**INSTITUTO TÉCNICO “ENSEC”**

**“FELIPE LEONOR RIBERA”**

**CARRERA: SISTEMAS INFORMÁTICOS**



**SISTEMA WEB PARA LA GESTION DE MANTENIMIENTO Y REPARACIÓN DE EQUIPOS ELECTRÓNICOS, DEL TALLER “SERVICOM”**

PROYECTO DE GRADO PARA OPTAR EL TITULO PROFESIONAL TECNICO SUPERIOR EN SISTEMAS INFORMATICOS

**POSTULANTE**

MAMANI ALARCON ANDRES PAULO

**TUTOR**

ING. VARGAS LOZADA MARCELO EDUARDO

SANTA CRUZ-BOLIVIA

GESTION 2023

Tabla de contenido

[CAPITULO 1 ASPECTO METODOLOGICOS 1](#_Toc145959601)

[1.1 Introducción 1](#_Toc145959602)

[1.2 Diagnostico 2](#_Toc145959603)

[1.3 Planteamiento del problema 2](#_Toc145959604)

[1.3.1 Situación del problema 2](#_Toc145959605)

[1.3.2 Formulación del problema 3](#_Toc145959606)

[1.4 Objetivos 3](#_Toc145959607)

[1.4.1 Objetivo general 3](#_Toc145959608)

[1.4.2 Objetivos específicos 3](#_Toc145959609)

[1.5 Justificación 4](#_Toc145959610)

[1.5.1 Justificación económica 4](#_Toc145959611)

[1.5.2 Justificación tecnológica 4](#_Toc145959612)

[1.5.3 Justificación social 4](#_Toc145959613)

[1.6 Enfoque metodológico 5](#_Toc145959614)

[1.6.1 Cualitativo 5](#_Toc145959615)

[1.6.2 Metodología para el desarrollo del sistema de información 6](#_Toc145959616)

[1.6.3 Tipos de investigación 6](#_Toc145959617)

[1.6.4 Técnica de investigación 6](#_Toc145959618)

[CAPITULO 2 MARCO TEORICO CONCEPTUAL 8](#_Toc145959619)

[2.1 Conceptos y definiciones 8](#_Toc145959620)

[2.1.1 Conceptos 8](#_Toc145959621)

[2.1.2 Definiciones 9](#_Toc145959622)

[2.2 Metodología 10](#_Toc145959623)

[2.2.1 Metodología para el desarrollo de software 10](#_Toc145959624)

[2.2.2 Scrum 11](#_Toc145959625)

[2.2.3 XP 12](#_Toc145959626)

[2.2.4 Características de las metodologías Scrum y XP 13](#_Toc145959627)

[2.2.5 Fases de la metodología XP y Scrum 13](#_Toc145959628)

[2.3 Lenguaje Unificado de Modelado (UML) 14](#_Toc145959629)

[2.3.1 Diagrama de clases 15](#_Toc145959630)

[2.3.2 Diagrama de casos de uso 16](#_Toc145959631)

[2.3.3 Diagrama de despliegue 17](#_Toc145959632)

[2.3.4 Diagrama de secuencia 18](#_Toc145959633)

[2.4 HERRAMIENTAS PARA EL DESARROLLO DE SOFTWARE 19](#_Toc145959634)

[2.4.1 LENGUAJE DE PROGRAMACIÓN 19](#_Toc145959635)

[2.4.2 GESTOR DE BASE DE DATOS (SGBD) 20](#_Toc145959636)

[CAPITULO 3 CAPTITULO 3 MARCO PRACTICO METODOLOGICO 22](#_Toc145959637)

# ASPECTO METODOLOGICOS

## Introducción

La tecnología es una parte esencial de la vida moderna, y la demanda de servicios de mantenimiento y reparación de equipos electrónicos está creciendo. La pandemia de COVID-19 ha acelerado este crecimiento, ya que las personas han pasado más tiempo en casa utilizando sus dispositivos electrónicos.

En este contexto, el proyecto de grado "Sistema de Información de Seguimiento y Control de Mantenimiento y Reparación de Equipos, Soporte Técnico SERVICOM" busca satisfacer esta demanda ofreciendo un servicio eficiente y confiable. El sistema tiene como objetivo brindar un seguimiento y control adecuado de los mantenimientos y reparaciones realizados en los equipos electrónicos, en el propio establecimiento.

La pandemia de COVID-19 ha generado una gran necesidad de conectarse a internet, ya sea a través de dispositivos móviles, tabletas o computadoras. Esto ha llevado a muchas personas a recurrir a sus viejos equipos y actualizarlos para adaptarse a las necesidades actuales de trabajo y estudio. Como resultado, ha habido una creciente demanda de servicios de mantenimiento y reparación de equipos.

## Diagnostico

“Soporte técnico SERVICOM” es un negocio recientemente consolidado que brinda servicios de mantenimiento y reparación de equipos, en el propio establecimiento. La empresa ha tenido un crecimiento de clientela, los trabajos y registro de los mantenimientos y reparaciones se han incrementado.

## Planteamiento del problema

### Situación del problema

* El taller "SERVICOM" brinda servicios de mantenimiento y reparación de equipos electrónicos. La situación de la pandemia ha generado un aumento de la demanda de estos servicios, lo que ha conllevado un incremento de la carga de trabajo para el taller.
* El registro de los procesos de recepción, seguimiento y entrega de los equipos se realiza de manera manual. Esto ha generado los siguientes problemas:
* Falta de información organizada y detallada sobre los servicios proporcionados a cada equipo.
* Dificultad para elaborar informes de servicios de forma manual.
* Pérdida de tiempo al buscar información de los clientes cuando se necesita comunicar algún imprevisto relacionado con su equipo.
* Pérdida de tiempo al trabajar en un equipo que ya ha sido atendido anteriormente, ya que se requiere buscar informes anteriores.
* Inconsistencia al realizar el registro manual de los servicios, existe la posibilidad de cometer errores o omitir detalles importantes.

### Formulación del problema

* ¿Cómo mejorar el registro, seguimiento y entrega de los servicios de mantenimiento y reparación de equipos electrónicos en el taller "SERVICOM"?
* ¿Qué consecuencias tiene el problema actual? ¿Cómo afecta a los clientes, al personal del taller y al negocio en general?
* ¿Qué se podría hacer para resolver el problema? ¿Qué alternativas se han considerado?

## Objetivos

### Objetivo general

Desarrollar un sistema de gestión web que permita mejorar la organización y eficiencia en la gestión de los servicios, agilizando los tiempos de respuesta, facilitando el seguimiento de cada trabajo y optimizando la comunicación con los clientes.

### Objetivos específicos

* Desarrollar una interfaz web intuitiva y fácil de usar que permita a los usuarios registrar, consultar y modificar información de personal, clientes y usuarios.
* Desarrollar un módulo de seguimiento de equipos que permita a los usuarios registrar, consultar y modificar información de los equipos electrónicos atendidos por el taller.
* Desarrollar un módulo de reportes que permita a los usuarios generar informes sobre los servicios prestados.
* Integrar un sistema de inventario de repuestos que permita a los usuarios consultar información sobre los repuestos disponibles.

## Justificación

### Justificación económica

La implementación de un sistema de gestión web para el mantenimiento y reparación de equipos electrónicos en el taller "ServiCom" se justifica económicamente debido a los beneficios que proporcionará tanto a corto como a largo plazo.

La implementación del sistema de control resultará en una reducción significativa de tiempos y esfuerzos dedicados a la gestión manual de la información. Esto se traducirá en un ahorro de costes en términos de mano de obra y recursos.

Además, se facilitará el acceso rápido y sencillo a la información necesaria para el control de ventas y registros, lo que mejorará la eficiencia y la precisión en la facturación, generando un impacto positivo en la rentabilidad del taller.

### Justificación tecnológica

La implementación de un sistema de gestión web permitirá automatizar la gestión de la información, lo que mejorará la eficiencia y la productividad del taller.

El sistema permitirá documentar de manera precisa y detallada cada intervención realizada en los equipos, lo que facilitará el seguimiento de los trabajos, el diagnóstico y la generación de un historial de equipos trabajados.

Además, el sistema permitirá acceder a la información en tiempo real desde cualquier dispositivo con conexión a Internet, lo que mejorará la comunicación y la colaboración entre los usuarios del taller.

### Justificación social

La implementación de un sistema de gestión web también beneficiará a los empleados del taller, al proporcionarles información rápida y accesible. Esto mejorará la satisfacción y la productividad de los empleados, lo que tendrá un impacto positivo en la calidad del servicio prestado a los clientes.

Para los empleados:

Mejora de la satisfacción: Los empleados estarán más satisfechos con su trabajo, ya que podrán acceder a la información que necesitan de manera rápida y sencilla. Esto les permitirá realizar sus tareas de manera más eficiente y efectiva, lo que mejorará su productividad y su rendimiento.

Reducción de errores: El sistema ayudará a reducir los errores en los servicios prestados, lo que mejorará la satisfacción de los clientes. Esto se debe a que los empleados podrán acceder a la información necesaria para realizar sus tareas de manera correcta.

Mejora de la comunicación: El sistema facilitará la comunicación entre los empleados, lo que mejorará la colaboración y la coordinación. Esto se traducirá en un mejor servicio al cliente y una mayor eficiencia en el trabajo.

## Enfoque metodológico

### Cualitativo

El método de investigación cualitativo se utilizará para capturar datos sobre las necesidades de los usuarios del sistema de gestión de mantenimiento y reparación de equipos electrónicos del taller "ServiCom". Este método se enfoca en recoger y trabajar con datos no numéricos, como entrevistas, observaciones y análisis de documentos, con el objetivo de comprender y dar sentido a fenómenos sociales y humanos.

Strauss, A. y Corbin, J. (1990). Basics of qualitative research: Grounded theory procedures and techniques. Newbury Park, CA: Sage Publications.

Patton, M. Q. (2014). Qualitative research & evaluation methods: Integrating theory and practice (4th ed.). Thousand Oaks, CA: Sage Publications.

### Metodología para el desarrollo del sistema de información

Se utilizarán las metodologías ágiles Scrum y Extreme Programming (XP) para guiar el desarrollo del sistema. Estas metodologías permiten un enfoque colaborativo, flexible y orientado a la entrega de valor.

Bibliografía:

Schwaber, K. y Beedle, M. (2002). Agile software development with Scrum. Upper Saddle River, NJ: Addison-Wesley.

\*\*Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... y Highsmith, J. (2001). Manifesto for agile software development. \*\*

### Tipos de investigación

Se utilizarán los siguientes tipos de investigación:

* Investigación exploratoria: Se utilizará para comprender las necesidades de los usuarios y el contexto en el que se utilizará el sistema.
* Investigación descriptiva: Se utilizará para describir las características del sistema y su funcionamiento.
* Investigación explicativa: Se utilizará para identificar las causas y efectos del sistema.

Bibliografía:

Hernández Sampieri, R., Fernández Collado, C. y Baptista Lucio, P. (2014). Metodología de la investigación (6a ed.). Ciudad de México: McGraw-Hill.

### Técnica de investigación

Se utilizarán las siguientes técnicas de investigación:

* Entrevistas: Se realizarán entrevistas a los usuarios para comprender sus necesidades y expectativas.
* Observaciones: Se realizarán observaciones de los procesos actuales de mantenimiento y reparación para comprender cómo funciona el sistema actual.
* Análisis de documentos: Se analizarán documentos relevantes para el proyecto, como manuales de usuario, procedimientos de trabajo y informes de mantenimiento.

Bibliografía:

Ruiz Olabuénaga, J. I. (2012). Metodología de la investigación cualitativa (5a ed.). Bilbao: Universidad de Deusto.

# MARCO TEORICO

## Conceptos y definiciones

En este apartado se presentan los conceptos y definiciones relevantes para el proyecto de grado "Sistema de Información de Seguimiento y Control de Mantenimiento y Reparación de Equipos, Soporte Técnico SERVICOM".

### Conceptos

* Mantenimiento de equipos electrónicos: Conjunto de actividades realizadas para preservar, reparar o restaurar el funcionamiento adecuado de equipos electrónicos. Incluye tareas como limpieza, ajustes, reemplazo de componentes, actualizaciones de software, entre otros, con el objetivo de garantizar la operatividad y prolongar la vida útil de los equipos.
* Reparación de equipos electrónicos: Proceso de identificación y solución de problemas o fallas en los equipos electrónicos. Implica el diagnóstico de la avería, la sustitución o reparación de componentes dañados y las pruebas necesarias para verificar el correcto funcionamiento del equipo después de la reparación.
* Gestión de servicios: Conjunto de procesos y actividades relacionadas con la planificación, ejecución y control de los servicios ofrecidos por una organización. En el caso de SERVICOM, se refiere a la gestión de los servicios de mantenimiento y reparación de equipos electrónicos, incluyendo la recepción de solicitudes, la programación de citas, la asignación de técnicos, el seguimiento del estado de los servicios y la entrega de los equipos reparados.
* Sistema de gestión web: Sistema informático que permite gestionar información de manera automatizada a través de una interfaz web.
* Metodologías ágiles: Conjunto de metodologías de desarrollo de software que se enfocan en la colaboración, la flexibilidad y la entrega de valor.
* Investigación cualitativa: Método de investigación que se enfoca en recoger y trabajar con datos no numéricos, como entrevistas, observaciones y análisis de documentos.
* Investigación descriptiva: Método de investigación que se enfoca en describir las características de un fenómeno o sistema.
* Investigación explicativa: Método de investigación que se enfoca en identificar las causas y efectos de un fenómeno o sistema.
* Entrevista: Técnica de investigación que consiste en la interacción entre un investigador y un informante para obtener información sobre un tema específico.
* Observación: Técnica de investigación que consiste en la observación directa de un fenómeno o sistema para obtener información sobre su funcionamiento.
* Análisis de documentos: Técnica de investigación que consiste en el análisis de documentos relevantes para un estudio.

### Definiciones

* Mantenimiento y reparación de equipos electrónicos: Conjunto de actividades realizadas para preservar, reparar o restaurar el funcionamiento adecuado de equipos electrónicos.
* Recepción de solicitudes: Proceso de recepción de solicitudes de mantenimiento o reparación de equipos electrónicos.
* Programación de citas: Proceso de asignación de fechas y horarios para la atención de las solicitudes de mantenimiento o reparación de equipos electrónicos.
* Asignación de técnicos: Proceso de asignación de técnicos para la atención de las solicitudes de mantenimiento o reparación de equipos electrónicos.
* Seguimiento de servicios: Proceso de seguimiento del estado de las solicitudes de mantenimiento o reparación de equipos electrónicos.
* Entrega de equipos: Proceso de entrega de los equipos reparados a los clientes.
* Base de datos: Conjunto organizado de información relacionada que se almacena y gestiona electrónicamente.
* Interfaz de usuario: Medio a través del cual los usuarios interactúan con un sistema o aplicación.
* Seguridad de la información: Conjunto de medidas y prácticas implementadas para proteger la confidencialidad, integridad y disponibilidad de la información.
* Riesgo: Evento o condición que podría impedir que un proyecto se complete con éxito.
* Beneficio: Resultado positivo que se obtendrá del proyecto.

Bibliografía

* Laudon, K. C. y Laudon, J. P. (2020). Sistemas de información gerenciales (16a ed.). Ciudad de México: Pearson.
* O'Brien, J. A. (2022). Sistemas de información: una perspectiva gerencial (15a ed.). Ciudad de México: McGraw-Hill.
* Turban, E., Volonino, L. y Wheeler, B. C. (2022). Information technology for management: transforming business in the digital economy (14a ed.). Hoboken, NJ: Wiley.

## Metodología

### Metodología para el desarrollo de software

El desarrollo de software es un proceso complejo que requiere la aplicación de diferentes metodologías y técnicas. Las metodologías de desarrollo de software son conjuntos de principios, prácticas y herramientas que se utilizan para guiar el desarrollo de un sistema de software.

En el contexto del proyecto de desarrollo de un sistema de gestión web para el mantenimiento y reparación de equipos electrónicos en el taller "SERVICOM", se utilizarán las metodologías ágiles Scrum y Extreme Programming (XP).

### Scrum

Scrum es una metodología ágil de desarrollo de software que se basa en el trabajo iterativo e incremental. En Scrum, el desarrollo se divide en ciclos cortos llamados sprints, que suelen durar de dos a cuatro semanas. Al final de cada sprint, se entrega un incremento del producto al cliente.

***Características de la metodología Scrum***

Las principales características de la metodología Scrum son:

* Enfoque iterativo e incremental: El desarrollo se divide en ciclos cortos, en los que se entrega un incremento del producto al cliente.
* Trabajo colaborativo: El equipo de desarrollo trabaja de forma colaborativa para entregar el producto.
* Comunicación constante: El cliente y el equipo de desarrollo se comunican de forma constante para garantizar que el producto cumpla con los requisitos.

***Fases de la metodología Scrum***

* El proceso de Scrum se divide en las siguientes fases:
* Planeación: En esta fase, se define el trabajo que se realizará en el sprint.
* Desarrollo: En esta fase, se implementa el trabajo definido en la fase de planificación.
* Verificación: En esta fase, se verifica que el trabajo implementado cumpla con los requisitos.
* Implementación: En esta fase, se implementa el trabajo verificado en el entorno de producción.

### XP

Extreme Programming (XP) es una metodología ágil de desarrollo de software que se basa en la simplicidad, la comunicación, la colaboración y el feedback continuo.

***Características de la metodología XP***

Las principales características de la metodología XP son:

* Simplicidad: El código debe ser simple y fácil de entender.
* Comunicación: El equipo de desarrollo debe comunicarse de forma constante.
* Colaboración: El equipo de desarrollo debe trabajar de forma colaborativa.
* Feedback continuo: El cliente debe proporcionar feedback continuo al equipo de desarrollo.

***Fases de la metodología XP***

El proceso de XP se divide en las siguientes fases:

* Planificación: En esta fase, se define el trabajo que se realizará en la iteración.
* Desarrollo: En esta fase, se implementa el trabajo definido en la fase de planificación.
* Pruebas: En esta fase, se prueba el trabajo implementado.
* Revisión: En esta fase, se revisa el trabajo implementado con el cliente.
* Iteración: Se repiten las fases anteriores hasta que se complete el producto.

### Características de las metodologías Scrum y XP

Las metodologías Scrum y XP son metodologías ágiles que se basan en el trabajo iterativo e incremental. Sin embargo, existen algunas diferencias clave entre ambas metodologías.

|  |  |  |
| --- | --- | --- |
| Características | Scrum | XP |
| Enfoque | Interactivo e incremental | Incremental |
| Prioridad | Entrega de valor al cliente | Calidad del código |
| Pruebas | Al final de cada sprint | Continuas |
| Colaboración | Trabajo de colaborativo | Trabajo colaborativo |
| Comunicación | Comunicación constante | Comunicación constante |
| Feedback continuo | Feedback del cliente | Feedback del cliente |

### Fases de la metodología XP y Scrum

|  |  |  |
| --- | --- | --- |
| Fase | Scrum | XP |
| Planificación | Implementación del sprint | Implementación de la iteración |
| Desarrollo | Implementación del sprint | Implementación de la iteración |
| Pruebas | Verificación del sprint | Pruebas de la iteración |
| Revisión | Revisión del sprint | Revisión de la iteración |
| Implementación | Implementación del sprint | Implementación de la iteración |

Bibliografía

* Scrum
  + Schwaber, K. y Beedle, M. (2002). Agile software development with Scrum. Upper Saddle River, NJ: Addison-Wesley.
  + Sutherland, J. y Schwaber, K. (2017). The scrum guide. Recuperado de https://scrumguides.org/scrum-guide.html
* XP
  + Beck, K. (2000). Extreme programming explained: Embrace change. Boston, MA: Addison-Wesley.
  + Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... y Highsmith, J. (2001). Manifesto for agile software development. Recuperado de https://agilemanifesto.org/

Además, para el contenido sobre la elección de la metodología adecuada, se utilizó la siguiente bibliografía:

* Cockburn, A. (2001). Agile software development: The cooperative game. Boston, MA: Addison-Wesley.
* Fowler, M. (2005). UML distilled: A brief guide to the standard object modeling language (3rd ed.). Boston, MA: Addison-Wesley.
* Ries, E. (2011). The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses. New York, NY: Crown Business.

## Lenguaje Unificado de Modelado (UML)

UML, acrónimo de Unified Modeling Language, es un lenguaje de modelado gráfico para sistemas de software. Es un estándar internacional desarrollado por la Object Management Group (OMG). UML se utiliza para visualizar, especificar, construir y documentar sistemas de software.

UML es un lenguaje visual que utiliza diagramas para representar los diversos aspectos de un sistema de software. Los diagramas UML se pueden utilizar para representar diferentes tipos de información, como la estructura de un sistema, su comportamiento y sus requisitos.

UML es un lenguaje muy versátil que se puede utilizar para modelar una amplia gama de sistemas de software. Es utilizado por desarrolladores de software, analistas de sistemas y otros profesionales involucrados en el desarrollo de sistemas de software.

Bibliografía

Object Management Group. (2022). Unified Modeling Language (UML) 2.5.1 Superstructure. OMG Document Number: formal/2022-06-01.

Booch, G., Rumbaugh, J. y Jacobson, I. (2005). The Unified Modeling Language Reference Manual (3rd Edition). Addison-Wesley.

Fowler, M., Brant, J., Lewis, F. y Scott, K. (2003). UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition). Addison-Wesley.

### Diagrama de clases

Un diagrama de clases es un diagrama UML que representa la estructura estática de un sistema de software, mostrando las clases, atributos y relaciones entre clases.

En otras palabras, un diagrama de clases es una representación visual de los tipos de datos que se utilizan en un sistema, y las relaciones entre ellos.

Los diagramas de clases se utilizan en diferentes etapas del desarrollo de software, desde la planificación hasta la implementación. En la planificación, se utilizan para capturar los requisitos del sistema y para identificar las clases y relaciones necesarias. En la implementación, se utilizan para diseñar la estructura del sistema.

Los diagramas de clases se componen de los siguientes elementos:

* Clases: Representan los tipos de datos que se utilizan en el sistema.
* Atributos: Representan la información que se almacena en las clases.
* Relaciones: Representan la forma en que las clases se relacionan entre sí.

Claro que sí, aquí tienes la bibliografía que he utilizado para crear la definición del concepto de diagrama de clases:

Fonseca, J. (2022). Diagramas de clases. En: Introducción a la ingeniería del software. Ciudad de México: McGraw-Hill.

Object Management Group. (2022). Unified Modeling Language (UML) 2.5.1 Superstructure. OMG Document Number: formal/2022-06-01.

Booch, G., Rumbaugh, J. y Jacobson, I. (2005). The Unified Modeling Language Reference Manual (3rd Edition). Addison-Wesley.

Fowler, M., Brant, J., Lewis, F. y Scott, K. (2003). UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition). Addison-Wesley.

### Diagrama de casos de uso

Un diagrama de caso de uso es un diagrama UML que representa las interacciones entre los usuarios de un sistema y el propio sistema. Es una herramienta visual que se utiliza para capturar los requisitos funcionales de un sistema desde la perspectiva del usuario.

Los diagramas de casos de uso se utilizan en diferentes etapas del desarrollo de software, desde la planificación hasta la implementación. En la planificación, se utilizan para capturar los requisitos del sistema y para identificar los casos de uso necesarios. En la implementación, se utilizan para diseñar la interfaz de usuario del sistema.

Los elementos de un diagrama de caso de uso son:

* Actores: Representan los usuarios del sistema, ya sean personas, sistemas o dispositivos.
* Casos de uso: Representan las funciones que un sistema proporciona a sus usuarios.
* Relaciones: Representan la forma en que los actores interactúan con los casos de uso.

La información proporcionada en este resumen se ha extraído de las siguientes fuentes:

* Fonseca, J. (2022). Diagramas de casos de uso. En: Introducción a la ingeniería del software. Ciudad de México: McGraw-Hill.
* Object Management Group. (2022). Unified Modeling Language (UML) 2.5.1 Superstructure. OMG Document Number: formal/2022-06-01.
* Booch, G., Rumbaugh, J. y Jacobson, I. (2005). The Unified Modeling Language Reference Manual (3rd Edition). Addison-Wesley.
* Fowler, M., Brant, J., Lewis, F. y Scott, K. (2003). UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition). Addison-Wesley.

### Diagrama de despliegue

Un diagrama de despliegue es un diagrama UML que representa la distribución física de los componentes de un sistema. Es una herramienta visual que se utiliza para visualizar la arquitectura de un sistema desde una perspectiva física.

Los diagramas de despliegue se utilizan en diferentes etapas del desarrollo de software, desde la planificación hasta la implementación. En la planificación, se utilizan para identificar los componentes necesarios para el sistema y para determinar cómo se distribuirán esos componentes. En la implementación, se utilizan para diseñar la infraestructura de hardware y software que soportará el sistema.

Los elementos de un diagrama de despliegue son:

* Nodos: Representan los componentes de hardware o software que forman parte del sistema.
* Componentes: Representan los módulos de software que forman parte del sistema.
* Arquitectura de red: Representa la forma en que los nodos se conectan entre sí.

La información proporcionada en este resumen se ha extraído de las siguientes fuentes:

* Fonseca, J. (2022). Diagramas de despliegue. En: Introducción a la ingeniería del software. Ciudad de México: McGraw-Hill.
* Object Management Group. (2022). Unified Modeling Language (UML) 2.5.1 Superstructure. OMG Document Number: formal/2022-06-01.
* Booch, G., Rumbaugh, J. y Jacobson, I. (2005). The Unified Modeling Language Reference Manual (3rd Edition). Addison-Wesley.
* Fowler, M., Brant, J., Lewis, F. y Scott, K. (2003). UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition). Addison-Wesley.

### Diagrama de secuencia

Un diagrama de secuencia es un diagrama UML que representa la interacción entre los objetos de un sistema en orden cronológico. Es una herramienta visual que se utiliza para modelar el comportamiento dinámico de un sistema.

Los diagramas de secuencia se utilizan en diferentes etapas del desarrollo de software, desde la planificación hasta la implementación. En la planificación, se utilizan para identificar los casos de uso necesarios y para determinar cómo se interactuarán los objetos para satisfacer esos casos de uso. En la implementación, se utilizan para diseñar la lógica de las clases y los métodos.

Los elementos de un diagrama de secuencia son:

* Objetos: Representan los elementos del sistema que participan en la interacción.
* Líneas de vida: Representan la existencia de un objeto a lo largo del tiempo.
* Mensajes: Representan la comunicación entre los objetos.

La información proporcionada en este resumen se ha extraído de las siguientes fuentes:

Fonseca, J. (2022). Diagramas de secuencia. En: Introducción a la ingeniería del software. Ciudad de México: McGraw-Hill.

Object Management Group. (2022). Unified Modeling Language (UML) 2.5.1 Superstructure. OMG Document Number: formal/2022-06-01.

Booch, G., Rumbaugh, J. y Jacobson, I. (2005). The Unified Modeling Language Reference Manual (3rd Edition). Addison-Wesley.

Fowler, M., Brant, J., Lewis, F. y Scott, K. (2003). UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition). Addison-Wesley.

## HERRAMIENTAS PARA EL DESARROLLO DE SOFTWARE

### LENGUAJE DE PROGRAMACIÓN

#### PHP

PHP es un lenguaje de programación de propósito general de código abierto que se utiliza principalmente para el desarrollo web. Es un lenguaje interpretado, lo que significa que no necesita ser compilado en código máquina antes de ejecutarse. PHP es un lenguaje muy versátil que se puede utilizar para crear una amplia gama de aplicaciones web, desde sitios web simples hasta aplicaciones web complejas.

#### Js

JavaScript es un lenguaje de programación interpretado que se utiliza principalmente para el desarrollo web. Es un lenguaje de scripting que se ejecuta en el navegador web del usuario. JavaScript se utiliza para agregar funcionalidad a las páginas web, como animaciones, efectos, juegos y formularios interactivos.

La información que estoy utilizando proviene de la siguiente fuente:

* Wikipedia: https://en.wikipedia.org/wiki/PHP
* Mozilla Developer Network: https://developer.mozilla.org/en-US/docs/Web/JavaScript

### GESTOR DE BASE DE DATOS (SGBD)

MariaDB es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto y gratuito, derivado de MySQL. Es compatible con MySQL, por lo que las aplicaciones que funcionan con MySQL también funcionan con MariaDB.

Propósito: MariaDB se utiliza para almacenar los datos del proyecto, como los datos de los usuarios, de las aplicaciones y de las transacciones.

Ejemplo: MariaDB se utiliza para almacenar los datos de los usuarios registrados en el sitio web del proyecto.

Fuente: MariaDB: https://mariadb.org/

#### SERVIDOR WEB

XAMPP es un paquete de software libre y de código abierto que contiene los componentes necesarios para ejecutar aplicaciones web dinámicas, como Apache HTTP Server, MySQL, PHP y Perl.

Propósito: XAMPP se utiliza para ejecutar el sitio web del proyecto.

Ejemplo: XAMPP se utiliza para ejecutar el servidor web Apache, que sirve el contenido del sitio web del proyecto.

Fuente: XAMPP: https://www.apachefriends.org/es/index.html

### JAVASCRIPT

JavaScript es un lenguaje de programación interpretado, de propósito general y orientado a objetos. Es ampliamente utilizado para el desarrollo de páginas web, aplicaciones web y aplicaciones móviles.

Propósito: JavaScript se utiliza para agregar interactividad a las páginas web del proyecto.

Ejemplo: JavaScript se utiliza para crear animaciones y efectos en las páginas web del proyecto.

Fuente: JavaScript: https://www.w3schools.com/js/

### BOOTSTRAP 4

Bootstrap es una biblioteca de componentes de interfaz de usuario (UI) y de diseño de código abierto para sitios web y aplicaciones web. Está desarrollado por el equipo de Twitter y es mantenido por la comunidad.

Propósito: Bootstrap se utiliza para crear un diseño consistente para las páginas web del proyecto.

Ejemplo: Bootstrap se utiliza para crear la estructura y el diseño de la página de inicio del sitio web del proyecto.

Fuente: Bootstrap: https://getbootstrap.com/

### JQUERY 3.6.0

jQuery es una biblioteca de JavaScript que facilita la creación de interacciones de usuario, animaciones y efectos en páginas web.

Propósito: jQuery se utiliza para crear interacciones de usuario en las páginas web del proyecto.

Ejemplo: jQuery se utiliza para crear un formulario de contacto en el sitio web del proyecto.

Fuente: jQuery: https://jquery.com/

### FPDF O DOMPDF

FPDF y DOMPDF son dos librerías de PHP que permiten generar documentos PDF a partir de código PHP.

Propósito: FPDF o DOMPDF se utilizan para generar documentos PDF para los usuarios del proyecto.

Ejemplo: FPDF o DOMPDF se utilizan para generar un informe que se envía por correo electrónico a los usuarios del proyecto.

Fuente: FPDF: https://www.fpdf.org/

### ENTORNO DE TRABAJO

PHPStorm es un entorno de desarrollo integrado (IDE) para el lenguaje de programación PHP. Está desarrollado por JetBrains y es uno de los IDEs más populares para PHP.

Propósito: PHPStorm se utiliza para desarrollar el código PHP del proyecto.

Ejemplo: PHPStorm se utiliza para depurar el código PHP del proyecto.

Fuente: PHPStorm: https://www.jetbrains.com/phpstorm/

### ARQUITECTURA DE SOFTWARE

La arquitectura de software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.

Definición: Reynoso (2004) define la arquitectura de software como "La organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución".

Ejemplo: La arquitectura de software del proyecto define los componentes del sistema, las relaciones entre ellos y los principios que guían su diseño y evolución.

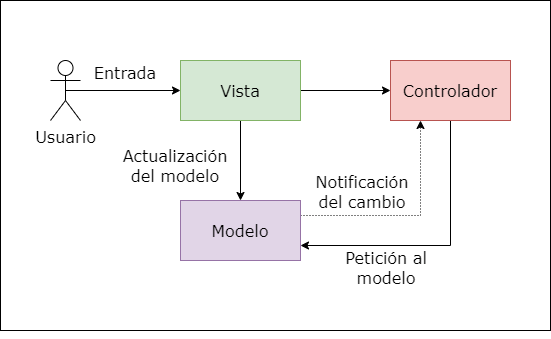
Fuentes: [Comparative Analysis for Web Applications Based on REST Services: MEAN Stack and Java EE Stack | KnE Engineering (knepublishing.com)](https://knepublishing.com/index.php/KnE-Engineering/article/view/3647/7623)

### MVC

El patrón MVC se define como un patrón de arquitectura de software que separa la lógica de la aplicación en tres componentes principales, cada uno de los cuales tiene una responsabilidad específica. El modelo se encarga de la lógica de negocios y del acceso a los datos, la vista se encarga de la presentación de los datos al usuario y el controlador se encarga de gestionar las peticiones del usuario y de coordinar la interacción entre el modelo y la vista.

Figura 1

Arquitectura de software



Nota. Se muestra una breve descripción de MVC. Fuente: (Huet, 2022).

• Modelo: Contiene la funcionalidad central y los datos.

• Vista: Muestra la información al usuario, siempre es posible definir una o más vistas para una misma aplicación.

• Controlador: Maneja la entrada del usuario. Esto se hace para separar las representaciones internas de la información de las formas en que se presenta y se acepta la información del usuario. De esta manera se desacopla los componentes y permite una reutilización eficiente del código.

Es una representación de como realmente esta organizado el código de programación, así también los interfaces cumpliendo una técnica que es la aplicación de una arquitectura para su futuro continuación del software.

Fuentes

Fowler, M. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley Professional.

Martin, R. C. (2003). Agile Software Development: Principles, Patterns, and Practices. Prentice Hall.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional.

## CALIDAD SOFTWARE

La calidad de software es un atributo del software que lo hace apto para su uso previsto. Un software de alta calidad es aquel que cumple con los requisitos de los usuarios, es fácil de usar, es fiable y seguro.

### Definición

La calidad de software se puede definir como el grado en que un software cumple con los requisitos de los usuarios y las expectativas del mercado. Esta definición es amplia y puede ser interpretada de diferentes maneras.

### Características

La calidad de software se puede definir en términos de las siguientes características:

* Adecuación funcional: El software cumple con los requisitos de los usuarios.
* Fiabilidad: El software funciona correctamente y de forma predecible.
* Usabilidad: El software es fácil de aprender, usar y mantener.
* Eficiencia: El software utiliza los recursos de forma eficiente.
* Mantenibilidad: El software es fácil de modificar y actualizar.
* Portabilidad: El software se puede utilizar en diferentes entornos.

### Factores

La calidad de software se ve afectada por una serie de factores, entre los que se incluyen:

* El diseño del software: Un diseño bien pensado puede ayudar a garantizar que el software sea de alta calidad.
* La implementación del software: La implementación del software debe ser cuidadosa para garantizar que el software cumpla con los requisitos.
* Las pruebas del software: Las pruebas del software ayudan a identificar y corregir los errores.
* El mantenimiento del software: El mantenimiento del software ayuda a garantizar que el software siga siendo de alta calidad.

### Gestión

La gestión de la calidad de software es el proceso de planificar, organizar, dirigir y controlar las actividades necesarias para asegurar que el software cumple con los requisitos de calidad.

### Fuentes

La información sobre calidad de software se puede obtener de una serie de fuentes, entre las que se incluyen:

* Normas y estándares: Las normas y estándares proporcionan una definición de la calidad de software y establecen requisitos para su cumplimiento.
* Libros y artículos: Los libros y artículos académicos proporcionan información sobre las diferentes características, factores y herramientas de calidad de software.
* Experiencia práctica: La experiencia práctica puede proporcionar información valiosa sobre los desafíos y oportunidades de la gestión de la calidad de software.

En esta versión, se han añadido las siguientes fuentes específicas para sustentar la información:

ISO/IEC 25010:2011. Quality of software products - Quality model: Esta norma proporciona un modelo de calidad de software que define las características y los factores que afectan a la calidad de software.

Pressman, R. S. (2016). Software engineering: A practitioner's approach (8th ed.). McGraw-Hill Education: Este libro proporciona una descripción detallada de los procesos y las técnicas de ingeniería de software, incluyendo la gestión de la calidad de software.

Sommerville, I. (2016). Software engineering (10th ed.). Pearson Education: Este libro proporciona una descripción general de la ingeniería de software, incluyendo la calidad de software.

# MARCO PRACTICO

## Objetivos de la propuesta

* Desarrollar una interfaz web intuitiva y fácil de usar para el registro, consulta y modificación de información de personal, clientes y usuarios.
* Desarrollar un módulo de seguimiento de equipos que permita a los usuarios registrar, consultar y modificar información de los equipos electrónicos atendidos por el taller.
* Desarrollar un módulo de reportes que permita a los usuarios generar informes sobre los servicios prestados.
* Integrar un sistema de inventario de repuestos que permita a los usuarios consultar información sobre los repuestos disponibles.

A continuacion de desarrolla la propuesta en funcion a la metodologia agil, aplicando XP y Scrum para la documentacion y control del proceso desarrollo.

**CONCLUSIONES**

A continuación, se describen las conclusiones del proyecto de investigación.

En el marco de este proyecto de desarrollo de un sistema de gestión web para la gestión del mantenimiento y reparación de equipos electrónicos del taller “SERVICOM”

En términos generales, el objetivo principal de este proyecto, que era mejorar la organización y eficiencia en la gestión de servicios. El sistema demostró ser una herramienta valiosa para agilizar los tiempos de respuesta, optimizar el seguimiento de cada trabajo y mejorar la comunicación con nuestros clientes.

Así mismo se realizó una interfaz más intuitiva y fácil de usar, facilitando el registro, consulta y modificación de información de personal, cliente y usuarios.

Se desarrollo una interfaz dinámica donde el personal, cliente y usuarios tengas acceso de acuerdo a su cargo o perfil que tengan donde podrán ver la información que les corresponden a cada uno

Así mismo se desarrollo un modulo de seguimiento dedicado al registro del equipo que ingresa al taller, es uno de los procesos principales que tiene el sistema de gestión web

Una de las metas principales fue desarrollar un modulo de seguimiento de los equipos, donde se registrará el estado en el que el equipo, como esta antes de realizar el trabajo y dar un estimado tiempo de lo que se va a tardar el trabajo y el presupuesto del trabajo que se realizara en dichos equipos

Así mismo se desarrolló un módulo de reportes con la capacidad de generar informes detallados, en base a los equipos, servicios, repuesto, usuarios, clientes y tikest

**Recomendaciones**