



**Universidad
Técnica de
Cotopaxi**

UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

CARRERA DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS

COMPUTACIONALES

PROYECTO DE INVESTIGACIÓN

ANÁLISIS DE METODOLOGÍAS SCRUM Y XP EN LA IMPLEMENTACIÓN DE UN SISTEMA MULTIPLATAFORMA DE GESTIÓN EN EL BANCO DE GERMOPLASMA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ MEDIANTE TECNOLOGÍAS OPEN SOURCE.

Proyecto de investigación previo a la obtención del Título de Ingeniero en Informática y Sistemas Computacionales

AUTORES:

Manobanda Tuapanta Edwin Ricardo

Millingalli Calo Mariela Elizabeth

TUTOR:


Ing. M.Sc. Edel Ángel Rodríguez Sánchez

**LA MANÁ-ECUADOR
FEBRERO-2020**

DECLARACIÓN DE AUTORÍA

Yo Manobanda Tuapanta Edwin Ricardo y Millingalli Calo Mariela Elizabeth declaramos ser autores del presente proyecto de investigación: ANÁLISIS DE METODOLOGÍAS SCRUM Y XP EN LA IMPLEMENTACIÓN DE UN SISTEMA MULTIPLATAFORMA DE GESTIÓN EN EL BANCO DE GERMOPLASMA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ MEDIANTE TECNOLOGÍAS OPEN SOURCE, siendo Ing. M.Sc Edel Ángel Rodríguez Sánchez tutor del presente trabajo; y eximo expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.



Manobanda Tuapanta Edwin Ricardo
C.I: 020222445-7



Millingalli Calo Mariela Elizabeth
C.I: 050425850-0

AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN

En calidad de Tutor del Trabajo de Investigación sobre el título:

ANÁLISIS DE METODOLOGÍAS SCRUM Y XP EN LA IMPLEMENTACIÓN DE UN SISTEMA MULTIPLATAFORMA DE GESTIÓN EN EL BANCO DE GERMOPLASMA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ MEDIANTE TECNOLOGÍAS OPEN SOURCE, de Manobanda Tuapanta Edwin Ricardo y Millingalli Calo Mariela Elizabeth, de la carrera de Ingeniería en Informática y Sistemas Computacionales, considero que dicho Informe Investigativo cumple con los requerimientos metodológicos y aportes científico-técnicos suficientes para ser sometidos a la evaluación del Tribunal de Validación de Proyecto que el Honorable Consejo Académico de la Facultad de Ciencias de la Ingeniería y Aplicadas de la Universidad Técnica de Cotopaxi designe, para su correspondiente estudio y calificación.

La Maná, Febrero 2020



Ing. M.Sc. Rodríguez Sánchez Edel Ángel
C.I: 175722381-1
TUTOR

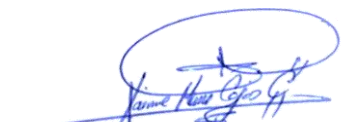
APROBACIÓN DEL TRIBUNAL DE TITULACIÓN


En calidad de Tribunal de Lectores, aprueban el presente Informe de Investigación de acuerdo a las disposiciones reglamentarias emitidas por la Universidad Técnica de Cotopaxi, y por la Unidad Académica de Ciencias de la Ingeniería y Aplicadas; por cuanto, los postulantes: Manobanda Tuapanta Edwin Ricardo y Millingalli Calo Mariela Elizabeth con el título de Proyecto de Investigación: ANÁLISIS DE METODOLOGÍAS SCRUM Y XP EN LA IMPLEMENTACIÓN DE UN SISTEMA MULTIPLATAFORMA DE GESTIÓN EN EL BANCO DE GERMOPLASMA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ MEDIANTE TECNOLOGÍAS OPEN SOURCE, han considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de Sustentación de Proyecto.


Por lo antes expuesto, se autoriza realizar los empastados correspondientes, según la normativa institucional.

La Maná, Febrero del 2020

Para constancia firman:


Ing. Mgr. Cajas Jaime Mesías
C.I: 050235925-0
LECTOR 1 (PRESIDENTE)


Ing. M.Sc. Silva Peñafiel Geovanny Euclides
C.I: 060289176-4
LECTOR 2


Ing. M.Sc. Bajaña Zajia Johnny Xavier
C.I: 120482711-5
LECTOR 3 (SECRETARIO)

AVAL DE IMPLEMENTACIÓN

El suscrito, Ing. Mgtr. Jaime Mesías Cajas con cédula de ciudadanía No. 050235925-0, Docente encargado del Laboratorio de Software de la Universidad Técnica de Cotopaxi extensión La Maná certifico que:

Los señores: Manobanda Tuapanta Edwin Ricardo y Millingalli Calo Mariela Elizabeth, estudiantes de la Universidad Técnica de Cotopaxi extensión La Maná, cumplieron a cabalidad con la realización e implementación del sistema en el mini DataCenter del Laboratorio de Software desde el cual se realiza la administración del sistema que se vincula al departamento de Germoplasma por medio de la dirección IP de conexión a la red de la UTC-La Maná, dando cumplimiento al proyecto de “IMPLEMENTACIÓN DE UN SISTEMA MULTIPLATAFORMA DE GESTIÓN EN EL BANCO DE GERMOPLASMA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ MEDIANTE TECNOLOGÍAS OPEN SOURCE”, el mismo que cumple con todo los requerimientos establecidos en el transcurso de su investigación.

Es todo en cuanto puedo certificar en honor a la verdad y autorizo a los peticionarios hacer uso del presente certificado de la manera ética que estimare conveniente, siempre y cuando este dentro de las leyes.

La Maná, Febrero del 2020

Atentamente;



Ing. Mgtr. Jaime Mesías Cajas

C.I: 050235925-0

DIRECTOR DE LA CARRERA DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN

AGRADECIMIENTO

En primer lugar agradezco a Dios por protegerme y cuidarme durante todo mi trayecto de vida. A mis padres hermanos y familiares por su apoyo incondicional en los buenos y malos momentos que he pasado, gracias por haberme cuidado y ayudado en toda mi formación profesional y vida.

A la Universidad Técnica de Cotopaxi en conjunto con todo su equipo de colaboradores, en especial a los docentes de la carrera de Ingeniería en Informática y Sistemas Computacionales, por guiarnos así una educación de excelencia, a mi compañera Mariela Millingalli, quien ha estado presente apoyando en el transcurso evolutivo de nuestra tesis.

Edwin M.

En este espacio deseo agradecer infinitamente a Dios quien fue el ente primordial que supo guiar mi camino, dándome aliento de vida, para poder realizar diferentes actividades, a la vez también agradecer a mis padres, a los docentes pertenecientes a la carrera de ingeniería en Informática quienes estuvieron en cada ciclo universitario demostrando su capacidad intelectual, a mi compañero Edwin Manobanda, quien ha estado presente en cada proceso evolutivo de nuestra tesis.

Agradezco a la Universidad Técnica de Cotopaxi, por ser acogedora de todos los jóvenes quienes desean estudiar en esta prestigiosa institución, ya que nos ha brindado una grandiosa oportunidad de formarnos como futuros profesionales.

Mariela M.

DEDICATORIA

El presente proyecto lo dedico a mis padres Ángel y María por estar a mi lado apoyándome y guiándome en todos los procesos de mi formación profesional, a mis hermanos y toda mi familia por su apoyo y ánimos recibidos, por confiar en mis capacidades, logrando tener como resultado la culminación de mi formación universitaria.

Edwin M.

La presente tesis está dedicada a mis padres, a mi familia en general debido a que sin su apoyo no hubiese logrado llegar hasta el lugar donde me encuentro, ya que en momentos de dificultad han estado apoyándome, seguidamente a mis docentes quienes en la etapa universitaria han sabido compartir sus conocimientos, de esta manera mejorando nuestro talento como estudiante, dándonos ánimos a seguir luchando por alcanzar nuestros sueños.

Mariela M.

UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

TITULO: “ANÁLISIS DE METODOLOGÍAS SCRUM Y XP EN LA IMPLEMENTACIÓN DE UN SISTEMA MULTIPLATAFORMA DE GESTIÓN EN EL BANCO DE GERMOPLASMA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ MEDIANTE TECNOLOGÍAS OPEN SOURCE.”

Autores:

Manobanda Tuapanta Edwin Ricardo

Millingalli Calo Mariela Elizabeth

RESUMEN

Las metodologías son modelos, marcos o métodos que permiten llevar un control de la estructura de los proyectos, contiene fases, roles y ciclos de vida de desarrollo permitiendo llevar un orden y estructuración en la realización de un proyecto de software estimando el tiempo de desarrollo, a través de métodos de trabajo para satisfacer las necesidades u objetivos específicos de diseño y desarrollo de proyectos. El proyecto desarrollado tuvo como objetivo principal la elaboración de un análisis comparativo sobre las metodologías ágiles Scrum y XP (Extreme Programming), aplicado al diseño de un software multiplataforma de gestión de especies de semillas en el Banco de Germoplasma de la Universidad Técnica de Cotopaxi extensión La Maná desarrollado mediante tecnologías informáticas Open Source, con la finalidad de obtener la metodología que mejor se adapte al diseño y desarrollo del software de gestión de especies de semillas del Banco de Germoplasma que faciliten su ordenamiento, organización, revisión y fase de desarrollo estimado de acuerdo a los requerimientos funcionales y peticiones del cliente que solicite el software, permitiendo calcular el tiempo y costo de un proyecto de desarrollo de software. Para la realización del proyecto de investigación se aplicaron metodologías de investigación como: investigación bibliográfica para la elaboración de la fundamentación científico técnico del proyecto a través de la obtención de información fidedigna de revistas científicas y libros que contribuyan y sustente el tema de investigación, además también se empleó el método analítico e hipotético deductivo, para la generación de un análisis de las metodologías Scrum y XP, al mismo tiempo se formuló una hipótesis como posible solución y comprobación que sustente lo investigado y posible solución a la problemática del proyecto. Como resultado al aplicar la metodología ágil Scrum y XP al diseño del software, se obtuvo como resultados que la metodología Scrum es mucho más eficiente en proyectos de desarrollo e implementación de software a corto tiempo con la ventaja de realizar cambios futuros durante el diseño del sistema sin afectar la estructuración y fecha de entrega del software, lo cual diferencia a la metodología XP que es más centralizada al diseño de proyectos de gran escala que conlleve un alto nivel de estructuración de los datos y códigos.

Palabras claves: Metodología, SCRUM, XP, Sistema, Multiplataforma, Open Source.

UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

TITLE: "SCRUM AND XP METHODOLOGY ANALYSIS IN THE IMPLEMENTATION OF A MULTIPLATFORM MANAGEMENT SYSTEM IN THE GERMPLASM BANK OF THE TECHNICAL UNIVERSITY OF COTOPAXI EXTENSIÓN LA MANÁ THROUGH OPEN SOURCE TECHNOLOGIES."

Author:

Manobanda Tuapanta Edwin Ricardo

Millingalli Calo Mariela Elizabeth

ABSTRACT

The methodologies are models, frameworks or methods that allow to take the control of the structure of the projects, contains phases, roles and development life cycles allowing to carry out an order and structuring in the realization of a software project estimating the development time, through work methods to meet the specific needs or objectives of project design and development. The main objective of the project was the development of a comparative analysis on the agile methodologies Scrum and XP (Extreme Programming), applied to the design of a multiplatform software for seed species management in the Germplasm Bank of the Technical University of Cotopaxi extension La Maná developed through Open Source information technologies, with the purpose of obtaining the methodology which suits the design and development of the seed management software of the Germplasm Bank that facilitate its ordering, organization, review and estimated development phase according to the functional needs and requests of the client that ask for the software, allowing to calculate the time and cost of a software development project. For the realization of the research project, research methodologies were applied such as: bibliographic research for the development of the technical scientific foundation of the project through the getting of reliable information from scientific journals and books that contribute and support the research topic, in addition also the deductive analytical and hypothetical method was used, for the generation of an analysis of the Scrum and XP methodologies, at the same time a hypothesis was formulated as a possible solution and verification that supports what was investigated and possible solution to the problem of the project. As a result of applying the agile Scrum and XP methodology to software design, it was obtained as a result that the Scrum methodology is much more efficient in short-term software development and implementation projects with the advantage of making future changes during system design. Without affecting the structuring and delivery date of the software, which distinguish from the XP methodology that is more centralized to the design of large-scale projects that entails a high level of data and code structuring.

Keywords: Methodology, SCRUM, XP, System, Multiplatform, Open Source

AVAL DE TRADUCCIÓN

En calidad de Docente del Idioma Inglés del Centro de Idiomas de la Universidad Técnica de Cotopaxi Extensión La Maná; en forma legal CERTIFICO que: la traducción del resumen del proyecto de investigación al idioma Inglés presentado por los estudiantes de la carrera de Ingeniería en Informática y Sistemas Computacionales de la facultad de Ciencias de la Ingeniería y Aplicadas, Manobanda Tuapanta Edwin Ricardo y Millingalli Calo Mariela Elizabeth cuyo título versa “ANÁLISIS DE METODOLOGÍAS SCRUM Y XP EN LA IMPLEMENTACIÓN DE UN SISTEMA MULTIPLATAFORMA DE GESTIÓN EN EL BANCO DE GERMOPLASMA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ MEDIANTE TECNOLOGÍAS OPEN SOURCE”, lo realizó bajo mi supervisión y cumple con una correcta estructura gramatical del idioma.

Es todo en cuanto puedo certificar en honor a la verdad y autorizo a los peticionarios hacer uso del presente certificado de la manera ética que estimare conveniente.

La Maná, Febrero del 2020

Atentamente,



Mg. Sebastián Fernando Ramón Amores
C.I: 050301668-5

COORDINADOR CENTRO CULTURAL DE IDIOMAS UTC-LA MANÁ

ÍNDICE GENERAL

| | |
|--|------|
| PORTADA..... | i |
| DECLARACIÓN DE AUTORÍA | ii |
| AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN..... | iii |
| APROBACIÓN DEL TRIBUNAL DE TITULACIÓN | iv |
| AVAL DE IMPLEMENTACIÓN..... | v |
| AGRADECIMIENTO..... | vi |
| DEDICATORIA | vii |
| RESUMEN..... | viii |
| ABSTRACT..... | ix |
| AVAL DE TRADUCCIÓN | x |
| ÍNDICE GENERAL..... | xi |
| ÍNDICE DE TABLAS | xv |
| ÍNDICE DE FIGURAS..... | xvi |
| ÍNDICE DE GRÁFICOS | xvii |
| 1. INFORMACIÓN GENERAL..... | 1 |
| 2. RESUMEN (DESCRIPCIÓN) DEL PROYECTO | 2 |
| 3. JUSTIFICACIÓN DEL PROYECTO..... | 2 |
| 4. BENEFICIARIOS DEL PROYECTO | 3 |
| 5. PROBLEMA DE INVESTIGACIÓN..... | 4 |
| 6. OBJETIVOS | 5 |
| 7. ACTIVIDADES Y SISTEMA DE TAREAS EN RELACIÓN A LOS OBJETIVOS PLANTEADOS | 6 |
| 8. FUNDAMENTACIÓN CIENTIFICO TÉCNICA..... | 7 |
| 8.1. Metodologías Agiles..... | 7 |
| 8.2. Metodologías, técnicas y herramientas | 7 |
| 8.2.1. Principales características de las metodologías agiles:..... | 8 |
| 8.2.2. Metodologías agiles más importantes | 9 |
| 8.2.2.1. Crystal Methodologies | 10 |
| 8.2.2.2. Kanban..... | 10 |
| 8.2.2.3. Programación Xtreme | 10 |
| 8.2.2.3.1. Roles..... | 11 |
| 8.2.2.3.2. Prácticas de Extreme Programming | 12 |
| 8.2.2.3.3. Usos de metáforas y analogías | 13 |
| 8.2.2.3.4. Ciclo de vida XP..... | 13 |
| 8.2.2.3.5. Ventajas de XP | 14 |

| | | |
|------------|---|----|
| 8.2.2.3.6. | Contras de XP | 14 |
| 8.2.2.4. | Scrum..... | 15 |
| 8.2.2.4.1. | Marco de trabajo de Scrum | 15 |
| 8.2.2.4.2. | Roles | 16 |
| 8.2.2.4.3. | Artefactos..... | 16 |
| 8.2.2.4.4. | Pilares de Scrum | 16 |
| 8.2.2.4.5. | Ciclo de vida de Scrum | 18 |
| 8.2.2.4.6. | Ventajas de Scrum | 19 |
| 8.2.2.4.7. | Desventajas de Scrum | 19 |
| 8.2.2.4.8. | Costo, plazo y perímetro | 19 |
| 8.3. | Similitudes y diferencias entre la metodología Scrum y XP | 20 |
| 8.4. | Evaluación de metodologías | 20 |
| 8.5. | Sistema | 21 |
| 8.6. | Sistema informático | 21 |
| 8.7. | Multiplataforma | 22 |
| 8.8. | Gestión..... | 22 |
| 8.9. | Germoplasma..... | 22 |
| 8.10. | Base de datos | 22 |
| 8.11. | Software..... | 23 |
| 8.12. | Open Source | 23 |
| 8.12.1. | Funciones de Open Source..... | 24 |
| 8.12.2. | Licencias de hardware de código abierto | 24 |
| 8.12.3. | Ejemplos de hardware de código abierto..... | 24 |
| 8.13. | Herramientas open source | 24 |
| 8.14. | Ventajas | 25 |
| 8.15. | Tecnologías open source de programación del lado del cliente..... | 25 |
| 8.15.1. | JavaScript..... | 25 |
| 8.15.2. | Características..... | 25 |
| 8.16. | Tecnologías Open Source de programación del lado servidor | 26 |
| 8.17. | Entorno de desarrollo integrado (IDE) | 26 |
| 8.17.1. | Eclipse | 26 |
| 8.17.2. | Netbeans | 26 |
| 8.17.2.1. | Ventajas | 26 |
| 9. | VALIDACIÓN DE PREGUNTAS CIENTÍFICAS O HIPOTESIS | 27 |
| 10. | METODOLOGÍAS Y DISEÑO EXPERIMENTAL | 27 |
| 10.1. | Tipo de investigación..... | 27 |
| 10.1.1. | Investigación bibliográfica..... | 27 |

| | | |
|-----------|---|----|
| 10.1.2. | Investigación descriptiva..... | 27 |
| 10.2. | Métodos de investigación..... | 28 |
| 10.2.1. | Método Analítico | 28 |
| 10.2.2. | Método Hipotético Deductivo..... | 28 |
| 10.3. | Metodología Scrum para el desarrollo de software | 28 |
| 10.4 | Técnica de investigación | 29 |
| 10.4.1 | Entrevista..... | 29 |
| 10.4.2 | Encuesta | 29 |
| 10.4.3 | Población y muestra..... | 29 |
| 10.4.3.1 | Población..... | 29 |
| 10.4.3.2 | Muestra..... | 30 |
| 11. | ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS | 31 |
| 11.1 | Proceso para el diseño del sistema | 31 |
| 11.2 | Aplicación de la metodología Ágil Scrum | 32 |
| 11.2.1 | Proceso de desarrollo de la metodología Scrum | 32 |
| 11.2.1.1 | Product Backlong..... | 32 |
| 11.2.1.2 | Objetivos del Product Backlong..... | 34 |
| 11.2.1.3 | Sprint Planning | 34 |
| 11.2.1.4 | Sprint Bakclong | 34 |
| 11.2.1.5 | Revisión de sprint | 35 |
| 11.2.1.6 | Requerimientos funcionales..... | 36 |
| 11.2.1.7 | Requisitos no funcionales | 37 |
| 11.2.1.8 | Casos de uso | 38 |
| 11.2.1.9 | Entrega de Proyecto | 48 |
| 11.3. | Aplicación de metodología ágil XP | 48 |
| 11.3.1. | Descripción del cliente | 48 |
| 11.3.2. | Planeación..... | 49 |
| 11.3.2.1. | Historia de usuarios | 49 |
| 11.3.2.2. | Asignación de roles del proyecto | 52 |
| 11.3.2.3. | Plan de entrega del proyecto | 52 |
| 11.3.3. | Diseño..... | 53 |
| 11.3.4. | Codificación..... | 54 |
| 11.3.5. | Pruebas | 56 |
| 11.4. | Análisis comparativo de metodologías ágiles Scrum y XP (Programación Xtreme)..... | 57 |
| 11.5. | Resultados de comparación de metodologías Scrum y XP aplicado al sistema..... | 58 |
| 12. | IMPACTOS (TÉCNICOS, SOCIALES, AMBIENTALES, O ECONÓMICOS). | 60 |
| 12.1. | Impacto técnico..... | 60 |

| | | |
|-------|--|----|
| 12.2. | Impacto social..... | 60 |
| 12.3. | Impacto ambiental..... | 60 |
| 12.4. | Impacto económico..... | 60 |
| 13. | PRESUPUESTO PARA LA PROPUESTA DEL PROYECTO | 60 |
| 14. | CONCLUSIONES Y RECOMENDACIONES | 62 |
| 15. | BIBLIOGRAFÍA | 63 |
| 16. | ANEXOS | 66 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1: Beneficiarios | 3 |
| Tabla 2: Actividades y tareas. | 6 |
| Tabla 3: Similitudes y diferencias entre la metodología Scrum y Programación Extrema | 20 |
| Tabla 4: Requisito Ingresar datos de inventario de semillas | 32 |
| Tabla 5: Requisito Guardar datos de inventario de semillas | 32 |
| Tabla 6: Requisito Modificar datos de inventario de semillas | 33 |
| Tabla 7: Requisito Editar datos de inventario de semillas | 33 |
| Tabla 8: Requisito Autenticar ingreso de usuario | 33 |
| Tabla 9: Requisito Almacenar información | 33 |
| Tabla 10: Product Backlong | 34 |
| Tabla 11: Requerimientos funcionales del software | 36 |
| Tabla 12: Inicio sesión administrador | 38 |
| Tabla 13: Validar autenticación de ingreso al sistema..... | 39 |
| Tabla 14: Registra usuario..... | 39 |
| Tabla 15: Acceder a todo los recursos del sistema | 39 |
| Tabla 16: Eliminar datos | 40 |
| Tabla 17: Generar reportes de semilla | 40 |
| Tabla 18: Visualizar datos almacenados..... | 40 |
| Tabla 19: Generar rol de usuarios | 41 |
| Tabla 20: Cerrar sesión | 41 |
| Tabla 21: Iniciar sesión | 41 |
| Tabla 22: Autenticar usuario..... | 42 |
| Tabla 23: Registrar semilla..... | 42 |
| Tabla 24: Registrar colector | 42 |
| Tabla 25: Buscar semilla | 43 |
| Tabla 26: Cerrar sesión | 43 |
| Tabla 27: Verificación de entrega de resultados | 48 |
| Tabla 28: Historia acceso al sistema..... | 50 |
| Tabla 29: Historia gestionar usuarios | 50 |
| Tabla 30: Historia registro de semillas | 50 |
| Tabla 31: Historia registro de colector | 51 |
| Tabla 32: Historia registro de colección y localización | 51 |
| Tabla 33: Historia acceso al sistema..... | 51 |
| Tabla 34: Asignación de roles del proyecto..... | 52 |
| Tabla 35: Plan de entrega del proyecto..... | 52 |
| Tabla 36: Ventajas y desventajas de metodologías Scrum y XP | 58 |

| | |
|---|----|
| Tabla 37: Valores de equivalencia..... | 59 |
| Tabla 38: Comparación de metodología..... | 59 |
| Tabla 39: Presupuesto del proyecto..... | 61 |
| Tabla 40: Herramientas tecnológicas Open Source | 61 |
| Tabla 41: Porcentaje de aplicación de metodologías | 68 |
| Tabla 42: Tipos de metodologías | 69 |
| Tabla 43: Tiempo y cronograma de ejecución..... | 70 |
| Tabla 44: Porcentaje de conocimiento de metodologías..... | 71 |
| Tabla 45: Utilización de metodologías | 72 |
| Tabla 46: Control y planificación..... | 73 |
| Tabla 47: Razón principal de utilizar metodologías | 74 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1: Resultados de la encuesta de Forrester | 9 |
| Figura 2: Ciclo de vida de la metodología XP..... | 13 |
| Figura 3: Ciclo de vida de la metodología Ágil Scrum..... | 18 |
| Figura 4: Caso de uso Acciones que realiza el administrador en el sistema | 38 |
| Figura 5: Acciones que realiza el usuario | 41 |
| Figura 6: Interfaz inicio de sesión | 44 |
| Figura 7: Interfaz registro de semillas | 44 |
| Figura 8: Interfaz registro de colección | 45 |
| Figura 9: Interfaz registro de colectores de semilla | 45 |
| Figura 10: Interfaz Registro de Usuarios..... | 46 |
| Figura 11: Interfaz validación de los campos | 46 |
| Figura 12: Modelo relacional de la base de datos..... | 47 |
| Figura 13: Reporte de semilla | 47 |
| Figura 14: Ingreso al sistema..... | 53 |
| Figura 15: Roles que puede cambiar el administrador..... | 53 |
| Figura 16: Registro de semillas | 54 |
| Figura 17: Script base de datos, localidad | 55 |
| Figura 18: Script de las tablas familia y especie de semillas | 55 |
| Figura 19: Validación de los campos | 56 |
| Figura 20: Código Validación de los campos | 56 |
| Figura 21: Código de conexión a base de datos..... | 57 |
| Figura 22: Conexión a la base de datos | 57 |
| Figura 23: Prueba de funcionamiento del script de la base de datos..... | 75 |
| Figura 24: Mensaje de conexión a la base de datos | 76 |

| | |
|--|----|
| Figura 25: Validación de los campos | 76 |
| Figura 26: Tabulación de los resultados de la encuesta | 77 |
| Figura 27: Diseño de la interfaz gráfica de usuario. | 77 |

ÍNDICE DE GRÁFICOS

| | |
|--|----|
| Gráfico 1: Porcentaje de aplicación de metodologías | 68 |
| Gráfico 2: Tipos de metodologías | 69 |
| Gráfico 3: Tiempo y cronograma de ejecución..... | 70 |
| Gráfico 4: Porcentaje de conocimiento de metodologías..... | 71 |
| Gráfico 5: Utilización de metodología | 72 |
| Gráfico 6: Control y planificación..... | 73 |
| Gráfico 7: Razón principal de aplicar metodologías..... | 74 |

1. INFORMACIÓN GENERAL

Título del Proyecto:

Análisis de metodologías SCRUM y XP en la implementación de un sistema multiplataforma mediante tecnologías Open Source (AMISOS).

Fecha de inicio: 27 de Marzo del 2019

Fecha de finalización: 20 de Febrero del 2020

Lugar de ejecución: Av. Los Almendros y calle Pujilí sector La Virgen. La Maná. Cotopaxi, Zona 3. Universidad Técnica de Cotopaxi Extensión La Maná

Facultad: Facultad de Ciencias de la Ingeniería y Aplicadas

Carrera que auspicia: Ingeniería en Informática y Sistemas Computacionales CIYA

Proyecto de investigación vinculado: Universidad Técnica de Cotopaxi Extensión La Maná

Equipo de trabajo:

Tutor: Ing. Mgtr. Edel Ángel Rodríguez Sánchez

Correo: edel.rodriguez@utc.edu.ec

Teléfono: 0989930089

Estudiante: Manobanda Tuapanta Edwin Ricardo

Correo: edwin.manobanda4457@utc.edu.ec

Teléfono: 0989280283

Estudiante: Millingalli Calo Mariela Elizabeth

Correo: mariela.millingalli8500@utc.edu.ec

Teléfono: 0986420850

Área de conocimiento: Desarrollo de software

Línea de Investigación: Tecnologías de la Información y Comunicación (TICs) y Diseño Gráfico

Sub líneas de investigación de la carrera: Ciencias Informáticas para la modelación de Sistemas de Información a través del desarrollo de software.

2. RESUMEN (DESCRIPCIÓN) DEL PROYECTO

El presente proyecto tiene como objetivo analizar las metodologías Scrum y XP en la implementación de un sistema multiplataforma de gestión desarrollado con tecnologías Open Source, con la finalidad de obtener la metodología adecuada para el desarrollo de sistemas informáticos que faciliten su estructuración, planificación, control y tiempo de desarrollo estimado de acuerdo a los requerimientos, permitiendo evaluar el tiempo y costo de un proyecto de software.

En ocasiones existe la necesidad de llevar un control y planificación del desarrollo de un proyecto, por ello son importantes la aplicación de metodologías de desarrollo de software de las cuales se mencionan las más utilizadas, sean para proyectos cortos o a largo plazo, las metodologías ágiles Scrum y XP son adecuadas a la hora de llevar una planificación de la ejecución de proyectos software. Cada metodología está compuesta por fases o ciclos de desarrollo de sistemas

Uno de los problemas que surgen al desarrollar software es de no saber que metodología de software es adecuada para la planificación de desarrollo del proyecto para ello se debe analizar cada una de las metodologías que existe y optar por una que se acople al tiempo que se requiere para su desarrollo.

Para la ejecución del proyecto se aplicaran metodologías de investigación como: investigación bibliográfica para la obtención de información verídica de libros y revistas científicas que aporten información coherente al tema de investigación, además también se emplea el método analítico e hipotético deductivo, permitiendo una vez obtenida la información aplicar un análisis de las metodologías Scrum y XP, al mismo tiempo generando una hipótesis que sustente lo investigado.

3. JUSTIFICACIÓN DEL PROYECTO

A medida que evolucionan las tecnologías informáticas las empresas e instituciones a nivel mundial, han optado por la automatización y digitalización de sus procesos administrativos de gestión informativa, a través de las implementaciones de sistemas informáticos adecuados a sus necesidades que faciliten dar soluciones a sus problemas de almacenamiento y gestión de la información.

En la actualidad los sistemas se ha convertido en un factor importante e indispensable en el día a día de las personas e instituciones, su constante evolución y surgimiento dan la solución a las problemáticas y necesidades de las empresas e instituciones facilitando su trabajo sin esfuerzo, obteniendo procesos rápidos, seguros al momento de gestionar, automatizar los procesamientos de información y la optimización de los recursos que manejan las empresas e instituciones.

En ocasiones, al desarrollar software no se tiene en cuenta la metodología adecuada lo cual no permite tener una planificación al diseñar dicho software. Para esto es recomendable realizar un pequeño estudio de las metodologías existentes optando por la que se ajuste mejor al de diseño de los sistemas software.

Por lo antes mencionado es importante la realización del análisis de las metodologías Scrum y XP para la implementación de un sistema multiplataforma desarrollado con tecnologías Open Source. Esto permite llevar una planificación, tiempo y presupuesto del proyecto.

4. BENEFICIARIOS DEL PROYECTO

Los beneficiarios directos del proyecto, serán los estudiantes de la carrera de Ingeniería en Informática y Sistemas Computacionales y el departamento del Banco de Germoplasma de la Universidad Técnica de Cotopaxi (UTC), en conjunto con los beneficiarios indirectos que conforma la cantidad de 1.700 estudiantes de la Universidad Técnica de Cotopaxi extensión La Maná, donde el 48,5% son varones y el 51,5% son mujeres aproximadamente.

Tabla 1: Beneficiarios

| BENEFICIARIOS DEL PROYECTO | | |
|-----------------------------------|---|-------|
| Directos | Estudiantes de la carrera de Ingeniería en Informática y Sistemas Computacionales | 250 |
| | Departamento de Germoplasma | 2 |
| Indirectos | Estudiantes de la UTC | 1.700 |
| TOTAL | | 1.952 |

Elaborado por: Los Autores

Fuente: UTC-La Maná

5. PROBLEMA DE INVESTIGACIÓN

A nivel mundial los países han optado por la automatización y digitalización de la información empresarial e institucional, optimizando los procesos administrativos, como por ejemplo Chile cuenta con una plataforma web SAG (Servicios Agrícolas y Ganaderos), contiene un área dedicado a la publicación de semillas que posee el país y las investigaciones que se desarrollan para el tratamiento y almacenamiento de la información de la semilla, otro caso es la Universidad ICESI de Colombia que cuenta con un sistema de administración de calidad para los procesos de conservación del banco de semillas de frijoles tropicales. Pero en ocasiones se tiene pérdida de calidad de software al no emplear una metodología ágil al desarrollar el software. La aplicación de un análisis de las metodologías ágiles de desarrollo de software, permiten optar por la metodología adecuada entorno a la estructuración, planificación y control del desarrollo del sistema software que garantice su calidad. En Perú se han desarrollado proyectos similares al análisis de metodologías Scrum y XP, una de ellas es el desarrollo e implementación de un sistema web de gestión documentaria aplicando las metodologías Scrum y XP con el fin de mejorar los procesos de venta de la empresa BRANUSAC. También se han desarrollado sistemas basados en la metodología Scrum y XP para el proceso de ventas de servicios de la empresa EMSOIR.

A nivel nacional se han desarrollado proyectos de investigación y desarrollo de software implementando combinaciones de las metodologías ágiles Scrum y XP, uno de los proyectos trata sobre el desarrollo de un sistema de control escolar para la escuela bilingüe Sangay, desarrollado por estudiantes de la Escuela Politécnica del Ejército. También en la ciudad de Latacunga se desarrolló un proyecto de análisis comparativo de las metodologías de desarrollo Mobil Hybrid Methodology Desing y Mobile D de una aplicación de visita virtual por parte de estudiantes de la Universidad Técnica de Cotopaxi. En Ecuador aún existe la deficiencia en aplicación de análisis de las metodologías ágiles adecuados a los proyectos de desarrollo de software.

En el cantón La Maná debido a la inexistencia de proyectos encaminados al análisis de metodologías Scrum y XP en la implementación de un sistema multiplataforma para la gestión de especies de semillas, es necesario la ejecución de la investigación y análisis de las metodologías ágiles permitiendo dar a conocer los resultados y optar por la metodología adecuada para el desarrollo de un proyecto software, obteniendo una estructuración,

planificación y control en el diseño del software mejorando su calidad. Actualmente la existencia de varias metodologías para el diseño de sistemas genera complicaciones al escoger la metodología adecuada para el desarrollo de software, para escoger una metodología se debe especificar el tiempo al cual estará enmarcado la entrega del software. La realización de un análisis de metodologías permitirá la selección de una metodología acorde a la planificación de realización del proyecto, realizando los procedimientos que posee cada una de las fases del ciclo de desarrollo del software de la metodología escogida.

6. OBJETIVOS

Objetivo General

- Analizar las metodologías Scrum y XP en la implementación de un sistema multiplataforma de gestión desarrollado con tecnologías Open Source.

Objetivos Específicos

- Investigar información sobre las metodologías ágiles de desarrollo de software que más se utilizan en la actualidad.
- Realizar un análisis comparativo de las metodologías ágiles Scrum y XP en el desarrollo de software.
- Plantear la metodología ágil a aplicarse en el desarrollo del software de gestión mediante las tecnologías seleccionadas.

7. ACTIVIDADES Y SISTEMA DE TAREAS EN RELACIÓN A LOS OBJETIVOS PLANTEADOS

Tabla 2: Actividades y tareas.

| Objetivo específico 1 | Actividad | Resultado de la actividad | Medio de verificación |
|---|---|--|--|
| Investigar información sobre las metodologías ágiles de desarrollo de software que más se utilizan en la actualidad. | Exploración de información de metodologías ágiles en artículos científicos, tesis y libros. | Recopilación de información de las metodologías de diseño de software | Marco Teórico |
| Objetivo específico 2 | Actividad | Resultado de la actividad | Medio de verificación |
| Realizar un análisis comparativo de las metodologías ágiles Scrum y XP en el desarrollo de software. | Análisis de las ventajas y desventajas de las metodologías Scrum y XP | Obtención de información y toma de decisión acerca de las metodologías para desarrollo del proyecto. | Cuadro comparativo entre la metodología Scrum y XP. |
| Objetivo específico 3 | Actividad | Resultado de la actividad | Medio de verificación |
| Plantear la metodología ágil a aplicarse en el desarrollo del software de gestión mediante las tecnologías seleccionadas. | *Selección de la metodología para el desarrollo del sistema. *Desarrollo del sistema. | *Planteamiento de la metodología para el desarrollo del proyecto. *Diseño del sistema informático multiplataforma de gestión, desarrollado de manera adecuada con herramientas Open Source. | *Resultado del cuadro comparativo. *Utilización de base de datos MySQL, lenguaje de programación java NetBeans IDE. . |

Fuente: Los Autores

8. FUNDAMENTACIÓN CIENTIFICO TÉCNICA

8.1. Metodologías Agiles

Agile es un conjunto de métodos y metodologías que ayudan a su equipo a pensar de manera más efectiva, trabajar de manera más eficiente y tomar mejores decisiones.

Estos métodos y metodologías abordan todas las áreas de la ingeniería de software tradicional, incluida la gestión de proyectos, el diseño y la arquitectura de software y la mejora de procesos. Cada uno de esos métodos y metodologías consiste en prácticas optimizadas y optimizadas para que sean más fáciles de adoptar (**Stellman, 2014**).

Este marco o forma de trabajo permite organizar el proceso de desarrollo de software a través de la definición de pautas a seguir y restricciones a cumplir. En consecuencia, existe una amplia variedad de metodologías definidas, cada una con ventajas y limitaciones inherentes al contexto de aplicación (**Rinaudo, 2015**).

Las metodologías ágiles tienen diferentes fases y procesos a diferencia de los metodologías tradicionales, porque las metodologías ágiles permiten no solo ahorrar tiempo, sino que además sus etapas son más fáciles y entendibles permitiendo que el desarrollo de un software sea más rápido no solo en relevamiento sino en la implementación. Los métodos de gestión de proyectos tradicionales (Cascada y Ciclo en V) y sus límites han llevado al nacimiento de los enfoques ágiles (**Bahit, 2012**).

8.2. Metodologías, técnicas y herramientas

Una metodología de desarrollo de software se fundamenta sobre tres pilares básicos: que hay que hacer y en qué orden, como deben realizarse las tareas y con que pueden llevarse a cabo.

Estas técnicas se basan en la diagramación, definición y especificación. La diagramación es el componente gráfico bajo el cual, de manera visual, se identifican los elementos del análisis o diseño. La definición es el componente descriptivo del elemento, llegando a su estructuración y descomposición unitaria, y la especificación es la codificación textual o formal del elemento. Por ejemplo, la técnica del Diagrama de flujo de Datos, utilizada para el análisis y diseño de alto nivel del sistema utiliza tres componentes (**Barranco, 2015**).

8.2.1. Principales características de las metodologías ágiles:

Objetivo: satisfacer al cliente entregando, rápidamente y a menudo, sistemas que tenga valor.

Proyectos Iterativos e incrementales.

- Los requisitos no se especifican en detalles en el inicio del proyecto. La solución evolucionara a partir del feedback del cliente/usuario.
- Reconocer si las estimaciones a largo plazo son groseras e imprecisas. Estas evolucionan a lo largo del proyecto.
- El esfuerzo de planificación se concentra en el futuro más próximo, es decir en el corto plazo.
- El equipo de desarrollo se auto-organiza y participa en la planificación del proyecto.
- Poca formalidad, es decir, los roles en la jerarquía se simplifica.
- El cliente participa activamente del proceso.
- Trabajan con líderes en vez de con jefes.

Estas características nos traen unos beneficios que principalmente son los siguientes:

Para el cliente, que al guía a los desarrolladores obtienen un producto que atienden sus necesidades. También sirve para percibir con antelación son los dos siguientes:

Para el cliente, que al guiar los desarrolladores obtiene un producto que atiende sus necesidades. También sirve para percibir con antelación si el proyecto es inviable evitando el gasto innecesario de recursos.

Para los desarrolladores, los cuales participan de forma activa en la planificación del proyecto, generando plazos realistas factibles, evitando la sobrecarga de trabajo innecesario que surge cuando se intentan realizar metas que se consideran imposibles (**Lainez, 2015**).

Cuando aplicar una metodología ágil

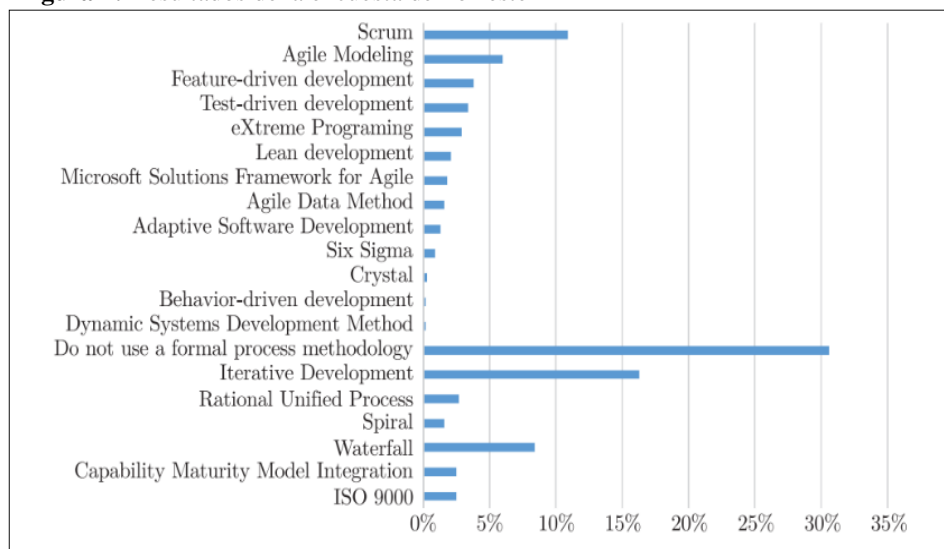
En si no se debería concentrarse en una sola metodología ágil, sino realizar diferente investigación, y analizar el proyecto a desarrollar para de esta manera obtener y usar la más adecuada. Para esto debemos saber adaptar de la manera más adecuada, ya que estas no solo nos sirven para optimizar tiempo sino que son para ayudarnos.

8.2.2. Metodologías ágiles más importantes

Podemos clasificar a aquellos en una escala de más pre-adaptativa a más adaptativa. Alguna programación Xtreme (XP) es un proceso muy estricto que aplica muchas reglas y regulaciones sobre cómo deben hacerse las cosas, y Kanban es algo con la cantidad mínima de conjuntos y regulaciones. Scrum está en el medio y puede cambiar para ser menos o más bases prescriptivas de la necesidad.

Existen varios métodos existentes considerados como ágil, es decir: La programación extrema, Desarrollo Scrum, Crystal familia de metodologías, Feature Driven, el Racional Unifica proceso, método de desarrollo de sistemas dinámicos, adaptativa de desarrollo de software, Agile Modeling.

Figura 1: Resultados de la encuesta de Forrester



Fuente. (Zima, 2015)

De acuerdo con el Forrester informe, Scrum es la opción más común del método de desarrollo de todas las técnicas ágiles - 10,9% de los desarrolladores han declarado que Scrum refleja más estrechamente los proceso de desarrollo que están utilizando actualmente. Scrum es un marco de proceso iterativo e incremental para el desarrollo de software (Zima, 2015).

8.2.2.1. Crystal Methodologies

No se trata de una única metodología sino de un conjunto de ellas centradas en las personas que tienen que desarrollar el software, el equipo es la base de estas metodologías creadas por Alistair Cockburn (**Fernandez, 2010**).

8.2.2.2. Kanban

Kanban fue inventado por Toyota en la década de 1940 para reducir el tiempo de inactividad durante la fabricación. En el mundo del software, básicamente ejecuta el trabajo tal como viene y usa un tablero con notas para rastrear los grados y los cuellos de botella.

Kanban tiene que ver con la visualización del proceso, por lo que la expresión "una imagen vale más que mil palabras" realmente describe el proceso kanban con mucha precisión (**Bibik, 2018**)

8.2.2.3. Programación Xtreme

XP es la más prescriptiva (específica) de las metodologías ágiles, cuyo enfoque principal es la prescripción de procesos de ingeniería. XP prescribe muchas prácticas básicas, que incluyen desarrollo basado en pruebas, pruebas de clientes, integración continua, iteraciones cortas, pequeños lanzamientos, programación de pares, juego de planificación, diseño simple, refactorización, propiedad de código colectivo, estándares de codificación, metáfora, ritmo sostenible, trabajo del cliente estrechamente en el sitio con el equipo, y la prueba de aceptación del cliente en cada lanzamiento.

La principal diferencia con Scrum: encuentro que XP está más impulsado por las metodologías de desarrollo de código que por la dinámica y los procesos del equipo dentro del equipo. XP tiene menos marcos flexibles que Scrum. Scrum puede adoptar prácticas de XP y básicamente hacer lo mismo que XP, pero cuando es necesario, Scrum se centra más en la productividad, mientras que XP se centra en la ingeniería.

Si desea que su equipo use una metodología XP eventualmente, Scrum puede ser un paso intermedio; el equipo puede comenzar con scrum y adoptar gradualmente las mejores prácticas de XP.

8.2.2.3.1. Roles

Rastreador: él o ella se asegura de que no haya problemas con la forma en que van las cosas, escuchan y toman medidas si es necesario. Las acciones incluyen pedirle ayuda a otro programador o asesoría. Si es necesario, él / ella organizaría una reunión con el cliente si eso mantuviera el proceso en marcha.

El cliente tiene el poder de decidir preguntas sobre las historias.

El cliente escribe las historias de usuario y declara las pruebas funcionales.

Programador (programador extremo): define las tareas de ingeniería a partir de las historias, evalúa las historias, realiza una estimación de cuánto tiempo durarán las historias y la tarea, implementa la prueba de la unidad y las historias.

Probador-él prueba los resultados del gráfico, implementa y ejecuta Pruebas Funcionales y se asegura de que la gente sepa cuándo caen los resultados de la prueba.

El entrenador programa la reunión, por ejemplo, el cronograma de compromiso y el plan de iteración, asegura que se siga el proceso de la reunión, toma nota de los resultados de la reunión para futuros informes y se lo entrega al rastreador. El entrenador no le dice a la demanda ni a las personas qué hacer y cuándo se debe hacer.

La codificación también puede ayudar a comunicar pensamientos sobre problemas de programación. Un programador puede usar su codificación para explicar una solución difícil a otros programadores.

Prueba: esto es importante porque si no se prueba no se puede estar seguro de que funciona. En la programación extrema, uno no puede estar seguro a menos que se pruebe adecuadamente antes de enviarlo al cliente.

Los programadores que escuchan no entienden el lado comercial del sistema que se está desarrollando. Sin embargo, el lado comercial determina la función de los sistemas y, por lo tanto, el programa necesita escuchar para saber cuál sería la funcionalidad de los sistemas.

Diseño: crear una estructura de diseño que organice la lógica en los sistemas ayudará a aclarar las dependencias dentro de los sistemas (**Gopaul, 2017**).

8.2.2.3.2. Prácticas de Extreme Programming

La aplicación de XP se orienta alrededor de trece prácticas, que se describen a continuación:

Entregas Frecuentes

Respetar este ritmo permite al cliente y al equipo de desarrollo realizar una introspección del producto.

Ritmo duradero

Para poder entregar con una frecuencia constante, es conveniente respetar un ritmo de trabajo constante y duradero.

Presencia del cliente

El cliente tiene la responsabilidad de definir las funcionalidades que se deben desarrollar, en función de su presupuesto y del tiempo asignado al proyecto. Por tanto, es necesario que sea miembro a tiempo completo del equipo de desarrollo, con el objetivo de responder a las diversas cuestiones que se le pueden formular y tener una visión continua de la calidad de los desarrollos, así como el avance del proyecto en sí mismo.

Diseño sencillo

La regla en términos sencillos es simple: “Desarrollar de la manera más sencilla posible y responder correctamente la necesidad”. La idea es no entrar en conflicto con los problemas relacionados con la complejidad de la solución a implementar, siempre que haya una manera sencilla de responder a la necesidad del cliente.

Implementación de las reglas de codificación

Las reglas de codificación permiten obtener una homogeneidad en el código de la aplicación, facilitando la reutilización del código o su modificación. Estas reglas se deben establecer por el conjunto de miembros del equipo de desarrollo.

Uso de Pruebas unitarias

Extreme Programming trabaja para la aplicación de los desarrollos gestionado por las pruebas (TDD, Test Driven Development). Es decir, antes de la escritura de cualquier línea de código, el desarrollador debe escribir dos casos de pruebas unitarias, uno comprobando el correcto funcionamiento del código desarrollado y el segundo generando un error. Esto permite asegurar la no regresión técnica de la aplicación durante las diferentes adiciones funcionales.

Prueba de aceptación

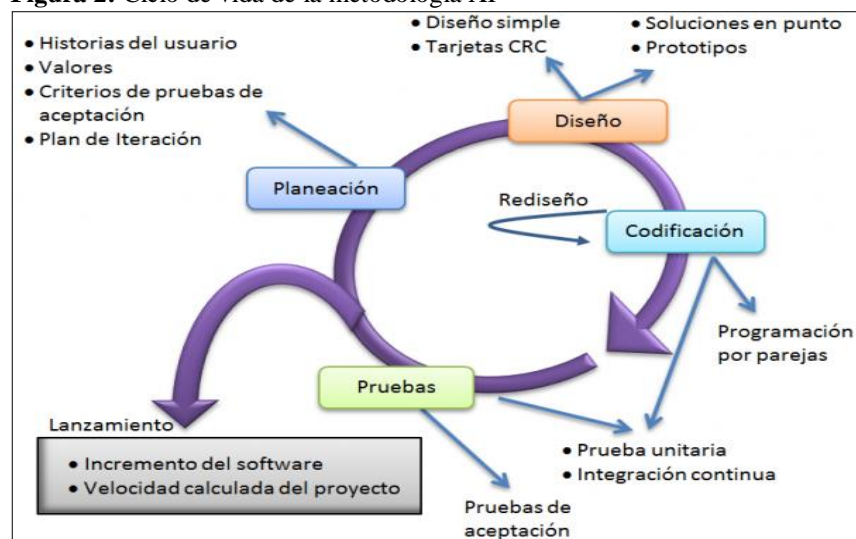
Las personas responsables de homologar software realizan de manera habitual estas pruebas, que tienen como objetivo comprobar que la aplicación está conforme a las necesidades funcionales del cliente. Si es el caso, entonces podemos considerar que la iteración ha terminado. En caso contrario, se deberán añadir correcciones. Estas pruebas se deben pasar por completo antes de cada entrega, para asegurar la no regresión funcional.

8.2.2.3.3. Usos de metáforas y analogías

Para que todos los interlocutores puedan entender y hablar, ya sean técnico o funcionales, es indispensable describir el sistema usado analogías y metáforas.

8.2.2.3.4. Ciclo de vida XP

Figura 2: Ciclo de vida de la metodología XP



Fuente. (Chicago, 2016)

Para terminar con el método XP, vamos a presentar las etapas de su ciclo estándar, es decir, el proceso XP tal y como debe implementarse.

1. El cliente escribe sus necesidades en forma de escenarios.
2. Los desarrolladores evalúan el coste de cada escenario, en colaboración con el cliente.
3. El cliente elige los escenarios que se deben integrar en la siguiente entrega.
4. Cada desarrollador se hace responsable de una tarea para realizar un escenario.
5. Se escribe las pruebas unitarias correspondientes al escenario que se va a implementar.
6. Se realiza la implementación, reorganización del código existente. (Subra, 2018)

8.2.2.3.5. Ventajas de XP

- **Reducción de costos:** la programación extrema está centrada en el código y disminuye algunas actividades para disminuir los cargos y los Desarrolladores se enfocan en la codificación como un reemplazo para reuniones y papeleo innecesarios.
- **Robustez:** XP logra crear un software más rápido con menos defectos al brindar simplicidad. En software, la simplicidad contribuye a la robustez y aporta calidad.
- **Resiliencia:** los requisitos siempre cambiantes hacen que el desarrollo de software sea más difícil, sin embargo, con la recomendación de CP de usar historias de usuarios de la estrella y obtener retroalimentación durante el curso de las iteraciones.

8.2.2.3.6. Contras de XP

- **Planificación detallada:** XP necesita un desarrollo completo desde el principio debido a la alteración de los costos y el alcance.
- **Medición débil:** La programación extrema no mide ni planifica el control de calidad de la codificación.
- **Problemas de duplicación:** XP se desarrolla con programación de pares que generalmente conduce a demasiada replicación de códigos y datos.
- **Dureza:** intente persuadir a los desarrolladores para que acepten esta práctica dura y no es una tarea fácil. Necesita más disciplina en el equipo y la dedicación de sus clientes.

- **Centrado en el código:** en proyectos más grandes, el centrado en el código puede ser agotador en lugar de centrado en el diseño, y XP está centrado en el código (**Moran, 2015**).

8.2.2.4. Scrum

Realmente es una metodología que nos permite desarrollar nuestro software en un corto periodo, de manera segura, a la vez que optimizamos muchos recursos. Ya que no solo nos permite disminuir tiempo sino que también al utilizar esta metodología de desarrollo ágil evitamos diferentes errores que suelen cometerse al desarrollar nuestro software, y de esta manera tener nuestro producto listo para entregar o presentar, ya que nos brinda un proyecto o trabajo sostenible.

Scrum proporciona un marco de gestión de proyectos que enfoca el desarrollo en ciclos de Sprint de 30 días en los que se entrega un conjunto específico de características de Backlog. La práctica central en Scrum es el uso de reuniones de equipo diarias de 15 minutos para la coordinación e integración. Scrum es manejado durante casi diez años y ha sido utilizado para entregar con éxito una amplia gama de productos (**Highsmith, 2014**).

Comparte muchas características con XP, como la participación activa del cliente, pero tiene su principal foco en las actividades de gestión del proyecto. El objetivo principal es transformar un conjunto de ítems requeridos por el cliente en un incremento de funcionalidad 100% operativa para el software.

8.2.2.4.1. Marco de trabajo de Scrum

En un sentido amplio, el marco de trabajo de Scrum se componen de una serie de reglas, que definen roles que integran los equipos, artefactos necesarios para los procesos, bloques de tiempo preestablecidos y ceremonias que deben respetarse.

Scrum es una metodología que nace ajena al desarrollo de software, de hecho sus principios fundamentales fueron desarrollados en procesos de reingeniería por Goldratt, Takeuchi y Nonaka en la década de 1980.

8.2.2.4.2. Roles

Los equipos de Scrum definen tres roles bien diferenciado.

- El Scrum Master. Responsable de asegurar los procesos.
- El Dueño del Producto, responsable de maximizar el valor del producto.
- El Equipo, responsable de realizar el trabajo.

8.2.2.4.3. Artefactos

Cómo herramientas aplicables a todos los procesos anteriores, Scrum emplea cuatro artefactos:

1. Backlog de Producto (una lista priorizada de todo lo que requiere el software)
2. Backlog de Sprint (una lista de tareas necesarias para convertir parte del Backlog de Producto, en incremento de funcionalidad del Software).
3. Scrum taskboard (un tablero físico que permite la transparencia de los ítems de trabajos pendientes, en curso y terminados de un mismo Sprint).
4. Diagrama de Burndow (un gráfico que permite medir visualmente, el progreso de los ítems del Blacklog de producto) (**Bahit, 2012**).

8.2.2.4.4. Pilares de Scrum

Transparencia

La transparencia en Scrum significa que toda persona implicada en el proyecto debe entender fácil y rápidamente el estado de proyecto. Este punto es de una importancia capital. Por tanto conviene forma a los futuros actores de un proyecto Scrum.

Inspección

Con frecuencia, el equipo Scrum debe inspeccionarlo que produce y el estado de avance respecto a sus objetivos, sin que esto entre en contradicción con la productividad del equipo.

Adaptación

Si durante la fase anterior hay desviaciones (Calidad del entregable, retrasos, etc) se debe realizar un ajuste para minimizar sus impactos, los elementos de Scrum que se describen más adelante son ideales para visualizar rápidamente cualquier aspecto positivo o negativo del avance por tanto no hay que descuidarlo.

Los eventos

En el método Scrum se añaden un determinado número de eventos específicos llamados ceremonias. Un proyecto Scrum es vivir un entorno donde tiene una duración limitada (time box). Esta duración se debe respetar escrupulosamente lo que permite una mejor gestión del tiempo y una ganancia en productividad (Subra, 2018).

Los eventos previstos por Scrum son cinco

- Sprint
- Reunión de planificación del Sprint
- Revisión del Sprint
- Retrospectiva del Sprint

Sprint

Un sprint corresponde con un periodo de tiempo, de una duración de un mes en el se realiza un incremento del producto en estado terminado, usable y potencialmente entregable al entorno de producción.

Reunión de planificación del Sprint

El contenido de un Sprint se prepara durante la reunión de planificación que no debe exceder de ocho horas para un Sprint de un mes o cuatro para Sprints de dos semanas.

Revisión del Sprint

Con una duración máxima de 4 horas para un sprint de un mes (2 horas para un Sprint de dos semanas). El objetivo de la demostración es presentar el trabajo realizado por el equipo de

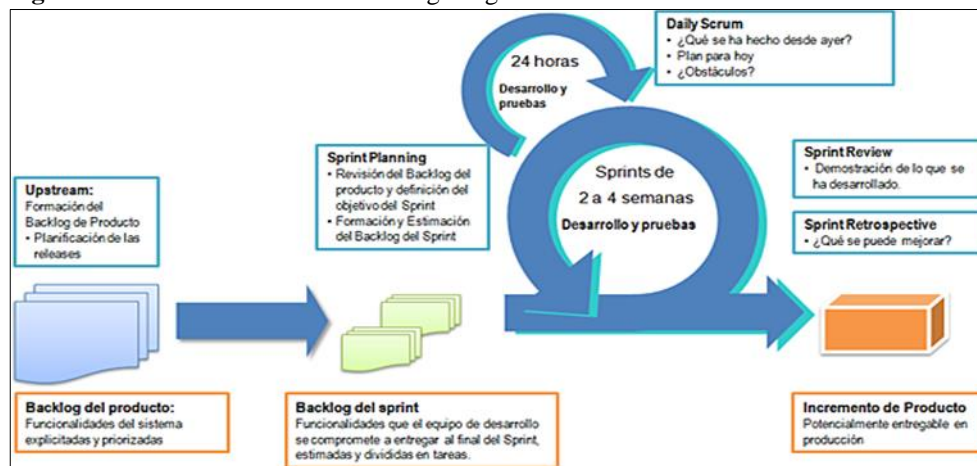
desarrollo y, por consiguiente, hacer un punto de sincronización preciso respecto al avance del proyecto.

Retrospectiva del Sprint

Para terminar la retrospectiva del sprint permite analizar cómo se ha desarrollado

8.2.2.4.5. Ciclo de vida de Scrum

Figura 3: Ciclo de vida de la metodología Ágil Scrum



Fuente: (Subra, 2018)

El ciclo de vida es el siguiente:

1. El Product Owner redacta las User Stories y las sitúa en el Product Backlog.
2. A continuación, el Product Owner prioriza estas User Stories y ordena el Product Backlog en consecuencia.
3. El equipo Scrum se junta en la reunión de planificación del Sprint, con el objetivo de establecer la lista de las User Stories que se tratarán durante el Sprint. Esto forma el Sprint Backlog y a continuación se descomponen en tareas por el equipo de desarrollo.
4. Entonces el Sprint puede comenzar con una iteración de 2, 3 o 4 semanas.
5. El equipo se reúne diariamente para realizar la Melé diaria.
6. Como consecuencia del Sprint, obtenemos un producto potencialmente entregable que forma parte de una demostración durante la revisión del Sprint.

7. El ciclo termina con la retrospectiva del Sprint.

8.2.2.4.6. Ventajas de Scrum

Las principales ventajas de utilizar Scrum son las siguientes:

Simple. SCRUM es un marco metodológico cuyos conceptos se pueden asimilar en poco tiempo.

Flexibles y operativa. Facilita la introducción de cambios en cada sprint pero respeta el trabajo del equipo de desarrollo a lo largo del sprint.

Minimiza riesgos. Propone ciclos de desarrollo muy cortos que minimizan los riesgos.

Se obtienen resultados en el corto plazo. SCRUM está enfocada a suministrar historias de usuario funcionando al finalizar cada sprint.

Inversión optimizada. Al ordenarse la pila del producto por orden de prioridad se asegura la optimización de la inversión.

8.2.2.4.7. Desventajas de Scrum

- Difícil de aplicar a grandes proyectos.
- Difícil de aplicar a proyectos de coste cerrado (**Ruiz, 2017**).

8.2.2.4.8. Costo, plazo y perímetro

En una gestión de proyecto clásica, solo se fija el perímetro. Los costes y los plazos son variables. Estos aspectos desconocidos influyen en el coste global del proyecto, porque no hay nada que indique los costes y los plazos se respetaran.

En Scrum, ocurre totalmente lo opuesto. Los costes y los plazos se respetaran pero el perímetro puede variar. Aquí se trata de una noción algunas veces difícil de hacer entender por parte de la gestión (así como el cliente). Los costes y plazos se respetaran, de modo que la duración de un Sprint siempre es constante y el número de personas que trabajan durante un Sprint es conocido (**Vannieuwenhuyse, 2018**).

8.3. Similitudes y diferencias entre la metodología Scrum y XP

Tabla 3: Similitudes y diferencias entre la metodología Scrum y Programación Extrema

| SCRUM | PROGRAMACION EXTREMA(XP) |
|---|--|
| <p>*Scrum es y, en una práctica ingeniosa, hay archivos que se ocupan de las pruebas automatizadas, el diseño, el desarrollo de la prueba de manejo, las innecesarias programaciones de pares y estos puntos pueden tratarse con campos de ingeniería.</p> <p>*Es una práctica de gestión en lugar de auto organizarse, por lo tanto, la discusión ocurrirá entre la gerencia, no entre los miembros del equipo. *La metodología Scrum lleva más tiempo que la Programación Extrema, Scrum puede tomar desde dos semanas hasta un mes hasta la finalización del proceso.</p> <p>*La metodología Scrum ahorra tiempo y dinero para el negocio. A medida que lo hacemos, Scrum codifica, prueba y también repara rápidamente el movimiento rápido y progresivo.</p> <p>*Scrum simplemente hace frente a los cambios en el proyecto; Además de eso, Scrum depende completamente de los cambios en el proyecto.</p> | <p>*La metodología de programación extrema toma aproximadamente dos semanas como máximo para el desarrollo de software.</p> <p>*La programación extrema, los desarrolladores no tienen que documentar y asistir a reuniones. Con esta metodología, si ofrece. Una gama más amplia de opciones para aprender es una nueva habilidad.</p> <p>*La programación extrema desarrolla el producto para los clientes de manera rápida y rápida con muy pocas imperfecciones.</p> <p>*Siempre que se produzcan cambios, XP permite que suceda con bajo costo.</p> <p>*La programación extrema produce satélites confiables para que la agenda pueda coordinarse sin esfuerzo.</p> |

Fuente: (Vannieuwenhuyse, 2018).

8.4. Evaluación de metodologías

La metodología de Programación Extrema (XP) cumple con menos objetivos establecidos por el proyecto y completa una cantidad menor de historias de desarrollo. La programación extrema también toma más tiempo en el período de desarrollo por historia. Aunque XP tuvo

menos errores en las pruebas, el ciclo de desarrollo también produce menos funciones de accesibilidad.

La metodología Scrum sin embargo cumple con más objetivos establecidos por el proyecto y completa una mayor cantidad de historias de desarrollo en el tiempo asignado.

Por lo tanto, según la evaluación estadística, se justificaría decir que Scrum es una metodología mejor que XP para el desarrollo de la solución de software de atención debido a las características de accesibilidad y cumplimiento de una sola historia en un corto tiempo de desarrollo que Extreme Programming (**Gopaul, 2017**).

8.5. Sistema

Este término es universalmente usado. Hablamos sobre sistemas informáticos, sistemas operativos, sistemas de pago, el sistema educacional, el sistema de gobierno, etcétera. Estos son obviamente usos bastante diferentes de la palabra sistema aunque coinciden en que, de algún modo, el sistema es más que simplemente la suma de sus partes (**Sommerville, 2015**).

El sistema es aquel que abarca tanto datos de entrada como de salida procesándose de manera inteligente al mandar ejecutar acciones las cuales se encuentre en el alcance del mismo. De esta manera logrando llevar y cumplir a cabo las necesidades del propietario así como las del cliente, que soliciten o requieran información proveniente del sistema con el que cuenten.

8.6. Sistema informático

El sistema informático de la empresa es un subsistema dentro del sistema de información y está formado por todos los recursos necesarios para dar respuesta a un tratamiento automático de la información y aquellos otros que posibiliten la comunicación de la misma. En definitiva, por tecnologías de la información y de las comunicaciones (TIC) (**Pablos, 2011**).

En si el sistema informático es el medio por el cual todo se está optimizando con el pasar del tiempo, existen diferentes entidades tanto públicas como privadas las cuales necesitan mejorar en todo aspecto , por lo que acuden a un medio informático, ahorrando no solo tiempo, sino disminuyendo grandes costos al hacer actividades en físico.

8.7. Multiplataforma

Se denomina multiplataforma a aquellos procesos informáticos que actúan y se ejecuta n interactúan en más de un tipo de plataforma para lograr que actúen en plataformas diferentes han de tener un lenguaje de programación adaptado a cada plataforma (**Durán, 2017**).

8.8. Gestión

Tiene como objetivo calcular y utilizar adecuadamente los recursos necesarios para aplicar correctamente estas medidas de prevención, detección y corrección de incidentes de seguridad. De esta manera obteniendo una mejora continua de gestión y tratamiento de incidentes (**Chicano, 2015**).

La gestión nos permite disminuir costo, optimizar tiempo, y mejorar los procesos informáticos que deseemos llevar a cabo de tal que facilite la forma y manera de realizar diferentes actividades las cuales se llevan físicamente, con el tiempo cada vez el ser humano necesita gestionar diferentes actividades, para lograr un objetivo es por tal motivo que es muy importante no solo dentro de la informática, sino de muchas otras áreas más.

8.9. Germoplasma

Es el suministro de muestras representativas de accesiones de semillas de un banco de germoplasma, en respuesta a solicitudes de los usuarios del germoplasma. El propósito de conservar un germoplasma es mejorar las variedades cultivadas través de Fito mejoramiento y la investigación o restaurar la diversidad perdida en fincas y habitas naturales (**Rao, 2012**).

8.10. Base de datos

Una base de datos es un conjunto de elementos interrelacionados y una serie de programas que permiten a varios usuarios tener acceso a estos archivos ya sea para consultarlos o actualizarlos.

Toda base de datos es indispensable en cualquier sistema ya que si esta no existiera, no se podría guardar o almacenar ningún tipo de información, quedando un programa o software obsoleto, ya que no podría recopilar ningún tipo de dato.

La abstracción se da en tres niveles:

Nivel Físico. Competese a la manera en la que se almacenan los datos en medios de almacenamiento

Nivel Conceptual. En el que se describen los datos que realmente se almacenan y las relaciones que existen entre ellos.

Nivel de visión. Describe solo una parte de la base de datos la cual se puede hacer de diferentes maneras para una misma base de datos (**León, 2010**).

8.11. Software

El software consiste en las instrucciones detalladas que controlan el funcionamiento de un sistema computacional. Hay dos tipos de software de sistemas y de aplicaciones cada tipo tiene funciones diferentes. El software de sistemas es un conjunto de programas más generalizados que administran los recursos de la computadora como el procesador central, enlaces de comunicación y dispositivos periféricos. El termino software de aplicaciones describe los programas que se escriben para los usuarios o son escritos por ellos, con el fin de aplicar a la computadora a una tarea específica (**Amaya, 2010**).

Las funciones del software son:

- Administrar los recursos computacionales de hardware.
- Proporcionar las herramientas para aprovechar dichos recursos.
- Actuar como intermediario entre las organizaciones y la información almacenada.

8.12. Open Source

Uno de los conceptos que se suelen mezclar con el software libre es el de Open Source (código abierto). Podemos considerarlo como un error justificable, ya que en la práctica el software Open Source y el software libre comparten las mismas licencias (**Albusae, 2013**).

Open source no solo es un término que nos permite disminuir redundancia al hablar acerca de un software libre, sino que nos brinda las mismas ventajas y uso de diferentes códigos abiertos, para los usuarios, de tal manera que se puedan modificar, estudiar e incluso reutilizarlo en otro software que se vaya a realizar.

El código abierto es una metodología de desarrollo de software basada en la comunidad, iterativa, incremental y evolutiva que enfatiza la experimentación y la experiencia sobre la planificación y el diseño formal.

El servidor de aplicaciones Zope basado en Python, que inicialmente era un software comercial pero tuvo más éxito como proyecto de código abierto, es un ejemplo de esta tendencia. MySQL, un popular programa de base de datos, se comercializa como fuente y como software comercial simultáneamente (Woods, 2015).

8.12.1. Funciones de Open Source

Open Source tiene una gran característica de escalabilidad y versatilidad, y por lo tanto, tiene una amplia gama de aplicaciones. Su característica única es su capacidad para desplegarse rápidamente, ya que uno puede personalizarlo en muy poco tiempo.

8.12.2. Licencias de hardware de código abierto

Las licencias de hardware de código abierto generalmente son compatibles con los destinatarios y los reconstructores del diseño y la documentación para estudiar, actualizar, redistribuir y finalmente distribuir cualquier modificación adicional. Además, las licencias de hardware abiertas no impiden que nadie divulgue o incluso venda el proyecto y su documentación. Los grupos de licencias de hardware abierto muy comunes son la Licencia pública GNU, Creative Common Licensing, MIT.

8.12.3. Ejemplos de hardware de código abierto

Raspberry Pi es compatible con varias plataformas de sistema operativo de código abierto, como Pidora Linux (versión Pi de Fedora), BSD Linux y RISC OS. Es compatible con varios entornos de programación como Python, C, C ++, JAVA, Arduino y Processing (Altheide, 2011).

8.13. Herramientas open source

Los lenguajes de programación son las herramientas básicas utilizadas por los desarrolladores para crear aplicaciones web. Dentro del desarrollo web la comunidad open source parece mostrar su predilección por el lenguaje PHP.

8.14. Ventajas

- Se enumeran a continuación algunas de las ventajas que proporciona el software open source, destacando que sí, bien las económicas son las más visibles, hay otras ventajas importantes:
- Ahorro de costes, las distribuciones de software open source son generalmente gratuitas o a un coste muy bajo, sus licencias dan al usuario libertad para hacer con las aplicaciones las modificaciones o distribuciones que consideren oportunas sin ningún coste añadido.
- El código de los programas es abierto.
- Mayor calidad y seguridad en los programas.
- Reactiva la competencia en un mercado con tendencias monopolísticas.
- Rapidez de desarrollo, la evolución y lanzamiento de versiones mejoradas en el software open source es muy superior al software cerrado (**Cobo, 2015**).

8.15. Tecnologías open source de programación del lado del cliente

8.15.1. JavaScript

Es un lenguaje interpretado basado en guiones que son integrados directamente en el código HTML. El código es transferido al cliente para que este lo interprete al cargar la página.

8.15.2. Características

- Es un lenguaje interpretado.
- No necesita compilación.
- Multiplataforma
- Lenguaje de alto nivel.
- Admite programación estructurada.
- Basado en objetos

8.16. Tecnologías Open Source de programación del lado servidor

8.16.1.1. PHP

Es un lenguaje interpretado del lado del servidor que surge dentro de la corriente llamada código abierto. Se caracteriza por su potencia, versatilidad, robustez y modularidad.

8.17. Entorno de desarrollo integrado (IDE)

Un entorno de desarrollo integrado (IDE) es una aplicación visual que sirve para la construcción de aplicaciones a partir de componentes **(Lozano, 2010)**.

En el mercado actual existen numerosas soluciones de software creadas para el desarrollo de aplicaciones informáticas de diversa índole todas ellas pensadas para ayudar y agilizar en las labores de codificación y desarrollo de programas:

Entre las soluciones existentes en el mercado cabe destacar las siguientes herramientas:

8.17.1. Eclipse

Es un entorno de software multi-lenguaje de programación que incluye un entorno de desarrollo integrado (IDE). Sobre él se pueden instalar herramientas de desarrollo para cualquier lenguaje, mediante la implementación de pluggins adecuados **(Robledo, 2014)**.

8.17.2. Netbeans

Esta plataforma gratuita y de código abierto no presenta restricciones de uso en su licencia, lo cual permite a los desarrolladores más avanzados poder extender dicha plataforma a sus necesidades. Su estructura interna con forma de frameworks se basa en módulos, los cuales ofrecen “extensibilidad” de la plataforma a innumerables funcionalidades **(Lopez, 2016)**.

8.17.2.1. Ventajas

Permite la programación de aplicaciones web con PHP, aplicaciones Java J2SE, web, EJB, aplicaciones para dispositivos móviles, AJAX, HTML, XML, JS, C y C++ entre otros.

También posee funciones más comunes como el autocompletado, el sistema automático de coloración, identificación, herramientas visuales, asistentes, etc.

9. VALIDACIÓN DE PREGUNTAS CIENTÍFICAS O HIPOTESIS

El análisis de las metodologías ágiles Scrum y XP permitirán obtener resultados a cerca de la metodología ágil adecuada para el desarrollo del sistema multiplataforma desarrollado mediante tecnologías Open Source, acorde a la planificación y tiempo de ejecución del proyecto.

10. METODOLOGÍAS Y DISEÑO EXPERIMENTAL

10.1. Tipo de investigación

10.1.1. Investigación bibliográfica

En la realización del proyecto investigativo se aplicó la investigación bibliográfica que es la parte esencial del proyecto, de soporte teórico de los conceptos e informaciones relevantes a cerca de las investigaciones de proyectos similares, recolectando información bibliográfica de temas desconocidos, ayudando a formar un documento de investigación con sustentación de argumentaciones teóricas coherentes.

Mediante esta investigación se recopiló información muy importante basada en el tema de investigación, mediante la revisión y búsqueda de información en libros, artículos científicos y sitios web durante el desarrollo del mismo con la finalidad de conceptualizar argumentos de diferentes autores puesto que es parte esencial para el cumplimiento de los objetivos de la investigación.

Este tipo de investigación permitió fortalecer el conocimiento teórico de los diferentes temas relacionados con la investigación lo cual será útil para el desarrollo del proyecto.

10.1.2. Investigación descriptiva

A través de esta investigación permite escoger instrumentos para medir adecuadamente el nivel de estudio del proyecto describir sus componentes para el desarrollo y las herramientas de software libre escogidas para el desarrollo del sistema. Un estudio descriptivo genera una apropiada familiarización con el proyecto de estudio ayudando a describir los implementos necesarios para la ejecución del proyecto.

10.2. Métodos de investigación

10.2.1. Método Analítico

A través de este método se realizó un análisis de la problemática y el análisis de las metodologías Scrum y XP en la implementación de sistemas multiplataforma desarrollada mediante tecnologías Open Source.

10.2.2. Método Hipotético Deductivo

La aplicación del método hipotético deductivo permitió conocer la estructura del diseño de una hipótesis, formulando de forma adecuada la hipótesis que sustente lo investigado de modo que se comprueba la importancia de nuestra investigación y el desarrollo de nuestro proyecto, determinando las variables dependientes e independientes al igual que la causa y efecto de la estructura de la hipótesis.

10.3. Metodología Scrum para el desarrollo de software

Para el desarrollo y diseño del software se utilizó la metodología Scrum que es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Scrum es útil para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto.

Además esta metodología nos proporciona la generación rápida de proyectos con entrega mensual (o quincenal) de resultados, teniendo una gestión regular de las expectativas del

cliente y basada en resultados tangibles, obteniendo resultados anticipados, posee la flexibilidad y adaptación respecto a las necesidades del cliente, cambios en el mercado, productividad, calidad y alineamiento entre el cliente y el equipo de desarrollo etc. Esto ayuda a la comprensión de las peticiones del cliente para el desarrollo adecuado del software.

10.4 Técnica de investigación

10.4.1 Entrevista

A través de la entrevista se determina los requerimientos que son necesarios para la realización del sistema, de igual forma permite obtener información acerca del funcionamiento que debe dar como resultado el software a desarrollarse. La entrevista se aplicó al desarrollador del sistema anterior del Banco de Germoplasma, por su mayor entendimiento a cerca del desarrollo y funcionamiento del sistema.

10.4.2 Encuesta

La aplicación de la encuesta permite obtener resultados a cerca de la investigación realizada obteniendo resultados que sustenten un proyecto de investigación. En el presente proyecto se aplicó la encuesta a los estudiantes de la carrera de Ingeniería en Sistemas de Información para determinar la aplicación de las metodologías ágil en el desarrollo de proyectos software.

Los resultados obtenidos de las encuestas se muestran que una pequeña parte de estudiantes no aplican metodologías ágil al desarrollo de sus proyectos, lo cual genera inconvenientes, además existe un porcentaje normal que si aplica metodologías ágil de las cual más se utilizan Scrum seguida por la metodología XP (ver Anexo 2)

10.4.3 Población y muestra

10.4.3.1 Población

La población en si es el conjunto de las diferentes personas las cuales vamos a realizar las encuestas, para el desarrollo de nuestro sistema, mediante diferentes consultas e investigaciones obtuvimos un total de 250 personas beneficiarias directas, las cuales de ellas tomaremos un porcentaje como estudio estadístico.

10.4.3.2 Muestra

Con el fin de lograr y obtener un porcentaje cercano a la realidad y útil para el desarrollo de nuestro proyecto, para la muestra hemos aplicado la siguiente fórmula: Fórmula obtenida de las investigaciones del autor (Aguilar, 2010) cuando se conoce el total de unidades de observación que la integran:

$$n = \frac{NZ^2pq}{d^2(N-1) + Z^2pq}$$

Donde:

n=tamaño de la muestra.

N=tamaño de la población.

Z=Valor de Z. Llamado nivel de confianza.

d= intervalo de confianza, diferencia máxima o error.

p=proporción aproximada a la población de referencia, 5%.

q=proporción de referencia 5% que no presenta el fenómeno en estudio (1-p), la suma de la p y la q siempre debe dar 1.

Equivalencia:

n=?, N=250 Z= 95% = 1.96. d= 0.06 p=0.05 q=0.05

Obtención de muestra:

$$n = \frac{NZ^2pq}{d^2(N-1) + Z^2pq}$$

$$n = \frac{250*(1.96)^2*0.05*0.05}{0.06^2(250-1) + 1.96^2*0.05*0.05}$$

$$n = \frac{250*3.84*0.0025}{0.0036(249) + 3.84*0.0025}$$

$$n = \frac{2.40}{0.89+0.96}$$

$$n = \frac{2.40}{1.85}$$

n=129 El resultado de muestra a la cual se aplica la encuesta es de 129 estudiantes.

11. ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS

Metodologías

Las metodologías ayudan a la elaboración de proyectos llevando el orden de los procesos que se deben cumplir para dar solución a las problemáticas, contiene métodos y técnicas en cada una de sus fases del ciclo de vida de un proyecto software. A su vez existe gran variedad de metodologías en el caso de metodologías para desarrollo de productos software, entre las más utilizadas están las metodologías ágiles Scrum y XP (Programación Xtreme)

11.1 Proceso para el diseño del sistema

Para el diseño y desarrollo del software se debe tener en cuenta puntos importantes que serán el pilar y soporte de un proyecto software para el alcance de los objetivos y propósitos de desarrollo:

- Identificación de los requisitos funcionales y no funcionales del Software.
- Análisis de requerimientos de Software.
- Mostrar una visión formal y llamativa del diseño del software.
- Mostrar las soluciones y detalles del software.
- Describir los detalles y diseño que contendrá un software para su correcto funcionamiento.

El software contendrá una interfaz dinámica, atractiva y fácil de utilizar por parte de los usuarios, esto garantiza que el software sea muy utilizado por que cumple con los niveles de integridad, seguridad y usabilidad por parte de los usuarios, la interfaz cuenta con menús ítem par el rápido y fácil manejo por parte de los usuarios la interfaz gráfica de usuario estará diseñada a través de la plataforma y lenguaje de programación Open Source java NetBeans IDE con conexión a una base de datos MySQL que también es un gestor de base de datos Open ource, estas herramientas de diseño son los que en la actualidad se manejan para plataformas actualizadas de software.

El diseño del software contendrá un inventario organizado y clasificado de las especies de semillas que posee el Banco de Germoplasma, en la cual se podrá ingresar nuevos registros de semillas, además de guardar información. Para el procedimiento de búsquedas de semillas o

del colector se tomara en cuenta el nombre o el código único en la base de datos. Para ello se desarrollará dar privilegios a los usuarios para desarrollar dicha actividad.

11.2 Aplicación de la metodología Ágil Scrum

Definición del equipo

Especificación de los roles para el diseño del software:

- **Scrum Master:** Edwin Manobanda, responsable de llevar control de la aplicación de la metodología y su fase o estructura.
- **Product Owner:** Se determina objetivos el será desempeñada por la Ing. Tatiana Gavilánez encargada del departamento de Germoplasma.
- **Scrum Team:** Desarrolladores responsable Edwin Manobanda, Mariela Millingalli

11.2.1 Proceso de desarrollo de la metodología Scrum

11.2.1.1 Product Backlong

En esta fase se procede a la obtención y recolección de los requisitos e historias de usuarios para la elaboración del software. A partir de los requisitos se estima una planificación para el cumplimiento de cada uno de los Sprint y objetivos.

Requisitos específicos

A continuación se detalla los requisitos que debe cumplir el software

Tabla 4: Requisito Ingresar datos de inventario de semillas

| | | |
|-------------------------|--|--|
| Número de requisito | R1 | |
| Nombre de requisito | Ingresar datos de inventario de semillas | |
| Tipo | <input checked="" type="checkbox"/> Requisito | <input type="checkbox"/> Restricción |
| Fuente del requisito | Entrevista | |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial | <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |

Elaborado por: Los Autores

Tabla 5: Requisito Guardar datos de inventario de semillas

| | | |
|-------------------------|--|--|
| Número de requisito | R2 | |
| Nombre de requisito | Guardar datos de inventario de semillas | |
| Tipo | <input checked="" type="checkbox"/> Requisito | <input type="checkbox"/> Restricción |
| Fuente del requisito | Entrevista | |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial | <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |

Elaborado por: Los Autores

Tabla 6: Requisito Modificar datos de inventario de semillas

| | |
|-------------------------|---|
| Número de requisito | R3 |
| Nombre de requisito | Modificar datos de inventario de semillas |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Entrevista |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |

Elaborado por: Los Autores

Tabla 7: Requisito Editar datos de inventario de semillas

| | |
|-------------------------|---|
| Número de requisito | R4 |
| Nombre de requisito | Editar datos de inventario de semillas |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Entrevista |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |

Elaborado por: Los Autores

Tabla 8: Requisito Autenticar ingreso de usuario

| | |
|-------------------------|---|
| Número de requisito | R5 |
| Nombre de requisito | Autenticar ingreso de usuario |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Entrevista |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |

Elaborado por: Los Autores

Tabla 9: Requisito Almacenar información

| | |
|-------------------------|---|
| Número de requisito | R6 |
| Nombre de requisito | Almacenar información |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Entrevista |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |

Elaborado por: Los Autores

11.2.1.2 Objetivos del Product Backlong

Tabla 10: Product Backlong

| Orden | Descripción | Semana de desarrollo (1-4) | Descripción |
|-------|-------------------------------------|----------------------------|---|
| 1 | Análisis de requisitos del software | 1 | Recopilación de requisitos funcionales. |
| 2 | Creación de base de datos | 2 | Crear base de datos, tablas, vistas y campos necesarios. |
| 3 | Creación del software | 4 | Diseño de interfaz gráfica de usuario |
| 3 | Validación de campos | 1 | Los campos deben estar validados de forma que no permita almacenar datos erróneos |
| 5 | Gestión de seguridad | 1 | Generación de respaldos de la base de datos |
| 6 | Conexión de base de datos | 2 | Inserción de datos correctamente. |

Elaborado por: Los Autores

11.2.1.3 Sprint Planning

Una vez obtenido los requisitos del sistema se procede a la planeación y estimación de tiempo para el cumplimiento de los Sprint que se detalla en el Sprint Bakclong. Se debe tener en cuenta que en el Sprint Bakclong no se permite modificaciones.

11.2.1.4 Sprint Bakclong

A continuación se procede a la determinación de las interacciones del Sprint Bakclong, permitiendo detallar las actividades que se deberán cumplir en cada Sprint durante los encuentro de resultados en cada reunión.

Sprint 1: El primer Sprint se analiza los requisitos del software a desarrollar al igual que el diseño de la interfaz

Sprint 2: El segundo Sprint, se procede a los ítems del software:

- Creación de la base de datos
- Creación del software

Sprint 3: El tercer Sprint, se procede a la aplicación de los siguientes ítems:

- Validación de campos
- Gestión de seguridad
- Conexión de base de datos

Sprint 4: El cuarto Sprint, se prepara entrega pruebas del Sistema

Planeación de las interacciones de los Sprints

A continuación especificamos las fechas de reuniones con todos los equipos que conforman el desarrollo del software, para el cumplimiento de las actividades. La reunión se procedió y termino en 30 minutos especificando las interacciones a cumplir en un lapso de 2 semanas, 10 días laborales cada reunión.

Sprint 1: Inicio del primer Sprint para la realización de actividades ya planeadas desde el 04 de Noviembre hasta el 11 de Noviembre del 2019.

Sprint 2: Inicio del segundo Sprint para la realización de actividades ya planeadas desde el 11 de Noviembre hasta el 21 de Diciembre del 2019.

Sprint 3: Inicio del primer Sprint para la realización de actividades ya planeadas desde el 21 de Diciembre del 2019 hasta el 10 de Enero del 2020.

Sprint 4: Inicio del primer Sprint para la realización de actividades ya planeadas desde el 10 de Enero hasta el 24 de Enero del 2020.

11.2.1.5 Revisión de sprint

Reunión de planificación del Sprint 1. En la primera reunión se recolecto los requerimientos funcionales que debe cumplir el software, a través de las especificaciones y requisitos se procede al diseño del software.

11.2.1.6 Requerimientos funcionales

Tabla 11: Requerimientos funcionales del software

| ID | Requerimientos | Actor | Observación |
|---------|--|-----------------------|--|
| RQF-001 | Ingresar datos de semillas | Sistema-usuario | Permitir la inserción de nuevos datos de semillas |
| RQF-002 | Guardar inventario de semillas | Usuario/sistema | Guardar información del inventario |
| RQF-003 | Editar datos de semillas, localización, colector y colección | Administrador/sistema | Editar información que posee el Banco de Germoplasma |
| RQF-004 | Eliminar detalles de inventario de semillas | Administrador | Eliminar detalles de semillas |
| RQF-005 | Autenticación de usuario | Sistema | Permitir autenticarse para hacer uso del software |
| RQF-006 | Crear usuario | Administrador | Permitir la creación de nuevos usuarios |
| RQF-007 | Eliminar usuario | Administrador | Eliminar usuarios no existentes |
| RQF-008 | Modificar usuario | Administrador | Modificar o realizar autenticación de inventarios de semillas. |

Elaborado por: Los Autores

Requisito funcional 001 el software debe permitir ingresar datos de semillas

- Al ingresar los datos debe ser validado los campos y detalles de semillas de manera que el usuario pueda ingresar datos no erróneos.
- Al realizar el registro dar de manera automática un código de inventario.

Requisito funcional 002 guardar inventario

- Debe guardarse de manera rápida.
- Debe permitir la visualización de existencia de datos o inventario redundante.
- Tenga o emita un mensaje de emergencia desea guardar

Requisito funcional 003 editar los registros de semillas

- Permitir realizar modificaciones de datos etc.
- Los campos deben ser validados.

- Guardar nuevas modificaciones.

Requisito funcional 004 eliminar detalles de registros

- Eliminar datos ya existentes
- Eliminar datos para renovar inventario

Requisito funcional 005 autenticación de usuario

- Debe permitir ingresar nombre y contraseña de usuario para seguridad del sistema
- Si no existe crear nuevo usuario.

Requisito funcional 006 crear usuario

- Permitir la creación de nuevo usuario
- Llenar datos de creación de usuarios

Requisito funcional 007 eliminar usuario

- Eliminar datos de usuarios repetitivos
- Eliminar usuarios no existentes

Requisito funcional 008 modificar usuario

- Corregir detalles mal digitalizadas
- Modificación datos de actualización del usuario.

11.2.1.7 Requisitos no funcionales

Software .Se utiliza las siguientes herramientas:

- Gestor de base de datos MYSQL
- Lenguaje de programación NetBeans IDE

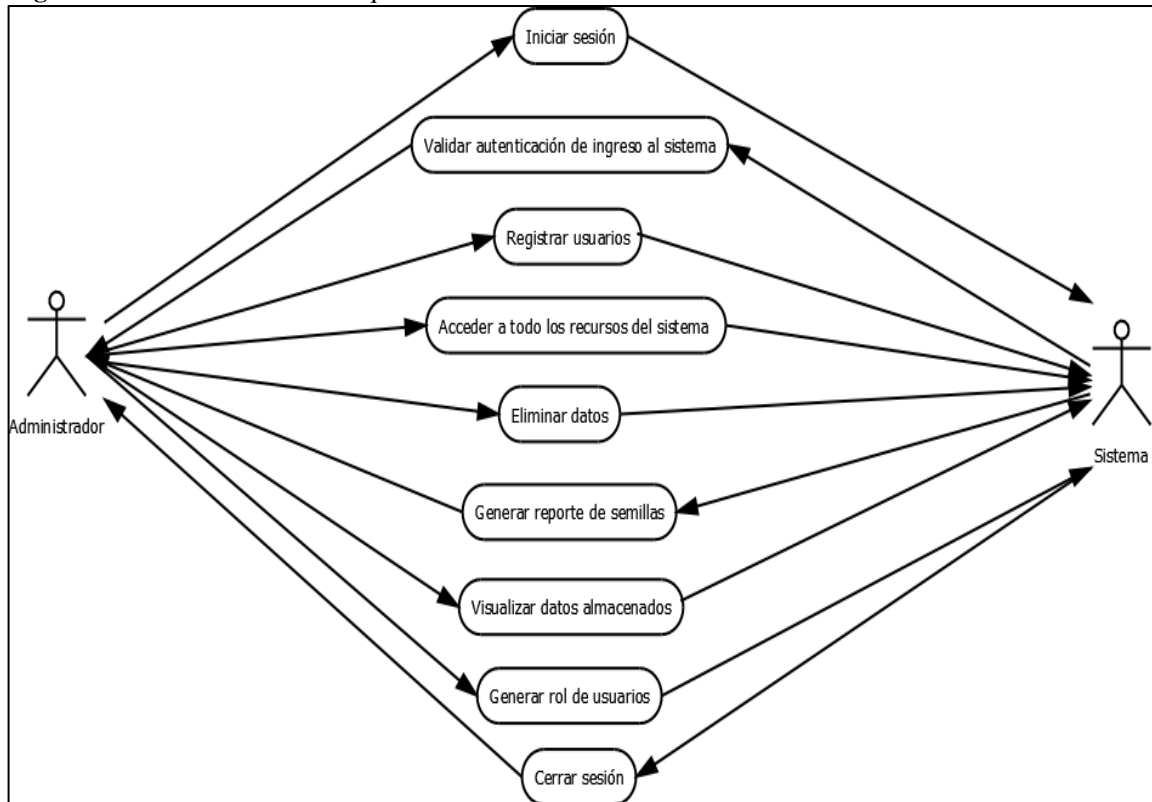
Hardware

Utilización de equipos:

- Computadoras 7 Gb RAM, 1 Tera disco duro, procesador 2,5 GHz

11.2.1.8 Casos de uso

Figura 4: Caso de uso Acciones que realiza el administrador en el sistema



Fuente: Los Autores

Descripción del caso de uso

Tabla 12: Inicio sesión administrador

| Nombre | Iniciar sesión | |
|-------------|---|--|
| Descripción | Permite al administrador iniciar sesión en cualquier momento. | |
| Secuencia | 1 | 1: Sistema: Pide nombre de cuenta y contraseña. 2: Administrador: Ingresar su nombre de cuenta y contraseña. |
| | 2 | 1: Sistema: Verifica nombre y contraseña del administrador. - Si el nombre y contraseña es correcto muestra página siguiente. - Si no regresa al punto 1:1 y muestra mensaje de error de cuenta o contraseña. 2: Administrador: Ingresar al Sistema |
| Excepciones | 1: Nombre y contraseña incorrecta emite mensaje de error. 2: Volver a ingresar una cuenta y contraseña válido. | |

Elaborado por: Los Autores

Tabla 13: Validar autenticación de ingreso al sistema

| | | |
|--------------------|---|--|
| Nombre | Validar autenticación de ingreso al sistema al Administrador | |
| Descripción | El sistema valida al usuario administrador que sea correcto los datos | |
| Secuencia | 1 | 1: Sistema: Pide nombre de cuenta y contraseña. 2: Administrador: Ingresa los datos y contraseña. |
| | 2 | 1: Sistema: Verifica los datos y valida rol administrador. - Si están los datos correctos permite ingresar. - Si no regresa al punto 1:1 y emite mensaje de error. 2: Administrador: Tiene acceso a todas las acciones del sistema. 3: Sistema: Realiza petición de administrador. |
| Excepciones | 1: Datos incorrectos del Administrador, emitir mensaje de error. 2: Emite mensaje ingrese de nuevo | |

Elaborado por: Los Autores

Tabla 14: Registra usuario

| | | |
|--------------------|---|--|
| Nombre | Registra usuario | |
| Descripción | Permite al administrador ingresar datos de usuarios nuevos | |
| Secuencia | 1 | 1: Sistema: Muestra campos a llenar de usuarios. 2: Administrador: Ingresa nuevos datos de usuarios. |
| | 2 | 1: Sistema: Verifica nuevos datos ingresados. - Si edición es correcto muestra interfaz siguiente. - Si no regresa al punto 1:1 y muestra datos a registrar de nuevo. 2: Sistema: Actualiza nueva modificación del administrador. |
| Excepciones | 1: Edición de campos mal editada emite mensaje de error. 2: Mostrar datos errónea si están mal editados. | |

Elaborado por: Los Autores

Tabla 15: Acceder a todo los recursos del sistema

| | | |
|--------------------|--|---|
| Nombre | Acceder a todo los recursos del sistema | |
| Descripción | Permite al administrador acceder a todo los registros almacenados | |
| Secuencia | 1 | 1: Sistema: Pide usuario y contraseña. 2: Administrador: Ingresa datos. |
| | 2 | 1: Sistema: Verifica nuevos datos ingresados. - Si los datos son correctos accede a recursos del sistema. - Si no regresa al punto 1:1 muestra usuario y contraseña incorrecto. 2: Administrador: Visualiza todos los datos almacenados. |
| Excepciones | 1: Usuario y contraseña mal escrita emite mensaje de error. 2: Recordar contraseña. | |

Elaborado por: Los Autores

Tabla 16: Eliminar datos

| | | |
|--------------------|--|---|
| Nombre | Eliminar datos | |
| Descripción | Permite al administrador eliminar y realizar cambios de datos de semillas | |
| Secuencia | 1 | 1: Sistema: Muestra datos de la semilla. 2: Administrador: verifica datos correctos. |
| | 2 | 1: Sistema: Muestra inventarios de los elementos que contiene. 2: Administrador: Actualiza nueva modificación de campos de las semillas 3: Administrador: elimina datos duplicados o usuarios inexistentes. |
| Excepciones | 1: Al eliminar emite mensaje está seguro de realizar acción eliminar. 2: Mostrar datos a restaurar. | |

Elaborado por: Los Autores

Tabla 17: Generar reportes de semilla

| | | |
|--------------------|---|--|
| Nombre | Generar reportes de semilla | |
| Descripción | Permite al administrador generar reporte de semillas | |
| Secuencia | 1 | 1: Sistema: Muestra datos de la semilla. 2: Administrador: Selecciona semilla. |
| | 2 | 1: Sistema: Muestra semilla a realizar reportes. - Si desea imprimir reporte muestra interfaz siguiente. - Si no regresa al punto 1:1 y muestra reporte de semilla 2: Sistema: Imprime reporte. |
| Excepciones | 1: Error de semilla no encontrada. 2: No posee registro de semillas. | |

Elaborado por: Los Autores

Tabla 18: Visualizar datos almacenados

| | | |
|--------------------|--|---|
| Nombre | Visualizar datos almacenados | |
| Descripción | Permite al administrador visualizar información almacenada en el sistema. | |
| Secuencia | 1 | 1: Sistema: Muestra información almacenada. 2: Administrador: Visualiza información. |
| | 2 | 1: Sistema: Da opción de búsqueda. 2: Administrador: Actualiza nueva búsqueda. |
| Excepciones | 1: Datos de búsqueda incorrectos muestra mensaje error. 2: Semilla inexistente. | |

Elaborado por: Los Autores

Tabla 19: Generar rol de usuarios

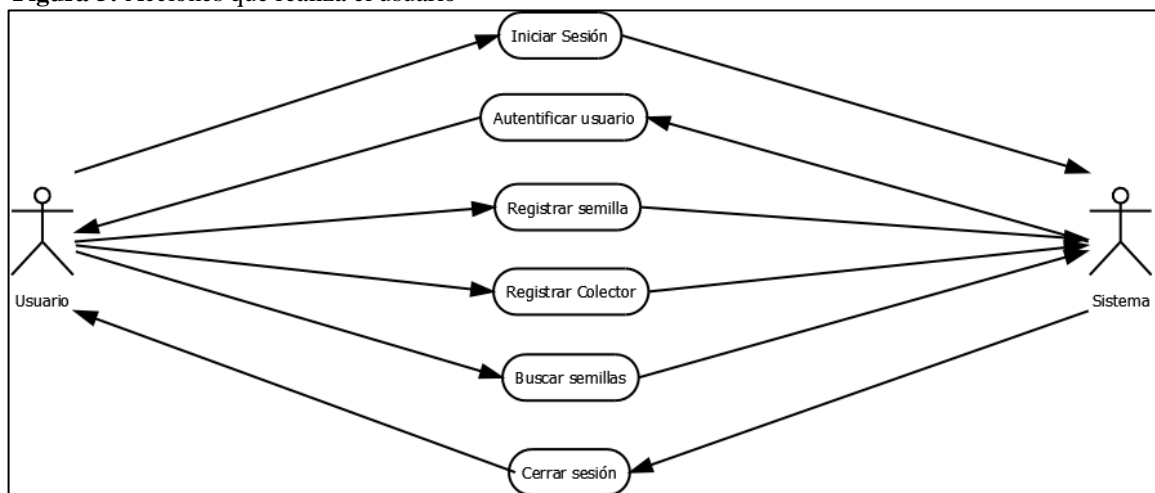
| Nombre | Generar rol de usuarios | |
|-------------|---|---|
| Descripción | Permite al administrador dar roles o permisos a usuarios. | |
| Secuencia | 1 | 1: Sistema: Muestra usuarios existentes. 2: Administrador: Asigna roles. |
| | 2 | 1: Sistema: Verifica nuevos roles ingresados. - Si aplica roles a usuarios muestra siguiente interfaz. - Si no regresa al punto 1:1 2: Sistema: Actualiza nueva aplicación de roles a usuario. |
| Excepciones | 1: Rol al escogido puede editar. | |

Elaborado por: Los Autores

Tabla 20: Cerrar sesión

| Nombre | Cerrar sesión | |
|-------------|--|---|
| Descripción | Permite al administrador cerrar sesión | |
| Secuencia | 1 | 1: Sistema: Muestra botón salir 2: Administrador: Selecciona opción. |
| | 2 | 1: Sistema: Cierra sesión |
| Excepciones | 1: Ingrese de nuevo. | |

Elaborado por: Los Autores

Figura 5: Acciones que realiza el usuario

Elaborado por: Los Autores

Tabla 21: Iniciar sesión

| Nombre | Iniciar sesión | |
|-------------|--|---|
| Descripción | Permite al usuario iniciar sesión | |
| Secuencia | 1 | 1: Sistema: Muestra usuario y contraseña a ingresar. 2: Usuario: Ingresa datos. |
| | 2 | 1: Sistema: Verifica datos. - Si dato es correcto muestra interfaz siguiente. - Si no regresa al punto 1:1 contraseña o usuario incorrecto. 2: Sistema: muestra acceso a la información. |
| Excepciones | 1: Contraseña y usuario mal ingresados emite mensaje de error. 2: Mensaje ingrese usuario valido. | |

Elaborado por: Los Autores

Tabla 22: Autenticar usuario

| | | |
|--------------------|---|--|
| Nombre | Autenticar usuario | |
| Descripción | Sistema autentifica a usuario | |
| Secuencia | 1 | 1: Sistema: Pide usuario y contraseña. 2: Usuario: Ingresa datos. |
| | 2 | 1: Sistema: Verifica datos ingresados. - Si usuario y contraseña es correcto muestra interfaz siguiente. - Si no regresa al punto 1:1 muestra usuario y contraseña erróneos. 2: Sistema: Permite acceso al sistema. |
| Excepciones | 1: Contraseña y usuario mal ingresado emite mensaje de error. 2: Emite ingreso de nuevo. | |

Elaborado por: Los Autores

Tabla 23: Registrar semilla

| | | |
|--------------------|---|--|
| Nombre | Registrar semilla | |
| Descripción | Permite al usuario ingresar datos de semillas | |
| Secuencia | 1 | 1: Sistema: Muestra datos a llenar de la semilla. 2: Usuario: Ingresa nuevos datos. |
| | 2 | 1: Sistema: Verifica nuevos datos ingresados. - Si datos de semilla es correcto muestra interfaz siguiente. - Si no regresa al punto 1:1 datos mal ingresados. 2: Sistema: Actualiza nueva modificación de usuario. |
| Excepciones | 1: Datos incorrectos emite mensaje de error. | |

Elaborado por: Los Autores

Tabla 24: Registrar colector

| | | |
|--------------------|---|--|
| Nombre | Registrar colector | |
| Descripción | Permite al usuario registra datos del colector. | |
| Secuencia | 1 | 1: Sistema: Muestra datos a llenar de colector. 2: Usuario: Ingresa nuevos datos. |
| | 2 | 1: Sistema: Verifica nuevos datos ingresados. - Si es correcto muestra interfaz siguiente. - Si no regresa al punto 1:1 y muestra datos incorrectos. 2: Sistema: Actualiza nuevo datos de colector. |
| Excepciones | 1: datos mal ingresados emite mensaje de error. 2: Duplicado mostrar colector existente. | |

Elaborado por: Los Autores

Tabla 25: Buscar semilla

| Nombre | Buscar semilla | |
|-------------|--|---|
| Descripción | Permite al usuario buscar y ver datos de semillas | |
| Secuencia | 1 | 1: Sistema: Muestra datos de búsqueda de la semilla. 2: Usuario: Ingresar datos de búsqueda. |
| | 2 | 1: Sistema: Verifica datos ingresados. - Si es correcto muestra interfaz siguiente. - Si no regresa al punto 1:1 y muestra ingresar nueva búsqueda. 2: Sistema: Muestra semilla resultado de búsqueda. |
| Excepciones | 1: Ingrese datos correctos de semilla a buscar. 2: Mostrar mensaje semilla no encontrada. | |

Elaborado por: Los Autores

Tabla 26: Cerrar sesión

| Nombre | Cerrar sesión | |
|-------------|---|---|
| Descripción | Sistema permite al usuario cerrar sesión. | |
| Secuencia | 1 | 1: Sistema: Muestra botón cerrar sesión. 2: Usuario: Selecciona salir. |
| | 2 | 1: Sistema: Cierra sesión de usuario. |
| Excepciones | 1: Mensaje desea salir | |

Elaborado por: Los Autores

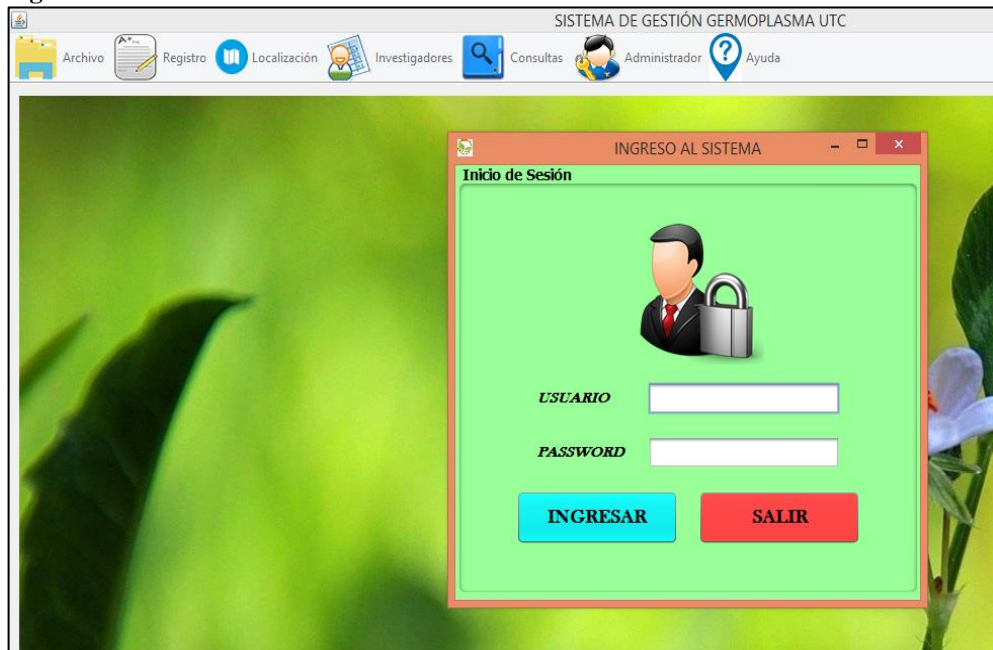
Reunión de planificación del Sprint 2

Diseño del sistema informático multiplataforma de gestión, mediante herramientas Open Source, que facilitan un diseño dinámico agradable al usuario.

Interfaz gráfica del software

A continuación se muestra el diseño del software desarrollado aplicando la metodología Scrum.

Figura 6: Interfaz inicio de sesión



Elaborado por: Los Autores

Figura 7: Interfaz registro de semillas

| Cod Semilla | Nombre | Observaciones | Número de Colec... | Repeticiones | Duplicado |
|-------------|------------------|---------------------|--------------------|--------------|-----------|
| 1 | Frejol Cuarenton | Alimento de cons... | 1 | 2 | 1 |
| 2 | maiz | semilla de regim... | 1 | 1 | 1 |

Elaborado por: Los Autores

Figura 8: Interfaz registro de colección



Registro de Colección de Semilla

Datos de colección de Semillas

Cod Colección: Nombre de Semilla:

Colector: Número de Colección:

Repeticiones: Duplicado:

Altura: Diámetro:

Uso: Nombre Común:

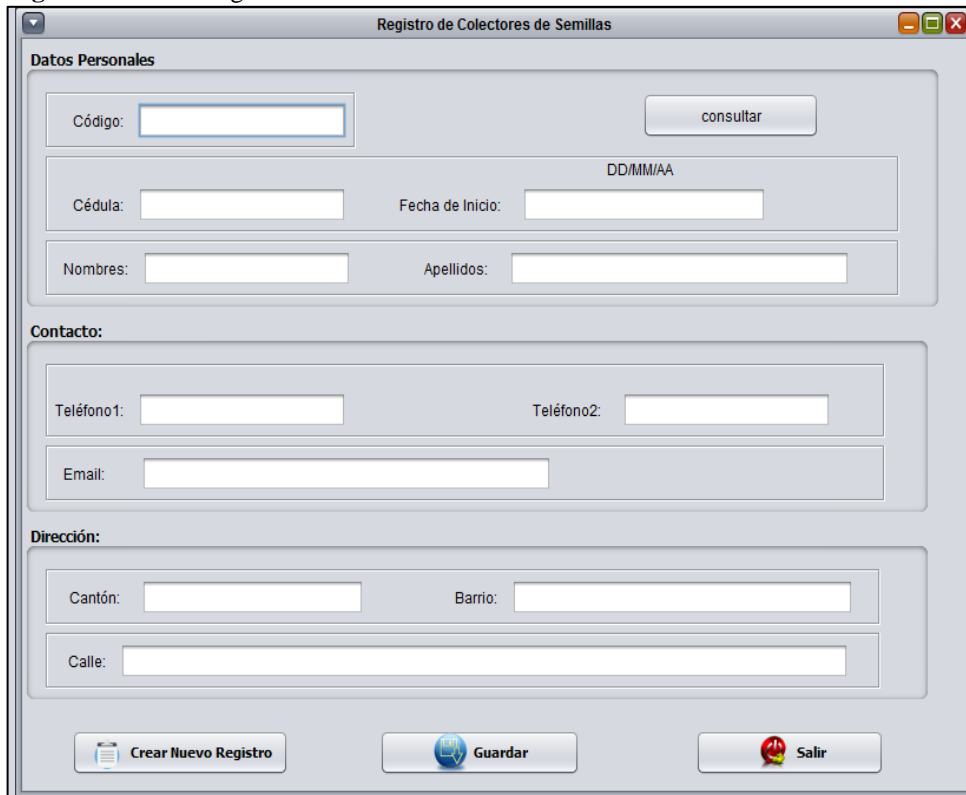
Fecha de Colección: Fecha de cierre de Colección:

Descripción: Observaciones:

Elaborado por: Los Autores

Figura 9: Interfaz registro de colectores de semilla



Registro de Colectores de Semillas

Datos Personales

Código:

Cédula: Fecha de Inicio: DD/MM/AA

Nombres: Apellidos:

Contacto:

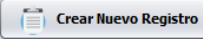


Teléfono 1: Teléfono 2:

Email:

Dirección:

Cantón: Barrio:

Calle:

Elaborado por: Los Autores

Figura 10: Interfaz Registro de Usuarios

Elaborado por: Los Autores

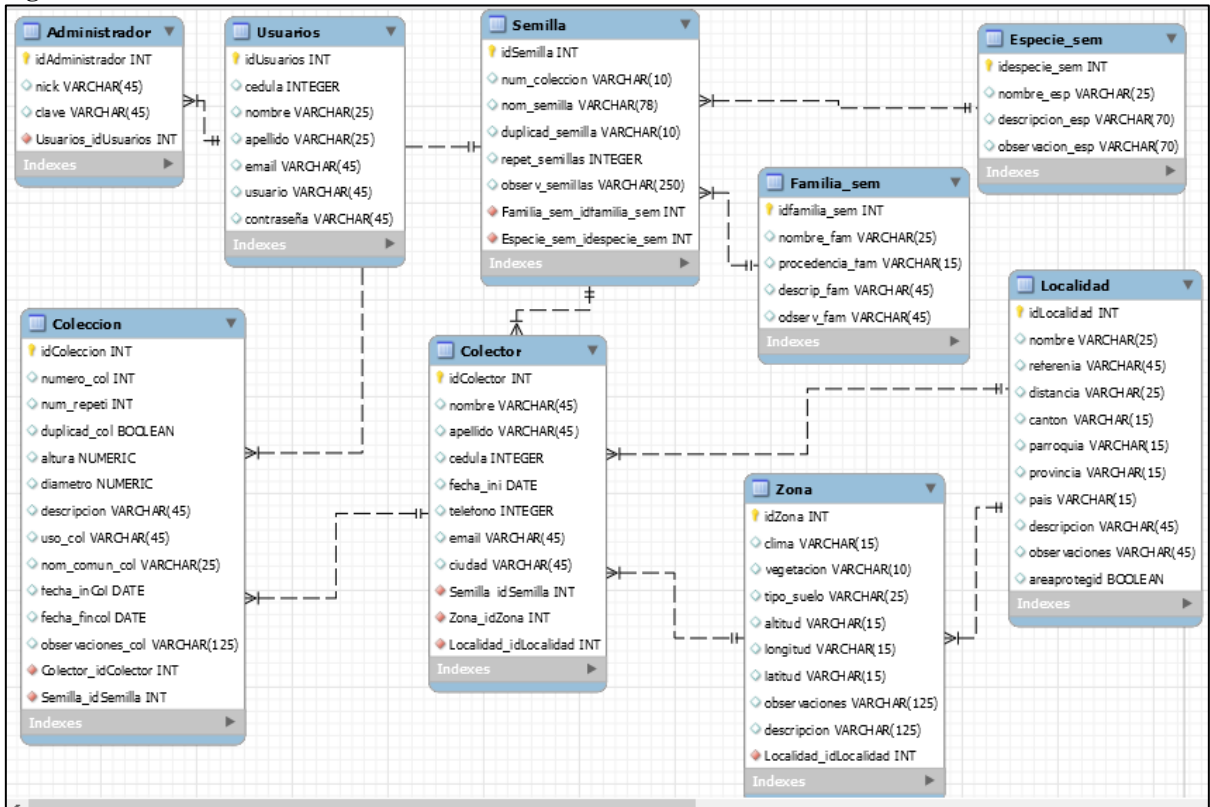
Reunión de planificación del Sprint 3. Verificación de la interfaz muestra de mensajes de advertencia de mal ingreso de datos.

Figura 11: Interfaz validación de los campos

| Cod Semilla | Nombre | Observaciones | Número de Colec... | Repeticiones | Duplicado |
|-------------|------------------|---------------------|--------------------|--------------|-----------|
| 1 | Frejol Cuarenton | Alimento de cons... | 1 | 2 | 1 |
| 2 | maiz | semilla de regim... | 1 | 1 | 1 |
| 3 | Naranja | citrico | 2 | 2 | 1 |

Elaborado por: Los Autores

Figura 12: Modelo relacional de la base de datos



Elaborado por: Los Autores

Reunión de planificación del Sprint 4. Pruebas de funcionamiento generación de reporte

Figura 13: Reporte de semilla

BANCO DE GERMOPLASMA

Fecha de reporte: 15/01/2020
 Colector: Tatiana Gavilánez
 Fecha de inicio colección: 16/06/2015 Fecha final colección: 12/12/2017

REPORTE

| | |
|----------------------|--|
| Cod. Semilla: | 03245 |
| Duplicado: | 02 |
| Nombre: | Frijol canario |
| Numero de colección: | 23 |
| Nombre común: | Yellow bean |
| Nombre científico: | Phaseolus vulgaris |
| Especie: | Leguminosa |
| Localidad: | Pucayacu |
| Provincia: | Cotopaxi |
| Canton: | La Mana |
| Zona: | Zona3 |
| Descripción: | Semilla de consumo humano, producto alimenticio, grano color amarillo, textura suave, contenido alimenticio hierro |

Elaborado por: Los Autores

11.2.1.9 Entrega de Proyecto

Tabla 27: Verificación de entrega de resultados

| Orden | Actividades | Descripción | Incumplimiento | Cumplimiento |
|-------|-------------------------------------|---|----------------|--------------|
| 1 | Análisis de requisitos del software | Recopilación de requisitos funcionales. | | ✓ |
| 2 | Creación de base de datos | Crear base de datos, tablas, vistas y campos necesarios. | | ✓ |
| 3 | Creación del software | Diseño de interfaz gráfica de usuario | | ✓ |
| 3 | Validación de campos | Los campos deben estar validados de forma que no permita almacenar datos erróneos | | ✓ |
| 5 | Gestión de seguridad | Generación de respaldos de la base de datos | | ✓ |
| 6 | Conexión de base de datos | Inserción de datos correctamente. | | ✓ |

Elaborado por: Los Autores

11.3. Aplicación de metodología ágil XP

11.3.1. Descripción del cliente

Actualmente el departamento de germoplasma de la Universidad Técnica de Cotopaxi extensión La Maná, realiza las gestiones de recolección de semillas originarias del Cantón La Maná y sus alrededores, para luego ser almacenados bajo casilleros a temperatura ambiente que no provoque la germinación o pérdida de la semilla. El departamento está conformada por dos personas que administran la gestión de semillas.

Definiciones, acrónimos y abreviaturas

R. Requisitos es una característica que debe cumplir el sistema.

Sist. Sistemas

RF. Requisitos funcionales, funcionalidad que debe contener el sistema para satisfacer los requerimientos de negocio.

RNF. Requisitos no funcionales

Id. Código de identidad

Atributo. Característica de algún objeto. Por ejemplo, el color y la forma de algún objeto.

Arquitectura. Estructura organizacional de un sistema o componente. Visión general de alto nivel para un sistema de software

Desarrollador. Persona encargada de realizar el software a nivel de código fuente

Diseño. Creación de modelos abstractos y planes para implementar los requerimientos del software.

Disponibilidad. El grado en el cual un sistema o componente está operando y accesible cuando requiere ser usado.

Hardware. Equipos físicos usados para procesar, guardar y transmitir programas de computadores o datos.

Interfaz de usuario. La interfaz entre el sistema y sus usuarios externos.

11.3.2. Planeación

11.3.2.1. Historia de usuarios

A través de la historia de usuario se recopila la información acerca de lo que contendrá el software a diseñar, a través de los requerimientos de los usuarios.

Las historias de usuarios del banco de germoplasma son:

- Acceder al sistema
- Gestionar usuarios
- Realizar registros de semillas
- Registros de colector
- Registros de colección y localización
- Creación de reportes

A continuación se muestra la historia de usuarios detallado.

Tabla 28: Historia acceso al sistema

| Historia de usuario | |
|---|---|
| Numero: 1 | Usuario: Administrador, usuario Colector |
| Nombre historia: Acceso al sistema | |
| Prioridad en Negocio: alta | Riesgo de desarrollo: media |
| Puntos estimados: 2 | Interacción asignada: 1 |
| Programador responsable: Edwin Manobanda | |
| Descripción: El sistema contara con un usuario administrador que se encargara de la gestión de cambios y asignación de roles a los demás usuarios. | |
| Observaciones: El administrador es el usuario único que puede acceder a todos los recursos del sistema. | |

Elaborado por: Los Autores

Tabla 29: Historia gestionar usuarios

| Historia de usuario | |
|--|------------------------------------|
| Numero: 2 | Usuario: Administrador |
| Nombre historia: Gestionar usuarios | |
| Prioridad en Negocio: alta | Riesgo de desarrollo: media |
| Puntos estimados: 2 | Interacción asignada: 1 |
| Programador responsable: Mariela Millingalli | |
| Descripción: A través del sistema el administrador insertara nuevos usuarios. | |
| Observaciones: El administrador es el usuario único insertar nuevos usuarios | |

Elaborado por: Los Autores

Tabla 30: Historia registro de semillas

| Historia de usuario | |
|---|---|
| Numero: 3 | Usuario: Administrador, usuario Colector |
| Nombre historia: Registro de semillas | |
| Prioridad en Negocio: alta | Riesgo de desarrollo: media |
| Puntos estimados: 2 | Interacción asignada: 1 |
| Programador responsable: Edwin Manobanda | |
| Descripción: El usuario administrador o colector puede realizar registro de semillas | |
| Observaciones: Determinadas funciones de como eliminar editar están solo disponible para el administrador. | |

Elaborado por: Los Autores

Tabla 31: Historia registro de colector

| Historia de usuario | |
|---|------------------------------------|
| Numero: 4 | Usuario: Administrador |
| Nombre historia: Registro de colector | |
| Prioridad en Negocio: alta | Riesgo de desarrollo: media |
| Puntos estimados: 2 | Interacción asignada: 1 |
| Programador responsable: Edwin Manobanda | |
| Descripción: Solo el administrador puede registrar nuevos colectores o actualizar datos. | |
| Observaciones: El administrador se encarga de ingresar colectores | |

Elaborado por: Los Autores

Tabla 32: Historia registro de colección y localización

| Historia de usuario | |
|---|---|
| Numero: 5 | Usuario: Administrador, usuario Colector |
| Nombre historia: Registro de colección y localización | |
| Prioridad en Negocio: alta | Riesgo de desarrollo: media |
| Puntos estimados: 2 | Interacción asignada: 1 |
| Programador responsable: Edwin Manobanda | |
| Descripción: El colector como el administrador puede realizar registro de colección y localización de semilla. | |
| Observaciones: El administrador es el usuario único que puede acceder a todos los recursos del sistema. | |

Elaborado por: Los Autores

Tabla 33: Historia acceso al sistema

| Historia de usuario | |
|---|---|
| Numero: 6 | Usuario: Administrador, usuario Colector |
| Nombre historia: Acceso al sistema | |
| Prioridad en Negocio: alta | Riesgo de desarrollo: media |
| Puntos estimados: 2 | Interacción asignada: 1 |
| Programador responsable: Edwin Manobanda | |
| Descripción: El sistema contará con un usuario administrador que se encargará de la gestión de cambios y asignación de roles a los demás usuarios. | |
| Observaciones: El administrador es el usuario único que puede acceder a todos los recursos del sistema. | |

Elaborado por: Los Autores

11.3.2.2. Asignación de roles del proyecto

A continuación se muestra el grupo de colaboradores para el desarrollo del software.

Tabla 34: Asignación de roles del proyecto

| Roles | Asignado A: |
|------------------------------------|------------------------|
| Programador | Edwin Manobanda |
| Ciente | Ing. Tatiana Gaviláñez |
| Encargado de pruebas y seguimiento | Ing. Edel Rodríguez |
| Consultor y gestor | Mariela Millingalli |

Elaborado por: Los Autores

11.3.2.3. Plan de entrega del proyecto

El plan de entrega del proyecto se genera acorde a la historia de usuarios definidas anteriormente, en donde a través de la metodología XP su realización de interacción se lo realiza diariamente.

Tabla 35: Plan de entrega del proyecto

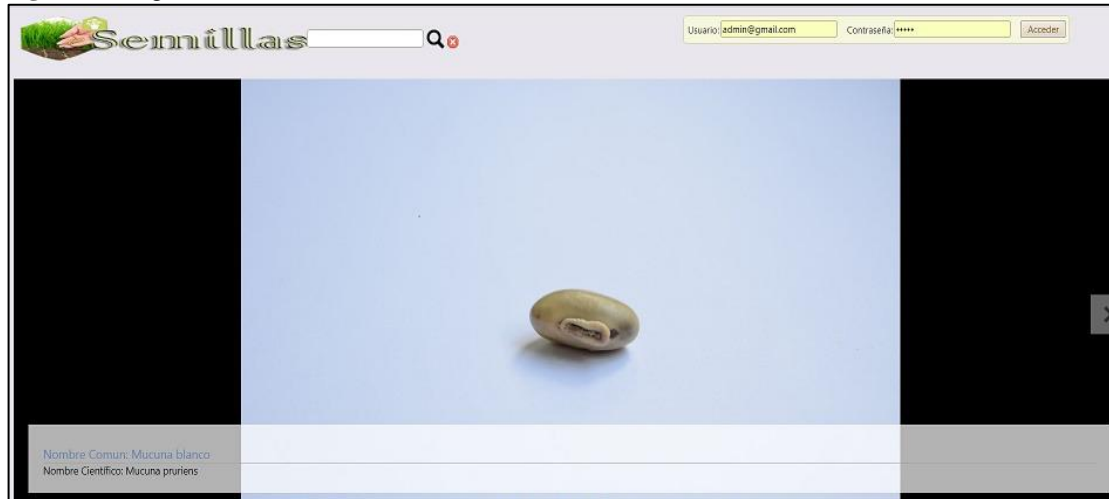
| Historia | Iteración | Prioridad | Fecha inicio | Fecha fin |
|-----------------|------------------|------------------|---------------------|------------------|
| Historia 1 | 1 | Alta | 04/11/2019 | 13/11/2019 |
| Historia 2 | 1 | Alta | 14/11/2019 | 29/11/2019 |
| Historia 3 | 1 | Alta | 02/12/2019 | 14/12/2019 |
| Historia 4 | 1 | Alta | 16/12/2019 | 06/01/2020 |
| Historia 5 | 1 | Alta | 07/01/2020 | 17/01/2020 |
| Historia 6 | 1 | Alta | 20/01/2020 | 07/02/2020 |

Elaborado por: Los Autores

11.3.3. Diseño

Una vez obtenido los requerimientos e historia de usuarios se procede al diseño del sistema. Esta metodología permite simplicidad de diseño del sistema

Figura 14: Ingreso al sistema



Elaborado por: Los Autores

Esta pantalla muestra el ingreso hacia el sistema, donde solo puede acceder con la contraseña y nombre de usuario. A parte puede realizar búsqueda de semillas almacenadas el Banco de Germoplasma.

Figura 15: Roles que puede cambiar el administrador

| Id | Fecha | Usuario | Acción | Tabla | Detalle |
|----|---------------------|-----------------|--------|-------------|----------------|
| 1 | 23/07/2017 10:20:53 | admin@gmail.com | INSERT | Usuarios | |
| 2 | 05/11/2017 14:42:55 | admin@gmail.com | INSERT | Alrededores | alr1 |
| 3 | 05/11/2017 14:56:31 | admin@gmail.com | EDIT | Usuarios | |
| 4 | 13/11/2017 12:11:48 | admin@gmail.com | EDIT | Alrededores | Un Alrededor |
| 5 | 13/11/2017 12:12:02 | admin@gmail.com | INSERT | Alrededores | Otro alrededor |
| 6 | 24/11/2017 15:47:03 | admin@gmail.com | EDIT | Usuarios | |

Elaborado por: Los Autores

Esta pantalla muestra los roles que el administrador puede crear a cada uno de los usuarios para que no puedan acceder a interfaces que solo administrador puede realizar, como

eliminación de usuarios, actualización e ingreso de nuevos colectores y eliminar datos redundantes.

Figura 16: Registro de semillas

The screenshot displays the 'Semillas' web application interface. The header shows the user is logged in as 'Administrador del Sistema'. The left sidebar contains a navigation menu with categories like 'Catalogos', 'Ubicabilidad', 'Elementos', and 'Administrador'. The main content area is titled 'Ingreso Semilla' and contains a form with the following fields:

| | | | |
|--------------------|-----------------------------------|--------------------|-----------------------------------|
| Código: | <input type="text"/> | Latitud: | <input type="text" value="0.00"/> |
| Nombre Común: | <input type="text"/> | Longitud: | <input type="text" value="0.00"/> |
| Nombre Científico: | <input type="text"/> | Altitud: | <input type="text" value="0.00"/> |
| Dimensión: | <input type="text" value="0.00"/> | Eliminado: | <input type="checkbox"/> No |
| Peso: | <input type="text" value="0.00"/> | Localidad: | Seleccione un elemento |
| Cantidad: | <input type="text" value="0.00"/> | Género: | Seleccione un elemento |
| Aislado: | <input type="checkbox"/> No | Instituto: | Seleccione un elemento |
| Parientes: | <input type="checkbox"/> No | Especie: | Seleccione un elemento |
| Herbario: | <input type="checkbox"/> No | Estado: | Seleccione un elemento |
| Muestra: | <input type="text"/> | Estado Fenológico: | Seleccione un elemento |
| Temperatura Clima: | <input type="text" value="0.00"/> | Tipo: | Seleccione un elemento |
| Humedad Clima: | <input type="text" value="0.00"/> | Topografía: | Seleccione un elemento |
| Luz: | <input type="text" value="0.00"/> | Microclima: | Seleccione un elemento |
| Método: | <input type="text" value="0.00"/> | Alrededores: | Seleccione un elemento |

Elaborado por: Los Autores

La aplicación de la metodología XP ase que el diseño de la interfaz gráfica del sistema sea sencilla pero contiene todos los elementos de forma organizada en un solo contorno a diferencia del diseño de del sistema aplicando metodología Scrum.

11.3.4. Codificación

A continuación se muestra las correcciones y estructura de programación y gestión de la base de datos, analizando la estructura de codificación del script de la base de datos del banco de germoplasma para determinar que no existan falencias en relaciones de tablas y almacenamiento de la información.

Figura 17: Script base de datos, localidad

```

-----
CREATE SCHEMA IF NOT EXISTS `BGermoplasma` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci ;
USE `BGermoplasma` ;

-----
-- Table `BGermoplasma`.`Localidad`
-----
CREATE TABLE IF NOT EXISTS `BGermoplasma`.`Localidad` (
  `idLocalidad` INT NOT NULL,
  `nombre` VARCHAR(25) NULL,
  `referencia` VARCHAR(45) NULL,
  `distancia` VARCHAR(25) NULL,
  `canton` VARCHAR(15) NULL,
  `parroquia` VARCHAR(15) NULL,
  `provincia` VARCHAR(15) NULL,
  `pais` VARCHAR(15) NULL,
  `descripcion` VARCHAR(45) NULL,
  `observaciones` VARCHAR(45) NULL,
  `areaprotegid` TINYINT(1) NULL,
  PRIMARY KEY (`idLocalidad`))
ENGINE = InnoDB;

```

Elaborado por: Los Autores

Figura 18: Script de las tablas familia y especie de semillas

```

-----
-- Table `BGermoplasma`.`Familia_sem`
-----
CREATE TABLE IF NOT EXISTS `BGermoplasma`.`Familia_sem` (
  `idfamilia_sem` INT NOT NULL,
  `nombre_fam` VARCHAR(25) NULL,
  `procedencia_fam` VARCHAR(15) NULL,
  `descrip_fam` VARCHAR(45) NULL,
  `odserv_fam` VARCHAR(45) NULL,
  PRIMARY KEY (`idfamilia_sem`))
ENGINE = InnoDB;

-----
-- Table `BGermoplasma`.`Especie_sem`
-----
CREATE TABLE IF NOT EXISTS `BGermoplasma`.`Especie_sem` (
  `idespecie_sem` INT NOT NULL,
  `nombre_esp` VARCHAR(25) NULL,
  `descripcion_esp` VARCHAR(70) NULL,
  `observacion_esp` VARCHAR(70) NULL,
  PRIMARY KEY (`idespecie_sem`))
ENGINE = InnoDB;

```

Elaborado por: Los Autores

Figura 19: Validación de los campos

```

CREATE TABLE IF NOT EXISTS `BGermoplasma`.`Colector` (
  `idColector` INT NOT NULL,
  `nombre` VARCHAR(45) NULL,
  `apellido` VARCHAR(45) NULL,
  `cedula` INT(11) NULL,
  `fecha_ini` DATE NULL,
  `telefono` INT(11) NULL,
  `email` VARCHAR(45) NULL,
  `ciudad` VARCHAR(45) NULL,
  `Semilla_idSemilla` INT NOT NULL,
  `Zona_idZona` INT NOT NULL,
  `Localidad_idLocalidad` INT NOT NULL,
  PRIMARY KEY (`idColector`),
  INDEX `fk_Colector_Semilla1_idx` (`Semilla_idSemilla` ASC),
  INDEX `fk_Colector_Zona1_idx` (`Zona_idZona` ASC),
  INDEX `fk_Colector_Localidad1_idx` (`Localidad_idLocalidad` ASC),
  CONSTRAINT `fk_Colector_Semilla1`
    FOREIGN KEY (`Semilla_idSemilla`)
      REFERENCES `BGermoplasma`.`Semilla` (`idSemilla`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Colector_Zona1`
    FOREIGN KEY (`Zona_idZona`)
      REFERENCES `BGermoplasma`.`Zona` (`idZona`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Colector_Localidad1`
    FOREIGN KEY (`Localidad_idLocalidad`)
      REFERENCES `BGermoplasma`.`Localidad` (`idLocalidad`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

```

Elaborado por: Los Autores

11.3.5. Pruebas

Pruebas de validación de campos del sistema permite no ingresar datos erróneos en los campos de inserción, aplicada a cada una de la interfaces graficas de usuario.

Figura 20: Código Validación de los campos

```

private void cedulatxtKeyTyped(java.awt.event.KeyEvent evt) {

    char car = evt.getKeyChar();

    if(cedulatxt.getText().length()>=10) evt.consume();

    if((car<'0' || car>'9')) evt.consume();

}

```

Elaborado por: Los Autores

La validación de campos permite no ingresar datos erróneos al sistema además de proporcionar espacio de almacenamiento.

Pruebas de conexión a la base de datos

Figura 21: Código de conexión a base de datos

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import javax.swing.JOptionPane;

/**
 *
 * @author HP14
 */
public class Conectar {

    Connection Conectar = null;

    public Connection conexion() {
        try {
            Class.forName("org.gjt.mm.mysql.Driver");
            Conectar = DriverManager.getConnection("jdbc:mysql://localhost/Almacen","root", "12345");
        } catch (Exception e) {
            System.out.print(e.getMessage());
        }
        return Conectar;
    }
}

```

Elaborado por: Los Autores

Al generar pruebas de conexión a la base de datos se debe tener muy en cuenta la similitud de las variables para su conexión de éxito.

Figura 22: Conexión a la base de datos

The screenshot shows a web-based application interface for managing seed records. It includes input fields for 'Observaciones' (citrico), 'Número de Colección' (2), 'Repeticiones' (2), and 'Duplicado' (1). There are buttons for 'Insertar', 'Nuevo Registro', and 'Modificar'. A 'Mensaje' dialog box is displayed in the foreground, stating 'Se agregó el registro de manera exitosa' (Record added successfully) with an 'Aceptar' button. Below the form is a table titled 'Registro de Semillas' with columns 'Cod Semilla', 'Nombre', and 'Observaciones'.

| Cod Semilla | Nombre | Observaciones |
|-------------|------------------|---------------------|
| 1 | Frejol Cuarenton | Alimento de cons... |
| 2 | maiz | semilla de regim... |

Fuente: Los Autores

11.4. Análisis comparativo de metodologías ágiles Scrum y XP (Programación Xtreme)

Para el análisis de las metodologías ágiles Scrum y XP se procede al diseño de un cuadro comparativo sobre las ventajas y desventajas que posee cada una de las metodologías.

Tabla 36: Ventajas y desventajas de metodologías Scrum y XP

| Metodologías Ágiles | Ventajas | Desventajas |
|--------------------------|--|--|
| Scrum | Desarrollo de software en corto periodo. | No es recomendable para proyectos grandes de desarrollo de software. |
| | Permite introducción de cambio en el software. | No aplica a proyecto de costos fijos. |
| | Posee ciclo de desarrollo corto minimizando riesgos | |
| | Se optimiza la inversión en proyectos software, generando resultados a cortos plazo de manera eficiente. | |
| XP (Programación Xtreme) | Reducción de costo | Planificación detallada desde inicio por alteración de costos |
| | Robustez del desarrollo del software de forma simple | No se planifica control de la codificación |
| | Permite cambios en requisitos del software | Conduce a demasiadas réplicas de códigos y datos Dedicado a proyectos grandes su utilización es compleja ya que está centrado en el código del software |

Elaborado por: Los Autores

Como resultado la metodología que más ventajas brinda y se asemeja al desarrollo del presente proyectos es la Metodología Ágil Scrum su facilidad de empleo, permite a un diseño de software culminar con menor tiempo, además permite la integración de cambios futuros en el sistema. La aplicación de la metodología Scrum en los proyectos software ahorra tiempo y dinero, presentando un software de calidad de la misma forma que se podría desarrollar el software aplicando otra metodología.

11.5. Resultados de comparación de metodologías Scrum y XP aplicado al sistema

Para la obtención de resultados a cerca de cual metodología ágil se adapta más al diseño del software entre Scrum y XP se genera una tabla de valores que a continuación se muestra con una escala de uno a tres para determinar el nivel de alcance que posee cada metodología.

Según las investigaciones realizadas de acuerdo a (Gopaul, 2017) se describe el siguiente cuadro de nivel de equivalencia

Tabla 37: Valores de equivalencia

| Nivel | Valor |
|-------|-------|
| Bajo | 1 |
| Medio | 2 |
| Alto | 3 |

Elaborado por: Los Autores

De acuerdo a (Gopaul, 2017) en su investigación Software Methodologies: SCRUM vs Extreme Programming, determina características comparativas de las metodologías ágiles de las cuales se realizó la comparación al estudio del presente proyecto

Tabla 38: Comparación de metodología

| Descripción | XP | SCRUM |
|---|-----------|-------------|
| Estrictamente sistemático | Medio | - |
| Documentación estricta | Medio | Medio |
| Independiente de tecnologías | - | Medio |
| Encaminado en los procesos | Medio | Medio |
| Dirigido a los desarrolladores | Alto | Medio |
| Resultados rápidos | Alto | Alto |
| Comunicación con el cliente | Medio | Medio |
| Administración de tiempo | Medio | Alto |
| Verificación y control de código interactivo en el software | Medio | Bajo |
| Respuesta a los cambios | Medio | Alto |
| Permite cambios futuros en desarrollo de software | Bajo | Alto |
| Equivalencia | 21 | ✓ 23 |

Elaborado por: Los Autores

Como resultado la metodología Scrum es una de las metodologías más utilizadas y aplicadas a los proyectos de desarrollo de software de manera ágil, segura y confiable permitiendo presentar un software de alta calidad en corto tiempo y disminución de costos evitando demoras de diseño del software, además permite realizar cambios futuros del software o implementar nuevas funciones que fortalezcan aún más el funcionamiento óptimo del software.

12. IMPACTOS (TÉCNICOS, SOCIALES, AMBIENTALES, O ECONÓMICOS).

12.1. Impacto técnico

El presente proyecto contiene un análisis exhaustivo de las metodologías Scrum y XP para la implementación de un sistema multiplataforma, a través del análisis se obtiene resultados de la metodología ágil adecuada para el diseño de un sistema software, empleado herramientas tecnológicas Open Source.

12.2. Impacto social

El proyecto sirve como base de investigación para la sociedad informática que se dedican a generar aplicaciones de desarrollo de software, contribuirá en el conocimiento de la metodologías ágiles adecuadas para el desarrollo de software.

12.3. Impacto ambiental

El presente proyecto en base al análisis de las metodologías ágiles, el diseño e implementación del sistema software, permite disminuir el excesivo manejo y cúmulo de documentos de informaciones registrado en hojas de papel, el software almacenará toda la información de forma digital, evitando la utilización de recursos naturales renovables como papeles, tinta, portafolios, etc.

12.4. Impacto económico

Al poseer un análisis de las metodologías ágiles Scrum y XP se reduce el tiempo y costo en el desarrollo del software, además al emplear herramientas tecnológicas Open Source se reduce el presupuesto de gastos por obtención de licencias de software, al ser un software libre implementado en una empresa o institución, genera menos inversión en su desarrollo con la misma o mayor calidad que los sistemas con licencia de software.

13. PRESUPUESTO PARA LA PROPUESTA DEL PROYECTO

El presupuesto general del proyecto es de 3.492,30 dólares americanos, detallados a continuación.

Tabla 39: Presupuesto del proyecto

| Presupuesto para el desarrollo del proyecto | | | |
|---|-----------------|--------------------|--------------------|
| Gastos directos | | | |
| Descripción | Cantidad | V. Unitario | V. Total |
| Recursos Tecnológicos | | | |
| Computador Portátil HP, COREi3, Disco duro 800 GB, RAM 8 GB | 2 | \$ 850,00 | \$ 1.700,00 |
| USB (memory flash 8GB) | 2 | \$ 8,00 | \$ 16,00 |
| Internet | 160 horas | \$ 1,00 | \$ 160,00 |
| Recursos Humanos | | | |
| Desarrolladores | 2 | \$ 600,00 | \$ 1.200,00 |
| Recursos Materiales | | | |
| Documentación del proyecto | 1 | \$ 30,00 | \$ 30,00 |
| Anillado | 9 | \$ 1,00 | \$ 9,00 |
| Impresiones blanco y negro y a color para revisiones | 9 | \$ 5,00 | \$ 45,00 |
| Carpetas | 3 | \$ 1,50 | \$ 4,50 |
| CD | 1 | \$ 1,50 | \$ 1,50 |
| Gastos indirectos | | | |
| Transporte movilización | 20 | \$ 6,00 | \$ 120,00 |
| Alimentación | 20 | \$ 2,00 | \$ 40,00 |
| Sub Total | | | \$ 3.326,00 |
| Gastos imprevistos 5 % | | | \$ 166,30 |
| TOTAL | | | \$ 3.492,30 |

Elaborado por: Los Autores

Detalle de herramientas de desarrollo del software de licencias sin costo.

Tabla 40: Herramientas tecnológicas Open Source

| Herramientas tecnológicas Open Source | Cantidad | V. Total |
|--|-----------------|---|
| Licencia Netbeans IDE | 1 | Licencia (GPL) Licencia Pública General |
| Licencia gestor de base de datos MySQL | 1 | Licencia (GPL) Licencia Pública General |

Elaborado por: Los Autores

14. CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- Actualmente las metodologías ágiles más utilizadas para el desarrollo de software son las metodologías Scrum y XP, estas metodologías proporcionan fases y etapas para desarrollar un proyecto de software sin tener dificultades de ejecución de entrega reduciendo el costo y tiempo para su instalación.
- A través del análisis comparativo de las metodologías ágiles Scrum y XP, se obtuvo como resultados que la metodología Scrum es mucho más eficiente en proyectos de entrega de software de desarrollo a corto tiempo una de las ventajas es que se puede realizar cambios durante el diseño del sistema, lo cual diferencia a la metodología XP que es más centralizada al diseño de proyectos de gran escala que conlleve un alto nivel de estructuración de los datos y códigos.
- Para el diseño del sistema se empleó la metodología Scrum por su ventaja de poder a futuro realizar cambios en el sistema sin afectar su proceso de desarrollo disminuyendo costo y tiempo, para ello se empleó lenguaje de programación NetBeans IDE 8.2 y gestor de base de datos MySQL que son herramientas Open Source

Recomendaciones

- Es necesario tener conocimientos acerca de las metodologías ágiles y utilizar de manera adecuada los procesos y fases que poseen para un correcto diseño y estructuración del software.
- Analizar cada uno de las fases que contiene las metodologías Scrum y XP para su correcta elección de la metodología con la cual se va a desarrollar un proyecto de software teniendo en cuenta el tiempo y estimación de ejecución del producto software.
- La aplicación de metodología ágiles a los proyectos de software mejoran y agilizan los procesos de diseño, permitiendo la satisfacción del usuario al entregar un sistema robusto sin pérdida de tiempo dinero. Para ello es recomendable utilizar herramientas Open Source en el diseño de sistemas, actualmente la mayor parte de los sistemas son más compatibles diseñados en software libre que los que contienen licencia.

15. BIBLIOGRAFÍA

- Albusae, J. (2013). Tecnologías Libres para Síntesis de Imagen Digital Tridimensional. (Lulu.com, Ed.).ISBN: 8468992801, 9788468992808.
- Altheide, C. (2011). Digital Forensics with Open Source Tool. (Elsevier, Ed.).ISBN: 1597495875, 9781597495875.
- Amaya, J. (2010). Sistemas de información gerenciales: Hardware, software, redes, Internet, diseño. (E. EDICIONES, Ed.).ISBN: 9586486354, 9789586486354.
- Bahit, E. (2012). Scrum & Extreme Programming para programadores.ISBN:1205011563509.
- Barranco, J. (2015). Metodología del análisis estructurado de sistemas (Ilustrada ed.). (U. P. Comillas, Ed.).ISBN: 8484680436, 9788484680437.
- Bibik, I. (2018). How to Kill the Scrum Monster: Quick Start to Agile Scrum Methodology and the Scrum Master Role. (Apress, Ed.)
- Chicago, M. (2016). Desarrollo Ágil. Obtenido de <http://marich.blogspot.es/1459536985/metodologia-extreme-programming/>
- Chicano, E. (2015). Gestión de incidentes de seguridad informática. IFCT0109. (I. Editorial, Ed.).ISBN: 8416351708, 9788416351701.
- Cobo, A. (2015). PHP y MySQL: Tecnologías para el desarrollo de aplicaciones Web. (E. D. Santos, Ed.).ISBN:8490520097,9781597495875.
- Durán, D. (2017). Desarrollo de prototipos de productos editoriales multimedia. ARGN0110. (I. Editorial, Ed.).ISBN: 8417224572, 9788417224578.
- Fernandez, J. (2010). Introduccion a las metodologias agile.ISBN: PID-00184460.
- Fernandez, V. (2010). Desarrollo de Sistemas de Información una Metodología Basada en el Modelado. (U. P. Catalunya, Ed.)
- Gopaul, D. (2017). Software Methodologies: SCRUM vs Extreme Programming. (I. Lulu Press, Ed.). ISBN: 138715821X, 9781387158218
- Highsmith, J. (2014). Agile Software Development Ecosystems (Ilustrada ed.). (A.-W. Professional, Ed.).ISBN: 0201760436, 9780201760439.
- Lainez, J. (2015). Desarrollo de Software Ágil: Extremme Programming y Scrum. 2ª Edición. (I. C. Academy, Ed.)
- León, F. (2010). Base de datos relacionales. (ITM, Ed.).ISBN: 9588351421, 9789588351421.
- López, G. (2015). Manual. Aplicaciones informáticas de bases de datos relacionales (Transversal: UF0322). Madrid: Certificados de profesionalidad. Editorial CEP, S.L.ISBN:9788499645971 - 338752015.
- Lopez, M. (2016). Programación Web en el Entorno Servidor. (MF0492_3). (G. E. RA-MA, Ed.).ISBN: 9788499645971 - 338752015.
- Lozano, M. (2010). Ingeniería del software y bases de datos: tendencias actuales (ilustrada+ ed.). (U. d. Mancha, Ed.).ISBN:8484270777,9788484270775.
- Moran, A. (2015). Managing Agile: Strategy, Implementation, Organisation and People (ilustrada ed.).ISBN: 3319162624, 9783319162621.
- Moreno, J. (2014). Adminitracion Software de un Sistema Informatico. (G. E. RA-MA, Ed.).ISBN: 8473563751, 9788473563758.

- Pablos, C. (2011). Informática y comunicaciones en la empresa (Ilustrada ed.). (ESIC Editorial, Ed.).ISBN: 8473563751, 9788473563758.
- Rao, K. (2012). Manual para el Manejo de Semillas en Bancos de Germoplasma (Manuales para Bancos de Germoplasma No. 8).ISBN: 929043757X, 9789290437574..
- Rinaudo, L. (2015). Ingeniería de Software. (A. G. Editor, Ed.).ISBN:9871609787,9789871609789.
- Robledo, F. (2014). Desarrollo de aplicaciones para Android II. (M. d. Educación, Ed.).ISBN: 8436955412, 9788436955415.
- Ruiz, E. (2017). Nuevas tendencias en los sistemas de información. (E. C. SA, Ed.).ISBN 8499612695, 9788499612690.
- Sommerville, I. (2015). Ingeniería de Software (Séptima ed.). España, Madrid: Pearson Educación, S.A.ISBN: 8478290745, 9788478290741.
- Stellman, A. (2014). Learning Agile: Understanding Scrum, XP, Lean, and Kanban. (I. "O'Reilly Media, Ed.).ISBN: 1449363857, 9781449363857.
- Subra, J. (2018). Scrum Un método ágil para sus proyectos (Ediciones ENI ed.).ISBN: 2409012922, 9782409012921.
- Vannieuwenhuyse, A. (2018). Scrum Un método ágil. ISBN: 9782409012921.
- Woods, D. (2015). Open Source for the Enterprise: Managing Risks, Reaping Rewards (ilustrada ed.). (I. "O'Reilly Media, Ed.).ISBN: 0596101198, 9780596101190.
- Zima, D. (2015). Modern Methods Of Software Development. 9, 14.

ANEXOS

16. ANEXOS

Anexo 1. Entrevista

ENTREVISTA

Entrevista realizado al desarrollador del software anterior de gestión especies de semillas del Banco de Germoplasma de la Universidad Técnica de Cotopaxi extensión La Maná.

Fecha: 14 de Noviembre del 2019

- 1. ¿Qué herramientas de desarrollo de software utilizan en la Universidad Técnica de Cotopaxi?**
Se utiliza varios IDE, como Java NetBeans, Dev c++, Visual Studio de Microsoft
- 2. ¿Qué lenguajes de programación utilizó para el desarrollo del Software?**
Para el desarrollo de la aplicación que esta en Germoplasma, se utilizó Visual Studio y C#.
- 3. ¿Qué gestor de base de datos utilizó para el almacenamiento de la información?**
El gestor de base de datos utilizado fue Microsoft SQL Server 2012
- 4. ¿Qué cantidad de usuarios ingresan al sistema de gestión del Banco de Germoplasma?**
Al sistema ingresa la responsable del Banco de Germoplasma Ing. Tatiana Gavilánez y estudiantes de investigación aproximadamente 15 estudiantes.
- 5. ¿Cuál es la capacidad de almacenamiento de especies de semillas del Banco de Germoplasma?**
Es menor al almacenamiento de 50.000 registros aproximadamente que puede ser registrado en la base de datos.
- 6. ¿Qué metodología de desarrollo de software implemento en el sistema?**
La metodología que se implemento fue XP que uno de los requerimientos es que el cliente es parte del equipo de desarrollo del software.
- 7. ¿Cree usted que la implementación de un nuevo software desarrollado con tecnologías Open Source, permita a los estudiantes acceder a la información de las semillas que posee el banco de Germoplasma de la UTC?**
Por supuesto que el desarrollo del software con herramientas Open Source va a facilitar bastante las cosas, pues el sistema anterior es dependiente del servidor. Internet Information Server, al ser desarrollado con herramientas open Source puede ser subido a cualquier dominio de contrato e incluso dominio gratuito.

Ing. Edel Ángel Rodríguez
ENTREVISTADO

Anexo 2. Encuesta

UNIVERSIDAD TÉCNICA DE COTOPAXI
CIENCIAS DE LA INGENIERIA Y APLICADAS
INGENIERÍA EN INFORMÁTICA Y SISTEMAS COMPUTACIONALES
ENCUESTAS

Encuestas aplicadas a los estudiantes de la carrera de Ingeniería en Informática y Sistemas Computacionales

1) **¿Usted al momento de desarrollar un software aplica alguna metodología para su desarrollo?**

SI NO

2) **¿Qué tipo de metodología utiliza para el desarrollo de software?**

Metodología Tradicional Metodología Ágil Ninguna

3) **¿Al diseñar o desarrollar un proyecto de software usted lleva a cabo el tiempo y cronograma de ejecución?**

SI NO

4) **¿Conoce las metodologías ágiles que existen para el desarrollo de software?**

SI NO

5) **¿Cuál de las siguientes metodologías ágiles de desarrollo de software ha utilizado?**

SCRUM XP (EXTREME PROGRAMMING
KANBAN NINGUNA

6) **¿Cree que es necesario llevar un control y planificación del proceso evolutivo de desarrollo de software?**

SI NO

7) **¿Cuál cree usted que es la principal razón de aplicar metodologías ágiles al desarrollar un sistema?**

MEJORAR LA CALIDAD DEL SOFTWARE
TERMINAR UN PROYECTO

Anexo 3. Análisis y tabulación de resultados de la encuesta

Análisis y tabulación de las encuestas aplicadas a los estudiantes de la carrera de Ingeniería en Informática y Sistemas Computacionales

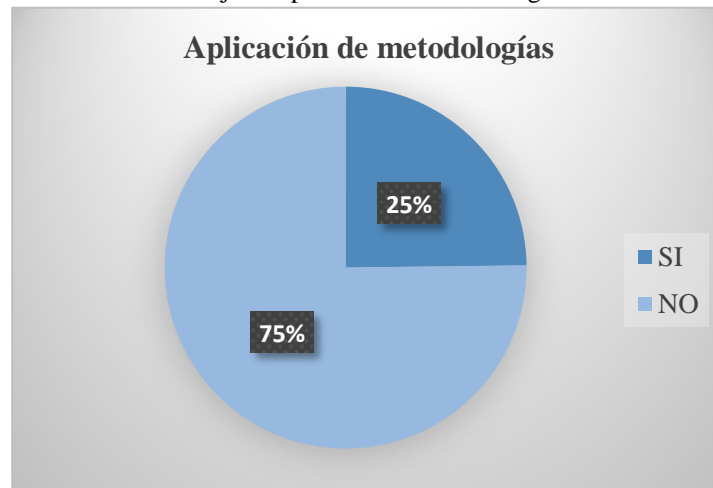
Pregunta 1: ¿Usted al momento de desarrollar un software aplica alguna metodología para su desarrollo?

Tabla 41: Porcentaje de aplicación de metodologías

| Variables | Cantidad | Porcentaje |
|--------------|----------|------------|
| Si | 32 | 25% |
| No | 97 | 75% |
| Total | 129 | 100% |

Fuente: Elaborado por los autores

Gráfico 1: Porcentaje de aplicación de metodologías



Fuente: Elaborado por los autores

Análisis e interpretación de resultados

De acuerdo al análisis de las encuestas realizadas obtuvimos un gran porcentaje de estudiantes de la carrera de Ingeniería en Informática que al momento de desarrollar un sistema no implementaban ninguna metodología para su desarrollo, simplemente lo realizaba de acuerdo al tiempo que le dé el docente de la asignatura, de acuerdo a estas investigaciones nos impulsa a seguir con el desarrollo de nuestra tesis.

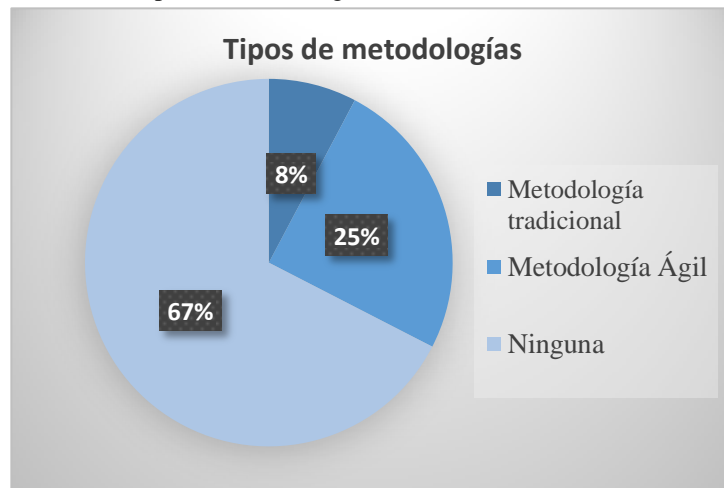
Pregunta 2: ¿Qué tipo de metodología utiliza para el desarrollo de software?

Tabla 42: Tipos de metodologías

| Variables | Cantidad | Porcentaje |
|-------------------------|------------|-------------|
| Metodología tradicional | 10 | 8% |
| Metodología Ágil | 22 | 25% |
| Ninguna | 97 | 67% |
| Total | 129 | 100% |

Fuente: Elaborado por los autores

Gráfico 9: Tipos de metodologías



Fuente: Elaborado por los autores

Análisis e interpretación de resultados

De acuerdo a las encuestas obtuvimos que apenas el 25% por ciento de estudiantes utilizan metodologías ágiles, el 8% metodologías tradicionales y un gran porcentaje de 67% no utilizan ninguna metodología lo cual confirma de que desarrollan los proyectos de software sin la aplicación de Sprint, ni ningún tipo de cronograma establecidos para el desarrollo del software.

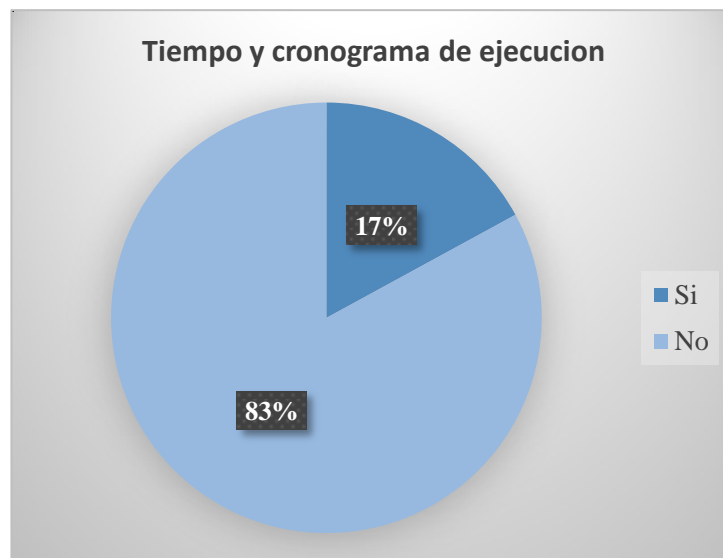
Pregunta 3: ¿Al diseñar o desarrollar un proyecto de software usted lleva a cabo el tiempo y cronograma de ejecución?

Tabla 43: Tiempo y cronograma de ejecución

| Variables | Cantidad | Porcentaje |
|--------------|----------|------------|
| Si | 22 | 17% |
| No | 107 | 83% |
| Total | 129 | 100% |

Fuente: Elaborado por los autores

Gráfico 17: Tiempo y cronograma de ejecución



Fuente: Elaborado por los autores

Análisis e interpretación de resultados

Como podemos visualizar solo el 17% por ciento de estudiantes planifican el desarrollo del software, con el propósito de obtener un mejor resultado al momento de generar reportes del funcionamiento del sistema, a la diferencia de los demás estudiantes que no llevan ninguna planificación de desarrollo.

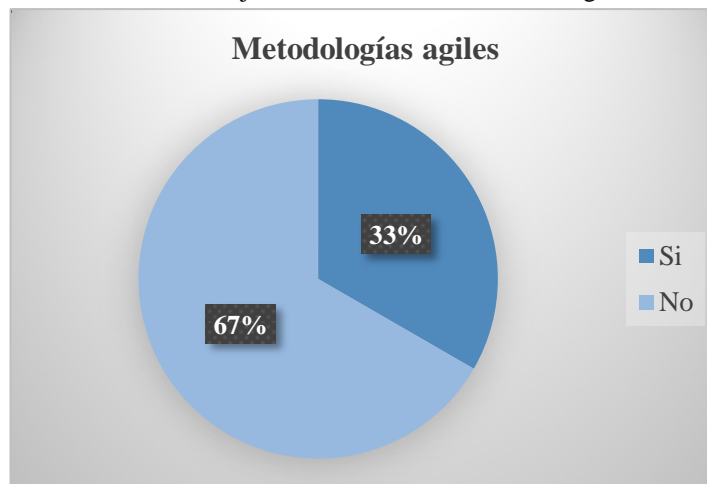
Pregunta 4: ¿Conoce las metodologías ágiles que existen para el desarrollo de software?

Tabla 44: Porcentaje de conocimiento de metodologías

| Variables | Cantidad | Porcentaje |
|-----------|----------|------------|
| Si | 43 | 33% |
| No | 86 | 67% |
| Total | 129 | 100% |

Fuente: Elaborado por los autores

Gráfico 25: Porcentaje de conocimiento de metodologías



Fuente: Elaborado por los autores

Análisis e interpretación de resultados

El 67% de estudiantes encuestados no conocen las diferentes metodologías de desarrollo de software, lo cual nos constata que debido a esto al momento de presentar depuración del sistema, genera diferentes errores, ya que no se desarrolla dentro de un marco de tiempo y con un lapso de sprints que aseguren el funcionamiento y calidad de software.

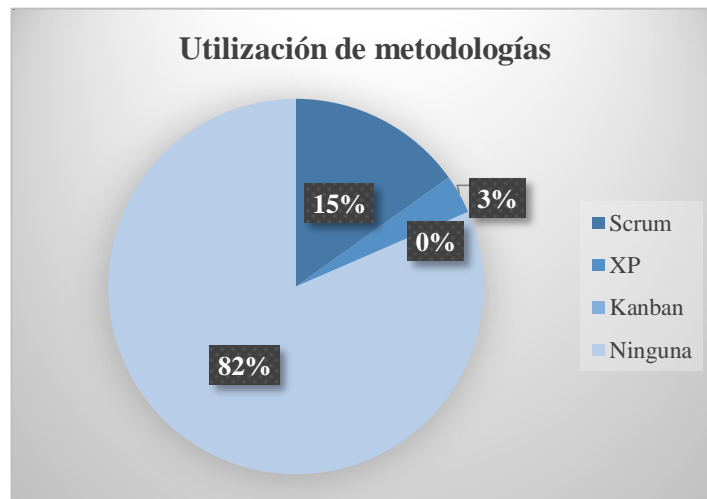
Pregunta 5: ¿Cuál de las siguientes metodologías ágil de desarrollo de software ha utilizado?

Tabla 45: Utilización de metodologías

| Variables | Cantidad | Porcentaje |
|------------------|-----------------|-------------------|
| Scrum | 18 | 15% |
| XP | 4 | 3% |
| Kanban | 0 | 0% |
| Ninguna | 97 | 82% |
| Total | 129 | 100% |

Fuente: Elaborado por los autores

Gráfico 33: Utilización de metodologías



Fuente: Elaborado por los autores

Análisis e interpretación de resultados

El 15% de encuestados tienen conocimientos acerca de la metodología Scrum y un porcentaje pequeño acerca de Xp, lo cual nos da como resultado que gran parte de los estudiantes desconocen las metodologías de desarrollo de software lo cual genera controversias al momento de desarrollarlos.

Pregunta 6: ¿Cree que es necesario llevar un control y planificación del proceso evolutivo de desarrollo de software?

Tabla 46: Control y planificación

| Variables | Cantidad | Porcentaje |
|--------------|----------|------------|
| Si | 129 | 100% |
| No | 0 | 0% |
| Total | 129 | 100% |

Fuente: Elaborado por los autores

Gráfico 41: Control y planificación



Fuente: Elaborado por los autores

Análisis e interpretación de resultados

Como observamos el gráfico el 100% de los estudiantes creen que es necesario llevar un control y planificación del proceso de desarrollo y evolutivo del software, pero al desconocer el proceso y tiempo que llevan los Sprint de cada metodología no lo han llevado a cabo.

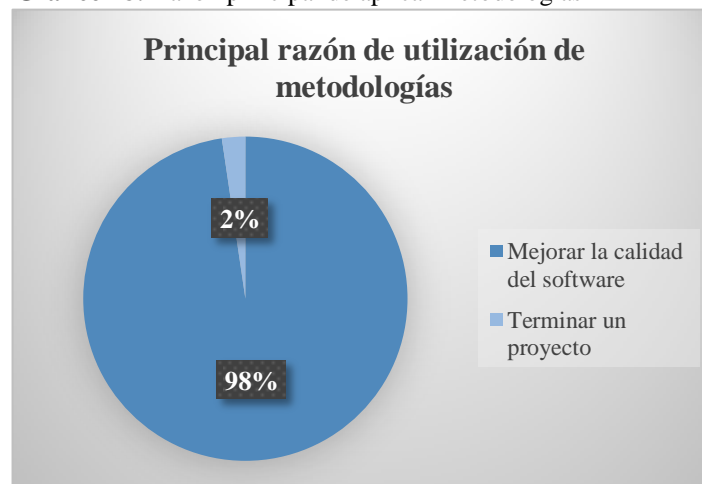
Pregunta 7: ¿Cuál cree usted que es la principal razón de aplicar metodologías ágiles al desarrollar un sistema?

Tabla 47: Razón principal de utilizar metodologías

| Variables | Cantidad | Porcentaje |
|---------------------------------|-----------------|-------------------|
| Mejorar la calidad del software | 126 | 98% |
| Terminar un proyecto | 3 | 2% |
| | | |
| Total | 129 | 100% |

Fuente: Elaborado por los autores

Gráfico 48: Razón principal de aplicar metodologías



Fuente: Elaborado por los autores

Análisis e interpretación de resultados

Las metodologías en si tienen como finalidad mejorar la calidad de software lo cual es beneficiario al momento de entregar un producto a un usuario cumpliendo con un tiempo determinado y planificado de su desarrollo, y es el principal motivo por el cual se lleva a cabo el desarrollo de proyecto, es por tal motivo que el 98% de estudiantes selecciono la opción que observamos en el gráfico.

Anexo 4. Prueba Caja Blanca

A través de las pruebas de caja blanca obtenemos el funcionamiento de los enlaces entre la base de datos y la interfaz gráfica de usuario, permitiendo corregir a tiempo los errores si se suscitan ayudando a mejorar falencias que se presenta.

Figura 23: Prueba de funcionamiento del script de la base de datos

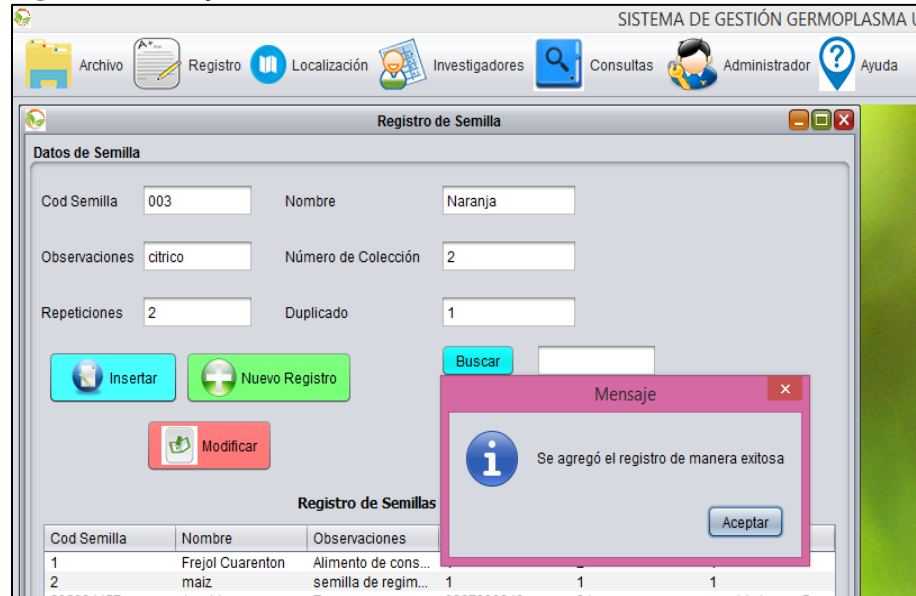
```
Schema BGermoplasma--
-----
Schema BGermoplasma--
-----
CREATE SCHEMA IF NOT EXISTS `BGermoplasma` DEFAULT
CHARACTER SET utf8 COLLATE utf8_general_ci ;USE
`BGermoplasma` ;--
-----
Table `BGermoplasma`.`Localidad`--
-----
CREATE TABLE IF NOT EXISTS `BGermoplasma`.`Localidad` (
`idLocalidad` INT NOT NULL, `nombre` VARCHAR(25) NULL,
`referencia` VARCHAR(45) NULL, `distancia` VARCHAR(25)
NULL, `canton` VARCHAR(15) NULL, `parroquia` VARCHAR(15)
NULL, `provincia` VARCHAR(15) NULL, `pais` VARCHAR(15)
NULL, `descripcion` VARCHAR(45) NULL, `observaciones`
VARCHAR(45) NULL, `areaprotegid` TINYINT(1) NULL,
PRIMARY KEY (`idLocalidad`))ENGINE = InnoDB;--
-----
Table `BGermoplasma`.`Familia_sem`--
-----
CREATE TABLE IF NOT EXISTS `BGermoplasma`.`Familia_sem` (
`idfamilia_sem` INT NOT NULL, `nombre_fam` VARCHAR(25)
NULL, `procedencia_fam` VARCHAR(15) NULL, `descrip_fam`
VARCHAR(45) NULL, `odserv_fam` VARCHAR(45) NULL, PRIMARY
KEY (`idfamilia_sem`))ENGINE = InnoDB;--
-----
Table `BGermoplasma`.`Especie_sem`--
-----
CREATE TABLE IF NOT EXISTS `BGermoplasma`.`Especie_sem` (
`idespecie_sem` INT NOT NULL, `nombre_esp` VARCHAR(25)
NULL, `descripcion_esp` VARCHAR(70) NULL,
`observacion_esp` VARCHAR(70) NULL, PRIMARY KEY
(`idespecie_sem`))ENGINE = InnoDB;--
-----
Table `BGermoplasma`.`Semilla`--
-----
CREATE TABLE IF NOT EXISTS `BGermoplasma`.`Semilla` (
`idSemilla` INT NOT NULL, `num_coleccion` VARCHAR(10)
NULL, `nom_semilla` VARCHAR(78) NULL, `duplicad_semilla`
VARCHAR(10) NULL, `repet_semillas` INT(11) NULL,
`observ_semillas` VARCHAR(250) NULL,
`familia_sem_idfamilia_sem` INT NOT NULL,
`especie_sem_idespecie_sem` INT NOT NULL, PRIMARY KEY
(`idSemilla`), INDEX `fk_Semilla_Familia_sem1_idx`
(`familia_sem_idfamilia_sem` ASC), INDEX
`fk_Semilla_Especie_sem1_idx` (`especie_sem_idespecie_sem`
ASC), CONSTRAINT `fk_Semilla_Familia_sem1` FOREIGN KEY
(`familia_sem_idfamilia_sem`) REFERENCES
`BGermoplasma`.`Familia_sem` (`idfamilia_sem`) ON
DELETE NO ACTION ON UPDATE NO ACTION, CONSTRAINT
`fk_Semilla_Especie_sem1` FOREIGN KEY
(`especie_sem_idespecie_sem`) REFERENCES
`BGermoplasma`.`Especie_sem` (`idespecie_sem`) ON
DELETE NO ACTION ON UPDATE NO ACTION)ENGINE = InnoDB;--
-----
Table `BGermoplasma`.`Zona`--
-----
CREATE TABLE IF NOT EXISTS `BGermoplasma`.`Zona` (
`idZona` INT NOT NULL, `clima` VARCHAR(15) NULL,
`vegetacion` VARCHAR(10) NULL, `tipo_suelo` VARCHAR(25)
NULL, `altitud` VARCHAR(15) NULL, `longitud` VARCHAR(15)
NULL, `latitud` VARCHAR(15) NULL, `observaciones`
VARCHAR(125) NULL, `descripcion` VARCHAR(125) NULL,
`localidad_idLocalidad` INT NOT NULL, PRIMARY KEY
(`idZona`), INDEX `fk_Zona_Localidad1_idx`
(`localidad_idLocalidad` ASC), CONSTRAINT
`fk_Zona_Localidad1` FOREIGN KEY
(`localidad_idLocalidad`) REFERENCES
`BGermoplasma`.`Localidad` (`idLocalidad`) ON DELETE NO
```

Fuente: Los Autores

Anexo 5. Prueba Caja Negra

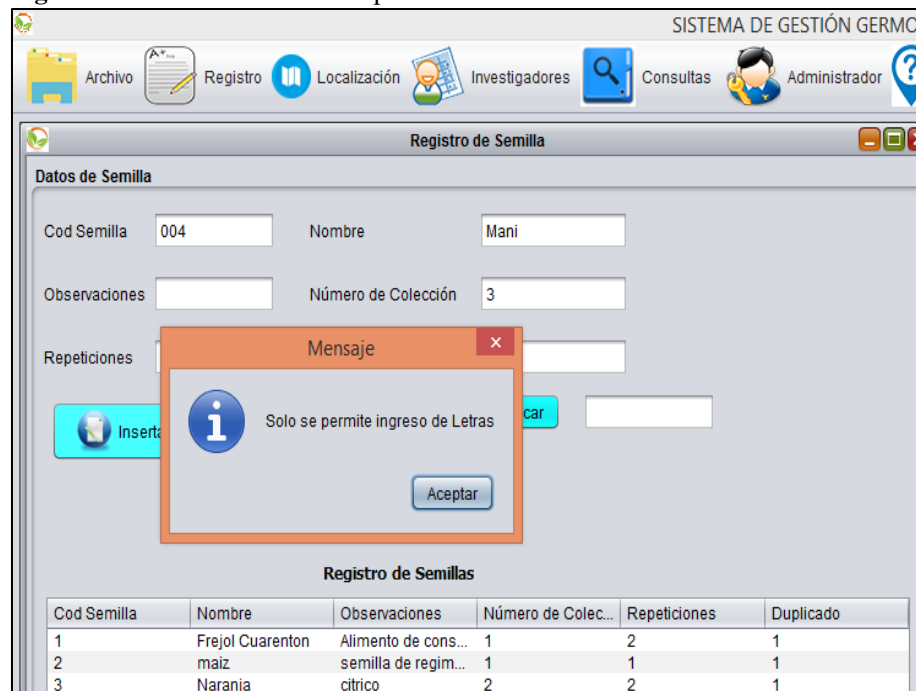
La realización de pruebas a la interfaz gráfica del usuario permite obtener resultados favorables para verificar la funcionalidad del software.

Figura 24: Mensaje de conexión a la base de datos



Fuente: Los Autores

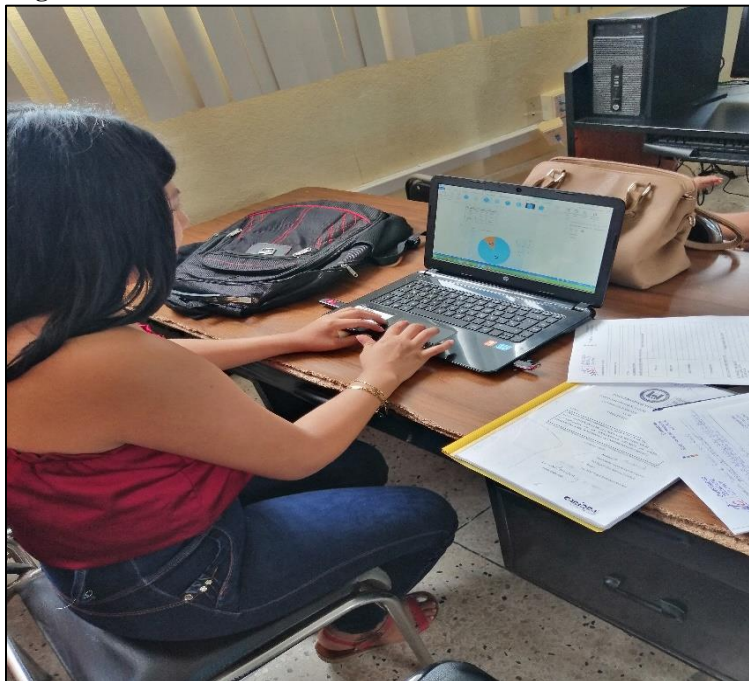
Figura 25: Validación de los campos



Fuente: Los Autores

Anexo 6. Tabulación e interpretación de la encuesta

Figura 26: Tabulación de los resultados de la encuesta



Fuente: Los Autores

Anexo 7. Desarrollo y diseño del software

Figura 27: Diseño de la interfaz gráfica de usuario.



Fuente: Los Autores

Anexo 8. Hoja de vida del tutor del proyecto de investigación.

CURRICULUM VITAE



1. INFORMACIÓN PERSONAL

| | |
|-------------------------------|------------------------------|
| Documento de identidad | 175722381-1 |
| No. Senescyt | 72413156 |
| Nacionalidad | cubano |
| Fecha de nacimiento | 11 julio 1980 |
| Teléfonos | 0983564541 |
| e-mail | soyedelangel@gmail.com |
| Idioma | Extranjero Inglés (Avanzado) |

2. ESTUDIOS SUPERIORES

Ingeniería Informática.

Institución Universidad de Granma. Cuba 2009

Máster Universitario en Ingeniería de Software e Inteligencia Artificial

Institución Universidad de Málaga España 2011

3. FORMACIÓN LABORAL

Técnico de sistemas Informáticos Delegación Provincial del CITMA Proyecto de Recuperación de la Cuenca del Cauto. Granma, Cuba.

Técnico B de Apoyo a la Docencia Universidad de Granma Centro de Información Científico Técnica Granma, Cuba.

Profesor instructor, Universidad de Granma Departamento de Informática Granma, Cuba



Ing. M.Sc. Rodríguez Sánchez Edel Ángel
C.I: 175722381-1

Anexo 9. Hoja de vida de los autores del proyecto de investigación.

CURRICULUM VITAE



1. INFORMACIÓN PERSONAL

Nombres y Apellidos: Edwin Ricardo Manobanda Tuapanta

Cédula de Identidad: 020222445-7

Lugar y fecha de nacimiento: Guanujo, 6 de Noviembre de 1995

Nacionalidad: Ecuatoriano

Estado Civil: Soltero

Dirección: Calle secundaria vía a Zapotal, Diagonal al Consejo de la Judicatura, Barrio Miraflores, Cantón Las Naves

Teléfono: 0989280283

Correo electrónico: edwin.manobanda4457@utc.edu.ec

2. ESTUDIOS REALIZADOS

Primer Nivel: Escuela Fiscal Mixta General Vicente Anda Aguirre

Segundo Nivel: Colegio Técnico en Comercio y Administración Zapotal

Tercer Nivel: Universidad Técnica de Cotopaxi

3. TITULOS

Bachiller Técnico en Comercio y Administración, Especialización Aplicaciones Informáticas.

4. IDIOMAS

Español (nativo)

Suficiencia en el Idioma Inglés "B1"

Manobanda Tuapanta Edwin Ricardo
C.I: 020222445-7

CURRICULUM VITAE



1. INFORMACIÓN PERSONAL

Nombres y Apellidos: Mariela Elizabeth Millingalli Calo

Cédula de Identidad: 0504258500

Lugar y fecha de nacimiento: La Maná, 30 de Junio de 1997

Estado Civil: soltera

Tipo de Sangre: O+

Domicilio: La Maná-Parroquia el Carmen-Barrio Angueta Moreno

Teléfonos: 0986420850

Correo electrónico: mariela.millingalli8500@utc.edu.ec

2. ESTUDIOS REALIZADOS

Primer Nivel: Escuela Fiscal de Niñas “Luis Andino Gallegos”

Segundo Nivel: Unidad Educativa La Maná

Tercer Nivel: Universidad Técnica de Cotopaxi

3. TITULOS

Bachiller en Aplicaciones Informáticas.

4. IDIOMAS

Español (nativo)

Suficiencia en el Idioma Inglés “B1”

Millingalli Calo Mariela Elizabeth
C.I: 050425850-0