

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DEFINICIÓN DE MEJORES PRÁCTICAS PARA MEJORAR LA CALIDAD DE PROCESOS DE SOFTWARE BASADAS EN LA NORMA ISO/IEC 29110-5-1-2: PERFIL BÁSICO EN MIPYMES

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL GRADO DE MAGISTER EN SOFTWARE CON MENCIÓN EN CALIDAD DE SOFTWARE

ING. LUIS ALBERTO CASTILLO SALINAS

luis.castillo01@epn.edu.ec

Director: PhD. Sandra Patricia Sánchez Gordón

sandra.sanchez@epn.edu.ec

Codirector: PhD. Edison Fernando Loza Aguirre

edison.loza@epn.edu.ec

Quito, Noviembre del 2018

APROBACIÓN DEL DIRECTOR

Como director del trabajo de titulación **DEFINICIÓN DE MEJORES PRÁCTICAS PARA MEJORAR LA CALIDAD DE PROCESOS DE SOFTWARE BASADAS EN LA NORMA ISO/IEC 29110-5-1-2: PERFIL BÁSICO EN MIPYMES** desarrollado por Luis Alberto Castillo Salinas, estudiante de la Maestría en Software con Mención en Calidad de Software, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, doy por aprobada la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la Defensa oral.

PhD. Sandra Patricia Sánchez

DIRECTOR

PhD. Edison Fernando Loza

CODIRECTOR

DECLARACIÓN DE AUTORÍA

Yo, Luis Alberto Castillo Salinas, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Ing. Luis Alberto Castillo Salinas

DEDICATORIA

A la memoria mi amada madre Mariana de Jesús Salinas Jiménez, quien fue pilar fundamental en mi educación y quien estuvo conmigo en cada momento de debilidad y fortaleza. Quisiste que llegue lejos en mi vida profesional, y aquí estoy cumpliendo lo que te prometí, este no es el final, es un paso más.

A mi padre Luis Alberto Castillo Velásquez y mis hermanos Bianca y Gonzalo, quienes a pesar de cualquier circunstancia han estado y estarán siempre apoyándome, y a quienes les debo una parte muy importante de mi vida. Estoy siendo el ejemplo que ven en mí.

A mi novia Katherine Calle, y mis hijas, Camila y Valentina, quienes todos los días están presentes a mi lado y me dan esa fortaleza que me permite superarme en cada momento. Todos mis éxitos futuros son dedicados para ustedes.

AGRADECIMIENTO

Mis más sinceros agradecimientos a mi directora, Sandra Sánchez, por haber hecho posible la obtención de este grado de master, a apoyarme hasta el final e incluso después para la publicación del artículo profesional.

A los estudiantes del curso “SIC734 Calidad de Software” del semestre 2018-A, quienes formaron parte del experimento realizado para la realización de este proyecto de titulación.

A mi novia Katherine Calle, quien fue un elemento muy importante para la redacción en inglés del artículo profesional que publicaré. Y quien estuvo apoyándome todos los días para culminar con éxito este proyecto.

A mis hijas Camila y Valentina por ser el motor que me impulsa todos los días a superarme.

A mi padre y mis hermanos, quienes fueron un apoyo incondicional para la obtención de este título. Siempre les estaré agradecido por todo lo que soy.

A amigos Christian, José Luis y Mauricio que han estado pendientes de que culmine con éxitos mi maestría, y han sido un apoyo extra para completar este proyecto.

Y a todas aquellas personas que de una manera u otra se han hecho presentes con mensajes y palabras de fortaleza, apoyando la obtención de esta maestría.

ÍNDICE DE CONTENIDO

RESUMEN	V
ABSTRACT	VI
1. INTRODUCCIÓN	1
1.1. PREGUNTA DE INVESTIGACIÓN	2
1.2. OBJETIVO GENERAL	3
1.3. OBJETIVOS ESPECÍFICOS	3
1.4. MARCO TEÓRICO	3
1.4.1. Calidad de Procesos y Productos de Software	3
1.4.2. Experimentación en Ingeniería de Software	4
1.4.3. ISO/IEC 29110	6
1.4.4. ISO/IEC 25000	11
1.4.5. Herramienta SonarQube	12
1.4.6. Trabajos relacionado	13
2. METODOLOGÍA	14
2.1. DISEÑO DEL EXPERIMENTO	14
2.2. SUJETOS EXPERIMENTALES	15
2.3. ALCANCE DEL PRODUCTO DE SOFTWARE	16
2.4. EVALUACIÓN DE LA CALIDAD DEL PROCESO SI	19
2.5. EVALUACIÓN DE LA CALIDAD DEL PRODUCTO DE SOFTWARE	21
3. RESULTADOS Y DISCUSIÓN	22
3.1. RESULTADOS	23
3.1.1. Resultados de la Evaluación de la Calidad del Proceso SI	23
3.1.2. Resultados de la Evaluación de la Calidad del Producto	32
3.2. DISCUSIÓN	39
3.2.1. Evaluación del Proceso	39
3.2.2. Evaluación del Producto	41
3.2.3. Conjunto de Prácticas Para la Mejora del Proceso de Implementación Software de la norma ISO/IEC 29110	42

3.2.3.1	Conjunto de Prácticas para la Actividad “ <i>Software Implementation Initiation</i> ”	43
3.2.3.2	Conjunto de Prácticas para la Actividad “ <i>Software Requirements Analysis</i> ”	44
3.2.3.3	Conjunto de Prácticas para la Actividad “ <i>Software Architectural and Detailed Design</i> ”	45
3.2.3.4	Conjunto de Prácticas para la Actividad “ <i>Software Construction</i> ”	46
3.2.3.5	Conjunto de Prácticas para la Actividad “ <i>Software Integration and Tests</i> ”	47
3.2.3.6	Conjunto de Prácticas para la Actividad “ <i>Product Delivery</i> ”	48
4	CONCLUSIONES Y RECOMENDACIONES	49
4.1.	CONCLUSIONES	49
4.2.	RECOMENDACIONES	51
	REFERENCIAS BIBLIOGRÁFICAS	52
	ANEXOS	56

LISTA DE FIGURAS

Figura 1 - Series ISO/IEC 29110 [23].....	8
Figura 2 - Guía de procesos para " <i>Basic Profile</i> " [23]	9
Figura 3 - Diagrama del proceso " <i>Software Implementation</i> " [28].....	10
Figura 4 - La calidad y el ciclo de vida [31]	11
Figura 5 – Formato de historia de usuario elaborado por el equipo #2	17
Figura 6 - Diagrama de Gantt del proyecto para los equipos controlados #2 y #4	18
Figura 7 - Diagrama de Gantt del proyecto para el equipo no controlado #1	18
Figura 8 - Diagrama de Gantt del proyecto para el equipo no controlado #3.....	19
Figura 9 - Scores obtenidos en la actividad " <i>Software Requirements Analysis</i> "	28
Figura 10 - Scores obtenidos en la actividad " <i>Software Architectural and Detailed Design</i> "	29
Figura 11 - Scores obtenidos en la actividad " <i>Software Construction</i> "	29
Figura 12 - Scores obtenidos en la actividad " <i>Software Integration and Tests</i> "	30
Figura 13 - Scores obtenidos en la actividad " <i>Product Delivery</i> "	31
Figura 14 – Tendencia de desempeño de los equipos.....	31
Figura 15 - Resultados de calidad interna del equipo no controlado #1	33
Figura 16 - Resultados de calidad interna del equipo controlado #2	33
Figura 17 - Resultados de Calidad Interna del equipo no controlado #3	34
Figura 18 - Resultados de calidad interna del equipo controlado #4	34
Figura 19 - Resumen de los problemas de calidad interna	35
Figura 20 - Resultados de las métricas de calidad externa "Efectividad de la documentación del usuario o ayuda del sistema" y "Claridad de mensajes"	36
Figura 21 - Resultados de las métricas de calidad externa "Verificación de entradas válidas" y "Métodos de autenticación"	36
Figura 22 - Encuesta de satisfacción. Preguntas 1 - 3.....	37
Figura 23 - Encuesta de satisfacción. Preguntas 4 - 6.....	38
Figura 24 - Resultados de la encuesta de satisfacción para la calidad en uso.....	38

LISTA DE TABLAS

Tabla 1 - Roles involucrados en el proceso SI. [23].....	16
Tabla 2 - Lista de requerimientos identificados del producto de software	16
Tabla 3 - Escala de clasificación N-P-L-F [33].....	19
Tabla 4 - Esquema de evaluación para equipos controlados (#2 y #4) y no controlados (#1 y #3).....	20
Tabla 5 - Calendario del proyecto para los equipos controlados #2 y #4	22
Tabla 6 - Calendario de proyecto del equipo no controlado #1	22
Tabla 7 - Calendario de proyecto del equipo no controlado #3	23
Tabla 8 – Equivalencias entre la escala NPLF y su respectivo score.....	23
Tabla 9 - Scores y sumatoria por cada actividad del proceso SI para el equipo no controlado #1	24
Tabla 10 - Scores y sumatoria por cada actividad del proceso SI para el equipo controlado #2.....	25
Tabla 11 - Scores y sumatoria por cada actividad del proceso SI para el equipo no controlado #3.....	26
Tabla 12 - Scores y sumatoria por cada actividad del proceso SI para el equipo controlado #4.....	27
Tabla 13 - Formato para detallar las prácticas determinadas	42
Tabla 14 - Prácticas para la actividad " <i>Software Implementation Initiation</i> ". Parte 1	43
Tabla 15 - Prácticas para la actividad " <i>Software Implementation Initiation</i> ". Parte 2	43
Tabla 16 - Prácticas para la actividad " <i>Software Implementation Initiation</i> ". Parte 3	44
Tabla 17 - Prácticas para la actividad " <i>Software Requirements Analysis</i> ". Parte 1	44
Tabla 18 - Prácticas para la actividad " <i>Software Requirements Analysis</i> ". Parte 2	45
Tabla 19 - Prácticas para la actividad " <i>Software Architectural and Detailed Design</i> ". Parte 1	45

Tabla 20 - Prácticas para la actividad " <i>Software Architectural and Detailed Design</i> ". Parte 2	46
Tabla 21 - Prácticas para la actividad " <i>Software Construction</i> ". Parte 1	46
Tabla 22 - Prácticas para la actividad " <i>Software Construction</i> ". Parte 2	47
Tabla 23 - Prácticas para la actividad " <i>Software Integration and Tests</i> ". Parte 1 ..	47
Tabla 24 - Prácticas para la actividad " <i>Software Integration and Tests</i> ". Parte 2 ..	48
Tabla 25 - Prácticas para la actividad " <i>Product Delivery</i> ". Parte 1	48
Tabla 26 - Prácticas para la actividad " <i>Product Delivery</i> ". Parte 2	49

LISTA DE ANEXOS

Anexo I - Interfaces del producto de software del equipo no controlado #1	57
Anexo II - Interfaces del producto de software del equipo controlado #2.....	62
Anexo III - Interfaces del producto de software del equipo no controlado #3.....	70
Anexo IV - Interfaces del producto de software del equipo controlado #4	77
Anexo V – Historias de Usuario del equipo no controlado #2	84

RESUMEN

La competitividad de las empresas de desarrollo de software depende de su capacidad para ofrecer productos de software con atributos de calidad. La gran mayoría de las pequeñas entidades (VSE) que desarrollan software no ven los beneficios que resultan de la aplicación de estándares de calidad de software en sus organizaciones. En consecuencia, limitan su potencial para ser reconocidos como empresas de desarrollo de software de calidad. En este estudio, los autores presentan los resultados obtenidos a través de la aplicación de la ingeniería de software empírica en un experimento en el que la norma ISO/IEC 29110-5-1-2: 2011 “Ingeniería de software - Perfiles de ciclo de vida para entidades muy pequeñas (VSE) - - Parte 5-1-2: Guía de administración e ingeniería: Grupo de perfil genérico: Perfil básico”. Dos equipos experimentales adoptaron el estándar, mientras que otros dos equipos experimentales no tuvieron una guía estándar, pero desarrollaron el mismo producto dentro del mismo período de tiempo y alcance. El objetivo del experimento fue determinar las mejores prácticas para la mejora de la calidad del proceso de implementación de software. La facilidad de uso de la norma, como guía de proceso para la implementación de software, permitió a los equipos experimentales que adoptaron la norma ejecutar las actividades de proceso con mejor calidad.

Palabras clave: ISO/IEC 29110, experimentación, Ingeniería de Software Empírica, calidad de procesos, calidad de producto.

ABSTRACT

The competitiveness of software development enterprises depends on their ability to offer software products with quality attributes. The vast majority of very small entities (VSEs) that develop software do not see the benefits that result from applying software quality standards in their organisations. Consequently, they limit their potential to be recognised as quality software development enterprises. In this study, the authors present results obtained through the application of empirical software engineering in an experiment in which the standard ISO/IEC 29110-5-1-2: 2011 “Software engineering -- Lifecycle profiles for Very Small Entities (VSEs) -- Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile” was used. Two experimental teams adopted the standard, while two other experimental teams did not have any standard guidance but developed the same product within the same timeframe and scope. The goal was to determine the best practices for the improvement of the software process implementation. The usability of the standard, as process guidance for software implementation, enabled the experimental teams that adopted the standard to execute the process activities with better quality.

Keywords: ISO/IEC 29110, experimentation, Empirical Software Engineering, process quality, product quality.

1. INTRODUCCIÓN

La correcta selección y aplicación de normas de calidad de software en una organización, debe incrementar la productividad y tener un impacto económico positivo en la misma [1]. Sin embargo, las micro, pequeñas y medianas empresas (con sus siglas en inglés VSEs – *Very Small Entities*) que desarrollan software tienen frecuentemente problemas para relacionar las normas ISO/IEC a sus necesidades de negocio. Las VSEs que desarrollan software, en su gran mayoría, no ven el beneficio que involucra la aplicación de normas de calidad de software en sus organizaciones. Esto se debe a que las VSEs consideran que estas prácticas son muy costosas; que están destinados a organizaciones grandes; que se requiere de mucha documentación o que no se cuenta con el tiempo y empleados necesarios [2] [3]. Cabe recalcar que, sin una certificación, las VSEs que desarrollan software tienen muy poca posibilidad de ser reconocidas como empresas que desarrollan sistemas o productos de software de calidad. Por lo cual no toman ventaja de algunas fuentes de ingreso para sus organizaciones.

En la actualidad, la capacidad de competir de las empresas que desarrollan software, depende de que éstas sean capaces de ofrecer productos de software con atributos de calidad [4]. Para ilustrar, la industria automotriz, incorpora en sus autos de gama alta más de 100 millones de líneas de código. Por lo que tienen proveedores de quienes consiguen productos de software necesarios para su industria. Estos proveedores a su vez adquieren los productos de software de productores de más bajo nivel (VSEs). En este sentido, si un fabricante usó un producto de software con un error que fue producido por uno de sus miles de productores, este error se transforma en millones de dólares de pérdida [1].

En todo el mundo existen casos de VSEs que han aplicado la norma ISO/IEC 29110. Tailandia es de los primeros países donde más de 350 empresas públicas y privadas, lograron obtener la certificación “*ISO/IEC 29110 Basic Profile*”. En México existen alrededor de 40 empresas han alcanzado dicha certificación [1]. En Colombia, solamente una empresa ha recibido esta certificación [4]. En Ecuador, las VSEs representan el 90% del sector empresarial [5], de las cuales existen más de 500 empresas dedicadas a consultorías, desarrollo y venta de productos de software [6]. De estas empresas, 160 están distribuidas en las principales ciudades del país: 98 (61,25%) en Quito, 36 (22,5%) en Guayaquil, y 26 (16,25%) en Cuenca [5]. Pero no existen registros de certificaciones otorgadas a alguna de estas empresas. Por tanto, los potenciales beneficiarios de este

trabajo son las 500 empresas dedicadas a consultorías, desarrollo y venta de productos de software, especialmente las 160 distribuidas en las tres principales ciudades del país. En la práctica resulta muy importante realizar evaluaciones, tanto de la calidad del proceso como de la calidad del producto de software [7]. La calidad del proceso SI y su certificación, permite, a una organización, obtener una ventaja competitiva en el mercado. Mientras que la calidad del producto de software, además de permitir alcanzar una ventaja competitiva en el mercado, ayuda a alcanzar una mayor satisfacción por parte del usuario final, generando confianza en el mismo y permitiendo que se refiera la organización con otros usuarios [8]. Por este motivo, además de la evaluación del proceso SI, se debe realizar la evaluación de la calidad los productos finales, esta evaluación se la puede realizar con la norma ISO/IEC 25000.

El problema a ser resuelto en este trabajo recae en la falta de calidad en los procesos de desarrollo de software en VSEs. El objetivo es definir las mejores prácticas para mejorar los procesos de implementación de software (por sus siglas en inglés SI) basadas en la norma ISO/IEC 29110-5-1-2: "Ingeniería de Software – Perfiles de Ciclo de Vida en VSEs - Parte 5-1-2: Guía de gestión e ingeniería: Grupo de perfil genérico: Perfil básico" [9]. Para cumplir con este objetivo se utilizará Ingeniería de "Software Empírica".

La introducción a la experimentación se proporciona a través de un enfoque de proceso que se centra en los pasos a seguir para realizar un experimento [10]. El propósito de la experimentación en la Ingeniería de Software es introducir a los interesados en estudios empíricos de Ingeniería de Software, utilizando experimentos bajo ambientes controlados [11]. Al realizar un experimento de la aplicación de la norma ISO/IEC29110-5-1-2 en una VSE, podremos determinar un conjunto de prácticas para mejorar el proceso SI de la misma. El éxito de la aplicación de esta norma dependerá de factores como: el compromiso por parte de la empresa u organización, la correcta implementación de la norma, los métodos de seguimiento y control, entre otras.

1.1. Pregunta de Investigación

¿Es factible definir prácticas para mejorar los procesos de desarrollo de software basándose en la norma ISO/IEC 29110-5-1-2: Perfil básico en VSEs?

1.2. Objetivo General

Definir las mejores prácticas para mejorar los procesos de software basadas en la norma ISO/IEC 29110-5-1-2: "Perfil básico en VSEs".

1.3. Objetivos Específicos

- Diseñar el experimento de software.
- Analizar los requerimientos de software
- Realizar una evaluación del experimento de software.
- Identificar las mejores prácticas para la mejora de procesos de software.

1.4. Marco Teórico

1.4.1. Calidad de Procesos y Productos de Software

Según [12], la calidad es el grado en el que un conjunto de características inherentes de un objeto cumple con los requisitos. Con la definición general de calidad podemos derivar hacia definiciones más específicas como la calidad de procesos y productos de software.

La calidad de los procesos de desarrollo de software es el pilar más importante de una organización, ya que permite obtener mayor importancia en su cultura empresarial y mantiene una diferenciación para sus clientes. Para que una organización pueda asegurar la calidad de sus procesos es necesario implementar un proceso de evaluación y mejora de los mismos, permitiendo alinear el desarrollo de la empresa a las mejores prácticas de la Ingeniería de software [13]. El objetivo principal de la evaluación y mejora de la calidad de procesos de desarrollo de software es controlar la calidad del mismo a través de la mejora continua de los procesos que se utilizan para su desarrollo.

Sin duda, la calidad de los procesos de software se está convirtiendo en un elemento estratégico de las organizaciones en general, ya que provee un impacto significativo en la competitividad de las empresas. El auge de los procesos de calidad se debe a que en los últimos años los productos de software están inmersos en la mayor parte de sectores de una organización. Además, representa un elemento clave en la innovación de las empresas, en cuanto a productos y procesos.

La calidad del producto de software según la norma ISO/IEC 25000 y la norma ISO/IEC 9000 [14], [12] se define como el grado en el que el producto de software incorpora un conjunto de características (funcionalidad, fiabilidad, usabilidad, eficiencia, satisfacción, entre otras), definidas por la industria, de tal manera que se garantiza su eficiencia de uso respecto a los requerimientos de los clientes. Es decir, el grado en que un producto de software cumple con los requisitos y expectativas del cliente.

En la gran mayoría de casos, un proceso de calidad asegura un producto de calidad. Pero, de la misma manera, podemos seguir un proceso de calidad y el producto puede no ser el esperado por el cliente. Es por esto que una organización o empresa debe preocuparse, tanto de la calidad del proceso, como de la calidad del producto final. Y el cliente debe preocuparse de la calidad el producto que obtiene más no de la calidad del proceso de la empresa de quien obtuvo el producto de software.

Para mejorar la calidad del proceso y producto, es necesario conocer el estado actual del proyecto y el rendimiento, identificar los problemas y resolverlos [15].

1.4.2. Experimentación en Ingeniería de Software

Una de las estrategias de la Ingeniería de Software Empírica es el experimento o experimento controlado, el cual, en ingeniería de software es una investigación empírica que manipula un factor o variable del escenario estudiado [10]. Los experimentos son usados cuando se desea tener control total del escenario estudiado o manipular su comportamiento directa, precisa y sistemáticamente [16].

Los experimentos pueden ser orientados a humanos u orientados a la tecnología. En los orientados a humanos, son las personas quienes aplican diferentes procedimientos a los objetos experimentales. Por otro lado, en los orientados a la tecnología, se utilizan diferentes herramientas sobre los objetos experimentales [16].

El enfoque científico es usado en diferentes campos de conocimiento como: Biología, Física, Ingeniería Física Nuclear, Biomedicina Informática y Ciencias de la Tierra; son dependientes de procesos de experimentación usando infraestructura computarizada [17]. Los procesos de experimentación son necesarios para poder comprobar, mediante la combinación de datos y variables, el comportamiento de un objeto en un ambiente controlado [18]. Si la experimentación usa recursos computarizados bien definidos, esta

puede obtener muchos beneficios para la construcción y evaluación de sistemas y productos de software [17].

En un experimento controlado, los usuarios son divididos randómicamente entre las variantes, de tal manera que los usuarios reciban la misma experiencia en múltiples ocasiones [18]. Un ejemplo puede ser, dos usuarios designados para dos diferentes diseños de la interfaz de un producto.

El propósito de la experimentación en la Ingeniería de Software es introducir a los interesados en estudios empíricos de Ingeniería de Software, utilizando experimentos bajo ambientes controlados [11]. Al igual que otros procedimientos experimentales, se construyen modelos del proceso o producto de software, las hipótesis acerca de este modelo son probadas y la información obtenida se utiliza para refinar la hipótesis planteada o crear nuevas [19].

La experimentación en Ingeniería de Software se convierte cada día en una norma en empresas de software avanzadas [18], donde se mide el impacto observado a lo largo del experimento [18]. Los resultados obtenidos de la medición son usados para realizar una evaluación confiable con los clientes y así priorizar las actividades del desarrollo de productos de software.

La experimentación está presente incluso en el área educativa. Muchos educadores mantienen la idea de realizar proyectos experimentales con clientes reales de la industria de software [20]. Con este tipo de proyectos, los estudiantes ganan experiencia y conocimientos en el área tecnológica, de desarrollo de productos de software, y de trabajo en equipo. La experimentación permite a los estudiantes, aprender aspectos específicos de los principios o técnicas y métodos de ingeniería de software, de manera directa y controlada. De esta manera la enseñanza no queda solo en la teoría, sino también tiene su componente práctico con situaciones reales pero en ambientes controlados.

Según [21], la Ingeniería de Software empírica aborda la consecución de cinco pasos para la elaboración de experimentos:

- **Alcance:** El propósito de esta fase es definir los objetivos del experimento de acuerdo al marco de trabajo definido. El alcance del experimento se establece definiendo sus objetivos. Generalmente se utiliza plantillas para definir los

objetivos, esto asegura que todos los aspectos importantes de un experimento sean definidos antes de la planificación y ejecución del experimento.

- **Planificación:** En esta fase se determina el “cómo” realizar el experimento (diseño experimental), la misma está dividida en siete pasos. La entrada para esta fase es la definición del objetivo para el experimento. Basado en el objetivo, la “selección del contexto” identifica el entorno en el cual se ejecutará el experimento. Luego, en los pasos de “formulación de la hipótesis” y “selección de variables” se identifican las variables independientes y dependientes. A continuación se realiza la “selección de los sujetos experimentales”. Se escoge el “tipo de diseño experimental”, basado en la hipótesis y las variables. Luego la “instrumentación” es preparada para la implementación práctica del experimento. Finalmente la “evaluación de validez” tiene como objetivo comprobar la validez del experimento.
- **Operación:** Es la fase donde se lleva a cabo el experimento, con el fin de recolectar los datos que deben ser analizados. Esta fase consta de tres pasos: “preparación” donde los sujetos experimentales son escogidos y preparados, “ejecución” donde los sujetos experimentales realizan sus tareas y los datos son recolectados, y “validación de datos” donde los datos recolectados son validados.
- **Análisis e interpretación:** En esta fase se debe interpretar los datos del experimento para poder dar conclusiones válidas basadas en los mismos. El primer paso es caracterizarlos usando estadísticas descriptivas. En el segundo paso se excluyen datos falsos o anormales, reduciendo así el conjunto de datos a un conjunto de datos válidos. En el tercer paso, los datos son analizados probando la hipótesis en un nivel dado de importancia.
- **Presentación:** En esta fase se presentan los hallazgos del experimento a diferentes audiencias. Esto se lo realiza mediante la elaboración de un artículo para una conferencia o revista, un informe para la toma de decisiones, un paquete para la replicación del experimento, o como material educativo.

1.4.3. ISO/IEC 29110

Desde el año 2000 *Software Research Associates Inc. (SRA)*, ha intentado promover la estandarización interna del proceso de desarrollo hacia la mejora de la calidad. Sin embargo, los procesos y los productos de trabajo no están realmente estandarizados en cada proyecto. Por lo tanto, la calidad del producto varía [15].

En enero del año 2011 se publicó la norma ISO/IEC 29110, siendo este un modelo de proceso de desarrollo de software [15]. Fue desarrollada por el Comité Técnico Conjunto 1 / Subcomité 7 con sus siglas en inglés JTC1/SC7, de la ISO/IEC. Esta norma lleva por nombre “Ingeniería de Software – Perfiles de Ciclo de Vida en VSEs” [22]. La ISO/IEC 29110 está basada en las normas ISO/IEC 12207, ISO/IEC 15289 y MoProSoft, que detalla aspectos de la organización [4]. Esta norma dicta el desarrollo de una serie de paquetes de despliegue con sus siglas en inglés *DP*. Un *DP* contiene un conjunto de artefactos, como: descripción de procesos, actividades, tareas, pasos, roles, productos, plantillas, listas de verificación, ejemplos, referencias y mapeo a estándares y modelos, y una lista de herramientas [23]. Estos *DP* están definidos como cuatro perfiles ISO/IEC 29110, incluidos dentro de un grupo de perfiles genéricos: Entrada, Básico, Intermedio y Avanzado, que son perfiles genéricos aplicables a las pequeñas entidades que no desarrollan software crítico [24].

La norma define una pequeña entidad como una organización, independientemente de su estado legal [22], conformada por un máximo de 25 personas [23]. La mayoría de las MIPYMES de software en Latinoamérica pertenecen a la categoría de pequeña empresa [25].

La norma ISO/IEC 29110 está compuesta por cinco partes, divididas en tres grupos, como se puede observar en la Figura 1 [26]:

- *Overview*
 - *Part 1 Overview*
- *Profiles*
 - *Part 2 Framework and Taxonomy*
 - *Part 4 Specifications of Very Small Entities Profiles*
- *Guides*
 - *Part 3 Assessment Guide*
 - *Part 5 Management and Engineering Guide*

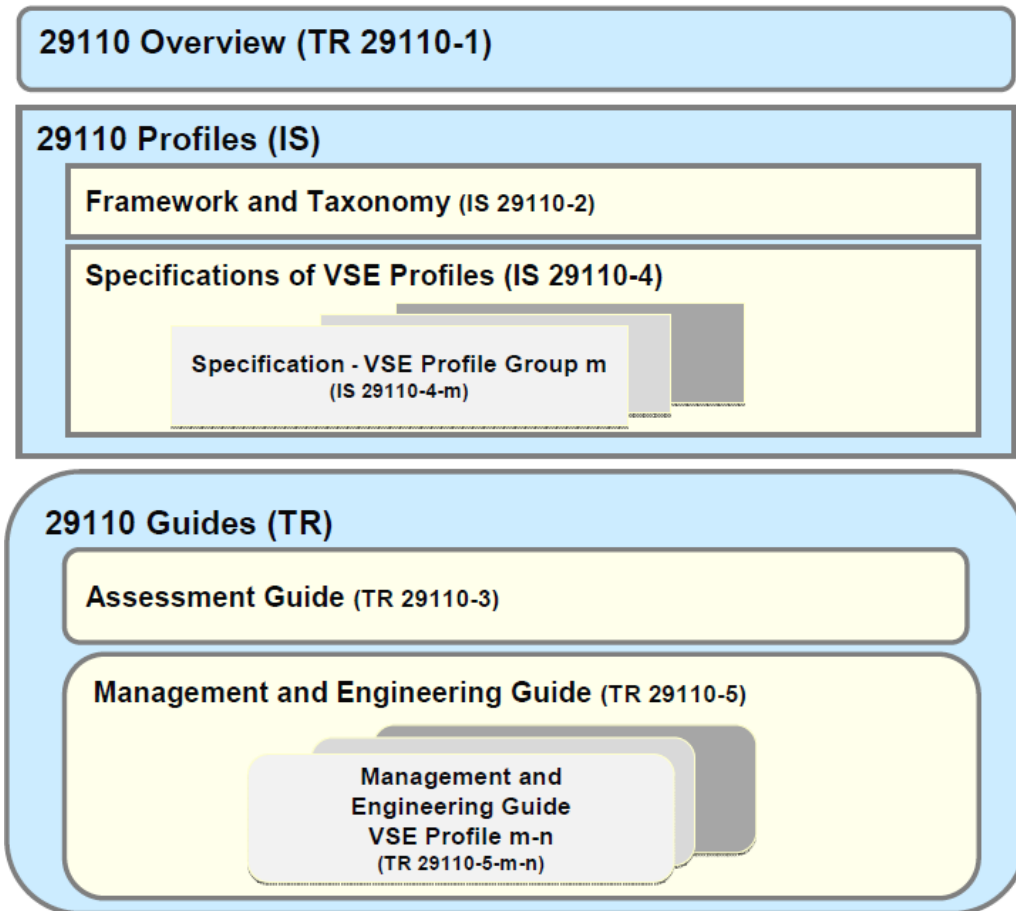


Figura 1 - Series ISO/IEC 29110 [23]

El perfil básico fue publicado en mayo del 2011 y nombrado ISO/IEC 29110-5-1-2, donde el dígito cinco corresponde al número de parte de la norma, el dígito 1 representa al Perfil Genérico, y el dígito 2 corresponde al Perfil Básico ya que antes de este, se encuentra el perfil de entrada [27].

En cuanto a los perfiles, el Perfil de Entrada detalla lo que se espera por una organización pequeña que inicia sus operaciones [4]. En el Perfil Básico se establecen dos procesos principales: Gestión de proyectos e Implementación de Software (ver Figura 2), mientras los otros Perfiles no han sido publicados.

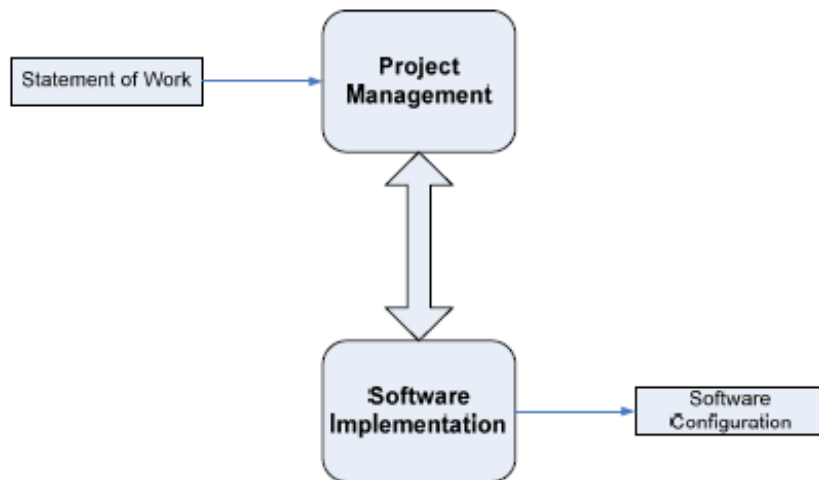


Figura 2 - Guía de procesos para "*Basic Profile*" [23]

Los principales beneficios al aplicar la ISO/IEC 29110, según lo señalado en la misma norma [23], son:

- Mejorar la calidad del software producido por la organización que adopta la norma
- Elevar la capacidad de las organizaciones para ofrecer servicios con calidad y alcanzar niveles internacionales de competitividad
- Permitir la fácil adopción del modelo, pues se brinda mayor detalle en la información
- Facilitar la migración de organizaciones que han implementado modelos como MoProSoft o ISO 9000
- Permitir el posicionamiento de las empresas que lo implementan, destacandolas sobre competidores y en beneficio de sus clientes

El propósito del proceso Gestión de Proyecto, con sus siglas en Inglés PM, es establecer y llevar a cabo las tareas del proyecto de implementación de software, el cual permite cumplir con los objetivos del proyecto en el tiempo, costo y calidad esperados [23]. El proceso PM inicia con el "*Statement of Work*" (ver Figura 2), el cual es proporcionado por el Cliente, el cual permite elaborar el Plan del Proyecto. Las tareas de evaluación y control, comparan el progreso del proyecto con el Plan de Proyecto, tomando acciones para eliminar errores o incorporar cambios al mismo. Para la actividad de cierre de PM, se entrega el producto "*Software Configuration*" (ver Figura 2) resultante del proceso SI, y se debe obtener la aceptación del cliente para poder formalizar el final del proyecto [23].

El propósito del proceso de Implementación de Software, con sus siglas en inglés SI, es el desempeño sistemático de las actividades de análisis, diseño, construcción, integración y pruebas para productos de software nuevos o modificados de acuerdo con los requisitos especificados [23]. La ejecución del proceso SI está impulsado por el Plan del Proyecto. El proceso SI inicia con la actividad “*Software Implementation Initiation*” (ver Figura 3), en la cual se realiza una revisión del plan del proyecto. Este plan guiará las actividades de: *Software Requirements Analysis*, *Software Architectural and Detailed Design*, *Software Construction*, *Software Integration and Tests* y *Product Delivery* [28].

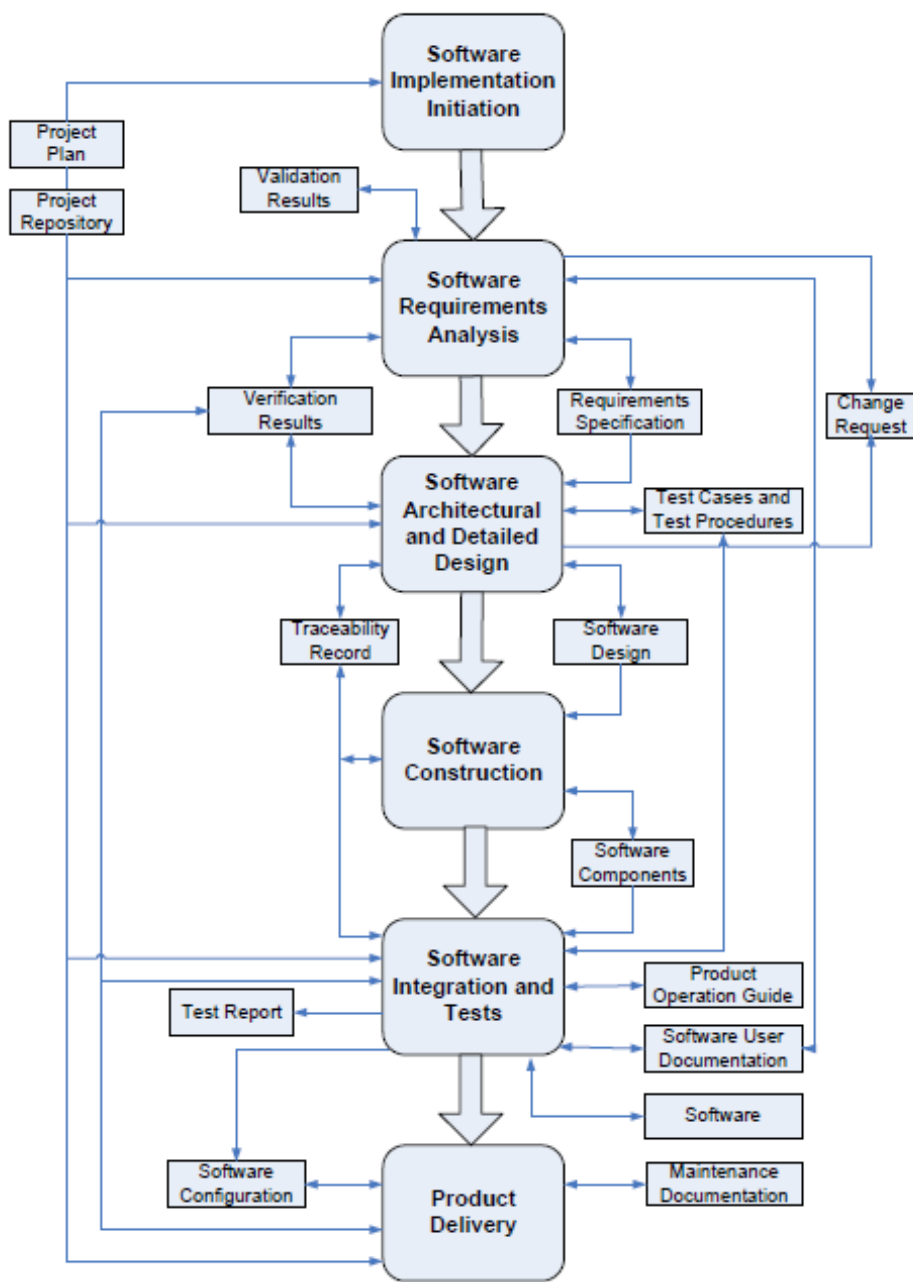


Figura 3 - Diagrama del proceso "Software Implementation" [28]

1.4.4. ISO/IEC 25000

La ISO/IEC 25000 conocida como “*Systems and Software Quality Requirements and Evaluation*” por sus siglas en inglés (SQuaRE), reemplaza a las normas ISO/IEC 9126 “*Software Quality Assessment*” e ISO/IEC 14598 “*Software Product Evaluation*” [29]. El objetivo de la creación de esta norma es cubrir dos procesos importantes: la especificación de requerimientos y la evaluación de la calidad de software, apalancada por un proceso de medición de calidad del producto de software [30]. La norma ISO/IEC 25000 establece criterios para la especificación de requisitos, e incluye un modelo de calidad para unificar las definiciones de calidad de los clientes con los atributos en el proceso de desarrollo [30]. El marco de trabajo del modelo de calidad, descrito en NTE INEN-ISO/IEC 25000, explica la relación entre los diferentes enfoques de calidad (ver Figura 4) [31].

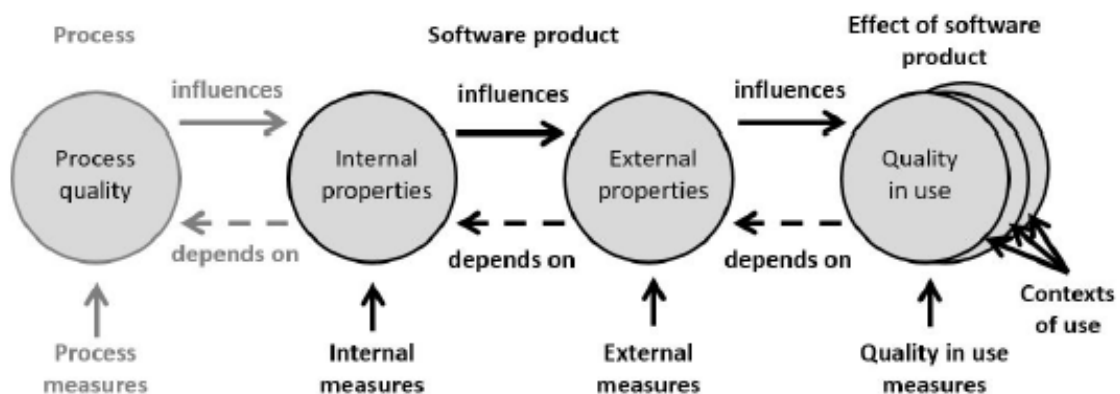


Figura 4 - La calidad y el ciclo de vida [31]

La calidad del producto de software se puede evaluar midiendo las cualidades internas, externas y en uso del software. La calidad en uso depende del comportamiento externo, el cual depende de las propiedades internas del software [29]. Al igual que la norma ISO/IEC 9126, la norma ISO/IEC 25000 define tres conceptos diferenciados en la evaluación de la calidad del producto:

- **Interna:** Se ocupa de las propiedades del software como: el tamaño, la complejidad, las vulnerabilidades de seguridad o la conformidad con las normas de orientación a objetos.
- **Externa:** Analiza el comportamiento del software en producción y estudia sus atributos, por ejemplo, el rendimiento de un software en una máquina determinada, el uso de memoria de un programa, la seguridad del programa, cómo

el programa ayuda al usuario final a entender un fallo o el tiempo de funcionamiento entre fallos.

- **En uso:** Mide la productividad, efectividad y satisfacción del uso del producto de software por parte del usuario final.

1.4.5. Herramienta SonarQube

Esta herramienta es una plataforma de gestión de la calidad de código dedicada a analizar y medir continuamente la calidad técnica, desde el portafolio de proyectos hasta los métodos del proyecto. Esta herramienta nos ayuda a detectar *bugs*, *code smells*, vulnerabilidades de seguridad, *bugs*, deuda técnica, bloques de código duplicado, estándares de codificación, pruebas unitarias; en más de 20 lenguajes de programación [32].

El termino deuda técnica significa que las malas prácticas en el proceso de desarrollo de software provocan una situación de deuda y repercute en un sobrecosto, no solo en el proceso de mantenimiento del producto de software, sino también en su propio funcionamiento [33]. La deuda técnica debe ser tomada en cuenta desde el principio del proyecto, para no empeorar el proyecto, manteniendo una deuda baja [34].

La herramienta SonarQube utiliza el método SQALE para realizar el cálculo de la deuda técnica de un proyecto de software [32]. El método SQALE permite gestionar la deuda técnica de nuestros proyectos de desarrollo de software y conocer qué características de nuestro código acumula más deuda [35].

El método SQALE se basa en ocho principios fundamentales y tiene una estructura que consta de 3 niveles [35]:

- **Nivel 1: Características**.- se definen 8 características de primer nivel, que se deducen del ciclo teórico del código fuente de un programa (codificar, testear, cambiar, entregar, mantener, portar, reutilizar)
- **Nivel 2: Subcaracterísticas**.- cada una de las características del primer nivel se dividen en un conjunto de subcaracterísticas, las cuales se utilizan para agrupar los requisitos permitiendo realizar análisis con varios niveles de abstracción. Existen dos tipos de subcaracterísticas.
 - Las que se corresponden con actividades del ciclo de vida como pruebas unitarias, pruebas de integración, optimización del uso de procesador o del tamaño del código generado.

- Las que resultan de aplicar reglas en términos de buenas o malas prácticas relacionadas con la arquitectura del software y la codificación.
- **Nivel 3: Requisitos.**- Este nivel del modelo contiene todos los requisitos de calidad del código, respetando siempre los criterios presentados en los principios fundamentales. Estos requisitos se refieren a aspectos que existan en el código fuente.

1.4.6. Trabajo relacionado

Existen varios países como Tailandia, México, Canadá, Perú, Colombia; en los cuales sus VSEs han incorporado la norma ISO/IEC 29110 a sus procesos SI [1]. Actualmente falta desarrollar, dentro del perfil genérico, los perfiles Intermedio y Avanzado de la norma [36]. Los trabajos que más se relacionan al presente, son los que se encargan de realizar evaluaciones al producto o proceso software, o los que documentan la aplicación de la norma como guía del desarrollo de un producto de software.

Utilizando como parámetros los mencionados anteriormente, el trabajo [37] realiza una evaluación de un conjunto de aplicaciones web para poder determinar la accesibilidad de las mismas. El resultado de este trabajo es una lista de tareas que deben tomarse en cuenta en cada una de las actividades del proceso SI de la norma ISO/IEC 29110, para poder eliminar las restricciones que se tiene para que un producto de software sea accesible.

El trabajo [38] está relacionado con la aplicación de la norma como guía de desarrollo. En este trabajo, se establece la guía de los dos procesos (PM y SI) de la ISO/IEC 29110-5-1-2 para el desarrollo del producto de software. Haciendo que su implementación sea recomendada por los autores, no solo para VSEs, sino también para grandes organizaciones que deseen realizar pequeños proyectos.

2. METODOLOGÍA

El objetivo de este trabajo fue definir las mejores prácticas para mejorar los procesos de software basadas en la norma ISO/IEC 29110-5-1-2: “Perfil básico en VSEs.

En este apartado se explica con detalle cómo se realizó y diseñó el experimento. Se detallan: los sujetos del experimento y sus roles, el alcance del producto de software a ser realizado, el calendario del experimento, cómo se realizó la evaluación del proceso y sus actividades, y cómo se evaluó el producto de software final.

2.1. Diseño del Experimento

Para alcanzar el objetivo de esta investigación, se propuso usar las técnicas de Ingeniería de Software Empírica que nos permiten realizar experimentos bajo ambientes controlados [11]. Para realizar el experimento, creamos:

- Dos equipos de desarrollo que realicen un proyecto de software aplicando la norma ISO/IEC 29110;
- Y otros dos equipos de desarrollo que realicen el mismo proyecto de software sin aplicar la norma, pero cumpliendo con el mismo alcance.

Los primeros dos equipos serán los encargados de aplicar la norma ISO/IEC 29110, de la cual se utilizará el proceso “*Implementación de Software*” [23], de ahora en adelante los llamaremos “equipos controlados”. Los otros dos equipos tendrán la guía de una metodología de desarrollo y se les proveerá del mismo alcance del producto, ambiente de desarrollo y tiempo del proyecto, de ahora en adelante los llamaremos “equipos no controlados”.

El experimento emula a VSEs latinoamericanas, en donde no se realizan capacitaciones previas a la implementación de algún proceso o estándar. Las capacitaciones se las realizan cuando es estrictamente necesario por cuestiones de presupuesto. Por este motivo, no se realizó la capacitación inicial en la norma, pero a lo largo del experimento ellos se auto-capacitaron para poder entregar los productos de salida especificados en la norma. Siguiendo el mismo concepto no se entregó los DP a los equipos controlados ya que esto provocaría un sesgo en el experimento a favor de los mismos. Por otro lado, a los equipos no controlados se les pidió que escojan una metodología de desarrollo, para

poder comparar si los entregables de la metodología escogida son equiparables a los que pide la norma ISO/IEC 29110-5-1-2.

El proceso “*Project Management*” (ver Figura 2), si bien no fue parte del experimento, fue usado como base para realizar la gestión de proyectos de los equipos controlados. Este fue realizado de forma informal. Cada equipo tenía un repositorio en la nube, en el cuál se debía subir todas las evidencias del trabajo realizado.

2.2. Sujetos Experimentales

En total, 25 estudiantes del curso “SIC734 Calidad de Software” de séptimo nivel de la Carrera de Ingeniería de Sistemas Informáticos y de Computación fueron parte de este experimento. Los estudiantes (5 mujeres y 20 hombres) tenían una edad entre 23 y 27 años, con una edad promedio de 25 años. Previamente, los estudiantes han obtenido conocimientos en Ingeniería de Software, Programación Orientada a Objetos, Metodologías Ágiles, Aplicaciones en Ambientes Libres, Aplicaciones en Ambientes propietarios, Gestión de Proyectos. Los estudiantes son todos de la misma nacionalidad (ecuatoriana), pero no todos están en el mismo nivel de la carrera, unos están aún cursando materias en niveles inferiores.

Los 25 estudiantes fueron repartidos aleatoriamente en tres equipos de 6 integrantes y uno de 7. El equipo #1 (7 integrantes) estuvo conformado por 7 hombres, el equipo #2 (6 integrantes) estuvo conformado por 3 mujeres y 3 hombres, el equipo #3 y el equipo #4 (6 integrantes) estuvieron conformado por 1 mujer y 5 hombres.

De los cuatro equipos enumerados anteriormente, los equipos #1 y #3 son los equipos no controlados, y los equipos #2 y #4 son los equipos controlados. Cada equipo escogió a un representante, el cuál fue el encargado de subir al repositorio todos los productos de salida y las evidencias del trabajo realizado, según lo planificado para cada equipo controlado y no controlado.

Roles

Los roles involucrados en el proceso SI son los descritos en la Tabla 1.

Tabla 1 - Roles involucrados en el proceso SI. [23]

Rol	Abreviación	Miembros del Equipo
Cliente	CUS	Sandra Sánchez-Gordon
Analista	AN	Equipos #2 y #4
Diseñador	DES	Equipos #2 y #4
Programador	PR	Equipos #2 y #4
Gerente del Proyecto	PM	Luis Castillo
Líder Técnico	TL	Luis Castillo
Equipo de Trabajo	WT	Equipos #2 y #4

Modificada de [23]

Todos los integrantes de los equipos cumplen el mismo rol según la actividad del proceso SI en el que se encuentre el ciclo de desarrollo. Por ejemplo, si se encuentran en la actividad “*Software Requirements Analysis*” (ver Figura 3) todos los integrantes cumplen el rol AN.

2.3. Alcance del Producto de Software

Los cuatro equipos tenían el objetivo de desarrollar un producto de software para el agendamiento de citas médicas para el Departamento de Bienestar Estudiantil de la Escuela Politécnica Nacional. Los requerimientos de calidad fueron dados por el rol CUS. En la Tabla 2 se listan los requerimientos identificados. En los Anexos I, II, III y IV se encuentran las capturas de pantalla de cada uno de los productos de software desarrollados por los equipos.

Tabla 2 - Lista de requerimientos identificados del producto de software

ID de la Historia de Usuario	Requerimiento
UHID 1	Ingresar los datos básicos de los médicos
UHID 2	Ingresar los datos básicos de los pacientes
UHID 3	Registrar las especialidades médicas que ofrece el Departamento de Bienestar Estudiantil
UHID 4	Programar una cita con un médico específico
UHID 5	Notificar por correo un día antes de la cita

UHID 6	Mostrar las citas pendientes
UHID 7	Ver el registro de cada cita con el médico
UHID 8	Mostrar las acciones de los médicos
UHID 9	Re-programar una cita con el médico
UHID 10	Imprimir las citas pendientes de cada médico
UHID 11	Opcionalmente, registrar el tratamiento de los pacientes
UHID 12	Las citas de tratamiento son prioridad para el mismo médico que los trata

En la Figura 5 se presenta un ejemplo de historia de usuario, usada para representar los requerimientos. Este formato fue usado por el equipo controlado #2. En el Anexo V se encuentran todas las historias de usuario de este equipo.

Historia de Usuario			
Código	UHID 2	Número de Sprint:	SP1
Nombre	Registrar información general del paciente		
Actor	Recepcionista		
Descripción	Como recepcionista quiero ingresar la información general de un nuevo paciente		
Criterios de aceptación			
Condición		Resultado	
Cuando se registra un paciente		Se debe validar que la cédula ingresada sea válida.	
		Se debe validar que el mail ingresado no pertenezca a otro usuario, y que sea única de una persona.	
		Se debe validar que los campos de cédula, nombre, apellidos, teléfono, e-mail, dirección sean obligatorios.	

Figura 5 – Formato de historia de usuario elaborado por el equipo #2

Calendario

El experimento tuvo una duración de 6 semanas, repartidas para cada una de las actividades del proceso SI (ver Figura 3). Al finalizar cada actividad del proceso SI, los equipos controlados #2 y #4 presentaron los productos de salida pertenecientes a dicha actividad como se puede observar en la Figura 6.

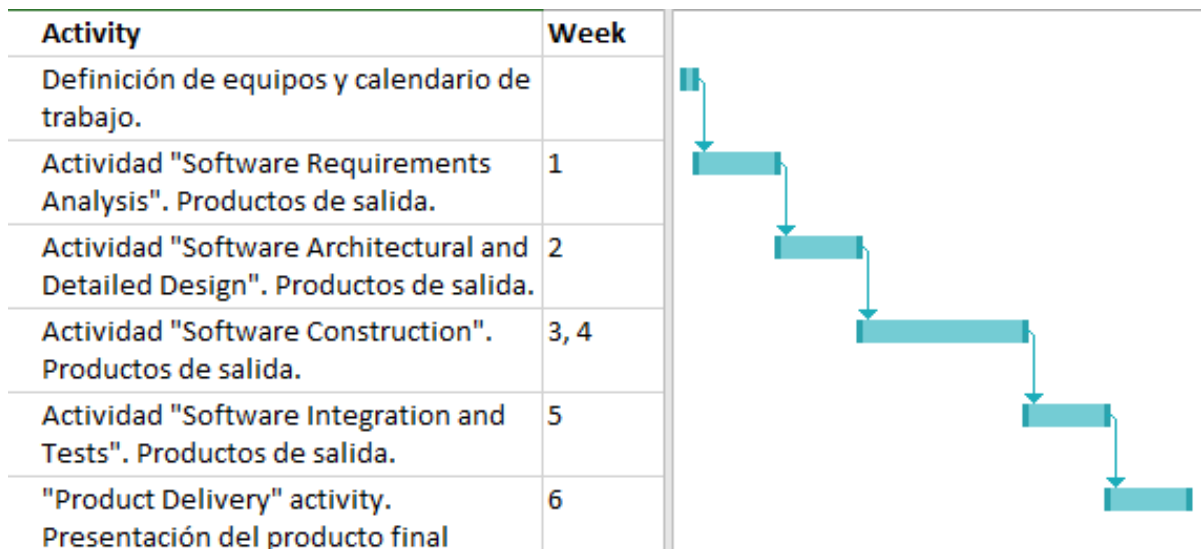


Figura 6 - Diagrama de Gantt del proyecto para los equipos controlados #2 y #4

En cuanto a los equipos no controlados, se les pidió elaborar su propio cronograma del proyecto, el cuál fue subido al repositorio respectivo de cada equipo. En la Figura 7 se presenta el diagrama Gantt del calendario del equipo no controlado #1.

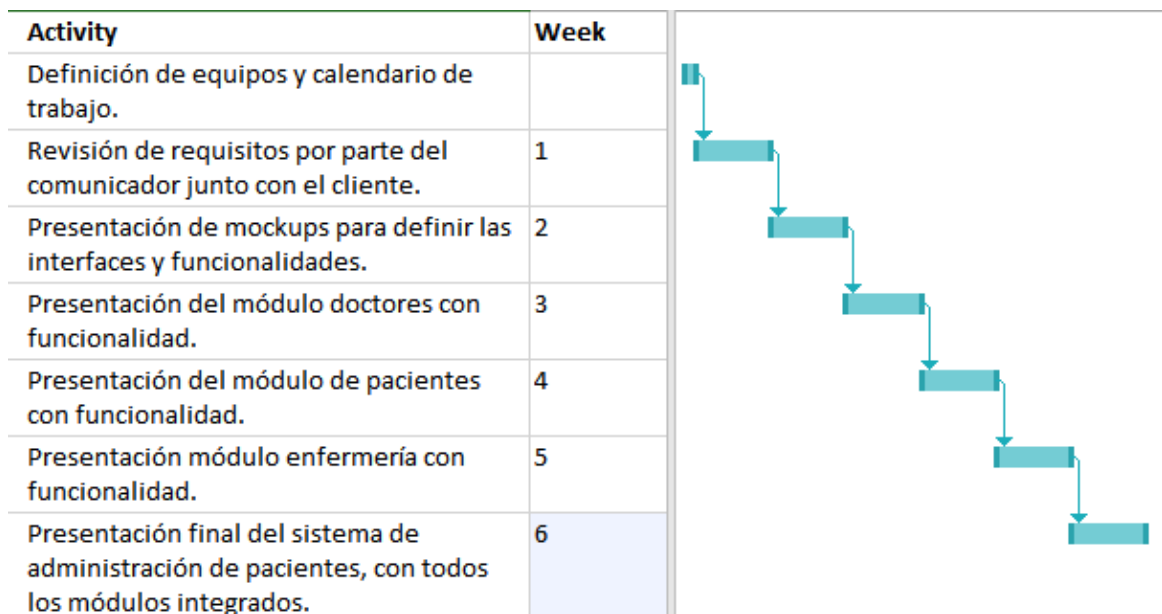


Figura 7 - Diagrama de Gantt del proyecto para el equipo no controlado #1

En la Figura 8 se presenta el diagrama de Gantt del calendario del equipo no controlado #3.

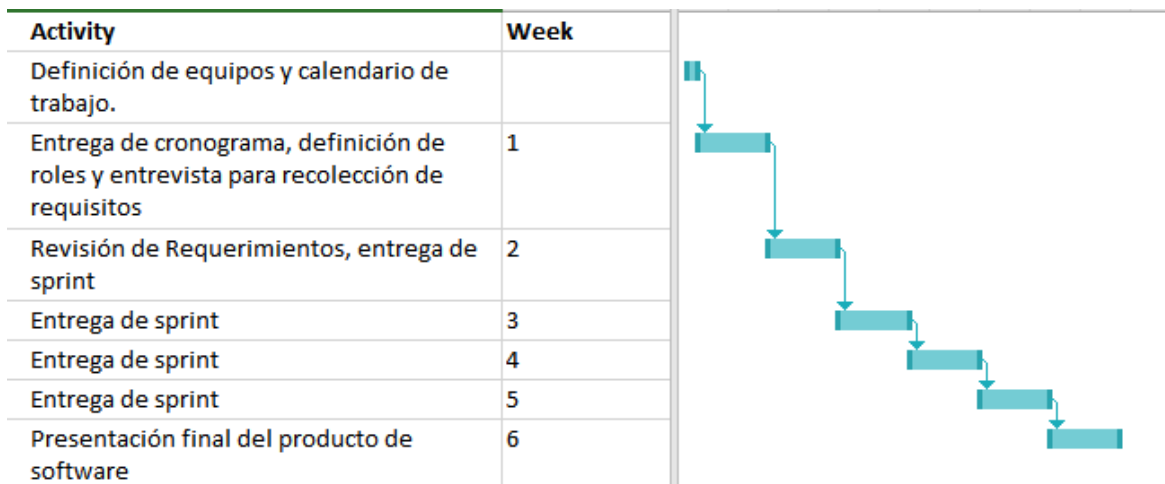


Figura 8 - Diagrama de Gantt del proyecto para el equipo no controlado #3

2.4. Evaluación de la Calidad del Proceso SI

Para la evaluación de los productos de salida al final de cada actividad del proceso SI, de los equipos controlados y no controlados, se utilizó la escala “*Not Achieved – Partially – Largely - Fully* (NPLF)”, tomada de la norma ISO/IEC 33020:2015 “*Information technology -- Process assessment -- Process measurement framework for assessment of process capability*”. Esta calificación está en base a la completitud de cada producto de salida [38]. La escala tiene los rangos de valores definidos en la Tabla 3.

Tabla 3 - Escala de clasificación N-P-L-F [33]

Escala	Significado	Rango
N	Not achieved	0 - 15%
P	Partially	>15% - 50%
L	Largely	>50%- 85%
F	Fully	>85% - 100%

Se creó un esquema de evaluación para cada uno de los productos de salida de cada actividad del proceso SI, con los rangos de calificación antes mencionados. Dicho esquema se presenta en la Tabla 4.

Tabla 4 - Esquema de evaluación para equipos controlados (#2 y #4) y no controlados (#1 y #3)

Activity	Output Products	N - (0 - 15)%	P - (>15-50)%	L - (>50-85)%	F - (>85-100)%
Software Implementation Initiation	Project Plan[reviewed]				
Software Requirements Analysis	Software Configuration				
	Requirements Specification [validated, baselined]				
	Software User Documentation [preliminary, verified, baselined]				
Software Architectural and Detailed Design	Software Configuration				
	Software Design [verified, baselined]				
	Test Cases and Test Procedures [verified]				
	Traceability Record [verified, baselined]				
Software Construction	Software Configuration				
	Software Components [corrected, baselined]				
	Traceability Record [updated baselined]				
Software Integration and Tests	Software Configuration				
	Test Cases and Test Procedures [baselined]				
	Software [tested, baselined]				
	Traceability Record [updated, baselined]				
	Test Report [baselined]				
	Product Operation Guide [verified, baselined]				
Product Delivery	Software User Documentation [verified, baselined]				
	Software Configuration				
	Maintenance Documentation [verified, baselined]				
	Software Configuration [delivered]				

Las evaluaciones realizadas a cada uno de los equipos se tabularon y analizaron para determinar la calidad en cada una de las actividades del proceso SI. De cada equipo se puede calcular una calificación total para el proceso SI, sumando las calificaciones obtenidas por cada producto de salida, de cada actividad del proceso SI; y así determinar los factores que influyeron para que un equipo obtenga una mayor calificación en la calidad de su proceso, que otro.

2.5. Evaluación de la Calidad del Producto de Software

El producto de software final de cada equipo se evaluó a modo de ejemplo para poder obtener más resultados en base a ISO/IEC 25000. Esta norma provee las métricas para la evaluación de la calidad externa, interna y de uso, de un producto de software [25].

Para la evaluación de la calidad en uso se realizó una encuesta de satisfacción a 10 usuarios finales, los cuales probaron los sistemas de los cuatro equipos. La encuesta se basó en la característica de Usabilidad de la norma ISO/IEC 25000 donde cada característica se evaluó con un rango de uno a cinco, siendo uno la más baja y cinco la más alta calificación. La evaluación de la calidad interna se realizó a los productos de software de los cuatro equipos, mediante la herramienta SonarQube. Los parámetros a tomar en cuenta del resultado de la herramienta son: las vulnerabilidades, los *bugs* y la cantidad de *code smells* con el esfuerzo adicional necesario para reparar o re-factorizar el código luego de su desarrollo (deuda técnica) [30] [31]. Y por último la evaluación de la calidad externa se la realizó mediante la matriz que provee la misma norma, tomando en cuenta la evaluación de las siguientes métricas:

- Efectividad de la documentación de usuario o ayuda.
- Claridad de mensajes.
- Verificación de entradas válidas.
- Métodos de autenticación.

La calidad del producto final se determinó con la tabulación y análisis de los resultados obtenidos de la evaluación de la calidad externa, interna y en uso, según la norma ISO/IEC 25000. Determinando que factores influyen en la calificación de calidad de producto.

3. RESULTADOS Y DISCUSIÓN

Después de haber realizado el experimento detallado en la sección anterior, en este apartado se presentan los resultados obtenidos después de haber realizado la evaluación, tanto al proceso SI como al producto de software. El experimento inició el día 24 de abril de 2018 y finalizó el día 05 de junio de 2018. Dentro de este intervalo de tiempo, los equipos controlados #2 y #4 realizaron la entrega de los productos de salida de acuerdo al calendario detallado en la Tabla 5.

Tabla 5 - Calendario del proyecto para los equipos controlados #2 y #4

Semana	Actividad
0	Definición de equipos y calendario de trabajo
1	Actividad " <i>Software Requirements Analysis</i> ". Productos de salida
2	Actividad " <i>Software Architectural and Detailed Design</i> ". Productos de salida
3, 4	Actividad " <i>Software Construction</i> ". Productos de salida
5	Actividad " <i>Software Integration and Tests</i> ". Productos de salida
6	Actividad " <i>Product Delivery</i> ". Presentación del producto final

Como se mencionó anteriormente, los equipos no controlados #1 y #3 establecieron sus propios calendarios en base a la metodología escogida por cada uno. La fecha de inicio y la fecha de presentación del producto final fueron las mismas para equipos controlados y no controlados. En las Tablas 6 y 7, se presentan los calendarios de los equipos #1 y #3 respectivamente.

Tabla 6 - Calendario de proyecto del equipo no controlado #1

Semana	Actividad
0	Definición de equipos y calendario de trabajo
1	Revisión de requisitos por parte del comunicador junto con el cliente
2	Presentación de mockups para definir las interfaces y funcionalidades
3	Presentación del módulo doctores con funcionalidad
4	Presentación del módulo de pacientes con funcionalidad
5	Presentación módulo enfermería con funcionalidad.
6	Presentación final del sistema de administración de pacientes, con todos los módulos integrados

Tabla 7 - Calendario de proyecto del equipo no controlado #3

Semana	Actividad
0	Definición de equipos y calendario de trabajo
1	Entrega de cronograma, definición de roles y entrevista para recolección de requisitos
2	Revisión de Requerimientos, entrega de sprint
3	Entrega de sprint
4	Entrega de sprint
5	Entrega de sprint
6	Presentación final del sistema

Los equipos controlados #2 y #4 usaron únicamente la guía de la norma ISO/IEC 29110 para el desarrollo de sus productos de software respectivos. En cambio, los equipos no controlados #1 y #3 optaron por utilizar la guía de la metodología ágil SCRUM. Los cuatro equipos desarrollaron sus productos de software utilizando el lenguaje de programación JAVA, mediante la herramienta NetBeans. La única diferencia fue que el equipo controlado #4 utilizó JavaFX para el desarrollo de su aplicación en conjunto con la librería JFoenix, para la construcción de las interfaces de usuario.

3.1. Resultados

3.1.1. Resultados de la Evaluación de la Calidad del Proceso SI

La evaluación del proceso SI se lo realizó a la par de su ejecución. A continuación, se presentan los resultados de las evaluaciones realizadas a cada uno de los equipos en cada una de las actividades del proceso SI. La actividad “*Software Implementation Initiation*” para todos los equipos tiene la máxima calificación, ya que se les entregó el plan del proyecto y el acceso al repositorio. Esta actividad no será tomada en cuenta para los resultados del proceso.

La calificación se la realizó en la escala NPLF como se muestra en la Tabla 4. Pero para tabular los resultados, y ya que la escala describe un rango de valores y no uno específico, se determinó un *score* a cada uno de los elementos de la escala, como se muestra en la Tabla 8.

Tabla 8 – Equivalencias entre la escala NPLF y su respectivo *score*

Escala	N	P	L	F
Score	0	1	2	3

Ya con estos valores determinados por cada producto de salida se realizó la sumatoria de los scores cada actividad del proceso. En las Tablas 9, 10,11 y 12; se muestran los scores alcanzados por cada equipo en cada actividad del proceso SI.

Tabla 9 - Scores y sumatoria por cada actividad del proceso SI para el equipo no controlado #1

Equipo 1							
Activity	Output Products	N - (0 - 15)%	P - (>15-50)%	L - (>50-85)%	F - (>85-100)%	Valor	Total
Software Implementation Initiation	Project Plan[reviewed]				X	3	3
Software Requirements Analysis	Software Configuration						
	Requirements Specification [validated, baselined]			X		2	2
	Software User Documentation [preliminary, verified, baselined]	X				0	
Software Architectural and Detailed Design	Software Configuration						
	Software Design [verified, baselined]			X		2	2
	Test Cases and Test Procedures [verified]	X				0	
	Traceability Record [verified, baselined]	X				0	
Software Construction	Software Configuration						
	Software Components [corrected, baselined]			X		2	2
	Traceability Record [updated baselined]	X				0	
Software Integration and Tests	Software Configuration						
	Test Cases and Test Procedures [baselined]	X				0	4
	Software [tested, baselined]		X			1	
	Traceability Record [updated, baselined]	X				0	
	Test Report [baselined]	X				0	
	Product Operation Guide [verified, baselined]	X				0	
	Software User Documentation [verified, baselined]					X	
Product Delivery	Software Configuration						
	Maintenance Documentation [verified, baselined]	X				0	1
	Software Configuration [delivered]		X			1	

Tabla 10 - Scores y sumatoria por cada actividad del proceso SI para el equipo controlado #2

Equipo 2								
Activity	Output Products	N - (0 - 15)%	P - (>15-50)%	L - (>50-85)%	F - (>85-100)%		Valor	Total
Software Implementation Initiation	Project Plan[reviewed]				X		3	3
Software Requirements Analysis	Software Configuration							
	Requirements Specification [validated, baselined]				X		3	3
	Software User Documentation [preliminary, verified, baselined]	X					0	
Software Architectural and Detailed Design	Software Configuration							
	Software Design [verified, baselined]				X		3	3
	Test Cases and Test Procedures [verified]	X					0	
	Traceability Record [verified, baselined]	X					0	
Software Construction	Software Configuration							
	Software Components [corrected, baselined]			X			2	2
	Traceability Record [updated baselined]	X					0	
Software Integration and Tests	Software Configuration							
	Test Cases and Test Procedures [baselined]			X			2	14
	Software [tested, baselined]			X			2	
	Traceability Record [updated, baselined]			X			2	
	Test Report [baselined]			X			2	
	Product Operation Guide [verified, baselined]					X	3	
	Software User Documentation [verified, baselined]					X	3	
Product Delivery	Software Configuration							
	Maintenance Documentation [verified, baselined]				X		3	5
	Software Configuration [delivered]			X			2	

Tabla 11 - Scores y sumatoria por cada actividad del proceso SI para el equipo no controlado #3

Equipo 3								
Activity	Output Products	N - (0 - 15)%	P - (>15-50)%	L - (>50-85)%	F - (>85-100)%		Valor	Total
Software Implementation Initiation	Project Plan[reviewed]				X		3	3
Software Requirements Analysis	Software Configuration							
	Requirements Specification [validated, baselined]			X			2	2
	Software User Documentation [preliminary, verified, baselined]	X					0	
Software Architectural and Detailed Design	Software Configuration							
	Software Design [verified, baselined]			X			2	2
	Test Cases and Test Procedures [verified]	X					0	
	Traceability Record [verified, baselined]	X					0	
Software Construction	Software Configuration							
	Software Components [corrected, baselined]			X			2	2
	Traceability Record [updated baselined]	X					0	
Software Integration and Tests	Software Configuration							
	Test Cases and Test Procedures [baselined]	X					0	6
	Software [tested, baselined]		X				1	
	Traceability Record [updated, baselined]	X					0	
	Test Report [baselined]	X					0	
	Product Operation Guide [verified, baselined]			X			2	
Product Delivery	Software User Documentation [verified, baselined]				X		3	
	Software Configuration							
	Maintenance Documentation [verified, baselined]	X					0	1
	Software Configuration [delivered]		X				1	

Tabla 12 - Scores y sumatoria por cada actividad del proceso SI para el equipo controlado #4

Equipo 4								
Activity	Output Products	N - (0 - 15)%	P - (>15-50)%	L - (>50-85)%	F - (>85-100)%		Valor	Total
Software Implementation Initiation	Project Plan[reviewed]				X		3	3
Software Requirements Analysis	Software Configuration							
	Requirements Specification [validated, baselined]				X		3	3
	Software User Documentation [preliminary, verified, baselined]	X					0	
Software Architectural and Detailed Design	Software Configuration							
	Software Design [verified, baselined]				X		3	3
	Test Cases and Test Procedures [verified]	X					0	
	Traceability Record [verified, baselined]	X					0	
Software Construction	Software Configuration							
	Software Components [corrected, baselined]			X			2	4
	Traceability Record [updated baselined]			X			2	
Software Integration and Tests	Software Configuration							
	Test Cases and Test Procedures [baselined]				X		3	15
	Software [tested, baselined]				X		3	
	Traceability Record [updated, baselined]				X		3	
	Test Report [baselined]				X		3	
	Product Operation Guide [verified, baselined]	X					0	
	Software User Documentation [verified, baselined]				X		3	
Product Delivery	Software Configuration							
	Maintenance Documentation [verified, baselined]				X		3	6
	Software Configuration [delivered]				X		3	

3.1.1.1. Resultados de la Actividad “Software Requirements Analysis”

En las Figuras 9, 10, 11, 12 y 13, se puede evidenciar los *scores* globales por cada uno de los equipos en cada una de las actividades del proceso SI. La línea superior de color tomate representa el valor máximo que se puede obtener en cada una de las actividades.

Como se puede visualizar en la Figura 9, en la actividad “*Software Requirements Analysis*” los equipos no controlados #1 y #3 (barras de color rojo) obtuvieron una misma calificación, dos sobre seis. De igual manera, los equipos controlados #2 y #4 (barras de color azul) obtuvieron una misma calificación, tres sobre seis.

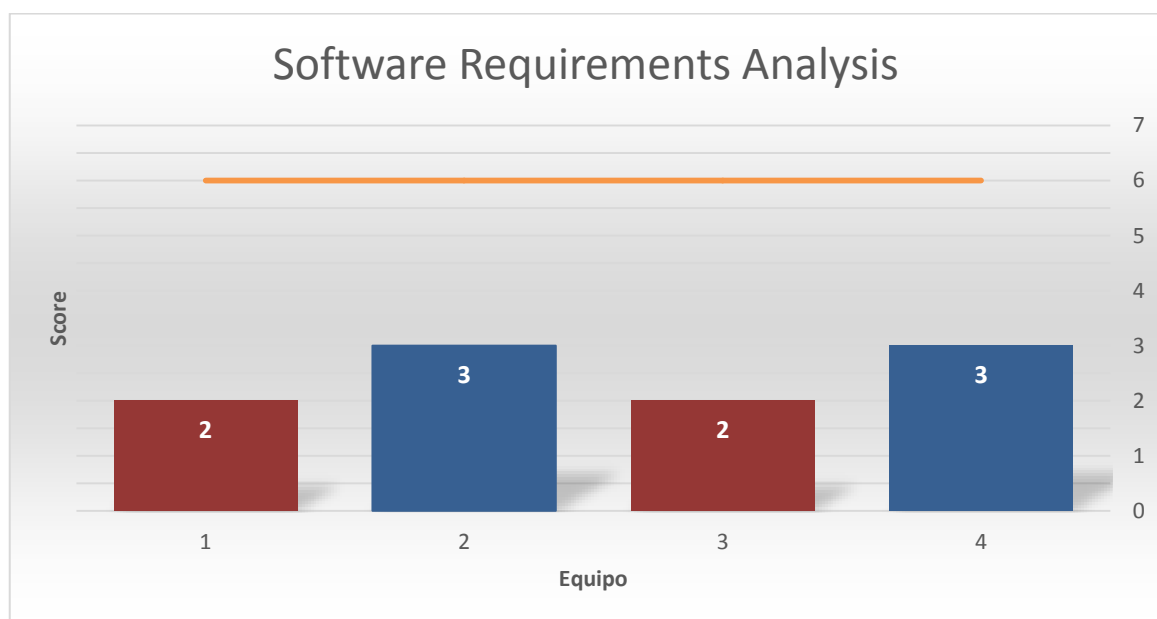


Figura 9 - Scores obtenidos en la actividad "Software Requirements Analysis"

3.1.1.2. Resultados de la Actividad “Software Architectural and Detailed Design”

Como se puede visualizar en la Figura 10, en la actividad “*Software Architectural and Detailed Design*” los equipos no controlados #1 y #3 (barras de color rojo) obtuvieron una misma calificación, dos sobre nueve. De igual manera, los equipos controlados #2 y #4 (barras de color azul) obtuvieron una misma calificación, tres sobre nueve.

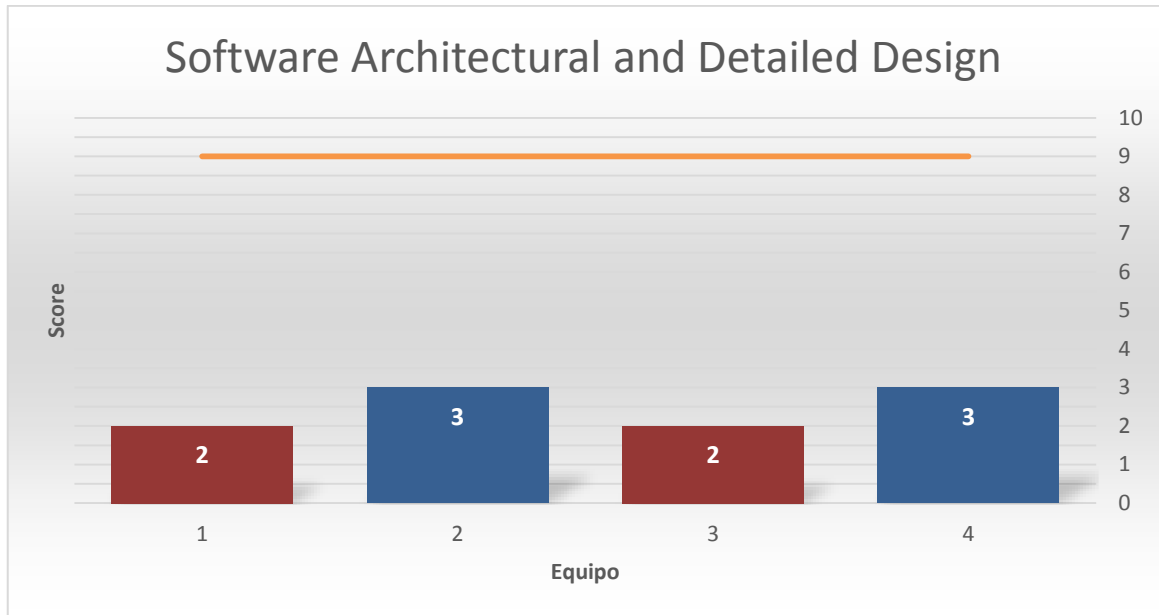


Figura 10 - Scores obtenidos en la actividad "Software Architectural and Detailed Design"

3.1.1.3. Resultados de la Actividad "Software Construction"

En la Figura 11 se puede evidenciar que existe un gran cambio en comparación con las anteriores dos actividades. En esta actividad, el equipo #4 obtuvo una calificación de cuatro sobre seis, mientras que los equipos #1, #2, #3 alcanzaron una calificación dos sobre seis.

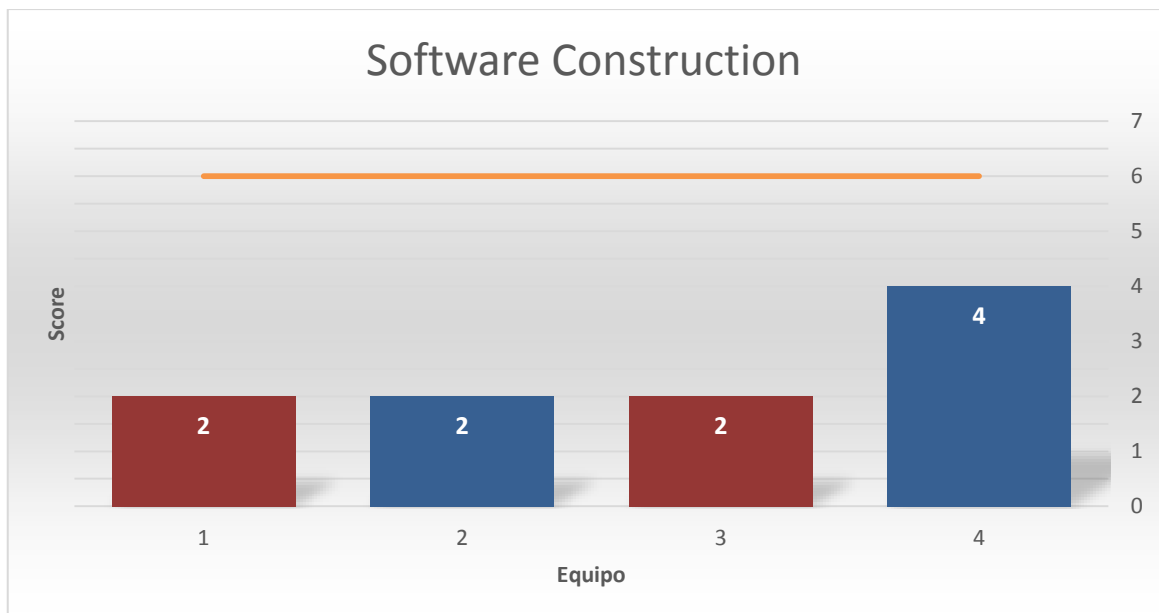


Figura 11 - Scores obtenidos en la actividad "Software Construction"

3.1.1.4. Resultados de la Actividad “Software Integration and Tests”

En la Figura 12 se puede visualizar que el equipo #4 es quien obtuvo una calificación de 15 sobre 18, seguido por el equipo #2 con 14 sobre 18, a continuación, el equipo #3 con seis sobre 18 y por último el equipo #1 con cuatro sobre 18.

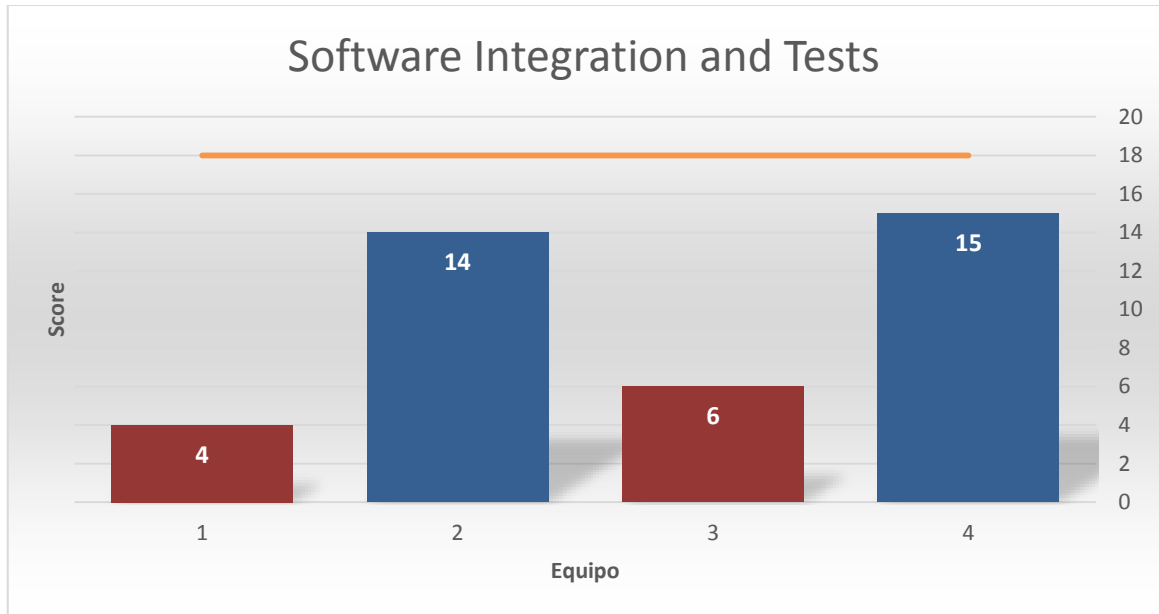


Figura 12 - Scores obtenidos en la actividad "Software Integration and Tests"

3.1.1.5. Resultados de la Actividad “Product Delivery”

La Figura 13 nos muestra los scores de la última actividad del proceso SI. Aquí se puede ver la gran diferencia que se tiene entre los equipos controlados #2 y #4 (barras de color azul) y los no controlados #1 y #3 (barras de color rojo). Mientras el equipo #4 obtuvo una calificación de seis sobre seis y el equipo #2 una de cinco sobre seis, los equipos #1 y #3 obtuvieron una calificación de uno sobre seis.

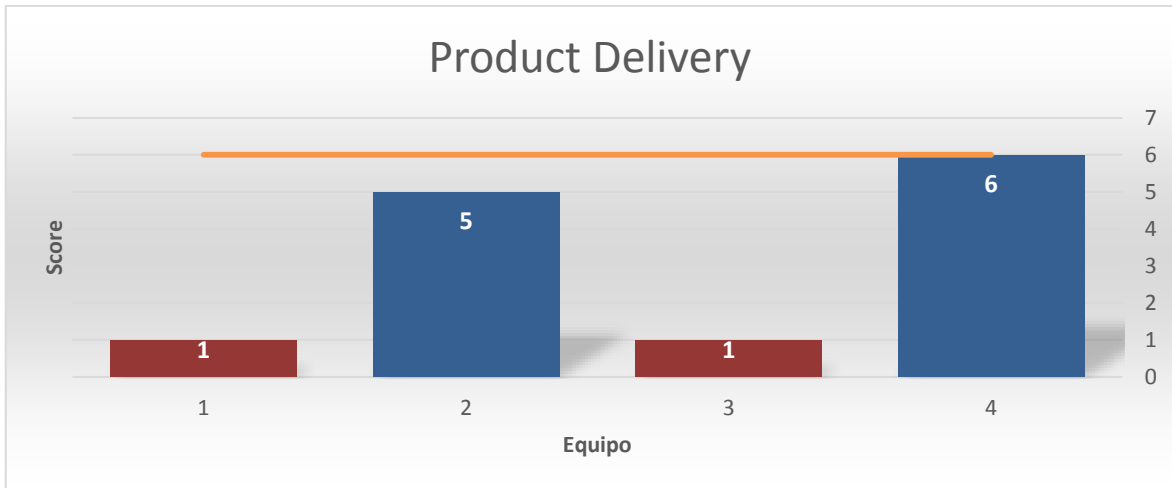


Figura 13 - Scores obtenidos en la actividad "Product Delivery"

La Figura 14 muestra los puntajes normalizados alcanzados por todos los equipos en cada una de las actividades del proceso de SI, y la tendencia de desempeño que cada uno de los equipos tiene a lo largo del proceso. La Figura 15 muestra que el rendimiento de los equipos controlados fue superior desde el inicio del proceso en comparación con los equipos no controlados. Al final del proceso, los equipos controlados alcanzaron una mejor calidad en su proceso. Los números en el eje de actividades corresponden a cada actividad del proceso SI, a excepción de la actividad "Iniciación de implementación de software" que no está incluida. La actividad 1 de la figura es la actividad "Análisis de requisitos de software" y así sucesivamente.

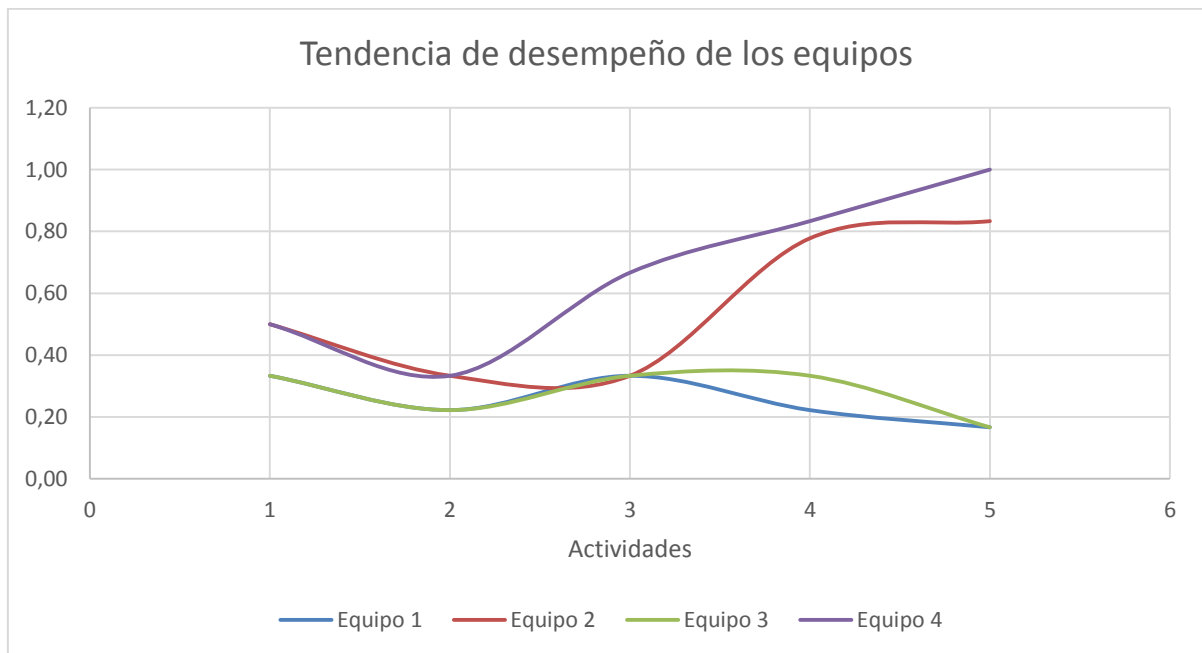


Figura 14 – Tendencia de desempeño de los equipos

3.1.2. Resultados de la Evaluación de la Calidad del Producto

Los resultados de calidad del producto final se presentan de la siguiente manera: primero se presentan los resultados de calidad interna, siguiendo con los resultados de calidad externa, y por último se presentan los resultados de calidad en uso.

3.1.2.1. Resultados de Calidad Interna

La calidad interna fue medida mediante la herramienta SonarQube, para cada uno de los proyectos dando como resultados los mostrados en las Figuras 15, 16, 17 y 18.

En la Figura 15 la herramienta muestra que el equipo no controlado #1 tiene: 20 *bugs* con una calificación "E", 15 vulnerabilidades con una calificación "D", 21 días de deuda técnica con una calificación "B", 781 *code smells*, y 16.5% de código duplicado.

En la Figura 16 la herramienta muestra que el equipo controlado #2 tiene: 1 *bug* con una calificación "C", 32 vulnerabilidades con una calificación "B", 38 días de deuda técnica con una calificación "B", 1.900 *code smells*, y 49% de código duplicado.

En la Figura 17 la herramienta muestra que el equipo no controlado #3 tiene: 5 *bugs* con una calificación "E", 37 vulnerabilidades con una calificación "B", 38 días de deuda técnica con una calificación "B", 1.900 *code smells*, y 36.7% de código duplicado.

En la Figura 18 la herramienta muestra que el equipo controlado #4 tiene: 36 *bugs* con una calificación "E", 4 vulnerabilidades con una calificación "B", tres días de deuda técnica con una calificación "A", 177 *code smells*, y 19.9% de código duplicado.

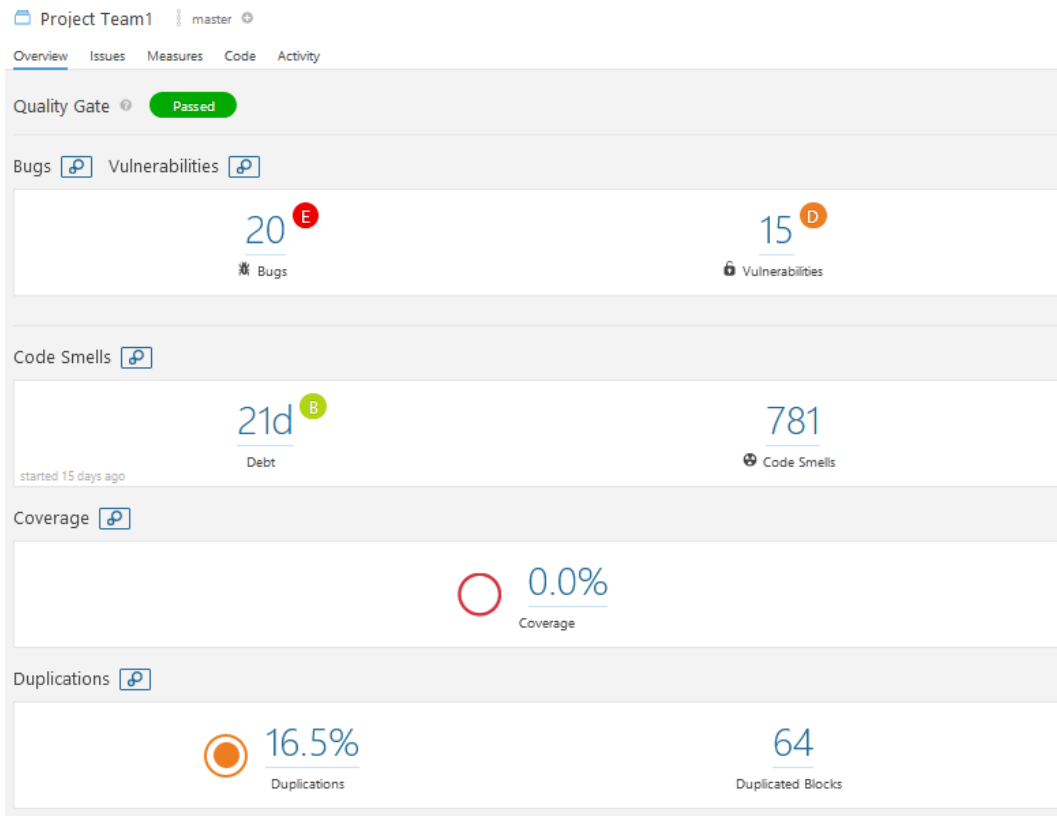


Figura 15 - Resultados de calidad interna del equipo no controlado #1

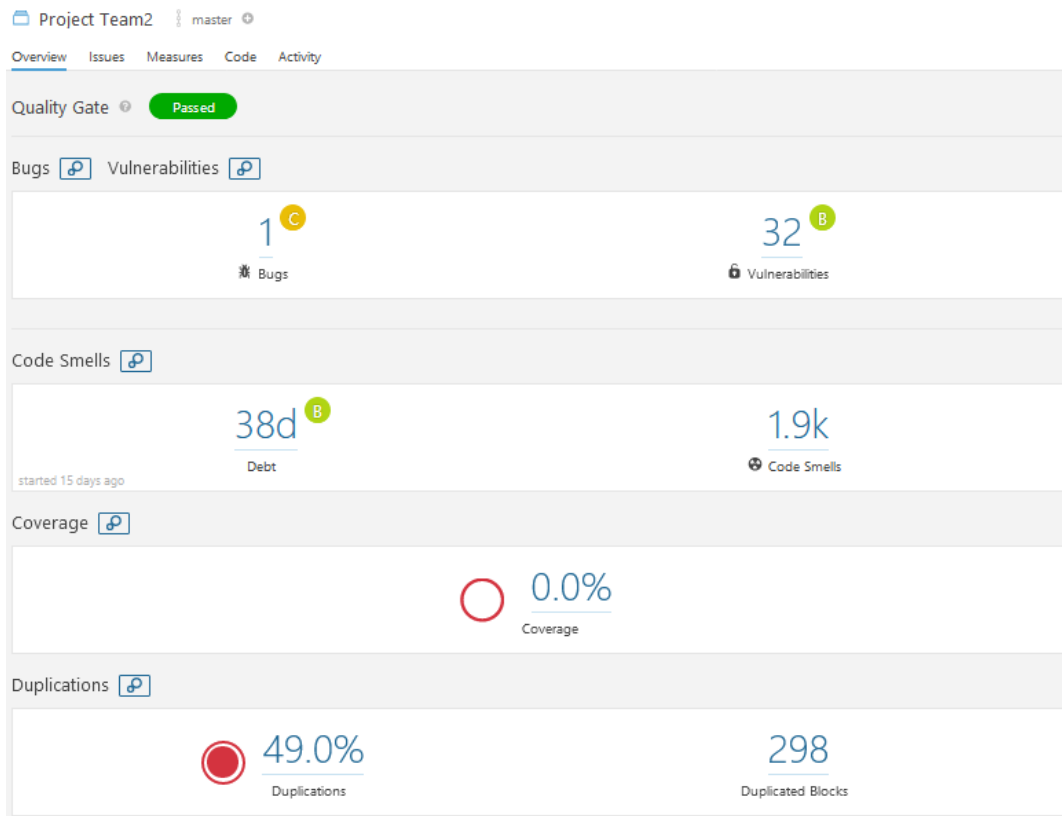


Figura 16 - Resultados de calidad interna del equipo controlado #2

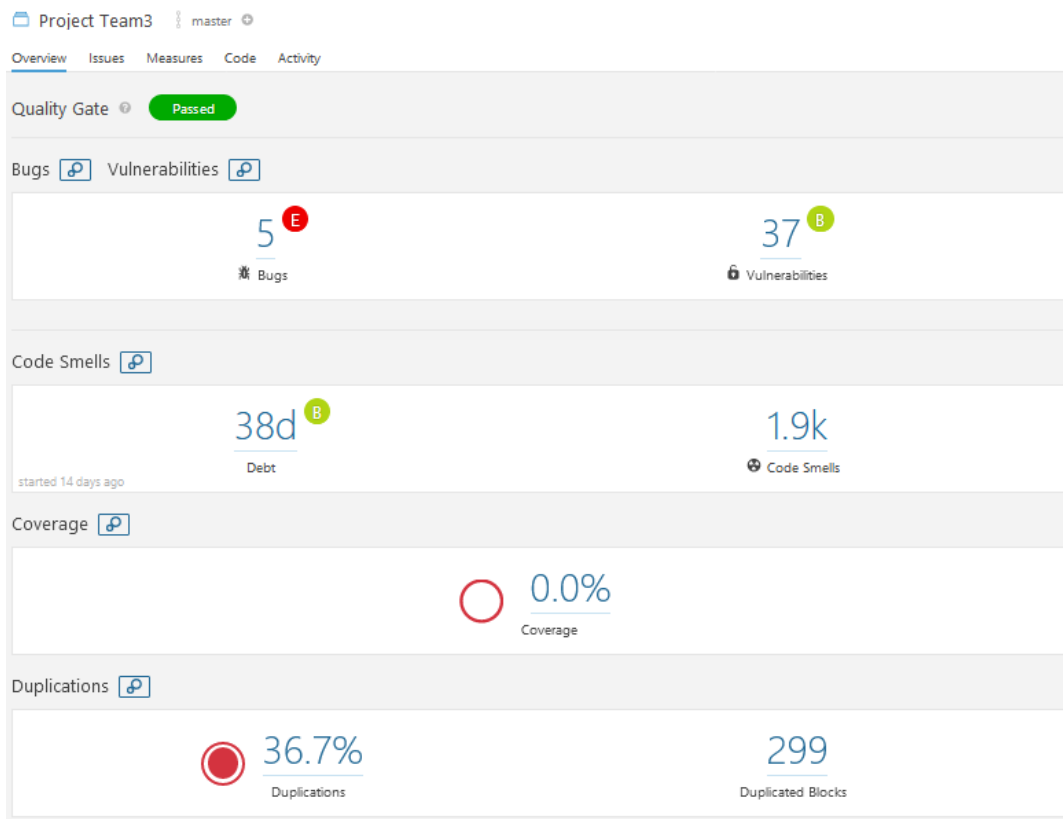


Figura 17 - Resultados de Calidad Interna del equipo no controlado #3

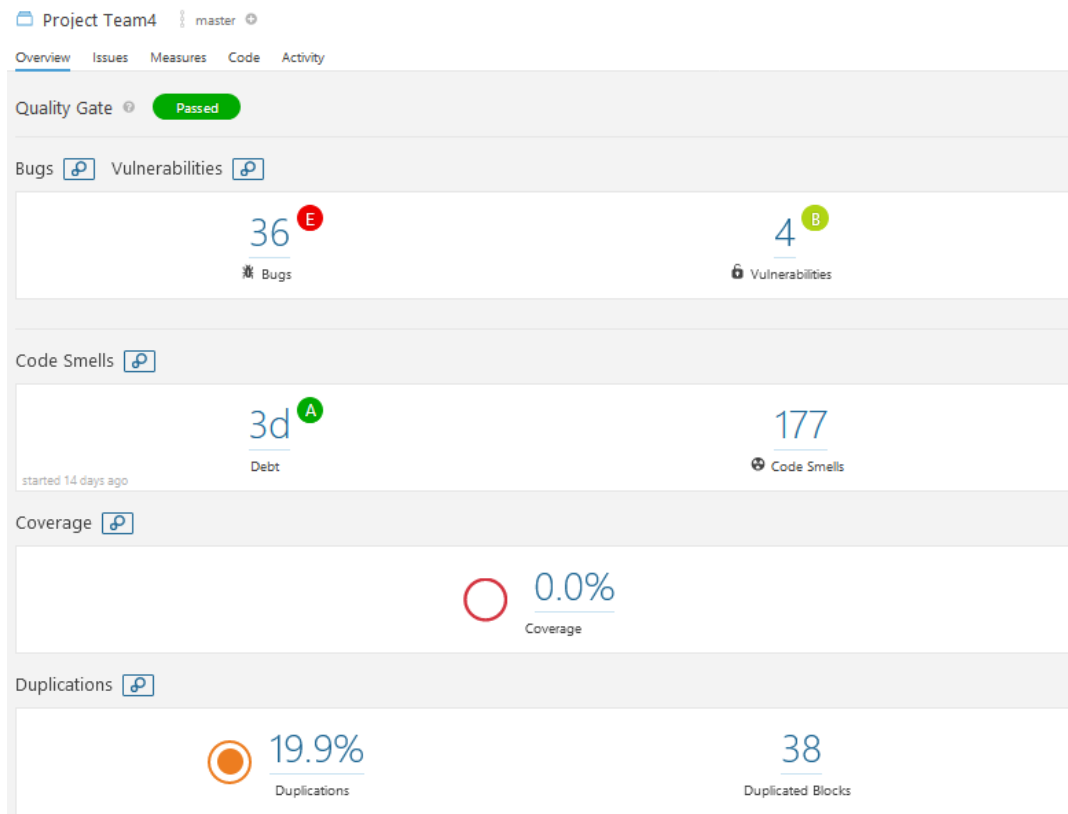


Figura 18 - Resultados de calidad interna del equipo controlado #4

La Figura 19 muestra un resumen de los resultados de la evaluación de calidad interna para los productos de software desarrollados por los equipos, en cuatro categorías: *bugs*, *code smells*, *vulnerabilities*, *technical debt* y *percentage of duplication code*. La categoría *code smells* tenía valores muy altos. Por esta razón, se decidió dividir estos valores por 100 para representarlos.

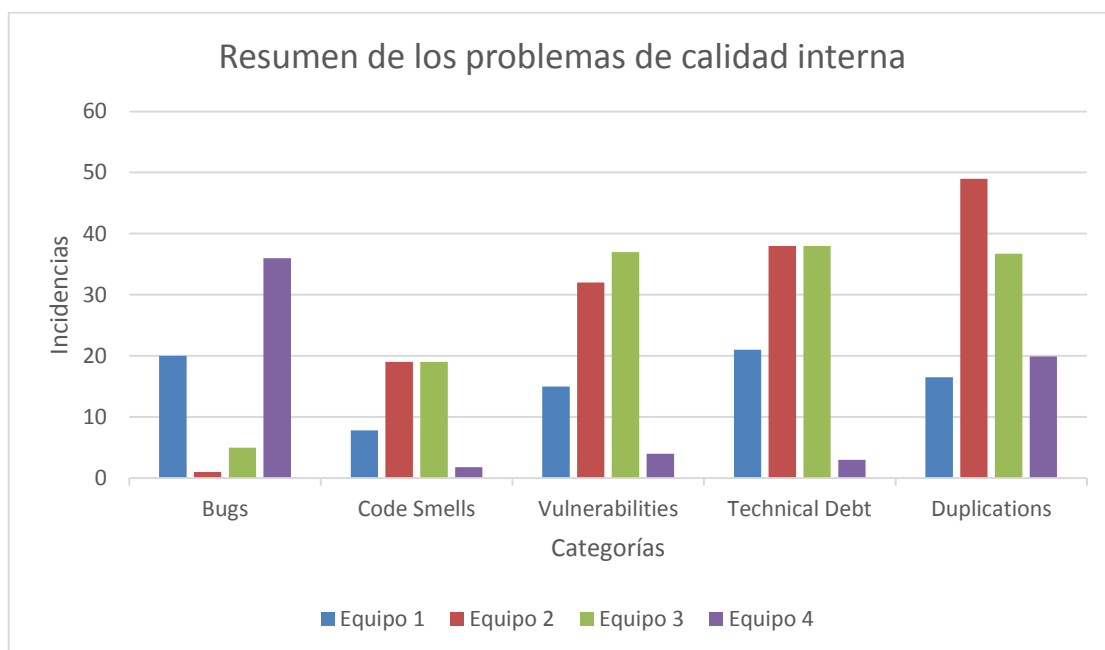


Figura 19 - Resumen de los problemas de calidad interna

3.1.2.2. Resultados de Calidad Externa

La calidad externa fue medida por cuatro métricas que provee la norma ISO/IEC 25000. Estas métricas fueron elegidas de las características “Facilidad de Uso” y “Seguridad”, y son: Efectividad de la documentación del usuario o ayuda del sistema, Claridad de mensajes, Verificación de entradas válidas y Métodos de autenticación. Los resultados de estas evaluaciones se muestran en las Figuras 20 y 21.

En la Figura 20, se puede evidenciar que en la métrica “Efectividad de la documentación del usuario o ayuda del sistema” los equipos #1 y #2 obtuvieron una calificación de 0,86 sobre uno, seguidos por el equipo #4 con 0,77 sobre uno y por último el equipo #3 con 0,58 sobre uno. Adicionalmente, respecto a la métrica “Claridad de mensajes” el equipo #4 alcanzó la calificación más alta (uno), seguido del equipo #1 con 0,91 sobre uno, luego está el equipo #3 con 0,68 sobre uno, y por último el equipo #2 con 0,63 sobre uno.

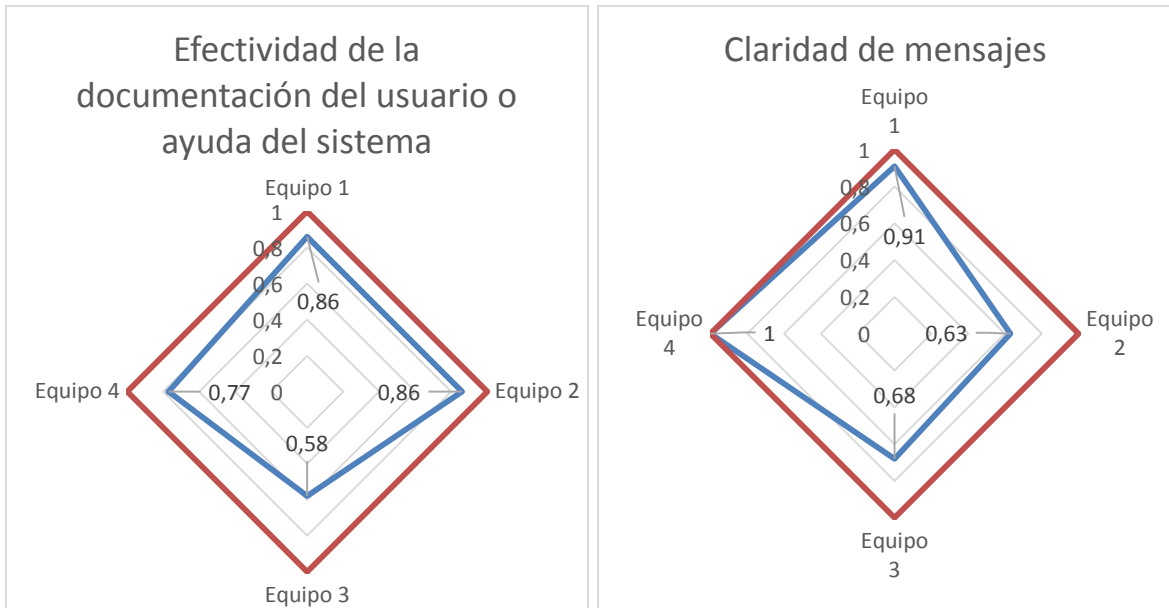


Figura 20 - Resultados de las métricas de calidad externa "Efectividad de la documentación del usuario o ayuda del sistema" y "Claridad de mensajes"

En la Figura 21, se puede evidenciar que para la métrica "Verificación de entradas válidas" los equipos #2 y #3 alcanzaron la calificación más alta (uno), seguido del equipo #1 con 0,75 sobre uno, y por último el equipo #4 con 0,06 sobre uno. Adicionalmente, se puede evidenciar que en la métrica "Métodos de autenticación" todos los equipos alcanzaron la calificación de 0,33 sobre uno.

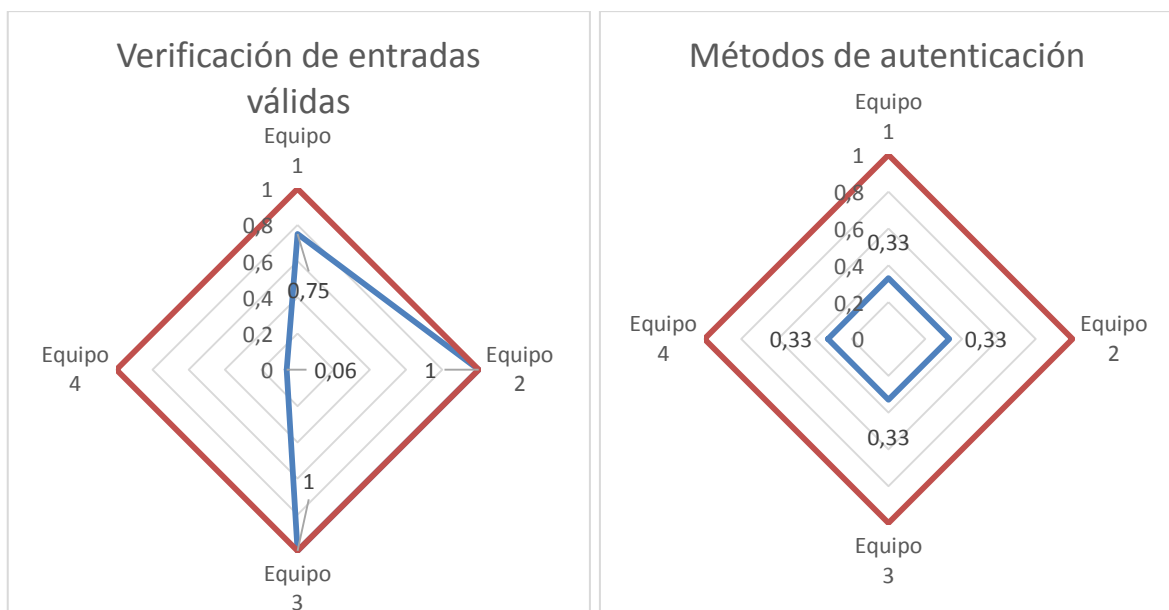
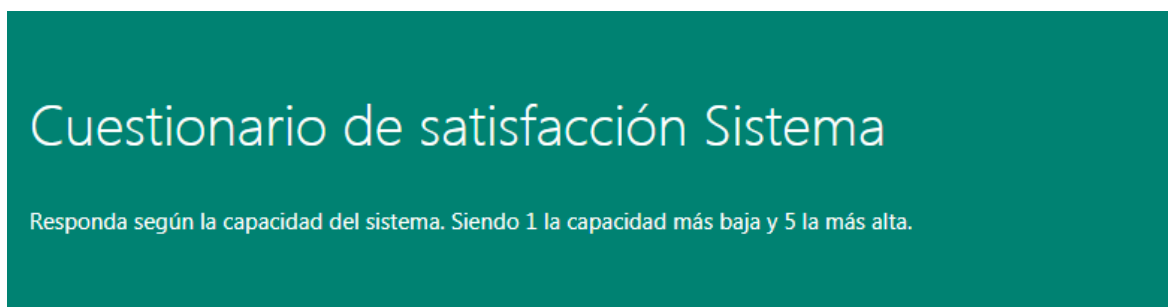


Figura 21 - Resultados de las métricas de calidad externa "Verificación de entradas válidas" y "Métodos de autenticación"

3.1.2.3. Resultados de Calidad en Uso

Se realizó una encuesta de satisfacción a 10 usuarios finales, quienes usaron cada uno de los productos de software implementados por los cuatro equipos. Las Figuras 22 y 23 muestran el cuestionario utilizado.

De cada pregunta se calculó el promedio entre las 10 calificaciones, luego se calculó el promedio de las seis preguntas del cuestionario para obtener la calificación total de cada producto de software. En la Figura 24 podemos comparar la calificación total obtenida para cada producto de software.



* Obligatorio

1. Que capacidad tiene el sistema, de ser adecuado para cubrir las necesidades de agendamiento de citas para un centro médico. *

1 2 3 4 5

2. Que capacidad tiene el sistema para que el usuario pueda operarlo y controlarlo con facilidad *

1 2 3 4 5

3. Que capacidad tiene el sistema para que le usuario aprenda de su manejo *

1 2 3 4 5

Figura 22 - Encuesta de satisfacción. Preguntas 1 - 3

4. Que capacidad tiene el sistema para proteger a los usuarios de cometer errores. *

1 2 3 4 5

5. La interfaz de usuario tiene la capacidad de agrandar y satisfacer la interacción con el usuario. *

1 2 3 4 5

6. Que capacidad tiene el sistema para permitir ser utilizado por usuarios con determinadas características y discapacidades. *

1 2 3 4 5

Enviar

Figura 23 - Encuesta de satisfacción. Preguntas 4 - 6

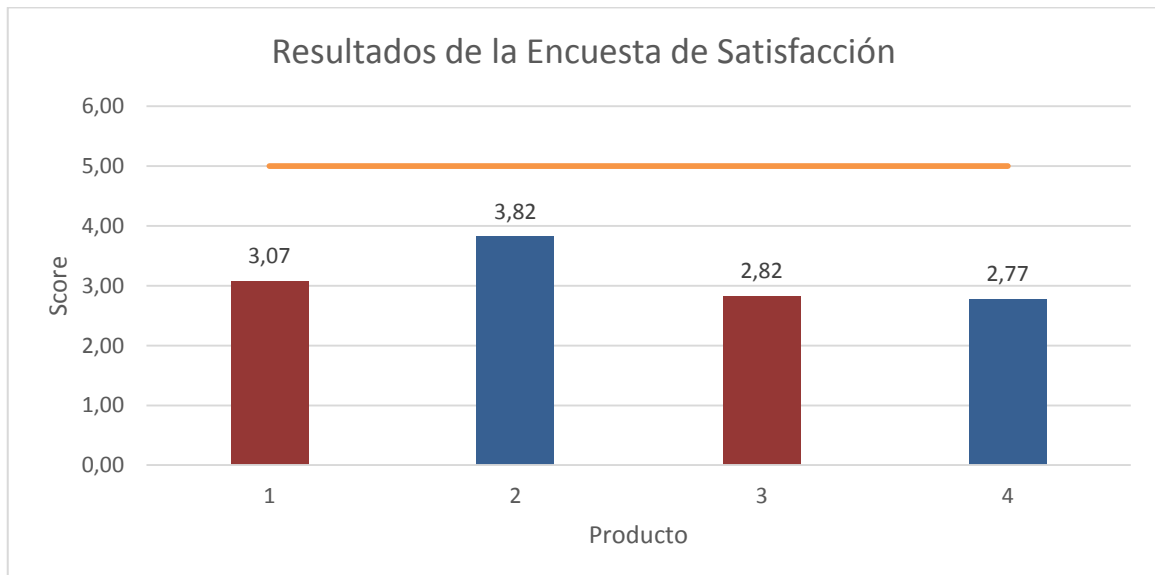


Figura 24 - Resultados de la encuesta de satisfacción para la calidad en uso

En esta encuesta de satisfacción el equipo controlado #2 está en primer lugar con una puntuación de 3,82 sobre cinco, seguido del equipo no controlado #1 con 3,07 sobre

cinco. A continuación, le sigue el equipo no controlado #3 con una puntuación de 2,82 sobre cinco, por último el equipo controlado #4 con una puntuación de 2,77 sobre cinco.

3.2. Discusión

Se realizará el análisis y discusión de la evaluación del proceso, la evaluación del producto y finalmente se expondrán un conjunto de prácticas para la mejora del proceso SI.

3.2.1. Evaluación del Proceso

Luego de haber analizado y tabulado los resultados del experimento podemos realizar la evaluación del mismo. Este análisis lo realizaremos a cada una de las actividades del proceso SI.

En los resultados de la actividad "*Software Requirements Analysis*" (ver Figura 14) podemos evidenciar dos situaciones importantes. La primera es que ningún equipo alcanza el score máximo (seis), y la segunda es que los dos equipos controlados obtuvieron el mismo score (tres) entre ellos, de la misma manera que los equipos no controlados (dos).

Este score bajo en los equipos controlados se debe a que no se tuvo una capacitación previa de la norma ISO/IEC 29110, por lo que no pusieron atención en realizar los documentos que la norma dicta en esta actividad del proceso SI. Por lo tanto, los equipos controlados iban capacitándose en la norma mientras estaba en ejecución el experimento. Por otro lado, los equipos no controlados tuvieron un score bajo por la falta de cumplimiento de los cronogramas determinados por ellos y además la falta de conocimiento de los productos de salida que deben realizar y que la norma pide.

Los resultados de la actividad "*Software Architectural and Detailed Design*", como se puede observar en la Figura 15, son los mismos que la actividad anterior. Además, se puede evidenciar las mismas dos situaciones que en la actividad anterior, con la diferencia del score máximo que en este caso es de nueve.

El score bajo en los equipos controlados se debe a la falta de capacitación en la norma ISO/IEC 29110 como pasó en la actividad anterior. En esta actividad debían entregar un producto de salida más que en la anterior actividad. Los equipos no controlados desconocían cuales eran los productos de salida necesarios para esta actividad.

En la actividad “*Software Construction*” podemos observar que ambos equipos no controlados (#1 y #3) y un equipo controlado (#2) tienen el mismo score (dos), la tercera parte del máximo (seis). Este score bajo, obtenido por parte del equipo controlado #2 se debe a que este equipo reutilizó código libre del Internet. El código reutilizado no era mantenible y el equipo no supo adaptarlo a los requerimientos del experimento, teniendo así funcionalidades extra que no fueron pedidas y funcionalidades incompletas o que no se ejecutaban.

El equipo controlado #4 fue el que más cerca estuvo de alcanzar el máximo score de esta actividad, por lo que se puede constatar que ya estaban más familiarizados con la norma y los productos de salida que en cada actividad se debe desarrollar.

Los resultados de la evaluación en la actividad “*Software Integration and Tests*” ya son más diferenciados entre equipos controlados y no controlados. Mientras los equipos controlados se acercan al máximo score (15 y 14 de 18), los equipos no controlados mantienen su score muy bajo incluso por debajo de la tercera parte del score máximo. En esta evaluación se puede evidenciar la gran diferencia que se tiene, entre guiarse por una norma, y hacer un proceso de desarrollo sin tener en cuenta aspectos importantes para el mismo.

Los equipos controlados ya tienen un conocimiento más amplio de la norma ISO/IEC 29110 y saben los productos de salida que se deben generar en cada actividad, además de que se igualaron con los productos de salida que no tenían realizados, haciendo que su score sea mayor y más cercano al máximo. Por otro lado, los equipos no controlados tenían escasos productos de salida al no seguir la norma, por lo que su score sigue su tendencia baja, haciendo que su proceso no sea de calidad.

La actividad “*Product Delivery*” tiene la misma tendencia de la anterior actividad. Los equipos controlados toman la superioridad en el score e incluso el equipo #4 alcanza el máximo score, seguido del equipo #2 solo una unidad más abajo del máximo score. Ambos equipos no controlados quedan muy abajo con solo una unidad de score.

Los equipos controlados en esta actividad alcanzaron un score superior por su compromiso de seguir la norma, además se igualaron en los productos de salida y entregaron el “*Software Configuration*” completo. Los equipos no controlados, al no tener la guía de la norma tuvieron escasos productos de salida e hicieron una implementación muy pobre en cuanto a documentación y funcionalidad, esto por no organizar el tiempo de mejor manera y además no respetar el cronograma impuesto por ellos mismo.

3.2.2. Evaluación del Producto

Los resultados de la evaluación a la calidad del producto fueron muy diferentes a los resultados del proceso.

En cuanto a evaluación de la calidad interna, realizada en la herramienta SonarQube, los resultados nos demuestran que el sistema con mayores vulnerabilidades es el del equipo #3 (37 vulnerabilidades) y el de menos vulnerabilidades es el del equipo #4 (4 vulnerabilidades). A pesar que el equipo #1 tiene solo 15 vulnerabilidades, su calificación es "D" esto a causa de que sus vulnerabilidades son más peligrosas que el del resto de equipos que obtienen una calificación "B". En cuanto a "Bugs", el equipo #2 tiene uno solo con una calificación "C", mientras que el resto de equipos tienen una calificación "E" (la peor calificación). Aunque el equipo #3 tiene solo cinco "Bugs" también obtiene la calificación "E", por lo críticos que estos son. La deuda técnica, es menor para el equipo #4 con tres días y con una calificación "A" (la más alta), el resto de equipos tiene una deuda técnica sobre los 20 días, pero menor a los 40 días y una calificación "B", lo que nos demuestra que sus implementaciones no son difíciles de reparar o re-factorizar.

Por otra parte, los resultados de la calidad externa nos indican que, en la métrica "Efectividad de la documentación del usuario o ayuda del sistema" de la norma ISO/IEC 25000 para calidad externa, el equipo #3 obtuvo el más bajo score mientras el resto de los equipos se mantienen más cerca del score máximo. Esto es debido a que el equipo #3 describió de forma correcta en su documentación la mayoría de las funcionalidades implementadas. En la métrica "Claridad de Mensajes" el equipo #4 alcanzó el máximo score por las explicaciones clara en todos sus mensajes, el equipo #1 se acerca al máximo score seguido de los equipos #3 y #2 los cuales tuvieron inconvenientes al poder mostrar mensajes claros a los usuarios finales. En la métrica "Verificación de entradas válidas" el equipo #4 obtiene un score muy bajo para la calidad externa de su producto, esto por no haber realizado las validaciones para los campos que necesitan ser validados y así poder evitar errores de los usuarios finales. Por otro lado, los equipos #2 y #3 obtienen el máximo score ya que sus validaciones fueron eficientes. En la métrica "Métodos de Autenticación" todos los equipos obtuvieron el mismo score ya que del número deseado de autenticaciones (tres) que dicta la norma, los cuatro equipos implementaron la tercera parte (uno).

Los resultados de las encuestas de satisfacción respondidas por los usuarios finales nos reflejan una perspectiva diferente en cuanto a la calidad del producto. De los equipos controlados, el equipo #2 es el que obtiene el más alto score (3,82), pero el score del

equipo #4 está por debajo de los dos equipos no controlados. Esto se debe a que el equipo #4 desarrollo su sistema usando una tecnología (JavaFX) que no dominaban, dejando muchas validaciones por realizarse. Esto llevó al usuario a cometer muchos errores y, a pesar de ser el proyecto con las interfaces más agradables a la vista, el usuario no se sintió cómodo usándolo, debido a la cantidad de errores que no validaba el sistema y se registraban, haciendo que los datos no sean consistentes y que el usuario realice un esfuerzo doble.

3.2.3. Conjunto de Prácticas Para Mejora la Calidad de Procesos de Software Basadas en la norma ISO/IEC 29110

Luego de haber realizado el análisis de los resultados obtenidos, podemos determinar el conjunto de prácticas para la mejora del proceso SI. A lo largo del experimento se pudo deducir algunas de prácticas que se mencionan a continuación, mientras el resto de prácticas expuestas se las determinó al concluir el experimento después de su entrega al CUS. El formato donde se detallan prácticas, se presenta en la Tabla 13.

Tabla 13 - Formato para detallar las prácticas determinadas

Actividad del proceso:	
Nombre:	
Descripción:	
Usos:	
Tareas asociadas:	

En el formato para detallar las prácticas determinadas, se puede identificar:

- La actividad del proceso donde se aplica la mejor práctica. El color de la celda depende de la actividad a la que se relaciona la mejor práctica.
- El nombre de la mejor práctica.
- La descripción de cómo y para que funciona la mejor práctica.

- Los usos que se le puede dar a la mejor práctica en la industria.
- Las tareas asociadas que apoyan a la mejor práctica.

El proceso SI, se complementa con la incorporación de un conjunto de prácticas para la mejora del proceso SI de la norma ISO/IEC 29110, determinadas en este documento. Gracias a este conjunto de prácticas y a la guía de la norma ISO/IEC 29110, el proceso SI en las VSEs será de calidad. Y por lo tanto sus productos tienen una alta probabilidad de cumplir con las expectativas del cliente y ser también de calidad

3.2.3.1 Conjunto de Prácticas para la Actividad “*Software Implementation Initiation*”

Las prácticas para la actividad "*Software Implementation Initiation*" se detallan en las Tablas 14, 15 y 16. Incluyen: capacitación en la norma ISO/IEC 29110, estimación correcta de tiempos y establecer normas de calidad y buenas prácticas de desarrollo.

Tabla 14 - Prácticas para la actividad "*Software Implementation Initiation*". Parte 1

Actividad del proceso:	Software Implementation Initiation
Nombre:	Capacitación en la norma ISO/IEC 29110
Descripción:	Se debe realizar una capacitación previa a la aplicación de la norma ISO/IEC 29110 en proyectos de desarrollo de software de un mínimo de 20 horas con un instructor con experiencia en la implementación de la misma.
Usos:	Capacitación e implementación.
Tareas asociadas:	
Realizar un curso de capacitación en la norma ISO/IEC 29110.	
Realizar un ejemplo práctico de la implementación de la norma ISO/IEC 29110.	
Evaluar el proceso de capacitación de la norma ISO/IEC 29110.	
Evaluar el impacto de la capacitación en la implementación de la norma.	

Tabla 15 - Prácticas para la actividad "*Software Implementation Initiation*". Parte 2

Actividad del proceso:	Software Implementation Initiation
Nombre:	Estimación correcta de tiempos
Descripción:	Se debe realizar la estimación de tiempos en base a las actividades a ser realizadas en el proyecto de desarrollo de software. Lo cual no afecte a las actividades del proyecto como tal, ni a las actividades del proceso de implementación de software.
Usos:	Estimación de tiempos de desarrollo.
Tareas asociadas:	

Establecer las actividades generales del proyecto.
Establecer las actividades de la norma ISO/IEC 29110.
Establecer los productos de salida han ser entregados.
Incluir tiempos de holgura para el posible retraso en alguna actividad.

Tabla 16 - Prácticas para la actividad "*Software Implementation Initiation*". Parte 3

Actividad del proceso:	Software Implementation Initiation
Nombre:	Establecer normas de calidad y buenas prácticas de desarrollo
Descripción:	Se deben establecer normas de calidad y buenas prácticas de desarrollo de software, que permitan guiar el proceso de implementación de inicio a fin. Con el objetivo aumentar la calidad de sus productos y procesos, además de facilitar el mantenimiento de sus productos de software.
Usos:	Buenas prácticas de desarrollo, calidad en el proceso de implementación.
Tareas asociadas:	
Establecer las normas de calidad y buenas prácticas de desarrollo.	
Capacitarse en las normas de calidad y buenas prácticas definidas para el proceso de implementación.	
Adscribirse a las normas de calidad y buenas prácticas de desarrollo.	
Monitorear el cumplimiento de las normas y buenas prácticas de desarrollo.	

3.2.3.2 Conjunto de Prácticas para la Actividad “*Software Requirements Analysis*”

Las prácticas para la actividad "*Software Requirements Analysis*" se detallan en las Tablas 17 y 18. Incluyen: brainwriting y uso de componentes reutilizables.

Tabla 17 - Prácticas para la actividad "*Software Requirements Analysis*". Parte 1

Actividad del proceso:	Software Requirements Analysis
Nombre:	Brainwriting
Descripción:	Se debe realizar una reunión con los stakeholders del proyecto de desarrollo de software, y conocer por medio de sus propias necesidades, los requisitos funcionales y requisitos de calidad del producto de software a ser implementado.
Usos:	Levantamiento o elicitación de requisitos.
Tareas asociadas:	
Realizar una lista de los stakeholders de la organización que solicita el producto de software.	
Realizar rondas en las cuales cada stakeholder escriba en un post-it una oración con uno de sus requerimientos.	
El stakeholder agrupará su requerimiento por afinidad con los requerimientos de los demás stakeholders.	
Al final de las rondas se identifican los grupos macros donde se sintetizan los requerimientos y se puede realizar el análisis de los mismos.	

Tabla 18 - Prácticas para la actividad "*Software Requirements Analysis*". Parte 2

Actividad del proceso:	Software Requirements Analysis
Nombre:	Uso de componentes reutilizables
Descripción:	Existen repositorios de componentes de software publicadas en la nube que pueden ser reutilizados para diferentes productos de software. En la realización del análisis de requisitos se debe identificar que componentes se pueden reutilizar.
Usos:	Cumplir con los requisitos en menor tiempo.
Tareas asociadas:	
Listar los requerimientos identificados con el cliente.	
Buscar en repositorios de código abierto los componentes que puedan cumplir con los requerimientos.	
Revisar el código del componente que se va a reutilizar para eliminar las funcionalidades que no cumplen con los requisitos.	
Integrar el componente reutilizado con la funcionalidad específica al producto de software en desarrollo.	

3.2.3.3 Conjunto de Prácticas para la Actividad “*Software Architectural and Detailed Design*”

Las prácticas para la actividad "*Software Architectural and Detailed Design*" se detallan en las Tablas 19 y 20. Incluyen: diseño detallado del producto y uso de patrones de software.

Tabla 19 - Prácticas para la actividad "*Software Architectural and Detailed Design*". Parte 1

Actividad del proceso:	Software Architectural and Detailed Design
Nombre:	Diseño detallado del producto
Descripción:	Se debe trasladar los requisitos funcionales y los requisitos de calidad identificados, con la ayuda de los stakeholders, a diagramas de diseño detallado para uso de los programadores.
Usos:	Definición de componentes e interfaces de usuario.
Tareas asociadas:	
Listar todos los requisitos identificados con los stakeholders.	
Definir para cada requisito funcional su(s) componente(s) o servicio(s) y opcionalmente su pseudocódigo.	
Definir estrategias para la implementación los requisitos de calidad.	
Elaborar mock-ups de las interfaces de usuario.	

Tabla 20 - Prácticas para la actividad "*Software Architectural and Detailed Design*". Parte 2

Actividad del proceso:	Software Architectural and Detailed Design
Nombre:	Uso de patrones de software
Descripción:	Para poder tener un producto de software de calidad, se recomienda utilizar patrones de software que son soluciones probadas a problemas de arquitectura y diseño recurrentes. Por ejemplo, el patrón Modelo-Vista-Controlador.
Usos:	Definición de la arquitectura del producto de software, implementación de funcionalidad genérica.
Tareas asociadas:	
Seleccionar patrones de software acorde a los requisitos.	
Guiar el desarrollo del producto de software por los patrones seleccionados.	
Integrar los componentes internos con los externos, definiendo protocolos de comunicación.	
Validar la arquitectura diseñada.	

3.2.3.4 Conjunto de Prácticas para la Actividad “*Software Construction*”

Las prácticas para la actividad "*Software Construction*" se detallan en las Tablas 21 y 22. Incluyen: conocimiento del ambiente integrado de desarrollo, lenguajes de programación y scripting; y uso de buenas prácticas de programación definidas en el proyecto.

Tabla 21 - Prácticas para la actividad "*Software Construction*". Parte 1

Actividad del proceso:	Software Construction
Nombre:	Conocimiento del ambiente integrado de desarrollo, lenguajes de programación y scripting
Descripción:	Los programadores deben tener un excelente conocimiento y manejo tanto del ambiente integrado de desarrollo, como de los lenguajes de programación y scripting, que se van a utilizar para la construcción y pruebas unitarias del producto de software.
Usos:	Creación de código fuente.
Tareas asociadas:	
Realizar capacitaciones del ambiente integrado de desarrollo, lenguajes de programación y scripting, que se usarán para el desarrollo del producto de software.	
Durante el desarrollo, de preferencia trabajar en parejas, a fin de que el conocimiento sea distribuido entre todos los programadores.	

Rotar el desarrollo de componentes para que todos los programadores participen de la propiedad colectiva de código para facilitar pruebas y mantenimiento futuro.

Tabla 22 - Prácticas para la actividad "*Software Construction*". Parte 2

Actividad del proceso:	Software Construction
Nombre:	Uso de buenas prácticas de programación definidas en el proyecto
Descripción:	Los programadores deben cumplir las buenas prácticas establecidas en el proyecto para la codificación del producto de software.
Usos:	Codificación cumpliendo buenas prácticas.
Tareas asociadas:	
Capacitarse en las mejoras prácticas que serán usadas durante la codificación.	
Adscribirse a las mejoras prácticas que serán usadas durante la codificación.	
Llevar a cabo revisiones de código entre pares para identificar si se están siguiendo o no las mejores prácticas y refactorizar el código revisado.	

3.2.3.5 Conjunto de Prácticas para la Actividad “*Software Integration and Tests*”

Las prácticas para la actividad "*Software Integration and Tests*" se detallan en las Tablas 23 y 24. Incluyen: integración continua; y documentación de diseño y ejecución de pruebas de sistemas.

Tabla 23 - Prácticas para la actividad "*Software Integration and Tests*". Parte 1

Actividad del proceso:	Software Integration and Tests
Nombre:	Integración continua
Descripción:	Los programadores integran su trabajo frecuentemente, al menos una vez al día en el repositorio de versionamiento. Cada integración se verifica con un build automático, incluyendo la ejecución de pruebas, para detectar defectos de integración.
Usos:	Integración de componentes y pruebas de integración.
Tareas asociadas:	
Realizar mínimo una integración diaria.	
Descargar el proyecto completo del repositorio de versionamiento.	
Ejecutar las pruebas de integración de los componentes nuevos o actualizados.	
Una vez que la integración este probada, subir el proyecto funcional al repositorio de versionamiento.	

Tabla 24 - Prácticas para la actividad "Software Integration and Tests". Parte 2

Actividad del proceso:	Software Integration and Tests
Nombre:	Documentación de diseño y ejecución de pruebas de sistemas
Descripción:	Se debe documentar los casos de prueba a ser realizados y los resultados obtenidos después de la ejecución de los casos de prueba. La ejecución puede ser manual o con apoyo de herramientas automatizadas de pruebas.
Usos:	Documentación de casos de prueba y sus resultados.
Tareas asociadas:	
Definir, preparar y documentar los casos de prueba y los datos de prueba.	
Documentar los resultados obtenidos de los casos de prueba ejecutados, para poder realizar correcciones o mejoras en el producto.	
Realizar correcciones y cambios en base a los resultados de los casos de prueba.	
Ejecutar pruebas de repetición y regresión para verificar que las correcciones y cambios estén correctos.	
Documentar los cambios realizados para tener constancia de la nueva funcionalidad establecida.	

3.2.3.6 Conjunto de Prácticas para la Actividad “*Product Delivery*”

Las prácticas para la actividad "*Product Delivery*" se detallan en las Tablas 25 y 26. Incluyen: ejecución de pruebas de aceptación y preparar presentación del producto de software.

Tabla 25 - Prácticas para la actividad "*Product Delivery*". Parte 1

Actividad del proceso:	Product Delivery
Nombre:	Ejecución de pruebas de aceptación
Descripción:	Se deben ejecutar pruebas de aceptación con la ayuda del cliente para determinar si el producto de software satisface los requerimientos planteados para su implementación. Estas pruebas son: <ul style="list-style-type: none"> - Pruebas de aceptación de usuario - Pruebas de aceptación operacional - Pruebas de aceptación de contrato y regulación - Pruebas alfa o beta
Usos de la práctica:	Aceptación del producto de software por parte del cliente, generar confianza en clientes y usuarios.
Tareas asociadas:	
Definir, preparar y documentar los casos y datos de prueba de aceptación, directamente con el cliente.	
Ejecutar, por parte del cliente, los casos de prueba de aceptación definidos.	
Generar confianza en clientes y usuarios, por los resultados obtenidos después de la ejecución de las pruebas de aceptación.	

Tabla 26 - Prácticas para la actividad "Product Delivery". Parte 2

Actividad del proceso:	Product Delivery
Nombre:	Preparar presentación del producto de software
Descripción:	El producto de software final debe ser presentado al cliente, de manera que él sea capaz de entenderlo y utilizarlo para mejorar su productividad. Para esto, la empresa de desarrollo de software debe tener la capacidad de generar confianza en el cliente respecto al producto de software que se está entregando, mediante una presentación donde explique el core del producto completamente funcional.
Usos de la práctica:	Generar confianza en clientes y usuarios.
Tareas asociadas:	
Escoger la funcionalidad que representa el core del producto de software, para ser expuesta al cliente.	
Identificar lo más relevante de la documentación realizada, a fin de presentar al cliente la misma.	
Realizar una presentación explicando el core del producto de software, y lo más relevante de la documentación generada del producto de software. Debe redactarse utilizando el vocabulario del cliente.	

4 CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

El experimento, a pesar de dar el resultado esperado, tuvo algunos inconvenientes, por ejemplo, el tiempo. El tiempo fue muy corto para realizar una implementación de software completa, los equipos tuvieron inconvenientes con el poco tiempo que se tuvo para la realización del experimento, especialmente para la construcción del producto de software, por lo que las actividades resultaron cortas de tiempo.

Se debe tener pleno conocimiento del ambiente integrado de desarrollo y lenguajes de programación y scripting para la construcción de un producto de software. El desconocimiento puede llevar a los programadores a tener problemas en la construcción del producto, y esto resulta en el incumplimiento de los criterios de aceptación del cliente. El uso de componentes reutilizables ayuda en el cumplimiento de los requisitos del cliente, siempre y cuando se revise el componente reutilizado y se adapte a las necesidades del cliente.

Al no tener una capacitación previa de la norma, los equipos controlados no realizaron los primeros productos de salida cuando se los debía realizar, y acumularon trabajo para las siguientes actividades haciendo que el trabajo se más pesado. Luego de haber estudiado

la norma, los equipos controlados supieron los productos de salida que debían entregar y recuperaron el trabajo retrasado que tenían. Además, siguieron la norma con mucho compromiso, como los resultados finales lo demuestran.

El lado positivo de este experimento es que los resultados fueron los esperados. Al término del experimento son los equipos controlados los que lideran los scores en cada una de las actividades del proceso SI. Dando como muestra que, si se sigue una norma para el proceso de desarrollo de productos de software, se pueden tener procesos de calidad y apalancar a que sus productos sean de calidad. Esto es una ventaja competitiva en el mercado o puede ser el punto de partida para abrirse camino a nuevos nichos de mercado: locales, nacionales o internacionales.

Es importante recalcar que un proceso de calidad no nos da como resultado un producto de calidad. En el experimento tenemos el claro ejemplo con el equipo #4, el cual tiene un proceso con la calificación más alta, pero su producto tiene una calificación baja por algunos inconvenientes que tuvieron en el desarrollo del mismo.

Se debe tener en cuenta las amenazas de validez del experimento realizado. En cualquier experimento pueden interferir tres tipos de amenazas de validez: internas, externas y de constructo [40]. Las amenazas de validez internas que pueden interferir en el experimento realizado, se pueden presentar ya que la organización de los equipos experimentales fue aleatoria. Por esta razón, es probable que algún equipo haya sido favorecido con miembros que tengan mayor experiencia en alguna actividad del proceso SI, o que haya existido algún tipo de afinidad entre miembros y eso produce mayor motivación o mayor expectativa del experimento. Al tener equipos controlados, el apoyo hacia estos fue mayor, lo que produce un favoritismo porque estos equipos tengan mejores resultados.

Las amenazas de validez externas se pueden presentar al tener equipos controlados y equipos no controlados. Los sujetos experimentales tienden a reaccionar al ambiente y a responder de modo diferente cuando están siendo observados y medidos, que cuando se encuentran en condiciones normales.

Las amenazas de validez de constructo en el experimento emergen por la medición de los objetos del experimento, en este sentido tenemos el número limitado de métricas usadas para la evaluación de la calidad del producto. La evaluación de la calidad del producto no fue completa sino que se realizó a modo de muestra, ya que la evaluación se

centra en la calidad del proceso. En un futuro trabajo sería favorable realizar una evaluación completa del producto de software.

4.2. Recomendaciones

Para un trabajo futuro se puede incluir en el experimento el proceso “Project Management” implementando en su totalidad la norma ISO/IEC 29110, como se lo realizó en [37]. Al tomar en cuenta ambos procesos de la norma, el tiempo del experimento debe ser más amplio y debe ser estimado tomando en cuenta colchones de tiempo para posibles inconvenientes que se pueden presentar. Además, además de probar ambos procesos de la norma, también se debe llevar a cabo un experimento complementario en VSEs establecidas, para que los resultados se obtengan en un entorno realista, en lugar de un entorno experimental, y sin diferenciar entre equipos controlados y equipos no controlados.

De la misma manera, se puede plantear un trabajo futuro donde se explore la calidad del producto en segmentos verticales de la industria, sean estos Financiero, Agropecuario, Comercial, Educación, y aumentar el tamaño de la muestra de métricas.

REFERENCIAS BIBLIOGRÁFICAS

- [1] C. Y. Laporte, M. Muñoz, J. M. Miranda y V. R. O'Connor, «Applying Software Engineering Standards in Very Small Entities,» IEEE SOFTWARE, 2018.
- [2] C. Y. Laporte y R. V. O'Connor, «Implementing Process Improvement in Very Small Enterprises with ISO/IEC 29110,» International Conference on the Quality of Information and Communications Technology, Montréal, 2016.
- [3] C. Y. Laporte, M. Muñoz y B. Gerançon, «The Education of Students About ISO/IEC 29110 Software Engineering Standards and Their Implementations in Very Small Entities,» Montreal, 2017.
- [4] M. Romero; R. Mamani; A. Dávila, «Implementation of the Basic Profile of ISO / IEC 29110 of A small software developer: Learned lessons,» Facultad de Ingeniería de Producción y Servicios – Universidad Nacional de San Agustín, Arequipa, 2015.
- [5] C. Andrade, J. S. Proaño, D. Ricaurte y J. Sáenz., «Estudio de mercado servicio desarrollo de software en Ecuador,» Guayaquil, 2012.
- [6] Asociación Ecuatoriana de Software, «Catálogo Aesoft 2015,» Quito, 2015.
- [7] J. P. Miguel, D. Mauricio y G. Rodríguez, «A Review of Software Quality Models for the Evaluation of Software Products,» *International Journal of Software Engineering & Applications*, vol. 5, nº 6, pp. 31-53, 2014.
- [8] G. A. García-Mireles, «Identifying relevant product quality characteristics in the context of very small organizations,» *Computer Science and Information Systems*, vol. 13, nº 3, p. 875–900, 2016.
- [9] A. Dávila y M. Pessoa, «Factors driving the adoption of ISO / IEC 29110: a case study of a small software enterprise,» Latin American Computing Conference, Lima, 2015.
- [10] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell y A. Wesslén, de *Experimentation in Software Engineering*, Londres, Springer, 2012, pp. 9-11.
- [11] B. Boehm, H. D. Rombach y M. V. Zelkowitz, Knoxville, 2005.
- [12] International Organization for Standardization, «ISO/IEC 9000: Quality managements systems -- Fundamentals and Vocabulary,» 2015.

- [13] International Organization for Standardization, «ISO/IEC 9001 Quality Management Systems -- Requirements,» 2015.
- [14] International Organization for Standardization, «ISO/IEC 25000:2014 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE,» 2014.
- [15] M. Takeuchi, N. Kohtake, S. Shirasaka, Y. Koishi y K. Shioya, «Report on an assessment experience based on ISO/IEC 29110,» Tokyo, 2014.
- [16] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell y A. Wesslén, de *Experimentation in Software Engineering*, Londres, Springer, 2012, pp. 16-18.
- [17] G. Travassos, P. Medeiros y P. Gomes, «An Environment to Support Large Scale Experimentation in Software Engineering,» IEEE, 2008.
- [18] A. Fabijan, P. Dmitriev, H. Holmström y J. Bosch, «The Evolution of Continuous Experimentation in Software Product Development,» IEEE, Malmö, 2017.
- [19] V. Basili, R. Selby y D. Hutchens, «Experimentation in Software Engineering,» IEEE, College Park, 1986.
- [20] M. Kuhrmann y J. Münch, «Enhancing Software Engineering Education Through Experimentation: An Experience Report,» IEEE, Goslar, 2018.
- [21] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell y A. Wesslén, de *Experimentation in Software Engineering*, Londres, Springer, 2012, pp. 76-81.
- [22] A. Dávila, P. Mamani, M. Abarca y R. Arisaca, «Analysis of adoption of ISO / IEC 29110 into two small companies Software in the city of Arequipa, Peru,» Arequipa, 2017.
- [23] International Organization for Standardization, «ISO/IEC TR 29110-5-1-2, Software Engineering—Lifecycle Profiles for Very Small Entities (VSEs) Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile,» ISO/IEC, 2011.
- [24] S. Sanchez- Gordon, M. Sánchez- Gordón, M. Yilmaz y R. O'Connor, «Integration of accessibility design patterns with the software implementation process of ISO/IEC 29110,» Software Evolution And Process, Quito, 2018.
- [25] Normalización Y Certificación NYCE S.C., «NYCE Seguridad y Confianza,» 2016. [En línea]. Available: <https://www.nyce.org.mx/ertificacion-iso-29110/>. [Último

acceso: 29 Enero 2018].

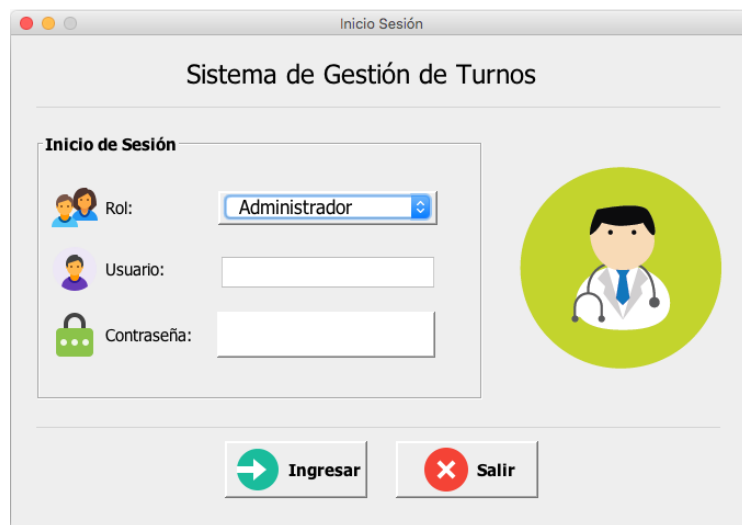
- [26] International Organization for Standardization, «ISO/IEC TR 29110-1, Software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 1: Overview,» ISO/IEC, 2011.
- [27] M. Morales-Trujillo, H. Oktaba, T. Ventura y R. Torres, «From MoProSoft Level 2 to ISO/IEC 29110 Basic Profile: Bridging the Gap,» CLEI Electronic Journal, Ciudad Universitaria, 2013.
- [28] International Organization for Standardization, *ISO/IEC TR 29110-5-1-2, Software Engineering—Lifecycle Profiles for Very Small Entities (VSEs) Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile*, 2011.
- [29] R. Ramos, S. Guun yoo, N. Velásquez y G. Ninahualpa, «Software Quality Assessment Applied for the Governmental Organizations using ISO/IEC 25000,» IEEE, Sangolqui, 2018.
- [30] International Organization for Standardization, «ISO/IEC 25000 SQuaRE (System and Software Quality Requirements and Evaluation),» 2004.
- [31] International Organization for Standardization, «ISO/IEC FDIS 25000 - Systems and Software Engineering -- Systems and Software Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE,» IEEE, 2005i.
- [32] SonarSource S.A, «sonarQube,» 2008. [En línea]. Available: <https://www.sonarqube.org/>.
- [33] K. Power, «Understanding the Impact of Technical Debt on the Capacity and Velocity of Teams and Organizations,» IEEE, San Francisco, 2013.
- [34] S. Akbarinasaji y A. Bener, «Adjusting the Balance Sheet by Appending Technical Debt,» IEEE, 2016.
- [35] J.-L. Letouzey, «The SQuaLE Method for Evaluating Technical Debt,» IEEE, Arcueli, 2012.
- [36] C. Y. Laporte y R. V. O'Connor, «Systems and software engineering standards for very small entities,» *9th International Conference on the Quality of Information and Communications Technology*, pp. 38-47, 2014.
- [37] S. Sanchez-Gordon, M.-L. Sánchez-Gordón y S. Luján-Mora, «Towards an Engineering Process for Developing Accessible Software in Small Software

Enterprises,» *Proceedings of the 11th International Conference on Evaluation of Novel Software Approaches to Software Engineering*, pp. 241-246, 2016.

- [38] C. Y. H. C. & M. C. Laporte, «Development of a social network website using the new ISO/IEC 29110 standard developed specifically for very small entities,» *Software Quality Professional*, vol. 16, n° 4, pp. 4-25, 2014.
- [39] International Organization for Standardization, «ISO/IEC 33020:2015 Information technology -- Process assessment -- Process measurement framework for assessment of process capability,» 2015.
- [40] R. Feldt y A. Magazinius, «Validity Threats in Empirical Software Engineering Research-An Initial Survey,» de *International Conference on Software Engineering and Knowledge Engineering (SEKE)*, 2010.

ANEXOS

Anexo I - Interfaces del producto de software del equipo no controlado #1



Registrar Paciente

Formulario Paciente Nuevo

* Cedula

* Primer Nombre

* Primer Apellido

* Fecha Nacimiento

* Direccion

* Email

Segundo Nombre

Segundo Apellido

* Telefono 1

Telefono 2

* Campos Obligatorios

Aceptar Cancelar

Buscar Paciente

Formulario Buscar Empleado

Ingrese los datos para buscar

Número de Cédula:

Buscar Limpiar

Resultados de búsqueda

CÉDULA	P.NOMBRE	S.NOMBRE	P.APELLI...	S.APELLI...	P.TELEF...	S.TELEF...	EMAIL	DIRECCI...	F.NACIM...

Ingresar los datos para buscar

Número de Cédula:

Resultados de búsqueda

CÉDULA	P.NOMBRE	S.NOMBRE	P.APELLIDO	S.APELLIDO	P.TELEFO...	S.TELEFO...	EMAIL	DIRECCION	F.NACIME...

* Cedula

* Primer Nombre Segundo Nombre

* Primer Apellido Segundo Apellido

* Fecha Nacimiento * Telefono 1

* Direccion Telefono 2

* Email

* Datos Obligatorios

Registrar Empleado

Registrar Empleado

Formulario Empleado Nuevo

Información Personal

Número de Cédula:

Primer Nombre: Segundo Nombre:

Primer Apellido: Segundo Apellido:

Fecha Nacimiento: Correo Electrónico:

Información Laboral

Departamento: Especialidad:

Buscar Empleado

Buscar Empleado

Formulario Buscar Empleado

Ingrese los datos para buscar

Número de Cédula:

Resultados de búsqueda

CÉDULA	P.NOMBRE	S.NOMBRE	P.APELLIDO	S.APELLIDO	F.NACIMIENTO	EMAIL

Actualizar Empleado

Actualizar Empleado

Formulario Actualizar Empleado

Ingresar los datos para buscar

Número de Cédula:

Resultados de búsqueda

CÉDULA	P.NOMBRE	S.NOMBRE	P.APELLIDO	S.APELLIDO	F.NACIMIENTO	EMAIL

Información Personal

Número de Cédula:

Primer Nombre: Segundo Nombre:

Primer Apellido: Segundo Apellido:

Fecha Nacimiento: Correo Electrónico:

Información Laboral

Departamento: Especialidad:

Agendar Turno

Agendar Turno

Formulario para Agendar Turnos

Seleccione un Profesional

Especialidad: Nombre del Profesional: Disponibilidad Profesional:

Fecha y Hora

Seleccione año, mes y día:

Detalle turno:

Detalle Turno:

Seleccione hora:

Seleccione un paciente

Ingrese los datos para buscar

Número de Cédula:

Resultados de búsqueda

CÉDULA	P.NOMBRE	S.NOMBRE	P.APELLIDO	S.APELLIDO	P.TELEFONO	EMAIL

Detalle de paciente

Cédula:

Primer Nombre: Segundo Nombre:

Primer Apellido: Segundo Apellido:

Teléfono: Correo Electrónico:

Crear Usuario

Datos

Rol de Usuario

Nombre de Usuario

Contraseña

Confirmar Contraseña

Usuarios Registrados

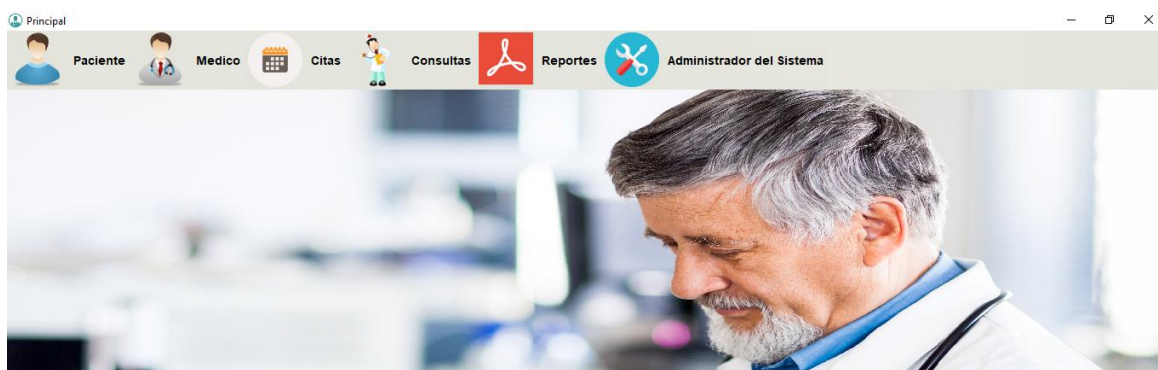
Anexo II - Interfaces del producto de software del equipo controlado #2

Iniciar Sesión

Usuario(*)

Contraseña(*)

Conectar Salir



Agregar Medico

Nombres(*)

Apellidos(*)

Cédula(*)

Correo(*)

Especialidad(*)

Guardar y Agregar Horario Limpiar Salir

Agregar Horario X

Días disponibles L M X J V S D

Lunes Desde: 10:00 A.M. Hasta: 1:00 P.M.

Martes Desde: 8:00 A.M. Hasta: 11:00 A.M.

Miercoles Desde: 9:00 A.M. Hasta: 2:00 P.M.

Jueves Desde: 2:00 P.M. Hasta: 5:00 P.M.

Ver Medico _ □ X

Buscar:

No.	Nombres	Apellidos	Cédula	Correo	Especialidad	Estado
1	Juan Carlos	Montero	0987654321	hfhua@ho.com	Dentista	Activo
2	Fabiola	Cadena	1234567891	hgu@gk.com	Dentista	Activo
3	Juan Fernando	Velazco	999999990	huj@hotmail.com	Dermatología	Activo

Modificar Medico

Nombres(*)

Apellidos(*)

Cédula(*)

Correo(*)

Especialidad(*)

Modificar Horario

Dias disponibles L M X J V S D

Lunes	Desde	<input type="text" value="<Seleccione>"/>	Hasta	<input type="text" value="<Seleccione>"/>
Martes	Desde	<input type="text" value="<Seleccione>"/>	Hasta	<input type="text" value="<Seleccione>"/>
Miercoles	Desde	<input type="text" value="<Seleccione>"/>	Hasta	<input type="text" value="<Seleccione>"/>
Jueves	Desde	<input type="text" value="<Seleccione>"/>	Hasta	<input type="text" value="<Seleccione>"/>
Viernes	Desde	<input type="text" value="<Seleccione>"/>	Hasta	<input type="text" value="<Seleccione>"/>
Sabado	Desde	<input type="text" value="<Seleccione>"/>	Hasta	<input type="text" value="<Seleccione>"/>
Domingo	Desde	<input type="text" value="<Seleccione>"/>	Hasta	<input type="text" value="<Seleccione>"/>

Agregar Paciente

Nombres(*)	<input type="text"/>
Apellidos(*)	<input type="text"/>
Telefono(*)	<input type="text" value="-"/>
Cédula(*)	<input type="text"/>
Correo(*)	<input type="text"/>
Genero(*)	<input type="text" value=" <Seleccione>"/>

Modificar Paciente

Nombres(*)	<input type="text" value="Hugo Patricio"/>
Apellidos(*)	<input type="text" value="Montero Cadena"/>
Telefono(*)	<input type="text" value="1231-3124"/>
Cédula(*)	<input type="text" value="0503934135"/>
Correo(*)	<input type="text" value="hugo29ks@hotmail.com"/>
Genero(*)	<input type="text" value="Masculino"/>

Agregar Cita

Paciente(*)

Especialidad(*)

Medico(*)

Días disponibles L M X J V S D

Seleccione la fecha de la cita en los días disponibles del medico

junio 2018

	lun	mar	mié	jue	vie	sáb	dom
22					1	2	3
23	4	5	6	7	8	9	10
24	11	12	13	14	15	16	17
25	18	19	20	21	22	23	24
26	25	26	27	28	29	30	

Seleccione la hora de la cita en la tabla de hora...

Hora	Estado
8:00 A.M	
9:00 A.M	
10:00 A.M	
11:00 A.M	
12:00 P.M	
1:00 P.M	
2:00 P.M	
3:00 P.M	
4:00 P.M	
5:00 P.M	

Guardar Limpiar Salir

Ver Cita

Buscar:

No.	Medico	Fecha	Dia	Hora	Paciente	Estado
2	Dr. Juan Carlos ...	29-may-2018	Martes	2:00 P.M	Hugo Patricio ...	Pendiente
3	Dr. Juan Carlos ...	29-may-2018	Martes	1:00 P.M	Hugo Patricio ...	Atendida
4	Dr. Fabiola Cad...	04-jun-2018	Lunes	10:00 A.M	Hugo Patricio ...	Atendida
5	Dr. Fabiola Cad...	04-jun-2018	Lunes	11:00 A.M	Hugo Patricio ...	Atendida
6	Dr. Fabiola Cad...	04-jun-2018	Lunes	12:00 P.M	Hugo Patricio ...	Atendida
7	Dr. Fabiola Cad...	04-jun-2018	Lunes	12:00 P.M	Hugo Patricio ...	Atendida
1	Dr. Juan Fernan...	28-may-2018	Lunes	9:00 A.M	Hugo Patricio ...	Atendida
8	Dr. Fabiola Cad...	11-jun-2018	Lunes	10:00 A.M	David Pazmiño	Pendiente

Cancelar Salir

Agregar Consulta

Dias disponibles L M X J V S D

Seleccione la fecha de la cita en los dias disponibles del medico

junio 2018

	lun	mar	mié	jue	vie	sáb	dom
22					1	2	3
23	4	5	6	7	8	9	10
24	11	12	13	14	15	16	17
25	18	19	20	21	22	23	24
26	25	26	27	28	29	30	

Seleccione la hora de la cita en la tabla de horario

Hora	Estado
10:00 A.M	Cita con David Pazmiño (Cancelada)
11:00 A.M	Libre
12:00 P.M	Cita con David Pazmiño (Pendiente)
1:00 P.M	Libre

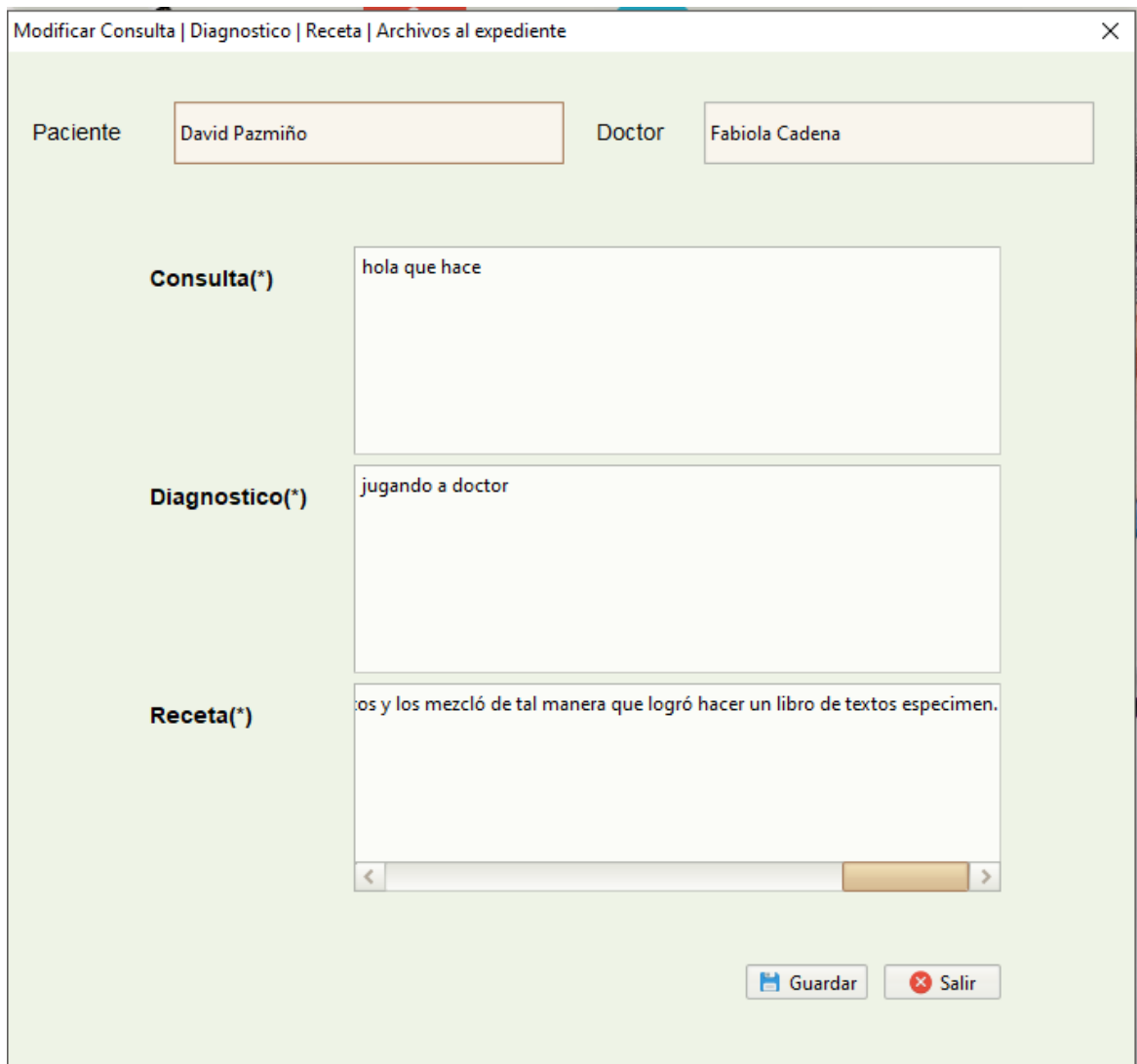
Agregar Consulta | Diagnostico | Receta | Archivos al expediente

Paciente Doctor

Consulta(*)

Diagnostico(*)

Receta(*)



Horario General - Mensual

Horario Mensual

No.	Fecha	Dia	Hora	Paciente	Estado
Dr. Fabiola Cadena					
4	04/jun/2018	Lunes	10:00 A.M	Hugo Patricio Montero Cadena	Atendida
5	04/jun/2018	Lunes	11:00 A.M	Hugo Patricio Montero Cadena	Atendida
6	04/jun/2018	Lunes	12:00 P.M	Hugo Patricio Montero Cadena	Atendida
7	04/jun/2018	Lunes	12:00 P.M	Hugo Patricio Montero Cadena	Atendida
8	11/jun/2018	Lunes	10:00 A.M	David Pazmiño	Cancelada
9	11/jun/2018	Lunes	12:00 P.M	David Pazmiño	Atendida

Horario General - Semanal

Horario Semanal

No.	Fecha	Dia	Hora	Paciente	Estado
Dr. Fabiola Cadena					
8	11/jun/2018	Lunes	10:00 A.M	David Pazmiño	Cancelada
9	11/jun/2018	Lunes	12:00 P.M	David Pazmiño	Atendida

Horario General - Mensual

Horario General Mensual

No.	Fecha	Dia	Hora	Paciente	Estado
Dr. Fabiola Cadena					
4	04/jun/2018	Lunes	10:00 A.M	Hugo Patricio Montero Cadena	Atendida
5	04/jun/2018	Lunes	11:00 A.M	Hugo Patricio Montero Cadena	Atendida
6	04/jun/2018	Lunes	12:00 P.M	Hugo Patricio Montero Cadena	Atendida
7	04/jun/2018	Lunes	12:00 P.M	Hugo Patricio Montero Cadena	Atendida
8	11/jun/2018	Lunes	10:00 A.M	David Pazmiño	Cancelada
9	11/jun/2018	Lunes	12:00 P.M	David Pazmiño	Atendida

Anexo III - Interfaces del producto de software del equipo no controlado #3

Iniciar Sesión ✕

DEPARTAMENTO MÉDICO

Login



Usuario*:


Contraseña*:


Ingresar al Sistema

Salir del Sistema

↑ VENTANA PRINCIPAL - □ ✕


 **Paciente**

 **Especialidad**

 **Medico**

 **Agendar Citas**

 **Atención Médica**

 **Usuarios**

 **REPORTES**

AÑADIR NUEVO PACIENTE

DATOS PERSONALES

Cédula:*

Nombres:* Apellidos:*

Teléfono:* Dirección:*

e-mail:*

DATOS MÉDICOS GENERALES

Peso (kg):* Altura:*

Edad:* Género:* Masculino Femeni...

Tipo de Sangre:*

Enfermedades: Alergias:

LISTA DE PACIENTES REGISTRADOS

Buscar Pacientes Registrados:

No.	Nom...	Apelli...	Peso	Altura	Edad	Telefo...	Alergi...	Enfer...	Tipo ...	Estado
2	peric...	de lo...	23.0	1.23	23	0996-...	ningu...	infarto	O+	Activo
3	peric...	de lo...	23.0	2.0	18	0996-...	ningu...	infarto	O+	Activo
5	Andres	Guam...	68.0	1.7	26	4564...	Queso	Todas	O+	Activo
9	Andres	Guam...	68.0	1.7	26	4564...	Queso	Todas	O+	Activo
13	Andres	Guam...	68.0	1.7	26	4564...	Queso	Todas	O+	Activo
14	Carlo...	Gutier...	70.0	170.0	26	0226-...	Ningu...	Ning...	O+	Activo

MODIFICAR DATOS DEL PACIENTE



DATOS PERSONALES



Cédula:* 1245984598

Nombres:* Andres Apellidos:* Guaman

Teléfono:* 4564-798 Dirección:* por ahi

e-mail:* jj@qma.com

DATOS MÉDICOS GENERALES

Peso:* 68.0 Altura:* 1.7

Edad:* 26 Género:* Masculino Femenino

Tipo de Sangre:* O+

Enfermedades: Todas Alergias: Queso

Guardar Cambios

Salir

Paciente Especialidad Medico Agendar Citas Atención Médica Usuarios Horario

LISTA DE PACIENTES REGISTRADOS

Buscar Pacientes Registrados:

No.	Nom...	Apelli...	Peso	Altura	Edad	Telefo...	Alergi...	Enfer...	Tipo ...	Estado
2	peric...	de lo...	23.0	1.23	23	0996-...	ningu...	infarto	O+	Activo
3	peric...	de lo...	23.0	2.0	18	0996-...	ningu...	infarto	O+	Activo
5	Andres	Guam...	68.0	1.7	26	-	Queso	Todas	O+	Activo
9	Andres	Guam...	68.0	1.7	26	4564...	Queso	Todas	O+	Activo
13	Andres	Guam...	68.0	1.7	26	4564...	Queso	Todas	O+	Activo
14	Carlo...	Gutier...	70.0	170.0	26	0226-...	Ningu...	Ning...	O+	Activo

Medico Agendar Citas Atención Médica Usuarios Horario

Ver Medico

Buscar:

No.	Nombres	Apellidos	Especialidad	Estado
2	PERICO	Perez	PROCTOLOGIA	Activo







AGREGAR ESPECIALIDAD

Nuevas Especialidades Médicas


Nombre de la Especialidad:*

Descripción:*

VENTANA PRINCIPAL

Paciente  Especialidad  Medico  Agendar Citas  Atención Médica  Usuarios  REPORTES

AÑADIR MÉDICO

Registrar Médicos 

Cédula:*

Nombres:*

Apellidos:*

Especialidad Médica:*

Horario del Médico:*

By LagartSoft

Modificar Horario

HORARIO DE TRABAJO

Días Laborables

Lunes
 Martes
 Miércoles
 Jueves
 Viernes
 Sábado

Seleccione el Horario Diario

Lunes Hora Entrada: 8:00 A.M. Hora Salida: 12:00 P.M.

Martes Hora Entrada: 8:00 A.M. Hora Salida: 12:00 P.M.

Miércoles Hora Entrada: 8:00 A.M. Hora Salida: 12:00 P.M.

Jueves Hora Entrada: <Selecione> Hora Salida: <Selecione>

Viernes

Sábado

Domingo

AGENDAR CITAS MÉDICAS

Criterio Previo Para Agendamento

Paciente:* <Selecione>

Especialidad:* <Selecione>

Médico:* <Selecione>

Dias disponibles del Médico

Lunes
 Martes
 Miércoles
 Jueves
 Viernes
 Sábado
 Domingo

Escoger Fecha de la Cita Médica

CALENDARIO

June 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Escoger Hora de la Cita Médica

Hora	Estado
7:00 A.M	
8:00 A.M	
9:00 A.M	
10:00 A.M	
11:00 A.M	
12:00 P.M	
1:00 P.M	
2:00 P.M	

LISTA DE CITAS AGENDADAS

CITAS AGENDADAS



Buscar Citas Médicas por: Paciente

No.	Medico	Fecha	Dia	Hora	Paciente	Estado
1	Dr. uio uio	11-jun-2018	Lunes	9:00 A.M	yhujk yuio	Pendiente

 Cancelar
 Salir

Seleccionar modo reporte

REPORTES

Tipo de reporte:



Semanal

Generar Reporte

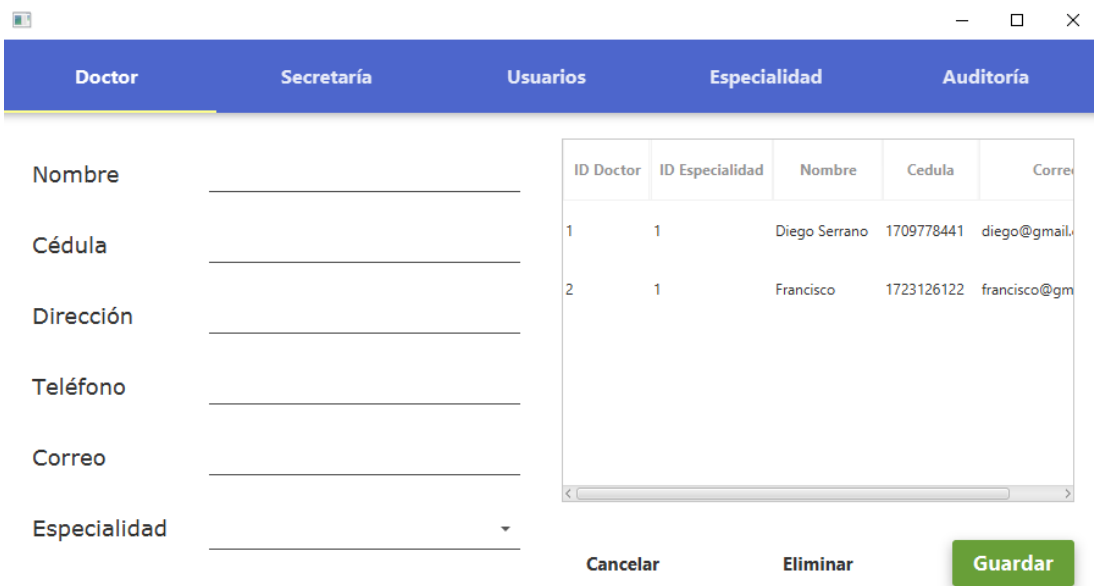


Mensual

Generar Reporte

 Salir

Anexo IV - Interfaces del producto de software del equipo controlado #4



Doctor **Secretaría** Usuarios Especialidad Auditoría

Nombre

Cédula

Dirección

Teléfono

Correo

ID Secretaria	Nombre	Cedula	Correo
1	Maria	1544877852	maria@gmail.com

Cancelar Eliminar **Guardar**

Doctor Secretaría **Usuarios** Especialidad Auditoría

Rol

Nombre

Usuario

Contraseña

ID Rol	Usuario	ID Asociado
1	admin	0
3	secretaria	2
2	francisco	2

Cancelar Eliminar **Guardar**

Doctor Secretaría Usuarios **Especialidad** Auditoría


Nombre Especialidad

Guardar


ID Especialidad	Nombre
1	Dentista
2	Medico General

Cancelar
Eliminar


Menu Principal Secretaria




Paciente



Doctor



Citas Medicas



Reportes

Registro
Historial

Nombre *

Cédula *

Dirección

Teléfono

Correo *

ID Paciente	Nombre	Cedula	Correo	
Tabla sin contenido				

Cancelar Eliminar Guardar

Horarios
Agenda
Historial Citas

Doctor * Buscar

Intervalo Dias Lunes Martes Miercoles Jueves Viernes *

Fecha Inicio *

Fecha Fin *

Hora Inicio *

Hora Fin *

Cancelar Guardar

80

Buscar Doctor

ID Doctor	ID Especialidad	Nombre	Cedula	Correo
1	1	Diego Serrano	1709778441	diego@gmail.com
4	1	Psic. Fernando Torres	1721458697	fernando.torres@hotmail.com

Cancelar **Seleccionar**

Menu Doctor

Horarios **Agenda** Historial Citas

< Hoy >

martes, 05. junio 2018

Agenda desde 5 de junio de 2018 hasta 5 de julio de 2018

No Hay Entradas

All Day

9:00

10:00

11:00

12:00

12:59

14:00

15:00

16:00

17:00

18:00

19:00

20:00

Menu Citas

< Hoy >

domingo, 03. junio 2018

Agenda desde 3 de junio de 2018 hasta 3 de julio de 2018

domingo	03-jun-2018
Dentista 1	9:15 to 11:15
lunes	04-jun-2018
Dentista 1	9:15 to 11:15
martes	05-jun-2018
Dentista 1	9:15 to 11:15
miércoles	06-jun-2018
Dentista 1	9:15 to 11:15

All Day

Nueva Cita
Reagendar Cita

Menu Citas

< Hoy >

domingo, 03. junio 2018

Agenda desde 3 de junio de 2018 hasta 3 de julio de 2018

domingo	03-jun-2018
Dentista 1	9:15 to 11:15
lunes	04-jun-2018
Dentista 1	9:15 to 11:15
martes	05-jun-2018
Dentista 1	9:15 to 11:15
miércoles	06-jun-2018
Dentista 1	9:15 to 11:15

All Day

Nueva Cita

Paciente: _____

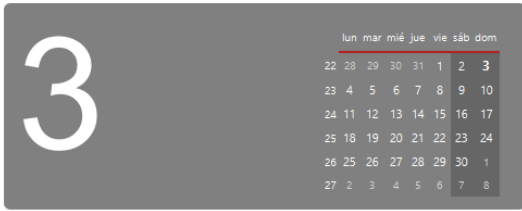
Fecha: 3/06/2018

Hora Inicio: _____

Hora Fin: _____

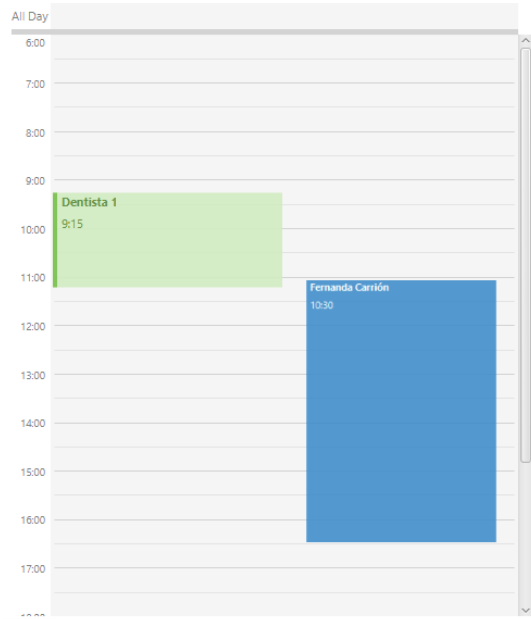
Guardar

Nueva Cita
Reagendar Cita



Agenda desde 3 de junio de 2018 hasta 3 de julio de 2018

domingo	03-jun-2018
Dentista 1	9:15 to 11:15
Fernanda Carrión	10:30 to 17:05
lunes	04-jun-2018
Dentista 1	9:15 to 11:15
martes	05-jun-2018
Dentista 1	9:15 to 11:15
miércoles	06-jun-2018
Dentista 1	9:15 to 11:15



Nueva Cita

Reagendar Cita

Anexo V – Historias de Usuario del equipo no controlado #2

Historia de Usuario			
Código	1	Número de Sprint:	2
Nombre	Inicio de sesión		
Actor	Médico		
Descripción	Como médico quiero iniciar sesión en la aplicación de escritorio para acceder a las funcionalidades específicas de acuerdo al perfil registrado.		
Criterios de aceptación			
Condición		Resultado	
Cuando se inicia sesión en la aplicación de escritorio		Se debe cumplir que el usuario y contraseña coincidan. "Usuario inició sesión correctamente"	
		Se debe cumplir que si el usuario es diferente a la contraseña o viceversa, se presente un mensaje "Usuario o contraseña incorrecta"	
		Se debe validar que se ingrese el valor del usuario y contraseña, que estos sean campos obligatorios	

Historia de Usuario			
Código	2	Número de Sprint:	4
Nombre	Ver los pacientes asignados y buscar un paciente por nombre		
Actor	Médico		
Descripción	Como médico quiero consultar los pacientes asignados con las respectivas horas y días.		
Criterios de aceptación			
Condición		Resultado	
Cuando se visualiza los pacientes asignados y se realiza una búsqueda específica proporcionando el nombre		Resultado cuando se consulta una cita del paciente Se debe cumplir que en la consulta se presente el nombre del paciente	
		Se debe cumplir que en la cita asignada se presente el día, hora y fecha en que la recepcionista agendó la cita con el médico	

Historia de Usuario			
Código	3	Número de Sprint:	2
Nombre	Inicio de sesión		
Actor	Recepcionista		
Descripción	Como recepcionista quiero iniciar sesión en la aplicación de escritorio para acceder a las funcionalidades específicas de acuerdo al perfil registrado.		
Criterios de aceptación			
Condición		Resultado	
Cuando se inicia sesión en la aplicación de escritorio		Se debe cumplir que el usuario y contraseña coincidan. "Usuario inició sesión correctamente"	
		Se debe cumplir que si el usuario es diferente a la contraseña o viceversa, se presente un mensaje "Usuario o contraseña incorrecta"	
		Se debe validar que se ingrese el valor del usuario y contraseña, que estos sean campos obligatorios	

Historia de Usuario			
Código	4	Número de Sprint:	2
Nombre	Registro información general del paciente		
Actor	Recepcionista		
Descripción	Como recepcionista quiero registrar en la aplicación la información general de cada paciente.		
Criterios de aceptación			
Condición		Resultado	
Cuando se registra un paciente		Se debe validar que la cédula ingresada sea válida	
		Se debe validar que el mail ingresado no pertenezca a otro usuario, y que sea única de una persona	
		Se debe validar que los campos de cédula, nombre, apellidos, teléfono, e-mail, dirección sean obligatorios	

Historia de Usuario			
Código	5	Número de Sprint:	2
Nombre	Registro información general del médico		
Actor	Recepcionista		
Descripción	Como recepcionista quiero registrar en la aplicación la información general de cada paciente.		
Criterios de aceptación			
Condición		Resultado	
Cuando se registra un médico		Se debe validar que la cédula ingresada sea válida	
		Se debe validar que el mail ingresado no pertenezca a otro usuario, y que sea única de una persona	
		Se debe validar que los campos de cédula, nombre, apellidos, teléfono, e-mail, dirección sean obligatorios	

Historia de Usuario			
Código	6	Número de Sprint:	3
Nombre	Calendario - Registro de Horarios disponibles para las citas médicas		
Actor	Recepcionista		
Descripción	Como recepcionista quiero registrar el horario disponible de atención por cada médico.		
Criterios de aceptación			
Condición		Resultado	
Calendario - Registro de Horarios disponibles para la consulta		Se debe cumplir que se agenda horarios disponibles únicamente en el mes en curso	
		Se debe cumplir que dos horarios no coincidan	
		Se debe cumplir que el horario agendado tenga una duración de 2 horas o menos	

Historia de Usuario			
Código	7	Número de Sprint:	3
Nombre	Registro de citas médicas		
Actor	Recepcionista		
Descripción	Como recepcionista quiero registrar una cita médica en el horario disponible de atención del médico y especialidad que necesite.		
Criterios de aceptación			

Condición	Resultado
Condición Resultado Registro de cita médica	Se debe cumplir que únicamente se pueda registrar citas médicas en el mes vigente
	Se debe cumplir que únicamente puede registrar citas en el horario que el médico ha definido previamente
	Se debe cumplir que el registro de la cita sea según el médico seleccionado
	Se debe cumplir que el registro de la cita sea según el consultorio seleccionado
	Se debe validar que se puede eliminar la cita médica hasta con 2 horas de anticipación
	Se debe cumplir que las citas visualizadas sean únicamente de los horarios disponibles

Historia de Usuario			
Código	8	Número de Sprint:	3
Nombre	Consulta de citas médicas registradas		
Actor	Recepcionista		
Descripción	Como recepcionista quiero visualizar las citas médicas registradas.		
Criterios de aceptación			
Condición	Resultado		
Cuando se consulta las citas registradas	Se debe cumplir que en la consulta se muestre el nombre del médico, el nombre del consultorio, la fecha y la hora		
	Se debe cumplir que el recepcionista seleccione la especialidad para visualizar las citas que tiene registradas en esa especialidad		
	Se debe cumplir que el recepcionista visualice la información actualizada de cada cita médica		

Historia de Usuario			
Código	9	Número de Sprint:	2
Nombre	Autorización de médicos		
Actor	Administrador		
Descripción	Requiero aprobar los médicos que cuenten con un registro valido de acceso al sistema.		

Criterios de aceptación	
Condición	Resultado
Cuando consulte los médicos registrados	Se debe cumplir que en el listado de médicos únicamente se muestren los que se encuentren en estado pendiente de autorización (activos)

Historia de Usuario			
Código	10	Número de Sprint:	3
Nombre	Registro de especialidades		
Actor	Administrador		
Descripción	Cuando el recepcionista quiere dejar registrar un consultorio.		
Criterios de aceptación			
Condición	Resultado		
Cuando registro una especialidad	Se debe cumplir que el valor ingresado para el nombre sea obligatorio		

Historia de Usuario			
Código	11	Número de Sprint:	2
Nombre	Creación de usuarios		
Actor	Administrador		
Descripción	Requiero crear un nuevo usuario para el sistema con rol de recepcionista o administrador.		
Criterios de aceptación			
Condición	Resultado		
Cuando cree un nuevo usuario	Se debe cumplir que se llene el campo de usuario y contraseña de manera obligatoria		