



Universidad de las Ciencias Informáticas

Facultad 1

“Aplicación informática para contribuir a la prevención de enfermedades profesionales en los informáticos”

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

Autora

Marielsis Alina Hernández Borrero.

Tutores

MSc. Nadia Porro Lugo.

MSc. Silvano Merced Len.

Ing. Hanny Valdés Hernández.

La Habana, 19 de mayo 2016

“Año 58 de la Revolución”

Declaración de Autoría

Declaro ser la única autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los días ____ del mes de _____ del año_____.

MSc. Nadia Porro Lugo.

MSc. Silvano Merced Len.

MSc. Hanny Valdés Hernández.

Pensamiento



“La cuestión sería combatir las causas y no conformarse con tener éxito en suprimir efectos.”

Ernesto Che Guevara.

Agradecimientos

Agradezco de manera especial a Dios, porque nunca me abandonó cuando más lo necesitaba.

A mis padres María Isabel y Eutimio por estar siempre a mi lado y apoyarme en los momentos más difíciles de mi vida. Por brindarme todo su amor y cariño y la confianza de poder contar con ellos. Por ser la luz de sus ojos y cuidarme en todo momento. Los Amo.

A mis abuelas Carmen Rosa y Felicita por estar presente en cada momento de mi vida y darme todo su amor y cariño. Por malcriarme como a nadie y por iluminar mi vida.

A Damaris, Yoandris, Lisbeth, Arelis, Maruca, Celeste, Miguel, Alberto, a toda mi familia le agradezco el apoyo y el cariño que me han dado.

A mi novio José Luis por estar estos cuatro años a mi lado y poder contar con él en los buenos y malos momentos. Por entenderme y dedicarme la paciencia que muchas veces necesitaba. Por formar parte de mi vida y aceptarme como soy. Le doy gracias a dios por haber conocido a una persona tan maravillosa como tú.

Le agradezco a mi amiga-hermana Daneysis por estar siempre a mi lado, aconsejarme, apoyarme en todo momento y regañarme muchas veces por mi bien.

A Maritza y Mireya por acogerme cariñosamente como nuevo miembro de la familia, aconsejarme y poder contar con ellas en todo momento.

Agradecimientos

A Odi por dedicarle tiempo a mi documento, revisarlo y preocuparse. Por la amistad que creamos en tan poco tiempo. Muachas gracias.

A las tutoras Nadia y Hanny le agradezco la paciencia y la dicha de trabajar con personas como ellas. Sin ellas el resultado de este trabajo no sería posible. Les doy las gracia por las intensas horas de trabajo y la dedicación que me otorgaron. Por el ánimo que me daban para seguir adelante y lograr obtener el título de ingeniero en ciencias informática.

A los miembros del tribunal por el trabajo tan excepcional que han realizado. A todos mis compañeros de aula en especial a Agnes, Felipe, Yosel, Yosmay, y Yordis por hacerme sentir un miembro importante del grupo.

A Iguelit y Carlos por la amistad que me brindaron y el cariño que me dieron en tan poco tiempo. Son unas personas maravillosas.

Les agradezco a todos en general por estar presente en este largo camino. Gracias.

Dedicatoria

Pido disculpa a mis padres, a mis abuelas y a mi novio, pero este trabajo de diploma tiene un solo nombre: "Eutimio".

Le dedico este trabajo a mi abuelo que aunque no se encuentra físicamente hoy conmigo lo amo y es el rayito de luz que me guía y me anima a salir adelante. Nunca lo olvidaré y siempre formará parte en mi vida aunque no esté presente.

A ti abuelo querido te dedico esta tesis.

Te Amo.

Marielsis Alina Hernández Borrero.

Resumen

La tecnología se ha desarrollado con el transcurso del tiempo permitiendo a la humanidad desarrollarse y aportar conocimientos en el ámbito político, científico, económico, social y cultural. Este avance tecnológico ha facilitado las actividades de la vida cotidiana y ha propiciado que las personas hayan creado dependencia de la tecnología en el quehacer diario de su labor u oficio. El uso continuado de la tecnología, a pesar de los beneficios que aporta, si no se utiliza adecuadamente puede causar afecciones que dañan la salud. Estas afecciones son conocidas como las enfermedades profesionales.

Las enfermedades profesionales afectan la calidad de vida y el rendimiento de los recursos humanos en las instituciones. En la Universidad de las Ciencias Informáticas se ha podido identificar la presencia de estas, como por ejemplo la tendinitis, el síndrome cervical por tensión, la bursitis, entre otras, producto de la exposición continuada del mal uso del ordenador y la poca actividad física.

Para evitar el desarrollo de estas enfermedades la Vicerrectoría de Extensión Universitaria de la universidad se propuso la construcción de una aplicación que permitiera alertar sobre estas y orientara sobre las acciones a realizar para ayudar a su prevención. De allí se deriva que el objetivo de este trabajo consista en desarrollar una aplicación informática que contribuya a la prevención de enfermedades profesionales en los informáticos.

Para guiar el proceso de desarrollo de la aplicación se seleccionó la metodología AUP-UCI y C++ como lenguaje de desarrollo, con el *framework* Qt y su IDE de desarrollo Qt Creator. Se seleccionó también el lenguaje HTML para mostrar la información de la aplicación con el IDE de desarrollo WebStorm.

Palabras Clave: aplicaciones, ejercicio físico, enfermedades profesionales, informática, prevenir enfermedades, salud.

Abstract

The technology has been developed over time allowing humanity to develop and provide knowledge in the political, scientific, economic, social and cultural. This technological progress has facilitated the activities of daily life and has meant that people have created dependence on technology in the daily tasks of their work or trade. The continued use of technology, despite the benefits, if not used properly can cause conditions that harm health. These conditions are known as occupational diseases.

Occupational diseases affect the quality of life and performance of human resources in institutions. At the University of Information Science, it has been able to identify the presence of these, such as tendinitis, cervical syndrome, stress, bursitis, among others, product of sustained exposure of computer misuse and little physical activity.

To prevent the development of these diseases the Vice-Rector of University Extension University building an application that would alert about these and guidance on the actions to take to help prevention was proposed. Hence it follows that the objective of this work consists in developing a software application that contributes to the prevention of occupational diseases in the computer.

To guide the development process of implementing the AUP-UCI and C++ language development methodology and was selected with the framework Qt and Qt Creator IDE development. HTML language is also selected to display information application development IDE with WebStorm.

Keywords: applications, physical exercise , occupational diseases, informatics, prevent disease, health.

Índice de Contenido

INTRODUCCIÓN	7
1.2. CONCEPTOS ASOCIADOS A LAS ENFERMEDADES PROFESIONALES DE LOS INFORMÁTICOS	7
1.2.1. Enfermedades Profesionales.....	7
1.2.2. Factores ambientales	8
1.2.3. Lesiones más frecuentes que se pueden producir.....	10
1.2.4. Consecuencias de las enfermedades profesionales informáticas	11
1.2.5. Posiciones adecuadas para evitar las enfermedades profesionales	12
1.2.6. El ejercicio y las enfermedades profesionales	14
1.3. APLICACIONES INFORMÁTICAS EXISTENTES PARA PREVENIR LAS ENFERMEDADES PROFESIONALES.....	15
1.3.1. Software Educativo: “Mente Sana en Cuerpo Sano”.....	15
1.3.2. Aplicación informática para la evaluación de los factores de riesgos laborales en el sector empresarial cubano 16	
1.3.3. Pomodoro	16
1.4. TENDENCIAS ACTUALES A CONSIDERAR EN EL DESARROLLO.....	18
1.5. METODOLOGÍA DE DESARROLLO.....	19
1.5.1. Selección de la metodología para el desarrollo de la aplicación.....	19
1.6. HERRAMIENTAS Y TECNOLOGÍAS	22
1.6.1. Herramienta CASE y lenguaje para el modelado.....	22
1.6.2. Lenguaje de programación	23
1.6.3. <i>Framework</i> de desarrollo	25
1.6.4. Entorno de desarrollo (IDE)	26
CONCLUSIONES PARCIALES	27
INTRODUCCIÓN	28
2.1. CARACTERÍSTICAS DE LA APLICACIÓN PREVENPRO.....	28
2.2. MODELO DE DOMINIO	29
2.3. ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE.....	30
2.3.1. Requerimientos Funcionales	30
2.3.2. Requerimientos no Funcionales	31
2.4. HISTORIAS DE USUARIOS	32
2.5. ARQUITECTURA DE SOFTWARE.....	39

Índice de Contenido

2.5.1. Estilo Arquitectónico	39
2.6. PATRONES DE DISEÑO.	40
2.7. MODELO DE DISEÑO.....	42
2.7.1. Diagrama de clases	43
CONCLUSIONES PARCIALES	44
INTRODUCCIÓN	45
3.1. MODELO DE DESPLIEGUE	45
3.2. MODELO DE COMPONENTE	45
3.3. ALMACENAMIENTO DE DATOS	46
3.3.1. Fichero de configuración	46
ESTÁNDARES DE CODIFICACIÓN	47
PRUEBAS DE SOFTWARE.....	50
3.1.1. Pruebas Funcionales	50
3.1.2. Resultados obtenidos	56
CONCLUSIONES PARCIALES	57

Índice de ilustraciones

Ilustración 1: Aplicación del método de la estrella.....	20
Ilustración 2: Modelo de dominio.....	29
Ilustración 3: Arquitectura n-capas (2).....	40
Ilustración 4: Diagrama de clases.	44
Ilustración 5: Diagrama de despliegue	45
Ilustración 6: Diagrama de componentes.....	46
Ilustración 7: Estructura del fichero de configuración.	47
Ilustración 8: Fragmento de código fuente de la clase UIMenuContenido.	48
Ilustración 9: Fragmento de código del método ControladorApp().....	48
Ilustración 10: Ejemplo de declaración de variable.....	49
Ilustración 11: Ejemplo de sentencias de inclusión	49
Ilustración 12: Ejemplo de sentencia if.	49
Ilustración 13: Ejemplo de sentencia for.....	50
Ilustración 14: Comportamiento por iteraciones de las pruebas funcionales.	57

Índice de Tablas

Tabla 1: Comparando las aplicaciones existentes.....	17
Tabla 2: Historia de Usuario 1	32
Tabla 3: Historia de Usuario 2.....	33
Tabla 4: Historia de usuario 5	36
Tabla 5: Historia de usuario 6	37
Tabla 6: Caso de Prueba Mostrar notificaciones.....	51
Tabla 7: Caso de Prueba Acceder al menú de contenidos.....	52
Tabla 8: Caso de Prueba Seleccionar tipo de notificaciones a mostrar.....	52
Tabla 9: Caso de Prueba Configurar notificaciones.....	53
Tabla 10: Caso de Prueba Adicionar soporte para agregar plugin a la barra de tarea.....	54
Tabla 11: Caso de Prueba Sugerir guía de ejercicio.....	55

Introducción

La tecnología se ha desarrollado con el transcurso del tiempo permitiendo a la humanidad desarrollarse y aportar conocimientos en el ámbito político, científico, económico, social y cultural. La automatización de procesos y actividades ha facilitado la realización de las tareas laborales y cotidianas con menos esfuerzo físico y mental, propiciando la creación de valor de manera más eficiente a través de la actividad económica. Como consecuencia las personas cada vez son más dependiente de la tecnología en el desarrollo de su vida diaria. Esto acarrea en ocasiones que se desarrollen afecciones que dañan la salud como resultado del mal uso continuado de la tecnología. Estas afecciones son conocidas como las enfermedades profesionales.

Se denomina enfermedad profesional a aquella enfermedad adquirida en el puesto de trabajo. En el mundo estas enfermedades ocasionan pérdida de recursos y de personal para las empresas, afectando a la economía de los países, donde la mayor fuente de ingreso es la producción. Por tal motivo, la Organización Mundial de Trabajo (OIT) estableció en el 2003, el 28 de abril como el Día Mundial de la Seguridad y la Salud en el Trabajo, que consiste en una campaña anual para promover el trabajo seguro, saludable y digno (Somavia, 2015).

La informática es una ciencia que ha avanzado con pasos agigantados, permitiéndole a las personas una vía de mejorar la educación y fortalecer el desarrollo comunitario. Se encarga de automatizar la información que tiene una fuerte dependencia del capital humano y una baja utilización del capital físico. Su dinamismo se basa en el conocimiento y la innovación, lo que otorga un fuerte potencial para generar valor agregado (Merced, 2012). Entre los beneficios que brinda esta ciencia están: la obtención de resultados de forma más rápida, el almacenamiento de grandes volúmenes de información, facilita la comunicación entre personas y permite acelerar los procesos de investigación.

Sin embargo, está considerada como una ciencia de bajo volumen de actividad motora, con posturas prolongadas en el tiempo, sedentarias e intensas horas de trabajo mental. Estos elementos unidos a los factores ergonómicos del trabajo traen como consecuencia el padecimiento de molestias y dolencias que se han calificado como enfermedades de la profesión (Merced, et al., 2012).

Introducción

Cuba, como muchos países se integra a la revolución tecnológica y a la búsqueda de distintos medios que permitan aprovechar sus beneficios. En el año 2002 la Universidad de las Ciencias Informáticas (UCI), abre sus puertas al país con la misión de formar profesionales comprometidos con la Patria y altamente calificados en la rama de la Informática. Fomentando la producción de aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación, y servir de soporte a la industria cubana de la informática (UCI, 2012).

En aras de velar por la salud de los profesionales de la institución se realizó una entrevista (**Ver Anexo 1**) a los especialistas del área de rehabilitación del Hospital Ernesto Che Guevara de la UCI, en la que pudieron identificarse las enfermedades presentes en la universidad:

- La fatiga visual por una larga exposición a los ordenadores, tabletas y televisión que traen como síntomas ardor en la vista, mareo y dolor de cabeza.
- El dolor de espalda, cuello y cervical son síntomas bastante frecuentes en las personas de la universidad, debido a las malas posturas adoptadas para trabajar frente al ordenador.
- La primera causa de absentismo laboral y como consecuencia responsable de la disminución de la productividad es el estrés, que arroja como síntoma la migraña.
- El síndrome del túnel carpiano es otra enfermedad que provoca la pérdida de fuerza en las manos, causada por la flexión reiterada de la muñeca, produce un fuerte dolor en la muñeca y en el codo y puede derivarse en una tendinitis.
- La tendinitis que afecta el codo, la rodilla y la muñeca.
- La bursitis causando inflamación debido a los movimientos repetitivos.

Al reflexionar sobre estos aspectos se puede comprobar que la necesidad del ejercicio físico ha perdido importancia y el ser humano se ha adaptado a las comodidades, y a las diversas funciones que adquiere. Lo que propicia que no se tengan en cuenta estas enfermedades profesionales, ni se le preste la atención que se necesita. Lo que hace necesario que las personas obtengan el conocimiento y las medidas de prevención en relación con su salud y bienestar acerca de las diferentes limitaciones físicas que genera.

Introducción

Existen hoy diferentes mecanismos en la universidad que brindan información de las enfermedades profesionales. Ejemplo de ello son los profesores de educación física que brindan distintos cursos electivos de diferentes temas, como por ejemplo el Curso de Masaje. Las instructoras educativas también realizan encuentros con los estudiantes donde brindan información sobre la importancia del ejercicio físico y el cuidado de la salud. Además, los especialistas del hospital, se encargan de informar a los pacientes que asisten a las consultas de cómo evitar las afectaciones causadas por las enfermedades profesionales a través de guías de ejercicios.

Pero estos medios y métodos empleados no son suficientes ya que están concentrados a un espacio limitado y no se utilizan con la debida frecuencia por parte de los profesionales de la institución. Además, la exigencia y dinámica de los proyectos hace difícil a los especialistas informáticos seguir un programa regular de ejercicios. Es por esto que surge la necesidad de crear un medio que permita prevenir las enfermedades profesionales de manera frecuente en la universidad.

De la problemática anterior surge como **problema de investigación** ¿Cómo automatizar los métodos de información que se brindan en la universidad sobre las enfermedades profesionales en los informáticos?

A partir del problema identificado se puede definir como **objeto de estudio**: las enfermedades profesionales en los informáticos. El **campo de acción** se enmarca en: las aplicaciones para la prevención de enfermedades profesionales en los informáticos.

El **objetivo general** de la investigación se centra en: desarrollar una aplicación informática que automatice los métodos de información que se brindan en la universidad sobre las enfermedades profesionales en los informáticos.

Del objetivo general se derivan los siguientes **objetivos específicos**:

- Caracterizar las enfermedades profesionales en los informáticos que afectan a la universidad y las aplicaciones que apoyan a su prevención, con el objetivo de determinar las funcionalidades que deben estar presente en la aplicación y las tecnologías que se utilizarán para construirla.

Introducción

- Analizar y diseñar una aplicación que automatice los métodos de información que se brindan en la universidad sobre las enfermedades profesionales.
- Implementar la aplicación en correspondencia de las características y tecnologías definidas y obtener resultados de prueba satisfactorios a través de la verificación de las funcionalidades.

Para darle respuestas a los objetivos específicos se derivan las siguientes **tareas investigativas**:

1. Análisis de los conceptos básicos y técnicos relacionados con la aplicación para prevenir las enfermedades profesionales en los informáticos.
2. Caracterización de las aplicaciones que apoyan a la prevención de enfermedades profesionales.
3. Identificación de los requisitos funcionales y no funcionales de la aplicación para prevenir las enfermedades profesionales en los informáticos.
4. Confección de los diagramas generados por la metodología seleccionada.
5. Implementación de la aplicación para automatizar los métodos de información en la universidad sobre las enfermedades profesionales en los informáticos.
6. Diseño de los casos de prueba a partir de los requisitos del sistema para llevar a cabo la realización de las pruebas.
7. Realización de las pruebas funcionales para probar el correcto funcionamiento de la aplicación y las pruebas de aceptación para validar la aplicación con el cliente.

Los métodos científicos que se utilizaron durante la investigación son:

Métodos Teóricos.

- **Histórico-Lógico:** se utilizó este método para el estudio de la trayectoria histórica. Además, para ver el desarrollo de las aplicaciones relacionadas con las enfermedades profesionales en el ámbito nacional e internacional. Este se complementa por el método lógico, al cual se recurrió para la interpretación de los fundamentos teóricos y las apreciaciones de los autores consultados.
- **Hipotético-Deductivo:** este método se utilizó para llegar a nuevos conocimientos y predicciones a partir del estudio de las funcionalidades de las aplicaciones existentes y establecer las funcionalidades que se implementarán en la aplicación. Permite adelantar y verificar nuevos

Introducción

problemas en la realidad, inferir conclusiones y establecer predicciones a través de conocimientos que ya se poseen.

- **Analítico-Sintético:** este método se utilizó con el objetivo de analizar y resumir las principales fuentes relacionadas con las enfermedades profesionales (incluyendo las lesiones y causas que provocan). Además, para analizar las características y los elementos fundamentales que poseen las aplicaciones que apoyan la prevención de enfermedades profesionales existentes para obtener un correcto diseño de la aplicación.

Métodos Empíricos.

Entrevistas: se realizó una entrevista para recoger información sobre las enfermedades profesionales, y de ellas identificar las que afectan a la población de la UCI. La población escogida para la realización de la entrevista fueron los especialistas del área de rehabilitación del Hospital Ernesto Che Guevara con un 100% en su totalidad.

La presente investigación está estructurada en tres capítulos, los cuales se muestran a continuación:

Capítulo I. Fundamentación teórica: en este capítulo se abordarán los principales conceptos básicos relacionados con las enfermedades profesionales en los informáticos y se analizan las aplicaciones existentes que apoyan a su prevención. Se realiza además la selección y descripción de la metodología seleccionada que guiará el desarrollo de la aplicación y de las herramientas empleadas para su construcción.

Capítulo II. Características de la aplicación PrevenPro para la prevención de las enfermedades profesionales: este capítulo muestra el análisis y diseño de la propuesta de solución para prevenir las enfermedades profesionales en los informáticos. Se describen y representan los conceptos de dominio del problema a través del modelo de dominio. Se proponen los requisitos funcionales y no funcionales y se describen a través de las historias de usuario. Se selecciona el estilo arquitectónico para lograr una correcta estructura de los componentes del sistema.

Introducción

Capítulo III. Implementación y prueba: en este capítulo se muestran los elementos de implementación y prueba de la propuesta de solución. Se presenta el modelo de implementación de la aplicación y se describen los métodos de pruebas a utilizar. Se exponen los resultados obtenidos de la aplicación que ratifican que esta cumple el objetivo para el cual fue construida.

El trabajo contiene además **Conclusiones, Recomendaciones, Referencias Bibliográficas, Bibliografía Consultada, Glosario de Términos y Anexos.**

Capítulo I. Fundamentación Teórica

Introducción

En el presente capítulo se abordan, los elementos principales que justifican y soportan técnicamente la investigación. Se realiza un análisis de las aplicaciones existentes que apoyan la contribución de las enfermedades profesionales. Se presentan las tecnologías y la metodología a utilizar en el desarrollo de la aplicación para contribuir a la prevención de las enfermedades profesionales en los informáticos.

1.2. Conceptos asociados a las enfermedades profesionales de los informáticos

Para lograr un mayor conocimiento del entorno de la aplicación, es necesario realizar un estudio de los principales conceptos asociados a las enfermedades profesionales en los informáticos. Es importante analizar la definición de enfermedades profesionales, identificar los factores ambientales que pueden condicionar un problema para la salud de las personas y las lesiones que pueden aparecer. Identificar las consecuencias que pueden ocurrir, las posiciones que debe adoptar una persona para evitar estas enfermedades profesionales y la importancia del ejercicio físico.

1.2.1. Enfermedades Profesionales

Una Enfermedad Profesional es aquella que es causada, de manera directa, por el ejercicio del trabajo que realice una persona y que le produzca incapacidad. Para ser considerada como Enfermedad Profesional, debe existir una relación causal entre el quehacer laboral y la patología que provoca la invalidez (ISL, 2014).

Estas enfermedades son generadas por diversos factores de condiciones de trabajo. Son de paulatino crecimiento y van afectando lentamente a las personas hasta llegar al punto máximo de inutilizarlos o producirle daños irreversibles, causado por el excesivo y mal uso de las innovaciones tecnológicas.

Existen múltiples factores de riesgo como la obesidad causada por sedentarismo, problemas posturales, deformaciones de miembros, estrés y dependencia por presiones laborales (Mendez, y otros, 2011).

La Unión General de Trabajadores de España define las enfermedades profesionales como aquel deterioro lento y paulatino de la salud del trabajador, producido por una exposición crónica a situaciones

Capítulo I. Fundamentación Teórica

adversas, sean estas producidas por el ambiente en el que se desarrolla el trabajo o por la forma en que este se encuentra organizado.

Las enfermedades profesionales pueden estar presentes en las actividades inmobiliarias, informáticas, domésticas, auxiliares, sanitarias, recreativas, culturales y deportivas. Estas enfermedades son clasificadas a partir de los riesgos físicos, químicos, biológicos, los derivados de la carga de trabajo y los psicosociales, que muchas veces cuando las medidas de prevención y control en el trabajo fallan, pueden aparecer las enfermedades profesionales.

Para clasificar una enfermedad profesional, no basta con la especificación antes mencionada, es necesario que haya sido causada por un específico agente material de riesgo con el cual estuviera el trabajador en contacto en su lugar de trabajo. Algunas de estas enfermedades son:

- Enfermedades profesionales producidas por agentes químicos.
- Enfermedades profesionales por la piel.
- Enfermedades profesionales provocadas por inhalación.
- Enfermedades profesionales sistemáticas o sistémicas.
- Enfermedades tecnológicas o patológicas informáticas.

También existen otros tipos de factores que posibilitan el padecimiento de estas enfermedades como las ambientales, que pueden traer como consecuencia las enfermedades profesionales cuando las empresas no disponen de las condiciones necesarias de higiene y seguridad.

1.2.2. Factores ambientales

Las inadecuadas condiciones ambientales generan incomodidad, malestar y dificultades en la concentración que afecta el desempeño laboral, el trabajo en equipo y sobre todo el cumplimiento de las funciones del trabajo (Quinga, 2015).

Capítulo I. Fundamentación Teórica

La ergonomía¹ ambiental analiza todos estos factores del entorno para prevenir su influencia negativa y conseguir la mayor comodidad y bienestar del trabajo para un óptimo rendimiento como la ventilación, el ruido y la iluminación. Julio Quinga en (Quinga, 2015) define estos problemas de la siguiente forma:

Ventilación

Un diseño incorrecto del sistema de ventilación puede contribuir a la formación de ambientes a los que no llegue el aire limpio. Las principales fuentes de contaminación debido a una mala ventilación son el humo del tabaco, para los trabajadores que fumen dentro de los locales, productos de limpieza e insecticidas. Un excesivo calor o frío produce incomodidad, pero también somnolencia o ansiedad e inquietud. La humedad relativa del aire puede provocar sequedad de las mucosas² respiratorias y molestias, por lo que es aconsejable mantener una temperatura ambiental entre 19^o y 24^o C y una humedad relativa entre 40 % y 70 %.

Iluminación

Se debe disponer de un equipo de iluminación adecuado al tipo de trabajo y tarea visual que se esté realizando. Es necesario tener en cuenta no solo la cantidad de luz necesaria, sino también la calidad de la luz, evitando contrastes y deslumbramientos. Es importante combinar las iluminaciones evitando luces intensas en el campo de visión. Una iluminación inadecuada supone un riesgo que puede desatar la fatiga visual y otros trastornos oculares y visuales. Una iluminación correcta aumenta la eficacia y comodidad del trabajo que se está realizando.

Ruido

El ruido es un contaminante ambiental que puede producir irritación, hipoacusia³ o fatiga auditiva, también puede generar daños y efectos indeseables de tipo extra-auditivo. Lo deseable sería que las exposiciones

¹ Es una disciplina científico-técnica y de diseño que estudia la relación entre el entorno de trabajo y quienes realizan el trabajo.

² La mucosa es un tejido orgánico de consistencia suave que recubre la pared interna de ciertos órganos y de ciertas cavidades naturales externas, como son las fosas nasales o los oídos.

³ Pérdida de audición.

Capítulo I. Fundamentación Teórica

al ruido no sobrepasen los 80 dB⁴. La música puede producir agradables sensaciones como el bienestar y la satisfacción, pero es importante tener en cuenta que el ritmo no se sobrepase. Un mal uso de la música puede causar grandes problemas de audición.

A parte de los factores ambientales, existen lesiones que aparecen con lentitud, provocadas por herramientas de trabajo y lugares de trabajo. Es importante que los trabajadores estén informados de estos tipos de lesiones, y así contribuir a la prevención de las mismas.

1.2.3. Lesiones más frecuentes que se pueden producir

Los riesgos laborales son peligros que existen alrededor de una actividad laboral como la realización de trabajos repetitivos. Pueden dar lugar a trastornos músculo-esquelético, es decir, lesiones de tipo inflamatorio o degenerativo de músculos, tendones, nervios, articulaciones y ligamentos. Estos trastornos afectan las zonas del cuello, la espalda, hombros, codos, muñecas, manos, dedos y piernas (Quinga, 2015). Aparecen de forma lenta y paulatina, y en un principio parecen inofensivas. A continuación, se mencionan ejemplos de estas lesiones:

- **Tendinitis:** la tendinitis es la inflamación de un tendón⁵ que puede ser causado a que esté repetidamente en tensión, doblado o en contacto con una superficie dura. Suele ocurrir en los hombros, codos, rodillas, talones y muñecas.
- **Tenosinovitis:** producción excesiva de líquido sinovial, hinchándose y produciendo dolor. Se originan por flexiones y/o extensiones extremas de la muñeca.
- **Síndrome del Túnel Carpiano:** se origina por la compresión del nervio de la muñeca, y por tanto la reducción del túnel. Los síntomas son dolor, entumecimiento, hormigueo y adormecimiento en la mano.
- **Síndrome Cervical por Tensión:** se origina por tensiones repetidas en la zona del cuello. Aparece al realizar trabajos por encima del nivel de la cabeza, o cuando el cuello se mantiene en flexión.
- **Dedo en Gatillo:** se origina por flexión repetida del dedo, o por mantener doblada la falange distal⁶ del dedo mientras permanecen rectas las falanges proximales⁷.

⁴ El decibelio (dB) es una unidad que se utiliza para medir la intensidad del sonido, unidad que recibe su nombre por Graham Bell.

⁵ Cuerdas resistentes de tejido que unen los músculos con los huesos y ayudan a los músculos a mover los huesos.

Capítulo I. Fundamentación Teórica

- **Bursitis:** inflamación o irritación de una “bursa⁸” debido a la realización de movimientos repetitivos.
- **Codo de tenista o Epicondilitis:** los tendones se irritan produciendo dolor a lo largo del brazo. Se debe a la realización de movimiento de extensión forzada de muñeca.
- **Fatiga:** la fatiga es causada por la falta de preparación física, y el poco fortalecimiento de los músculos. Puede causar tendencia a la depresión, ansiedad injustificada, pérdida de la iniciativa y falta de fuerza. Se puede manifestar de varias formas: disminución de la agudeza visual⁹, aumento de frecuencia respiratoria y cardiaca, disminución de la fuerza muscular, dolores, irritabilidad, agresividad y sueño. También provoca causas psíquicas como preocupaciones, conflictos, problemas para adaptarse al trabajo, incapacidad, insatisfacción laboral y falta de motivación.

1.2.4. Consecuencias de las enfermedades profesionales informáticas

El desconocimiento de estas enfermedades arroja como consecuencias problemas de salud y malestares convirtiéndose en graves lesiones musculares. Existen además personas que llevan un estilo de vida sedentario sin saber las consecuencias que pueden ocasionar.

Inactividad física y estilo de vida sedentario

Según Christina Pérez en (Pérez, 2015) el sedentarismo es la actitud que asume la persona que generalmente lleva un estilo de vida carente de agitación o movilidad. Las personas sedentarias mantienen una baja actividad física, de forma que no tienden a practicar ejercicio físico y solo se limitan a moverse lo estrictamente necesario. El sedentarismo es una de las principales causas del sobrepeso y la obesidad, una gran diversidad de estudios ha demostrado que la mayoría de las personas tienden a seguir un estilo de vida poco saludable. Tienden a tener dolores articulares y contracturas y estar propensos a determinadas enfermedades, además, tienden a tener un estado de ánimo bajo.

Lesiones músculo esqueléticas

⁶ Son los huesos localizados en los extremos de los dedos, manos y pies.

⁷ Son los huesos localizados en la base de los dedos de manos y pies.

⁸ Pequeñas bolsas situadas entre el hueso, los músculos y la piel.

⁹ La agudeza visual es la capacidad del sistema de visión para percibir, detectar o identificar objetos especiales con unas condiciones de iluminación buenas.

Capítulo I. Fundamentación Teórica

Las lesiones músculo esqueléticas son lesiones que afectan a los músculos, tendones, huesos y ligamentos, la mayoría de estas lesiones aparecen como resultado de traumatismo pequeños y repetitivos (ERGODEP, 2006). Aparecen de forma lenta y de carácter inofensivo, en apariencia, suele ignorar el síntoma hasta que se hace crónico y aparece el daño permanente. Se localiza fundamentalmente en el tejido conectivo, sobre todo en tendones y sus vainas. Ocasiona daño o irritación localmente de los nervios. Impide el flujo sanguíneo a través de venas y arterias y su ubicación más frecuente es en la zona de hombros, cuello y espalda.

1.2.5. Posiciones adecuadas para evitar las enfermedades profesionales

El objetivo principal de este epígrafe es ofrecer información para contribuir a la prevención de las enfermedades profesionales en los informáticos y destacar la importancia que tiene adoptar una buena postura frente al ordenador. Por tal motivo se explica a continuación detalladamente cada uno de los pasos para utilizar correctamente la computadora. Estos conceptos se identificaron con la ayuda del libro de Ergonomía en (Ergonomía, 2005), que explica de forma detallada las posturas que se deben adoptar.

Enfermedades tecnológicas o patológicas informática

Estas enfermedades son consecuencia del trabajo constante frente al computador y causan molestias cervicales, tendinitis, el síndrome del túnel carpiano, afecciones en la vista y estrés. Para evitar estas molestias es recomendable utilizar el monitor con colores claros y mates para evitar reflejos. Los caracteres e íconos definidos con un nivel de contraste a gusto con el fondo. Es recomendable descansar la vista 5 minutos cada hora viendo escenas lejanas.

Colocación del Monitor

Es importante ordenar los elementos de trabajo de forma que las tareas que se realicen sean llevadas a cabo de la manera más cómoda. La pantalla debe estar de frente y situada entre 50-55 cm de los ojos. Es necesario evitar las posturas forzadas, por ejemplo: poner la pantalla en la esquina de la mesa y tener que virar el cuello. La pantalla tiene que estar perpendicular a las ventanas y a todas las fuentes de luz generadas en el puesto de trabajo. El borde superior de la carcasa del monitor debe quedar a la altura de la mitad de los ojos. Esta altura debe obtenerse después de que la persona se siente correctamente en su puesto de trabajo.

Capítulo I. Fundamentación Teórica

Colocación del teclado

El teclado se debe colocar de tal forma que no quede justo al borde de la mesa, entre uno y otro debe quedar como mínimo 10 cm para apoyar la muñeca, evitando así posibles lesiones por movimientos repetidos. Es necesario tener presente que el teclado no puede quedar demasiado alto, la posición correcta se consigue cuando el antebrazo, la muñeca y la mano forman una línea recta. Su inclinación debe estar comprendida entre 0 y 25 grados. Su grosor debe ser menor o igual a 3 cm, contados desde la mesa de apoyo hasta la parte superior de la tercera fila de teclas.

La postura del sentado

La postura correcta para trabajar delante del ordenador es aquella en que la parte superior del cuerpo y la inferior están formando un ángulo recto de 90° con la espalda completamente apoyada al respaldo de la silla. Es necesario ajustar la altura del asiento, de manera que los codos queden a la altura de la superficie en la que se va a trabajar. Los pies deben estar bien apoyados al suelo y la espalda debe permanecer en contacto con el respaldo del asiento. Se debe inclinar el asiento a la mesa de trabajo de forma tal que no se tenga que inclinar el tronco y los antebrazos y así estos puedan tener espacio suficiente para apoyarse.

Colocación y uso del ratón

Se debe situar justo al lado del teclado, dejando los cables libres para manejarlo con comodidad. La mano, la muñeca y el antebrazo deben estar alineados. Se debe llevar el ratón a los lados, hacia arriba y abajo, realizando movimientos suaves de la mano y los dedos. Se deben evitar realizar gestos bruscos y flexiones sobre los dedos del ratón que puedan dañar las articulaciones y los tendones. El ratón se debe coger de tal forma que la mano descansa totalmente sobre él, y la muñeca y el antebrazo estén en contacto con la mesa.

Mesa de Trabajo

El largo de la mesa debe ser de 120-180 cm y el ancho debe ser de 80 cm. El alto de la mesa debe estar entre 67-77 cm, ya que el espacio que está debajo de la mesa debe ser suficiente para alojar las piernas, de forma que pueda cambiar la postura. El espacio debajo del tablero debe ser suficiente como para situar las piernas con holgura, sin que sufran ningún tipo de presión.

Capítulo I. Fundamentación Teórica

Resolución de la Pantalla

La pantalla deberá ser estable, sin destellos u otras formas de inestabilidad, debe ser orientable a voluntad, con facilidad para adaptarse a las necesidades del usuario. Es recomendable usar las pantallas de la siguiente forma: tamaño diagonal de 35 cm, resolución de 640 x 480, frecuencia de imagen 60 Hz. Las imágenes que se muestran en pantalla no deben ser pequeñas y así evita fijar la vista. Cuando esté trabajando frente al ordenador, evite acercarse mucho a la pantalla, esto podría dañar su vista. Si lleva más de 45 minutos trabajando frente al ordenador, tómese un descanso de 15 minutos viendo escenas lejanas.

Silla de trabajo

La altura de la silla debe ser ajustable, el respaldo debe tener una suave prominencia¹⁰ que permita el apoyo lumbar (para la parte baja de la espalda) y ayude a mantener una postura correcta. Debe ser ajustable en altura y también en inclinación para facilitar la relajación ocasional de la espalda. Es muy importante apoyar la muñeca sobre el borde de la mesa, que el brazo se mantenga pegado al tronco y el brazo y el antebrazo formen un ángulo de 90°. Además, el muslo y la pierna también deben formar un ángulo de 90° al apoyar los pies en el suelo.

1.2.6. El ejercicio y las enfermedades profesionales

Teniendo en cuenta la importancia de la actividad física y las enfermedades profesionales mencionadas anteriormente, es necesario mencionar los beneficios del ejercicio físico para la salud de las personas. En los anexos se mostrará una guía de ejercicios para evitar estas enfermedades (**Ver Anexo 2**).

Importancia del ejercicio físico

La práctica de ejercicio físico es fundamental para la salud, el ejercicio practicado especialmente a una intensidad adecuada, posee una serie de beneficios que han sido elevados por gran cantidad de estudios e investigaciones a lo largo de los años. El ejercicio beneficia el sistema cardio-respiratorio y el aparato locomotor. Mejora los mecanismos del sueño y aumenta el gasto energético, por tanto, el consumo de

¹⁰ Altura relativa.

Capítulo I. Fundamentación Teórica

grasa. Disminuye las cifras de tensión arterial y aumenta las cifras del colesterol HDL¹¹ (colesterol bueno). Es efectivo contra los diferentes dolores musculares y es un efecto anti-estrés eliminando la ansiedad y el estrés en sí. Ayuda en el metabolismo del calcio.

1.3. Aplicaciones informáticas existentes para prevenir las enfermedades profesionales

Las bibliografías consultadas evidencian que existen aplicaciones informáticas relacionadas con las enfermedades profesionales. Estas aplicaciones brindan información de dichas enfermedades con el objetivo de apoyar a su contribución y prevención.

1.3.1. Software Educativo: “Mente Sana en Cuerpo Sano”

Este software es una multimedia interactiva que le permite al usuario conocer el significado de enfermedades profesionales para evitar las afecciones que traen para la salud. Está desarrollada en HAEduc¹² la cual cuenta con un compilador de Sora Script basado en el lenguaje Visual Basic. Además, cuenta con varios juegos para que el usuario tenga interacción con la multimedia y a la vez obtenga los conocimientos necesarios para proteger su salud. También emplea un test de conocimiento para que el usuario pueda verificar el grado de conocimiento que tiene sobre las enfermedades profesionales.

La multimedia brinda distintas opciones como galerías, curiosidades, videos, ejercicios y toda la información sobre las enfermedades profesionales relacionadas con el síndrome del ordenador, síndrome del túnel carpiano, osteo-musculares de hombro y codo, dolor cervical y tendinitis (Mora, 2011).

Se construye con las bases de la metodología SXP (híbrido cubano de metodologías ágiles). Para modelar los artefactos generados utilizaron el lenguaje UML, con la herramienta Visual Paradigm. Es una multimedia que contiene 5 requerimientos funcionales: Autenticar Administrador, Administrar Juegos, Jugar, Mostrar y Controlar Audio. Además, utiliza la arquitectura N-Capas con el patrón arquitectónico top-down.

¹¹ Son las lipoproteínas (moléculas compuestas de proteínas y de grasa) más pequeñas y más densas, están compuestas de una alta proporción de proteínas.

¹² Herramienta de autor para la creación de aplicaciones multimedia con la posibilidad de generar dichas aplicaciones para los sistemas operativos Linux y Windows

Capítulo I. Fundamentación Teórica

1.3.2. Aplicación informática para la evaluación de los factores de riesgos laborales en el sector empresarial cubano

Este software se desarrolla para garantizar que el proceso de identificación y evaluación de los riesgos laborales sean más eficientes. Facilita la toma de decisiones oportunas y la disminución de la probabilidad de ocurrencia de accidentes en el sector empresarial cubano. Tiene como objetivo contribuir a la eliminación de las enfermedades profesionales, la disminución de los indicadores de accidentes y obtener un nivel de salud adecuado. Es una aplicación de escritorio desarrollada en java que evalúa los riesgos a nivel de empresa para identificar los factores de riesgos que existen y a nivel de los puestos de trabajo para tomar las medidas correspondientes y evitar las enfermedades profesionales (Sánchez, y otros, 2010). El desarrollo del trabajo está guiado por la metodología RUP, que se compone por las fases de Inicio, Elaboración, Construcción, y Transición.

Contiene requisitos funcionales como: Identificar riesgo a nivel de empresa, Evaluar los riesgos identificados en la empresa, Identificar riesgos a nivel de puesto de trabajo, Crear lista de riesgos, Evaluar riesgos a puestos de trabajo y Buscar resultados en la Base de Datos según criterios de búsqueda. Utiliza la arquitectura en tres capas la de presentación, negocio y datos, para separar la lógica del negocio de la lógica de diseño.

1.3.3. Pomodoro

Pomodoro es una técnica para administrar el tiempo y aumentar la productividad. Beneficia a las personas a realizar el máximo posible de tareas durante el día sin perder el tiempo procrastinado¹³ y al mismo tiempo tomando varios descansos para despejar la mente y relajarse. Esta es una actividad muy importante para evitar las enfermedades profesionales.

¿Cómo funciona?, durante 25 minutos las personas deberán trabajar sin distracciones de ningún tipo. Cuando el pomodoro finalice, deberá tomar un descanso de 5 minutos, al terminar los 5 minutos deberán incorporarse nuevamente al trabajo que estaban realizando. Después que pasen 4 pomodoros (100

¹³ Es la acción o hábito de retrasar actividades o situaciones que deben atenderse, sustituyéndolas por otras situaciones más irrelevantes. Es un problema de autorregulación y de organización del tiempo.

Capítulo I. Fundamentación Teórica

minutos), tendrá un descanso de 15 minutos donde nuevamente deben levantarse de la silla, caminar y distraer la vista, para que se relajen y empiecen a trabajar con nuevas ideas. Esta aplicación está desarrollada para los sistemas operativos Linux, Mac, Windows y Android (GIZMODO, 2015).

Tabla 1: Comparando las aplicaciones existentes.

Herramientas Característica	Pomodoro	Mente Sana en Cuerpo sano	Evaluación de los factores de riesgos laborales en el sector empresarial cubano.
Utilidad	Técnica de administra el tiempo	Multimedia educativa	Sistema de gestión de riesgos
Software Libre	Si	Si	Si
Interactuar con el usuario	Si	Si	Si
Aplicaciones de escritorio	Si	No	Si
Mostrar notificaciones	Si (1 solo tipo)	No	No
Informar al usuario	No	Si	No
Guías de ejercicios	No	No	No

A partir del análisis realizado sobre las aplicaciones estudiadas y teniendo en cuenta las características que se resumen en la tabla 1, se pudo apreciar qué, a pesar de las posibilidades que brindan en cuanto a la intención de la prevención de las enfermedades profesionales, fueron diseñadas en su generalidad según los objetivos en donde son aplicados.

En el caso de la aplicación “Evaluación de los factores de riesgos laborales en el sector empresarial cubano” se puede señalar que, aunque cumple con ser una aplicación de escritorio desarrollada con tecnologías de código abierto y que permite la interacción con el usuario, el objetivo con la que fue creada no se adecúa a las necesidades de la aplicación que se desea crear. Por lo que queda descartada de la comparación que se realiza.

Las dos aplicaciones restantes “Pomodoro y Mente Sana en cuerpo sano” también son aplicaciones desarrolladas sobre tecnologías de código abierto y permiten la interacción usuario-ordenador. Sin

Capítulo I. Fundamentación Teórica

embargo, solo una de ellas es una aplicación de escritorio y se limita a notificar al usuario de las posiciones adecuadas que debe adoptar para trabajar frente al ordenador. Las características mencionadas anteriormente son vitales para la propuesta de solución por las ventajas que proporciona en un entorno sin conexiones de red y la posibilidad de crear una conciencia sobre la importancia que requiere este tema con respecto a la salud.

Aunque la necesidad de esta investigación está encaminada a que la propuesta de solución pueda generar una variedad en cuanto al tipo de notificación que le debe mostrar al usuario. Se descarta la aplicación pomodoro ya que tampoco le brinda información al usuario de las enfermedades profesionales ni las lesiones que provocan, ya que está destinada únicamente a administrar el tiempo. Además, no proporciona una guía de ejercicio que ayude a proteger su salud.

Por todas las razones planteadas se pudo evidenciar que, las aplicaciones analizadas no son capaces de cumplir con la necesidad de obtener una aplicación de escritorio desarrollada sobre software libre que permita: informar al usuario sobre los temas relacionados con las enfermedades profesionales, brindarle diferentes tipos de notificaciones y guías de ejercicios adaptadas al tipo de enfermedad que desean prevenir los informáticos de la UCI en un primer momento.

1.4. Tendencias actuales a considerar en el desarrollo

Para el desarrollo de la aplicación se realizó un estudio sobre las principales herramientas a utilizar, así como de la metodología a ser empleada durante su elaboración. Con el análisis de las tendencias actuales y las novedades existentes en el campo de desarrollo, se llega a las siguientes propuestas sobre la metodología, los lenguajes de desarrollo y otras herramientas que serán empleadas.

“El software libre, es un movimiento tecnológico que ha revolucionado la sociedad. Presenta características especiales que han permitido la experimentación de nuevas formas de desarrollo y mantenimiento del programa, nuevos modelos económicos, y nuevas normas legales. Es un asunto de libertad, no de precio. Para entender el concepto se debe pensar en libre como en libertad de expresión.”
(DerechoDeInternet, 2013).

Capítulo I. Fundamentación Teórica

1.5. Metodología de desarrollo

Las metodologías de desarrollo posibilitan definir un conjunto de reglas a seguir por parte de los desarrolladores de software para lograr un proyecto con la más alta calidad. Estas se encierran en dos grupos, las tradicionales o pesadas y las ágiles que se explican a continuación.

Metodología tradicional

Pressman en (Pressman, 2005) hace referencia a las metodologías tradicionales como un “...*enfoque prescriptivo, denominado en algunas bibliografías como tradicional o pesado, busca la estructura, orden y consistencia del proyecto de desarrollo de software en cuestión. Se les llama prescriptivos porque prescriben un conjunto de elementos del proceso (acciones, tareas, productos de trabajo, mecanismos de control y aseguramiento de la calidad). Además, definen la forma en que los elementos del proceso mencionados anteriormente deben relacionarse entre sí*”.

Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías pesadas no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar.

Metodología Ágil

Las metodologías ágiles como menciona Mairelys Boeras en (Boeras, 2012) se centran en los miembros del equipo y su interacción. En la entrega rápida de versiones de software funcional, en la colaboración constante del cliente y la facilidad para manejar los cambios.

1.5.1. Selección de la metodología para el desarrollo de la aplicación

Existen varios métodos para elegir el enfoque a utilizar, en este caso se estará utilizando el método Bohem y Turner (método de la estrella) que caracteriza el proyecto de software a partir de 5 criterios. Además, estima cuan ágil o prescriptivo debe ser el enfoque a utilizar, siendo estos: tamaño del equipo, criticidad del producto, dinamismo de los cambios, cultura del equipo y personal con que se cuenta. Para

Capítulo I. Fundamentación Teórica

la selección del valor se debe tener en cuenta el comportamiento de estos criterios en el proyecto y la selección del enfoque se obtiene en correspondencia de los puntos o valores asignados a cada criterio.

Cuando los puntos ubicados en los ejes se encuentran más cerca del centro, el enfoque más adecuado a usar es el ágil y en caso contrario, se recomienda un enfoque tradicional (Boeras, 2012). En la figura 1 se muestra la representación de los criterios analizados en el método de la estrella.

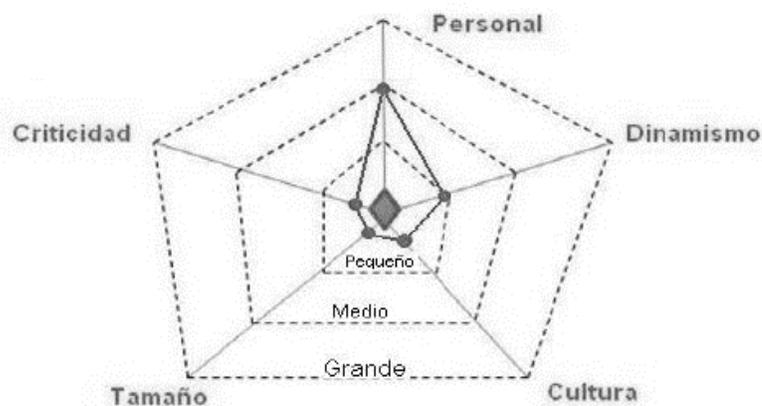


Ilustración 1: Aplicación del método de la estrella.

Fuente: <http://publicaciones.uci.cu/index.php/SC/article/view/891/546>.

Al aplicar el método de la estrella se obtienen los siguientes resultados:

Tamaño: el trabajo de desarrollo está conformado por una persona, por lo que se puede clasificar como pequeño.

Criticidad: el valor se pondera como pequeño, debido a que los efectos por errores del producto no provocan pérdidas de vida, social ni monetario para la universidad.

Dinamismo: en caso de que se necesite realizar algún cambio en los requisitos se realizará de forma adecuada. Se adoptarán mecanismos que faciliten la asimilación y adaptación rápida a dichos cambios.

Personal: el valor para este criterio se considera medio ya que el programador no es un experto, pero tiene los conocimientos básicos para realizar un buen desarrollo de la aplicación.

Capítulo I. Fundamentación Teórica

Cultura: se realiza un estudio del estado del arte para obtener toda la información en orden y tomar decisiones adecuadas que son consultadas con el cliente de la aplicación.

Como es un proyecto de desarrollo pequeño, los valores de asignación de cada arista de la estrella poseen un dominio (grande, medio, pequeño) y no un valor numérico específico. Al analizar todos los elementos que define el método empleado, se evidencia que es conveniente aplicar un enfoque ágil, ya que los puntos definidos están bien cercanos al centro de la estrella.

De las metodologías que utilizan este enfoque ágil el proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio utilizando técnicas ágiles. El proceso unificado se refiere al desarrollo iterativo incremental que sigue el producto después de haber realizado una funcionalidad ya probada (Ambler, 2015).

La UCI realizó una adaptación de la metodología AUP de forma tal que se ajuste al ciclo de vida definido para la actividad productiva de la universidad. De las 4 fases que propone, AUP-UCI mantiene la fase de inicio, pero unifica las otras tres restantes fases en una sola que se llama Ejecución y se agrega la fase de Cierre, quedando de la siguiente forma:

Inicio: se llevan a cabo las actividades relacionadas con la planeación del proyecto, se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realiza estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.

Ejecución: en esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.

Cierre: se analiza tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre de proyecto.

Capítulo I. Fundamentación Teórica

Fundamentación de la selección de la metodología

Se seleccionó la metodología AUP-UCI teniendo en cuenta que brinda un proceso de desarrollo sencillo y rápido. Tiene bien definida todas sus fases y disciplinas utilizando un proceso iterativo incremental, que propone con claridad los elementos a tener en cuenta para la posterior implementación de los requisitos. Permite definir prototipos que evidencian la validez de la arquitectura para los requisitos claves de la aplicación y está enfocada al ciclo de vida que se define para la actividad productiva de la universidad.

1.6. Herramientas y Tecnologías

Las herramientas son programas, aplicaciones o simplemente instrucciones que ofrecen la posibilidad de realizar varias funcionalidades con diferentes propósitos. Por otra parte, las tecnologías son el conjunto de conocimientos técnicos, ordenados científicamente que permiten diseñar y crear bienes y servicios. A continuación, se muestran todas las herramientas y tecnologías utilizadas para el desarrollo de la aplicación.

1.6.1. Herramienta CASE y lenguaje para el modelado

La herramienta CASE (**C**omputer **A**ided **S**oftware **E**ngineering por sus siglas en inglés) se puede definir como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software (Investigación Preliminar, Análisis, Diseño, Implementación e Instalación). Se trata también del conjunto de métodos, utilidades y técnicas que facilitan el mejoramiento del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases. Se pueden ver las herramientas CASE como la unión de las herramientas automáticas de software y las metodologías de desarrollo de software formales (González, et al., 2013) (Pressman, 2015).

Visual Paradigm

Potente herramienta para el desarrollo de aplicaciones utilizando modelado UML, ideal para ingenieros de software, analistas y arquitectos de sistemas que están interesados en construir sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos (Internet, 2013). Posee una licencia gratuita y comercial. Tiene las características de ser un producto de calidad, presenta generación de código. Es fácil de instalar y actualizar y tiene compatibilidad entre ediciones.

Capítulo I. Fundamentación Teórica

UML

El Lenguaje Unificado de Modelado más conocido por todos como UML (por sus siglas en inglés Unified Modeling Language), se utiliza para el modelado del análisis y el diseño. UML proporciona la tecnología necesaria para apoyar la práctica de la ingeniería de software orientada a objetos. Es un modelo incremental que define 5 fases. La primera fase es la de inicio que abarca la comunicación con el cliente y las actividades de planeación, y destaca el desarrollo y el refinamiento de casos de uso como un modelo primario. La segunda fase es la de elaboración, que abarca la comunicación con el cliente y las actividades de modelado con un enfoque en la creación de modelos de análisis y diseño, con énfasis en las definiciones de clase y representaciones arquitectónicas. La tercera es la de construcción que refina y después traduce el modelo de diseño en componentes de software implementado. La cuarta fase es la de transición, que trasfiere el software del desarrollador al usuario final para realizar las pruebas y obtener la aceptación. La última fase es la de producción en la cual se realiza el monitoreo (Pressman, 2014).

Fundamentación de la selección del lenguaje y la herramienta de modelado

Para el modelado fue seleccionado Visual Paradigm en su versión 8.0 teniendo en cuenta las características que posee para realizar el diseño. Tiene características gráficas muy cómodas que facilitan la realización de los diagramas de modelado que siguen el estándar UML. También posee cualidades como la generación automática de diagramas, permitiendo la agilidad en el trabajo del analista. Además, utiliza el lenguaje de modelado UML 2.0 que es común para todo tipo de desarrollo. Como lenguaje permite visualizar, construir y documentar los artefactos de un proyecto de software.

1.6.2. Lenguaje de programación

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente. Es un elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis.

Java

Capítulo I. Fundamentación Teórica

Java es un lenguaje orientado a objetos, es un lenguaje compilado, generando ficheros de clases compilados. Estas clases compiladas son en realidad interpretadas por la máquina virtual java. Siendo esta la que mantiene el control sobre las clases que se estén ejecutando. Es un lenguaje multiplataforma, el mismo código java que funciona en un sistema operativo funcionará en cualquier otro sistema operativo que tenga instalada la máquina virtual java. La máquina virtual al ejecutar el código java realiza comprobaciones de seguridad, además el propio lenguaje carece de características inseguras, como por ejemplo los punteros. Gracias al API de java se puede ampliar el lenguaje para que sea capaz de comunicarse con equipos mediante red, acceder a bases de datos, crear páginas HTML dinámicas y crear aplicaciones visuales al estilo Windows (EXES, 2008).

C++

C++ es un lenguaje de programación diseñado a mediados de los años 80 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitieran la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, C++ es un lenguaje híbrido. Posteriormente se añadieron facilidades de programación genérica, que se sumó a los otros dos paradigmas que ya estaban admitidos (programación estructurada y la programación orientada a objetos). Por esto se suele decir que es un lenguaje de programación multiparadigma (Atom, 2013). Es un lenguaje de programación orientado a objeto, didáctico y potente en lo que se refiere a creación de sistemas complejos. Es un lenguaje robusto que actualmente puede compilar y ejecutar código de C, ya que viene con librerías para realizar esta labor.

Visual Basic

Es un lenguaje que le permite al programador desarrollar aplicaciones complejas en poco tiempo comparado con otras aplicaciones como C++ y Java. Es un lenguaje de programación visual llamado como 4ta generación. Permite realizar tareas sin escribir código, simplemente con operaciones gráficas realizadas con el mouse sobre la pantalla. Es un programa basado en objetos, aunque no orientado a ellos como C++ y Java. Este lenguaje utiliza objetos con propiedades y métodos, aunque carece de herencia y polimorfismo.

HTML

Capítulo I. Fundamentación Teórica

HTML es un lenguaje que se utiliza para la creación de páginas Web. Es utilizado para describir la estructura y el contenido en forma de texto. Está compuesto por una serie de comandos que se interpretan por el visualizador o programa que es usado para navegar. Emplea un conjunto de elementos predefinidos que permiten identificar los distintos tipos de elementos. Estos elementos contienen una o más etiquetas que contienen o expresan el contenido. Estas etiquetas suelen ir encapsuladas entre los símbolos <>, y las etiquetas de cierre (que indican el final de un determinado contenido) están precedidas por una barra / (developer, 2016).

Fundamentación de la selección del lenguaje de programación

De los lenguajes de desarrollo analizados que permiten la creación de aplicaciones de escritorio con calidad, se selecciona C++ por utilizar el paradigma de programación orientada a objetos. Es capaz de reunir necesidades de trabajo a bajo nivel con abstracciones de alto nivel para la productividad y permitirle al programador un mayor control sobre lo que está haciendo consiguiendo así una mayor rapidez. También se selecciona el lenguaje HTML por lo sencillo que es trabajar con él y permitir enmarcar la información que contiene la aplicación que será visible para el usuario.

1.6.3. Framework de desarrollo

La palabra inglesa “*framework*” define como plataforma, entorno o marco de trabajo. Es una estructura conceptual y tecnológica de soporte, en lo cual el desarrollador de software puede reutilizar este código. Puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto (Renguifo, y otros, 2011).

Qt

Qt 5.4 es un *framework* multiplataforma que se utiliza para desarrollar aplicaciones con interfaz gráfica de usuario. Es desarrollado como software libre y de código abierto y utiliza el lenguaje C++. Es un *framework* orientado a entornos embebidos y móviles. Proporciona flexibilidad y productividad para el desarrollador. Brinda rendimiento, funcionalidad y facilidad de implementación en todo tipo de proyectos de aplicaciones de escritorio a móviles. Un punto especial para este *framework* ya que Digia promete la mayor integración posible con Android e iOS. Integra además un nuevo código modular (núcleos para el desarrollo de apps y servicios y complementos que pueden ser añadidos solo cuando se necesita) que

Capítulo I. Fundamentación Teórica

mejora la capacidad multiplataforma y mejora el tamaño final de las aplicaciones creadas (SOFTPEI, 2013)

1.6.4. Entorno de desarrollo (IDE)

Un entorno de desarrollo interactivo o en inglés Integrated Development Environment (IDE) es una aplicación informática que proporciona servicios integrales para facilitar al desarrollador o programador el desarrollo del software. Normalmente un IDE consiste de un editor de código fuente, herramientas de construcción automáticas y un depurador¹⁴ (Rischpater, 2014).

Qt Creator

Qt Creator 3.3.1 es un IDE que proporciona las herramientas necesarias para diseñar y desarrollar aplicaciones con el marco de aplicaciones Qt. Permite que un equipo de desarrollo comparta un proyecto a través de diferentes plataformas de desarrollo (Microsoft Windows, Mac Os X y Linux) con una herramienta común para desarrollo y depurado. El objetivo principal de Qt Creator según Daniel Martín en (Consuegra, 2012) es satisfacer las necesidades de desarrollo de los desarrolladores de Qt que buscan simplicidad, facilidad de uso, productividad, extensibilidad y apertura.

Qt puede desarrollar aplicaciones con el avanzado editor de código C++ que proporciona nuevas características de gran alcance para completar fragmentos de código y ver la jerarquía de símbolos de un archivo. Crea, ejecuta y despliega proyectos de Qt que se dirigen a múltiples plataformas de escritorio y móviles. Depura con los depuradores GNU y el CDB mediante una interfaz gráfica de usuario con mayor conocimiento de la estructura de clases Qt. Utiliza las herramientas de análisis de código para comprobar si existen problemas de gestión de memoria en las aplicaciones. Implementa aplicaciones para dispositivos móviles y accede a la información con el módulo contextual sistema de ayuda Qt.

WebStorm

Es un IDE para el desarrollo de proyectos HTML, JavaScript y PHP. Contiene un editor HTML que muestra el CSS asociado a cada etiqueta de la página que se esté editando. Una de sus características es

¹⁴ Programa usado para probar y depurar (eliminar) los errores de otros programas.

Capítulo I. Fundamentación Teórica

que guarda automáticamente los cambios. Este IDE se utiliza para desarrollar los contenidos HTML que van a ser mostrados en los componentes de la aplicación.

Fundamentación del *framework* y el IDE de desarrollo

Para el proceso de desarrollo de la aplicación se escoge el *framework* Qt por la siguiente razón:

Es un *framework* desarrollado en software libre que se utiliza para desarrollar aplicaciones con interfaz gráfica. Brinda rendimiento, funcionalidad y facilidad de implementación en todo tipo de proyecto de aplicaciones móviles y de escritorio. Se selecciona además como IDE de desarrollo Qt Creator porque le brinda las herramientas necesarias para diseñar y desarrollar aplicaciones con el marco de aplicaciones Qt. Permite utilizar las herramientas de análisis de código para comprobar si existen problemas de gestión de memoria en las aplicaciones. Además, contiene un editor de texto que le permite escribir código y formato, completamiento de código.

Conclusiones Parciales

Con el estudio de las enfermedades profesionales en los informáticos se define que, deben ser tratadas la tendinitis, bursitis, síndrome del túnel carpiano, síndrome cervical por tensión y dedo en gatillo, por ser las que mayores afecciones causan en la universidad.

A partir del análisis de las aplicaciones existentes se puede evidenciar que éstas no cuentan por si solas con las funcionalidades necesarias para ayudar en la prevención de las enfermedades profesionales. Por lo tanto, no es posible aplicar ninguna de ellas como propuesta de solución.

Se puede concluir también que, con el análisis de las tecnologías y herramientas queda definido para el progreso de la aplicación un proceso de desarrollo con enfoque ágil acorde con las necesidades del cliente. Se define además utilizar UML y Visual Paradigm para el modelado, C++ como lenguaje de programación, HTML para el diseño de las interfaces y Qt Creator como *framework* de desarrollo garantizando así, contar con las libertades de software libre.

Capítulo II. Características de la aplicación PrevenPro para la prevención de las enfermedades profesionales

Introducción

En el presente capítulo se identifica y describe un diagnóstico asociado al campo de acción del trabajo de diploma, donde se describen elementos del proceso de creación de la aplicación informática para contribuir a la prevención de enfermedades profesionales en los informáticos. Durante su elaboración se presentan los conceptos claves del dominio del problema. Se realiza el levantamiento de los requisitos funcionales y no funcionales necesarios para lograr el correcto funcionamiento de la aplicación. Se construyen los artefactos correspondientes al análisis y diseño según la metodología seleccionada en el capítulo anterior. Además, se modela la estructura de la aplicación para la prevención de las enfermedades profesionales informáticas.

2.1. Características de la aplicación PrevenPro

Prevención de enfermedades Profesionales en los informáticos (**PrevenPro**) es una aplicación de escritorio que se desea implementar para prevenir las enfermedades profesionales en los informáticos de la UCI en primer momento. La aplicación cuenta con un menú donde le muestra al usuario toda la información relacionada con las enfermedades profesionales como: su definición, los problemas ambientales, las lesiones y consecuencias que pueden ocasionar, las posiciones adecuadas que debe optar para trabajar frente al ordenador, la importancia del ejercicio físico y una guía de ejercicio por enfermedades para que el usuario conozca qué ejercicio debe realizar.

Permite, además configurar las notificaciones que se le muestran al usuario, teniendo en cuenta la administración de los siguientes parámetros: el tiempo establecido entre notificaciones, activar o desactivarlas, elegir las enfermedades que se desea prevenir y según estas, el tipo de lesión que puede evitar. Además, pasado un tiempo de inactividad la aplicación detiene el contador de notificaciones y este vuelve a activarse cuando el usuario interactúa nuevamente con el ordenador. Además, brinda información de la aplicación como la versión, el nombre completo de la aplicación y los datos del

Capítulo II. Características de la aplicación PrevenPro para la prevención de las enfermedades profesionales

desarrollador. La aplicación también le brinda al usuario un plugin en la barra de tarea para realizar las funcionalidades que se realizan en el menú de contenido, pero de forma rápida.

2.2. Modelo de dominio

El Modelo de Dominio o Modelo Conceptual permite representar los conceptos más importantes de los objetos que se muestran en el dominio del problema. Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes de software. No se trata de un conjunto de diagramas que describen clases software, u objetos de software con responsabilidades, es el encargado de capturar los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema (Larman, 2003).

El modelo de dominio constituye el punto de arranque para lograr realizar un diseño acertado del producto a desarrollar, definiendo a partir del mismo un conjunto de clases conceptuales u objetos existentes del mundo real en el dominio del problema planteado.

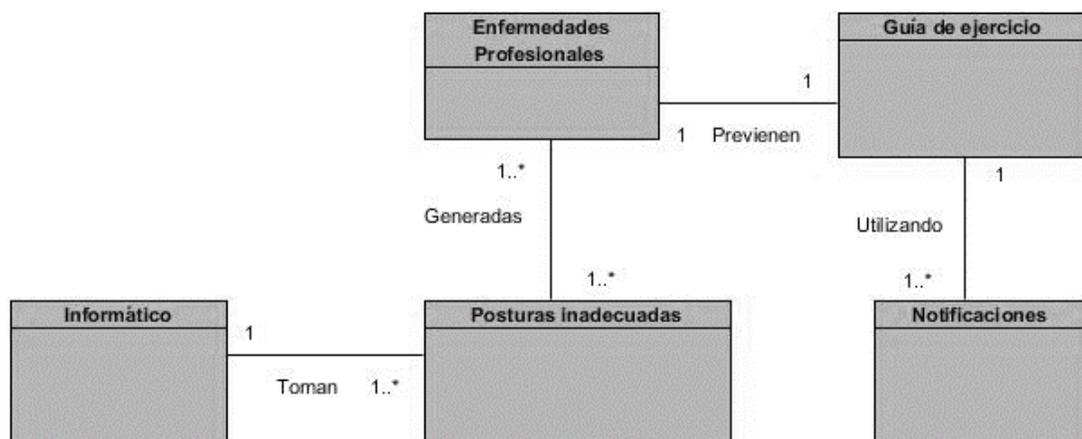


Ilustración 2: Modelo de dominio

Capítulo II. Características de la aplicación PrevenPro para la prevención de las enfermedades profesionales

Informático: es un usuario que trabaja constantemente frente a una computadora.

Notificaciones: son mostradas al usuario para prevenir las enfermedades profesionales a través de información o guías de ejercicio.

Guía de ejercicio: se utilizan para indicar al usuario a través de ejercicios físicos cómo debe prevenir las enfermedades.

Enfermedad Profesional: son aquellas generadas a través del trabajo diario frente al ordenador.

Posturas inadecuadas: posiciones que adopta el usuario de forma incorrecta en largas jornadas de trabajo.

2.3. Especificación de los requisitos de software

En la ingeniería del software, los requisitos les permiten a los ingenieros de software, entender el problema en el que trabajarán. Incluye un conjunto de tareas que conduce a comprender cuál será el impacto del software sobre el negocio. Se especifica lo que el cliente quiere y cómo van a interactuar los usuarios finales con el software (Pressman, 2013).

2.3.1. Requerimientos Funcionales

Los requisitos funcionales definen el comportamiento interno de un software, son condiciones que el sistema ha de cumplir. Estos muestran las funcionalidades que deben satisfacerse para cumplir con las especificaciones de software (Somerville, 2007).

A continuación, se muestran los requisitos funcionales de la aplicación para contribuir a la prevención de las enfermedades profesionales.

RF_1: Mostrar notificaciones.

RF_2: Acceder al menú de contenidos.

Capítulo II. Características de la aplicación PrevenPro para la prevención de las enfermedades profesionales

RF_3: Configurar notificaciones.

RF_4: Adicionar soporte para agregar plugin a la barra de tarea.

RF_5: Sugerir guía de ejercicios.

2.3.2. Requerimientos no Funcionales

Los requisitos no funcionales son propiedades que hacen al producto atractivo, usable, rápido o confiable. Se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar. Además, se conocen como un conjunto de características de calidad, que es necesario tener en cuenta al diseñar e implementar el software (Somerville, 2007).

Software

RnF_1: la PC donde se instale la aplicación debe utilizar el sistema operativo Nova en su versión 6.0.

Hardware

RnF_2: las PC deben tener como mínimo un procesador (CPU) Dual Core con una memoria RAM de 512 Mb.

Restricción de diseño e implementación

RnF_3: la aplicación deberá ejecutarse con las librerías de Qt 5.4.

RnF_4: el IDE de desarrollo debe ser Qt Creator en su versión 3.3.1.

RnF_5: el modelado de los artefactos de software deber realizarse en Visual Paradigm en su versión 8.0.

Apariencia o interfaz externa

RnF_6: empleo de imágenes y colores planos.

RnF_7: se debe detener el contador de notificaciones cuando detecte inactividad en el sistema.

Capítulo II. Características de la aplicación PrevenPro para la prevención de las enfermedades profesionales

2.4. Historias de Usuarios

Las historias de usuario (HU) describen funcionalidades que serán útiles para el usuario o comprador del sistema de un software. Las HU muestran funcionalidad que serán desarrolladas, pero no cómo se desarrollarán, además deben ser pequeñas y memorizables (Cohn, 2004).

Para encapsular los requisitos funcionales y seguir la metodología elegida AUP-UCI, se presentan a continuación las principales historias de usuario.

Tabla 2: Historia de Usuario 1

Número: HU1		Nombre del requisito: Mostrar notificaciones.	
Programador: Marielsis Alina		Iteración Asignada: 1	
Prioridad: Alta		Tiempo Estimado: 160 horas	
Riesgo en Desarrollo: -----		Tiempo Real: 145 horas	
Descripción: para el usuario poder visualizar las notificaciones que se muestran en pantalla con el objetivo de prevenir las enfermedades profesionales tiene que ejecutar el flujo descrito en la HU3.			
Observaciones: las notificaciones tienen que estar activas para que se le muestren al usuario.			
Prototipo de interfaz:			

Capítulo II. Características de la aplicación PrevenPro para la prevención de las enfermedades profesionales



Tabla 3: Historia de Usuario 2

Número: HU2		Nombre del requisito: Acceder al menú de contenidos.	
Programador: Marielsis Alina		Iteración Asignada: 1	
Prioridad: Alta		Tiempo Estimado: 40 horas	
Riesgo en Desarrollo:		Tiempo Real: 40 horas	
Descripción: cuando el usuario instale la aplicación, se le mostrará en pantalla un menú de contenido relacionado con las enfermedades profesionales para que esté consiente de los problemas que les pueden ocasionar. Los temas son: <ul style="list-style-type: none">➤ Enfermedades Profesionales.➤ Factores ambientales.➤ Lesiones.➤ Consecuencias.			

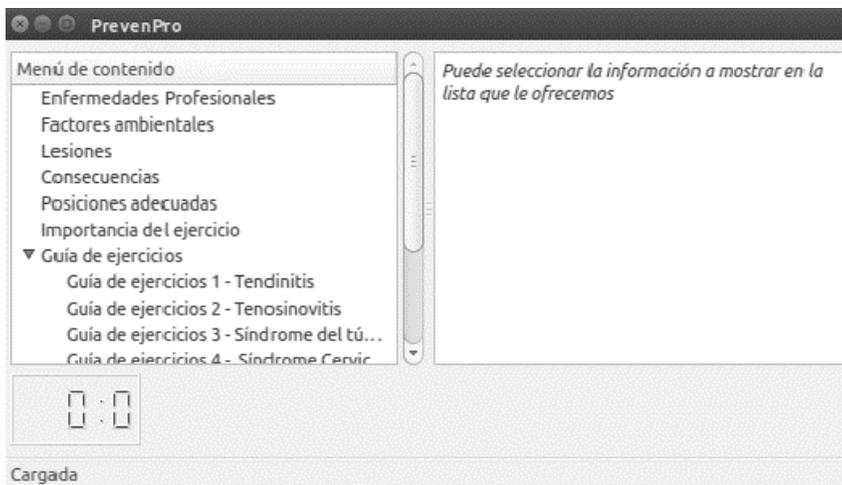
Capítulo II. Características de la aplicación PrevenPro para la prevención de las enfermedades profesionales

- Posiciones adecuadas.
- Importancia del ejercicio físico.
- Guías de ejercicios.

Dentro de las enfermedades profesionales se le explicará al usuario la definición de enfermedades profesionales. Los problemas ambientales le permitirán conocer los temas relacionados con la iluminación, ventilación y ruido que pueden poner en riesgo su salud. Podrá conocer las lesiones que pueden ocurrir por causa de las enfermedades profesionales y sus consecuencias. Además, se mostrará de forma detallada como se deben usar las herramientas informáticas y las posturas que debe adoptar en el puesto de trabajo. Por último, explicará la importancia del ejercicio físico para la salud de las personas y se mostrarán guías de ejercicios divididas por los tipos de enfermedades para mantener un cuerpo saludable.

Observaciones:

Prototipo de interfaz:



Capítulo II. Características de la aplicación PrevenPro para la prevención de las enfermedades profesionales

Tabla 4: Historia de usuario 3

Número: HU3		Nombre del requisito: Configurar notificaciones.	
Programador: Marielsis Alina		Iteración Asignada: 1	
Prioridad: Alta		Tiempo Estimado: 80 horas	
Riesgo en Desarrollo:		Tiempo Real: 72 horas	
Descripción: cuando el usuario acceda a la ventana de configuraciones tendrá las siguientes opciones: <ol style="list-style-type: none">1. Elegir el tiempo en el que se muestran las notificaciones. Se le dará la opción de elegir dentro del rango de una hora cada que tiempo desea recibir las notificaciones. El usuario no podrá seleccionar más de ese tiempo porque el sistema no le dará esa opción.2. Activar las notificaciones o desactivarlas en caso de que no quiera recibir en un momento dado notificaciones. Muchas veces las personas están trabajando y no desean ser molestadas, por tal motivo se le brindará esta opción. Aunque es recomendable que tengan en cuenta las consecuencias de no prevenir las enfermedades y las consecuencias que le causa a la salud de las personas.3. Seleccionar las enfermedades que desea prevenir: tendinitis, Tenosinovitis, síndrome del túnel carpiano, síndrome cervical por tensión, dedo en gatillo, bursitis, codo de tenista/Epicondilitis, fatiga visual.4. A partir de las enfermedades seleccionadas podrá indicar los temas relacionadas con: el monitor, el teclado, la pantalla, la posición del ratón y la posición del sentado. De los cuales desea recibir las notificaciones.			

Capítulo II. Características de la aplicación PrevenPro para la prevención de las enfermedades profesionales

Observaciones:

Prototipo de interfaz:

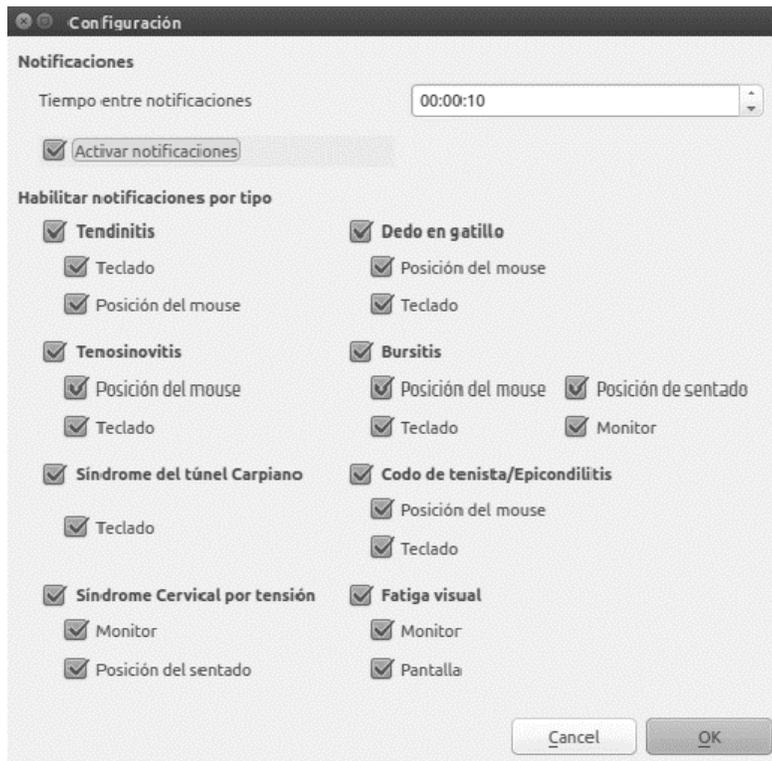


Tabla 4: Historia de usuario 5

Número: HU5	Nombre del requisito: Adicionar soporte para agregar plugin a la barra de tarea.
Programador: Marielsis Alina	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 40 horas

Capítulo II. Características de la aplicación PrevenPro para la prevención de las enfermedades profesionales

Riesgo en Desarrollo:	Tiempo Real: 40 horas
<p>Descripción: esta opción le brinda al usuario comodidad para interactuar con la aplicación sin la necesidad de ir al menú de contenidos. Cuando el sistema inicie se le activará automáticamente un plugin en la barra de tarea. Cuando el usuario de clic derecho sobre el plugin se mostrará:</p> <ul style="list-style-type: none"> ➤ Configuración: a dar clic se mostrará la ventana para configurar las notificaciones. ➤ Acerca de PrevenPro: para mostrar la información de la aplicación y el programador. ➤ Mostrar ventana principal: para acceder a la ventana principal y poder visualizar los contenidos de enfermedades profesionales. ➤ Habilitar notificaciones: para activar o desactivar las notificaciones. ➤ Salir: para salir de la aplicación. 	
Observaciones:	
Prototipo de interfaz:	
	

Tabla 5: Historia de usuario 6

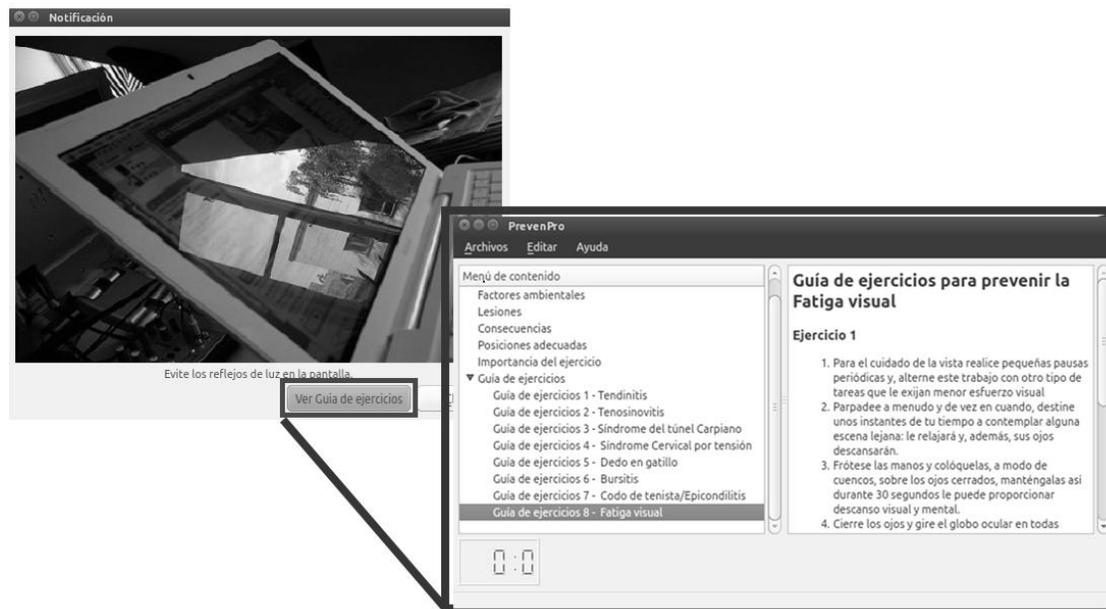
Número: HU6	Nombre del requisito: Sugerir guía de ejercicios.
Programador: Marielsis Alina	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 80 horas
Riesgo en Desarrollo:	Tiempo Real: 72 horas

Capítulo II. Características de la aplicación PrevenPro para la prevención de las enfermedades profesionales

El sistema le muestra una notificación al usuario, este le podrá sugerir una guía de ejercicios en dependencia de la enfermedad que sea elegida. De esta forma el usuario tendrá la opción de ver los ejercicios adecuados para prevenir esa enfermedad que se le está notificando.

Observaciones:

Prototipo de interfaz:



Capítulo II. Características de la aplicación PrevenPro para la prevención de las enfermedades profesionales

2.5. Arquitectura de Software

La arquitectura de software de un programa o sistema de computación es la (s) estructura (s) del sistema que comprende los componentes del software, las propiedades visibles de esos componentes y las relaciones entre ellos. Facilita la comunicación entre todas las partes interesadas en el desarrollo de un sistema basado en computadora. Destaca decisiones tempranas de diseño que tendrán un profundo impacto en todo el trabajo de ingeniería de software. Además, incluye un modelo comprensible de cómo está estructurado el sistema y cómo trabajan juntos sus componentes (Pressman, 2015).

2.5.1. Estilo Arquitectónico

Un estilo arquitectónico es una transformación impuesta al diseño de todo un sistema. El objetivo es establecer una estructura para todos los componentes del sistema. En caso de que una arquitectura existente se someta a reingeniería, la imposición de un estilo arquitectónico desembocará un cambio fundamental en la estructura del software (Pressman, 2013).

El estilo arquitectónico seleccionado fue la **Arquitectura n capas**, esta arquitectura separa los componentes de una aplicación en n niveles o capas lógicas. Permite modificar una capa sin necesidad de modificar toda la aplicación. Define el patrón en capas como una organización jerárquica, facilitando a los implementadores, particionar un problema en secuencia de pasos incrementales (Introducción a la Arquitectura de Software II, 2013).

Se definió la arquitectura en dos capas como se describen a continuación:

Capa de presentación: presenta el sistema al usuario, comunica la información y captura la información del usuario en un mínimo proceso. Esta capa reúne todos los aspectos del software que tienen que ver con las interfaces incluyendo el manejo y aspecto de las ventanas.

Capítulo II. Características de la aplicación PrevenPro para la prevención de las enfermedades profesionales

Capa de negocio: en esta capa se modela el negocio y se obtienen las solicitudes de la capa de presentación. Esta capa reúne todos los aspectos del software que apoyan los procesos que se llevan a cabo.

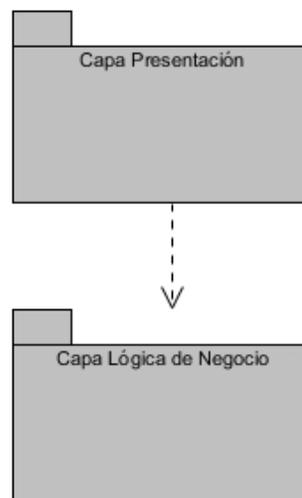


Ilustración 3: Arquitectura n-capas (2).

2.6. Patrones de diseño.

Los patrones de diseño, permiten describir fragmentos y reutilizar ideas de diseño. Craig Larman en (Larman, 2004) define los patrones de diseño como “*un par problema/solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos*”. Los patrones se codifican en un formato estructurado que describe el problema y la solución. Cada patrón aborda un problema específico y recurrente en el diseño o implementación de un sistema de software.

Capítulo II. Características de la aplicación PrevenPro para la prevención de las enfermedades profesionales

Con el objetivo de lograr un sistema robusto y reutilizable se utilizaron los **patrones Grasp**¹⁵, ya que son la base para la búsqueda de soluciones a problemas comunes con el desarrollo de la aplicación. Los patrones Grasp describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones. La siguiente selección presenta los patrones de asignación de responsabilidades.

Experto: este patrón es el principio básico de asignación de responsabilidad. Garantiza que las clases cumplan con su responsabilidad, según las clases que contengan y las funcionalidades que se deseen implementar. Por ejemplo: en la aplicación la clase UIInfoApp es la experta en conocer la información necesaria para construir la interfaz que brinda la información relacionada con el tema de la aplicación, y la información del programador.

Creador: responde a la pregunta ¿Quién debería ser responsable de crear una nueva instancia de alguna clase? Que una clase contenga objetos de otra clase. Que una clase registre instancias de objetos en otra clase. Que una clase utilice objetos de otra clase. Que una clase sea creadora de los objetos de otra clase. Por ejemplo: este patrón se evidencia en la clase ControladoraApp, que en los métodos de ella se instancian objetos de otras clases.

Alta Cohesión: este patrón tiene como objetivo asignar una responsabilidad de manera que la cohesión permanezca alta. Una clase con baja cohesión puede provocar que esta sea difícil de entender, de reutilizar, de mantener y que se vean constantemente afectada por los cambios. En la aplicación se utiliza para asignar responsabilidades de manera que la complejidad se mantenga dentro de los límites, asumiendo solamente las acciones que debe manejar, evitando un trabajo excesivo. Su utilización mejora la claridad y facilita que se entienda el diseño, simplifica el mantenimiento y las mejoras de funcionalidad,

¹⁵ General Responsibility Assignment Software Patterns (Patrones generales de software para asignar responsabilidades).

Capítulo II. Características de la aplicación PrevenPro para la prevención de las enfermedades profesionales

genera un bajo acoplamiento y soporta mayor capacidad de reutilización. En la solución se evidencia este patrón en la clase ControladoraApp debido a que esta tiene una gran interdependencia con otras clases.

Bajo Acoplamiento: este patrón se tiene en cuenta para asignar una responsabilidad de manera que el acoplamiento permanezca bajo. Además, impulsa la asignación de responsabilidades de manera que su localización no incremente el acoplamiento hasta un nivel que lleve a los resultados a un acoplamiento alto. Este patrón soporta el diseño de clase que son independientes, lo que reduce el impacto del cambio. Se dice que es de bajo acoplamiento cuando no existe acoplamiento entre clases. Si el bajo acoplamiento llega al extremo, se puede producir un diseño pobre. Por ejemplo: con este patrón se logra que la clase ONotificacion posea un grado de dependencia mínimo.

Controlador: el patrón controlador tiene como objetivo asignar la responsabilidad de recibir o manejar un mensaje de evento del sistema a una clase. Este patrón define el método para la operación del sistema. Por ejemplo: la clase este patrón se evidencia en la clase ControladoraApp que se encarga de la comunicación entre los componentes de la interfaz de la aplicación. Es la encargada de enviar la información de los objetos a la interfaz para que se muestren.

2.7. Modelo de diseño.

El modelo de diseño representa las especificaciones de las clases e interfaces de software en una aplicación. Estas muestran las definiciones de las clases de software en lugar de los conceptos del mundo real. Entre las informaciones generales se encuentran: clases, asociaciones y atributos; interfaces con sus operaciones y constantes; métodos e información acerca del tipo de atributo y dependencias (Larman, 2006).

Capítulo II. Características de la aplicación PrevenPro para la prevención de las enfermedades profesionales

2.7.1. Diagrama de clases

Los diagramas de clase del diseño describen el nombre de las clases, las etiquetas de los métodos y los atributos. Proporciona una perspectiva estática que representa el diseño estructural del sistema mostrando un conjunto de clases (Larman, 2003).

En este diagrama se muestran las clases agrupadas en el paquete Negocio y Presentación. En el paquete del negocio se encuentran las clases UIMenuContenido, ControladorApp, UINotificaciones, UIConfigNotificaciones y UIInfoApp. En el paquete Presentación se encuentran las clases OPersistencia, OGuia, OEnfermedad y ONotificacion. A continuación, se muestra un resumen de cómo quedo estructurado el diagrama de clase. El diagrama de clases completo se encuentra en los anexos.

(Ver [Anexo 3](#))

Capítulo III. Implementación y Prueba

Introducción

En este capítulo se muestra el modelo de implementación como resultado del diseño anteriormente desarrollado. Se realiza una descripción de las pruebas que se utilizarán en el desarrollo de la aplicación, con el objetivo de probar el correcto funcionamiento de la aplicación para la prevención de las enfermedades profesionales en los informáticos. Se realiza además un estudio de los estándares de codificación.

3.1. Modelo de despliegue

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware y muestra cómo los elementos y artefactos del software se trazan en esos nodos (Sparxsystems, 2016).

A continuación, se presenta el diagrama de despliegue que es muy sencillo porque el producto es una aplicación de escritorio que cargará cuando inicie la PC y en su interior incluirá todo.

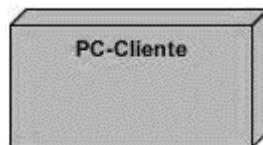


Ilustración 5: Diagrama de despliegue

3.2. Modelo de componente

La vista de implementación muestra el empaquetado físico de las partes reutilizables del sistema en unidades sustituibles, llamadas componentes. Una vista de implementación muestra los elementos físicos del sistema mediante componentes, así como sus interfaces y dependencias entre componentes. Los componentes son piezas reutilizables de alto nivel a partir de las cuales se pueden construir los sistemas. El diagrama de componentes describe la descomposición física del sistema de software en componentes, a efectos de construcción y funcionamiento. Los componentes identifican objetos físicos que hay en tiempos de ejecución, de compilación o de desarrollo y tienen identidad propia con una interfaz bien

Capítulo III. Implementación y Prueba

definida (Dpto.Electrónica, 2016). A continuación, se muestra el diagrama de componentes de la aplicación para prevenir las enfermedades profesionales.

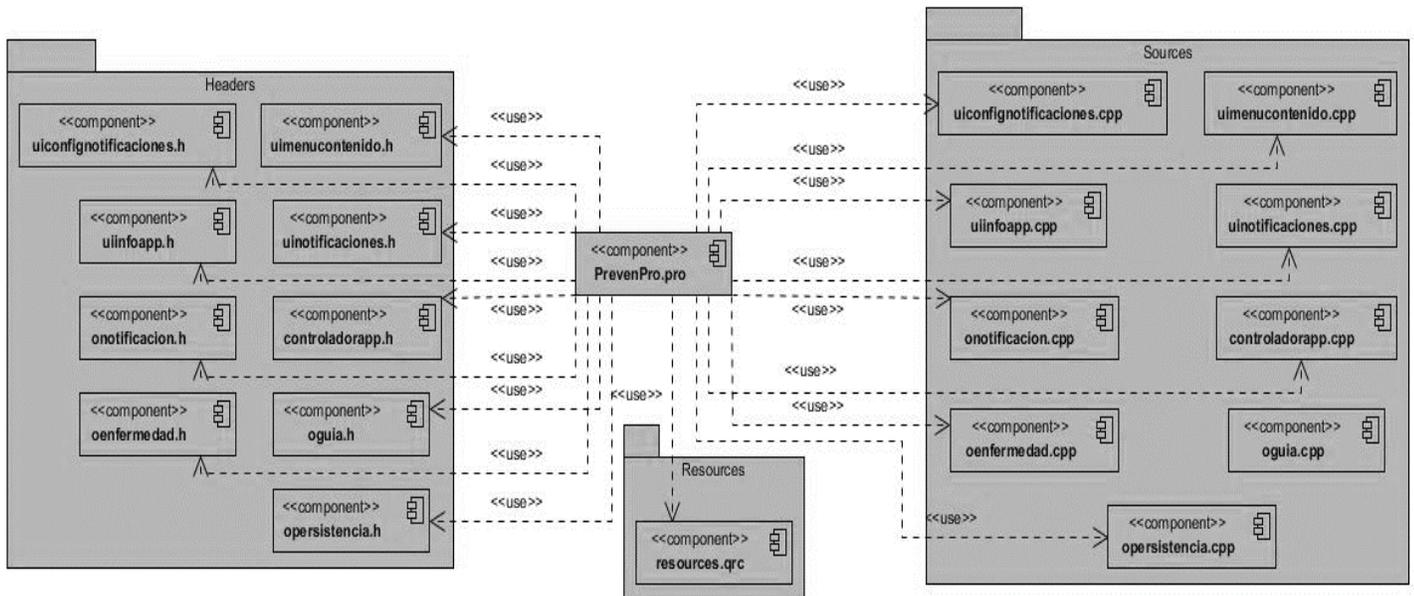


Ilustración 6: Diagrama de componentes.

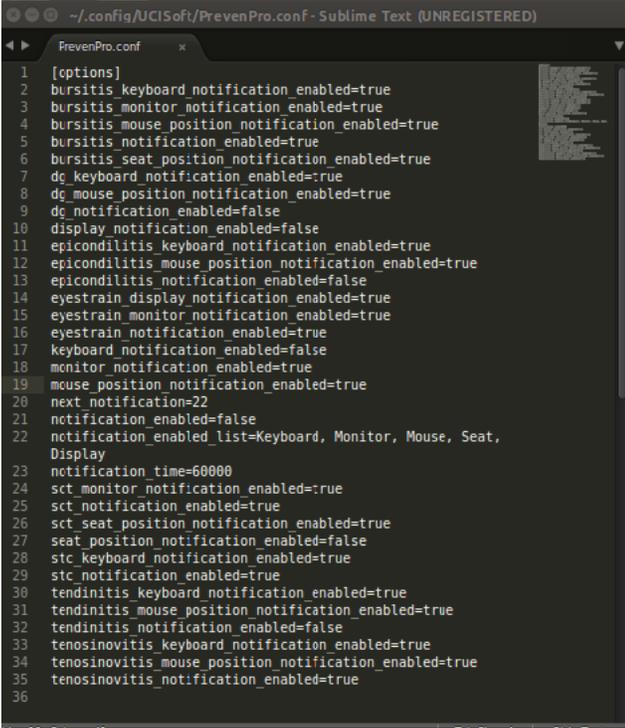
En el presente diagrama aparecen los componentes y paquetes imprescindibles que intervienen en todo el proceso. Los paquetes Headers y Source son los contenedores de las clases que intervienen en la aplicación.

3.3. Almacenamiento de datos

3.3.1. Fichero de configuración

La aplicación para almacenar los datos relacionados con las configuraciones de las notificaciones almacenará la información referente a variables de configuración y los valores establecidos por el usuario en un fichero (.conf). Esto permite cargar y guardar la información que se almacena de forma eficiente. El fichero contará con un formato propio definido por el desarrollador como se muestra en la figura.

Capítulo III. Implementación y Prueba



```
1 [options]
2 bursitis_keyboard_notification_enabled=true
3 bursitis_monitor_notification_enabled=true
4 bursitis_mouse_position_notification_enabled=true
5 bursitis_notification_enabled=true
6 bursitis_seat_position_notification_enabled=true
7 dc_keyboard_notification_enabled=true
8 dc_mouse_position_notification_enabled=true
9 dc_notification_enabled=false
10 display_notification_enabled=false
11 epicondylitis_keyboard_notification_enabled=true
12 epicondylitis_mouse_position_notification_enabled=true
13 epicondylitis_notification_enabled=false
14 eyestrain_display_notification_enabled=true
15 eyestrain_monitor_notification_enabled=true
16 eyestrain_notification_enabled=true
17 keyboard_notification_enabled=false
18 monitor_notification_enabled=true
19 mouse_position_notification_enabled=true
20 next_notification=22
21 notification_enabled=false
22 notification_enabled_list=Keyboard, Monitor, Mouse, Seat,
  Display
23 notification_time=60000
24 sct_monitor_notification_enabled=true
25 sct_notification_enabled=true
26 sct_seat_position_notification_enabled=true
27 seat_position_notification_enabled=false
28 stc_keyboard_notification_enabled=true
29 stc_notification_enabled=true
30 tendinitis_keyboard_notification_enabled=true
31 tendinitis_mouse_position_notification_enabled=true
32 tendinitis_notification_enabled=false
33 tenosinovitis_keyboard_notification_enabled=true
34 tenosinovitis_mouse_position_notification_enabled=true
35 tenosinovitis_notification_enabled=true
36
```

Ilustración 7: Estructura del fichero de configuración.

Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez, el estándar de codificación debería establecer cómo operar con la base de código existente (MICROSOFT, 2015).

Para el desarrollo de la aplicación informática para contribuir a la prevención de las enfermedades profesionales en los informáticos, se tuvo en cuenta el estándar que define el estilo Qt programming style en **qtapistyle** (Ezust, y otros, 2012).

CamelCase: es un estilo de escritura que se aplica a frases o palabras compuestas. Se caracteriza por unir las palabras entre sí, sin dejar espacio; con la peculiaridad de que la primera letra de cada término se

Capítulo III. Implementación y Prueba

encuentra con mayúscula para hacer legible el conjunto. A continuación, se muestra como queda definida la clase con el estándar CamelCase:

- UpperCamelCase: la primera letra de la nueva palabra debe estar en mayúscula, permitiendo que sea fácil de distinguir de un nombre lowerCamelCase. En la aplicación se utiliza para nombrar las clases.

```
class UIMenuContenido : public QMainWindow
{
```

Ilustración 8: Fragmento de código fuente de la clase UIMenuContenido.

- lowerCamelCase: la primera letra del nombre debe estar en minúscula. En la aplicación se utiliza este estilo para nombrar las variables y los métodos.

```
void ControladorApp::showOptions()
{
    int leftTime = notificationTimer->remainingTime();
    notificationStop();
    optionDialog = new UIConfigNotificaciones(m_options);
    optionDialog->exec();
    if(m_options->isAccepted()){
        loadOptions();
    }else{
        notificationTimer->start(leftTime);
    }
}
```

Ilustración 9: Fragmento de código del método ControladorApp().

- Número de declaraciones por línea: en la aplicación se declara cada variable en una línea distinta.

Capítulo III. Implementación y Prueba

```
bool m_EyestrainMonitorChecked;  
bool m_EyestrainMonitorCheckedTemp;  
bool m_EyestrainDisplayChecked;  
bool m_EyestrainDisplayCheckedTemp;  
QStringList m_NotificationEnabledList;  
bool m_accepted;
```

Ilustración 10: Ejemplo de declaración de variable.

- Las formas de incluir las clases se muestran a continuación:

```
#include <QMainWindow>  
#include <QTextEdit>  
#include <QSignalMapper>  
#include <QMdiArea>  
#include <QSystemTrayIcon>  
#include <QVBoxLayout>  
#include <QDialog>  
#include <QTime>  
#include <QSplitter>  
#include <QTreeView>
```

Ilustración 11: Ejemplo de sentencias de inclusión

- Las sentencias if y for de la aplicación contienen la siguiente estructura:

```
void ControladorApp::nextNotificationEnabled(QList<QString> notificationEnabledList)  
{  
    int cont = 0;  
    if(!notificationEnabledList.isEmpty()){  
        while(cont<m_NotificationList.length()){  
            m_CurrentNotification = nextNotification();  
            if(notificationEnabledList.contains(m_CurrentNotification->type())){  
                setCurrentNotification(m_CurrentNotification);  
                return;  
            }  
            cont++;  
        }  
    }else{  
        setCurrentNotification();  
    }  
}
```

Ilustración 12: Ejemplo de sentencia if.

Capítulo III. Implementación y Prueba

```
for(int i=0; i<m_SicknessList.length(); i++)
{
    QStandardItem *item = new QStandardItem(m_SicknessList.value(i)->getName());
    QStandardItem *content = new QStandardItem(m_SicknessList.value(i)->getDescription());

    if( i == 6 )
        for( int c=0; c<m_GuideList.length(); c++ ){
            item->appendRow( new QStandardItem( QString(m_GuideList.value(c)->getName() ) ) );
            content->appendRow( new QStandardItem( QString(m_GuideList.value(c)->getDescription()) ) );
        }
    m_SicknessModel->setItem(i, 0, item);
    m_GuidesModel->setItem(i, 0, content);
}
```

Ilustración 13: Ejemplo de sentencia for.

Pruebas de software

El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales, es decir, la realización de las pruebas para la detección de errores. Además, son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa (Pressman, 2010).

3.1.1. Pruebas Funcionales

Con estas pruebas se asegura el trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados. A este tipo de prueba se le denomina también pruebas de comportamiento y para realizarlas se emplea el método de caja negra, donde los probadores o analistas de pruebas no enfocan su atención en cómo se generan las respuestas del sistema, sino en el funcionamiento de la interfaz del sistema.

El objetivo de estas pruebas es demostrar que las funciones del software son operativas, que la entrada y salida se producen de forma correcta y que además se mantiene la integridad de la información.

Método de caja negra

Las pruebas de caja negra son las que se le aplica a la interfaz del software. Estas pruebas examinan los aspectos funcionales del sistema que tienen poca relación con la estructura de la lógica interna del

Capítulo III. Implementación y Prueba

software. Se prueban las respuestas lógicas del software y la colaboración entre componentes. Estas pruebas se concentran en los requisitos funcionales del software. Le permiten al ingeniero de software derivar conjuntos de condiciones de entrada que ejercitarán por completo todos los requisitos funcionales de un programa (Pressman, 2013).

Casos de Prueba

A partir de la técnica seleccionada, se realizó un caso de prueba basado en las HU, diseñando un caso de prueba por cada uno de estos.

Caso de Prueba: Mostrar notificaciones.

Descripción general: el usuario selecciona el tiempo que desea ser notificado y el tipo de enfermedad que desea tratar. También deberá seleccionar el tipo de notificación que desea recibir. Cuando estas acciones se realicen, el usuario recibirá las notificaciones para prevenir las enfermedades profesionales.

Condiciones de ejecución: el usuario debe tener la aplicación instalada y las notificaciones deben estar activadas.

Tabla 6: Caso de Prueba Mostrar notificaciones.

Nombre de la Sección	Escenario de la sección	Descripción de la funcionalidad
SC 1: Mostrar notificación	EC 1.1 Activar notificaciones	Se activan las notificaciones y el usuario comienza a recibirlas.

Caso de Prueba: Acceder al menú de contenidos.

Descripción general: el usuario accede al menú de contenidos que le permite visualizar la información relacionada con las enfermedades profesionales, en este menú encontrará información de enfermedades profesionales, factores ambientales, lesiones, consecuencias, posiciones adecuadas, la importancia del ejercicio físico y ocho guías de ejercicio para prevenir estas enfermedades.

Condiciones de ejecución: el usuario debe tener la aplicación instalada.

Capítulo III. Implementación y Prueba

Tabla 7: Caso de Prueba Acceder al menú de contenidos.

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad
SC1: Acceder al menú de contenidos.	EC 1.1 Abrir menú	Se presiona en el plugin de la barra de tarea y aparece el menú de contenidos.
	EC 1.2 Seleccionar contenido	Se selecciona el contenido que se desea visualizar y se muestra la información correspondiente.

Caso de Prueba: Seleccionar tipo de notificaciones a mostrar.

Descripción general: el usuario debe seleccionar las enfermedades que desea prevenir y el tipo de notificación que desea recibir. En este caso existen cinco tipos: Monitor, Teclado, Posición de sentado, Colocación del mouse y Pantalla.

Condiciones de ejecución: el usuario debe tener la aplicación instalada.

Tabla 8: Caso de Prueba Seleccionar tipo de notificaciones a mostrar.

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad
SC1: Seleccionar notificaciones.	EC 1.1 Seleccionar enfermedad	Se selecciona la enfermedad que el usuario desea tratar. Puede seleccionar más de una enfermedad.
	EC 1.2 Seleccionar tipo de notificación.	Se selecciona el tipo de notificación que el usuario desea recibir.

Capítulo III. Implementación y Prueba

Caso de Prueba: Configurar notificaciones.

Descripción general: el usuario tendrá la opción de elegir el tiempo en el que se muestran las notificaciones. En este caso el usuario no tendrá la opción de elegir más de 1 hora. También tendrá la opción de activar las notificaciones o desactivarlas.

Condiciones de ejecución: el usuario debe tener la aplicación instalada.

Tabla 9: Caso de Prueba Configurar notificaciones.

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad
SC 1 Elegir tiempo de notificaciones	EC 1.1 Seleccionar el menú de contenidos.	Se presiona en el plugin de la barra de tarea y aparece el menú de contenidos.
	EC1.2 Abrir las configuraciones.	Se selecciona la ventana Editar y se presiona en "Configuración". El usuario puede seleccionar Ctrl+K y aparece la ventana de configuración.
	EC1.3 Definir tiempo de las notificaciones	Configura cada que tiempo desea ser notificado para prevenir las enfermedades.
SC 2 Activar/Desactivar notificaciones	EC 2.1 Seleccionar activar notificaciones	El usuario selecciona la opción activar notificaciones que se encuentra en el menú de contenidos o simplemente da clic derecho sobre el plugin que se encuentra en la barra de tarea y

Capítulo III. Implementación y Prueba

		aparece la opción de activar las notificaciones.
--	--	--

Caso de Prueba: Adicionar soporte para agregar plugin a la barra de tarea.

Descripción general: esta opción le brinda al usuario comodidad para interactuar con la aplicación sin la necesidad de ir al menú de contenidos. Cuando el sistema inicie se le activara automáticamente un plugin en la barra de tarea. Cuando el usuario de clic derecho sobre el plugin se mostrarán las opciones Configurar, Acerca de PrevenPro, Mostrar ventana principal, Habilitar/Deshabilitar notificaciones y Salir.

Condiciones de ejecución: el usuario debe tener la aplicación instalada.

Tabla 10: Caso de Prueba Adicionar soporte para agregar plugin a la barra de tarea.

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad
SC 1 Acceso a las opciones del plugin.	EC 1.1 Seleccionar configuración	El usuario presiona clic derecho sobre el plugin y selecciona "Configuración" y se muestra la ventana de configuración de las notificaciones.
	EC 1.2 Acerca de PrevenPro	El usuario presiona clic derecho sobre el plugin y selecciona "Acerca de PrevenPro" y se muestra la información de la aplicación.
	EC 1.3 Mostrar ventana principal	El usuario presiona clic derecho sobre el plugin y selecciona "Mostrar ventana principal" y se

Capítulo III. Implementación y Prueba

		muestra el menú de contenido donde se encuentra la información de las enfermedades profesionales.
	EC 1.4 Habilitar/Deshabilitar notificaciones.	El usuario presiona clic derecho sobre el plugin y selecciona “Habilitar notificaciones” y el usuario comienza a recibir las notificaciones. Si selecciona Deshabilitar notificaciones la aplicación dejará de notificar al usuario.
	EC 1.4 Salir	El usuario presiona clic derecho sobre el plugin y selecciona “Salir” y se cierra la aplicación.

Caso de Prueba: Sugerir guía de ejercicio.

Descripción general: cuando el sistema le muestre una notificación al usuario, este le podrá sugerir una guía de ejercicio en dependencia de la enfermedad que se esté previniendo. De esta forma el usuario tendrá la opción de ver los ejercicios adecuados para prevenir esa enfermedad que se le está notificando.

Condiciones de ejecución: el usuario debe tener la aplicación instalada.

Tabla 11: Caso de Prueba Sugerir guía de ejercicio.

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad
----------------------	-------------------------	---------------------------------

Capítulo III. Implementación y Prueba

SC 1 Guía de ejercicios	EC 1.1 Sugerir guía	Se muestra una notificación. El usuario selecciona el botón “Ver Guía” y se muestra la guía de ejercicios en correspondencia con la notificación que se esté mostrando en ese momento
--------------------------------	----------------------------	---

Ejecución de los casos de prueba

La ejecución de las pruebas se realizó en tres iteraciones por cada escenario según el comportamiento de la aplicación.

3.1.2. Resultados obtenidos

Al aplicar las pruebas funcionales se alcanzaron los siguientes resultados en cada una de las iteraciones aplicadas a la herramienta:

- En la primera iteración de 15 eventos probados, 9 de ellos arrojaron resultados satisfactorios y 6 no satisfactorios. Los 6 no satisfactorios están dados por la detección de errores tales como: funcionalidades incorrectas, errores de interfaz y falta de ortografía.
- Al efectuarse la segunda iteración, se mostró que se habían corregido los errores anteriores, detectándose 13 eventos satisfactorios y 2 no satisfactorios. Manteniendo errores de interfaz y falta de ortografía.
- En la tercera iteración no se encontraron eventos no satisfactorios, demostrando que la aplicación cumple con los requerimientos funcionales expuestos anteriormente.

La siguiente ilustración muestra gráficamente el resultado obtenido en las pruebas por cada una de las iteraciones mencionadas.

Capítulo III. Implementación y Prueba

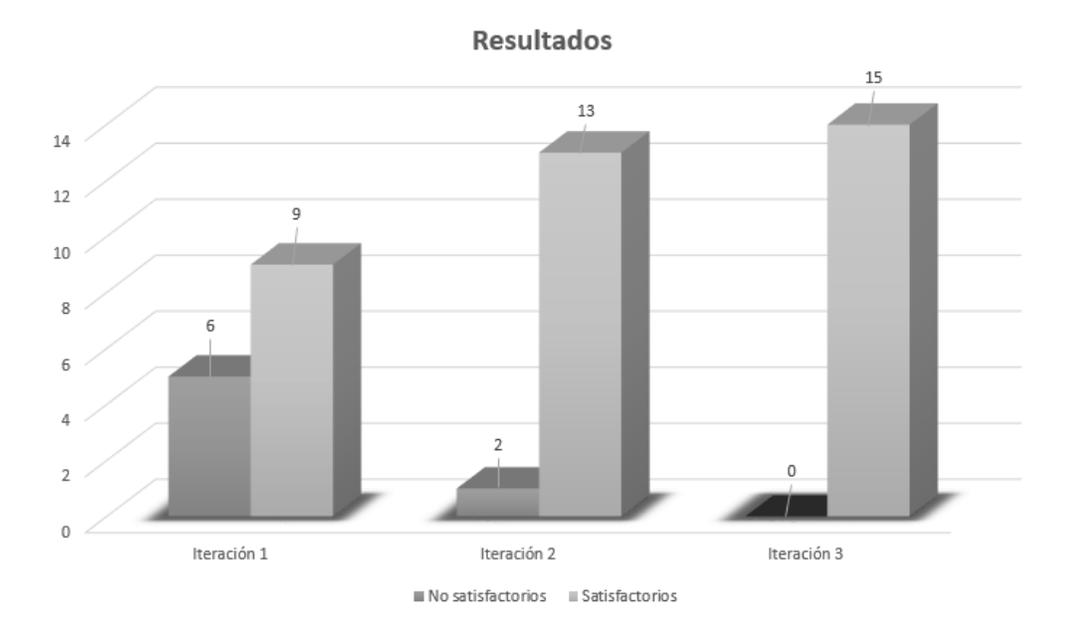


Ilustración 14: Comportamiento por iteraciones de las pruebas funcionales.

Conclusiones Parciales

La aplicación de las pruebas funcionales a través del método de caja negra, permitió detectar los errores funcionales, ortográficos y de interfaz. Al finalizar los casos de prueba se obtuvieron resultados satisfactorios logrando un correcto funcionamiento de la aplicación. Por lo que se demuestra que la aplicación PrevenPro se encuentra correctamente diseñada y se presupone será efectiva al aplicarla para prevenir las enfermedades profesionales, dando cumplimiento a los objetivos trazados en la investigación.

Conclusiones Generales

Con la realización del presente trabajo de diploma se obtuvo una propuesta que da cumplimiento al objetivo general planteado, desarrollar la aplicación informática para contribuir a la prevención de enfermedades profesionales en los informáticos. Para llegar a este resultado se concluye lo siguiente:

1. El estudio del arte de las enfermedades profesionales confirma que, los conocimientos y mecanismos existentes con relación a los problemas de salud que pueden provocar, no son suficientes para prevenir estas enfermedades. Se debe a que están concentrados en un espacio limitado y no se utilizan con la debida frecuencia por parte de los profesionales de la institución, fundamentando las bases para el desarrollo de la aplicación PrevenPro.
2. La búsqueda y análisis de sistemas semejantes demostró que las aplicaciones internacionales, nacionales y las encontradas en la UCI, no poseen las funcionalidades requeridas, evidenciándose la necesidad de crear un producto nuevo para los informáticos de la institución.
3. La aplicación cuenta con un diseño adecuado para su propósito, con todos sus componentes visuales y todas las funcionalidades en correspondencia con las características diseñadas y los requisitos trazados, de acuerdo a la revisión de la arquitectura y el funcionamiento de la aplicación.
4. Se comprobó la efectividad de la solución propuesta a partir de los resultados satisfactorios obtenidos en las pruebas funcionales y por el criterio de especialistas del área de rehabilitación que valoraron la utilidad de la aplicación para la prevención de las enfermedades profesionales.

Recomendaciones

Con el desarrollo de futuras investigaciones se recomienda perfeccionar e incorporar nuevas funcionalidades para lograr mejoras en la prevención de enfermedades profesionales dentro del objeto de estudio de la presente investigación, entre las cuales se pueden considerar:

- Extender el uso de la aplicación para prevenir enfermedades profesionales generadas por los factores ambientales, tales como: el ruido y la iluminación.
- Utilizar como referencia el código de la aplicación para adaptarlo a dispositivo móviles con sistema operativo Android.

Referencias Bibliográficas

- Acosta, Roman, Brisely, Crislia y Lucero, Edith. 2010.** Impacto de la informatica en la sociedad actual. [En línea] 13 de octubre de 2010.
- Alonso, José María. 2003.** Prevención de accidentes de trabajo y enfermedades profesionales. [En línea] 2003. <http://riberdis.cedd.net>.
- Atom. 2013.** Lenguaje de Programación C++. [En línea] 2013.
- Belloch, Consuelo.** Las tecnologías de la información y comunicación (TIC) en el aprendizaje. . [En línea] <http://www.uv.es/bellochc/pdf/pwtic2.pdf>.
- Benavides, Fernando G., Frutos, Carlos Ruiz y Gracia, Ana M. 1997.** Salud laboral, conceptos asociados y tecnicas para la prevencion de riesgos laborales. [En línea] Masson, 1997.
- Chamizo, Daniel. 2013.** Entorno de programación gráfico OPENCV. [En línea] Septiembre de 2013. http://e-archivo.uc3m.es/bitstream/handle/10016/21422/TFG_Daniel_CHAMIZO_ALBERTO.pdf?sequence=1.
- Cohn, Mike. 2004.** *User Stories Applied: for agile software development*. 2004.
- Consuegra, Daniel M. 2012.** Ingeniería Técnica Industrial: Electrónica Industrial. . *Diseño y Desarrollo de una Interfaz Gráfica de usuario para la prueba de DAQS basado en arduino mediante ROS*. [En línea] Noviembre de 2012. http://e-archivo.uc3m.es/bitstream/handle/10016/16723/PFC_Daniel_Martin_de_Consuegra_Martinez.pdf?sequence=1.
- Del Río, Gregorio. 2013.** Riesgos laborales ergonómicos: una visión jurisprudencial. [En línea] 2013.
- DerechoDeInternet. 2013.** Las esferas socioculturales del Software Libre . [En línea] 2013. <http://www.derecho-internet.org>.
- developer. 2016.** Introduccion al HTML. [En línea] 2016. https://developer.mozilla.org/es/docs/Web/Guide/HTML/Introduction_alhtml.
- developerWorks. 2011.** XML y Tecnologías Relacionadas: Introducción. [En línea] 2011.
- Dpto.Electrónica, Automática e Informática industrial. 2016.** 5UML. El modelo dinámico y de implementación. [En línea] 2016. <http://www.elai.upm.es/webantigua/spain/Asignaturas/InfoInd/apuntesAOOD/cap5UMLDinamicoImpl.pdf>.

Referencias Bibliográficas

- ERGODEP. 2006.** Prevención de riesgos ergonómicos y psicosociales en los centros de atención a personas en situación de dependencia en la comunidad valenciana. [En línea] 2006.
- Ergonomía. 2005.** *Ergonomía. Proyecto espadaleda.* . Galicia, España : s.n., 2005.
- EXES. 2008.** Área de Programación y Desarrollo. Curso de introducción a java. [En línea] 2008. http://www.mundojava.net/caracteristicas-del-lenguaje.html?Pg=java_inicial_4_1.html.
- Ezust, Alan y Paul, Ezust. 2012.** *An Introduction to Design Patterns in C++ with Qt. 2da Edición.* 2012. ISBN 978-0-13-282645-7 (hardback : alk. paper).
- Garrido, Salvador Alemany. 2009.** *Introducción a QT. Programación gráfica en C++ con QT4.* 2009.
- GIZMODO. 2015.** Cómo utilizar la técnica de productividad Pomodoro en cualquier SO. [En línea] 2015. <http://es.gizmodo.com/como-utilizar-la-tecnica-pomodoro-en-cualquier-sistema-1697435287>.
- GNUSTEP. 2016.** Sitio de GNUstep en español. . [En línea] 2016. <https://gnustep.wordpress.com/gnustep-a-fondo/%C2%BFque-es-un-framework-%C2%BFcomo-se-utiliza/>.
- González, Pascual, L., A. A. G. y Gallud, José Antonio. 2013.** *Herramientas CASE ¿Cómo incorporarlas con éxito en nuestra organización? Edición ed. 14 p.* 2013.
- Internet, Software.com.ar. Available from. 2013.** <<http://www.software.com.ar/visual-paradigm-para-uml.html>>. [En línea] 2013.
- Introducción a la Arquitectura de Software II. Universidad de Sevilla. 2013.* 2013.
- ISL. 2014.** Enfermedades Profesionales. Instituto de Seguridad Laboral Ministerio del Trabajo y Previsión Social. [En línea] 2014. http://www.isl.gob.cl/wp-content/uploads/2014/04/Enfermedad_Profesional.pdf.
- Larman. 2003.** *Parte v. Fase de construcción.* . 2003.
- Larman, Craig. 2003.** *UML y Patrones.* 2003. 2da Edición.
- . **2004.** *UML y Patrones 2da Edición.* 2004.
- . **2006.** *UML y Patrones. Cap 18, Modelo de Diseño, creación de los DCD.* 2006. ISBN.
- Mendez, Elita, y otros. 2011.** Ciencia de la Educación. Cultura Preventiva del riesgo laboral del docente universitario. *Cultura Preventiva del riesgo laboral del docente universitario.* 2011, Vol. 11.
- Mendez, Elita, y otros. 2011.** Cultura Preventiva del riesgo laboral del docente universitario. [En línea] 2011.
- Merced, Silvano. 2013.** Revista Cubana de Salud y Trabajo. [En línea] 2013.

Referencias Bibliográficas

- . 2012. Tesis: SISTEMA DE ACTIVIDADES FÍSICO PROFILÁCTICAS FORMATIVAS PARA CONTRIBUIR A LA PREVENCIÓN DE LAS ENFERMEDADES PROFESIONALES EN LOS ESTUDIANTES DE INGENIERÍA EN CIENCIAS INFORMÁTICAS. 2012.
- Merced, Silvano y Velazquez, Alionuska. 2014.** La prevención de enfermedades profesionales en el informático y su impacto social. 2014.
- Merced, Silvano, y otros. 2012.** Multimedia "Mente sana en cuerpo sano": Una alternativa para prevenir las enfermedades profesionales en el informático. [En línea] 2012.
- MICROSOFT. 2015.** Revisiones de código y estándares de codificación. [En línea] 2015. <https://msdn.microsoft.com/es-es/library/aa291591>.
- Mora, Aylín Maria. 2011.** Software Educativo: "Mente Sana en Cuerpo sano". [En línea] 2011.
- Mora, Aylín. 2011.** Software Educativo "Mente Sana en Cuerpo sano". Trabajo de diploma. [En línea] 2011.
- Pérez, Christian. 2015.** Consecuencias y efectos del sedentarismo. [En línea] 2015.
- Pressman. 2013.** *Diseño Arquitectónico Parte 1. 6ta Edición.* 2013. ISBN.
- . 2013. *Ingeniería de Requisitos Parte 1. 6ta Edición.* 2013. ISBN:9789701054734.
- . 2014. *Modelos de Proceso. Parte 2.* 2014.
- Pressman, Roger. 2010.** Ingeniería del Software, un enfoque práctico. 2010.
- . 2013. Técnicas de prueba Parte 1 6ta Edición. 2013.
- Pressman, Roger.** *Ingeniería de Software . Un enfoque práctico.*
- . 2015. Ingeniería del Software. Un enfoque práctico. s.l. : Edtion ed. 614 p, 2015.
- Pressman, Rogger. 2005.** *Ingeniería de Software. Un enfoque práctico.* 2005.
- Quinga, Julio E. 2015.** Riesgos ergonomicos y su incidencia en el desempeño laboral de los colaboradores del área administrativa en la empresa "Importadora Alvarado Vasconez", ciudad de Ambato Provincia de Tungurahua. [En línea] 2015.
- Renguifo, Johanna y Carlos., Betancourt. 2011.** Frameworks y Herramientas para el desarrollo de aplicaciones orientadas a dispositivos móviles. [En línea] 2011. <http://repositorio.utp.edu.co/dspace/bitstream/11059/2490/1/00416R412.pdf>.
- Rischpater, Ray. 2014.** *Application Development with QT Creator.* 2014.

Referencias Bibliográficas

- Sánchez, Elizabeth y Estrada, Odiel. 2010.** Aplicación Informática para la evaluación de los factores de riesgos laborales en el sector empresarial cubano. [En línea] Junio de 2010.
- SOFTPEI. 2013.** Introducción al Framework Qt 5 para el desarrollo de aplicaciones multiplataforma. [En línea] 14 de Mayo de 2013.
- Somavia, Juan. 2015.** Día Mundial de la Seguridad y la Salud en el Trabajo. [En línea] 2015. <http://www.un.org/es/events/safeworkday/>.
- Somerville. 2007.** Software Engineering. [En línea] 2007. http://eva.uci.cu/mod/resource/view.php?id=9269&subdir=/Sommerville_8va_edicion..
- Sparxsystems. 2016.** Diagrama de despliegue UML 2. [En línea] 2016. http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.
- UCI. 2012.** Portal de Internet. Misiones UCI. Informació sobre estudios de pregrado. [En línea] 2012. <http://www.uci.cu/pregrado#estudiantes>.
- UGT. Manual Informativo de PRL: Enfermedades Profesionales.** [En línea] <http://www.ugtbalears.com/es/PRL/Mutuas/Documents/MANUAL%20EEPP.pdf>.
- USERS.DCC. 2015.** Tutorial de UML. Caso de Uso. [En línea] 2015. <http://users.dcc.uchile.cl/~psalinas/uml/casosuso.html..>
- Velazquez, Martin. 2012.** Administración Pública y Política Informática. *El impacto social de la informática*. [En línea] 10 de noviembre de 2012. [Citado el: 10 de noviembre de 2012.] <http://admonpublicaypoliticainformatica.blogspot.com/>.
- Villagran, Migue. 2013.** El impacto de la informática en la sociedad actual. [En línea] 18 de Agosto de 2013. [Citado el: 18 de Agosto de 2013.] <http://aspectosprofesionales.blogspot.com/>.

Glosario de términos

A continuación, se muestra en orden alfabético el significado de algunos términos utilizados en el trabajo de tesis cuyo uso no es común y puede dificultar la comprensión del mismo.

A

Aplicación: programa informático que proporciona servicios de alto nivel a las personas, generalmente utilizando otros programas más básicos que se sitúan por debajo.

Automatizar: disponer complicadas operaciones mecánicas de carácter industrial, comercial o científico, para que se realicen automáticamente mediante la aplicación de dispositivos electrónicos.

Absentismo laboral: falta de asistencia al trabajo practicada habitualmente.

Aparato locomotor: conjunto de estructuras que le permite al cuerpo realizar cualquier tipo de movimiento.

AUP-UCI: metodología que describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles. Es una adaptación de la metodología AUP.

B

Bursitis: hinchazón e irritación de una bursa (bolsa), un saco lleno de líquido que actúa como amortiguador entre los músculos, los tendones y los huesos.

C

Codo de tenista: inflamación o dolor en el lado externo (lateral) de la parte superior del brazo cerca del codo.

Cardio-respiratorio: Oferta a los tejidos y órganos el oxígeno suficiente, según las necesidades individuales y dependiendo del estado o actividad. Es el encargado de proveer y hacer llegar hasta el músculo el oxígeno necesario para su funcionamiento.

Colesterol: molécula grasa que se produce generalmente en el hígado y los intestinos o se ingiere con los alimentos y cuya excesiva acumulación causa enfermedades circulatorias.

CASE: (Computer Aided Software Engineering): aplicaciones informáticas para aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y dinero.

Glosario de términos

C++: lenguaje orientado a objetos diseñado por Bjarne Stroustrup. Se creó con el objetivo de extender el lenguaje C con mecanismos que permitieran la manipulación de objetos. Es un lenguaje híbrido y multiplataforma.

D

dB: el decibelio (dB) es una unidad que se utiliza para medir la intensidad del sonido, unidad que recibe su nombre por Graham Bell.

Dedo en Gatillo: limita el movimiento del dedo. Cuando usted trata de poner el dedo recto, este se trabará o inmovilizará antes de ponerse recto. Es una condición que afecta a los tendones de los dedos de su mano.

E

Estrés: alteración física de una persona por exigir a su cuerpo un rendimiento superior al normal.

Ergonomía: disciplina científico-técnica y de diseño que estudia la relación entre el entorno de trabajo y quienes realizan el trabajo.

F

Fomentar: impulsar, promover, favorecer.

Fatiga visual: alteración que sufren las personas que aparecen generalmente como consecuencia de un esfuerzo prolongado de la visión.

Fatiga auditiva: disminución temporal de la audición que aparece tras la exposición prolongada a un ruido intenso.

Framework: representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

H

Hipoacusia: disminución de la capacidad auditiva.

HAEduc: Herramienta de autor para la creación de aplicaciones multimedia con la posibilidad de generar dichas aplicaciones para los sistemas operativos Linux y Windows.

HU:

Hardware: corresponde a todas las partes físicas y tangibles de una computadora.

Glosario de términos

HTML: (Hypertext Markup Language): lenguaje de marcado para hipertextos, utilizado para la confección de páginas web.

I

IDE: (Integrated Development Environment): entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

Interfaz: parte del programa informático que permite el flujo de información entre varias aplicaciones o entre el propio programa y el usuario.

J

Java: lenguaje orientado a objetos, compilado, generando ficheros de clases compilados. Es un lenguaje multiplataforma, el mismo código java que funciona en un sistema operativo funcionará en cualquier otro sistema operativo que tenga instalada la máquina virtual java.

L

Lumbar: la zona lumbar (espalda baja) está situada entre la última costilla y los riñones.

Lenguaje: en informática cuando se habla de lenguaje se refiere generalmente al de programación, conjunto de instrucciones que las aplicaciones necesitan para que el ordenador ejecute determinadas operaciones.

M

Músculo esqueléticas: lesiones que afectan a los músculos, tendones, huesos y ligamentos.

Metabolismo: conjunto de reacciones que efectúan las células de los seres vivos para descomponer y asimilar los alimentos.

Multimedia: integración de soportes o procedimientos que emplean sonido, imágenes o textos para difundir información, especialmente si es de forma interactiva.

Multiplataforma: término utilizado para referirse a los programas, lenguajes de programación, u otra clase de software, que puede funcionar en diversas plataformas.

P

Prolongadas: alargar, dilatar en cuanto al espacio. Hacer que dure una cosa más de lo normal.

Paulatino crecimiento: procede o actúa despacio y de forma gradual (lenta).

Glosario de términos

Prominencia: Elevación o abultamiento de una cosa sobre lo que está alrededor. Altura relativa.

Procrastinado: acción o hábito de retrasar actividades o situaciones que deben atenderse, sustituyéndolas por otras situaciones más irrelevantes. Es un problema de autorregulación y de organización del tiempo.

S

Sedentaria: persona de poca agitación o movimiento. Ejemplo: el trabajo de oficinista.

Síndrome del túnel carpiano: es una afección en la cual existe una presión excesiva en el nervio mediano. Este es el nervio en la muñeca que permite la sensibilidad y el movimiento a partes de la mano.

Sinovial: revestimiento de la vaina protectora que cubre los tendones.

Síndrome Cervical por Tensión: contractura muscular incontrolable y persistente en la región cervical posterior, que afecta a un músculo o a un grupo muscular.

Software: equipamiento lógico o soporte lógico de una computadora digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de tareas específicas.

Software Libre: software que brinda libertad a los usuarios sobre un producto adquirido, por tanto, una vez obtenido, puede ser utilizado, copiado, estudiado, modificado y redistribuido libremente.

T

Tendinitis: inflamación de un tendón. Causa dolor y sensibilidad cerca de una articulación.

Tendón: cuerdas resistentes de tejido que unen los músculos con los huesos y ayudan a los músculos a mover los huesos.

Tenosinovitis: inflamación del revestimiento de la vaina que rodea al tendón, el cordón que une el músculo con el hueso.

Traumatismo: lesiones internas o externas provocadas en los tejidos.

Tejido: cada una de las estructuras de células de naturaleza y origen idénticos que desempeñan en conjunto una determinada función en los organismos vivos

U

UCI: Universidad de las Ciencias Informáticas.

Glosario de términos

UML: Lenguaje Unificado de Modelado (Unified Modeling Language en inglés) prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo estos diagramas y símbolos significan.