



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

**CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**(TECNOLOGÍAS DE LA INFORMACIÓN)**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO**

**DE INGENIERO EN SISTEMAS E INFORMÁTICA**

**TEMA: DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UNA  
APLICACIÓN DE MONITOREO DE SIGNOS VITALES DEL PERSONAL  
MILITAR EN MISIONES DE COMBATE, APLICANDO METODOLOGÍA  
MDD.**

**AUTORES: TNTE. DE COM. HIDALGO ANANGONO, LUIS GABRIEL**

**SR. SINGAÑA TUTILLO, ESTEBAN DANIEL**

**DIRECTOR: RON EGAS, MARIO BERNABE**

**SANGOLQUÍ**

**2018**

## CERTIFICADO DEL DIRECTOR



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERIA EN SISTEMAS

### CERTIFICACIÓN

Certifico que el trabajo de titulación, "DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN DE MONITOREO DE SIGNOS VITALES DEL PERSONAL MILITAR EN MISIONES DE COMBATE, APLICANDO METODOLOGÍA MDD" fue realizado por el señor Esteban Daniel, Singaña Tutillo y Tnte. Hidalgo Anangono, Luis Gabriel el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 20 de diciembre del 2018

Firma:

ING. RON EGAS MARIO BERNABE

C.C. 1704229797

## AUTORÍA DE RESPONSABILIDAD



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA EN SISTEMAS

### AUTORÍA DE RESPONSABILIDAD

Nosotros, **Esteban Daniel Singaña Tutillo** y **Tnte. Hidalgo Anangono Luis Gabriel**, declaramos que el contenido, ideas y criterios del trabajo de titulación: **DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN DE MONITOREO DE SIGNOS VITALES DEL PERSONAL MILITAR EN MISIONES DE COMBATE, APLICANDO METODOLOGÍA MDD**. Es de nuestra autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 12 de Noviembre del 2018

Firma

Singaña Tutillo Esteban Daniel  
Gabriel  
C.C.:1724124167

Firma

Tnte. Hidalgo Anangono Luis  
C.C.:1722004775

## AUTORIZACIÓN



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERIA EN SISTEMAS

### AUTORIZACIÓN

Nosotros, Esteban Daniel Singaña Tutillo y Tnte. Hidalgo Anangono Luis Gabriel, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación, **DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN DE MONITOREO DE SIGNOS VITALES DEL PERSONAL MILITAR EN MISIONES DE COMBATE, APLICANDO METODOLOGÍA MDD** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Sangoquí, 12 de Noviembre del 2018

Firma

Singaña Tutillo Esteban Daniel  
Gabriel  
C.C.:1724124167

Firma

Tnte. Hidalgo Anangono Luis  
C.C.:1722004775

## **DEDICATORIA**

*A mi madre Mónica Tutillo por todo su amor, consejos y apoyo incondicional, lo que me ha llevado a cumplir una meta más en mi vida.*

Daniel Singaña

*A mi familia en especial a mi madre MARIA BEATRIZ ANANGONO MORALES junto con mis hermanos ROBERTO Y CRISTINA me han estado apoyando en cada paso de mi vida y han hecho posible la culminación de esta grandiosa etapa.*

Luis Hidalgo

## AGRADECIMIENTO

Quiero agradecer a Dios por darme la vida, por guiarme, especialmente por las personas que ha puesto en mi camino. A mis padres por todo su esfuerzo que han puesto en mí, especialmente quiero agradecer a mi madre quien con sus consejos y especialmente con amor me ha impulsado para tener un éxito más en mi vida. A mi director de tesis Ing. Mario Ron quien ha puesto todos sus conocimientos en beneficio de cumplir con este proyecto.

Daniel Singaña

Agradezco a mis padres y hermanos quienes con su paciencia y amor han sido mi soporte, ejemplo y guía, al brindarme todo su amor incondicional.

A mis amigos quienes han estado junto a mí durante esta larga etapa de la vida, me han brindado su mano en todo momento y he tenido grandes experiencias durante el recorrido por la universidad.

También, agradezco a la Universidad de las Fuerzas Armadas-ESPE, por permitirme conocer a grandes seres humanos y profesionales que han sido ejemplo y motivación para continuar y superarme día a día. Al equipo del Departamento de Ciencias de la Computación por su conocimiento y experiencia en el tema de esta investigación, en especial a mi director de proyecto Ing. Mario Ron, quien me ha guiado con paciencia y dedicación hasta culminar esta etapa.

Luis Hidalgo

## INDICE DE CONTENIDOS

<b>CERTIFICADO DEL DIRECTOR.....</b>	<b>i</b>
<b>AUTORÍA DE RESPONSABILIDAD.....</b>	<b>ii</b>
<b>AUTORIZACIÓN.....</b>	<b>iii</b>
<b>DEDICATORIA.....</b>	<b>iv</b>
<b>AGRADECIMIENTO.....</b>	<b>v</b>
<b>INDICE DE CONTENIDOS.....</b>	<b>vi</b>
<b>ÍNDICE DE TABLAS.....</b>	<b>viii</b>
<b>ÍNDICE DE FIGURAS.....</b>	<b>ix</b>
<b>RESUMEN.....</b>	<b>xi</b>
<b>ABSTRACT.....</b>	<b>xii</b>
<b>CAPÍTULO I</b>	
<b>PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACIÓN</b>	
<b>INTRODUCCIÓN.....</b>	<b>2</b>
1.1 Antecedentes.....	2
1.2 Planteamiento del problema.....	4
1.3 Descripción resumida del proyecto.....	6
1.4 Justificación e importancia.....	7
1.5 Objetivos.....	8

1.5.1	Objetivo General .....	8
1.5.2	Objetivos Específicos .....	8

## **CAPÍTULO II MARCO TEÓRICO Y ESTADO DEL ARTE**

2.1	MARCO TEÓRICO.....	10
2.1.1	Eclipse .....	10
2.1.2	Base de datos Mysql.....	10
2.1.3	Lenguaje Java.....	11
2.1.4	Openxava.....	12
2.1.5	Primefaces .....	13
2.1.6	Sockets .....	13
2.1.7	Simulación.....	13
2.2	ESTADO DEL ARTE.....	15
2.3	METODOLOGÍA MDD.....	19
2.3.1	Tipos de modelos .....	20
2.3.2	Proceso .....	20

## **CAPÍTULO III PRUEBAS DE CAJA NEGRA Y CAJA BLANCA**

3.1	Pruebas del sistema .....	22
-----	---------------------------	----

## **CAPÍTULO IV CONCLUSIONES Y RECOMENDACIONES**

4.1	Conclusiones y Recomendaciones .....	43
-----	--------------------------------------	----

## ÍNDICE DE TABLAS

<b>Tabla 1</b> <i>Validación de dígitos para ingresar a la base de datos.</i> .....	27
<b>Tabla 2</b> <i>Registro persona en la base de datos.</i> .....	28
<b>Tabla 3</b> <i>Valores para ingresar a la base de datos.</i> .....	33
<b>Tabla 4</b> <i>Resultados Prueba Condiciones Múltiples</i> .....	36
<b>Tabla 5.</b> <i>Comparación De Cronogramas</i> .....	43

## ÍNDICE DE FIGURAS

<i>Figura 1.</i> Desarrollo Dirigido por Modelos OpenXava. ....	12
<i>Figura 2.</i> Desarrollo Dirigido por Modelos.....	19
<i>Figura 3.</i> Fotografía de entrenamiento al sistema .....	23
<i>Figura 4.</i> Fotografía de entrenamiento al sistema (Monitoreo).....	23
<i>Figura 5.</i> Fotografía de ingreso de datos al sistema. ....	24
<i>Figura 6.</i> Fotografía de visualización de la posición.....	24
<i>Figura 7.</i> Fotografía de visualización de la presión arterial .....	25
<i>Figura 8.</i> Fotografía de visualización de la temperatura corporal.....	25
<i>Figura 9.</i> Fotografía de visualización del ritmo cardíaco.....	26
<i>Figura 10.</i> Fotografía de ingreso de cédula 0 dígitos. ....	27
<i>Figura 11.</i> Fotografía de ingreso de cédula 1 dígitos. ....	27
<i>Figura 12.</i> Fotografía de ingreso de cédula 10 dígitos.....	28
<i>Figura 13.</i> Fotografía de integridad de la base de datos1.....	29
<i>Figura 14.</i> Fotografía de integridad de la base de datos 2.....	29
<i>Figura 15.</i> Fotografía de integridad de la base de datos.....	30
<i>Figura 16.</i> Fotografía prueba de similitud.....	30
<i>Figura 17.</i> Fotografía de prueba. ....	31
<i>Figura 18.</i> Fotografía de prueba.....	32
<i>Figura 19.</i> Fotografía de prueba.....	32

<b>Figura 20.</b> Fotografía de inserción a la base de datos .....	33
<b>Figura 21.</b> Fotografía de inserción nula a la base de datos .....	34
<b>Figura 22.</b> Fotografía de una correcta inserción a la base de datos.....	34
<b>Figura 23.</b> Algoritmo para la prueba de Decisión y Condición .....	35
<b>Figura 24.</b> Resultado del Algoritmo para la prueba de Decisión y Condición. ....	36
<b>Figura 25.</b> Resultados Prueba Condiciones Múltiples Escenario 1.....	37
<b>Figura 26.</b> Resultados Prueba Condiciones Múltiples Escenario 2.....	37
<b>Figura 27.</b> Resultados Prueba Condiciones Múltiples Escenario 3.....	38
<b>Figura 28.</b> Cronograma Del Proyecto. ....	42
<b>Figura 29.</b> Cronograma Al Inicio Del Proyecto sin MDD. ....	42
<b>Figura 30.</b> Cronograma Al Final Del Proyecto con MDD. ....	42

## **RESUMEN**

Las operaciones de combate tanto en áreas abiertas como en sitios poblados requieren capacidad física adecuada y condiciones óptimas de salud del personal militar para ejecutarlas de manera eficaz y eficiente. De igual manera la información acerca de esas condiciones y de la ubicación exacta de cada combatiente es indispensable y esencial para el planeamiento y la ejecución de las misiones encomendadas a las unidades militares. Por lo cual los signos vitales constituyen una herramienta valiosa como indicadores del estado funcional de una persona. En el presente proyecto de investigación se realiza la implementación de un software, del monitoreo de los signos vitales del personal militar, para lo cual el presente proyecto contempla los siguientes signos vitales como la presión, la temperatura, el ritmo cardíaco y la posición, utilizando una metodología ágil para la construcción del software para lo cual se decidió la metodología dirigida por modelos, ya que esta metodología permite reducir el tiempo de desarrollo. Para validar el software se realizaron pruebas de caja negra y caja blanca dichas pruebas se detallan en este documento, dando como resultado un software aceptable. Cabe recalcar que el software es en tiempo real lo que facilita la toma de decisiones sobre un militar.

### **PALABRAS CLAVE:**

- **MDD**
- **IoT**
- **CICTE**
- **MILNOVA**
- **OPENXAVA**

## **ABSTRACT**

The combat operations. In the same way, information about these conditions and the exact location of each combat is essential and essential for the planning and execution of the missions entrusted to the military units. Therefore, vital signs are a valuable tool as indicators of a person's functional status. In the present research project the implementation of a software, the monitoring of the vital signs of the military personnel is carried out, for which the present project contemplates the following vital signs as the pressure, the temperature, the cardiac rhythm and the position, using a methodology for the construction of the software for what has worked the way of working for the models, so that it can be used to reduce the development time. To validate the software, black box and white box tests have been carried out. It should be noted that the software is in real time that facilitates decision making about a military.

### **KEYWORDS:**

- **MDD**
- **IoT**
- **CICTE**
- **MILNOVA**
- **OPENXAVA**

**DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN DE  
MONITOREO DE SIGNOS VITALES DEL PERSONAL MILITAR EN MISIONES DE  
COMBATE, APLICANDO METODOLOGÍA MDD.**

# CAPÍTULO I

## INTRODUCCIÓN

### 1.1 Antecedentes

En la actualidad las organizaciones que se dedican a la construcción de software, tienen una gran variedad de metodologías para el desarrollo del software. Hasta el momento las metodologías se agrupan en tres grandes categorías: metodología tradicional, metodología ágil y metodología híbrida, que son comúnmente utilizadas en nuestro medio y describen a continuación (Mukhtar, Hafeez, Riaz, & Afzaal, 2013):

- **Metodologías tradicionales:** se refieren específicamente al desarrollo de software que se realiza en cascada o en espiral y que han sido muy importantes en el proceso de construcción del software. Cada una brinda opciones diferentes, por ejemplo: la creación de una aplicación o sistema software bajo la metodología en cascada permite, ordenar meticulosamente todas las fases del ciclo de vida de un producto software, de tal manera que cada fase pueda finalizar para continuar con la siguiente. Las etapas que comprenden esta metodología son: Análisis de requisitos, Diseño del Sistema, Codificación, Pruebas, Implementación y finalmente, Mantenimiento (Mukhtar, Hafeez, Riaz, & Afzaal, 2013). Por otra parte, para la creación de un software utilizando la metodología en espiral, las etapas de desarrollo se realizan de forma de una espiral; de forma que las iteraciones se

representan mediante un grupo de etapas, que se establecen en función de los requerimientos. Las etapas son; Determinar Objetivos, Análisis del riesgo, Desarrollo y finalmente, la Planificación (Rakshith & Annapurn, 2013). Cabe recalcar que las metodologías tradicionales siguen un orden muy riguroso en su aplicación, por lo que es muy difícil alterar el orden de desarrollo (Rakshith & Annapurn, 2013).

- **Metodologías ágiles:** se refieren al diseño y creación de un software de forma rápida. Se desarrolla el producto de software en forma iterativa e incremental, incluso en un entorno donde los requisitos son cambiantes. Es por ello que esta metodología es muy adecuada para los escenarios actuales de negocios, donde los requisitos cambian constantemente. Tiene varias etapas de adaptación e iterativas que no son formales, es decir que no siguen un estándar como las metodologías tradicionales (Rakshith & Annapurn, 2013).

Las etapas en la metodología ágil, implican la intercalación entre la especificación, implementación, diseño y pruebas. Lo más importante, es que permiten alterar el orden de desarrollo. Lo que no ocurre con las metodologías tradicionales (Rakshith & Annapurn, 2013).

- o **Metodología híbrida:** hace referencia a la unión de las metodologías tradicionales con las ágiles, haciendo que coexisten para formar una sola metodología. Este tipo de metodología toma las ventajas de ambos tipos de metodologías mencionadas anteriormente, por ejemplo, de las metodologías tradicionales se beneficia en documentar, permitiéndole de esta manera entender, extender y mantener el software y de las metodologías ágiles se obtiene el beneficio de las respuestas efectivas durante los cambios rápidos de requisitos, que están basados en un plan de proyecto flexible, cosa

que las metodologías tradicionales no permiten compartir la simplicidad en la creación de un producto o sistema software (Kapteijns, Jansen, Brinkkemper, Houët, & Barendse, 2009).

## **1.2 Planteamiento del problema**

Las operaciones de combate tanto en áreas abiertas como en sitios poblados requieren capacidad física adecuada y condiciones óptimas de salud del personal militar para ejecutarlas de manera eficaz y eficiente (Janzen, 2006).

La información acerca de esas condiciones y de la ubicación exacta de cada combatiente es indispensable y esencial para el planeamiento y la ejecución de las misiones encomendadas a las unidades militares.

La dinámica de las misiones de combate y las condiciones geográficas en las que se desarrollan, presentan muchas veces condiciones difíciles para el tratamiento de la información necesaria en la toma de decisiones oportunas por parte de los comandantes tácticos, operativos y estratégicos, quienes deben conocer en tiempo real las condiciones y la localización de los recursos humanos a su cargo (Beck, Beedle, & Bennekum, 2018). En caso de existir condiciones críticas en la salud de un combatiente, el conocimiento de manera oportuna de ésta condición y su ubicación es importante para que pueda ser recuperado sin que el daño sea significativo.

Existen algunas condiciones que afectan a la capacidad física operativa del personal de combate:

- Falta de acondicionamiento físico (resistencia al esfuerzo físico prolongado).
- Déficit en alimentación y consumo de agua
- Alteración psicológica (ansiedad, desesperación, angustia)
- Aislamiento o pérdida de contacto con su unidad de combate
- Falta de información oportuna de la condición previa al incidente.
- Heridas, lesiones o enfermedad.
- Condiciones ambientales adversas (frío, humedad, calor extremo).
- Falta de puntos de atención oportuna.
- Arquitectura urbana, topografía, obstáculos naturales o antrópicos.

En conformidad con lo expresado, el problema actual es que no se cuenta al momento con información confiable y oportuna, que permita conocer en tiempo real, la disponibilidad operativa del personal militar durante el desarrollo de una misión de combate, tanto en áreas urbanas como en rurales, para efectos de planificación, conducción y evaluación de las misiones encomendadas, así como la toma de decisiones oportunas en caso de incidentes que afecten a la integridad física del combatiente.

### 1.3 Descripción resumida del proyecto

En 2015, Zheng & Carter (2015) nos menciona el impacto del Internet of Things (IoT) el cual alcanzaría un despliegue exponencial en todos los tipos de industria mejorando la eficiencia y la efectividad de sus operaciones. En el ámbito militar, no ha sido diferente.

Las Fuerzas Armadas Ecuatorianas están tratando de equipar a sus soldados con tecnología IoT para monitorear y visualizar en tiempo real el estado de salud de cada persona durante las misiones de combate (Janzen, 2006).

Sin embargo, la puesta en producción, implementación y la investigación, al menos en Ecuador; ha sido muy complejo debido a dos problemas: (1) la organización en el diseño y la integración de los componentes y aplicaciones cuando participan varios investigadores e ingenieros de múltiples disciplinas (por ejemplo, control, software) y (2) la implementación compleja debido a la fragmentación tecnológica entre los contextos militares (por ejemplo, la fuerza aérea, el ejército y la marina).

Ha habido avances significativos en el campo de Internet of Things (IoT) recientemente. Al mismo tiempo, existe una demanda cada vez mayor de sistemas de salud para mejorar la salud y el bienestar humanos. En la mayoría de los sistemas de monitoreo de pacientes basados en IoT, especialmente en hogares inteligentes u hospitales, existe un punto de enlace (es decir, puerta de enlace) entre una red de sensores e Internet que a menudo sólo realiza funciones básicas como la traducción entre los protocolos utilizados en Internet y redes de sensores (Yagüe, 2007).

## 1.4 Justificación e importancia

El desarrollo de software para monitoreo de indicadores de la condición de salud y signos vitales de las personas, es una alternativa muy actual que permite procesar datos estadísticamente, para determinar las variaciones en la condición de salud y así brindar tratamiento en caso de que las variaciones condujeran a riesgos potenciales en el paciente.

La adquisición de los datos referentes a los signos vitales o indicadores específicos de la condición de salud de las personas se puede hacer mediante sensores que permiten capturar esta valiosa información. Los datos procesados configuran un diagnóstico de lo que es observado por especialistas médicos, y en ciertos casos, que estos datos permitan prever una condición crítica de la salud del paciente, en forma automática. De tal manera que se active una alarma de esta condición a la vez que es monitoreado mediante sistemas de geolocalización a fin de ubicarlo y prestar una atención de emergencia.

Actualmente, el Centro de Investigación Tecnológica del Ejército (CICTE) está en la búsqueda de mejorar todos sus procesos, por tal motivo, en el área de la tecnología requiere del desarrollo e implementación de un aplicativo (prototipo) de monitoreo para la salud del personal militar en misiones de combate. La aplicación pertenece al proyecto de investigación denominado MILNOVA con la que se pretende optimizar los recursos

militares en las operaciones de combate tanto en áreas abiertas como en sitios poblados, mientras se monitorea su capacidad física y condiciones óptimas de salud. La importancia del software es tal, que un software de monitoreo, al menos de estas características, servirá para tomar decisiones y acciones adecuadas para el rescate y auxilio del personal en el caso de que sus signos vitales y sus condiciones de supervivencia sean de alto riesgo.

Ahora bien, el CICTE dada la importancia del software y que este sea desarrollado con las mejores metodologías de desarrollo, ha decidido como metodología más idónea la utilización de MDD. Los resultados que puedan ser obtenidos durante la aplicación de la metodología MDD permitirán a las Fuerzas Armadas utilizar de mejor manera este tipo de metodología ágil de desarrollo. Al momento las Fuerzas Armadas del Ecuador no cuentan con estudios que le permita aplicar correctamente la metodología MDD para el desarrollo sus aplicaciones, software y sistemas.

## **1.5 Objetivos**

### **1.5.1 Objetivo General**

Diseñar, desarrollar e implementar una aplicación de monitoreo de signos vitales del personal militar en misiones de combate, aplicando metodologías ágiles (MDD).

### **1.5.2 Objetivos Específicos**

- Realizar una revisión de marco teórico acerca de la metodología MDD aplicada en los desarrollos de software de monitoreo de salud en el ambiente militar.

- Analizar, diseñar, y desarrollar el módulo de monitoreo de los signos vitales del personal militar en misiones de combate.
- Analizar, diseñar, y desarrollar el módulo de operaciones militares del personal militar en misiones de combate y el módulo de sensores.
- Implementar y probar los módulos desarrollados bajo el esquema de evaluación dinámica de pruebas de caja blanca y caja negra. Las cuales se detallan a continuación:

#### Pruebas de caja negra

- comprobación de valores límite
- integridad base de datos
- pruebas de situaciones de excepción,

#### Pruebas caja blanca

- comprobación de bucles
- pruebas de decisión y condición
- pruebas de flujo de datos
- pruebas de condiciones múltiples.

## **CAPÍTULO II**

### **MARCO TEÓRICO Y ESTADO DEL ARTE**

Esta parte del capítulo tiene como objetivo el dar a conocer varios conceptos con referencia al tema de investigación como lo es las herramientas que se van a utilizar y el sistema de monitorización detallando los dispositivos, técnicas y herramientas empleadas.

#### **2.1 MARCO TEÓRICO**

##### **2.1.1 Eclipse**

Eclipse es una plataforma de software que contiene un conjunto de herramientas open source multiplataforma basada en Java. Esta plataforma es utilizada para entornos de desarrollos integrados. Eclipse es un marco y un conjunto de servicios para construir un entorno de desarrollo a partir de componentes conectados (plug-in). Hay plug-ins para el desarrollo de Java (JDT Java Development Tools) así como para el desarrollo en C/C++, COBOL, etc (Universitat de Valencia, 2004).

##### **2.1.2 Base de datos Mysql**

Las bases de datos están en todas partes, desde las aplicaciones de escritorio más pequeñas hasta los sitios web más grandes. La información comercial importante se almacena en servidores de bases de datos que a menudo son poco seguros. MySQL

es el sistema de gestión de bases de datos relacionales de código abierto más utilizado en el mundo desarrollado y respaldado por Oracle Corporation. Su código fuente está disponible bajo los términos de la Licencia Pública General de GNU y otros acuerdos de licencia de propiedad. MySQL es la elección correcta cuando se trata de crear una aplicación de código abierto (Taran & Silnov, 2017).

La base de datos MySQL está bien protegida, pero como cualquier software, disponible para el público, no debe darse por hecho. Hoy en día, la seguridad del almacenamiento de datos es una gran preocupación y atrae cada vez más la atención de los usuarios. Cualquier persona que tenga acceso a esta información puede tener control sobre la infraestructura de una empresa, incluso podría vender esta información a los competidores de una compañía (Taran & Silnov, 2017).

### **2.1.3 Lenguaje Java**

Java es uno de los lenguajes de programación orientado a objetos más populares creado en el año de 1991. Los programas escritos en el lenguaje Java generalmente se traducen en bytecode de Java, que es interpretado por una máquina virtual Java. Este enfoque nos permite agregar un conjunto de características convenientes en el lenguaje, por ejemplo, administración de memoria automática, que reduce o elimina por completo el tipo de defectos como fugas de memoria y desbordamientos de búfer. Sin embargo, puede disminuir significativamente el rendimiento de la aplicación en comparación con el programa análogo bien diseñado escrito en lenguajes compilados. A veces, es conveniente utilizar las características de los

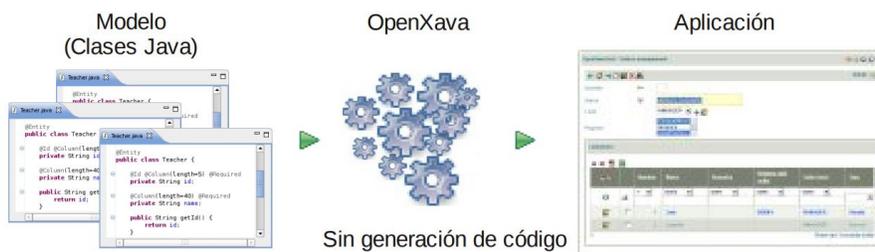
enfoques basados en la compilación y en la interpretación dentro de un sistema de software. La interfaz nativa de Java (JNI) se creó como parte de la especificación de la máquina virtual Java para cubrir dichos problemas (Vartanov, 2018).

### 2.1.4 Openxava

OpenXava es un marco para el desarrollo rápido de aplicaciones web, adecuado para aplicaciones comerciales orientadas a bases de datos; las aplicaciones OpenXava son aplicaciones estándar JavaEE / J2EE (Pelechano, 2012).

Sus características principales son:

- Utiliza JPA, Hibernate o EJB CMP2 para dar soporte a la persistencia de datos y cierta persistencia de estado a nivel de servlets.
- Proporciona módulos que se implementan como portlets estándar que permiten la implementación en cualquier portal que es compatible con el JSR-168.



**Figura 1.** Desarrollo Dirigido por Modelos OpenXava.

### **2.1.5 Primefaces**

PrimeFaces es un popular marco de código abierto para JavaServer Faces (JSF) que cuenta con más de 100 componentes como touchkit móvil optimizado, validación del lado del cliente, motor de temas y más; que facilitan la escritura y desarrollo de aplicativos web. PrimeFaces está establecido con la licencia de Apache V2, y una de las mayores ventajas de usar esta librería es que se puede utilizar e integrar otro tipo de componentes (OmniFaces, RichFaces, entre otros) (PrimeTek, 2018).

### **2.1.6 Sockets**

Socket es la interfaz básica de la aplicación de Internet que se basa en el protocolo TCP / IP que propone la Universidad de California BSD UNIX Branch. A través de la interfaz, puede comunicarse con cualquier computadora que tenga una interfaz de socket. También puede lograr transmisión de información, envío y recepción. En la actualidad, Internet se encuentra en una fase de transición de IPv4 a IPv6, con el fin de aprovechar mejor la gran cantidad existente de recursos de IPv4 y para lograr una transición sin problemas de la red se utiliza la teoría de programación de Sockets (Wang, 2010).

### **2.1.7 Simulación**

La simulación es considerada actualmente como una de las mayores y más útiles herramientas para analizar diversos aspectos de procesos y sistemas que pueden

llegar a ser muy complejos. A continuación se presenta una lista en la cual explica los pasos para realizar en una simulación:

1. Definición del sistema. Se plantea los objetivos del estudio, definiendo los límites o fronteras del modelo, las restricciones e interrelaciones en el sistema.
2. Creación del modelo. Se determina qué aspectos o partes del sistema se deben incluir y qué tan detallado debe ser, se realiza la reducción del sistema real a un diagrama de flujo lógico. Existen varias formas de estructurar el modelo conceptual. Una de ellas es con diagramas de flujo, que son una representación ordenada de las actividades que se llevan a cabo.
3. Recolección de datos. Recolectar los datos necesarios que el modelo requiere para la simulación.
4. Traducción del modelo. Elegir un lenguaje de programación para la simulación.
5. Validación. Comprobar que el sistema debe ser modelado conforme a los requerimientos.
6. Experimentación. Corrida de simulación para generar datos deseados, se puede observar el comportamiento del sistema con alguna modificación.
7. Interpretación. Se interpretan los datos arrojados por la simulación.

8. Implantación. Uso del modelo y resultados.
9. Documentación. Se registran los pasos del proyecto, sus resultados, además de la documentación del modelo y de su uso.

## 2.2 ESTADO DEL ARTE

El presente apartado tiene como objetivo recopilar varios estudios relacionados a la presente tesis, que permita examinar tecnologías similares y proporcionen una visión clara del problema que se requiere solucionar con el desarrollo de la aplicación planteada. Para conocer la literatura referente, se realizó una búsqueda minuciosa en varias bases digitales (IEEE Explore, Springer Link) cuyos resultados se muestran a continuación:

### **A Middleware Platform for Integrating Devices and Developing Applications in E-Health**

En este estudio se presenta a EcoHealth (Ecosystem of Health Care Devices), una plataforma de middleware para IdC (Internet de las Cosas) basada en la Web que integra sensores corporales heterogéneos que permiten el monitoreo remoto de pacientes con el fin de mejorar los diagnósticos médicos, además incluye la visualización, procesamiento y almacenamiento de datos en tiempo real. El proyecto de EcoHealth se basa en varias tecnologías Web bien establecidas (HTTP, REST y EEML) con el fin de estandarizar y simplificar el desarrollo de aplicaciones en el contexto de IdC, minimizando así problemas de compatibilidad e

interoperabilidad entre fabricantes, protocolos propietarios y formatos de datos. En esta investigación también se describe la aplicación de una evaluación cuantitativa del rendimiento de la plataforma, la cual fue sometida a una gran cantidad de solicitudes de actualizaciones de feeds en un corto periodo de tiempo. La evaluación mostró que la plataforma puede soportar una cantidad considerable de dispositivos físicos (por ejemplo, sensores corporales) enviando datos con frecuencias adecuadas para el monitoreo de señales vitales (Maia, y otros, 2015).

### **A SOA-based middleware for WBAN**

En este artículo, los autores proponen un diseño de middleware orientado a servicios para middleware WBAN (Wireless Body Area Network). En esta propuesta, los sensores que se utilizan están coordinados por un nodo de puerta de enlace, que a su vez retransmite datos a una unidad central remota que recibe información de control de WBAN y realiza consultas desde esta unidad central. La unidad central, por otro lado, se encarga de almacenar los datos de los sensores, la reconfiguración del sensor, la gestión de recursos del cliente, la detección y envío de información de los pacientes al personal médico. El usuario objetivo es un paciente que necesita un monitoreo regular. WBAN en este caso aumentará la comodidad del paciente y reducirá los controles periódicos que permiten el monitoreo remoto. Los investigadores afirman que el uso de servicios web y la estandarización de los mensajes intercambiados son una solución potencial para la interoperabilidad, la facilidad de uso y los desafíos de configuración (Abousharkh & Mouftah, 2011).

## **MMIG: A Middleware for Communicating a WSN to a Web Site for Monitoring Vital Signs**

Laflor y otros, diseñaron e implementaron el sistema multicapa MMIM (Middleware for Monitoring and Information Management), que permite establecer una comunicación confiable entre una red inalámbrica de sensores (RIS) basada en el estándar IEEE 802.15.4 y un portal web. MMIM tiene por objetivo brindar servicios de monitorización de signos vitales de pacientes que portan dispositivos con sensores, mediante un web, adicionalmente este sistema permite el establecimiento de intervalos de muestreo y límites para emisión de alertas. Este sistema, ofrece la adquisición de signos vitales con alto grado de confiabilidad en el orden de minutos o fracciones de los mismos, con un límite de 20 segundos para un total de 8 nodos sensores por cada nodo coordinador (Laflor, Nieto Hipólito, & Vázquez Briseño, 2011).

### **A proposed architectural model for vital sign monitoring system**

En esta investigación se presenta un sistema portátil de monitoreo de signos vitales utilizando GSM (Global System for Mobile Communications). El diseño e implementación del sistema de monitoreo de signos vitales se basa en Internet y en la red de sensores inalámbricos. El sistema consta de tres niveles: detección, comunicación y presentación. El sistema monitorea las mediciones de los signos vitales que incluyen: electrocardiogramas, mediciones de la presión arterial y temperatura. Además de la información de salud sobre el historial de los pacientes

se guarda en la base de datos. La aplicación de Internet de las cosas en los sistemas de atención médica mejora la eficacia y la eficiencia de la actividad de la organización de atención médica (Deshmukh & Wagh, 2015).

### **A Web-based vital sign telemonitor and recorder for telemedicine applications**

Por otra parte en este estudio, se describe un telemonitor de signos vitales (VST) que adquiere, registra, visualiza y proporciona lecturas tales como: electrocardiogramas (ECG), temperatura (T) y saturación de oxígeno (SaO<sub>2</sub>) a través de Internet en cualquier sitio. El diseño de este sistema constaba de tres partes: sensores, circuitos de procesamiento de señales analógicas y una interfaz gráfica de usuario (GUI) fácil de usar. La primera parte involucró la selección de sensores apropiados. Para ECG, electrodos Ag / AgCl desechables; para la temperatura, sensor de temperatura de precisión LM35; y para SaO<sub>2</sub> se seleccionó el kit de desarrollo de oximetría de Nonin equipado con un clip de dedo. La segunda parte consistió en procesar las señales analógicas obtenidas de estos sensores. Esto se logró implementando amplificadores y filtros adecuados para los signos vitales. La parte final se centró en el desarrollo de una GUI para mostrar los signos vitales en el entorno de LabVIEW. A partir de estas mediciones, se calcularon valores importantes como la frecuencia cardíaca (FC), los intervalos latido a latido (RR), SaO<sub>2</sub> porcentajes y T en ambos grados Celsius y Fahrenheit. Se podía acceder a la GUI a través de Internet en una página web que facilita la posibilidad de telemonitorización paciente en tiempo real. El sistema final

fue completado y probado en voluntarios con resultados satisfactorios (Mendoza, Gonzalez, & Villanueva, 2005).

## 2.3 METODOLOGÍA MDD

La idea detrás de MDD (Model Driven Development) es que modelar y transformar es una mejor base para el desarrollo y mantenimiento de sistemas que la programación. Los objetivos principales de MDD son la portabilidad, la interoperabilidad, productividad, mantenimiento y documentación y la reutilización a través de la separación arquitectónica de las prioridades. Un enfoque basado en modelos requiere lenguajes para la especificación de modelos, la definición de transformaciones y la descripción de metamodelos (Koch, 2006).

MDD permite crear software basados en el diseño de modelos y proporciona visión general del software a desarrollar; a pesar de esto MDD no define ninguna técnica proceso o metodología a emplearse. Es por esta razón que se han creado varios enfoques que utilicen MDD, uno de los más ampliamente usados MDA (Model Driven Architecture) (López, González, López, & Iduñate, 2006).



**Figura 2.** Desarrollo Dirigido por Modelos

### 2.3.1 Tipos de modelos

Un modelo de un sistema es una especificación de ese sistema y su entorno para ciertos fines. Los modelos consisten en un conjunto de elementos con una representación gráfica y / o textual. La idea de MDD es crear diferentes modelos de un sistema en diferentes niveles de abstracción y usar transformaciones para producir la implementación del sistema. MDA sugiere construir modelos independientes computacionales (CIM), modelos independientes de plataforma (PIM) y modelos específicos de plataforma (PSM) correspondientes a diferentes niveles de abstracción o puntos de vista (López, González, López, & Iduñate, 2006).

### 2.3.2 Proceso

La idea principal de utilizar MDA es obtener varios beneficios al realizar un desarrollo guiado por modelos de software, para ello se plantean el siguiente proceso:

- Como primer paso se debe obtener todos los requerimientos para poder representar el sistema mediante un modelo CIM, en el cual se detalla los futuros escenarios en el que el sistema será utilizado.
- El siguiente paso es transformar el modelo anterior a un modelo PIM cuyo objetivo será describir la funcionalidad del sistema, a pesar de esto, el

modelo no desplegara los detalles de cómo será la implementación de la aplicación en un plataforma tecnológica específica.

- Posteriormente se obtendrá el un modelo PSM en base al modelo PIM; aquí se muestran los detalles técnicos necesarios para manipular la plataforma tecnológica en la cual el sistema correrá.
- Como parte final, se emplea el modelo PSM para generar todo el código fuente que se necesita para obtener una solución coherente o un modelo completo y ejecutable.

## CAPÍTULO IV

### PRUEBAS

En este capítulo se realizará las pruebas en el programa y conjuntamente con los resultados obtenidos se realizará el análisis de los resultados obtenidos. Además, se podrá visualizar el software Milnova, el cual se lo ha presentado a la empresa CICTE con el fin de que se apruebe y este de acuerdo a los requerimientos acordados para la conformidad con el mismo.

Para realizar las pruebas del programa se presentará un escenario el cual se obtiene simulando la trama de datos que se obtienen de los sensores. A continuación se presentara los escenarios de realización de las pruebas.

5.1. Respecto al escenario de pruebas del sistema estas se las realizaron en las aulas de la Universidad de las Fuerzas Armadas “ESPE”.

#### 5.1.1. Paso 1: Utilización del sistema de MILNOVA para entrenamiento

Se realizó una introducción de cuál era la finalidad del programa así como de los componentes principales del software. El objetivo de esto, es realizar una consulta de que tan amigable es el sistema para el usuario. Los participantes nos proporcionan problemas que se tengan en la utilización del programa y así posteriormente plantear una solución al mismo.

## Bienvenido a Prueba

-  Hola `welcome_point1` en tus archivos de mensajes i18n.
-  Hols `welcome_point2` en tus archivos de mensajes i18n.
-  Modifica este párrafo con la entrada `welcome_point3` en tus archivos de mensajes i18n.

Para empezar con Prueba inicia sesión con el usuario: **admin**, contraseña: **admin**

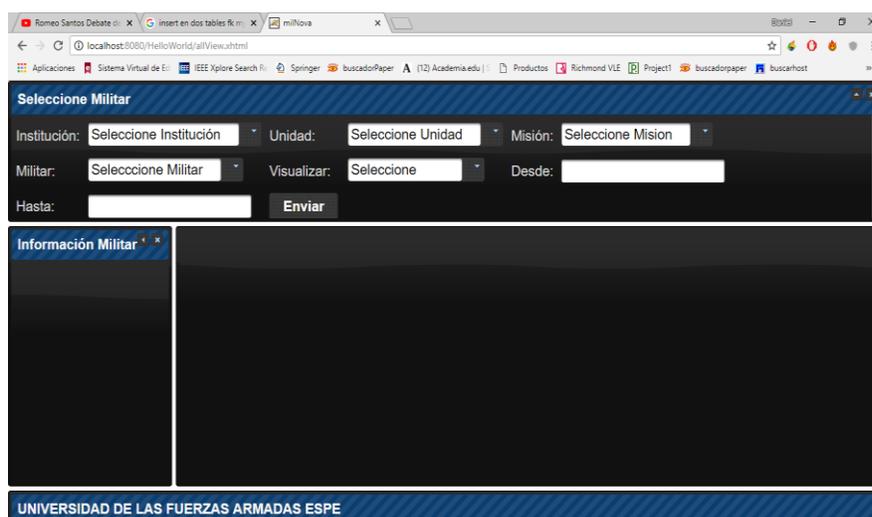


Usuario

Contraseña

**Figura 3.** Fotografía de entrenamiento al sistema

### 5.1.2. Paso 2: Utilización del sistema Milnova para entrenamiento



**Figura 4.** Fotografía de entrenamiento al sistema (Monitoreo)

### 5.1.3. Paso 3: Ingreso de datos al sistema

The screenshot shows a web browser window displaying the 'Selección Militar' application. The interface includes several dropdown menus and input fields for data entry:

- Institución:** FAE
- Unidad:** APOYO
- Misión:** RESCATAR
- Militar:** HIDALGO ANANGONO
- Visualizar:** Posición
- Desde:** (empty field)
- Hasta:** (empty field)

Below the input fields is an 'Enviar' button. To the right, there is a calendar for August 2018 with the date 29th highlighted. Below the calendar are input fields for 'Time', 'Hour', 'Minute', and 'Second', with '00:00:00' entered in the 'Time' field.

The footer of the application displays 'UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE'.

*Figura 5.* Fotografía de ingreso de datos al sistema.

### 5.1.4. Paso 4: Visualización de la posición

The screenshot shows the same web browser window as Figure 5, but now displaying a map and additional information:

- Institución:** FAE
- Unidad:** APOYO
- Misión:** RESCATAR
- Militar:** HIDALGO ANANGONO
- Visualizar:** Posición
- Desde:** 08/02/2018 15:37:32
- Hasta:** 08/02/2018 23:47:42

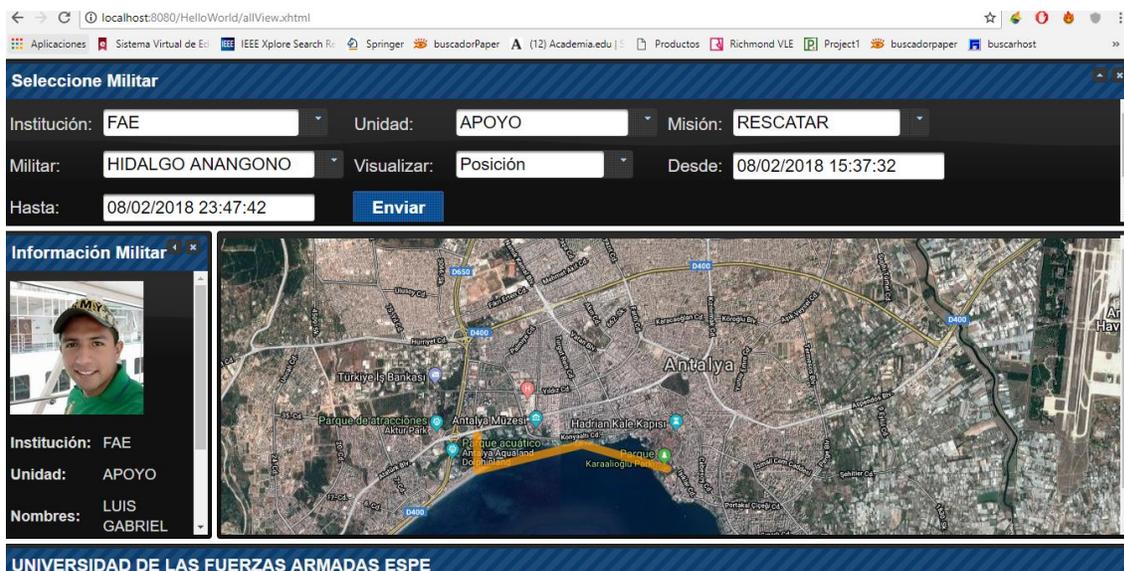
The 'Enviar' button is now highlighted in blue. Below the input fields, there is a section titled 'Información Militar' which includes a profile picture of a man in a military uniform and the following text:

- Institución:** FAE
- Unidad:** APOYO
- Nombres:** LUIS GABRIEL

To the right of the profile information is a satellite map of Antalya, Turkey, with a red location pin and a yellow path. The footer of the application displays 'UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE'.

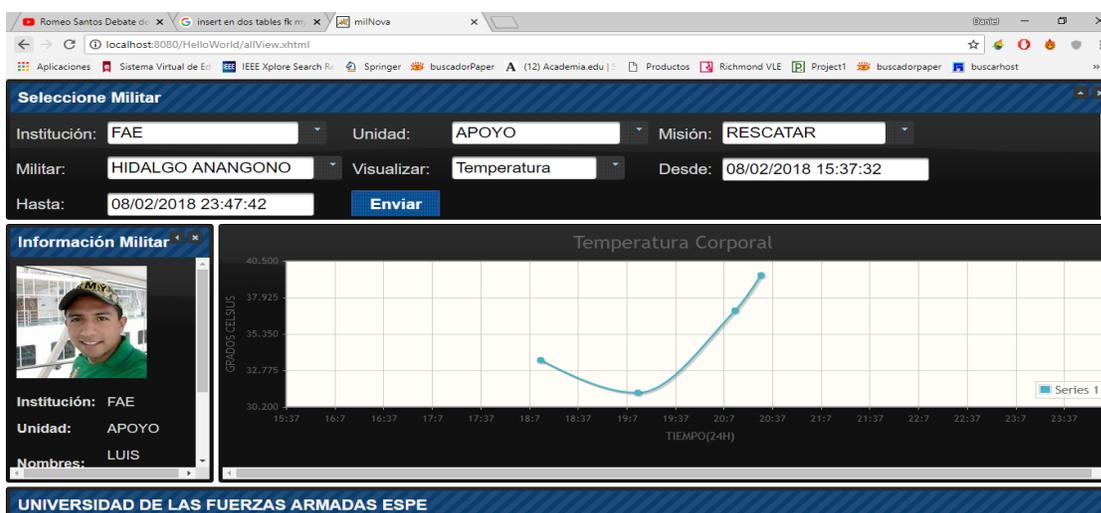
*Figura 6.* Fotografía de visualización de la posición

### 5.1.5. Paso 5: Visualización de la presión arterial



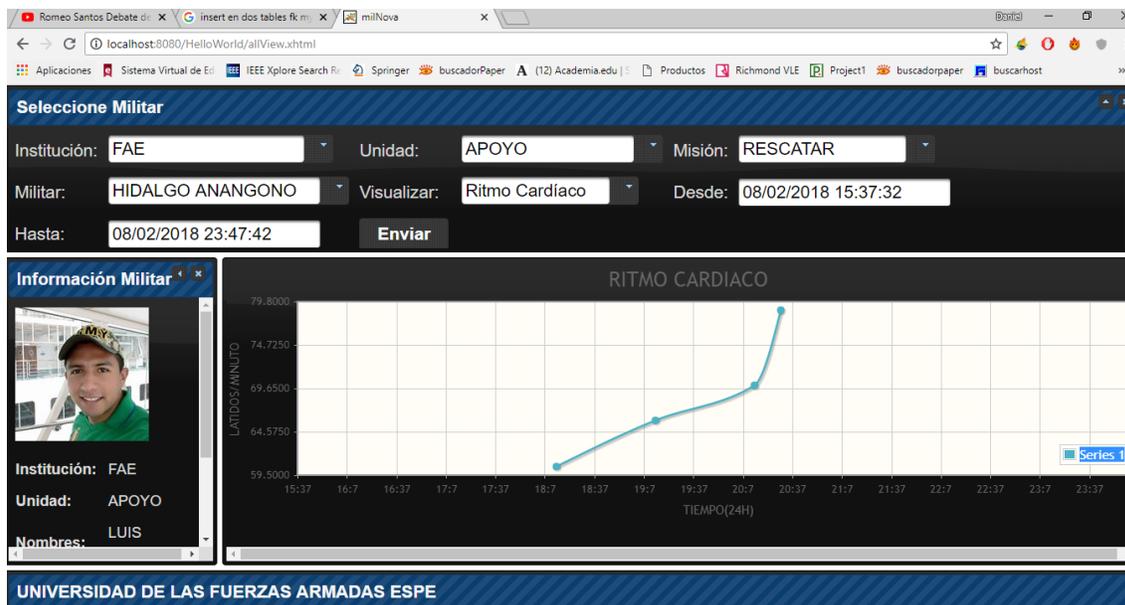
*Figura 7.* Fotografía de visualización de la presión arterial

### 5.1.6. Paso 6: Visualización de la temperatura corporal



*Figura 8.* Fotografía de visualización de la temperatura corporal

### 5.1.7. Paso 7: Visualización del ritmo cardiaco



**Figura 9.** Fotografía de visualización del ritmo cardiaco

## CAJA NEGRA

### COMPROBACIÓN DE VALORES LÍMITE

Realizamos en ingreso de los números de la cédula los cuales están dentro de este rango  $0 < \text{cedula} \leq 10$ .

Para tal caso se procede a realizar la comprobación límite a través de la cantidad de dígitos del número de la cédula de las personas en donde la cantidad permitida es de 10 dígitos. En donde el sistema únicamente permite el ingreso de 10 dígitos los mismos que se validan con un algoritmo para verificar que la cédula ingresada es válida.

**Tabla 1**

*Validación de dígitos para ingresar a la base de datos.*

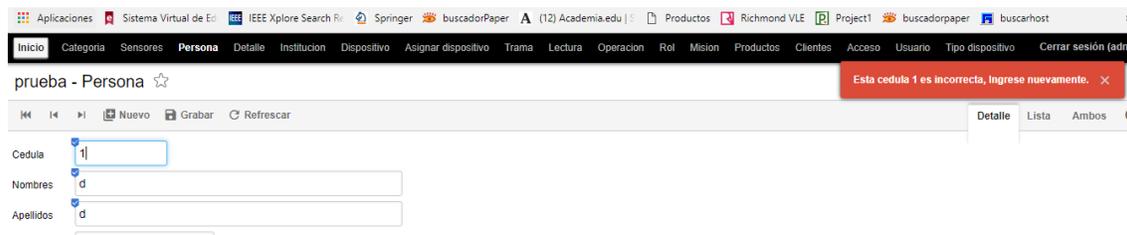
Nro.	Cantidad dígitos	Respuesta Sistema	Valor
1	C=0	No válido	0
2	C>0	No válido	1
3	C=10	Válido	10
4	C>10	No válido	11

### Dígitos Cédula=0



**Figura 10.** Fotografía de ingreso de cédula 0 dígitos.

### Dígitos Cédula>0



**Figura 11.** Fotografía de ingreso de cédula 1 dígitos.

### Dígitos Cédula=10

**Figura 12.** Fotografía de ingreso de cédula 10 dígitos

## INTEGRIDAD DE LA BASE DE DATOS

Para tal caso se va a realizar la inserción de una persona en la base de datos.

**Tabla 2**

*Registro persona en la base de datos*

Nro.	Actividad	Respuesta Sistema
1	Registrar Persona	Registro Correctamente
2	Registrar Persona con cédula repetida	No válido
3	Registrar Persona con cédula inválida	No válido

## Escenario 1

The screenshot shows a web application interface for creating a person record. A green notification banner at the top reads "Persona creada satisfactoriamente". The form fields are filled with the following data:

Campo	Valor
Cedula	1722004775
Nombres	Daniel
Apellidos	Sani
Tipo persona	LISTA MILITAR
Institucion	NAVAL
Unidad	RESCTAR VIDAD
Ciudad	Quito
Parroquia	Cayambe
Canton	Quito
Tipo sangre	o

A blue "Grabar" button is visible at the bottom left of the form.

**Figura 13.** Fotografía de integridad de la base de datos 1.

## Escenario 2

The screenshot shows the same web application interface as in Scenario 1, but with a red error notification banner at the top that reads "Imposible crear: ya existe un registro con esa clave". The form fields are filled with the same data as in Scenario 1:

Campo	Valor
Cedula	1722004775
Nombres	Daniel
Apellidos	Sani
Tipo persona	LISTA MILITAR
Institucion	NAVAL
Unidad	RESCTAR VIDAD
Ciudad	Quito
Parroquia	Cayambe
Canton	Quito
Tipo sangre	o

The blue "Grabar" button is still present at the bottom left.

**Figura 14.** Fotografía de integridad de la base de datos 2.

### Escenario 3

prueba - Persona

Esta cedula 1722004770 es incorrecta, Ingrese nuevamente.

Cedula: 1722004770

Nombres: Daniel

Apellidos: Sani

Tipo persona: LISTA MILITAR

Institucion: NAVAL

Unidad: RESCTAR VIDAD

Ciudad: Quito

Parroquia: Cayambe

Canton: Quito

Tipo sangre: 0

Grabar

**Figura 15.** Fotografía de integridad de la base de datos.

### Pruebas De Simulación De Excepción

Escenario 1: Cuando un arreglo está vacío se lanza esta excepción y el aplicativo la puede tratar.

```

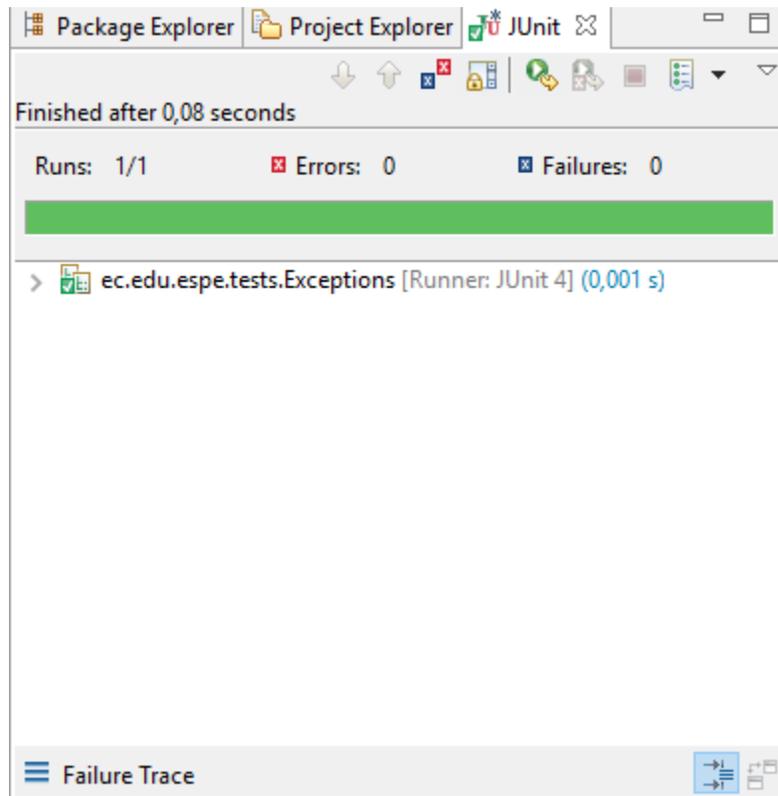
5
6 import org.junit.*;
7 import org.junit.rules.*;
8
9 import antlr.collections.*;
10
11 public class Exceptions
12 {
13     @Rule
14     public ExpectedException thrown = ExpectedException.none();
15
16     @Test
17     public void shouldTestExceptionMessage() throws IndexOutOfBoundsException {
18         List<Object> list = new ArrayList<Object>();
19
20         thrown.expect(IndexOutOfBoundsException.class);
21         thrown.expectMessage("Indice: 0, Tamaño: 0");
22         list.get(0);
23     }
24

```

**Figura 16.** Fotografía prueba de similitud

**Figura 16.** Fotografía prueba de similitud

RESULTADO:



**Figura 17.** Fotografía de prueba.

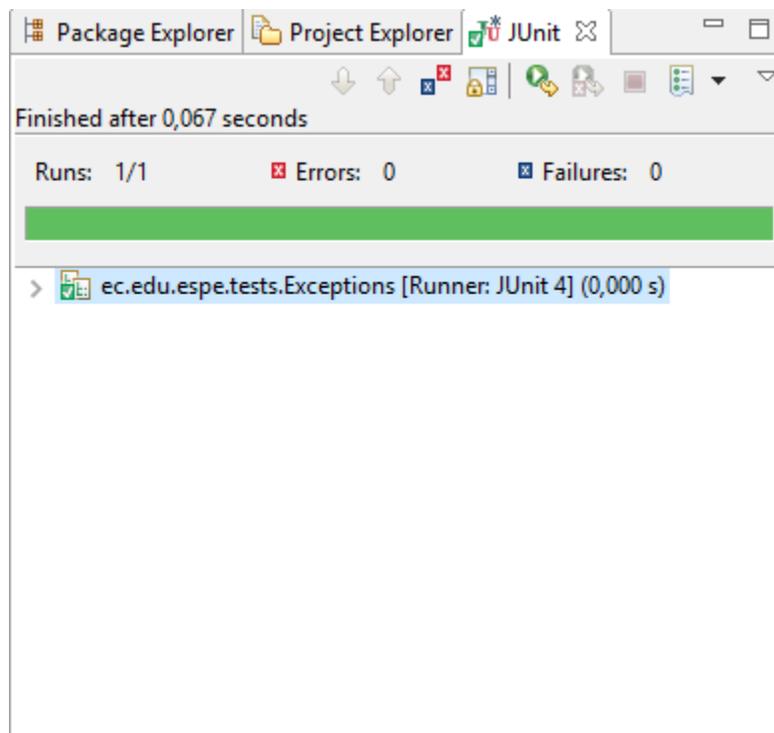
Escenario 2: Cuando se espera un valor entero y se introduce una cadena de caracteres



```
5
6 import org.junit.*;
7 import org.junit.rules.*;
8
9 import antlr.collections.*;
10
11 public class Exceptions
12 {
13     @Rule
14     public ExpectedException exceptionRule = ExpectedException.none();
15
16     @Test
17     public void whenExceptionThrown_thenRuleIsApplied() {
18         exceptionRule.expect(NumberFormatException.class);
19         exceptionRule.expectMessage("Para salida de una cadena");
20         Integer.parseInt("tres");
21     }
22 }
23
24
```

*Figura 18.* Fotografía de prueba

## RESULTADO



Package Explorer | Project Explorer | JUnit

Finished after 0,067 seconds

Runs: 1/1      Errors: 0      Failures: 0

> ec.edu.espe.tests.Exceptions [Runner: JUnit 4] (0,000 s)

*Figura 19.* Fotografía de prueba

**CAJA BLANCA**

## Comprobación De Bucles

Para tal caso se va a realizar la validación en el bucle generado por el servidor hacia el cliente utilizando sockets para realizar la inserción en la base de datos del servidor.

**Tabla 3**

*Valores para ingresar a la base de datos.*

Valor entrada Cliente	valor	Resultado
Valores negativos	-1	Valor inválido
Valor cero	0	Valor inválido
Valor mayor a cero	2	Valor válido

### Escenario 1

```

Ingresar el código de la persona:
1
Ingresar la fecha de Inicio (YYYY-MM-dd):
2018/08/02
Ingresar la fecha de Fin (YYYY-MM-dd):
2018/08/02
Ingresar cantidad de registros:
-1
VALOR INVALIDO

```

**Figura 20.** Fotografía de inserción a la base de datos

### Escenario 2

```

clave=7, valor=JUAN ORTIZ
clave=8, valor=JUAN TUTILLO
Ingresar el código de la persona:
3
Ingresar la fecha de Inicio (YYYY-MM-dd):
2018
Ingresar la fecha de Fin (YYYY-MM-dd):
2018
Ingresar cantidad de registros:
0
|VALOR INVALIDO

```

**Figura 21.** Fotografía de inserción nula a la base de datos

### Escenario 3

```

Ingresar el código de la persona:
3
Ingresar la fecha de Inicio (YYYY-MM-dd):
2018-08-02
Ingresar la fecha de Fin (YYYY-MM-dd):
2018-08-02
Ingresar cantidad de registros:
2
|Registro 1 insertado correctamente.
|Registro 2 insertado correctamente.

```

**Figura 22.** Fotografía de una correcta inserción a la base de datos.

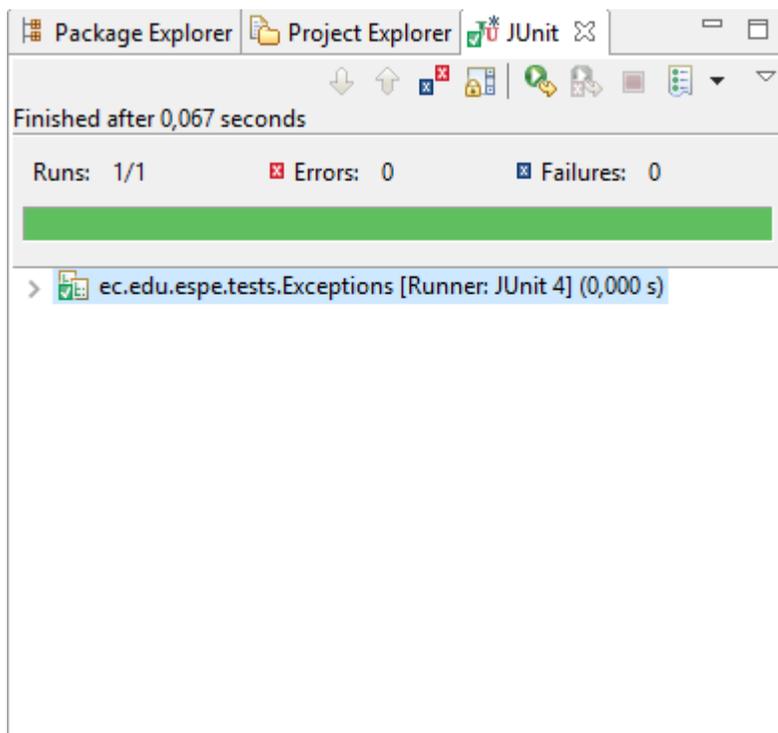
### Pruebas De Decisión y Condición

Para tal caso si el usuario esta logeado únicamente como administrador podrá realizar inserciones en la base de datos como muestra la prueba.

```
8 public UnidadTest(String testName)
9 {
10     super(testName, "prueba", "Unidad");
11 }
12
13 public void testCreateReadUpdateDelete() throws Exception
14 {
15     login("admin", "admin");
16
17     //CREAR
18     execute("CRUD.new");
19     setValue("nombre", "1");
20     setValue("tipo", "s");
21     setValue("descripcion", "2");
22     setValue("institucion", "2");
23     execute("CRUD.save");
24     assertNoErrors();
25
26 }
27
```

*Figura 23.* Algoritmo para la prueba de Decisión y Condición

## RESULTADO



**Figura 24.** Resultado del Algoritmo para la prueba de Decisión y Condición.

#### Pruebas De Condiciones Múltiples

Para tal caso vamos a utilizar la prueba con asignar un dispositivo a un militar ya que esta consta de varias condiciones para que puede ser realizada correctamente, para llevar a la práctica hemos seleccionado los siguientes escenarios en donde se obtuvieron las respuestas siguientes.

**Tabla 4**

*Resultados Prueba Condiciones Múltiples*

MISION	PERSONA	DISPOSITIVO	FECHA	RESULTADO
	P1	D1	F1	CORRECTO
M1	P2	D1	F1	INCORRECTO

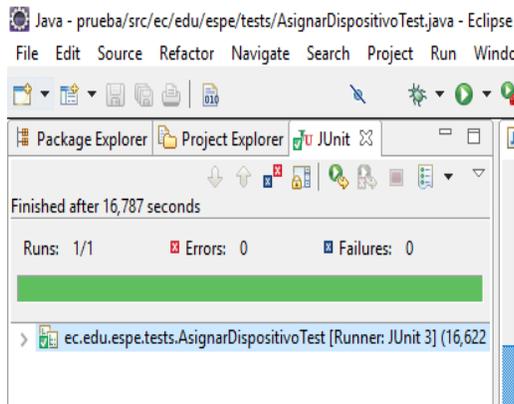
P2

D2

F1

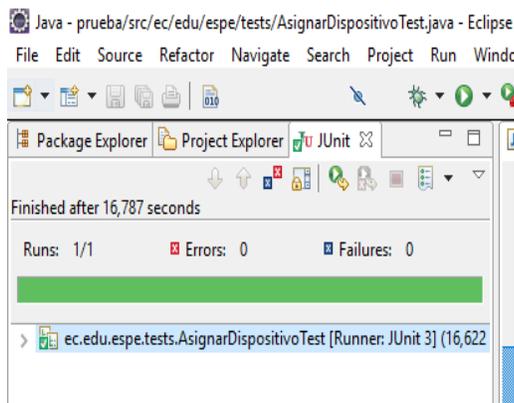
CORRECTO

## Escenario 1 Respuesta



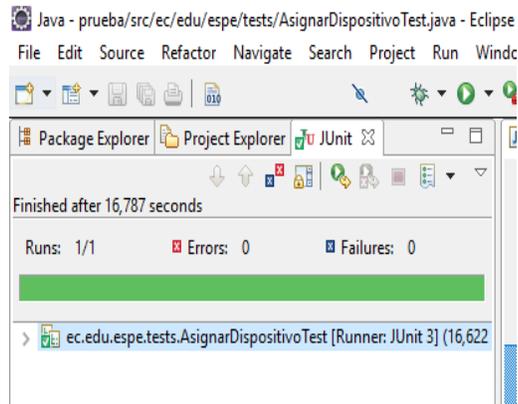
**Figura 25.** Resultados Prueba Condiciones Múltiples Escenario 1

## Escenario 2: Respuesta



**Figura 26.** Resultados Prueba Condiciones Múltiples Escenario 2

## Escenario 3: Respuesta



**Figura 27.** Resultados Prueba Condiciones Múltiples Escenario 3

## MODULO DE MONITOREO

### TEMPLATE

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

      xmlns:h="http://java.sun.com/jsf/html"

      xmlns:f="http://java.sun.com/jsf/core"

      xmlns:ui="http://java.sun.com/jsf/facelets"

      xmlns:p="http://primefaces.org/ui">

<f:view contentType="text/html">

    <h:head>

        <script>

            PrimeFaces.locales['es'] = {

                closeText : 'Cerrar',

                prevText : 'Anterior',

                nextText : 'Siguiente',

                monthNames : [ 'Enero', 'Febrero', 'Marzo', 'Abril',
'Mayo',
```

```
        'Junio', 'Julio', 'Agosto', 'Septiembre',  
'Octubre',  
  
        'Noviembre', 'Diciembre' ],  
monthNamesShort : [ 'Ene', 'Feb', 'Mar', 'Abr', 'May',  
'Jun',  
  
        'Jul', 'Ago', 'Sep', 'Oct', 'Nov', 'Dic' ],  
dayNames : [ 'Domingo', 'Lunes', 'Martes', 'Miércoles',  
  
        'Jueves', 'Viernes', 'Sábado' ],  
dayNamesShort : [ 'Dom', 'Lun', 'Mar', 'Mie', 'Jue',  
'Vie',  
  
        'Sab' ],  
dayNamesMin : [ 'D', 'L', 'M', 'M', 'J', 'V', 'S' ],  
weekHeader : 'Semana',  
firstDay : 1,  
isRTL : false,  
showMonthAfterYear : false,  
yearSuffix : '',  
timeOnlyTitle : 'Sólo hora',  
timeText : 'Tiempo',  
hourText : 'Hora',
```

```
minuteText : 'Minuto',  
  
secondText : 'Segundo',  
  
currentText : 'Fecha actual',  
  
ampm : false,  
  
month : 'Mes',  
  
week : 'Semana',  
  
day : 'Día',  
  
allDayText : 'Todo el día'  
  
};
```

## CRONOGRAMA MILNOVA

▷ Alcanze	6 days	Wed 4/11/18	Wed 4/18/18
▷ Analisis/Software Requerimientos	9 days	Thu 4/19/18	Tue 5/1/18
▷ Diseño	12 days	Wed 5/2/18	Thu 5/17/18
▷ Desarrollo	91 days	Fri 5/18/18	Fri 9/21/18
▷ Pruebas	5 days	Mon 9/24/18	Fri 9/28/18

*Figura 28.* Cronograma Del Proyecto.

★	↳ Desarrollo	91 days	Fri 5/18/18	Fri 9/21/18		
↳	Modulo Seguridad	20 days	Fri 5/18/18	Thu 6/14/18		Developer
↳	Modulo Operaciones	20 days	Fri 6/15/18	Thu 7/12/18	24	Developer
↳	Modulo Sensores	20 days	Fri 7/13/18	Thu 8/9/18	25	Developer
↳	Modulo Monitoreo	30 days	Fri 8/10/18	Thu 9/20/18	26	Developer

*Figura 29.* Cronograma Al Inicio Del Proyecto sin MDD.

★	↳ Desarrollo	67 days	Fri 5/18/18	<u>Mon 8/20/18</u>		
↳	Modulo De Seguridad	15 days	Fri 5/18/18	Thu 6/7/18		Developer
↳	Modulo Operaciones	12 days	Fri 6/8/18	Mon 6/25/18	24	Developer
↳	Modulo De Sensores	10 days	Tue 6/26/18	Mon 7/9/18	25	Developer
↳	Modulo De Monitoreo	30 days	Tue 7/10/18	Mon 8/20/18	26	Developer

*Figura 30.* Cronograma Al Final Del Proyecto con MDD.

## DESARROLLO DEL SOFTWARE CRONOGRAMA

**Tabla 5**

*Comparación De Cronogramas*

	Módulo	Módulo	Módulo	Módulo	Tiempo
	Seguridad	Operaciones	Sensores	Monitoreo	Total
<b>SIN MDD</b>	20 d	20 d	20 d	30 d	90 d
<b>CON MDD</b>	15 d	12 d	10 d	30 d	67 d

Utilizando MDD en el desarrollo del aplicativo se pudo reducir el tiempo en un 25% del tiempo previsto al inicio del proyecto.

## CAPÍTULO V

### CONCLUSIONES Y RECOMENDACIONES

En este capítulo se expondrán las conclusiones y recomendaciones obtenidas a lo largo del desarrollo del proyecto. Con esto se permitirá conocer si se cumplieron los objetivos planteados al inicio y si se logró proporcionar una solución al problemática planteada, dando por finalizado el presente proyecto de investigación, para beneficio de la institución militar del Ecuador.

#### CONCLUSIONES

La metodología MDD puede ser utilizada para proyectos posteriores de software militar debido a que nos ofreció muchas facilidades como la optimización de tiempo, en lo que respecta a la generación del código a través de los modelos.

OPENXAVA permitió desarrollar los módulos de seguridad, de sensores y de operaciones en menor tiempo debido a que utiliza clasesjava con anotaciones las cuales pueden ser reutilizadas en el aplicativo.

Se implementó el prototipo de monitoreo de la salud militar lo cual permitió un beneficio en la toma de decisiones.

Mediante las pruebas de caja negra y caja blanca el aplicativo fue validado.

La diferencia del factor tiempo al concluir el módulo de monitoreo desarrollado con el plugin OPENXAVA para MDD y el desarrollo tradicional fue mínimo.

## **RECOMENDACIONES**

Para un correcto desarrollo con la metodología MDD se recomienda definir correctamente el modelo a utilizarse en el desarrollo de software, debido a que los modelos son la clave que contempla la lógica del negocio.

Para un correcto desarrollo con la metodología MDD se recomienda tener conocimientos previos, específicamente JPA, anotaciones y definir correctamente el modelo a utilizarse

en el desarrollo de software, debido a que los modelos son la clave que contemplan la lógica del negocio.

Se recomienda realizar una nueva actualización del aplicativo con una base de datos no relacional debido a que estas pueden manejar miles de datos sin dificultad.

## XV. REFERENCIAS

- Abousharkh, M., & Mouftah, H. (2011). A SOA-based middleware for WBAN. *IEEE International Symposium on Medical Measurements and Applications*.
- Beck, K., Beedle, M., & Bennekum, A. (2 de Mayo de 2018). *agilemanifesto*. Obtenido de <http://agilemanifesto.org/>
- Deshmukh, V., & Wagh, S. (2015). A proposed architectural model for vital sign monitoring system. *International Conference on Communications and Signal Processing (ICCSP)*, 1758-1762.
- Janzen, D. (2006). . An empirical evaluation of the impact of test-driven development on software quality. *Dissertation*.
- Kaptejns, T., Jansen, S., Brinkkemper, S., Houët, H., & Barendse, R. (2009). A Comparative Case Study of Model Driven Development vs Traditional Development: The Tortoise or the Hare. *4th European Workshop on "From code centric to model centric software engineering: Practices, Implications and ROI"*.
- Koch, N. (2006). Transformation techniques in the model-driven development process of UWE. *ICWE '06 Workshop proceedings of the sixth international conference on Web engineering*.
- Laflor, A., Nieto Hipólito, J., & Vázquez Briseño, M. (2011). MMIG: A Middleware for Communicating a WSN to a Web Site for Monitoring Vital Signs. *Intercambio de servicios de salud panamericanos*.

- López, E., González, M., López, M., & Iduñate, E. (2006). Proceso de Desarrollo de Software Mediante Herramientas MDA. *SISTEMAS, CIBERNÉTICA E INFORMÁTICA*.
- Maia, P., Baffa, A., Cavalcante, E., Delicato, F., Batista, T., & Pires, P. (2015). A Middleware Platform for Integrating Devices and Developing Applications in E-Health. *XXXIII Brazilian Symposium on Computer Networks and Distributed Systems*, 10-18.
- Mendoza, P., Gonzalez, P., & Villanueva, B. (2005). A Web-based vital sign telemonitor and recorder for telemedicine applications. *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*.
- Mukhtar, M., Hafeez, Y., Riaz, M., & Afzaal, M. (2013). un modelo híbrido para prácticas ágiles Uso de Razonamiento Basado Case. *IEEE*.
- Pelechano, V. (2012). Automating the development of information systems with the MOSKitt open source tool. *2012 Sixth International Conference on Research Challenges in Information Science (RCIS)*.
- PrimeTek. (5 de Julio de 2018). *Primefaces*. Obtenido de <https://www.primefaces.org/>
- Rakshith , H., & Annapurn, P. (2013). Una perspectiva empobrecido Agile Metodología. *Conferencia Internacional sobre Ingeniería de Sistemas y Tecnología* .
- Taran, A., & Silnov, D. (2017). Research of attacks on MySQL servers using HoneyPot technology. *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIconRus)*, 224-226.

Universidad de Valencia. (2004). *Universidad de Valencia*. Obtenido de [https://www.uv.es/~jgutierr/MySQL\\_Java/TutorialEclipse.pdf](https://www.uv.es/~jgutierr/MySQL_Java/TutorialEclipse.pdf)

Vartanov, S. (2018). Dynamic symbolic execution of Java programs using JNI. *2017 Computer Science and Information Technologies (CSIT)*, 83-86.

Wang, Y. y. (2010). Transition of Socket applications from IPv4 to IPv6. *2010 2nd International Conference on Computer Engineering and Technology*, 75-78.

Yagüe, A. (2007). Test driven development: fortalezas y debilidades. *Reportes Técnicos 07-13*.