

UNIVERSIDAD METROPOLITANA DEL ECUADOR



FACULTAD DE INGENIERÍA

INGENIERÍA EN SISTEMAS DE INFORMACIÓN

SEDE QUITO

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCION DEL TÍTULO
DE INGENIERO EN SISTEMAS DE INFORMACION**

**TEMA: Sistema informático para el registro y control de asistencias
y actividades semanales para la Empresa Eléctrica Quito**

Autor: José Emilio Fonseca Paz

Tutor: Ing. Tonyse de la Rosa Martín, M.sc.

Quito – 2021

CERTIFICACIÓN DEL TUTOR

Ing. Tonyse de la Rosa Martín, M.sc., calidad de Asesor del Trabajo de Graduación o Titulación.

CERTIFICA:

Que el trabajo de Graduación ha sido desarrollado de manera autónoma por el señor José Emilio Fonseca Paz, portador de la cédula de identidad N°. 172330435-6 con el tema: “Sistema informático para el registro y control de asistencias y actividades semanales para la Empresa Eléctrica Quito” el mencionado señor ha cumplido con todos los requisitos legales exigidos.

Es todo cuanto puedo decir en honor a la verdad, facultando al interesado hacer uso de la presente.

Ing. Tonyse de la Rosa Martín, M.Sc.
Cl. Atentamente,

CERTIFICACIÓN DE AUTORÍA DE TRABAJO DE TITULACIÓN

Yo, **José Emilio Fonseca Paz**, estudiante de la Universidad Metropolitana del Ecuador “UMET”, **Ingeniería en Sistemas de Información**, declaro en forma libre y voluntaria que el presente trabajo de investigación que versa sobre: **Sistema informático para el registro y control de asistencias y actividades semanales para la Empresa Eléctrica Quito** y las expresiones vertidas en la misma, son autoría del compareciente, las cuales se han realizado en base a recopilación bibliográfica, consultas de internet y consultas de campo.

En consecuencia, asumo la responsabilidad de la originalidad de la misma y el cuidado al referirme a las fuentes bibliográficas respectivas para fundamentar el contenido expuesto.

Atentamente,

JOSE EMILIO FONSECA PAZ
C.I. 1723304356
AUTOR

CESIÓN DE DERECHOS DE AUTOR

Yo, JOSE EMILIO FONSECA PAZ, en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación, Sistema informático para el registro y control de asistencias y actividades semanales para la Empresa Eléctrica Quito, modalidad Proyecto de investigación de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN, cedo a favor de la Universidad Metropolitana del Ecuador una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Conservo a mi favor todos los derechos de autor sobre la obra, establecidos en la normativa citada.

Así mismo, autorizo a la Universidad Metropolitana del Ecuador para que realice la digitalización y publicación de este trabajo de titulación en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

El autor declara que la obra objeto de la presente autorización es original en su forma de expresión y no infringe el derecho de autor de terceros, asumiendo la responsabilidad por cualquier reclamación que pudiera presentarse por esta causa y liberando a la Universidad de toda responsabilidad.

José Emilio Fonseca Paz
Ci: 1723304356

DEDICATORIA

A Jesucristo mi Señor y Salvador que pacientemente espere en él, la provisión para generar los medios de mi formación profesional y se inclinó a mí y oyó mi clamor. Y me hizo sacar del pozo de la desesperación del lodo cenagoso; puso mis pies sobre peña, y enderezo mis pasos. Puso luego en mi boca cántico nuevo, alabanza a nuestro Dios. Verán esto muchos, y temerán, y confiarán en Dios. Te amo, oh Jehová, fortaleza mía. Jehová roca mía y castillo mío, y mi libertador; Dios mío, fortaleza mía, en él confiaré; mi escudo, y la fuerza de mi salvación, mi alto refugio. Invocare a Jehová, quien es digno de ser alabado, y seré salvo de mis enemigos.

AGRADECIMIENTO

Mi agradecimiento total es a Jesucristo por ser mi Señor y Salvador. Bendice, alma mía, a Jehová, y bendiga todo mi ser su santo nombre. Bendice, alma mía, a Jehová, y no olvides ninguno de sus beneficios. Él es quien perdona todas tus iniquidades. Él es quien sana todas tus dolencias. El que rescata del hoyo tu vida. El que te corona de favores y misericordias. El que sacia de bien tu boca de modo que te rejuvenezcas como el águila. Jehová es el que hace justicia y derecho a todos los que padecen violencia. Sus caminos notifico a Moisés y a los hijos de Israel sus obras. Misericordioso y clemente es Jehová. Lento para la ira y grande en misericordia. No contendrá para siempre, ni para siempre guardará el enojo. No ha hecho con nosotros conforme a nuestras iniquidades, ni nos ha pagado conforme a nuestros pecados.

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN DEL TUTOR.....	II
CERTIFICACIÓN DE AUTORÍA DE TRABAJO DE TITULACIÓN	III
CESIÓN DE DERECHOS DE AUTOR.....	IV
DEDICATORIA	V
AGRADECIMIENTO	VI
ÍNDICE DE CONTENIDOS	VII
Índice de figuras	X
Índice de tablas.....	X
Índice de anexos.....	XII
RESUMEN	XIII
ABSTRACT	XIV
INTRODUCCIÓN	1
Misión	2
Visión.....	2
Situación problemática	2
Formulación del problema	2
Objetivo general.....	3
Objetivos específicos.....	3
Preguntas de investigación.....	3
Justificación de la investigación.....	4
Viabilidad de la investigación.....	4
Tipo de investigación	5
Hipótesis de la investigación	5
Estructura del documento	5
CAPÍTULO I.....	7
1. FUNDAMENTACIÓN TEÓRICA.....	7
1.1. Breve descripción del contenido del capítulo.....	7
1.2. Objetivos estratégicos de la organización objeto de la investigación	7
1.3. Objeto de estudio.....	8

1.3.1. Flujo actual del proceso	8
1.3.2. Análisis crítico de la ejecución de los procesos	10
1.3.3. Procesos de objeto de automatización	10
1.4. Sistemas automatizados existentes vinculados al campo de acción	10
1.4.1. Bixpe app de fichaje y control de horario	11
1.4.2. Work Time App registro horario legal	12
1.5. Caracterización de la problemática.....	12
1.6. Tendencias y metodologías actuales.....	13
1.6.1. Fundamentación de la metodología utilizada	13
1.6.2. Metodología ágil XP (Programación Extrema)	24
1.7. Fundamentación del entorno de desarrollo	26
1.7.1. Node.js	27
1.7.2. Angular.....	28
1.7.3. Framework Express.js.....	29
1.7.4. Oracle	29
1.7.5. Lucidchart.....	30
1.8. Conclusiones del capítulo	30
CAPÍTULO II	32
2. MODELO DE NEGOCIO Y REQUISITOS	32
2.1. Breve descripción del contenido del capítulo.....	32
2.2. Análisis de mejora de procesos	32
2.3. Reglas del negocio a considerar.....	33
2.4. Requisitos de negocio	34
2.4.1. Especificación de los requisitos funcionales	35
2.4.2. Especificación de los requisitos no funcionales	38
2.5. Actores del sistema a automatizar	41
2.6. Diagrama de casos de uso del sistema	41
2.6.1. Descripción del CU Autenticar Usuario	43
2.6.2. Descripción del CU Gestionar Asistencias	44
2.6.3. Descripción del CU Gestionar Actividades.....	45
2.6.4. Descripción del CU Gestionar Aprobaciones	46
2.6.5. Descripción del CU Gestionar Reportes	47
2.7. Conclusiones del capítulo	47
CAPITULO III	49

3. DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.....	49
3.1. Breve descripción del contenido del capítulo.....	49
3.2. Arquitectura del sistema	49
3.3. Diagrama de clases	50
3.3.1. Diagrama de clases de asistencias.....	51
3.3.2. Diagrama de clases de actividades.....	51
3.4. Diagrama de secuencia	51
3.4.1. Diagrama de secuencia de asistencias.....	53
3.4.2. Diagrama de secuencia de actividades.....	54
3.5. Tratamiento de errores	55
3.6. Diseño de la base de datos	55
3.7. Diagrama de despliegue.....	56
3.8. Diagrama de componentes.....	57
3.9. Conclusiones de capítulo.....	58
CAPÍTULO IV	59
4. PRUEBAS Y ANÁLISIS DE FACTIBILIDAD.....	59
4.1. Breve descripción del contenido del capítulo.....	59
4.2. Planificación basada en método de estimación	59
4.2.1. Método de estimación de esfuerzo	59
4.2.2. Factores de complejidad	60
4.2.3. Factores técnicos	61
4.2.4. Factores ambientales.....	63
4.2.5. Métricas a medir para realizar la estimación	65
4.2.6. Porcentaje de actividades por ciclo de vida	65
4.3. Casos de pruebas.....	67
4.3.1. Clases de equivalencia	67
4.4. Verificación del sistema	74
4.5. Conclusiones del capítulo.....	74
CONCLUSIONES	76
RECOMENDACIONES	77
BIBLIOGRAFIA	78
ANEXOS	84

Índice de figuras

Figura 1: Proceso actual registro de asistencias con dispositivo biométrico	9
Figura 2: Bixpe app de fichaje y control de asistencias.....	11
Figura 3: Work Time registro horario legal	12
Figura 4: Proceso Unificado Racional	15
Figura 5: Modelo de equipo de MSF	16
Figura 6: Esquema de trabajo SCRUM	20
Figura 7: Crystal Methodologies.....	22
Figura 8: Dinamic System Development Method	23
Figura 9: Comparación entre metodologías SCRUM vs XP	23
Figura 10: Fases metodología XP	24
Figura 11: Node.js	27
Figura 12: Angular 9.....	28
Figura 13: Express.js.....	29
Figura 14: Oracle Base de Datos	30
Figura 15: Lucidchart.....	30
Figura 16: Proceso general del negocio.....	32
Figura 17: Clasificación de requerimientos no funcionales	38
Figura 18: Diagrama de casos de uso del sistema	42
Figura 19: Modelo - Vista – Controlador.....	50
Figura 20: Diagrama de clases de asistencias.....	51
Figura 21: Diagrama de clases de actividades.....	51
Figura 22: Diagrama de secuencia de asistencias	53
Figura 23: Diagrama de secuencia de actividades.....	54
Figura 24: Notificación de errores en el sistema	55
Figura 25: Modelo físico de base de datos.....	56
Figura 26: Diagrama de despliegue	57
Figura 27: Diagrama de componentes	57
Figura 28. Resultados obtenidos del método de aplicación	66
Figura 29: Certificado de desarrollo software.....	74

Índice de tablas

Tabla 1: Comparación Metodologías Tradicionales vs Metodologías Ágiles.....	19
---	----

Tabla 2: Registrar asistencias mediante biométrico	33
Tabla 3: Descripción de procesos y reglas del negocio.	34
Tabla 4: Descripción requisitos funcionales	37
Tabla 5: Requisitos funcionales por usabilidad	39
Tabla 6: Requisitos no funcionales por confiabilidad	39
Tabla 7: Requisitos no funcionales por eficiencia	39
Tabla 8 Requisitos no funcionales por disponibilidad.....	40
Tabla 9: Requisitos no funcionales por seguridad.....	40
Tabla 10: Requisitos no funcionales por interoperabilidad	40
Tabla 11: Descripción de actores del negocio.....	41
Tabla 12: Descripción del CU - Autenticar Usuario	43
Tabla 13: Descripción del caso de uso - Gestionar Asistencias.....	44
Tabla 14: Descripción del CU - Gestionar Actividades.....	45
Tabla 15: Descripción del CU - Gestionar Aprobaciones	46
Tabla 16: Descripción del CU - Gestionar Reportes	47
Tabla 17: Definición de factores de complejidad	60
Tabla 18: Definición de factores para medir la complejidad	61
Tabla 19: Descripción de los factores técnicos	62
Tabla 20: Descripción de los valores de ponderación de factores técnicos	63
Tabla 21: Descripción de los factores ambientales	64
Tabla 22: Descripción de los valores de ponderación de factores ambientales	64
Tabla 23: Porcentaje de actividades	67
Tabla 24: Identificación de clases de equivalencias crear asistencia.....	68
Tabla 25: Clases válidas para crear asistencias	68
Tabla 26: Clases inválidas para crear asistencias	68
Tabla 27: Matriz de casos de prueba crear asistencias	68
Tabla 28: Identificación de clases de equivalencias crear periodo.....	69
Tabla 29: Clases válidas para crear periodo	69
Tabla 30: Clases inválidas para crear periodos	69
Tabla 31: Matriz de casos de prueba crear periodo	69
Tabla 32: Identificación de clases de equivalencias crear actividades.....	70
Tabla 33: Clases válidas para crear actividad.....	71
Tabla 34: Clases inválidas para crear actividades	72
Tabla 35: Matriz de casos de prueba crear actividad.....	73

Índice de anexos

Anexo 1. Historia de usuario – autenticar usuario	84
Anexo 2. - historia de usuario – gestionar asistencia	84
Anexo 3. Historia de usuario – gestionar actividades.....	84
Anexo 4. Historia de usuario – gestionar aprobaciones	85
Anexo 5. Historia de usuario – gestionar reportes	85
Anexo 6. Tarjeta crc – autenticar usuario.....	85
Anexo 7. Tarjeta crc – gestionar asistencias.....	85
Anexo 8. Tarjeta crc – gestionar actividades.....	86
Anexo 9. Tarjeta crc – gestionar aprobaciones	86
Anexo 10. Tarjeta crc – gestionar reportes	86

RESUMEN

La finalidad de esta investigación es desarrollar un sistema que ayude a la gestión virtual de los procedimientos de registro y control de asistencias, registro de actividades por periodos semanales de máximo 7 días en la Empresa Eléctrica Quito para los empleados de todas las áreas. El sistema fue desarrollado bajo la metodología de Programación Extrema (XP), la cual es flexible a los cambios y nuevos requerimientos, fomenta la comunicación entre el cliente y los desarrolladores además usa buenas prácticas con la finalidad de realizar los procesos de manera más eficaz. Para el diseño de los procesos se ha utilizado Lenguaje Unificado de Modelamiento (UML) y para la diagramación de artefactos se utilizó el aplicativo de diagramación Lucidchart. La función del sistema es de controlar la asistencia de todo el personal que esté realizando teletrabajo o trabajo presencial para que los empleados dejen de utilizar los biométricos que posee la empresa puesto que su uso representa un riesgo potencial para el aumento de contagios de COVID. Por otra parte, el Ministerio de Trabajo, según el acuerdo ministerial ACUERDO-MDT-2020-076-TELETRABAJO que en su artículo 5 indica: todos los empleados del sector público deben emitir un informe semanal de las actividades diarias que realicen en la modalidad de teletrabajo. Considerando estos requisitos la solución que da el sistema, adicional al registro y control de asistencias es el registro de actividades diarias dependiendo el periodo de trabajo, requerimiento solicitado por la Empresa Eléctrica Quito.

Palabras clave: bases de datos, frameworks, metodologías.

ABSTRACT

The aim of this Project is to develop a system that helps with the virtual management of all the procedures related to the attendance record and control as well as the record and control of activities which is made within a maximum of seven days. The system was developed under the Extreme Programming methodology (XP) that is adaptable to changes and new requirements, enhances client and developer communication and uses practices that allow to develop the processes in a more efficient way. To design the processes it has been used a Unified Modeling Language (UML) and for the devices diagraming it was used the diagraming applicative Lucidchart. The system's function is to control the attendance of the personnel who is working remotely or on-site in order that workers stop using the biometric device due to the fact that they represent a potential risk to get infected by COVID. On the other hand, according to the Ministerial agreement ACUERDO-MDT-2020-076-TELETRABAJO on its article 5 states that all public workers must stand a weekly report of their daily activities. Taking into account these requirements, the solution provided by the system and the record and control of attendance is the daily activities record depending on the working period which was demanded by the Empresa Eléctrica de Quito.

Keywords: Databases, Frameworks, Methodologies.

INTRODUCCIÓN

Frente a la emergencia sanitaria COVID 19 el mundo está sufriendo un cambio en cuanto a su forma de convivencia, actualmente estamos viviendo un cambio en nuestra forma de vivir tanto a nivel personal como a nivel laboral. Según la encuesta realizada en abril del 2020 por Deloitte Consulting a 129 empresas ecuatorianas, el 91% de ellas se acogió al teletrabajo. Esto pone a pensar por un momento si es posible realizar de la misma manera las actividades laborales que se han venido llevando a cabo o si la organización deberá llegar a un nivel más alto en cuanto a tecnología se trata. Es así que se ha dado pasos acelerados en el uso de nuevas herramientas tecnológicas.

La Empresa Eléctrica Quito cuenta con varias sucursales localizadas en la ciudad de Quito, la cual tiene como principal agencia la que está en la Av. 10 de Agosto y Bartolomé de las Casas. La empresa cuenta con 1870 empleados en nómina los cuales utilizaban el biométrico para marcar las entradas y salidas de sus jornadas laborales, a los cuales se les garantizaba las medidas de bioseguridad hasta el inicio de la pandemia.

Empresa Eléctrica Quito (EEQ)

La Empresa Eléctrica Quito es la responsable de generar energía eléctrica para la ciudad de Quito, por lo tanto, sus empleados deben contar con las herramientas necesarias para el buen cumplimiento de sus funciones. Debido a la emergencia sanitaria producida por el COVID 19, gran cantidad de empresas migraron sus puestos de trabajo a los hogares de cada empleado con la modalidad de teletrabajo, en las cuales se encuentra la Empresa Eléctrica Quito. Esta situación ha creado la necesidad de desarrollar un sistema de control y monitoreo de la asistencia de todos los empleados para registrar su hora de ingreso y salida; en virtud que por las circunstancias descritas anteriormente el sistema biométrico no se encuentra en servicio. Para complementar un efectivo control se requiere adicionalmente registrar las actividades que realizan los empleados diariamente, adicional a esto “la Institución rectora de políticas públicas de trabajo, empleo y del talento humano del servicio público” (Ecuador, Ministerio del Trabajo, 2020) denominada Ministerio de Trabajo solicita a la Empresa Eléctrica Quito un informe con las actividades de los

trabajadores.

Misión

“Apoyar el desarrollo integral de Quito y su región, suministrando energía limpia y de bajo costo para dinamizar el aparato productivo y mejorar la calidad de vida de los habitantes”. (Ecuador, Empresa Eléctrica Quito, 2020)

Visión

“Ser una empresa eficiente y moderna, líder en el sector eléctrico ecuatoriano y una de las primeras en el contexto latinoamericano”.(Ecuador, Empresa Eléctrica Quito, 2020)

Situación problemática

La presente investigación se centraliza en la Empresa Eléctrica Quito, la cual ha optado por dejar de usar los biométricos de la empresa los cuales son un riesgo para la salud ya que al tener contacto con ellos el empleado está expuesto a un contagio al COVID 19, de esta manera actualmente el control de asistencias de la empresa ha quedado inactivo y a partir de esto se busca solucionar la problemática en cuanto al proceso de control de asistencias y registro de actividades semanales para la empresa.

La manera de contagio del COVID 19 es principalmente a través de las gotículas generadas cuando una persona infectada tose, estornuda o respira. Estas gotículas son demasiado pesadas para permanecer suspendidas en el aire y caen rápidamente sobre las superficies o el suelo, una persona puede infectarse al inhalar el virus si está cerca de una persona con COVID 19 o tras tocar una superficie contaminada la cual puede ser un biométrico.

Formulación del problema

¿Cómo realizar de manera virtual el control y registro de asistencias y actividades de los empleados de la Empresa Eléctrica Quito?

Objetivo general

Desarrollar un sistema informático para el control y registro de asistencias y actividades semanales para la Empresa Eléctrica Quito, utilizando tecnologías web.

Objetivos específicos

- Describir los procesos que implica el ingreso y salida del personal de su jornada laboral con el fin de seleccionar cuales serán automatizados.
- Analizar las metodologías y tecnologías a utilizar para el desarrollo de este proyecto determinar cuáles serán las utilizadas.
- Construir el aplicativo conforme a las exigencias del departamento de Talento Humano de la Empresa Eléctrica Quito con la intención de reemplazar el anterior sistema.
- Verificar dicho sistema informático a partir de pruebas de caja negra de tal manera que se compruebe su operatividad.
- Estimar esfuerzo técnico y costo para el desarrollo del aplicativo a través del método de puntos de función para cuantificar el ahorro que representa el sistema para la Empresa Eléctrica de Quito.

Preguntas de investigación

- ¿Cuáles son los procesos actuales para el registro de ingreso y salida de la jornada laboral de los empleados de la Empresa Eléctrica Quito?
- ¿Qué metodologías y tecnologías existen para el desarrollo del software para el control y registro de asistencias y actividades?
- ¿Cuál es el proceso de la técnica de caja negra para la verificar del software informático?
- ¿Cómo calcular el esfuerzo técnico y costo de la solución informática a través del método de puntos de función?

Justificación de la investigación

Frente a la emergencia sanitaria COVID 19 el mundo se ha visto en la obligación de cambiar su forma de vivir a nivel social y laboral. En el ámbito laboral, múltiples empresas han adaptado sus funciones al Teletrabajo. La empresa Eléctrica de Quito se suma a estos cambios y ha buscado alternativas en cuando a herramientas tecnológicas con el fin de salvaguardar la salud de sus empleados. La modalidad de Teletrabajo requiere un cambio en el registro de asistencias de los trabajadores puesto que el uso del biométrico no es posible, además, es necesario hacer un control de actividades semanales solicitado por el Ministerio de Trabajo para los servidores del sector público. Como respuesta a esta necesidad, este proyecto entregará un software al cual los empleados de puedan acceder desde entornos web, lo que evitará que los empleados realicen presencialmente los procesos antes mencionados.

Viabilidad de la investigación

Para el desarrollo del proyecto se ha considerado un conjunto de recursos humanos, materiales, tecnológicos y bibliográficos. Como recurso humano figura el autor del proyecto. Los recursos materiales utilizados fueron la computadora proporcionada y financiada por la Empresa Eléctrica de Quito, además del servicio de Internet, el cual será financiado por el autor.

Las herramientas de programación de código abierto, tales como: Node.js, Angular.js, y, Nginx son consideradas como herramientas tecnológicas al hacer posible el desarrollo del mencionado software. Debido a que dichas herramientas son gratuitas y de códigos abiertos, no es necesario financiamiento por parte de la Empresa Eléctrica de Quito o del autor de este proyecto lo cual representa un ahorro.

Finalmente, debido a que se debe realizar un estudio previo de las diferentes tecnologías disponibles en materia de desarrollo de software para identificar la que sería utilizada en el desarrollo del software, también se contará con recurso bibliográfico. La información bibliográfica será obtenida en su mayor parte de sitios web orientados a desarrollo web.

Tipo de investigación

Según (Vargas Cordero, 2009)

La investigación aplicada se caracteriza por la búsqueda y aplicación de los conocimientos adquiridos, al mismo tiempo la adquisición de otros para luego implementar y sistematizar la práctica que está basada en la investigación, de acuerdo a esto se obtiene un resultado riguroso, organizado y sistemático de conocer la realidad. Consecuentemente, esta investigación es de tipo aplicada dado que los conocimientos adquiridos antes y durante la ejecución de este proyecto serán aplicados en beneficio de los empleados de la Empresa Eléctrica Quito, para el control de asistencias y registro de actividades por periodos.

Hipótesis de la investigación

La creación de un aplicativo para el control de asistencia y actividades de los trabajadores de la EEQ, que reemplaza al reloj biométrico y lleva un registro semanal de actividades de acuerdo a las exigencias del ministerio de trabajo reduce el riesgo de contagio de COVID puesto que elimina el contacto físico entre personas y posibles focos de infección como el reloj biométrico

Estructura del documento

Este proyecto se divide en cuatro capítulos.

El primer capítulo, consiste en el análisis de los procesos que se llevan a cabo con el fin de llevar un control del inicio y fin de la jornada laboral de los empleados de la Empresa Eléctrica Quito. De igual manera contará con las bases teóricas de las herramientas y tecnologías que se utilizarán en el desarrollo del software.

El segundo capítulo, se describirá los diferentes componentes que deben ser tomados en consideración para el desarrollo del software solicitado por el área de Talento Humano. Adicional a esto serán presentados los requerimientos funcionales y no funcionales que comprenderá el software informático.

El tercer capítulo, estará compuesto por los diagramas y componentes UML los cuales mostrarán la estructura técnica necesaria para el desarrollo del aplicativo.

El cuarto capítulo, detalla cómo será calculado el esfuerzo técnico a utilizarse en el desarrollo del software en cuestión con la finalidad de poder calcular su costo final para su posible venta en el mercado, de igual manera incluirá el proceso de verificación a través de la técnica pruebas de caja negra.

CAPÍTULO I

1. FUNDAMENTACIÓN TEÓRICA

1.1. Breve descripción del contenido del capítulo

En este capítulo se describirá la labor que realiza la Empresa Eléctrica de Quito (EEQ) y su importancia en la ciudad. También se llevará a cabo un análisis de los procesos que se deben desarrollar para llevar un control de asistencia de sus empleados, con el fin de determinar los requisitos con los que el producto de este proyecto debe cumplir. Por otra parte, se analizarán los sistemas que cumplen con funciones similares a las que se necesitan en la EEQ, de esta manera se contará con una referencia para desarrollar el aplicativo que cumplirá con las necesidades de la empresa mencionada. Finalmente, se dispondrá de un estudio sistemático del entorno, herramientas y tecnologías necesarias para la creación del software.

1.2. Objetivos estratégicos de la organización objeto de la investigación

La (Ecuador, Empresa Eléctrica Quito, 2020) tiene entre sus propósitos brindar energía eléctrica al Distrito Metropolitano de Quito y otras áreas, al ser un servicio público, se encuentra comprometida en que este sea de calidad y sin causar daños al medio ambiente, de esta manera favorece el desarrollo económico de la ciudad. Para lograr satisfacer estas necesidades la empresa, junto con el estado, maneja procesos a fin de alcanzar los siguientes objetivos que plantea la Empresa Eléctrica Quito:

- Contar con la cantidad de energía necesaria para sus usuarios y que esta sea sostenible.
- Satisfacción del usuario.
- Personal entrenado, dedicado y motivado.
- Gestión profesional.
- Finanzas sanas.

- Rendición de cuentas y auditoría social.
- Uso y desarrollo de tecnología de punta.

1.3. Objeto de estudio

1.3.1. Flujo actual del proceso

La Empresa Eléctrica Quito tiene aproximadamente 1800 empleados quienes registran su asistencia a diario. Por este motivo, en esta investigación se detallará los procesos actuales que conllevan al registro.

En la Figura 1, se observa el procedimiento actual para realizar el registro de asistencia, este proceso es obligatorio para todos los empleados de la Empresa Eléctrica Quito para mantener el control de asistencias a su jornada de trabajo.

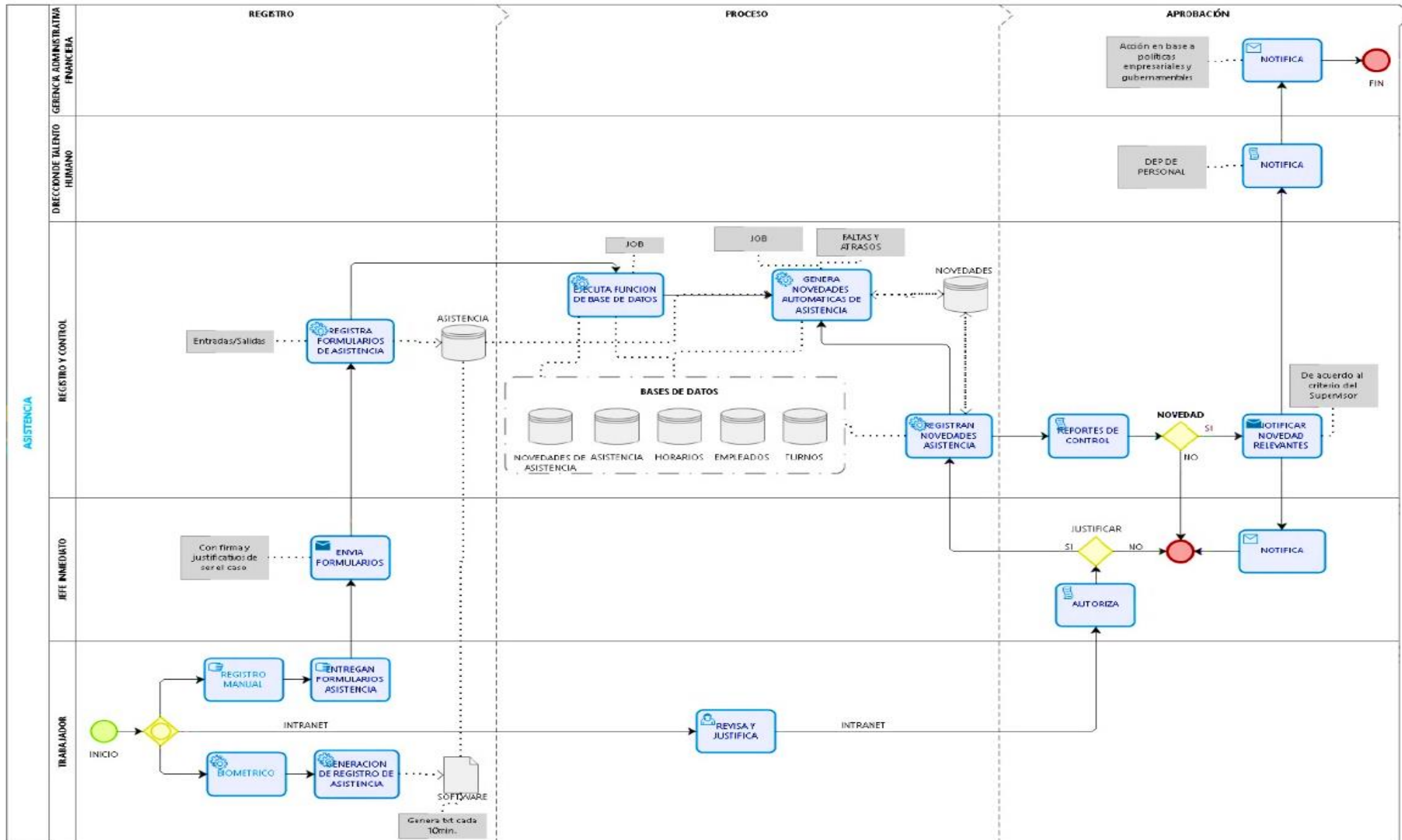


Figura 1: Proceso actual registro de asistencias con dispositivo biométrico
 Fuente (Lucid Software, 2021)
 Elaborado por: José Emilio Fonseca

1.3.2. Análisis crítico de la ejecución de los procesos

El registro de asistencia en la Empresa Eléctrica Quito inicia a través de un dispositivo biométrico utilizado por todo el personal. Cada oficina en la ciudad de Quito dispone de uno de estos mecanismos destinados a controlar la asistencia laboral. La problemática se centra en un solo aspecto: la probabilidad de propagación del COVID 19 al usar estos mecanismos debido a que se requiere contacto directo con estos.

Solución: La virtualización del proceso de registro de asistencias permitirá que los datos de ingreso y salida de personal sean almacenados de la misma forma en que se hacía con el uso del biométrico.

El registro de actividades en la Empresa Eléctrica Quito se da a causa del inicio de la pandemia por COVID 19 a raíz de que la mayoría del personal de la empresa se dedicó a trabajar desde casa. Consecuentemente, el (Ecuador, Ministerio de Trabajo, 2020) según el acuerdo ministerial ACUERDO-MDT-2020-076-TELETRABAJO artículo 5 indicó que todos los empleados del sector público deben emitir un reporte semanal de sus actividades diarias. El problema radica en la inexistencia de un método de registro de actividades.

Solución: El desarrollo de software del proceso de registro de actividades, para que los empleados puedan reportar sus actividades diarias.

1.3.3. Procesos de objeto de automatización

La Empresa Eléctrica Quito actualmente dispone de un proceso de registro de asistencias de tipo manual. Por otro lado, no existe un proceso de registro de actividades. Ambos procedimientos son vitales para el control de las asistencias del personal y para cumplir con una disposición ministerial respectivamente. Por lo que se vuelve prioridad el desarrollo de estos dos procesos y su integración en un solo software.

1.4. Sistemas automatizados existentes vinculados al campo de acción

Luego de haber investigado y analizado datos, se obtiene como resultado que existen varios aplicativos informáticos semejantes al que se desea desarrollar, los

cuales registran información del ingreso y salida de las jornadas laborales de los empleados. Según los datos recolectados de fuentes bibliográficas, existen dos aplicativos específicamente para el manejo de registro de asistencias en España.

1.4.1. Bixpe app de fichaje y control de horario

Según (Bixpe, 2019):

Bixpe es una marca de movilidad empresarial líder en el mercado hispanohablante. Esta marca fue creada por la empresa Abbanza Research Int SL. Por lo tanto, Abbanza es la empresa responsable de Bixpe. Constituida a principio de 2006, Abbanza nace con el objetivo de adaptar las últimas tecnologías del mercado al incremento de la productividad empresarial, ofreciendo siempre un servicio profesional y personalizado.

Bixpe, control de horario es una aplicación web que realiza el control de asistencias, registrando los distintos tipos de pausas que haga el usuario durante su jornada de trabajo. Verifica los ingresos y salidas laborales esta información se almacena en los servidores del proveedor. A este aplicativo se puede acceder únicamente a través de una licencia Premium el pago se lo hace por cada usuario que haga uso del aplicativo. La suscripción a este servicio es muy costosa y la Empresa Eléctrica Quito no estableció un presupuesto financiero para la compra de estas licencias por lo que esta solución no resuelve la problemática planteada.

The screenshot displays the Bixpe web application interface. At the top, it shows 'Control Horario' and the user's name 'Jose'. The main dashboard includes a 'PANEL' section with 'ESTATUS GLOBAL EMPLEADOS' showing 1 employee 'Trabajando', 0 'En pausa', and 0 'Fuera de Jornada'. Below this is a table of 'EMPLEADOS' with columns for 'Estado' and 'Empleado', listing 'Jose' as 'Trabajando'. A right-hand panel shows 'EMPIEZA AQUÍ TU JORNADA' with a timer at '234:24:04' and various control icons. At the bottom, 'ULTIMOS FICHAJES' shows a recent entry for 'Jose' on '27/3/2021 21:50:06' with the note 'Reanuda la jornada'.

Figura 2: Bixpe app de fichaje y control de asistencias
Fuente: (Bixpe, 2019)

1.4.2. Work Time App registro horario legal

“Internalia Group es una compañía con más de 15 años de experiencia en la implementación de soluciones tecnológicas en empresas con el principal objetivo de optimización de sus procesos y mejora de resultados y beneficios.” (Internalia Group, 2019)

Work Time App es un aplicativo móvil de fácil uso, el acceso a la app es único e intransferible, permite el registro de varios tipos de pausas e ingresos y salidas a la jornada laboral y puede registrar la asistencia a través de la firma o reconocimiento facial del empleado. El módulo de administración es a través de un entorno web. El acceso a este servicio es mediante la compra de licencias para cada usuario. Lamentablemente, al ser muy costosa la adquisición, no es posible para la EEQ disponer de este servicio puesto que no posee el presupuesto necesario.

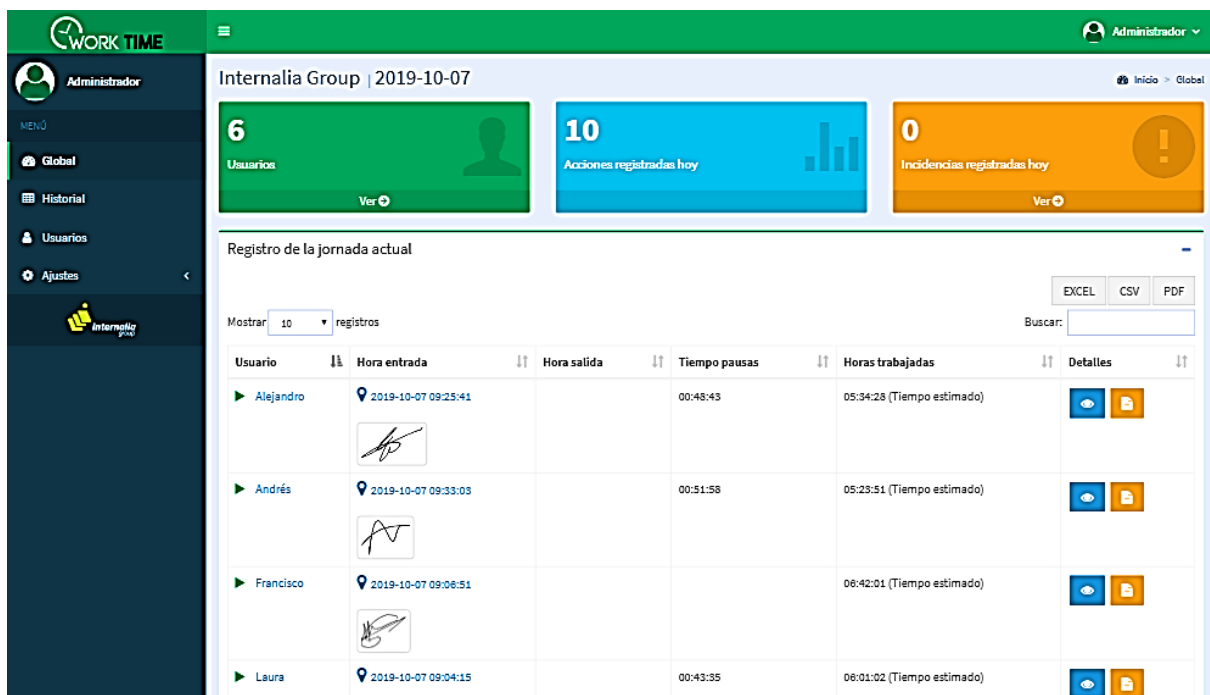


Figura 3: Work Time registro horario legal
Fuente: (Internalia Group, 2019)

1.5. Caracterización de la problemática

A causa de la pandemia por el COVID 19 la Empresa Eléctrica Quito se vio en la necesidad de modificar los procesos para registro y control de asistencias e implementar uno para el registro de actividades. Como solución a esta problemática, es indispensable desarrollar un software de información basado en una arquitectura

Cliente/Servidor en un ambiente web, los empleados de la Empresa Eléctrica Quito tendrán acceso al software para realizar los procesos requeridos. La información que se almacena en el aplicativo deberá guardarse en la misma base de datos que usa el módulo de recursos humanos donde a través de la información generada por el biométrico realizan el control de asistencias. La arquitectura utilizada Cliente/Servidor estará dividida en 3 capas:

- Capa de presentación
- Capa de negocio
- Capa de acceso a datos

1.6. Tendencias y metodologías actuales

1.6.1. Fundamentación de la metodología utilizada

Antes de poder dar una definición de lo es una metodología de software, es necesario comprender el concepto de Ingeniería de software. Resumiendo la investigación de (Maida & Pacienza, 2015), se trata de un campo o una rama de la informática cuya finalidad o propósito es el buen manejo de las herramientas informáticas para desarrollar una solución económica accesible y que se acomode a las exigencias de la compañía del cliente. Para ello se toma en cuenta varios aspectos relacionados con los costos estimados del desarrollo de software, los procedimientos a seguir para el paso a producción y el tiempo invertido para el desarrollo.

La Ingeniería suele encontrarse relacionada con múltiples áreas de conocimiento inherentes a las ciencias de la computación e informática, por ejemplo: desarrollo de Intranet/Internet el cual permite cumplir con las fases necesarias para desarrollar un sistema que se adapte a las necesidades de prácticamente cualquier negocio que así requiera; entre ellos destacan: bancos, logística, medicina, comercio, instituciones educativas, entre otros. A lo anteriormente mencionado se suman otros componentes como: sistemas operativos y la elaboración de compiladores.

En conclusión la intención de la ingeniería de software es la de poner en marcha los principios y métodos que conducen a un desarrollo con buenas prácticas,

funcional y eficaz en todas sus fases desde el comienzo incluso hasta la finalización de su puesta en marcha y mantenimiento.

Una vez establecida la definición de ingeniería de software, es posible comprender el concepto de una metodología y cuál es el uso, para ello hay que conocer que una metodología es usada un conjunto de procesos lógicos para llegar a un fin el cual implica conocimientos y habilidades especializadas.

La metodología representa una fase determinada de un plan de trabajo que tiene su inicio en la teoría y que supone una elección cuidadosa de técnicas particulares y de métodos correspondientes a los objetivos. Es el resultado de la agrupación de métodos usados en una actividad específica de tal manera que sea formalizada y optimizada. Se encarga de establecer los pasos que serán cumplidos, y, a su vez mostrar cómo estos deben ser ejecutados para culminar una asignación.

Dentro de un proyecto de software la metodología es la encargada de diseñar los procedimientos, utilizar técnicas y elegir herramientas las cuales permiten a los programadores desarrollar o implementar nuevos sistemas informáticos. (Rosales López, Salas Bacalla, & Tinoco Gómez, 2010)

Para el desarrollo de software existen dos tipos de metodologías:

1.6.1.1. Metodologías tradicionales

De acuerdo a (Maida & Pacienza, Metodologías de desarrollo de software, 2015) Surgen como respuesta a la necesidad de mejorar el proceso de desarrollo de software, que, en su inicio era completamente artesanal. Para esto fueron tomados como referencia los fundamentos de metodologías existentes en otros contextos y fueron adaptados al contexto de desarrollo de software dando origen al desarrollo dividido en fases secuenciales respondiendo a la necesidad.

Son conocidas como metodologías pesadas debido a que representan un proceso exhaustivo para desarrollar un sistema eficiente. Para esto, se enfocan principalmente en controlar el proceso, asignar roles, y, determinar actividades, artefactos, herramientas y notaciones.

Se enfocan principalmente en manejar una documentación detallada acerca de

todo el proyecto con el fin de cumplir con el plan que fue definido en la primera fase del desarrollo del proyecto. De la mano de esto, va el costo elevado que representa la implementación de un posible cambio y la carencia de una solución eficiente para proyectos en los cuales el entorno es variable. Entre las metodologías tradicionales más populares se encuentran: Rational Unified Process RUP y Microsoft Solution Framework MSF.

RUP

Se trata según lo expuesto por (Figueroa Díaz, Solís, & Cabrera, 2007) de un proceso formal que permite distribuir misiones y responsabilidades a los integrantes de una organización encargada del desarrollo de software. Busca garantizar la calidad de la producción del software de tal manera que satisfaga las exigencias del cliente siempre y cuando el cronograma y el presupuesto sean respetados. Su construcción es atribuida a Rational Software y se combina con la Suite Rational de herramientas. La organización que haga uso de esta metodología cuenta con la posibilidad de adaptarlo y extenderlo para que cumpla con sus necesidades.

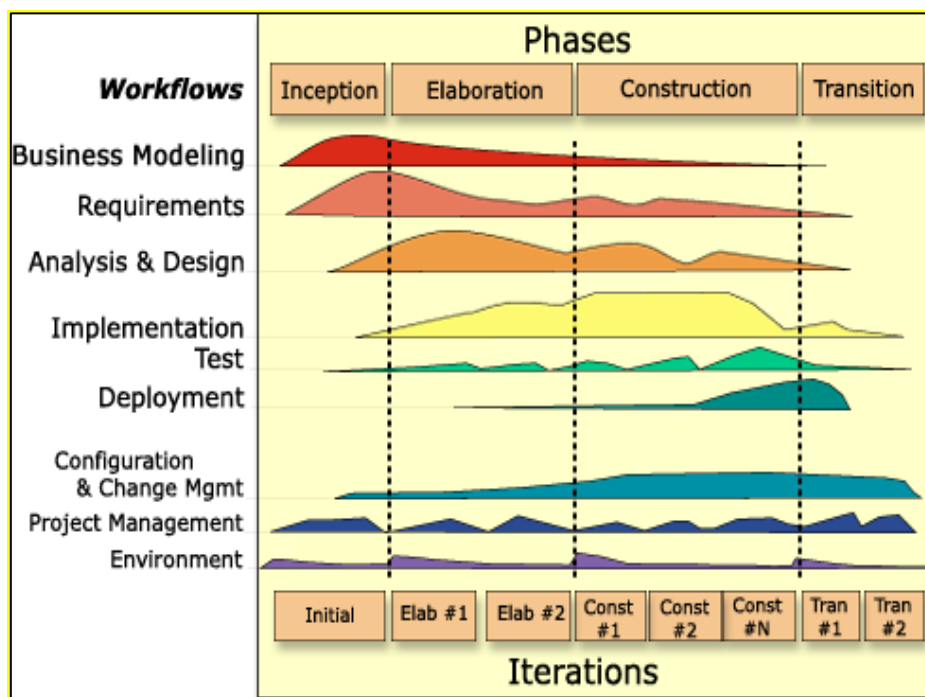


Figura 4: Proceso Unificado Racional
Fuente: (Figueroa Díaz, Solís, & Cabrera, 2007)

De acuerdo a (Maida & Pacienza, 2015), y realizando una síntesis de su trabajo, esta metodología se basa en cuatro fases para su desarrollo:

- **Inicio:** También conocida como la etapa de concepción. Es establecido el alcance que tendrá el proyecto junto con los clientes. Así también son identificados los riesgos relacionados con el proyecto y se crea un plan de las etapas al igual que el de la iteración.

- **Elaboración:** Se crea el borrador del sistema y se elige los casos de uso que harán posible el establecer la arquitectura del sistema.

- **Desarrollo:** O fase de implementación tiene como función completar la funcionalidad del sistema, los requerimientos pendientes, aplican los cambios según los análisis realizados por el usuario y el proyecto.

- **Transición:** En esta fase se ponen en marcha las pruebas necesarias para asegurar que el software funcione adecuadamente para los usuarios finales, se corrigen falencias detectadas durante las pruebas. Los usuarios reciben la capacitación para el uso del software junto al soporte necesario.

MSF

De acuerdo a lo descrito por (Figueroa Díaz, Solís, & Cabrera, 2007):

MSF es un compendio de las mejores prácticas en cuanto a administración de proyectos se refiere. Más que una metodología rígida de administración de proyectos, MSF es una serie de modelos que puede adaptarse a cualquier proyecto de tecnología de información.



Figura 5: Modelo de equipo de MSF
Fuente: (Figueroa Díaz, Solís, & Cabrera, 2007)

Destacando lo más relevante de las etapas se hace referencia a (Figueroa

Díaz, Solís, & Cabrera, 2007), que en resumen dice que:

- **Visión y Alcances:** Es una fase esencial pues determina el éxito del proyecto. En esta fase es necesario que el equipo tenga una perspectiva clara de lo que se aspira entregar al cliente, así mismo debe tener la capacidad de expresarlo de tal manera que el equipo como el cliente se sientan motivados. En esta etapa son seleccionadas las personas que estarán al frente de este proyecto, también se identifican los riesgos de este.

- **Planificación:** El equipo de trabajo se encarga de formular las especificaciones funcionales, diseñar el plan de trabajo y soluciones, estimar costos y tiempos de las versiones del proyecto.

- **Desarrollo:** El equipo construye la mayoría de la documentación y del código aunque es posible que se ejecuten tareas de desarrollo en la etapa de estabilización como consecuencia de los resultados obtenidos en las pruebas. Durante esta etapa también se define la infraestructura a utilizar.

- **Estabilización:** El proyecto es puesto a prueba para verificar su uso y funcionalidad en situaciones de uso cotidianas, en esta etapa el equipo invierte su mayor esfuerzo en solución de errores y preparación de soluciones para la salida a producción.

- **Implantación:** El software es implementado en el hardware requerido en la fase de planificación. El personal de soporte y de operaciones es entrenado para su uso, para finalmente, recibir la aprobación del cliente.

El desarrollo del este modelo de equipos ocurre como compensación de las deficiencias resultantes de la organización jerárquica para trabajar proporcionada a los desarrolladores en los proyectos tradicionales.

Las unidades de trabajo organizadas por este modelo son de tamaño reducido y multidisciplinarias, sus integrantes tienen responsabilidades compartidas y sus destrezas son balanceadas de tal manera, que se mantengan enfocados en el proyecto a su cargo.

Dentro de esta modalidad de trabajo se encuentran seis roles relacionados a

las metas de un proyecto y se responsabilizan por estas. Cada rol puede ser desempeñado por una o varias personas su estructura circular homogénea muestra la ausencia de jerarquía y que el aporte de cada uno de los roles es de igual importancia. Ningún rol puede ser suprimido a pesar de tener diferentes niveles. En el centro de esta estructura se encuentra la comunicación demostrando que ocurre en todas las direcciones y que forma parte de la estructura.

1.6.1.2. Metodologías ágiles

(Maida & Pacienza, 2015), realiza un extenso estudio de estas metodologías, lo que en síntesis las define como metodologías que poseen la capacidad de adaptarse a los posibles cambios que sea necesario hacer. Se caracteriza por la posibilidad que brinda de corregir errores o hacer modificaciones. Así mismo permite que desarrolladores y clientes trabajen en conjunto en el sistema, además su uso es fácil de aprender. Finalmente, no requiere de tanta documentación a diferencia de la metodología tradicional.

Es un enfoque que tiene su origen luego de varios puntos de vista divididos, unos que apoyan las metodologías tradicionales y otros en su contra. Es una solución a los conflictos previamente detallados y tiene como fundamento dos ejes puntuales: retraso de decisiones y la planificación adaptativa. Impulsando el desarrollo a gran escala. Sus ideas principales son:

- Las herramientas y los procesos utilizados pasan a un segundo plano después de los sujetos involucrados en el proceso y sus relaciones.
- Más que elaborar documentación exhaustiva es de vital importancia la creación de un producto software funcional.
- El trabajo colaborativo con el cliente es más importante que la negociación de contratos.
- La importancia de la capacidad para responder a cambios es mayor que el cumplimiento de un plan.

A continuación una comparación entre las metodologías ágiles y tradicionales

Tabla 1: Comparación Metodologías Tradicionales vs Metodologías Ágiles

Metodologías Tradicionales	Metodologías Ágiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.	Basadas en heurísticas provenientes de prácticas de producción de código.
Cierta resistencia a los cambios.	Especialmente preparados para cambios durante el proyecto.
Impuestas externamente.	Impuestas internamente (por el equipo).
Proceso mucho más controlado, con numerosas políticas/normas.	Proceso menos controlado, con pocos principios.
El cliente interactúa con el equipo de desarrollo mediante reuniones.	El cliente es parte del equipo de desarrollo.
Más artefactos.	Pocos artefactos.
Más roles.	Pocos roles.

Fuente: (Arevalo Lizardo, 2011)

Luego de haber analizado los tipos de metodologías y haber realizado una comparación entre las dos, se determina que la metodología ideal para el desarrollo de este proyecto es la ágil. Lo que la hace la metodología más idónea para llevar a cabo el proyecto es, en primer lugar, la facilidad de modificar el software conforme a requerimientos del cliente lo que a su vez permite al desarrollador interactuar de manera activa con el cliente. La demanda de documentación por parte de esta metodología es baja, lo cual hace más flexible el proceso de desarrollo de software.

1.6.1.3. Metodologías ágiles existentes

(Canós, Letelier, & Penadés, 2003, págs. 3-4), enfatizan que a pesar de que las metodologías ágiles comparten sus fundamentos, cada una posee características que las diferencian unas de otras. Es necesario hacer un análisis de las metodologías ágiles y sus características para, de esta manera, identificar la que más beneficios aporte al proyecto.

SCRUM

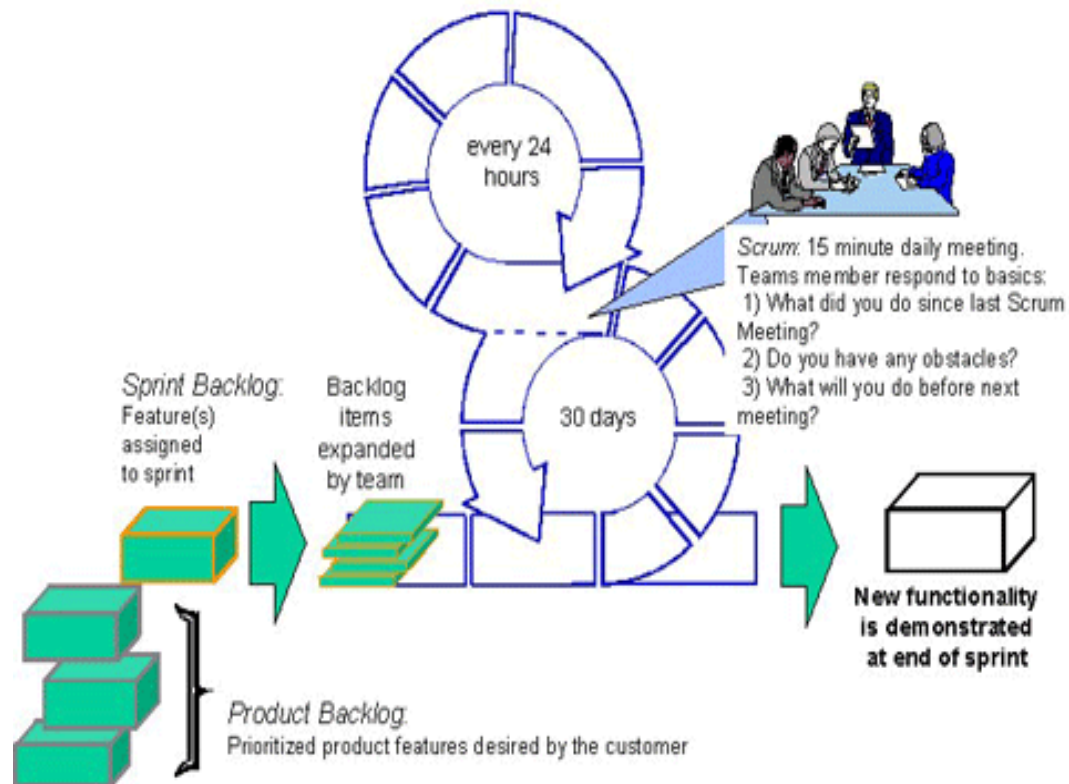


Figura 6: Esquema de trabajo SCRUM
 Fuente: (Figuerola Díaz, Solís, & Cabrera, 2007)

De acuerdo a (Figuerola Díaz, Solís, & Cabrera, 2007):

Scrum es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software. El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza de software ejecutable que incorpora nueva funcionalidad. Las iteraciones en general tienen una duración entre 2 y 4 semanas. Scrum se utiliza como marco para otras prácticas de ingeniería de software como RUP o Extreme Programming.

Scrum se focaliza en priorizar el trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión. Está diseñado especialmente para adaptarse a los cambios en los requerimientos, por ejemplo en un mercado de alta competitividad. Los requerimientos y las prioridades se revisan y ajustan durante el proyecto en intervalos muy cortos y regulares. De esta forma se puede adaptar en tiempo real el producto que se está construyendo a las necesidades del cliente. Se busca entregar software que realmente resuelva las necesidades, aumentando la satisfacción del cliente.

En Scrum, el equipo se focaliza en una única cosa: construir software de calidad. Por el otro lado, la gestión de un proyecto Scrum se focaliza en definir cuales son las características que debe tener el producto a construir (qué construir, qué no y en qué orden) y en remover cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo. Se busca que los equipos sean lo más efectivos y productivos posible.

Scrum tiene un conjunto de reglas muy pequeño y muy simple y está basado en los principios de inspección continua, adaptación, auto-gestión e innovación. El cliente se entusiasma y se compromete con el proyecto dado que ve crecer el producto iteración a iteración y encuentra las herramientas para alinear el desarrollo con los objetivos de negocio de su empresa.

Por otro lado, los desarrolladores encuentran un ámbito propicio para desarrollar sus capacidades profesionales y esto resulta en un incremento en la motivación de los integrantes del equipo.

De acuerdo a (Abellan, 2020), es una metodología con un enfoque para permitir cambios en periodos de tiempo reducidos. Está orientada en dos elementos principales los cuales son:

- La programación del sistema se realiza a través de tareas las cuales son conocidas como sprints que duran, dependiendo el requerimiento, hasta 30 días.
- La necesidad de encuentros entre el equipo desarrollador durante 15 minutos diariamente con el fin de integrar y adecuar los sprints.

Crystal Methodologies

La publicación de (Virender Singh, 2019) afirma que es un grupo de metodologías enfocadas al equipo desarrollador y producción de artefactos limitada. Su componente principal es el equipo cuya prioridad es la constante mejora, debilidades y destrezas entre ellas el trabajo en equipo. Los equipos se clasifican en función de la cantidad de miembros y les son asignados colores.



Figura 7: Crystal Methodologies
Fuente: (Cuofano, s.f.)

Dynamic Systems Development Method (DSDM):

Para el desarrollo de software es necesario definir esquema de trabajo. Sus elementos principales son: iteración, incremento y cooperación entre usuarios y desarrolladores. Sucede en cinco etapas: estudio de viabilidad, modelado funcional, estudio de negocio, diseño o construcción e implementación (Agile Business Consortium Limited, 2021).

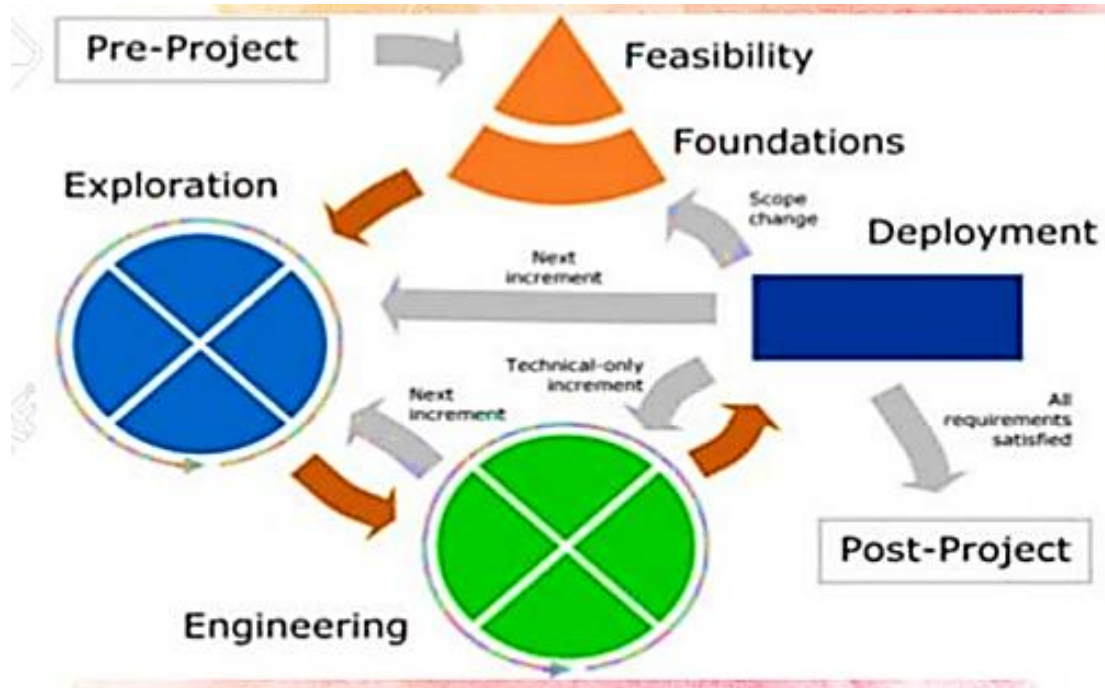


Figura 8: Dinamic System Development Method
Fuente: (Duero, 2018)

Programación extrema (XP)

Se centra en potenciar las relaciones interpersonales del equipo de desarrollo como clave del éxito mediante el trabajo en equipo, el aprendizaje continuo y el buen clima de trabajo. Esta metodología pone el énfasis en la retroalimentación continua entre cliente y el equipo de desarrollo y es idónea para proyectos con requisitos imprecisos y muy cambiantes (Calvo, 2018)

SCRUM	XP
Es una metodología de desarrollo ágil basada en la administración del proyecto.	Es una metodología de desarrollo que está más centrada en la programación o creación del producto.
Cada miembro de del equipo trabaja de forma individual.	Los miembro del equipo programan en parejas.
Las ietraciones de entrega son de 1 a 4 semanas.	Las ietraciones de entrega son de 1 a 3 semanas
Al finalizar un Sprint, las tareas del Sprint Backlog que se hayan realizado y que el Product Owner (propietario del producto) haya mostrado su conformidad ya no se retoca. Si funciona y está bien, se aparta y a otra cosa.	Las tareas se van terminando aunque son susceptibles de ser modificadas durante el transcurso de proyecto, incluso, después de que funcionen correctamente.
Trata de seguir el orden de prioridades que marca el Product Owner en el Sprint Backlog pero puede cambiarlo si es mejor para el desarrollo de la tareas.	El equipo de desarrollo sigue estrictamente el orden de prioridad de las tareas definido por el cliente.

Figura 9: Comparación entre metodologías SCRUM vs XP
Fuente: (Isla Visual, 2014)

En función de la comparativa previamente presentada, la metodología XP es la indicada para el desarrollo del proyecto porque cumple con una serie de requerimientos que se apegan a los del presente proyecto tales como:

- Se centra en el desarrollo del software o creación del producto

- La programación del sistema se da de forma individual o en parejas.
- Es posible realizar actualizaciones a las actividades a lo largo del proyecto, a pesar de que estas funciones adecuadamente.
- Las tareas deben ser cumplidas en el orden indicado de acuerdo a la prioridad que el cliente requiera.

1.6.2. Metodología ágil XP (Programación Extrema)



Figura 10: Fases metodología XP
Fuente: (Calvo, 2018)

El rendimiento óptimo del programa se encuentra ligado al equipo de programadores, los cuales se encuentran en contacto constante con el cliente. Esta metodología permite realizar cambios a la brevedad conforme a la solicitud del cliente con la posibilidad de intervenir en cualquiera de las fases sin importar que esta ya haya culminado.

Las fases de acuerdo a una síntesis de la publicación de (Castillo, Figueroa, & Sevilla, s.f.), son:

1. Planificación

Está orientada al contacto entre el usuario solicitante y el o los desarrolladores encargados de crear el software, la cantidad de reuniones que se mantenga influye directamente para el resultado de un mejor producto. En esta etapa se considera varios factores como el orden de atención para las tareas, la tarifa de cada proceso,

los obstáculos que se pueden presentar para el desarrollo del software y un cronograma con fechas tentativas. Dentro de la etapa se tendrá 6 elementos que se deben cumplir para el desarrollo del proyecto:

- **Historias de usuario:** Estas son el primer paso para el desarrollo de cualquier proyecto, tienen la misma finalidad que los casos de uso con la diferencia de que existen un par de líneas escritas por el cliente en un lenguaje no técnico. Las historias de usuario correspondientes a este proyecto se encuentran anexadas.
- **Planificación de lanzamientos:** En este punto después de haber definido las historias de usuario es necesario elaborar un plan de publicaciones para las versiones del programa juntamente con fechas tentativas.
- **Iteraciones:** Cualquier proyecto que se haya desarrollado con la metodología XP se divide en iteraciones con alrededor de un periodo de 3 semanas.
- **Velocidad del proyecto:** Representa la rapidez con la que se pueda desarrollar las historias de usuario.
- **Programación en pareja:** En la metodología se recomienda la programación en parejas ya que mejora la calidad y aumenta la productividad del software.
- **Reuniones diarias:** Son interacciones que se realiza entre los desarrolladores y el cliente para resolver problemas de forma conjunta.

2. Diseño:

El producto de esta etapa es una versión preliminar del aplicativo el cual será obtenido a partir de un código sencillo. Es en este punto que se crean las tarjetas CRC (Clase-Responsabilidad-Colaboración) en el caso de software de objetos. Para esta etapa hay que tener en cuenta 6 elementos:

- **Diseños simples:** Hay que tomar en consideración asegurarse de usar diseños sencillos y simples.
- **Glosario de terminología:** Para reusar el código es necesario definir una terminología especificada de los nombres de clases y métodos lo que ayudará a entender el diseño de mejor manera.
- **Riesgos:** En el caso de que ocurra algún problema XP sugiere usar una pareja de desarrolladores para que investiguen y minimicen los riesgos de un problema.
- **Funcionalidad extra:** Jamás se debe crear una funcionalidad extra al programa.
- **Refactorización:** Mejorar y actualizar la estructura de los códigos ya diseñados sin cambiar operatividad.
- **Tarjetas C.R.C:** Representan objetos, la clase de la que es miembro ese objeto,

además de describir las responsabilidades de los objetos. Las tarjetas C.R.C correspondientes a este proyecto se encuentran anexadas.

3. Codificación

Esta fase requiere un trabajo colaborativo e integrado en el cual los miembros del equipo, y, de ser necesario las parejas son intercambiadas. Lo que se busca es un código que cualquier desarrollador pueda comprender con el fin de que los desarrollos tengan un esquema ordenado.

4. Pruebas:

Debido a que no son proyectos extensos es necesario correr pruebas periódicamente. Así también el cliente tiene la capacidad de generar nuevas pruebas y revisar las versiones que se va desarrollando. Se consideran tres tipos de pruebas:

- **Pruebas Unitarias:** Es un área definida por el programador en su entorno de trabajo en el cual realiza pruebas de la función desarrollada, pero para ello es necesario definir lo siguiente: establecer el ambiente de ejecución para la prueba, invocar al método desarrollado, definir los recursos y revisar si el resultado es el esperado.
- **Pruebas de Aceptación:** La ejecución de esta prueba ratifica las pruebas unitarias realizadas anteriormente al igual que el funcionamiento del producto final.
- **Pruebas de corrección de errores:** En caso de encontrar un error es obligatorio escribirlo y reportar inmediatamente tanto al usuario final como al equipo de desarrollo antes de intentar solventar el inconveniente. De esta manera se mantiene un historial de errores del cual los implicados están al tanto.

5. Lanzamiento

Es la última fase luego de haber desarrollado, aprobado y verificado las historial de usuario en virtud de las exigencias del cliente. Lo que se entregará es un producto útil y funcional.

1.7. Fundamentación del entorno de desarrollo

Se entiende como entorno de desarrollo al conjunto de lenguajes de programación utilizados, gestor de base de datos y tecnologías utilizadas para desarrollar. En el siguiente apartado se caracterizará cada uno de los elementos a ser

utilizados en este proyecto. Es necesario señalar que las tecnologías han sido seleccionadas debido a que son los utilizados por la Empresa Eléctrica Quito.

1.7.1. Node.js

Node.js es un entorno de código abierto que se ejecuta de lado del servidor (back-end) en tiempo de ejecución multiplataforma la cual se ejecuta de manera asíncrona con un motor basado en el motor V8 de Google y tiene una arquitectura orientada a eventos. (Equipo Geek, 2020)



Figura 11: Node.js
Fuente: (Villatoro, 2018)

Node.js contribuye al presente proyecto con las siguientes utilidades:

- Funciona con un modelo de evaluación de un único hilo de ejecución, por lo que usa entradas y salidas asíncronas las cuales se ejecutan concurrentemente en un número de hasta miles sin comprometer los recursos de hardware.
- Muy rápido: al estar construido en el motor JavaScript V8 de Google Chrome, la biblioteca Node.js es muy rápida en la ejecución de código.
- Sin almacenamiento en búfer: las aplicaciones Node.js nunca almacenan en búfer ningún dato. Estas aplicaciones simplemente generan los datos en fragmentos

El mecanismo de eventos ayuda al servidor a responder sin bloqueos y hace que el servidor sea altamente escalable en comparación con los servidores tradicionales que crean hilos limitados para manejar las solicitudes. Node.js utiliza un solo programa de

subprocesos y el mismo programa puede proporcionar servicio a un número mucho mayor de solicitudes que los servidores tradicionales como el Servidor Apache HTTP (Equipo Geek, 2020).

1.7.2. Angular

Haciendo referencia a la publicación de (Oriol, 2020), se sabe que Angular es un framework con licencia MIT la cual es de código abierto desarrollado en TypeScript es cual se usa para desarrollar aplicativos de tipo web o móviles en una sola página. La fortaleza del uso de Angular está en el potencial que poseen sus componentes para realizar páginas dinámicas.



Figura 12: Angular 9
Fuente: (Oriol, 2020)

Angular proporciona las siguientes utilidades en el presente proyecto:

- Transforma las plantillas en código optimizado para el uso de las máquinas virtuales de JavaScript.
- Permite crear interfaces de usuario con una sintaxis rápida, potente y sencilla.
- Gracias al enrutador de componentes que maneja Angular el aplicativo se carga de manera inmediata, esto a su vez proporciona a que los usuarios solo carguen el código o componente que van a utilizar para el proceso.
- Permite crear animaciones complejas con muy poco código y con una alta utilidad (Oriol, 2020).

1.7.3. Framework Express.js

Express.js es un framework rápido y flexible de Node.js, el cual ayuda a crear APIs o aplicativos webs y que proporciona un grupo de herramientas para el manejo de direccionamientos o rutas, integraciones con bases de datos, archivos dinámicos o estáticos, middlewares y control de logs. Una de las características principales es el ajuste de que puerto se va a utilizar para consumir una aplicativo. Cabe destacar que express.js es de uso gratuito y tiene mucha documentación como soporte para realizar consultas (López Sosa, 2019).

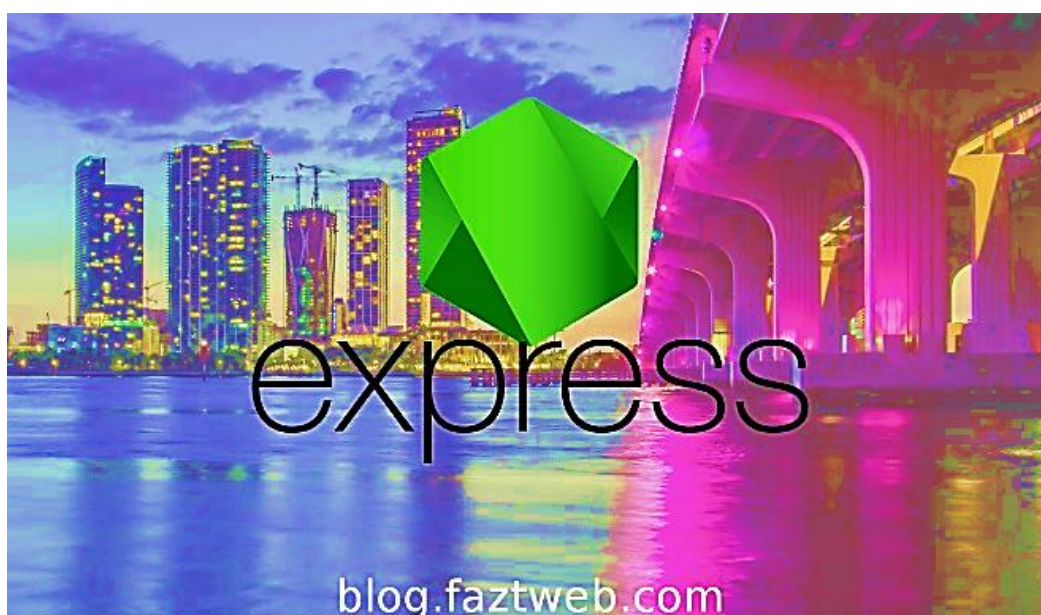


Figura 13: Express.js
Fuente: (Fazt Blog, 2017)

1.7.4. Oracle

En una síntesis de la página de (Oracle, 2021) se sabe que es un sistema de gestión de base de datos principalmente usado por medianas y grandes empresas. Las consultas hacia la base de datos se realizan a través de lenguaje estructurado SQL. Fue diseñado con el propósito de gestionar grandes cantidades de información reduciendo riesgos de pérdida a un costo bajo. Entre sus principales características se encuentran:

- Los datos son visualizados en formato filas/columnas.
- Interfaz gráfica de usuario amigable y sencillo para usar.

- Controla la entrada a los datos con tecnología innovadora.



Figura 14: Oracle Base de Datos
Fuente: (López Roldán, 2020)

1.7.5. Lucidchart

Es una herramienta para el análisis, modelado y diseño de tipo UML (Unified Modeling Language), la cual se puede instalar en la plataforma de Windows o vía web. Está facilitada la creación de software y ayuda a tener alta categoría de documentación. Para este proyecto se utilizará Lucidchart para el diseño de los requerimientos.



Figura 15: Lucidchart
Fuente: (Lucid Software, 2021)

1.8. Conclusiones del capítulo

Al finalizar este capítulo, se cuenta con información respecto a las actividades de la Empresa Eléctrica de Quito, su relevancia para la ciudad y objetivos. Así mismo, se ha explicado cual es la situación actual en cuanto a procesos de registro y control de asistencias y sus deficiencias, junto con las necesidades para dar solución al problema que surgió a partir de la pandemia por el COVID-19. Ha sido contextualizada la importancia de llevar un control de actividades semanales realizadas por el

personal de la mencionada empresa y determinados los requisitos con los que debe cumplir el software que hará posible este proceso el mismo que cumplirá con dos funciones:

- Registro y control de asistencias
- Registro de actividades del personal.

Junto con esto, han sido descritas las tecnologías, herramientas y metodologías que existen en materia de desarrollo de software para posteriormente seleccionar las que serán empleadas para la creación del software que dará solución al problema descrito anteriormente.

CAPÍTULO II

2. MODELO DE NEGOCIO Y REQUISITOS

2.1. Breve descripción del contenido del capítulo

Este capítulo comprenderá de una descripción detallada del modelo de negocio, al igual que de los requerimientos funcionales y no funcionales solicitados por el área de Talento Humano de la Empresa Eléctrica Quito, a partir de esto se desarrollarán los casos de uso respectivos. Se mostrará y explicará cada fase de los protocolos de que cumplirá el sistema a desarrollar, dando cumplimiento a las peticiones recolectadas a los funcionarios solicitantes. Se presentará también los actores del proceso junto con los roles que desempeñaran en las distintas etapas.

2.2. Análisis de mejora de procesos

Actualmente el registro de entradas y salidas para las jornadas laborales de los empleados de la Empresa Eléctrica Quito se realizan por medio de los relojes biométricos de huella digital, esta información se pasa a un software que ya se encuentra desarrollado perteneciente al departamento de Talento Humano los cuales verifican la información enviada por los biométricos para realizar los procedimientos que les corresponde y así llevar un control del registro de las asistencias de todo el personal de la Empresa.



Figura 16: Proceso general del negocio
Elaborado por: José Emilio Fonseca

Lo que ahora procede es un desglose del proceso que, en la actualidad, se lleva a cabo en la EEQ para controlar las asistencias de su personal.

Tabla 2: Registrar asistencias mediante biométrico

Proceso	EEQ-SRCAA-PN1 Registro de asistencias
Subproceso	N/A
Objetivo	La información referente a asistencias del empleado queda registrada en la base de datos el acceso y control por parte del departamento de Talento Humano.
Alcance	El procedimiento tiene su origen en la necesidad de llevar un control de asistencias de los empleados de la EEQ. De tal manera que se pueda calcular el total de horas laboradas y realizar el pago al personal.
Pre-condiciones	Ser Parte de la nómina
Disparador	Tener registrada la huella dactilar en el sistema biométrico
Entradas	Ingreso manual de huella dactilar en el sistema biométrico. Registro de ingreso del empleado
Trabajadores	Registro de inicio de hora de almuerzo del empleado Registro de fin de hora de almuerzo del empleado Registro de salida del empleado
Actores	Trabajadores de las distintas áreas de la EEQ, Departamento de Talento Humano
Base Legal	Empleado EEQ
Recursos	Reglamento Interno EEQ, Reglamento Ministerio de Trabajo
Frecuencia	Dispositivo Biométrico
Salidas	Diaria
Post-Condicion	Reporte mensual de asistencias
Post-Condicion	N/A
ACTOR	ACTIVIDAD
	INICIO
1. Empleado	Registra huella dactilar en sistema biométrico
	Recibe información del registro de asistencia
2. Dep. Talento Humano	Verifica la información en la base de datos del Talento a través del software manejado por el área.

Elaborado por: José Emilio Fonseca

2.3. Reglas del negocio a considerar

Para establecer los procesos se toma como referencia el reglamento interno de la EEQ el cual cuenta con las políticas del funcionamiento de la empresa. Dentro del mismo documento se encuentran especificadas las responsabilidades de sus trabajadores junto con sus derechos. Este conjunto de normas son de vital importancia para la interacción armoniosa entre empleados.

Tabla 3: Descripción de procesos y reglas del negocio.

DESCRIPCIÓN DEL PROCESO	ACTORES	REGLA DE NEGOCIO
GESTIÓN DE ASISTENCIAS		
El empleado registra su asistencia de entrada a la jornada laboral.		
El empleado registra su asistencia de entrada al lunch.		
El empleado registra su asistencia de salida del lunch.		
El empleado registra su asistencia de salida de la jornada laboral.		
En caso de que biométrico se encuentre fuera de servicio, la asistencia no haya sido registrada o por inasistencia.	Empleado	La asistencia de inicio y salida se debe registrar obligatoriamente todos los días de la jornada laboral, al igual que las jornadas extraordinarias.
El empleado debe llenar un formulario con su firma para registrar su asistencia	Empleado	El periodo de tolerancia para registrar la asistencia es de máximo 10 minutos, cualquier registro realizado luego de ese tiempo será considerado atraso.
El empleado envía el formulario a su empleado jefe para la aprobación respectiva	Jefe	No es obligatorio el registro de ingreso o salida al lunch.
El empleado jefe ingresa la aprobación en el software de justificaciones de faltas	Supervisor de Talento Humano	
El supervisor de Talento Humano realiza la revisión de novedades y aprueba según el reglamento interno.		

Elaborado por: José Emilio Fonseca

2.4. Requisitos de negocio

Un requerimiento es la condición o capacidad que debe tener un sistema, producto, servicio o componente para satisfacer un contrato, estándar, especificación, u otros documentos formalmente establecido. Se deben definir en la fase inicial junto con los implicados para obtener una visión completa y compartida de todas las piezas y poder priorizar en base a los objetivos del proyecto. Los requerimientos no te indican que diseño debe tener tu producto o como desarrollarlo. Te indican que features, funciones y contenidos se espera que tenga, y como deben los usuarios interactuar con él (Saraclip, 2017)

Los requerimientos del sistema se encuentran clasificados en base a lo que estos van a describir. Se identifican dos tipos:

- Requerimientos funcionales
- Requerimientos no funcionales

2.4.1. Especificación de los requisitos funcionales

Los requerimientos funcionales de un sistema describen lo que el sistema realmente debe de hacer o ejecutar. Estos requerimientos dependen del tipo de software que se vaya a desarrollar, de los posibles usuarios del software y del enfoque general tomado por la organización al redactar requerimientos. Adicional a esto también pueden ser interacciones con otros sistemas, respuestas automáticas o procesos predefinidos (Noren, 2020)

Es menester tomar en cuenta varios pasos para definir un requerimiento funcional. Una vez recolectados todos los requerimientos que tiene que cumplir el software, se debe definir la prioridad y el orden con el que se va a cumplir cada uno de estos, hay que tener presente que el resultado de identificar correctamente estos requisitos será el alcance de la solución del proyecto. En base a esto se determina que existen 3 tipos de prioridades para un requerimiento funcional:

- **Alta:** El sistema debe ajustarse a estos obligatoriamente. Para que se considere de alta prioridad se necesitan 8 o más transacciones y 5 o más objetos del software a los cuales se ejecutan operaciones de consulta, creación, modificación o eliminación. El requerimiento debe ser programado por un desarrollador ya que es una función personalizada tanto en la información requerida como en la interfaz que se muestra al usuario final.

- **Media:** Son requerimientos deseables por el usuario. Para que se considere de media prioridad se necesitan de 5 a 8 transacciones y de 3 a 5 objetos del software a los cuales se ejecutan operaciones de consulta, creación, modificación o eliminación. En cuanto a la interfaz de usuario muchas veces son pantallas que se conectan a través de servicios web, XML, aplicativos terceros o correo electrónico.

- **Baja:** Son requerimientos deseables por el usuario pero que pueden ser borrados sin afectar a la regla del negocio. Para que se considere de baja prioridad se necesita de 1 a 3 transacciones y de 1 a 2 objetos del software a los cuales se ejecutan operaciones de consulta, creación, modificación o eliminación. Para la interfaz de usuario se maneja solo componentes estándar y no se asocia con otras pantallas del software.

Una vez explicadas las prioridades acerca de los requerimientos, a continuación se detallará los requisitos funcionales a desarrollar:

Tabla 4: Descripción requisitos funcionales

ID	Requisito	Descripción	Complejidad	Entidades
SRCAA-RF1	Autenticar Usuario	El requerimiento empieza al momento que el usuario ingresa al software, se provee del usuario y clave y el sistema realiza la validación en caso de que sea el correcto permite el ingreso a la página principal, caso contrario muestra un mensaje que indica que la clave es errónea. Para ejecutar esta validación se usará el LDAP el cual es un protocolo del servicio Active Directory institucional de la Empresa Eléctrica Quito. Finalmente en el caso de que no se pueda acceder al servicio como plan de contingencia se realizará contra la base de datos local.	Media	Usuario
SRCAA-RF2	Gestionar Asistencias			
SRCAA-RF2.1	Crear nueva asistencias de entrada	El requerimiento inicia cuando el empleado requiere crear una nueva asistencia de entrada. El sistema verificará que el usuario se encuentre logueado y si tiene esta precondición la asistencia se creará de manera exitosa.	Baja	Asistencia
SRCAA-RF2.2	Crear nueva asistencias de salida	El requerimiento inicia cuando el empleado requiere crear una nueva asistencia de salida. El sistema verificará que el usuario se encuentre logueado y si tiene esta precondición la asistencia se creará de manera exitosa.	Baja	Asistencia
SRCAA-RF2.3	Consultar asistencias	El requerimiento inicia cuando el empleado desea realizar la consulta de sus asistencias en un rango determinado de fechas. Para ello el usuario debe seleccionar obligatoriamente dos fechas para realizar la consulta, en caso de que solo se ingrese una fecha la consulta no podrá ser llevada a cabo, caso contrario mostrará un listado de la consulta de las fechas y horas de las asistencias realizadas por el usuario.	Baja	Asistencia
SRCAA-RF3	Gestionar Actividades			
SRCAA-RF3.1	Listar Periodos	El requerimiento inicia cuando el empleado desea visualizar todos los periodos creados por él. Para esto se realizará una búsqueda sobre los periodos registrados en el sistema según el rol de usuario que le pertenece al empleado. En caso de que la búsqueda sea exitosa se listará todos los periodos en el sistema.	Baja	Actividades
SRCAA-RF3.2	Crear periodo	El requerimiento inicia cuando el empleado desea crear un nuevo periodo. El sistema verificará que el usuario se encuentre logueado y si tiene esta precondición el periodo se creará de manera exitosa.	Baja	Actividades
SRCAA-RF3.3	Eliminar Periodo	El requerimiento inicia cuando el empleado desea eliminar un periodo determinado, para esto se debe seleccionar el periodo que se requiere eliminar, en caso de que existan actividades realizadas en este periodo mostrará un mensaje el cual indique que en ese periodo existen actividades creadas y si el usuario acepta eliminar el periodo se eliminará sin importar las actividades que se encuentren dentro de ese periodo.	Baja	Actividades
SRCAA-RF3.4	Buscar periodo	El requerimiento inicia cuando el empleado desea realizar una búsqueda sobre los periodos registrados en el sistema. Esta búsqueda se podrá realizar a través de un criterio definido de fechas de inicio, fechas fin o estado del periodo. En caso de que la búsqueda se realice sin inconvenientes se podrá visualizar un listado con la información requerida.	Baja	Actividades
SRCAA-RF3.5	Enviar periodo a Jefatura	El requerimiento inicia cuando el empleado desea enviar el periodo a la jefatura asignada para que realice la aprobación de sus actividades diarias creadas por periodos semanales. En caso de que el periodo se envíe correctamente internamente pasará a un estado de ENVIADO y el empleado ya no podrá visualizar ese periodo solo en el caso de que la jefatura decida devolverlo.	Baja	Actividades
SRCAA-RF3.6	Crear actividad	El requerimiento inicia cuando el empleado desea crear una nueva actividad. El sistema verificará que el usuario se encuentre logueado y que se haya creado un periodo, si tiene estas precondiciones la actividad se creará de manera exitosa.	Baja	Actividades
SRCAA-RF3.7	Eliminar actividad	El requerimiento inicia cuando el empleado desea eliminar una actividad determinada, para esto se debe seleccionar la actividad que se requiere eliminar, mostrará un mensaje el cual indique si está seguro de eliminar la actividad y si el usuario acepta eliminar la actividad se eliminará.	Baja	Actividades
SRCAA-RF3.8	Buscar actividad	El requerimiento inicia cuando el empleado desea realizar una búsqueda sobre las actividades registradas en el sistema. Esta búsqueda se podrá realizar a través de un criterio definido de fechas de creación, descripción, número de referencia, producto entregable, observación, etapa o estado de la actividad. En caso de que la búsqueda se realice sin inconvenientes se podrá visualizar un listado con la información requerida.	Baja	Actividad
SRCAA-RF3.9	Listar actividades	El requerimiento inicia cuando el empleado desea visualizar todas las actividades creados por él en un determinado periodo. Para esto se realizará una búsqueda sobre las actividades registrados en el sistema según el rol de usuario que le pertenece al empleado y el periodo de esa actividad. En caso de que la búsqueda sea exitosa se listará todas las actividades en el sistema.	Baja	Actividades
SRCAA-RF3.10	Modificar actividad	El requerimiento inicia cuando el empleado desea modificar una actividad determinada, por lo que es necesario seleccionar la actividad que se desea modificar y en caso de que todo este correcto se modificará y se podrá visualizar en el listado de actividades.	Baja	Actividades
SRCAA-RF4	Gestionar Aprobaciones			
SRCAA-RF4.1	Listar periodos recibidos	El requerimiento inicia cuando el empleado jefe desea visualizar los periodos enviados por su empleado a cargo, por lo que es necesario que el empleado a cargo haya enviado el periodo a su empleado jefe. Se mostrará un listado de todos los periodos que han sido enviados de todos sus empleados a cargo.	Baja	Actividades
SRCAA-RF4.2	Aprobar periodo recibido	El requerimiento inicia cuando el empleado jefe desea aprobar las actividades que contiene el periodo enviado por su empleado a cargo, por lo que es necesario que el empleado jefe vaya aprobando cada actividad hasta aprobar todas y que no haya ninguna más por aprobar. En caso de que el empleado jefe apruebe todas las actividades el periodo automáticamente pasará a gestión del área Talento Humano.	Baja	Actividades
SRCAA-RF4.3	Devolver periodo recibido	El requerimiento inicia cuando el empleado jefe desea devolver alguna actividad que contiene el periodo enviado por su empleado a cargo, por lo que es necesario que el empleado jefe devuelva la actividad seleccionada. En caso de que el empleado jefe devuelva una actividad a su empleado a cargo automáticamente pasará de la bandeja del jefe al empleado a cargo.	Baja	Actividades
SRCAA-RF5	Gestionar reportes			
SRCAA-RF5.1	Exportar reporte actividades por periodo	El requerimiento inicia cuando el empleado desea descargar un reporte en formato Excel de sus actividades por periodos, se debe seleccionar el periodo que el empleado necesita descargar. En caso de que se exporte se deberá descargar las actividades creadas por el empleado en ese periodo.	Baja	Actividades
SRCAA-RF5.2	Exportar reporte actividades por parámetros	El requerimiento inicia cuando el empleado jefe desea descargar un reporte en formato Excel de las actividades que su empleado a cargo tiene creadas, enviadas o aprobadas, para ello se debe ingresar los parámetros de consulta para descargar este reporte personalizado, el empleado jefe debe ingresar el estado de las actividades, fecha de inicio, fecha fin y seleccionar el empleado de cual desea descargar las actividades. En caso de que se exporte se deberá descargar un reporte personalizado con las actividades del empleado seleccionado.	Baja	Actividades

Elaborado por: José Emilio Fonseca

2.4.2. Especificación de los requisitos no funcionales

Según la información recopilada de la página web (PMOinformatica, 2015), se entiende que estos requerimientos son el resultado de las limitaciones o necesidades específicas del cliente. Entre las principales causas se encuentran: falta de recursos financieros, normas institucionales, la demanda de la interoperabilidad con diferentes sistemas de hardware o software, y, normativas de confidencialidad. Los requerimientos no funcionales son los que indican diferentes criterios para medir la calidad de un servicio de tecnología informática. La siguiente figura muestra la clasificación de los requerimientos no funcionales:

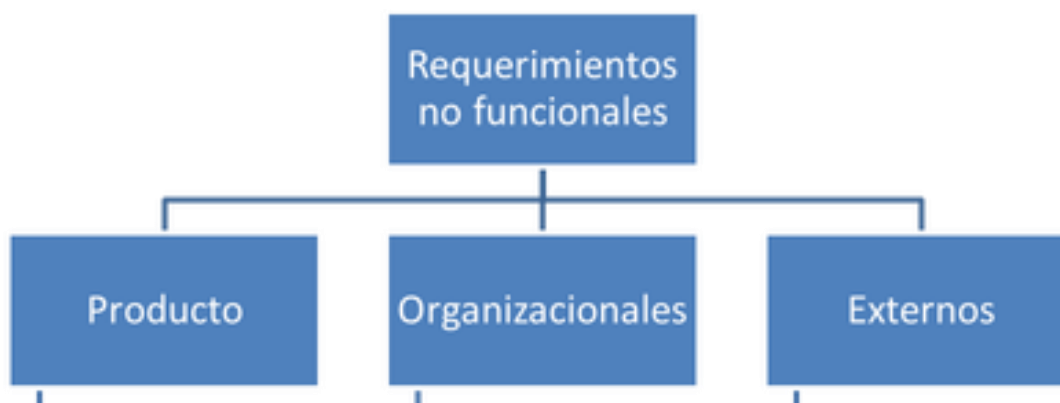


Figura 17: Clasificación de requerimientos no funcionales
Fuente: (PMOinformatica, 2015)

Como se observa en la imagen dentro de los tipos de requerimientos funcionales se encuentra enfocados al: producto, organizacionales y externos. En esta ocasión se detallará los requerimientos no funcionales enfocados al producto que son:

2.4.2.1. Usabilidad

Se refiere de acuerdo con (Serrano Mascaraque, 2009), a la característica del sistema que le permite ser manipulado por el usuario. A su vez, involucra los elementos que harán a la interfaz agradable en determinadas situaciones.

Tabla 5: Requisitos funcionales por usabilidad

ID	Descripción	Prioridad
	El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 4 horas.	Alta
U2	Es necesario que la interfaz se adapte para que pueda ser visualizada en distintos dispositivos. (Responsive)	Alta
U3	El tiempo de respuesta no debe superar los 10 segundos.	Alta

Elaborado por: José Emilio Fonseca

2.4.2.2. Confiabilidad

“En caso de presentar errores, el sistema debe tener la capacidad de seguir funcionando. Además de que las operaciones deben ser transaccionales”. (Ortíz, 2021)

Tabla 6: Requisitos no funcionales por confiabilidad

ID	Descripción	Prioridad
C1	Los mensajes de error deben brindar al usuario instrucciones claras acerca de cómo proceder en caso de presentarse alguna excepción.	Alta
C2	Una descripción de las razones por las cuales ocurren los errores debe ser generada y presentada siempre y cuando el usuario lo quiera.	Media
C3	El idioma en el cual los errores deben ser presentados es el español.	Alta

Elaborado por: José Emilio Fonseca

2.4.2.3. Eficiencia

Esta característica muestra la relación que existe entre el esfuerzo y los recursos utilizados bajo situaciones específicas.

Tabla 7: Requisitos no funcionales por eficiencia

ID	Descripción	Prioridad
E1	Deben ser cargados en caché todos aquella información, que no es modificada con el transcurso del tiempo, o aquella que sus intervalos de cambio son prolongados.	Alta
E2	En el caso de los usuarios con un tiempo de ingreso inferior a dos segundos, es necesario que su información sea cambiada en la base de datos.	Alta
E3	La capacidad del sistema debe ser igual o mayor a 3000 usuarios con sesiones simultaneas.	Alta

Elaborado por: José Emilio Fonseca

2.4.2.4. Disponibilidad

Capacidad del sistema o componente de estar operativo y accesible para su

uso cuando se requiere.

Tabla 8 Requisitos no funcionales por disponibilidad

ID	Descripción	Prioridad
D1	La disponibilidad para el acceso al sistema debe ser del 99,99% de las ocasiones en que un usuario intente ingresar.	Alta
D2	El tiempo de inicio o reinicio del sistema deberá ser de un máximo de 5 minutos.	Alta
D3	La duración de fallas no puede exceder de 10 minutos en promedio.	
D4	La probabilidad de que el sistema presente fallas es de un máximo de 0,05.	Alta

Elaborado por: José Emilio Fonseca

2.4.2.5. Seguridad

Se refiere a la capacidad que tiene el sistema de evitar que personas o sistemas no deseados tengan acceso a la información o datos almacenados en el software.

Tabla 9: Requisitos no funcionales por seguridad

ID	Descripción	Prioridad
S1	La autenticación para ingresar al sistema debe ser mediante del servicio de active directory empresarial con LDAP.	Alta
S2	La opción de recordar usuario al momento de Login, debe ser configurada con token de seguridad anclado al local storage del navegador.	Alta
S3	La seguridad debe ser verificada en cada una de las capas del sistema.	
S4	La interfaz debe ser desarrollada libre de ataques XSS y SQL Injections.	Alta

Elaborado por: José Emilio Fonseca

2.4.2.6. Interoperabilidad

Es la interacción que tiene lugar entre dos sistemas o componentes cuando estos intercambian información o hacen uso de esta.

Tabla 10: Requisitos no funcionales por interoperabilidad

ID	Descripción	Prioridad
I1	Es indispensable que el sistema proporcione un API cuyas funcionalidades estén completas correctamente especificadas.	Alta

Elaborado por: José Emilio Fonseca

2.5. Actores del sistema a automatizar

Recibe el nombre de actor, cualquier individuo independiente del sistema que mantenga una relación con este y que haga uso de sus funcionalidades. Por lo tanto, el actor puede ser un operario humano al igual que un sistema ajeno al software en cuestión. Al referirse al usuario humano se puede hablar de roles, consecuentemente, un actor puede poseer más de un rol. En la regla de negocio se tienen 2 actores principales los cuales van a cumplir un rol en específico del sistema y ellos adicional cumplen con un caso de uso específico para cada uno. Los cuales se detallan a continuación:

- Empleado
- Empleado Jefe

Seguidamente, se encuentra una caracterización de todos los actores del sistema junto con vínculo con los requerimientos del sistema anteriormente detallados.

Tabla 11: Descripción de actores del negocio

Actor	Descripción
Empleado	Se entiende por Empleado a la persona que trabaja en la Empresa Eléctrica Quito bajo nombramiento indefinido o temporal que consta en nómina y tiene un contrato. El cual registra su huella dactilar en el biométrico para registrar sus asistencias.
Empleado Jefe	Se entiendo por Empleado Jefe a la persona que trabaja en la Empresa Eléctrica Quito bajo nombramiento indefinido o temporal que consta en nómina y tiene contrato. Esta persona tiene empleados bajo su responsabilidad, además debe de igual manera registrar su huella dactilar en el biométrico para registrar sus asistencias.
Supervisor de Talento Humano	Se entiende por Supervisor de Talento Humano a la persona que trabaja en la Empresa Eléctrica Quito bajo nombramiento indefinido o temporal que consta en nómina y tiene contrato. Esta persona trabaja en el área de Talento Humano y es la asignada para revisar las novedades de las asistencias del personal de la Empresa Eléctrica Quito.

Elaborado por: José Emilio Fonseca

2.6. Diagrama de casos de uso del sistema

Un caso de uso es la forma de describir las reacciones y acciones del comportamiento de un sistema desde una perspectiva de usuario, además de permitir definir límites entre las relaciones del sistema y el entorno. Para un caso de uso de

negocio describe que procesos de negocio están vinculados al campo de acción, como se benefician e interactúan los socios y clientes de este proceso.

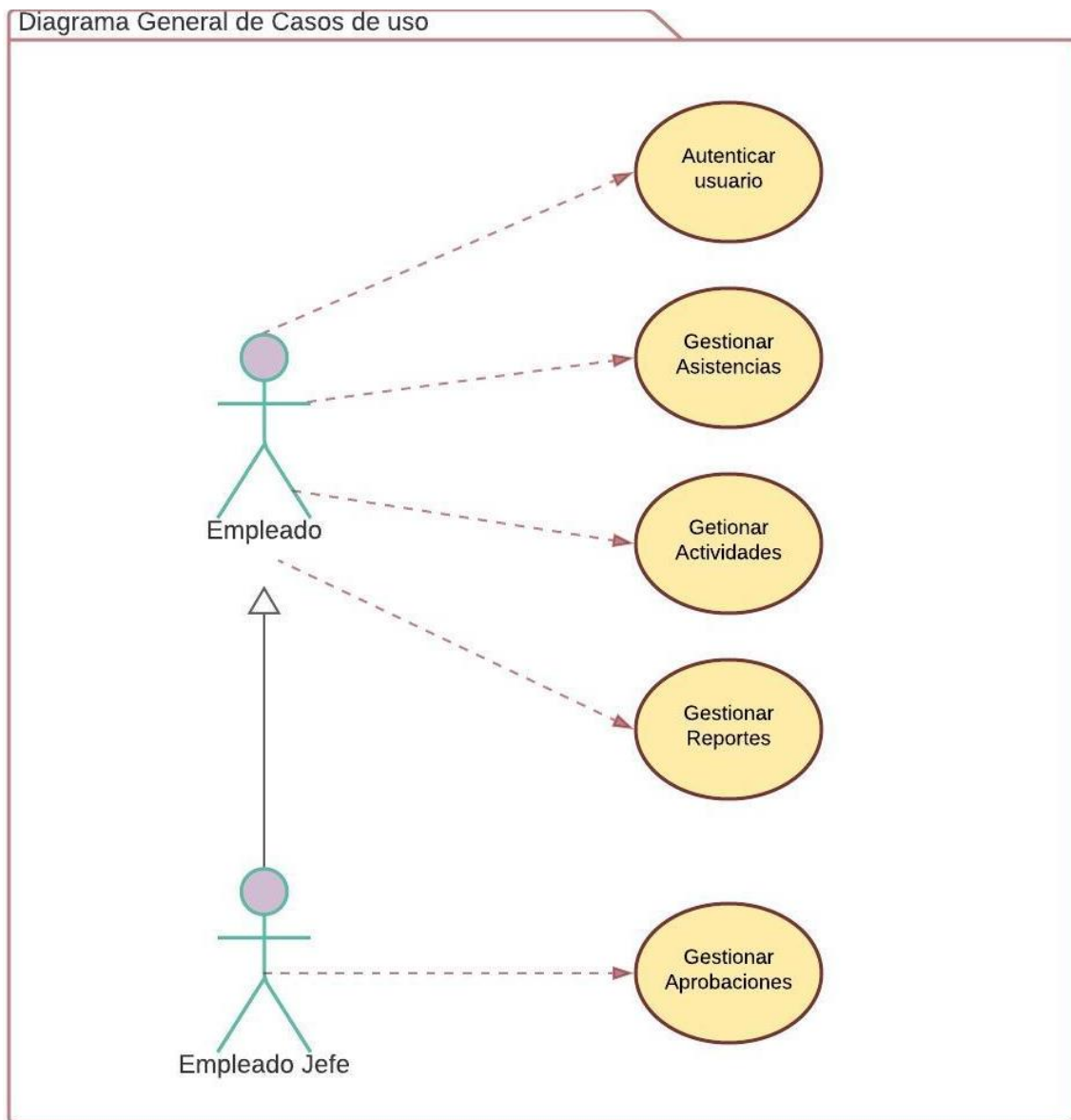


Figura 18: Diagrama de casos de uso del sistema
Elaborado por: José Emilio Fonseca

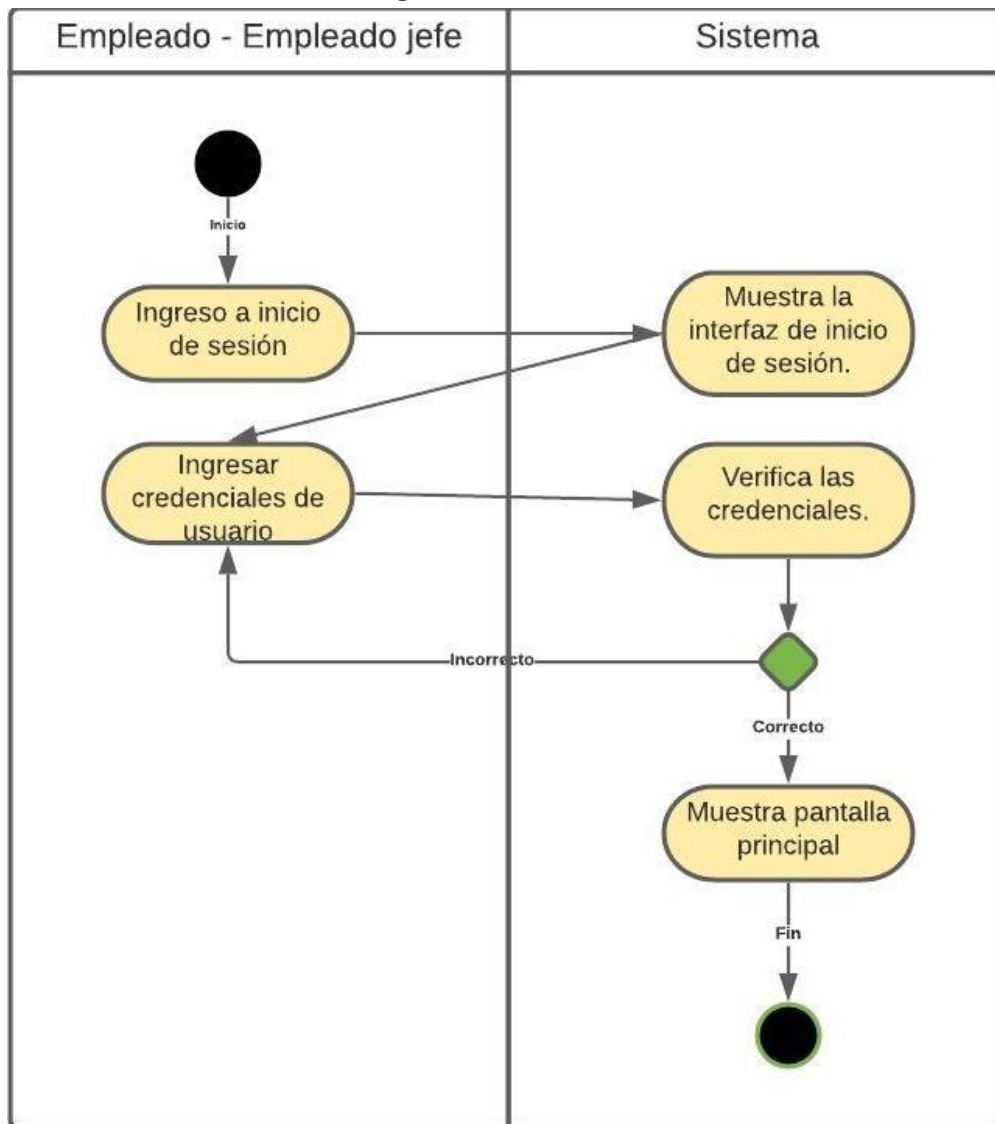
La descripción de los casos de uso del sistema, se realizó con el objetivo de documentar adecuadamente todo el proceso lógico de las actividades.

2.6.1. Descripción del CU Autenticar Usuario

Tabla 12: Descripción del CU - Autenticar Usuario

Caso de uso	Autenticar Usuario
Actores:	Empleado, Empleado Jefe
Propósito:	Verificar usuario para ingreso al sistema.
Resumen:	El caso de uso inicia cuando el usuario debe ingresar al software desarrollado

Diagrama de actividades



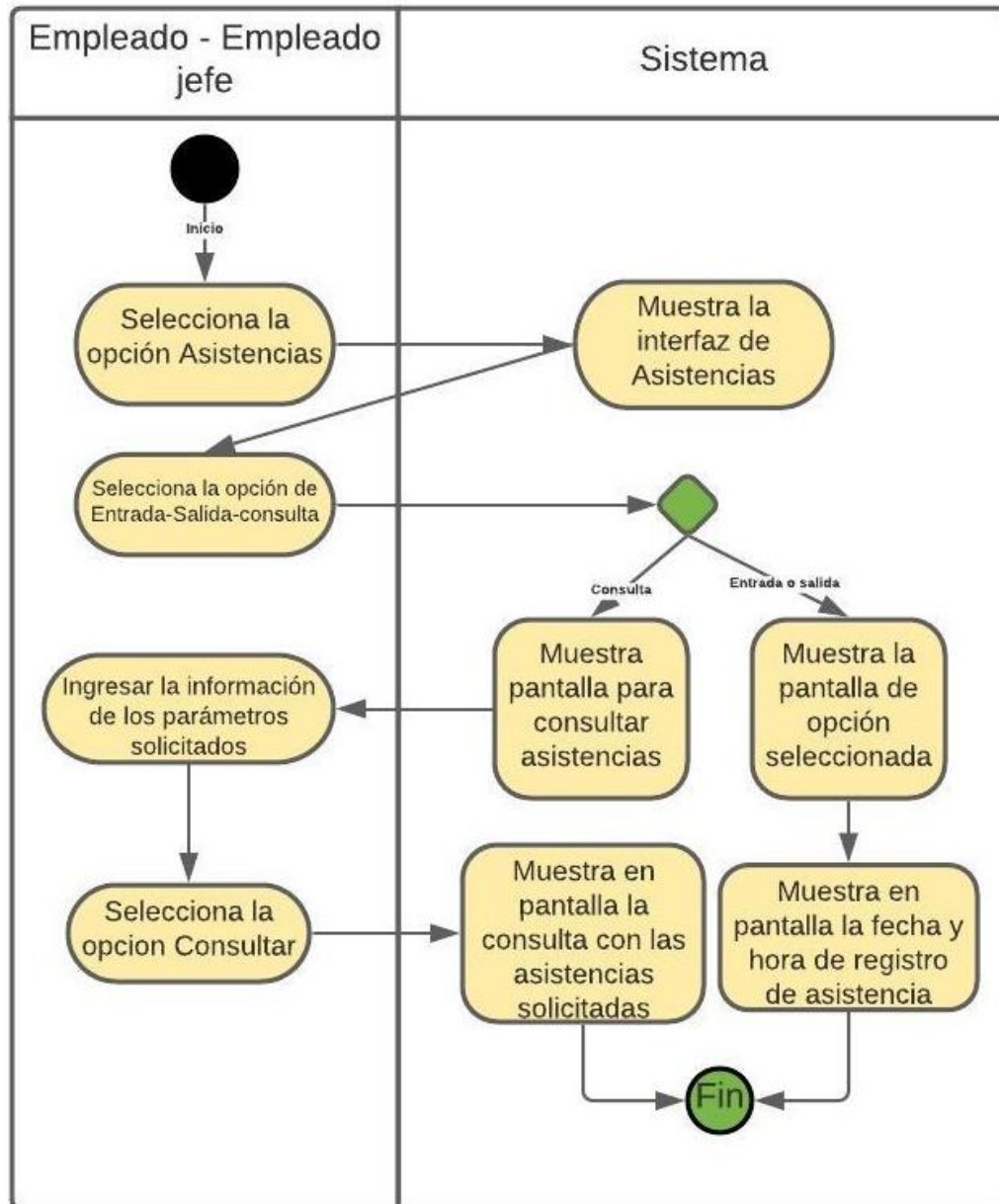
Elaborado por: José Emilio Fonseca

2.6.2. Descripción del CU Gestionar Asistencias

Tabla 13: Descripción del caso de uso - Gestionar Asistencias

Caso de uso	Gestionar Asistencias
Actores:	Empleado, Empleado Jefe
Propósito:	Registrar las asistencias que se tomarán en cuenta para el control de Talento Humano.
Resumen:	El caso de uso inicia cuando el usuario realiza el registro de sus asistencias laborales.

Diagrama de actividades



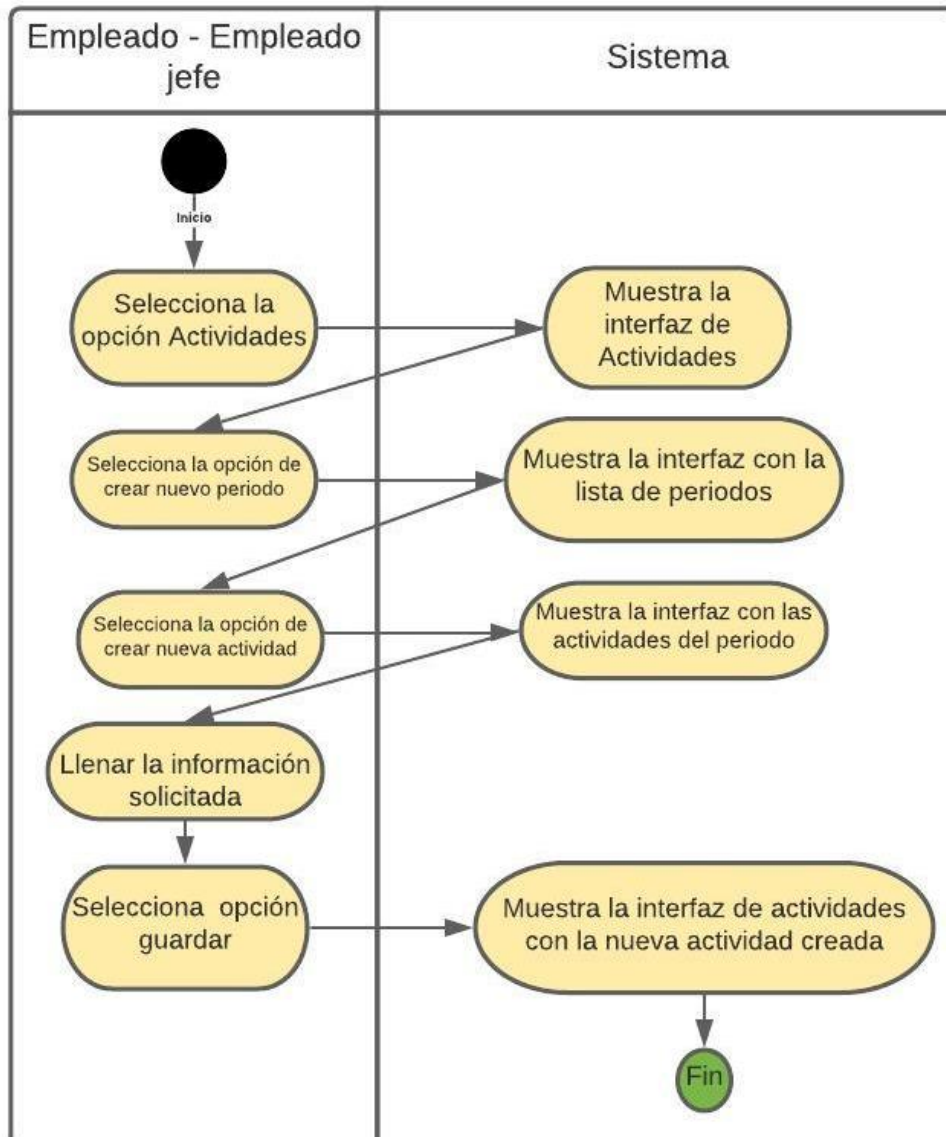
Elaborado por: José Emilio Fonseca

2.6.3. Descripción del CU Gestionar Actividades

Tabla 14: Descripción del CU - Gestionar Actividades

Caso de uso	Gestionar Actividades
Actores:	Empleado, Empleado Jefe
Propósito:	Registrar las actividades que se tomarán en cuenta para el control de Talento Humano.
Resumen:	El caso de uso inicia cuando el usuario realiza el registro de sus actividades semanales.

Diagrama de actividades



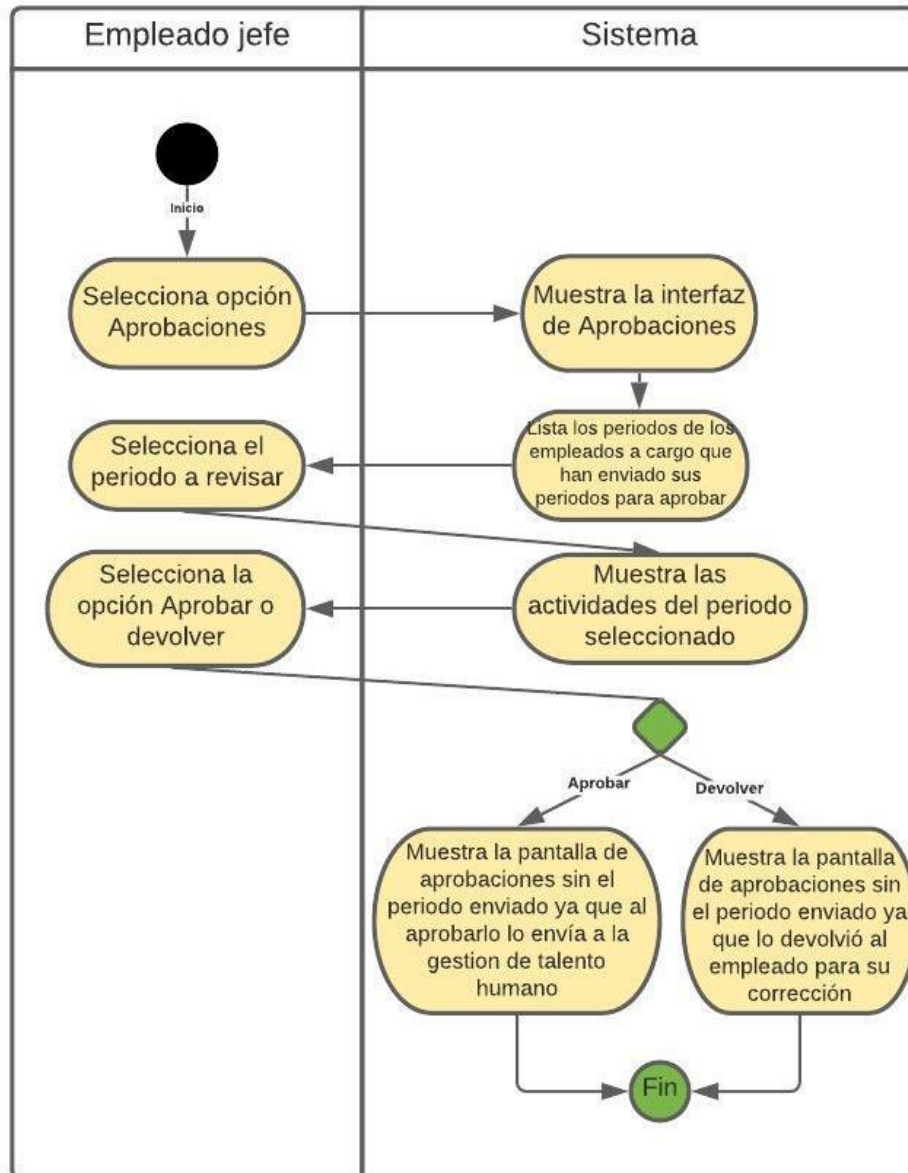
Elaborado por: José Emilio Fonseca

2.6.4. Descripción del CU Gestionar Aprobaciones

Tabla 15: Descripción del CU - Gestionar Aprobaciones

Caso de uso	Gestionar Aprobaciones
Actores:	Empleado Jefe
Propósito:	Aprobar o devolver las actividades de sus empleados a cargo. El caso de uso inicia cuando el usuario con rol de empleado enviar sus actividades semanales a su respectiva jefatura para su aprobación o devolución.
Resumen:	

Diagrama de actividades



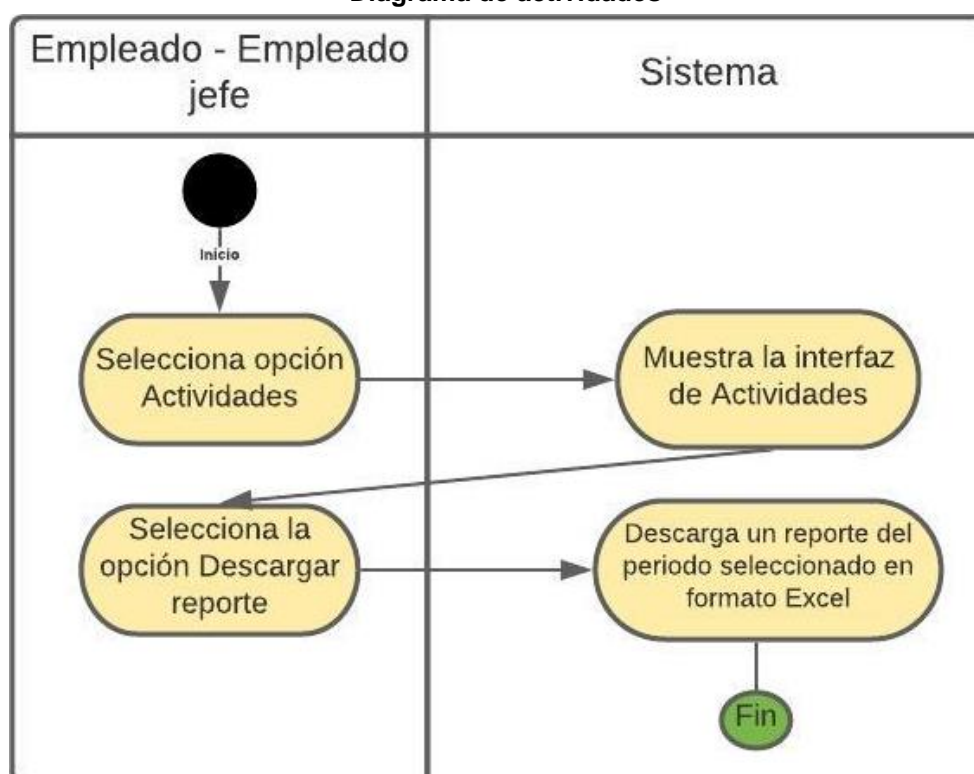
Elaborado por: José Emilio Fonseca

2.6.5. Descripción del CU Gestionar Reportes

Tabla 16: Descripción del CU - Gestionar Reportes

Caso de uso	Gestionar Reportes
Actores:	Empleado, Empleado Jefe
Propósito:	Exportar reportes de las actividades realizadas del empleado o de los empleados a cargo. El caso de uso inicia cuando el usuario empleado requiere descargar un reporte de sus actividades semanales realizadas o cuando el usuario empleado jefe requiere descargar un reporte personalizados de las actividades de sus empleados a cargo.
Resumen:	

Diagrama de actividades



Elaborado por: José Emilio Fonseca

2.7. Conclusiones del capítulo

- El funcionamiento interno de la Empresa Eléctrica de Quito fue descrito satisfactoriamente, en base a eso fue posible identificar que el proceso de control de asistencia se encuentra normado por el reglamento interno de la empresa y por el reglamento del Ministerio de Trabajo.

- El proceso de control de asistencias, el cual será virtualizado, ha sido descrito a detalle. Igualmente, han sido descritos los roles de los actores de este proceso.

- Se han identificado y detallado los requerimientos funcionales y no

funcionales del sistema a desarrollarse, mismos que han sido organizados jerárquicamente en función del nivel de complejidad de cada uno empezando desde el nivel más alto de complejidad y terminando en el nivel más bajo. Los diferentes casos de uso han sido descritos a detalle.

CAPITULO III

3. DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

3.1. Breve descripción del contenido del capítulo

Dentro de este capítulo será descrita la arquitectura del sistema en conjunto con los componentes de diagramación UML. Los componentes de diagramación son: diagramas de clase, secuencia, y despliegue. De igual manera se presentará el modelo relacional de la base de datos. El capítulo será cerrado con el tratamiento de errores del sistema.

3.2. Arquitectura del sistema

En la ingeniería de software se usa patrones de diseño con el único enfoque que son soluciones reutilizables para un problema determinado, un patrón básicamente es una plantilla que ayuda a resolver una problemática. En esta ocasión se ha elegido utilizar el patrón de diseño Modelo-Vista-Controlador (MVC), la cual divide a la aplicación en 3 elementos interconectados. Este patrón es uno de los marcos de trabajo para desarrollo web más utilizados y es usado en casi todos los aplicativos webs modernos. Existe el controlador que se ocupa de las peticiones entrantes, por ejemplo: de los usuarios que navegan por la página web, este delega información y define las interacciones entre los componentes del modelo y la vista. El modelo es el componente central del patrón se encarga de manejar y validar los datos, la lógica y las reglas de la aplicación, además interactúa con la base de datos y pasa los datos al controlador cuando éste lo requiere. Y finalmente la vista que se encarga de presentar la información, representa la interfaz de usuario y muestra los datos que el usuario requiere. (Kopca, 2020)

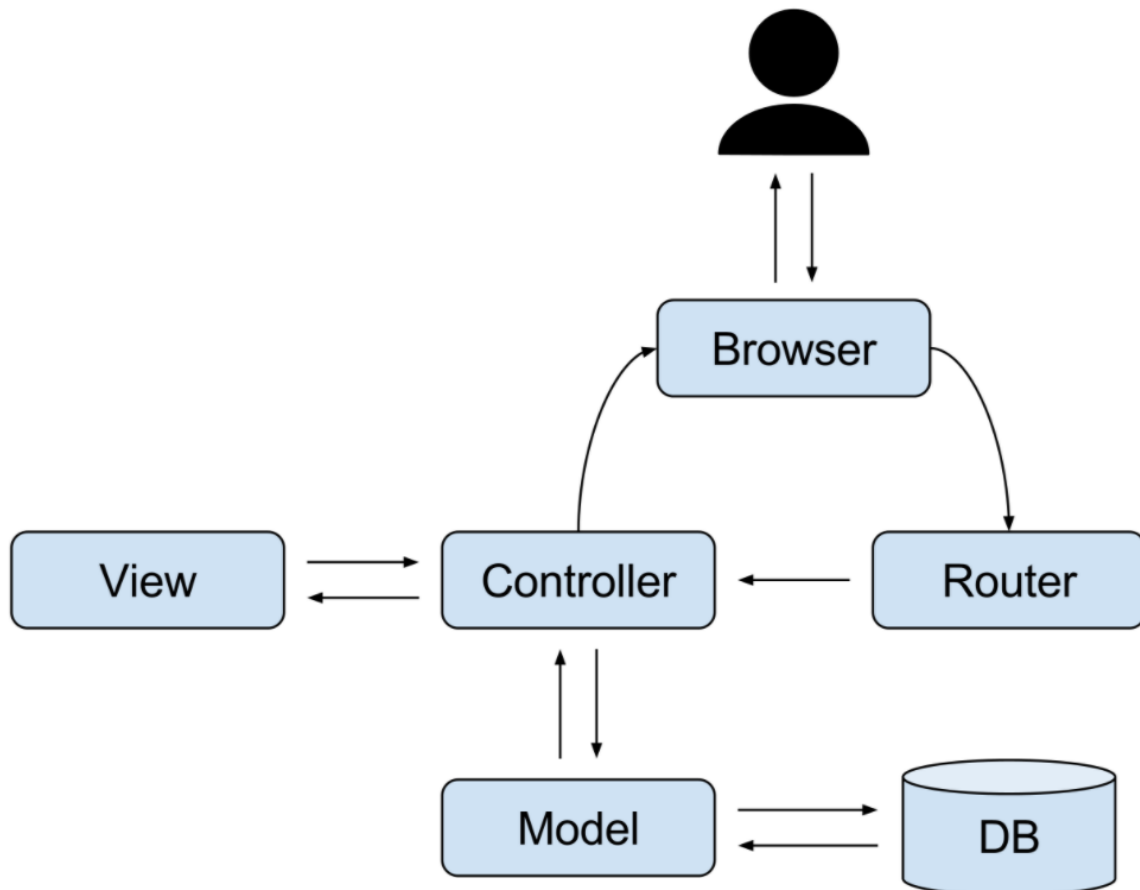


Figura 19: Modelo - Vista – Controlador
Fuente: (Kopca, 2020)

3.3. Diagrama de clases

Resumiendo a (López, Pereira, & Vidal, 2014), se obtiene que el diagrama de clases recoge las clases de los objetos y a sus asociaciones, representa la estructura y el comportamiento de cada uno de los objetos del sistema y sus relaciones con los demás objetos, con el fin de facilitar la comprensión del diagrama, se puede incluir paquetes como elementos del mismo, donde cada uno de ellos agrupa un conjunto de clases. El diagrama de clases permite representar clases abstractas, una clase abstracta es una clase que no puede existir en realidad, pero es útil conceptualmente para el diseño del modelo orientado a objetos.

3.3.1. Diagrama de clases de asistencias



Figura 20: Diagrama de clases de asistencias
Elaborado por: José Emilio Fonseca

3.3.2. Diagrama de clases de actividades

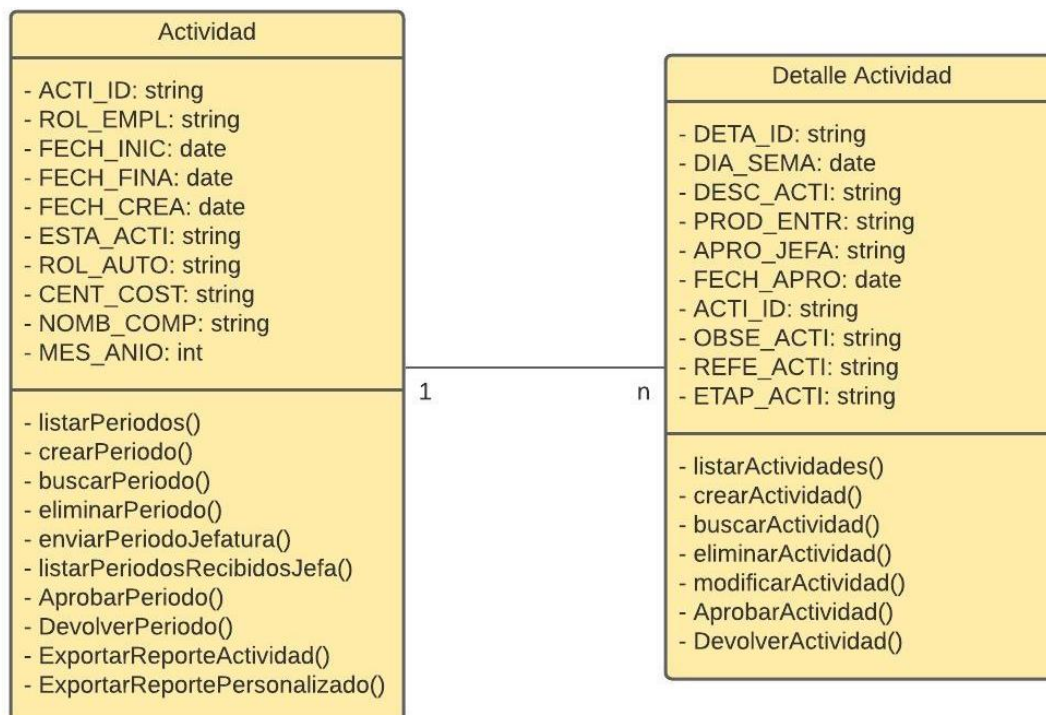


Figura 21: Diagrama de clases de actividades
Elaborado por: José Emilio Fonseca

3.4. Diagrama de secuencia

El diagrama de secuencia es un tipo de diagrama de interacción cuyo objetivo es

describir el comportamiento dinámico del sistema de información haciendo énfasis en la secuencia de los mensajes intercambiados por los objetos, tiene 2 dimensiones el eje vertical el cual representa el tiempo y el eje horizontal que representa diferentes objetos. Respecto a los objetos, es irrelevante el orden en que se representan, aunque su colocación debería poseer la mayor claridad posible. (López, Rubio, Vidal Silva, & Villarroel, 2019, pág. 270)

3.4.1. Diagrama de secuencia de asistencias

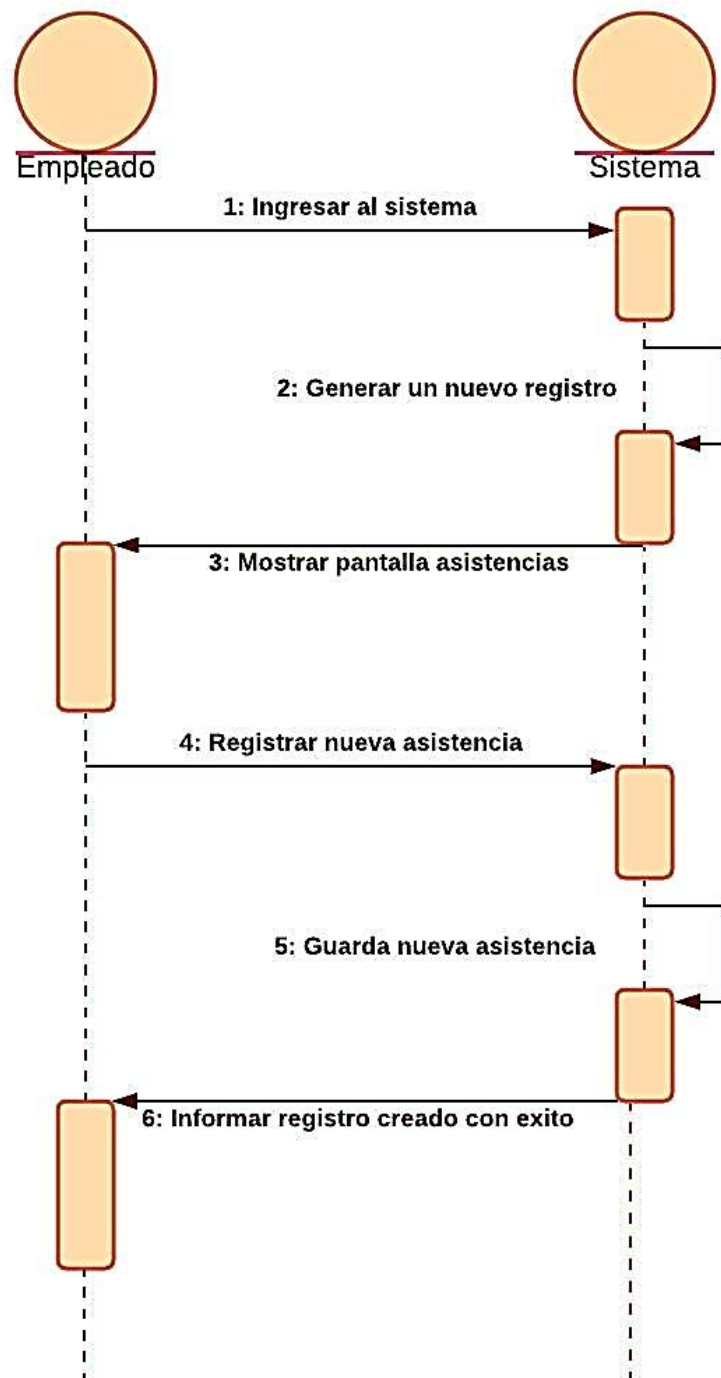


Figura 22: Diagrama de secuencia de asistencias
Elaborado por: José Emilio Fonseca

3.4.2. Diagrama de secuencia de actividades

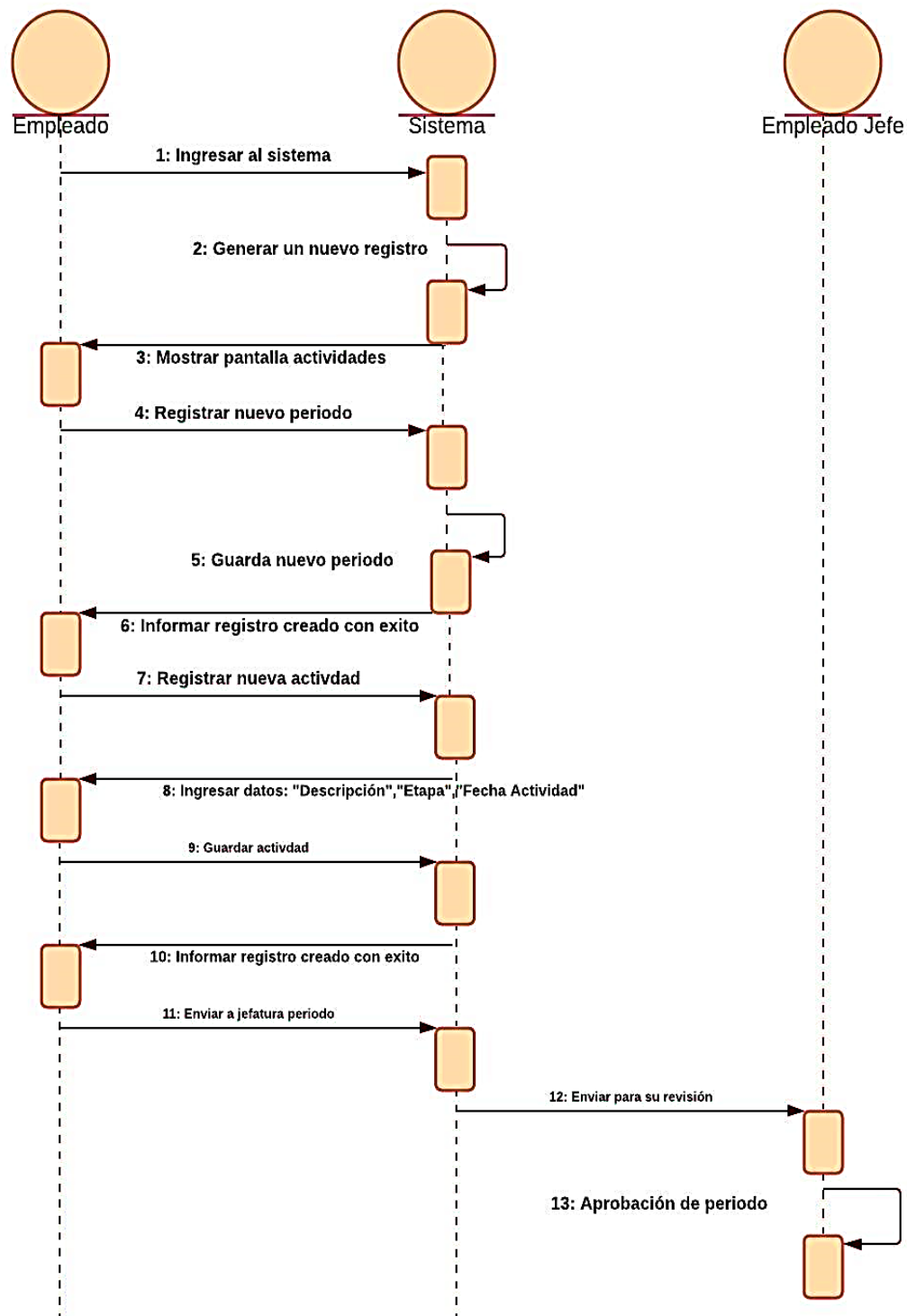


Figura 23: Diagrama de secuencia de actividades
Elaborado por: José Emilio Fonseca

3.5. Tratamiento de errores

En los últimos años, las innovaciones tecnológicas han tenido un fuerte impacto en cualquier ámbito, tanto personal como profesional. Muchas compañías basan gran parte de su actividad empresarial en la tecnología y en consecuencia cada vez más estas soluciones están convirtiendo en herramientas fundamentales para tomar decisiones y detectar problemas de negocio. Las empresas se apoyan en sistemas que generan una gran cantidad de datos en forma de trazas textuales, llamadas técnicamente Logs. Los logs que nos explican el comportamiento de nuestros sistemas o programas, normalmente suelen escribirse en ficheros. (Diaz, 2017)

En esta ocasión se utiliza los logs como buena práctica, la cual aporta distintos beneficios, tanto a nivel de funcionamiento de los sistemas como de objetivos del negocio entre las cuales tenemos:

- Mejorar la gestión y el control de la información
- Detectar amenazas
- Prevenir fugas de información
- Registrar errores de código o datos

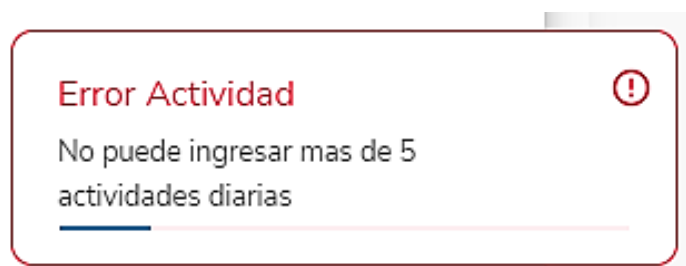


Figura 24: Notificación de errores en el sistema
Elaborado por: José Emilio Fonseca

3.6. Diseño de la base de datos

La página de la empresa (Tecnologías Información, 2018) afirma que:

Se trata de una representación gráfica de cómo será construido el modelo en la base de datos. Comprende las estructuras de tabla, junto con el nombre de columna y el tipo de información contenida, las restricciones, contraseña principal y externa, las relaciones establecidas entre las tablas. Los pasos para generar un modelo físico son:

convertir entidades en tablas, convertir relaciones en claves externas, convertir atributos en columnas y modificar el modelo de datos físicos en función de las restricciones.

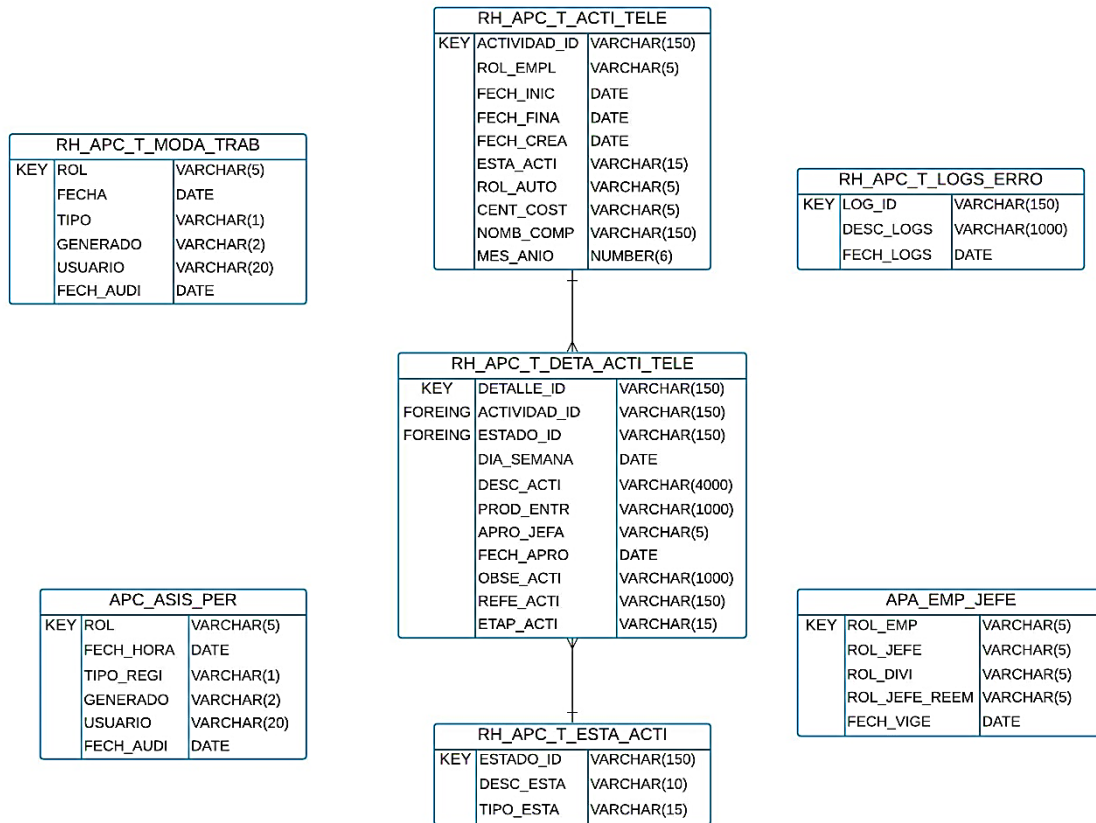


Figura 25: Modelo físico de base de datos
Elaborado por: José Emilio Fonseca

3.7. Diagrama de despliegue

“Es un tipo de diagrama del Lenguaje Unificado de Modelado que muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos”. (Ecured, 2019)

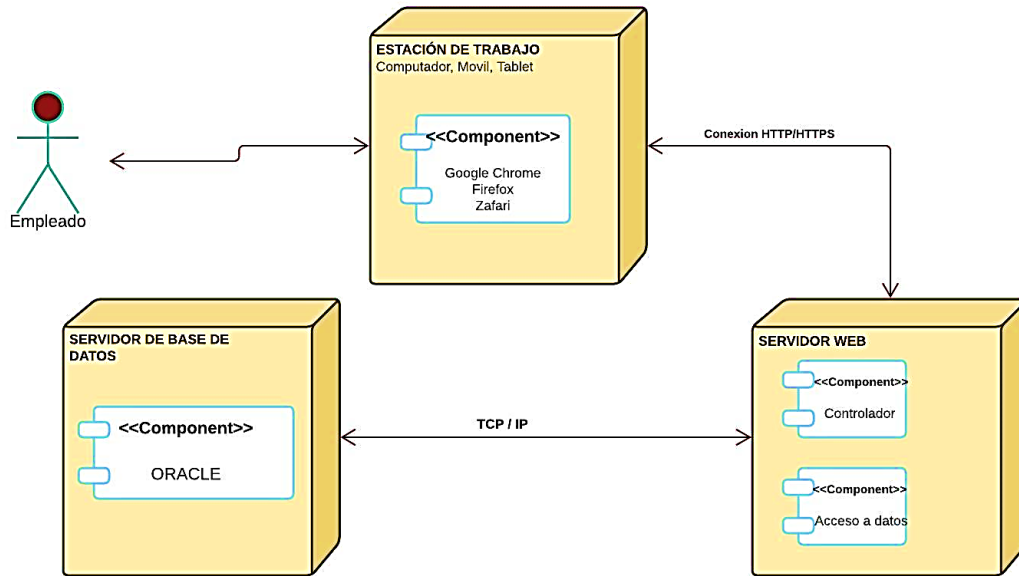


Figura 26: Diagrama de despliegue
Elaborado por: José Emilio Fonseca

3.8. Diagrama de componentes

Este tipo de diagramas son utilizados para mostrar como los componentes individuales se relacionan a través de una vista de diseño estática. En el entorno UML los componentes representan partes modulares del sistema y no dependen el uno del otro y que pueden ser sustituidos por componentes equivalentes. Se caracterizan por ser auto contenidos y comprenden estructuras con variedad de complejidad. La interacción entre los elementos encapsulados ocurre mediante interfaces.

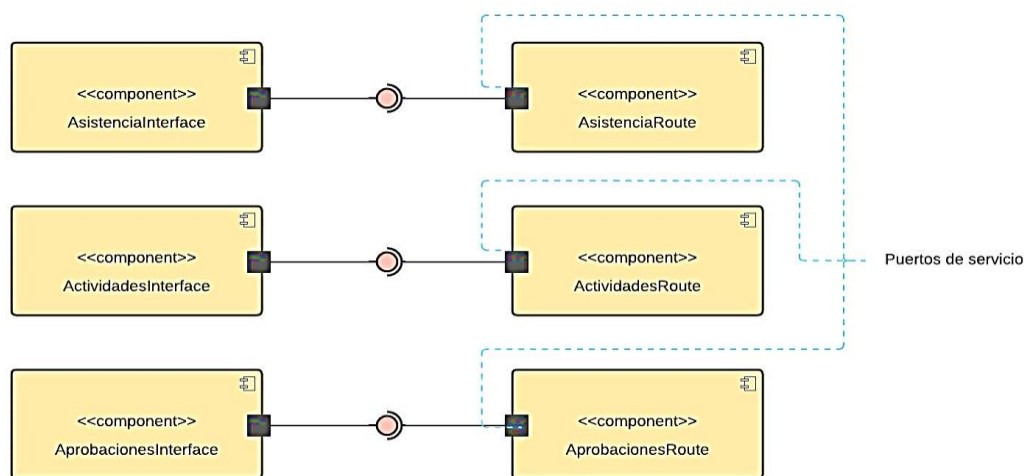


Figura 27: Diagrama de componentes
Elaborado por: José Emilio Fonseca

3.9. Conclusiones de capítulo

Los diagramas UML de clase, secuencia y despliegue fueron elaborados en conformidad de los requerimientos del cliente con el fin de tener documentados los procesos a realizarse. Lo que se busca con esta documentación es proporcionar al equipo de desarrollo las herramientas necesarias para que sean capaces de analizar los requerimientos. Finalmente, la intención de presentar el tratamiento de errores es tener un control a nivel lógico y de usuario para que el equipo de desarrollo y el cliente puedan tomar medidas necesarias en caso de que se llegara a presentar un error en el desarrollo o uso del software.

CAPÍTULO IV

4. PRUEBAS Y ANÁLISIS DE FACTIBILIDAD

4.1. Breve descripción del contenido del capítulo

Este capítulo está destinado a detallar los procesos de pruebas de caja negra junto con el método de cálculo de costo, el cual será el método de estimación de puntos de casos de uso.

4.2. Planificación basada en método de estimación

Una estimación de software es un pronóstico de los valores y tiempo que llevará a cabo el desarrollo de un software a través de los requerimientos solicitados por el cliente. Para el caso de que la estimación se requiera hacer en base al costo es el equivalente a la moneda de elegida, pero si se trata en de una estimación en base al tiempo el equivalente puede expresarse en hora-persona. El único fin de las estimaciones es:

- Desarrollar proyectos
- Construir estimación de costos
- Hacer un diagnóstico de inversiones
- Determinación de un aplicativo de software para un cliente corporativo.

Para ello existen varias técnicas de estimación de costos de software, para este proyecto se ha utilizado el método de puntos de caso de uso para construir la estimación de los costos.

4.2.1. Método de estimación de esfuerzo

De acuerdo a (Garzas, 2021),

El método de punto de casos de uso fue desarrollado por Gustav Karner en el año de 1993, esto con el fin de realizar cálculos del tiempo para completar un proyecto y saber qué tipo de circunstancias pueden llevarlo a ser afectado como:

- La cantidad de actores y su complejidad
- Los requisitos técnicos de un caso de uso como: rendimiento, concurrencia y seguridad.
- Las diferentes situaciones ambientales de los programadores como: conocimientos y experiencia.

Cuando el equipo de desarrollo de software toma en cuenta estos factores descritos es mucho más sencillo de realizar los cálculos para:

- Tiempo de desarrollo
- Costos
- Asignación de recursos

4.2.2. Factores de complejidad

Seguidamente se enlistarán los factores para la complejidad de los requerimientos junto con sus definiciones. Para esto se considerará el grado de complejidad como criterio para organizarlos.

Tabla 17: Definición de factores de complejidad

ID	Complejidad del requerimiento	Peso
C1	Alta	9
C2	Media	6
C3	Baja	3

Elaborado por: José Emilio Fonseca

La complejidad de los actores que hacen uso del sistema será medida bajo los siguientes criterios:

Tabla 18: Definición de factores para medir la complejidad

ID	Actores	Peso	Criterio de clasificación
C1	Alta	0,5	Usuario que a través de una interfaz gráfica mantiene una interacción con el sistema Puede ser otro sistema cuya interacción ocurre a través de un protocolo u otra persona que interactúa mediante una interfaz en modo de consola.
C2	Media	0,3	Un sistema externo que, interactúa mediante una interfaz de programación, interactúa con el sistema en desarrollo.
C3	Baja	0	

Elaborado por: José Emilio Fonseca

4.2.3. Factores técnicos

Los factores técnicos se componen de 12 puntos los cuales calculan la dificultad de cada módulo del software a desarrollar, todos los factores descritos tienen un peso con el cual se consigue puntos ponderados dependiendo la calificación que se le establezca.

Tabla 19: Descripción de los factores técnicos

ID	Factor técnico	Peso (0.5-2)	Valor(0.1)	Descripción
FT1	Sistema distribuido	0.5	Toma el valor de definido	Está relacionado con requisitos de los cuales su hardware o software están alojados en otros equipos conectados a la red. Sus acciones son comunicadas o coordinadas mediante el intercambio de mensajes, para alcanzar un objetivo. Se trata de la capacidad que el requisito tiene para dar cumplimiento a sus funciones. Generalmente está vinculado a como el requisito reacciona cuando hay varios usuario conectados al mismo tiempo y a su tiempo de respuesta.
FT2	Rendimiento y Concurrencia	0.5	Toma el valor de definido	Se trata de la composición visual diseñada en la versión preliminar del requisito
FT3	Complejidad de diseño gráfico	0.3	Toma el valor de definido	Se trata de la complejidad al definir el requisito y de los algoritmos que serán utilizados.
FT4	Algoritmos y procesamiento interno	0.5	Toma el valor de definido	Es la característica que tiene el requisito para manipular la identidad de las entidades, el ingreso a los recursos, el historial cronológico de las acciones, asegurarse que la información sea accesible únicamente para las entidades autorizadas, protección de la información contra manipulación.
FT5	Seguridad	0.5	Toma el valor de definido	Se trata de la cantidad de veces que se validará una determinada acción del sistema.
FT6	Monto de validaciones	0.5	Toma el valor de definido	Es la posibilidad de hacer uso de un mismo código en diferentes requisitos y sistemas.
FT7	Reutilización	0.5	Toma el valor de definido	Son los estándares que necesitan ser integrados al requisito.
FT8	Utilización de estándares	0.5	Toma el valor de definido	La necesidad de la interacción entre el requisito con el hardware externo (lector de huella).
FT9	Integración con dispositivos externos	0.5	Toma el valor de definido	Necesidad de uso de componentes desarrollados, API's y librerías para que el requisito funcione.
FT10	Uso de librerías externas	0.5	Toma el valor de definido	Capacidad del requisito para funcionar en otras plataformas o dispositivos.
FT11	Interoperabilidad	0.5	Toma el valor de definido	Los usuarios finales deben estar entrenados o capacitados para hacer uso del requisito.
FT12	Entrenamiento a usuario	0.5	Toma el valor de definido	

Elaborado por: José Emilio Fonseca

Seguido el cálculo de los valores de los factores técnicos:

Tabla 20: Descripción de los valores de ponderación de factores técnicos

Valor	Descripción
0	No presencia del factor técnico.
1	Presencia del factor técnico.

Elaborado por: José Emilio Fonseca

4.2.4. Factores ambientales

Las habilidades y la formación de los equipos implicados en el desarrollo tienen un impacto significativo en el tiempo estimado. Estos puntos deben tenerse en cuenta al calcular los factores ambientales.

Tabla 21: Descripción de los factores ambientales

ID	Factor ambiental	Peso(0.5-2)	Valor(1,2,3)	Descripción
A1	Familiaridad con el modelo del proyecto usado	1	Toma el valor definido	Es el conocimiento que el equipo maneja acerca del negocio que va a ser desarrollado.
A2	Disponibilidad de equipo	1	Toma el valor definido	Es el tiempo que el equipo tiene para destinar al desarrollo del sistema.
A3	Experiencia en el entorno de desarrollo	1	Toma el valor definido	Es el conjunto de habilidades, conocimiento y experiencia que el equipo tiene para el uso de herramientas y tecnologías que serán usadas en el desarrollo del requisito.
A4	Familiarización con el requisito	1	Toma el valor definido	Se trata de la experiencia que el desarrollador tiene en cuanto a requisitos similares.
A5	Estabilidad del requisito	1	Toma el valor definido	El requisito no necesita modificaciones ya que se encuentra definido correctamente.
A6	Conocimiento en herramientas de control de versiones	1	Toma el valor definido	Se trata de cuanto sabe el equipo y que habilidades tiene para manipular herramientas de control de código y documentación.
A7	Motivación	0.5	Toma el valor definido	Es lo que estimula al equipo a dar cumplimiento a los objetivos ya sea para alcanzar recompensas o no.
A8	Conocimiento en herramientas de control de ticket	0.5	Toma el valor definido	Es el dominio que tiene el equipo para manejar las herramientas de ticket.
A9	Equipo distribuido geográficamente	1	Toma el valor definido	Se trata de la posición geográfica de los miembros del equipo.
A10	Dificultad del lenguaje de programación	1	Toma el valor definido	Se refiere a cuán difícil de usar es el lenguaje utilizado.

Elaborado por: José Emilio Fonseca

Seguido el cálculo de los valores de los factores ambientales:

Tabla 22: Descripción de los valores de ponderación de factores ambientales

Valor	Descripción
3	No presencia del factor ambiental.
2	Presencia parcial del factor ambiental.
1	Presencia total del factor ambiental.

Elaborado por: José Emilio Fonseca

4.2.5. Métricas a medir para realizar la estimación

Las métricas a evaluar son:

- Puntos de Requisitos Sin Ajustar (PRSA): $PRSA = FPASA + FPRSA$
- Factor de Peso del Actor Sin Ajustar (FPASA): $FPASA = \sum_{i=1}^n \text{Peso}_i * \text{Actor}_i$
- Factor de Peso de Requisitos Sin Ajustar (FPRSA): $FPRSA = \text{Peso}$
- Factor de Complejidad Técnica (FCT): $FCT = 0,6 + (\sum_{i=1}^n \text{Peso}_i * \text{Valor}_i) / 100$
- Factor Ambiental (FA): $FA = 1,4 - (0,03 * \sum_{i=1}^n \text{Peso}_i * \text{Valor}_i)$
- Puntos de Requisitos Ajustado (PRA): $PRA = PRSA * FTC * FA$

El esfuerzo estimado en horas persona:

- $E = PRA * HH$
- HH = 6 horas

Se utilizará como valor constante 6 horas por cada punto de requisito ajustado, esto en cuanto al valor HH (Hora/Hombre), el cual se tomó como referencia del método de Karner expuesto anteriormente.

4.2.6. Porcentaje de actividades por ciclo de vida

Usando este criterio y la estimación de tiempo calculada a partir del punto del caso de uso como entrada, se pueden calcular otras estimaciones para saber el tiempo total de duración proyecto.

A continuación, se muestran los resultados obtenidos luego de haber aplicado el método de estimación de puntos de caso de uso al Sistema de registro y control de asistencias y actividades semanales para la Empresa Eléctrica Quito.

ID REQUISITO	COMPLEJIDAD	ACTORES			Factores Técnicos												Factores Ambientales										PRA	HH	HORAS																															
		A1	A2	A3	FPASA	FPRSA	PRSA	FT1	FT2	FT3	FT4	FT5	FT6	FT7	FT8	FT9	FT10	FT11	FT12	FCT	FA1	FA2	FA3	FA4	FA5	FA6				FA7	FA8	FA9	FA10	FA																										
SRCAA- RF1	Media	0	1	0	0,3	6	6,30	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	0	1	1,13	4,271	6	25,63																										
SRCAA- RF2.1	Baja	1	0	0	0	3	3,00	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	1	1	1,10	1,98	6	11,88																										
SRCAA- RF2.2	Baja	1	0	0	0	3	3,00	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	1	1	1,10	1,98	6	11,88																										
SRCAA- RF2.3	Baja	1	0	0	0	3	3,00	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	1	1	1,10	1,98	6	11,88																										
SRCAA- RF3.1	Baja	1	0	0	0	3	3,00	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	1	1	1,10	1,98	6	11,88																										
SRCAA- RF3.2	Baja	1	0	0	0	3	3,00	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	1	1	1,10	1,98	6	11,88																										
SRCAA- RF3.3	Baja	1	0	0	0	3	3,00	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	1	1	1,10	1,98	6	11,88																										
SRCAA- RF3.4	Baja	1	0	0	0	3	3,00	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	1	1	1,10	1,98	6	11,88																										
SRCAA- RF3.5	Baja	1	0	0	0	3	3,00	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	1	1	1,10	1,98	6	11,88																										
SRCAA- RF3.6	Baja	1	0	0	0	3	3,00	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	1	1	1,10	1,98	6	11,88																										
SRCAA- RF3.7	Baja	1	0	0	0	3	3,00	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	1	1	1,10	1,98	6	11,88																										
SRCAA- RF3.8	Baja	1	0	0	0	3	3,00	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	1	1	1,10	1,98	6	11,88																										
SRCAA- RF3.9	Baja	1	0	0	0	3	3,00	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	1	1	1,10	1,98	6	11,88																										
SRCAA- RF3.10	Baja	1	0	0	0	3	3,00	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	1	1	1,10	1,98	6	11,88																										
SRCAA- RF4.1	Baja	1	0	0	0	3	3,00	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	1	1	1,10	1,98	6	11,88																										
SRCAA- RF4.2	Baja	1	0	0	0	3	3,00	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	1	1	1,10	1,98	6	11,88																										
SRCAA- RF4.3	Baja	1	0	0	0	3	3,00	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	1	1	1,10	1,98	6	11,88																										
SRCAA- RF5.1	Baja	1	0	0	0	3	3,00	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	1	1	1,10	1,98	6	11,88																										
SRCAA- RF5.2	Baja	1	0	0	0	3	3,00	0	0	0	0	0	0	0	0	0	0	0	0,60	1	1	1	1	1	1	1	1	1	1	1	1,10	1,98	6	11,88																										
TOTAL ESFUERZO OBTENIDO																																																												239,47

Figura 28. Resultados obtenidos del método de aplicación
Elaborados por: José Emilio Fonseca

Tabla 23: Porcentaje de actividades

Actividad	Porcentaje	Tiempo (Horas)
Análisis	15%	35,92
Diseño	15%	35,92
Programación	40%	95,78
Pruebas	15%	35,92
Otras actividades	15%	35,92

Elaborado por: José Emilio Fonseca

Como se determina en la tabla anterior, se aplicó el método de puntos de casos de uso para determinar el esfuerzo del desarrollo para el software de la Empresa Eléctrica Quito, el cual arrojó el resultado de un total **239,47** horas/hombre, a ese valor se le realiza una multiplicación de 8 ya que actualmente en el Ecuador el costo de cada hora de desarrollo de software de un Ingeniero en Sistemas es de 8 USD. Por lo que el total del valor aproximado del software es de **1,915.15 USD**.

4.3. Casos de pruebas

“La prueba de caja negra es una tecnología de prueba de software en la que las funciones se verifican sin considerar la estructura del código interno, los detalles de implementación o los escenarios de ejecución interna en el software”. (Testing Baires, 2017)

4.3.1. Clases de equivalencia

Dentro de las pruebas de caja negra existen técnicas de caja de prueba negra, por lo que una clase de equivalencia representa un grupo de estados validos o inválidos los requisitos de entrada.

4.3.1.1. Identificación de las clases de equivalencias: crear asistencia

Clase de equivalencias

Tabla 24: Identificación de clases de equivalencias crear asistencia

Parámetro de entrada	Regla heurística a aplicar	Clases válidas (CV)	Clases inválidas (CI)
Iniciar Jornada Laboral	Seleccionar botón con descripción de Inicio Jornada Laboral	CV31: Seleccionar botón iniciar jornada	CI31: No se registra asistencia de entrada.
Finalizar Jornada Laboral	Seleccionar botón con descripción de Finalizar Jornada Laboral	CV32: Seleccionar botón finalizar jornada	CI32: No se registra asistencia de salida

Elaborado por: José Emilio Fonseca

Diseño de los casos de pruebas para clases válidas

Tabla 25: Clases válidas para crear asistencias

Iniciar Jornada Laboral	Finalizar Jornada Laboral	Clases validas cubiertas
10/01/2021 7:30:00	10/01/2021 15:30:00	CV31, CV32

Elaborado por: José Emilio Fonseca

Diseño de los casos de pruebas para clases inválidas

Tabla 26: Clases inválidas para crear asistencias

Iniciar Jornada Laboral	Finalizar Jornada Laboral	Clases invalidas cubiertas
No se seleccionan	No se seleccionan	CI31, CI32

Elaborado por: José Emilio Fonseca

Matriz de casos de prueba

Tabla 27: Matriz de casos de prueba crear asistencias

Clases de equivalencia	Resultados esperados	Resultados reales
CV31	Hora de registro de entrada (fecha y hora)	Hora de registro de entrada 10/01/2021 7:30:00
CV32	Hora de registro de salida (fecha y hora)	Hora de registro de salida 10/01/2021 15:30:00
CI31	No se registra asistencia	No se registra asistencia
CI32	No se registra asistencia	No se registra asistencia

Elaborado por: José Emilio Fonseca

4.3.1.2. Identificación de las clases de equivalencias: crear periodo

Clase de equivalencias

Tabla 28: Identificación de clases de equivalencias crear periodo

Parámetro de entrada	Regla heurística a aplicar	Clases válidas (CV)	Clases inválidas (CI)
Fecha Inicio	Seleccionar fecha inicio	CV41: Seleccionar fecha inicio	CI41: No seleccionar fecha de inicio
Fecha Fin	Seleccionar fecha fin	CV42: Seleccionar fecha fin	C42: No seleccionar fecha de fin

Elaborado por: José Emilio Fonseca

Diseño de los casos de pruebas para clases válidas

Tabla 29: Clases válidas para crear periodo

Fecha Inicio	Fecha Fin	Clases validas cubiertas
11/01/2021	15/01/2021	CV41, CV42

Elaborado por: José Emilio Fonseca

Diseño de los casos de pruebas para clases inválidas

Tabla 30: Clases inválidas para crear periodos

Fecha Inicio	Fecha Fin	Clases invalidas cubiertas
No se seleccionan	No se seleccionan	CI41, CI42

Elaborado por: José Emilio Fonseca

Matriz de casos de prueba

Tabla 31: Matriz de casos de prueba crear periodo

Clases de equivalencia	Resultados esperados	Resultados reales
CV41, CV42	Periodo creado	Periodo creado
CI41	Mensaje de que "La fecha inicio es obligatoria"	Mensaje de que "La fecha inicio es obligatoria"
CI42	Mensaje de que "La fecha fin es obligatoria"	Mensaje de que "La fecha fin es obligatoria"
CI43	Mensaje de que "Periodo no puede ser mayor a 7 días"	Mensaje de que "Periodo no puede ser mayor a 7 días"

Elaborado por: José Emilio Fonseca

4.3.1.3. Identificación de las clases de equivalencias: crear actividad

Clase de equivalencia

Tabla 32: Identificación de clases de equivalencias crear actividades

Parámetro de entrada	Regla heurística a aplicar	Clases válidas (CV)	Clases inválidas (CI)
Descripción	Llenar campo con la descripción de la actividad	CV51: Llenar el campo descripción	CI51: No llenar el campo descripción
Referencia	Llenar campo con la referencia de la actividad	CV52: Llenar el campo referencia	CI52: Sobrepasar los 120 caracteres.
Producto entregable	Llenar campo con el producto entregable de la actividad	CV53: Llenar el campo producto entregable	CI53: Sobrepasar los 1000 caracteres.
Observación	Llenar campo con la observación de la actividad	CV54: Llenar el campo observación	CI54: Sobrepasar los 1000 caracteres.
Fecha actividad	Seleccionar fecha actividad	CV55: Seleccionar la fecha de la actividad	CI55: Seleccionar una fecha fuera del rango del periodo.
Etapas	Seleccionar la etapa de la actividad	CV56: Seleccionar etapa	CI56: No seleccionar la etapa

Elaborado por: José Emilio Fonseca

Diseño de los casos de pruebas para clases válidas

Tabla 33: Clases válidas para crear actividad

Descripción	Referencia	Producto entregable	Observación	Fecha actividad	Etapas	Clases validas cubiertas
Atención a tarea SGI	215958	NO	Se realizó la revisión del envío del comprobante y se modificó el correo ya que existía 2 destinatarios diferentes.	11/01/2021	CULMINADO	CV51, CV52, CV53, CV54, CV55, CV56

Elaborado por: José Emilio Fonseca

Diseño de los casos de pruebas para clases inválidas

Tabla 34: Clases inválidas para crear actividades

Descripción	Referencia	Producto entregable	Observación	Fecha actividad	Etapas	Clases validas cubiertas
Se deja vacío	215958	NO	Se realizó la revisión del envío del comprobante y se modificó el correo ya que existía 2 destinatarios diferentes.	11/01/2021	CULMINADO	CI51
Atención a tarea SGI	Referencia con más de 120 caracteres	NO	Se realizó la revisión del envío del comprobante y se modificó el correo ya que existía 2 destinatarios diferentes.	11/01/2021	CULMINADO	CI52
Atención a tarea SGI	215958	Producto entregable con más de 1000 caracteres	Se realizó la revisión del envío del comprobante y se modificó el correo ya que existía 2 destinatarios diferentes.	11/01/2021	CULMINADO	CI53
Atención a tarea SGI	215958	NO	Observación con más de 1000 caracteres	11/01/2021	CULMINADO	CI54
Atención a tarea SGI	215958	NO	Se realizó la revisión del envío del comprobante y se modificó el correo ya que existía 2 destinatarios diferentes.	20/01/2021	CULMINADO	CI55
Atención a tarea SGI	215958	NO	Se realizó la revisión del envío del comprobante y se modificó el correo ya que existía 2 destinatarios diferentes.	11/01/2021	No se selecciona	CI56

Elaborado por: José Emilio Fonseca

Matriz de casos de prueba

Tabla 35: Matriz de casos de prueba crear actividad

Clases de equivalencia	Resultados esperados	Resultados reales
CV51, CV52, CV53, CV54, CV55, CV56	Actividad creada	Actividad creada
CI51	Mensaje de que “La descripción de la actividad es obligatoria”	Mensaje de que “La descripción de la actividad es obligatoria”
CI52	Mensaje de que “La referencia de la actividad no puede ser mayor a 120 caracteres”	Mensaje de que “La referencia de la actividad no puede ser mayor a 120 caracteres”
CI53	Mensaje de que “El producto entregable de la actividad no puede ser mayor a 1000 caracteres”	Mensaje de que “El producto entregable de la actividad no puede ser mayor a 1000 caracteres”
CI54	Mensaje de que “La observación de la actividad no puede ser mayor a 1000 caracteres”	Mensaje de que “La observación de la actividad no puede ser mayor a 1000 caracteres”
CI55	Mensaje de que “La fecha de la actividad no puede ser fuera del rango del periodo”	Mensaje de que “La fecha de la actividad no puede ser fuera del rango del periodo”
CI56	Mensaje de que “La etapa de la actividad es obligatoria”	Mensaje de que “La etapa de la actividad es obligatoria”

Elaborado por: José Emilio Fonseca

4.4. Verificación del sistema

Se realizó la salida a producción del software para control de asistencias y registro de actividades en la Empresa Eléctrica Quito. Para la constancia del mismo a continuación se puede ver un certificado del funcionamiento:



Quito D.M. enero 13 de 2021

Por medio del presente, certifico que el señor José Emilio Fonseca Paz, con cedula de ciudadanía 1723304356, luego de cumplir con todas las fases de desarrollo ha implementado el Sistema "Registro de Asistencia y Actividades para la Empresa Eléctrica Quito", el cual cumple con los requisitos de confiabilidad, disponibilidad, integridad, mismo que apoya la gestión de Talento Humano de la EEQ.

Debo agregar que el Señor José Emilio Fonseca ha demostrado valores éticos, morales y de honestidad que hacen de él un Profesional que cuenta con el apoyo del Área de Tecnología.

El interesado puede hacer uso del presente certificado mientras se encuentre dentro del ámbito legal.

Atentamente,



Ing. Héctor Segovia Ramírez
Director de Tecnología de la Información y
Comunicaciones EEQ

Figura 29: Certificado de desarrollo software
Elaborado por: Empresa Eléctrica Quito

4.5. Conclusiones del capítulo

Los resultados obtenidos en este capítulo son los siguientes: Las pruebas de

caja negra han sido ejecutadas satisfactoriamente conforme a lo que se ha detallado anteriormente. EL método de estimación de puntos de casos de uso ha permitido calcular el esfuerzo técnico invertido en el desarrollo del sistema, lo cual hace posible calcular el costo tentativo que tendrá el producto final.

CONCLUSIONES

- Los procesos requeridos para el control de ingreso y salida de la jornada laboral del personal de la Empresa Eléctrica Quito han sido identificados exitosamente. Posteriormente, la selección de aquellos procesos que necesitan ser automatizados se llevó a cabo en compañía del departamento de recursos humanos.

- Las metodologías y tecnologías disponibles para el desarrollo de software han sido analizadas. En base al mencionado análisis, fue posible seleccionar las herramientas que hicieron posible el desarrollo del sistema de control y registro de asistencias y actividades.

- La fase de diseño del aplicativo fue realizada por el encargado del proyecto tomando en consideración los requerimientos detallados por el departamento de recursos humanos. Esto, con la intención de entregar un sistema que se adapta a las necesidades de la empresa en cuanto a asistencias y actividades.

- El sistema desarrollado fue sometido a pruebas de caja negra y su comportamiento fue el esperado conforme a los criterios establecidos.

- El método de estimación de puntos de casos de uso hizo posible la estimación de esfuerzo y costo requerido en el desarrollo del aplicativo. De esta manera, se pudo que el sistema desarrollado representa un ahorro económico debido a que no fue necesario contratar personal externo o el servicio de plataformas que brindan este servicio.

RECOMENDACIONES

- Una vez desarrollado el software para el control de asistencias y registro de actividades y en virtud que su ejecución ha sido exitosa se debería considerar la posibilidad de migrar progresivamente los procesos web que maneja la empresa a herramientas tecnológicas Node.js y Angular para los aplicativos desarrollados a futuro con el fin de actualizar los ambientes de desarrollo web de la Empresa Eléctrica de Quito.

- Como plan de mejora del sistema se recomienda el uso de la librería FaceApi.js para realizar el reconocimiento facial del empleado y de esa manera registrar la asistencia por medio de su rostro. Esto proporcionará más filtros de seguridad de ingreso al sistema, lo cual evitará posible suplantación de identidad o accesos al sistema no deseados.

- Luego de haber dado cumplimiento a los objetivos del presente proyecto se recomienda migrar el módulo de justificación de atrasos y faltas al sistema recientemente integrado con el fin de agilizar los procesos en un solo módulo y que el acceso a toda la información sea más fácil. De esta manera, los empleados podrán reportar la información inherente a su jornada laboral en un mismo sitio y el departamento de recursos humanos podrá generar el reporte de una manera más rápida.

BIBLIOGRAFIA

Abellan, E. (05 de marzo de 2020). *Scrum: qué es y cómo funciona esta metodología*. Recuperado el 20 de marzo de 2021, de <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>

Agile Business Consortium Limited. (2021). *What is DSDM?* Recuperado el 12 de marzo de 2021, de <https://www.agilebusiness.org/page/whatisdsdm#:~:text=What%20is%20DSDM%3F-,DSDM%20is%20an%20Agile%20method%20that%20focuses%20on%20the%20full,new%20iterative%20way%20of%20working.>

Arevalo Lizardo, M. E. (15 de noviembre de 2011). *Diferencias entre metodologías tradicionales y ágiles*. Recuperado el 14 de marzo de 2021, de <https://arevalomaria.wordpress.com/2011/11/15/diferencias-entre-metodologias-tradicionales-y-agiles-metodologiasagiles/>

Bixpe. (19 de marzo de 2019). *Empresa*. Recuperado el 10 de enero de 2021, de <https://www.bixpe.com/empresa/>

Calvo, D. (07 de abril de 2018). *Metodología XP programación extrema (metodología ágil)*. Recuperado el 20 de marzo de 2021, de <https://www.diegocalvo.es/metodologia-xp-programacion-extrema-metodologia-agil/#:~:text=Definici%C3%B3n,para%20la%20gesti%C3%B3n%20de%20proyectos.&text=Esta%20metodolog%C3%ADa%20pone%20el%20%C3%A9nfasis,requisitos%20imprecisos%20y%20muy%20cambiantes.>

Canós, J. H., Letelier, P., & Penadés, M. C. (2003). *Métodologías ágiles en el desarrollo de software*. Recuperado el 20 de marzo de 2021, de <https://docplayer.es/704660-Metodologias-agiles-en-el-desarrollo-de-software.html>

Castillo, O., Figueroa, D., & Sevilla, H. (s.f.). *Fases de la programación extrema*.

Recuperado el 14 de marzo de 2021, de <https://programacionextrema.tripod.com/fases.htm>

Cuofano, G. (s.f.). *Crystal Agile Framework In A Nutshell*. Recuperado el 18 de marzo de 2021, de FourWeekMBA: <https://fourweekmba.com/crystal-agile-framework/>

Diaz, A. (19 de enero de 2017). *¿Qué son los Logs y por qué deben interesarte?* Recuperado el 19 de marzo de 2021, de <https://dbibyhas.io/es/blog/que-son-los-logs/>

Duero, L. D. (21 de junio de 2018). *Dynamic system development method*. Recuperado el 18 de marzo de 2021, de SlideShare: <https://www.slideshare.net/LennonDukeDuero/dynamic-system-development-method-dsdm-102751942>

Ecuador, Empresa Eléctrica Quito. (2020). *Misión*. Recuperado el 24 de marzo de 2021, de <http://www.eeq.com.ec/laEmpresa/mision.php?mn=1>

Ecuador, Empresa Eléctrica Quito. (2020). *Políticas*. Recuperado el 10 de abril de 2021, de <http://www.eeq.com.ec/laEmpresa/politicas.php?mn=1>

Ecuador, Empresa Eléctrica Quito. (2020). *Visión*. Recuperado el 24 de marzo de 2021, de <http://www.eeq.com.ec/laEmpresa/vision.php?mn=1>

Ecuador, Ministerio de Trabajo. (2020). *Directrices para la aplicación de teletrabajo emergente durante la declaratoria de emergencia sanitaria*. Quito, Ecuador: Acuerdo ministerial N° MDT-2020-076.

Ecuador, Ministerio del Trabajo. (junio de 2020). *Valores / Misión / Visión*. Recuperado el 22 de marzo de 2021, de <https://www.trabajo.gob.ec/valores-mision-vision/#:~:text=%E2%80%9CSomos%20la%20Instituci%C3%B3n%20rectora%20de,modelos%20de%20gesti%C3%B3n%20integral%2C%20para>

Ecured. (4 de agosto de 2019). *Diagrama de despliegue*. Recuperado el 19 de marzo de 2021, de https://www.ecured.cu/index.php?title=Diagrama_de_despliegue&oldid=3488

755

Equipo Geek. (08 de abril de 2020). *Qué es NodeJS y primeros pasos*. Recuperado el 20 de marzo de 2021, de <https://ifgeekthen.everis.com/es/que-es-node-js-y-primeros-pasos>

Fazt Blog. (08 de septiembre de 2017). *Curso de Express.js, el Framework de Nodejs | ¿Qué es Express?* Recuperado el 14 de marzo de 2021, de <https://blog.faztweb.com/2017/09/expressjs-el-framework-de-nodejs-que-es.html>

Figueroa Díaz, R., Solís, C., & Cabrera, A. (2007). *Metodologías tradicionales vs metodologías ágiles*. Recuperado el 19 de abril de 2021, de https://www.researchgate.net/profile/Roberth-Figueroa-Diaz/publication/299506242_METODOLOGIAS_TRADICIONALES_VS_METODOLOGIAS_AGILES/links/56fc17ff08ae8239f6dc4498/METODOLOGIAS-TRADICIONALES-VS-METODOLOGIAS-AGILES?_sg%5B0%5D=72MQ5g9DcWESvmpIz4AcVdm6k-4RqDjh

Garzas, J. (2021). *Método de estimación de puntos de caso de uso*. Recuperado el 14 de marzo de 2021, de <http://233gradosdeti.com/articulos/metodo-de-estimacion-de-puntos-de-caso-de-uso/>

Internalia Group. (2019). *Nosotros*. Recuperado el 16 de marzo de 2021, de <https://www.internaliagroup.com/#nosotros>

Isla Visual. (2014). *Diferencias entre Scrum y XP*. Recuperado el 22 de marzo de 2021, de https://islavisual.com/articulos/desarrollo_web/diferencias-entre-scrum-y-xp.php

Kopca, T. (04 de noviembre de 2020). *El concepto de Modelo-Vista-Controlador (MVC) explicado*. Recuperado el 16 de marzo de 2021, de <https://www.mano.org/es/software/desarrollo/el-concepto-de-modelo-vista-controlador-mvc-explicado>

López Roldán, J. J. (1 de noviembre de 2020). *Instalación y Configuración de Oracle 12c en Debian 8 – Jessie*. Recuperado el 16 de marzo de 2021, de El Rincón

De Juanjo: <https://juanjoselo.wordpress.com/2017/10/25/instalacion-y-configuracion-de-oracle-12c-en-debian-8-jessie/>

López Sosa, A. (22 de mayo de 2019). *Introducción a express js*. Recuperado el 17 de marzo de 2021, de <https://medium.com/@aarnlpezsosa/introducci%C3%B3n-a-express-js-a1ebe16dbcf4>

López, L., Pereira, C., & Vidal, C. (2014). Propuesta y aplicación de diagramas de clases UML JPI. *Información Tecnológica*, 25(5), 113-120. Recuperado el 15 de marzo de 2021, de https://scielo.conicyt.cl/scielo.php?pid=S0718-07642014000500016&script=sci_arttext&tIng=en

López, X., Rubio, J., Vidal Silva, C., & Villarroel, R. (febrero de 2019). Una propuesta de algoritmo Spin/Promela para el análisis y diagnóstico de errores en diagramas de secuencia UML. *Información tecnológica*, 263-272. Recuperado el 12 de Febrero de 2021, de https://scielo.conicyt.cl/scielo.php?pid=S0718-07642019000100263&script=sci_arttext&tIng=en

Lucid Software. (2021). *Diagramación inteligente para todos los equipos*. Recuperado el 22 de marzo de 2021, de <https://www.lucidchart.com/pages/es>

Maida, E. G., & Pacienza, J. (2015). *Metodologías de desarrollo de software*. Recuperado el 12 de marzo de 2021, de <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>

Noren, A. (2020). *Requerimientos funcionales y no funcionales, ejemplos y tips*. Recuperado el 19 de marzo de 2021, de <https://requeridos.com/requerimientos-funcionales-y-no-funcionales/>

Oracle. (2021). *Definición de base de datos*. Recuperado el 24 de marzo de 2021, de <https://www.oracle.com/mx/database/what-is-database/>

Oriol, E. (2 de marzo de 2020). *Angular 9: Lo más destacado*. Recuperado el 19 de marzo de 2021, de <http://blog.enriqueoriol.com/2020/03/angular-9-novedades.html>

- Ortíz, K. H. (2021). *Plataforma para el control del uso de softwares educativos*. Recuperado el 16 de marzo de 2021, de <https://www.eumed.net/libros-gratis/2009c/583/Requerimientos%20no%20Funcionales.htm>
- PMOinformatica. (13 de abril de 2015). *Requerimientos no funcionales: Una clasificación*. Recuperado el 14 de marzo de 2021, de <http://www.pmoinformatica.com/2015/04/requerimientos-no-funcionales-una.html>
- Rosales Lopez, P. P., Salas Bacalla, J., & Tinoco Gomez, O. (julio de 2010). Criterios de selección de metodologías de desarrollo de software. *Industrial Data, XIII(2)*, 70-74. Recuperado el 10 de marzo de 2021, de <https://www.redalyc.org/pdf/816/81619984009.pdf>
- Saraclip. (15 de junio de 2017). *Requerimientos de un proyecto*. Recuperado el 12 de marzo de 2021, de <https://www.saraclip.com/requerimientos-de-un-proyecto/>
- Serrano Mascaraque, E. (5 de agosto de 2009). Accesibilidad vs usabilidad web: evaluación y correlación. *Investigación bibliotecológica, 23(48)*, 61-103. Recuperado el 16 de marzo de 2021, de <http://www.scielo.org.mx/pdf/ib/v23n48/v23n48a4.pdf>
- Tecnologías Información. (2018). *Modelos de datos: Modelo conceptual, físico y lógico*. Recuperado el 08 de marzo de 2021, de <https://www.tecnologias-informacion.com/modelos-datos.html>
- Testing Baires. (26 de febrero de 2017). *Pruebas de caja negra y un enfoque práctico*. Recuperado el 10 de marzo de 2021, de <https://testingbaires.com/2017/02/26/pruebas-caja-negra-enfoque-practico/#:~:text=Las%20Pruebas%20de%20Caja%20Negra,ejecuci%C3%B3n%20internos%20en%20el%20software.>
- Vargas Cordero, Z. R. (2009). La investigación aplicada: una forma de conocer las realidades con evidencia científica. *Educación, 33(1)*, 155-165. Recuperado el 10 de abril de 2021, de <https://www.redalyc.org/pdf/440/44015082010.pdf>
- Villatoro, G. (26 de abril de 2018). *Node.js anuncia la primera versión en su línea 10.x*.

Recuperado el 13 de marzo de 2021, de <https://www.lomasnuevo.net/noticias/node-js-anuncia-la-primer-version-en-su-linea-10-x/>

Virender Singh. (01 de octubre de 2019). *Crystal method in agile*. Recuperado el 15 de marzo de 2021, de <https://www.toolsqa.com/agile/crystal-method/>

ANEXOS

Anexo 1. Historia de usuario – autenticar usuario

Historia de usuario	
Número: 1	Nombre: Autenticar Usuario
Usuario: Cliente – Lcdo. Paul Dávila	
Modificación de historia número: N/A	Iteración asignada: 1
Prioridad en Negocio: Media	Puntos estimados: 2
Riesgo en desarrollo: Baja	Puntos Reales: 2
Programador responsable: José Emilio Fonseca	
Descripción: Cliente manifiesta que es necesario que los empleados se autentifiquen con el rol asignado por la empresa y la misma clave del correo electrónico.	
Observaciones:	

Elaborado por: José Emilio Fonseca

Anexo 2. - historia de usuario – gestionar asistencia

Historia de usuario	
Número: 2	Nombre: Gestionar Asistencia
Usuario: Cliente – Lcdo. Paul Dávila	
Modificación de historia número: N/A	Iteración asignada: 2
Prioridad en Negocio: Alta	Puntos estimados: 2
Riesgo en desarrollo: Baja	Puntos Reales: 2
Programador responsable: José Emilio Fonseca	
Descripción: Cliente manifiesta que es necesario que los empleados realicen el registro de asistencia de entrada y salida a través del aplicativo, independientemente de ser un empleado con jefatura.	
Observaciones:	

Elaborado por: José Emilio Fonseca

Anexo 3. Historia de usuario – gestionar actividades

Historia de usuario	
Número: 3	Nombre: Gestionar Actividades
Usuario: Cliente – Lcdo. Paul Dávila	
Modificación de historia número: N/A	Iteración asignada: 3
Prioridad en Negocio: Alta	Puntos estimados: 2
Riesgo en desarrollo: Baja	Puntos Reales: 2
Programador responsable: José Emilio Fonseca	
Descripción: Cliente manifiesta que es necesario que los empleados realicen el registro de actividades en periodos de 7 días y que en cada día se pueda ingresar como un máximo de 5 actividades diarias. Posterior a esto se envíe todas las actividades de ese periodo a la jefatura correspondiente.	
Observaciones:	

Elaborado por: José Emilio Fonseca

Anexo 4. Historia de usuario – gestionar aprobaciones

Historia de usuario	
Número: 4	Nombre: Gestionar Aprobaciones
Usuario: Cliente – Lcdo. Paul Dávila	
Modificación de historia número: N/A	Iteración asignada: 4
Prioridad en Negocio: Alta	Puntos estimados: 2
Riesgo en desarrollo: Baja	Puntos Reales: 2
Programador responsable: José Emilio Fonseca	
Descripción: Cliente manifiesta que es necesario que las jefaturas de las distintas áreas tengan la potestad de aprobar o devolver cada una de las actividades enviadas, esto con el fin de que si son aprobadas pasen a potestad del departamento de Talento Humano y si son devueltas los empleados a cargo puedan corregir las actividades.	
Observaciones:	

Elaborado por: José Emilio Fonseca

Anexo 5. Historia de usuario – gestionar reportes

Historia de usuario	
Número: 5	Nombre: Gestionar Reportes
Usuario: Cliente – Lcdo. Paul Dávila	
Modificación de historia número: N/A	Iteración asignada: 5
Prioridad en Negocio: Alta	Puntos estimados: 2
Riesgo en desarrollo: Baja	Puntos Reales: 2
Programador responsable: José Emilio Fonseca	
Descripción: Cliente manifiesta que es necesario que los empleados sean capaces de descargar un reporte en formato Excel de sus periodos con las actividades correspondientes y que los empleados jefes puedan descargar un reporte personalizado bajo parámetros.	
Observaciones: Los parámetros para reporte personalizado de empleado jefe son: Selección de empleado a cargo, fecha inicio, fecha fin y estado.	

Elaborado por: José Emilio Fonseca

Anexo 6. Tarjeta crc – autenticar usuario

Usuario	
Funcionalidades	Colaboraciones
Validar Usuario Ldap	Empleado

Elaborado por: José Emilio Fonseca

Anexo 7. Tarjeta crc – gestionar asistencias

Asistencia	
Funcionalidades	Colaboraciones
Crear asistencia entrada	Empleado
Crear asistencia salida	Empleado Jefe
Consultar asistencias	

Elaborado por: José Emilio Fonseca

Anexo 8. Tarjeta crc – gestionar actividades

Actividades	
Funcionalidades	Colaboraciones
Crear periodos Listar periodos Eliminar periodo Buscar periodo Enviar periodo a jefatura Crear actividad Eliminar actividad Buscar actividad Listar actividades Modificar actividades	Empleado Empleado Jefe

Elaborado por: José Emilio Fonseca

Anexo 9. Tarjeta crc – gestionar aprobaciones

Actividades	
Funcionalidades	Colaboraciones
Listar periodos recibidos Aprobar periodos recibidos Devolver periodos recibidos	Empleado Jefe

Elaborado por: José Emilio Fonseca

Anexo 10. Tarjeta crc – gestionar reportes

Actividades	
Funcionalidades	Colaboraciones
Exportar reporte actividades por periodo Exportar reporte personalizado	Empleado Empleado Jefe

Elaborado por: José Emilio Fonseca