

UNIVERSIDAD TECNOLÓGICA ISRAEL

INGENIERÍA EN SISTEMAS INFORMÁTICOS

PROYECTO INTEGRADOR DE SABERES INVESTIGACIÓN Y PRÁCTICAS

Sistema interprete de datos suministrados por reloj biométrico

David A. Taday

Alex F. Sandoval

Andrés F. Reinoso

Ing. Martha San Andrés, MSc.

D. M. Quito, 24 de febrero del 2018

Contenido

[Plan 5](#_Toc507109830)

[1. Título del Proyecto 5](#_Toc507109831)

[2. Definición y Justificación del problema 5](#_Toc507109832)

[2.1. Definición del negocio 5](#_Toc507109833)

[2.2. Justificación del problema 5](#_Toc507109834)

[3. Objetivos 6](#_Toc507109835)

[3.1. Principal 6](#_Toc507109836)

[3.2. Secundarios 6](#_Toc507109837)

[4. Marco teórico introductorio 6](#_Toc507109838)

[4.1. Ingeniería de requerimientos 6](#_Toc507109839)

[5. Hipótesis o ideas a defender 7](#_Toc507109840)

[6. Metodología investigativa 7](#_Toc507109841)

[Para el desarrollo del presente proyecto se han hecho uso de diversos conceptos y metodologías de investigación y desarrollo, los cuales han sido listados en el marco referencial, en tal sentido como primer punto se realizó una investigación de la empresa Dentales Pablo Herman S.A. para obtener los parámetros sobre los cuales se desarrollará el proyecto, posterior a ello se realizaron entrevistas a los usuarios como se señala en los archivos anexos, de esa investigación se pudieron obtener herramientas de diagnostico como el grafico de espinas de pescado, con lo cual se pudo obtener los requerimientos funcionales y no funcionales. 7](#_Toc507109842)

[Metodología ágil Scrum 7](#_Toc507109843)

[6.1. Identificación de actores principales 8](#_Toc507109844)

[7. Resultados esperados 8](#_Toc507109845)

[8. Marco Referencial 9](#_Toc507109846)

[Sistemas similares que existen en el mercado, comparativos. 9](#_Toc507109847)

[En este campo descriptivo, se muestra un cuadro comparativo en función del Proyecto *“Sistema Intérprete de Datos suministrados por un Reloj Biométrico”*, se eligió dos Sistema los cuales se enfocan en funcionalidades similares a la del Proyecto. 9](#_Toc507109848)

[Reloj Biométrico de Control de Personal, Biotech IN04. 9](#_Toc507109849)

[9. Bibliografía 10](#_Toc507109850)

[INFORME DE RESULTADOS DEL PROYECTO 11](#_Toc507109851)

[1. Título del Proyecto 11](#_Toc507109852)

[2. Introducción 11](#_Toc507109853)

[La empresa Dentales Pablo Herman S.A. es una compañía localizada el distrito metropolitano de Quito, En la actualidad su actividad comercial se basa importación y comercialización al por mayor y menor de equipamiento médico, odontológico y laboratorio forense, debido a ello su manejo de talento humano se ha vuelto mas complejo con el paso de los años, al momento se hace uso de un reloj biométrico que suministra un archivo poco administrable, debido a ello el presente proyecto se ha desarrollado con el objetivo de dar solución a la problemática de análisis de datos suministrarlos por el reloj biométrico. 11](#_Toc507109854)

[Sección 1 11](#_Toc507109855)

[3. Marco Teórico 11](#_Toc507109856)

[3.1. Software. - Definición. - 11](#_Toc507109857)

[3.2. Enfoques de calidad 14](#_Toc507109858)

[3.3. METOGOLOGIA DE DESARROLLO AGIL SCRUM 16](#_Toc507109859)

[3.4. Diagramación 23](#_Toc507109860)

[3.5. Diagrama de flujo y participantes 24](#_Toc507109861)

[3.6. Técnica del diagrama causa-efecto: 24](#_Toc507109862)

[3.7. Arquitectura del Software 26](#_Toc507109863)

[3.8. Verificación y validación 26](#_Toc507109864)

[3.9. Prototipado 27](#_Toc507109865)

[3.10. Control de Versiones - GIT 30](#_Toc507109866)

[Para el desarrollo del presente proyecto se ha hecho uso de herramientas de control de versiones como GIT, Esta herramienta permite alojar los diferentes proyectos de trabajo, permitiendo gestionar de forma muy simple y gráfica las diferentes versiones o modificaciones que se vayan realizando. Git es un sistema de control de versiones distribuido, mediante el cual podemos trabajar a nivel local sin tener que comprometer el repositorio principal a cada cambio que realicemos. 30](#_Toc507109867)

[Gracias a este sistema, los desarrolladores del proyecto trabajaron de forma independiente hasta el momento en el que tengan que poner en común con el resto del equipo su código, controlando los cambios en las versiones y mejorando el seguimiento al desarrollo. 30](#_Toc507109868)

[3.11. PRUEBAS DE ACEPTACION 30](#_Toc507109869)

[3.12. IDE – Netbens 30](#_Toc507109870)

[3.13. My SQL 30](#_Toc507109871)

[4. Marco Legal 30](#_Toc507109872)

[5. Marco referencial 35](#_Toc507109873)

[6. Marco ambiental 36](#_Toc507109874)

[7. Marco tecnológico 37](#_Toc507109875)

[7.1. Terminología 37](#_Toc507109876)

[7.1.1. Modelo de Casos de Uso del Negocio 38](#_Toc507109877)

[7.1.2. Modelo de Objetos del Negocio 38](#_Toc507109878)

[7.1.3. Modelo de Casos de Uso 38](#_Toc507109879)

[7.1.4. Especificaciones de Casos de Uso 38](#_Toc507109880)

[7.1.5. Especificaciones Adicionales 39](#_Toc507109881)

[7.1.6. Modelo de Análisis y Diseño 39](#_Toc507109882)

[7.1.7. Modelo de Datos 39](#_Toc507109883)

[7.1.8. Modelo de Implementación 39](#_Toc507109884)

[7.1.9. Casos de Prueba 39](#_Toc507109885)

[7.1.10. Plan de Iteración 40](#_Toc507109886)

[7.1.11. Material de Apoyo al Usuario Final 40](#_Toc507109887)

[7.1.12. Producto 40](#_Toc507109888)

[7.1.13. UML 40](#_Toc507109889)

[Sección II Diagnostico 41](#_Toc507109890)

[8. Diagrama de flujo y participantes 41](#_Toc507109891)

[8.1. Definición de requerimientos funcionales y no funcionales 46](#_Toc507109892)

[Diagnóstico del Problema: 46](#_Toc507109893)

[Sección III Propuesta 49](#_Toc507109894)

[9. Solución.- 49](#_Toc507109895)

[Para la solución del problema del cliente final Dentales Pablo Herman S.A. se ha desarrollarlo el sistema interprete de datos de reloj biométrico, para el cual se ha utilizado diversas metodologías investigativas para obtener la mayor cantidad de información posible para que sea analizada y contextualizada en el presente proyecto todo con el objetivo de solucionar el problema planteado por el cliente en el diseño y la implementación del sistema. 49](#_Toc507109896)

[9.1. Uso de Scrum 49](#_Toc507109897)

[Para aquello el presente proyecto se desarrolló bajo la metodología de desarrollo ágil SCRUM la cual se encuentra ampliamente expuesta en documento adjunto, donde se han hecho uso de las herramientas product Backlog y Sprint Backlog que fueron desarrollados en base a los requerimientos funcionales y no funcionales antes descritos. 49](#_Toc507109898)

[9.2. Control de versiones.- 49](#_Toc507109899)

[Con el objetivo de manejar de manera más organizada las versiones de los diferentes sprints que se realizaron en el presente proyecto se hizo uso de herramientas de control de versiones, es el caso de GIT y GIT HUB, de donde además de controlar las versiones se pudo hacer un seguimiento de los progresos realizados por cada uno de los participantes del proyecto, lo cual se verifica en archivo anexo 49](#_Toc507109900)

[9.3. IDE Desarrollo 50](#_Toc507109901)

[IDE DESARROLLO 50](#_Toc507109902)

[MY SQL 50](#_Toc507109903)

[PLATAFORMA DE DESARROLLO SWING 50](#_Toc507109904)

[Cabe destacar que la presente herramienta de software fue desarrollada bajo la metodología Scrum, utilizando todos los parámetros de ella, los cuales se detallan en los anexos. 50](#_Toc507109905)

[9.4. PROTOTIPO 50](#_Toc507109906)

[Con el objetivo de validar que los requerimientos fueron correctamente interpretados por el Product Owner, se procedió a realizar prototipos de funcionalidades del software, el objetivo de los prototipos del software es mostrar al usuario final las funcionalidades más relevantes en relación a los requisitos funcionales y parte de los no funcionales, tomando en cuenta la prioridad de estos siendo los de categoría 5 como los más primordiales, para este proceso se realizó utilizando el prototipado de estilo vertical. 50](#_Toc507109907)

[Los presentes prototipos sirvieron de base para que el usuario final haga sus observaciones y sugerencias acerca del desarrollo del sistema en donde se puede observar del acta de revisión adjunta. 53](#_Toc507109908)

[Conclusiones y recomendaciones 53](#_Toc507109909)

[10. Conclusiones 53](#_Toc507109910)

[11. Recomendaciones 54](#_Toc507109911)

[Bibliografía 54](#_Toc507109912)

[ANEXOS 56](#_Toc507109913)

Plan

1. Título del Proyecto

Sistema interprete de datos suministrados por reloj biométrico.

Quito-Ecuador

1. Definición y Justificación del problema
   1. Definición del negocio

La empresa Dentales Pablo Herman S.A. es una compañía localizada el distrito metropolitano de Quito, específicamente en el centro-norte de la capital, sector de Santa Clara, su actividad comercial se basa importación y comercialización al por mayor y menor de equipamiento médico, odontológico y laboratorio forense, con su respectivo servicio técnico profesional.

* 1. Justificación del problema

La empresa Dentales Pablo Herman S.A. a través de su departamento de recursos humanos realiza el control de personal apoyándose en un reloj biométrico, dentro de las actividades del operador de este departamento está la emisión de reportes semanales y mensuales con relación a la información de control de personal tales como:

* Registro de ingreso y salida del personal
* Carga de horas extras
* Verificación de atrasos
* Verificación de horas trabajadas, etc.

El problema detectado en el proceso de elaboración de reportes es principalmente el excesivo tiempo que toma el realizarlos, debido a que la información que se sube desde el reloj biométrico hacia el computador es muy básica sin orden ni formato, en tal sentido la información dada requiere de un análisis extenso por parte del personal del departamento.

1. Objetivos
   1. Principal

* Interpretar datos suministrados por un reloj biométrico y presentar reportes de manera estructurada y amigable para el usuario.
  1. Secundarios
* Simplificar el trabajo del usuario y realzar diversos tipos de consultas y reportes con los datos administrados
* Optimizar el proceso de trabajo del personal de recursos humanos de la empresa.

1. Marco teórico introductorio
   1. Ingeniería de requerimientos

El proceso de recopilar, analizar y verificar las necesidades del cliente para un sistema de software es conocido como Ingeniería de Requerimientos. La meta del mismo es entregar una especificación de requerimientos de software correcta y completa. La ingeniería de requerimientos apunta a mejorar la forma en que comprendemos y definimos sistemas de software complejos.

Para todo este proceso se ha utilizado la Especificación de Requisitos Software según el estándar de IEEE 830.

Una buena especificación de requisitos software ofrece una serie de ventajas entre las que destacan el contrato entre cliente y desarrolladores, la reducción del esfuerzo en el desarrollo, una buena base para la estimación de costes y planificación, un punto de referencia para procesos de verificación y validación, y una base para la identificación de posibles mejoras en los procesos analizados.

Las características deseables para una buena especificación de requisitos software que se indican en el IEEE son las siguientes [Chalmeta, 2000][Piattini, 1996]:

* + Correcta
  + No ambigua
  + Completa
  + Verificable
  + Consistente
  + Clasificada
  + Modificable
  + Explorable
  + Utilizable durante las tareas de mantenimiento y uso

1. Hipótesis o ideas a defender

La implementación del sistema de gestión de horarios permite interpretar los datos relacionados con la administración de recursos humanos de la compañía Dentales Pablo Herman S.A., esto con el objetivo de brindar una mejor interpretación de los datos y presentarlos de manera más grafica para un análisis más sencillo por parte usuario.

1. Metodología investigativa

Para el desarrollo del presente proyecto se han hecho uso de diversos conceptos y metodologías de investigación y desarrollo, los cuales han sido listados en el marco referencial, en tal sentido como primer punto se realizó una investigación de la empresa Dentales Pablo Herman S.A. para obtener los parámetros sobre los cuales se desarrollará el proyecto, posterior a ello se realizaron entrevistas a los usuarios como se señala en los archivos anexos, de esa investigación se pudieron obtener herramientas de diagnostico como el grafico de espinas de pescado, con lo cual se pudo obtener los requerimientos funcionales y no funcionales.

Metodología ágil Scrum

Toda vez que se ha realizado un diagnostico se procede a utilizar la metodología Scrum la cual se utiliza con el objetivo de evitar situaciones en que no se este entregando al cliente lo que necesita, cuando las entregas se alarguen demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesite capacidad de reacción ante la competencia, cuando la moral de los equipos este baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto.

* 1. Identificación de actores principales

El sistema interprete de datos suministrados por reloj biométrico tiene dos tipos de usuarios que a continuación se describen a cada uno de ellos, así como las actividades que realizan respectivamente.

|  |  |
| --- | --- |
| Jefe de Departamento | *El usuario que esta como jefe del departamento, es el que cuenta con todos los privilegios modificación eliminación y sobre escritura, así mismo puede utilizar todas las funciones del sistema* |
| Actividad | *Además de las especificaciones anteriores podrá:*  *-Cambiar contraseñas.*  *-Modificar el nombre del Jefe de departamento.*  *-Consulta de reportes*  *-Generación de reportes* |
| Administrador RRHH | *El usuario administrador RRHH es quién podrá crear los reportes, solo podrá consultar y crear los reportes con los datos que se encuentren en la base de datos actual.* |
| Actividad | *-Ingresar al sistema bajo una validación del usuario y contraseña.*  *-Pre visualizar el reporte para posible corrección.*  *-Impresión del reporte.*  *-Consulta de Reportes según (rango, fechas, empleados, etc.)* |

1. Resultados esperados

El presente documento tiene como finalidad dar a conocer al usuario final las propuestas establecidas dentro de la interpretación, como del análisis de resultados podemos citar la metodología empleada, ya que gracias a los métodos inductivo y deductivo se pudo mejorar la investigación para una mejor interpretación. Con la aplicación de herramientas como lo es una encuesta la cual contó con preguntas cerradas, permitió obtener datos referenciales de la muestra., con el único objetivo de cumplir con lo planteado y así lograr un aporte importante para la empresa.

1. Marco Referencial

Sistemas similares que existen en el mercado, comparativos.

En este campo descriptivo, se muestra un cuadro comparativo en función del Proyecto *“Sistema Intérprete de Datos suministrados por un Reloj Biométrico”*, se eligió dos Sistema los cuales se enfocan en funcionalidades similares a la del Proyecto.

Reloj Biométrico de Control de Personal, Biotech IN04.

|  |  |  |
| --- | --- | --- |
| COMPARATIVA | Biotech IN04 | DSoftware |
| Software de terceros | Software propio |
| NOTIFICACION DE NOVEDADES | Área de notificaciones, se despliega un campo que contiene todas las notificaciones con respecto al control de asistencia. | En cuanto a novedades del empleado el Sistema en modulos anexos por desarrollarse podrá enviar un correo electrónico cuando se registre alguna de las siguientes acciones: atraso o falta al día de trabajo por parte de un trabajador. |
| MODIFICACION | Control de Sistema Automatizado, el Sistema se maneja de forma independiente con la facultad de ofrecer recursos limitados de Funcionalidad | El registro datos podrá configurarse para realizarse de forma automatizada a su registro, o manualmente en lotes. |
| El sistema también permitirá el registro de asistencias manuales, sin embargo, estas requerirán autorización por parte del Gerente. |
| EXCLUSIVIDAD | Evita el fraude que se puede dar por el intercambio de tarjetas o pines de acceso. Ningún empleado podrá “marcarle” a otro empleado. | El campo fecha de consulta acepta únicamente fechas anteriores al día de hoy (día actual). |
| Sistema de Encriptación Básica para acceso a su base de Datos de ingreso personal. | Los usuarios deben ingresar al sistema con un nombre de usuario y contraseña. |

1. Bibliografía

* Alarcon, A., & Sandoval, E. (2011). Herramientas CASE para ingeniería de Requisitos. Cultura Científica, 6(6), 70–74.
* Jesús Bustínduy Basterrechea. (s/f). Desarrollo detallado de la fase de aprobación de un proyecto informático. 2004. Recuperado a partir de http://openaccess.uoc.edu/webapps/o2/bitstream/10609/575/1/27733tfc.pdf
* Méndez, G. (s/f). Ingeniería de Requisitos. Recuperado a partir de https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/03-Requisitos.pdf
* Molina Quiroga, E. P. (2015). Los principios del derecho colectivo de trabajo en la constitución, en el código de trabajo y en el proyecto de código orgánico de relaciones laborales (B.S. thesis). Guayaquil: Universidad Del Pacífico. Recuperado a partir de http://repositorio.upacifico.edu.ec/handle/40000/129
* Rodríguez, J. B. (s/f-a). Trabajo Fin de Grado Grado en Ingeniería de Organización Industrial. Recuperado a partir de <http://bibing.us.es/proyectos/abreproy/91033/fichero/91033.pdf>
* Mendoza González, G. (2008). Herramienta de Desarrollo Netbeans. Colombia.

INFORME DE RESULTADOS DEL PROYECTO

1. Título del Proyecto

Sistema interprete de datos suministrados por reloj biométrico.

1. Introducción

La empresa Dentales Pablo Herman S.A. es una compañía localizada el distrito metropolitano de Quito, En la actualidad su actividad comercial se basa importación y comercialización al por mayor y menor de equipamiento médico, odontológico y laboratorio forense, debido a ello su manejo de talento humano se ha vuelto mas complejo con el paso de los años, al momento se hace uso de un reloj biométrico que suministra un archivo poco administrable, debido a ello el presente proyecto se ha desarrollado con el objetivo de dar solución a la problemática de análisis de datos suministrarlos por el reloj biométrico.

Sección 1

1. Marco Teórico

Con el objeto de proseguir el desarrollo de esta investigación los autores han realizado una exhaustiva investigación de conceptos, normativas y procedimientos que se deben realizar previo al desarrollo del software propuesto, en tal sentido se es menester profundizar en el estudio de lo siguiente:

* 1. Software. - Definición. -

El software es una parte integral de la mayoría de los sistemas y por lo tanto se requiere ejecutar proyectos de software de forma satisfactoria y de alta calidad. El software son los programas y la documentación asociada tal como requisitos, modelos de diseño y manuales de usuario.

Los productos de software pueden ser de varios tipos tal que:

**Genéricos:** Desarrollados para ser vendidos a un ámbito de clientes diferentes

**Hechos a medida:** En este apartado es software desarrollado a personalización del cliente y sus respectivas especificaciones

Para la creación de software se lo puede realizar a partir de cero, configurando sistemas de software genéricos o reutilizando código y mejorándolo como tal.

De la investigación realizada se desprende que existen ciertos tipos de atributos que facultan a un software como bueno, entre los cuales tenemos:

**Mantenibilidad:** Evolución constate

**Fiabilidad:** Digno de confianza

**Eficiencia:** Uso eficaz de los recursos del sistema

**Aceptabilidad:** Aceptación de los usuarios

El software puede definirse como el conjunto de tres componentes

**Programas:** Conjuntos de instrucciones que proporcionan la funcionalidad deseada y el rendimiento cuando se ejecutan.

**Datos:** Datos necesarios para manejar y probar los programas y las estructuras requeridas para mantener y manipular estos datos.

**Documentos:** Operación y uso del programa, son todos los documentos necesarios para poder ejecutar modificar y mantener el software.

**Características del software**

Para poder comprender lo que es realmente el software es importante conocer de forma exhaustiva algunas características que lo diferencian de otras cosas que el hombre pude construir.

**El Software se desarrolla**

Construir un producto software implica entender que es necesario, se lo debe realizar en función de que cumpla los requisitos que el cliente está solicitando, en el software el recurso principal son las personas, no siempre es posible acelerar la construcción del software añadiendo personas porque el desarrollo del software requiere un esfuerzo de equipo.

**El Software no se estropea**

Los defectos no detectados harán que falle el programa durante las primeras etapas de su vida, sin embargo, una vez que se corrigen los fallos disminuyen.

El software no se estropea, pero se deteriora. Conforme se realicen cambios es bastante probable que se introduzcan nuevos defectos, lo que hace que el software se deteriore con el tiempo.

**El software se construye a medida**

La mayoría de software se construye a medida. Aunque la reutilización y ensamblaje de componentes está aumentando, el software con frecuencia se construye de acuerdo a los requisitos específicos de un cliente.

Las ventajas de construir el software a medida son:

* Reduce el coste y el tiempo de lanzamiento al mercado
* Buena calidad
* Ingeniería rápida
* Mantenimiento
* Fácil mejora

**Definición de Ingeniería del Software**

La ingeniería del software es la aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento del software, que es la aplicación de la ingeniería del software (IEEE, 1990).

**Mitos del software**

Los mitos tienen cierto número de atributos que los hacen insidiosos. Por ejemplo, parecen enunciados razonables de hechos (a veces contienen elementos de verdad), tienen una sensación intuitiva y es frecuente que los manifiesten profesionales experimentados que “conocen la historia”

Para entender de mejor manera revisaremos los siguientes ejemplos:

**Mitos del Gerente**

**Mito:** Tenemos un libro lleno de estándares y procedimientos para elaborar software. ¿No le dará a mi personal todo lo que necesita saber?

**Realidad:** Tal vez exista el libro de estándares, pero ¿se utiliza? ¿Saben de su existencia los trabajadores del software? ¿Refleja la práctica moderna de la ingeniería de software? ¿Es completo? ¿Es adaptable? ¿Está dirigido a mejorar la entrega a tiempo y también se centra en la calidad? En muchos casos, la respuesta a todas estas preguntas es “no”.

**Mitos del cliente**

**Mito:** Para comenzar a escribir programas, es suficiente el enunciado general de los objetivos —podremos entrar en detalles más adelante.

**Realidad:** Aunque no siempre es posible tener el enunciado exhaustivo y estable de los requerimientos, un “planteamiento de objetivos” ambiguo es una receta para el desastre. Los requerimientos que no son ambiguos (que por lo general se obtienen en forma iterativa) se desarrollan sólo por medio de una comunicación eficaz y continua entre el cliente y el desarrollador

**Mitos del profesional**

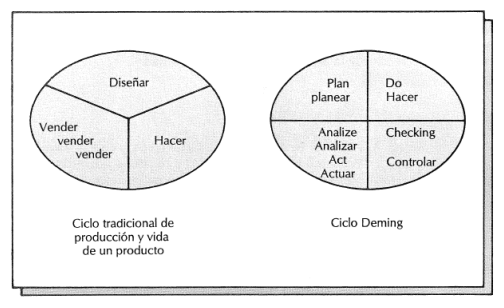
**Mito:** Una vez que escribimos el programa y hacemos que funcione, nuestro trabajo ha terminado.

**Realidad:** Alguien dijo alguna vez que “entre más pronto se comience a ‘escribir el código’, más tiempo tomará hacer que funcione”. Los datos de la industria indican que entre 60 y 80% de todo el esfuerzo dedicado al software ocurrirá después de entregarlo al cliente por primera vez.

(Pressman, 2014)

* 1. Enfoques de calidad

Para el desarrollo del presente proyecto los autores realizaron un estudio de los enfoques de calidad sobre los cuales se debe realizar el análisis, en tal sentido; fruto de esa investigación se ha decidido que el Ciclo Deming, desarrollado por Shewhart, es un modelo de proceso administrativo que será utilizado, en mismo que está dividido en cuatro fases, como se muestra en el grafico N°1



*Grafico N° 1 – Enfoque de calidad*

Profundizando el enfoque de calidad se ha establecido que el enfoque Japonés es el adecuado para ser implementado en este desarrollo debido a sus fases del modelo de calidad las mismas que son:

**Enfoque Japonés**

Su proceso de planear, hacer, verificar y actuar.

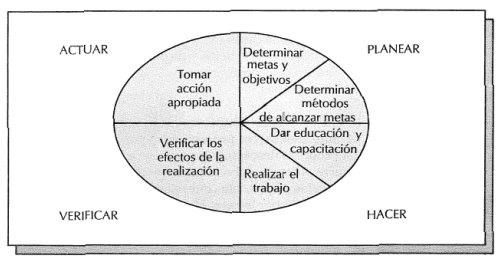
Planear a su vez es determinar objetivos y métodos.

Hacer es realizar el trabajo, previa educación y capacitación de los ejecutores.

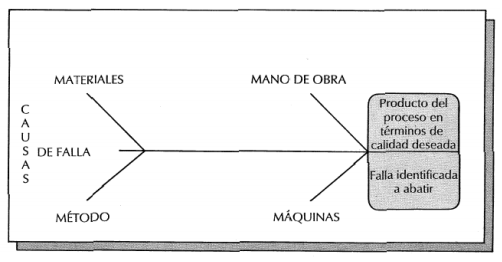
Verificar permanentemente si el producto satisface lo planeado, incluyendo la satisfacción del cliente.

Actuar implica tomar la acción correctiva necesaria.

Su metodología de análisis causal para la solución de problemas, comúnmente conocida como "Espina de Pescado" o diagrama de causa-efecto. Se llama espina de pescado porque dice que hay que ir a la raíz de los problemas



*Grafico 2.- Modelo Japonés*



*Grafico N°3 – Modelo de espina de pescado*

* 1. METOGOLOGIA DE DESARROLLO AGIL SCRUM

Para el desarrollo del presente proyecto se ha hecho uso de metodologías de agiles de desarrollo como es el caso de SCRUM el cual es definido como una metodología agil de desarrollo de software creado en el año 1986 por Takeuchi y Nonaka mediante la publicaron del artículo “The New Product Developroent Game”

el cual daría a conocer una nueva forma de gestionar proyectos en la que la agilidad, flexibilidad, y la incertidumbre son los elementos principales.

Nonaka y Takeuchi se fijaron en empresas tecnológicas que, estando en el mismo entorno en el que se encontraban otras empresas, realizaban productos en menos tiempo, de buena calidad y menos costes.

Observando a empresas como Honda, HP, Canon…etc., se dieron cuenta de que el producto no seguía unas fases en las que había un equipo especializado en cada una de ellas, si no que se partía de unos requisitos muy generales y el producto lo realizaba un equipo multidisciplinar que trabajaba desde el comienzo del proyecto hasta el final. Se comparó esta forma de trabajo en equipo, con la colaboración que hacen los jugadores de Rugby y la utilización de una formación denominada SCRUM

Se puede concluir que Scrum es adecuado para aquellas empresas en las que el desarrollo de los productos se realiza en entornos que se caracterizan por tener:

1. Incertidumbre: Sobre esta variable se plantea el objetivo que se quiere alcanzar sin proporcionar un plan detallado del producto.

Esto genera un reto y da una autonomía que sirve para generar una “tensión” adecuada para la motivación de los equipos.

1. Auto-organización: Los equipos son capaces de organizarse por sí solos, no necesitan roles para la gestión.

Scrum al ser una metodología de desarrollo ágil tiene como base la idea de creación de ciclos breves para el desarrollo, que comúnmente se llaman iteraciones y que en Scrum se llamarán “Sprints”. Para entender el ciclo de desarrollo de Scrum es necesario conocer las 5 fases que definen el ciclo de desarrollo ágil:

1. Concepto: Se define de forma general las características del producto y se asigna el equipo que se encargará de su desarrollo.

2. Especulación: en esta fase se hacen disposiciones con la información obtenida y se establecen los límites que marcarán el desarrollo del producto, tales como costes y agendas.

Se construirá el producto a partir de las ideas principales y se comprueban las partes realizadas y su impacto en el entorno.

Esta fase se repite en cada iteración y consiste, en rasgos generales, en:

* Desarrollar y revisar los requisitos generales.
* Mantener la lista de las funcionalidades que se esperan.
* Plan de entrega. Se establecen las fechas de las versiones, hitos e iteraciones. Medirá el esfuerzo realizado en el proyecto.

3. Exploración: Se incrementa el producto en el que se añaden las funcionalidades de la fase de especulación.

4. Revisión: El equipo revisa todo lo que se ha construido y se contrasta con el objetivo deseado.

5. Cierre: Se entregará en la fecha acordada una versión del producto deseado. Al tratarse de una versión, el cierre no indica que se ha finalizado el proyecto, sino que seguirá habiendo cambios, denominados “mantenimiento”, que hará que el producto final se acerque al producto final deseado.



Gráfico Nº 4 – Ciclo de desarrollo ágil SCRUM

Scrum gestiona estas iteraciones a través de reuniones diarias, uno de los elementos fundamentales de esta metodología.

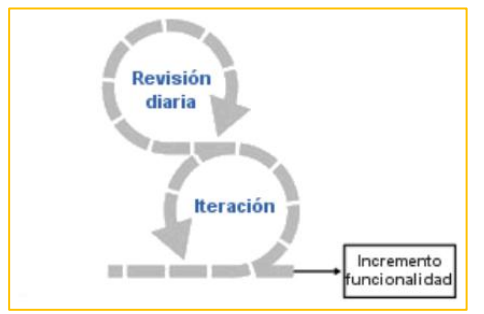


Gráfico Nº 5 – Ciclo principal de SCRUM

**Componentes de Scrum.**

Para entender todo el proceso de desarrollo del Scrum, se describirá de forma general las fases y los roles. Estas fases y roles se detallarán de forma más concisa más adelante.

Scrum se puede dividir de forma general en 3 fases, que podemos entender como reuniones. Las reuniones forman parte de los artefactos de esta metodología junto con los roles y los elementos que lo forman.

**Las Reuniones.**

1. Planificación del Backlog

Se definirá un documento en el que se reflejarán los requisitos del sistema por prioridades. En esta fase se definirá también la planificación del Sprint 0, en la que se decidirá cuáles van a ser los objetivos y el trabajo que hay que realizar para esa iteración. Se obtendrá además en esta reunión un Sprint Backlog, que es la lista de tareas y que es el objetivo más importante del Sprint.

2. Seguimiento del Sprint

En esta fase se hacen reuniones diarias en las que las 3 preguntas principales para evaluar el avance de las tareas serán:

* ¿Qué trabajo se realizó desde la reunión anterior?
* ¿Qué trabajo se hará hasta una nueva reunión?
* Inconvenientes que han surgido y qué hay que solucionar para poder continuar.

3. Revisión del Sprint

Cuando se finaliza el Sprint se realizará una revisión del incremento que se ha generado. Se presentarán los resultados finales y una demo o versión, esto ayudará a mejorar el feedback con el cliente.

4 Roles

Son las personas que están comprometidas con el proyecto y el proceso de Scrum.

* Product Owner: Es la persona que toma las decisiones, y es la que realmente conoce el negocio del cliente y su visión del producto. Se encarga de escribir las ideas del cliente, las ordena por prioridad y las coloca en el Product Backlog.
* ScrumMaster: Es el encargado de comprobar que el modelo y la metodología funciona. Eliminará todos los inconvenientes que hagan que el proceso no fluya e interactuará con el cliente y con los gestores.
* Equipo De Desarrollo: suele ser un equipo pequeño de unas 5-9 personas y tienen autoridad para organizar y tomar decisiones para conseguir su objetivo. Está involucrado en la estimación del esfuerzo de las tareas del Backlog.

Aunque no son parte del proceso de Scrum, es necesario que parte de la retroalimentación dé la salida del proceso y así poder revisar y planear cada sprint y son:

* Usuarios: Es el destinatario final del producto.
* Stakeholders: Las personas a las que el proyecto les producirá un beneficio. Participan durante las revisiones del Sprint.
* Managers: Toma las decisiones finales participando en la selección de los objetivos y de los requisitos.

**Elementos de Scrum**.

Los elementos que forman a Scrum son:

* Product Backlog: lista de necesidades del cliente.
* Sprint Backlog: lista de tareas que se realizan en un Sprint.
* Incremento: parte añadida o desarrollada en un Sprint, es un parte terminada y totalmente operativa

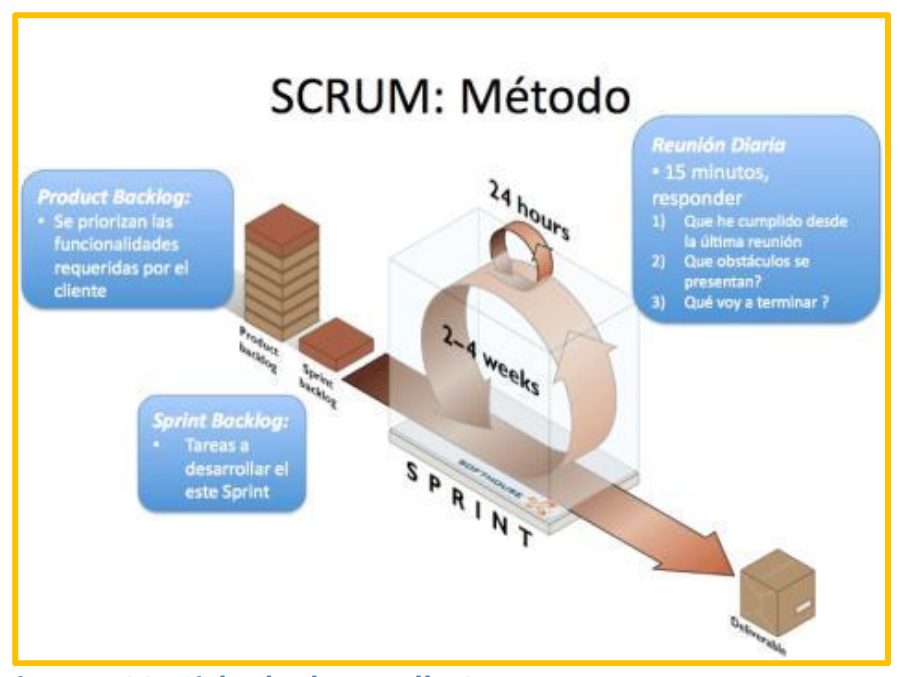


Grafico Nº 7 Ciclo de desarrollo SCRUM

**Product Backlog.**

Es el inventario en el que se almacenan todas las funcionalidades o requisitos en forma de lista priorizada, en el presente proyecto este instrumento se ha detallado de manera minuciosa y se encuentra en un formulario anexo. Estos requisitos son los que se adquirieron en sucesivas iteraciones.

La lista será gestionada y creada por el cliente con la ayuda del Scrum Master, quien indicará el coste estimado para completar un requisito, y además contendrá todo lo que aporte un valor final al producto.

Las tres características principales de esta lista de objetivos serán:

* Contendrá los objetivos del producto, se suele usar para expresarlos las historias de usuario.
* En cada objetivo, se indicará el valor que le da el cliente y el coste estimado; de esta manera, se realiza la lista, priorizando por valor y coste, se basará en el ROI.
* En la lista se tendrán que indicar las posibles iteraciones y los releases que se han indicado al cliente.
* La lista ha de incluir los posibles riesgos e incluir las tareas necesarias para solventarlos.

**Las historias de Usuario.**

Son las descripciones de las funcionalidades que va a tener el software.

Estas historias de usuario, serán el resultado de la colaboración entre el cliente y el equipo, e irán evolucionando durante toda la vida del proyecto.

Las historias de usuario se componen de tres fases denominadas “Las 3 C”:

* Card: Será una breve descripción escrita que servirá como recordatorio.
* Conversation: Es una conversación que servirá para asegurarse de que se ha entendido bien todo, y concretar el objetivo.
* Confirmation: Tests funcionales para fijar detalles que sean relevantes e indicar cuál va a ser el límite.

**Formato de la Pila Del Producto (Product Backlog).**

En Scrum, la preferencia por tener documentación en todo momento es menos estricta. Se encuentra más necesario el mantener una comunicación directa con el equipo, por eso se usa como herramienta el Backlog.

Aunque no hay ningún producto especial a la hora de confeccionar la lista, es conveniente que incluya información relativa a:

* Identificador para la funcionalidad.
* Descripción de la funcionalidad.
* Sistema de priorización u orden.
* Estimación.

**Sprint Backlog.**

Es la lista de tareas que elabora el equipo durante la planificación de un Sprint.

Se asignan las tareas a cada persona y el tiempo que queda para terminarlas.

De esta manera el proyecto se descompone en unidades más pequeñas y se puede determinar o ver en qué tareas no se está avanzando e intentar eliminar el problema.

**Incremento.**

Representa los requisitos que se han completado en una iteración y que son perfectamente operativos.

Según los resultados que se obtengan, el cliente puede ir haciendo los cambios necesarios y replanteando el proyecto.

* 1. Diagramación

Para el desarrollo de este software los autores han hecho uso de diversas metodologías y herramientas para diagramar los flujos y procesos del software a desarrollarse, es el caso de diagramas se diferente objeto y funcionamiento para representar en forma gráfica los procesos de una empresa y observar las actividades en conjunto, sus relaciones y cualquier incompatibilidad, cuello de botella o fuente de posibles ineficiencias.

Debido a la investigación se ha definido en base a diferentes investigaciones que:

el análisis de procesos es el cambio más importante ocurrido en la evolución de la ciencia administrativa. (Hernández & Alfredo, 2003)

Debido a ello para el desarrollo de un diagrama se debe cuidar el detalle del uniforme, cada uno de los cuadros de actividad puede ser considerado como un proceso; los diagramas de dichos cuadros constituyen el siguiente nivel de detalle. De esta forma se puede ir penetrando en el detalle hasta donde resulte conveniente.

En la actualidad la tendencia de desarrollo se caracteriza por la creciente dedicación a la búsqueda de soluciones completas donde se aborden todos los ángulos de una situación, con este objetivo se desarrollaron diferentes tipos de simbología que representan procesos, preguntas, bucles, etc

* 1. Diagrama de flujo y participantes

Una de las características de este tipo de diagrama es que considera a los diversos participantes en el proceso y también describe el flujo de las actividades y la secuencias, es el caso de los diagramas que

Normalmente, contiene más detalles que los dos diagramas anteriores, puesto que, en una actividad, puede participar más de un área y ello da pie a desglosar cada una de dichas participaciones.

En la presente investigación los autores han hecho uso de este tipo de diagramas, para lo cual se ha realizado un diagrama de flujo y participantes del proceso de selección y presentación del presente desarrollo al cliente final, en el mencionado desarrollo se realizaron entrevistas con el personal designado para el manejo del software asi como con el gerente de la empresa para asegurar que los requerimientos que se están levantando cumplan las necesidades del cliente.

* 1. Técnica del diagrama causa-efecto:

Descrita por Ishikawa en 1943 para explicar a un grupo de ingenieros de Kawasaki la clasificación y relación entre diversos factores. Conocida también como diagrama

de espina de pescado, diagrama de Ishikawa o diagrama de las 4M: maquina, material, mano de obra y método, forma clásica de agrupar las causas. Aunque ha sido una herramienta de análisis de gran aceptación, actualmente se utilizan más a menudo dos variantes, las 5M, añadiendo el medio y el método PEM-PEM (personal, entorno y método-planta, equipos y materiales).

Su objetivo final es desglosar problemas en componentes elementales para permitir una mayor facilidad de manejo e interpretación de sus causas. El efecto es el resultado obtenido y las causa el factor que contribuye al efecto obtenido. Cualquier efecto puede tener varias causas. Este método fomenta la reflexión, relaciona exhaustivamente y de forma jerárquica las causas que concurren en un problema, investiga estas causas siguiendo el orden jerárquico establecido y verifica los resultados.

Para la construcción de este diagrama, es necesario la realización de una serie de acciones:

•Selección de características a mejorar o el problema a resolver, que constituirá el denominado efecto.

•Dibujo de una flecha horizontal que termine en un rectángulo que contenga el efecto.

•Confección de una lista con las causas posibles mediante técnicas de generación de ideas como brainstorming, NGT, etc. Ordenación, estratificación y clasificación de estas posibles causas en base a criterios de las 5M.

•Realización de unas ramas principales incidentes en la línea horizontal. En los extremos de cada una se escriben las causas principales dentro de otros tantos rectángulos (medio, material, maquina, método y mano de obra).

•Determinación de causas secundarios o menores, también denominadas concretas, que dan lugar las causas principales. Se agrupan alrededor de la causa principal. Se buscan el máximo número de causas, sin descartar ninguna hasta la realización de su análisis.

•Análisis exhaustivo de todas las causas. Eliminación, mediante pruebas de aquellas que no afectan al problema. Para la realización de este análisis se puede utilizar el método QQDCCP o método de las seis preguntas, descrito en los siguientes apartados del trabajo. De este análisis se obtienen las causas más probables, procediendo posteriormente a la toma de acciones que se consideren oportunas.

Para aumentar la creatividad, conviene disponer de un grupo heterogéneo, con perspectivas diferentes. Se debe desarrollar de forma sistemática y concisa, no teniendo más de 6 o 7 causas principales. Debido a su naturaleza gráfica, se debe facilitar su lectura, teniendo la precaución de subdividir el problema en sub-problemas con gráficos independientes, en los casos necesarios. (Jesús Bustínduy Basterrechea, s/f)

En el desarrollo del presente proyecto se tomaron en cuenta diversas metodologías de desarrollo, todo ello con el objetivo de asegurar la calidad del desarrollo del software en tal sentido se utilizaron diferentes conceptos y se definió el ciclo de vida del software, el mismo que se detalla de la siguiente manera:

* 1. Arquitectura del Software

El diseño de la arquitectura del software se refiere a la estructura global del software y las maneras en que esa estructura proporciona integridad conceptual a un sistema. De acuerdo con Pressman, en su forma más simple, la arquitectura es la estructura jerárquica de los módulos del programa, la manera de interactuar de estos componentes, y la estructura de los dato s usados por estos módulos (Méndez, s/f, p. 5)

La arquitectura del software que este proyecto propone como una herramienta aplicativa dentro de un contexto real y centralizado; es en tal sentido que los autores realizaron investigaciones de ventajas y desventajas de arquitecturas, de donde se pudo verificar que la opción más eficiente y recomendable para el desarrollo de este software es la arquitectura de escritorio

Descripción de los Requisitos funcionales del Proyecto “Sistema intérprete de datos Suministrados por reloj Biométrico”

* 1. Verificación y validación

Con el objetivo de verificar lo peticionado por el cliente Dentales Pablo Herman S.A. los autores han establecido diferentes técnicas de verificación y validación, el cual es el nombre que se da a los procesos de comprobación y análisis que aseguran que el software que se esta desarrollando está acorde a su especificación y cumple las necesidades de Dentales Pablo Herman S.A.. La Verificación y Validación es un proceso de ciclo de vida completo. Inicia con las revisiones de los requerimientos y continúa con las revisiones del diseño y las inspecciones del código hasta la prueba del producto. Existen actividades de V&V en cada etapa del proceso de desarrollo del software.

La verificación y la validación no son la misma cosa , aunque es muy fácil confundirlas, Boehm (1979) expresó la diferencia entre ellas de forma sucinta:

*•Verificación: ¿Estamos construyendo el producto correctamente?*

*El papel de la verificación comprende comprobar que el software está de*

*acuerdo con su especificación. Se comprueba que el sistema cumple los*

*requerimientos funcionales y no funcionales que se le han especificado.*

*•Validación: ¿Estamos construyendo el producto concreto?*

*La validación es un proceso mas general. Se debe asegurar que el software*

*cumple las expectativas del cliente. Va mas allá de comprobar si el*

*sistema está acorde con su especificación, para probar que el software*

*hace lo que el usuario espera a diferencia de lo que se ha especificado.*

Es fácil cometer errores y omisiones durante la fase de análisis de requerimientos del sistema y, en tales casos, el software final no cumplirá la expectativas de los clientes. Sin embargo, en la realidad, la validación de los requerimientos no puede descubrir todos los problemas que presenta la aplicación. Algunos defectos en los requerimientos solo pueden descubrirse cuando la implementación del sistema es completa.

* 1. Prototipado

Como una técnica de verificación y validación se han hecho uso de técnicas de como el prototipoado el cual genera un modelo (representación, demostración o simulación) fácilmente ampliable y modificable de un sistema planificado, probablemente incluyendo su interfaz y su funcionalidad de entradas y salidas.

El prototipado modela el producto final y permite efectuar un test sobre determinados atributos del mismo sin necesidad de que está disponible. Se trata, simplemente, de testar haciendo uso del modelo.

Se puede hablar de una amplia variedad de técnicas de prototipado en base a las

características del prototipo a desarrollar:

1. Según la funcionalidad reproducida

2. Según la fidelidad de la reproducción de la interfaz

3. Según su propósito

4. Según la porción del sistema representado

5. Según su ejecutabilidad u operatividad

**. Según la funcionalidad reproducida**

Prototipado Horizontal (Horizontal Prototyping)

Los prototipos horizontales exhiben un amplio espectro de las características del producto, pero sin el respaldo de una funcionalidad relativamente amplia. Los prototipos horizontales se utilizan con frecuencia para evaluar las preferencias de los usuarios respecto de las interfaces de usuario, cuando las funciones reales operativas aún no han sido implementadas. Tales prototipos permiten una evaluación del diseño de la interfaz, así como la ubicación y accesibilidad de determinados aspectos y características, sin requerir el funcionamiento real de las funciones que representan.

A menudo, los prototipos horizontales se confeccionan como de baja fidelidad, consistiendo en pocos más que dibujos y listas manuscritas, pasando por maquetas realizadas por ordenador que tienen el mismo aspecto que el producto final. Las maquetas realistas pueden construirse con la mayoría de los entornos de desarrollo gráfico. Esta técnica se utiliza con preferencia en las etapas tempranas del proceso de desarrollo, cuando el trabajo sobre las funciones reales del producto aún no ha dado comienzo, pero el conjunto de características es conocido.

Prototipado Vertical (Vertical Prototyping)

Los prototipos verticales muestran la funcionalidad exacta de un producto para una pequeña parte del conjunto completo. Por ejemplo, un prototipo vertical de un procesador de textos podría mostrar todas las funciones de comprobación de ortografía y gramática, pero ninguna función relacionada con la entrada de texto o su formato. Todas las funciones de un prototipo vertical imitan sus equivalente reales tanto como sea posible. Dado que un prototipo vertical ha de ser funcional prácticamente de forma completa (aunque sólo para una pequeña parte de la interfaz del producto), quizás la mejor forma de obtener un prototipo vertical es utilizar un módulo completamente operativo de un producto. Para aplicaciones software que se han desarrollado con arquitectura modular, esto resulta relativamente sencillo, aunque las interfaces a otros módulos no funcionarán (lo que no supone ningún problema porque es la funcionalidad de las secciones dadas las que serán inspeccionadas o testadas, y no otras). En el caso de un coche, sería el asiento y el tablero de mandos lo que podría ser testado, mientras que el chasis, la carrocería y otros elementos aún no están disponibles.

Esta técnica se utiliza cuando el diseño para una parte del producto en particular está prácticamente completa y merece la pena evaluarla en tanto que es un elemento contiguo a otro. Aun cuando algunas partes del producto aún no están listas para el test, es posible determinar ciertos problemas con alguna parte en concreto mientras las demás se encuentran aún en fase de desarrollo.

**Prototipado Diagonal**

Se trata de un prototipo horizontal hasta un cierto nivel, a partir del cual se puede

considerar vertical

**2. Según la fidelidad de la reproducción de la interfaz**

Prototipado de Alta Fidelidad (High-Fidelity Prototyping)

El prototipado de alta fidelidad es un método donde el prototipo utilizado para el test se corresponde con la interfaz real en la mayor medida posible. Normalmente, y en particular para interfaces de software, es otra herramienta de software la utilizada para maquetar la interfaz. Dicha herramienta acepta entradas desde ratón o teclado, tal y como haría la interfaz real y responde a esos eventos de idéntica forma (mostrando una ventana en particular, un mensaje, cambiando de estado, etc).

Se ha de utilizar una herramienta que permita imitar el aspecto y el comportamiento de la interfaz real en la mayor medida de los posible. Muchos prototipos de software se construyen utilizando herramientas multimedia. Dichas herramientas, además, permiten imitar la existencia de retrasos en máquinas más lentas o la espera a la respuesta de un servidor.

Las interfaces basadas en la Web se prototipan bastante bien con HTML y HTML Dinámico, siendo posible reutilizar algunos fragmentos de código del prototipo en la interfaz real.

**Prototipado de Baja Fidelidad (Low-Fidelity Prototyping)**

El prototipado de baja fidelidad es una manera barata de proporcionar prototipos para su uso en test y sesiones de diseño participativo. Baja fidelidad significa que los prototipos a utilizar no tienen el aspecto real de la interfaz que se está testando, aun cuando operan de la misma forma.

La idea es conseguir una gran cantidad de información de la interacción entre la interfaz y el usuario mediante la evaluación de este prototipo. Dado que los prototipos de baja fidelidad son baratos, tanto en términos de dinero como de tiempo, es posible permitirse un mayor número de ciclos de test, más sujetos o más prototipos.

**Prototipado de Papel (Paper Prototyping)**

Este método se caracteriza por el uso de materiales y equipo sencillos para crear una simulación basada en papel de la interfaz de un sistema con el objetivo de explorar los requerimiento de usuario (Después, durante el proceso de diseño, los prototipos de papel van a constituirse como medios valiosos y rentables para evaluar las opciones de diseño. Los elementos de la interfaz, como menús, ventanas, diálogos e iconos, se crean utilizando papel, tarjetas, acetato, bolígrafos,..). El resultado obtenido se denomina frecuentemente prototipo de baja fidelidad

* 1. Control de Versiones - GIT

Para el desarrollo del presente proyecto se ha hecho uso de herramientas de control de versiones como GIT, Esta herramienta permite alojar los diferentes proyectos de trabajo, permitiendo gestionar de forma muy simple y gráfica las diferentes versiones o modificaciones que se vayan realizando. Git es un sistema de control de versiones distribuido, mediante el cual podemos trabajar a nivel local sin tener que comprometer el repositorio principal a cada cambio que realicemos.

Gracias a este sistema, los desarrolladores del proyecto trabajaron de forma independiente hasta el momento en el que tengan que poner en común con el resto del equipo su código, controlando los cambios en las versiones y mejorando el seguimiento al desarrollo.

* 1. GESTIÓN DE VALIDACIÓN Y VERIFICACIÓN DEL SOFTWARE

El interés por la calidad crece de forma continua, a medida que los clientes se vuelven más selectivos y comienzan a rechazar los productos poco fiables o que realmente no dan respuesta a sus necesidades. Como primera aproximación es importante diferenciar entre la calidad del PRODUCTO software y la calidad del PROCESO de desarrollo. Las metas que se establezcan para la calidad del producto van a determinar las metas a establecer para la calidad del proceso de desarrollo, ya que la calidad del producto va a estar en función de la calidad del proceso de desarrollo. Sin un buen proceso de desarrollo es casi imposible obtener un buen producto. También es importante destacar que la calidad de un producto software debe ser considerada en todos sus estados de evolución a medida que avanza el desarrollo de acuerdo al ciclo de vida seleccionado para su construcción (especificaciones, diseño, código, etc.). No basta con tener en cuenta la calidad del producto una vez finalizado, cuando los problemas de mala calidad ya no tienen solución o la solución es muy costosa.

* 1. TÉCNICAS DE PRUEBA

Las técnicas de evaluación dinámica o prueba proporcionan distintos criterios para generar casos de prueba que provoquen fallos en los programas. Estas técnicas se agrupan en:

* Técnicas de caja blanca o estructurales, que se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa.
* Técnicas de caja negra o funcionales, que realizan pruebas sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar.
  1. ESTRATEGIA DE PRUEBAS

La estrategia que se ha de seguir a la hora de evaluar dinámicamente un sistema software debe permitir comenzar por los componentes más simples y más pequeños e ir avanzando progresivamente hasta probar todo el software en su conjunto. Más concretamente, los pasos a seguir son:

1. Pruebas Unitarias. Comienzan con la prueba de cada módulo.

2. Pruebas de Integración. A partir del esquema del diseño, los módulos probados se vuelven a probar combinados para probar sus interfaces.

3. Prueba del Sistema. El software ensamblado totalmente con cualquier componente hardware que requiere se prueba para comprobar que se cumplen los requisitos funcionales.

4. Pruebas de Aceptación. El cliente comprueba que el software funciona según sus expectativas.

* 1. IDE

IDE en su sigla en inglés (Environment Development Integrated) como su nombre lo indica es un “Entorno de Desarrollo Integrado”. Debido que contiene un editor de texto, un editor de diseño, compilador y un depurador, el primer IDE fue creado para BASIC Dartmouth en ese entonces se utilizaba en la terminal o la consola de programación no tiene en su fila un IDE para realizar sus respectivas tareas como el diseño y desarrollo, tiende hacer muy desgastante para las personas porque estará desintegrado y necesitaría ser configurado, por lo tanto no ayudaría al desarrollador o programador cumplir sus respectivas funciones. Así se puede afirmar que para utilizar un Lenguaje de programación es necesario tener un IDE completo en una empresa para realizar un buen desarrollo de software

* 1. NETBEANS.-

Netbeans es una Herramienta que se utiliza para desarrollar aplicaciones Web, Móvil y de Escritorio para diferentes lenguajes de programación como son Java, C++, Ruby y PHP entre otros. Es de código abierto, es multiplataforma, multilenguaje, contiene servidores web y es fácil de instalarlo e utilizarlo,

* 1. MY SQL.-

Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca. MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL.

El lenguaje de programación que utiliza MySQL es Structured Query Language

( SQL ) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales.

1. Marco Legal

Para el desarrollo de esta investigación los autores han realizado una exhaustiva búsqueda de normativas que se relacionen con el control, desarrollo y manejo que pueda verse relacionado con el software propuesto, en tal sentido se han encontrado relación directa con los diferentes actores y usuarios del software (cliente-empleador y usuario-empleado), en tal sentido la normativa legal del Ecuador ampara al empleado, brinda derechos, beneficios y responsabilidades que se resaltan de la siguiente manera.

Constitución de la República del Ecuador

Los derechos de un trabajador son irrenunciables y todos los ciudadanos Ecuatorianos gozamos del derecho de trabajar como lo señala la constitución en su artículo 33

El trabajo es un derecho y un deber social, y un derecho económico, fuente de realización personal y base de la economía. El Estado garantizará a las personas trabajadoras el pleno respeto a su dignidad, una vida decorosa, remuneraciones y retribuciones justas y el desempeño de un trabajo saludable y libremente escogido o aceptado

(Constitución, 2008)

En tal sentido haciendo referencia al texto constitucional los autores se centran en los derechos irrenunciables del trabajador según el articulo 66 literal 17 del mismo cuerpo constitucional

El derecho a la libertad de trabajo. Nadie será obligado a realizar un trabajo gratuito o forzoso, salvo los casos que determine la ley. (Constitución, 2008)

El Codigo de Trabajo señala los parámetros sobre los cuales el empleado debe desarrollar su actividad laboral, hablando en relación al horario de trabajo se establece en el articulo 47.

De la jornada máxima.­ La jornada máxima de trabajo será de ocho horas diarias, de manera que no exceda de cuarenta horas semanales, salvo disposición de la ley en contrario.

El tiempo máximo de trabajo efectivo en el subsuelo será de seis horas diarias y solamente por concepto de horas suplementarias, extraordinarias o de recuperación, podrá prolongarse por una hora más, con la remuneración y los recargos correspondientes (Codigo de Trabajo, 2005)

**Art. 48.-** Jornada especial.- Las comisiones sectoriales y las comisiones de trabajo determinarán las industrias en que no sea permitido el trabajo durante la jornada completa, y fijarán el número de horas de labor

**Art. 50.-** Límite de jornada y descanso forzosos

Las jornadas de trabajo obligatorio no pueden exceder de cinco en la semana, o sea de cuarenta horas hebdomadarias.

Los días sábados y domingos serán de descanso forzoso y, si en razón de las circunstancias, no pudiere interrumpirse el trabajo en tales días, se designará otro tiempo igual de la semana para el descanso, mediante acuerdo entre empleador y trabajadores.

**Art. 51.-** Duración del descanso.­ El descanso de que trata el artículo anterior lo gozarán a la vez todos los trabajadores, o por turnos si así lo exigiere la índole de las labores que realicen.

Comprenderá un mínimo de cuarenta y ocho horas consecutivas.

**Art. 52.-** Trabajo en sábados y domingos.­

Las circunstancias por las que, accidental o permanentemente, se autorice el trabajo en los días sábados y domingos, no podrán ser otras que éstas:

1. Necesidad de evitar un grave daño al establecimiento o explotación amenazado por la inminencia de un accidente; y, en general, por caso fortuito o fuerza mayor que demande atención impostergable. Cuando esto ocurra no es necesario que preceda autorización del inspector del trabajo, pero el empleador quedará obligado a comunicárselo dentro de las veinticuatro horas siguientes al peligro o accidente, bajo multa que será impuesta de conformidad con lo previsto en el artículo 628 de este Código, que impondrá el inspector del trabajo.

En estos casos, el trabajo deberá limitarse al tiempo estrictamente necesario para atender al daño o peligro.

2. La condición manifiesta de que la industria, explotación o labor no pueda interrumpirse por la naturaleza de las necesidades que satisfacen, por razones de carácter técnico o porque su interrupción irrogue perjuicios al interés público. (Código de Trabajo, 2005)

Uno de los objetivos principales del desarrollo de este software es el control de las horas laborables del empleado, brindando así una herramienta que mantenga el control sobre remuneraciones suplementarias o extraordinarias, de igual manera el número de horas trabajadas por el personal diariamente, en tal sentido en el Código de Trabajo los autores señalan la siguiente normativa legal.

**Art. 55.-** Remuneración por horas suplementarias y extraordinarias. ­

Por convenio escrito entre las partes, la jornada de trabajo podrá exceder del límite fijado en los artículos 47 y 49 de este Código, siempre que se proceda con autorización del inspector de trabajo y se observen las siguientes prescripciones:

1. Las horas suplementarias no podrán exceder de cuatro en un día, ni de doce en la semana;

2. Si tuvieren lugar durante el día o hasta las 24H00, el empleador pagará la remuneración correspondiente a cada una de las horas suplementarias con más un cincuenta por ciento de recargo. Si dichas horas estuvieren comprendidas entre las 24H00 y las 06H00, el trabajador tendrá derecho a un ciento por ciento de recargo. Para calcularlo se tomará como base la remuneración que corresponda a la hora de trabajo diurno;

3. En el trabajo a destajo se tomarán en cuenta para el recargo de la remuneración las unidades de obra ejecutadas durante las horas excedentes de las ocho obligatorias; en tal caso, se aumentará la remuneración correspondiente a cada unidad en un cincuenta por ciento o en un ciento por ciento, respectivamente, de acuerdo con la regla anterior. Para calcular este recargo, se tomará como base el valor de la unidad de la obra realizada durante el trabajo diurno; y,

4. El trabajo que se ejecutare el sábado o el domingo deberá ser pagado con el ciento por ciento de recargo

De igual manera La Ley Orgánica Para La Justicia Laboral Y Reconocimiento Del Trabajo En El Hogar proteje al trabajador bajo los siguientes parámetros establecidos en el articulo 59 numeral 5

Cálculo de horas suplementarias o de tiempo extraordinario, sin considerar la semana integral por debajo de las 240 horas al mes. Dicho trabajo suplementario o extraordinario deberá calcularse sobre 240 horas mensuales. (Ley Orgánica Para La Justicia Laboral Y Reconocimiento Del Trabajo En El Hogar, 2015)

Como se puede apreciar la normativa ecuatoriana brinda derechos y responsabilidades de los trabajadores para cumplir con su derecho constitucional de trabajo, sin embargo en el mismo sentido se brindan derechos al empleador para que pueda desarrollar sus actividades económicas de manera ágil y eficiente, es en tal sentido que se crearon diferentes preventas a favor del empleador en referencia al tiempo y horario de trabajo, dichos beneficios se encuentran establecidos en el Código De Trabajo bajo los siguientes numerales:

**Art. 54.-** Pérdida de la remuneración.­ El trabajador que faltare injustificadamente a media jornada continua de trabajo en el curso de la semana, tendrá derecho a la remuneración de seis días, y el trabajador que faltare injustificadamente a una jornada completa de trabajo en la semana, sólo tendrá derecho a la remuneración de cinco jornadas.

Tanto en el primer caso como en el segundo, el trabajador no perderá la remuneración si la falta estuvo autorizada por el empleador o por la ley, o si se debiere a enfermedad, calamidad doméstica o fuerza mayor debidamente comprobadas, y no excediere de los máximos permitidos.

La jornada completa de falta puede integrarse con medias jornadas en días distintos.

No podrá el empleador imponer indemnización al trabajador por concepto de faltas.

**Art. 172.-** Causas por las que el empleador puede dar por terminado el contrato.- El empleador podrá dar por terminado el contrato de trabajo, previo visto bueno, en los siguientes casos:

1. Por faltas repetidas e injustificadas de puntualidad o de asistencia al trabajo o por abandono de éste por un tiempo mayor de tres días consecutivos, sin causa justa y siempre que dichas causales se hayan producido dentro de un período mensual de labor;

2. Por indisciplina o desobediencia graves a los reglamentos internos legalmente aprobados;

3. Por falta de probidad o por conducta inmoral del trabajador;

4. Por injurias graves irrogadas al empleador, su cónyuge o conviviente en unión de hecho, ascendientes o descendientes, o a su representante;

5. Por ineptitud manifiesta del trabajador, respecto de la ocupación o labor para la cual se comprometió;

6. Por denuncia injustificada contra el empleador respecto de sus obligaciones en el Seguro Social; más, si fuere justificada la denuncia, quedará asegurada la estabilidad del trabajador, por dos años, en trabajos permanentes; y,

7. Por no acatar las medidas de seguridad, prevención e higiene exigidas por la ley, por sus reglamentos o por la autoridad competente; o por contrariar, sin debida justificación, las prescripciones y dictámenes médicos.

Como podemos apreciar, el control por parte del empleador en el aspecto de horarios de trabajo es legalmente justificado y valedero, es en ese sentido que el presente desarrollo de software brinda esa facilidad para el control de horarios laborales de los empleados, cabe destacar que al momento este procedimiento se realiza de manera manual y genérica, lo cual se presta a inconvenientes con el personal asignado, así como con el mecanismo actualmente usado.

Es en tal sentido que el presente proyecto se encuentra justificado legalmente bajo normativa vigente en la legislación de Ecuador.

1. Marco referencial

Sistemas similares que existen en el mercado, comparativos.

En este campo descriptivo, se muestra un cuadro comparativo en función del Proyecto *“Sistema Intérprete de Datos suministrados por un Reloj Biométrico”*, se eligió dos Sistema los cuales se enfocan en funcionalidades similares a la del Proyecto.

**Reloj Biométrico de Control de Personal, Biotech IN04.**

|  |  |  |
| --- | --- | --- |
| COMPARATIVA | Biotech IN04 | DSoftware |
| Software de terceros | Software propio |
| NOTIFICACION DE NOVEDADES | Área de notificaciones, se despliega un campo que contiene todas las notificaciones con respecto al control de asistencia. | En cuanto a novedades del empleado el Sistema enviará un correo electrónico cuando se registre alguna de las siguientes acciones: atraso o falta al día de trabajo por parte de un trabajador. |
| MODIFICACION | Control de Sistema Automatizado, el Sistema se maneja de forma independiente con la facultad de ofrecer recursos limitados de Funcionalidad | El registro datos podrá configurarse para realizarse de forma automatizada a su registro, o manualmente en lotes. |
| El sistema también permitirá el registro de asistencias manuales, sin embargo, estas requerirán autorización por parte del Gerente. |
| EXCLUSIVIDAD | Evita el fraude que se puede dar por el intercambio de tarjetas o pines de acceso. Ningún empleado podrá “marcarle” a otro empleado. | El campo fecha de consulta acepta únicamente fechas anteriores al día de hoy (día actual). |
| Sistema de Encriptación Básica para acceso a su base de Datos de ingreso personal. | Los usuarios deben ingresar al sistema con un nombre de usuario y contraseña. |

*Fuente de Información del Sistema comparado:*

[*http://www.biotracksoftware.com/esp/asistencia.htm*](http://www.biotracksoftware.com/esp/asistencia.htm)

1. Marco ambiental

Como todo Proyecto implementado en un campo determinado, existe un impacto que puede ser tanto directo como indirecto en el Medio Ambiente, el mismo que se presenta sin excepciones, pero niveles diferentes.

Los puntos desarrollados a continuación, son perspectivas del impacto Ambiental que desde el punto de vista de los autores se tiene del Proyecto.

IMPACTO DE TIPO CONTAMINACIÓN AMBIENTAL:

**El “Lector Biométrico”,** es el componente del Sistema Biométrico físico que, en el campo de DESECHO, es el que más atención requiere.

Su lapso de Vida Útil, en el caso de ser un lector de “Biométrico Físico”, es de 5 años con mantenimiento moderado, según el “Departamento de Sistemas de la Universidad de CALIFORNIA”[[1]](#footnote-1). Siendo este elemento el componente que genera Contaminación de tipo Tecnológico

**Energía,** El incremento en el consumo de energía ha hecho que aumenten considerablemente las proporciones de determinados gases, un Sistema completo como en este caso: “Sistema Biométrico”, usa un enfoque considerable de energía eléctrica ya que es un Administrador de Base de Datos, compuesto por diferentes componentes de “Almacenamiento y Distribución de Información”.

**Efecto Invernadero,** todos los desechos que dejan los Sistemas Tecnológicos generan recalentamiento global del planeta. (Efecto de invernadero)

Las transformaciones traen consigo cambios en los regímenes de lluvia y en los ecosistemas marinos, aumentan las probabilidades de ciclones, vulnerabilidad de las zonas templadas a enfermedades de malaria, del dengue, fiebre amarilla y afectaciones en muchos cultivos de esa área, ejemplo el trigo.

(SYSTEM-GLOBAL, 2017)

IMPACTO DE TIPO NO CONTAMINANTE

**Papel**, en un análisis de directrices secundarias, se estableció este punto, fundamentándose en el hecho de que el Sistema Biométrico genera un ahorro de material físico (hojas), para el registro que se hacía tradicionalmente en el seguimiento del Control Asistencial de los empleados en una Empresa determinada.

**Impacto Ambiental,** en comparación a la Industria y la Producción, el Sistema Biométrico tiene un impacto de poca proporción su nivel de liberación de Gases de Desecho es mínima o nula.

1. Marco tecnológico
   1. Terminología

Es preciso destacar que de acuerdo a la filosofía de PISIP (y de todo proceso iterativo e incremental), todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso podríamos tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de común.

* + 1. Modelo de Casos de Uso del Negocio

Es un modelo de las funciones de negocio vistas desde la perspectiva de los actores externos (Agentes de registro, solicitantes finales, otros sistemas etc.). permite situar al sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito. Este modelo se representa con un Diagrama de Casos de Uso usando estereotipos específicos para este modelo.

* + 1. Modelo de Objetos del Negocio

Es un modelo que describe la realización de cada caso de uso del negocio, estableciendo los actores internos, la información que en términos generales manipulan y los flujos de trabajo (workflows) asociados al caso de uso del negocio. Para la representación de este modelo se utilizan Diagramas de Colaboración (para mostrar actores externos, internos y las entidades (información) que manipulan, un Diagrama de Clases para mostrar gráficamente las entidades del sistema y sus relaciones, y Diagramas de Actividad para mostrar los flujos de trabajo.

* + 1. Modelo de Casos de Uso

El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso.

* + 1. Especificaciones de Casos de Uso

Para los casos de uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, flujo de eventos, requisitos no-funcionales asociados. También, para casos de uso cuyo flujo de eventos sea complejo podrá adjuntarse una representación gráfica mediante un Diagrama de Actividad.

* + 1. Especificaciones Adicionales

En este documento se plasmaron todos los requisitos que no han sido incluidos como parte de los casos de uso y se refieren requisitos no-funcionales globales. Dichos requisitos incluyen: requisitos legales o normas, aplicación de estándares, requisitos de calidad del producto, tales como: confiabilidad, desempeño, etc., u otros requisitos de ambiente, tales como: sistema operativo, requisitos de compatibilidad, etc.

* + 1. Modelo de Análisis y Diseño

Para este modelo se estableció la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo al avance del proyecto.

* + 1. Modelo de Datos

Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Clases (donde se utiliza un profile UML para Modelado de Datos, para conseguir la representación de tablas, claves, etc.).

* + 1. Modelo de Implementación

Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema. (Este modelo es sólo una versión preliminar al final de la fase de Elaboración, posteriormente tiene bastante refinamiento).

* + 1. Casos de Prueba

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados. Estos casos de prueba son aplicados como pruebas de regresión en cada iteración. Cada caso de prueba llevará asociado un procedimiento de prueba con las instrucciones para realizar la prueba, y dependiendo del tipo de prueba dicho procedimiento podrá ser automatizable mediante un script de prueba.

* + 1. Plan de Iteración

Es un conjunto de actividades y tareas ordenadas temporalmente, con recursos asignados, dependencias entre ellas. Se realiza para cada iteración, y para todas las fases.

* + 1. Material de Apoyo al Usuario Final

Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo: Guías del Usuario, Guías de Operación, Guías de Mantenimiento y Sistema de Ayuda los cuales estarían disponibles en almacenamiento en la nube.

* + 1. Producto

Los ficheros del producto empaquetados y almacenadas en un CD con los mecanismos apropiados para facilitar su instalación. El producto, a partir de la primera iteración de la fase de Construcción es desarrollado en cascada

* + 1. UML

Unified Modeling Language. Lenguaje de programación gráfico para el modelado de proyectos software orientados a objetos.

Sección II Diagnostico

Siguiendo en el recorrido del enfoque de calidad antes descrito, conjuntamente con el modelo de desarrollo agil SRCUM que se está haciendo uso, los autores han desarrollado un proceso de análisis con el objeto de diagnosticar la problemática a resolver para posteriormente definir el proyecto, es en tal sentido que es necesario definir cuál es el problema principal, de tal manera que ¿Que es un problema? Es el lazo de unión entre una situación real y la situación que cabría esperar o desear. El problema está afectando o afectará a la calidad que percibe el cliente del producto o servicio, o bien eleva los costes de producción, lo cual acarrea un descenso en los beneficios, implicando una pérdida de competitividad.

La conceptualización del problema en el presente proyecto de desarrollo de software consta de una introducción a la naturaleza del problema para aquello se hizo uso esencialmente de dos técnicas para definir el diagnóstico del proyecto un diagrama de flujo y participantes y un diagrama de espina de pescado, de donde podemos obtener:

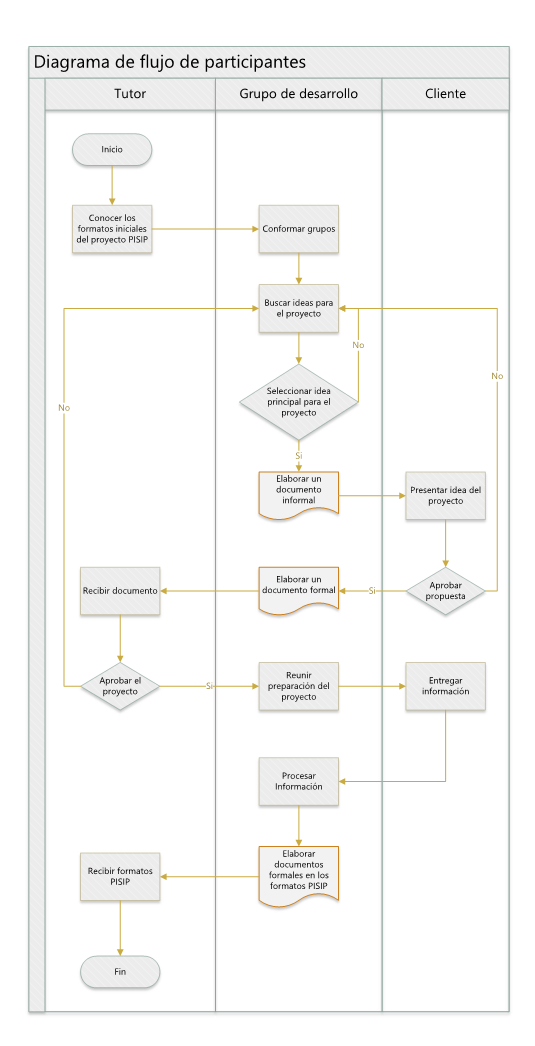
1. Diagrama de flujo y participantes

Con el objetivo de definir los parámetros sobre los cuales se iniciaría el desarrollo del presente software se realizó dos reuniones de trabajo en las instalación de la empresa usuaria del software con el objetivo de identificar a los actores que tendrían manejo y afectación directa con el sistema a implementarse, el análisis del diagrama de Flujo y Participantes se realizó tomando como referencia la interacción inicial que los autores tuvieron con el tutor de la materia Ingeniería de Software cuando se hicieron conocer los formatos sobre los cuales se trabajaría para el desarrollo del proyecto.

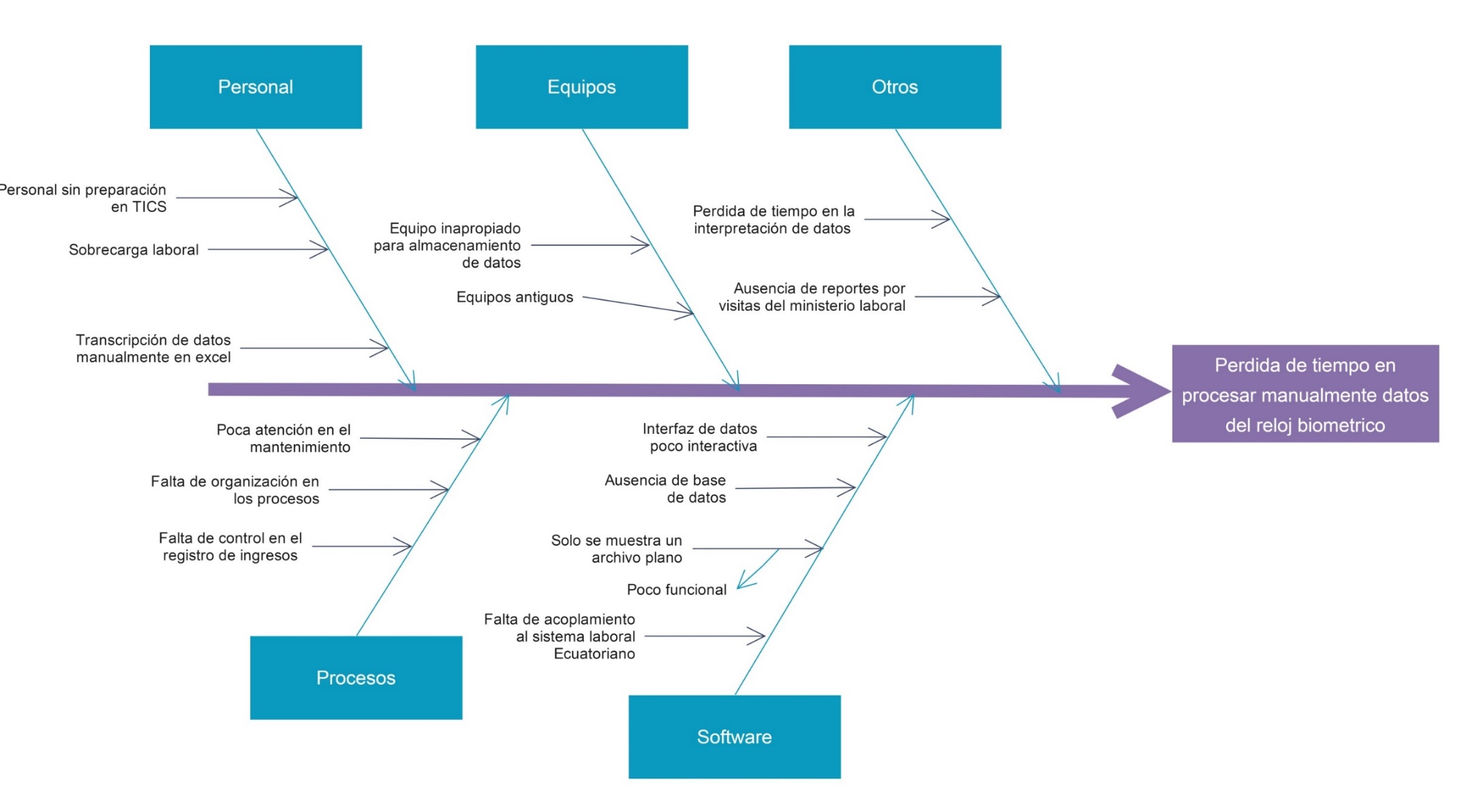
El presente grupo de trabajo se integró con alumnos de 7mo nivel de la catedra rectora Ing. de software II, materia en la cual se presentó ideas para el desarrollo del presente proyecto de donde se tomó como punto de referencia la problemática de la empresa beneficiaria del software. Una vez que se aprobó el tema del proyecto se elaboró el documento que se señala en el diagrama de espina de pescado

En primera instancia para el desarrollo de este proyecto se mantuvo reuniones con el gerente de la compañía con el objeto de obtener la aprobación para poder trabajar y desarrollar la solución a su problema. En tal sentido, posterior a las reuniones mantenidas se llegó a un consenso con la empresa usuaria del software.

Todo este procedimiento se puede verificar en el siguiente grafico de Flujo y Participantes en donde plasma los procedimientos que fueron realizados en la concepción y aceptación del proyecto de desarrollo de software.

*Grafico N° 4 diagrama de flujo de participantes aprobación de proyecto*

Con el objetivo principal de desglosar los problemas en componentes elementales para permitir una mayor facilidad de manejo e interpretación de sus causas. Se utilizó la metodología de la espina de pescado o diagrama de Ishikawa con el cual se ha establecido las causas y efectos de la problemática existente en la empresa, de tal manera que el resultado obtenido y las causas contribuyen directamente al efecto obtenido. Cualquier efecto puede tener varias causas. Este método fomenta la reflexión, relaciona exhaustivamente y de forma jerárquica las causas que concurren en un problema, investiga estas causas siguiendo el orden jerárquico establecido y verifica los resultados como se denota del siguiente gráfico:

****

*Grafico N°5 Grafico de espina de pescado*

De esta manera basados en el método investigación entrevistas (actas adjuntas) se pudo determinar la problemática exacta que es la siguiente:

Pérdida de tiempo en procesar manualmente datos de reloj biométrico.

Con esa información podemos determinar que el problema principal tiene afectación en diversos campos dentro de la empresa, en tal sentido es fundamental brindar solución a la problemática, en tal sentido se justifica el desarrollo del software propuesto que brindará las siguientes soluciones:

* Interpretará datos suministrados por un reloj biométrico y presentar reportes de manera estructurada y amigable al usuario.
* Simplificará el trabajo del usuario y realzar diversos tipos de consultas y reportes con los datos administrados
* Optimizará el proceso de trabajo del personal de control administrativo de la empresa
  1. Definición de requerimientos funcionales y no funcionales

Diagnóstico del Problema:

De acuerdo a la metodología de desarrollo ágil SCRUM se puede hacer uso de herramientas de diferentes metodologías de desarrollo, en tal sentido se ha hecho uso de información y resultados obtenidos mediante encuestas y entrevistas de donde se han obtenido los siguientes requerimientos funcionales y no funcionales, los mismos que serán utilizados para la elaboración del product Backlog y posterior Sprint Backlog:

**Requerimiento No Funcional**

* Optimización de Tiempo:

Rendimiento en la Ocupación del Tiempo, al implementar el Sistema Intérprete de Datos suministrados por Reloj Biométrico en el campo al que se apuntó.

* Optimización de tiempo a los Administradores, generar un reporte con la utilidad de una base de datos y un Sistema de Interpretación, genera un ahorro de tiempo de más de 5 veces menos tiempo, que realizar un reporte físico.
* Cumplimiento de normativa legal, sistemas de datos laborales: El nuevo sistema y sus procedimientos de mantenimiento de datos deben cumplir con las leyes y reglamentos del Ministerio de Relaciones Laborales.
* **Código abierto**, El nuevo sistema se acogerá a las reglas de las licencias generales públicas (GNU), es decir será gratuito, código abierto en el que cualquiera podrá cambiar el software, sin patentes y sin garantías.
* Desempeño, toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.
* Desempeño en transacciones con base de datos, los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.
* Nivel de Usuario, los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.
* Proceso de seguridad, el nuevo sistema debe desarrollarse aplicando patrones y recomendaciones de programación que incrementen la seguridad de datos.
* Seguridad en información, si se identifican ataques de seguridad o brecha del sistema, el mismo no continuará operando hasta ser desbloqueado por un administrador de seguridad.
* Errores del stakeholder, la tasa de errores cometidos por el usuario deberá ser menor del 1% de las transacciones totales ejecutadas en el sistema.
* Diseño de interfaz, el sistema debe poseer interfaces gráficas bien formadas e intuitivas.

**Requerimiento Funcional**

1. Notificaciones al usuario, el sistema notificará cuando se registre alguna de las siguientes acciones: atraso o falta al día de trabajo por parte de un trabajador.
2. **Registrar usuarios o personal de la misma institución**, El sistema permitirá el registro de todos los trabajadores de la empresa con posibilidad de agregar o eliminar trabajadores en caso de requerirlo
3. Realizar consultas, el sistema permitirá a los usuarios autorizados realizar consultas de acuerdo al criterio que se requiera.
4. Actualización de horarios, el sistema permitirá aprobar, cambiar o actualizar horarios de trabajo y las diferentes jornadas laborales posibles.
5. Registro de datos manuales, el sistema también permitirá el registro de asistencias manuales, sin embargo, estas requerirán autorización por parte del Gerente.
6. **Registro por lotes o automática**, El registro datos podrá configurarse para realizarse de forma automatizada a su registro, o manualmente en lotes.
7. Generar Reportes, el software debe poder emitir los siguientes reportes: carga laboral por empleado, horarios de trabajo, retrasos del personal de manera, semanal mensual, anual.
8. Validación de usuarios, el sistema validara automáticamente el empleado con la base de datos de trabajadores de la empresa.
9. **Control de acceso**, El sistema controlará el acceso y lo permitirá solamente a usuarios autorizados.
10. Plataforma de funcionamiento, la aplicación debe ser de escritorio con conexión local.

Sección III Propuesta

1. Solución.-

Para la solución del problema del cliente final Dentales Pablo Herman S.A. se ha desarrollarlo el sistema interprete de datos de reloj biométrico, para el cual se ha utilizado diversas metodologías investigativas para obtener la mayor cantidad de información posible para que sea analizada y contextualizada en el presente proyecto, todo con el objetivo de solucionar el problema planteado por el cliente en el diseño y la implementación del sistema.

1. Scrum

Para el desarrollo del presente proyecto se hizo uso la metodología de desarrollo ágil SCRUM, donde se han seguido los procedimientos de: Requerimientos, Análisis, Diseño, Evolución, de esa manera se han generado los documentos product Backlog y Sprint Backlog que fueron desarrollados en base a los requerimientos funcionales, no funcionales e historias de usuario recopiladas en la fase de análisis, dentro del proceso de desarrollo de se han establecido 6 Sprints que se detallan en:

* 1. SPRINT 1: Diseño Módulo de Ingreso de Usuarios (login)

Dentro de este sprint, se integraron todos los datos. Estos fueron enlazados de

acuerdo al tipo de información sobre el proceso manejado en la unidad

educativa. Para el diseño de la base de datos se utilizó el modelo entidadrelación, en el cual se seleccionaron las entidades con sus respectivos atributos

y luego se efectuaron las respectivas relaciones

* 1. Control de versiones.-

Con el objetivo de manejar de manera más organizada las versiones de los diferentes sprints que se realizaron en el presente proyecto se hizo uso de herramientas de control de versiones, es el caso de GIT y GIT HUB, de donde además de controlar las versiones se pudo hacer un seguimiento de los progresos realizados por cada uno de los participantes del proyecto, lo cual se verifica en archivo anexo

* 1. IDE Desarrollo

Para el desarrollo de la presente aplicación se ha usado el IDE de desarrollo Netbeans

IDE DESARROLLO

MY SQL

PLATAFORMA DE DESARROLLO SWING

Cabe destacar que la presente herramienta de software fue desarrollada bajo la metodología Scrum, utilizando todos los parámetros de ella, los cuales se detallan en los anexos.

* 1. PROTOTIPO

Con el objetivo de validar que los requerimientos fueron correctamente interpretados por el Product Owner, se procedió a realizar prototipos de funcionalidades del software, el objetivo de los prototipos del software es mostrar al usuario final las funcionalidades más relevantes en relación a los requisitos funcionales y parte de los no funcionales, tomando en cuenta la prioridad de estos siendo los de categoría 5 como los más primordiales, para este proceso se realizó utilizando el prototipado de estilo vertical.

**Herramienta a usar para el prototipado del software**

Para este proyecto se ha considerado conveniente usar **Pencil Project**, ya que es una herramienta de código abierto y de distribución gratuita la cual nos permitirá presentar a nuestro cliente el o los prototipos del software a desarrollar de una forma sencilla y práctica.

**Usuarios con los que aplicará el test**

Los usuarios con quienes se aplicarán los test de aceptación en relación a los prototipos del software son:

Ab. Manuel Quijo Villamarin – Gerente General

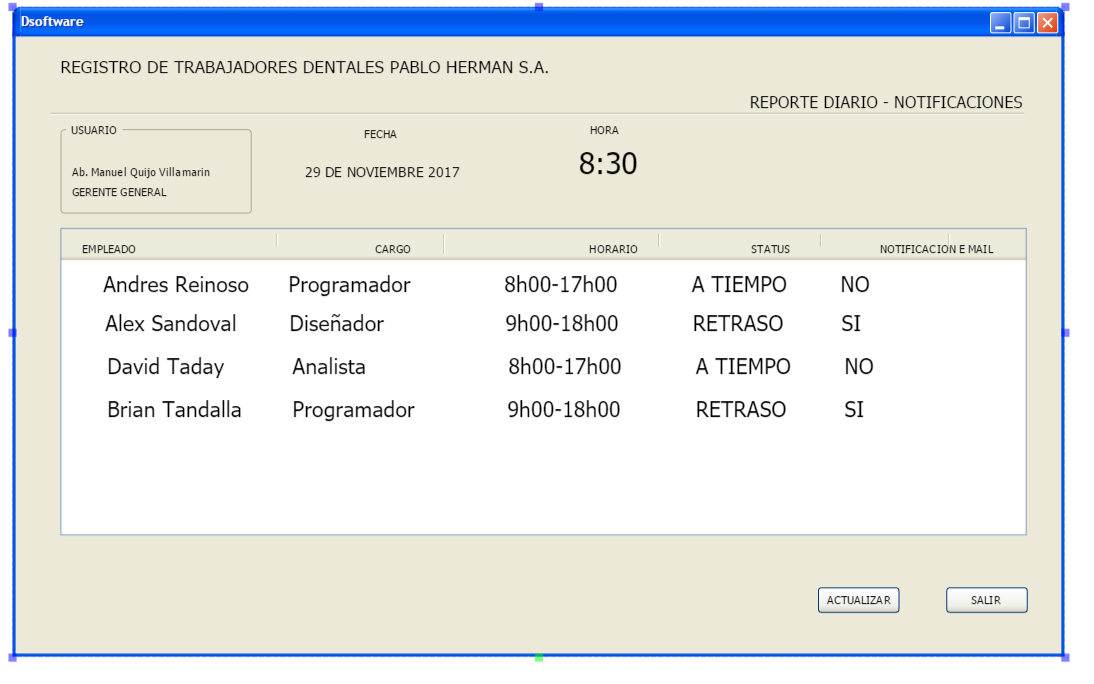
Lic. Laura Palacios – Jefa de RRHH

**Funcionalidades a mostrar en el prototipo**

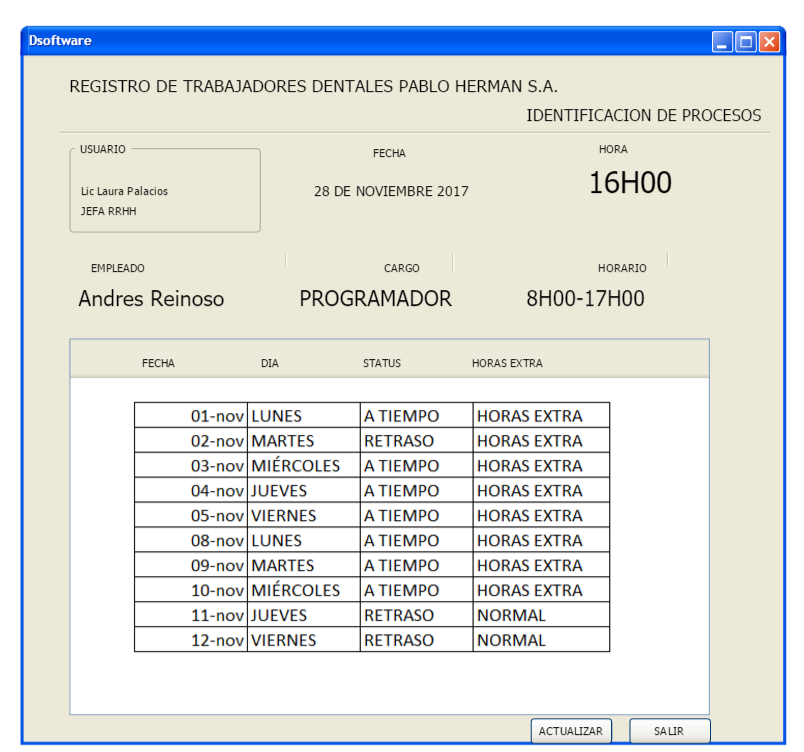
Las funcionalidades del prototipo son:

NOTIFICACIONES DEL USUARIO

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF01 |
| **Nombre del Requerimiento:** | Notificaciones el usuario |
| **Características:** | Los usuarios recibirán notificaciones según las acciones que sean programadas en el sistema. |
| **Descripción del requerimiento:** | El sistema enviará un correo electrónico cuando se registre alguna de las siguientes acciones: atraso o falta al día de trabajo por parte de un trabajador. |
| **Requerimiento NO funcional:** | * RNF011 * RNF010 |
| **Prioridad del requerimiento:** Baja | |



|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF05 |
| **Nombre del Requerimiento:** | Registro de trabajadores Dentales Pablo Herman S.A. |
| **Características:** | El sistema permitirá que se pueda verificar el estatus de horas de trabajo de cada trabajador |
| **Descripción del requerimiento:** | A cada día de trabajo se le asignará un identificador único, que será utilizado para identificarla en todos los procesos subsecuentes que se realicen sobre este como el control de horas de trabajo. |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 |
| **Prioridad del requerimiento:** Media – Alta | |



Los presentes prototipos sirvieron de base para que el usuario final haga sus observaciones y sugerencias acerca del desarrollo del sistema en donde se puede observar del acta de revisión adjunta en el anexo.

Conclusiones y recomendaciones

1. Conclusiones

Finalmente, el desarrollo investigativo del software ha permitido conocer todas las etapas que este proceso implica desde el análisis del problema hasta el factor más mínimo, todo esto suman para lograr un proyecto de calidad y altamente competitivo dentro del mercado.

Fruto de las pruebas de aceptación realizadas en la implementación del presente software podemos señalar que se ha cumplido con los requerimientos solicitados por el cliente final, entregando de esta manera software de calidad bajo procesos de optimización de tiempo y procesos.

La implementación del presente proyecto se realizó sin inconvenientes en la empresa Dentales Pablo Herman S.A. al momento se esta haciendo uso de la herramienta y se ha reportado mejoras en el rendimiento laboral y control administrativo.

Del diseño, implementación, validación y pruebas del presente proyecto se concluye que el proceso fue exitoso, cumplimento a cabalidad lo peticionado por el cliente final.

1. Recomendaciones

En base a la investigación realizada se recomienda el desarrollo del segundo modulo de la aplicación, beneficiando de esta manera el rendimiento laboral del personal administrativo de control de la empresa Dentales Pablo Herman S.A.

Como resultado del estudio comparativo realizado se pudo verificar que no existe implementaciones tecnológicas de esta naturaleza, en tal sentido se recomienda su comercialización de manera masiva.

Bibliografía

1. Alarcon, A., & Sandoval, E. (2011). Herramientas CASE para ingeniería de Requisitos. *Cultura Científica*, *6*(6), 70–74.
2. Constituyente, E. A. (2008). Constitución de la República del Ecuador. Recuperado a partir de <http://repositorio.dpe.gob.ec/bitstream/39000/638/1/NN-001-Constituci%C3%B3n.pdf>
3. DECSAI Universidad de Gra ada. (s/f). Especificacion de requerimientos, Diseño de Base de datos.
4. Fernando Barraza A. MS.c. (s/f). Arquitectura conceptos de modelado.
5. Hernández, V., & Alfredo, L. (2003). Manual para la Diagramación de Procesos. *Noviembre del*. Recuperado a partir de http://docencia.fca.unam.mx/~lvaldes/cal\_pdf/cal18.pdf
6. ITECO Gobierno de España,. (s/f). INGENIERIA\_DEL\_SOFTWARE\_CicloVida.pdf.
7. Jesús Bustínduy Basterrechea. (s/f). Desarrollo detallado de la fase de aprobación de un proyecto informático. *2004*. Recuperado a partir de http://openaccess.uoc.edu/webapps/o2/bitstream/10609/575/1/27733tfc.pdf
8. Méndez, G. (s/f). Ingeniería de Requisitos. Recuperado a partir de https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/03-Requisitos.pdf
9. Molina Quiroga, E. P. (2015). *Los principios del derecho colectivo de trabajo en la constitución, en el código de trabajo y en el proyecto de código orgánico de relaciones laborales* (B.S. thesis). Guayaquil: Universidad Del Pacífico. Recuperado a partir de http://repositorio.upacifico.edu.ec/handle/40000/129
10. Rodríguez, J. B. (s/f-a). Trabajo Fin de Grado Grado en Ingeniería de Organización Industrial. Recuperado a partir de http://bibing.us.es/proyectos/abreproy/91033/fichero/91033.pdf
11. Rodríguez, J. B. (s/f-b). Trabajo Fin de Grado Grado en Ingeniería de Organización Industrial. Recuperado a partir de http://bibing.us.es/proyectos/abreproy/91033/fichero/91033.pdf
12. TipoDeSoftware.pdf. (s/f).
13. Veronica Valdez Alvarado. (2012, enero). Tecnicas efectivas para la toma de requerimientos. NorthWale Software Development.
14. (INTECO), I. N. (2009). Curso de introducción a la ingeniería de software. *Plan Avanza 2*, 22.
15. Pressman, R. (2014). *Ingeniería de Software enfoque practico.* McGraw-Hill local.
16. SYSTEM-GLOBAL. (2017). *SYSTEM-GLOBAL*. Obtenido de http://www.sdgku.com
17. Manuel Trigas Gallego. (s/f). *Metodologia Scrum*.
18. Ingeniería de Programación (4º Físicas) J.M. Drake
19. Gutiérrez, M. G. C. (s/f). Grafic 1.0 Un Recurso Tecnológico como Apoyo al Aprendizaje de Geometría Análitica-Edición Única.
20. Fonseca, E. (2016). Evaluación de la efectividad de las técnicas de pruebas de software estructurales y funcionales mediante replicación experimental, caso práctico ESPE sede Latacunga. (Master’s Thesis). Universidad de las Fuerzas Armadas ESPE. Vicerrectorado de Investigación, Innovación y Transferencia de Tecnología.
21. Cordova Castañeda, A. O. (2016). Conceptos de Base de Datos (B.S. thesis).

ANEXOS

1. (UNIVERSIDAD DE CALIFORNIA, LOS ÁNGELES, 2017) [↑](#footnote-ref-1)