



Universidad de las Ciencias Informáticas
Facultad 4

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Aplicación informática para la gestión de los entornos de pruebas del Laboratorio Industrial de Pruebas de Software.

Autores: Deyrin Perez Pedroso.
Carlos Fuentes Puentes.

Tutores: Danelis Padilla Pérez.
Aniubis Rodríguez Batista.

La Habana, Cuba
Curso: 2013-2014

Declaración de autoría

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) a que haga el uso que estime pertinente con el mismo.

Para que así conste firmamos la presente a los _____ días del mes de _____ del año _____.

Autor(es):

Deyrin Perez Pedroso.

Carlos Fuentes Puentes.

Tutoras:

Ing. Danelis Padilla Pérez.

Ing. Aniubis Rodríguez Batista.

Datos del contacto

Datos del contacto

Tutores

Ing. Danelis Padilla Pérez.

Email: dpperez@uci.cu

Ing. Aniubis Rodríguez Batista.

Email: arbatista@uci.cu

Autores

Deyrin Perez Pedroso.

Email: dpedrozo@estudiantes.uci.cu

Carlos Fuentes Puentes.

Email: cpuentes@estudiantes.uci.cu

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Frases



“...Tu tiempo es limitado, de modo que no lo malgastes viviendo la vida de alguien distinto. No quedes atrapado en el dogma, que es vivir como otros piensan que deberías vivir. No dejes que los ruidos de las opiniones de los demás acallen tu propia voz interior. Y, lo que es más importante, ten el coraje para hacer lo que te dice tu corazón y tu intuición”.

Steve Jobs

Dedicatoria y agradecimientos



*Dedico esta tesis a la persona más importante de mi vida, la que ha sacrificado todo por mí, por su amor, ayuda y apoyo incondicional, por confiar siempre en mí. Por ser mi motivación, ejemplo, consejera y amiga. Por darme la vida y estar siempre a mi lado en los momentos buenos y malos, y por quien hoy, he llegado a ser quien soy. Esta tesis está dedicada a **Evelin Pedroso Blanco**, mi Madre.*

Agradezco a mi madre, por ser la mujer más luchadora del mundo, por ser madre y padre y por ser ese motor impulsor que me guió durante todo este largo camino.

*A mi padrastro (**Jaime**), por ser como un padre para mí y por todo el apoyo que siempre me brindó.*

*A mi abuela (**Zelma**) por ser mi segunda mamá, por educarme, apoyarme y aconsejarme en todos los momentos de mi vida. Por siempre estar presente dándome su amor.*

*A mi hermanita linda (**Denis**), es el regalo más grande que me ha dado la vida, gracias por siempre creer en mí.*

*A mi novio (**Yunierkis**), por ser la persona que ha estado estos dos últimos años a mi lado, apoyándome, por preocuparse tanto por mí, por haber sido mi sostén y por amarme tanto, muchas gracias mi amor.*

*A mi padre (**Ernesto**), por apoyarme y ayudarme.*

A mis amigos, los cuales siempre me han ayudado y apoyado en lo largo de la carrera, en fin, a todas las personas que de una forma u otra me ha dado su ayuda.

Muchas Gracias.

Dedicatoria y agradecimientos



Dedico este trabajo a mi Dios Jesucristo, gracias por su amor y por su paz que sobrepasa todo entendimiento. A mis padres Carlos Fuentes Gonzales y Beatriz Puentes Rodríguez, por su apoyo y amor todos estos años de estudios, a mi familia y amigos.

Agradecer primeramente a Dios, por las fuerzas para realizar este trabajo, a esa fuerza y paz que solo puede dar Él, agradecer a mis padres por la educación que me han dado, por el amor y comprensión que me han dado hasta el día de hoy. Agradecer a cada uno de los miembros de mi familia, mis abuelas, mi abuelo, mi tío, a su esposa, agradecer a mis hermanas Mairyn y Yusmila, son un regalo de Dios para mí. Agradecer a todos mis hermanos en Cristo, tanto los que están en la escuela como los que se han ido, a Annia, Yignelys, Lizardo, Mairyn, Alejandro, a Dailen, a Elizabeth, Ariannis, Aime, Eliza, Yanelis, Laritza, Alien, Leandro, Yasmany, agradecer a esa persona que este último año en la universidad siempre ha estado muy atenta a mí, la persona que me ha dedicado largas horas de conversaciones, María, a todos los quiero mucho, agradecer a mis amigos del apartamento, mis amigos de la escuela q nunca olvidaré, El Yoe, Joel, Diego, Roney, gracias a todos por compartir y estar tanto en los momentos alegres como en los tristes.

Ante la necesidad que presenta el Laboratorio Industrial de Pruebas de Software de gestionar los entornos de prueba; surge el presente trabajo, que tiene como objetivo general desarrollar una

Dedicatoria y agradecimientos

aplicación que permita la gestión de los entornos de prueba según las características de Calidad en el centro. En la confección del sistema se utilizó la metodología de desarrollo XP; Visual Paradigm8.0, como herramienta CASE; Symfony2 como framework de desarrollo; PostgreSQL como sistema gestor de base de datos; Apache2 como servidor web; PHP5, HTML5 como lenguajes de programación y NetBeans7.4 como entorno de desarrollo. Se determinaron e implementaron todas las funcionalidades que debía tener el sistema y se realizaron las pruebas a la solución propuesta, donde se obtuvieron resultados favorables para el equipo de desarrollo, que muestran un alto grado de satisfacción por parte del cliente. La creación de la aplicación para la gestión de los entornos de prueba según las características de Calidad contribuirá a la sistematización y organización de los entornos de prueba del LIPS de CALISOFT.

Palabras clave

Entornos de prueba, gestión de los entornos de prueba.

Índice

Introducción	1
Capítulo 1. Fundamentación Teórica	5
Introducción.....	5
1.1 Principales conceptos asociados al dominio del problema	5
1.2 Estado del arte a nivel internacional.....	6
1.2.1 Redmine	7
1.2.2 Project KickStart 4	7
1.2.3 OpenWorkbench.....	7
1.2.4 QATraq.....	8
1.3 Estado del arte a nivel nacional.....	9
1.3.1 GESPRO	9
1.4 Valoraciones de estas herramientas.....	9
1.5 Tendencias y tecnologías a utilizar.....	10
1.5.1 Metodología de desarrollo de software	10
1.5.1.1 SCRUM5	12
1.5.1.2 Crystal Methodologies6	13
1.5.1.3 Dynamic Systems Development Method7 (DSDM).....	13
1.5.1.4 Adaptive Software Development8 (ASD)	13
1.5.1.5 Feature -Driven Development9 (FDD)	14
1.5.1.6 Lean Development10 (LD).....	14
1.5.1.7 Programación extrema (eXtreme Programming, XP).....	14
1.6 Framework.....	15
1.6.1 Symfony	15
1.6.2 Zend Framework	16
1.7 Entornos de desarrollo integrado (IDE)	17
1.7.1 NetBeans	17
1.7.2 Eclipse	18
1.8 Sistema Gestor de Base de Datos	18

1.8.1 PostgreSQL.....	19
1.8.2 MySQL.....	19
1.9 Lenguaje de programación web.....	20
1.9.1 Lenguajes de Programación del lado del cliente HTML (Hyper Text Markup Language) .	20
1.9.1.2 JavaScript.....	21
1.9.2 Lenguajes de Programación del lado del servidor. PHP (Hypertext Preprocessor).....	21
1.10 Herramientas de Modelado.....	22
1.10.1 Rational Rose Enterprise.....	22
1.10.2 Visual Paradigm-UML.....	23
1.11 Conclusiones Parciales.....	23
Capítulo 2. Características del sistema.....	24
Introducción.....	24
2.1 Descripción del flujo actual del proceso.....	24
2.1.1 Descripción del subproceso: Inicio y Planificación.....	25
2.1.1.2 Descripción textual del subproceso: Inicio y Planificación.....	25
2.1.2 Descripción del subproceso: Ejecución.....	28
2.1.2.1 Descripción textual del subproceso: Ejecución.....	29
2.1.3 Descripción del subproceso: Cierre.....	31
2.1.3.1 Descripción textual del subproceso: Cierre.....	32
2.2 Propuesta de solución.....	33
2.3 Usuarios relacionados con el sistema.....	36
2.4 Propuesta de arquitectura.....	36
2.4.1 Patrón Modelo Vista Controlador (MVC).....	37
2.5 Planificación.....	38
2.5.1 Historia de Usuario (HU).....	39
2.5.2 Estimación de esfuerzo por HU.....	42
2.5.3 Plan de iteraciones.....	42
2.5.4 Plan de duración de las iteraciones.....	43
2.5.5 Plan de entregas.....	43
2.6 Diseño del sistema.....	45

2.6.1 Tarjetas CRC	46
2.7 Conclusiones Parciales	48
Capítulo 3. Implementación y Prueba	50
Introducción	50
3.1 Implementación	50
3.1.1 Iteración 1	50
3.1.2 Iteración 2	51
3.1.3 Iteración 3	51
3.2 Codificación	52
3.3 Tareas de ingeniería	52
3.4 Patrones de diseño usados en la aplicación	55
3.5 Estándares de codificación	57
3.6 Pruebas de software	58
3.6.1 Pruebas de Aceptación	58
3.7 Conclusiones Parciales	72
Conclusiones Generales	73
Recomendaciones	74
Referencias bibliográficas	75
Bibliografía consultada	79
Anexos	81

Índice de figuras

Figura 1_Diferencias entre metodologías ágiles y no ágiles.....	12
Figura 2_Etapas de procedimiento de evaluación.....	24
Figura 3_Patrón MVC	38
Figura 4_Diseño de estructura global del sistema.....	45
Figura 5_Ejemplo del patrón Experto.	56

Índice de tablas

Tabla 1_Descripción textual del subproceso: Inicio y Planificación.	26
Tabla 2 Descripción textual del subproceso: Ejecución.	29
Tabla 3_Descripción textual del subproceso: Cierre.....	32
Tabla 4_Usuarios relacionados con el sistema.	36
Tabla 5_HU.Gestionar usuarios.	39
Tabla 6_HU.Gestionar las solicitudes de instalación de Entornos de Prueba.....	40
Tabla 7_HU.Gestionar entorno de prueba real (EP).....	40
Tabla 8_HU.Gestionar roles.	41
Tabla 9_HU.Gestionar proyectos.....	41
Tabla 10_Estimación de esfuerzo por HU.	42
Tabla 11_Plan de duración de las iteración.	43
Tabla 12_Plan de entregas.	44
Tabla 13_Tarjeta CRC.....	46
Tabla 14_Funcionalidades implementadas en la primera iteración.	51
Tabla 15_Funcionalidades Implementadas en la segunda iteración.	51
Tabla 16_Funcionalidades implementadas en la tercera iteración.....	52
Tabla 17_Tarea #1.	53
Tabla 18_Tarea #2.	53
Tabla 19_Tarea #3.	54
Tabla 20_Tarea #4.	54
Tabla 21_Tarea #5.	54
Tabla 22_Tarea #6.	55
Tabla 23_Tarea #7	55
Tabla 24_Tarea #8.	55
Tabla 25_CPA HU1_P1 Gestionar usuarios.	59
Tabla 26_CPA HU3_P1 Gestionar solicitud de instalación de Entorno de Prueba (EP).	59
Tabla 27_CPA HU4_P1 Gestionar entorno de prueba real.	60

Índice de tablas

Tabla 28_CPA HU5_P1 Gestionar roles.....	60
Tabla 29_CPA HU6_P1 Gestionar proyectos.....	61
Tabla 30_CPA HU7_P1 Gestionar planificación.....	62
Tabla 31_CPA HU8_P1 Gestionar turnos.....	62
Tabla 32_CPA HU9_P1 Gestionar puestos de trabajo.....	63
Tabla 33_CPA HU10_P1 Gestionar incidencias.....	63
Tabla 34_CPA HU10_P3 Gestionar incidencias.....	64
Tabla 35_CPA HU10_P2 Gestionar incidencias.....	65
Tabla 36_CPA HU9_P2 Gestionar puestos de trabajo.....	65
Tabla 37_CPU HU8_P2 Gestionar turnos.....	66
Tabla 38_CPU HU8_P3 Gestionar turnos.....	66
Tabla 39_CPU HU9_P3 Gestionar puestos de trabajo.....	67
Tabla 40_CPU HU8_P3 Gestionar turnos.....	67
Tabla 41_CPU HU6_P4 Gestionar proyectos.....	68
Tabla 42_CPU HU6_P3 Gestionar proyectos.....	69
Tabla 43_CPU HU6_P2 Gestionar proyectos.....	69
Tabla 44_CPU HU7_P4 Gestionar planificación.....	70
Tabla 45_CPU HU5_P3 Gestionar rol.....	71
Tabla 46_CPU HU5_P2 Gestionar rol.....	71

Introducción

El desarrollo tecnológico, a lo largo de la historia es uno de los más enfatizados procesos ejecutados por el hombre. En estos tiempos se ha podido percibir que el desarrollo tecnológico ha evolucionado paulatinamente. (1)

“Este año se cumplieron 54 años de la entrada de la primera computadora electrónica en Cuba –una IBM RAMAC 650– y 42 años de la fabricación por primera vez de una computadora en el país. En esta etapa de la evolución humana, donde los cambios sociales, económicos y tecnológicos se suceden a velocidades vertiginosas, 54 y 42 años respectivamente son plazos de tiempo suficientemente largos” (1)

“Cuba exporta actualmente productos y servicios de software a más de 20 países de Europa, América del Sur y Centroamérica, y se propone sumar mercados en su propia vecindad, el Caribe insular. La mitad de esas ventas externas las realiza la empresa ICID-COMBIOMED¹, dedicada a la informática médica”.(2)

“Con el desarrollo de la computación, se hizo necesario extender la rama de la informática al pueblo, por lo que se incluyeron carreras afines a esta disciplina en distintas universidades del país. Esto provocó un gran auge en el desarrollo del software cubano”. (1)

La Universidad de las Ciencias Informáticas (UCI), inició su funcionamiento en el curso 2002 – 2003, es una universidad de nuevo tipo, primera de la Batalla de Ideas que lleva a cabo el pueblo cubano, con un modelo de formación que combina el estudio con la producción y la investigación.

La UCI a partir del año 2004 comienza a tener una presencia productiva en la Industria del Software con el desarrollo de proyectos informáticos fundamentalmente relacionados con la informatización de la sociedad y la exportación. Para asegurar la calidad de los productos dentro de la UCI, surge la Dirección de Calidad de Software, actualmente Centro Nacional de Calidad de Software (CALISOFT). Este centro es responsable de la verificación y validación de productos y procesos según normas

¹Instituto Central de Investigación Digital (ICID)- COMBIOMED es la empresa de comercio exterior del ICID con derechos exclusivos de venta de sus equipos en el extranjero.

nacionales e internacionales. Dentro de su estructura organizativa se encuentra el Departamento de Evaluación de Productos de Software (DEPSW).

El DEPSW gestiona las solicitudes de prueba realizadas por entidades externas y principalmente los centros de desarrollo de la UCI, pero es justamente el Laboratorio Industrial de Pruebas de Software (LIPS) quien posee como principal tarea la gestión de los procesos tecnológicos llevados a cabo con el objetivo de garantizar la correcta ejecución de los procesos de prueba y que estos sean preparados de una manera rápida y sobre todo con la calidad requerida.

Actualmente la gestión de los entornos de pruebas en el Laboratorio Industrial de Pruebas de Software (LIPS) se controla mediante hojas electrónicas de cálculo (Microsoft excel) que permiten registrar la información de los productos, sus proyectos y características, y especificaciones de los servidores solicitados por estos últimos en cada proceso de prueba por el que transitan. Además, se realiza y registra la planificación de los proyectos para la instalación y actualización de sus aplicaciones, teniendo en cuenta las máquinas y horarios disponibles, reflejándose la siguiente situación problemática.

Estas hojas de cálculo hacen que el proceso sea complejo, pues es necesario introducir los datos en diferentes documentos; se torna complicado registrar una descripción en una celda. Las solicitudes para la actualización o instalación de los proyectos al laboratorio son numerosas. Al gestionar las mismas a través de un Excel en un repositorio de control de versiones, si se realizan cambios al mismo tiempo por diferentes usuarios se puede generar pérdida de información, complicando y haciendo delicado el proceso. Como consecuencia hoy realiza la planificación un pequeño grupo del laboratorio aumentando su carga de trabajo e interviniendo en la calidad de sus actividades, siendo esta una tarea que corresponde realizar directamente a los proyectos.

Resulta sumamente complejo realizar análisis a la información obtenida, pues está separada en varios documentos. A medida que pasa el tiempo el cúmulo de información aumenta.

El LIPS no cuenta actualmente con una herramienta que permita la gestión de los entornos² de prueba de manera factible. Esto incluye la planificación de los proyectos para instalar o actualizar sus

² Ambiente controlado en el cual se ejecutan las actividades de las pruebas.

aplicaciones, registro de incidencias, datos relacionados con dicha acción, la gestión de turnos y puestos de trabajo para la actualización. Además del registro de toda la información relacionada con los servidores y las reuniones de inicio. No se tiene constancia de notificaciones o de la realización correcta del proceso por parte de los miembros del grupo.

Todo esto, unido a la cantidad de pasos que resultan necesarios para culminar un proceso y el hecho de que la información esté separada en diferentes documentos y para realizar una acción sea a veces necesario abrir más de un documento y consultar otros, genera atrasos en el proceso.

Por lo anteriormente planteado se define el siguiente **problema a resolver**: ¿Cómo contribuir a la gestión de entornos de pruebas del Laboratorio Industrial de Pruebas de Software?

Teniendo en cuenta el problema planteado se define el **siguiente objetivo general**: Desarrollar una aplicación informática que contribuya a la gestión de entornos de pruebas del Laboratorio Industrial de Pruebas de Software.

A partir del objetivo general definido se derivan los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación.
- Realizar el análisis y diseño del sistema de gestión de los entornos de pruebas del LIPS.
- Realizar la implementación y prueba del sistema de gestión de los entornos de pruebas del LIPS.

Se define como **objeto de estudio**: Procesos de gestión de entornos de pruebas, enmarcándose en el **campo de acción**: Procesos de gestión de entornos de pruebas del Laboratorio Industrial de Pruebas de Software.

Idea a defender: Con el desarrollo de la aplicación informática se contribuirá a la gestión de los entornos de pruebas del Laboratorio Industrial de Pruebas de Software.

Para dar cumplimiento a los objetivos específicos propuestos, se definen las **tareas de la investigación** siguientes:

- Análisis de sistemas similares existentes a nivel nacional e internacional.
- Descripción de las herramientas, tecnologías y metodología utilizada en el desarrollo del sistema.

- Identificación de los artefactos a utilizar en el diseño de la propuesta de solución.
- Confección del diseño del sistema.
- Implementación del sistema.
- Realización de pruebas al sistema.
- Revisión de funcionalidades implementadas con el cliente.

En la presente investigación se utilizaron los métodos científicos de investigación siguientes:

Métodos teóricos

- Histórico – lógico: Mediante este se lleva a cabo una mejor comprensión de la evolución de las Pruebas de Software, así como otras temáticas relacionadas con la investigación.
- Analítico – sintético: Se maneja para el estudio de la literatura especializada correspondida con el tema, lo que permite la fabricación de los fundamentos teóricos y las conclusiones.

Métodos empíricos

- Observación: Para observar las actividades que se desarrollan en el Laboratorio Industrial de Pruebas de Software y determinar el flujo de los procesos.

El presente documento se estructura por 3 capítulos que abordan los temas fundamentales distribuidos de la siguiente manera:

Capítulo 1: “Fundamentación Teórica de la Investigación”

Se exponen los principales conceptos de necesario estudio para el desarrollo de la solución propuesta, así como las metodologías, herramientas y tecnologías a usar para el desarrollo de la solución.

Capítulo 2: “Concepción y características del sistema”

Se describe la propuesta del sistema y se detallan las funcionalidades que el sistema debe ser capaz de cumplir.

Capítulo 3: “Implementación y pruebas”

Se abordan los aspectos relacionados con la construcción de la solución propuesta, además de la documentación correspondida a las pruebas aplicadas al sistema.

Capítulo 1. Fundamentación teórica

Capítulo 1. Fundamentación Teórica

Introducción

En este capítulo se desarrolla el marco teórico referencial de la investigación. Se ofrece una breve panorámica de diferentes herramientas que se utilizan para planificar y gestionar entornos de pruebas, sus características y ventajas. Se expone el proceso de selección de las herramientas y tecnologías a utilizar en la implementación de la aplicación, así como la metodología que guiará su proceso de desarrollo.

1.1 Principales conceptos asociados al dominio del problema

En la problemática de la vigente investigación se ven expresados varios conceptos que por su importancia son expuestos a continuación:

Proceso

Conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados.(3)

Planificación

La planificación constituye una función básica y punto de partida de la gestión. Se concibe como "un proceso continuo y sistemático en el que las personas toman decisiones sobre acciones futuras, sobre el respaldo que deben tener dichas acciones futuras y cómo evaluar y medir el éxito.(4)

Según Charles R. McClure, la planificación se divide en cuatro pasos esenciales:

- Identificar las necesidades de la comunidad.
- Desarrollar la misión, metas y objetivos de la organización.
- Diseñar programas y actividades para cumplir los objetivos.
- Evaluar el éxito de dichas actividades.

La planificación, como se ha observado, requiere la capacidad de determinar, definir y estructurar el conjunto de acciones estratégicas, tácticas y operativas para el desempeño de la organización en el

Capítulo 1. Fundamentación teórica

presente y en el futuro, en concordancia con su entorno, lo que obligatoriamente exige un examen continuo de los resultados obtenidos. (4)

En términos generales, puede concebirse como un proceso eminentemente creativo, de decisión y orientador; cuyo valor radica, en mayor medida, en su propio decurso y su éxito depende de la realización completa e ininterrumpida del ciclo gerencial por lo que no concluye con la confección de un plan. Su sistematicidad, elemento distintivo, está determinada por la posesión de una metodología específica y la secuencia lógica de eventos donde se exponen las vías para lograr los deseos de la organización. Por consiguiente, involucra a todos los recursos humanos, tanto para la toma de decisiones como para la ejecución de las acciones orientadas y conducentes al éxito organizacional, consignadas y descritas con nitidez en el denominado plan, que no sólo constituye guía del accionar sino que ha de permitir la evaluación cuantitativa y cualitativa de la organización en el transitar de la situación actual a la deseada. (4)

Gestión

Se denomina gestión al correcto manejo de los recursos que dispone una determinada organización, como por ejemplo, empresas, organismos públicos, organismos no gubernamentales, etc. El término gestión puede abarcar una larga lista de actividades, pero siempre se enfoca en la utilización eficiente de estos recursos, en la medida en que debe maximizarse sus rendimientos.(5)

Entorno

Son los ordenadores e infraestructuras donde los programadores trabajan para hacer los programas. Se caracteriza por no tener unos requerimientos de disponibilidad y rendimiento demasiado exigentes, ya que los programas no están a disposición del usuario final, y por lo tanto los errores que puedan surgir no tienen especial importancia.(6)

1.2 Estado del arte a nivel internacional

Para la gestión de los entornos de prueba del Laboratorio Industrial de Pruebas de Software se estudiaron diferentes herramientas buscando una solución a la situación problemática descrita. Dentro de las herramientas estudiadas se pudo constatar que varias de ellas son muy populares para planificar actividades y gestionar entornos de pruebas.

Capítulo 1. Fundamentación teórica

1.2.1 Redmine

Es una herramienta de gestión de proyectos de software con interface web. Es un sistema multi-plataforma³ y soporta múltiples motores de bases de datos relacionales. Resiste variados proyectos. Hace seguimiento de los tiempos relacionados. Maneja archivos, documentos y noticias. Genera información en base a calendarios y diagramas Gantt. Permite el auto registro de usuarios. Admite la creación de tareas vía correo electrónico.

1.2.2 Project KickStart 4

Project KickStart4 es una herramienta que posee nuevas características con respecto a sus versiones anteriores, entre ellas una interfaz gráfica mucho más fácil de usar, incluso para aquellos usuarios que se inician en el mundo de la planificación de software y que aún no tienen gran experiencia en el uso de estas herramientas.

Entre sus inconvenientes se encuentra que es caro, para usarse debe usar una computadora con tecnología Pentium, Sistema Operativo (SO) Windows 95, 98, 2000, NT y XP y 32 MB RAM. Sus necesidades tecnológicas lo hacen imposible de usar para usuarios del tercer mundo u otros lugares subdesarrollados que no usen dicha tecnología. Además, teniendo en cuenta su precio y el no conocimiento de su código fuente, hace imposible realizarle modificaciones para familiarizarlo con el entorno de trabajo en que se desea emplear. (7)

Otro de los inconvenientes de esta herramienta es que se usa para la planificación, pero la misma no incluye los elementos que se necesitan para la gestión de los entornos de prueba. Además, la planificación que se lleva a cabo en CALISOFT tiene sus particularidades pues se tiene en cuenta los recursos (humanos, tecnológicos) y el tiempo.

1.2.3 OpenWorkbench

OpenWorkbench es una herramienta de planificación de proyectos, al estilo de Microsoft Project pero con la diferencia de que es de código abierto. Se ejecuta en entornos Windows y es completamente libre y gratuito. Soporta un amplio abanico de funciones, tales como la asignación de tareas hacia delante y hacia atrás, análisis de hitos y tiempo estimado para la terminación de la solución. Posee

³ Poder funcionar o mantener una interoperabilidad de forma similar en diferentes sistemas operativos.

Capítulo 1. Fundamentación teórica

una práctica única de programar tareas de acuerdo con restricciones de recursos, siendo esta una variedad nueva con respecto a otro software de planificación de proyectos. Esta característica ofrece un plan mucho más real y utiliza menos tiempo, pues calcula la duración de la tarea en esfuerzo por disponibilidad de recursos. Esta herramienta contiene un programa que permite afinar funcionalidades del calendario como son las fechas de comienzo y finalización. Permite además un sinfín de tareas, como es el diseño de gráficas de Gantt, asignación de tareas, relaciones, tiempos de actividades, entre otros. Su principal inconveniente es que no es compatible con Linux, por tanto, es imposible de usar para usuarios amantes de este sistema. Sin embargo, es una herramienta que al ser OpenSource⁴ viabiliza el problema de licencias en Cuba. Es adaptable al entorno de trabajo con tan solo modificar cualquiera de sus componentes. Se ajusta y es compatible con los ideales subyacentes en la administración de proyectos, a la vez presenta información de manera intuitiva y fácil de aprender. Decenas de miles de administradores de proyectos de todo el mundo usan Open Workbench para planificar y ejecutar proyectos complejos. Todos los proyectos avanzan a través de una serie de tareas (o etapas) durante su ciclo de vida. Al usar Open Workbench, estas tareas o etapas críticas se vuelven más administrables, lo cual facilita cada vez más el éxito de los proyectos.(8) El inconveniente de esta herramienta es que no se adapta a las características que debe poseer el sistema, pues este solo se utiliza para la planificación de tareas y también se requiere que la propuesta gestione las actividades orientadas a gestionar los entornos de prueba dirigidos a garantizar el soporte tecnológico de las pruebas que se realizan en CALISOFT. Además, desde el punto de vista futurista no es viable un sistema que solo funcione en Windows, ya que la tendencia en nuestro país es a la migración al software libre dada las dificultades que el bloqueo económico impuesto a Cuba trae para la adquisición de licencias de software.

1.2.4 QATraq

QATraq es una herramienta gratuita para la gestión de entornos de prueba, es de código abierto y está desarrollada en PHP. Muy fácil de instalar en cualquier entorno (Linux, Windows, Solaris ó Mac). Esta herramienta permite gestionar el ciclo de vida de las pruebas. Desde la creación de un Plan de test, escribir vuestros Casos de Prueba o la ejecución. QATraq es ideal para equipos de Test de pequeño y mediano tamaño. Permite obtener informes y saber en qué parte del proceso de ejecución

⁴ Software cuyo código fuente es publicado bajo una licencia de software.

Capítulo 1. Fundamentación teórica

encuentra. Está basado en LAMP (Linux, Apache, Mysql y PHP) por lo que es de fácil instalación en cualquier plataforma. (9)

1.3 Estado del arte a nivel nacional

A nivel nacional existen diferentes herramientas para la planificación de actividades.

1.3.1 GESPRO

Actualmente en la Universidad de las Ciencias Informáticas existe un equipo de trabajo dedicado a la expansión del Redmine, esta nueva versión es llamada Ecosistema de software para la Dirección Integrada de Proyectos (GESPRO): combina el uso de una solución informática para la dirección integrada de proyectos y un sistema de formación especializada en gestión de proyectos. Posibilita la planificación, el control, seguimiento de los proyectos y los recursos, en alineación con la proyección estratégica de las organizaciones. GESPRO se comercializa bajo un modelo de negocios basado en servicios entre los que se destacan: personalización de la solución informática, despliegue en diferentes escenarios, actualizaciones, soportes, consultorías y formaciones especializadas.(10)

La solución informática está completamente basada en tecnologías libres y se comercializa bajo licencia GNU-GPL 2. El núcleo de la misma es la plataforma GESPRO 13.05 que es un producto registrado en el Centro Nacional de Derecho de Autor de Cuba, No Registro CENDA 1540-2010. Es una herramienta muy útil para el monitoreo, la gestión de riesgos, incluso es altamente integrable con otras herramientas de gestión web, sin embargo no está acorde con las características del software que se necesita, además hacerle los cambios que se requieren costaría más tiempo y esfuerzo que hacer una aplicación nueva. El equipo de desarrollo de GESPRO acepta solicitudes de personalización, sin embargo el equipo es muy pequeño y responder estas solicitudes podría demorar meses. (10)

1.4 Valoraciones de estas herramientas

Luego de elaborado un análisis sobre varias herramientas existentes, a nivel nacional e internacional se llega a la conclusión de que pese a las ventajas que las aplicaciones antes mencionadas poseen, existen algunas limitaciones para su uso. Estas herramientas tienen una infraestructura y funcionalidad básica establecidas esencialmente para el campo específico en que se despliegan, de

Capítulo 1. Fundamentación teórica

modo que no satisfacen los intereses del proceso que se desea informatizar. El principal inconveniente de las herramientas antes expuestas es que unas se especializan en la planificación y otras en la gestión de entornos de prueba, y para poder adaptarlas a las particularidades del proceso realizado en CALISOFT se necesitaría tener acceso a su código fuente además de que la magnitud de los cambios traerían consigo un gasto de tiempo, esfuerzo y recursos mayor que lo planificado. Algunas de estas aplicaciones antes aludidas son en su totalidad, software propietario, o sea, son herramientas donde la adquisición de sus licencias en porcentos monetarios se tornan muy elevadas, además, de las restricciones que el bloqueo económico trae para la adquisición de estas licencias, y en caso de poder adquirirlas el soporte y mantenimiento de las mismas resultaría un proceso muy engorroso. Dada la proyección futurista de nuestro país de migrar a software libre se desecharon herramientas que no sean multiplataforma o que su uso dependan de algún software o componente privativo. Por las razones antes expuestas, se considera que no es posible utilizar ninguna de las herramientas antes aludidas. Por la carencia de un sistema con los requerimientos necesarios para gestionar los entornos de pruebas del LIPS de manera factible, se hace inminente el desarrollo de una aplicación que contribuya a mejorar el proceso de gestión de entornos de pruebas realizado.

1.5 Tendencias y tecnologías a utilizar

Teniendo en cuenta que la herramienta a desarrollar es una aplicación que gestione los entornos de prueba del LIPS, se determinaron las herramientas, metodologías y lenguaje de programación, para garantizar que el desarrollo de la misma sea satisfactorio, se procede entonces a la realización de un estudio de las metodologías de desarrollo de software, herramientas y tecnologías actuales, determinando así las más apropiadas para emplearse en la solución.

1.5.1 Metodología de desarrollo de software

El desarrollo de un software depende de actividades, etapas, donde tiene gran peso elegir la metodología más adecuada para lograr el éxito del producto. El objetivo de las metodologías de software es guiar a los desarrolladores a la hora de implementar un nuevo programa, definiendo dentro de cualquier proyecto ¿Quién debe hacer?, ¿Qué?, ¿Cuándo y Cómo debe hacerlo? Las metodologías contribuyen en gran medida a que el trabajo se realice de forma estructurada y organizada. El proceso de desarrollo de software no es una tarea sencilla, es la existencia de

Capítulo 1. Fundamentación teórica

numerosas propuestas metodológicas. Decidir cuál metodología usar es de vital importancia, pues es la encargada de guiar todo el proceso de desarrollo del software. Por una parte se tienen las propuestas más tradicionales, que se centran en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones que se usarán. Estas propuestas tradicionales, han exhibido ser efectivas en un gran número de proyectos, pero también han presentado problemas en otros. Por otra parte se tienen las metodologías ágiles, las cuales se centran en otras dimensiones, como por ejemplo el factor humano o el producto software, dando así mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque muestra su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Las metodologías ágiles están revolucionando la manera de producir software, y a la vez generando un amplio debate entre sus seguidores y quienes por escepticismo o convencimiento no las ven como alternativa para las metodologías tradicionales.

(11)

Capítulo 1. Fundamentación teórica

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Figura 1_Diferencias entre metodologías ágiles y no ágiles.

Podría resumirse todo lo antes expuesto de la siguiente forma:

- ✓ **Metodologías tradicionales (formales):** se focalizan fundamentalmente en la documentación, planificación y control de procesos.
- ✓ **Metodologías ágiles:** están orientadas a la interacción continua con el cliente y el desarrollo incremental del software.

Después de haber realizado un estudio y análisis de las características de las metodologías, y atendiendo a que se cuenta con un equipo de trabajo de dos personas con muy poco tiempo de desarrollo, y un alto riesgo a cambios en los requisitos del software, se decide adoptar la filosofía de las metodologías ágiles para guiar el proceso de desarrollo de la solución propuesta, ya que tienen características propias y hacen hincapiés en algunos aspectos más específicos. A continuación se realiza un resumen de algunas de estas metodologías ágiles.

1.5.1.1 SCRUM5

Define un marco para la gestión de proyectos, utilizándose con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales

Capítulo 1. Fundamentación teórica

características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. Su segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.(12)

1.5.1.2 Crystal Methodologies⁶

Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo *Crystal Clear* (3 a 8 miembros) y *Crystal Orange* (25 a 50 miembros).(13)

1.5.1.3 Dynamic Systems Development Method⁷ (DSDM)

Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología RAD unificada. Sus principales características son: proceso iterativo e incremental donde el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases.(14)

1.5.1.4 Adaptive Software Development⁸ (ASD)

Es impulsado por Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software, en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.(15)

Capítulo 1. Fundamentación teórica

1.5.1.5 Feature -Driven Development⁹ (FDD)

Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software. Sus impulsores son Jeff De Luca y Peter Coad.(16)

1.5.1.6 Lean Development¹⁰ (LD)

Definida por Bob Charette.s a partir de su experiencia en proyectos con la industria japonesa del automóvil en los años 80 y utilizada en numerosos proyectos de telecomunicaciones en Europa. En LD, los cambios se consideran riesgos, pero si se manejan adecuadamente se pueden convertir en oportunidades que mejoren la productividad del cliente. Su principal característica es introducir un mecanismo para implementar dichos cambios. (17)

1.5.1.7 Programación extrema (eXtreme Programming, XP)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (18)

Esta metodología fue formulada por Kent Beck, se basa en la reutilización del código desarrollado. Sus objetivos fundamentales son la satisfacción del cliente y potenciar las relaciones interpersonales como clave para el éxito del desarrollo de software, promoviendo el trabajo en equipo.(18)

Esta metodología presenta las siguientes características(18)

- Comunicación: se refiere a crear un ambiente de cooperación y unidad en el equipo de desarrollo.
- Simpleza: busca hacer solo lo necesario evitando el gasto de cualquier recurso.
- Retroalimentación: se basa en la socialización del conocimiento y las soluciones.
- Coraje: se refiere al valor que necesita el grupo de trabajo para enfrentar los cambios.

Ventajas:

- Fomenta la comunicación entre los clientes y los desarrolladores.

Capítulo 1. Fundamentación teórica

- Facilita los cambios.
- Permite ahorrar tiempo y recursos.
- Puede ser aplicada a cualquier lenguaje de programación.
- El cliente tiene el control sobre las prioridades.
- Se hacen pruebas continuas durante el proyecto.

Desventajas:

- Es recomendable emplearla solo en proyectos a corto plazo.

De las metodologías ágiles antes expuestas, se decide utilizar la metodología XP para guiar el proceso de desarrollo de software, ya que el proyecto tiene un plazo corto para su desarrollo y el equipo de trabajo es pequeño, además que XP es flexible a cambios en cualquier momento.

1.6 Framework

Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Se han convertido en la piedra angular de la moderna ingeniería del software. Entre sus características se encuentra que es una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras es una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. Entre sus objetivos se puede encontrar(19)

- Implementar rápido.
- Proceso estructurado.
- Reutilizar el código.
- Reducir costos de producción.
- Disminuir el esfuerzo en el desarrollo.
- Aprovechar las funcionalidades ya implementadas.

1.6.1 Symfony

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web, es decir está basado en la arquitectura MVC (Modelo-Vista-Controlador). Está

Capítulo 1. Fundamentación teórica

desarrollado completamente con PHP5. Brinda mayor organización del código, donde el tratamiento de errores y mantenimiento se realiza de manera sencilla.

Cuenta con un gran número de librerías, herramientas y helpers que ayudan a desarrollar una aplicación mucho más rápida. Tiene una documentación muy extensa que se actualiza con mucha frecuencia. Symfony dispone de una moderna librería que se encarga de las validaciones de acceso y seguridad. (14) Es compatible con los gestores de bases de datos, como MySQL y PostgreSQL. Se puede ejecutar tanto en plataformas *nix (Unix, Linux) como en plataformas Windows. (20)

Características de Symfony

- Fácil de instalar y configurar en la mayoría de las plataformas.
- Independiente del sistema gestor de bases de datos.
- Utiliza programación orientada a objetos.
- Es compatible con NetBeans IDE.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja.
- Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Una potente línea de comandos que facilitan generación de código, lo cual contribuye a ahorrar tiempo de trabajo.

Por las características antes expuestas se decide seleccionar Symfony en su versión 2.3.7 para el desarrollo de la presente aplicación informática.

1.6.2 Zend Framework

Es un framework que utiliza el patrón de arquitectura Modelo Vista Controlador. Una de sus características es que automatiza las tareas más específicas, como el acceso a base de datos, el filtrado de datos ingresados a la aplicación o la búsqueda en un sitio web ordenando resultados por relevancia. Es independiente de la base de datos, posee manejo propio de sesiones por usuarios y es

Capítulo 1. Fundamentación teórica

compatible con PHP5. Entre sus inconvenientes se encuentra que no posee mapeado de objetos a bases de datos relacionales (ORM), interfaz de línea de comandos para la creación y mantenimiento de aplicaciones y en la seguridad no se maneja la verificación de la salida generada en HTML por procesamiento de peticiones.(21)

1.7 Entornos de desarrollo integrado (IDE)

Un IDE es un programa informático compuesto por un conjunto de herramientas de programación. Es un entorno que ha sido empaquetado como un programa de aplicación; es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI)(22)

- **Editor de código fuente:** Editor de texto que sirve para editar el código fuente de aplicaciones informáticas.
- **Un compilador:** Es un traductor de código fuente, lo traduce a un lenguaje que sea legible para las máquinas.
- **Un depurador:** Es una aplicación que tiene como función probar y eliminar posibles errores en un programa en desarrollo.
- **Constructor de interfaz gráfica:** Herramienta que sirve para crear y diseñar las interfaces con las cuales habrá interacción entre la aplicación y el usuario.

1.7.1 NetBeans

Es un proyecto exitoso de código abierto con una gran base de usuarios y una comunidad en constante crecimiento. Es un entorno de desarrollo y una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además, un número importante de módulos para extender la herramienta. Se caracteriza por presentar mejoras en el editor de código, su instalación y actualización es más simple que otros IDE, presenta características visuales para el desarrollo web, mejoras para SOA⁵, UML y soporte para PHP. Puede ser ejecutado en sistemas operativos como Linux y Windows.(23)

⁵ SOA es una arquitectura orientada a servicios.

Capítulo 1. Fundamentación teórica

Se decide utilizar el NetBeans en su versión 7.4 para la implementación de la aplicación porque presenta un rendimiento óptimo en tiempo de ejecución y optimización de recursos. NetBeans IDE proporciona diferentes vistas de los datos de múltiples ventanas del proyecto a herramientas útiles para la creación de las aplicaciones y la gestión de manera eficiente, lo que le permite profundizar en los datos de forma rápida y sencilla. Puede ejecutarse en una gran variedad de sistemas operativos, incluyendo Windows y Linux.

1.7.2 Eclipse

Eclipse es un entorno de desarrollo integrado, de código abierto y multiplataforma. Mayoritariamente se utiliza para desarrollar lo que se conoce como "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Es una potente y completa plataforma de Programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java. Es un entorno de desarrollo integrado (IDE) en el que se encuentran todas las herramientas y funciones necesarias para el desarrollo del trabajo, recogidas además en una atractiva interfaz que lo hace fácil y agradable de usar. (24)

1.8 Sistema Gestor de Base de Datos

Es un software cuyo objetivo es proporcionar una interfaz entre la base de datos (BD), el usuario y las aplicaciones que la utilizan. Las principales funciones que debe cumplir un SGBD es la creación y mantenimiento de la BD, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.(25)

Características de los SGBD(26)

- ✓ **Abstracción de la información:** los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos.
- ✓ **Independencia:** consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven en ella.
- ✓ **Consistencia:** consiste en que todos los datos repetidos se actualicen de forma simultánea.

Capítulo 1. Fundamentación teórica

- ✓ **Seguridad:** los SGBD deben garantizar que la información almacenada en la base de datos se encuentre segura frente a usuarios malintencionados que intenten leer información privilegiada y proteger los datos de ataques deseen manipular o destruir la información.
- ✓ **Integridad:** se trata de adoptar todas las medidas necesarias para garantizar la validez de los datos almacenados.

1.8.1 PostgreSQL

Es un gestor de base de datos que incluye características de la orientación a objeto, la herencia, tipos de datos, funciones, restricciones, reglas e integridad. Es compatible con los sistemas operativos: Linux y Windows. Posee soporte de protocolo de comunicación encriptado por SSL⁶, extensiones para alta disponibilidad, nuevos tipos de índices, datos espaciales y minería de datos.

Compatible con el lenguaje PHP. Entre otras de sus características se encuentra la limpieza de la base de datos, utilidades para análisis y optimización de Querys, almacenaje especial para tipos de datos grandes y varios tipos de índices.

Algunas características (27)(28):

- Es multiplataforma.
- Extensible.

Se decide utilizar como gestor de base de datos PostgreSQL en su versión 9.2.4.1 por las diversas características antes expuestas.

1.8.2 MySQL

MySQL es un gestor de bases de datos de código abierto, tiene gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos. Utiliza como lenguaje de programación *Structured Query Language* (SQL). Es un sistema de gestión de base de datos relacional. Posee velocidad al realizar las operaciones. Algunas de sus ventajas son(29)

- Velocidad y robustez.

⁶ Protocolo criptográfico empleado para realizar conexiones seguras entre un cliente (como lo es un navegador de Internet) y un servidor (como lo son las computadoras con páginas web).

Capítulo 1. Fundamentación teórica

- Escalable.
- Acceso a la base de dato de forma simultánea.
- Aplicable a: Windows, Linux, Python, Perl, entre otras.
- Seguro en cuanto a conectividad y permisos.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está instalada la base de dato.
- Conectividad y seguridad.

1.9 Lenguaje de programación web

Desde el inicio de Internet, florecieron diferentes demandas por parte de los usuarios y se proporcionaron soluciones mediante lenguajes estáticos. Las tecnologías han ido evolucionando y con ellas nuevos problemas por resolver. Por lo que se han desarrollado lenguajes de programación dinámicos para la web, que permiten interactuar con los usuarios y utilizan sistemas de Bases de Datos.

En la actualidad existen gran variedad de lenguajes de programación web, lenguajes de programación del lado del cliente que indican al navegador donde colocar cada imagen, video o texto y de qué forma estarán colocados en la página; y los lenguajes del lado del servidor, que son ejecutados en el servidor web antes de que se envíe la página a través de Internet al cliente, permitiendo que se muestre en la misma los datos deseados por el servidor.

1.9.1 Lenguajes de Programación del lado del cliente HTML (Hyper Text Markup Language)

HTML5: lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores del tipo Internet Explorer, Opera, Firefox o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos y también de los más fáciles de aprender.(30)

Capítulo 1. Fundamentación teórica

Se decide utilizar HTML5, para la implementación de la aplicación porque es un lenguaje estático y está basado en etiquetas. Brinda la posibilidad de organizar el texto. Es compatible con todos los exploradores y fácil de aprender.

1.9.1.2 JavaScript

Java Script se utiliza principalmente para crear páginas web dinámicas. Es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos, es decir los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Este lenguaje se incluye dentro del código de una página XHTML.

En JavaScript no es obligatorio inicializar las variables, ya que se pueden declarar por una parte y asignarles un valor posteriormente. Incorpora una serie de funciones y propiedades para el manejo de las variables, de esta forma, muchas de las operaciones básicas con las variables, se pueden realizar directamente con las utilidades que ofrece JavaScript. Este lenguaje utiliza el Modelo de Objeto de Documento⁷ por sus siglas en inglés Document Object Model (DOM), que permite manipular las páginas XHTML como si fueran documentos XML (31)

1.9.2 Lenguajes de Programación del lado del servidor. PHP (Hypertext Preprocessor)

Para realizar la implementación del sistema se utilizó el lenguaje PHP, pues se trata de un tipo de lenguaje de programación de código del lado del servidor, diseñado fundamentalmente para el desarrollo web de contenido dinámico.

Entre los aspectos positivos de PHP a destacar se encuentra que su código fuente es invisible al navegador y al cliente, ya que es el propio servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Es también multiplataforma, sencillo y ofrece una gran variedad de funciones para la explicación de bases de datos sin complicaciones.(32)

⁷ Estructura de objetos que genera el navegador cuando se carga un documento y se puede alterar mediante Java Script para cambiar dinámicamente los contenidos y aspecto de la página.

Capítulo 1. Fundamentación teórica

1.10 Herramientas de Modelado

Las herramientas de modelado hacen la vida más fácil al desarrollador de software, porque reducen la complejidad del sistema al abstraerlo de detalles innecesarios. Se representan por notaciones en su mayoría visuales, que intentan representar un sistema de software a un nivel mucho más alto que los lenguajes de programación, representándolo en forma más intuitiva para personas sin especialización en Informática.

UML, es el lenguaje de modelado de software más usado en el mundo. Actualmente casi todas las herramientas CASE (*Computer Aided Software Engineering*, lo que en español significa Ingeniería de Software Asistida por Computadora) y de desarrollo de software han adoptado al UML, como lenguaje de modelado. En su creación y desarrollo participaron estudiosos e investigadores de gran prestigio y fue apoyado por la mayoría de las instituciones más importantes del mundo de la Informática.(33)

1.10.1 Rational Rose Enterprise

Esta herramienta cubre todo el ciclo de vida de un proyecto. Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción.(34) Permite acelerar el desarrollo de las aplicaciones con código generado a partir de modelos visuales mediante el lenguaje UML (Unified Modeling Language). Algunas características de la herramienta:

- Proporciona prestaciones de modelado visual para desarrollar muchos tipos de aplicaciones de software.
- Contiene herramientas web y XML⁸ para el modelado de aplicaciones web.
- Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- Brinda la posibilidad de analizar la calidad del código.

⁸ Sistema estándar de codificación de información organizada.

Capítulo 1. Fundamentación teórica

1.10.2 Visual Paradigm-UML

Entre las herramientas CASE más usadas actualmente se encuentra Visual Paradigm. Es una herramienta UML profesional considerada en el mercado mundial como muy completa y fácil de usar, es multiplataforma y proporciona excelentes facilidades de interoperabilidad con otras aplicaciones, la misma soporta el ciclo de vida completo del desarrollo de software. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, y a un menor coste. Tiene disponibilidad de múltiples versiones para cada necesidad y recurso de integrarse en los principales IDEs. Apoya los estándares más recientes de las notaciones de Java y de UML. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros(35)

Se escogió como herramienta de modelado Visual Paradigm en su versión 8.0, puesto que presenta varias ventajas, tales como: emplea una rápida respuesta con poca memoria utilizando moderadamente los tiempos del procesador, lo que le permite manejar grandes y complicadas estructuras de un proyecto en una forma muy eficiente. A diferencia del Rational Rose que es un software propietario mientras que el Visual Paradigm es un software libre.

1.11 Conclusiones Parciales

Luego del estudio y análisis del objeto de investigación del presente trabajo de diploma, apoyado en los métodos de la investigación científicos definidos, se concluye lo siguiente:

Los métodos científicos y las técnicas de recopilación de datos empleadas en el estudio que se presenta, permitieron desarrollar la teoría que sustenta esta investigación. La solución será desarrollada utilizando la metodología XP para guiar el proceso de desarrollo. Como herramienta de modelado Visual Paradigm. Como lenguaje del lado de servidor PHP5, HTML5 como lenguaje de descripción de hipertexto, como framework de desarrollo se escoge Symfony en su versión 2. Como servidor web Apache en su versión 2, PostgreSQL como gestor de base de datos. Como IDE de desarrollo para el sistema, NetBeans en su versión 7.4.

Capítulo 2. Características del sistema

Capítulo 2. Características del sistema

Introducción

En el presente capítulo tiene el objetivo de describir el proceso de construcción de la solución, así como los roles que interactúan con el mismo. Además, se muestran los principios, prácticas y técnicas que sirven de guía para el desarrollo de la presente investigación, entre las que se encuentran: historias de usuarios (HU), plan de iteraciones, plan de entregas y tarjetas CRC (Clases, Responsabilidad y Colaboración). De igual forma se hace una descripción del desarrollo del sistema.

2.1 Descripción del flujo actual del proceso

A continuación se describe y explica el procedimiento de la evaluación, el cual consta de 3 etapas que deben ser ejecutadas de manera secuencial comenzando por la primera:

1. Inicio y Planificación
2. Ejecución
3. Cierre

El procedimiento propuesto está basado en la norma ISO/IEC 9126, estándar que propone una guía para evaluar la calidad de los productos. Este estándar describe las características con las que debe cumplir el software, pero no dice como evaluarlas. El procedimiento que se describe a continuación es una propuesta de cómo evaluar la Manipulación de los Entornos de Prueba en el Laboratorio Industrial de Pruebas de Software.

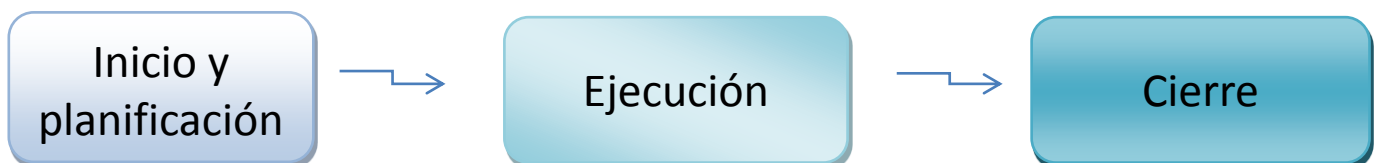


Figura 2_Etapas de procedimiento de evaluación.

Capítulo 2. Características del sistema

2.1.1 Descripción del subproceso: Inicio y Planificación

La etapa Inicio y Planificación consta de cuatro actividades que tienen como objetivo principal recolectar la información inicial necesaria para realizar la manipulación de los Entornos de Pruebas. Para efectuar esta etapa la solicitud de evaluación de un producto debe ser aceptada por Calisoft y asignada a un especialista que será el coordinador de la evaluación durante todo el proceso de pruebas. De manera general esta etapa se define por la obtención de información referente al componente a evaluar y en ella quedan constituidos documentos necesarios para las etapas posteriores. A continuación se muestran y describen las actividades definidas en esta primera etapa.

2.1.1.2 Descripción textual del subproceso: Inicio y Planificación.

Capítulo 2. Características del sistema

Tabla 1_Descripción textual del subproceso: Inicio y Planificación.

IPP-0000:2013 Inicio y Planificación		
Criterios de entrada	La solicitud del servicio debe estar aceptada.	
Criterios de salida	Debe quedar aceptado el Acuerdo de Entorno de Prueba en la Reunión de Inicio.	
No.	Descripción	Salida
1	<p>Planificación con el Coordinador de la Evaluación.</p> <ul style="list-style-type: none"> ✓ Cuando los Coordinadores de la Evaluación del producto reciben la asignación de un producto a evaluar deben asistir al Laboratorio Industrial de Pruebas de Software a rectificar que todos los requisitos del entorno de prueba que se solicitan por el Equipo de Desarrollo pueden ser soportadas en el entorno de virtualización. ✓ El Especialista del Laboratorio Industrial de Pruebas de Software asesora al Coordinador de la Evaluación y le aconseja sobre las características del Entorno de Prueba de la solicitud. Además le recuerda que le ha de informar al Equipo de Desarrollo que los requisitos a solicitar deben ser los básicos para que funcione el producto, no los del futuro Entorno de despliegue del mismo. ✓ El Coordinador de la evaluación debe dirigirse a los especialistas del Laboratorio Industrial de Pruebas de Software encargados de la planificación de la actualización en los entornos de prueba para concebir un encuentro con las personas del proyecto para que realicen la instalación del producto. <p>Esta planificación se recoge en el documento <u>Registro de Actualización de Servidores 2013-2014</u> que realizan los encargados de la planificación del Laboratorio Industrial de Pruebas de Software.</p>	<ul style="list-style-type: none"> ✓ Registro de actualización de servidores(actualizado)

Capítulo 2. Características del sistema

2	<p>Recibir Acuerdos de Entornos de Pruebas de Reunión de Inicio.</p> <ul style="list-style-type: none"> ✓ El Coordinador del Servicio debe hacer entrega del Acuerdo de Entorno de Prueba al especialista del Laboratorio Industrial de Pruebas de Software al concluir la Reunión de Inicio en caso de que las características que estaban en la solicitud, fueran las correctas. ✓ El Especialista del Laboratorio industrial de Pruebas de Software revisa que todos los datos estén correctos. ✓ El Coordinador de la Evaluación en caso que por alguna razón se haya modificado o no planificado con anterioridad la fecha en la que el cliente debe venir a instalar su producto, debe planificarla al finalizar la Reunión de Inicio. Lo cual ocurre en muy pocos casos pues siempre se trata en las reuniones de inicio tenerlo todo bien asegurado para evitar atrasos. 	<ul style="list-style-type: none"> ✓ Acuerdo de Entorno de Prueba Aceptado (creado)
3	<p>Realizar y Enviar Documentación.</p> <ul style="list-style-type: none"> ✓ Luego de recibir los Acuerdos de Entornos de Pruebas el Especialista del Laboratorio Industrial de Pruebas de Software realiza la creación de 2 documentos: <ul style="list-style-type: none"> -<u>Acuerdos de Entornos de Pruebas</u> que se lleva como constancia en el repositorio para tener archivados todos los Acuerdos de Entornos de Pruebas tomados en las Reuniones de Inicio con los datos que se aceptaron en las mismas. En caso que por alguna razón no se haya podido llegar a un acuerdo en dichas características se detiene la evaluación del producto 1 semana hasta que el cliente actualice la solicitud. -<u>Entornos de prueba a instalar</u> que se le hace llegar a la Jefa del Laboratorio Industrial de Pruebas de Software para que constate los datos de los Entornos de Prueba que se instalarán, el especialista que se encargará de ello y el día que están planificados para que venga a instalar el cliente su producto al Laboratorio Industrial de Pruebas de Software. 	<ul style="list-style-type: none"> ✓ Archivos de Acuerdos de Entornos de Prueba (creado) ✓ Notificación de Entornos de Prueba a instalar (enviado)
4	<p>Asignar Especialista.</p> <ul style="list-style-type: none"> ✓ El Especialista del Laboratorio Industrial de Pruebas de Software envía el 	<ul style="list-style-type: none"> ✓ Notificación de asignación

Capítulo 2. Características del sistema

	documento de <u>Entornos de prueba a instalar</u> al especialista que se encargará del montaje del Entorno de Pruebas.	de especialista (enviado)
--	--	---------------------------

2.1.2 Descripción del subproceso: Ejecución

En la etapa de Ejecución es en la que se realiza todo el trabajo en los Entornos de Pruebas. En esta se realizan la instalación y también actualizaciones durante todo el proceso de evaluación de producto e intervienen tanto el coordinador de la evaluación como el equipo de proyecto además de los especialistas del Laboratorio Industrial de Pruebas de Software.

La correcta realización de esta etapa determinará la calidad de evaluaciones en el departamento de Pruebas, por ello es de gran importancia realizar cada actividad de manera minuciosa y detallada. A continuación se muestran y describen las actividades definidas en esta etapa.

Capítulo 2. Características del sistema

2.1.2.1 Descripción textual del subproceso: Ejecución

Tabla 2 Descripción textual del subproceso: Ejecución.

Ejecución		
Criterios de entrada	Se posee toda la documentación solicitada al proyecto.	
Criterios de salida	Queda definida la estrategia de la evaluación y montado el entorno de pruebas.	
No.	Descripción	Salida
1	<p>Montar Entorno de Prueba</p> <p>En el Laboratorio Industrial de Pruebas de Software hay especialistas que se encargan de llevar a cabo las instalaciones de los entornos de pruebas, donde se montarán los productos a probar en el Laboratorio, basados en los datos de los Acuerdos de Entornos de Prueba acordados previamente en la reunión de inicio de dicho proyecto y de la Solicitud de Servicio del mismo.</p> <p>A este entorno de prueba luego de instalado se le crea un Snapshot, el cual se va a encargarse de salvar momentos importantes de este entorno de prueba por si en el transcurso de la misma aparecen problemas, se pueda retomar el entorno de prueba en un punto importante anterior y evitar empezar desde cero. Cuando el entorno de pruebas está listo para que el proyecto vaya a realizar su instalación es el primer momento en que se crea un Snapshot ya que estaría listo con todo lo necesario para el montaje del producto a probar.</p>	<p>✓ Entorno de prueba (creado)</p> <p>✓ Snapshot del Entorno de Prueba Recién Montado (creado)</p>
2	<p>Actualizar Documentos.</p> <p>Para que quede constancia de las características del entorno de pruebas instalado se realiza la actualización del Registro de Aplicaciones que contiene los datos de todos los entornos de pruebas que se instalan en el Laboratorio industrial de Pruebas de Software.</p>	<p>✓ Registro de instalación de servidores (actualizado)</p>
3	<p>Notificar al Coordinador de la Evaluación</p> <p>El Especialista del Laboratorio Industrial de pruebas de Software encargado de realizar los documentos Acuerdos de Entornos de Pruebas y Entornos de Pruebas a Instalar le notifica vía correo al Coordinador de</p>	<p>✓ Notificación de Entorno de Prueba listo (enviado)</p>

Capítulo 2. Características del sistema

	la Evaluación que el Entorno de Prueba está preparado y listo para la instalación.	
4	<p>Instalar Producto y Actualizar Documentación</p> <ul style="list-style-type: none"> ✓ Luego del montaje del entorno de prueba el proyecto debe venir a instalar su producto en el mismo, ya sea el día que se haya planificado en la reunión de inicio o enviado por el especialista luego de la misma. ✓ Cuando el proyecto va al Laboratorio Industrial de Pruebas de Software para la instalación inicial de su producto en el entorno de pruebas, van a haber especialistas que se encargaran de atenderlos, los cuales son las personas encargadas, con los permisos apropiados para trabajar y manipular los entornos de pruebas y además se encargan de brindarle apoyo a las personas que vienen del proyecto, a los cuales nunca se les da permisos para evitar que puedan cambiar o realizar alguna acción indebida en el entorno de prueba, y el equipo de desarrollo además debe firmar la libreta de Instalación de Servidores. ✓ El especialista del Laboratorio Industrial de Pruebas de Software luego de la instalación del producto realiza la creación de un Snapshot y realiza la actualización del Registro_de_Actualización_de_Servidores_2013-2014 para dejar plasmado la acción que el equipo de desarrollo realizó sobre el Entorno de Prueba o si no se presentó. ✓ Luego de la instalación del producto, el entorno de prueba donde queda instalado no se vuelve a utilizar directamente, los especialistas del Laboratorio Industrial de Pruebas de Software tienen solo permisos para prender y apagar los mismos para que los coordinadores de las pruebas puedan realizar las pruebas necesarias y solicitadas por el cliente al producto. 	<ul style="list-style-type: none"> ✓ Entorno de Prueba (Instalado). ✓ Snapshot del Entorno de Prueba Recién Instalado. ✓ Registro de planificación de actualización de servidores (actualizado)
5	<p>Planificar Actualización.</p> <ul style="list-style-type: none"> ✓ Al concluir cada iteración de las pruebas el coordinador de la misma debe dirigirse a los especialistas del Laboratorio Industrial de Pruebas de Software encargados de la planificación de la actualización en los 	<ul style="list-style-type: none"> ✓ Registro de planificación de actualización

Capítulo 2. Características del sistema

	<p>entornos de prueba para concebir un encuentro con las personas del proyecto para que realicen la actualización del producto para una nueva iteración. Esta planificación se recoge en el documento <u>Registro de Actualización de Servidores 2013-2014</u> que realizan los encargados de la planificación del Laboratorio Industrial de Pruebas de Software.</p>	<p>de servidores (actualizado)</p>
6	<p>Actualizar Producto.</p> <p>Cuando el proyecto va al Laboratorio Industrial de Pruebas de <i>Software</i> para la actualización de su producto en el entorno de pruebas, van a haber especialistas que se encargaran de atenderlos, los cuales son las personas encargadas, con los permisos apropiados para trabajar y manipular los entornos de pruebas y además se encargan de brindarle apoyo a las personas que vienen del proyecto, a los cuales nunca se les da permisos para evitar que puedan cambiar o realizar alguna acción indebida en el entorno de prueba, y el equipo de desarrollo además debe firmar la libreta de Instalación de Servidores.</p> <p>El especialista del Laboratorio Industrial de Pruebas de Software luego de la actualización del producto realiza la creación de un Snapshot y realiza la actualización del <u>Registro de Actualización de Servidores 2013-2014</u> para dejar plasmado la acción que el equipo de desarrollo realizó sobre el Entorno de Prueba o si no se presentó.</p>	<p>✓ Entorno de Prueba (Actualizado)</p> <p>✓ Snapshot del Entorno de Prueba (Actualizado)</p> <p>✓ Registro de planificación de actualización de servidores (actualizado)</p>
7	<p>¿Terminó la Evaluación?</p> <p>Si cuando el proyecto viene a instalar te informa que ya concluyó la evaluación del producto y vienen a actualizar para la versión final entonces se culmina la etapa de Ejecución, en caso contrario se continúa el procedimiento.</p>	

2.1.3 Descripción del subproceso: Cierre

En la etapa de Cierre se realizan los últimos procedimientos en los entornos de prueba, la exportación y eliminación del mismo de los servidores del Laboratorio industrial de Pruebas. A continuación se muestran y describen las actividades definidas en esta etapa.

Capítulo 2. Características del sistema

2.1.3.1 Descripción textual del subproceso: Cierre.

Tabla 3_Descripción textual del subproceso: Cierre.

Evaluación y Cierre		
Criterios de entrada	Se posee toda la documentación solicitada al proyecto.	
Criterios de salida	Queda definida la estrategia de la evaluación y montado el entorno de pruebas.	
No.	Descripción	Salida
1	<p>Enviar Parte.</p> <ul style="list-style-type: none"> ✓ Luego que los productos son liberados en el Departamento de Pruebas se le envía al Jefe del Laboratorio Industrial de Pruebas de Software el parte semanal donde aparecerá el mismo y este lo envía al especialista encargado de exportar los entornos de prueba correspondientes a los productos liberados. 	Notificación del Parte semanal del DEPSW (enviado)
2	<p>Exportar Entornos de Pruebas.</p> <ul style="list-style-type: none"> ✓ El especialista del Laboratorio Industrial de Pruebas de Software luego de recibir la notificación del Jefe de Laboratorio Industrial de Pruebas de Software exporta el Entorno de Prueba y luego de hacerlo el mismo se mantiene guardado con su última versión. ✓ A continuación notifica al especialista que se encarga de eliminar los Entornos de Prueba para que lo elimine del servidor. ✓ Actualiza el <u>Registro de Máquinas Exportadas y Eliminadas</u>. Documento que se lleva como control mensual de las máquinas exportadas y eliminadas, para llevar un mejor control de las mismas y que sirva para cualquier parte o reporte que se desee realizar. 	<ul style="list-style-type: none"> ✓ Registro de Maquinas Exportadas y Eliminadas (modificado) ✓ Notificación de máquinas exportadas (enviado)
3	<p>Eliminar Entornos de Pruebas.</p> <ul style="list-style-type: none"> ✓ El Especialista del Laboratorio Industrial de Pruebas de Software encargado, luego de recibir la notificación elimina el Entorno de Prueba del producto liberado. ✓ Actualiza el <u>Registro de Máquinas Exportadas y Eliminadas</u>. 	<ul style="list-style-type: none"> ✓ Registro de Máquinas Exportadas y Eliminadas (modificado)

Capítulo 2. Características del sistema

✓	Actualiza el <u>Registro de Aplicaciones</u> .	✓	Registro de Aplicaciones (modificado)
---	--	---	---------------------------------------

2.2 Propuesta de solución

Como propuesta de solución que fundamenta esta investigación se plantea la creación de un sistema desarrollado mediante el framework Symfony2, que permita gestionar toda la información referente a los entornos de prueba del Laboratorio Industrial de Pruebas de Software de CALISOFT, el mismo será un sistema web que admitirá un manejo adecuado de toda la información y la gestión de los procesos planificar, gestionar y registrar.

Las funcionalidades que el sistema debe cumplir son las siguientes:

1. Gestionar usuarios.

- 1.1. Insertar usuarios.
- 1.2. Modificar usuarios.
- 1.3. Eliminar usuarios.
- 1.4. Visualizar datos de los usuarios.

2. Autenticar usuarios.

3. Gestionar solicitud de instalación de Entornos de Prueba (EP).

- 3.1 Crear solicitud de instalación de EP.
- 3.2 Modificar solicitud de instalación de EP.
- 3.3 Eliminar solicitud de instalación de EP.
- 3.4 Mostrar listado de solicitudes de instalación de EP.
- 3.5 Aceptar solicitud de instalación de EP.
- 3.6 Rechazar solicitud de instalación de EP.
- 3.7 Enviar notificación de solicitud de instalación de EP.

Capítulo 2. Características del sistema

3.8 Asignar solicitudes al rol especialista.

4. Gestionar entorno de prueba real.

4.1 Agregar entorno de prueba real.

4.2 Modificar entorno de prueba real.

4.3 Visualizar entorno de prueba real.

4.4 Eliminar entorno de prueba real.

5. Gestionar roles.

5.1 Agregar rol.

5.2 Modificar rol.

5.3 Visualizar rol.

5.4 Eliminar rol.

6. Gestionar proyectos.

6.1 Agregar proyecto.

6.2 Modificar proyecto.

6.3 Visualizar proyecto.

6.4 Eliminar proyecto.

7. Gestionar planificación.

7.1 Agregar planificación.

7.2 Modificar planificación.

7.3 Visualizar planificación.

7.4 Eliminar planificación.

8. Gestionar calendarios.

8.1 Gestionar turnos.

Capítulo 2. Características del sistema

8.1.1 Agregar turnos.

8.1.2 Modificar turnos.

8.1.3 Visualizar turnos.

8.1.4 Eliminar turnos.

9. Gestionar puestos de trabajo.

9.1 Agregar puesto de trabajo.

9.2 Modificar puesto de trabajo.

9.3 Visualizar puesto de trabajo.

9.4 Eliminar puesto de trabajo.

10. Gestionar incidencias.

10.1 Agregar incidencias.

10.2 Modificar incidencias.

10.3 Visualizar incidencias.

Las características definidas con las cuales la PC cliente y la PC servidor deben contar para la gestión de la información son las siguientes:

- En la PC cliente debe ejecutarse cualquier navegador.
- El sistema podrá ser ejecutado en la PC cliente con los sistemas operativos: Windows 98 o superior o Linux.
- La PC servidor debe tener instalado PHP5, Servidor web Apache versión 2.0, y PostgreSQL.
- La herramienta debe ser sencilla, fácil de emplear y amigable para el usuario.
- El sistema estará orientado al LIPS (Laboratorio Industrial de Pruebas de Software) del Departamento de Evaluación de Productos de CALISOFT.
- Debe estar disponible las 24 horas.
- Documentación necesaria para el uso o desarrollo de las funcionalidades de la herramienta.

Capítulo 2. Características del sistema

- El sistema debe proteger la información que se maneja.
- Permitirá la autenticación de forma necesaria para entrar a la aplicación.
- Garantizará que la información sea vista por aquellas personas que tengan los permisos para esto.
- Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario.
- El sistema debe permitirle al usuario conectarse desde cualquier PC de la red.

2.3 Usuarios relacionados con el sistema

Uno de los puntos fundamentales que se debe tener en cuenta al desarrollar un sistema, es la especificación de la audiencia a la que está dirigido el mismo. Los roles y las funciones que realizarán estos usuarios en el sistema, se detallan a continuación:

Tabla 4_Usuarios relacionados con el sistema.

Personal	Descripción
Administrador	Responsable de gestionar la información del sistema y de administrar todas las cuentas de los usuarios.
Usuario	Responsable de gestionar la información del entorno de pruebas, y aceptar o rechazar las solicitudes, posee los mismos permisos del Administrador, puede modificar el rol de todos los usuarios, eliminar usuarios, excepto al Administrador.

2.4 Propuesta de arquitectura

En XP no se destaca la definición temprana de una arquitectura constante para el sistema. Dicha arquitectura se asume evolutiva y los posibles inconvenientes que se generarían por no contar con ella explícitamente en el inicio del proyecto se determinan con la existencia de una metáfora. Las metáforas son una de las prácticas más utilizadas que propone la metodología XP, que se llevan a cabo durante un proyecto, y que hacen que ésta metodología sea tan utilizada y efectiva.(36)

Capítulo 2. Características del sistema

El sistema se define en base a una metáfora o conjuntos de metáforas entre el cliente y los programadores. La metáfora es utilizada por los programadores para describir a grandes rasgos la estructura del sistema de manera tal que pueda ser comprendida por todos los integrantes.(36)(37)

Una vez que la metáfora es creada, se pasa al establecimiento de clases, métodos y relaciones primordiales del sistema. Además, de servir como fuente de comunicación entre los clientes y los programadores sirve para poder mantener el diseño del sistema lo más simple posible.

Los conceptos relacionados con el término metáfora son diversos, entre ellos se encuentra el emitido por Kent Beck, quien es considerado el padre de XP y autor de los libros más influyentes sobre el tema, en su libro *Extreme Programming Explained*: “la metáfora del sistema es algo que todos los clientes y programadores entienden sobre cómo funciona el sistema. Las metáforas ayudan con la abstracción y modelado del sistema...”(37)

2.4.1 Patrón Modelo Vista Controlador (MVC)

Christopher Alexander, un reconocido arquitecto inglés dice que “cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, así como la solución a ese problema, de tal modo que se pueda aplicar esta solución un millón de veces, sin hacer lo mismo dos veces”. (38)

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC, que está formado por tres niveles:

- El Modelo: representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La Vista: transforma el modelo en una página web que permite al usuario interactuar con ella.
- El Controlador: se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

Capítulo 2. Características del sistema

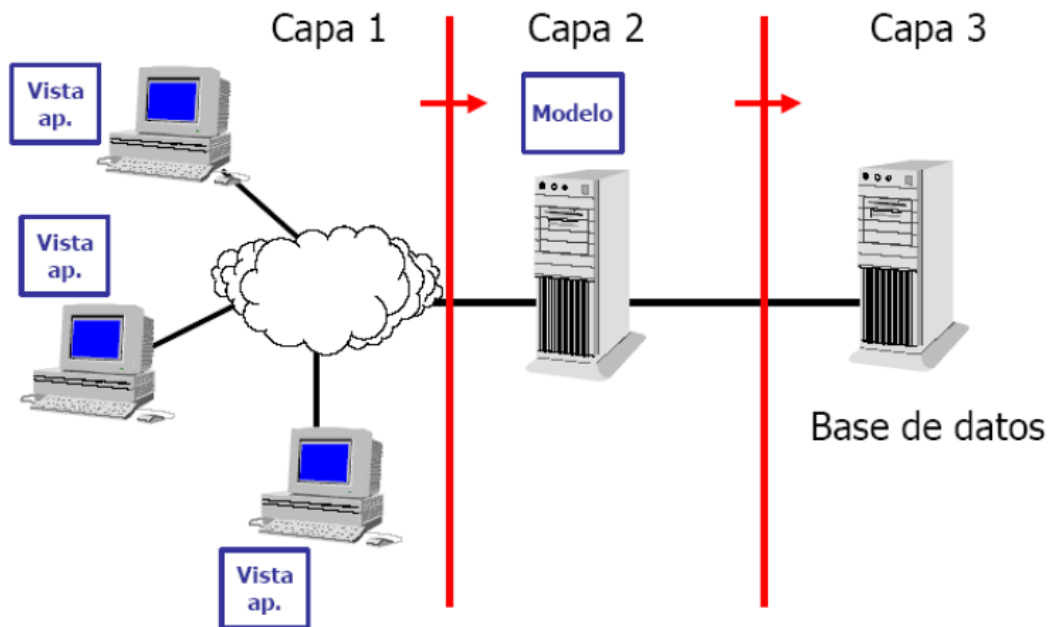


Figura 3_Patrón MVC

El modelo, contiene la lógica de negocio de la aplicación, la vista muestra al usuario la información que éste necesita y el controlador recibe e interpreta la interacción del usuario, actuando sobre modelo y vista de manera adecuada para provocar cambios de estado en la representación interna de los datos, así como en su visualización.

2.5 Planificación

Es la primera fase de la metodología XP donde el cliente establece la prioridad de cada historia de usuario (HU), y respectivamente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se establece un cronograma en conjunto con el cliente. Además, precisa el Plan de entrega, se aconseja la programación en parejas pues aumenta la productividad y la calidad del software desarrollado y se proponen las reuniones diarias donde los programadores muestran sus problemas, soluciones e ideas de forma conjunta.(39)

Capítulo 2. Características del sistema

2.5.1 Historia de Usuario (HU)

Las historias de usuario son utilizadas como herramienta para dar a conocer los requerimientos del sistema al equipo de desarrollo. Son escritas por el cliente con asistencia de los diseñadores, en un lenguaje común y sin detalles de elementos de programación. Además, los clientes son los que forman la prioridad de las mismas. Las historias de usuario sólo exponen la forma de una tarea a realizarse.

Las HU descritas están compuestas por los siguientes campos:

- **Número:** contiene el identificador de la HU.
- **Nombre:** contiene el nombre que identifica a la HU.
- **Usuario:** roles que pueden acceder a la HU.
- **Iteración Asignada:** número de la iteración en que se va a implementar la HU.
- **Puntos Estimados:** estimación hecha por el equipo de desarrollo del tiempo de duración.

Cuando el valor es 1 equivale a una semana ideal de trabajo. En la metodología XP está definida una semana ideal como 5 días hábiles trabajando 40 horas, es decir, 8 horas diarias.

- **Riesgo en Desarrollo:** nivel de riesgo que posee la HU en caso de no realizarse (Alto, Medio o Bajo).
- **Puntos Reales:** tiempo real en el que se realizó la HU.
- **Descripción:** breve descripción del proceso que define la HU.
- **Observaciones:** comentarios aclaratorios relacionados con la HU.

A continuación se muestran algunas Historias de Usuarios del sistema a realizarse. Las demás HU se incluyen en el **Anexo 1**.

Tabla 5_HU.Gestionar usuarios.

Historia de Usuario	
Número: 1	Nombre: Gestionar usuarios.
Usuario: Administrador.	

Capítulo 2. Características del sistema

Puntos Estimados: 1	Iteración Asignada: 1
Riesgo en Desarrollo: Alta	Puntos Reales: 1
<p>Descripción: Permite administrar las cuentas de los usuarios del sistema, dando la posibilidad de insertar, modificar, eliminar y visualizar los datos de los usuarios. A cada usuario se le darán los permisos necesarios en dependencia de lo que realizará en el sistema. Los datos necesarios para crear un usuario son: nombre, apellidos, usuario y contraseña.</p>	
<p>Observación: El gestionar incluye insertar, modificar, visualizar y eliminar los datos de un usuario; en el sistema intervendrá 2 tipos de usuario, el administrador y los especialistas. A cada usuario se le darán los permisos necesarios en dependencia de lo que realizará en el sistema.</p>	

Tabla 6_HU.Gestionar las solicitudes de instalación de Entornos de Prueba.

Historia de Usuario	
Número: 3	Nombre: Gestionar las solicitudes de instalación de Entornos de Prueba (EP).
Usuario: Administrador, Usuario.	
Puntos Estimados: 1	Iteración Asignada: 1
Riesgo en Desarrollo: Alta	Puntos Reales: 1.3
<p>Descripción: Permite administrar las solicitudes de entornos de pruebas en el sistema, dando la posibilidad de aceptar, rechazar, crear, modificar y eliminar las solicitudes. Los datos requeridos para crear una solicitud de entorno de prueba son: etapa, sistema operativo, tipo sistema operativo, capacidad de disco duro, tipo de servidor, versión o distribución, arquitectura, capacidad de memoria RAM.</p>	
Observación:	

Tabla 7_HU.Gestionar entorno de prueba real (EP).

Historia de Usuario	
Número: 4	Nombre: Gestionar entorno de prueba real.
Usuario: Administrador, Usuario.	

Capítulo 2. Características del sistema

Puntos Estimados: 1	Iteración Asignada: 3
Riesgo en Desarrollo: Alta	Puntos Reales: 1
Descripción Una vez creada la solicitud, podrá modificarse el entorno de prueba y todos los datos de la misma, y agregarle tantos entornos de pruebas como sean necesario.	
Observación:	

Tabla 8_HU.Gestionar roles.

Historia de Usuario	
Número: 5	Nombre: Gestionar roles.
Usuario: Administrador.	
Puntos Estimados: 0.2	Iteración Asignada: 4
Riesgo en Desarrollo: Baja	Puntos Reales: 0.2
Descripción: El administrador será el encargado de crear los distintos roles, y dar los distintos permisos a los usuarios.	
Observación: El sistema contará con dos tipos de roles, administrador, y usuario, el administrador será el encargado de dar los permisos a los usuarios que interactúen en el sistema.	

Tabla 9_HU.Gestionar proyectos.

Historia de Usuario	
Número: 6	Nombre: Gestionar proyectos.
Usuario: Administrador, Usuario.	
Puntos Estimados: 0.4	Iteración Asignada: 4
Riesgo en Desarrollo: Media	Puntos Reales: 0.3
Descripción: Permite crear, listar, ver, editar uno o varios proyectos. Un proyecto debe contener los siguientes datos: nombre del proyecto, nombre del producto, prioridad, cliente, comentario, número de solicitud, versión, clasificación, artefactos, y si presenta o no entorno de prueba.	

Capítulo 2. Características del sistema

Observación: Los usuarios tendrán derecho a visualizar y realizar modificaciones solo en los proyectos creados por ellos, solo el administrador podrá ver todos los proyectos creados, y realizarle modificaciones.

2.5.2 Estimación de esfuerzo por HU

Después de definir las HU, los desarrolladores crean una planificación del tiempo que requieren para implementarlas y así establecer el tiempo total para el desarrollo de la aplicación. Un punto, equivale a una semana ideal de programación. Las historias generalmente van de 1 a 3 puntos. (40)

Tabla 10_Estimación de esfuerzo por HU.

Número de HU	Historia de usuario	Puntos estimados (semanas)
1	Gestionar usuarios.	1
2	Autenticar usuarios.	1
3	Gestionar solicitud de instalación de Entorno de prueba (EP).	1
4	Gestionar entorno de prueba real.	1
5	Gestionar roles.	0.2
6	Gestionar proyectos.	0.3
7	Gestionar planificación.	1
8	Gestionar calendarios.	1
9	Gestionar turnos.	1
10	Gestionar puestos de trabajo.	0.2
11	Gestionar Incidencias.	0.3

2.5.3 Plan de iteraciones

En la metodología XP la creación del sistema se divide en etapas para facilitar su realización. Por lo general, los proyectos constan de 3 etapas en adelante, las cuales toman el nombre de iteraciones, de

Capítulo 2. Características del sistema

ahí se adquiere el nombre de metodología iterativa. La duración ideal de una iteración es de 1 a 3 semanas. Para cada iteración se precisa un conjunto de HU a implementar, donde se obtiene como resultado la entrega de una versión funcional del producto. De acuerdo a las características propias del presente proyecto el mismo es dividido en 3 iteraciones, a continuación se exponen los detalles de cada una de ellas.(40)

2.5.4 Plan de duración de las iteraciones

En el plan de duración de las iteraciones se especifica detalladamente el orden de desarrollo de las HU dentro de cada iteración así como la estimación completa de dicha iteración.

Tabla 11_Plan de duración de las iteración.

Iteración	Historia de usuario	Duración(semanas)
1	Gestionar usuarios. Autenticar usuarios. Gestionar solicitud de instalación de Entornos de Prueba (EP).	3
2	Gestionar entorno de prueba real. Gestionar roles. Gestionar proyectos. Gestionar planificación.	3
3	Gestionar puestos de trabajo. Gestionar incidencias. Gestionar calendarios Gestionar turnos.	3

2.5.5 Plan de entregas

Mediante el cronograma de entregas se establece cuales HU son agrupadas para conformar una entrega y el orden de las mismas. Este cronograma se realiza en función de las estimaciones de tiempos de desarrollo realizadas.

- El final de la primera iteración se corresponde con la cuarta semana de febrero.

Capítulo 2. Características del sistema

- El final de la segunda iteración se corresponde con la cuarta semana de marzo.
- El final de la tercera iteración se corresponde con la cuarta semana de abril.

Tabla 12_Plan de entregas.

Historia de usuario	Primera iteración	Segunda iteración	Tercera iteración
Gestionar usuarios	V 1.0	Finalizado	Finalizado
Autenticar usuarios.	V 1.0	Finalizado	Finalizado
Gestionar solicitud de instalación de Entorno de Prueba (EP).	V 1.0	Finalizado	Finalizado
Gestionar entorno de prueba real.	–	V 1.0	Finalizado
Gestionar roles.	–	V 1.0	Finalizado
Gestionar proyectos.	–	V 1.0	Finalizado
Gestionar planificación.	–	V 1.0	Finalizado
Gestionar turnos.	–	–	V 1.0
Gestionar puestos de trabajo.	–	–	V 1.0
Gestionar calendarios.	–	–	V 1.0
Gestionar incidencias.	–	–	V 1.0

Capítulo 2. Características del sistema

2.6 Diseño del sistema

En esta fase la metodología XP sugiere que hay que conseguir diseños simples y sencillos, para lograr diseños fácilmente entendibles que a la larga costará menos tiempo y esfuerzo desarrollar.

Nunca se debe añadir funcionalidad extra al programa aunque se piense que en un futuro será utilizada. Se considera un desperdicio de tiempo y recursos. A la hora de utilizar códigos ya creados, se propone refactorizar que no es más que revisar de nuevo estos códigos para procurar optimizar su funcionamiento. Esto evita generar código completamente inestable y muy mal diseñado.

Un prototipo es una visión preliminar del sistema futuro, es un modelo operable, sencillamente ampliable y modificable, que posee todas las características que hasta el momento debe tener el sistema. En este trabajo se propone un prototipo de interfaz de usuario, que constituye el primer punto de partida dirigida a la interfaz inicial que presentará el sistema para la gestión de los entornos de prueba del LIPS. A continuación se muestra, de forma esquematizada cómo será la estructura principal del sistema.

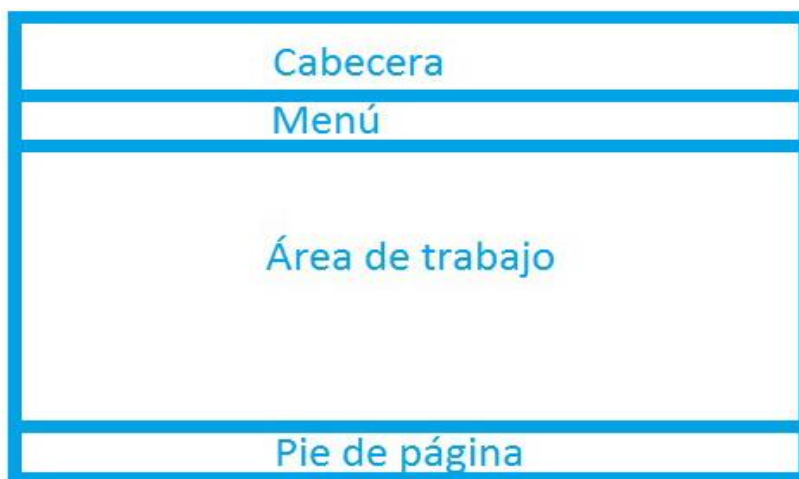


Figura 4_Diseño de estructura global del sistema.

Como se muestra en la figura, la estructura global de los elementos que componen al sistema existen cuatro zonas diferenciadas:

Capítulo 2. Características del sistema

- Cabecera: contendrá un identificador (banner) que le permitirá al usuario obtener, de forma fácil y rápida la información referente al sitio en el que se encuentra. La razón por la que se coloca esta información en la parte superior es porque la identificación del sitio lo representa y como tal, ocupa el lugar más alto en la jerarquía lógica del mismo.
- Menú: es el siguiente elemento de navegación que se le muestra al usuario del sistema, en esta sección el usuario encontrará elementos de navegación que le permitan acceder a las distintas secciones.
- Área de trabajo: es la zona donde el usuario realiza las modificaciones del sistema.
- Pie de página: mostrará un pequeño apunte informativo sobre el sistema.

2.6.1 Tarjetas CRC

En la metodología XP, no se requiere de la representación del sistema mediante diagramas de clases utilizando notación UML, en su lugar usa tarjetas CRC (Clase-Responsabilidad-Colaboración). Una tarjeta CRC no es más que una ficha de papel o cartón que representa una entidad del sistema.

Establece 3 dimensiones, las cuales identifican el rol de un objeto en análisis y/o diseño: nombre de la clase, responsabilidades y colaboraciones.

- **Una clase** representa a una colección de objetos similares.
- **Una responsabilidad** es aquello que la clase sabe o hace. Una clase puede cambiar el valor de lo que sabe pero no puede cambiar el valor de lo que saben otras clases. Algunas veces una clase tiene una responsabilidad que cumplir pero no tiene toda la información para hacerlo. Esto hace que deba interactuar con otras clases para obtener su colaboración.
- **Los colaboradores** toman una de dos formas: un pedido de información o un pedido de que se realice una operación.(41)

Tabla 13_Tarjeta CRC.

Tarjeta CRC	
Nombre del proyecto: Gestión de Solicitud de Prueba	
Responsabilidades	Colaboradores
Usuario	loginAction()

Capítulo 2. Características del sistema

	<code>indexAction()</code> <code>createAction(Request \$request)</code> <code>newAction()</code> <code>showAction(\$id)</code> <code>editAction(\$id)</code> <code>updateAction(Request \$request, \$id)</code> <code>deleteAction(Request \$request, \$id)</code> <code>buscarAction()</code> <code>new_rolAction()</code> <code>rol_createAction()</code>
Artefacto	<code>indexAction(\$idproducto)</code> <code>createAction(Request \$request)</code> <code>newAction()</code> <code>showAction(\$id)</code> <code>editAction(\$id)</code> <code>updateAction(Request \$request, \$id)</code> <code>deleteAction(Request \$request, \$id)</code> <code>createDeleteForm(\$id)</code>
Entorno de Prueba	<code>indexAction()</code> <code>datos_epAction(Request \$request)</code> <code>createAction()</code> <code>create_masAction()</code> <code>newAction(Request \$request)</code> <code>showAction(\$id)</code> <code>editAction(\$id)</code> <code>updateAction(Request \$request, \$id)</code> <code>deleteAction(Request \$request, \$id)</code> <code>createDeleteForm(\$id)</code>
Producto	<code>indexAction()</code> <code>crearConSolicitudAction()</code> <code>createAction(Request \$request)</code> <code>newAction()</code>

Capítulo 2. Características del sistema

	<code>showAction(\$id)</code> <code>editAction(\$id)</code> <code>updateAction(Request \$request, \$id)</code> <code>deleteAction(Request \$request, \$id)</code> <code>createDeleteForm(\$id)</code>
Proyecto	<code>indexAction()</code> <code>createAction(Request \$request)</code> <code>newAction()</code> <code>showAction(\$id)</code> <code>editAction(\$id)</code> <code>updateAction(Request \$request, \$id)</code> <code>deleteAction(Request \$request, \$id)</code> <code>createDeleteForm(\$id)</code>
Solicitud	<code>indexAction()</code> <code>createAction(Request \$request)</code> <code>newAction()</code> <code>showAction(\$id)</code> <code>editAction(\$id)</code> <code>updateAction(Request \$request, \$id)</code> <code>deleteAction(Request \$request, \$id)</code> <code>createDeleteForm(\$id)</code>

2.7 Conclusiones Parciales

En este capítulo se describió la propuesta de solución a realizarse y la descripción de los procesos, determinando de esta manera las actividades que serán automatizadas dentro del proceso de gestión de entornos de prueba. Se definieron dos roles que estarán interactuando con el sistema, los mismos contarán con diferentes niveles de acceso, de acuerdo a los privilegios que posea. Las fases de Planificación y Diseño posibilitaron la confección de las historias de usuarios, que permitieron delimitar las tareas a realizar por los desarrolladores para llevar a cabo la implementación de cada funcionalidad, así como establecer el tiempo aproximado que se dedicará para el cumplimiento de las

Capítulo 2. Características del sistema

mismas. Las tarjetas CRC confeccionadas ayudaron a establecer la relación entre las funcionalidades a implementar y determinar la responsabilidad y colaboración entre ellas.

La metodología XP permitió minimizar el tiempo de trabajo, en cuanto a reducir la cantidad y complejidad de los artefactos a modelar. Se define la arquitectura MVC para un mejor manejo y organización del código a implementar.

Capítulo 3. Implementación y prueba

Capítulo 3. Implementación y Prueba

Introducción

Este capítulo se enmarca en los aspectos relacionados con la construcción del sistema para gestionar los entornos de prueba del Laboratorio Industrial de Pruebas de Software. Se explica los artefactos que define XP para las fases de implementación y prueba: casos de pruebas necesarios para poder ejecutar las pruebas al software y así validar la calidad del mismo, asegurando el cumplimiento de los requisitos. Durante este periodo el cliente mantiene una constante comunicación con el equipo de desarrollo, no solo para la aclaración de posibles dudas sino para formar parte de este.

En la fase de iteración el cliente realiza las pruebas funcionales para verificar que todo funciona correctamente y de ocurrir errores se solucionarán en iteraciones posteriores. Durante el desarrollo del capítulo, se afrontarán elementos esenciales relacionados con la implementación del sistema, estándares de codificación empleados y patrones de diseño aplicados durante el desarrollo del sistema. Además, se realizarán las pruebas de software, con el objetivo de lograr una aplicación que cumpla con los requerimientos previamente definidos.

3.1 Implementación

La metodología XP propone que los proyectos sean desarrollados por pares de programadores, esto propicia que se minimizan los errores y se logren mejores diseños. Usando ésta práctica el producto obtenido es por lo general de mejor calidad que cuando el desarrollo se realiza por programadores individuales.

En esta fase, las HU serán implementadas de acuerdo a la planificación realizada. Por esta razón se procede a verificar el plan realizado para comenzar la implementación.

Para crear el presente trabajo, se definen 3 iteraciones que serán detalladas a continuación:

3.1.1 Iteración 1

Capítulo 3. Implementación y prueba

Tabla 14_Funcionalidades implementadas en la primera iteración.

Tiempo de Implementación (semanas)		
Nombre de la HU	Estimación	Real
Gestionar usuarios.	1	1
Autenticar usuarios.	1	1
Gestionar solicitud de instalación de Entorno de Prueba (EP).	1	1

3.1.2 Iteración 2

Tabla 15_Funcionalidades Implementadas en la segunda iteración.

Tiempo de Implementación (semanas)		
Nombre de la HU	Estimación	Real
Gestionar entornos de prueba real.	1	1
Gestionar roles.	0.2	0.2
Gestionar proyectos.	0.4	0.3
Gestionar planificación.	1	1

3.1.3 Iteración 3

Capítulo 3. Implementación y prueba

Tabla 16_Funcionalidades implementadas en la tercera iteración.

Nombre de la HU	Tiempo de Implementación (semanas)	
	Estimación	Real
Gestionar calendarios.	1	1
Gestionar turnos.	1	1
Gestionar puestos de trabajo.	0.2	0.2
Gestionar incidencias.	0.3	0.3

3.2 Codificación

El cliente es una parte más del equipo de desarrollo, su presencia es indispensable en las distintas fases de X.P. A la hora de codificar una historia de usuario su presencia es aún más necesaria. No se puede olvidar que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple la funcionalidad especificada. En esta etapa de la codificación los clientes y los desarrolladores del proyecto deben estar en comunicación para que los desarrolladores puedan codificar todo lo necesario para el proyecto que se requiere, en esta período está incluido todo lo de codificación o programación por parte de los desarrolladores del proyecto. (42)

3.3 Tareas de ingeniería

Según la metodología XP, dentro de la etapa de planificación de la iteración, el cliente escoge el conjunto de historias de usuario de mayor valor para que sean efectuadas en la iteración creada. Estas historias de usuario son divididas en tareas más específicas denominadas tareas de ingeniería o tareas de programación (se plasman en tarjetas de papel donde se describe qué se debe realizar y

Capítulo 3. Implementación y prueba

son muy dinámicas y flexibles, pueden ser cambiadas por otras más generales o más específicas, agregarse nuevas o modificarse según las necesidades existentes, cada una de estas tareas puede ser comprobada a través de los casos de prueba). Luego, cada programador elige las tareas que desea implementar, las analiza en mayor detalle y realiza una estimación de su tiempo de desarrollo.(43)

A continuación se muestran algunas tareas de ingeniería (TI), las demás se encuentran en el **Anexo 2**; para la elaboración de las mismas se tomaran los parámetros siguientes:

- **Número tarea:** Número consecutivo a partir del 1.
- **Número historia:** Identifica la HU.
- **Nombre tarea:** Identifica la TI.
- **Tipo de tarea:** Desarrollo/ Corrección/ Mejora/ Otra.
- **Programador responsable:** Persona encargada de la realización de la TI.
- **Descripción:** Explica en qué consiste la tarea de ingeniería.

Tabla 17_Tarea #1.

Tarea #1
Número de la HU (No.1): Gestionar usuarios.
Nombre de la tarea: Desarrollar Create, Read, Update, Delete (CRUD) de los usuarios.
Tipo de tarea: Desarrollo.
Programador responsable: Deyrin Pérez Pedroso y Carlos Fuentes Puentes.
Descripción: Implementa las funcionalidades de insertar, modificar, eliminar y visualizar y autenticar usuarios.

Tabla 18_Tarea #2.

Tarea #2
Número de la HU (No.3): Gestionar las solicitudes de instalación de Entornos de Prueba (EP).
Nombre de la tarea: Desarrollar Create, Read, Update, Delete (CRUD) de las solicitudes de instalación de Entornos de Prueba.
Tipo de tarea: Desarrollo.
Programador responsable: Deyrin Pérez Pedroso y Carlos Fuentes Puentes.

Capítulo 3. Implementación y prueba

Descripción: Implementa las solicitudes de instalación de Entornos de Prueba especificando los datos correspondientes.

Tabla 19_Tarea #3.

Tarea #3
Número de la HU (No.4): Gestionar entorno de prueba real.
Nombre de la tarea: Desarrollar las funcionalidades del entorno real.
Tipo de tarea: Desarrollo.
Programador responsable: Deyrin Pérez Pedroso y Carlos Fuentes Puentes.
Descripción: Implementa las funcionales correspondiente al entorno real.

Tabla 20_Tarea #4.

Tarea #4
Número de la HU (No.5): Gestionar roles.
Nombre de la tarea: Desarrollar Create, Read, Update, Delete (CRUD) de los usuarios.
Tipo de tarea: Desarrollo.
Programador responsable: Deyrin Pérez Pedroso y Carlos Fuentes Puentes.
Descripción: Implementa las funcionalidades necesarias para permitir cambiar los roles a los usuarios.

Tabla 21_Tarea #5.

Tarea #5
Número de la HU (No.6): Gestionar proyectos.
Nombre de la tarea: Desarrollar Create, Read, Update, Delete (CRUD) de los proyectos.
Tipo de tarea: Desarrollo.
Programador responsable: Deyrin Pérez Pedroso y Carlos Fuentes Puentes.
Descripción: Implementar las funcionalidades de insertar, modificar, eliminar y visualizar los datos de los proyectos.

Capítulo 3. Implementación y prueba

Tabla 22_Tarea #6.

Tarea #6
Número de la HU (No.7): Gestionar planificación.
Nombre de la tarea: Desarrollar Create, Read, Update, Delete (CRUD) de la planificación.
Tipo de tarea: Desarrollo.
Programador responsable: Deyrin Pérez Pedroso y Carlos Fuentes Puentes.
Descripción: Implementar las funcionalidades de insertar, modificar, eliminar y visualizar la planificación.

Tabla 23_Tarea #7

Tarea #7
Número de la HU (No.8): Gestionar calendarios.
Nombre de la tarea: Desarrollar calendarios.
Tipo de tarea: Desarrollo.
Programador responsable: Deyrin Pérez Pedroso y Carlos Fuentes Puentes.
Descripción: Implementar las funcionalidades de visualizar las planificaciones.

Tabla 24_Tarea #8.

Tarea #8
Número de la HU (No.8.1): Gestionar turnos.
Nombre de la tarea: Gestionar turnos.
Tipo de tarea: Desarrollo.
Programador responsable: Deyrin Pérez Pedroso y Carlos Fuentes Puentes.
Descripción: Implementa las funcionalidades de insertar, modificar, eliminar y visualizar turnos de planificación.

3.4 Patrones de diseño usados en la aplicación

Los patrones de diseño describen las clases y objetos del diseño que se comunican entre sí. Tienen menor nivel de abstracción, y se aproxima más a la implementación final. Dentro de ellos se encuentran los patrones GRASP, que describen los principios fundamentales de diseño de objetos

Capítulo 3. Implementación y prueba

para la asignación de responsabilidades. A continuación, se exponen algunos ejemplos de patrones y se especifican los que fueron tenidos en cuenta al momento de desarrollar la propuesta:

Experto (GRASP): Este patrón indica que la asignación de responsabilidades será para la clase que contiene la información necesaria para cumplir la responsabilidad. En la aplicación se evidencia este patrón en la clase: Producto.php

```
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145

/**
 * Set nombre_proyecto
 *
 * @param string $nombreProyecto
 * @return Producto
 */
public function setNombreProyecto($nombreProyecto) {
    $this->nombre_proyecto = $nombreProyecto;

    return $this;
}

/**
 * Get nombre_proyecto
 *
 * @return string
 */
public function getNombreProyecto() {
    return $this->nombre_proyecto;
}
```

Figura 5_Ejemplo del patrón Experto.

Creador (GRASP): Este patrón guía la asignación de responsabilidades a una clase de crear una instancia de otra. En la aplicación se evidencia este patrón en la clase: ProductoController.php

Capítulo 3. Implementación y prueba

```
23 public function indexAction() {
24     $em = $this->getDoctrine()->getManager();
25     $user_actual = $this->get('security.context')->getToken()->getUser();
26
27     $entities = $em->getRepository('SolicitudBundle:Producto')->findAll();
28     $lista_product_x_user = $em->getRepository('SolicitudBundle:Producto')->findBy(array('usuario' => $user_actual));
29     return $this->render('SolicitudBundle:Producto:index.html.twig', array(
30         'entities' => $entities,
31         'lista_product_x_user' => $lista_product_x_user,
32     ));

```

Figura 5_Ejemplo del patrón Creador.

Bajo Acoplamiento (GRASP): Este patrón se encarga de minimizar la relación que tiene una clase con las demás clases, mediante la asignación de responsabilidades para que esa dependencia sea baja. En la aplicación se evidencia en las clases: ProductoController.php y EPController.php.

Alta Cohesión (GRASP): Este patrón facilita que las clases con responsabilidades estrechamente relacionadas no realicen un trabajo enorme, asignándole responsabilidades afines a cada clase, de modo que la cohesión sea alta. En la aplicación se evidencia este patrón en la clase: ProductoController.php.

Controlador (GRASP): Este patrón se encarga de asignar la responsabilidad del manejo de los eventos del sistema a una clase. En la aplicación se evidencia este patrón en las clases controladoras, una de las cuales es ProductoController.php.

3.5 Estándares de codificación

Los estándares de codificación son reglas que se siguen para la escritura del código fuente. De tal manera que a los programadores se les facilite entender el código escrito por otros. Su propósito fundamental es que el código tenga un estilo consistente, independiente del autor, con lo cual el sistema resulte fácil de entender y por supuesto fácil de mantener. Forman parte de una de las buenas prácticas que propone la metodología XP, donde todos los programadores deben escribir y documentar en código en la misma manera.(44)

Capítulo 3. Implementación y prueba

Los estándares de codificación son un complemento a la programación por pares y no sólo es importante tener un estándar, sino tener un buen estándar de codificación, de esta forma un buen estándar de codificación deberá:

- Clarificar más que confundir.
- Promover la intención del código.
- Permitir que los programas se acerquen lo mejor posible al lenguaje natural.
- Incorporar las mejores prácticas de la codificación.

Para la implementación del sistema se siguió el siguiente estándar de codificación:

El nombre de todas las clases comienzan con la primera letra en mayúscula y las demás en minúscula por ejemplo: Usuario. Se utilizará el convenio Pascal para la capitalización de los identificadores (45), en caso de ser un nombre compuesto que se escribe de la siguiente manera, la inicial de cada uno con mayúscula; ejemplo: EP. En el caso de los atributos de las entidades, se declaran con minúscula ejemplo: \$versión. Si los atributos son nombres compuestos se declaran de la siguiente forma: \$num_solicitud, en abreviatura la primera palabra seguida de "_" y la otra palabra completa.

3.6 Pruebas de software

“Las pruebas de Software son la ejecución del código usando combinaciones de entradas, en un determinado estado, para revelar defectos.”(46) Su objetivo fundamental es verificar los requisitos del sistema, por lo que son los propios requisitos la principal fuente de información a la hora de construir las pruebas del sistema.

Uno de los pilares fundamentales de la metodología XP es el proceso de prueba. XP anima a probar constantemente tanto como sea posible para asegurar todo el proceso de desarrollo de un producto. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección.(47)

3.6.1 Pruebas de Aceptación

Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una

Capítulo 3. Implementación y prueba

historia de usuario ha sido correctamente implementada. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. (48)

A continuación se detallan algunos casos de pruebas (CP) de aceptación los demás se encuentran en el Anexo 3.

Tabla 25_CPA HU1_P1 Gestionar usuarios.

Caso de prueba de aceptación	
Código: HU1_P1	Historia de usuario: Gestionar usuarios.
Nombre: Insertar usuario.	
Descripción: Para registrarse debe introducir: Nombre, Apellidos, Email, Usuario y Contraseña.	
Condiciones de ejecución:	
Pasos de ejecución: El usuario selecciona la opción "Registrar Usuario" que se encuentra en la vista inicial de la aplicación, donde deberá introducir los siguientes datos: nombre, apellidos, usuario, correo y contraseña. Luego se acciona sobre el botón "Aceptar" para guardar los datos y "Cancelar" muestra la vista anterior.	
Resultado esperado: Los datos son registrados satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 26_CPA HU3_P1 Gestionar solicitud de instalación de Entorno de Prueba (EP).

Caso de prueba de aceptación	
Código: HU3_P1	Historia de usuario: Gestionar solicitud de instalación de Entornos de Prueba (EP).
Nombre: Crear solicitud de instalación de EP.	
Descripción: Prueba para la funcionalidad que permite crear solicitud de instalación de EP.	

Capítulo 3. Implementación y prueba

Condiciones de ejecución: El usuario debe estar autenticado con el rol de administrador o especialista.
Pasos de ejecución: <ol style="list-style-type: none">1. El administrador o especialista se registran en la aplicación.2. En su pantalla inicial se le muestra la opción “+Nuevo Proyecto”.3. Posteriormente la aplicación mostrará el formulario con todos los datos correspondientes para crear una solicitud de EP. Una vez llenado todos los campos, él usuario ejecuta el botón “Siguiente”.
Resultado esperado: Los datos son registrados satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 27_CPA HU4_P1 Gestionar entorno de prueba real.

Caso de prueba de aceptación	
Código: HU4_P1	Historia de usuario: Gestionar entorno de prueba real.
Nombre: Agregar entorno de prueba real.	
Descripción: Prueba para la funcionalidad que permite agregar entorno de prueba real.	
Condiciones de ejecución: El usuario debe estar autenticado con el rol de administrador o especialista para acceder a todas los entornos reales, en caso de ser un usuario solo tendrá acceso a sus entornos reales.	
Pasos de ejecución: <p>El administrador o especialista dispondrá al modificar una solicitud de instalación de EP, un botón “Adicionar” donde podrá agregar tantos entornos reales que desee. Una vez llenado todos los campos, él usuario ejecuta el botón “Siguiente”.</p>	
Resultado esperado: El entorno de prueba real es agregado satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 28_CPA HU5_P1 Gestionar roles.

Capítulo 3. Implementación y prueba

Caso de prueba de aceptación	
Código: HU5_P1	Historia de usuario: Gestionar roles.
Nombre: Agregar rol.	
Descripción: Prueba para la funcionalidad que permite agregar un rol.	
Condiciones de ejecución: El usuario debe estar autenticado con el rol de administrador o especialista.	
Pasos de ejecución: <ol style="list-style-type: none">1. Al administrador o especialista acceder a la aplicación en su pantalla inicial se le mostrará el botón "Editar Usuario".2. Posteriormente se le brindará la opción "Adicionar".	
Resultado esperado: El rol es agregado satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 29_CPA HU6_P1 Gestionar proyectos.

Caso de prueba de aceptación	
Código: HU6_P1	Historia de usuario: Gestionar proyectos.
Nombre: Agregar proyecto.	
Descripción: Prueba para la funcionalidad que permite agregar un proyecto.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: <ol style="list-style-type: none">1. Cualquiera de los usuarios de la aplicación pueden crear un proyecto, al autenticarse se le mostrará una opción "Crear Nuevo Proyecto".2. Una vez dado clic en la opción "Crear Nuevo Proyecto", se le mostrará los datos correspondientes para crear un proyecto, y un botón "Cancelar" que lo llevará a la vista anterior si no desea crear un nuevo	

Capítulo 3. Implementación y prueba

proyecto.
Resultado esperado: El proyecto es agregado satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 30_CPA HU7_P1 Gestionar planificación.

Caso de prueba de aceptación	
Código: HU7_P1	Historia de usuario: Gestionar planificación.
Nombre: Agregar planificación.	
Descripción: Prueba para la funcionalidad que permite agregar una planificación.	
Condiciones de ejecución: El usuario debe estar autenticado con el rol de administrador o usuario.	
Pasos de ejecución: <ol style="list-style-type: none">1. Los usuarios al acceder a la aplicación dispondrán en el menú la opción “Planificación”2. En el calendario podrá seleccionar el día que desee realizar la planificación.	
Resultado esperado: La planificación es agregado satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 31_CPA HU8_P1 Gestionar turnos.

Caso de prueba de aceptación	
Código: HU8_P1	Historia de usuario: Gestionar turnos.
Nombre: Agregar turnos.	

Capítulo 3. Implementación y prueba

Descripción: Prueba para la funcionalidad que permite agregar un turno.
Condiciones de ejecución: El usuario debe estar autenticado con el rol de administrador o usuario.
Pasos de ejecución: <ol style="list-style-type: none"> 1. Los usuarios al acceder a la aplicación dispondrán en el menú la opción “Planificación”. 2. Dando clic encima del calendario podrán agregar el puesto de trabajo.
Resultado esperado: Los turnos son agregados satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 32_CPA HU9_P1 Gestionar puestos de trabajo.

Caso de prueba de aceptación	
Código: HU9_P1	Historia de usuario: Gestionar puestos de trabajo.
Nombre: Agregar puesto de trabajo.	
Descripción: Prueba para la funcionalidad que permite agregar puesto de trabajo.	
Condiciones de ejecución: El usuario debe estar autenticado con el rol de administrador o usuario.	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Los usuarios al acceder a la aplicación dispondrán en el menú la opción “Planificación”. 2. En el calendario al dar clic encima de un día que deseen tendrán la posibilidad de agregar un puesto de trabajo. 	
Resultado esperado: Los puestos de trabajo son agregados satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 33_CPA HU10_P1 Gestionar incidencias.

Capítulo 3. Implementación y prueba

Caso de prueba de aceptación	
Código: HU10_P1	Historia de usuario: Gestionar incidencias.
Nombre: Agregar incidencias.	
Descripción: Prueba para la funcionalidad que permite agregar incidencias.	
Condiciones de ejecución: <ul style="list-style-type: none">• El usuario debe estar autenticado con el rol de administrador o usuario.	
Pasos de ejecución: <ol style="list-style-type: none">1. Los usuarios en el momento de crear un nuevo proyecto, dispondrán de un campo donde podrán agregar las incidencias.	
Resultado esperado: Las incidencias son agregadas satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 34_CPA HU10_P3 Gestionar incidencias.

Caso de prueba de aceptación	
Código: HU10_P3	Historia de usuario: Gestionar incidencias.
Nombre: Visualizar incidencias.	
Descripción: Prueba para la funcionalidad que permite visualizar incidencias.	
Condiciones de ejecución: <ul style="list-style-type: none">• El usuario debe estar autenticado con el rol de administrador o usuario.	
Pasos de ejecución: <ol style="list-style-type: none">1. En la pantalla inicial del usuario dispondrán al lado de cada solicitud de instalación de EP, un botón “Detalles”, donde podrán visualizar cada incidencia.	

Capítulo 3. Implementación y prueba

Resultado esperado: Las incidencias son visualizadas satisfactoriamente.

Evaluación de la prueba: Prueba satisfactoria.

Tabla 35_CPA HU10_P2 Gestionar incidencias.

Caso de prueba de aceptación	
Código: HU10_P2	Historia de usuario: Gestionar incidencias.
Nombre: Modificar incidencias.	
Descripción: Prueba para la funcionalidad que permite modificar incidencias.	
Condiciones de ejecución: <ul style="list-style-type: none">• El usuario debe estar autenticado con el rol de administrador o usuario.	
Pasos de ejecución: <ol style="list-style-type: none">1. En la pantalla inicial del usuario dispondrán al lado de cada solicitud de instalación de EP, un botón "Editar", donde podrán modificar las incidencias.	
Resultado esperado: Las incidencias son modificadas satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 36_CPA HU9_P2 Gestionar puestos de trabajo.

Caso de prueba de aceptación	
Código: HU9_P2	Historia de usuario: Gestionar puestos de trabajo.
Nombre: Modificar puestos de trabajo.	
Descripción: Prueba para la funcionalidad que permite modificar puestos de trabajo.	
Condiciones de ejecución: El usuario debe estar autenticado con el rol de administrador o de usuario.	
Pasos de ejecución:	

Capítulo 3. Implementación y prueba

1. Los usuarios al acceder a la aplicación dispondrán en el menú la opción “Planificación”.

En el calendario al dar clic encima de un día que deseen tendrán la posibilidad de modificar un puesto de trabajo.

Resultado esperado: Los puestos de trabajo son modificados satisfactoriamente.

Evaluación de la prueba: Prueba satisfactoria.

Tabla 37_CPU HU8_P2 Gestionar turnos.

Caso de prueba de aceptación	
Código: HU8_P2	Historia de usuario: Gestionar turnos.
Nombre: Modificar turnos.	
Descripción: Prueba para la funcionalidad que permite modificar turnos.	
Condiciones de ejecución: <ul style="list-style-type: none">• El usuario debe estar autenticado con el rol de administrador o usuario.	
Pasos de ejecución <ol style="list-style-type: none">1. Los usuarios al acceder a la aplicación dispondrán en el menú la opción “Planificación”.2. Dando clic encima del calendario podrán modificar los puestos de trabajo.	
Resultado esperado: Los turnos son modificados satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 38_CPU HU8_P3 Gestionar turnos.

Caso de prueba de aceptación	
Código: HU8_P3	Historia de usuario: Gestionar turnos.
Nombre: Eliminar turnos.	

Capítulo 3. Implementación y prueba

Descripción: Prueba para la funcionalidad que permite eliminar turnos.
Condiciones de ejecución: <ul style="list-style-type: none">El usuario debe estar autenticado con el rol de administrador o de usuario.
Pasos de ejecución <ol style="list-style-type: none">Los usuarios al acceder a la aplicación dispondrán en el menú la opción “Planificación”.En el calendario al dar clic encima del turno que deseen eliminar, le saldrá una pantalla y en uno de sus laterales un botón “Eliminar”, si desean eliminar el turno de trabajo.
Resultado esperado: Los turnos son eliminados satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 39_CPU HU9_P3 Gestionar puestos de trabajo.

Caso de prueba de aceptación	
Código: HU9_P3	Historia de usuario: Gestionar puestos de trabajo.
Nombre: Visualizar puestos de trabajo.	
Descripción: Prueba para la funcionalidad que permite visualizar puestos de trabajo.	
Condiciones de ejecución: El usuario debe estar autenticado con el rol de administrador o de usuario.	
Pasos de ejecución: <ol style="list-style-type: none">Los usuarios al acceder a la aplicación dispondrán en el menú la opción “Planificación”.En el calendario se visualizan todos los puestos de trabajo.	
Resultado esperado: Los puestos de trabajo son visualizados satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 40_CPU HU8_P3 Gestionar turnos.

Capítulo 3. Implementación y prueba

Caso de prueba de aceptación	
Código: HU8_P3	Historia de usuario: Gestionar turnos.
Nombre: Visualizar turnos.	
Descripción: Prueba para la funcionalidad que permite visualizar turnos.	
Condiciones de ejecución: <ul style="list-style-type: none">• El usuario debe estar autenticado con el rol de administrador o de usuario.	
Pasos de ejecución <ol style="list-style-type: none">1. Los usuarios al acceder a la aplicación dispondrán en el menú la opción “Planificación”.2. En el calendario podrán visualizar los puestos de trabajo.	
Resultado esperado: Los turnos son visualizados satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 41_CPU HU6_P4 Gestionar proyectos.

Caso de prueba de aceptación	
Código: HU6_P4	Historia de usuario: Gestionar proyecto.
Nombre: Eliminar proyecto.	
Descripción: Prueba para la funcionalidad que permite eliminar un proyecto.	
Condiciones de ejecución: <ul style="list-style-type: none">• El usuario debe estar autenticado con el rol de administrador o de usuario.	
Pasos de ejecución <ol style="list-style-type: none">1. Los usuarios al acceder a la aplicación dispondrán una lista de los proyectos con la opción “Detalles”.2. Una vez mostrados todos los datos referentes al proyecto, dispondrá de un botón “Eliminar”, el cual elimina el proyecto.	

Capítulo 3. Implementación y prueba

Resultado esperado: El proyecto es eliminado satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 42_CPU HU6_P3 Gestionar proyectos.

Caso de prueba de aceptación	
Código: HU6_P3	Historia de usuario: Gestionar proyecto.
Nombre: Visualizar proyecto.	
Descripción: Prueba para la funcionalidad que permite visualizar un proyecto.	
Condiciones de ejecución: <ul style="list-style-type: none">El usuario debe estar autenticado con el rol de administrador o de usuario, en caso de un usuario normal solo puede visualizar los proyectos creados por él.	
Pasos de ejecución <ol style="list-style-type: none">Los usuarios al acceder a la aplicación dispondrán una lista de los proyectos.	
Resultado esperado: El proyecto es visualizado satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 43_CPU HU6_P2 Gestionar proyectos.

Caso de prueba de aceptación	
Código: HU6_P2	Historia de usuario: Gestionar proyecto.
Nombre: Modificar proyecto.	
Descripción: Prueba para la funcionalidad que permite modificar un proyecto.	

Capítulo 3. Implementación y prueba

Condiciones de ejecución: <ul style="list-style-type: none">El usuario debe estar autenticado con el rol de administrador o de usuario, en caso de un usuario normal solo puede modificar los proyectos creados por él y también debe poseer algún proyecto para modificar.
Pasos de ejecución <ol style="list-style-type: none">Los usuarios al acceder a la aplicación dispondrán una lista de los proyectosPosteriormente se le brindará la opción "Editar", donde podrá modificar los datos del proyecto.
Resultado esperado: El proyecto es modificado satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 44_CPU HU7_P4 Gestionar planificación.

Caso de prueba de aceptación	
Código: HU7_P4	Historia de usuario: Gestionar planificación.
Nombre: Eliminar planificación.	
Descripción: Prueba para la funcionalidad que permite eliminar una planificación.	
Condiciones de ejecución: <ul style="list-style-type: none">El usuario debe estar autenticado con el rol de administrador o de usuario.	
Pasos de ejecución <ol style="list-style-type: none">Los usuarios al acceder a la aplicación dispondrán en el menú la opción "Planificación".Dando clic encima de la planificación que desea eliminar, se le mostrará un botón "Eliminar Planificación".	
Resultado esperado: La planificación es eliminada satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Capítulo 3. Implementación y prueba

Tabla 45_CPU HU5_P3 Gestionar rol.

Caso de prueba de aceptación	
Código: HU5_P3	Historia de usuario: Gestionar rol.
Nombre: Eliminar rol.	
Descripción: Prueba para la funcionalidad que permite eliminar un rol.	
Condiciones de ejecución: <ul style="list-style-type: none">• El usuario debe estar autenticado con el rol de administrador.	
Pasos de ejecución <ol style="list-style-type: none">1. Al administrador acceder a la aplicación en su pantalla inicial se le mostrará el botón “Editar Usuario”.2. Posteriormente se le brindará la opción “Editar”.3. Luego tendrá la posibilidad de dar en él botón “Eliminar”.	
Resultado esperado: El rol es eliminado satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 46_CPU HU5_P2 Gestionar rol.

Caso de prueba de aceptación	
Código: HU5_P2	Historia de usuario: Gestionar rol.
Nombre: Modificar rol.	
Descripción: Prueba para la funcionalidad que permite modificar un rol.	
Condiciones de ejecución: <ul style="list-style-type: none">• El usuario debe estar autenticado con el rol de administrador.	
Pasos de ejecución <ol style="list-style-type: none">1. Al administrador acceder a la aplicación en su pantalla inicial se le mostrará el botón “Editar Usuario”.2. Posteriormente se le brindará la opción “Editar rol”.	

Capítulo 3. Implementación y prueba

3. Luego se le mostrará una pantalla donde podrá cambiar el rol al usuario.

Resultado esperado: El rol es modificado satisfactoriamente.

Evaluación de la prueba: Prueba satisfactoria.

3.7 Conclusiones Parciales

La descripción de la propuesta de solución en el capítulo anterior permitió desarrollar las funcionalidades descritas a partir de las herramientas, lenguajes y tecnologías seleccionados en el capítulo 1. Los estándares de codificación permitieron que ambos desarrolladores pudieran trabajar de manera coordinada y pudiesen interpretar eficientemente el código. Se validó la aplicación propuesta mediante la realización de pruebas de aceptación, a través de tres iteraciones, donde se pudo documentar las no conformidades detectadas en la aplicación durante su desarrollo, las cuales fueron posteriormente corregidas. Al finalizar la última iteración se obtuvo una aplicación que cumple con las especificaciones establecidas por el cliente.

Conclusiones Generales

Conclusiones Generales

Después de desarrollar el presente trabajo y analizar los resultados obtenidos, se arriban a las siguientes conclusiones:

- Los métodos científicos utilizados permitieron definir los principales conceptos y teorías que sustentan la presente investigación.
- La selección de la metodología, herramientas y tecnologías a utilizar posibilitaron al equipo de desarrollo cumplir con los objetivos propuestos.
- Mediante la aplicación de la metodología ágil XP se logró la realización del análisis y diseño, permitiendo desarrollar en un corto periodo de tiempo una aplicación informática acorde con las necesidades planteadas.
- La ejecución de las pruebas a la aplicación facilitó valorar la calidad y el correcto funcionamiento de la misma, pudiendo asegurar que las funcionalidades alcanzadas se encuentran en correspondencia con las necesidades del cliente.

Como resultado de la investigación se logró desarrollar una aplicación con la que se da cumplimiento a las especificidades de los objetivos propuestos. Con el principal propósito de mejorar el proceso de gestión de los entornos de prueba en el LIPS.

Recomendaciones

La investigación realizada logra sus objetivos luego de su puesta en práctica, no obstante se hace necesario realizar las siguientes recomendaciones:

- A los administradores de la aplicación, mantener actualizada la misma con las nuevas tecnologías y herramientas informáticas.
- Agregarle nuevas funcionalidades para mejorar el empleo de la aplicación.

Referencias bibliográficas

Referencias bibliográficas

1. SANTANA., Pedro Ernesto Matos. La Industria Cubana Del Software - Investigaciones - Pedritos18. [Online]. [Accessed 27 January 2014]. Available from: <http://www.buenastareas.com/ensayos/La-Industria-Cubana-Del-Software/6991410.html>
2. Cuba en Noticias: Presentan Industria Cubana del Software. [online]. [Accessed 27 January 2014]. Available from: http://old.cubahora.cu/index.php?tpl=buscar/ver-not_buscar.tpl.html&newsid_obj_id=2859
3. *SISTEMAS DE GESTIÓN DE LA CALIDAD — FUNDAMENTOS Y VOCABULARIO*[ISO 9000:2005, (Traducción certificada), IDT]. 2005.
4. VEGA ALMEIDA ROSA LIDIA. *La planificación* [online]. 2014. Ciencias Médicas. Available from: acimed@infomed.sld.cu
5. Definición de Gestión - Qué es y Concepto. [online]. [Accessed 22 May 2014]. Available from: <http://definicion.mx/gestion/>
6. ESPINAR, Pablo. Tradingsys.org - Reflexión independiente sobre sistemas de trading - Factores Prácticos del trading: Entorno de Producción. [Online]. 2014. [Accessed 22 May 2014]. Available from: http://www.tradingsys.org/index.php?option=com_content&task=view&id=162&Itemid=54
7. Project KickStart 4 - Download. [online]. [Accessed 8 April 2014]. Available from: <http://www.mbaware.com/prki.html>
8. CA Open Workbench - CA Technologies. [online]. [Accessed 3 May 2014]. Available from: <http://www.ca.com/ar/collateral/demos/na/ca-open-workbench.aspx>
9. ALVARGONZÁLEZ, Javier Fernández, Pello. QATraq Gestiona tu entorno de Pruebas | Softqanetwork.com. [Online]. 2006. [Accessed 1 May 2014]. Available from: <http://www.softqanetwork.com/qatraq-gestiona-tu-entorno-de-pruebas>
10. LABORATORIO DE INVESTIGACIONES EN GESTIÓN DE PROYECTOS, UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS. FORTES- - Centro de Tecnologías para la Formación. [online]. 2013. [Accessed 25 May 2014]. Available from: <http://gespro.fortes.prod.uci.cu/>
11. AREVALO, Maria Eugenia and LIZARDO. Diferencias entre Metodologías Tradicionales y Ágiles #MetodologiasAgiles. [online]. [Accessed 3 May 2014]. Available from: <http://arevalomaria.wordpress.com/2011/11/15/diferencias-entre-metodologias-tradicionales-y-agiles-metodologiasagiles/>
12. Servicios de Soporte — Soporte de TiendaLinux.com Linux en Colombia , Bogota , RedHat ,

Referencias bibliográficas

Red Hat , SuSE , Novell , Debian , Instalacion / Implementacion de Soluciones de Software Libre /

Open Source , Hosting , Servidores de Correo , Web , Archivos , Directorio , Postfix , Apache , Samba , LDAP , Amavis , Seguridad , Firewall , Firewalls , . [online]. 2005. [Accessed 23 May 2014]. Available from: http://soporte.tiendalinux.com/portal/index_html

13. CANÓS, José H., PATRICIO LETELIER and M^a CARMEN PENADÉ. *Metodologías Ágiles en el Desarrollo de Software*.
14. BENUSSI, Giovanni. Programación en C/C++: Trabajando con punteros [Maestros del Web. [Online]. August 2012. [Accessed 23 May 2014]. Available from: <http://www.maestrosdelweb.com/editorial/programacion-en-c-trabajando-con-punteros/>
15. HIGHSMITH, Jim. Agile Project Management Adaptive Software Development. [Online]. 2006 2002. [Accessed 23 May 2014]. Available from: <http://www.adaptivesd.com/>
16. Feature Driven Development | the portal for all things FDD. [Online]. 2004. [Accessed 23 May 2014]. Available from: <http://www.featuredrivendevelopment.com/>
17. POPPENDIECK, Tom. THE LEAN MINDSET. [Online]. [Accessed 23 May 2014]. Available from: <http://www.poppendieck.com/>
18. ESCRIBANO, Gerardo Fernández. *Introducción a Extreme Programming*. 9 December 2012.
19. *Repositorio* [online]. 14 January 2013. Available from: <http://repositorio.utn.edu.ec/bitstream/123456789/993/1/04%20ISC%20207%20TESIS%20DOCUMENTO.pdf>
20. FIGUEROA IDALGO, DAIRA, ORTIZ, Vega, Yurisbel and FERNANDEZ, Martell. *Publicaciones UC* [online]. Available from: <http://publicaciones.uci.cu/index.php/SC/article/view/208/228>
21. *Zend* [online]. Available from: <http://tuxpuc.pucp.edu.pe/articulo/comparativa-de-frameworks-en-php-cakephp-symfony-y-zend-framework>
22. ALAN2793. *Entorno de Desarrollo Integrado (IDE)* [online]. 2013. Available from: <http://alanss18.wordpress.com/2013/01/25/entorno-de-desarrollo-integradoide>
23. *Netbeans* [online]. Available from: http://netbeans.org/index_es.html
24. *Picando Java* [online]. Available from: <https://sites.google.com/site/picandojavacom/mundo-java/ide-eclipse/%C2%BFqueeseclipseinstalacion>
25. PÉREZ GARZÓN and TERESA, María. *Innovación y experiencias educativas*. 2010.
26. PICHES IGLESIA and LISANDRA. *Diseño de una aplicación para la Gestión de la información de las asignaturas Práctica Profesional en la Facultad 7. Ciudad Habana*. 2008.

Referencias bibliográficas

27. *Script* [online]. Available from: <http://es.scribd.com/doc/58356519/Teoria-PostgreSQL>
28. *Soporte.tiendalinux* [online]. Available from: http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html
29. COBO, Angel, GOMEZ, Patricia, PEREZ, Daniel and ROCHA, Rocio. *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web. España. 2005.* Ediciones Diaz de Santos.
30. CRUZ IÑIGO, Dayron. *Sistema Gestión de la Información de la Facultad 8.Módulo para la Gestión de la Información Docente. Ciudad Habana. 2007 2006.*
31. PÉREZ, Javier Eguíluz. *Librosweb* [online]. [no date]. Available from: www.librosweb.es/javascript
32. GONZÁALEZ, Martins and ANGEL, Miguel. *Los lenguajes de programación actuales, Universidad de Alcalá de Henares. 2013.*
33. GONZÁLEZ, Blanco and YANEISY. *Sistema de UML para los proyectos de la línea Soluciones Integrales. 2011.*
34. *Scribd* [online]. Available from: <http://es.scribd.com/doc/136393787/38/Herramienta-Utilizada-Rational-Rose>
35. RUIZ LEYVA ILEN. *Análisis y diseño de un sistema para automatizar la gestión de la información de los profesores de la facultad 4 de la UCI. 2008.*
36. CALBARIA, LUIS and PÍRIS PABLO. *Metodología XP.*
37. LETELIER TORRES PTRICIO and SÁNCHEZ LÓPEZ EMILIO A. *“Metodologías ágiles en el desarrollo de software. España.*
38. RODRIGUEZ JUANDA. *juandarodriguez* [online]. Available from: <http://juandarodriguez.es/cursosf14/unidad7.html>
39. ECHEVERRY TOBÓN LUIS MIGUEL, DELGADO CARMONA and LUZ ELENA. *Repositorio.utp* [online]. Available from: <http://repositorio.utp.edu.co/dspace/bitstream/11059/794/1/0053E18cp.pdf>
40. LETELIER, PATRICIO and PENADÉS, MA.CARMEN. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Metodologías ágiles para el desarrollo de software: eXtreme.*
41. CASAS, SANDRA Y REINAGA and HÉCTOR. *Identificación y Modelado de los Aspectos Tempranos por Tarjetas de Responsabilidad y Colaboraciones.*

Referencias bibliográficas

42. LEAL, Wendy. Metodologías Ágiles de desarrollo de software (XP) Fases. [online]. 22 June 2008. [Accessed 10 May 2014]. Available from: http://boards5.melodysoft.com/UBV_INGS/metodologias-agiles-de-desarrollo-43.html
43. ERLIJMAN PIWEN ARIEL and GOYÉN FROS ALEJANDRO. *Problemas y soluciones en la implementación de Extreme Programing.*
44. CALLEJA, Manuel Arias. *Estándares de codificación.*
45. DE LAS VEGAS GARCÍA, ERIK and MASSÓ VERRIER, ELISKA FRANCY. *ESTÁNDARES DE CODIFICACIÓN.* 2009.
46. PROF. CAROLINA, IVETTE. *Pruebas de software. Universidad Simón Bolívar.*
47. J.J. GUTIÉRREZ, M.J. ESCALONA, M. MEJÍAS and TORRES, J. *Pruebas del sistema en la Programación Extrema. Seville.*
48. MYERS, GLENFORD J. *The art of software testing.* 2004.

Bibliografía consultada

1. Acuña, B. (2009). *Selección de metodologías de desarrollo para aplicaciones web*. Cienfuegos: Universidad de Cienfuegos:"Carlos Rafael Rodríguez".
2. Blanco Bueno, C. (n.d.). *Ingeniería de software II. Tema 1:Construcción y pruebas de software*. . Universidad de Cantabria.
3. Calabria, L. y. (2003). *Metodología XP*. Uruguay.
4. Casillas Santillan, L. A., Giber Ginesta, M., & Pérez Mora, O. (2011). *Universitat Oberta de Catalunya*. Retrieved 2 20, 2014, from Universitat Oberta de Catalunya: <http://ocw.uoc.edu/computer-science-technology-and-multimedias-bases-de-datos/bases-de-datos/p06-m2109-02151>
5. Castillo, O. (2011). *Programació Extrema*.
6. Clavería, C. (2013). Recuperado el 1 de 5 de 2014, de http://cesclaveria.blogspot.com/2008/04/valores-de-xp-extreme_programming.html.
7. CodeIgniter. (2013). *Guía del Usuario de CodeIgniter*. Retrieved 2 22, 2014, from Guía del Usuario de CodeIgniter.
8. Corporation, O. (2012). *NetBeans*. Retrieved 4 6, 2014, from NetBeans: <http://netbeans.org/features/platform/index.html>.
9. Eguíluz Pérez, J. (2008). *Introducción a JavaScript*.
10. Española, R. A. (2010). *DICCIONARIO DE LA LENGUA ESPAÑOLA*. Retrieved from DICCIONARIO DE LA LENGUA ESPAÑOLA: <http://buscon.rae.es/draef/>.
11. Granville G., M. (2011). *Visual Studio Team System. Innovation and integration across the entire application development life cycle*. Retrieved 1 25, 2014, from Visual Studio Team

Bibliografía consultada

12. System. Innovation and integration across the entire application development life cycle:
<http://blogs.msdn.com>.
13. Identidad Cuba, P. (2011). *Manual de normas de diseño del SIEC*. La Habana.
14. Informática, D. d. (2010). *Universidad de Jaén*.
15. Ing. Mendoza Vázquez, I. (n.d.). *Definición de un framework para aplicaciones web con navegación sensible a concers*. República Argentina.
16. Ing. Joscowicz, J. (2008). *Reglas y Prácticas en eXtreme Programming*. España.
17. J, P. (2011, 6 25). *Optimus Software y Consultoría C.A.* Retrieved 3 16, 2014, from Optimus Software y Consultoría C.A: <http://optimus-software.com/noticias/2011/06/25/asi-era-la-primera-implementacion-de-scrum-para-software>
18. Jffries, R., Anderson, A., & Hendrickson, C. (2011). *Extremen Programming Installed*. Addison Wesley.
19. Landrian García, M. J., Blanco Zambrano, I., & Zalazar Pacio, I. H. (2010). *Arquitectura de software*. La Habana.
20. Larman, C. (2008). *UMLy patrones*.
21. Letelier Torres, P. y. (2003). *Metodologías ágiles en el desarrollo de software*. España.
22. Palacio, J. (2008). *Flexibilidad con SCRUM*.
23. Pressman, R. S. (2002). *Ingeniería de software. Un enfoque práctico*. Madrid.

Anexos

Anexo 1. Historias de usuario.

Historia de Usuario	
Número: 2	Nombre: Autenticar usuarios.
Usuario: Administrador, usuario.	
Puntos Estimados: 1	Iteración Asignada: 1
Riesgo en Desarrollo: Alta	Puntos Reales: 1
Descripción: Al acceder a la aplicación en la página principal el usuario deberá introducir usuario y contraseña para autenticarse. Según sus privilegios accederá a la sección correspondiente.	
Observación:	

Historia de Usuario	
Número: 8	Nombre: Gestionar calendarios.
Usuario: Administrador, Usuario.	
Puntos Estimados: 1	Iteración Asignada: 4
Riesgo en Desarrollo: Alta	Puntos Reales: 1
Descripción: Permite crear, listar, ver, editar y eliminar una o varias planificaciones de prueba en el calendario.	
Observación:	

Historia de Usuario	
Número: 8.1	Nombre: gestionar turnos.
Usuario: Administrador, Usuario.	
Puntos Estimados: 1	Iteración Asignada: 4

Riesgo en Desarrollo: Alta	Puntos Reales: 1
Descripción: Permite crear, listar, ver, editar y eliminar uno o varios turnos de trabajo.	
Observación:	

Historia de Usuario	
Número: 9	Nombre: Gestionar puestos de trabajo.
Usuario: Administrador, Usuario.	
Puntos Estimados: 0.2	Iteración Asignada: 1
Riesgo en Desarrollo: Baja	Puntos Reales: 0.2
Descripción: Se realizan las acciones agregar, modificar, eliminar los puestos de trabajos.	
Observación:	

Historia de Usuario	
Número: 10	Nombre: Gestionar incidencias.
Usuario: Administrador, Usuario.	
Puntos Estimados: 0.3	Iteración Asignada: 1
Riesgo en Desarrollo: Media	Puntos Reales: 0.3
Descripción: Se realizan las acciones agregar, modificar, eliminar incidencias.	
Observación:	

Historia de Usuario	
Número: 7	Nombre: Gestionar planificación.
Usuario: Administrador, Usuario.	
Puntos Estimados: 1	Iteración Asignada: 4
Riesgo en Desarrollo: Alta	Puntos Reales: 1
Descripción: Permite crear, listar, ver, editar y eliminar una o varias planificaciones de prueba. Solo se puede planificar una solicitud de prueba previamente "Aceptada".	
Observación:	

Anexo 2. Tareas de Ingeniería

Tarea #9
Número de la HU (No.2): Autenticar usuario.
Nombre de la tarea: Desarrollar la autenticación de los usuarios.
Tipo de tarea: Desarrollo.
Programador responsable: Deyrin Pérez Pedroso y Carlos Fuentes Puentes.
Descripción: Implementar las funcionalidades necesarias para permitir la autenticación del usuarios en el sistema.

Tarea #10
Número de la HU (No.11): Gestionar incidencias.
Nombre de la tarea: Desarrollar Create, Read, Update, Delete (CRUD) de las incidencias.
Tipo de tarea: Desarrollo.
Programador responsable: Deyrin Pérez Pedroso y Carlos Fuentes Puentes.
Descripción: Implementa las funcionalidades correspondientes a las incidencias.

Tarea #11
Número de la HU (No.9): Gestionar puestos de trabajo.
Nombre de la tarea: Desarrollar los puestos de trabajo.
Tipo de tarea: Desarrollo.
Programador responsable: Deyrin Pérez Pedroso y Carlos Fuentes Puentes
Descripción: Implementa las funcionalidades de insertar, modificar, eliminar y visualizar los puestos de trabajo.

Anexo 3.Casos de pruebas de aceptación

Caso de prueba de aceptación	
Código: HU1_P2	Historia de usuario: Gestionar usuario.
Nombre: Modificar usuario.	
Descripción: Prueba para la funcionalidad que permite editar el perfil de usuario.	

<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> El usuario debe estar autenticado con el rol de administrador o de usuario.
<p>Pasos de ejecución:</p> <p>El administrador o usuario al acceder a la aplicación, selecciona la opción “Editar Usuario”, luego se muestran los datos referentes al usuario brindando la posibilidad de editar sus datos.</p>
<p>Resultado esperado: Autenticación satisfactoria.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Caso de prueba de aceptación	
Código: HU1_P2	Historia de usuario: Gestionar usuario.
Nombre: Eliminar usuarios.	
Descripción: Se muestran los datos del usuario a eliminar y el botón “Eliminar”, el cual elimina el usuario de la base de datos.	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol de administrador o usuario, la aplicación posee un administrador supremo al cual nadie puede eliminar. 	
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> El administrador al iniciar su sesión, en su pantalla inicial dispondrá del botón “Editar Usuario”. Se le mostrarán todos los usuarios del sistema, al lado de cada dato de los usuarios el sistema posee un botón “Editar”. Se le mostrará los datos del usuario seleccionado y un botón “Eliminar”. 	
Resultado esperado: Los datos son editados satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación

Código: HU1_P3	Historia de usuario: Gestionar usuario.
Nombre: Visualizar datos de los usuarios.	
Descripción: Prueba para la funcionalidad que permite visualizar los datos de los usuarios.	
Condiciones de ejecución: <ul style="list-style-type: none">✓ El usuario debe estar autenticado con el rol de administrador o usuario.	
Pasos de ejecución: <ol style="list-style-type: none">1. El administrador al acceder a la aplicación se le mostrará un botón en su pantalla inicial: "Editar Usuario".2. Una vez dentro se le mostrarán todos los datos de los usuarios relacionados con la aplicación.	
Resultado esperado: Los datos de los usuarios relacionados con el sistema son visualizados correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación

Código: HU2_P1	Historia de usuario: Autenticar usuario.
Nombre: Autenticar usuario.	
Descripción: Prueba para la funcionalidad que permite autenticar a los usuarios en la aplicación.	
Condiciones de ejecución: <ul style="list-style-type: none">✓ El usuario debe tener que haber sido registrado.	
Pasos de ejecución: <p>Al acceder a la aplicación el usuario previamente registrado introduce su usuario y contraseña, en caso de ser incorrectos cualquiera de los datos introducidos, la aplicación no le permitirá acceder a la misma.</p>	

Resultado esperado: La funcionalidad de autenticar usuario se realiza de forma correcta.	
Evaluación de la prueba: Prueba satisfactoria.	
Caso de prueba de aceptación	
Código: HU3_P2	Historia de usuario: Gestionar solicitud de instalación de Entornos de Prueba (EP).
Nombre: Modificar solicitud de instalación de EP.	
Descripción: Prueba para la funcionalidad que permite modificar la solicitud de instalación de EP.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> El usuario debe estar autenticado con el rol de administrador o usuario para acceder a todas las solicitudes, un usuario solo dispondrá de las solicitudes creadas por él. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> El administrador o especialista en su pantalla inicial visualizará todas las solicitudes de entornos de prueba creadas, en caso de un usuario normal solo dispondrá de las que el haya creado. Seleccionando la solicitud que desea modificar, se le mostraran todos los datos referentes a dicha solicitud. Finalmente ejecutando en el botón "Editar", podrá realizar los cambios deseados. Selecciona el botón "Cancelar", en caso de que desee regresar a la pantalla inicial, si no selecciona el botón "Aceptar" y es modificada la solicitud de EP. 	
Resultado esperado: La solicitud de instalación de EP es modificada de forma correcta.	
Evaluación de la prueba: Prueba satisfactoria.	
Caso de prueba de aceptación	
Código: HU3_P2	Historia de usuario: Gestionar solicitud de instalación de Entornos de Prueba (EP).

Nombre: Eliminar solicitud de instalación de EP.
Descripción: Prueba para la funcionalidad que permite eliminar solicitud de instalación de EP.
Condiciones de ejecución: <ul style="list-style-type: none"> El usuario debe estar autenticado con el rol de administrador o especialista para borrar cualquier solicitud de instalación de EP; un usuario solo dispondrá de las solicitudes creadas por él.
Pasos de ejecución: <ol style="list-style-type: none"> El administrador en su pantalla inicial visualizará todas las solicitudes de entornos de prueba creadas, en caso de un usuario normal solo dispondrá de las que el haya creado. Seleccionando la solicitud que desea modificar, se le mostraran todos los datos referentes a dicha solicitud. Finalmente ejecutando en el botón “Eliminar”, y la solicitud será eliminada. Selecciona el botón “Cerrar”, en caso de que desee regresar a la pantalla anterior.
Resultado esperado: La solicitud de instalación de EP es eliminada satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación

Código: HU3_P3	Historia de usuario: Gestionar solicitud de instalación de Entornos de Prueba (EP).
Nombre: Mostrar listado de solicitud de instalación de EP.	
Descripción: Prueba para la funcionalidad que permite mostrar el listado de la solicitudes de instalación de EP.	
Condiciones de ejecución: <ul style="list-style-type: none"> El usuario debe estar autenticado con el rol de administrador o de usuario para visualizar cualquier solicitud de instalación de EP; un usuario solo visualizará las solicitudes creadas por él. 	

Pasos de ejecución

1. El administrador o especialista en su pantalla inicial visualizará todas las solicitudes de entornos de prueba creadas, en caso de un usuario normal solo dispondrá de las que él haya creado.

Resultado esperado: Se visualizarán el listado de solicitudes de instalación de EP.

Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación

Código: HU3_P4

Historia de usuario: Gestionar solicitud de instalación de Entornos de Prueba (EP).

Nombre: Aceptar solicitud de instalación de EP.

Descripción: Prueba para la funcionalidad que permite aceptar la solicitud de instalación de EP.

Condiciones de ejecución:

- El usuario debe estar autenticado con el rol de administrador o de usuario.

Pasos de ejecución

El administrador llena los datos en un formulario con el nombre del departamento (es único), se presiona el botón "Crear", luego se muestra un mensaje de confirmación: "El departamento ha sido creado" y se agrega a la lista de los departamentos creados.

Resultado esperado: El departamento es creado satisfactoriamente.

Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación

Código: HU3_P5

Historia de usuario: Gestionar solicitud de instalación de Entornos de Prueba (EP).

Nombre: Rechazar solicitud de instalación de EP.

Descripción: Prueba para la funcionalidad que permite rechazar una solicitud de instalación de EP.

<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> El usuario debe estar autenticado con el rol de administrador o usuario.
<p>Pasos de ejecución</p> <p>El administrador o usuario al modificar los datos de una solicitud de instalación de EP, dispondrá de un campo para cambiar de “aceptada” a “rechazada”.</p>
<p>Resultado esperado: La solicitud de instalación de EP es rechazada satisfactoriamente.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Caso de prueba de aceptación	
<p>Código: HU3_P6</p>	<p>Historia de usuario: Gestionar solicitud de instalación de Entornos de Prueba (EP).</p>
<p>Nombre: Enviar notificación de solicitud de instalación de EP.</p>	
<p>Descripción: Prueba para la funcionalidad que permite enviar notificación de una solicitud de instalación de EP.</p>	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> El usuario debe estar autenticado con el rol de administrador o usuario. 	
<p>Pasos de ejecución</p> <p>El administrador o especialista al aceptar una solicitud de instalación de EP, la aplicación enviara una notificación a la persona que ha sido asignada dicha solicitud.</p>	
<p>Resultado esperado: La solicitud de instalación de EP es notificada satisfactoriamente.</p>	
<p>Evaluación de la prueba: Prueba satisfactoria.</p>	

Caso de prueba de aceptación	
<p>Código: HU3_P7</p>	<p>Historia de usuario: Gestionar solicitud de instalación de Entornos de Prueba (EP).</p>
<p>Nombre: Asignar solicitudes al rol especialista.</p>	

Descripción: Prueba para la funcionalidad que permite asignar solicitudes al rol usuario.
Condiciones de ejecución: <ul style="list-style-type: none"> El usuario debe estar autenticado con el rol de administrador o usuario.
Pasos de ejecución <ol style="list-style-type: none"> El administrador o usuario dispondrá al modificar una solicitud de instalación de EP, un campo donde introducirá el nombre del especialista al que desee asignar la solicitud de instalación de EP.
Resultado esperado: La solicitud de instalación de EP es asignada satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: HU3_P7	Historia de usuario: Gestionar solicitud de instalación de Entornos de Prueba (EP).
Nombre: Asignar solicitudes al rol usuario.	
Descripción: Prueba para la funcionalidad que permite asignar solicitudes al rol usuario.	
Condiciones de ejecución: <ul style="list-style-type: none"> El usuario debe estar autenticado con el rol de administrador o usuario. 	
Pasos de ejecución <ol style="list-style-type: none"> El administrador o especialista dispondrá al modificar una solicitud de instalación de EP, un campo donde introducirá el nombre del usuario al que desee asignar la solicitud de instalación de EP. 	
Resultado esperado: La solicitud de instalación de EP es asignada satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: HU4_P2	Historia de usuario: Gestionar entorno de prueba real.

Nombre: Modificar entorno de prueba real.
Descripción: Prueba para la funcionalidad que permite modificar un entorno de prueba real.
Condiciones de ejecución: <ul style="list-style-type: none"> El usuario debe estar autenticado con el rol de administrador o usuario para acceder a todos los entornos reales, en caso de ser un usuario solo tendrá acceso a sus entornos reales.
Pasos de ejecución <ol style="list-style-type: none"> El administrador o especialista dispondrá al modificar una solicitud de instalación de EP, un botón "Editar" donde podrá modificar los datos del entorno real que desee.
Resultado esperado: El entorno de prueba real es modificado satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: HU4_P3	Historia de usuario: Gestionar entorno de prueba real.
Nombre: Visualizar entorno de prueba real.	
Descripción: Prueba para la funcionalidad que permite visualizar un entorno de prueba real.	
Condiciones de ejecución: <ul style="list-style-type: none"> El usuario debe estar autenticado con el rol de administrador o usuario para acceder a todos los entornos reales, en caso de ser un usuario solo tendrá acceso a sus entornos reales. 	
Pasos de ejecución <ol style="list-style-type: none"> Los usuarios relacionados con la aplicación al acceder a su pantalla principal dispondrán de un link: "Ver Entorno Real", el cual les muestra todos los datos referentes al entorno real. 	
Resultado esperado: El entorno de prueba real es visualizado satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: HU4_P4	Historia de usuario: Gestionar entorno de prueba real.
Nombre: Eliminar entorno de prueba real.	
Descripción: Prueba para la funcionalidad que permite eliminar un entorno de prueba real.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> El usuario debe estar autenticado con el rol de administrador o usuario para acceder a todas los entornos reales, en caso de ser un usuario solo tendrá acceso a sus entornos reales. 	
Pasos de ejecución	
<ol style="list-style-type: none"> Los usuarios relacionados con la aplicación al acceder a su pantalla principal dispondrán de un link: "Ver Entorno Real", el cual les muestra todos los datos referentes al entorno real, una vez dentro dispondrá de un botón "Eliminar", el cual elimina el entorno de prueba real. 	
Resultado esperado: El entorno de prueba real es eliminado satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: HU4_P5	Historia de usuario: Gestionar entorno de prueba real.
Nombre: Eliminar entorno de prueba real.	
Descripción: Prueba para la funcionalidad que permite eliminar un entorno de prueba real.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> El usuario debe estar autenticado con el rol de administrador o usuario para acceder a todas los entornos reales, en caso de ser un usuario solo tendrá acceso a sus entornos reales. 	
Pasos de ejecución	
<ol style="list-style-type: none"> Los usuarios relacionados con la aplicación al acceder a su pantalla principal dispondrán de un link: "Ver Entorno Real", el cual les muestra todos los datos referentes al entorno real, una vez dentro dispondrá 	

de un botón “Eliminar”, el cual elimina el entorno de prueba real.

Resultado esperado: El entorno de prueba real es eliminado satisfactoriamente.

Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación

Código: HU5_P2

Historia de usuario: Gestionar rol.

Nombre: Visualizar rol.

Descripción: Prueba para la funcionalidad que permite visualizar un rol.

Condiciones de ejecución:

- El usuario debe estar autenticado con el rol de administrador.

Pasos de ejecución

1. Al administrador acceder a la aplicación en su pantalla inicial se le mostrará el botón “Editar Usuario”.
2. Posteriormente se le brindará la opción “Editar rol”.
3. Luego se le mostrará una pantalla donde están registrados todos los usuarios con sus roles correspondientes

Resultado esperado: El rol es visualizado satisfactoriamente.

Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación

Código: HU7_P2

Historia de usuario: Gestionar planificación.

Nombre: Modificar planificación.

Descripción: Prueba para la funcionalidad que permite modificar una planificación.
Condiciones de ejecución: <ul style="list-style-type: none"> El usuario debe estar autenticado con el rol de administrador.
Pasos de ejecución <ol style="list-style-type: none"> Los usuarios al acceder a la aplicación dispondrán en el menú la opción “Planificación” En el calendario dando clic encima de una planificación podrá modificar los datos de una planificación.
Resultado esperado: La planificación es modificada satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.
Caso de prueba de aceptación
Código: HU7_P3 Historia de usuario: Gestionar planificación.
Nombre: Visualizar planificación.
Descripción: Prueba para la funcionalidad que permite modificar una planificación.
Condiciones de ejecución: <ul style="list-style-type: none"> El usuario debe estar autenticado con el rol de administrador o usuario.
Pasos de ejecución <ol style="list-style-type: none"> Los usuarios al acceder a la aplicación dispondrán en el menú la opción “Planificación” En el calendario se visualizan todas las planificaciones realizadas.
Resultado esperado: La planificación es visualizada satisfactoriamente.

Evaluación de la prueba: Prueba satisfactoria.