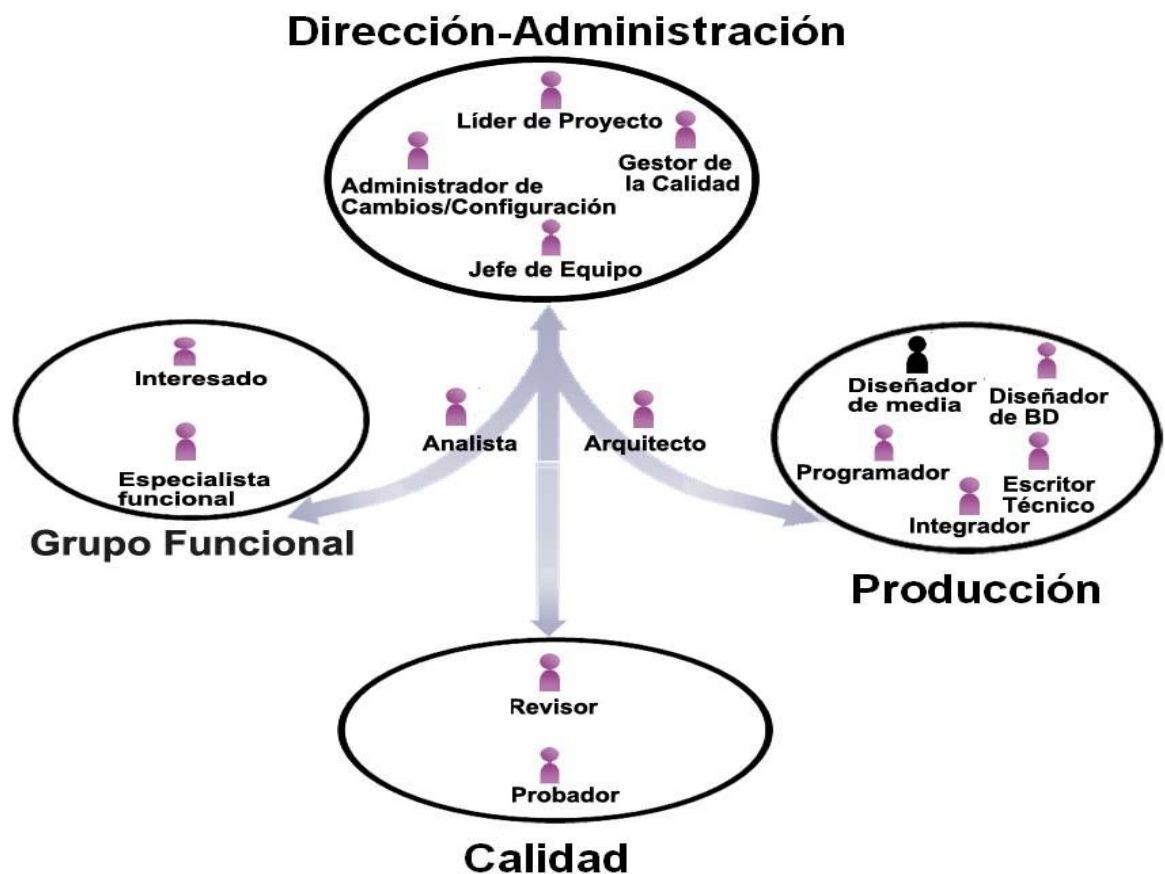


Figura 24. Estructura organizativa de MERVI.

¹⁴ Este rol estará representado en todos los grupos, y subgrupos de trabajo dedicados a una línea de producción específica en los proyectos del PRV. Generalmente, una persona puede asumir el rol de Jefe de Equipo, y a la misma vez otros roles (especialmente del equipo que dirige). En el caso individual de la edición de videos y sonidos, será dirigida y organizada por un director.

Para la determinación de las responsabilidades, conocimientos, y habilidades de los roles que a continuación se describen, se hace referencia además, a la propuesta [Álvarez, Pita, 2007] realizada por ingenieros de la UCI.

Rol	Responsabilidades	Conocimientos y habilidades
Líder de proyecto	<ul style="list-style-type: none"> ➤ Define la organización y estructura (líneas de trabajo) del proyecto. ➤ Gestiona y asigna recursos humanos y de otro tipo. ➤ Establece los horarios de trabajo del proyecto. ➤ Establece las estrategias de desarrollo del proyecto. ➤ Planifica las fases e iteraciones. ➤ Define, planifica, asigna y controla las tareas del proyecto. ➤ Coordina las interacciones con los clientes y los usuarios finales. ➤ Define el plan de capacitación y evaluación del personal. ➤ Planifica y realiza reuniones de información y control del proyecto. ➤ Realiza talleres y consejos técnicos con todos los miembros del proyecto. ➤ Informa sobre el estado actual del proyecto a los miembros del mismo y a instancias superiores. ➤ Motiva y organiza el equipo de trabajo para lograr un objetivo definido. ➤ Participa en la selección del personal del proyecto. ➤ Participa en los cursos de capacitación para líderes de proyecto. 	<ul style="list-style-type: none"> ➤ Metodologías de Desarrollo de Software. ➤ Tecnologías de desarrollo de software. ➤ Gestión de proyectos. ➤ Liderazgo y dirección (Motivar, Organizar, Innovar).
Jefe de equipo	<ul style="list-style-type: none"> ➤ Guía al equipo de desarrollo según las estrategias trazadas. ➤ Guía al equipo en la especificación del diseño del software. ➤ Guía al equipo en las pruebas del sistema. ➤ Guía al equipo en la producción de la documentación del usuario. ➤ Participa en la producción del reporte de desarrollo. ➤ Mide los requisitos de calidad de la línea de producto que se genera. 	<ul style="list-style-type: none"> ➤ Técnicas de desarrollo de los productos. ➤ Herramientas de desarrollo de los productos. ➤ Dirección (Motivar, Organizar, Innovar).
Analista	<ul style="list-style-type: none"> ➤ Dirige y coordina el proceso de extracción de requisitos y desarrollo del modelo de casos de uso, definiendo la funcionalidad y límites del sistema. 	<ul style="list-style-type: none"> ➤ Metodologías de Desarrollo de Software ➤ Lenguajes de modelado. ➤ Captura de requisitos y

Arquitecto	<ul style="list-style-type: none"> ➤ Especifica los requerimientos del sistema. ➤ Diseña el sistema en cuanto a requisitos, arquitectura, y proceso de desarrollo. ➤ Asesora a los desarrolladores en cuanto a preparación como analistas de software. ➤ Responsable del guión de contenido. ➤ Identifica técnicas apropiadas, herramientas e instrucciones para la implementación de las pruebas. ➤ Diseña los casos de pruebas. ➤ Define listas de chequeo para las pruebas. 	<ul style="list-style-type: none"> análisis de sistema. ➤ Herramientas de modelado visual. ➤ Guiones. ➤ Diseño. ➤ Pruebas de software. ➤ Estrategias de pruebas.
Rol general	<ul style="list-style-type: none"> ➤ Responsable de la arquitectura del SW. ➤ Lidera el proceso de arquitectura. ➤ Toma las decisiones técnicas más importantes en cuanto a las restricciones del diseño global e implementación del proyecto. ➤ Responsable de los planes de arquitectura. ➤ Define la forma en la cual los elementos del sistema trabajan en conjunto. ➤ Responsable de integrar requerimientos no funcionales en el sistema. 	<ul style="list-style-type: none"> ➤ Tecnologías de desarrollo de software. ➤ Lenguajes de programación. ➤ Patrones de diseño. ➤ Estilos arquitectónicos. ➤ Metodologías de Desarrollo de Software y lenguajes de modelado. ➤ Base de datos. ➤ Protocolos de comunicación.
Integrador	<p>Efectúa las tareas que puede realizar cualquier miembro del proyecto, o las comunes para todos los demás roles:</p> <ul style="list-style-type: none"> ➤ Enviar una solicitud de cambio. ➤ Realizar investigaciones. ➤ Reportar riesgos a Líder de proyecto. 	<ul style="list-style-type: none"> ➤ Capacidad para desempeñar cualquier otro rol definido en el proyecto. ➤ Herramientas de la infraestructura general. ➤ Búsqueda de Información.
Administrador de cambios/configuración.	<ul style="list-style-type: none"> ➤ Integrar los elementos implementados para producir versiones compiladas. ➤ Incorporar sonidos a los ambientes virtuales. ➤ Unir todos los fragmentos de videos en una sola línea y mezclarlos con el ambiente sonoro. 	<ul style="list-style-type: none"> ➤ Conocimiento del sistema o de parte del sistema que se integra. ➤ Interdependencias entre elementos y subsistemas de implementación. ➤ Expectativas de cambios en el desarrollo y las dependencias a lo largo del tiempo. ➤ Herramientas de integración, y de edición. ➤ Técnicas de edición. ➤ Plataformas de desarrollo.
	<ul style="list-style-type: none"> ➤ Planificar y chequear el proceso de gestión de configuración. ➤ Responsable del sistema de gestión de información, salvadas, etc. ➤ Define y supervisa el proceso de control de cambios. ➤ Proporciona al equipo de desarrollo la infraestructura general, de gestión de cambios y ambiente de trabajo. ➤ Es responsable del funcionamiento correcto de 	<ul style="list-style-type: none"> ➤ Herramientas control de versiones. ➤ Calidad de software (Estándares y modelos). ➤ Administración de la configuración. ➤ Documentación del SGC. ➤ Sistema operativo de la plataforma de desarrollo, la red, mecanismos de

Programador	<p>los servidores y las estaciones de trabajo durante el desarrollo y pruebas del software.</p> <ul style="list-style-type: none"> ➤ Responsable de la correcta instalación y configuración del producto en el lugar de destino. ➤ Define y selecciona la herramienta a utilizar para la gestión de la configuración. ➤ Capacita al personal en dicha herramienta. ➤ Responsable de la selección, gestión y obtención de las herramientas que se utilizarán en el proyecto. ➤ Debe instalar, configurar y asegurar que estas herramientas funcionan como se espera. 	<ul style="list-style-type: none"> ➤ seguridad y distribución. ➤ Resolución de problemas y el diagnóstico de anomalías.
Diseñador de Base de Datos	<ul style="list-style-type: none"> ➤ Participa en la etapa de diseño junto con los demás especialistas. ➤ Desarrolla los componentes, módulos o subsistemas a los que esté a cargo. ➤ Documenta el código escrito durante el desarrollo de las tareas que se le asignen. ➤ Realiza las pruebas de unidad de los componentes. ➤ Documenta los juegos de datos utilizados en las pruebas así como los resultados de las mismas. ➤ Da soporte técnico al código realizado por él u otros programadores. ➤ Participa en la preparación del desempeño como desarrollador. 	<ul style="list-style-type: none"> ➤ Lenguajes de programación. ➤ Herramientas de desarrollo. ➤ Plataformas de desarrollo
Escritor técnico	<ul style="list-style-type: none"> ➤ Define el gestor de base de datos a usar. ➤ Diseña la Base de Datos (BD), y garantiza la integridad referencial. ➤ Define la herramienta de modelado para BD relacionales. ➤ Define las políticas de cambio sobre los elementos de datos. ➤ Define los algoritmos de réplica, sincronización, respaldo, recuperación de la BD. ➤ Define las políticas de almacenamiento de los datos. ➤ Define las políticas de uso de los diferentes objetos de BD ante situaciones particulares. ➤ Responsable de las actualizaciones, correcciones, y mantenimiento de la BD del Sistema. ➤ Utiliza la información del análisis del negocio para identificar, definir y catalogar todos los datos que la aplicación almacenará en la BD. ➤ Documenta los datos mediante un diagrama Entidad-Relación. 	<ul style="list-style-type: none"> ➤ Herramienta de modelado visual. ➤ Modelo Relacional. ➤ SQL.
	<ul style="list-style-type: none"> ➤ Responsable de producir los materiales de soporte a los usuarios finales (guías de 	<p>Herramientas ofimáticas</p>

	usuarios, textos de la ayuda, notas asociadas a la salida del software, etc.).	
Probador	<ul style="list-style-type: none"> ➤ Ejecuta las pruebas diseñadas. ➤ Registra los resultados. 	<ul style="list-style-type: none"> ➤ Conocimientos del negocio. ➤ Pruebas de software. ➤ Calidad de software (Estándares y modelos).
Revisor	<ul style="list-style-type: none"> ➤ Elabora la lista de chequeo para las revisiones. ➤ Revisa todos los artefactos que se generan en el proyecto. ➤ Registra los resultados de las revisiones. ➤ Identificación, documentación, y seguimiento de las desviaciones del trabajo en el proceso del software. ➤ Identificación, documentación, y seguimiento de las no conformidades del producto. ➤ Realizar reportes para el líder del proyecto. 	<ul style="list-style-type: none"> ➤ Ingeniería de software. ➤ Calidad de software: Estándares y modelos. ➤ Revisiones técnicas formales.
Gestor de la calidad	<ul style="list-style-type: none"> ➤ Planifica el proceso de Aseguramiento de la Calidad en el proyecto. ➤ Realiza y da seguimiento al Plan de prueba, de revisión y auditoría de cada iteración. ➤ Coordina el proceso de recopilación, análisis y reporte de las estadísticas de calidad. ➤ Guía las revisiones técnicas formales. ➤ Guía las pruebas y las auditorías que se realicen. ➤ Maneja todo lo relacionado con los riesgos de calidad. ➤ Realiza el Resumen de Evaluación de Pruebas. 	<ul style="list-style-type: none"> ➤ Calidad de Software. ➤ Ingeniería de Software. ➤ Métricas y análisis estadístico. ➤ Monitoreo de proyectos informáticos. ➤ Herramientas de gestión de riesgos.
Diseñador de media	<ul style="list-style-type: none"> ➤ Diseña las escenas, modelos y componentes necesarios para el ambiente virtual. ➤ Define el diseño gráfico de la aplicación. ➤ Interviene en la creación del prototipo. ➤ Define las pautas para el diseño de la interfaz. ➤ Modela las escenas, modelos y elementos necesarios para el ambiente virtual. ➤ Participa en la edición, y realización de videos. ➤ Participa en la edición y realización de sonidos. ➤ Realiza animaciones necesarias para el ambiente virtual. ➤ Realiza los efectos especiales necesarios en el ambiente virtual. 	<ul style="list-style-type: none"> ➤ Técnicas de texturizado. ➤ Herramientas de texturizado. ➤ Herramientas de edición de sonidos. ➤ Lenguajes de programación. ➤ Técnicas de animación 3D. ➤ Comunicación visual. ➤ Herramientas de desarrollo gráfico (2D, 3D, cinemáticas)

Tabla 5. Propuesta de roles, responsabilidades y conocimientos.

Hay que tener especial cuidado para determinar la utilización de estos roles, ya que se puede caer en la redundancia de la asignación de responsabilidades. Estos roles deben ser utilizados según las necesidades específicas del proyecto, y de la línea de productos que se desarrolla.

3.1.4 DISCIPLINAS

Las disciplinas (también llamadas flujos de actividades) que propone MERVI, tienen participación a lo largo de todo el ciclo de vida del proyecto. Teniendo en cuenta aspectos conceptuales, tecnológicos, metodológicos, organizacionales, y gerenciales de las LPS. Las disciplinas del presente trabajo, están divididas en flujos de protección, y flujos de trabajo. A continuación se describen cada una de ellas.

3.1.4.1 FLUJOS DE PROTECCIÓN

Estas disciplinas están centradas en el personal, producto, proceso, y proyecto para la construcción de un SRV o servicio. Mediante la aplicación de una serie de prácticas, consideradas como críticas, las cuales se mencionan a continuación:

- Gestión rigurosa de riesgos.
- Estimación de la planificación a dos niveles (valor, y riesgos).
- Gestión basada en métricas.
- Seguimiento del valor ganado.
- Seguimiento de insuficiencias, y aciertos frente a los hitos de calidad.
- Gestión del programa del personal.

Aunque no es el propósito del presente trabajo, proponer métodos para la Gestión de proyecto, Administración de cambios/configuración, y Aseguramiento de la calidad. A continuación se describen los elementos más significativos de los flujos de protección propuestos por MERVI, para una mejor comprensión de lo expuesto anteriormente.

Gestión de proyecto: Esta disciplina está relacionada fundamentalmente con la gestión organizacional, y está descrita por el Líder del proyecto. Se basa en la planificación sobre cambios, y entregas cortas del proyecto, la gestión de riesgos (organizacionales principalmente), monitoreo del progreso, y la métrica. Así como la incorporación iterativa de nuevos elementos de SW. También se encarga de seleccionar, y capacitar al personal adecuado, para enfrentar el proyecto con éxito. Así como ayudarlos a ocuparse de los conflictos, y dificultades encontrados durante la elaboración del SW.

Los propósitos que persigue la Gestión de proyecto son:

- Proporcionar una infraestructura para gestionar intensivamente la LPS de un proyecto. Durante todo el ciclo de vida, y para una iteración determinada.
- Proporcionar reglas prácticas para la planificación, selección y capacitación de personal, métrica, monitoreo y control del progreso.
- Proporcionar una infraestructura para gestionar los riesgos.
- Administración del personal, presupuesto, y recursos.
- Administración de los contratos, y relaciones con los proveedores y los clientes.
- Análisis de mercados.
- Concepción de operaciones, y tecnologías.

En la tabla siguiente se pueden observar las actividades, tareas, y resultados que se obtienen:

Entradas	Actividades	Salidas
Visión.	Desarrollar un plan de proyecto.	Plan de proyecto.
Plan de proyecto.	Desarrollar un plan de iteración.	Plan de iteración.

Tabla 6. Actividades, entradas y salidas de la disciplina Gestión de proyecto.

Administración de cambios/configuración: Se relaciona principalmente con la gestión técnica en el proyecto, y es referida por el Administrador de cambios/configuración. Esta disciplina explica cómo controlar y adaptar la evolución de los productos de trabajo que componen el SW. Está diseñada para que el equipo, y los interesados sean retroactivos. También los cambios deben ser considerados como riesgos inherentes; por lo que deben ser reconocidos, registrados, y gestionados. De manera resumida, la Administración de cambios/configuración es un conjunto de actividades dedicadas a gestionar los cambios producidos durante todo el ciclo de vida del proyecto. A medida que van evolucionado los productos de trabajo, se convierten en elementos de configuración del SW, y se gestionan sin importar el momento, o los motivos por los que estos cambios son producidos.

Esta disciplina persigue los siguientes objetivos:

- Definición, y soporte de herramientas al proceso de desarrollo.
- Mantener la integridad del producto.
- Garantizar el correcto cumplimiento del producto de trabajo configurado.
- Proporcionar un entorno estable de desarrollo, control de versiones, y documentación para la LPS del proyecto.

- Limitar los cambios a los elementos de configuración del SW, basados en políticas del proyecto.
- Proporcionar informes de estado, e indicios para las auditorías sobre los motivos, momentos, y autores de los cambios en los productos de trabajo.
- Gestión tecnológica, y de riesgos (técnicos fundamentalmente).

A continuación se observan las actividades, tareas, y salidas que se deben cumplir durante la Administración de cambios/configuración:

Entradas	Actividades	Salidas
Proceso de desarrollo.	Configurar el ambiente de trabajo.	Proceso de desarrollo especificado. Infraestructura general.
Solicitud de cambio.	Desarrollar un plan de gestión de configuración.	Plan de gestión de configuración.

Tabla 7. Actividades, entradas, y salidas de la disciplina Administración de cambios/configuración.

Aseguramiento de la calidad: Involucra la aplicación de lineamientos, procedimientos, y estándares de calidad que regulen la producción en el PRV. Aunque otros roles, como por ejemplo, el Probador, realizan actividades que garantizan la calidad del producto. El Aseguramiento de la calidad, es una disciplina que se describe esencialmente por el Gestor de la calidad, el cual estará centrado en el control de la evolución del proyecto, el proceso, las personas, y los productos. Para reducir las diferencias entre los expedientes de proyecto terminados, y los reales, incluyendo personal, equipos, y tiempos de avance requeridos. Esto tiene que ver con el alcance de un porcentaje de calidad conocido, de una solución a otra. Esta disciplina consiste en la auditoría, y el reporte de los informes de calidad del producto, teniendo en cuenta atributos de calidad para los SRV desarrollados por el Polo.

El Aseguramiento de la Calidad tiene las siguientes intenciones fundamentales:

- Control de la calidad (inspecciones, revisiones, y pruebas).
- Garantía de la calidad (auditorías, y reportes de la gestión).
- Valor de la calidad (prevención, evaluación, y fallos para mejorar los procesos).

En la siguiente tabla se puede observar las actividades con sus responsables, las tareas, y los resultados que se generan:

Entradas	Actividad	Salidas
----------	-----------	---------

Tabla 8. Actividades, entradas, y salidas de la disciplina Aseguramiento de la calidad.

3.1.4.2 FLUJOS DE TRABAJO

Un flujo de trabajo es el conjunto de actividades ordenadas, descritas en el marco de trabajo de un proceso. Orientan a los miembros de un proyecto en cuanto a cómo hacer para lograr un producto. Son aplicables a todos los proyectos, independientemente de su diversidad. En el actual epígrafe, se describirán cada uno de los flujos de trabajo propuestos por MERVI. Los cuales, deben adaptarse a las necesidades de los proyectos, y las personas del PRV.

Ingeniería de requisitos: Explica cómo identificar, analizar, especificar, validar y gestionar los requerimientos del cliente, y del producto que será desarrollado. Está dirigida fundamentalmente al análisis técnico, de las actividades del usuario, y de mercado. Durante la ingeniería de requisitos, los clientes plantean los requisitos que son importantes para ellos, en la entrega del producto. Para esto, el analista utiliza métodos de captura de información. Además de Historias de usuario, que pueden ser aplicadas al cliente, mediante entrevistas o encuestas. Esta disciplina persigue los siguientes objetivos:

- Comprender el contexto del problema que será solucionado.
- Comprender las necesidades e intenciones de los interesados.
- Identificar interfaces de comunicación para el proyecto.
- Definir los requisitos. O sea, lo que la aplicación debe hacer.
- Definir el alcance del proyecto.
- Identificar requisitos técnicos de la solución, y los arquitectónicamente significativos.
- Proporciona la base para la previsión, planificación, y estimación de las iteraciones.

La ingeniería de requisitos ajustada a un producto de RV, incluye además, los siguientes objetivos:

- Identificar los componentes que interactúan.
- Identificar los elementos que participarán en el ambiente virtual.
- Describir el ambiente del problema.
- Definir detalles de los elementos del ambiente virtual.

Diseño y codificación: Explica cómo transformar los requisitos en productos de trabajo, que definan el diseño del SW, y su arquitectura. Así como el diseño, la construcción de la arquitectura, y la solución técnica. Durante el diseño y codificación se desarrolla, organiza, integran, y prueban los componentes. La finalidad de este flujo de trabajo consiste en:

- Definir lo que será diseñado e implementado.
- Adaptar un diseño al entorno de implementación, pensado para el rendimiento y la reutilización de activos de SW.
- Realizar diseños gráficos, e implementaciones de las soluciones pertinentes, y una arquitectura consistente.
- Verificar la estabilidad, la exactitud de la arquitectura, el diseño, y la implementación.

Entre estas metas, para la elaboración de un producto de RV, se incluyen también las siguientes:

- Descomponer elementos del ambiente virtual en partes constituyentes, válidas para el problema.
- Diseñar y construir la aplicación virtual, a partir de activos reutilizables.
- Diseñar y construir programas de comportamiento para los elementos del ambiente virtual.
- Incorporar atributos (texturas, sonidos, etc.) a las entidades de la aplicación virtual.
- Optimizar el diseño gráfico, y la implementación.

La disciplina Diseño y codificación para una aplicación de RV, está enfocada en el diseño del HW/SW, de la interacción, de entidades o escenarios virtuales, y de la historia. Así como en la implementación de prototipos de la plataforma, la interacción, y el ambiente virtual.

Prueba: Describe cómo diseñar, implementar, ejecutar, y evaluar pruebas. Para proveer información retroactiva acerca de la capacidad operacional de un producto en evolución. Los propósitos de esta disciplina son los siguientes:

- Buscar y registrar defectos sobre la calidad de la solución de manera temprana.
- Proveer información retroactiva sobre la calidad percibida, de manera eficaz y frecuente.
- Medir objetivamente el progreso, en los incrementos.
- Verificar y demostrar la validez de la solución.

Entre estos objetivos, en la construcción de soluciones de RV, el flujo de pruebas incluye además los siguientes:

- Verificar definición de componentes.
- Validar el diseño y la implementación, con la definición del problema.
- Realizar pruebas de comportamiento y de interacción.

La prueba debe estar centrada en la ejecución, el servicio, y el trabajo artístico del producto.

Entrega: Detalla las actividades relacionadas con la disposición e instalación del producto, el soporte, y la garantía para los usuarios. Registrando los reportes de mantenimiento, y soporte realizados. Así como los ajustes, y las versiones obtenidas.

En el desarrollo de soluciones de RV, este flujo de trabajo tiene también los siguientes propósitos:

- Optimización del diseño.
- Incorporación de periféricos (si posee alguno).

Es muy importante aclarar, que esta disciplina se limita a lo plasmado en la visión del proyecto. En caso de que se generen requisitos, y características nuevas. Que no hayan sido tenidos en cuenta, o se salen del alcance del proyecto. Es muy probable, que se genere otro proyecto de desarrollo de productos, y se comience un nuevo ciclo en la espiral (que se muestra en la figura 19).

3.1.5 FASES

MERVI propone cuatro fases a través de todo el ciclo:

Pre-Producción: La meta principal en esta fase, es alcanzar un acuerdo con todos los interesados respecto a los objetivos del ciclo vital para el proyecto. Obtener una visión compartida, comprendida y alineada con los objetivos del dominio. Se determina al menos una solución del problema, identificando beneficios, requerimientos funcionales, alcances, limitaciones y riesgos. También se obtendrá un cronograma de trabajo, en el cual se identifiquen puntos de control que proporcionen pequeñas liberaciones funcionales, teniendo en cuenta presupuesto, tiempo, recursos y riesgos. Además, en esta

etapa es cuando se comienza la capacitación del equipo de trabajo, y se familiariza con en el ambiente de trabajo, las herramientas, tecnologías y prácticas del proceso.

Hito: Objetivos del proyecto.

La siguiente tabla resume los flujos de trabajo, los responsables, las actividades, las entradas y salidas que debe cumplirse, cada vez que se realice una iteración.

Flujos de trabajo	Responsable	Entrada	Actividades	Salida
Ingeniería de requisitos.	Analista	Historias de usuario.	Definir ambiente del problema.	Guión de contenidos.
		Guión de contenidos.	Desarrollar la visión técnica.	Visión.
		Guión de contenidos. Visión.	Desarrollar un plan de gestión de requisitos.	Plan de gestión de requisitos.
		Guión de contenidos. Visión.	Definir el ambiente virtual.	Guión técnico.
		Plan de gestión de requisitos. Guión técnico.	Diseñar los casos de prueba.	Caso de prueba.
	Arquitecto	Visión. Guión técnico.	Definir la arquitectura candidata.	Arquitectura de software.
		Revisor	Guión de contenidos. Visión. Plan de gestión de requisitos. Guión técnico. Caso de prueba. Arquitectura de software.	Realizar revisiones.

Tabla 9. Flujos de trabajo, actividades, responsables, entradas, y salidas en la fase Pre - Producción.

Ingeniería de Dominio (ID): La finalidad principal en esta fase es el establecimiento de una línea base para la arquitectura de la LPS, además de proporcionar una plataforma estable para el diseño refinado, y la implementación de la fase siguiente. También se captura información, y constituye la comprensión del dominio de la línea. Con el fin principal de adquirir un entendimiento de los requerimientos, con un nivel de detalle más alto, y de crear activos de SW reutilizables en la solución de cualquier nuevo producto de la LPS. En esta fase se crean mecanismos que adiestren a los miembros del proyecto para reutilizar los componentes a lo largo del trabajo, en sistemas actuales o futuros.

Hito: Arquitectura del proyecto.

La tabla siguiente resume los flujos de trabajo, los responsables, las actividades, las entradas, y salidas que debe efectuarse, en cada iteración que se realice.

Flujos de trabajo	Responsable	Entrada	Actividades	Salida
Diseño y codificación.	Analista.	Plan de gestión de requisitos. Guión técnico.	Diseñar la solución.	Modelo de diseño.
	Arquitecto.	Visión. Guión técnico. Modelo diseño.	Perfeccionar la arquitectura.	Arquitectura de software. (detallada)
	Programador.	Plan de gestión de requisitos. Arquitectura de software.	Desarrollar componentes	Modelo de implementación.
		Modelo de implementación. Caso de prueba.	Realizar pruebas unitarias.	Plan de pruebas.
	Diseñador de BD.	Plan de gestión de requisitos. Arquitectura de software.	Desarrollar la Base de Datos.	Modelo de Datos.
		Modelo de Datos. Caso de prueba.	Realizar pruebas unitarias.	Plan de pruebas.
	Diseñador de media.	Plan de gestión de requisitos. Arquitectura de software. Modelo de implementación.	Realizar diseño de media.	Modelo de media.
Modelo de media. Caso de prueba.		Realizar pruebas unitarias.	Plan de pruebas.	
Prueba.	Revisor	Modelo de diseño. Arquitectura de software. Modelo de implementación. Plan de pruebas. Modelo de Datos. Modelo de media.	Realizar revisiones.	Registro de la calidad.
	Probador	Modelo de implementación. Modelo de Datos. Modelo de media. Caso de prueba.	Realizar pruebas de dominio.	Plan de pruebas.

Tabla 10. Flujos de trabajo, actividades, responsable, entradas, y salidas en la fase Dominio.

Ingeniería de Aplicaciones (IA): En esta fase se refinan los requisitos, y se completa el desarrollo de la aplicación, apoyándose en la arquitectura de LPS definida en la fase anterior. Durante la IA, se procede iterativamente a la construcción de los productos de la línea, a través de la reutilización de activos de SW, y los planes de producción. Minimizando los costos de desarrollo, y alcanzando un positivo nivel de equilibrio. También la arquitectura de dominio de la etapa anterior, sirve como modelo de referencia en esta fase para diseñar los productos de la línea. Por otra parte, en esta etapa es imprescindible la utilización de un Sistema de Gestión de Activos de Software Reutilizables (repositorio LPS); que

suministre los activos de SW, necesarios para el desarrollo de cada nuevo producto de la línea. Finalmente, se debe asegurar que el producto está listo para ser entregado a los clientes, y usuarios.

Hito: Capacidad operacional de la aplicación.

A continuación se resume en una tabla los flujos de trabajo, los responsables, las actividades, las entradas, y salidas que debe realizarse, en cada iteración que se efectúe.

Flujos de trabajo	Responsables	Entradas	Actividades	Salidas
Diseño y codificación.	Integrador	Modelo de implementación. Modelo de Datos. Modelo de media.	Construir ambiente virtual.	Producto.
	Revisor	Producto. Modelo de entrega. Plan de pruebas.	Realizar revisiones.	Registro de calidad.
	Escritor técnico	Producto.	Proporcionar soporte a usuarios.	Modelo de entrega.
Prueba.	Probador	Producto. Caso de prueba.	Realizar pruebas de integración.	Plan de pruebas.

Tabla 11. Flujos de trabajo, actividades, responsable, entradas, y salidas en la fase Aplicación.

Post-Producción: Es la última fase, cuyo objetivo principal es determinar si se ha cumplido con las expectativas del cliente, llegar a un acuerdo con los interesados acerca de la continuación del proyecto, así como realizar un fuerte, y retrospectivo análisis postmortem del proyecto. Esto se logra realizando la evaluación de versiones Beta, garantizando que el producto está dispuesto para entregarlo con calidad a los interesados. También se necesita un empeño agregado por parte de los desarrolladores para satisfacer las exigencias de los interesados. Si el cliente está satisfecho con el sistema, si la aplicación no llena las perspectivas esperadas por los interesados, o si es insuficiente el presupuesto; entonces el proyecto finaliza. En cambio, si se producen nuevos requisitos, y características que no fueron pactadas inicialmente. El equipo de trabajo inicializa un nuevo ciclo en la espiral (que se muestra en la figura 19), comenzando de esta manera un nuevo proyecto de desarrollo de productos.

Hito: Entrega de versiones.

La siguiente tabla resume los flujos de trabajo, los responsables, las actividades, las entradas, y salidas que debe cumplirse, cada vez que se realice una iteración.

Flujos de trabajo	Responsables	Entradas	Actividades	Salidas
Entrega.	Integrador	Plan de Pruebas. Producto.	Perfeccionar el ambiente virtual.	Producto. (mejorado)
	Escritor técnico.	Producto.	Proporcionar soporte a los usuarios.	Modelo de entrega. (actualizado)
	Revisor.	Producto. Modelo de entrega. Plan de Pruebas.	Realizar revisiones. Registrar resultados de pruebas de aceptación.	Registro de calidad.

Tabla 12. Flujos de trabajo, actividades, responsable, entradas, y salidas en la fase Post-Producción.

3.1.6 ARTEFACTOS

Los artefactos y la documentación generada en el proceso referido en MERVI, están ubicados en un expediente de proyecto. El cual recopila y organiza, toda la información referente a los dominios de un proyecto. Un dominio puede subdividirse en subdominios, también puede ser catalogado como la agrupación de productos de trabajo asociados entre sí. De acuerdo con los recursos, el tiempo, o la relación.

El expediente de proyecto está estructurado según la enumeración a continuación:

1. Expediente de proyecto

1.1. Trabajo

1.1.1. Ingeniería de requisitos

- 1.1.1.1. Historia de usuario.
- 1.1.1.2. Plan de gestión de requisitos.
- 1.1.1.3. Visión.
- 1.1.1.4. Guión de contenido.
- 1.1.1.5. Guión técnico.
- 1.1.1.6. Caso de prueba.

1.1.2. Diseño y codificación

- 1.1.2.1. Arquitectura de software.
- 1.1.2.2. Modelo¹⁵ de datos.

¹⁵ En cada modelo del expediente, se incluyen además de la plantilla del modelo correspondiente, componentes de dominios asociados. Es decir: Diagramas, modelos gráficos, dibujos, códigos fuentes, ejecutables, etc.

- 1.1.2.3. Modelo de diseño.
- 1.1.2.4. Modelo de implementación.
- 1.1.2.5. Modelo de media.¹⁶
- 1.1.3. Prueba**
 - 1.1.3.1. Plan de pruebas.
- 1.1.4. Entrega**
 - 1.1.4.1. Modelo de entrega¹⁷
- 1.2. Protección**
 - 1.2.1. Gestión de Proyecto.**
 - 1.2.1.1. Plan de proyecto.
 - 1.2.1.2. Plan de iteración.
 - 1.2.1.3. Acta de reunión.
 - 1.2.2. Administración de cambios/configuración.**
 - 1.2.2.1. Plan de gestión de configuración.
 - 1.2.2.2. Acta de reunión.
 - 1.2.3. Aseguramiento de la calidad.**
 - 1.2.3.1. Registro de calidad.
 - 1.2.3.2. Plan aseguramiento de la calidad.
 - 1.2.3.3. Acta de reunión.
- 1.3. Adicional¹⁸**
 - 1.3.1. Solicitud de cambio.
 - 1.3.2. Investigación.

A continuación se ofrece una breve descripción, de la documentación generada por en la presente guía.

¹⁶ Dentro del modelo de media, pueden encontrarse subdominios para: Escenarios virtuales, objetos y personajes, texturas, animaciones, efectos especiales, videos, y sonidos.

¹⁷ En esta parte del expediente, se incluyen los manuales de usuario, ayudas, videos, presentaciones, u otros archivos de soporte para el interesado, y marketing del producto.

¹⁸ En esta parte del expediente, se encuentran los productos de trabajo generados por Cualquier rol, u otra información que necesite ser archivada o que no haya sido prevista.

Historia de usuario: Constituye una descripción breve de alguna funcionalidad prevista por el sistema desde el punto de vista del usuario. Su objetivo, es conocer las características y algún segmento de funcionalidad del sistema que se construirá. Detalla una vista, usando representaciones escritas y visuales. Una Historia de usuario se usa también, para ilustrar detalles en un panorama antes de que sea diseñado o codificado. Esta plantilla puede ser manuscrita, o creada en cualquier herramienta de redacción de documentos, o visualización. La Historia de usuario es aplicada por el analista a los interesados, y puede ser desarrollada en forma de encuesta, entrevista, u otro método de recolección de información. Este proceso de captura de requisitos en historias de usuario, puede estar apoyado por el equipo de producción. (Ver Anexo 4: Plantilla de Historia de usuario.)

Plan de gestión de requisitos: En este documento se describe cómo desarrollar un plan para documentar requisitos, sus atributos, directrices para la rastreabilidad y gestión de los requisitos del producto. El objetivo es desarrollar un plan de gestión de requisitos que especifique la información, además de los mecanismos de control que se recopilarán y utilizarán para medir, informar y controlar los cambios efectuados en los requisitos del producto. Esta plantilla es sugerida para especificar necesidades y limitaciones. (Ver Anexo 5: Plantilla de Plan de gestión de requisitos.)

Visión: Este artefacto describe la visión general del sistema, que incluye el problema que se va a resolver. Además, es especificada en términos de características, y necesidades cruciales de los interesados. Provee una base de alto nivel para detallar los requisitos más técnicos, y constituye una guía para el equipo durante el ciclo de desarrollo. Con este artefacto, se acuerda qué problemas se deben resolver, se identifican los interesados, definen los límites del sistema y se describen las características principales del sistema. (Ver Anexo 6: Plantilla Visión.)

Guión de contenido: Es la parte narrativa o descripción del problema en el ambiente virtual, es decir, la historia. En este documento se describen los personajes/actores en el ambiente virtual, objetos/ítems, reglas/mecánicas, escenas, eventos, y otras especificaciones que se consideren necesarias en su descripción. Estos elementos del ambiente del problema se obtienen a partir de los análisis de las historias de usuario, que son explicados en el Guión de contenidos. Luego, en el Guión técnico son diseñados, y detallados con la ayuda de los interesados. (Ver Anexo 7: Plantilla de Guión de contenido.)

Guión técnico: Este documento describe la parte artística e industrial del software, es decir, como está formada la estructura de la aplicación técnicamente. Describe las tecnologías, modos, motores, plataformas que se utilizarán para dar las soluciones. También describe los elementos de diseño gráfico (mensajes de alertas, iconos, botones, etc.), las media (videos, sonidos, textos, animaciones, efectos especiales, videos, etc.) que compone, los módulos, pantallas, atributos (color, formas, etc.) de los elementos del ambiente virtual. Además de los eventos y funcionalidades, que debe proporcionar el ambiente virtual al usuario; los cuales se especifican con diagramas, modelos y descripciones textuales. También proporciona una vista previa del contenido de la aplicación con figuras e imágenes. (Ver Anexo 8: Plantilla de Guión técnico.)

Caso de prueba: Este artefacto es la especificación de un grupo de aportes experimentales, condiciones de ejecución y resultados esperados; identificados al evaluar un aspecto particular de un escenario o Caso de uso. Tiene como objetivo proveer una manera de ver cómo responde el sistema o producto de software al introducirle información, y ejecutarlo. En este documento, también se identifican sistemáticamente elementos a probar del software. Se especifica si un resultado esperado ha sido alcanzado, basado en la verificación de un requisito de sistema, el grupo de requisitos, o escenario. Un caso de prueba especifica las condiciones que deben ser validadas, para permitir una valoración de aspectos del sistema bajo prueba. Esta plantilla contiene además las instrucciones graduales que componen una prueba automatizada, permitiendo ver el estado en el que se encuentra un software puesto en marcha. (Ver Anexo 9: Plantilla de Caso de prueba.)

Arquitectura de software: Este artefacto describe la justificación razonada, explicación, suposiciones e implicaciones acerca de las decisiones que fueron tomadas para formar la arquitectura. La arquitectura de software representa la estructura del producto, también proporciona una visión general arquitectónica completa de la aplicación; mediante una serie de vistas arquitectónicas diferentes para representar diferentes aspectos del sistema. El objetivo de este artefacto es crear un documento de arquitectura de software, que proporcione una visión general completa de la arquitectura del proyecto de software. Sirve como medio de comunicación, entre el arquitecto de software y otros miembros del equipo de trabajo; respecto a las decisiones significativas para la arquitectura que se llevan a cabo en el proyecto. (Ver Anexo 10: Plantilla de Arquitectura de software.)

Plan de pruebas: Especifica los objetivos de las pruebas en el ámbito de desarrollo unitario, la iteración, o el proyecto para el enfoque que se adopta, los recursos necesarios y los entregables que se deben generar. Persigue esquematizar, y comunicar la intención de las pruebas de una planificación determinada. Provee además, un registro detallado del conjunto de pruebas realizadas, mediante la construcción de los casos de prueba. Esta información registrada, es analizada y utilizada para obtener la aceptación, o aprobación de los interesados. (Ver Anexo 11: Plantilla de Plan de pruebas.)

Plan de proyecto: Esboza un acuerdo inicial, que constituye el punto de partida para planificar cómo logrará el proyecto sus metas. Este plan provee una visión general, nivelada y resumida del proyecto. Su objetivo es obtener las condiciones de los interesados, para planificar como se llevará a cabo el desarrollo del proyecto. Provee la lista de tareas que necesitan ser realizadas, antes de comenzar un proyecto. (Ver Anexo 12: Plantilla de Plan de proyecto.)

Plan de iteración: Permite la planificación del alcance, y las responsabilidades para una iteración. Su propósito es identificar, y crear un plan para el siguiente incremento de la aplicación. Un plan de iteración es el conjunto de visiones, requisitos y tareas para una iteración. El plan de iteración comienza, y finaliza en la iteración. (Ver Anexo 13: Plantilla de Plan de iteración.)

Plan de gestión de configuración: El Plan de gestión de configuración, detalla los elementos necesarios para la administración y dirección de control de cambio/configuración que se realizarán en el proyecto. El propósito de este plan es definir, o establecer los pasos y actividades necesarias para la gestión de cambios/configuración en el desarrollo de un producto del software. (Ver Anexo 14: Plantilla de Plan de gestión de configuración.)

Registro de calidad: Descripciones de varios registros de revisiones, y auditorías que se mantendrán durante el proyecto. Incluyendo cuando, por quien, cómo y dónde, cada tipo de registro se guardará, y por cuánto tiempo. (Ver Anexo 15: Plantilla de Registro de calidad.)

Plan aseguramiento de la calidad: Establece métodos, normas y estándares concretos que apoyen la gestión de calidad del producto, el proceso y las personas en el proyecto. Proporciona las herramientas de planificación, gestión y cumplimiento del aseguramiento de la calidad. Este plan es muy útil para que los miembros del proyecto, e involucrados, trabajen con procedimientos y visión de mejoramiento continuo.

Con este documento se busca responder a la necesidad de que las personas, puedan interiorizarse y manejar los conceptos, así como las herramientas de gestión y garantía de la calidad. (Ver Anexo 16: Plantilla de Plan aseguramiento de la calidad.)

Solicitud de cambio: Las peticiones (o solicitudes de cambio) se usan para documentar y rastrear defectos, peticiones de importancia y algún otro tipo de solicitud para un cambio del producto. La ventaja de usar solicitudes de cambio, es que proveen un registro de decisiones, predicciones y, debido a su proceso de valoración, asegura que los impactos de cambio son comprendidos a través del proyecto. La necesidad para el cambio, es inherente en la evolución y existencia de soportes lógicos. La solicitud de cambio, es un método para asegurar que la producción de aplicaciones se realiza de forma controlada. (Ver Anexo 17: Plantilla de Solicitud de cambio.)

Investigación: Este documento es producto del carácter investigativo – productivo del PRV, es decir, es generado por la necesidad de documentar las investigaciones realizadas en el proyecto. Como consecuencias del aprendizaje continuo, y la búsqueda de soluciones a problemas presentados durante la marcha en el proyecto. Con la intención principal de proveer a los miembros del proyecto, de capacidad valorativa, colaboración mutua y aprendizaje continuo. También para proporcionar una base sólida de conocimientos, y experiencias para el desarrollo de futuras aplicaciones. (Anexo 18: Plantilla de Investigación.)

Acta de reunión: Proporciona la información generada en reuniones realizadas durante el transcurso del proyecto. La reunión es la oportunidad que tienen los miembros del proyecto para fomentar las relaciones personales, organizarse, comunicarse y adquirir conocimientos. A través del acta de reunión, se puede registrar los debates acerca de temas como la gestión del proyecto (para obtener compromisos, estimular, amonestar, orientar, etc.), la gestión de cambios/configuración (acerca de la infraestructura general del proyecto, cómo hacer una solicitud de cambio, etc.), o el aseguramiento de la calidad. También puede ser sobre el comienzo, la evolución y conclusión del trabajo en las iteraciones; así como de los planes de revisiones y auditorías. (Anexo 18: Acta de reunión.)

3.2 APORTES

El principal aporte del presente trabajo lo constituye la guía, teniendo en cuenta el problema real que tiene el PRV, donde se generan más artefactos que los necesarios con el empleo de algunas metodologías, ocasionando mucha pérdida de tiempo para el desarrollo de los productos. Además, existen roles, actividades y productos de trabajo que son de vital importancia para la construcción de las aplicaciones de RV en el Polo, que las metodologías existentes no definen. Esta guía se propone resolver todos estos problemas, pues contiene los artefactos, actividades y roles necesarios para el llevar a cabo soluciones de RV. El presente trabajo constituye una idea de los beneficios y soluciones proporcionadas, para algunas dificultades que presenta el PRV.

El hecho de que se combinen los modelos espiral, incremental y basado en reutilización, es un aporte muy importante. La tabla 13, proporciona una idea del nivel efectividad de cada uno de los modelos de proceso referidos en esta guía, en algunos criterios básicos que se deben tener en cuenta en el desarrollo de productos de SW.

Modelos de proceso	Desempeño si no se predefinen requisitos	Produce software fiable	Gestión de riesgos	Gestión de cambios	Visibilidad del progreso
Desarrollo basado en reutilización	Medio	Bajo a Alto	Bajo a Medio	Alto	Alto
Desarrollo Incremental	Bajo	Alto	Medio	Medio	Bajo
Desarrollo Espiral	Alto	Alto	Alto	Medio	Medio

Tabla 13. Comparación entre modelos de proceso de software. [16]

Otro de los aportes que benefician inmensamente a la organización, y orientación en el proceso de asignación de roles, es el modelo de equipos, y su estructura organizativa. Ya que contextualizan el problema entorno a la determinación de roles y equipos; cómo colaboran, comunican, y se desempeñan en el ambiente de trabajo.

El presente trabajo, incluye otros elementos que constituyen ajustes al desarrollo de las aplicaciones de RV, los cuales se describen a continuación.

Una de las aportaciones novedosas, constituye la definición del rol Diseñador de media. Pues, este término está especialmente definido en la presente propuesta, con el propósito de agrupar personas con

la responsabilidad y capacidad, de llevar a cabo tareas específicas (modelado de objetos y escenas virtuales, texturizado, edición de videos y sonidos, desarrollo de animaciones y efectos especiales, etc.), para desarrollar soluciones de diseño de mundos virtuales, y animación. Diseñador de media también puede ser considerado como un conjunto de roles, ya que muchos proyectos del PRV utilizan diferentes roles cuando desarrollan productos de RV en sus líneas. En este caso, los roles representados bajo el conjunto Diseñador de media son los siguientes:

Modelador	<ul style="list-style-type: none"> ➤ Dibuja las escenas, modelos y objetos necesarios para la tarea. ➤ Modela las escenas, modelos y objetos necesarios para la tarea. ➤ Define el desarrollo gráfico de la aplicación. ➤ Interviene en la creación del prototipo. ➤ Define las pautas para el diseño de la interfaz. 	<ul style="list-style-type: none"> ➤ Comunicación visual. ➤ Herramientas de desarrollo gráfico (2D, 3D, Cinemáticas)
Texturizador	<ul style="list-style-type: none"> ➤ Encargado de crear las texturas de los personajes, objetos y entornos en general. ➤ Texturiza los componentes de la escena. ➤ Produce varios canales de textura para aumentar el realismo en la escena. ➤ Produce imágenes según las necesidades de un cliente. 	<ul style="list-style-type: none"> ➤ Técnicas de texturizado. ➤ Herramientas de texturizado.
Diseñador gráfico.	<ul style="list-style-type: none"> ➤ Diseña prototipos de interfaz de usuario según los requerimientos de usabilidad del cliente. ➤ Encargado de realizar el trabajo artístico que requiera el proyecto (íconos, logos, pantallas, gráficos, etc.). 	<ul style="list-style-type: none"> ➤ Lenguajes de programación. ➤ Técnicas de animación 3D. ➤ Herramientas de diseño gráfico (2D, 3D, cinemáticas).
Sonidista	<ul style="list-style-type: none"> ➤ Construye el ambiente sonoro de un ambiente virtual. ➤ Responsable de la edición y realización de los sonidos. ➤ Edición de Videos. ➤ Incorporación de sonidos a los ambientes virtuales. ➤ Programa y mejora las herramientas. 	<ul style="list-style-type: none"> ➤ Herramientas de edición de sonidos.
Animador	<ul style="list-style-type: none"> ➤ Realiza las animaciones de cualquier vértice que se mueva. 	<ul style="list-style-type: none"> ➤ Lenguajes de programación. ➤ Técnicas de animación 3D.
Desarrollador de Efectos Especiales	<ul style="list-style-type: none"> ➤ Realiza todos los efectos especiales necesarios. 	<ul style="list-style-type: none"> ➤ Herramientas.

Tabla 14. Roles, responsabilidades, y conocimientos del conjunto Diseñador de media.

Entre las actividades se debe mencionar:

Definir ambiente del problema: Explica cómo declarar los objetivos del ambiente virtual, y describir el ambiente del problema. También se propone analizar el enfoque que se la dará a la aplicación. En este

caso, se analiza el ambiente virtual con las perspectivas propias del autor según el futuro usuario. Este toque artístico puede ser real o ficticio, dependiendo de la madurez que tendrán los futuros usuarios en el tema de la realidad virtual, y en el ambiente virtual en sí mismo. Durante la definición del ambiente del problema, se debe identificar los conceptos en el problema, así como la relación que existe entre ellos. Esta actividad describe cómo desarrollar un guión de contenido que contenga la línea de acción propia del mundo virtual. Dicha línea es única, y culmina en un determinado desenlace.

Definir el ambiente virtual: Describe cómo definir cada elemento, y escenario del ambiente virtual desde la perspectiva del sistema a desarrollar en cuanto a forma, nivel de detalle, comportamiento, interacción, texturización, sonido, etc. Se debe describir la composición de cada elemento y escenario del mundo virtual, así como la relación entre ellos. Elaborar especificaciones, y criterios de evaluación detallados respecto al mundo virtual. Durante esta actividad, deberá definirse que el mundo virtual está únicamente comprometido por las respuestas que debe dar a las acciones desarrolladas por el usuario. El objetivo principal de esta labor, es desarrollar un guión técnico que describa técnicamente las líneas de acción del ambiente virtual, por las cuales puede navegar el usuario. Esta actividad es preliminar al desarrollo de la aplicación, y constituye un preparativo que involucra el contacto con el computador.

Realizar diseño de media: Explica cómo construir los elementos (media) del ambiente virtual a partir de componentes, incluyendo programas de comportamiento en aquellos que lo requieran. En esta actividad, se reutilizan los elementos de similar comportamiento, realizando copias según se necesite. También se deben agrupar los productos de trabajo generados por esta tarea, en el modelo de media, según sus características o participación en la aplicación. Además, es aconsejable agrupar los elementos en una estructura jerárquica, teniendo en cuenta aquellos que poseen dinámica o rotación, y que están conformados por unidades menores.

Construir ambiente virtual: Explica el desarrollo del modelo del mundo virtual, en función de las especificaciones elaboradas, durante la producción de activos reutilizables. Debido a la necesidad de contar con mejores métodos de comunicación con el usuario, se vienen desarrollando recursos que faciliten la acción del usuario. También se realizan las copias necesarias de componentes según sea el caso. Esta actividad tiene como propósito fundamental, construir el SRV a partir de los componentes, comenzando por la composición del ambiente virtual, mediante la distribución de los componentes en el mismo.

Otros de los aportes que ofrece este trabajo, es el expediente de proyecto propuesto, y dentro de éste, los artefactos organizados en dominios, necesarios para el desarrollo de aplicaciones virtuales en el PRV. Uno de los subdominios que merecen ser mencionados es el Modelo de media, y entre las plantillas ajustadas al desarrollo de SRV en el Polo, tienen gran importancia el Guión de contenido y Guión técnico. Los productos de trabajo mencionados, fueron descritos de manera resumida anteriormente.

3.3 VALIDACIÓN

Luego de estar confeccionada la propuesta, se hace necesaria su validación para tener conocimiento del nivel de aceptación de la misma. La validación se realizó en cooperación con especialistas, mediante el procesamiento, y análisis de la información obtenida a través de encuestas, y entrevistas. El cuestionario (presentado en el anexo 2) le fue aplicado a un grupo de 8 profesionales que laboran en la UCI, entre ellos: Jefe del Polo, Jefes de Áreas Temáticas, Arquitecto principal del Polo, Jefes de proyecto, y especialistas de la Dirección General de Producción en la UCI. Los cuales fueron escogidos cuidadosamente por la especialidad, experiencia, y nivel de conocimiento en la ISW, y el desarrollo de soluciones de RV.

3.3.1 MÉTODO

Primeramente, se procedió a la elaboración de preguntas para obtener los criterios de evaluación de la propuesta. Las preguntas fueron las siguientes:

- **P1:** ¿Cree UD que la guía está a la altura de las necesidades de los proyectos de Realidad Virtual?
- **P2:** ¿Con la guía propuesta en el trabajo, cree UD que se pueda lograr eficacia en un proyecto de Realidad Virtual?
- **P3:** ¿Cree UD que la propuesta ajustada a Sistemas de Realidad Virtual es integral?
- **P4:** ¿En qué nivel de creatividad, y efectividad colocaría UD la propuesta adaptada?

Posteriormente, se determinó un comité de 8 especialistas, escogidos principalmente por los conocimientos que poseen acerca del tema que se está tratando. Coincidentemente, la mayoría de estas personas poseen cargos, o responsabilidades importantes en el PRV de la Facultad 5, y en la UCI. (Ver Anexo 19: Comité de especialistas.)

Una vez explicado, y entregado a los especialistas la propuesta para que examinen el tema. Se les hace llegar el cuestionario para que ofrezcan sus respuestas. Los resultados, se muestran en la tabla siguiente, los cuales fueron obtenidos a partir de la información presentada en el anexo 20.

Preguntas	Respuestas	Frecuencia absoluta	%
P1	Si	6	75%
	No	0	0%
	Parcialmente	2	25%
P2	Si	6	75%
	No	0	0%
	Parcialmente	2	25%
P3	Si	5	62,5%
	No	2	25%
	Parcialmente	1	12,5%
P4	Nivel bajo	0	0%
	Nivel medio	2	25%
	Nivel alto	6	75%

Tabla 15. Muestra de los cruzamientos de datos por preguntas e indicadores.

Es importante señalar, que las validaciones efectuadas en este epígrafe tienen como propósito esencial, observar el grado de aceptación que tiene la propuesta en el PRV. De manera que los resultados de esta validación, son puramente teóricos. Es decir, se realiza un procesamiento, y análisis cualitativo de la información, basado principalmente en lo que creen los especialistas sobre la validez de la guía adaptada a SRV, aún sin tener resultados reales de su aplicación en algún proyecto. Debido a la falta de tiempo fundamentalmente, para obtener resultados prácticos, y actuales de cuánto mejora el modo de producción de SRV utilizando MERVI. Como consecuencia, el nivel de la encuesta aplicada, es óptima para un mayor grado de terminación del presente trabajo. Sin embargo, la presente validación constituye un paso de avance, y valoración para decidir la puesta en marcha de una guía de desarrollo de SW ajustada a SRV, y así, continuar perfeccionando las áreas de procesos en el PRV.

En general, las personas encuestadas opinan que es necesario esperar a que se hagan pruebas de campo, para poder tener la certeza acerca de la eficacia del proceso en el PRV. Partiendo de los criterios cualitativos presentados en la tabla 15 acerca del trabajo realizado, se puede observar que la mayoría de los especialistas creen que la presente propuesta está a la altura de las necesidades de los proyectos del Polo, y que los elementos están bastante bien descritos. Dejando poco que modificar, eliminar, o agregar

en el trabajo realizado, teniendo en cuenta su alcance. También les parece que con la guía propuesta, se puede lograr eficacia en un proyecto de RV. Asimismo, el presente trabajo se coloca en un alto nivel de creatividad, y efectividad.

3.3.2 EVALUACIÓN

Tras la aplicación de la encuesta que se ha realizado, para conocer el grado de aceptación que tendrá la propuesta, se pudo recopilar las impresiones de los especialistas encuestados. Ellos plantean que un proceso de SW ajustado a las aplicaciones de RV desarrolladas en el Polo, debe apoyar la idea de que las Áreas Temáticas (AT), sean entidades que preparen personal capacitado para realizar cualquier producto relacionado con el AT en la que se trabaje. Precisamente, la propuesta, demuestra que no se puede separar a los diseñadores de media del equipo de desarrollo, y que éste, debe participar en el marketing del producto. Además, todo el proyecto debe participar en el aseguramiento de la calidad, y contar con un Gestor de la calidad dentro del grupo de trabajo. Con el tiempo, es necesario observar y mostrar resultados de cuánto mejora el trabajo, el expediente de proyecto propuesto por la guía, respecto al que actualmente es concebido por los proyectos del PRV. Asimismo, estudiar cómo se manifiesta la puesta en práctica de esta guía, respecto a la que actualmente es utilizada por cada proyecto del Polo. También, se debe tener en cuenta la valoración de algunos conceptos manejados, y el ajuste de la documentación generada por el proceso. Así como la integración entre los roles de diseños de media, los de desarrollo de guión técnico, guión de contenido y el resto de los roles de programación que son decisivos para un proceso de este tipo. Por lo que se considera necesario, poner la guía a prueba en proyectos pilotos. Al mismo tiempo, continuar el estudio del proceso, y la especificación de las actividades, hasta que llegue el momento en que pueda ser formalizado. Para dar así, una mejor validez a la propuesta, basada en su aplicación a los proyectos de RV, en aras de buscar una variante que aumente la productividad y calidad en el trabajo. La cantidad de información que se maneja en el trabajo realizado es abundante, y las propuestas contienen ideas interesantes, creativas, innovadoras, novedosas e integradoras. Resultado de una profunda investigación, y capaz de optimizar el proceso de desarrollo de aplicaciones de RV. Pues, está en total acuerdo con el proceso de desarrollo de SW que se aplica en la mayoría de los proyectos del PRV.

CONCLUSIONES PARCIALES

- Se proponen modelos para contextualizar las necesidades, e intereses del PRV.
- Se ajustan métodos y procesos, que están siendo (o han sido) utilizados por el PRV.
- Se proponen nuevos roles, artefactos y actividades, necesarios para llevar a cabo el desarrollo de SRV.
- La propuesta ha sido evaluada con un alto nivel de aceptación por especialistas.

Los proyectos del PRV no poseen una guía de desarrollo de SW ajustada a SRV, que describa el proceso para llevar a cabo el desarrollo de sus productos. Esto trae como consecuencias, que el trabajo no sea tan eficiente como se quisiera.

Durante la realización del Trabajo de Diploma “Propuesta de guía de desarrollo de software con ajustes a Sistemas de Realidad Virtual”, se logró desarrollar una guía llamada MERVI, aplicable a los proyectos del PRV. En la misma, se describen los roles, artefactos, disciplinas que se deben tener en cuenta para llevar a cabo un proceso de desarrollo de SW, configurable a los SRV.

La guía promueve el desarrollo basado en LPS y además, propone varios modelos para contextualizar el modo de producción del PRV, entre los cuales se encuentran el modelo de proceso, y el modelo de equipos de trabajo. También plantea un expediente de proyecto, para ayudar en la organización, y representación de los productos de trabajo obtenidos durante el desarrollo de aplicaciones de RV.

- Establecer comparaciones con otros procesos de desarrollo de software.
- Perfeccionar la guía propuesta, a través de su aplicación en proyectos pilotos de RV.
- Profundizar en el estudio de cada elemento de la guía, centrándose en SRV.
- Establecer métricas para medir la eficacia de la propuesta.
- Desarrollar un Entorno Integral de Desarrollo para el proceso de SW.

REFERENCIAS BIBLIOGRÁFICAS

- [Álvarez, Pita; 2007] Álvarez Zapata, Pita Hernández, "Determinación de los roles, responsabilidades y conocimientos necesarios para el Proceso de Producción de Software de Realidad Virtual", Universidad de las Ciencias Informática, Cuba, Ciudad de la Habana, 2007.
- [Clements and Northrop, 2002] Clements, P. and Northrop, L. *Software Product Lines: Practices and Patterns*. Addison Wesley, 2002
- [CMM, 1993] Paulk, M. C.; Garcia, G. M.; Chrissis, M. B. y Bush, M.: *Capability Maturity Model for Software*, version 1.1 CMU/SEI-93-TR-24, Software Engineering Institute and University Carnegie Mellon, febrero, 1993.
- [CMMI, 2008] CMMI-Development v 1.2, Seminario de CMMI, Junio, 2008.
- [Concepción, 2001] Parra Marquez, Juan Carlos; García Alvarado, Rodrigo; Santelices Malfanti, Iván, *Introducción Práctica a la Realidad Virtual*, ©Ediciones U. Bío-Bío, 2001.
- [Domínguez, 2007] Jandrich Domínguez Fortún, "Propuesta Sistema de Gestión de la Calidad", Universidad de las Ciencias Informática, Cuba, Ciudad de la Habana, 2007.
- [GUIDANCE, 2005] GUIDANCE, P., *MSF for Agile Software Development*, 2005. Disponible en: <http://www.microsoft.com/downloads/details.aspx?familyid=9F3EA426-C2B2-4264-BA0F-35A021D85234&displaylang=en>.
- [GUIDO, 2007] GUIDO, L. E. F. Guía para diseñar y procesar encuestas en organismos públicos. GOBIERNO, Subsecretaría de la Gestión Pública, Buenos Aires, Argentina 2007.
- [ISO, 2006] ISO/IEC 90003:2004, IDT, "INGENIERIA DE SOFTWARE — DIRECTIVAS PARA LA APLICACIÓN DE LA NC ISO 9001:2001 AL SOFTWARE DE COMPUTACIÓN"
- [MDE, 93] McDermid, J. y P. Rook, "Software Development Process Models", en *Software Engineer's Reference Book*, CRC Press, 1993, pp. 15126-15128
- [Pressman, 2001] Pressman Roger S.; *Ingeniería del Software. Un enfoque práctico*; 4ta ed.; España; Madrid, Carchelejo; 2001
- [RUP, 2000] JAMES RUMBAUGH, I. J., GRADY BOOCH. *El Proceso Unificado de Software (RUP)*. 2000. p.

[RUP, 2005] Rational Unified Process Versión 7.0.1 Copyright (C) IBM Corporation 1987, 2005. Reservados todos los derechos.

[Schwaber, 2001] Schwaber, Ken, Mike Beedle, *Agile Software Development with Scrum*, Prentice Hall, 2001.

[Valencia, 2006] CANÓS H. JOSÉ, P. L., PENADÉS MA. CARMEN, *Metodologías Ágiles en el Desarrollo de Software*.

- BAUER, F. L.: *Software Engineering*, Information Processing, 71, North Holland Publishing Co., Amsterdam, 1972.
- Beck, Kent, *Extreme Programming Explained*, Addison-Wesley The XP Series, 2000.
- BOEHM, B., W.: *software Engineering*, IEEE Transactions on Computers, C-25, núm. 12, diciembre, pp. 1226-1241.
- Boehm, Barry W., *A Spiral Model of Software Development and Enhancement*, IEEE Computer, Vol. 21, No. 15 (May 1988) pp. 61-72.
- Cockburn, Alistair, *Crystal "Clear": A human-powered software development methodology for small teams*, <http://members.aol.com/humansandt/crystal/clear/>, Humans and Technology, 1998- 2001.
- Davis, A., y P. Sitaram, "A Concurrent Process Model for Software Development", *Software Engineering Notes*, ACM Press, vol. 19, n. 2, Abril 1994, pp. 38-51.
- Davis, Bersoff, and Comer *IEEE/TSE* Vol.14, No. 10, October 1988.
- Durán Benejam Mayra, Sancho Fernández Miguel, Alonso Hernández Lidiexy, López Baracaldo René; "Una aproximación al perfil de Entornos Virtuales"; *UCIENCIA* 2006; Universidad de las Ciencias Informáticas; Cuba; Ciudad de la Habana; 2006.
- Hall, A., "Seven Myths of Formal Methods", *IEEE Software*, Septiembre 1990.
- Hernán Schenone Marcelo; "Diseño de una Metodología Ágil de Desarrollo de Software"; FIUBA; Argentina; Buenos Aires; 2004.
- Hernández Meléndrez Edelsys; "Cómo escribir una tesis"; Escuela Nacional de Salud Pública; 2006.
- Highsmith, Jim, *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*, Dorset House, 1999.
- ISO 9001:2000, Quality management systems — Requirements.
- Ivar Jacobson, *Concepts for Modeling Large Real Time Systems*, Chapter 2, Dissertation, Department of Computer Systems, The Royal Institute of Technology, Stockholm, Sept, 1985.
- Ivar Jacobson, Magnus Christerson, Patrick Jonhson, and Gunnar Övergaard, *Object-Oriented Software Engineering: A Use – Case Driven Approach*, Reading, MA: Addison Wesley, 1992.
- Ivar Jacobson, "Object-Oriented as a Competitive Advantage", *American Programmer*, Oct. 1992.
- Jacobson, I. 1998. "Applying UML in The Unified Process" Presentación Rational Software. Disponible en <http://www.rational.com/uml como UMLconf.zip>.
- Jerry Isdale, "What Is Virtual Reality?", <http://vr.isdale.com/WhatIsVR.html>.
- Ken; J., y R. Hunter, *Inside RAD*, McGraw-Hill, 1994.
- Ignais La Paz Trujillo; "Propuesta de Proceso de Software para sistemas de Realidad Virtual"; Universidad de las Ciencias Informática, Cuba, Ciudad de la Habana, 2007.

- Larman, Craig, *Agile and Iterative Development: A Manager's Guide*, Addison-Wesley Agile Software Development Series, 2003.
- Ledesma Carbayo Jesús; Introducción a la Realidad Virtual; Ingeniería Neurosensorial; Departamento de Ingeniería Electrónica.
- Martin, J., *Rapid Application Development*, Macmillan Inc., New York, 1991.
- Rodríguez, Fernández, Mayelín; Izquierdo, Yanes, Greicy Danay; Vila Bello, "Como producir software de Entornos Virtuales basados en la ISO 90003:2004", Universidad de las Ciencias Informática, Cuba, Ciudad de la Habana, 2007.
- Nierstrasz, "Component-Oriented Software Development", CACM, vol. 35, n. 9, Septiembre 1992, pp.160-165.
- NTP-ISO/IEC 12207:2006, *TECNOLOGÍA DE LA INFORMACIÓN. Procesos del ciclo de vida del software*, 2ª Edición, el 28 de julio de 2006.
- Philippe Coiffet, John Wiley and Sons, "Virtual Reality Technology" Grigore Burdea, New York, 1993.
- Philippe Krutchen, *The Rational Unified Process: An Introduction*, Reading, Reading, MA: Addison Wesley, 1998.
- Pere Brunet, "Sistemas y Equipos de Realidad Virtual", Universidad Politécnica de Catalunya, Febrero 2009.
- Royce, Winston, *Managing the Development of Large Software Systems*, Proceedings of IEEE Westcon, 1970.
- Stapleton, Jennifer, *DSDM Dynamic Systems Development Method*, Addison-Wesley, 1997.
- Sutherland, Jeff. ScrumWeb Home Page: *A Guide to the SCRUM Development Process*. Jeff Sutherland's Object Technology Web Page, 1996
<<http://www.tiac.net/users/jsuth/scrums/index.html>>
- Takeuchi, Hirotaka and Nonaka, Ikujiro. January-February 1986. *The New New Product Development Game*. Harvard Business Review.

- [1]<http://agilemanifesto.org>
- [2]<http://www.agilealliance.com>
- [3]<http://www.monografias.com/trabajos51/programacion-extrema/programacion-extrema.shtml>
- [4]<http://www.navegapolis.net>
- [5]<http://calidadsoft.prod.uci.cu>
- [6]http://ucipedia/index.php/Polo_de_Realidad_Virtual
- [7]<http://epf.eclipse.org/wikis/openup>
- [8]<http://atenea.ucauca.edu.co/~gramirez/archivos/AnotacionesRUP.pdf>
- [9]<http://www.upedu.org/upedu/>
- [10]<http://epf.eclipse.org/wikis/openup>
- [11]<http://eisc.univalle.edu.co/materias/WWW/material/lecturas/xp.pdf>
- [12]http://www.eclipse.org/epf/downloads/xp/xp_downloads.php
- [13]<http://www.e-gattaca.com/eContent/library/documents/DocNewsNo50DocumentNo6.PDF>
- [14]http://www.eclipse.org/downloads/download.php?file=/technology/epf/Scrum/published/scrum_published-1.5-20080820.zip
- [15]<http://vchamorro334.blogspot.es/img/grafico.jpg>
- [16]http://eva.uci.cu/file.php/102/Curso_2008-2009/Materiales_Basicos/Materiales_Basicos_Conf_1/Clase_1_Curso_Proceso_de_Desarrollo_de_Software.pdf

Ingeniería de Software

La Ingeniería de Software es capaz de brindar modelos de producción de software con el propósito fundamental de lograr productos con mayor productividad, confiabilidad y simplicidad. Es una tecnología estratificada que integra herramientas, métodos y procesos necesarios para construir, mantener y mejorar un sistema informático con calidad. Sobre la base de un enfoque sistémico que establezca prácticas, valores y principios para la aplicación disciplinada de la Ingeniería de Software.

Proceso de desarrollo de software

El proceso de desarrollo de software establece un conjunto de actividades de protección, y del marco de trabajo para un área de producción de un sistema informático, que se debe establecer para la entrega efectiva de las tecnologías de la Ingeniería de Software, en un ambiente de trabajo común del proceso.

Metodología de desarrollo de software

La metodología de desarrollo de software, es un proceso basado en la combinación de modelos de proceso, y establece principios que promuevan prácticas adaptativas. Define productos de trabajo, roles y actividades involucradas. Además, proporciona métodos para referirse a técnicas, notaciones y guías asociadas que son aplicables a las actividades del proceso de desarrollo de software.

Realidad Virtual

La Realidad Virtual es un conjunto de tecnologías capaces de simular la realidad, o realidades creadas por la fantasía a través de un ordenador. Que apoyado de periféricos y equipos especialmente fabricados, hacen que el usuario interactúe, imagine y emerja en una situación artificial de la realidad.

Sistema de Realidad Virtual

Un Sistema de Realidad Virtual es aquel que sea capaz de simular aspectos suficientes de un objeto o ambiente de forma que pueda convencer al usuario de su casi realidad. El sistema debe permitir que el usuario interactúe con él, y pueda percibir el mundo virtual con sus sentidos (vista, oído, tacto) por medio de elementos externos.

Método de Ingeniería de Software

Los métodos de la Ingeniería de Software abarcan tareas que indican cómo construir técnicamente el software. Dependen de un conjunto de principios básicos que gobiernan cada área de la tecnología e incluyen actividades de modelado y otras técnicas descriptivas.

Guía de desarrollo de software

Una guía de desarrollo de software es una descripción detallada del conjunto de actividades inherente a un proceso o procedimiento de desarrollo de SW, y que por su nivel de complejidad o por facilidad de entendimiento, requiere la aclaración de quiénes realizan dichas actividades, los productos de trabajo u otras características básicas.

Modelo de proceso de software

Un modelo de proceso de software, es una abstracción del proceso real. Es una representación simplificada de un proceso de software, la cual indica qué hacer para desarrollar software. Define el orden para las tareas o actividades involucradas, así como la coordinación, el enlace y la retroalimentación entre ellas.

A

AT: Área Temática.

B

BD: Base de Datos.

C

CMM: Modelo de Capacidad de Madurez.

CU: Caso de Uso.

D

DRA: Desarrollo Rápido de Aplicaciones.

DCS: Dirección de la Calidad del Software.

UCD: Diseño centrado en el usuario.

G

GIRV: Grupo de Investigación de Realidad Virtual.

GETC: Grupos Estudiantiles de Trabajo Científico.

H

HW: Hardware.

HMD: Head Mounted Displays (Monitor Montado en la Cabeza).

I

ISW: Ingeniería de Software.

ID: Ingeniería de Dominio.

IA: Ingeniería de Aplicaciones.

L

LOD: Level Of Detail (Nivel de Detalle).

LPS: Línea de Producto de Software.

M

MIC: Ministerio de la Informática y las Comunicaciones.

O

OO: Orientada a Objetos.

OMT: Object Modeling Technique (Técnica Modelamiento de Objetos).

P

PP: Polos Productivos .

PRV: Polo de Realidad Virtual.

PSP: Personal Software Process (Proceso Personal de Software).

R

RUP: Rational Unified Process (Proceso Unificado de Rational).

RV: Realidad Virtual.

ROP: Rational Objectory Process (Proceso de Rational Objectory).

S

SO: Sistemas Operativos.

SRV: Sistemas de Realidad Virtual.

SW: Software.

U

UCI: Universidad de Ciencias Informáticas.

UE: Ingeniería de la Usabilidad.

UML: Unified Modeling Language (Lenguaje de Modelado Unificado).

UP: Proceso Unificado.

V

VRML: Lenguaje de Marcación de Realidad Virtual.

X

XP: Programación Extrema.

Anexo 1: Encuesta realizada a los miembros de proyectos.

Compañero (a)

El Trabajo de Diploma titulado “Propuesta de guía de desarrollo de software con ajuste a Sistemas de Realidad Virtual” de los autores Yanisleis Pérez Díaz y Grenny Negret Rodríguez, se encuentra haciendo la presente encuesta para conocer criterios sobre los proyectos, específicamente sobre el proceso de desarrollo utilizado en el mismo. No es necesario que ponga su nombre, sólo le pedimos su más sincera respuesta. Lea detenidamente cada una de las preguntas y responda debidamente los correspondientes formularios.

Muchas gracias.

Necesitamos que nos diga:

1. ¿Cuál es el nombre del proyecto en el que participa?

a. _____

2. ¿Qué metodología ustedes utilizan en el desarrollo de los productos dentro del proyecto productivo?

a. RUP

b. XP

c. Scrum

d. Otras.Cuál o cuáles

3. ¿En qué fases o etapas se divide el proceso de desarrollo en su proyecto?

4. ¿Cuáles son las actividades o disciplinas dentro de las fases que se realizan en su proyecto?

5. ¿Qué roles o responsabilidades intervienen en su proyecto? Mencione los que considere más importantes.

6. ¿Qué documentación u otros artefactos se generan en su proyecto? Mencione los que considere más importantes.

7. ¿Nos podría UD describir brevemente el proceso de desarrollo realizado en su proyecto?

Anexo 2: Encuesta para la validación de la propuesta.

Compañero (a):

El Trabajo de Diploma titulado “Propuesta de guía de desarrollo de software con ajuste a Sistemas de Realidad Virtual” de los autores Yanisleis Pérez Díaz y Grenny Negret Rodriguez, se encuentra haciendo la presente encuesta para el proceso de validación de la propuesta. Le pedimos su más sincera respuesta, y por favor, lea detenidamente cada una de las preguntas.

Muchas gracias.

Necesitamos que nos diga:

1. ¿Cree UD que la guía está a la altura de las necesidades de los proyectos de Realidad Virtual?

Si ____ No ____ Parcialmente ____

2. ¿Con la guía propuesta en el presente trabajo, cree UD que se pueda lograr eficacia en un proyecto de Realidad Virtual?

Si ____ No ____ Parcialmente ____

3. ¿Cree UD que la propuesta ajustada a Sistemas de Realidad Virtual es integral? Mencione qué elemento(s) UD agregaría, modificaría o eliminaría; y argumente su respuesta.

Si ____ No ____ Parcialmente ____

4. ¿En qué escala colocaría UD la guía propuesta? Teniendo en cuenta que:

Nivel bajo de creatividad y efectividad. ____

Nivel medio de creatividad y efectividad. ____

Nivel alto de creatividad y efectividad. ____

5. ¿Qué valoración en general diera UD a la guía propuesta?

6. Datos de control

a. Nombre:

b. Contacto:

c. Categoría docente: Estudiante Profesor

d. Nivel Escolar: Técnico medio Estudios superiores

e. Año, lugar, y especialidad de graduado:

f. Lugar, y responsabilidad(es) donde labora:

g. Años, y oportunidades de experiencias adquiridas:

Anexo 3: Presentación de los resultados.

Con el propósito de evaluar la situación actual de los proyectos, y determinar la información necesaria. A continuación se presentan algunos resultados generales obtenidos de los proyectos del PRV mediante las encuestas y entrevistas. En las tablas siguientes, cada número de columna representa un proyecto enumerado de la siguiente forma:

- | | |
|---|--------------------------|
| 1. Juegos de Consola | 2. Simulador de Auto |
| 3. Entrenadores Aduana | 4. Simulador de Tiro |
| 5. Herramientas de Desarrollo para Sistemas de Realidad Virtual (HDSRV) | 6. Escenarios Virtuales |
| 7. Paseos Virtuales | 8. Meñique |
| 9. Juegos CNEURO | 10. Juegos ICAIC |
| 11. Compilación de juegos | 12. Realidad Aumentada |
| 13. Laboratorios Virtuales | 14. Simulador Quirúrgico |

Las columnas agrupadas por un color representan un AT específica de la siguiente manera:

- Simuladores
- Diseño y Animación
- Video-Juegos
- Visualización

Metodologías	1	2	3	4	5	6	7	8	9	10	11	12	13	14
RUP	x	x	x	x	x		x		x		x	x		x
OpenUP										x				
UPEDU													x	

Tabla #1. Metodologías que utilizan los proyectos del Polo de Realidad Virtual.

Flujos de trabajo	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Modelado del negocio		x	x		x							x		x
Levantamiento de requisitos	x	x	x	x	x		x		x		x	x	x	x
Análisis y diseño	x	x	x	x	x		x		x	x	x	x	x	x
Implementación	x	x	x	x	x		x		x	x	x	x	x	x
Prueba	x	x	x	x	x		x		x	x	x	x	x	x
Despliegue	x	x	x	x	x							x		x
Administración de cambios/configuración	x	x	x	x	x		x		x	x	x	x	x	x
Administración de proyecto	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Tabla #2. Flujos de trabajo que desarrollan los proyectos del Polo de Realidad Virtual.

Roles	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Líder de proyecto	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Administrador del proyecto													x	
Arquitecto de software	x		x	x	x		x		x	x	x	x	x	x
Analista del sistema													x	x
Administrador de configuración	x	x	x		x		x		x			x	x	x
Administrador de control de cambios													x	

Diseñador de casos de prueba		x	x		x		x									x	x
Diseñador de prueba																x	
Auditor de la calidad	x		x		x							x					x
Revisor																x	
Revisor técnico formal			x		x		x										x
Analista de Prueba	x															x	
Diseñador	x					x						x		x	x	x	x
Modelador						x											
Realizador 3D	x																
Diseñador de escena																x	
Desarrollador de Texturas y Materiales.									x								
Diseñador gráfico																x	
Arquitectura y tecnologías	x		x		x		x							x			x
Arquitecto de Información														x			
Escritor técnico			x		x		x										
Documentador	x									x		x					
Planificador de la calidad				x	x		x	x				x				x	x
Ingeniero de pruebas																x	
Administrador de la Calidad																x	x
Especialista Funcional																x	
Asuntos legales			x		x												
Psicólogo											x						
Rol adicional												x					
Trabajo educativo											x						
Texturizador							x										
Desarrollador Setup de personajes																	
Sonidista																	x
Analista de software	x		x	x	x						x	x	x	x	x	x	x
Integrador			x		x											x	
Desarrollador (programador)	x	x	x	x	x						x	x	x	x	x	x	x
Guionista técnico																	x
Programador Script																	
Jefe de equipo		x	x		x												x
Jefe de Modelación																	
Jefe de Texturizado																	
Diseñador de interfaz de usuario			x		x												x
Diseñador BD																	x
Probador		x	x		x												x
Investigador	x		x		x												x

Tabla #3. Roles principales que participan en los proyectos del Polo de Realidad Virtual.

Artefactos y documentación	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Diagrama de Proceso					x									
Evaluación de Áreas de la Organización				x	x					x	x			
Especificación de requisitos	x	x	x	x	x		x		x	x	x	x	x	x
Plan de Gestión de requisitos	x	x	x	x	x		x	x	x	x		x	x	x
Modelo del Dominio	x	x		x	x				x	x				
Modelo de Negocio		x	x	x						x	x	x		x
Modelo de Casos de Uso del sistema	x	x	x	x	x				x	x	x	x		x
Arquitectura de Información (AI)		x									x			
Documento de Arquitectura de Software	x	x		x	x				x	x	x	x	x	x

Modelo de Diseño	X	X	X	X	X	X	X	X	X	X	X	X	X
Código fuente	X	X	X	X	X	X	X	X	X	X	X	X	X
Manual de Usuario	X	X	X	X	X	X	X	X	X	X	X	X	X
Diseño casos de prueba	X	X	X	X	X	X	X	X	X	X	X	X	X
Plan de pruebas	X	X	X	X	X	X	X	X	X	X	X	X	X
Modelo de Despliegue	X	X	X	X	X	X	X	X	X	X	X	X	X
Plan de Instalación	X					X	X	X	X	X	X	X	X
Guión de contenidos		X	X			X	X	X	X	X	X	X	X
Guión técnico		X	X			X	X	X	X	X	X	X	X
Modelo 3D						X	X	X	X	X	X	X	X
Vídeos						X	X	X	X	X	X	X	X
Texturas						X	X	X	X	X	X	X	X
Locaciones de personajes						X	X	X	X	X	X	X	X
Scripting						X	X	X	X	X	X	X	X

Tabla #4. Artefactos fundamentales generados en los proyectos del Polo de Realidad Virtual.

Anexo 4: Plantilla de Historia de usuario.

<Logo de la Empresa> Historia de usuario													
Nombre del proyecto:	<Logo del proyecto>												
Nombre del producto:	<Logo del producto>												
Fecha: [mm/dd/aa] Versión: <x.x> Clasificación: <<clasificación>>													
Reglas de confidencialidad.													
Tabla de contenidos.													
<p>1. Introducción. [Ofrece una breve explicación del documento Historia de usuario. Su propósito, importancia, alcance (fases, y disciplinas con las que se involucra. Ya sea en proyectos actuales, o futuros), y su manejo en el proceso.]</p> <p>1.1 Glosario de términos. [definiciones, acrónimos, abreviaturas]</p> <p>1.2 Referencias.</p> <table border="1"> <thead> <tr> <th>Código</th> <th>Título</th> <th>Ubicación real</th> </tr> </thead> <tbody> <tr> <td>[1]</td> <td><Nombre de documento a referenciar></td> <td>Expediente de proyecto/...</td> </tr> </tbody> </table>		Código	Título	Ubicación real	[1]	<Nombre de documento a referenciar>	Expediente de proyecto/...						
Código	Título	Ubicación real											
[1]	<Nombre de documento a referenciar>	Expediente de proyecto/...											
<p>2. Especificaciones de Historia de Usuario. [Se describen detalles de narraciones hechas por los interesados. Si se decide tener un documento independiente para cada Historia de Usuario, en esta sección se haría referencia a ese documento]</p> <table border="1"> <thead> <tr> <th colspan="2">Especificación de Historia de Usuario</th> </tr> </thead> <tbody> <tr> <td>Número:</td> <td>Usuario:</td> </tr> <tr> <td colspan="2">Nombre historia:</td> </tr> <tr> <td>Prioridad:</td> <td>Riesgo:</td> </tr> <tr> <td>Puntos estimados:</td> <td>Iteración asignada:</td> </tr> <tr> <td colspan="2">Responsable:</td> </tr> </tbody> </table>		Especificación de Historia de Usuario		Número:	Usuario:	Nombre historia:		Prioridad:	Riesgo:	Puntos estimados:	Iteración asignada:	Responsable:	
Especificación de Historia de Usuario													
Número:	Usuario:												
Nombre historia:													
Prioridad:	Riesgo:												
Puntos estimados:	Iteración asignada:												
Responsable:													

Descripción:		
Observaciones:		
3. Otras especificaciones.		
4. Observaciones. <i>[Pueden describirse entre otras cosas las lecciones aprendidas]</i>		
5. Anexos.		
Requerido por:	Responsable:	Firma:

Anexo 5: Plantilla de Plan de gestión de requisitos.

<Logo de la Empresa> Plan de gestión de requisitos			
Nombre del proyecto:	<Logo del proyecto>		
Nombre del producto:	<Logo del producto>		
Fecha: <i>[mm/dd/aa]</i> Versión: <x.x> Clasificación: <<clasificación>>			
Reglas de confidencialidad.			
Tabla de contenidos.			
1. Introducción. <i>[Ofrece una breve explicación del documento Plan de gestión de requisitos. Su propósito, importancia, alcance (fases, y disciplinas con las que se involucra. Ya sea en proyectos actuales, o futuros), y su manejo en el proceso.]</i> 1.1 Glosario de términos. <i>[definiciones, acrónimos, abreviaturas]</i> 1.2 Referencias.			
Código	Título	Ubicación real	
<i>[1]</i>	<Nombre de documento a referenciar>	Expediente de proyecto/...	
2. Identificación de requisitos. <i>[Describe los elementos que participan en la obtención de requisitos (personas, restricciones de dominio, métodos, herramientas, etc.) Los requisitos se identifican, asociándoles un código, nombre, descripción.]</i>			
Código	Nombre	Descripción	Observación
3. Análisis y pacto de requisitos. <i>[Los requisitos se estudian respecto a: consistencia, completitud,</i>			

ambigüedad, y prioridad, riesgos, estimaciones]

Código	Consistencia	Compleitud	Ambigüedad	Prioridad	Riesgos	Estimaciones	Observación

4. Especificación de requisitos. *[Se detallan los requisitos identificados, y analizados en secciones anteriores, y se dividen en categorías (funcionalidad, usabilidad, fiabilidad, eficiencia, soporte, diseño, interfaz, licencia, estándares aplicables, componentes comprados, etc.)]*

Código	Categoría	Descripción	Observación

5. Seguimiento. *[Para cada elemento se definen un grupo de reglas o guías para aplicar la trazabilidad. También se identifican la lista de los atributos que serán utilizados para caracterizarlo. Algunos atributos pueden ser el estado (propuesto, aprobado, Rechazado, Incorporado), beneficio (crítico, importante, útil), esfuerzo (bajo, medio, alto), estabilidad (bajo, medio, alto)]*

Código	Elemento de trazabilidad	Criterio	Atributos	Descripción	Observación

6. Otras especificaciones.

7. Observaciones. *[Pueden describirse entre otras cosas las lecciones aprendidas]*

8. Anexos.

Requerido por:

Responsable:

Firma:

Anexo 6: Plantilla Visión.

<Logo de la Empresa>
Visión

Nombre del proyecto:

<Logo del proyecto>

Nombre del producto:

<Logo del producto>

Fecha: *[mm/dd/aa]*

Versión: <x.x>

Clasificación: <<clasificación>>

Reglas de confidencialidad.

Tabla de contenidos.

1. Introducción. *[Ofrece una breve explicación del documento Visión. Su propósito, importancia, alcance (fases, y disciplinas con las que se involucra. Ya sea en proyectos actuales, o futuros), y su manejo en el proceso.]*

1.1 Glosario de términos. *[definiciones, acrónimos, abreviaturas]*

1.2 Referencias.

Código	Título	Ubicación real
<i>[1]</i>	<Nombre de documento a referenciar>	<i>Expediente de proyecto/...</i>

2. Posición. *[Breve descripción de las oportunidades de comercio, los posibles problemas a resolver, y las áreas en los que el producto intenta incidir.]*

3. Descripción de Grupo Funcional. <i>[Resumen de las principales características del Mercado que motivan el proyecto. Perfil detallado de los involucrados, así como del usuario principal, y su entorno de trabajo. Lista de problemas claves con las soluciones existentes, y como son percibidas por los involucrados. Identificar las alternativas disponibles que perciben los involucrados.]</i>		
4. Visión general del producto. <i>[Esta sección proporciona en alto nivel, una visión de las capacidades del producto, interfaces con otras aplicaciones, y configuración del sistema. En cuanto a: perspectivas, capacidades, suposiciones y dependencias, costo y precio, así como características básicas del proyecto]</i>		
5. Otras especificaciones.		
6. Observaciones. <i>[Pueden describirse entre otras cosas las lecciones aprendidas]</i>		
7. Anexos.		
Requerido por:	Responsable:	Firma:

Anexo 7: Plantilla de Guión de contenido.

<Logo de la Empresa> Guión de contenido								
Nombre del proyecto:	<Logo del proyecto>							
Nombre del producto:	<Logo del producto>							
Fecha: <i>[mm/dd/aa]</i> Versión: <x.x> Clasificación: <<clasificación>>								
Reglas de confidencialidad.								
Tabla de contenidos.								
1. Introducción. <i>[Ofrece una breve explicación del documento Guión de contenido. Su propósito, importancia, alcance (fases, y disciplinas con las que se involucra. Ya sea en proyectos actuales, o futuros), y su manejo en el proceso.]</i> 1.1 Glosario de términos. <i>[definiciones, acrónimos, abreviaturas]</i> 1.2 Referencias. <table border="1"> <thead> <tr> <th>Código</th> <th>Título</th> <th>Ubicación real</th> </tr> </thead> <tbody> <tr> <td>[1]</td> <td><Nombre de documento a referenciar></td> <td>Expediente de proyecto/...</td> </tr> </tbody> </table>			Código	Título	Ubicación real	[1]	<Nombre de documento a referenciar>	Expediente de proyecto/...
Código	Título	Ubicación real						
[1]	<Nombre de documento a referenciar>	Expediente de proyecto/...						
2. Enunciado del problema. <i>[Descripción del problema, explicando de qué trata, cuáles son sus objetivos, sus antecedentes, mencionar los aspectos interesantes y/o innovadores que tenga el proyecto. Describir la estructura narrativa del ambiente del problema.]</i>								
3. Reglas/Mecánicas. <i>[Se describen las reglas que deberán ser consideradas en cada historia. En un video-juego, estas reglas suelen ser mecánicas del juego, que no son más que ideas para adornar la historia.]</i>								
4. Actores/Personajes. <i>[Se describen los personajes, que hay en cada historia. Pueden ser personas, animales, entidades inteligentes (no controlados por el usuario) u otra forma que pueda ser considerado un actor en la historia. Para cada actor se describe sus características psicológicas, físicas y morfológicas, su rol, aparición en la historia, y todos los demás elementos que lo definen.]</i>								

<p>5. Objetos/Ítems. <i>[Un objeto es cualquier cosa que aparece en la historia que no tiene actuación propia, su aparición está determinada por los actores o por algún suceso trascendente del entorno. Debe describirse todos los objetos que aparecen en la historia, explicar el rol que tiene y cómo es la interacción entre los personajes, el entorno, y ellos.]</i></p>		
<p>6. Entornos. <i>[Se trata de hacer un listado de todos los escenarios y subescenarios donde la historia se desarrolla. Hay que definir no solo el escenario sino también lo que contiene.]</i></p>		
<p>7. Clases de dominio. <i>[Identificar clases del dominio a partir de las especificaciones realizadas en las secciones anteriores. Para cada clase se identifican solo los atributos y pueden ser modeladas utilizando UML]</i></p>		
<p>8. Diagrama de clases de dominio. <i>[Realizar un diagrama de clases del dominio a partir de las especificaciones realizadas en las secciones anteriores. Donde se muestra y describe las relaciones entre las clases. Puede ser modeladas utilizando UML]</i></p>		
<p>9. Otras especificaciones.</p>		
<p>10. Observaciones. <i>[Pueden describirse entre otras cosas las lecciones aprendidas]</i></p>		
<p>11. Anexos. <i>[Aquí se deben colocar ideas y referencias en las cuales se inspiraron para realizar el proyecto. Agregar imágenes que ilustren de dónde han salido las ideas, fotos de ciudades que ayuden a comprender mejor cómo serán los escenarios. Además, no basta con poner las imágenes hay que explicar por qué una foto nos inspira y por qué ahí hay algo parecido a nuestro personaje, objeto, o escena.]</i></p>		
<p>Requerido por:</p>	<p>Responsable:</p>	<p>Firma:</p>

Anexo 8: Plantilla de Guión técnico.

<p><Logo de la Empresa> Guión de contenido</p>								
<p>Nombre del proyecto:</p>	<p><Logo del proyecto></p>							
<p>Nombre del producto:</p>	<p><Logo del producto></p>							
<p>Fecha: <i>[mm/dd/aa]</i> Versión: <x.x> Clasificación: <<clasificación>></p>								
<p>Reglas de confidencialidad.</p>								
<p>Tabla de contenidos.</p>								
<p>1. Introducción. <i>[Ofrece una breve explicación del documento Guion de contenido. Su propósito, importancia, alcance (fases, y disciplinas con las que se involucra. Ya sea en proyectos actuales, o futuros), y su manejo en el proceso.]</i></p> <p>1.1 Glosario de términos. <i>[definiciones, acrónimos, abreviaturas]</i></p> <p>1.2 Referencias.</p> <table border="1"> <thead> <tr> <th>Código</th> <th>Título</th> <th>Ubicación real</th> </tr> </thead> <tbody> <tr> <td><i>[1]</i></td> <td><Nombre de documento a referenciar></td> <td>Expediente de proyecto/...</td> </tr> </tbody> </table>			Código	Título	Ubicación real	<i>[1]</i>	<Nombre de documento a referenciar>	Expediente de proyecto/...
Código	Título	Ubicación real						
<i>[1]</i>	<Nombre de documento a referenciar>	Expediente de proyecto/...						
<p>2. Tecnologías. <i>[Explicar con qué motor se va a hacer la aplicación, para qué plataformas. Con que tecnología se cuenta para la realización de algún suceso específico.]</i></p>								

3. Modos. *[Aquí se explica los modos actuales y posibles a incorporar que tenemos en la aplicación. Los modos pueden ser para un usuario singular, para usuarios múltiples, en red, sistema on-line, si hay modos de entrenamiento, etc. También especificar qué opciones tiene el usuario en los diferentes modos.]*

4. Hardware. *[Especificar los controles y/o periféricos, que se utilizaran en dependencia del hardware que se emplee para interactuar con el entorno del sistema. También explicar de qué manera son manejados los actores y objetos en el sistema, para estos se especifica tipo, nombre, ubicación, nomenclatura, descripción.]*

Tipo	Nomenclatura	Nombre	Ubicación real	Descripción

5. Sonidos. *[Describir la ambientación sonora de la aplicación, se debe especificar qué tipo de ambiente se quiere crear en cada escena y en cada caso especificar bien el momento de aparición y duración en la escena.]*

Tipo	Nomenclatura	Nombre	Ubicación real	Descripción

6. Cinemática. *[Esta sección recoge todo aquello que no es manejable por el usuario, y es animado en el ambiente virtual. Se hacen con el objetivo de ambientar, o resaltar algo en específico. Hay que hacer una descripción general de cómo van a ser la secuencias cinemáticas, especificando si estarán realizadas con el motor gráfico, o se trata de videos generados desde un programa de 3D.]*

Tipo	Nomenclatura	Nombre	Ubicación real	Descripción

7. Interfaz de Usuario. *[Mediante la interfaz se establece la comunicación entre el software y el usuario. Por tanto, es importante describir cada elemento de la pantalla, por ejemplo: iconos, botones, componentes, o cualquier otro elemento que permita interactuar con el software. Representación simple de la interfaz de usuario de la aplicación en cuanto a formalidades y jerarquía de los contenidos de información para el usuario que tiene la interfaz. Descripción textual de cada uno de los elementos de la estructura, características, comportamiento.]*

Tipo	Nomenclatura	Nombre	Ubicación real	Descripción

8. Mapa. *[Representación en forma de árbol de módulos, pantallas, ventanas emergentes, escenarios y otros contenidos relacionados. Estos elementos se listan de forma tabular indicando su tipo, nomenclatura o indicador, nombre, ubicación real, identificación, etc.]*

Tipo	Nomenclatura	Nombre	Ubicación real	Descripción

9. Diseño. *[Representación mediante figuras, imágenes, o prototipos de cada uno de los elementos que componen el mapa, con el objetivo de verificar la ubicación de cada uno de ellos e identificar algunas observaciones.]*

10. Modelado de escenarios. *[Realizar un modelado de los escenarios del sistema, representando sus elementos funcionales, como están relacionados entre ellos, y con los actores del sistema identificados. Esto puede hacerse utilizando UML mediante diagramas de casos de usos del sistema. Se realiza una breve descripción de cada elemento del modelado]*

11. Descripción textual de escenarios. *[Se describe paso a paso cada uno de los elementos funcionales de la estructura. Los cuales equivalen a los CU en caso de realizarse un diagrama de CU. Teniendo en cuenta nombre, actores del sistema, resumen (breve descripción), precondiciones, referencias, prioridad, medias (videos, animaciones, textos, sonidos, imágenes, etc.), flujo básico de eventos, flujos alternos, postcondiciones. Para cada uno de los flujos, agregar acción actor del sistema, y respuesta del sistema, además de una vista previa para el flujo.]*

Escenario:	
------------	--

Actores:					
Resumen:					
Precondiciones:					
Referencias					
Prioridad					
Recursos					
Imágenes	Animaciones	Sonidos	Videos	Efectos Especiales	Textos
Flujo Normal de Eventos					
Sección ""					
Acción del Actor			Respuesta del Sistema		
Vista previa					
Flujos Alternos					
Acción del Actor			Respuesta del Sistema		
Vista previa					
Poscondiciones					
12. Otras especificaciones.					
13. Observaciones. <i>[Pueden describirse entre otras cosas las lecciones aprendidas]</i>					
14. Anexos.					
Requerido por:		Responsable:		Firma:	

Anexo 9: Plantilla de Caso de prueba.

<Logo de la Empresa> Caso de prueba	
Nombre del proyecto:	<Logo del proyecto>
Nombre del producto:	<Logo del producto>
Fecha: <i>[mm/dd/aa]</i> Versión: <x.x> Clasificación: <<clasificación>>	

Reglas de confidencialidad.

Tabla de contenidos.

1. **Introducción.** [Ofrece una breve explicación del documento Caso de prueba. Su propósito, importancia, alcance (fases, y disciplinas con las que se involucra. Ya sea en proyectos actuales, o futuros), y su manejo en el proceso.]

1.1 **Glosario de términos.** [definiciones, acrónimos, abreviaturas]

1.2 Referencias.

Código	Título	Ubicación real
[1]	<Nombre de documento a referenciar>	Expediente de proyecto/...

2. **Secciones.** [Se describen para cada sección los escenarios, los cuales van a estar formados por flujo básico + flujos alternos.]

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
[Nombre de la sección 1]	EC 1.1: Nombre del escenario.	[Descripción de la funcionalidad]	[Pasos a desarrollar para probar la funcionalidad indicada]
	EC 1.n: Nombre del escenario.	[Descripción de la funcionalidad]	[Pasos a desarrollar para probar la funcionalidad] indicada]
Nombre de la sección n]	EC n.1: Nombre del escenario.	[Descripción de la funcionalidad]	[Pasos a desarrollar para probar la funcionalidad indicada]
	EC n.n: Nombre del escenario.	[Descripción de la funcionalidad]	[Pasos a desarrollar para probar la funcionalidad] indicada]

3. **Variables.** [Se describen en una lista tabular, todas las variables que serán probadas Para cada variable se definirá número, nombre, clasificación, valor, descripción.]

No	Nombre de campo	Clasificación	Valor	Descripción
<Se enumera cada variable, descrita en el Escenario>	<Nombre de variable de entrada>	<La clasificación es según el componente de diseño utilizado] >	<Se especifica el valor por defecto, si puede ser nulo o no, o tener otro valor predeterminado.>	<Una breve descripción de los datos que deben introducirse (Reglas que tiene que cumplir el campo.)>

4. **Matriz de datos.** [Para cada una de las variables que se examinan, en cada uno de los escenarios de sus respectivas secciones, se describen los pasos de la prueba, y resultados esperados.]

Escenario	Variable 1 (Nombre de la variable)	Variable 2 (Nombre de la variable)	Variable N (Nombre de la variable)	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
<Nombre del escenario.>	V (Valido)	V (Valido)	V (Valido)	<Se escribe el resultado que se espera al realizar la prueba.>	<Se escribe el resultado que se obtiene al realizar la prueba. Ejemplo: Satisfactorio. No Satisfactorio.>	<Pasos a desarrollar para probar la Funcionalidad que se indicó. Ejemplo: <ul style="list-style-type: none">• Paso 1• Paso 2 >
	I(Invalido)	I(Invalido)	I(Invalido)			
	NA(Irrelevante)	NA(Irrelevante)	NA(Irrelevante)			

5. **Otras especificaciones.**

6. Observaciones. <i>[Pueden describirse entre otras cosas las lecciones aprendidas]</i>		
7. Anexos. <i>[Aquí se deben colocar ideas y referencias en las cuales se inspiraron para realizar el proyecto. Agregar imágenes que ilustren de dónde han salido las ideas, fotos de ciudades que ayuden a comprender mejor cómo serán los escenarios. Además, no basta con poner las imágenes hay que explicar por qué una foto nos inspira y por qué ahí hay algo parecido a nuestro personaje, objeto, o escena.]</i>		
Requerido por:	Responsable:	Firma:

Anexo 10: Plantilla de Arquitectura de software.

<Logo de la Empresa> Arquitectura de software								
Nombre del proyecto:	<Logo del proyecto>							
Nombre del producto:	<Logo del producto>							
Fecha: <i>[mm/dd/aa]</i> Versión: <x.x> Clasificación: <<clasificación>>								
Reglas de confidencialidad.								
Tabla de contenidos.								
<p>1. Introducción. <i>[Ofrece una breve explicación del documento Arquitectura de software. Su propósito, importancia, alcance (fases, y disciplinas con las que se involucra. Ya sea en proyectos actuales, o futuros), y su manejo en el proceso.]</i></p> <p>1.1 Glosario de términos. <i>[definiciones, acrónimos, abreviaturas]</i></p> <p>1.2 Referencias.</p> <table border="1"> <thead> <tr> <th>Código</th> <th>Título</th> <th>Ubicación real</th> </tr> </thead> <tbody> <tr> <td><i>[1]</i></td> <td><Nombre de documento a referenciar></td> <td>Expediente de proyecto/...</td> </tr> </tbody> </table>			Código	Título	Ubicación real	<i>[1]</i>	<Nombre de documento a referenciar>	Expediente de proyecto/...
Código	Título	Ubicación real						
<i>[1]</i>	<Nombre de documento a referenciar>	Expediente de proyecto/...						
<p>2. Representación Arquitectónica. <i>[Incluir en esta sección una descripción de que es la arquitectura de software para el sistema, y como esta es representada. La descripción de la representación aborda sobre el cómo la arquitectura y el diseño son representados, las convenciones de modelado y los artefactos usados para presentar la información, patrones arquitectónicos, tendencia, clasificación.]</i></p>								
<p>3. Objetivos y Restricciones Arquitectónicas. <i>[Incluir en esta sección una descripción de los requerimientos y objetivos del software que tienen un impacto significativo en la arquitectura.]</i></p>								
<p>4. Tamaño y rendimiento. <i>[Incluir en esta sección una descripción de las características de dimensiones importantes del software que afectan la arquitectura, así como las restricciones de rendimiento para la puesta a punto.]</i></p>								
<p>5. Vista de escenarios. <i>[Se incluye la realización de cada escenario o CU (descripción textual, diagramas de interacción) arquitectónicamente significativo. En esta sección se puede incluir el diagrama de casos de uso del sistema arquitectónicamente significantes. También pueden ser añadidas notas al diagrama para explicar porqué un caso de uso en específico es arquitectónicamente significativo.]</i></p> <table border="1"> <tr> <td>Escenario arquitectónicamente significativo:</td> <td colspan="2"></td> </tr> </table>			Escenario arquitectónicamente significativo:					
Escenario arquitectónicamente significativo:								

Actores:					
Resumen:					
Precondiciones:					
Referencias					
Prioridad					
Recursos					
Imágenes	Animaciones	Sonidos	Videos	Efectos Especiales	Textos
Flujo Normal de Eventos					
Sección ""					
Acción del Actor			Respuesta del Sistema		
Vista previa					
Flujos Alternos					
Acción del Actor			Respuesta del Sistema		
Vista previa					
Poscondiciones					

6. Estilos arquitectónicos. *[Incluir en esta sección los patrones arquitectónicos que se utilizan en la vista lógica, en la misma estarán expresadas las partes correspondientes de cada patrón. Se puede dar una visión general si se quiere hacer el próximo punto menos extenso.]*

7. Vista Lógica. *[Incluir la descripción de las clases más importantes, su organización en paquetes y subsistemas, y la organización de estos subsistemas en capas. La descripción se realiza a través de diagramas de clases para ilustrar la relación entre las clases arquitectónicamente significantes, subsistemas, paquetes y capas.]*

7.1 Elementos del modelo de diseño arquitectónicamente significantes. *[Incluir un diagrama con los elementos del modelo de diseño que son arquitectónicamente significantes.]*

7.2 Visión general de la arquitectura – Alineamiento de paquetes, subsistemas y capas. *[Incluir en esta sección un diagrama que ilustre la organización del modelo de diseño en capas lógicas e incluir la descripción de cada una de ellas. Enumeración de las tecnologías a aplicar por cada una de las capas y subsistemas.]*

8. Vista de procesos. *[Incluir en esta sección la descripción de las tareas (procesos, hilos, tareas programadas, eventos y notificaciones) involucrados en la ejecución del sistema. Además de incluir la ubicación de las clases y objetos necesarios para estas tareas de ser necesario. También puede incluirse un diagrama d procesos que muestra la composición de los procesos e hilos, y la distribución de las clases en estos procesos e hilos.]*

9. Vista de despliegue. *[Incluir en esta sección la descripción de la funcionalidad y capacidad de los*

nodos físicos (dispositivos y procesadores) para la mayoría de las configuraciones tanto para usuarios finales como para desarrolladores y probadores. Además ubicar las tareas (de la vista de procesos) y las capas lógicas en los nodos físicos. Esta descripción se realiza a través del diagrama de despliegue seguido por la distribución de los procesos y las capas lógicas en cada uno de los procesadores.]

10. Vista de Implementación. *[Incluir en esta sección la colección de componentes y subsistemas de implementación mediante el modelo de implementación (diagrama de componentes). Esta vista define además, ejecutables, bibliotecas, ficheros, subsistemas, dependencias entre ellos.]*

11. Vista de datos. *[Incluir en esta sección los elementos persistentes arquitectónicamente significativos en el modelo de datos. Una vista general de la tecnología usada para lograr la persistencia de las entidades del sistema además de la descripción del modelo de datos en términos de tablas, vistas, triggers y procedimientos almacenados.]*

12. Calidad. *[En esta sección, listar las dimensiones claves de calidad del sistema que forman la arquitectura. Para cada dimensión, abordar como la arquitectura soporta el requerimiento. Puede organizar esta sección por las diferentes vistas (lógica, implementación,...) o por calidad. Cuando una característica particular es importante para el sistema, por ejemplo seguridad, privacidad, el papel que juega la arquitectura en ese requisito debe ser cuidadosamente delineado en esta sección.]*

13. Otras especificaciones.

14. Observaciones. *[Pueden describirse entre otras cosas las lecciones aprendidas]*

15. Anexos.

Requerido por:

Responsable:

Firma:

Anexo 11: Plantilla de Plan de pruebas.

<Logo de la Empresa> Plan de pruebas								
Nombre del proyecto:	<Logo del proyecto>							
Nombre del producto:	<Logo del producto>							
Fecha: <i>[mm/dd/aa]</i> Versión: <i><x.x></i> Clasificación: <i><<clasificación>></i>								
Reglas de confidencialidad.								
Tabla de contenidos.								
<p>1. Introducción. <i>[Ofrece una breve explicación del documento Plan de pruebas. Su propósito, importancia, alcance (fases, y disciplinas con las que se involucra. Ya sea en proyectos actuales, o futuros), y su manejo en el proceso.]</i></p> <p>1.1 Glosario de términos. <i>[definiciones, acrónimos, abreviaturas]</i></p> <p>1.2 Referencias.</p> <table border="1"> <thead> <tr> <th>Código</th> <th>Título</th> <th>Ubicación real</th> </tr> </thead> <tbody> <tr> <td><i>[1]</i></td> <td><i><Nombre de documento a referenciar></i></td> <td><i>Expediente de proyecto/...</i></td> </tr> </tbody> </table>			Código	Título	Ubicación real	<i>[1]</i>	<i><Nombre de documento a referenciar></i>	<i>Expediente de proyecto/...</i>
Código	Título	Ubicación real						
<i>[1]</i>	<i><Nombre de documento a referenciar></i>	<i>Expediente de proyecto/...</i>						
<p>2. Cronograma. <i>[Desarrollar una planificación de las pruebas, teniendo en cuenta las tareas, responsable, participantes, recursos, fecha de ejecución de las pruebas.]</i></p>								

No.	Tarea	Caso de prueba	Fecha de ejecución	Responsable	Participantes	Recursos	Observaciones
	<i>Nombre de la tarea</i>						

3. Estrategia de pruebas. *[Se describe el flujo de trabajo que se utilizará para la ejecución de las pruebas. También para cada tipo de prueba, se provee una descripción y por qué está siendo implementada y ejecutada.]*

Tipo de prueba:	<i>[Los tipos de pruebas pueden ser de: funcionamiento, interfaz de usuario, datos y prueba de integridad de BD, ejecución, carga, stress, seguridad y control de acceso, configuración, instalación,...]</i>
Objetivo de prueba:	<i>[Explique el enfoque y la meta de la técnica en relación a los riesgos de calidad de los que se ocupa.]</i>
Técnicas:	<i>[Esboce el método de alto nivel para la técnica, posiblemente como pasos marcados con puntos en el nivel de visión general.]</i>
Criterios de aceptación:	<i>[Explique cómo será la técnica evaluada exitosamente, dando criterios específicos que pueden y serán medidos.]</i>
Otras consideraciones:	<i>[Provea una lista de cualquier suposición, restricciones, dependencias u otras consideraciones que tendrán un impacto en la técnica, como las habilidades del probador o los requisitos experimentales del recurso.]</i>

4. Evaluación de las pruebas. *[Se resumen brevemente los resultados de prueba, y se describen los criterios de evaluación para las pruebas, como clasificación de las No Conformidades, Pedidos de cambios y listas de chequeo. También puede incluir diagramas para visualizar los resultados de prueba.]*

5. Registro de prueba. *[Para cada una de las pruebas realizadas, se registra descripción, clase, nombre o identificador, estado, tipo, fecha de última ejecución, ejecutor, verificador, entrada, aceptación, resultados de última prueba.]*

Prueba de: <i>[unitaria, integración]</i>		
Nombre o identificador de prueba:		
Estado: <i>[Estado actual de la prueba. Ejemplo: Satisfactoria (Cuando arrojó los resultados esperados), Fallida(No arrojó los resultados esperados), Pendiente (Cuando se pospone su ejecución), Cancelada (Cuando se decide no realizarse)]</i>	Tipo: <i>[funcionamiento, interfaz de usuario, datos y prueba de integridad de BD, ejecución, carga, stress, seguridad y control de acceso, configuración, instalación,...]</i>	Ultima ejecución: DD/MM/AA <i>[Fecha en que la prueba se ejecutó por última vez]</i>
Ejecutado por: <i>[Nombre de la persona que ejecutó la prueba]</i>	Verificado por: <i>[Nombre de la persona que verificó la ejecución de la prueba]</i>	
Descripción: <i>[Descripción de Aspectos Generales de la prueba unitaria o de integración ya sean sus características particulares, incidencias en el momento de su desarrollo y otros aspectos relevantes]</i>		
Entrada: <i>[Dato de entrada]</i>		
Criterio de aceptación: <i>[Aceptación o condiciones de éxito de las pruebas]</i>		
Resultados: <i>[Resultados de la prueba]</i>		

6. Otras especificaciones.

7. Observaciones. <i>[Pueden describirse entre otras cosas las lecciones aprendidas]</i>		
8. Anexos.		
Requerido por:	Responsable:	Firma:

Anexo 12: Plantilla de Plan de proyecto.

<Logo de la Empresa> Plan de proyecto										
Nombre del proyecto:	<Logo del proyecto>									
Nombre del producto:	<Logo del producto>									
Fecha: <i>[mm/dd/aa]</i> Versión: <x.x> Clasificación: <<clasificación>>										
Reglas de confidencialidad.										
Tabla de contenidos.										
<p>1. Introducción. <i>[Ofrece una breve explicación del documento Plan de proyecto. Su propósito, importancia, alcance (fases, y disciplinas con las que se involucra. Ya sea en proyectos actuales, o futuros), y su manejo en el proceso.]</i></p> <p>1.2 Glosario de términos. <i>[definiciones, acrónimos, abreviaturas]</i></p> <p>1.2 Referencias.</p> <table border="1"> <thead> <tr> <th>Código</th> <th>Título</th> <th>Ubicación real</th> </tr> </thead> <tbody> <tr> <td>[1]</td> <td><Nombre de documento a referenciar></td> <td>Expediente de proyecto/...</td> </tr> </tbody> </table>			Código	Título	Ubicación real	[1]	<Nombre de documento a referenciar>	Expediente de proyecto/...		
Código	Título	Ubicación real								
[1]	<Nombre de documento a referenciar>	Expediente de proyecto/...								
<p>2. Organización. <i>[Descripción de la estructura organizativa de todos los equipos y grupos de trabajo del proyecto. También cómo el proyecto interactúa con grupos externos. Se detallan todos los roles que participan en el proyecto, en cuanto a sus responsabilidades, conocimientos, y habilidades.]</i></p> <p>2.1 Estructura organizativa. <i>[Una descripción de la estructura organizativa del equipo del proyecto, incluso la dirección y otras autoridades de la revisión.]</i></p> <p>2.2 Interfaces externas. <i>[Para cada grupo externo, identifique los nombres del contacto internos y externos.]</i></p> <p>2.3 Roles y responsabilidades. <i>[Se enuncian los roles que van a existir en el proyecto y las responsabilidades que van a tener dichos roles]</i></p> <table border="1"> <thead> <tr> <th>Rol</th> <th>Responsabilidad</th> <th>Competencias</th> <th>Equipo de trabajo</th> </tr> </thead> <tbody> <tr> <td><i>[Nombre de Rol]</i></td> <td><i>[Responsabilidad asociada al rol]</i></td> <td><i>[Listar las competencias, experiencia y formación necesaria para el desempeñar el rol.]</i></td> <td><i>[Nombre del(los) miembro(s) del equipo asignado a rol]</i></td> </tr> </tbody> </table>			Rol	Responsabilidad	Competencias	Equipo de trabajo	<i>[Nombre de Rol]</i>	<i>[Responsabilidad asociada al rol]</i>	<i>[Listar las competencias, experiencia y formación necesaria para el desempeñar el rol.]</i>	<i>[Nombre del(los) miembro(s) del equipo asignado a rol]</i>
Rol	Responsabilidad	Competencias	Equipo de trabajo							
<i>[Nombre de Rol]</i>	<i>[Responsabilidad asociada al rol]</i>	<i>[Listar las competencias, experiencia y formación necesaria para el desempeñar el rol.]</i>	<i>[Nombre del(los) miembro(s) del equipo asignado a rol]</i>							
<p>3. Estimaciones. <i>[Definir toda la información necesaria para realizar la planificación del proyecto, además del ciclo de vida del proyecto. Desarrollar estimaciones del proyecto en cuanto a: alcance, atributos de trabajo y tareas, esfuerzo y costo.]</i></p> <p>3.1 Alcance: <i>[Desarrollar una estructura desglosada del trabajo (ETD ó WBS) basada en la arquitectura del producto. Un WBS puede estar basado en los requisitos, productos de trabajo, actividades o una combinación de estos. Identificar los paquetes de trabajo, los componentes o productos que serán adquiridos externamente y los productos de trabajo que pueden ser rehusados. La identificación de los paquetes de trabajo debe realizarse con suficiente detalle para que permita especificar las tareas del proyecto, responsabilidades.]</i></p>										

3.2 Atributos de trabajo y tareas: [Establecer y mantener estimaciones de los atributos de los productos de trabajo identificados. Atributos como el tamaño, complejidad y estructura son utilizados generalmente por los métodos de estimación.]

3.3 Definir ciclo de vida del proyecto: [Describir el ciclo de vida utilizado para el desarrollo del software, incluyendo la metodología, las fases, las iteraciones y sus objetivos.]

3.4 Esfuerzo y costo: [Proporcionar estimaciones de esfuerzo y costo para el proyecto, así como la base para esas estimaciones, y los puntos y/o circunstancias en las cuales será necesaria una re-estimación.]

4. Planificación.

4.1 Hitos.

Fecha	Hitos	Comentarios
MM-DD -YYYY		

4.2 Suposiciones y restricciones. [Una lista de Suposiciones en que se basa el Plan y cualquier restricción, por ejemplo. el presupuesto, personal, el equipo, cronograma que aplica al proyecto.]

4.3 Criterios para acciones correctivas. [Describir los criterios que se utilizaron para determinar lo que constituye una importante desviación del plan de proyecto y aplicación de acciones correctivas.]

4.4 Cronograma detallado del proyecto. [Diagramas o tablas que muestra las fechas de completar las iteraciones, fases, puntos de liberaciones, demos y otros hitos.]

5. Plan de gestión de riesgos. [Desarrollar una lista tabular o plan con el nombre o número descriptivo, tipo, impacto, descripción, probabilidad, efecto, indicador (cómo monitorear o detectar que el riesgo ha ocurrido o está próximo), estrategias de mitigación de riesgos (reducir el impacto), así como el plan de contingencia (acciones si se materializa el riesgo) para cada uno de los riesgos identificados. Publicando, y haciendo visible finalmente, los riesgos más significativos.]

5.1 Identificación de riesgos.

Riesgo	Tipo de riesgo	Impacto	Descripción	Probabilidad	Efectos
	[Los tipos de riesgos pueden ser: Tecnológico, Personal, Organización, Herramienta, Requerimientos, Estimación]	[Lista de impactos en el proyecto o producto. Estimar el impacto sobre el proyecto en una escala del 1 al 5, donde 1 = bajo impacto sobre el éxito del proyecto. 5= impacto catastrófico sobre el éxito del proyecto]		[Estimar la probabilidad de ocurrencia. La probabilidad puede ser: Alta, Media, Baja, Muy alta]	[Los efectos pueden ser: Catastrófico, Serio, Tolerable, Insignificante]

5.2 Control de riesgos.

Riesgo	Indicadores	Estrategia de mitigación	Plan de contingencia	Monitoreo	Estado actual
	[Describe como monitorear o detectar que el riesgo ha ocurrido o está próximo. Incluye cosas como métricas y umbrales, resultados de prueba, eventos específicos, y así sucesivamente.]	[Describe que se hace actualmente en el proyecto para reducir el impacto del riesgo. ¿Cómo se puede evitar el riesgo?]	[Describe que curso seguirán las acciones si el riesgo se materializa: solución alternativa, reducción de su efecto, y así sucesivamente ¿Con qué planes de contingencia contamos si el riesgo se vuelve realidad?]	[¿Qué factores podemos vigilar que nos permitan ser capaces de determinar si el riesgo es más o menos probable?]	

5.3 Tareas para la gestión de riesgos. [Breve descripción de las tareas de gestión durante el proyecto. Se debe describir lo siguiente:

- La estrategia a utilizar para identificar el riesgo y cómo serán analizados y priorizados.
- Estrategias para la mitigación, evasión, y/o prevención para los riesgos más importante.
- Como se van a dar seguimiento al estado de cada riesgo significativo y las actividades de mitigación

- *Cronograma de revisión y reporte de los riesgos. La revisión de los riesgos debe formar parte de cada revisión de iteración y de aceptación de fases.]*

5.4 Elementos de riesgos a gestionar. *[Lista de los elementos de riesgo más importantes.]*

6. Plan de gestión de datos. *[Describir los requerimientos y procesos para asegurar la privacidad de los datos, incluyendo el mecanismo para archivarlos y el acceso a los mismos. Identificar cuáles datos del proyecto serán descritos, coleccionados y distribuidos. Describir la estructura del repositorio]*

7. Plan de gestión de recursos. *[Definición de los recursos del proyecto (equipamiento, materiales, herramientas) y las cantidades necesarias para llevar a cabo las actividades del proyecto basado en las estimaciones iniciales.]*

Recurso:	
Descripción	
Software Base	
Servicios	
Descripción	Observaciones

8. Plan de capacitación. *[Fundamentar la necesidad de realizar la capacitación, el alcance de la capacitación, objetivos, estrategias, acciones a desarrollar. Mostrar las fechas de completar las actividades de la capacitación en Diagramas o tablas]*

9. Plan de participación de los interesados. *[Describir actividades o acciones para identificar o involucrar a los interesados]*

10. Plan de monitoreo y control.

10.1 Plan de control de cronograma. *[Describe en un plan de control de cronograma, la estrategia para supervisar el progreso contra lo planeado y cómo tomar las acciones correctivas cuando se requiera.]*

10.2 Plan de control de presupuesto. *[Describe en un plan de control de presupuesto, la estrategia para monitorear lo gastado del presupuesto del proyecto y cómo tomar las acciones correctivas cuando se requiera.]*

10.3 Plan de reportes. *[Describe en un plan de reportes, los reportes externos e internos que serán generados y la frecuencia y distribución]*

Reporte	Frecuencia	A quién	Ubicación	Observaciones

11. Plan de cierre. *[Describe las actividades para ordenar el cierre del proyecto, reasignación del equipo, archivar los materiales del proyecto, etc. Se lleva a cabo una grafica de cierre, que se va actualizando al final de cada iteración con el rastreo del progreso en las iteraciones.]*

12. Reunión. *[Registrar en un acta los datos (horario, lugar, asunto, asistencia, acuerdos tomados, puntos tomados) de la reunión que se realizara para debatir temas acerca de la gestión de proyecto (puede ser de compromiso, estímulo, amonestación, orientación, etc.). (Ver el acta de reunión en el anexo 18)]*

13. Otras especificaciones.

14. Observaciones. *[Pueden describirse entre otras cosas las lecciones aprendidas]*

15. Anexos.

Requerido por:	Responsable:	Firma:
----------------	--------------	--------

Anexo 13: Plantilla de Plan de iteración.

<Logo de la Empresa> Plan de iteración																																					
Nombre del proyecto:						<Logo del proyecto>																															
Nombre del producto:						<Logo del producto>																															
Fecha: [mm/dd/aa] Versión: <x.x> Clasificación: <<clasificación>>																																					
Reglas de confidencialidad.																																					
Tabla de contenidos.																																					
<p>1. Introducción. [Ofrece una breve explicación del documento Plan de iteración. Su propósito, importancia, alcance (fases, y disciplinas con las que se involucra. Ya sea en proyectos actuales, o futuros), y su manejo en el proceso.]</p> <p>1.1 Glosario de términos. [definiciones, acrónimos, abreviaturas]</p> <p>1.2 Referencias.</p> <table border="1"> <thead> <tr> <th>Código</th> <th>Título</th> <th>Ubicación real</th> </tr> </thead> <tbody> <tr> <td>[1]</td> <td><Nombre de documento a referenciar></td> <td>Expediente de proyecto/...</td> </tr> </tbody> </table>								Código	Título	Ubicación real	[1]	<Nombre de documento a referenciar>	Expediente de proyecto/...																								
Código	Título	Ubicación real																																			
[1]	<Nombre de documento a referenciar>	Expediente de proyecto/...																																			
<p>2. Plan. [Diagramas detallados que muestran la estructura de desglose del trabajo, las líneas de tiempo, los objetivos intermedios, cuándo empieza la verificación, la versión, la demostración, etc. para la iteración.]</p>																																					
<p>3. Control. [Lista que describe qué entregables se desarrollan en la iteración. Esta sección se actualiza constantemente, llevando a cabo graficas para controlar el progreso de la iteración. Se realiza Gráfica individual (horas pendientes), grafica de tareas (tareas pendientes), grafica de esfuerzo (trabajo pendientes)]</p> <p>3.1 Control de activos. [Para cada activo se especifica responsable, y cuando lo realiza, el estado (terminado pendiente, en curso), la prioridad, observaciones, fecha inicio, fecha fin, estimaciones en cuanto a tiempo dedicado, y tiempo restante.]</p> <table border="1"> <thead> <tr> <th>Nombre o descripción</th> <th>Prioridad</th> <th>Responsable</th> <th>Fecha entrega</th> <th>Tamaño estimado</th> <th>Estado</th> <th>Esfuerzo estimado</th> <th>Referencia/ Ubicación</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p>3.2 Control individual. [También se controla el trabajo individual diario, teniendo en cuenta lo que cada individuo ha hecho, lo que hará, y si tiene alguna dificultad para trabajar.]</p> <table border="1"> <thead> <tr> <th>Individuo</th> <th>Tiempo dedicado</th> <th>Tiempo restante</th> <th>¿Qué ha hecho?</th> <th>¿Qué hará?</th> <th>Dificultades</th> <th>Observaciones</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>								Nombre o descripción	Prioridad	Responsable	Fecha entrega	Tamaño estimado	Estado	Esfuerzo estimado	Referencia/ Ubicación									Individuo	Tiempo dedicado	Tiempo restante	¿Qué ha hecho?	¿Qué hará?	Dificultades	Observaciones							
Nombre o descripción	Prioridad	Responsable	Fecha entrega	Tamaño estimado	Estado	Esfuerzo estimado	Referencia/ Ubicación																														
Individuo	Tiempo dedicado	Tiempo restante	¿Qué ha hecho?	¿Qué hará?	Dificultades	Observaciones																															
<p>4. Criterios de evaluación. [Funcionalidad, rendimiento, capacidad, medidas de calidad, objetivos de calidad, etc. Evaluar y describir cómo los objetivos fueron logrados.]</p>																																					
<p>5. Especificaciones suplementarias. [Mediciones de rendimiento técnico, requisitos y restricciones no funcionales que se van a tratar en esta iteración.]</p>																																					
<p>6. Reunión. [Registrar en un acta los datos (horario, lugar, asunto, asistencia, acuerdos tomados, puntos tomados) de la reunión que se realizara para decidir el comienzo, el final, o la continuación y evolución del trabajo en la iteración. (Ver el acta de reunión en el anexo 18)]</p>																																					

7. Otras especificaciones.		
8. Observaciones. <i>[Pueden describirse entre otras cosas las lecciones aprendidas]</i>		
9. Anexos.		
Requerido por:	Responsable:	Firma:

Anexo 14: Plantilla de Plan de gestión de configuración.

<Logo de la Empresa> Plan de gestión de configuración								
Nombre del proyecto:	<Logo del proyecto>							
Nombre del producto:	<Logo del producto>							
Fecha: <i>[mm/dd/aa]</i> Versión: <x.x> Clasificación: <<clasificación>>								
Reglas de confidencialidad.								
Tabla de contenidos.								
<p>1. Introducción. <i>[Ofrece una breve explicación del documento Plan de gestión de configuración. Su propósito, importancia, alcance (fases, y disciplinas con las que se involucra. Ya sea en proyectos actuales, o futuros), y su manejo en el proceso.]</i></p> <p>1.2Glosario de términos. <i>[definiciones, acrónimos, abreviaturas]</i></p> <p>1.2 Referencias.</p> <table border="1"> <thead> <tr> <th>Código</th> <th>Título</th> <th>Ubicación real</th> </tr> </thead> <tbody> <tr> <td><i>[1]</i></td> <td><Nombre de documento a referenciar></td> <td><i>Expediente de proyecto/...</i></td> </tr> </tbody> </table>			Código	Título	Ubicación real	<i>[1]</i>	<Nombre de documento a referenciar>	<i>Expediente de proyecto/...</i>
Código	Título	Ubicación real						
<i>[1]</i>	<Nombre de documento a referenciar>	<i>Expediente de proyecto/...</i>						
<p>2. Identificación de la configuración. <i>[Especificación de la identificación, formulario de control de cambios, líneas base del proyecto, bibliotecas]</i></p> <p>2.1Especificación de la configuración. <i>[Define:</i></p> <ul style="list-style-type: none"> • <i>Esquema para etiquetado y numerado de documentos y directorios</i> • <i>Cómo identificar las relaciones</i> • <i>Descripción del esquema para el seguimiento de la identificación</i> • <i>Cómo identificar las versiones y los entregables</i> • <i>Esquema para identificar hardware y software]</i> <p>2.2 Identificación para el formulario de cambios. <i>[Esquema de identificación para cada formulario, cómo nombre, código, sub proyecto]</i></p> <p>2.3 Líneas bases del proyecto. <i>[Identificación de las diferentes líneas base del proyecto</i></p> <ul style="list-style-type: none"> • <i>Cómo y cuándo van a ser creadas</i> • <i>Quién las autoriza y quién las verifica</i> • <i>El propósito</i> • <i>Cuál será su contenido (especificar Elementos de Configuración)]</i> <p>2.4 Bibliotecas. <i>[</i></p> <ul style="list-style-type: none"> • <i>Mecanismos de identificación y control utilizados</i> • <i>Número de bibliotecas y sus tipos</i> • <i>Procedimientos y planes de resguardo</i> • <i>Procesos de recuperación ante cualquier tipo de pérdida</i> • <i>Política de acceso a bibliotecas]</i> 								
3. Control de la configuración. <i>[Procedimientos para cambiar una línea base, procedimiento para</i>								

procesar pedidos de cambios y su aceptación, Comité de Control de Cambios, revisión de documentos.]

3.1 Procedimientos para cambiar una línea base. *[Se describe el procedimiento a seguir para la modificación de la Línea Base del proyecto]*

3.2 Procedimiento para procesar pedidos de cambio y su aceptación. *[Flujo para el control de cambios]*

3.3 Comité de Control de cambios. *[Miembros, Roles, Procedimientos, Mecanismos de aprobación]*

3.4 Revisión de documentos. *[Describe cómo serán manipulados los documentos para el Control de Cambios]*

4. Estado de la configuración.

4.1 Almacenamiento, manipulación y entregables del proyecto. *[Describe dónde y cuándo serán almacenados los entregables del proyecto, esto incluye especificar los mecanismos de salva]*

4.2 Reportes. *[Mención a todos los reportes que se obtendrán sobre la configuración del proyecto]*

4.3 Proceso de entregas. *[Se refiere a las entregas a los clientes del sistema]*

- *Qué se encuentra en la entrega*
- *Problemas conocidos en el entregable*
- *Mecanismo de entrega del entregable*
- *Instrucciones de instalación*

5. Control de versiones.

Fecha	Versión	Descripción	Autor
<i>[DD/MM/AA]</i>	<i>[x.x]</i>	<i>[detalles]</i>	<i>[Nombre y apellidos]</i>

6. Reuniones. *[En esta sección debe plasmarse los datos (horario, lugar, asunto, asistencia, acuerdos tomados, puntos tomados) de las reuniones realizadas para debatir temas acerca del proceso de gestión de cambios y configuración en el proyecto. (Ver el acta de reunión en el anexo 18)]*

7. Otras especificaciones.

8. Observaciones. *[Pueden describirse entre otras cosas las lecciones aprendidas]*

9. Anexos.

Requerido por:

Responsable:

Firma:

Anexo 15: Plantilla de Registro de calidad.

<Logo de la Empresa> Registro de calidad	
Nombre del proyecto:	<Logo del proyecto>
Nombre del producto:	<Logo del producto>
Fecha: <i>[mm/dd/aa]</i> Versión: <i><x.x></i> Clasificación: <i><<clasificación>></i>	
Reglas de confidencialidad.	
Tabla de contenidos.	

1. Introducción. [Ofrece una breve explicación del documento Registro de calidad. Su propósito, importancia, alcance (fases, y disciplinas con las que se involucra. Ya sea en proyectos actuales, o futuros), y su manejo en el proceso.]

1.1 Glosario de términos. [definiciones, acrónimos, abreviaturas]

1.2 Referencias.

Código	Título	Ubicación real
[1]	<Nombre de documento a referenciar>	Expediente de proyecto/...

2. Reuniones. [En esta sección debe plasmarse los datos (horario, lugar, asunto, asistencia, acuerdos tomados, puntos tomados) de las reuniones realizadas para la planificación de las revisiones y auditorías. (Ver el acta de reunión en el anexo 18)]

3. Registro de las revisiones. [Realizar un registro histórico de las revisiones ¿Qué fue revisado?, ¿Quién lo revisó?, ¿Descubrimientos, y conclusiones? Se identifican áreas problemáticas, listas de chequeo de puntos de acción (que guíen para hacer correcciones), no conformidades, y se realizan reportes para el líder de proyectos u otros involucrados.]

3.1 Identificación de la revisión. [Identifique con un ID único, y el tipo de revisión; Por ejemplo, inspección de código, revisión de requisitos, revisión de documentos.]

3.2 Artefactos y objetivos de la revisión. [Liste los artefactos que serán el tema de esta revisión y describa los objetivos de la revisión.]

3.3 Identificación de problemas y resolución. [Liste cualquier problema identificado durante la revisión. El equipo retrospectivo puede hacer recomendaciones en la resolución problemática.]

3.4 Registro de defectos y dificultades detectadas.

Elemento: <Nombre del Elemento>

Número: <#>

Revisor: <Nombre de la persona que revisa el artefacto.>

No conformidad	Aspecto correspondiente	Etapas de detección	Significativa	No Significativa	Recomendación	Estado NC	Respuesta del Equipo Desarrollo
<Descripción de la No Conformidad>	<Descripción del Aspecto correspondiente>	<Etapas de detección del error>	<X>	<X>	<X>	[Se coloca el estado de la NC y la fecha, cada vez que se revise se deja el estado anterior y se coloca el nuevo con la fecha en que se revisó.] RA: Resuelta PD: Pendiente NP: No Procede	[Esta columna se comienza a llenar a partir de la 2da iteración, y es responsabilidad del equipo de desarrollo, quien especifica la conformidad con lo encontrado o no y en caso de no proceder la no conformidad explica por qué.]

Evaluación del artefacto

< Se evalúa si es abortada la revisión del artefacto revisado. También como puede ser evaluada la calidad del artefacto revisado, la cual puede ser regular, bien >

3.5 Reportes. [Ciertos problemas o ciertas anomalías pueden ser descubiertos para las cuales un curso de la acción no puede quedar concertado por el equipo retrospectivo, y que necesita ser escalado para la resolución.]

3.6 Registro de esfuerzo. [Se registran las horas de esfuerzo gastadas, la preparación, y conductas durante las revisiones.]

4. Registro de las auditorías. [Realizar un registro histórico de las auditorías ¿Qué fue auditado?, ¿Quién lo hizo?, ¿Desviaciones, perfecciones, orientaciones, conclusiones?]

5. Otras especificaciones.

6. Observaciones. [Pueden describirse entre otras cosas las lecciones aprendidas]

7. Anexos.

Requerido por:

Responsable:

Firma:

Anexo 16: Plantilla de Plan aseguramiento de la calidad.

<Logo de la Empresa> Registro de calidad				
Nombre del proyecto:			<Logo del proyecto>	
Nombre del producto:			<Logo del producto>	
Fecha: [mm/dd/aa] Versión: <x.x> Clasificación: <<clasificación>>				
Reglas de confidencialidad.				
Tabla de contenidos.				
1. Introducción. [Ofrece una breve explicación del documento Plan aseguramiento de la calidad. Su propósito, importancia, alcance (fases, y disciplinas con las que se involucra. Ya sea en proyectos actuales, o futuros), y su manejo en el proceso.] 1.1 Glosario de términos. [definiciones, acrónimos, abreviaturas] 1.2 Referencias.				
Código	Título	Ubicación real		
[1]	<Nombre de documento a referenciar>	Expediente de proyecto/...		
2. Plan de mediciones. [En esta sección se describe todo el proceso de métricas que se realizará como producto del monitoreo del trabajo.] 2.1 Gestión de metas y submetas. [Especificación de las metas del programa de medida que se relacionan con el proyecto en términos de logros, mejoras y calidad.] 2.2 Métricas. [Enumere las métricas que serán sintetizadas en los intervalos del proyecto para apoyar las metas.]				
Métrica:				
Categoría:				
Conceptos a medir:				
Nombre	Definición	Metas	Procedimientos de análisis	Responsabilidades
[Nombre de la métrica o un sinónimo conocido.]	[Atributos de las entidades que son medibles utilizando esta métrica, cómo se calcula la métrica, y desde que métrica primitiva es calculada]	[Lista de metas relacionadas con la métrica.]	[Como se pretende usar la métrica. Precondiciones para interpretar la métrica. Valores posibles o tendencias Modelos para el análisis de tendencias y herramientas que serán utilizadas Procedimientos de calibración Almacenamiento]	[Quien realizará la recolección de los datos, preparará los reportes y analizará los datos]
2.3 Métricas primitivas. [Enumera métricas primitivas que serán recolectadas automática o manualmente para computar las métricas finales]				
Nombre:	[Nombre de la métrica Primitiva]			
Definición:	[Descripción no ambigua de la métrica en términos del entorno del proyecto.]			

Procedimiento de recolección:	<i>[Describe el procedimiento de recolección Herramienta para la recolección de los datos formularios utilizados. Puntos del ciclo de vida donde se recolectan los datos. Procedimiento de verificación a utilizar. ¿Donde se almacenarán los datos, formularios?]</i>				
Responsabilidades:	<i>[¿Quién es responsable de la recolección y verificación de los datos?]</i>				
3. Estándares y Guías. <i>[Lista de los estándares y guías utilizados por el proyecto.]</i>					
Estándar	Ubicación			Comentarios	
4. Plan de Revisiones y Auditorías. <i>[Describe brevemente cada tipo de revisión y auditoría que se llevará a cabo en el proyecto. Para cada tipo, identifique los artefactos del proyecto que serán el asunto de la revisión o auditoría. Éstos pueden incluir Revisiones Técnicas y de Gestión Conjuntas entre Cliente y Desarrollador, Revisiones y Auditorías de Proceso, Auditorías de Cliente, Revisiones Internas, Técnicas y de Gestión.]</i>					
No. de revisión	Tipo	Objetivos	Descripción (Iteración)	Fase del proyecto	Responsable
5. Procesos, Herramientas, Técnicas y Metodologías utilizadas en las actividades del Plan. <i>[Describir el proceso explícito a ser seguido para cada tipo de revisión o auditoría. Su organización puede tener un Manual de Procedimientos de Revisión y de Auditoría estándar que puede ser referenciado. Estas descripciones de los procedimientos también deben dirigir la recolección, almacenamiento y archivo de los Registros de calidad del proyecto. Deben describirse las listas de chequeo a utilizar en cada revisión y los atributos de calidad que serán abordados en cada una de ellas. Debe de plasmar la lista de todas las herramientas, técnicas y metodologías utilizadas en las actividades del proyecto.]</i>					
6. Reuniones. <i>[En esta sección debe plasmarse los datos (horario, lugar, asunto, asistencia, acuerdos tomados, puntos tomados) de las reuniones realizadas para debatir temas acerca de la calidad en el proyecto, las personas, el producto, y el proceso. (Ver el acta de reunión en el anexo 18)]</i>					
7. Otras especificaciones.					
8. Observaciones. <i>[Pueden describirse entre otras cosas las lecciones aprendidas]</i>					
9. Anexos. <i>[Incluye Métodos computacionales, tablas para la estimación, detalles de los procedimientos.]</i>					
Requerido por:	Responsable:			Firma:	

Anexo 17: Plantilla de Solicitud de cambio.

<Logo de la Empresa> Formulario de cambio	
Nombre del proyecto:	<Logo del proyecto>
Nombre del producto:	<Logo del producto>
Proyecto/Producto: <i>[Nombre o número del proyecto o producto al que pertenece el Pedido de Cambio]</i>	
Número: <i>[Número consecutivo identificador del Pedido de cambio]</i>	Tipo: <i>[Tipo de Pedido de Cambio: error o mejora]</i>

Título: <i>[Título del pedido de cambio que refleje de manera sintetizada el contenido del pedido]</i>		
Creador: <i>[Nombre de la persona que creó el pedido de cambio, y contacto de la misma]</i>		
Fecha de creación: <i>[DD/MM/AA]</i>	Prioridad: <i>[Prioridad del pedido de cambio, esta puede ser alta, media o baja en dependencia de la importancia del mismo]</i>	
Problema actual		
Descripción: <i>[Breve descripción del pedido de cambio que abarque los aspectos más importantes del mismo]</i>		
Cómo repetir: <i>[Secuencia de pasos que permitan repetir el problema]</i>		
Nuevos requerimientos: <i>[Listado de nuevos requerimientos que desean ser agregados]</i>		
Condiciones bajo las que fue observadas el problema: <i>[Condiciones de software bajo las que fue detectado el problema]</i>		
Ambiente actual: <i>[Hardware y software correspondiente al ordenador donde se detectó el problema]</i>		
Cambios propuestos:		
Acción: <i>[Explicación de la decisión tomada respecto al cambio. Pudiera ser aplazarlo para una próxima iteración del software]</i>		
Decisión tomada: <i>[Aceptar, rechazar]</i>		
Elementos de configuración afectados: <i>[Elementos de configuración que se ven afectados por el pedido de cambio]</i>		
Errores corregidos: <i>[Errores corregidos durante la implementación del pedido]</i>		
Nuevas funcionalidades: <i>[Nuevas funcionalidades que pudieron ser agregadas durante la implementación del pedido]</i>		
Resolución		
Tiempo y costo estimado del cambio propuesto: <i>[Tiempo y costo estimado por parte del equipo de desarrollo para la realización del cambio]</i>		
Desarrollador: <i>[Nombre del desarrollador del pedido de cambio]</i>		
Observaciones:		
Requerido por:	Responsable:	Firma:

Anexo 18: Plantilla de Investigación.

<Logo de la Empresa> Investigación	
Nombre del proyecto:	<Logo del proyecto>
Nombre del producto:	<Logo del producto>

Fecha: [mm/dd/aa]
Versión: <x.x>
Clasificación: <<clasificación>>

Reglas de confidencialidad.

Tabla de contenidos.

1. Introducción. [Ofrece una breve explicación del documento Investigación. Su propósito, importancia, alcance (fases, y disciplinas con las que se involucra. Ya sea en proyectos actuales, o futuros), y su manejo en el proceso.]

1.1 Glosario de términos. [definiciones, acrónimos, abreviaturas]

1.2 Referencias.

Código	Título	Ubicación real
[1]	<Nombre de documento a referenciar>	Expediente de proyecto/...

2. Estado del arte.

2.1 Marco Teórico. [Es la posición o corriente que asume el investigador ante el tema como fundamentación teórica para la justificación de su trabajo científico y la formulación de hipótesis. Aquí se concreta el trabajo de la preparación previa y se hace un análisis detallado del estado del arte, de lo que hay en la bibliografía nacional e internacional sobre el tema que se investiga, destacando la posición del investigador al respecto, haciendo su valoración crítica de cada una de las posiciones que existen.]

2.1.1 Tendencia tecnológica de avanzada. [Describir, cuando proceda, las tendencias tecnológicas de avanzada, los titulares de las tecnologías, las soluciones protegidas en el país cuya explotación infringiría los derechos de los titulares tanto en Cuba como en otros países.]

2.2 Modelo Teórico. [Es una representación ideal del objeto de investigación de acuerdo con la concepción que se tiene de la investigación a partir del análisis de la bibliografía, las indagaciones realizadas y la experiencia del investigador y da origen a la hipótesis que es el núcleo de ese modelo teórico.]

3. Diseño de la investigación. [Resumen de las principales características del Mercado que motivan el proyecto. Perfil detallado de los involucrados, así como del usuario principal, y su entorno de trabajo. Lista de problemas claves con las soluciones existentes, y como son percibidas por los involucrados. Identificar las alternativas disponibles que perciben los involucrados.]

3.1 Objetivo general.

3.2 Necesidades de la investigación.

3.2.1 Nombre de la investigación específica.

3.2.2 Objetivos específicos.

3.2.3 Funcionalidades que se benefician con la investigación.

Antecedentes y justificación. [Donde se exprese claramente: Importancia de la solución para el proyecto específico. Debe especificarse el marco teórico y el modelo teórico por lo que puede referenciar dicho contenido del documento del Estado del Arte del Producto.]

3.2.4 Metas a lograr para alcanzar los objetivos específicos. [Metas a lograr para alcanzar los objetivos específicos.]

3.2.5 Indicadores objetivamente verificables. [Evidencia objetiva que permita comprobar en qué medida se han alcanzado los resultados, el objetivo específico y el impacto del proyecto. Pueden ser cuantitativos y/o cualitativos. Definen (¿quién?) cuantifican (¿cuánto?) cualifican (¿qué tan bien?) fijan tiempos (¿para cuándo?) y determinan el lugar (¿dónde?). Proporcionan la base para evaluar la marcha del proyecto.]

3.2.6 Medios / fuentes de verificación. [Recursos que permiten validar los indicadores seleccionados. Fuentes fidedignas de datos que proporcionan la información necesaria, así como los métodos de recopilación de estos datos. Se expresan en informes, metodologías, procedimientos, normas, bases de datos, software, publicaciones, tesis de grados, patentes, prototipos, maquetas, instalaciones pilotos, resultados de encuestas y otros]

3.2.7 Premisas/ factores externos que pueden poner en riesgo la investigación. [Factores y condiciones del entorno que están fuera del control del proyecto y que son importantes para que las actividades planificadas produzcan los resultados esperados. Determinan el éxito o fracaso del mismo por lo que deben tomarse en consideración. Incluir solo los más importantes. Estos factores

se deben incorporar a los riesgos del proyecto de software.]

3.2.8 Recursos requeridos por la investigación. [Se declaran los recursos materiales, incluyendo los rubros que se consideran como inversión así como todas las actividades que requieren financiamiento, tales como: la investigación del estado de la técnica, la vigilancia tecnológica; la protección legal de los resultados en Cuba y en el extranjero; las consultorías para la elaboración de contratos de transferencia o de colaboración; el aseguramiento de la calidad; la gestión ambiental; la formación de recursos humanos; la publicación de documentos, los viajes y dietas; el pago de licencias y los gastos de celebración de eventos, entre otros aspectos importantes.]

4. Reporte de investigación.

4.1 Descripción del resultado. [Se deben describir de forma precisa y clara el resultado, como acciones o trabajo terminado. Se debe definir su novedad en el ámbito nacional y/o internacional. Debe especificarse la capacidad de protección legal que se estima, por lo que se recomienda consultar la legislación vigente al respecto.]

4.2 Sostenibilidad de resultados. [Señalar las posibilidades reales de que se apliquen los resultados. Analizar posibilidad real de que se mantengan o incrementen los beneficios derivados de la investigación una vez concluida esta. Análisis que refleje el uso potencial de la investigación en otros proyectos y prototipos de situaciones donde se pueda aplicar especificando claramente los requerimientos tecnológicos para su aplicabilidad.]

4.3 Acceso al resultado. [En caso de que el resultado sea algoritmo, componentes de software etc. cuya esencia esté reflejada fuera de este artefacto entonces indicar claramente su localización y referencia. Con vistas a su posible uso y explotación.]

4.4 Dictamen del consejo científico. [Documento de evaluación que emite el Consejo Científico del Ejecutor sobre la culminación de la investigación y la coherencia entre sus objetivos y los resultados obtenidos.]

5. Otras especificaciones.

6. Observaciones.

7. Anexos.

Requerido por:

Responsable:

Firma:

Anexo 19: Plantilla Acta de reunión.

<Logo de la Empresa> Acta de reunión	
Proyecto	<Logo del proyecto>
Producto	<Logo del producto>
Datos de reunión	
Secretario de acta: [Nombre y apellidos]	
Responsable de reunión: [Nombre y apellidos, cargo]	
Asunto: [Asunto de la reunión]	
Asistentes: [#. Nombre y apellidos, contacto]	
Lugar: [Lugar de la reunión]	Fecha: [DD/MM/AA]

Hora inicio: [HH:MM]		Hora fin: [HH:MM]	
Acuerdos tomados			
No.	Acuerdo:	Responsable:	Fecha de cumplimiento:
Puntos tratados			

Anexo 20: Comité de especialistas.

Especialistas	Descripción
E1	Ingeniero en Informática. Graduado de la CUJAE en el 2004 (Instituto Superior Politécnico José Antonio Echeverría). Cuatro años de experiencia en desarrollo de gráficos, y aplicaciones de Realidad Virtual, así como de Líder de proyecto de RV. Tiene Diplomado en Diseño y Programación de Videojuegos de la Universidad Abierta de Catalunya. Además, ha ejercido el cargo de Arquitecto Principal del Polo por un año.
E2	Ingeniero Industrial, graduado en Julio del 2005. Profesor de Matemática Aplicada. Asesor de la Dirección General de Producción. Cursó el diplomado de Docencia Universitaria y de Dirección, entre otros cursos de postgrado. Ha participado en UCIENCIA 2006 y 2007, Fórum de Ciencia y Técnica 2006 y 2007, 13 Convención Científica de Ingeniería y Arquitectura, Informática 2007, Octavo Congreso Nacional, y Cuarto Internacional de red de la investigación y docencia sobre innovación tecnológica.
E3	Ingeniero en Ciencias Informáticas, graduado en la Universidad de Ciencias Informáticas en el 2007. Actualmente es el Jefe de Área Temática de Simuladores, y líder de proyecto de Juegos de Consola. Desarrollador experimentado de proyectos de RV. Participante de eventos como Informática 2007, y UCIENCIA 2007, 2008.
E4	Ingeniero en Ciencias Informáticas, graduado en la Universidad de Ciencias Informáticas en el 2008. Actualmente es el Jefe de Área Temática de Visualización, y líder de proyecto "Simulador Quirúrgico".
E5	Ingeniero en Ciencias Informáticas, graduado con título de oro en la Universidad de las Ciencias Informáticas en el 2008. Actualmente Jefe de Área Temática de Video-Juegos.
E6	Ingeniero en Ciencias Informáticas, graduado en la Universidad de Ciencias Informáticas en el 2007. Actualmente es el Líder del proyecto "Desarrollo de juegos en K Dimensiones (DJKD)". Consta con 6 años de experiencia en el tema de RV, ha participado en el desarrollo de productos como Simuladores, y Video - Juegos. Participante de Informática 2004, 2006, 2008.
E7	Ingeniero en Ciencias Informáticas, graduado en la Universidad de Ciencias Informáticas en el 2007. Tiene 4 años de experiencia en el tema de Realidad Virtual, sobre el que ha realizado publicaciones, y participado en eventos científicos. Actualmente, es Jefe de Área Temática de Diseño y Animación. Además, está ejerciendo Maestría en Informática Aplicada a la Realidad Virtual.
E8	Graduado en Ingeniería Informática, en la Ciudad Universitaria José Antonio Echeverría (CUJAE), con 5 años de experiencia en el tema de la Gráfica Computacional. Actualmente Profesor Instructor de la Universidad de las Ciencias Informáticas impartiendo la asignatura de Gráficos por Computadora, además de ser el jefe del Polo de Realidad Virtual en la Facultad 5 de la Universidad de las Ciencias Informáticas.

Anexo 21: Resultados del trabajo con especialistas.

Preguntas	Respuestas	E1	E2	E3	E4	E5	E6	E7	E8
P1	Si	x		x	x	x	x	x	
	No								
	Parcialmente		x						x
P2	Si	x		x	x	x	x	x	
	No								
	Parcialmente		x						x
P3	Si	x	x	x	x		x		
	No							x	x
	Parcialmente					x			
P4	Nivel bajo								
	Nivel medio		x						x
	Nivel alto	x		x	x	x	x	x	