

METODOLOGÍAS ÁGILES, UNA ALTERNATIVA PARA EL DESARROLLO DE SOFTWARE



ANEXOS

PAOLA ANDREA MANQUILLO MANQUILLO

JESUS EMILIO RIVERA DAZA

Director

Ing. Carlos Alberto Ardila Albarracín

UNIVERSIDAD DEL CAUCA

FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES

Departamento de Sistemas

Línea de Investigación en Ingeniería de Software

POPAYÁN

2006

CONTENIDO

Anexo 1	1
Anexo 2	35
Anexo 3	107
Anexo 4	111
Anexo 5	130

ANEXO 1: FUNDAMENTOS SOBRE METODOLOGIAS ÁGILES

1. INTRODUCCION

En el presente anexo se documentan las metodologías ágiles que sirvieron como guía para el desarrollo de un caso de estudio empleando programación extrema y modelamiento ágil. Para una mejor comprensión de las metodologías ágiles en general se presenta una reseña de las metodologías ágiles existentes.

El proceso de desarrollo de software es considerado como un conjunto de personas, estructuras organizacionales, reglas, políticas, actividades, componentes de software, metodologías y herramientas usadas o creadas específicamente para conceptualizar, desarrollar, ofrecer un servicio, innovar o extender un producto de software, es decir la forma en que la organización realiza sus distintos proyectos de generación de software [1]

El campo del desarrollo de software ha introducido numerosas nuevas metodologías en los últimos años, de las cuales solamente unas pocas han sobrevivido para ser utilizados hoy. Estudios recientes han revelado que la práctica metodológica fuerte, con sus exigencias de planeamiento y sus técnicas de control, en muchos casos no brindaba resultados que estuvieran a la altura de sus costos en tiempo, complejidad y dinero. Investigaciones como la de Joe Nandhakumar y David Avison [2], en un trabajo de campo sobre "la ficción del desarrollo metodológico", denunciaba que las metodologías clásicas de sistemas de información "se tratan primariamente como una ficción necesaria para presentar una imagen de control o para proporcionar estatus simbólico" y que dichas metodologías son demasiado ideales, rígidas y mecanicistas para ser utilizadas al pie de la letra. Duane Truex, Richard Baskerville y Julie Travis [3] toman una posición aún más extrema y aseguran que es posible que los métodos tradicionales sean "ideales inalcanzables hipotéticos que proporcionan una guía normativa en situaciones de desarrollo utópicas". Consideran que las metodologías estándares se basan en una fijación de objetivos pasada de moda e incorrecta y que la obsesión de los ingenieros con los métodos puede ser inhibidora de una adecuada implementación, tanto a nivel de sistemas como en el plano de negocios.

Estas metodologías tradicionales no han logrado resolver una discusión que aún subsiste entre la calidad, el coste, tiempo y el impacto en los procesos de desarrollo de software, por ello, los desarrolladores de software tienen que tratar dos preguntas que prevalecen: ¿Por qué el software cuesta tanto? ¿Por qué es tan largo el tiempo para desarrollar un producto de software? Los procesos software de hoy hacen énfasis en calidad sobre eficiencia. Estudios muestran que solucionar problemas después de la introducción del producto puede costar de 60 a 100 veces más que encontrando y eliminando los problemas durante la fase del diseño [4].

Las corrientes de procesos software de hoy no tratan con eficacia las dos cuestiones principales de coste y tiempo. Las metodologías ágiles, sin embargo, son una filosofía emergente que produce nuevos procesos de desarrollo de software conocidos como "Agile Software Development". Los nuevos procesos conocidos como "desarrollo Ágil de software"

se centran más en interacciones con la gente y en el desarrollo temprano del código que en la documentación y la planeación.

Lo que las metodologías ágiles tienen en común (y lo que de aquí en adelante obrará como una definición de los mismos), es su modelo de desarrollo incremental (pequeñas entregas con ciclos rápidos), cooperativo (desarrolladores y usuarios trabajan juntos en estrecha comunicación), directo (el método es simple y fácil de aprender) y adaptativo (capaz de incorporar los cambios). Las claves de las metodologías ágiles son la velocidad y la simplicidad. De acuerdo con ello, los equipos de trabajo se concentran en obtener lo antes posible una pieza útil que implemente sólo lo que sea más urgente; de inmediato requieren realimentación de lo que han hecho y lo tienen muy en cuenta. Luego prosiguen con ciclos igualmente breves, desarrollando de manera incremental.

2. PRINCIPIOS DE LAS METODOLOGIAS AGILES

Las metodologías ágiles para el desarrollo de software estuvieron motivadas:

- (a) Por una conciencia particularmente aguda de la crisis del software.
- (b) Por la responsabilidad que se imputa a las grandes metodologías en la gestación de esa crisis.
- (c) Por el propósito de articular soluciones.

El término “ágil” aplicado al desarrollo de software nace en febrero de 2001, tras una reunión celebrada en Utah, Estados Unidos; tras esta reunión se creó *The Agile Alliance* –La Alianza Ágil- [5], una organización sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo de software ágil y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida fue el manifiesto ágil, un documento que resume la filosofía ágil. Según el manifiesto se valora [5]:

-Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas: El movimiento ágil enfatiza la relación y comunicación de los desarrolladores de software y los roles personales reflejados en los contratos, como oposición a los procesos institucionalizados y las herramientas de desarrollo. En las prácticas ágiles existentes, estas manifiestan estar muy cercanas a la interrelación de los miembros del equipo, disposición del ambiente de trabajo, y otros procedimientos que impulsan el espíritu de equipo.

-Desarrollar software que funciona más que conseguir una buena documentación: El gran objetivo del equipo de desarrollo es continuamente al final de cada lanzamiento realizar pruebas de funcionamiento de software. Nuevos lanzamientos son producidos en frecuentes intervalos, en algunos casos diariamente, pero usualmente mensualmente. Los desarrolladores son urgidos a mantener el código simple y técnicamente tan avanzado como sea posible.

-La colaboración con el cliente más que la negociación de un contrato: la relación y cooperación entre desarrolladores y clientes dio preferencia sobre el estricto contrato. Desde el punto de vista del negocio, el desarrollo ágil esta enfocado en entregar valor al negocio desde la iniciación del proyecto, así se reduce inmediatamente el riesgo de no cumplimiento del contrato.

-Responder a los cambios más que seguir estrictamente un plan: El grupo de desarrollo, comprende desarrolladores y representantes del cliente. El grupo debe estar bien informado, competente y autorizado para considerar posibles ajustes y necesidades que surgen durante el ciclo del proceso de desarrollo. Esto significa que los participantes están preparados para realizar cambios y que los contratos existentes contienen herramientas que permiten y apoyan que estos cambios se realicen.

2.1. PRINCIPIOS DE LAS METODOLOGIAS AGILES

Junto al manifiesto se han especificado los principios que rigen al grupo de metodologías ágiles [6].

1. Nuestra prioridad más alta es satisfacer al cliente a través de la entrega temprana y continua de software valioso.

2. Los requerimientos cambiantes son bienvenidos, incluso cuando llegan tarde en el desarrollo. Los procesos ágiles se adaptan al cambio en procura de una ventaja competitiva para el cliente.
3. Entregas frecuentes de software que funcione, desde un par de semanas hasta un par de meses, con preferencia por las escalas de tiempo más breves.
4. La gente de negocios y los desarrolladores deben trabajar juntos cotidianamente a través de todo el proyecto.
5. Construir proyectos en torno de individuos motivados. Darles la oportunidad y el respaldo que necesitan y procurarles confianza para que realicen la tarea.
6. La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
7. El software que funciona es la medida primaria de progreso.
8. Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante indefinidamente.
9. La atención continua a la excelencia técnica enaltece la agilidad.
10. La simplicidad (el arte de maximizar la cantidad de trabajo que no se hace) es esencial.
11. Las mejores arquitecturas, requerimientos y diseños emergen de equipos que se auto-organizan.
12. A intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo, y ajusta su conducta en consecuencia.

3. METODOLOGIAS AGILES MÁS DESTACADAS

3.1. PROGRAMACION EXTREMA, EXTREME PROGRAMMING (XP)

3.1.1 Proceso. El ciclo de vida de XP consta de cinco fases: Exploración, planeación, iteraciones, producción, mantenimiento y muerte, a continuación se describen según [7]

3.1.1.1. Fase de exploración: Los clientes ponen por escrito en las tarjetas de historia de usuario lo que desean sea incluido en el primer lanzamiento. Paralelamente a las historias de usuario se hace una exploración de las posibles soluciones, un experimento dinámico de código, realizado para determinarse como podría solucionarse el problema. Es una versión ágil de la idea de prototipo. Se le llama spike porque aborda la solución total pero es muy fino.

Historias de usuario. Las historias de usuario tienen el mismo propósito que los casos de uso pero no son lo mismo. Se usan para crear las estimaciones de tiempo en la reunión de planeación del lanzamiento. También se usan en reemplazo de un gran documento de requisitos. Las historias de usuario son escritas por el cliente en su lenguaje propio sin términos técnicos, también conducen la creación de las pruebas de aceptación automatizadas que se utilizan para verificar al usuario que la historia se ha puesto en ejecución correctamente.

Las historias de usuario se diferencian de la especificación de requisitos tradicionales en que solo proporcionan suficiente detalle para hacer una estimación de cuanto tiempo tomará la implementación de la entrega durante la reunión de planificación con bajo riesgo. Otra diferencia importante de las historias de usuario con respecto al documento de requisitos, es que las historias evitan los detalles de tecnología, de la disposición de la base de datos, y de algoritmos específicos.

3.1.1.2. Fase de planeamiento:

Reunión de planificación de lanzamiento. En la reunión de planificación se crea un plan del lanzamiento para el proyecto total. El plan del lanzamiento entonces se utiliza para crear los planes individuales de cada iteración.

La planeación se hace por tiempo o por alcance. La velocidad del proyecto se utiliza para determinar cuántas historias se pueden poner en ejecución antes de una fecha determinada (tiempo) o cuánto tiempo tomará poner en ejecución un conjunto de historias de usuario (alcance). Al planear por tiempo multiplique el número de iteraciones por la velocidad del proyecto para determinar cuántas historias de usuario pueden ser terminadas. Al planear por alcance divida las semanas totales estimadas de las historias de usuario por la velocidad del proyecto para determinar cuántas iteraciones estarán listas hasta el lanzamiento.

La base filosófica del plan de lanzamiento es que un proyecto se puede cuantificar por cuatro variables: alcance, recursos, tiempo, y calidad. El alcance es cuánto debe ser realizado. Los recursos son cuántas personas están disponibles. El tiempo es cuando el proyecto o el lanzamiento serán realizados. Y la calidad es que tan bueno y que tan bien probado será el software a desarrollar.

La gerencia puede elegir solamente 3 de las 4 variables del proyecto para seguir. Observe que bajar la calidad tiene impacto imprevisto en las otras. Esencialmente hay solamente 3 variables que usted desea realmente cambiar.

El plan de lanzamiento. Después que las historias de usuario han sido escritas se puede convocar una reunión de planificación de lanzamiento. El plan de lanzamiento especifica exactamente que historias de usuario van a ser implementadas en cada lanzamiento del sistema y las fechas de lanzamiento para cada una de ellas. Esto da un conjunto de historias de usuario para que los clientes elijan durante la reunión de planificación de la iteración lo que se pondrá en ejecución durante la iteración siguiente. Estas historias seleccionadas se traducen entonces en tareas de programación individuales para ser implementadas durante la iteración para completar las historias.

Las historias también se traducen en pruebas de aceptación durante la iteración. Estas pruebas de aceptación se corren durante esta iteración, y las iteraciones subsiguientes para verificar cuando las historias se acaban correctamente para continuar trabajando eficientemente.

Cuando la velocidad del proyecto cambia dramáticamente para un par de iteraciones o en cualquier caso para varias iteraciones es recomendable programar una reunión de planificación de lanzamiento con los clientes y crear un nuevo plan de lanzamiento.

3.1.1.3. Fase de iteraciones: La figura 1 muestra como se desarrolla una iteración en XP. El desarrollo en una iteración XP se muestra en la figura 2.

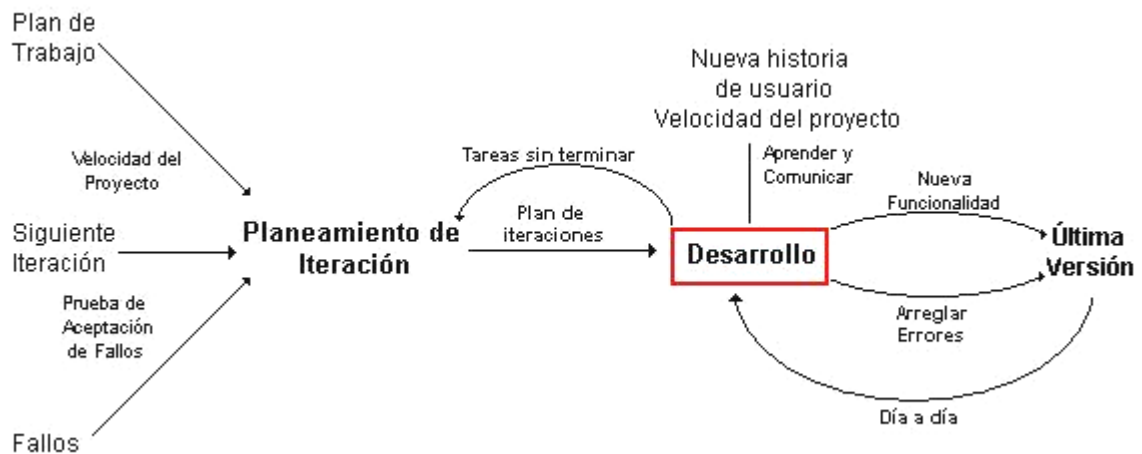


Figura 1. Iteración en XP

Desarrollo iterativo. El desarrollo iterativo adiciona agilidad al proceso de desarrollo. Divida el horario de desarrollo en aproximadamente una docena de iteraciones de una a tres semanas de duración cada una. Manteniendo la duración constante de cada iteración en todo el proyecto, este es el corazón del proyecto. Es esta constante lo que marca la medida del progreso y la planificación simple y confiable en XP.

No programar las tareas de programación por adelantado. En cambio tenga una reunión de planificación de iteración al inicio de cada iteración para planear lo que será realizado. La

planificación justo a tiempo es una manera fácil de permanecer a la altura de las exigencias cambiantes del cliente.

Esta también contra las reglas anticiparse y poner en ejecución tareas que no estén programadas para una determinada iteración. Cada funcionalidad tendrá su momento cuando esta historia de usuario se convierta en la más importante del plan de lanzamiento.

Tome los plazos de la iteración seriamente y haga un seguimiento del progreso durante cada iteración. Si se tiene la impresión de no terminar las tareas programadas por una iteración convoque otra reunión de planificación de iteración, estime nuevamente los horarios y suprima algunas tareas. Concentre todo su esfuerzo en terminar las tareas más importantes según lo elegido por el cliente, en lugar de hacer varias tareas inacabadas elegidas por los desarrolladores.

Planificación de iteración. Una reunión de planificación de iteración se convoca al principio de cada iteración para producir el plan de tareas de programación de esa iteración. Cada iteración se programa para una duración de una a tres semanas. Las historias de usuario a implementarse para cada iteración son escogidas por el cliente. También son seleccionadas las pruebas de aceptación. El cliente selecciona las historias de usuario con la velocidad del proyecto de la iteración pasada.

Las historias de usuario y las pruebas elegidas generan unas tareas de programación que deben ser analizadas. Las tareas se anotan en tarjetas como historias de usuario. Mientras que las historias de usuario están en el lenguaje del cliente las pruebas están en el lenguaje del desarrollador. Las tareas duplicadas pueden ser eliminadas. Este conjunto de tarjetas con sus respectivas tareas serán el plan detallado para cada iteración.

Los desarrolladores asignan las tareas a desarrollar y además el mismo desarrollador que tiene una tarea a su cargo, estima la duración de la misma en términos de días de programación ideales, es decir, sin interrupciones. Cada tarea se debe estimar con una duración de unos, dos o tres días de programación ideales. Las tareas de menos de un día de duración se deben agrupar y las tareas de más de tres días de duración requieren un mayor nivel de detalle.

Ahora se analiza nuevamente la velocidad del proyecto teniendo en cuenta el total de las estimaciones en los días de programación ideales, en todo caso no debe exceder la velocidad del proyecto de la iteración anterior. Si la iteración tiene demasiadas tareas entonces el cliente deberá escoger historias de usuario para ser implementadas más adelante. Si la iteración tiene muy poco entonces es posible aceptar otra historia de usuario.

Rotación de personal. La rotación de personal es importante para evitar serios cuellos de botella en la pérdida y codificación del conocimiento. Si solo una persona del equipo de desarrollo puede trabajar en un área dada y esa persona se va, el proyecto se enfrenta a numerosas tareas para ser realizadas sin la persona adecuada para hacerse cargo de ellas, esto implica una reducción de la velocidad del proyecto.

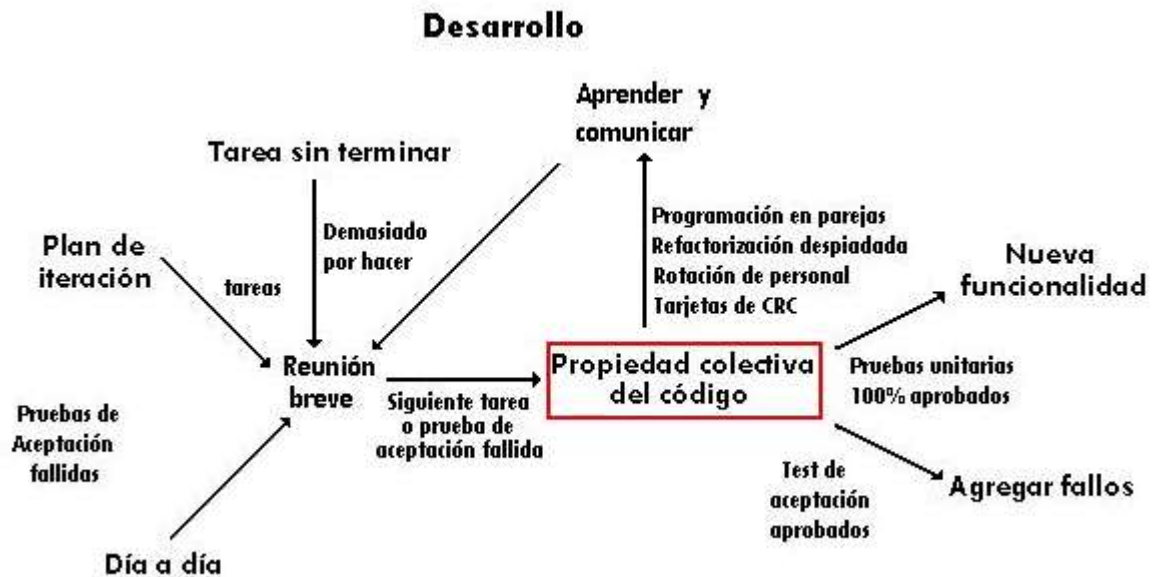


Figura 2. Desarrollo en una iteración XP.

Reunión diaria. El propósito de la reunión diaria es la comunicación entre todos los miembros del equipo, es la oportunidad para comunicar los problemas, plantear posibles soluciones y enrutarse hacia el objetivo del equipo. Es lógico, además, que es mucho más eficiente tener una sola reunión con todos los miembros del equipo y no múltiples reuniones con diferentes miembros. Es posible incluso evitar programar la mayoría de las reuniones. La mayoría de las reuniones puede ocurrir espontáneamente frente al computador, donde el código puede ser revisado y probado realmente.

Correcciones en XP. Es sabido que es necesario realizar correcciones para un proyecto específico. Se recomienda seguir las reglas XP al inicio pero no dude en cambiar aquello que no trabaja. Esto no significa que el equipo puede hacer lo que desea. Las reglas deben seguirse hasta que el equipo las cambie.

Nunca adicione funcionalidad temprano. Mantenga el sistema libre de material que se supone será utilizado mas adelante. Solamente el 10% de ese material adicional logrará ser utilizado mas adelante, así que si se llena de funcionalidades adicionales probablemente estará perdiendo el 90% de su tiempo. No todos los desarrolladores acceden a agregar funcionalidad temprano, más bien lo hacen posteriormente cuando sepan exactamente como agregarla o porque eso hace que constituya una mejora en el sistema. Puede parecer que es más eficiente agregarla ahora, pero es necesario recordar constantemente que no vamos a necesitarla realmente. Las funcionalidades adicionales provocan retrasos y malgastarán nuestros recursos.

3.1.1.4. Fase de producción: Requiere prueba y comprobación extra del funcionamiento del sistema antes de que éste se pueda lanzar al cliente. En esta fase, los nuevos cambios pueden todavía ser encontrados y debe tomarse la decisión de si se incluyen o no en el lanzamiento actual. Durante esta fase, las iteraciones pueden necesitar ser aceleradas de una a tres semanas. Las ideas y las sugerencias propuestas se documentan para una puesta en práctica posterior, por ejemplo en la fase de mantenimiento. Después de que se

realice el primer lanzamiento productivo para uso del cliente, el proyecto de XP debe mantener el funcionamiento del sistema mientras que produce nuevas iteraciones.

3.1.1.5. Fase de mantenimiento: requiere de un mayor esfuerzo para satisfacer también las tareas del cliente. Así, la velocidad del desarrollo puede desacelerar después que el sistema esté en producción. La fase de mantenimiento puede requerir la incorporación de nuevo personal y cambiar la estructura del equipo.

3.1.1.6. Fase de muerte: es cuando el cliente no tiene más historias para poner en ejecución. Esto requiere que el sistema satisfaga necesidades del cliente respecto al funcionamiento y confiabilidad. En esta etapa se escribe la documentación necesaria y ya no se realizan más cambios a la arquitectura, diseño o código elaborado. La fase de muerte también puede ocurrir si el sistema no satisface los resultados esperados o si llega a ser demasiado costoso para el desarrollo adicional.

3.1.2. ROLES Y RESPONSABILIDADES

Programador: los programadores escriben las pruebas y mantienen el código del programa tan simple y definido como sea posible. El primer aspecto para tener éxito con XP es comunicarse y coordinar con los otros miembros del equipo.

Cliente: Escribe las historias y pruebas funcionales, y decide cuando cada requerimiento es satisfecho. El cliente selecciona el orden de prioridad para los requerimientos.

Probador: Ayuda al cliente a escribir las pruebas funcionales. Realizan pruebas funcionales regularmente, difunden resultados de la prueba y mantienen las herramientas de pruebas.

Tracker/rastreador: Rastrea los estimativos hechos por el equipo, por ejemplo: esfuerzo estimado; y proporciona realimentación de cómo mejorar estimativos futuros. Hace seguimiento al progreso de cada iteración y evalúa si el objetivo es alcanzable con los recursos dados y en el tiempo límite o si son necesarios cambios en el proceso.

Coach/entrenador: es la persona responsable por el proceso completo. Está en capacidad de guiar a otros miembros del equipo en el seguimiento del proceso.

Consultor: es una persona externa al proyecto que posee el conocimiento técnico específico necesario. El consultor guía al equipo en la resolución de sus problemas específicos.

Director: es quien toma las decisiones. Para poder hacer esto, él se comunica con el equipo del proyecto para determinar la situación actual, y distinguir alguna dificultad en el proceso.

3.1.3. PRACTICAS

Comunicación: El cliente tiene que hacerse parte del equipo de trabajo, haciendo la definición de los requerimientos, permitiendo producir versiones cada 2 a 4 semanas. Esto permite también que, si el programador necesita aclarar algo en los requerimientos, se reúne con el equipo de trabajo y se resuelven sus dudas. La programación por pares y la propiedad colectiva del código (prácticas de XP que veremos más adelante) promueven la

comunicación de conocimiento técnico a través de todo el equipo. Cuando aparecen retos técnicos, el equipo es más capaz de solucionar el problema [8].

Realimentación: Mientras más rápido se identifica el cambio, más rápido se le puede manejar. En general, el principio es encontrar el error lo más cerca posible al tiempo en que fue introducido. XP se esfuerza para que se pueda recibir la realimentación lo más rápido posible en todos los aspectos del proyecto [8]. Se generan pruebas unitarias para la mayor parte del código producido. El código debe pasar las pruebas unitarias sin errores antes de ser actualizado en el repositorio central. Además se elaboran pruebas de aceptación con el cliente para verificar que la aplicación está haciendo lo que el cliente necesita que haga [8].

Simplicidad: Un diseño simple toma siempre menos tiempo para ser terminado que un diseño complejo. El objetivo es lograr que las cosas puedan trabajar tan simples como sea posible. Si encuentra algo complejo sustitúyalo por algo simple. Es siempre más rápido y más barato sustituir ahora código complejo, que perder mucho más tiempo en él.

Valentía: XP permite que el programador modifique el código existente cuando lo considere necesario, de forma responsable, sin temor a que los cambios que realice afecten la funcionalidad existente o causen errores en otras partes, esto gracias a las pruebas unitarias, que nos exigen que cada vez que se cambie algo antes de que se pueda subir al repositorio central debe pasar todas las pruebas [8]

Las doce prácticas de la programación extrema: XP cree en la posibilidad de disminuir la curva exponencial del costo de cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las siguientes prácticas:

El juego de la planificación: La planificación en XP permite contestar dos preguntas clave en el desarrollo de software: Predecir lo que se hará ahora, y determinar que se hará después. El énfasis se pone en la dirección del proyecto, más que en hacer una predicción exacta de que será necesario y cuanto tardará en hacerlo. Hay una comunicación frecuente del cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y el tiempo de las entregas [8]

Entregas pequeñas: Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Los equipos de XP entregan software a sus usuarios finales tan frecuentemente como sea posible [8]

Metáfora: La metáfora del proyecto es más o menos una descripción informal de la arquitectura del sistema; describe el sistema en conceptos simples. Los conceptos pueden ser literales o figurativos, dependiendo de la claridad del sistema actual. La idea es que en base a esta metáfora, todos los miembros del equipo, clientes y desarrolladores, puedan entender cómo trabaja el sistema de manera general [8]

Diseño simple: Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. Esto se basa en la filosofía de que el mayor valor de negocio es entregado por el programa más sencillo que cumpla los requerimientos [8]

Pruebas: Las pruebas son una práctica crucial en un proyecto XP. Una de las fuertes de XP es que hace que los proyectos sean flexibles; la flexibilidad se basa en la realimentación exacta y frecuente, y las pruebas proveen esta realimentación. En XP existen dos categorías de pruebas: pruebas unitarias y pruebas de aceptación.

Refactorización: Refactorizar ayuda a mantener el diseño simple y evita complejidad innecesaria, de modo que el código se mantiene limpio, y mucho más fácil de entender, modificar y extender. Al final toma menos tiempo producir un sistema bien preparado. La refactorización requiere aceptar que el diseño previsto en un inicio puede ser obsoleto después.

Programación en parejas: Todo el software producido con XP es construido por dos programadores, sentados uno al lado del otro en la misma máquina. Esta práctica asegura que todo el código producido es revisado por al menos un programador, y resulta en unas mejores pruebas y mejor código. Puede parecer ineficiente tener dos programadores haciendo el trabajo de uno, pero la investigación hecha acerca de programación de pares demuestra que se obtiene un código de mayor calidad en casi el mismo tiempo que un solo programador. El proceso de programar en parejas conlleva ventajas implícitas: menor tasa de errores, mejor diseño, mayor satisfacción de los programadores [8]

Propiedad colectiva del código: En un proyecto XP, cualquier par de programadores pueden mejorar cualquier parte del código en cualquier momento. Esto significa que todo el código recibe el beneficio de la atención de bastante gente, lo que incrementa la calidad de código y reduce los defectos. El uso de estándares de codificación y las pruebas dan la confianza que todo va a seguir funcionando bien tras una modificación [8] La propiedad colectiva del código es en la práctica realmente más confiable que tener solo una persona, pues esta puede dejar el proyecto en cualquier momento

Integración continua: Cada pocas horas, o máximo al cabo de un día de programación se integra el sistema completo. Para ello debe existir una máquina llamada "máquina de integración", a la que se acercará una pareja de programadores cada vez que tengan una clase que haya sido probada unitariamente. Si al añadir la nueva clase junto con sus pruebas unitarias, el sistema completo sigue funcionando correctamente, pasa todas las pruebas, los programadores darán por finalizada esa tarea. Si no, serán los responsables de dejar el sistema de nuevo con las pruebas funcionando al 100%. Si después de un cierto tiempo no son capaces de descubrir qué es lo que falla, tirarán el código a la basura y volverán a comenzar de nuevo [9]

Cuarenta horas por semana: XP promueve contar con un equipo de trabajo bien descansado. Los desarrolladores cansados tienen más propensión a cometer errores y a empezar a desear un nuevo trabajo [8]

Cliente en el sitio de trabajo: Uno de los pocos requisitos de XP, al menos uno de los clientes debe estar disponible en el sitio de trabajo. Ayuda no solo al equipo de desarrollo sino que hace parte de él. El cliente también es necesario durante las pruebas funcionales, con éstas se verifica que el sistema está listo para ser lanzado en producción. Puede suceder que el sistema no pasa todas las pruebas funcionales justo un poco antes del lanzamiento, en este caso el cliente revisa la prueba y decidirá si el sistema continúa en producción o se detiene. Si el cliente argumenta que su tiempo es demasiado valioso, deberemos entender que realmente el proyecto que nos ha encomendado es lo suficientemente trivial como para

que no merezca su atención, y que no importa que esté construido a partir de suposiciones hechas por programadores que nada, o muy poco, saben del negocio real [9].

Estándares de codificación: La codificación se debe ajustar al formato de los estándares de codificación convenidos. Los estándares de codificación mantienen el código completo fácil de leer, entender y refactorizar al equipo. Es decisivo para poder plantear con éxito la propiedad colectiva del código. Ésta sería impensable sin una codificación basada en estándares que haga que todo el mundo se sienta cómodo con el código escrito por cualquier otro miembro del equipo [9].

El mayor beneficio de las prácticas se consigue con una aplicación conjunta y equilibrada, puesto que se apoyan unas con otras. La mayoría de las prácticas propuestas por XP no son novedosas sino que de algún modo ya habían sido propuestas en ingeniería de software e incluso demostrado su valor en la práctica, el mérito de XP es integrarlas de una forma efectiva y complementarlas con otras ideas desde las perspectivas del negocio, los valores humanos y el trabajo en equipo.

3.2. SCRUM

3.2.1. PROCESO.

Fase de prejuego: esta fase incluye dos subfases: planeación y diseño de alto nivel o arquitectura.

Planeación: Incluye la definición del sistema a ser desarrollado, se crea el *Product Backlog* que contiene todos los requerimientos acumulados conocidos a la fecha.

Durante *Sprint Planning Meeting* el Propietario del producto asigna prioridades a cada uno de los ítems contenidos en el *Product Backlog*, los describe al equipo y se estima el esfuerzo necesario para la implementación de éstos. Los requisitos pueden provenir del cliente, la división de mercadeo y ventas o desarrolladores de software.

La acumulación de requisitos es constantemente actualizada con nuevos detalles, así mismo con más exactitud en las estimaciones y nuevo orden de prioridades. La planeación también incluye la definición del equipo del proyecto, herramientas y otros recursos, valoración y control de riesgos, entrenamiento necesario y verificación y aceptación de la gerencia. A cada iteración, la actualización de requisitos es revisada por el equipo Scrum para lograr el objetivo en la iteración siguiente.

Arquitectura: El diseño de alto nivel del sistema incluye la arquitectura basada en los ítems del *Product Backlog*. En caso de un mejoramiento al sistema existente, se identifican los cambios necesarios para la implementación de los ítems del *Product Backlog* con los problemas que esto puede generar.

Fase de desarrollo: también llamada fase de juego es la parte ágil de la metodología Scrum. Esta fase es tratada como una "caja negra" donde se espera lo imprevisible. Las diversas variables ambientales y técnicas (cronograma, calidad, requisitos, recursos, tecnologías y herramientas de la puesta en práctica, e incluso métodos de desarrollo), identificadas en Scrum que puede cambiar durante el proceso, se observan y se controlan con varias prácticas de Scrum durante el Sprint de la fase de desarrollo.

En esta fase el sistema se desarrolla en *Sprints*. Un *Sprint* es un ciclo iterativo donde la funcionalidad se desarrolla o se realiza para producir nuevos incrementos. Cada *Sprint* incluye las fases tradicionales del desarrollo de software: requisitos, análisis, diseño, evolución y entrega. La arquitectura y el diseño del sistema se desarrollan durante el desarrollo del *Sprint*.

Fase de post-juego: contiene la clausura del lanzamiento. Esta fase es completada con la aceptación de las variables ambientales y con el requerimiento cumplido. En este caso, ya no se encuentran más ítems por desarrollar y el sistema se encuentra listo para el lanzamiento. Esta fase incluye integración, prueba del sistema y documentación.

3.2.2. ROLES Y RESPONSABILIDADES

Existen seis roles identificados en scrum que tienen diferentes funciones y prácticas en el proceso scrum, estos se describen según [10].

Scrum Master: el Scrum Master es un nuevo rol de la gerencia introducido por scrum. El Scrum Master es responsable de asegurar que el proyecto está llevándose a cabo según las prácticas, los valores y las reglas de scrum, y que progresa según lo previsto. El scrum master obra recíprocamente con el equipo de proyecto así como con el cliente y la gerencia durante el proyecto. Él es también responsable de asegurar que cualquier impedimento sea eliminado en el proceso para mantener el funcionamiento del equipo tan productivo como sea posible.

Propietario del producto: El propietario del producto prioriza el Product Backlog. El propietario del producto es oficialmente responsable del proyecto, manejando, controlando y haciendo visible el *Product Backlog*. Este es seleccionado por el Scrum Master, el cliente y el administrador. El propietario del producto toma la decisión final de las tareas relacionadas con el *Product Backlog*, participa en la estimación del esfuerzo de desarrollo para los ítems del *Backlog* y convierte las problemáticas en características a ser desarrolladas.

Equipo Scrum: el equipo Scrum es quien tiene la autoridad para decidir sobre las acciones necesarias y organizarlas para alcanzar las metas de cada *sprint*. El equipo scrum está implicado, por ejemplo, en la estimación del esfuerzo, creación del *Sprint Backlog*, revisión del *Product Backlog* y sugiere impedimentos que necesitan ser eliminados del proyecto.

Cliente: el cliente participa en las tareas relacionadas con los ítems de la lista del *Product Backlog* a ser desarrolladas.

Director: el director es el encargado de tomar las decisiones finales, también decide los estándares y convenciones a seguir en el proyecto. Igualmente participa en la determinación de los objetivos y requerimientos. Por ejemplo, el director también participa en la elección del propietario del producto, medición del progreso y reducción del *Backlog* junto con el Scrum Master.

3.2.3. PRACTICAS

A continuación, la descripción de las prácticas de Scrum basada en [10]

Product Backlog: Define todo lo que se necesita en el producto final basado en el conocimiento actual. *Product Backlog* define el trabajo a ser realizado en el proyecto. Abarca una lista con los requerimientos técnicos y del negocio, nivel de prioridades y constantemente actualizada. Los ítems del *Backlog* pueden incluir, por ejemplo, las características, funciones, fallas del sistema a corregir, solicitudes de mejoramientos o revisión tecnológica. Múltiples actores pueden participar en la generación del *Backlog Product*, como el cliente, el equipo del proyecto, división de mercadeo y ventas, gerencia.

Estimación de esfuerzo: la estimación del esfuerzo es un proceso iterativo, en que los estimativos de los ítems del Backlog está enfocada en un nivel mas exacto cuando más información está disponible sobre cierto ítem de un producto dado. El propietario del producto conjuntamente con el equipo Scrum es responsable de realizar la estimación del esfuerzo.

Sprint: Sprint es el procedimiento de adaptación al cambio de las variables ambientales (requerimientos, tiempo, recursos, conocimiento, tecnología, etc.). El equipo Scrum se organiza para producir un nuevo incremento ejecutable del producto en un Sprint que dura aproximadamente treinta días calendario. Las herramientas de trabajo del equipo son reuniones de planeamiento del Sprint, *Sprint Backlog* y reuniones diarias de Scrum.

Reunión de planificación del Sprint: La reunión de planificación consta de dos fases organizadas por el Scrum master. Los clientes, usuarios, director, propietario del producto y equipo Scrum participan en la primera fase de la reunión para decidir sobre los objetivos y funcionalidad del nuevo sprint. La segunda fase de la reunión se mantiene entre el Scrum master y el equipo scrum, se enfoca en como el incremento del producto es implementado durante el proceso.

Sprint Backlog: es el punto de partida para cada sprint. Este es una lista de ítems del *Product Backlog* seleccionados para ser implementados en el próximo sprint. Los ítems son seleccionados por el equipo Scrum en conjunto con el Scrum Master y el propietario del producto en la reunión de planificación del sprint, sobre la base del nivel de prioridad de los ítems y los objetivos para el sprint. Distinta del *Product Backlog*, el sprint backlog es estable hasta que se termina el sprint (es decir 30 días). Cuando todos los ítems del pedido sprint son completados, se entrega una nueva iteración del sistema.

Reunión diaria scrum: las reuniones diarias o Scrum son organizadas para mantener revisión constante del progreso del equipo y también sirven como reunión de planificación: que se ha hecho desde la última reunión y que debe hacerse antes de la siguiente. Igualmente, en estas cortas reuniones de aproximadamente 15 minutos, se discuten las dificultades y otras variables. Se buscan, identifican y combaten posibles deficiencias para mejorar el proceso o impedimentos en la práctica del proceso o de la ingeniería del desarrollo del sistema. El scrum master dirige las reuniones diarias. Además del equipo Scrum también la gerencia puede participar de la reunión.

Revisión a reunión de sprint: En el último día del sprint, el equipo Scrum y el Scrum Master presentan el resultado (por ejemplo, incremento del producto trabajado) del sprint a la gerencia, clientes, usuarios y propietario del producto en una reunión informal. Los participantes evalúan el incremento del producto y toman decisiones acerca de la siguiente actividad. La reunión de revisión genera nuevos ítems e incluso cambia el rumbo del sistema que está siendo construido.

3.3. METODOLOGÍAS CRYSTAL

La familia de Metodologías Crystal incluye diversas metodologías para seleccionar la más conveniente en cada proyecto individual. Además de las metodologías el enfoque Crystal también incluye los principios para adoptar las metodologías según las circunstancias de los diversos proyectos.

La familia Crystal dispone de un código de colores para marcar la complejidad de una metodología: cuanto más oscuro un color, más "pesado" es el método. Cuanto más crítico es un sistema, más rigor se requiere. El código cromático se aplica a una forma tabular elaborada por Cockburn que se usa en muchas metodologías ágiles para situar el rango de complejidad al cual se aplica una metodología. En la figura se muestra una evaluación de las pérdidas que puede ocasionar la falla de un sistema y el método requerido según este criterio. Los parámetros son Comodidad (C), Dinero Discrecional (D), Dinero Esencial (E) y Vidas (L). En otras palabras, la caída de un sistema que ocasione incomodidades indica que su nivel de criticidad es C, mientras que si causa pérdidas de vidas su nivel es L. Los números del cuadro indican el número de personas afectadas en un proyecto. [11].

Las dimensiones de criticidad y tamaño son representados por un símbolo de la categoría del proyecto, como se describe en la figura 3. Así, por ejemplo, D6 denota un proyecto con un máximo de seis personas desarrollando un sistema de máxima criticidad del dinero discrecional.

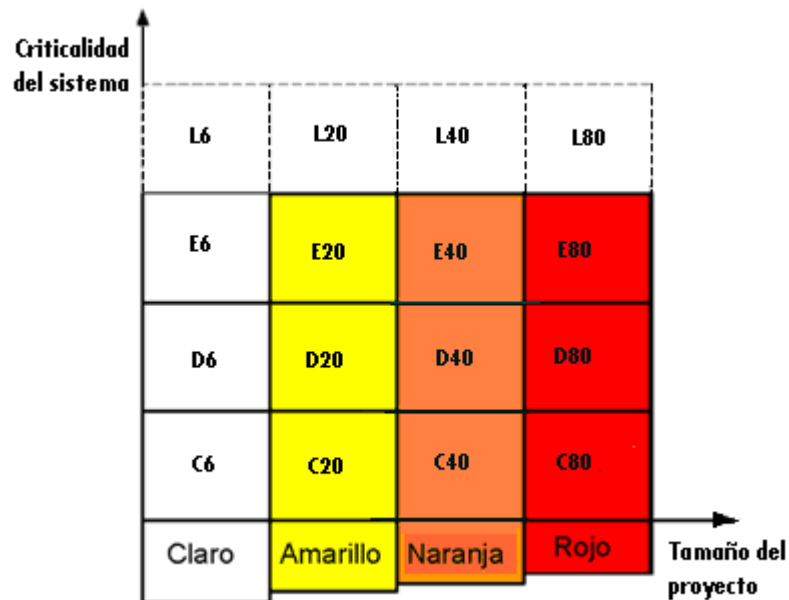


Figura 3. Familia de metodologías Crystal [11]

Existen ciertas reglas, características y valores que son comunes a todos los métodos en la familia Crystal. Primero que todo, los proyectos siempre utilizan ciclos de desarrollo incremental con una máxima duración de cuatro meses, pero preferiblemente entre uno y tres meses [12]. El énfasis está en la comunicación y la cooperación de la gente. Las Metodologías Crystal no limitan ninguna práctica de desarrollo, herramientas o productos de

trabajo, mientras que también permiten la adopción de prácticas como XP y Scrum [12]. Además, el enfoque Crystal incluye objetivos para reducir productos intermedios de trabajo y forma convenciones dentro de una metodología para los proyectos individuales y desarrollarlos mientras que el proyecto evoluciona.

3.3.1 PROCESO

Todas las metodologías de la familia Crystal proporcionan directrices de políticas, estándares, productos de trabajo, herramientas, roles a ser seguidos en el proceso de desarrollo. Crystal Clear y Crystal naranja son los dos miembros de la familia Crystal que se han construido y utilizado [13].

Crystal Clear se diseña para los proyectos muy pequeños (como los proyectos de la categoría D6), abarcando hasta seis desarrolladores. Sin embargo, con una cierta extensión a la comunicación y pruebas puede ser aplicado también a los proyectos E8/D10. Un equipo que utilice Crystal Clear debe ser situado en una oficina compartida debido a las limitaciones en su estructura de comunicación [12]. Crystal naranja se diseña para los proyectos de tamaño mediano, con un total de 10 a 40 miembros del proyecto (categoría D40), y con una duración de proyecto de uno a dos años. También los proyectos de la categoría E50 pueden ser aplicables dentro de Crystal naranja con las adiciones de procesos de verificación y pruebas. [12]

El método Crystal Clear (CC), se encuentra muy documentado y es por eso que se describe a continuación. Como casi todos los otros métodos, CC comprende valores, técnicas y procesos.

Los siete valores o propiedades de CC [14] son:

Entrega frecuente. Consiste en entregar software a los clientes con frecuencia, no solamente en compilar el código. La frecuencia dependerá del proyecto, pero puede ser diaria, semanal, mensual o lo que fuere. La entrega puede hacerse sin despliegue, si es que se consigue algún usuario cortés o curioso que suministre realimentación.

Comunicación osmótica. Todos juntos en el mismo cuarto. Una variante especial es disponer en la sala de un diseñador experto al alcance de la mano. Una reunión separada para que los concurrentes se concentren mejor es descrita como El Cono del Silencio.

Mejora reflexiva. Tomarse un pequeño tiempo (unas pocas horas en varias semanas o una vez al mes) para pensar bien qué se está haciendo, cotejar notas, reflexionar, discutir.

Seguridad personal. Hablar cuando algo molesta: decirle amigablemente al director que la agenda no es realista, o a un colega que su código necesita mejorarse. Esto es importante porque el equipo puede descubrir y reparar sus debilidades. No es provechoso encubrir los desacuerdos con gentileza y conciliación. Técnicamente, estas cuestiones se han caracterizado como una importante variable de confianza y han sido estudiadas con seriedad en la literatura.

Foco. Saber lo que se está haciendo y tener la tranquilidad y el tiempo para hacerlo. Lo primero debe venir de la comunicación sobre dirección y prioridades, típicamente con el

Patrocinador Ejecutivo. Lo segundo, de un ambiente en que la gente no se vea obligada a hacer otras cosas incompatibles.

Fácil acceso a usuarios expertos. Una comunicación de Keil a la ACM (<http://portal.acm.org/citation.cfm?id=975819>) demostró hace tiempo, sobre una amplia muestra estadística, la importancia del contacto directo con expertos en el desarrollo de un proyecto. No hay un dogma de vida o muerte sobre esto, como sí lo hay en XP. Un encuentro semanal o semi-semanal con llamadas telefónicas adicionales parece ser una buena pauta. Otra variante es que los programadores se entrenen para ser usuarios durante un tiempo. El equipo de desarrollo, de todas maneras, incluye un Experto en Negocios.

Ambiente técnico con prueba automatizada, administración de configuración e integración frecuente. Microsoft estableció la idea de los avances cotidianos, y no es una mala práctica. Muchos equipos ágiles compilan e integran varias veces al día.

Crystal Clear no requiere ninguna estrategia o técnica, pero conviene tener unas cuantas a mano para empezar. Las estrategias documentadas favoritas, comunes a otras metodologías ágiles son:

Exploración de 360°. Verificar o tomar una muestra del valor de negocios del proyecto, los requerimientos, el modelo de dominio, la tecnología, el plan del proyecto y el proceso.

Victoria temprana. Es mejor buscar pequeños triunfos iniciales que aspirar a una gran victoria tardía. La fundamentación de esta estrategia proviene de algunos estudios sociológicos específicos. Usualmente la primera victoria temprana consiste en la construcción de un Esqueleto Ambulante. Conviene no utilizar la técnica de “lo peor primero” de XP, porque puede bajar la moral. La preferencia de Cockburn es “lo más fácil primero, lo más difícil segundo”.

Esqueleto ambulante. Es una transacción que debe ser simple pero completa. Podría ser una rutina de consulta y actualización en un sistema cliente-servidor, o la ejecución de una transacción en un sistema transaccional de negocios. Un Esqueleto Ambulante no suele ser robusto; sólo camina, y carece de la carne de la funcionalidad de la aplicación real, que se agregará incrementalmente. Es diferente de un Spike, porque éste es “la más pequeña implementación que demuestra un éxito técnico plausible” y después se tira, porque puede ser desprolijo y peligroso; un Spike sólo se usa para saber si se está en la dirección correcta. El Esqueleto debe producirse con buenos hábitos de producción y pruebas de regresión, y está destinado a crecer con el sistema.

Rearquitectura incremental. Se ha demostrado que no es conveniente interrumpir el desarrollo para corregir la arquitectura. Más bien la arquitectura debe evolucionar en etapas, manteniendo el sistema en ejecución mientras ella se modifica.

Radiadores de información. Es una lámina pegada en algún lugar que el equipo pueda observar mientras trabaja o camina. Tiene que ser comprensible para el observador casual, entendida de un vistazo y renovada periódicamente para que valga la pena visitarla. Puede estar en una página Web, pero es mejor si está en una pared, porque en una máquina se acumulan tantas cosas que ninguna llama la atención. Podría mostrar el conjunto de la iteración actual, el número de pruebas pasadas o pendientes, el número de casos de uso o historias entregadas, el estado de los servidores, etc.

En cuanto a las técnicas, se favorecen:

Entrevistas a proyectos. Se suele entrevistar a más de un responsable para tener visiones más ricas. Cockburn ha elaborado una útil plantilla de dos páginas con entrevistas que probó ser útil en diversos proyectos. La idea es averiguar cuales son las prioridades, obtener una lista de rasgos deseados, saber cuáles son los requerimientos más críticos y cuáles los más negociables. Si se trata de una actualización o corrección, saber cuáles son las cosas que se hicieron bien y merecen preservarse y los errores que no se quieren repetir.

Talleres de reflexión. El equipo debe detenerse treinta minutos o una hora para reflexionar sobre sus convenciones de trabajo, discutir inconvenientes y mejoras y planificar para el período siguiente. De aquí puede salir material para poner en un póster como Radiador de Información.

Planeamiento Blitz. Se ponen tarjetas indexadas en una mesa, con una historia de usuario o función visible en cada una. El grupo finge que no hay dependencias entre tarjetas, y las alinea en secuencias de desarrollo preferidas. Los programadores escriben en cada tarjeta el tiempo estimado para desarrollar cada función. El patrocinador o embajador del usuario escribe la secuencia de prioridades, teniendo en cuenta los tiempos referidos y el valor de negocio de cada función. Las tarjetas se agrupan en períodos de tres semanas llamados iteraciones que se agrupan en entregas (lanzamientos), usualmente no más largas de tres meses [7].

Estimación Delphi con estimaciones de pericia. En el proceso Delphi se reúnen los expertos responsables y proceden como en un remate para proponer el tamaño del sistema, su tiempo de ejecución, la fecha de las entregas según dependencias técnicas y de negocios y para equilibrar las entregas en paquetes de igual tamaño.

Encuentros diarios de pie. La palabra clave es "brevedad", cinco a diez minutos como máximo. No se trata de discutir problemas, sino de identificarlos. Los problemas sólo se discuten en otros encuentros posteriores, con la gente que tiene que ver en ellos. La técnica se origina en Scrum. Se deben hacer de pie para mantener dinamismo.

Miniatura de procesos. La "Hora Extrema" fue inventada por Peter Merel para introducir a la gente en XP en 60 minutos [15] y proporciona lineamientos canónicos que pueden usarse para articular esta práctica.

Gráficos de quemado. Se trata de una técnica de graficación para descubrir demoras y problemas tempranamente en el proceso, evitando que se descubra demasiado tarde que todavía no se sabe cuánto falta. Para ello se hace una estimación del tiempo faltante para programar lo que resta al ritmo actual, lo cual sirve para tener dominio de proyectos en los cuales las prioridades cambian bruscamente y con frecuencia.

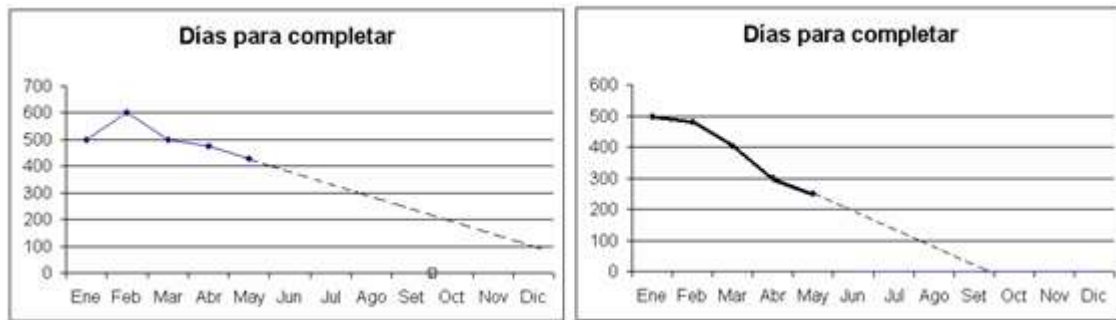


Figura 4: gráficos de quemado – Con necesidad de recortar retrasos (izq.) y con entrega proyectada en término. Medición realizada en mayo – La fecha de entrega proyectada es el 1° de octubre

Programación por pares: Mucha gente siente que la programación en pares de XP involucra una presión excesiva; la versión de Crystal Clear establece proximidad, pero cada quien se aboca a su trabajo asignado, prestando un ojo a lo que hace su compañero, quien tiene su propia máquina. Esta es una ampliación de la Comunicación Osmótica al contexto de la programación.

A pesar que no contempla el desarrollo de software propiamente dicho, CC involucra unos veinte productos de trabajo o artefactos. Mencionamos los más importantes:

- Declaración de la misión. Documento de máximo una página, describiendo el propósito.
- Estructura del equipo. Lista de equipos y miembros.
- Metodología. Comprende roles, estructura, proceso, productos de trabajo que mantienen, métodos de revisión.
- Secuencia de entrega. Declaración o diagrama de dependencia; muestra el orden de las entregas y lo que hay en cada una.
- Cronograma de visualización y entrega. Lista, planilla de hoja de cálculo o herramienta de gestión de proyectos.
- Lista de riesgos. Descripción de riesgos por orden descendente de prioridad.
- Estatus del proyecto. Lista hitos, fecha prevista, fecha efectiva y comentarios.
- Lista de actores-objetivos. Lista de dos columnas, planilla de hoja de cálculo, diagrama de caso de uso o similar.
- Casos de uso anotados. Requerimientos funcionales.
- Archivo de requerimientos. Colección de información indicando qué se debe construir, quiénes han de utilizarlo, cómo proporciona valor y qué restricciones afectan al diseño.

Los métodos Crystal no prescriben las prácticas de desarrollo, las herramientas o los productos que pueden usarse, pudiendo combinarse con otros métodos como Scrum, XP y Microsoft Solutions Framework.

3.3.2. ROLES Y RESPONSABILIDADES

Hay ocho roles nominados en CC: Patrocinador, Usuario Experto, Diseñador Principal, Diseñador-Programador, Experto en Negocios, Coordinador, Verificador, Escritor. En Crystal Naranja se agregan aun más roles: Diseñador de IU, Diseñador de Base de Datos, Experto en Uso, Facilitador Técnico, Analista/Diseñador de Negocios, Arquitecto, Mentor de Diseño, Punto de Reutilización. A continuación se describen según [12], [14] los artefactos de los que son responsables los roles de CC.

Patrocinador. Produce la Declaración de Misión con Prioridades de Compromiso. Consigue los recursos y define la totalidad del proyecto.

Usuario Experto. Junto con el Experto en Negocios produce la Lista de Actores-Objetivos y el Archivo de Casos de Uso y Requerimientos. Debe familiarizarse con el uso del sistema, sugerir atajos de teclado, modos de operación, información a visualizar simultáneamente, navegación, etcétera.

Diseñador Principal. Produce la Descripción Arquitectónica. Se supone que debe ser al menos un profesional de Nivel 3. (En metodologías ágiles se definen tres niveles de experiencia: Nivel 1 es capaz de "seguir los procedimientos"; Nivel 2 es capaz de "apartarse de los procedimientos específicos" y encontrar otros distintos; Nivel 3 es capaz de manejar con fluidez, mezclar e inventar procedimientos). El Diseñador Principal tiene roles de coordinador, arquitecto, mentor y programador más experto.

Diseñador-Programador. Produce, junto con el Diseñador Principal, los Borradores de Pantallas, el Modelo Común de Dominio, las Notas y Diagramas de Diseño, el Código Fuente, el Código de Migración, las Pruebas y el Sistema Empaquetado. Cockburn no distingue entre diseñadores y programadores. Un programa en CC es "diseño y programa"; sus programadores son diseñadores-programadores. En CC un diseñador que no programe no tiene cabida.

Experto en Negocios. Junto con el Usuario Experto produce la Lista de Actores-Objetivos y el Archivo de Casos de Uso y Requerimientos. Debe conocer las reglas y políticas del negocio.

Coordinador. Con la ayuda del equipo, produce el Mapa de Proyecto, el Plan de Entrega, el Estado del Proyecto, la Lista de Riesgos, el Plan y Estado de Iteración y la Agenda de Visualización.

Verificador. Produce el Reporte de errores. Puede ser un programador en tiempo parcial, o un equipo de varias personas.

Escritor. Produce el Manual de Usuario.

3.3.3 PRÁCTICAS

Todas las metodologías Crystal involucran una serie de prácticas, así como desarrollo incremental. En la descripción de Crystal naranja [12] el incremento incluye actividades como: estacionamiento, supervisión, revisión, junto con paralelismo y flujo, ver figura.

también otras actividades y prácticas pueden ser identificadas. Estas son incluidas en la siguiente descripción.

Estacionamiento: el estacionamiento incluye la planificación del próximo incremento del sistema. Este debería ser programado para producir una liberación trabajando cada tres o cuatro meses [13] como máximo. Un tiempo entre uno y tres meses es el preferido. El equipo selecciona los requerimientos a ser implementados en el incremento y programa cuales de ellos consideran deben entregar [13].

Revisión: cada incremento incluye varias iteraciones. Cada iteración incluye las siguientes actividades: construcción, demostración y revisión de los objetivos del incremento [13].

Monitoreo: el progreso es monitoreado teniendo en cuenta las entregas del equipo durante el proceso de desarrollo con respecto al progreso y estabilidad de éstas [13]. El progreso es medido por avances (inicio, revision1, revision2, pruebas, entrega) y fases de estabilidad (fluctuación intensa, fluctuación y suficientemente estable para revisar). Monitoreo es requerido en Crystal Clear y Crystal Orange.

Paralelismo y flujo: Una vez el monitoreo es estable se da el resultado “suficientemente estable revisar”, para las entregas la nueva tarea puede empezar. En Crystal orange, esto significa que los múltiples equipos pueden proceder con máximo paralelismo satisfactoriamente. Para alcanzar esto, el equipo de monitoreo y arquitectura revisan sus planes de trabajo, estabilidad y sincronización [13].

Diversidad de estrategia holística: Crystal orange incluye un método llamado diversidad de estrategia holística para dividir grandes grupos funcionales en grupos cruzados. La idea central de esto es incluir múltiples especialistas en un solo equipo [13] (Pág. 214). La diversidad de estrategia holística también permite formar pequeños grupos con el conocimiento específico necesario “know-how”, y también considera aspectos como la localización de los equipos, comunicación y documentación y coordinación de equipos múltiples [13].

Técnica de sintonización metodológica: la técnica de sintonización metodológica es una de las técnicas básicas de Crystal clear y crystal orange. Utiliza entrevistas del proyecto y talleres del equipo para resolver un método Crystal específico para cada proyecto individual [12]. Una de las ideas del desarrollo incremental es permitir la fijación o mejorar el proceso de desarrollo [13]. En cada incremento, puede utilizar el conocimiento aprendido para desarrollar el proceso para el incremento siguiente.

Puntos de vista del usuario: ambos puntos de vista del usuario son sugeridos por Crystal Clear para una liberación [12]. En Crystal naranja los puntos de vista del usuario deben ser organizados tres veces por cada incremento [13].

Taller de reflexión: tanto Crystal naranja como Crystal Clear incluyen una regla que los equipos deben mantener los talleres de reflexión previos y posteriores al incremento, (se recomiendan también los talleres de reflexión en la mitad del incremento) [12].

Crystal Clear y Crystal naranja no definen ninguna práctica o técnica a ser usada por los miembros del proyecto en sus tareas de desarrollo de software. La adopción de prácticas de otras metodologías como XP y SCRUM es permitida en Crystal para reemplazar algunas prácticas propias, como por ejemplo, los talleres de reflexión.

3.4. DISEÑO DIRIGIDO POR RASGOS (FEATURE DESIGN DRIVEN – FDD).

A diferencia de otras metodologías ágiles, no cubre todo el ciclo de vida sino sólo las fases de diseño y construcción y se considera adecuado para proyectos mayores y de misión crítica [16]. FDD contiene desarrollo iterativo con las mejores prácticas encontradas para ser efectivo en la industria. Hace énfasis en aspectos de calidad a lo largo de los procesos e incluye entregas tangibles y pequeñas, junto con monitoreo preciso del progreso del proyecto.

3.4.1. PROCESO

FDD consiste de cinco procesos secuenciales durante los cuales se diseña y construye el sistema. La parte iterativa soporta desarrollo ágil con rápidas adaptaciones a cambios en requerimientos y necesidades del negocio. Cada fase del proceso tiene un criterio de entrada, tareas, pruebas y un criterio de salida. Típicamente, la iteración de un rasgo invierte de una a tres semanas. Las fases se describen de acuerdo a [16]; estos son:

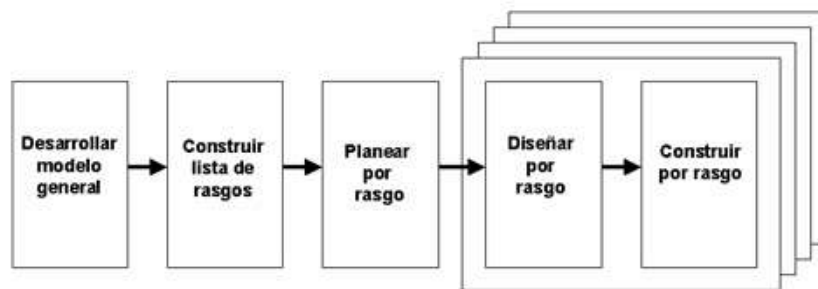


Figura 5: proceso FDD, basado en [16]

Desarrollo de un modelo general. Los expertos de dominio presentan un ensayo en el que los miembros del equipo y el arquitecto principal se informan de la descripción de alto nivel del sistema. El dominio general se subdivide en áreas más específicas y se define un ensayo más detallado para cada uno de los miembros del dominio. Luego de cada ensayo, un equipo de desarrollo trabaja en pequeños grupos para producir modelos de objeto de cada área de dominio. Simultáneamente, se construye un gran modelo general para todo el sistema [16].

Construcción de la lista de rasgos. Los ensayos, modelos de objeto y documentación de requerimientos proporcionan la base para construir una amplia lista de rasgos. Los rasgos son pequeños ítems útiles a los ojos del cliente. Son similares a las tarjetas de historias de XP y se escriben en un lenguaje que todas las partes puedan entender. Las funciones se agrupan conforme a diversas actividades en áreas de dominio específicas. La lista de rasgos es revisada por los usuarios y patrocinadores para asegurar su validez y exhaustividad. Los rasgos que requieran más de diez días se descomponen en otros más pequeños.

Planeamiento por rasgo. Incluye la creación de un plan de alto nivel, en el que los conjuntos de rasgos se ponen en secuencia conforme a su prioridad y dependencia, y se asigna a los programadores jefes. Las listas se priorizan en secciones que se llaman paquetes de diseño. Luego se asignan las clases definidas en la selección del modelo general a programadores individuales, o sea propietarios de clases.

Diseño por rasgo y Construcción por rasgo. Se selecciona un pequeño conjunto de rasgos del conjunto y los propietarios de clases seleccionan los correspondientes equipos dispuestos por rasgos. Se procede luego iterativamente hasta que se producen los rasgos seleccionados. Una iteración puede tomar de unos pocos días a un máximo de dos semanas. Puede haber varios grupos trabajando en paralelo. El proceso iterativo incluye inspección de diseño, codificación, prueba de unidad, integración e inspección de código. Luego de una iteración exitosa, los rasgos completos se promueven al repositorio principal. Este proceso puede demorar una o dos semanas en implementarse.

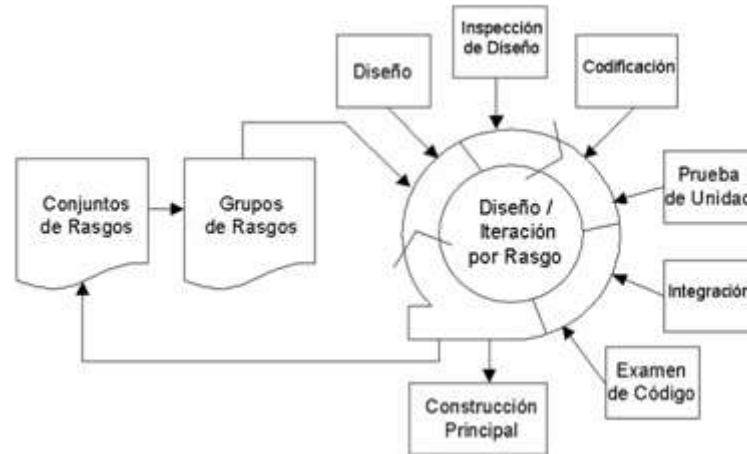


Figura 6: Ciclo de FDD, basado en [17]

3.4.2. ROLES Y RESPONSABILIDADES

Hay tres categorías de rol en FDD: roles claves, roles de soporte y roles adicionales. A continuación se presentan los roles y responsabilidades según [16]

Roles clave:

(1) administrador del proyecto: es el líder administrativo y financiero del proyecto. Una de sus tareas es proteger al equipo del proyecto de distracciones externas y capacitar al equipo para trabajar conjuntamente en condiciones apropiadas. Es quien tiene la última palabra en materia de visión, cronograma y asignación del personal.

(2) arquitecto jefe: El diseñador jefe es responsable por el diseño completo del sistema, es quien toma la decisión final en todos los aspectos de diseño. Si es necesario este rol puede ser dividido en arquitecto de dominio y arquitecto técnico.

(3) manager de desarrollo: es el encargado de las actividades diarias de desarrollo y de solucionar cualquier conflicto que pueda presentarse en el equipo. Además, este rol incluye la responsabilidad de solucionar problemas de recursos. Las responsabilidades de este rol pueden combinarse con el arquitecto jefe o manager de proyecto.

(4) programador jefe: es un desarrollador experimentado, que participa en el análisis de requisitos y el diseño de proyectos. El programador jefe es responsable de conducir a equipos pequeños en el análisis, diseño, y el desarrollo de nuevos rasgos. Selecciona rasgos

del conjunto de rasgos para ser desarrollados en la iteración siguiente e identifica la clase y los propietarios de clase que se necesitan en el equipo.

(5) propietarios de clases: trabajan bajo la guía del programador jefe en diseño, codificación, prueba y documentación, repartidos por rasgos. Es el responsables del desarrollo de la clase que le han asignado como propietario. Los propietarios de clase forman equipos de rasgos.

(6) experto de dominio, que puede ser un cliente, patrocinador, analista de negocios o una mezcla de todo eso. Su tarea es poseer el conocimiento de cómo los diversos requisitos para el sistema en desarrollo deben realizarse. Los expertos de dominio pasan este conocimiento a los desarrolladores para asegurarse que los desarrolladores entregan un sistema competente.

Roles de soporte:

(1) administrador de entrega: que controla el progreso del proceso revisando los reportes del programador jefe y manteniendo reuniones breves con él; además reporta el progreso al manager del proyecto.

(2) abogado/guru de lenguaje: Es un miembro del equipo que debe tener todo el conocimiento necesario relacionado con un lenguaje o una tecnología específica. Este rol es particularmente importante cuando el equipo de desarrollo se esta ocupando de una nueva tecnología.

(3) ingeniero de construcción: Es el responsable de la creación, funcionamiento y mantenimiento de la estructura del proceso, incluye la tarea de manejar el sistema de control y de publicar la documentación.

(4) herramentista (toolsmith): Es un rol para construcción de pequeñas herramientas para desarrollo, pruebas y conversión de datos en el equipo del proyecto. también puede trabajar manteniendo bases de datos y sitios Web para proyectos de propósito específico.

(5) administrador del sistema: Debe configurar, manejar y localizar averías en los servidores, sitios de red y de los ambientes de desarrollo y pruebas usados por el equipo del proyecto.

Roles adicionales:

(1)Verificadores: Verifican que el sistema que es producido satisface los requisitos del cliente. Puede ser un equipo independiente o una parte del equipo del proyecto.

(2)Deployer: El trabajo del *deployer* esta relacionado con la conversión de datos existentes al formato requerido por el nuevo sistema y participan en el despliegue de nuevos lanzamientos. Pueden ser un equipo independiente o parte del equipo del proyecto.

(3)Escritor técnico: La documentación del usuario es preparada por el escritor técnico, que también puede ser un equipo independiente o parte del equipo del proyecto.

3.4.3 PRÁCTICAS

FDD consiste en un conjunto de “mejores prácticas” y los desarrolladores argumentan que aunque las prácticas seleccionadas no son nuevas, la mezcla específica de estos ingredientes hace los cinco procesos de FDD únicos para cada caso.

- Modelado de objetos del dominio, resultante en un *framework* cuando se agregan los rasgos. Esta forma de modelado descompone un problema mayor en otros menores; el diseño y la implementación de cada clase u objeto es un problema pequeño a resolver. Cuando se combinan las clases completas, constituyen la solución al problema mayor.
- Desarrollo por rasgo. Hacer simplemente que las clases y objetos funcionen no refleja lo que el cliente pide. El seguimiento del progreso se realiza mediante examen de pequeñas funcionalidades descompuestas y funciones valoradas por el cliente. Un rasgo en FDD es una función pequeña expresada en la forma <acción> <resultado> <por | para | de | a> <objeto> con los operadores adecuados entre los términos. Por ejemplo, calcular el importe total de una venta; determinar la última operación de un cajero; validar la contraseña de un usuario.
- Propiedad individual de clases (código). Cada clase tiene una sola persona nominada como responsable por su consistencia, rendimiento e integridad conceptual.
- Equipos de Rasgos, pequeños y dinámicamente formados. La existencia de un equipo garantiza que un conjunto de mentes se apliquen a cada decisión y se tomen en cuenta múltiples alternativas.
- Administración de configuración. Permite realizar seguimiento histórico de las últimas versiones completas de código fuente.
- Reporte de progreso. Se comunica a todos los niveles organizacionales necesarios.

FDD suministra un rico conjunto de artefactos para la planificación y control de los proyectos. En [18] se encuentran diversos formularios y tablas con información real de implementaciones de FDD: Vistas de desarrollo, Vistas de planificación, Reportes de progreso, Reportes de tendencia, Vista de plan (<acción><resultado><objeto>), etcétera. Se han desarrollado también algunas herramientas que generan vistas combinadas o específicas.

En síntesis, FDD es un método de desarrollo de ciclos cortos que se concentra en la fase de diseño y construcción. En la primera fase, el modelo global de dominio es elaborado por expertos de dominio y desarrolladores; el modelo de dominio consiste en diagramas de clases con clases, relaciones, métodos y atributos. Los métodos no reflejan conveniencias de programación sino rasgos funcionales.

Algunos agilistas sienten que FDD es demasiado jerárquico para ser un método ágil, porque demanda un programador jefe, quien dirige a los propietarios de clases, quienes dirigen equipos de rasgos. Otros críticos sienten que la ausencia de procedimientos detallados de prueba en FDD es llamativa e impropia. Los promotores del método aducen que las empresas ya tienen implementadas sus herramientas de prueba, pero subsiste el problema de su adecuación a FDD. Un rasgo llamativo de FDD es que no exige la presencia del cliente.

FDD se utilizó por primera vez en grandes aplicaciones bancarias a fines de la década de 1990. Los autores sugieren su uso para proyectos nuevos o actualizaciones de sistemas existentes, y recomiendan adoptarlo en forma gradual. En [17] se asegura que aunque no hay evidencia amplia que documente sus éxitos, las grandes consultoras suelen recomendarlo incluso para delicados proyectos de misión crítica.

4. MODELAMIENTO AGIL (AGILE MODELING – AM)

4.1. DEFINICION

Modelamiento ágil es un conjunto de prácticas para modelamiento y documentación efectiva de sistemas basados en software. AM es una colección de prácticas, principios y valores para modelamiento de software que pueden ser aplicadas en un proyecto de desarrollo de software de una manera ligera y efectiva. Se puede tomar modelamiento ágil para requerimiento, análisis, arquitectura y diseño.

AM no define un procedimiento detallado para como crear un tipo de modelo dado, en cambio provee consejos para ser efectivo como modelador. AM no reduce el modelamiento, de hecho muchos desarrolladores encontrarán que ellos están haciendo más modelamiento del que hacían en el pasado antes de usar AM. AM reconoce que los valores del desarrollo ágil, promovidos por la alianza ágil, son la llave para ser efectivo. Los valores son:

- Individuos e interacciones sobre procesos y herramientas.
- Trabajo de software sobre comprensiva documentación.
- Colaboración con el cliente sobre negociación de contrato.
- Responder al cambio sobre seguir un plan.

4.2. PRINCIPIOS DE AM

AM define una serie de de principios básicos y suplementarios que al aplicarse en un proyecto de desarrollo de software fijan la fase para una colección de prácticas de modelamiento. Algunas de las prácticas han sido adoptadas de programación extrema y ya han sido bien documentadas.

Los principios de AM son clasificados como principios básicos que se deben adoptar para poder proclamar que verdaderamente esta tomando un enfoque AMDD agile model driven development,- desarrollo dirigido por modelamiento ágil y principios suplementarios que se deben considerar adaptar en un proceso software al conocer las necesidades exactas del ambiente. En enero de 2005 se adiciono una tercera lista, principios desaprobados, que el creador de AM ha decidido eliminar en la segunda versión de AMDD para simplificarla.

4.2.1. Principios básicos.

Modelar con un propósito. El primer paso es identificar un propósito valido para crear un modelo y las audiencias para ese modelo. Una vez que un modelo haya satisfecho sus metas ha finalizado con este por ahora y debería seguir con algo mas, como escribir código o mostrar que el modelo trabaja. Este principio también se aplica a un cambio a un modelo existente: si usted está realizando un cambio, tal vez utilizando un patrón conocido, después tendría una razón válida para hacer que cambie (talvez soportar un nuevo requerimiento o refactorizar su trabajo para obtener algo mas limpio). Una importante implicación de este principio es que es necesario conocer su audiencia, incluso cuando la audiencia es usted mismo.

Maximice inversión del stakeholder. Los stakeholders del proyecto están invirtiendo recursos –tiempo, dinero, instalaciones, etc.- para desarrollar el software que resuelva sus necesidades. Los stakeholders merecen invertir sus recursos de la mejor manera posible y

no tener recursos fragmentados lejos de su equipo. Además, ameritan tener la opinión final de cómo se invierten o no se invierten esos recursos.

Luz del recorrido. Cada artefacto que usted crea, y que luego decide conservar, necesitará mantenimiento a lo largo del tiempo. Si decide conservar siete modelos, entonces siempre que ocurra un cambio es necesario considerar el impacto del cambio sobre todos los siete modelos y actuar consecuentemente. Si se decide conservar solamente tres modelos entonces claramente se tiene menos trabajo para soportar los mismos cambios. Similarmente, cuando los modelos son más complejos y detallados, es más probable que cualquier cambio dado resulte más duro de lograr (el modelo individual es mas pesado y es por lo tanto mas de una carga a mantener). Cada vez que se decide mantener un modelo usted intercambia menos agilidad por la conveniencia de tener la información disponible para su equipo de una manera abstracta. Un equipo de desarrollo que decide mantener un documento de requisitos detallados, una colección detallada de modelos de análisis, una colección detallada de modelos arquitectónicos, y una detallada colección de modelos de diseño rápidamente descubrirá que están gastando la mayor parte del tiempo en actualizar dichos documentos y modelos en lugar de escribir el código.

Múltiples modelos. Usted necesitará potencialmente utilizar múltiples modelos para desarrollar software porque cada modelo describe un aspecto único de su software. Un importante punto es que no será necesario desarrollar todos estos modelos para un sistema dado, pero dependiendo de la naturaleza del software a desarrollar será necesario al menos un subconjunto de modelos.

Realimentación rápida. El tiempo entre una acción y la realimentación a esa acción es crítico. En el trabajo de cerca con el cliente, para entender los requisitos, para analizar esos requisitos o para desarrollar una interfaz de usuario que resuelva sus necesidades, es necesaria la realimentación rápida.

Asuma simplicidad. Mientras usted desarrolla debe asumir que la solución más simple es la mejor solución. No sobredimensionar su software, o en caso de que el AM no describa las características adicionales en sus modelos que usted no necesite hoy. Tenga el valor para no remodelar su sistema hoy, ya que se puede modelar basado en sus requerimientos existentes hoy y refactorizar el sistema cuando los requerimientos evolucionen. Mantenga los modelos tan simples como sea posible.

Aceptar el cambio. Los requerimientos evolucionan sobre el tiempo. La gente entiende que los requerimientos cambian en el tiempo. Los inversionistas del proyecto pueden cambiar características del proyecto a medida que este avanza, nuevo personal puede ser adicionado al proyecto o algunos existentes lo pueden dejar. Los inversionistas del proyecto pueden cambiar los puntos de vista, potencialmente cambiando los objetivos y los criterios de éxito para su esfuerzo. La implicación es que el ambiente del proyecto cambia mientras progresan sus esfuerzos, y consecuentemente su enfoque de desarrollo debe reflejar esta realidad.

Cambio incremental. Un importante concepto a entender con respecto a modelar es que no es necesario conseguir el modelado correcto la primera vez, de hecho, es muy inusual que se pueda lograr. Además, no se necesita capturar cada detalle en sus modelos, lo que justamente se necesita es conseguir un buen modelo a tiempo. En vez de intentar en vano desarrollar todo el modelo desde el comienzo usted puede jugársela por desarrollar un pequeño modelo, o talvez un modelo de alto nivel, y evolucionar este de una manera incremental.

Trabajo de calidad. A nadie le gusta el trabajo mediocre. La gente que hace el trabajo no gusta de él porque es algo de lo que no puede estar orgullosa, el personal que más tarde realiza la refactorización no gusta de él porque será más duro de entender y actualizar, y el usuario final no gusta de él porque probablemente el trabajo es frágil o no cumple sus expectativas.

El software es su objetivo fundamental. El objetivo del desarrollo de software es producir software que satisfaga las necesidades de los inversionistas del proyecto de una manera efectiva. El objetivo primordial es no producir documentación extraña, artefactos extraños a la gerencia, o aun modelos extraños. Cualquier actividad que no contribuya a esta meta debe ser cuestionada y evitada si no puede ser justificada.

Permitir el próximo esfuerzo en los objetivos secundarios. Parte de satisfacer las necesidades de los inversionistas/clientes es asegurar que el sistema es bastante robusto de modo que pueda ser extendido en un momento dado. Según Cockburn, cuando se esta planeando el desarrollo de software el objetivo secundario es configurar el juego de la planificación de la siguiente. El esfuerzo siguiente puede ser el desarrollo del próximo lanzamiento importante del sistema o puede ser simplemente las operaciones y soporte de la versión actual que se está construyendo. Para lograrlo no se deseará solamente desarrollar software de calidad sino también crear justo la documentación suficiente de modo que el personal de la siguiente fase pueda ser eficaz. Los factores que se necesita considerar incluyen si los miembros del equipo actual van a ser parte del siguiente esfuerzo, la naturaleza en si misma del siguiente esfuerzo, y la importancia del siguiente esfuerzo para la organización. En pocas palabras, cuando estemos trabajando en nuestro proyecto necesitamos tener un ojo puesto en el futuro.

4.2.2. Principios Suplementarios

Contenido es más importante que la documentación. Cualquier modelo dado puede tener muchas formas de representarse. Por ejemplo, una especificación de historias de usuario usando un post de notas a lo largo de una hoja grande de papel (un modelo esencial o prototipo de baja fidelidad), mientras un prototipo tradicional lo construye utilizando un lenguaje de programación, o como un documento formal incluyendo ambos una representación visual como una descripción textual de la historia de usuario. Una interesante implicación es que el modelo no necesita ser un documento. Incluso un completo conjunto de diagramas creados usando CASE puede no hacer parte de un documento, en cambio estas se utilizan como entradas en otros artefactos, muy comúnmente código fuente, pero nunca formalizada como documento inicial. El punto es que se debe aprovechar las ventajas de los beneficios del modelado sin incurrir en los costos de mantener una documentación.

Comunicación abierta y honesta. La gente necesita ser libre, y percibir que lo es para ofrecer sugerencias. Esto incluye ideas pertinentes a uno o más modelos, entrega de malas noticias, como estar retrasado en el desarrollo del cronograma. La comunicación abierta y honesta capacita al personal para tomar mejores decisiones porque la calidad de la información en la que se están basando es más exacta.

4.3. PRACTICAS DE AM

4.3.1. PRÁCTICAS BÁSICAS

Participación activa del cliente. Una extensión de la práctica XP del cliente en el sitio de trabajo que describe la necesidad de tener acceso en el sitio a usuarios que tienen la autoridad y habilidad de proveer información pertinente al sistema que está siendo construido y tomar decisiones a tiempo teniendo en cuenta los requisitos y la prioridad de estos. La extensión de la práctica xp del cliente en el sitio de trabajo incluye usuarios directos, la gerencia de estos, personal de operaciones, y personal de soporte (activamente vinculados al proyecto). Esto incluye tomar decisiones oportunas para la gerencia, soporte público y privado para el proyecto por el gerente, y participación activa del personal de operaciones y soporte en el desarrollo de requisitos y modelos pertenecientes a sus respectivas áreas.

Modele con otros. Cuando se modela con un propósito se encuentra frecuentemente que está modelando para entender algo, que está modelando para comunicar sus ideas con otros, o está buscando desarrollar una visión común de un proyecto. Esta es una actividad del grupo, una en la cual se desea la entrada de varias personas que trabajan juntas con eficacia. Frecuentemente se descubre que el equipo necesita trabajar conjuntamente para crear el conjunto de modelos básicos críticos al proyecto. Por ejemplo, desarrollar la metáfora o arquitectura del sistema, a menudo será necesario un grupo para desarrollar una solución, tan simple como sea posible. Para conseguir esto la mayoría de las veces el mejor camino es hablar de esto con una o más personas.

Aplique el artefacto correcto. Cada artefacto tiene sus propias aplicaciones específicas. Muchas veces un diagrama es mejor que código fuente –Si una imagen es mejor que mil palabras entonces frecuentemente un modelo es mejor que mil líneas de código cuando se aplica en las circunstancias correctas porque es posible explorar más alternativas de diseño eficientemente dibujando una pareja de diagramas. La implicación es que es necesario conocer las fortalezas y debilidades de cada tipo de artefacto, de modo que sabemos cuándo utilizarlos y cuándo no utilizarlos.

Iterar a otro artefacto. Cuando se trabaja en el desarrollo de un artefacto, como casos de uso, tarjetas de CRC, diagramas de secuencia, o incluso código fuente, y encuentra que está en aprietos, entonces debe considerar mientras tanto trabajar en otro artefacto. Cada artefacto es bueno para cierto tipo de trabajo. Cuando encontramos dificultades trabajando en un artefacto, tal vez estamos trabajando en un caso de uso y encuentra que esta tratando de describir la lógica del negocio, esto es un signo de que se debe iterar a otro artefacto. Por ejemplo, si estamos trabajando en un caso de uso esencial entonces puede considerar cambiar el enfoque para comenzar a trabajar en un prototipo esencial de interfaz de usuario, un modelo CRC, una regla de negocio, un sistema de casos de uso, o un cambio de caso. Iterando a otro artefacto es posible inmediatamente salir del aprieto porque esta trabajando progresivamente en otro artefacto.

Pruébalo con código. Un modelo es una abstracción que debe reflejar exactamente un aspecto de lo que está construyendo. Para saber si funciona debe demostrarlo con código.

Utilice las herramientas más simples. La gran mayoría de modelos puede dibujarse simplemente con lápiz y papel o marcadores y tablero. Si se desea ahorrar alguno de estos dibujos es posible tomar una fotografía del diagrama con una cámara digital. La mayoría de

estos diagramas una vez resuelta la aplicación no ofrecen mucho valor como resultado. Un tablero y marcadores, o simplemente papel y lápiz pueden convertirse en su mejor herramienta de modelado.

Modele en incrementos pequeños. Desarrollo incremental en que se organiza una gran labor en pequeñas porciones que se lanzan en un cierto plazo. Preferiblemente en incrementos de semanas o un mes o dos, lo que incrementa la agilidad permitiendo entregar software más rápido.

Fuente de información simple. La información se debe guardar en un lugar solamente. En otras palabras, no solamente se debe aplicar el artefacto correcto, también se debe modelar un concepto sólo una vez, almacenando la información en el mejor lugar posible. Cuando alguien está modelando debe hacerse la pregunta si necesita conservar esa información permanentemente, de ser así, cual es el mejor lugar para almacenar dicha información y si esta información ya ha sido capturada y debe referenciarla simplemente. A veces el mejor lugar para almacenar la información es un documento ágil, frecuentemente el código fuente.

Propiedad colectiva. Cada uno puede trabajar en un modelo, y de hecho cualquier artefacto del proyecto si es necesario.

Crear varios modelos en paralelo. Cada tipo de modelo tiene sus fortalezas y debilidades, un modelo simple no es suficiente para las necesidades de modelamiento. Por ejemplo, cuando se está explorando requerimientos puede ser necesario desarrollar casos de uso esenciales o historias de usuario, un prototipo esencial de interfaz de usuario, y algunas reglas de negocio. En combinación con la práctica de iteración a otro artefacto los modeladores frecuentemente descubrirán que pueden ser más productivos trabajando en varios modelos simultáneamente que si ellos solo se enfocan en uno un cierto tiempo.

Crear contenido simple. Se debe mantener el contenido actual en los modelos – requerimientos, análisis, arquitectura, o diseño– tan simple como sea posible mientras se satisfagan los requisitos de los clientes del proyecto. Lo anterior implica que no se debe agregar aspectos adicionales a los modelos a menos que sean justificables. Tenga el valor de confiar que se está en capacidad de adicionar una nueva característica si es solicitada y cuando sea solicitada.

Represente los modelos en su forma más simple. Cuando se considera los diagramas potenciales que se pueden utilizar (diagramas de UML, diagramas de interfaz de usuario, modelo de datos y otros), nos damos cuenta que la mayoría de las veces solo se requiere un subconjunto de la notación de los diagramas disponibles. Un simple modelo que muestre las características dominantes que se están intentando comprender, talvez un modelo de clases que represente las responsabilidades primarias de clases y las relaciones entre ellas, frecuentemente demuestra ser suficiente.

Exhibición de modelos públicamente. Los modelos deberían mostrarse públicamente, frecuentemente sobre algo denominado pared de modelado o pared maravilla. Esta permite una comunicación abierta y honesta en el equipo de desarrollo porque todos los modelos son rápidamente accesibles al grupo, igualmente permite que la información esté disponible para los clientes. La pared de modelado es donde se colocan los modelos para que cada uno los vea, esta debe ser accesible al equipo de desarrollo y a los clientes. La pared de modelado puede ser un lugar físico, un tablero donde se fijan los modelos, o un lugar virtual, como una pagina Web en Internet que es actualizada con imágenes escaneadas.

4.3.2. PRÁCTICAS SUPLEMENTARIAS

Aplicar estándares de modelado. Esta práctica se deriva de la práctica estándares de codificación de XP, la idea básica es que los desarrolladores deben permitir y seguir un conjunto común de estándares de modelado en un proyecto de software. Así como hay valor al seguir unas convenciones comunes de codificación, código limpio que sigue los lineamientos de codificación escogidos es más fácil de entender y de desarrollar, hay un valor similar en seguir convenciones comunes de modelado. Existe una amplia variedad de estándares de modelado disponibles.

Aplique patrones suavemente. Los modeladores efectivos aprenden y después aplican apropiadamente patrones arquitectónicos, de análisis y de diseño comunes en sus modelos.

Descarte modelos temporales. La gran mayoría de modelos que se crean son modelos temporales, diseño de bosquejos, prototipos de baja fidelidad, potenciales alternativas de arquitectura/diseño, etcétera. Modelos que han satisfecho un propósito pero ya no adicionan valor ahora que ya lo han hecho. Los modelos están rápidamente fuera de sincronización con el código, por lo que se debe tomar la decisión de sincronizar los modelos o desecharlos simplemente porque la inversión para poner al día los modelos no será recuperada. Esta práctica es particularmente importante para la documentación ágil.

Formalice modelo de contrato. El modelo de contrato se requiere a menudo cuando un grupo externo controla un recurso de la información requerido por el sistema, por ejemplo una base de datos. El modelo de contrato es algo que ambas partes deben convenir mutuamente y cambiar mutuamente en un cierto plazo. Los ejemplos del modelo de contrato incluyen documentación detallada del uso de una interfaz de programación (API). Como un contrato legal, un modelo del contrato requiere a menudo invertir recursos significativos para desarrollar y mantener el contrato y para asegurarse que es exacto y suficientemente detallado. Su meta es reducir al mínimo el número de modelos del contrato. Obsérvese que casi siempre se utilizará una herramienta electrónica para desarrollar un modelo del contrato porque el modelo se debe mantener en un cierto plazo.

Actualice solamente cuando sea necesario. Los modelos deben ser actualizados solamente cuando sea absolutamente necesario, cuando no actualizar el modelo sea más doloroso que actualizarlo. Con este enfoque se descubre que se actualizan un número menor de modelos de los que se tenía en el pasado y que la realidad es que los modelos no tienen que ser perfectos para proporcionar valor. Demasiado tiempo y dinero es desperdiciado al intentar mantener sincronizado modelos y documentación con código fuente, es una tarea imposible de llevar, tiempo y dinero que debería ser mejor invertido desarrollando nuevo software. Esta práctica es particularmente importante para una documentación ágil.

REFERENCIAS

- [1] Weitzenfeld, Alfredo. El Proceso para el Desarrollo de Software. <http://cannes.rhon.itam.mx/Alfredo/Espaniol/Publicaciones/MINT/Procsww.pdf>
- [2] Joe Nandhakumar y David Avison. "The fiction of methodological development: a field study of information systems development". *Information Technology & People*, 12(2), pp. 176-191, 1999.
- [3] Duane Truex, Richard Baskerville y Julie Travis. "Amethodical Systems Development: The deferred meaning of systems development methods". *Accounting, Management and Information Technology*, 10, pp. 53-79, 2000.
- [4] Agile Software Development Processes A Different Approach to Software Design - Everette R. Keith - año 2002 <http://www.agilealliance.org/articles/articles/ADifferentApproach.pdf> pagina 1
- [5] Manifiesto for agile software development. <http://www.agilemanifesto.org/>
- [6] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland y Dave Thomas. "Principles behind the Agile Manifiesto". <http://agilemanifesto.org/principles.html>, 2001
- [7] Beck, K. "Extreme programming explained. Embrace Change". Reading. Addison Wesley Longman, 2000
- [8] Programación Extrema - Experimentar para crear. <http://www.publicaciones.spc.org.pe/SPCMagazine/editions/II-1/html/articulos/articulo1.htm>
- [9] Extreme programming (XP): Un nuevo método de desarrollo. <http://www.ati.es/novatica/2002/156/156-8.pdf>
- [10] Schwaber, K. y Beedle, M. *Agile software with Scrum*. Upper Saddle River, NJ, Prentice-Hall. 2002.
- [11] WWW.Crystalmethodologies.org
- [12] Cockburn 2002a= [48] *Agile software development*. Boston, Addison-Wesley. 2002
- [13] (Cockburn 1998=[50] *Surviving Object-Oriented Projects: A Manager's guide*. Addison Wesley.
- [14] Alistair Cockburn. "Crystal Clear. A human-powered methodology for small teams, including The Seven Properties of Effective Software Projects". Borrador. *Humans and Technology*, versión del 27 de febrero de 2002.
- [15] <http://c2.com/cgi/wiki?ExtremeHour>

[16] Palmer, S.R. y Felsing, J.M. A practical guide to feature-driven developmen. Upper Saddle River, NJ, Prentice-Hall

[17] Abrahamsson, Outi Salo, Jussi Ronkainen y Juhani Warsta. "Agile Software Development Methods". VTT Publications 478, Universidad de Oulu, Suecia, 2002.

[18] <http://www.nebulon.com/articles/fdd/fddimplementations.html>

ANEXO 2: HISTORIAS DE USUARIO Y PRUEBAS

1. HISTORIAS DE USUARIO

En el primer capítulo se presenta la planificación de historias de usuario realizado durante el transcurso del caso de estudio, tras estudiar el proyecto y mantener conversaciones con el cliente. De esta redacción inicial de historias de usuario se realizó una planificación inicial y posteriormente fue cambiada a lo largo del proyecto. El presente documento presenta la versión final de las historias de usuario desarrolladas.

El segundo capítulo corresponde a las pruebas realizadas en cada una de las historias de usuario, con sus respectivos resultados o mensajes de error obtenidos para los distintos casos considerados.

Historia de Usuario	
Número: 1	Prioridad en Negocio: Alto
Nombre historia: Crear/eliminar microproyectos estándar	
Iteración Asignada: Primera	Riesgo en Desarrollo: Alto
Programador responsable: Paola Andrea Manquillo, Jesús Rivera	
Descripción: El usuario ingresará los datos de los distintos microproyectos estándar con que cuenta la granja. Las tareas asociadas a esta historia son: -Diseño de interfaz de usuario principal. -Diseño de interfaz Menú Proyectos, la cual le da la opción al usuario si va a ingresar proyectos estándar o especiales. -Diseño de interfaz Proyectos Estándar -Inserción de los datos del proyecto en la base de Datos.	
Observaciones:	

Historia de Usuario	
Número: 2	Prioridad en Negocio: Alto
Nombre historia: Crear/eliminar microproyectos especiales	
Iteración Asignada: Primera	Riesgo en Desarrollo: Alto
Programador responsable: Paola Andrea Manquillo, Jesús Rivera	
Descripción: El usuario ingresará los datos de los distintos microproyectos especiales con que cuenta la granja. Las tareas asociadas a esta historia son: -Diseño de interfaz Proyectos Estándar -Inserción de los datos del proyecto en la base de Datos.	
Observaciones:	

Historia de Usuario	
Número: 3	Prioridad en Negocio: Media
Nombre historia: Crear/eliminar productos	
Iteración Asignada: Primera	Riesgo en Desarrollo: Medio
Programador responsable: Paola Andrea Manquillo, Jesús Rivera	
Descripción: El usuario ingresará cada uno de los productos extraídos en cada uno de los microproyectos. Las tareas para llevar a cabo esta historia de usuario fueron: -Diseño de menú productos -Diseño de interfaz para almacenamiento de productos -Inserción de datos	
Observaciones:	

Historia de Usuario	
Número: 4	Prioridad en Negocio: media
Nombre historia: Crear/eliminar unidades	
Iteración Asignada: Primera	Riesgo en Desarrollo: Bajo
Programador responsable: Paola Andrea Manquillo, Jesús Rivera	
Descripción: El usuario ingresará las unidades de medida en que se comercializan los productos. Tareas -Diseño de interfaz unidades -Inserción de datos	
Observaciones:	

Historia de Usuario	
Número: 5	Prioridad en Negocio: Alta
Nombre historia: Introducir costos individuales:	
Iteración Asignada: Primera	Riesgo en Desarrollo: Medio
Programador responsable: Paola Andrea Manquillo, Jesús Rivera	
Descripción: Una vez el usuario ha creado proyectos, productos y unidades podrá calcular los costos individuales de cada uno de los productos de un determinado microproyecto. Las tareas fueron: -Diseño de menú Costos -Desarrollo de interfaz para ingreso de costos individuales -Inserción y almacenamiento de costos individuales en la base de datos	
Observaciones:	

Historia de Usuario	
Número: 6	Prioridad en Negocio: Alta
Nombre historia: Calcular costos totales	
Iteración Asignada: Primera	Riesgo en Desarrollo: Medio
Programador responsable: Paola Andrea Manquillo, Jesús Rivera	
Descripción: Una vez se han introducido costos individuales se pueden calcular los costos totales para un producto perteneciente a un microproyecto determinado. Tareas que se llevaron a cabo: - Desarrollo de interfaz para cálculo de costos totales - Lectura base de datos - Obtención de costos totales y almacenamiento en base de datos	
Observaciones:	

Historia de Usuario	
Número: 7	Prioridad en Negocio: Alta
Nombre historia: Introducir ingresos individuales	
Iteración Asignada: Segunda	Riesgo en Desarrollo: Medio
Programador responsable: Paola Andrea Manquillo	
Descripción: Una vez el usuario ha creado proyectos, productos y unidades podrá introducir los ingresos individuales de cada uno de los productos de un determinado microproyecto. Las tareas fueron: -Diseño de menú Ingresos -Desarrollo de interfaz para entrada de ingresos individuales -Inserción y almacenamiento de ingresos individuales en la base de datos	
Observaciones:	

Historia de Usuario	
Número: 8	Prioridad en Negocio: Alta
Nombre historia: Calcular ingresos totales	
Iteración Asignada: Segunda	Riesgo en Desarrollo: Media
Programador responsable: Paola Andrea Manquillo	
Descripción: Una vez se han introducido ingresos individuales se pueden calcular los ingresos totales para un producto perteneciente a un microproyecto determinado. Tareas que se llevaron a cabo: <ul style="list-style-type: none">- Desarrollo de interfaz para cálculo de ingresos totales- Lectura base de datos- Obtención de ingresos totales y almacenamiento en base de datos	
Observaciones:	

Historia de Usuario	
Número: 9	Prioridad en Negocio: Alta
Nombre historia: Calcular costos por fecha	
Iteración Asignada: Segunda	Riesgo en Desarrollo: Alto
Programador responsable: Paola Andrea Manquillo	
Descripción: El administrador agropecuario podrá calcular los costos de un determinado producto de un microproyecto, para un período de tiempo comprendido entre una fecha inicial y una fecha final escogidos por él mismo. Las tareas fueron: <ul style="list-style-type: none">- Desarrollo de interfaz para cálculo de costos por fecha- Lectura base de datos- Obtención de costos por fecha y almacenamiento en base de datos	
Observaciones:	

Historia de Usuario	
Número: 10	Prioridad en Negocio: Alta
Nombre historia: Calcular ingresos por fecha	
Iteración Asignada: Segunda	Riesgo en Desarrollo: Medio
Programador responsable: Paola Andrea Manquillo	
Descripción: El administrador agropecuario podrá calcular los ingresos de un determinado producto de un microproyecto, para un período de tiempo comprendido entre una fecha inicial y una fecha final escogidas por él mismo. Las tareas fueron: - Desarrollo de interfaz para cálculo de ingresos por fecha - Lectura base de datos - Obtención de ingresos por fecha y almacenamiento en base de datos	
Observaciones:	

Historia de Usuario	
Número: 11	Prioridad en Negocio:
Nombre historia: Calcular utilidades totales	
Iteración Asignada: Segunda	Riesgo en Desarrollo: Medio
Programador responsable: Paola Andrea Manquillo	
Descripción: Una vez calculados los costos e ingresos totales es posible calcular las utilidades totales para un producto perteneciente a un determinado microproyecto. Las tareas necesarias fueron: - Desarrollo de interfaz para cálculo de utilidades - Desarrollo de interfaz para cálculo de utilidades totales - Lectura a base de datos - Obtención de utilidades totales y almacenamiento en base de datos	
Observaciones:	

Historia de Usuario	
Número: 12	Prioridad en Negocio: Alta
Nombre historia: Calcular utilidades por fecha	
Iteración Asignada: Segunda	Riesgo en Desarrollo: Medio
Programador responsable: Paola Andrea Manquillo	
Descripción: El administrador agropecuario podrá calcular las utilidades de un determinado producto de un microproyecto, para un período de tiempo comprendido entre una ficha inicial y una fecha final escogidos por Él mismo. Las tareas fueron: - Desarrollo de interfaz para cálculo de utilidades por fecha - Lectura base de datos - Obtención de utilidades por fecha y almacenamiento en base de datos	
Observaciones:	

Historia de Usuario	
Número: 13	Prioridad en Negocio: Baja
Nombre historia: Introducción de fichas técnicas agrícolas	
Iteración Asignada: Tercera	Riesgo en Desarrollo: Medio
Programador responsable: Paola Andrea Manquillo	
Descripción: El usuario podrá introducir y almacenar los datos correspondientes a las fichas técnicas agrícolas. Las tareas para esta historia fueron: -Diseño de menú fichas técnicas -Diseño de interfaz de usuario para ingreso de datos fichas agrícolas -Inserción de datos en la interfaz y almacenamiento en la base de datos	
Observaciones:	

Historia de Usuario	
Número: 14	Prioridad en Negocio: alta
Nombre historia: Ingreso y almacenamiento de datos fichas técnicas porcinos, etapa lactancia	
Iteración Asignada: Tercera	Riesgo en Desarrollo: Alta
Programador responsable: Paola Andrea Manquillo	
Descripción: El usuario podrá ingresar y almacenar los datos correspondientes a la ficha técnica del proyecto porcinos en su etapa de lactancia. Tareas: -Diseño de menú fichas técnicas pecuarias -Diseño de menú fichas técnicas porcinos -Diseño de interfaz de usuario para ingreso de datos ficha técnica porcinos etapa lactancia -Inserción de datos y almacenamiento de los mismos	
Observaciones:	

Historia de Usuario	
Número: 15	Prioridad en Negocio: Alta
Nombre historia: Ingreso y almacenamiento de datos fichas técnicas porcinos, otras etapas	
Iteración Asignada: Tercera	Riesgo en Desarrollo: Alto
Programador responsable: Paola Andrea Manquillo	
Descripción: El usuario podrá ingresar y almacenar los datos correspondientes a la ficha técnica del proyecto porcinos en otras etapas (distintas de la etapa de lactancia). Tareas: -Diseño de interfaz de usuario para el ingreso de datos de ficha técnica porcinos en otras etapas -Inserción de datos y almacenamiento de los mismos	
Observaciones:	

Historia de Usuario	
Número: 16	Prioridad en Negocio: Alta
Nombre historia: Introducción y almacenamiento de datos fichas técnicas vacas lecheras	
Iteración Asignada: Tercera	Riesgo en Desarrollo: Medio
Programador responsable: Paola Andrea Manquillo	
Descripción: El usuario podrá ingresar y almacenar los datos correspondientes a la ficha técnica del proyecto vacas lecheras. Tareas -Diseño de menú bovinos -Diseño de interfaz de usuario para ingreso de datos de la ficha técnica vacas lecheras -Inserción de datos y almacenamiento de los mismos en la base de datos	
Observaciones:	

Historia de Usuario	
Número: 17	Prioridad en Negocio: Alta
Nombre historia: Introducción y almacenamiento de datos fichas técnicas vacas lecheras, registro diario	
Iteración Asignada: Tercera	Riesgo en Desarrollo: Medio
Programador responsable: Paola Andrea Manquillo	
Descripción: El usuario podrá ingresar y almacenar diariamente los datos correspondientes a la ficha técnica de cada animal del proyecto vacas lecheras. Tareas realizadas -Diseño de interfaz de usuario para ingreso de datos ficha técnica vacas lecheras por día -Inserción de datos y almacenamiento de los mismos	
Observaciones:	

Historia de Usuario	
Número: 18	Prioridad en Negocio: Media
Nombre historia: Introducción y almacenamiento de datos fichas técnicas terneros	
Iteración Asignada: Cuarta	Riesgo en Desarrollo: Bajo
Programador responsable: Paola Andrea Manquillo	
Descripción: El usuario podrá ingresar y almacenar los datos correspondientes a la ficha técnica del proyecto terneros. Tareas -Diseño de interfaz de usuario para ingreso de datos ficha técnica terneros -Inserción de datos y almacenamiento de los mismos	
Observaciones:	

Historia de Usuario	
Número: 19	Prioridad en Negocio: Alta
Nombre historia: Introducción y almacenamiento de datos fichas técnicas aves de postura	
Iteración Asignada: Cuarta	Riesgo en Desarrollo: Medio
Programador responsable: Paola Andrea Manquillo	
Descripción: El usuario podrá ingresar y almacenar los datos correspondientes a la ficha técnica de cada galpón. Tareas -Diseño de interfaz de usuario para ingreso de datos de la ficha técnica de aves de postura -Inserción de datos y almacenamiento de los mismos en la base de datos	
Observaciones:	

Historia de Usuario	
Número: 20	Prioridad en Negocio: Alta
Nombre historia: Introducción y almacenamiento de datos fichas técnicas aves de postura/semana	
Iteración Asignada: cuarta	Riesgo en Desarrollo: Medio
Programador responsable: Paola Andrea Manquillo	
Descripción: El usuario podrá ingresar y almacenar los datos semanales correspondientes a la ficha técnica de cada galpón. Tareas -Diseño de interfaz de usuario para ingreso de datos de la ficha técnica de aves de postura por semana. -Inserción de datos y almacenamiento de los mismos	
Observaciones:	

Historia de Usuario	
Número: 21	Prioridad en Negocio: Medio
Nombre historia: Introducción y almacenamiento de datos fichas técnicas proyecto apiarios	
Iteración Asignada: Cuarta	Riesgo en Desarrollo: Medio
Programador responsable: Paola Andrea Manquillo	
Descripción: El usuario podrá ingresar y almacenar los datos correspondientes a cada uno de los productos del proyecto apiarios. Tareas realizadas -Diseño de interfaz de usuario para ingreso de datos de la ficha técnica de proyecto apiarios. -Inserción de datos y almacenamiento de los mismos en la base de datos	
Observaciones:	

Historia de Usuario	
Número: 22	Prioridad en Negocio: Media
Nombre historia: Introducción y almacenamiento de datos fichas técnicas proyecto apiarios/labores semanales	
Iteración Asignada: Cuarta	Riesgo en Desarrollo: Medio
Programador responsable: Paola Andrea Manquillo	
Descripción: El usuario podrá ingresar y almacenar los datos correspondientes a las labores semanales del proyecto apiarios. Tareas realizadas -Diseño de interfaz de usuario para ingreso de datos ficha técnica proyecto apiarios. -Inserción de datos y almacenamiento de los mismos en la base de datos	
Observaciones:	

Historia de Usuario	
Número: 23	Prioridad en Negocio: Media
Nombre historia: Introducción y almacenamiento de datos ficha técnica hembras de cría otras especies	
Iteración Asignada: Quinta	Riesgo en Desarrollo: Bajo
Programador responsable: Paola Andrea Manquillo	
Descripción: El usuario podrá ingresar y almacenar los datos correspondientes al registro de la ficha técnica de otras especies diferentes a las ya tratadas. Tareas relacionadas. -Diseño de interfaz de usuario para ingreso de datos ficha técnica otras especies -Inserción de datos y almacenamiento de los mismos en la base de datos	
Observaciones:	

Historia de Usuario	
Número: 24	Prioridad en Negocio: alta
Nombre historia: Introducción y almacenamiento de datos fichas técnicas hembras de cría otras especies/labores semanales	
Iteración Asignada: Quinta	Riesgo en Desarrollo: Bajo
Programador responsable:	
Descripción: El usuario podrá ingresar y almacenar los datos correspondientes a las labores semanales de cada uno de los proyectos de otras especies. Tareas realizadas. -Diseño de interfaz de usuario para ingreso de datos de labores semanales en otras especies. -Inserción de datos y almacenamiento de los mismos en la base de datos.	
Observaciones:	

Historia de Usuario	
Número: 25	Prioridad en Negocio:
Nombre historia: Introducción y almacenamiento de datos fichas técnicas crías otras especies	
Iteración Asignada: Quinta	Riesgo en Desarrollo: Bajo
Programador responsable:	
Descripción: El usuario podrá ingresar y almacenar los datos correspondientes a las crías pertenecientes a proyectos de otras especies. Tareas realizadas. -Diseño de interfaz de usuario para ingreso de datos crías otras especies -Inserción de datos y almacenamiento de los mismos en la base de datos.	
Observaciones:	

Historia de Usuario	
Número: 26	Prioridad en Negocio:
Nombre historia: Introducción y almacenamiento de datos fichas técnicas crías otras especies/labores semanales	
Iteración Asignada: Quinta	Riesgo en Desarrollo: Bajo
Programador responsable:	
Descripción: El usuario podrá ingresar y almacenar los datos correspondientes a las labores semanales pertenecientes a proyectos crías otras especies. Tareas realizadas. -Diseño de interfaz de usuario para ingreso de datos crías otras especies/labores semanales -Inserción de datos y almacenamiento de los mismos en la base de datos.	
Observaciones:	

Historia de Usuario	
Número: 27	Prioridad en Negocio: Bajo
Nombre historia: Introducción y almacenamiento de datos bienes e insumos	
Iteración Asignada: Quinta	Riesgo en Desarrollo: Medio
Programador responsable: Paola Andrea Manquillo	
Descripción: El usuario podrá ingresar y almacenar los datos correspondientes a los bienes e insumos que ubicados en la finca y el estado en que éstos se encuentran. Tareas realizadas: -Diseño de menú inventarios -Diseño de interfaz de usuario para ingreso de datos bienes e insumos. -Inserción de datos y almacenamiento de los mismos en la base de datos	
Observaciones:	

Historia de Usuario	
Número: 28	Prioridad en Negocio: Baja
Nombre historia: Introducción y almacenamiento de datos inventario agropecuario	
Iteración Asignada: Quinta	Riesgo en Desarrollo: Bajo
Programador responsable: Paola Andrea Manquillo	
Descripción: El usuario podrá ingresar y almacenar los datos correspondientes al inventario agropecuario. Tareas realizadas -Diseño de interfaz de usuario para ingreso de datos inventario agropecuario -Inserción de datos y almacenamiento de los mismos	
Observaciones:	

Historia de Usuario	
Número: 29	Prioridad en Negocio: baja
Nombre historia: Introducción y almacenamiento de datos fichas técnicas porcinos, etapa lactancia por fechas	
Iteración Asignada: Sexta	Riesgo en Desarrollo: Alto
Programador responsable: Paola Andrea Manquillo	
Descripción: El usuario podrá ingresar y almacenar los datos correspondientes a la ficha técnica de cada cerda de cría y su camada entre un par de fechas determinadas. Tareas: -Diseño de interfaz de usuario para ingreso de datos ficha técnica porcinos, etapa lactancia por fechas. -Inserción de datos y almacenamiento de los mismos.	
Observaciones:	

Historia de Usuario	
Número: 30	Prioridad en Negocio: media
Nombre historia: Introducción y almacenamiento de datos fichas técnicas porcinos, otras etapas por fechas	
Iteración Asignada: Sexta	Riesgo en Desarrollo: Alto
Programador responsable: Paola Andrea Manquillo	
Descripción: El usuario podrá ingresar y almacenar los datos correspondientes a la ficha técnica de cada etapa entre un par de fechas determinadas. Tareas: -Diseño de interfaz de usuario para ingreso de datos ficha técnica porcinos, otras etapas por fechas. -Inserción de datos y almacenamiento de los mismos.	
Observaciones	

Historia de Usuario	
Número: 31	Prioridad en Negocio: baja
Nombre historia: Agregar calendario	
Iteración Asignada: Sexta	Riesgo en Desarrollo: Bajo
Programador responsable: Paola Andrea Manquillo	
Descripción: El usuario dispondrá de un calendario para ingreso de fechas, de modo que se reduce la posibilidad de errores en el ingreso de fechas en formato java.	
Observaciones:	

Historia de Usuario	
Número: 32	Prioridad en Negocio: Media
Nombre historia: Introducción de fichas técnicas agrícolas por fechas	
Iteración Asignada: Sexta	Riesgo en Desarrollo: Alto
Programador responsable: Paola Andrea Manquillo	
Descripción: El usuario podrá ingresar y almacenar los datos correspondientes a la ficha técnica de cada cultivo entre un par de fechas determinadas. Tareas: -Diseño de interfaz de usuario para ingreso de datos ficha técnica agrícola por fechas. -Inserción de datos y almacenamiento de los mismos.	
Observaciones:	

Historia de Usuario	
Número: 33	Prioridad en Negocio: Media
Nombre historia: Generación de reporte en formato Excel de costos individuales	
Iteración Asignada: Sexta	Riesgo en Desarrollo: Alto
Programador responsable: Paola Andrea Manquillo	
Descripción: Se generará un reporte en formato Excel donde se muestra el registro de costos individuales, el detalle de los costos. Esto le permite al usuario imprimir dicho reporte con lo que podrá sustentar sus informes a la Vicerectoría Administrativa.	
Observaciones: Consulta en base de datos para selección de la tabla correspondiente. Creación de hoja de cálculo e incluir la información de la tabla	

Historia de Usuario	
Número: 34	Prioridad en Negocio: Media
Nombre historia: Generación de reporte en formato Excel de ventas diarias	
Iteración Asignada: Sexta	Riesgo en Desarrollo: Alto
Programador responsable: Paola Andrea Manquillo	
Descripción: El usuario tendrá acceso a un reporte en formato Excel, que muestra los productos vendidos, la fecha de venta y el valor de la venta. Una vez generado este reporte el usuario podrá imprimirlo y así soportar sus informes a la Vicerectoría Administrativa.	
Observaciones: Consulta en base de datos para selección de la tabla correspondiente. Creación de hoja de cálculo e incluir la información de la tabla	

Historia de Usuario	
Número: 35	Prioridad en Negocio: Media
Nombre historia: Generación de reporte en formato Excel de utilidades para un producto determinado.	
Iteración Asignada: Sexta	Riesgo en Desarrollo: Alto
Programador responsable: Paola Andrea Manquillo	
Descripción: El reporte generado mostrará las utilidades totales para todos los productos a la fecha. También se mostrarán aquí los costos totales y los ingresos totales a la fecha.	
Observaciones:	

2. PRUEBAS

Este documento cubre el conjunto de pruebas funcionales relacionadas con cada una de las distintas historias de usuario:

2.1 Historia de usuario 1: Crear/eliminar microproyectos estándar

Descripción

En este caso hay que comprobar la introducción y almacenamiento correcto de los datos correspondientes a los microproyectos estándar. Un microproyecto estándar consta de múltiples datos y puede ocurrir que algunos sean válidos mientras que otros no. En el proceso se guardan sólo aquellos microproyectos que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y el proyecto no será guardado y si los datos son incompletos se le solicita al cliente que introduzca los datos completos. En el caso de que la introducción de datos sea correcta, éstos son almacenados en la base de datos. Las pruebas fueron:

2.1.1 Ingreso y almacenamiento correcto de microproyectos estándar

Descripción

El usuario (administrador agropecuario), una vez haya entrado en el sistema, seleccionará la opción "Proyectos" del Menú principal. En la ventana proyectos seleccionará "Proyectos Estándar". Tras esa elección se le mostrará la interfaz en la que ingresa los datos correctos correspondientes al proyecto, internamente se procesan los datos y dado que no hay ningún error (datos completos y válidos), el proyecto es guardado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Entrada

- El Administrador Agropecuario seleccionará "proyectos" del Menú Principal.
- Del Menú Proyectos seleccionará "Proyectos Estándar".
- A continuación se muestra una interfaz donde el debe ingresar los siguientes datos:

Datos solicitados	Datos validos
Nombre del proyecto	Tipo texto
Tipo de proyecto	Seleccionar una de las opciones :agrícola o pecuario
Área utilizada	Introducir cantidad (con punto flotante) y seleccionar unidad (hectáreas o metros cuadrados)
Numero de unidades	Entero
Fecha de entrada	Seleccionar fecha del calendario

Tabla 2.1: Datos de entrada Historia de usuario 1

-Finalmente selecciona guardar. El proceso de ingreso de microproyectos estándar se considera como finalizado.

Resultado Esperado

El proyecto estándar es almacenado en la base de datos y el usuario podrá visualizarlo mediante la lista desplegada en la interfaz y, haciendo clic sobre el nombre del proyecto, podrá visualizar los datos correspondientes a dicho microproyecto.

Evaluación de la Prueba: Prueba satisfactoria

2.1.2 Introducción de microproyectos estándar con datos erróneos

Descripción

El usuario (administrador agropecuario), una vez haya entrado en el sistema, seleccionará la opción "Proyectos" del Menú principal. En la ventana proyectos seleccionará "Proyectos Estándar". Tras esa elección se mostrará la interfaz en la que ingresa los datos correspondientes al proyecto, internamente se procesan los datos y se encuentra algún error en uno o más de los datos ingresados, (alguno/s de los datos ingresados no corresponde al tipo de dato valido). En este caso el proyecto no es guardado

Condiciones de ejecución

Instalación y ejecución de SIGA

Entrada

- El Administrador Agropecuario seleccionará "proyectos" del Menú Principal.
- Del Menú Proyectos seleccionará "Proyectos Estándar".
- A continuación se muestra una interfaz en donde se ingresa uno o más datos de tipo distinto al indicado en la tabla 2.1. El proceso de introducción de proyectos estándar se considera como erróneo.

Resultado Esperado

Internamente se procesan los datos y se encuentran datos no válidos por el sistema en alguno de los campos, en ese caso, se le muestra al usuario un mensaje de error que le informa que ocurrió un error en su ingreso de datos.

Evaluación de la Prueba: Prueba satisfactoria

2.1.3 Ingreso incompleto de datos en microproyectos estándar

El usuario omite el ingreso de uno o más de los datos solicitados, en este caso, se mostrará un mensaje solicitando ingresar los datos completos.

Condiciones de ejecución

Instalación y ejecución de SIGA

Entrada

- El Administrador Agropecuario seleccionará "proyectos" del Menú Principal.
- Del Menú Proyectos seleccionará "Proyectos Estándar".
- A continuación se muestra una interfaz donde el usuario omite ingresar uno de los datos indicados en la tabla 2.1 finalmente selecciona guardar
- El proceso de ingreso de microproyectos estándar se considera como incompleto y debe repetirse.

Resultado Esperado

El proyecto no es almacenado en la base de datos, en cambio se muestra un mensaje al usuario que le sugiere introducir los datos completos.

Evaluación de la Prueba: Prueba satisfactoria

2.1.4 Eliminación de un microproyecto estándar

Descripción

El usuario (administrador agropecuario), una vez haya entrado en el sistema, seleccionará la opción "Proyectos" del Menú principal. En la ventana proyectos seleccionará "Proyectos Estándar". Tras esa elección se mostrará la interfaz para ingreso de datos correspondientes a un macroproyecto estándar, en la misma interfaz podrá visualizar también un listado de los proyectos estándar ingresados hasta el momento. El usuario seleccionará uno de los proyectos del listado y a continuación la opción borrar, con ello, el proyecto será eliminado de la base de datos.

Condiciones de ejecución

Instalación y ejecución de SIGA

Entrada

- El Administrador Agropecuario seleccionará "proyectos" del Menú Principal.
- Del Menú Proyectos seleccionará "Proyectos Estándar".
- En la interfaz de ingreso de datos, se muestra además la lista de los proyectos estándar ingresados, el usuario selecciona uno de los proyectos.
- Selecciona la opción borrar.

Resultado Esperado

El proyecto es eliminado de la base de datos y por lo tanto, ya no se mostrará en el listado de los proyectos estándar existentes.

Evaluación de la Prueba: Prueba satisfactoria

2.2 Historia de usuario 2: Crear/eliminar microproyectos especiales

En este caso hay que comprobar la introducción y almacenamiento correcto de los datos correspondientes a los microproyectos especiales. Un microproyecto especial consta de múltiples datos y puede ocurrir que algunos sean válidos mientras que otros no. En el proceso se guardan sólo aquellos microproyectos que ingresen los datos correctos. Si los datos ingresados no son válidos, es decir, si se ingresa un tipo de dato distinto al válido, se genera un mensaje de error y el proyecto no será guardado y si los datos son incompletos se le solicita al cliente que introduzca los datos completos. En el caso de que la introducción de datos sea correcta, éstos son almacenados en la base de datos. Las pruebas fueron:

Entrada

- El Administrador Agropecuario seleccionará "proyectos" del Menú Principal.
- Del Menú Proyectos seleccionará "Proyectos Especiales".
- A continuación se muestra una interfaz donde el debe ingresar los siguientes datos:

Datos solicitados	Datos validos
Nombre del proyecto	Tipo texto
Tipo de proyecto	Seleccionar una de las opciones: porcino, bovino, apiario, aves de postura, otro pecuario)
Área utilizada	Introducir cantidad (con punto flotante) y seleccionar unidad (hectáreas o metros cuadrados)
Nombre de animal	Tipo texto
Tipo de animal	Seleccionar una opción (etapa gestación, lactancia, levante, precebo, ceba)
UDC	Tipo entero
Fecha de entrada	De la interfaz proyectos especiales seleccionar una fecha del calendario

Tabla 2.2: Datos de entrada historia de usuario 2

2.2.1 Ingreso y almacenamiento correcto de microproyectos especiales

El usuario (administrador agropecuario), una vez haya entrado en el sistema, seleccionará la opción "Proyectos" del Menú principal. En la ventana proyectos seleccionará "Proyectos Especiales". Tras esa elección se le mostrará la interfaz en la que ingresa los datos correctos, como lo indica la tabla 2.2, posteriormente, selecciona la opción guardar. Internamente se procesan los datos y dado que no hay ningún error (datos completos y válidos), el proyecto es guardado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

El proyecto especial es almacenado en la base de datos y el usuario podrá visualizarlo mediante la lista desplegada en la interfaz y, haciendo clic sobre el nombre del proyecto especial, podrá visualizar los datos correspondientes a dicho microproyecto. El proceso de considera finalizado

Evaluación de la Prueba: Prueba satisfactoria

2.2.2 Introducción de microproyectos especiales con datos erróneos

El usuario (administrador agropecuario), una vez haya entrado en el sistema, seleccionará la opción "Proyectos" del Menú principal. En la ventana proyectos seleccionará "Proyectos Especiales". Tras esa elección se le mostrará la interfaz en la que ingresa un tipo de dato distinto al indicado en la tabla 2.2, posteriormente, selecciona la opción guardar. Internamente se procesan los datos y en este caso, se muestra un mensaje de error y el proyecto especial no es guardado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Internamente se procesan los datos y se encuentran datos no válidos (distintos de los indicados en la tabla 2.2), en ese caso, se le muestra al usuario un mensaje de error que le informa que ocurrió un error en su ingreso de datos.

Evaluación de la Prueba: Prueba satisfactoria

2.2.3 Ingreso incompleto de datos en microproyectos especiales

El usuario (administrador agropecuario), una vez haya entrado en el sistema, seleccionará la opción "Proyectos" del Menú principal. En la ventana proyectos seleccionará "Proyectos Especiales". Tras esa elección se le mostrará la interfaz en la que omite el ingreso de uno o más de los datos mostrados en la tabla 2.2 y posteriormente seleccionar guardar.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

El proyecto especial no es almacenado en la base de datos, y se muestra un mensaje al usuario que le sugiere introducir los datos completos.

Evaluación de la Prueba: Prueba satisfactoria

2.2.4 Eliminación de un microproyecto especial

El usuario (administrador agropecuario), una vez haya entrado en el sistema, seleccionará la opción "Proyectos" del Menú principal. En la ventana proyectos seleccionará "Proyectos Especiales". Tras esa elección se le mostrará la interfaz para el ingreso de datos correspondientes, el usuario seleccionará uno de los proyectos del listado y a continuación la opción borrar, con ello, el proyecto será eliminado de la base de datos.

Condiciones de ejecución

Instalación y ejecución de SIGA

Entrada

- El Administrador Agropecuario seleccionará "proyectos" del Menú Principal.
- Del Menú Proyectos seleccionará "Proyectos Especiales".
- En la interfaz de ingreso de datos, se muestra además la lista de los proyectos especiales ingresados, el usuario selecciona uno de los proyectos.
- Selecciona la opción borrar.

Resultado Esperado

El proyecto es eliminado de la base de datos y por lo tanto, ya no se mostrará en el listado de los proyectos estándar existentes.

Evaluación de la Prueba: Prueba satisfactoria

2.3 historia de usuario 3: crear/eliminar productos

En este caso hay que comprobar la introducción y almacenamiento correcto de los datos correspondientes a los productos de un microproyecto estándar o especial, si intenta

guardar un producto sin su introducción previa de los datos se despliega un mensaje en pantalla indicando al usuario que es necesario que ingrese los datos. En el caso de que la introducción de datos sea correcta, éstos son almacenados en la base de datos y el producto ingresado será mostrado cuando el usuario seleccione el microproyecto al que pertenece.

2.3.1 Ingreso correcto de productos

El usuario (administrador agropecuario), una vez haya entrado en el sistema, seleccionará la opción "Productos" del Menú principal, del menú productos seleccionará "Crear productos". A continuación se mostrará una interfaz en la que se muestra un listado de todos los proyectos creados especiales o estándar, de donde él selecciona uno de los proyectos y enseguida ingresa los productos que se generan en dicho proyecto. El tipo de dato válido para productos es texto, a continuación el usuario selecciona la opción guardar, internamente se procesan los datos y dado que no hay ningún error (datos completos y válidos), el producto es guardado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Entrada

- El Administrador Agropecuario seleccionará "productos" del Menú Principal.
- Del menú productos seleccionará "Crear productos"
- A continuación se muestra una interfaz donde el usuario debe: seleccionar el proyecto, a continuación ingresará el nombre del producto. Finalmente debe seleccionar la opción "Guardar" para que el producto quede almacenado en la base de datos. En caso de que un proyecto genere más de un producto, se repite este procedimiento para cada uno de los productos.
- El proceso de ingreso de productos se considera como finalizado.

Resultado Esperado

El producto es almacenado en la base de datos y el usuario podrá visualizarlo en la interfaz de ingreso de productos

Evaluación de la Prueba: Prueba satisfactoria

2.3.2 Ingreso incompleto de datos en interfaz productos

Descripción

El usuario (administrador agropecuario), una vez haya entrado en el sistema, seleccionará la opción "Productos" del Menú principal, del menú productos seleccionará "Crear productos"

A continuación se mostrará una interfaz en la que se muestra un listado de todos los proyectos creados especiales o estándar, de donde él selecciona uno de los proyectos pero omite ingresar los productos que se generan en dicho proyecto. A continuación el usuario selecciona la opción guardar, internamente se procesan los datos y dado que no hay algún dato ingresado, se desplegará en pantalla un mensaje que le sugiere al usuario ingresar los datos.

Condiciones de ejecución

Instalación y ejecución de SIGA

Entrada

- El Administrador Agropecuario seleccionará "productos" del Menú Principal.
- Del menú productos seleccionará "Crear productos"
- A continuación se muestra una interfaz donde el usuario debe: seleccionar el proyecto. Finalmente debe seleccionar la opción "Guardar", pero debido a que no ha ingresado el nombre del producto, internamente se procesan los datos y se muestra un mensaje en pantalla solicitando al usuario ingresar los datos.

Resultado Esperado

Se despliega un mensaje en pantalla que le indica al usuario que no se han introducido todos los datos.

Evaluación de la Prueba: Prueba satisfactoria

2.3.3 Eliminación de un producto

Descripción

El usuario (administrador agropecuario), una vez haya entrado en el sistema, seleccionará la opción "Productos" del Menú principal, del menú productos seleccionará "Crear productos" A continuación se mostrará una interfaz en la que se muestra un listado de todos los proyectos creados especiales y estándar, de donde él selecciona uno de los proyectos. Al seleccionar uno de los proyectos se despliegan los productos ingresados para dicho proyecto, el usuario seleccionará uno de éstos y a continuación la opción borrar, internamente se procesa la opción seleccionada y el producto escogido es eliminado de la base de datos.

Condiciones de ejecución

Instalación y ejecución de SIGA

Entrada

- El Administrador Agropecuario seleccionará "productos" del Menú Principal
- Del menú productos seleccionará "Crear productos"
- En la interfaz de ingreso de productos, selecciona uno de los proyectos y se mostrarán los productos pertenecientes al proyecto seleccionado.
- Selecciona el producto que desea borrar
- Selecciona la opción borrar

Resultado Esperado

El producto escogido es borrado de la base de datos, y no se mostrará en pantalla cuando el usuario seleccione desde la interfaz de productos, el proyecto al que pertenece.

Evaluación de la Prueba: Prueba satisfactoria

2.4 Historia de usuario 4: Crear/eliminar unidades

Descripción

En este caso se comprobará la introducción correcta de los datos correspondientes a las unidades de medida de los diferentes productos, éste es un listado general, no hará falta especificar para cada producto sus unidades, pues las unidades ingresadas estarán disponibles para todos los productos. El usuario seleccionará la opción Productos del Menú

Principal, del menú productos seleccionará "Crear unidades", a continuación se mostrará la interfaz para ingreso de unidades, en donde introduce el nombre de la unidad y posteriormente la opción guardar. Internamente se procesan los datos ingresados y el nombre de la unidad será guardado en la base de datos.

2.4.1 Ingreso correcto de Unidades

Condiciones de ejecución

Instalación y ejecución de SIGA

Entrada

- El Administrador Agropecuario seleccionará "Productos" del Menú Principal.
- Del menú productos seleccionará "Crear unidades"
- Ingresa el nombre de la unidad de medida de los productos. Finalmente debe seleccionar la opción "Guardar" y la unidad ingresada quedará almacenada en la base de datos.
- El proceso de ingreso de unidades se considera como finalizado.

Resultado Esperado

Las unidades son almacenadas en la base de datos y el usuario podrá visualizarlas en el momento que desea registrar los ingresos.

Evaluación de la Prueba: Prueba satisfactoria

2.4.2 Ingreso incompleto de datos en interfaz unidades

Condiciones de ejecución

Instalación y ejecución de SIGA

Entrada

- El Administrador Agropecuario seleccionará "Productos" del Menú Principal.
- Del menú productos seleccionará "Crear unidades"
- A continuación se muestra una interfaz donde el debe ingresar el nombre de la unidad pero no ingresa los datos respectivos, a continuación debe seleccionar la opción "Guardar", pero debido a que no ha ingresado el nombre de la unidad, internamente se procesan los datos y se muestra un mensaje en pantalla solicitando al usuario ingresar los datos.

Resultado Esperado

Se despliega un mensaje en pantalla que le indica al usuario que no se han introducido todos los datos.

Evaluación de la Prueba: Prueba satisfactoria

2.4.3 Eliminación de unidades

Descripción

El usuario (administrador agropecuario), una vez haya entrado en el sistema, seleccionará la opción "Productos" del Menú principal, del menú productos seleccionará la opción "crear unidades". A continuación se muestra la interfaz de ingreso de unidades, donde aparece el listado de las unidades ingresadas hasta el momento, el usuario seleccionará de dicho

listado la unidad que desea eliminar y posteriormente la opción borrar. Internamente se procesan los datos y la unidad escogida es borrada de la base de datos, (la unidad también se borra del listado de unidades de la interfaz de ingreso de unidades).

Condiciones de ejecución

Instalación y ejecución de SIGA

Entrada

- El Administrador Agropecuario seleccionará "productos" del Menú Principal.
- Del menú productos seleccionará "crear unidades"
- En la interfaz de ingreso de unidades, selecciona una de las unidades del listado, (la que desea eliminar)
- Selecciona la opción borrar

Resultado Esperado

La unidad escogida es borrada de la base de datos, y no se mostrará en la interfaz "Crear Unidades".

Evaluación de la Prueba: Prueba satisfactoria

2.5 Historia de usuario 5: Introducir costos individuales

Descripción

En este caso hay que comprobar la introducción correcta de los datos correspondientes a los costos individuales de cada uno de los productos que hacen parte de los microproyectos. En este proceso de introducción de costos pueden ocurrir errores por ingreso de datos no válidos. Si los datos ingresados no son válidos, se genera un mensaje de error y el costo no podrá ser guardado, si los datos son incompletos se le solicita al cliente que introduzca los datos completos. En caso que la introducción de datos sea correcta, éstos son almacenados en la base de datos.

Entrada

- El Administrador Agropecuario seleccionará "gastos" del Menú Principal.
- Del Menú gastos seleccionará "Costos individuales".
- A continuación se muestra una interfaz donde el debe ingresar los siguientes datos:

Datos solicitados	Datos Validos
Seleccionar nombre del proyecto	De la interfaz costos individuales seleccionar uno de los proyectos ya ingresados
Seleccionar Nombre del producto	De la interfaz costos individuales seleccionar uno de los productos ya ingresados
Nombre del gasto	Descripción corta del gasto (tipo texto)
Tipo de gasto	Seleccionar una opción entre inversión, producción, mantenimiento y otros
Valor del gasto	Valor flotante
Fecha de gasto	Seleccionar una fecha del calendario

Tabla2.3: Datos de entrada historia de usuario 5

- El proceso de ingreso de costos individuales se considera como finalizado.

2.5.1 Ingreso correcto de costos individuales

En este caso se ingresan todos los datos según se indica en la tabla 2.3 y se selecciona guardar.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Los costos individuales son almacenados en la base de datos y estarán disponibles para cálculos posteriores

Evaluación de la Prueba: Prueba satisfactoria

2.5.2 Introducción de costos individuales con datos erróneos

En este caso se ingresa uno o más datos diferentes de los datos válidos indicados en la tabla 2.3 y se seleccionar guardar.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Internamente se procesan los datos y se encuentran datos no válidos por el sistema en alguno de los campos, en ese caso, se le muestra al usuario un mensaje de error que le informa que ocurrió un error en su ingreso de datos.

Evaluación de la Prueba: Prueba satisfactoria

2.5.3 Ingreso incompleto de datos en costos individuales

En este caso se omite el ingreso de uno o más datos de la tabla 2.3, y se selecciona la opción guardar.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

El costo individual no es almacenado en la base de datos, en cambio se le muestra un mensaje al usuario que le sugiere introducir los datos completos.

Evaluación de la Prueba: Prueba satisfactoria

2.6 Historia de usuario 6: Calcular costos totales

Descripción:

En este caso hay que comprobar el cálculo correcto de los costos totales, a partir de los costos individuales ingresados. En este proceso de cálculo de costos pueden ocurrir errores por omisión de datos de entrada o errores de cálculo. Si se omite ingresar datos necesarios para calcular el costo, internamente se procesan los datos y se desplegará un mensaje en

pantalla informando al usuario que debe ingresar los datos completos. En el caso de que la introducción de datos sea correcta, se procesan los datos y se procede a realizar el cálculo, dicho costo total es almacenado en la base de datos.

Entrada

- El Administrador Agropecuario seleccionará "gastos" del Menú Principal.
- Del Menú gastos seleccionará "Costos totales".
- A continuación se muestra una interfaz donde el debe seleccionar los siguientes datos:

Datos solicitados	Datos Validos
Seleccionar nombre del proyecto	De la interfaz costos totales seleccionar uno de los proyectos ya ingresados
Seleccionar Nombre del producto	De la interfaz costos totales seleccionar uno de los productos ya ingresados

Tabla 2.4: Datos de entrada Historia de usuario 6

2.6.1 Cálculo correcto de costos totales

El usuario (administrador agropecuario), una vez haya entrado en el sistema, seleccionará la opción "gastos" del Menú principal. En el Menú gastos seleccionará "costos totales". Tras esa elección se le mostrará la interfaz de cálculo de costos totales, en la que selecciona primero el microproyecto, luego el producto y por último selecciona la opción calcular, el sistema procesa los datos seleccionados y mostrará en pantalla el resultado de la suma de los costos individuales para el producto seleccionado. Finalmente debe seleccionar la opción guardar para almacenar este valor en la base de datos. El proceso de cálculo y almacenamiento de costos totales se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se debe mostrar al usuario el resultado de la suma de cada uno de los costos individuales para el producto seleccionado. Este resultado puede ser guardado en la base de datos.

Evaluación de la Prueba: Prueba satisfactoria

2.6.2 Cálculo incompleto de costos totales

El usuario (administrador agropecuario), una vez haya entrado en el sistema, seleccionará la opción "gastos" del Menú principal. En el Menú gastos seleccionará "costos totales". Tras esa elección se le mostrará la interfaz de cálculo de costos totales. Se considera que el cálculo es incorrecto si el usuario omite el ingreso de uno o más datos solicitados de la tabla 2.4. Internamente se procesan los datos y al no ser posible realizar el cálculo por la falta de algún dato éste no se lleva a cabo, entonces se le mostrará al usuario un mensaje en el que se le informa que es necesario ingresar todos los datos necesarios.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Internamente se procesan los datos y se encuentra que no se puede realizar el cálculo, entonces se le muestra un mensaje al usuario donde se le informa que es necesario ingresar todos los datos.

Evaluación de la prueba: prueba satisfactoria

2.7 Historia de usuario 7: Introducir Ingresos individuales

Descripción

En este caso hay que comprobar la introducción correcta de los datos correspondientes a los ingresos individuales de cada uno de los productos que hacen parte de los microproyectos. En este proceso de introducción de ingresos pueden ocurrir errores por ingreso de datos no válidos. Si los datos introducidos no son válidos, se genera un mensaje de error y el ingreso no podrá ser guardado, si los datos son incompletos se le solicita al cliente que introduzca los datos completos. En el caso de que la introducción de datos sea correcta, éstos son almacenados en la base de datos.

Entrada

- El Administrador Agropecuario seleccionará "Ingresos" del Menú Principal.
- Del Menú ingresos seleccionará "Ingresos individuales".
- A continuación se muestra una interfaz donde el debe introducir los siguientes datos:

Datos solicitados	Datos Validos
Seleccionar nombre del proyecto	De la interfaz ventas diarias seleccionar uno de los proyectos ya ingresados
Seleccionar Nombre del producto	De la interfaz ventas diarias seleccionar uno de los productos ya ingresados
Cantidad vendida,	Tipo entero
Seleccionar unidad	Seleccionar una de las opciones de unidad disponibles
Valor unitario	Valor flotante
Fecha de venta	De la interfaz ventas diarias seleccionar la fecha del calendario

Tabla 2.5 Datos de entrada Historia de Usuario 7

2.7.1 Introducción correcta de ingresos individuales

Se ingresan los datos según lo indica la tabla 2.5 y posteriormente se selecciona la opción calcular. El proceso de introducción de costos individuales se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se calcula el valor del ingreso individual y tras seleccionar guardar, este resultado se almacenará en la base de datos y estará disponibles para cálculos posteriores

Evaluación de la Prueba: Prueba satisfactoria

2.7.2 Introducción de ingresos individuales con datos erróneos

Se ingresan uno o más datos diferentes a los indicados en la tabla 2.5 y posteriormente se selecciona la opción calcular. El proceso de introducción de costos individuales se considera como erróneo.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Internamente se procesan los datos y se encuentran datos no válidos por el sistema en alguno de los campos, en ese caso, estos ingresos no son almacenados en la base de datos.

Evaluación de la Prueba: Prueba satisfactoria

2.7.3 Ingreso incompleto de datos en ingresos individuales

Se omite el ingreso de uno o más datos de la tabla 2.5 y posteriormente se selecciona la opción calcular. El proceso de introducción de costos individuales se considera como incompleto.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se mostrará en pantalla un mensaje que informa al usuario que no se han ingresado todos los datos.

Evaluación de la Prueba: Prueba satisfactoria

2.8 Historia de usuario 8: Calcular ingresos totales

Descripción

En este caso hay que comprobar el cálculo correcto de los costos totales, obtenidos de la suma de los distintos ingresos individuales. En este proceso de cálculo de ingresos totales pueden ocurrir errores. Si los datos ingresados no son suficientes, se genera un mensaje que se despliega en pantalla y solicita todos los datos. En el caso de que la introducción de datos sea correcta, son almacenados en la base de datos.

Entrada

- El Administrador Agropecuario seleccionará "Ingresos" del Menú Principal.
- Del Menú ingresos seleccionará "Ingresos totales".
- A continuación se muestra una interfaz donde el debe ingresar los siguientes datos:

Datos solicitados	Datos Validos
Seleccionar nombre del proyecto	De la interfaz "calcular ingresos" seleccionar uno de los proyectos ya ingresados
Seleccionar Nombre del producto	De la interfaz "calcular ingresos" seleccionar uno de los productos ya ingresados

Seleccionar unidad	De la interfaz "calcular ingresos" seleccionar una de las opciones de unidad disponibles
--------------------	--

Tabla 2.6: Datos de entrada historia de usuario 8

2.8.1 Cálculo correcto de ingresos totales

Se ingresan los datos según lo indica la tabla 2.6 y posteriormente se selecciona la opción calcular. El proceso de cálculo de costos totales se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se calcula el valor de los ingresos totales y tras seleccionar guardar, el resultado es almacenado en la base de datos y estará disponible para cálculos posteriores.

Evaluación de la Prueba: Prueba satisfactoria

2.8.2 Inserción incompleta de datos para cálculo de ingresos totales

Se omite el ingreso de uno o más datos de la tabla 2.6 y posteriormente se selecciona la opción calcular. El proceso de cálculo de costos totales se considera como incompleto.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se muestra en pantalla un mensaje donde se le informa al usuario que debe ingresar todos los datos completos.

2.9 Historia de usuario 9: Cálculo de costos por fecha

Descripción

En este caso hay que comprobar el cálculo correcto de los costos comprendido entre una fecha inicial y una fecha final. En este proceso se pueden presentar errores por omitir ingresar alguno de los datos necesarios, si los datos ingresados no son suficientes, se genera un mensaje que se despliega en pantalla y solicita todos los datos. En el caso de que la introducción de datos sea correcta, éstos son procesados y se genera el resultado de los costos comprendidos entre las fechas ingresadas.

Entradas:

- El Administrador Agropecuario seleccionará "gastos" del Menú Principal.
- Del Menú Costos seleccionará "costos por fecha".
- A continuación se muestra una interfaz donde el debe ingresar los siguientes datos:

Datos solicitados	Datos Validos
Seleccionar nombre del proyecto	De la interfaz costos por fecha seleccionar uno de los proyectos ya ingresados
Seleccionar Nombre del producto	De la interfaz costos por fecha seleccionar uno de los productos ya ingresados
fecha inicial de semana	De la interfaz costos por fecha, seleccionar fecha

Fecha final de semana	De la interfaz costos por fecha, seleccionar fecha
-----------------------	--

Tabla 2.7: Datos de entrada historia de usuario 9

2.9.1 Cálculo correcto de costos por fecha

Se ingresan los datos según lo indica la tabla 2.7 y posteriormente se selecciona la opción calcular. El proceso de cálculo de costos por fecha se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se muestra en pantalla el resultado de los costos totales para el período comprendido entre las fechas ingresadas.

Evaluación de la Prueba: Prueba satisfactoria

2.9.2 Cálculo de costos por fecha, cuando ingreso de datos es incompleto

Se omite ingresar uno o más datos de la tabla 2.7 y posteriormente se selecciona la opción calcular. El proceso de cálculo de costos por fecha se considera como incompleto.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se muestra en pantalla un mensaje donde se solicita al usuario el ingreso de todos los datos.

Evaluación de la Prueba: Prueba satisfactoria

2.10 Historia de Usuario 10: Calcular ingresos por fecha

Descripción

En este caso hay que comprobar el cálculo correcto de los ingresos comprendidos entre una fecha inicial y una fecha final. En este proceso se pueden presentar errores por omitir ingresar alguno de los datos necesarios, si los datos ingresados no son suficientes, se genera un mensaje que se despliega en pantalla y solicita todos los datos. En el caso de que la introducción de datos sea correcta, éstos son procesados y se genera el resultado de los ingresos comprendidos entre las fechas ingresadas.

Entrada

- El Administrador Agropecuario seleccionará "Ingresos" del Menú Principal.
- Del Menú Ingresos seleccionará "Ingresos por fecha".
- A continuación se muestra una interfaz donde el debe seleccionar los siguientes datos:

Datos solicitados	Datos Validos
Seleccionar nombre del proyecto	De la interfaz "Ingresos por fecha" seleccionar uno de los proyectos ya ingresados
Seleccionar Nombre del producto	De la interfaz "Ingresos por fecha" seleccionar uno de los productos ya ingresados
fecha inicial de semana	De la interfaz "Ingresos por fecha", seleccionar

	fecha
Fecha final de semana	De la interfaz "Ingresos por fecha", seleccionar fecha

Tabla 2.8: Datos de entrada historia de usuario 10

2.10.1 Cálculo correcto de ingresos por fecha

Se ingresan los datos según lo indica la tabla 2.8 y posteriormente se selecciona la opción calcular. El proceso de cálculo de ingresos por fecha se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se muestra en pantalla el resultado de los ingresos totales para el período comprendido entre las fechas ingresadas.

Evaluación de la Prueba: Prueba satisfactoria

2.10.2 Cálculo de ingresos por fecha, cuando ingreso de datos es incompleto

Se omite ingresar uno o más datos de la tabla 2.8 y posteriormente se selecciona la opción calcular. El proceso de cálculo de ingresos por fecha se considera como incompleto.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se muestra en pantalla un mensaje donde se solicita al usuario el ingreso de todos los datos.

Evaluación de la Prueba: Prueba satisfactoria

2.11 Historia de usuario 11: Calcular Utilidades totales

Descripción

En este caso hay que comprobar el cálculo correcto de las utilidades obtenidas de la diferencia entre los ingresos totales y los costos totales. En este proceso de cálculo de utilidades totales pueden presentarse errores por cálculos que arrojen resultados erróneos. En el caso de que la introducción de datos sea correcta, se realiza correctamente el cálculo y el resultado es almacenado en la base de datos.

2.11.1 Cálculo correcto de utilidades totales cuando costos totales son distintos de cero e ingresos totales son iguales a cero

Descripción

El usuario (administrador agropecuario), una vez haya entrado en el sistema, seleccionará la opción "Utilidades" del Menú principal. En el Menú Utilidades seleccionará "Utilidades totales". Tras esa elección se le mostrará la interfaz de cálculo de utilidades totales, el usuario seleccionará el proyecto y el producto del que desea conocer sus utilidades, finalmente selecciona la opción calcular, y se mostrará en pantalla el resultado de la

operación. En este caso se tuvo en cuenta realizar la operación para casos en que los costos totales son mayores que cero, un valor cualquiera, y los ingresos son cero.

Condiciones de ejecución

Instalación y ejecución de SIGA

Entrada

- El Administrador Agropecuario seleccionará "utilidades" del Menú Principal.
- Del Menú utilidades seleccionará "utilidades totales".
- A continuación se muestra una interfaz donde el debe seleccionar los siguientes datos: microproyecto, producto de ese microproyecto, por último se selecciona la opción calcular.
- El proceso de cálculo de utilidades totales se considera como finalizado.

Resultado Esperado

Se muestra en pantalla el resultado del cálculo de utilidades totales, que en este caso era el mismo valor de los costos con signo negativo

Evaluación de la Prueba: Prueba satisfactoria

2.11.2 Cálculo de utilidades cuando costos son mayores que ingresos, e ingresos mayores que cero

Descripción

El usuario (administrador agropecuario), una vez haya entrado en el sistema, seleccionará la opción "Utilidades" del Menú principal. En el Menú utilidades seleccionará "Utilidades totales". Tras esa elección se le mostrará la interfaz de cálculo de utilidades totales, en donde seleccionará primero el proyecto y luego el producto del que desea calcular las utilidades, finalmente selecciona la opción calcular, el sistema le mostrará el resultado de la diferencia entre los ingresos y costos, y el usuario tiene la opción de almacenar dicho resultado en la base de datos escogiendo la opción guardar.

Condiciones de ejecución

Instalación y ejecución de SIGA

Entrada

- El Administrador Agropecuario seleccionará "utilidades" del Menú Principal.
- Del Menú utilidades seleccionará "utilidades totales".
- A continuación se muestra una interfaz donde el debe seleccionar los siguientes datos: microproyecto, producto de ese microproyecto, por último se selecciona la opción calcular.
- El proceso de cálculo de utilidades totales se considera como finalizado.

Resultado Esperado

Se muestra en pantalla el resultado del cálculo de utilidades totales, que en este caso es la diferencia entre costos totales e ingresos totales, (será un número entero mayor que cero con signo negativo, ya que costos son mayores que ingresos).

Evaluación de la prueba: Satisfactoria

2.11.3 Cálculo de utilidades cuando ingresos son mayores que costos y costos diferentes de cero

Descripción

El usuario (administrador agropecuario), una vez haya entrado en el sistema, seleccionará la opción "Utilidades" del Menú principal. En el Menú utilidades seleccionará "Utilidades totales". Tras esa elección se le mostrará la interfaz de cálculo de utilidades totales, en donde seleccionará primero el proyecto y luego el producto del que desea calcular las utilidades, finalmente selecciona la opción calcular, el sistema le mostrará el resultado de la diferencia entre los ingresos y costos, y el usuario tiene la opción de almacenar dicho resultado en la base de datos escogiendo la opción guardar.

Condiciones de ejecución

Instalación y ejecución de SIGA

Entrada

- El Administrador Agropecuario seleccionará "utilidades" del Menú Principal.
- Del Menú utilidades seleccionará "utilidades totales".
- A continuación se muestra una interfaz donde el debe seleccionar los siguientes datos: microproyecto, producto de ese microproyecto, por último se selecciona la opción calcular.
- El proceso de cálculo de utilidades totales se considera como finalizado.

Resultado Esperado

Se muestra en pantalla el resultado del cálculo de utilidades totales, que en este caso es la diferencia entre costos totales e ingresos totales, (será un número entero mayor que cero ya que ingresos son mayores que costos)

Evaluación de la prueba: Satisfactoria

2.11.4 Cálculo de utilidades cuando ingresos son mayores que costos y costos iguales a cero

Descripción

El usuario (administrador agropecuario), una vez haya entrado en el sistema, seleccionará la opción "Utilidades" del Menú principal. En el Menú utilidades seleccionará "Utilidades totales". Tras esa elección se le mostrará la interfaz de cálculo de utilidades totales, en donde seleccionará primero el proyecto y luego el producto del que desea calcular las utilidades, finalmente selecciona la opción calcular, el sistema le mostrará el resultado de la diferencia entre los ingresos y costos, en este caso el resultado de las utilidades es igual al de los ingresos puesto que no se han introducido costos. El usuario tiene la opción de almacenar dicho resultado en la base de datos escogiendo la opción guardar.

Condiciones de ejecución

Instalación y ejecución de SIGA

Entrada

- El Administrador Agropecuario seleccionará "utilidades" del Menú Principal.
- Del Menú utilidades seleccionará "utilidades totales".
- A continuación se muestra una interfaz donde el debe seleccionar los siguientes datos: microproyecto, producto de ese microproyecto, por último se selecciona la opción calcular.
- El proceso de cálculo de utilidades totales se considera como finalizado.

Resultado Esperado

Se muestra en pantalla el resultado del cálculo de utilidades totales, que en este caso es la diferencia entre costos totales e ingresos totales, dado que costos totales son iguales a cero, las utilidades son iguales a los ingresos totales

Evaluación de la prueba: Satisfactoria

2.12 Historia de usuario 12: Calcular utilidades por fecha

En este caso hay que comprobar el cálculo correcto de las utilidades comprendido entre una fecha inicial y una fecha final. En este proceso se pueden presentar errores por omitir ingresar alguno de los datos necesarios, si los datos ingresados no son suficientes, se genera un mensaje que se despliega en pantalla y solicita todos los datos. En el caso de que la introducción de datos sea correcta, éstos son procesados y se genera el resultado de las utilidades obtenidas entre las fechas ingresadas.

Entrada

- El Administrador Agropecuario seleccionará "Utilidades" del Menú Principal.
- Del Menú utilidades seleccionará "Utilidades por fecha".
- A continuación se muestra una interfaz donde el debe ingresar los siguientes datos:

Datos solicitados	Datos Validos
Seleccionar nombre del proyecto	De la interfaz "Utilidades por fecha" seleccionar uno de los proyectos ya ingresados
Seleccionar Nombre del producto	De la interfaz "Utilidades por fecha" seleccionar uno de los productos ya ingresados
fecha inicial de semana	De la interfaz "Utilidades" por fecha, seleccionar fecha
Fecha final de semana	De la interfaz "utilidades" por fecha, seleccionar fecha

Tabla 2.9: Datos de entrada historia de usuario 12

2.12.1 Cálculo correcto de utilidades por fecha

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.9 y posteriormente se selecciona la opción calcular. El proceso cálculo de utilidades por fecha se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se muestra en pantalla el resultado de las utilidades para el período comprendido entre las fechas ingresadas.

Evaluación de la Prueba: Prueba satisfactoria

2.12.2 Cálculo de Utilidades por fecha, cuando ingreso de datos es incompleto

Durante el proceso de ingreso de datos, se omite el ingreso de uno o más datos de la tabla 2.9 y posteriormente se selecciona la opción calcular. El proceso de cálculo de utilidades por fecha se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se muestra en pantalla un mensaje donde se solicita al usuario el ingreso de todos los datos.

Evaluación de la Prueba: Prueba satisfactoria

2.13 Historia de usuario 13: Introducción de fichas técnicas agrícolas

Descripción

En este caso hay que comprobar la introducción correcta de los datos correspondientes a las fichas técnicas agrícolas. Una ficha técnica agrícola consta de múltiples datos y puede ocurrir que algunos sean válidos mientras que otros no. En el proceso se guardan sólo aquellas fichas técnicas que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y el proyecto no será guardado y si los datos son incompletos se le solicita al cliente que introduzca los datos completos. En el caso de que la introducción de datos sea correcta, éstos son almacenados en la base de datos.

Entradas

- El Administrador Agropecuario seleccionará "Fichas Técnicas" del Menú Principal.
- Del Menú Fichas Técnicas seleccionará "Agrícolas".
- A continuación se muestra una interfaz donde el debe ingresar los siguientes datos:

Datos solicitados	Datos validos
Cultivo	Tipo texto
Fecha de siembra	Del menú fichas técnicas agrícolas seleccionar la fecha
Fecha cosecha	Del menú fichas técnicas agrícolas seleccionar la fecha
Unidades sembradas	Entero (está predeterminado por las unidades sembradas que fueron ingresadas en proyectos estándar)
Cantidad cosechada (valor)	Valor flotante
Cantidad cosechada (Unidad)	Del menú fichas técnicas agrícolas

Tabla 2.10: Datos de entrada historia de usuario 13

2.13.1 Ingreso correcto de fichas técnicas agrícolas

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.10 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica agrícola se considera como finalizado.

Resultado Esperado

La ficha técnica agrícola es almacenada en la base de datos y el usuario podrá visualizarlo mediante la lista desplegada en la ventana y, haciendo click en éste, podrá visualizar los datos correspondientes a dicha ficha técnica.

Evaluación de la Prueba: Prueba satisfactoria

2.13.2 Introducción de fichas técnicas agrícolas con datos erróneos

Durante el proceso de ingreso de datos, se introducen datos distintos a los indicados en la tabla 2.10 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica agrícola se considera como erróneo.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Internamente se procesan los datos y se encuentran datos no válidos por el sistema en alguno de los campos, en ese caso, se le muestra al usuario un mensaje de error que le informa que ocurrió un error en su ingreso de datos.

Evaluación de la Prueba: Prueba satisfactoria

2.13.3 Ingreso incompleto de datos en fichas técnicas agrícolas

Durante el proceso de ingreso de datos, se omiten datos de la tabla 2.10 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica agrícola se considera como incompleto. Aquí debe tenerse en cuenta que los datos de entrada "fecha de cosecha" y "cantidad cosechada" solo se conocen cuando el evento de la cosecha ocurre, por lo tanto, SIGA acepta como válido el ingreso de fichas técnicas agrícolas, con o sin el ingreso de dichos datos. Cuando ocurra la cosecha el usuario podrá actualizar la ficha técnica ingresando solo esos datos.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Internamente se procesan los datos y se encuentran que no se han ingresado todos los datos, se mostrará un mensaje que solicita al usuario introducir todos los datos.

Evaluación de la Prueba: Prueba satisfactoria

2.14 Historia de usuario 14: Ingreso de datos ficha técnica porcinos, etapa lactancia

Descripción

En este caso hay que comprobar la introducción correcta de los datos correspondientes a las fichas técnicas de porcinos en la etapa de lactancia, que consta de múltiples datos y puede ocurrir que algunos sean válidos mientras que otros no. En el proceso se guardan sólo aquellas fichas técnicas que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y el proyecto no será guardado y si los datos son incompletos se le solicita al cliente que introduzca los datos completos.

Entrada

- El Administrador Agropecuario seleccionará "Fichas Técnicas" del Menú Principal.
- Del Menú Fichas Técnicas seleccionará "Pecuarios".
- Del Menú Pecuarios seleccionará "Porcinos"
- Del Menú Porcinos seleccionará "Lactancia"
- A continuación se muestra una interfaz donde el debe ingresar los siguientes datos:

Datos solicitados	Datos válidos
Nombre de la Madres	texto
Raza	texto
Fecha inicial	De la interfaz ficha técnica etapa lactancia selecciona la fecha del calendario
Fecha final	De la interfaz ficha técnica etapa lactancia selecciona la fecha del calendario
Lechones nacidos	entero
Lechones al final de lactancia	entero
Peso promedio Inicial (valor)	Valor flotante
Peso promedio Inicial (unidad)	De la interfaz ficha técnica etapa lactancia, selecciona una de las opciones (kilos, libras, gramos)
Peso promedio final (valor)	entero
Peso promedio final (unidad)	De la interfaz ficha técnica etapa lactancia, selecciona una de las opciones (kilos, libras, gramos)

Tabla 2.11: Datos de entrada historia de usuario 14

2.14.1 Ingreso de datos correctos en fichas técnicas porcinos etapa lactancia

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.11 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica porcinos etapa lactancia se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica es almacenada en la base de datos y el usuario podrá visualizarla mediante la lista desplegada en la ventana y, haciendo click en éste, podrá visualizar los datos correspondientes a dicha ficha técnica.

Evaluación de la Prueba: Prueba satisfactoria

2.14.2 Introducción de fichas técnicas porcinos etapa lactancia con datos erróneos

Durante el proceso de ingreso de datos, se introducen datos distintos a los indicados en la tabla 2.11 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica porcinos etapa lactancia se considera como erróneo.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Internamente se procesan los datos y se encuentran datos no válidos por el sistema en alguno de los campos, en ese caso, la ficha técnica no es guardada en la base de datos

Evaluación de la Prueba: Prueba satisfactoria

2.14.3 Ingreso incompleto de datos en fichas técnicas porcinos etapa lactancia

Durante el proceso de ingreso de datos, se omiten datos de la tabla 2.11 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica porcinos etapa lactancia se considera como incompleto.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Internamente se procesan los datos y se encuentran que no se han ingresado todos los datos, se mostrará un mensaje que solicita al usuario introducir todos los datos.

Evaluación de la Prueba: Prueba satisfactoria

2.14.4 Eliminación ficha técnica porcinos etapa lactancia

Condiciones de ejecución

Instalación y ejecución de SIGA

Entrada

- El Administrador Agropecuario seleccionará "Fichas Técnicas" del Menú Principal.
- Del Menú Fichas Técnicas seleccionará "Pecuarios".
- Del Menú Pecuarios seleccionará "Porcinos"
- Del Menú Porcinos seleccionará "Lactancia"
- De la interfaz ficha técnica porcinos etapa lactancia selecciona uno de los proyectos almacenados. A continuación selecciona la opción borrar

Resultado esperado

La ficha técnica de porcinos etapa lactancia seleccionada es eliminada de la base de datos.

Evaluación de la prueba: Prueba satisfactoria

2.15 Historia de usuario 15: Ingreso y almacenamiento de datos fichas técnicas porcinos, otras etapas

Descripción

En este caso hay que comprobar la introducción correcta de los datos correspondientes a las fichas técnicas de porcinos en otras etapas (distintas a lactancia), que consta de múltiples datos y puede ocurrir que algunos sean válidos mientras que otros no. En el proceso se guardan sólo aquellas fichas técnicas que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y el proyecto no será guardado y si los datos son incompletos se le solicita al cliente que introduzca los datos completos.

Entrada

-El Administrador Agropecuario seleccionará "Proyectos" del menú principal

-Del menú "Proyectos" seleccionará "Proyectos Especiales"

En la interfaz proyectos especiales seleccionará del listado el animal al que se le desea realizar la ficha técnica y posteriormente seleccionará la opción Fichas técnicas.

-A continuación se muestra una interfaz donde el debe ingresar los siguientes datos:

Datos solicitados	Datos válidos
Etapas	Debe seleccionarla de la interfaz: Ceba, levante, precebo, gestación
Raza	texto
Fecha inicial	De la interfaz ficha técnica porcinos otras etapas selecciona la fecha del calendario
Fecha final	De la interfaz ficha técnica porcinos otras etapas selecciona la fecha del calendario
Animales iniciales	entero
Animales finales	entero
Peso promedio total inicial	Valor flotante, y seleccionar unidad (gramos, libras, kilos)
Peso promedio final	Valor flotante, y seleccionar unidad (gramos, libras, kilos)

Tabla 2.12: Datos de entrada historia de usuario 15

2.15.1 Ingreso de datos correctos en fichas técnicas porcinos otras etapas

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.12 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica porcinos otras etapas se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica es almacenada en la base de datos y el usuario podrá visualizarla mediante la lista desplegada en la ventana y, haciendo click en éste, podrá visualizar los datos correspondientes a dicha ficha técnica.

Evaluación de la Prueba: Prueba satisfactoria

2.15.2 Ingreso de datos erróneos en fichas técnicas porcinos otras etapas

Durante el proceso de ingreso de datos, se introducen datos distintos a los indicados en la tabla 2.12 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica porcinos otras etapas se considera como erróneo.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Al detectarse datos no válidos la ficha técnica no es almacenada en la base de datos.

Evaluación de la Prueba: Prueba satisfactoria

2.15.3 Ingreso de datos incompletos en fichas técnicas porcinos otras etapas

Durante el proceso de ingreso de datos, se omiten datos de la tabla 2.12 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica porcinos otras etapas se considera como incompleto.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Desplegar un mensaje en pantalla en el que se solicita ingresar todos los datos.

Evaluación de la Prueba: Prueba satisfactoria

2.15.4 Eliminación ficha técnica porcinos otras etapas

Condiciones de ejecución

Instalación y ejecución de SIGA

Entrada

- El Administrador Agropecuario seleccionará "fichas técnicas" del menú principal
- Del menú "fichas técnicas" seleccionará "Pecuarios"
- Del menú pecuario seleccionará "porcinos"
- Del menú "porcinos" seleccionará otras etapas

En la interfaz ficha técnica otras etapas seleccionará uno de los elementos de la lista, y posteriormente la opción borrar.

Resultado esperado

El elemento seleccionado de la ficha técnica otras etapas seleccionado es eliminado de la base de datos.

Evaluación de la prueba: Prueba satisfactoria

2.16 Historia de usuario 16: Introducción y almacenamiento de datos fichas técnicas vacas lecheras

Descripción

En este caso hay que comprobar la introducción correcta de los datos correspondientes a las fichas técnicas de vacas lecheras, que consta de múltiples datos y puede ocurrir que algunos sean válidos mientras que otros no. En el proceso se guardan sólo aquellas fichas técnicas que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y el proyecto no será guardado y si los datos son incompletos se le solicita al cliente que introduzca los datos completos.

Entrada

-El Administrador Agropecuario seleccionará "Proyectos" del menú principal

-Del menú "Proyectos" seleccionará "Proyectos Especiales"

En la interfaz proyectos especiales seleccionará del listado a la vaca lechera a la que se le desea realizar la ficha técnica y posteriormente seleccionará la opción Fichas técnicas.

-A continuación se muestra una interfaz donde selecciona una de las vacas del listado, en dicha interfaz deberá ingresar los siguientes datos:

Datos solicitados	Datos Válidos
Nombre del animal	De la interfaz ficha técnica vacas lecheras selecciona el animal
Raza	Texto
Fecha de inseminación	De la interfaz ficha técnica vacas lecheras selecciona la fecha del calendario
Fecha de parto	De la interfaz ficha técnica vacas lecheras selecciona la fecha del calendario
Fecha final de lactancia	De la interfaz ficha técnica vacas lecheras selecciona la fecha del calendario
Fecha posible inseminación	De la interfaz ficha técnica vacas lecheras selecciona la fecha del calendario

Tabla 2.13: Datos de entrada historia de usuario 16

2.16.1 Ingreso de datos correctos en fichas técnicas vacas lecheras

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.13 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica vacas lecheras se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica es almacenada en la base de datos y el usuario podrá visualizarla mediante la lista desplegada en la ventana y, haciendo click en éste, podrá visualizar los datos correspondientes a dicha ficha técnica.

Evaluación de la Prueba: Prueba satisfactoria

2.16.2 Ingreso de datos erróneos en fichas técnicas vacas lecheras

Durante el proceso de ingreso de datos, se introducen datos distintos a los indicados en la tabla 2.13 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica vacas lecheras se considera como erróneo.

2.16.3 Ingreso de fichas técnicas vacas lecheras con datos incompletos

Durante el proceso de ingreso de datos, al omitirse el ingreso de los datos “nombre del animal” o “raza”, el proceso de guardar la ficha técnica se considera como incompleto y la ficha técnica no es guardada. Los datos de fecha de inseminación, fecha de parto, fecha final de lactancia y fecha posible próxima inseminación se actualizarán en la medida en que dichos eventos ocurran, por tanto se consideran válidos los ingresos, con o sin esos datos.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se despliega un mensaje en pantalla que solicita al usuario introducir los datos completos

Evaluación de la Prueba: Prueba satisfactoria

2.16.3 Eliminación ficha técnica vacas lecheras

Condiciones de ejecución

Instalación y ejecución de SIGA

Entrada

El Administrador Agropecuario seleccionará “Proyectos” del menú principal

-Del menú “Proyectos” seleccionará “Proyectos Especiales”

En la interfaz proyectos especiales seleccionará del listado el animal al que se le desea realizar la ficha técnica y posteriormente seleccionará la opción Fichas técnicas.

En la interfaz fichas técnicas vacas lecheras seleccionará el nombre del animal del que desea borrar los datos.

Resultado esperado

Los datos correspondientes al animal seleccionado son eliminados de la base de datos.

Evaluación de la prueba: Prueba satisfactoria

2.17 Historia de usuario 17: Ingreso y almacenamiento de registro diario vacas lecheras

Descripción

En este caso hay que comprobar la introducción correcta de los datos correspondientes al registro diario de vacas lecheras, que consta de múltiples datos y puede ocurrir que algunos sean válidos mientras que otros no. En el proceso se guardan sólo aquellas fichas técnicas que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y el proyecto no será guardado y si los datos son incompletos se le solicita al cliente que introduzca los datos completos.

Entrada

-El Administrador Agropecuario seleccionará “Proyectos” del menú principal

-Del menú “Proyectos” seleccionará “Proyectos Especiales”

En la interfaz proyectos especiales seleccionará del listado el animal al que se le desea realizar la ficha técnica y posteriormente seleccionará la opción Fichas técnicas.

- A continuación se muestra la interfaz ficha técnica vacas lecheras, de donde selecciona el animal al que se le desea hacer el registro diario.
- En la interfaz de ficha técnica vacas lecheras seleccionará la opción "Por día"
- A continuación se muestra una interfaz donde el debe ingresar los siguientes datos:

Datos solicitados	Datos válidos
Fecha	De la interfaz ficha técnica diaria vacas lecheras selecciona la fecha del calendario
Producción de leche	Introducir cantidad (valor flotante) y seleccionar en la interfaz la unidad (Litros, botellas o cm. Cúbicos)
Tipo de alimento consumido Concentrado	Introducir cantidad (valor flotante) y seleccionar en la interfaz la unidad (gramos, libras o kilos)
Tipo de alimento consumido -Melaza	Introducir cantidad (valor flotante) y seleccionar en la interfaz la unidad (gramos, libras o kilos)
Tipo de alimento consumido -Sal	Introducir cantidad (valor flotante) y seleccionar en la interfaz la unidad (gramos, libras o kilos)

Tabla 2.14: Datos de entrada historia de usuario 17

2.17.1 Ingreso de datos correctos en registro diario de vacas lecheras

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.14 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica vacas lecheras registro diario se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica es almacenada en la base de datos y el usuario podrá visualizarla.

Evaluación de la Prueba: Prueba satisfactoria

2.17.2 Ingreso de datos erróneos en registro diario de vacas lecheras

Durante el proceso de ingreso de datos, se introducen datos distintos a los indicados en la tabla 2.14 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica vacas lecheras registro diario se considera como erróneo.

Condiciones de ejecución

Instalación y ejecución de SIGA

- Resultado Esperado

La ficha técnica no es almacenada en la base de datos

Evaluación de la Prueba: Prueba satisfactoria

2.17.3 Ingreso de datos incompletos en registro diario de vacas lecheras

Durante el proceso de ingreso de datos, se omiten datos de la tabla 2.14 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica vacas lecheras registro diario se considera como incompleto.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se despliega un mensaje donde se solicita al cliente ingresar todos los datos

Evaluación de la Prueba: Prueba satisfactoria

2.18 Historia de usuario 18: Introducción y almacenamiento de datos fichas técnicas terneros

Descripción:

En este caso hay que comprobar la introducción correcta de los datos correspondiente a las fichas técnicas terneros, que consta de múltiples datos y puede ocurrir que algunos sean válidos mientras que otros no. En el proceso se guardan sólo aquellas fichas técnicas que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y la ficha técnica no será guardada y si los datos son incompletos se le solicita al cliente que introduzca los datos completos.

Entrada

-El Administrador Agropecuario seleccionará "Proyectos" del menú principal

-Del menú "Proyectos" seleccionará "Proyectos Especiales"

En la interfaz proyectos especiales seleccionará del listado el ternero al que se le desea realizar la ficha técnica y posteriormente seleccionará la opción Fichas técnicas.

-A continuación se muestra la interfaz ficha técnica terneros, de donde selecciona el ternero al que se le desea crear la ficha técnica, en esta interfaz debe ingresar los siguientes datos:

Datos solicitados	Datos Válidos
Nombre del animal	De la interfaz ficha técnica terneros selecciona uno de los animales.
Raza	Texto
Fecha de nacimiento	De la interfaz ficha técnica terneros selecciona la fecha del calendario
Peso al nacer	Introducir cantidad (tipo entero) y seleccionar de la interfaz la unidad (gramos, libras o kilos)
Fecha de destete	De la interfaz ficha técnica terneros selecciona la fecha del calendario
Peso de salida	Introducir cantidad (tipo entero) y seleccionar de la interfaz la unidad (gramos, libras o kilos)

Tabla 2.15: Datos de entrada historia de usuario 18

2.18.1 Ingreso de datos correctos en fichas técnicas terneros

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.15 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica terneros se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica es almacenada en la base de datos y el usuario podrá visualizarla mediante la lista desplegada en la ventana y, haciendo click en éste, podrá visualizar los datos correspondientes a dicha ficha técnica.

Evaluación de la Prueba: Prueba satisfactoria

2.18.2 Ingreso de fichas técnicas terneros con datos erróneos

Durante el proceso de ingreso de datos, se introducen datos distintos a los indicados en la tabla 2.15 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica terneros se considera como erróneo.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica no es almacenada en la base de datos

Evaluación de la Prueba: Prueba satisfactoria

2.18.3 Ingreso de fichas técnicas terneros con datos incompletos

Durante el proceso de ingreso de datos, se omiten datos de la tabla 2.15 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica terneros se considera como incompleto. Debe aclararse que el peso de salida es un dato que se conoce cuando el ternero sale, por ello, se acepta el ingreso y almacenamiento de fichas técnicas sin dicho dato, que podrá ser actualizado cuando dicho evento ocurre.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se despliega un mensaje en pantalla que solicita al usuario introducir los datos completos

Evaluación de la Prueba: Prueba satisfactoria

2.18.4 Eliminación de fichas técnicas terneros

- El Administrador Agropecuario seleccionará "Proyectos" del menú principal
- Del menú "Proyectos" seleccionará "Proyectos Especiales"

En la interfaz proyectos especiales seleccionará del listado el ternero al que se le desea realizar la ficha técnica y posteriormente seleccionará la opción Fichas técnicas.

-A continuación se muestra la interfaz ficha técnica terneros, de donde selecciona el ternero al que le desea eliminar los datos ingresados.

-Selecciona la opción borrar

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Los datos pertenecientes al elemento seleccionado son borrados de la base de datos

Evaluación de la Prueba: Prueba satisfactoria

2.19 Historia de usuario19: Introducción y almacenamiento de datos fichas técnicas aves de postura

Descripción:

En este caso hay que comprobar la introducción correcta de los datos correspondiente a las fichas técnicas aves de postura, que consta de múltiples datos y puede ocurrir que algunos sean válidos mientras que otros no. En el proceso se guardan sólo aquellas fichas técnicas que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y la ficha técnica no será guardada y si los datos son incompletos se le solicita al cliente que introduzca los datos completos.

Entrada

-El Administrador Agropecuario seleccionará "Fichas Técnicas" del Menú Principal.

-Del Menú Fichas Técnicas seleccionará "Pecuarios".

-Del Menú Pecuarios seleccionará "aves de postura"

A continuación se muestra una interfaz donde el debe ingresar los siguientes datos:

Dato solicitado	Dato válido
Numero de galpón	entero
Raza	texto
Número de aves inicio	entero
Número de aves final	entero

Tabla 2.16: Datos de entrada historia de usuario 19

2.19.1 Ingreso de datos correctos en fichas técnicas aves de postura

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.16 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica aves de postura se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica es almacenada en la base de datos y el usuario podrá visualizarla mediante la lista desplegada en la interfaz.

Evaluación de la Prueba: Prueba satisfactoria

2.19.2 Ingreso de fichas técnicas aves de postura con datos erróneos

Durante el proceso de ingreso de datos, se introducen datos distintos a los indicados en la tabla 2.16 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica aves de postura se considera como erróneo.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica no es almacenada en la base de datos

Evaluación de la Prueba: Prueba satisfactoria

2.19.3 Ingreso de fichas técnicas aves de postura con datos incompletos

Durante el proceso de ingreso de datos, se omiten datos de la tabla 2.16 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica aves de postura se considera como incompleto. El dato número de aves al final, es una excepción, pues se conoce al final del período productivo de las aves, y por ello el ingreso de fichas técnicas que no contienen dicho dato es aceptado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se despliega un mensaje en pantalla que solicita al usuario introducir los datos completos

Evaluación de la Prueba: Prueba satisfactoria

2.19.4 Eliminación fichas técnicas aves postura

Entrada

- El Administrador Agropecuario seleccionará "Fichas Técnicas" del Menú Principal.
- Del Menú Fichas Técnicas seleccionará "Pecuarios".
- Del Menú Pecuarios seleccionará "aves de postura"
- De la interfaz ficha técnica aves de postura selecciona uno de los elementos almacenados
- Selecciona la opción borrar.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

El elemento seleccionado es eliminado de la base de datos, y es borrado de la interfaz de ficha técnica aves de postura

Evaluación de la Prueba: Prueba satisfactoria

2.20 Historia de usuario 20: Introducción y almacenamiento Registro semanal aves de postura

En este caso hay que comprobar la introducción correcta de los datos correspondiente al registro semanal de aves de postura, que consta de múltiples datos y puede ocurrir que algunos sean válidos mientras que otros no. En el proceso se guardan sólo aquellos registros que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y la ficha técnica no será guardada y si los datos son incompletos se le solicita al usuario que introduzca los datos completos.

Entrada

- El Administrador Agropecuario seleccionará "Fichas Técnicas" del Menú Principal.
- Del Menú Fichas Técnicas seleccionará "Pecuarios".
- Del Menú Pecuarios seleccionará "aves de postura"
- De la ficha técnica "Aves de postura" seleccionará uno de los galpones existentes y posteriormente la opción "por semana".

A continuación se muestra una interfaz donde el debe ingresar los siguientes datos:

Dato solicitado	Dato válido
Semana desde	Seleccionar una fecha del calendario de la interfaz ficha semanal aves de postura
Semana Hasta	Seleccionar una fecha del calendario de la interfaz ficha semanal aves de postura
Alimento consumido	Ingresar cantidad (valor flotante) y seleccionar la unidad (gramos, libras o kilos).
Número de aves descartadas vendidas	Entero
Número de aves descartadas no aptas	Entero
Número de huevos puestos tipo C	Entero
Número de huevos puestos tipo B	Entero
Número de huevos puestos tipo A	Entero
Número de huevos puestos tipo Súper:	Entero
Número de huevos puestos tipo Extra	Entero
Número de huevos puestos tipo Jumbo	Entero

Tabla 2.17: Datos de entrada historia de usuario 20

2.20.1 Ingreso de datos correctos en fichas técnicas aves de postura registro semanal.

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.17 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica aves de postura registro semanal se considera como finalizado

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

El registro semanal es almacenado en la base de dato.

Evaluación de la Prueba: Prueba satisfactoria

2.20.2 Ingreso de datos erróneos en registro semanal aves de postura

Durante el proceso de ingreso de datos, se introducen datos distintos a los indicados en la tabla 2.17 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica aves de postura registro semanal se considera como erróneo.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

El registro semanal no es almacenado en la base de datos

Evaluación de la Prueba: Prueba satisfactoria

2.20.3 Ingreso de datos incompletos en registro semanal de aves de postura

Durante el proceso de ingreso de datos, se omiten datos de la tabla 2.17 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica aves de postura registro semanal se considera como incompleto. Debe tenerse en cuenta el número de huevos puestos (tipo A, B, C, súper, extra y jumbo), puede ser cero para algunas de las clasificaciones, por ello, también es aceptado el ingreso de registros semanales que no ingresen el número de huevos puestos para todos los tipos, SIGA asume que si uno de esos datos no es ingresado, el número de huevos puestos es cero.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se despliega un mensaje que solicita al usuario ingresar todos los datos

Evaluación de la Prueba: Prueba satisfactoria

2.21 Historia de usuario 21: Introducción y almacenamiento de datos fichas técnicas proyecto apiarios

Descripción:

En este caso hay que comprobar la introducción y almacenamiento correcto de los datos correspondientes a las fichas técnicas proyecto apiarios, que consta de múltiples datos y

puede ocurrir que algunos sean válidos mientras que otros no. En el proceso se guardan sólo aquellas fichas técnicas que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y la ficha técnica no será guardada y si los datos son incompletos se le solicita al cliente que introduzca los datos completos.

Entrada

- El Administrador Agropecuario seleccionará "Fichas Técnicas" del Menú Principal.
- Del Menú Fichas Técnicas seleccionará "Pecuarios".
- Del Menú Pecuarios seleccionará "Apiarios"

A continuación se muestra una interfaz donde el debe ingresar los siguientes datos:

Datos solicitados	Datos válidos
colmena	texto
Fecha de inicio	Seleccionar fecha del calendario de la interfaz ficha técnica apiarios
UDC	Entero
Fecha cosecha de miel	Seleccionar una fecha del calendario de la interfaz ficha técnica apiarios
Fecha cosecha polen	Seleccionar una fecha del calendario de la interfaz ficha técnica apiarios
Fecha cosecha otros	Seleccionar una fecha del calendario de la interfaz ficha técnica apiarios
Cantidad cosecha miel	Debe ingresar valor flotante y seleccionar unidad de la interfaz ficha técnica apiarios (gramo, libra, cm. cúbico, litros)
Cantidad cosecha polen	Debe ingresar valor flotante y seleccionar unidad de la interfaz ficha técnica apiarios (gramo, libra, cm. cúbico, litros)
Cantidad cosecha polen	Debe ingresar valor flotante y seleccionar unidad de la interfaz ficha técnica apiarios (gramo, libra, cm. cúbico, litros)

Tabla 2.18: Datos de entrada fichas técnicas proyecto apiarios

2.21.1 Introducción y almacenamiento correcto de fichas técnicas proyecto apiarios

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.18 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica apiarios se considera como finalizado.

Resultado Esperado

La ficha técnica es almacenada en la base de datos

Evaluación de la prueba: Prueba satisfactoria

2.21.2 Introducción y almacenamiento erróneo de fichas técnicas proyecto apiarios

Durante el proceso de ingreso de datos, se introducen datos distintos a los indicados en la tabla 2.18 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica apiarios se considera como erróneo.

Resultado Esperado

Se produce un mensaje de error y la ficha técnica no es guardada en la base de datos, el proceso de ingreso de fichas técnicas de proyecto apiarios se considera erróneo.

Evaluación de la prueba: Prueba satisfactoria

2.21.3 Introducción y almacenamiento de fichas técnicas proyecto apiarios, con datos incompletos

Se consideran datos necesarios para ingresar la ficha técnica: Colmena, fecha de inicio y UDC, los otros datos de fechas y cantidades cosechadas se llenan y actualizan en el transcurso del proyecto. Por ello se considera como incompleto el caso en que falte alguno de los tres primeros datos.

Resultado Esperado

La ficha técnica no es guardada y se despliega un mensaje que solicita se ingresen todos los datos.

Evaluación de la prueba: Prueba satisfactoria.

2.21.4. Eliminación de fichas proyecto apiarios

Entrada

- El Administrador Agropecuario seleccionará "Fichas Técnicas" del Menú Principal.
- Del Menú Fichas Técnicas seleccionará "Pecuarios".
- Del Menú Pecuarios seleccionará "Apiarios"
- De la interfaz ficha técnica apiarios seleccionar uno de los proyectos ya almacenados.
- Seleccionar la opción borrar.

El proceso de eliminación de fichas técnicas apiarios se considera como finalizado

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

El elemento seleccionado es eliminado de la base de datos, y es borrado de la interfaz de ficha técnica apiarios

Evaluación de la Prueba: Prueba satisfactoria

2.22 Historia de usuario 22. Introducción y almacenamiento de datos fichas técnicas proyecto apiarios/labores semanales

Descripción:

En este caso hay que comprobar la introducción y almacenamiento correcto de los datos correspondientes a las labores semanales de proyecto apiarios, que consta de múltiples datos y puede ocurrir que algunos sean válidos mientras que otros no. En el proceso se guardan sólo aquellas fichas técnicas que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y la ficha técnica no será guardada y si los datos son incompletos se le solicita al cliente que introduzca los datos completos.

Entrada

- El Administrador Agropecuario seleccionará "Fichas Técnicas" del Menú Principal.
- Del Menú Fichas Técnicas seleccionará "Pecuarios".
- Del Menú Pecuarios seleccionará "Apiarios"
- De la interfaz de proyecto apiarios selecciona uno de los proyectos almacenados (al que pertenece la labor a almacenar), posteriormente seleccionará la opción labores semanales.

A continuación se muestra una interfaz donde el debe ingresar los siguientes datos:

Datos solicitados	Datos válidos
Fecha de inicio de semana	De la interfaz de fichas técnicas labores semanales apiarios seleccionar la fecha del calendario
Fecha final de semana	De la interfaz de fichas técnicas labores semanales apiarios seleccionar la fecha del calendario
Labores Realizadas	Modo texto

Tabla 2.19 Datos entrada historia de usuario 22

2.22.1 Introducción y almacenamiento correcto de fichas técnicas proyecto apiarios/labores semanales

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.19 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica apiarios/labores semanales se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica es almacenada en la base de datos

Evaluación de la prueba: Prueba satisfactoria

2.22.2 Introducción y almacenamiento de fichas técnicas proyecto apiarios/labores semanales, con datos incompletos

Durante el proceso de ingreso de datos, se omiten datos de la tabla 2.19 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica aves de postura se considera como incompleto.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica no es guardada y se despliega un mensaje que solicita se ingresen todos los datos.

valuación de la prueba: Prueba satisfactoria

2.23 Historia de Usuario 23: Introducción y almacenamiento de datos ficha técnica hembras de cría otras especies

Descripción:

En este caso hay que comprobar la introducción y almacenamiento correcto de los datos correspondientes a las fichas técnicas del proyecto hembras de cría otras especies (se consideran otras especies las distintas de bovinos, porcinos, aves de postura y apiarios), que consta de múltiples datos y puede ocurrir que algunos sean válidos mientras que otros no. En el proceso se guardan sólo aquellas fichas técnicas que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y la ficha técnica no será guardada y si los datos son incompletos se le solicita al cliente que introduzca todos los datos.

Entrada

- El Administrador Agropecuario seleccionará "Fichas Técnicas" del Menú Principal.
- Del Menú Fichas Técnicas seleccionará "Pecuarios".
- Del Menú Pecuarios seleccionará "Otros"
- Del Menú Otros seleccionará "Hembras de Cría".

A continuación se muestra una interfaz donde el debe ingresar los siguientes datos:

Dato	Dato válido
Proyecto	Tipo texto
Hembras de cría	Tipo entero
Fecha de partos	De la interfaz ficha técnica hembras de cría otras especies seleccionar fecha.

Tabla 2.20 Datos de entrada historia de usuario 20

2.23.1 Introducción y almacenamiento correcto de fichas técnicas hembras de cría otras especies

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.20 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica hembras de cría otras especies se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado: La ficha técnica es almacenada en la base de datos

Evaluación de la prueba: prueba satisfactoria

2.23.2 Introducción y almacenamiento erróneo de fichas técnicas hembras de cría otras especies

Durante el proceso de ingreso de datos, se introducen datos distintos a los indicados en la tabla 2.20 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica hembras de cría otras especies se considera como erróneo.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se produce un mensaje de error y la ficha técnica no es guardada en la base de datos

Evaluación de la prueba: Prueba satisfactoria

2.23.3 Introducción y almacenamiento de fichas técnicas hembras de cría otras especies con datos incompletos.

Durante el proceso de ingreso de datos, se omiten datos de la tabla 2.20 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica hembras de cría otras especies se considera como incompleto. La fecha de partos puede estar vacía y el proceso se considera completo.

Condiciones de ejecución: Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica no es guardada y se despliega un mensaje de solicita se ingresen todos los datos.

Evaluación de la prueba: Prueba satisfactoria.

2.23.4 Eliminación fichas técnicas hembras de cría otras especies

Entrada

- El Administrador Agropecuario seleccionará "Fichas Técnicas" del Menú Principal.
- Del Menú Fichas Técnicas seleccionará "Pecuarios".
- Del Menú Pecuarios seleccionará "Otros"
- Del Menú Otros seleccionará "Hembras de Cría"
- De la interfaz ficha técnica hembras de cría otras especies selecciona el proyecto almacenado que desea eliminar
- Selecciona la opción borrar

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

El elemento seleccionado es eliminado de la base de datos, y es borrado de la interfaz de ficha técnica hembras de cría otras especies

Evaluación de la prueba: Prueba satisfactoria.

2.24 Historia de Usuario 24: Introducción y almacenamiento de datos fichas técnicas hembras de cría otras especies/labores semanales

Descripción:

En este caso hay que comprobar la introducción y almacenamiento correcto de los datos correspondientes a las fichas técnicas del proyecto hembras de cría otras especies/labores semanales (se consideran otras especies las distintas de bovinos, porcinos, aves de postura y apiarios), que consta de múltiples datos y puede ocurrir que algunos sean válidos mientras que otros no. En el proceso se guardan sólo aquellas fichas técnicas que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y la ficha técnica no será guardada y si los datos son incompletos se le solicita al cliente que introduzca los datos completos.

Entrada

-El Administrador Agropecuario seleccionará "Fichas Técnicas" del Menú Principal.

-Del Menú Fichas Técnicas seleccionará "Pecuarios".

-Del Menú Pecuarios seleccionará "Otros"

Del Menú Otros seleccionará "Hembras de Cría".

De la interfaz hembras de cría otras especies seleccionará uno de los proyectos ya almacenados que estará visible en un listado, posteriormente seleccionará la opción por semana.

A continuación se le presenta una interfaz donde el debe ingresar los siguientes datos:

Datos de entrada	Datos válidos
Fecha de inicio de semana	De la interfaz ficha técnica hembras de cría otras especies, seleccionar la fecha del calendario
Fecha final de semana	De la interfaz ficha técnica hembras de cría otras especies, seleccionar la fecha del calendario
Labores Realizadas	Modo texto

Tabla 2.21 Datos entrada historia de usuario 24

2.24.1 Introducción y almacenamiento correcto de fichas hembras de cría otras especies/labores semanales

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.21 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica hembras de cría otras especies/labores semanales se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado: La ficha técnica es almacenada en la base de datos

Evaluación de la prueba: Prueba satisfactoria

2.24.2 Introducción y almacenamiento de fichas técnicas proyecto hembras de cría otras especies/labores semanales con datos incompletos

Durante el proceso de ingreso de datos, se omiten datos de la tabla 2.21 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica hembras de cría otras especies/labores semanales se considera como incompleto.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica no es guardada y se despliega un mensaje que solicita se ingresen todos los datos.

Evaluación de la prueba: Prueba satisfactoria.

2.25 Historia de Usuario 25: Introducción y almacenamiento de datos fichas técnicas crías otras especies

Descripción:

En este caso hay que comprobar la introducción y almacenamiento correcto de los datos correspondientes a las fichas técnicas crías otras especies (se consideran otras especies las distintas de bovinos, porcinos, aves de postura y apiarios), que consta de múltiples datos y puede ocurrir que algunos sean válidos mientras que otros no. En el proceso se guardan sólo aquellas fichas técnicas que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y la ficha técnica no será guardada y si los datos son incompletos se le solicita al cliente que introduzca los datos completos.

Entrada

- El Administrador Agropecuario seleccionará "Fichas Técnicas" del Menú Principal.
- Del Menú Fichas Técnicas seleccionará "Pecuarios".
- Del Menú Pecuarios seleccionará "Otros"
- Del Menú Otros seleccionará "crías".

A continuación se muestra una interfaz donde el debe ingresar los siguientes datos:

Datos solicitados	Datos validos
Proyecto	Tipo texto
Total animales	Tipo entero
Numero de gazapos	Tipo entero
Numero de cebados	Tipo entero
Fecha cebada	De la interfaz ficha técnica crías otras especies seleccionar fecha del calendario

Tabla 2.22 Datos de entrada historia de usuario 25

2.25.1 Introducción y almacenamiento correcto de fichas técnicas crías otras especies.

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.22 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica crías otras especies se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado: La ficha técnica es almacenada en la base de datos.

Evaluación de la prueba: Prueba satisfactoria

2.25.2 Introducción y almacenamiento erróneo de fichas técnicas crías otras especies

Durante el proceso de ingreso de datos, se introducen datos distintos a los indicados en la tabla 2.22 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica crías otras especies se considera como erróneo.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se produce un mensaje de error y la ficha técnica no es guardada en la base de datos.

Evaluación de la prueba: Prueba satisfactoria

2.25.3 Introducción y almacenamiento de fichas técnicas crías otras especies con datos incompletos

Durante el proceso de ingreso de datos, se omiten datos de la tabla 2.22 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica crías otras especies se considera como incompleto. El proceso de guardar esta ficha técnica se considera completo cuando faltan los datos de número de gazapos, número de cebados, fecha cebada, puesto que estos datos se conocen si existen gazapos, cebados.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica no es guardada y se despliega un mensaje de error que solicita se ingresen todos los datos

Evaluación de la prueba: Prueba satisfactoria

2.25.4 Eliminación de fichas técnicas crías otras especies

Entradas

- El Administrador Agropecuario seleccionará "Fichas Técnicas" del Menú Principal.
- Del Menú Fichas Técnicas seleccionará "Pecuarios".
- Del Menú Pecuarios seleccionará "Otros"
- Del Menú Otros seleccionará "crías".

A continuación se muestra la interfaz de ficha técnica crías otras especies, de donde el selecciona del listado el proyecto de crías que desee eliminar. Posteriormente selecciona la opción borrar.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

El elemento seleccionado es eliminado de la base de datos, y es borrado de la interfaz de ficha técnica crías otras especies

Evaluación de la prueba: Prueba satisfactoria

2.26 Historia de usuario 26: Introducción y almacenamiento de datos fichas técnicas crías otras especies/labores semanales

Descripción:

En este caso hay que comprobar la introducción y almacenamiento correcto de los datos correspondientes a las fichas técnicas crías otras especies/labores semanales (se consideran otras especies las distintas de bovinos, porcinos, aves de postura y apiarios), que consta de múltiples datos y puede ocurrir que algunos sean válidos mientras que otros no. En el proceso se guardan sólo aquellas fichas técnicas que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y la ficha técnica no será guardada y si los datos son incompletos se le solicita al cliente que introduzca los datos completos.

Entrada

- El Administrador Agropecuario seleccionará "Fichas Técnicas" del Menú Principal.
- Del Menú Fichas Técnicas seleccionará "Pecuarios".
- Del Menú Pecuarios seleccionará "Otros"
- Del Menú Otros seleccionará "crías".
- En la interfaz para ingreso de fichas técnicas de crías otras especies, seleccionará el proyecto al que pertenece la labor a registrar, posteriormente seleccionará la opción x semana

A continuación se despliega una interfaz en la que debe ingresar los siguientes datos:

Datos solicitados	Datos válidos
Fecha de inicio de semana	De la interfaz de ficha técnica de crías otras especies/labores semanales seleccionar la fecha del calendario.
Fecha final de semana	De la interfaz de ficha técnica de crías otras especies/labores semanales seleccionar la fecha del calendario.
Labores Realizadas	Modo texto

Tabla 2.23 Datos de entrada historia de usuario 26

2.26.1 Introducción y almacenamiento correcto de fichas técnicas crías otras especies/labores semanales

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.23 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica crías otras especies/labores semanales se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica es almacenada en la base de datos.

Evaluación de la prueba: Prueba satisfactoria

2.26.2 Introducción y almacenamiento de fichas técnicas crías otras especies/labores semanales con datos incompletos

Durante el proceso de ingreso de datos, se omite el ingreso de alguno de los datos indicados en la tabla 2.23 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica crías otras especies/labores semanales se considera como incompleto.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado: La ficha técnica no es guardada y se despliega un mensaje que solicita se ingresen todos los datos.

Evaluación de la prueba: Prueba satisfactoria

2.27 Historia de usuario 27: Introducción/eliminación y almacenamiento de datos inventario bienes e insumos

Descripción:

En este caso hay que comprobar la introducción y almacenamiento correcto de los datos correspondientes al inventario de bienes e insumos, que consta de múltiples datos y puede ocurrir que algunos sean válidos mientras que otros no. En el proceso se guardan sólo aquellas fichas técnicas que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y la ficha técnica no será guardada y si los datos son incompletos se le solicita al cliente que introduzca los datos completos.

Entrada

-El Administrador Agropecuario seleccionará "Inventarios" del Menú Principal.

-Del Menú Inventarios seleccionará "Bienes e insumos".

A continuación se muestra una interfaz donde el debe ingresar los siguientes datos:

Datos solicitados	Datos validos
Artículo	Tipo texto
UDC	Tipo entero

Fecha de ingreso	De la interfaz inventario bienes e insumos seleccionar la fecha del calendario
Cantidad	Tipo entero
Estado	Tipo texto

Tabla 2.24 Datos de entrada inventario bienes e insumos

2.27.1 Introducción y almacenamiento correcto de inventario bienes e insumos

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.24 y posteriormente se selecciona la opción guardar. El proceso de guardar el registro del bien o insumo se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Los datos correspondientes al bien o insumo son almacenados en la base de datos.

Evaluación de la prueba: Prueba satisfactoria

2.27.2 Introducción y almacenamiento erróneo de inventario bienes e insumos

Durante el proceso de ingreso de datos, se introduce uno o más datos distintos a los indicados en la tabla 2.24 y posteriormente se selecciona la opción guardar. El proceso de guardar el registro del bien o insumo se considera como erróneo.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se produce un mensaje de error y el registro de inventario del bien o insumo no es guardado en la base de datos.

Evaluación de la prueba: Prueba satisfactoria

2.27.3 Introducción y almacenamiento de inventario bienes e insumos con datos incompletos

Durante el proceso de ingreso de datos, se omiten datos de la tabla 2.24 y posteriormente se selecciona la opción guardar. El proceso de guardar el registro de inventario bienes e insumos se considera como incompleto.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica no es guardada y se despliega un mensaje de error que solicita se ingresen todos los datos. El proceso de ingreso de inventario bienes e insumos se considera como incompleto

Evaluación de la prueba: Prueba satisfactoria.

2.27.4 Eliminación de inventario bienes e insumos

Entrada

-El Administrador Agropecuario seleccionará "Inventarios" del Menú Principal.

-Del Menú Inventarios seleccionará "Bienes e insumos".

A continuación se muestra una interfaz donde el selecciona el bien o insumo a eliminar y posteriormente la escogerá la opción borrar:

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado esperado

El bien o insumo seleccionado de la interfaz inventario bienes o insumos es eliminado de la base de datos.

Evaluación de la prueba: Prueba satisfactoria

2.28 Historia de usuario 28: Introducción y almacenamiento de datos inventario agropecuario

Descripción:

En este caso hay que comprobar la introducción y almacenamiento correcto de los datos correspondientes al inventario agropecuario, que consta de múltiples datos y puede ocurrir que algunos sean válidos mientras que otros no. En el proceso se guardan sólo aquellas fichas técnicas que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y la ficha técnica no será guardada y si los datos son incompletos se le solicita al cliente que introduzca los datos completos.

Entrada

-El Administrador Agropecuario seleccionará "Inventarios" del Menú Principal.

-Del Menú Inventarios seleccionará "Agropecuario".

A continuación se muestra una interfaz donde el debe ingresar los siguientes datos:

Datos solicitados	Datos validos
Artículo	Tipo texto
UDC	Tipo entero
Fecha ingreso	De la interfaz de inventario agropecuario seleccionar fecha del calendario
Edad	De la interfaz de inventario agropecuario un Valor flotante y seleccionar unidad (días, mes o año)
Cantidad	entero

Estado	texto
--------	-------

Tabla 2.25 Datos de entrada historia de usuario 28

2.28.1 Introducción y almacenamiento correcto inventario agropecuario

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.25 y posteriormente se selecciona la opción guardar. El proceso de guardar el registro del inventario agropecuario se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica es almacenada en la base de datos,

Evaluación de la prueba: Prueba satisfactoria

2.28.2 Introducción y almacenamiento erróneo de inventario agropecuario

Durante el proceso de ingreso de datos, se introduce uno o más datos distintos a los indicados en la tabla 2.25 y posteriormente se selecciona la opción guardar. El proceso de guardar el inventario agropecuario se considera como erróneo.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Se produce un mensaje de error y la ficha técnica no es guardada en la base de datos

Evaluación de la prueba: Prueba satisfactoria

2.28.3 Introducción y almacenamiento de inventario agropecuario con datos incompletos.

Durante el proceso de ingreso de datos, se omiten datos de la tabla 2.25 y posteriormente se selecciona la opción guardar. El proceso de guardar el inventario agropecuario se considera como incompleto.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica no es guardada y se despliega un mensaje de error que solicita se ingresen todos los datos.

Evaluación de la prueba: Prueba satisfactoria.

2.28.4 Eliminación de inventario agropecuario

Entrada:

- El Administrador Agropecuario seleccionará "Inventarios" del Menú Principal.
- Del Menú Inventarios seleccionará "Agropecuario".
- En el listado de interfaz de inventario agropecuario seleccionará uno de los artículos almacenados
- selecciona borrar

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

El artículo será borrado y es eliminado de la base de datos

Evaluación de la prueba: Prueba satisfactoria

2.29 Historia de usuario 29: introducción y almacenamiento de datos fichas técnicas porcinos, etapa lactancia por fechas

Descripción

En este caso hay que comprobar la introducción correcta de los datos correspondientes a las fichas técnicas de porcinos en la etapa de lactancia para un período comprendido desde y hasta una fecha que se solicitan al usuario. En el proceso se guardan sólo aquellas fichas técnicas que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y el proyecto no será guardado y si los datos son incompletos se le solicita al cliente que introduzca los datos completos.

Entrada

- El Administrador Agropecuario seleccionará "Fichas Técnicas" del Menú Principal.
- Del Menú Fichas Técnicas seleccionará "Pecuarios".
- Del Menú Pecuarios seleccionará "Porcinos"
- Del Menú Porcinos seleccionará "Lactancia"
- A continuación se muestra una interfaz donde el usuario seleccionará uno de los proyectos de porcinos etapa lactancia y posteriormente la opción "por fecha", en donde ingresará los siguientes datos.

Datos solicitados	Datos válidos
Fecha desde	De la interfaz ficha técnica porcinos etapa lactancia por fecha selecciona la fecha del calendario
Fecha hasta	De la interfaz ficha técnica porcinos etapa lactancia por fecha seleccionará la fecha del calendario
Tipo de alimento consumido	Tipo texto
Consumo total por periodo	Valor flotante

Unidades	Selecciona una de las unidades disponibles en la interfaz
----------	---

Tabla 2.26: Datos de entrada historia de usuario 29

2.29.1 Ingreso de datos correctos en fichas técnicas porcinos etapa lactancia por fechas

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.26 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica porcinos etapa lactancia por fechas se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica es almacenada en la base de datos y el usuario podrá visualizarla mediante la interfaz de ingreso de fichas técnicas porcinos etapa lactancia por fechas.

Evaluación de la Prueba: Prueba satisfactoria

2.29.2 Introducción de fichas técnicas porcinos etapa lactancia por fecha con datos erróneos

Durante el proceso de ingreso de datos, se introducen datos distintos a los indicados en la tabla 2.26 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica porcinos etapa lactancia se considera como erróneo.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Internamente se procesan los datos y se encuentran datos no válidos por el sistema en alguno de los campos, en ese caso, la ficha técnica no es guardada en la base de datos

Evaluación de la Prueba: Prueba satisfactoria

2.29.3 Ingreso incompleto de datos en fichas técnicas porcinos etapa lactancia

Durante el proceso de ingreso de datos, se omiten datos de la tabla 2.26 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica porcinos etapa lactancia se considera como incompleto.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Internamente se procesan los datos y se encuentran que no se han ingresado todos los datos, se mostrará un mensaje que solicita al usuario introducir todos los datos.

Evaluación de la Prueba: Prueba satisfactoria

2.30: Historia de usuario 30: Introducción y almacenamiento de datos fichas técnicas porcinos, otras etapas por fechas

Descripción

En este caso hay que comprobar la introducción correcta de los datos correspondientes a las fichas técnicas de porcinos otras etapas para un período comprendido desde y hasta una fecha que se solicitan al usuario. En el proceso se guardan sólo aquellas fichas técnicas que ingresen los datos correctos. Si los datos ingresados no son válidos, se genera un mensaje de error y el proyecto no será guardado y si los datos son incompletos se le solicita al cliente que introduzca los datos completos.

Entrada

- El Administrador Agropecuario seleccionará "Fichas Técnicas" del Menú Principal.
- Del Menú Fichas Técnicas seleccionará "Pecuarios".
- Del Menú Pecuarios seleccionará "Porcinos"
- Del Menú Porcinos seleccionará "Otras etapas"
- A continuación se muestra una interfaz donde el usuario seleccionará uno de los proyectos de porcinos otras etapas y posteriormente la opción "por fecha", en donde ingresará los siguientes datos:

Datos solicitados	Datos válidos
Fecha desde	De la interfaz ficha técnica porcinos etapa lactancia por fecha selecciona la fecha del calendario
Fecha hasta	De la interfaz ficha técnica porcinos etapa lactancia por fecha seleccionará la fecha del calendario
Tipo de alimento consumido	Tipo texto
Consumo total por periodo	Valor flotante
Unidades	Selecciona una de las unidades disponibles en la interfaz

Tabla 2.27: Datos de entrada historia de usuario 29

2.30.1 Ingreso de datos correctos en fichas técnicas porcinos otras etapas por fechas

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.27 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica porcinos etapa lactancia por fechas se considera como finalizado.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

La ficha técnica es almacenada en la base de datos y el usuario podrá visualizarla mediante la interfaz de ingreso de fichas técnicas porcinos otras etapas por fechas.

Evaluación de la Prueba: Prueba satisfactoria

2.30.2 Introducción de fichas técnicas porcinos otras etapas por fecha con datos erróneos

Durante el proceso de ingreso de datos, se introducen datos distintos a los indicados en la tabla 2.27 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica porcinos otras etapas por fecha se considera como erróneo.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Internamente se procesan los datos y se encuentran datos no válidos por el sistema en alguno de los campos, en ese caso, la ficha técnica no es guardada en la base de datos

Evaluación de la Prueba: Prueba satisfactoria

2.30 Ingreso incompleto de datos en fichas técnicas porcinos otras etapas por fecha

Durante el proceso de ingreso de datos, se omiten datos de la tabla 2.27 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica porcinos etapa lactancia se considera como incompleto.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Internamente se procesan los datos y se encuentran que no se han ingresado todos los datos, se mostrará un mensaje que solicita al usuario introducir todos los datos.

Evaluación de la Prueba: Prueba satisfactoria

2.31 Historia de usuario 31: Agregar calendario

El proceso de ingreso de datos, implica en muchos casos introducir fechas, en versiones preliminares este se hacía ingresando los datos manualmente, lo cual implicaba mayor probabilidad de error por parte de quien ingrese los datos, que debía ser conocedor del formato java de las fechas e ingresarlas correctamente. Para evitar que se ingresen fechas incorrectas se agregó esta historia de usuario que permite el ingreso de la fecha seleccionándola de la interfaz.

Previamente en las tablas de entrada de datos ya se ha dicho que interfaces cuentan con esta opción

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado esperado

Se mostrará un calendario en las interfases o formularios que solicitan el ingreso de fechas.

Evaluación de la prueba: Prueba satisfactoria

2.32 Historia de usuario 32: Introducción de fichas técnicas agrícolas por fechas

Descripción

En este caso hay que comprobar la introducción correcta de los datos correspondientes a las labores realizadas en fichas técnicas agrícolas para un período comprendido entre una fecha inicial y una fecha final. Para realizar este proceso satisfactoriamente es necesario seguir una serie de pasos e ingresar correctamente los datos solicitados. Sólo son almacenadas en la base de datos las fichas técnicas que realicen todo el procedimiento correcto e ingresen los datos válidos. Si los datos ingresados son incompletos se le solicita al cliente que introduzca los datos completos.

Entradas

- El Administrador Agropecuario seleccionará "Fichas Técnicas" del Menú Principal.
- Del Menú Fichas Técnicas seleccionará "Agrícolas".
- Del Menú agrícolas seleccionará uno de los proyectos y posteriormente la opción por fecha. Posteriormente se mostrará la interfaz de ingreso y almacenamiento de datos para fichas técnicas agrícolas por fechas en donde el debe llenar los siguiente datos:

Datos solicitados	Datos validos
Fecha desde	En el calendrio de la interfaz fichas técnicas agrícolas por fechas seleccionar la fecha
Fecha hasta	En el calendario de la interfaz fichas técnicas agrícolas por fecha seleccionar la fecha
Labores realizadas	Modo texto, el usuario describe la labor realizada para el periodo comprendido entre las fechas ingresadas

Tabla 2.28: Datos de entrada historia de usuario 32

2.32.1 Ingreso correcto de fichas técnicas agrícolas por fecha

Durante el proceso de ingreso de datos, se introducen los datos según lo indica la tabla 2.28 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica agrícola se considera como finalizado.

Resultado Esperado

La ficha técnica agrícola por fechas es almacenada en la base de datos y el usuario podrá visualizarlo desde la interfaz de ingreso de fichas técnicas agrícola por fechas

Evaluación de la Prueba: Prueba satisfactoria

2.32.2 Ingreso incompleto de datos en fichas técnicas agrícolas por fechas

Durante el proceso de ingreso de datos, se omiten datos de la tabla 2.32 y posteriormente se selecciona la opción guardar. El proceso de guardar ficha técnica agrícola por fecha se considera como incompleto.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado Esperado

Internamente se procesan los datos y se encuentran que no se han ingresado todos los datos, se mostrará un mensaje que solicita al usuario introducir todos los datos.

Evaluación de la Prueba: Prueba satisfactoria

2.33 Historia de usuario 33: Generación de reporte en formato Excel de costos individuales

Descripción:

Se generarán reportes en formato Excel de los costos individuales, que contiene el valor y una descripción del gasto. Durante el proceso pueden ocurrir errores o que no se genere el reporte esperado.

Entradas:

- Seleccionar gastos del menú principal
- Seleccionar costos individuales del menú gastos
- Seleccionar generar reporte de la interfaz de costos individuales.

Internamente se consultan las tablas de la base de datos relacionadas y se procesan los datos y se generará el reporte en formato Excel con los costos individuales de todos los productos hasta la fecha.

Condiciones de ejecución

Instalación y ejecución de SIGA

Resultado esperado

Se generará un informe en formato Excel con la información descrita

Evaluación de la prueba: Prueba satisfactoria

2.34 Historia de usuario 34: Generación de reporte en formato Excel de ventas diarias.

Descripción:

Se generarán reportes en formato Excel de las ventas individuales. Durante el proceso pueden ocurrir errores o que no se genere el reporte esperado.

Entradas:

- Seleccionar costos del menú principal
- Seleccionar ingresos individuales del menú ingresos

-Seleccionar generar reporte de la interfaz de ingresos individuales (ventas diarias).

Internamente se consultan las tablas de la base de datos relacionadas y se procesan los datos y se generará el reporte en formato Excel con las ventas diarias de todos los productos hasta la fecha.

Condiciones de ejecución

Instalación y ejecución de SIGA

2.35 Historia de usuario 35: Generación de reporte en formato Excel de utilidades para un producto determinado.

Descripción:

Se generarán reportes en formato Excel de las utilidades totales, junto con los ingresos y los costos totales para todos los productos registrados. Durante el proceso pueden ocurrir errores o que no se genere el reporte esperado.

Entradas:

- Seleccionar utilidades del menú principal.
- Seleccionar utilidades totales del menú utilidades.
- Seleccionar generar reporte de la interfaz de utilidades totales

Internamente se consultan las tablas de la base de datos relacionadas y se procesan los datos y se generará el reporte en formato Excel con las utilidades totales de todos los productos hasta la fecha.

Condiciones de ejecución

Instalación y ejecución de SIGA

ANEXO 3

ESTADO DE LA INDUSTRIA DEL SOFTWARE EN EL SUROCCIDENTE COLOMBIANO

Este anexo recopila la información producto de encuestas a empresas del sector de la informática, realizadas por el departamento de sistemas de la universidad del Cauca. Los resultados ampliados y completos de las encuestas se encuentran publicados en el documento: "Reporte técnico sobre el estado de la práctica del proceso de desarrollo de software en la región suroccidental colombiana Versión 1.0" del Ing. Carlos Alberto Ardila Albarracín , de donde se extrajo la información para este anexo.

1. PERFIL GENERAL DE LAS EMPRESAS.

Período de Realización de la encuesta: Abril de 2005.

Empresas encuestadas:

EMPRESA	CIUDAD
CYBEREXITO LTDA HEXUM LTDA PASSWORD S.A. SINTEVSA SITIS LTDA TOTEMS SOFTWARE UNISOFT COLOMBIA LTDA	Popayán
COLUMBIA TECNOLOGIA LTDA GEOSOFT LTDA INGENIAR SOLUCIONES I.T. SW MEDICAL SYSTEMS NEXUS SOFTWARE NUEVOS MEDIOS ONLINE DIAGNOSTIC SYSTEM S.A. VIANET	Cali
AES DATTEL LATIN BUSSINES LTDA SOLUCIONES INTEGRALES EN TECNOLOGIA DE LA INFORMACION	Pasto

1.1. Distribución por ciudades.

CIUDAD	NÚMERO DE EMPRESAS	(%)
Popayán	7	35
Cali	9	45
Pasto	4	20

1.2. Trayectoria (en años).

CIUDAD	NÚMERO DE EMPRESAS	(%)
5 años o menos	14	70
Entre 5 y 10 años	3	15
10 años o más	3	15

1.3. Volumen de Ventas (en millones de pesos).

RANGO	NÚMERO DE EMPRESAS	(%)
Menos de 50	11	55
Entre 50 y 200	5	25
Entre 200 y 500	3	15
Más de 500	1	5

1.4. Número total de empleados.

RANGO	NÚMERO DE EMPRESAS	(%)
Menos de 5	7	35
Entre 6 y 10	7	35
Entre 11 y 20	4	20
21 o más	2	10

1.5. Número de empleados dedicados al desarrollo de software.

RANGO	NÚMERO DE EMPRESAS	(%)
Menos de 5	14	70
Entre 6 y 10	4	20
Más de 10	2	10

1.6. Alcance y proyección.

ALCANCE	NÚMERO DE EMPRESAS	(%)
Nacional	14	70
Internacional	6	30

1.7. Cantidad de proyectos iniciados.

Estas empresas registran un intervalo de proyectos iniciados de entre 2 y 14. El promedio es de 6 proyectos.

1.8. Cantidad de proyectos terminados.

RANGO	NÚMERO DE EMPRESAS	(%)
Entre 0 y 25	6	30
Entre 26 y 50	2	10
Entre 51 y 75	3	15
Entre 76 y 100	9	45

1.9. Cantidad de proyectos cancelados.

RANGO	NÚMERO DE EMPRESAS	(%)
Entre 0 y 25	16	80
Entre 26 y 50	2	10
Entre 51 y 75	2	10

1.10. Esfuerzo promedio invertido en proyectos de desarrollo (personas-mes).

RANGO	NÚMERO DE EMPRESAS	(%)
Menos de 3	9	45
Entre 3 y 5	8	40
Más de 5	1	5
No responde	2	10

1.11. Procesos de desarrollo seguidos por las empresas.

PROCESO	NÚMERO DE EMPRESAS QUE LO USAN
Cascada	6
Proceso Unificado de Desarrollo	5
Espiral	5
Rational UP	4
Xprogramming	2
Proceso Propietario	1
Otro	6

Se destacan en esos otros procesos:

- CommonKADS.
- Prueba y avance.
- Proceso Orientación a Objetos.

1.12. Estrategias utilizadas para fomentar la productividad.

ESTRATEGIA	NÚMERO DE EMPRESAS QUE LA USAN
Desarrollo de software basado en componentes	10
Generación Automática de Código	5
Desarrollo orientado a familias o líneas de productos	5
Desarrollo de software basado en reglas	3
Metodologías Ágiles	2
Desarrollo Orientado a Modelos de Conocimiento	1
Ninguna	4

1.13. Tipos de software desarrollados por estas empresas.

Las diferentes empresas expresaron que se dedicaban al desarrollo de software de los siguientes tipos:

- Sector Salud y Gubernamental.
- Educativo.
- Sistemas y Servicios de Información Geográfica.
- Dispositivos Móviles.
- Seguridad.
- LMS (Learning Management Systems).
- Software para Telecomunicaciones.
- Aplicaciones WEB.
- Analisis Financiero y Gerencial.
- Control de activos y digitalización de documentos.
- Software para el Sector Solidario.

ANEXO 4: PROCESO PROGRAMACION EXTREMA DOCUMENTADO CON MODELAMIENTO AGIL V 1.0

4.1 Fase 1: Exploración

Objetivo:

Obtener los requisitos del sistema en las palabras del cliente. Éstos pueden variar en el transcurso del desarrollo.

El resultado de esta fase es un documento en el que el cliente deja por escrito sus necesidades y cómo deben suplirse. Este caso de estudio estuvo centrado en obtener un Sistema de información para Granjas Agroindustriales (en adelante lo designaremos mediante la abreviatura SIGA), el cual debe estar en capacidad de:

- Gestionar la contabilidad de gastos e ingresos de cada microproyecto en períodos mensuales.

Los proyectos bovinos y porcinos deben considerarse como proyectos especiales, esto es, la contabilidad de gastos e ingresos no se hace en su totalidad para todo el proyecto sino por cada unidad de producción (por cada porcino o por cada bovino); además, en estos proyectos la contabilidad de gastos e ingresos se lleva para todo el ciclo de producción y no para un período de un mes.

- Debe permitir agregar o eliminar proyectos según sean las disposiciones de la granja.

Debe tener una base de datos que permita la recolección de información de las ventas o los gastos realizados, de modo que se permita además de ver los resultados por cada macroproyecto, ver también los resultados de la totalidad de los macroproyectos, y al final del año ver el resultado total de la suma de los meses.

- Debe permitir imprimir los resultados de la contabilidad, esto con el fin de que sirvan como anexos a los informes que deben presentarse a la Vicerectoría administrativa de la Universidad del Cauca.

4.1.1 Flujo de trabajo fase de exploración

En los diagramas de flujo de trabajo se utilizan calles para mostrar que trabajadores ejecutan cada actividad, (representada por ruedas dentadas) situadas en el mismo campo del trabajador que la ejecuta.

Primero el cliente escribe las historias de usuario, que son entregadas al equipo de trabajo (en esta etapa aun no se han definido roles) para que explore las alternativas de solución, lo que en XP recibe el nombre de spike arquitectónico, adicionalmente se realiza la metáfora del sistema.

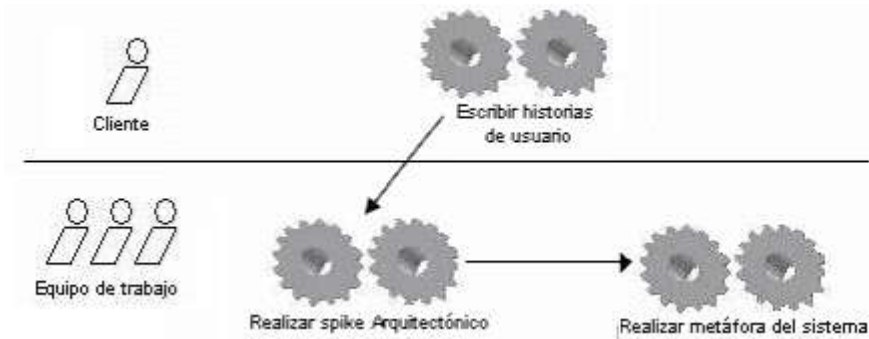


Figura 4.1: Flujo de trabajo fase de exploración

4.1.2 Actividades fase de exploración:

- **Escribir historias de usuario:** el cliente describe las características y especificaciones de aquello que necesita sea implementado. Esta actividad no tiene entradas y su resultado se representa en la figura 4.2 Responsable: cliente.

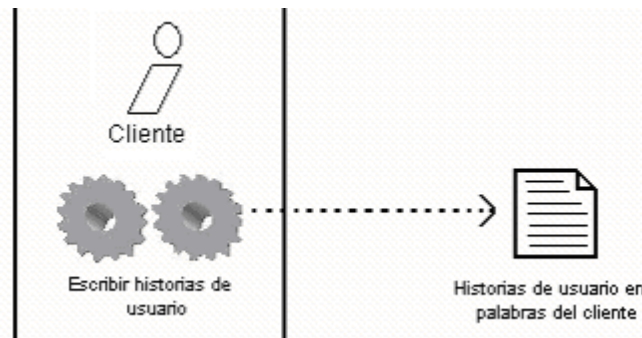


Figura 4.2: Entradas y resultado de escribir historias de usuario

- **Realizar el spike arquitectónico:** se lleva a cabo una exploración y prueba de las posibles alternativas de solución para el cliente y se selecciona la más adecuada. Esta actividad es responsabilidad del equipo de trabajo y se representa en la figura 4.3.

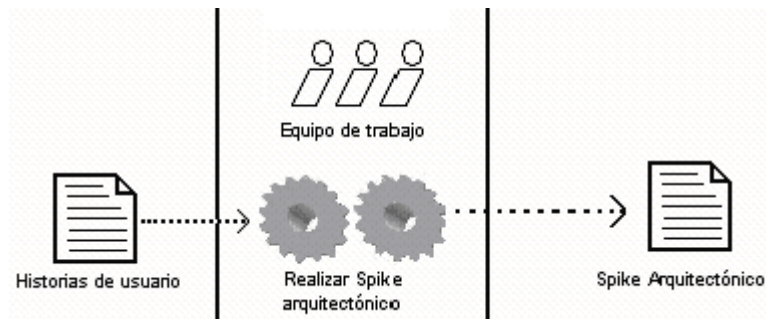


Figura 4.3: Entradas y resultados de realizar el spike arquitectónico

- **Realizar la metáfora del sistema:** la metáfora es una descripción sencilla y que todo el mundo puede contar con facilidad de cómo funciona el sistema. La aplicación

se puede definir por una o varias metáforas compartidas por el cliente y el equipo de desarrollo, consiste en formar un conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema. La figura 4.4 muestra entradas y resultado de esta actividad. Responsable: equipo de trabajo

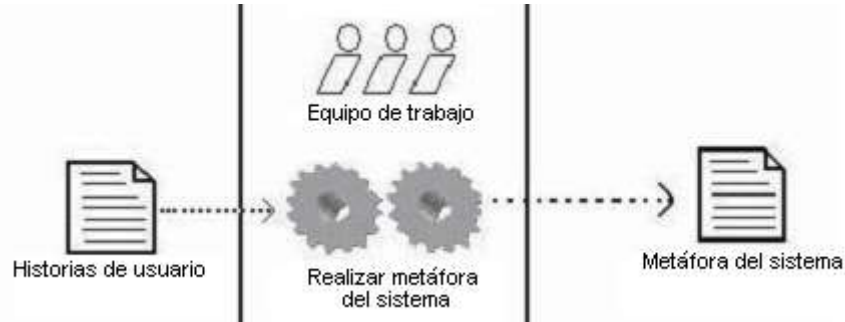


Figura 4.4: Entradas y resultado de realizar la metáfora del sistema

4.1.3 Entradas fase de exploración:

Ninguna

4.1.4 Salidas fase de exploración:

- **Historias de usuario:** Se busca obtener las necesidades del cliente (que deben ser traducidas a requerimientos del sistema) escritos por su propia mano y explicadas en sus palabras. El formato empleado para este fin se muestra en la tabla 4.1.

Historia de Usuario	
Número:	Prioridad en Negocio:
Nombre historia:	
Iteración Asignada:	Riesgo en Desarrollo: (Alto / Medio / Bajo)
Programador responsable:	
Descripción:	
Observaciones:	

Figura 4.1: Formato Historias de Usuario

- **Spike arquitectónico:** se genera un programa simple para identificar los riesgos y problemas que se pueden presentar en el desarrollo del sistema. Este programa es basado en los requerimientos del cliente tomados de las historias de usuario. Con esto se busca determinar si el sistema es viable o no. Si el sistema es viable, se desarrolla una metáfora del sistema.

- **Metáfora del sistema:** la metáfora se describió en la sección 4.1.3

4.2 Fase 2: Planeamiento

Objetivo:

Obtener un estimativo del número de iteraciones a realizar y del tiempo a emplear en cada una de ellas.

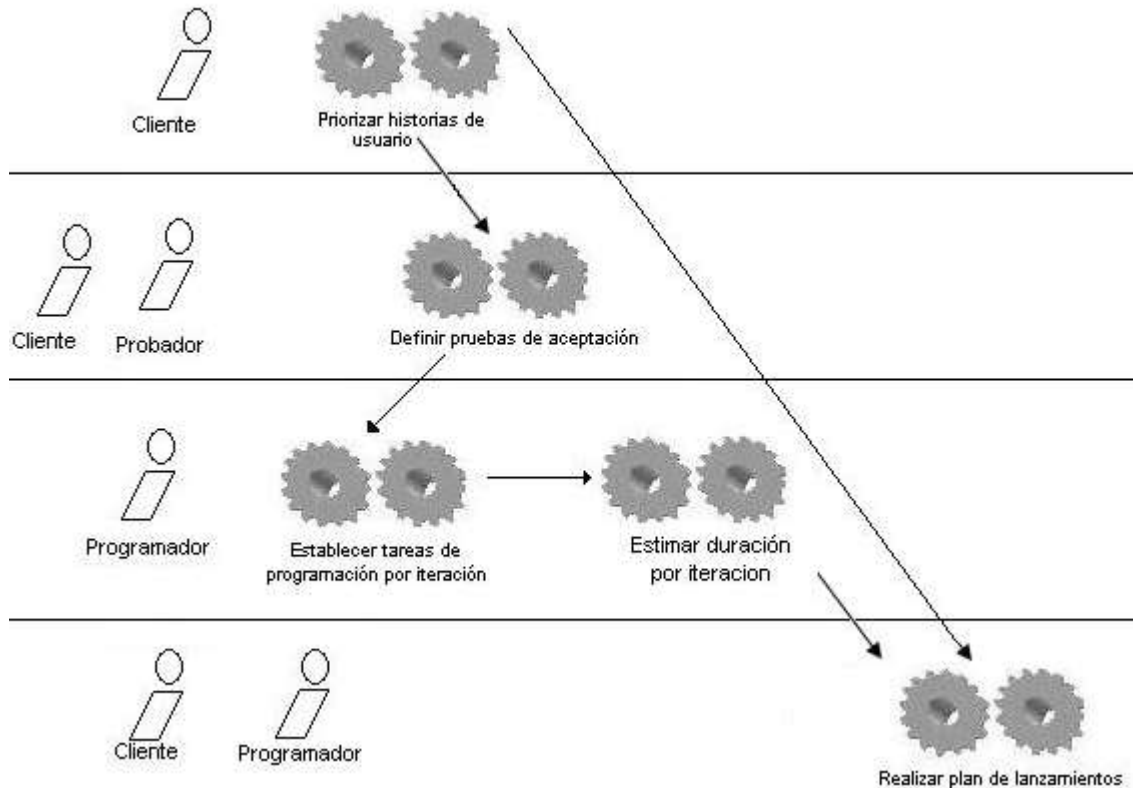


Figura 4.5: Flujo de trabajo fase de planeamiento

4.2.1 Flujo de trabajo fase de planeamiento

Inicialmente el cliente asigna prioridad a las historias de usuario. Luego se lleva a cabo una reunión de planeamiento en donde se asignan los distintos roles y responsabilidades, el probador ayuda al cliente a definir las pruebas de aceptación y los programadores se encargan de asignar las tareas de programación para cada iteración, como resultado de este proceso se obtiene un plan de lanzamientos con los estimativos para cada iteración. El flujo de trabajo se muestra en la figura 4.5.

4.2.2 Actividades fase de planeamiento

- **Priorizar (asignar prioridad a) historias de usuario:** el cliente asigna el orden y la iteración en que las historias de usuario se deben implementar. El formato de asignación de prioridades se muestra en la tabla 4.1.



Figura 4.6: Entradas y resultado de actividad asignar prioridad a historias de usuario

Asignación de prioridad a historias de usuario			
Nombre de historia	Descripción	Orden de prioridad	Iteración asignada

Tabla 4.1: Asignación de prioridad a historias de usuario

- Definir pruebas de aceptación:** Una vez el equipo de trabajo ha definido los diferentes roles en la reunión de planeamiento, y teniendo en cuenta las historias de usuario ordenadas prioritariamente, el probador ayuda al cliente en la definición de las diferentes pruebas de aceptación correspondientes a cada iteración. La figura 4.7 muestra el diagrama de la actividad.

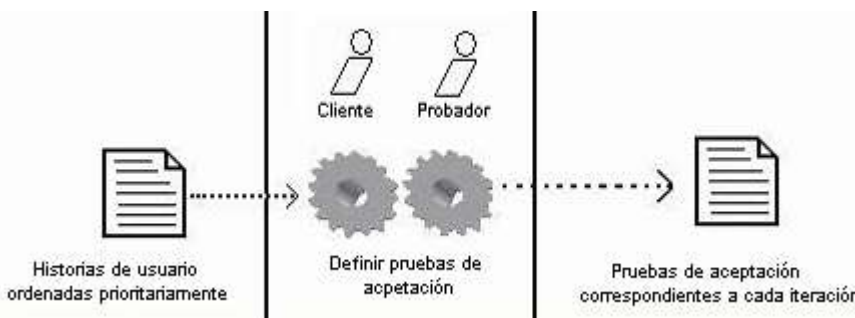


Figura 4.7: Entradas y resultados de definir pruebas de aceptación

- Establecer tareas de programación:** Una vez se tienen las pruebas de aceptación a satisfacer en cada iteración los programadores deben determinar las tareas de programación necesarias para satisfacerlas. El diagrama con las entradas y resultado de la actividad es el mostrado en la figura 4.8.

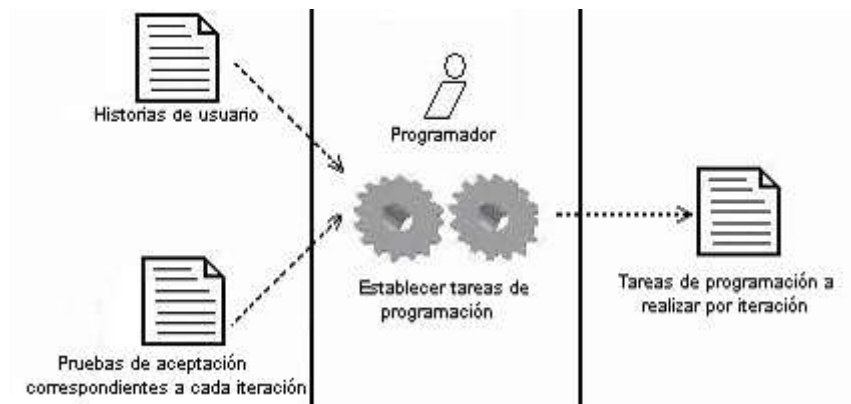


Figura 4.8: Entradas y resultados de establecer tareas de programación

- **Estimar duración por iteración:** Los programadores a partir de las tareas de programación a realizar en cada iteración estiman la duración de cada una de ellas. El diagrama con las entradas y resultado de la actividad es el mostrado en la figura 4.9.



Figura 4.9: Entradas y resultados de estimar duración por iteración

- **Realizar plan de lanzamientos:** La estimación de duración de cada una de las iteraciones es de utilidad para la realización del plan de lanzamientos, donde se definen las fechas de las posibles entregas correspondientes a cada iteración. El diagrama con las entradas y resultado de la actividad es el mostrado en la figura 4.10

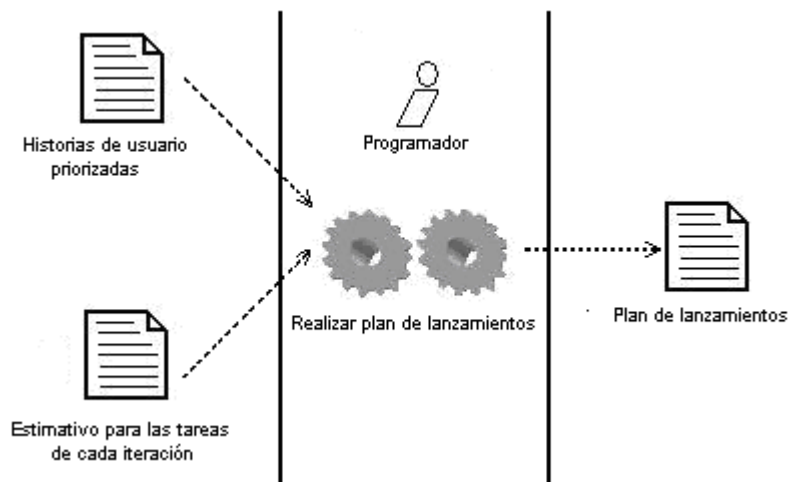


Figura 4.10: Entradas y resultados de realizar el plan de lanzamientos

Las anteriores actividades se llevan a cabo en la reunión de planeamiento donde también se asignaron los diferentes roles de la siguiente manera:

- **Desarrolladores (programadores):** los programadores escriben las pruebas y mantienen el código del programa tan simple y definido como sea posible. La pareja de programadores la conformaron Paola Andrea Manquillo y Jesús Emilio Rivera Daza.
- **Cliente:** El cliente selecciona el orden de prioridad para los requerimientos. El cliente en este caso fue el administrador de la granja agroindustrial de la Universidad del Cauca, localizada en la vereda La Rejoia.
- **Probador:** el probador o tester ayuda al cliente a escribir las pruebas funcionales. Ellos realizan pruebas funcionales regularmente, difunden resultados de la prueba y mantienen las herramientas de pruebas. En este caso las pruebas estuvieron a cargo de los programadores.
- **Coach/entrenador:** No hubo alguien que asumiera este rol, pero se puede considerar que el director de tesis (director del proyecto), asumió la guía de los otros miembros del equipo, de modo que puede decirse que él fue también el entrenador.
- **Tracker/rastreador:** Este rol no fue necesario debido a que, por diversas razones que se explicarán en detalle en el desarrollo del capítulo 5 (sección 5.3.2), no se llevó a cabo la estimación de la duración de las historias de usuario en las iteraciones.
- **Consultor:** El proyecto no contó con este rol de manera oficial, pero el ingeniero Francisco Pino, Docente del departamento de sistemas de la FIET, quien cuenta con experiencia en metodologías ágiles, colaboró en la aclaración de ciertas dudas.
- **Director:** El director de proyecto fue el director de tesis, inicialmente este rol lo asumió el Ingeniero Julio Ariel Hurtado, y en su ausencia asumió la dirección el Ingeniero Carlos Alberto Ardila Albarracín, ambos, docentes del Departamento de Sistemas de la Universidad del Cauca

4.2.3 Entradas fase de planeamiento:

- Historias de usuario

4.2.4 Salidas fase de planeamiento:

- **Plan de lanzamientos:** Es el plan de trabajo dividido en etapas, o iteraciones, una iteración corresponde a un grupo de historias de usuario a desarrollar. El plan de lanzamientos define las historias de usuario incluidas en cada iteración y la fecha de las entregas, también conocidas como lanzamientos. Cuando se cumple una iteración, ésta es revisada por el cliente quien la aprueba o sugiere modificaciones. Previamente se había convenido un plan de lanzamientos que consistía en seis iteraciones, aunque el grupo de trabajo fue conciente de la naturaleza cambiante de los requisitos y del alcance del proyecto, por ello se planeó que la última iteración sea básicamente para corregir errores o asuntos pendientes o nuevos que resulten de todo el proceso de desarrollo, esto permitió en cierta medida, que los requisitos no cambiaran ni se prolongaran de forma indefinida, sino que el cliente sabía hasta que momento podía sugerir modificaciones o agregar historias de usuario.

Plan de lanzamientos			
No. historia de usuario	Nombre historia de usuario	Iteración asignada	Fecha de terminación

Tabla4.2: Formato plan de lanzamientos

Realimentación:

- **Nueva historia de usuario:** Se toman las historias de usuario que siguen en el orden de prioridad.

4.3 Fase 3: Iteraciones

Objetivo:

En esta fase se busca desarrollar las historias de usuario prioritarias para el cliente siguiendo el plan de lanzamientos.

4.3.1 Flujo de trabajo fase de iteraciones

Se programan las historias de usuario pertenecientes a cada una de las iteraciones (una a una) según el plan de lanzamientos obtenido en la reunión de planificación de iteración y se genera una funcionalidad del sistema. A continuación, se valida el correcto desempeño de la

funcionalidad con las pruebas de aceptación; si son superadas satisfactoriamente, se repite este proceso hasta finalizar las iteraciones, en caso contrario, se corrigen los errores hasta que la funcionalidad cumpla completamente con las pruebas de aceptación, momento en el cual se repite este proceso hasta finalizar con todas las iteraciones. Responsables: desarrolladores y cliente. El flujo de trabajo se representa en la figura 4.11.

4.3.2 Actividades fase de iteraciones

- **Realizar reunión de planificación de iteración:** Al inicio de cada iteración se realiza una reunión en la que se asigna el orden en que serán solucionadas las historias de usuario pertenecientes a la iteración. El diagrama con las entradas y resultado de la actividad es el mostrado en la figura 4.12
- **Diseñar pruebas de aceptación:** Se definen las pruebas que deben cumplir las funcionalidades para que pueda considerarse que esta ha sido correctamente implementada. Esto se hace en conjunto con el cliente. El diagrama con las entradas y resultado de la actividad es el mostrado en la figura 4.13

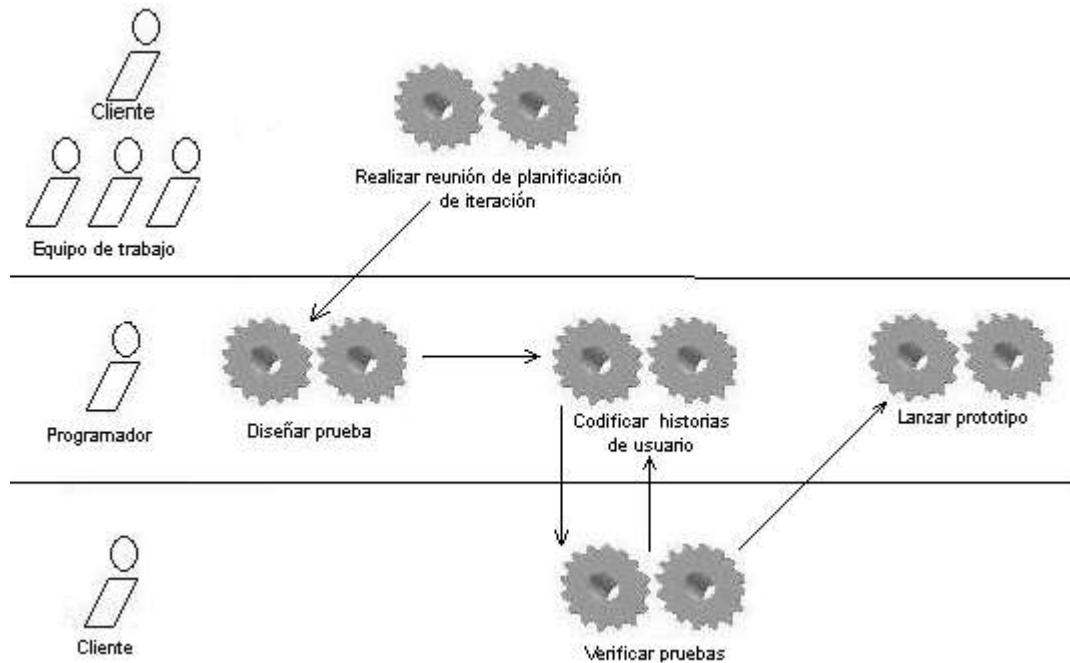


Figura 4.11: Flujo de trabajo iteración 3

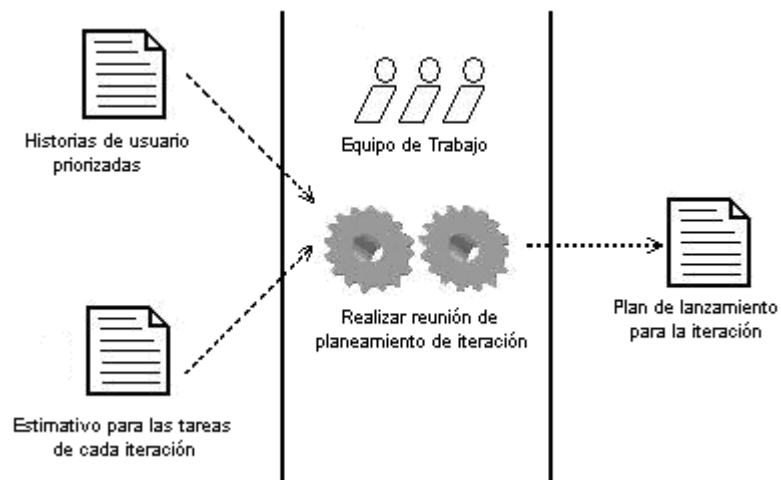


Figura 4.12: Entradas y resultado de realizar reunión de planeamiento de iteración

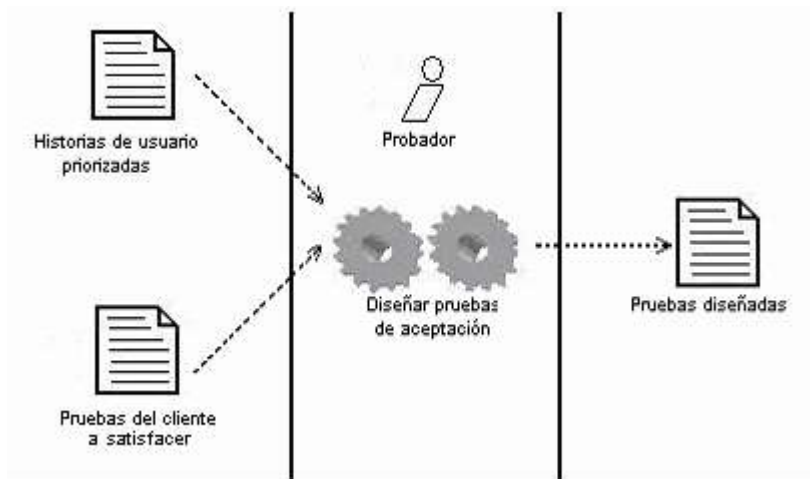


Figura 4.13: Entradas y resultado de diseñar pruebas de aceptación

- **Codificar historias de usuario:** Los programadores implementan cada una de las historias de usuario pertenecientes a la iteración, se considera que el desarrollo termina cuando todas las historias de usuario han sido programadas y además satisfacen las pruebas de aceptación. Como resultado del desarrollo se obtienen entonces nuevas funcionalidades terminadas, esto se muestra en la figura 4.14.

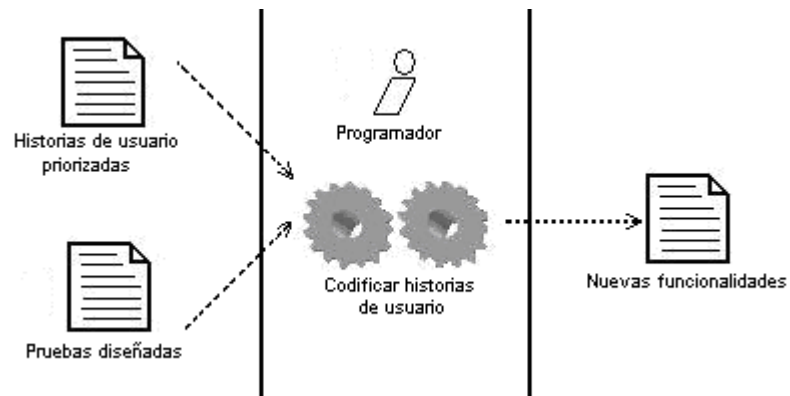


Figura 4.14: Entradas y resultado de codificar historias de usuario

- **Verificar pruebas:** Las pruebas se dan por satisfechas si una vez desarrolladas cumplen con cada uno de los casos considerado previamente para cada una de ellas. En caso de que no se certifiquen en su totalidad se retorna nuevamente el desarrollo, donde se corrige los fallos existentes hasta que no se encuentren errores. El diagrama de esta actividad se muestra en la figura 4.15.

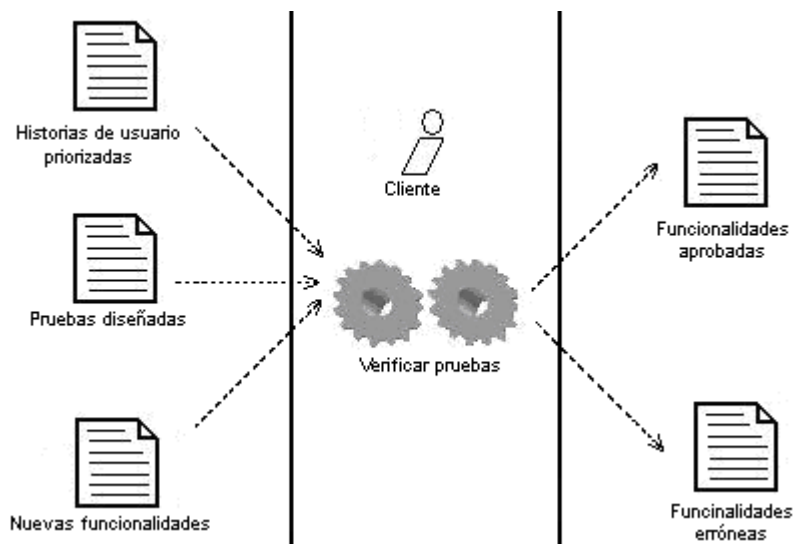


Figura 4.15: Entradas y resultado de verificar pruebas

- **Lanzar prototipo:** Una vez culminado el desarrollo y verificación de pruebas de cada iteración se obtiene un prototipo que es mostrado al cliente, quien puede sugerir cambios o ajustes a esta versión parcial del sistema a desarrollar. Las entradas y resultado de esta actividad se muestran en la figura 4.16.

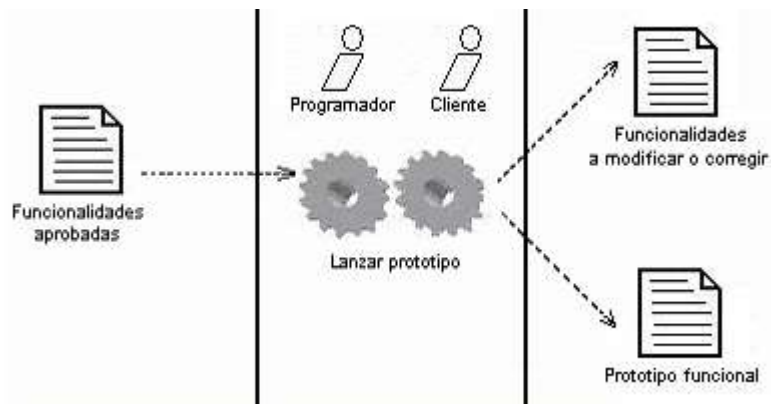


Figura 4.16: Entradas y resultado de lanzar prototipo

4.3.3 Entradas fase de iteraciones

- **Plan de lanzamientos:** Pone en ejecución el trabajo teniendo en cuenta el orden de prioridad de las historias de usuario y la duración estimada de la implementación de cada iteración. Posteriormente se ejecutan las pruebas de aceptación para cada iteración.
- **Prueba de aceptación de fallos (Entrada: Fallos):** En el caso de que las pruebas de aceptación no sean superadas de forma satisfactoria, se procede a corregir los fallos existentes y se vuelve a ejecutar las pruebas de aceptación. Este proceso se repite hasta que las pruebas de aceptación sean superadas en su totalidad.

4.3.4 Salidas fase de iteraciones:

- **Nueva funcionalidad:** Al ser superadas las pruebas de aceptación de forma satisfactoria, el sistema en general cuenta con nuevas funcionalidades o nuevas capacidades.
- **Fallos corregidos:** Cuando las pruebas de aceptación no son superadas, se procede a buscar los fallos que impiden su ejecución correcta, y, al encontrarlos, se los corrige. La fase de iteraciones termina cuando las pruebas de aceptación son aprobadas.

4.4 Fase 4: Producción

Objetivo:

Obtener un sistema adaptable y flexible, acorde a las necesidades del cliente y evaluar la metodología aplicada.

4.4.1 Flujo de trabajo fase de producción

La figura 4.17 muestra el flujo de las actividades de la fase de producción y sus responsables.

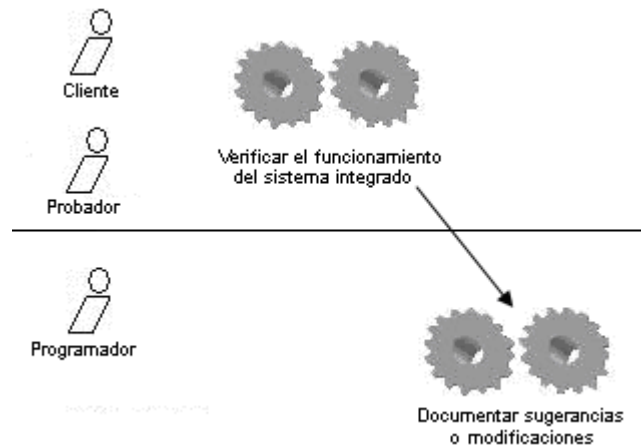


Figura 4.17: Flujo de trabajo fase de producción

4.4.2 Actividades fase de producción

- **Verificar funcionamiento del sistema integrado:** se comprueba el cumplimiento de las pruebas funcionales para el sistema al final de las iteraciones planeadas. Las entradas y el resultado de la actividad se muestran en la figura 4.18.
- **Documentar sugerencias o modificaciones:** (en caso de que éstas existan), se documentan las indicaciones del cliente para que puedan ser tenidas en cuenta en posteriores versiones del sistema. Responsable: desarrolladores. La actividad es mostrada en la figura 4.19.

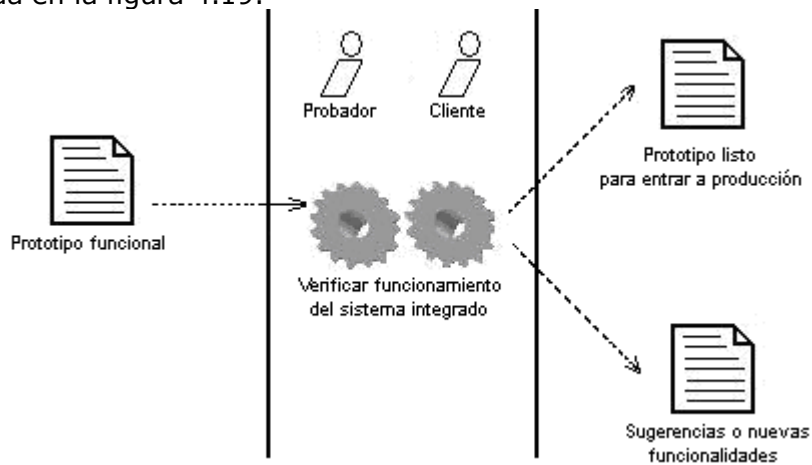


Figura 4.18: Entradas y resultado de verificar el funcionamiento del sistema integrado

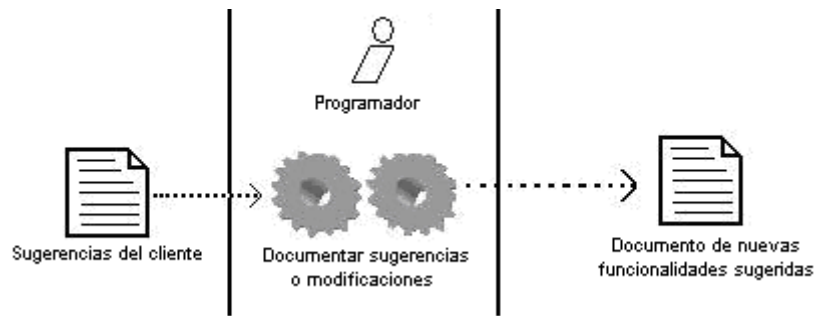


Figura 4.19: Entradas y resultado de documentar sugerencias o modificaciones

4.4.3 Entradas fase de producción:

- **Prototipo funcional:** Cuando las pruebas de aceptación son aprobadas, se verifica el correcto funcionamiento de la totalidad del sistema (el ensamblaje de las funcionalidades obtenidas) hasta que esté listo para ser lanzado al cliente.

4.4.4 Salidas fase de producción:

- **Producto liberado:** Se entrega al cliente el producto software en funcionamiento, que en este instante debe satisfacer la totalidad de especificaciones.
- **Nuevas funcionalidades:** Las sugerencias finales del cliente se documentan para ser tenidas en cuenta en la fase de mantenimiento y muerte.

4.5 Fase 5: Mantenimiento y Muerte

Objetivos:

- Capacitar al usuario en la instalación y manejo del software desarrollado.
- Satisfacer las nuevas funcionalidades aceptadas del prototipo funcional.
- Realizar un informe sobre la experiencia con el modelo seguido, al igual que documentar los manuales para el usuario.

4.5.1 Flujo de Trabajo fase de mantenimiento y muerte

En esta fase el equipo de trabajo se apartó un poco de los lineamientos propuestos por los autores que sugieren una fase de mantenimiento y otra de muerte. Un factor que determinó algunas modificaciones en las fases finales es que este proyecto se desarrolló como trabajo de grado, sujeto a un límite de tiempo, por tal motivo se acordó con el cliente que el recibiría un prototipo funcional del sistema y tendría una fecha límite para sugerir ajustes. Paralelamente se realizó la documentación pertinente, estas labores se llevaron a cabo simultáneamente, por tanto, se decidió unir las como una sola fase de mantenimiento y muerte. El flujo de trabajo para esta fase se muestra en la figura 4.20.

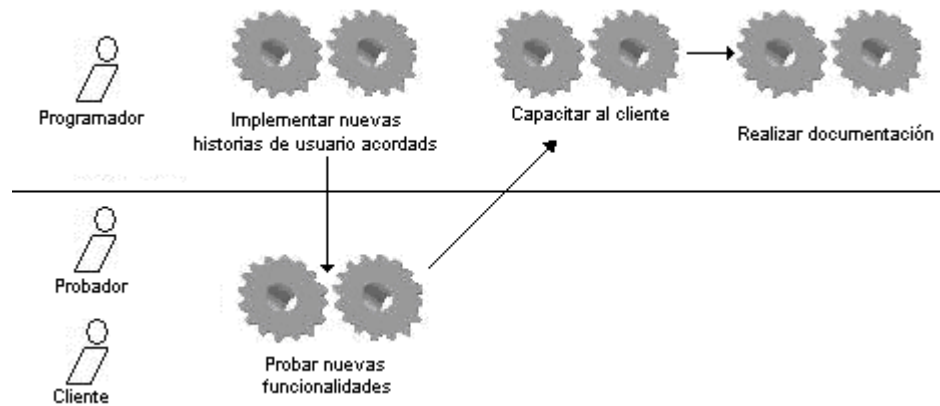


Figura 4.20: Flujo de trabajo fase de mantenimiento y muerte

4.5.2 Actividades fase de mantenimiento y muerte:

- **Implementar nuevas historias de usuario:** se codifican las nuevas historias de usuario sugeridas por el cliente que hayan sido aceptadas. Las entradas y resultado de la actividad se muestran en la figura 4.21

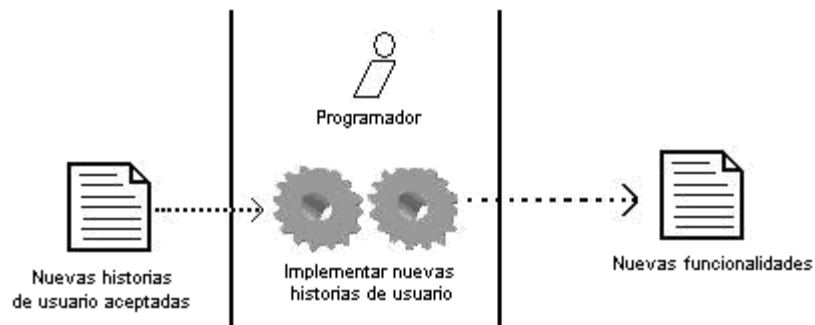
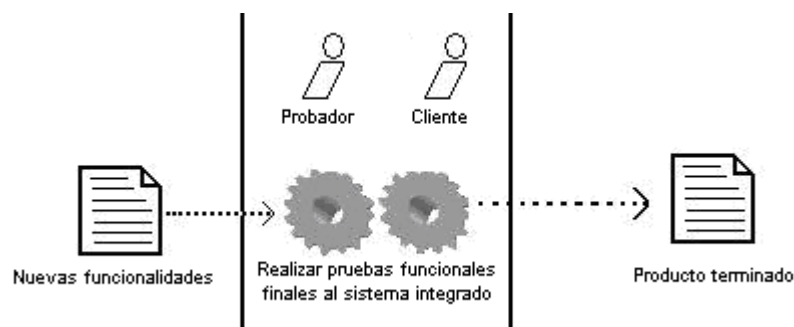


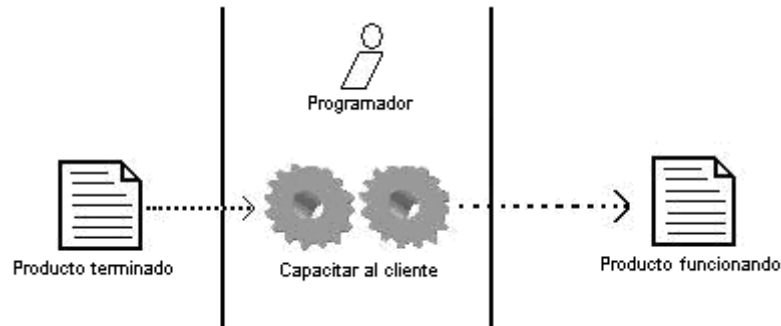
Figura 4.21: Entradas y resultado de implementar nuevas historias de usuario

- **Realizar nuevas pruebas funcionales:** se realizan las pruebas funcionales correspondientes a las nuevas historias de usuario y se determina el grado de satisfacción de los requerimientos del cliente. Las entradas y resultado de la actividad se muestran en la figura 4.22.



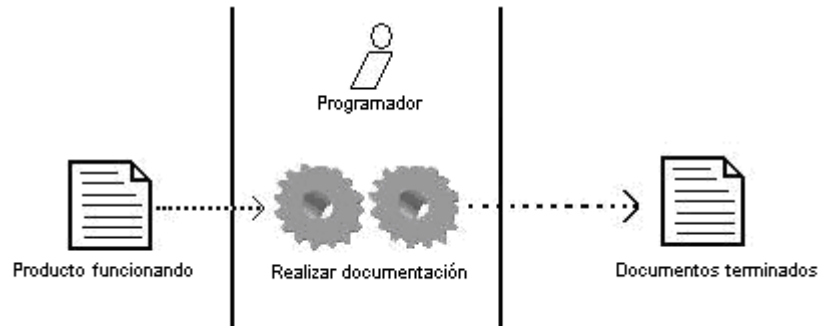
4.22: Entradas y resultado de realizar pruebas funcionales finales

- **Capacitar al cliente:** se entrena al cliente en la instalación y manejo del producto terminado, la figura 4.23 muestra las entradas y resultado de la actividad.



4.23: Entradas y resultado de capacitar al cliente

- **Realizar documentación:** Se completa la documentación relacionada con el proceso, en la figura 4.24 se muestra entradas y resultados de la práctica



4.24: Entradas y resultado de realizar documentación

4.5.3 Entradas fase de mantenimiento y muerte

- **Prototipo Funcional:** la versión en funcionamiento obtenida en la etapa de producción.
- **Nuevas funcionalidades:** correspondientes a las nuevas historias de usuario y las modificaciones que de común acuerdo programadores y cliente deciden poner en funcionamiento para el producto final.

4.5.4 Salidas fase de mantenimiento y muerte

- **Producto en funcionamiento:** como resultado de la capacitación al cliente se obtiene el sistema trabajando en manos del cliente.
- **Monografía y Anexos:** Al finalizar todo el proceso se realiza la monografía (documento que expone la metodología y sus resultados) y la documentación relacionada que amplía y explica en mayor nivel de detalle aspectos que no se profundizan en la monografía.

1. Metodologías Ágiles
2. Historias de Usuario y pruebas

3. Estado de la industria del software en el suroccidente colombiano
4. Manuales de usuario e instalación: contienen las instrucciones, que deben ser seguidas por el cliente, para instalación y funcionamiento correctos del software desarrollado.
5. Otros anexos

4.6 Consideraciones de Aplicación

En esta sección se describen los lineamientos a tener en cuenta para la aplicación del proceso XP adaptado.

- **Fase de exploración:** debe tenerse en cuenta que en la región, la metodología es poco conocida y su aplicación hasta el momento se limita a un escaso número de organizaciones, por lo tanto, quienes estén interesados en aplicar XP deben capacitarse para lograr el conocimiento que les permita la comprensión y aceptación del proceso, los valores y las prácticas que involucra.

Ante la posibilidad de que el cliente no siempre tenga un nivel de comprensión que le permita escribir las historias de usuario, (que fue lo ocurrido en este caso), uno de los integrantes del equipo de desarrollo (programador o probador), debe estar dispuesto a instruir al cliente, o de ser necesario refinar las historias de usuario, ya que es fundamental una total claridad de lo que se debe implementar en las etapas posteriores. Es probable que esto implique un poco más de tiempo, pero con toda seguridad esto se verá recompensado en las etapas futuras.

Una ventaja de esta fase es la realización del spike arquitectónico, pues la exploración de las diferentes alternativas de solución permite seleccionar la más adecuada.

- **Fase de planeamiento:** actividades como estimación de duración de implementación de cada historia de usuario, implica que los programadores sean experimentados. Para el caso de grupos de programación no avanzados, puede servir de guía un plan de lanzamiento menos detallado teniendo en cuenta los siguientes aspectos: asignar prioridad a las historias de usuario (el cliente es quien decide la prioridad), los programadores implementan en las primeras iteraciones las historias de mayor prioridad, el plan de lanzamientos debe contener las historias asignadas a cada iteración, pero no cuenta con una estimación tan detallada, lo que se puede hacer y también es útil es estimar en qué fecha finalizará cada una de las iteraciones.

Una ventaja que se desprende de esta etapa es que el desarrollo posterior estará enfocado a dar solución a aquellas funcionalidades prioritarias para el cliente, adicionalmente permite una aproximación del tiempo, recursos y alcance del proyecto.

La desventaja más notable es que la metodología es tolerante al cambio de requisitos, por ello, en el transcurso del desarrollo pueden ser agregadas o modificadas algunas historias de usuario, con la correspondiente alteración del plan de lanzamientos realizado.

Otra desventaja en esta etapa es que las estimaciones realizadas suponen unos escenarios bastante ideales, los cuales es difícil que se cumplan en su totalidad. Por eso,

aunque el plan de lanzamiento es importante para dimensionar el proyecto, probablemente presente modificaciones en el transcurso del mismo.

- **Fase de Iteraciones:** el entorno regional está compuesto sobre todo por empresas pequeñas, por ello, en algunos casos no será posible que cada uno de los roles implicados en esta etapa sea desempeñado por una persona diferente. También debe considerarse que el equipo de programadores lo pueden conformar un número impar de personas y en esos casos no se puede poner en práctica la programación por pares en su totalidad.

Para el caso de equipos de menor número de personas, una alternativa posible es que uno de los programadores sea también probador. En la medida de lo posible se debe intentar realizar el código por pares. Cuando el número de programadores es impar, una alternativa para mantener la programación por pares es que una de las personas trabaje sola y se rote de manera que luego trabaje en pareja, de tal modo que todos los programadores estén en capacidad de implementar cualquier sección del código.

El mayor beneficio de la metodología se logra de articular y complementar preferiblemente la totalidad de las doce prácticas, pero algunas están fuera del dominio de las empresas desarrolladoras de software, como es el caso del cliente en el sitio de trabajo. Las posibles alternativas para suplir la ausencia del cliente pueden ser: un representante del cliente con capacidad de decisión disponible, utilización de medios de comunicación como teléfono, servicios de mensajería instantánea; programar reuniones diarias de corta duración en las que tanto el cliente como los programadores aclaren posibles dudas o realicen sugerencias.

La programación extrema es muy abierta y tolerante al cambio, al punto que esto puede convertirse en un obstáculo para la medida del avance y alcance real del proyecto, y aunque parte de su filosofía es darle mayor importancia a la colaboración con el cliente, más que a la negociación de un contrato, es fundamental establecer unos parámetros claros de en que circunstancias y tiempos son aceptables los cambios y adiciones de funcionalidades, de no hacerse así, las organizaciones desarrolladoras corren el riesgo de terminar invirtiendo mucho más de lo estimado en la implementación.

El modelamiento ágil es una buena práctica para complementar el proceso XP, no se debe buscar que estos modelos sean los más completos y detallados, para evitar la inversión en tiempo que significa mantener unos modelos actualizados con los requisitos son cambiantes. Los modelos deben estar enfocados en el propósito del modelamiento y a quien van dirigidos, así es posible evitar la realización de modelos que no tendrán utilidad pero que implican tiempo y recursos.

Una ventaja de esta fase del proceso es que al poner en práctica las entregas frecuentes de historias de usuario implementadas se identifican a tiempo los posibles cambios necesarios, al realizarse los cambios en etapas tempranas, se reduce el costo para las organizaciones. Una ventaja adicional de las entregas frecuentes es que proporcionan agilidad al proceso de desarrollo y brindan seguridad y confianza a ambas partes.

Otra ventaja adicional es que todo el código producido esta siendo probado, se realiza pruebas a las nuevas funcionalidades de forma individual y posteriormente cuando son integradas, adicionalmente también se debe refactorizar, esto hace que el código producido sea de calidad.

La desventaja más notable para el proceso se presenta cuando el cliente o un representante autorizado no pueden estar disponibles, porque sin ellos no es posible realizar las entregas frecuentes deseadas, ni tampoco se puede cumplir con el plan de lanzamientos establecido.

- **Fase de producción:** en esta etapa el cliente puede sugerir nuevos cambios, por ello el equipo de desarrollo debe ser muy cuidadoso en aceptar o no estos últimos cambios. Adicionalmente debe comprometerse al cliente a una mayor presencia en el sitio de trabajo ya que es necesaria una comprobación extra del sistema.

Una desventaja se presenta si el cliente no puede estar disponible para la aceptación de la totalidad de las pruebas y de la aceptación de todo el producto desarrollado, pero el proceso incluye la presencia del cliente como una de las prácticas.

No se tiene consideraciones especiales para la fase de mantenimiento y muerte

ANEXO 5: MODELOS

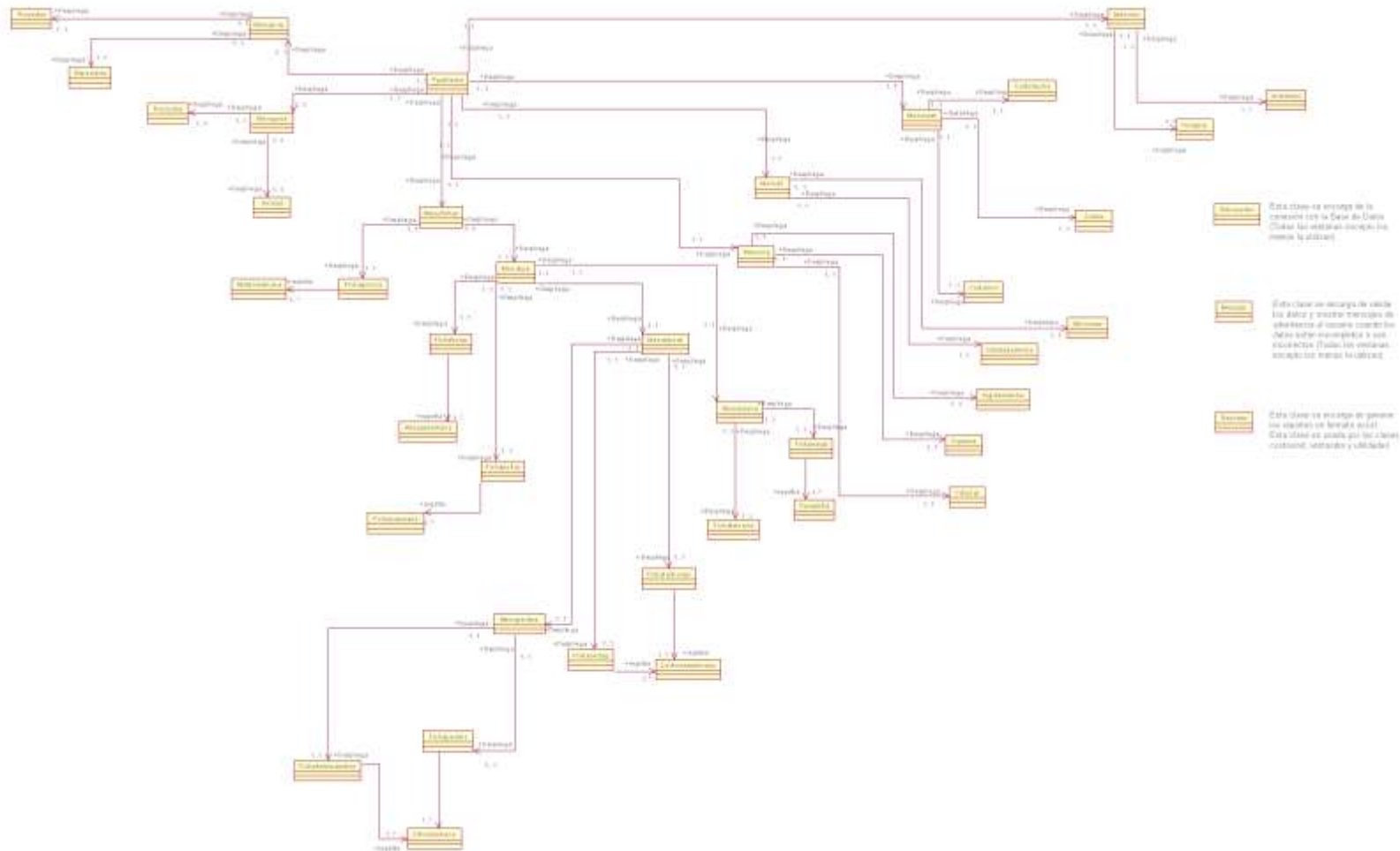


Figura 5.1: Diagrama de clases

