

# UNIVERSIDAD PÚBLICA DE EL ALTO

## CARRERA INGENIERÍA DE SISTEMAS



### PROYECTO DE GRADO

#### **“SISTEMA DE REGISTRO Y CONTROL PARA EL PAGO DEL BONO A PERSONAS CON DISCAPACIDAD CASO: MUNICIPIO DE TIAHUANACU”**

Para optar al título de Licenciatura en Ingeniería de Sistemas

**Mención: Informática y Telecomunicaciones**

**Postulante: Marcos Juan Zuazo Caso**

**Tutor Metodológico: M.Sc. Ing. Wilfredo Pérez Aquino**

**Tutor Especialista: Lic. Freddy Salgueiro Trujillo**

**Tutor Revisor: Ing. Gabriel Sirpa Huayhua**

**EL ALTO – BOLIVIA**

**2019**

## DEDICATORIA

A Dios, por permitirme llegar hasta esta instancia y poder tener la oportunidad de demostrar que uno a pesar de las adversidades uno puede llegar a sus metas u objetivos, gracias Dios por tu bondad y tu infinito amor.

A mis padres Avi Caso y Juan Zuazo, por su apoyo en todas las etapas de mi vida demostrándome que el trabajo y sacrificio como ejemplo y filosofía de vida son los valores más importantes que me pudieron enseñar.

## AGRADECIMIENTO

Ante todo a Dios la vida es la oportunidad que todo ser humano tiene para demostrar que con voluntad y perseverancia puede alcanzar sus objetivos y metas.

A mi tutor metodológico MSc. Wilfredo Perez Aquino por sus enseñanzas, comentarios y observaciones que me permiten culminar el presente trabajo.

A mi asesor Lic. Freddy Salgueiro Trujillo por guiarme, corregirme y las ideas propuestas hoy implementadas en el presente trabajo.

A mi tutor revisor Ing. Reynaldo Sirpa Huayhua por sus correcciones, la confianza que se ve hoy en trabajo culminado con éxito.

En especial a la Dr. Olga Poma directora SLIM-DNA del municipio de Tiahuanacu por brindarme la oportunidad y confianza, del sistema hoy en día implementado con mucho éxito en dicha unidad.

## Resumen

Hoy en día el gobierno central dispone de muchos software o sistemas para la administración y gestión de sus actividades, pero no así todos los municipios del departamento de La Paz. El presente proyecto se propuso para el gobierno central autónomo municipal de Tiahuanacu dentro de la unidad del servicio legal integral municipal y defensoría de la niñez y adolescencia que brinda asesoramiento legal, psicológico y social a personas víctimas de violencia psicológica, familiar o física. El sistema que se propuso tiene como objetivos el centralizar información generada, también de hacer seguimiento de los casos que tengan que ver las personas con discapacidad y de hacer seguimiento de sus documentaciones, generar planillas y recibos de pago del bono a personas con discapacidad, dicho proyecto se trabajó con la metodología uwe, scrum y mysql para el gestor de base de datos.

## INDICE DE CONTENIDO

|  |    |
|--|----|
| CAPITULO I .....                                 | 1  |
| 1. MARCO PRELIMINAR.....                         | 1  |
| 1.1. INTRODUCCIÓN .....                          | 1  |
| 1.2. ANTECEDENTES.....                           | 3  |
| 1.2.1. Antecedentes de la institución.....       | 3  |
| 1.2.2. Antecedentes de proyectos similares ..... | 3  |
| 1.3. PLANTEAMIENTO DEL PROBLEMA .....            | 4  |
| 1.3.1. Problema principal .....                  | 4  |
| 1.3.2. Problema secundario.....                  | 4  |
| 1.3.3. Formulación del problema .....            | 5  |
| 1.4. OBJETIVOS .....                             | 6  |
| 1.4.1. Objetivo general .....                    | 6  |
| 1.4.2. Objetivos específicos .....               | 6  |
| 1.5. JUSTIFICACIÓN .....                         | 6  |
| 1.5.1. Justificación tecnológica .....           | 6  |
| 1.5.2. Justificación técnica .....               | 7  |
| 1.5.3. Justificación económica .....             | 8  |
| 1.5.4. Justificación social .....                | 9  |
| 1.6. METODOLOGIA .....                           | 10 |

|                  |  |    |
|------------------|--|----|
| 1.6.2.           | Método de ingeniería web.....                        | 11 |
| 1.8.             | LIMITES Y ALCANCES.....                              | 14 |
| 1.8.1.           | Limites.....   | 14 |
| 1.9.             | APORTES.....   | 16 |
| CAPITULO II..... |  | 17 |
| 2.               | MARCO TEORICO.....                                   | 17 |
| 2.1.             | SISTEMA.....   | 17 |
| 2.2.             | DISCAPACIDAD.....                                    | 18 |
| 2.3.             | PERSONAS CON DISCAPACIDAD.....                       | 19 |
| 2.4.             | MARCO LEGAL LEY 977.....                             | 19 |
| 2.4.1.           | Objetivo de la ley 977.....                          | 19 |
| 2.5.             | MODELOS AGILES.....                                  | 24 |
| 2.6.             | ANTECEDENTES.....                                    | 25 |
| 2.7.             | METODOLOGÍA DE DESARROLLO DE APLICACIONES WEB.....   | 26 |
| 2.7.1.           | Metodología de gestión de relaciones RMM.....        | 26 |
| 2.7.2.           | Método de diseño hipermedia orientado a objetos..... | 27 |
| 2.8.             | INGENIERÍA WEB.....                                  | 29 |
| 2.8.1            | UML.....   | 29 |
| 2.8.2.           | UML – Based web engineering (UWE).....               | 31 |
| 2.8.3.           | Tipos de diagrama en UML.....                        | 32 |

|   |    |
|---|----|
| 2.8.3.1. Modelo de caso de uso.....                   | 32 |
| 2.8.3.2 Diagramas de casos de uso .....               | 33 |
| 2.8.3.3. Diagrama de clases .....                     | 34 |
| 2.8.3.4. Modelo de contenido .....                    | 35 |
| 2.8.3.5. Diagrama de objetos .....                    | 36 |
| 2.8.3.6. Modelo de navegación .....                   | 37 |
| 2.8.3.7. Modelo de presentación .....                 | 37 |
| 2.8.3.8. Modelo de flujo de proceso .....             | 37 |
| 2.8.3.9. Diagramas de comportamientos.....            | 38 |
| 2.8.3.10. Diagrama de estados .....                   | 40 |
| 2.8.3.11. Diagrama de secuencia .....                 | 41 |
| 2.8.3.12. Diagrama de iteración.....                  | 42 |
| 2.8.3.13. Diagrama de implementación .....            | 43 |
| 2.9. METODOLOGIA DE DESARROLLO UWE .....              | 44 |
| 2.9.1. Tipos de aplicación web.....                   | 44 |
| 2.9.2. Características de una Aplicación Web.....     | 45 |
| 2.9.3. Fases del desarrollo web .....                 | 46 |
| 2.11. CICLO DE VIDA DE LA INGENIERÍA DE SOFTWARE..... | 48 |
| 2.12. HERRAMIENTAS .....                              | 50 |
| 2.12.1. Php.....                                      | 50 |

|  |    |
|--|----|
| 2.12.2. JavaScript.....  | 52 |
| 2.12.3. MySql.....   | 52 |
| 2.12.4. Codeigniter.....   | 53 |
| 2.13. TEORIA DE REDES.....                                       | 56 |
| 2.13.1. Topología de redes.....                                  | 56 |
| 2.13.2 Modelo cliente servidor .....                             | 58 |
| 2.13. INGENIERIA DE SOFTWARE .....                               | 60 |
| 2.13.1. Las Métricas y la calidad de software.....               | 60 |
| 2.13.2. Medidas de fiabilidad y de disponibilidad.....           | 60 |
| 2.13.3. Modelo (Cocomo II) .....                                 | 61 |
| 2.13.4. Factores de calidad de estándar ISO 9126.....            | 62 |
| 2.13.5. Métricas basadas en la función .....                     | 63 |
| 2.13.6. Modelo de construcción de costos.....                    | 64 |
| CAPITULO III.....  | 66 |
| 3. MARCO APLICATIVO .....  | 66 |
| 3.1. DESCRIPCIÓN GOBIERNO AUTÓNOMO MUNICIPAL DE TIAHUANACU ..... | 66 |
| 3.2. INGENIRIA DE REQUERIMIENTOS.....                            | 66 |
| 3.3. GAME .....  | 68 |
| 3.3.1. Primera iteración .....                                   | 68 |
| 3.3.2. Segunda iteración .....                                   | 71 |



|          |  |    |
|----------|--|----|
| 3.3.3.   | Tercera iteración.....                 | 72 |
| 3.3.4.   | Cuarta iteración.....                  | 75 |
| 3.3.5.   | Modelo de casos de uso.....            | 77 |
| 3.3.5.1. | Diagrama de casos de uso general.....  | 77 |
| 3.3.5.2. | Descripción de caso de uso.....        | 79 |
| 3.3.6.   | Diagrama de clases.....                | 81 |
| 3.3.6.   | Modelado de la base de datos.....      | 81 |
| 3.3.7.   | Modelo de contenidos.....              | 85 |
| 3.3.8.   | Modelo de Secuencia.....               | 86 |
| 3.3.9.   | Modelo de colaboración.....            | 87 |
| 3.3.10.  | Modelo de actividades.....             | 88 |
| 3.3.11.  | Modelo de despliegue.....              | 89 |
| 3.4.     | POST-GAME.....                         | 90 |
| 3.4.1.   | Identificación de roles.....           | 91 |
| 3.5.     | METRICAS DE CALIDAD.....               | 92 |
| 3.6.1.   | Operación del sistema – facilidad..... | 96 |
| 3.7.     | SEGURIDAD.....                         | 97 |
| 3.7.1.   | Integridad.....                        | 98 |
| 3.7.2.   | Confidencialidad.....                  | 98 |
| 3.7.3.   | Disponibilidad.....                    | 99 |

|   |     |
|---|-----|
| 3.7.4. Irrefutabilidad.....                         | 99  |
| 3.8. COSTOS .....                                   | 99  |
| 3.8.1. Modelo cocomo .....                          | 100 |
| 3.8.1. Estudio de factibilidad .....                | 101 |
| 3.8.2. Factibilidad técnica .....                   | 101 |
| 3.8.3. Requerimientos de hardware .....             | 102 |
| 3.8.4. Requerimientos de software.....              | 103 |
| 3.8.5. Factibilidad operativa .....                 | 103 |
| 3.8.6. Costo Laboral en el ciclo de ejecución ..... | 104 |
| CAPITULO IV .....                                   | 108 |
| 4. CONCLUSIONES Y RECOMENDACIONES.....              | 108 |
| 4.1. CONCLUSIONES .....                             | 108 |
| 4.2. RECOMENDACIONES .....                          | 109 |
| BIBLIOGRAFIA .....                                  | 110 |

## INDICE DE FIGURAS

|   |    |
|---|----|
| Figura No 1.1 modelo scrum .....                    | 10 |
| Figura No 1.2 modelo vista controlador .....        | 12 |
| Figura No 2.1 diagrama de casos de uso .....        | 34 |
| Figura No 2.2 diagrama de clases .....              | 35 |
| Figura No 2.3 diagrama de objeto .....              | 36 |
| Figura No 2.4 diagrama de componentes .....         | 39 |
| Figura No 2.5 diagrama de estados.....              | 40 |
| Figura No 2.6 diagrama de secuencia.....            | 41 |
| Figura No 2.7 Diagrama de iteraciones .....         | 42 |
| Figura No 2.8 Diagrama de implementación.....       | 43 |
| Figura No 2.9 Fases de UWE .....                    | 47 |
| Figura No 2.10 Ciclo de vida del proceso .....      | 49 |
| Figura No 2. 11modelo vista controlador .....       | 55 |
| Figura No 2.12 Topología de red estrella .....      | 57 |
| Figura No 2.13 Topología de red en Malla.....       | 57 |
| Figura No 2 14 Topología de red Anillo.....         | 58 |
| Figura No 2. 15 Topología de red estrella .....     | 59 |
| Figura No 3.1 Módulo Beneficiarios listado .....    | 70 |
| Figura No 3. 2 Módulo Beneficiarios Adicionar ..... | 70 |
| Figura No 3.3 Módulo Beneficiarios Actualizar ..... | 71 |

|  |    |
|--|----|
| Figura No 3.4 Módulo Tutores Listado .....             | 72 |
| Figura No 3.5 Módulo Registro de Pagos.....            | 74 |
| Figura No 3.6 Módulo Recibo de Pagos.....              | 74 |
| Figura No 3.7 Módulo Usuarios Lista .....              | 76 |
| Figura No 3.8 Módulo adicionar Usuario.....            | 76 |
| Figura No 3.9 Descripción de Actores.....              | 77 |
| Figura No 3.10 Diagrama de Caso de uso General.....    | 78 |
| Figura No 3.11 Caso de uso Gestor .....                | 78 |
| Figura No 3.12 Diagrama de clases .....                | 81 |
| Figura No 3.13 Modelo Físico de la Base de Datos ..... | 82 |
| Figura No 3.14 Modelo de contenidos.....               | 86 |
| Figura No 3.15 Diagrama de Secuencia .....             | 87 |
| Figura No 3.16 diagrama de colaboración.....           | 87 |
| Figura No 3.17 Diagrama de actividades.....            | 88 |
| Figura No 3.18 Diagrama de Despliegue.....             | 89 |
| Figura No 3.19 Topología de red Estrella.....          | 90 |
| Figura No 3.20 Clasificación de roles .....            | 92 |

## INDICE DE TABLAS

|  |     |
|--|-----|
| Tabla No 3.1 Ingeniería de Requerimientos .....                          | 67  |
| Tabla No 3. 2 Primera Iteración.....                                     | 69  |
| Tabla No 3.3 Segunda Iteración .....                                     | 71  |
| Tabla No 3.4 Tercera Iteración.....                                      | 73  |
| Tabla No 3.5 Cuarta Iteración.....                                       | 75  |
| Tabla No 3.6 Descripción de Caso de Uso “Registro de Gestor”.....        | 79  |
| Tabla No 3.7 Descripción de Caso de Uso "Asignar rol al usuario" .....   | 79  |
| Tabla No 3.8 Descripción de Caso de Uso "Registro de Beneficiario" ..... | 79  |
| Tabla No 3.9 Descripción de Caso de Uso "Registro de Tutor" .....        | 80  |
| Tabla No 3.10 Descripción de Caso de Uso "Registro de Pagos" .....       | 80  |
| Tabla No 3.11 Beneficiarios .....  | 83  |
| Tabla No 3.12 Tutores .....  | 84  |
| Tabla No 3.13 Registros .....  | 84  |
| Tabla No 3.14 Mes Pagado.....  | 84  |
| Tabla No 3.15 Tipo de Beneficiarios.....                                 | 85  |
| Tabla No 3.16 Usuarios .....   | 85  |
| Tabla No 3.18 Descripción de roles del sistema.....                      | 91  |
| Tabla No 3.19 Métricas de calidad .....                                  | 93  |
| Tabla No 3.20 Constantes del modelo Semiorgánico.....                    | 101 |

|   |     |
|---|-----|
| Tabla No 3.21 Requerimientos mínimos de hardware .....      | 102 |
| Tabla No 3.22 Requerimientos recomendados de hardware ..... | 102 |
| Tabla No 3.23 Costos del Software .....                     | 103 |
| Tabla No 3.24 Costo Operativo por Año.....                  | 104 |
| Tabla No 3.25 Tarifa del programador para el sistema .....  | 104 |
| Tabla No 3.26 Tarifa del programador para el sistema .....  | 105 |

## **CAPITULO I**

### **1. MARCO PRELIMINAR**

#### **1.1. INTRODUCCIÓN**

La ONU a través de su programa de acción mundial para personas con discapacidad informa que a causa de deficiencias mentales, físicas o sensoriales, millones de personas impedidas, a las que se deben reconocer los mismos derechos, y brindar iguales oportunidades, que a todos los demás seres humanos. Con demasiada frecuencia, estas personas han de vivir en condiciones de desventaja debido a barreras físicas y sociales existentes en la sociedad que se oponen a su plena participación. El resultado es que millones de niños y adultos del mundo entero arrastran a menudo una existencia marcada por la segregación y degradación.

En el panorama general alrededor del 10% de la población mundial, o sea 650 millones de personas, vive con una discapacidad, constituyen la mayor minoría del mundo, y la esperanza de vida es superior a los 70 años, en promedio alrededor de 8 años o el 11.5% de la vida de un individuo transcurre con incapacidades, el 80 % de las personas con discapacidad vive en países en desarrollo, según el Programa de las Naciones Unidas para el Desarrollo (PNUD), el Banco Mundial estima que el 20% de los más pobres del mundo tienen discapacidades, y tienden a ser considerados dentro de sus propias comunidades como las personas en situación más desventajosa

El gobierno del estado plurinacional de Bolivia el 26 de septiembre de 2017 decreto la ley de inserción laboral y de ayuda económica para personas con discapacidad, que obliga a las empresas e instituciones públicas y privadas a dar empleo a discapacitados o sus familiares.

Los mismos que tuvieran a su cargo el cuidado de este sector vulnerable de la sociedad esta ley delega a instituciones públicas a cubrir el 4% de sus planillas y el 2% a empresas privadas, así mismo obliga a los gobiernos autónomos municipales a pagar un bono de 250 Bs, para efectivizar este pago, las personas deben portar un carnet con discapacidad grave o muy grave.

El municipio de Tiahuanacu se encuentra localizada en la provincia Ingavi a 74 Km de la ciudad de La Paz cuenta con una población de 24,378 habitantes de los cuales de los cuales se estima que 4,876 pertenecería a este sector. El gobierno autónomo de Tiahuanacu tiene por propósito contribuir y satisfacer a las necesidades de las personas con discapacidad cuando realicen el trámite del cobro de este bono, así mismo prevé garantizar la participación e integración de sus ciudadanos para el desarrollo sostenible del proyecto en el tiempo.

La idea de un sistema para el registro y control del pago del bono a personas con discapacidad nace a partir de una visita a esta localidad lo que se quiere lograr es tener un registro de las personas a las cuales se les pagara, tener la información básica de ellos y saber cuándo se les tiene que pagar, esto con el fin de evitar confusiones o fraudes.

Este proyecto de grado se enmarca en la ley de telecomunicaciones en su artículo 77 que insta a las instituciones públicas a utilizar software libre, asimismo contribuye con la aplicación de gobierno electrónico. El sistema se utilizara las metodologías UWE para el (análisis y diseño) y SCRUM para el desarrollo herramientas como HTML5, JAVASCRIPT, PHP, BOTSTRAP, y para interfaz de usuario se desarrollara en el framework de angular4 y para la administración de bases de datos MYSQL.



## **1.2. ANTECEDENTES**

### **1.2.1. Antecedentes de la institución**

El municipio de Tiahuanacu se encuentra en el departamento de La Paz dentro de la provincia Ingavi, así mismo el municipio está constituido por 24,378 habitantes las cuales son dependientes del municipio de Tiahuanacu. Podríamos decir que sus principales actividades son el turismo y la agricultura, en la actividad turística se encuentra los museos más importantes de la región sud americana Tiahuanacu tiene como principales museos al museo monolítico, cerámicos también son dependientes de actividades como la hotelería y la gastronomía, dentro de la actividad agropecuaria se caracteriza por la crianza de ganado lechero y ovino, además de la producción de cebada, papa y sus derivados como chuño y la tunta.

La misión del municipio de Tiahuanacu, es contribuir a la satisfacción de las necesidades colectivas y garantizar la integración y participación de sus ciudadanos en la planificación, implementación y el desarrollo humano sostenible del municipio de Tiahuanacu.

### **1.2.2. Antecedentes de proyectos similares**

- [Agencia - AGETIC] PLATAFORMA EUSTAQUIO MOTO MENDEZ PARA EL REGISTRO DE PERSONAS CON DISCAPACIDAD.
- [Municipio – Gobierno Autónomo Municipal De El Alto] SISTEMA DE PAGO DEL BONO A PERSONAS CON DISCAPACIDAD, Desarrollo de un sistema para registrar y controlar los pagos realizados, que apoye los procesos de pago del bono para personas con discapacidad agilizando y reduciendo colas si es existiera.

- [Municipio – Gobierno Autónomo Municipal De La paz] SISTEMA DE PAGO DEL BONO A PERSONAS CON DISCAPACIDAD, Desarrollo de un sistema para registrar y controlar los pagos realizados, que apoye los procesos de pago del bono para personas con discapacidad agilizando y reduciendo colas si es existiera.

### **1.3. PLANTEAMIENTO DEL PROBLEMA**

#### **1.3.1. Problema principal**

El municipio de Tiahuanacu en el marco de sus atribuciones y funciones delegadas por el gobierno central tiene proyectado realizar el pago del bono a personas con discapacidad, a partir de la gestión 2018 en adelante lo que genera bases de datos en listas o documentos Word o Excel de beneficiarios, la falta de organización en el cronograma de pagos y estos a su vez se realizan en distintos ordenadores lo que dificulta la planeación y organización de pago del bono.

#### **1.3.2. Problema pecundario**

- Falta de una base de datos que pueda ser capaz de centralizar información que se genera, no existe y dificulta la búsqueda de información oportuna pues no se sabe con exactitud dirección, teléfono ocupación y otros datos necesarios para realizar seguimientos.
- Procesos lentos al realizar el cobro del bono a personas con discapacidad, las personas cuando llegan caminando buscando a la persona encargada de finanzas para poder cobrar su bono siendo que los encargados de realizar el pago son el personal del SLIM DNA del municipio.

- Falta de organización del cronograma de pago a personas con discapacidad, al no existir una socialización de la información no se coordina con los beneficiarios que son los más perjudicados y tienen que volver otro día.
- No existen reportes automáticos que respalden la ejecución de operaciones, pagos o el registro de nuevos beneficiarios que necesariamente deben de respaldarse con documentación oficial.
- Existe ausencia de información no se conoce cuanto es que se debe asignar al pago del bono a personas con discapacidad, pues al no existir la cantidad real de personas con discapacidad siempre se tendrá el presupuesto.
- No se cuenta con una herramienta que determine la cantidad aproximada de personas con discapacidad.
- La seguridad de la información generada de las operaciones de pago del bono a personas con discapacidad es vulnerable por que se encuentra en hojas o documentos Word y Excel.
- La ausencia de una herramienta capaz de generar copias de seguridad en caso de que ocurra cualquier percance o problemas con los ordenadores.
- La disponibilidad de datos en una intranet dentro de la alcaldía del municipio de Tiahuanacu y sus unidades.

### **1.3.3. Formulación del problema**

¿Cómo mejorar el manejo de información y los procesos y operaciones del registro y control de pago del bono a personas con discapacidad en el municipio de Tiahuanacu?

## **1.4. OBJETIVOS**

### **1.4.1. Objetivo general**

Desarrollar e implementar un sistema de registro y control para el pago del bono a personas con discapacidad en el municipio de Tiahuanacu, que optimice las operaciones relacionadas, organizando, reduciendo colas y brindar una mejor atención a los usuarios, contar con respaldos de seguridad de la información.

### **1.4.2. Objetivos específicos**

- Realizar un diagnóstico de datos transaccionales.
- Diseñar un sistema utilizando metodologías y modelos tradicionales y ágiles para garantizar la calidad del sistema.
- Diseñar interfaces intuitivas, dinámicas y modernas para el usuario dentro del marco de funcionalidad y usabilidad.
- Realizar un módulo de alerta de caducidad de carnets de codepedis.
- Realizar pruebas de calidad del sistema implementado en coordinación con los usuarios del municipio de Tiahuanacu.

## **1.5. JUSTIFICACIÓN**

### **1.5.1. Justificación tecnológica**

Las tecnologías emergentes cada día dan pasos gigantescos en aplicaciones del lado de los clientes y de los servidores.

Una de las formas de trabajo son las del MVC (Modelo Vista Controlador), los sistemas que se dividen en frontend y backend, así como las implantaciones de firebases o Apis como acompañamientos de live reload o reload view que consisten en consultas a bases de datos en tiempo real que lo hacen mucho más eficientes en su trabajo.

Pero nada sería posible sin el uso de adecuado de los ordenadores que se han convertido en herramientas indispensables dentro de las instituciones públicas y privadas pues así también son totalmente indispensables en la vida cotidiana de los seres humanos y las actividades de los profesionales y aún más para los desarrolladores de sistemas. Es en ese entendido que los sistemas junto a las computadoras hacen una sinergia para convertir el producto final en herramientas que ayudan en realización de operaciones como registros búsquedas personalizadas y demás.

Así pues el municipio de Tiahuanacu haciendo uso de las herramientas y tecnologías mencionadas y con el sistema de registro y control para el pago a personas con discapacidad contara con información relacionada a las personas con discapacidad para un manejo adecuado de sus recursos económicos así también como sus recursos humanos.

### **1.5.2. Justificación técnica**

El municipio de Tiahuanacu actualmente cuenta con infraestructura y con equipos de computación para la implantación del sistema de registro y control para el pago del bono a personas con discapacidad, el municipio de Tiahuanacu cuenta con equipos de computación conectados en red y que además tienen acceso a internet sujetos a modificaciones o mejoramiento de arquitectura de red.

Además cabe resaltar que las dependencias de la institución estarán a disposición de requerimientos del sistema así como también oficinas e inmobiliario por eso podemos decir que el sistema tiene las condiciones técnicas en el proceso de su implementación.

También tendríamos que decir que el sistema de registro y control para el pago a personas con discapacidad será escalable para que en el tiempo esta pueda fácilmente actualizada o sin ninguna clase de pérdida de datos esto es muy importante en el aspecto de la vida útil del sistema y así pueda adicionarse nuevos módulos o cambios.

### **1.5.3. Justificación económica**

Por lo general las instituciones públicas tiene como consigna atender las necesidades de sus ciudadanos por ende una forma seria la de reducir costos de operación y administración para lograr mayor ejecución presupuestaria.

También hacemos mención a la ley de telecomunicaciones en su artículo 77 dice forma clara “I. Los Órganos Ejecutivos, Legislativo, Judicial y Electoral en todos sus niveles, promoverán y priorizaran la utilización del software libre y estándares abiertos, en el marco de la soberanía y seguridad nacional”. Lo cual implica una determinación del estado para que las aplicaciones sean desarrolladas en software libre.

Por otra parte AGETIC dependiente de la vicepresidencia, recomienda que las instituciones públicas deben buscar la interoperabilidad estos se logra con el cruce de información entre entidades públicas a pedido o requerimientos habría muchos modos de realizar estas operaciones una de estas.

También tendríamos que decir que el único aparato extra aparte de las computadoras será una cámara digital o en su defecto las cámaras de una laptop si tomamos en cuenta el costo de este equipo es parte de los activos de la secretaria de desarrollo productivo usado en la verificación y documentación de avances de desarrollo de su agricultura.

Por lo ya mencionado anteriormente nuestro proyecto es factible económicamente para el municipio de Tiahuanacu, porque su desarrollo estará enfocado totalmente en lenguajes de programación y gestores de bases de datos y sus herramientas open source con licencia GPL (Licencia Publica General).

#### **1.5.4. Justificación social**

La región de Tiahuanacu está comprendida por varias subcentrales las cuales las cuales conforman el municipio de Tiahuanacu, al estar la población dispersada lo que hace que sea un poco más difícil su acceso a información y por ende a el pago del bono a personas con discapacidad, es que surge la necesidad de contar con herramientas como el sistema que se propone.

La entrega del bono a personas con discapacidad es sin duda un apoyo importante para las personas con discapacidad, pues en cierta medida ayuda con los gastos de medicinas, alimentación, etc.

Este proyecto lo que quiere lograr es ayudar al gobierno autónomo municipal de Tiahuanacu en el proceso de este pago, será útil en la medida que la alcaldía podrá realizar copias de seguridad búsquedas instantáneas registros rápidos de personas nuevas.

## 1.6. METODOLOGIA

### 1.6.1. Método de ingeniería

SCRUM es una metodología ágil para desarrollar productos y servicios innovadores que depende de la colaboración de sus colaboradores.

Figura No 1.1 Modelo de SCRUM



Fuente: [www1]

SCRUM es una de las metodologías ágiles más populares existentes hoy por hoy en el mundo que depende exclusivamente del trabajo en equipo en el desarrollo de software y otros tipos de productos pero en lo veremos desde el enfoque de la creación de software este equipo se encarga principalmente de la reducción de tiempo, costo, alcance, calidad, recursos, capacidades organizativas, y otro tipos de limitaciones que dificultan su planificación y ejecución de proyectos. SCRUM tiene como objetivo primordial garantizar y maximizar la calidad del sistema apoyado en el equipo que necesita de:



- Colaboración diaria con el cliente.
- Comunicación verbal directa entre los implicados en el proyecto.
- Predisposición y respuesta al cambio.
- Desarrollo incremental con entregas funcionales frecuentes.
- Prefiere el conocimiento táctico de las personas al explícito de los procesos.
- Auto-organización y compromiso.
- Simplicidad y supresión de artefactos innecesarios en la gestión del proyecto.

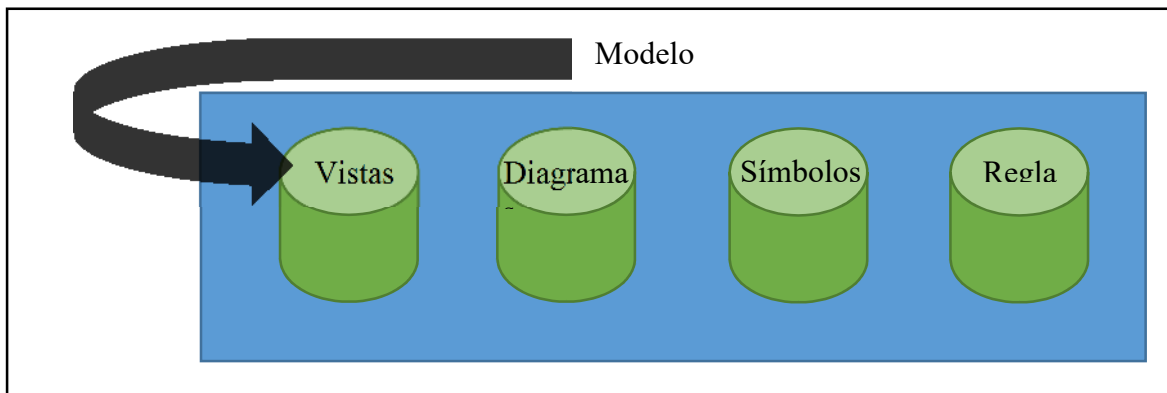
### **1.6.2. Método de ingeniería web**

El método de desarrollo UWE se constituye en un modelo de técnicas, actividades y procesos para la construcción de aplicaciones web. Existen varios tipos de aplicaciones web entre ellas podemos mencionar las informaciones que interactúan con una base de datos, las orientadas a descarga de datos, las interactivas, orientadas al servicio, transaccionales y de flujo de datos.

Así mismo hace la utilización de herramientas como UML (Lenguaje Unificado de Modelado) y este a su vez integra a OCL (Lenguaje de Restricciones de Objetos). UML es un estándar OMG diseñado para visualizar, diseñar, construir, especificar y documentar software orientado a objetos el modelo UML estandariza específicamente 9 tipos de diagramas para representar gráficamente un sistema desde distintos puntos de vista. Debido a las características de UML es una herramienta muy requerida y utilizada muy popular por su notación orientada a objetos es la más utilizada en la actualidad usada en la ingeniería web adaptado a nuestras necesidades.

OCL es el estándar de especificación de invariantes de clases y operaciones de precondiciones y post condiciones donde se define las precisiones del modelado y restricciones de la aplicación web a ser implementada.

Figura No 1.2 Modelo vista controlador



Fuente: (Fernando Berzal Diagramas de UML - 2014)

Las principales características utilizadas por UWE son los siguientes:

- Para la definición de métodos se utilizan los pasos de construcción de los diferentes modelos existentes.
- Para la especificación de restricciones se hace uso de restricciones estrictas para aumentar la exactitud de los modelos (OCL<sup>1</sup>).
- Uso de la notación estándar y sus esquemas conceptuales en el lenguaje (UML: Modelo de lenguaje unificado).

---

<sup>1</sup> OCL: Lenguaje de restricciones de objetos que es utilizada para aumentar la exactitud de los modelos dentro de UML.

## 1.7. HERRAMIENTAS

Para elaborar este proyecto se realizó una descripción, análisis y comparación de datos del pago del bono a personas con discapacidad para realizar una propuesta de un sistema.

Para el desarrollo y la implementación del sistema, y luego de un análisis pertinente se llegó a concluir que podríamos utilizar las siguientes herramientas en el proceso de desarrollo del sistema de registro y control para el pago del bono a personas con discapacidad.

PHP es un lenguaje de programación interpretado que se utiliza para la generación de páginas web de forma dinámica. Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web. (Fuente: [www1](#))

JavaScript es un lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos.

HTML5 es un lenguaje markup usado para estructuras y presentar el contenido para la web. Es la quinta versión del lenguaje HTML, se trata de un sistema para formatear el layout de nuestras páginas, así como hacer algunos ajustes a su aspecto.

Como gestor de base de datos se utilizará mysql. En el año 2009, Widenius abandonó el proyecto de software junto con otros de los desarrolladores principales de MySQL para concentrarse en el desarrollo de MariaDB, la bifurcación de MySQL. El equipo de desarrolladores temía que Oracle entrara en un conflicto de intereses si la empresa desarrollaba el software de código abierto.

Con este propósito, Widenius creó en el año 2012 la MariaDB Foundation. El desarrollo de la bifurcación de MySQL se llevó a cabo en primera instancia a través de Monty Program hasta que en el 2014 se creó MariaDB Corporación.

Bootstrap es un framework web o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en html y css, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end, Bootstrap, originalmente llamado blueprint de twitter, fue desarrollado por Mark Otto y Jacob Thornton de Twitter, como un marco de trabajo (framework) para fomentar la consistencia entre las herramientas internas. (Fuente: www2)

También haremos la utilización del sistema operativo Windows con un ordenador de características mínimas Core I3 con 4 Gbts de memoria RAM, una cámara web o digital y en todo el proceso de se utilizara entrevistas estructuradas y encuestas.

## **1.8. LIMITES Y ALCANCES**

### **1.8.1. Limites**

El sistema está diseñado para la secretaria de finanzas del gobierno autónomo municipal de Tiahuanacu, el sistema será desarrollado específicamente al proceso de pago del bono a personas con discapacidad en el municipio de Tiahuanacu.

El desarrollo del sistema contemplara específicamente esta labor y por lo que no contemplara otros módulos concernientes al pago del bono a personas con discapacidad.

El sistema de registro y control de pago del bono a personas con discapacidad también realizara reportes sobre cantidades de personas en la localidad, cantidad de pago que se realiza trimestralmente, alertar sobre carnet de afiliación no vigentes, entre otros.

### **1.8.2. Alcances**

El sistema proporcionara a los usuarios, la capacidad de ver todas las opciones posibles para su uso según las funciones que realice, en base a listas de acceso según sus roles.

El sistema ofrecerá la capacidad de búsquedas ágiles y precisos, registros rápidos, actualización de información así como copias de seguridad y reportes de control.

El sistema brindara a la institución la seguridad de un sistema integro en todos sus módulos en especial en la base de datos pues una de las prioridades del sistema es utilizar nuevos métodos de seguridad emergentes, sesiones encriptados, etc.

Tomando muy en serio la seguridad para el sistema se toma en cuenta los siguientes aspectos para en sistema.

- Control de horarios en la atención a personas con discapacidad.
- Seguimiento de cobro del bono de personas con discapacidad.
- Registro de personas con discapacidad.
- Validación de datos.
- Manejo estructurado de datos.
- Copias de seguridad en la nube.

## **1.9. APORTES**

El sistema de registro y control para el pago del bono a personas con discapacidad realizara operaciones como registros y búsquedas disminuyendo los tiempos de atención, optimizara el manejo de la información generada por el proceso de pago, automatizara las tareas de copias de seguridad por futuros percances.

El sistema ayudara al municipio de Tiahuanacu a cumplir sus objetivos en el proceso de pago del bono a personas con discapacidad, gestionando de manera óptima el manejo de la información, y de esta manera también ellas puedan obtener información en lo referente al seguimiento de casos de salud o de vivienda, pero en específico para que los recursos sean utilizados de manera eficiente destinada al sector de personas con discapacidad.

## CAPITULO II

### 2. MARCO TEORICO

#### 2.1. SISTEMA

La palabra sistema tal vez es el término más usado y del que más se abusa en el léxico técnico. Se habla de sistemas políticos, sistemas educativos, de sistemas de aviación y sistemas de fabricación, de sistemas bancarios y sistemas de locomoción. La palabra dice muy poco. El adjetivo se utiliza para describir el sistema y así entender el contexto en el que se usa la palabra. El diccionario Webster define sistema de la siguiente manera:

Un conjunto o disposición de cosas relacionadas que forman una unidad o un todo orgánico;  
Una serie de hechos, principios, reglas, etcétera, clasificado y dispuestos de manera ordenada que muestran un plan lógico de la unión de las partes; Un método o plan de clasificación o disposición; Una manera establecida de hacer algo método procedimiento. La ingeniería de sistemas asume distintas formas, según el dominio de aplicación en que se utilice es decir que se podría aplicar a casi cualquier ámbito en el mundo en que este pueda realizar actividad.

Sistema es un conjunto de componentes interrelacionados que recolectan, procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control en una organización. Además de apoyar la toma de decisiones, la coordinación y el control, los sistemas de información también pueden ayudar a los gerentes y trabajadores del conocimiento a analizar problemas, visualizar temas complejos y crear nuevos productos.

(Roger Pressman, Ingeniería del software, 6th ed. p 136)

Los sistemas de información contienen información sobre personas, lugares y cosas importantes dentro de la organización, o en el entorno que la rodea. Por información nos referimos a los datos que se han modelado en una forma significativa y útil para los seres humanos. Por el contrario, los datos son flujos de elementos en bruto que representan los eventos que ocurren en las organizaciones o en el entorno físico antes de ordenarlos e interpretarlos en una forma que las personas puedan comprender y usar. (Kenneth C. Laudon y Jane P. Laudon, Sistemas De Información Gerencial, 12va ed. P 82)

## **2.2. DISCAPACIDAD**

La discapacidad es aquella condición bajo la cual ciertas personas presentan alguna deficiencia física, mental, intelectual o sensorial que a largo plazo afectan la forma de interactuar y participar plenamente en la sociedad. En la mayoría de países, las personas con discapacidad pueden solicitar el reconocimiento de su condición y, a partir de cierto grado, un certificado de discapacidad, que les permite acceder a una serie de ventajas. El término minusvalía se considera peyorativo fuera del ámbito legal.

La Convención Internacional sobre los Derechos de las Personas con Discapacidad, aprobada por la ONU en 2006, define de manera genérica a quien posee una o más discapacidades como persona con discapacidad. En ciertos ámbitos, términos como discapacitados, ciegos, sordos, etc., aun siendo correctamente empleados, pueden ser considerados despectivos o peyorativos, ya que para algunas personas dichos términos etiquetan a quien padece la discapacidad, lo cual interpretan como una forma de discriminación. En esos casos, para evitar conflictos de tipo semántico, se usan a veces las formas (www3).



### **2.3. PERSONAS CON DISCAPACIDAD**

La discapacidad es una condición que afecta el nivel de vida de un individuo o de un grupo. El término se usa para definir una deficiencia física o mental, como la discapacidad sensorial, cognitiva o intelectual, la enfermedad mental o varios tipos de enfermedades crónicas.

Las personas con discapacidad, la "minoría más amplia del mundo", suelen tener menos oportunidades económicas, peor acceso a la educación y tasas de pobreza más altas. Eso se debe principalmente a la falta de servicios que les puedan facilitar la vida (como acceso a la información o al transporte) y porque tienen menos recursos para defender sus derechos. A estos obstáculos cotidianos se suman la discriminación social y la falta de legislación adecuada para proteger a los discapacitados.

Las personas con discapacidad son más a menudo víctimas de la violencia: los niños discapacitados tienen cuatro veces más posibilidades de ser víctimas de actos violentos, la misma proporción que los adultos con problemas mentales. (www4)

### **2.4. MARCO LEGAL LEY 977**

El 26 de septiembre de 2017 se promulgo la ley de inserción laboral y de ayuda económica para personas con discapacidad.

#### **2.4.1. Objetivo de la ley 977**

ARTÍCULO 1. (OBJETIVO). La presente ley tiene por objeto:

- a) Establecer la inserción laboral en los sectores público y privado, de personas con discapacidad, así como de la madre o el padre, cónyuge, cónyuge, tutora o tutor que se encuentre a cargo de una o más personas con discapacidad menores de dieciocho (18) años o con discapacidad grave y muy grave.
- b) Crear un Bono mensual para las Personas con discapacidad grave y muy grave.

## ARTÍCULO 2. (INSERCIÓN LABORAL OBLIGATORIA E INTERMEDIACIÓN).

I. Todas las instituciones del sector público que comprenden los Órganos del Estado Plurinacional, instituciones que ejercen funciones de control, de defensa de la sociedad y del Estado, gobiernos autónomos departamentales, regionales, municipales e indígena originario campesinos, universidades públicas, empresas públicas, instituciones financieras bancarias y no bancarias, instituciones públicas de seguridad social y todas aquellas personas naturales y jurídicas que perciban, generen y10 administren recursos públicos, tienen la obligación de insertar laboralmente a personas con discapacidad, a la madre o al padre, cónyuge, tutora o tutor que se encuentre a cargo de una o más personas con discapacidad menores de dieciocho (18) años o con discapacidad grave o muy grave, en un porcentaje no menor al cuatro por ciento (4%) de su personal.

En el mismo porcentaje, están obligados a aplicar las Fuerzas Armadas y la Policía Boliviana respecto a su personal administrativo.

II. Todas las empresas o establecimientos laborales del sector privado, que desarrollen cualquier actividad en el territorio nacional, tienen la obligación de insertar laboralmente a personas con discapacidad, a la madre o al padre, cónyuge.

III. Las instituciones del sector público señaladas en el párrafo I del presente artículo, podrán insertar laboralmente mediante invitación directa, a personas con discapacidad, de la misma manera hacerlo en el caso de la madre o el padre, cónyuge, tutora o tutor que se encuentre a cargo de una o más personas con discapacidad menores de dieciocho (18) años o con discapacidad grave y muy grave.

En aquellos casos en los que se requiera la intermediación laboral, ésta será ejercida por el Ministerio de Trabajo, Empleo y previsión social, a través de la Bolsa de trabajo del Servicio Público de Empleo.

El ministerio de Trabajo, Empleo y Previsión social, es una institución autorizada para ejercer la autorización de intermediación laboral. Cualquier persona natural o jurídica<sup>2</sup> que realice intermediación laboral de persona con discapacidad, o la madre o el padre, cónyuge o la tutora o el tutor que se encuentre a cargo de una o más personas con discapacidad, será denunciada ante el Ministerio Público por presunta comisión de delitos de trata y tráfico de personas.

IV. Las instituciones del sector público y las empresas o establecimientos laborales del sector privado que inserten laboralmente en porcentajes superiores a los establecidos en el presente Artículo, obtendrán distinciones y reconocimientos a establecerse en norma reglamentaria.

---

<sup>2</sup> Persona jurídica: es persona moral o ficticia, es una organización o institución formada por varias personas físicas y tiene capacidad independiente de sus miembros.( [www.conceptosjuridicos.com/persona-juridica](http://www.conceptosjuridicos.com/persona-juridica))

V. El Estado Plurinacional de Bolivia garantizará la inamovilidad de las personas con discapacidad, así como de la madre o del padre, cónyuge, tutora o tutor que se encuentre a cargo de una o más personas con discapacidad menores de dieciocho (18) años o con discapacidad grave, en los sectores público y privado, siempre y cuando cumplan con la normativa vigente y no existan causales que justifiquen su desvinculación.

VI. Las instituciones del sector público y las empresas o establecimientos laborales del sector privado deberán:

- a) Brindar accesibilidad a su personal con discapacidad.
- b) Realizar reportes trimestrales al Ministerio de Trabajo, Empleo y Previsión Social, conforme al reglamento especial.

VII. El Ministerio de Trabajo, Empleo y Previsión Social, es la entidad encargada de coordinar con el Ministerio de Salud, la interoperabilidad de datos en el Sistema de información del Programa de Registro Único Nacional de Personas con discapacidad – SIPRUN PCD, y del sistema de control de afiliados – SICOA del instituto Boliviano de la Ceguera, de acuerdo al Artículo 3 del Decreto Supremo N° 1893.

VIII. Las instituciones del sector público y las empresas o establecimientos laborales del sector privado, tienen la obligación de adjuntar a las planillas que se entregan trimestralmente al Ministerio de Trabajo, Empleo y Previsión social, información.

Sobre las personas con discapacidad, así como de la madre o del padre, cónyuge, tutora o tutor que se encuentre a cargo de una o más personas con discapacidad menores de dieciocho (18) años o con discapacidad grave y muy grave,

### ARTÍCULO 3.

(BONO MENSUAL PARA LAS PERSONAS CON DISCAPACIDAD GRAVE Y MUY GRAVE).

I. Es responsabilidad de los Gobiernos Autónomos Municipales, el pago de un Bono mensual para las Personas con Discapacidad grave y muy grave, que tengan acreditado legalmente su domicilio en su respectiva jurisdicción.

II. Se exceptúa el pago del Bono mensual a las personas que ya se encuentren beneficiadas con la inserción laboral, establecida en el Artículo 2 de la presente Ley.

III. El monto del Bono mensual para las Personas con Discapacidad grave y muy grave, es de Bs250.- (Doscientos Cincuenta 001100 Bolivianos), el cual entrará en vigencia a partir de la gestión 2018.

IV. Los Gobiernos Autónomos Municipales financiarán el pago del Bono mensual para Personas con Discapacidad grave y muy grave, con recursos de cualquiera de sus fuentes de ingresos.

V. El Ministerio de Economía y Finanzas Públicas, a través del Tesoro General de la Nación - TGN, asignará anualmente recursos del Fondo Nacional de Solidaridad y Equidad, por un monto máximo de hasta Bs15.000.000.- (Quince Millones 001100 Bolivianos) destinados a aportar al pago del Bono mensual para las Personas con Discapacidad grave y muy grave, conforme a reglamentación específica.

VI. Las y los beneficiarios del Bono mensual para las Personas con Discapacidad grave y muy grave, deberán estar registrados en el sistema de Información del Programa de Registro Único Nacional de Personas con discapacidad – SIPRUN PCD y del Ministerio de Trabajo, Empleo y Previsión Social, de acuerdo a reglamento.

VII. Los Gobiernos Autónomos Municipales tendrán acceso a la base de datos del SIPRUN PCD y del Ministerio de Trabajo, Empleo y Previsión Social, de acuerdo a reglamento.

VIII. Los Gobiernos Autónomos Municipales elaborarán los procedimientos para ejecutar el pago del Bono mensual para las Personas con Discapacidad grave y muy grave.

IX. Se exceptúa de las disposiciones del presente Artículo, las personas con discapacidad que perciban el bono de indigencia. Disponible en:([www5](#))

## **2.5. MODELOS AGILES**

En febrero de 2001, Kent Beck y otros 16 notables desarrolladores, escritores y consultores (conocidos como la “Alianza Ágil”) firmaron el “Manifiesto para el desarrollo ágil de software”. Hemos descubierto mejores formas de desarrollar software al construirlo por nuestra cuenta y ayudar a otros a hacerlo. Por medio de este trabajo hemos llegado a valorar:

A los individuos y sus interacciones sobre los procesos y las herramientas.

Al software en funcionamiento sobre la documentación extensa.

A la colaboración del cliente sobre la negociación del contrato.

A la respuesta al cambio sobre el seguimiento de un plan.

Un manifiesto se asocia por lo general con un movimiento político emergente aquel que ataca la vieja vanguardia y sugiere un cambio revolucionario (en el mejor de los casos para mejorar). De alguna forma, esto es con exactitud de lo que se trata el desarrollo ágil.

A pesar de que las ideas subyacentes que conducen al desarrollo ágil han estado presentes por muchos años, no ha sido sino hasta la década pasada que estas ideas han cristalizado en un “movimiento”. En esencia, los métodos ágiles se desarrollaron en un intento por superar las debilidades advertidas y reales en la ingeniería del software convencional. El desarrollo ágil proporciona beneficios importantes, pero es imposible aplicarlo en todos los proyectos, productos, personas y situaciones.

Tras esta reunión se creó The Agile Alliance (La Alianza Ágil), una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida es fue el Manifiesto Ágil, un documento que resume la filosofía “ágil”. (Roger Pressman, Ingeniería del software, 6th ed. p 77)

## **2.6. ANTECEDENTES**

El 06 de febrero del 2018 con la participación de diferentes instituciones y dando cumplimiento al Decreto Supremo 3437, se ha construido la Plataforma Plurinacional de Información de personas con discapacidad “Eustaquio Moto Méndez” cuyo objetivo es centralizar los datos de las personas con discapacidad que existen en Bolivia y generar listados para el pago del bono mensual de Bs. 250. Para aquellas personas con discapacidad grave muy grave.

El objetivo de la plataforma “Eustaquio Moto Méndez” es generar los reportes anuales y mensuales necesarios para que se efectúe el pago del bono mensual consecuentemente disponer de un observatorio que permita conocer la información asociada a las personas con discapacidad. (www6)

## **2.7. METODOLOGÍA DE DESARROLLO DE APLICACIONES WEB**

### **2.7.1. Metodología de gestión de relaciones RMM**

Se define como un proceso de análisis, diseño y desarrollo de aplicaciones hipermedia. Esta metodología es apropiada para dominios con estructuras regulares es decir, con clases de objetos bien definidas, y con claras relaciones entre esas clases:

- Aproximación para el diseño de sitios web, bajo una aproximación centrada en la información.
- Lenguaje de modelado de sitios web a nivel lógico (dominio de información más estructuras de navegación más elementos de presentación).
- Integrado en una metodología de desarrollo.
- Facilitar la estructuración de páginas web complejas que contienen elementos de distintas entidades (vistas múltiples).
- Permitir la reutilización de elementos en el diseño es decir programación inteligente (vistas jerárquicas).
- Diseño de enlaces más potentes y versátiles.
- mantener el contexto durante la navegación.



### **2.7.2. Método de diseño hipermedia orientado a objetos**

OOHDM propone el desarrollo de aplicaciones hipermedia a través de un proceso compuesto por cuatro etapas: diseño conceptual, diseño navegacional, diseño de interfaces abstractas e implementación este método trata de dividir el problema para que en el momento de afrontar problemas no sea grande.

- **Diseño Conceptual.** En OOHDM, el desarrollo se inicia diseñando la capa conceptual, siendo el principal objetivo de esta etapa capturar los conceptos involucrados en el dominio de la aplicación y describirlos en detalle, haciendo uso de diagramas que permitan expresar con claridad el comportamiento, la estructura y las relaciones entre dichos conceptos. La Programación Orientada a Objetos facilita el traslado del diseño conceptual a la implementación, proveyendo al programador con herramientas que permiten reducir la distancia entre el problema del mundo real y la programación de la solución en la computadora.
- **Diseño Navegacional.** La capa navegacional se compone de objetos construidos a partir de objetos conceptuales, y constituyen en general los elementos canónicos de las aplicaciones hipermedia tradicionales: nodos, enlaces, anclas y estructuras de acceso. Sin embargo, estas clases pueden extender el comportamiento característico de los objetos entonces, los objetos navegacionales pueden actuar como observadores, para construir vistas de objetos conceptuales, y como adaptadores, para extender la actividad navegacional de un nodo y poder aprovechar el comportamiento conceptual del objeto.

- Diseño de Interfaz Abstracta.

Una vez que las estructuras navegacionales son definidas, se deben especificar los aspectos de interfaz. Esto significa definir la forma en la cual los objetos navegacionales pueden aparecer, cómo los objetos de interfaz activarán la navegación y el resto de la funcionalidad de la aplicación, qué transformaciones de la interfaz son pertinentes y cuándo es necesario realizarlas.

Una clara separación entre diseño navegacional y diseño de interfaz abstracta permite construir diferentes interfaces para el mismo modelo navegacional, dejando un alto grado de independencia de la tecnología de interfaz de usuario.

El aspecto de la interfaz de usuario de aplicaciones interactivas (en particular las aplicaciones web) es un punto crítico en el desarrollo que las modernas metodologías tienden a descuidar. En OOHDM se utiliza el diseño de interfaz abstracta para describir la interfaz del usuario de la aplicación de hipermedia. El modelo de interfaz ADVs (Vista de Datos Abstracta) especifica la organización y comportamiento de la interfaz, pero la apariencia física real o de los atributos, y la disposición de las propiedades de las ADVs en la pantalla real son hechas en la fase de implementación.

- Implementación. En esta fase, el diseñador debe implementar el diseño. Hasta ahora, todos los modelos fueron construidos en forma independiente de la plataforma de implementación; en esta fase es tenido en cuenta el entorno particular en el cual se va a correr la aplicación.

Debe identificar también, cómo son organizados los ítems de acuerdo con el perfil, decidir qué interfaz debería ver y cómo debería comportarse. A fin de implementar todo en un entorno web, el diseñador debe decidir además qué información debe ser almacenada o que información es relevante para mejorar la atención que se brindara, en esta etapa se debe tomar en cuenta datos relevantes que serían de utilidad en la administración pública.

## **2.8. INGENIERÍA WEB**

Es un proceso de desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML.

### **2.8.1 UML**

UML son las siglas de “Lenguaje Unificado de Modelado”. Se trata de un estándar que se ha adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollos de software (programas informáticos).

UML es una herramienta propia de personas que tienen conocimientos relativamente avanzados de programación y es frecuentemente usada por analistas funcionales (aquellos que definen qué debe hacer un programa sin entrar a escribir el código) y analistas-programadores (aquellos que dado un problema, lo estudian y escriben el código informático para resolverlo en un lenguaje como Java, C#, Python o cualquier otro).

Por tanto si estás dando tus primeros pasos en programación, te recomendaríamos que te olvides de UML hasta que tengas unos conocimientos mínimos como uso de condicionales, bucles, y conocimiento de la programación orientada a objetos. Esto es solo una recomendación, en realidad prácticamente cualquier persona puede usar UML, incluso podría usarse para realizar esquemas o documentación de procesos que no tengan que ver con la informática

UWE plantea un enfoque iterativo incremental o progresivo en el desarrollo del producto, cuyas actividades fundamentales son el análisis de requisitos y el diseño conceptual, de la navegación y de la presentación. Los elementos hipermedia se representan por medio de elementos propios de los diagramas de clases UML. Así por ejemplo, los nodos son clases, los enlaces son asociaciones estereotipadas y las ayudas a la navegación (como índices o mapas) son clases estereotipadas. Para modelar aspectos dinámicos se hace uso de modelos de tarea y diagramas de estado, mientras la navegación y la presentación se representan por medio de UML y de estereotipos creados al efecto. Los principales aspectos en los que se fundamenta UWE son los siguientes: Uso de una notación estándar para todos los modelos Definición de métodos: Definición de los pasos para la construcción de los diferentes modelos.

Especificación de Restricciones: Se recomienda el uso de restricciones escritas (OCL: Lenguaje de restricciones de objetos) para aumentar la exactitud de los modelos.

Este proceso de autoría está dividido en cuatro pasos o actividades:

- Análisis de Requisitos: Fija los requisitos funcionales de la aplicación Web para reflejarlos en un modelo de casos de uso. Esto da lugar a los diagramas de casos de uso.
- Diseño Conceptual: Se construye el modelo conceptual del dominio de la aplicación considerando los requisitos reflejados en los casos de uso. El resultado es el diagrama de clases de dominio.
- Diseño Navegacional: Se obtienen el modelo de espacio de navegación y el de estructura de navegación, que muestra como navegar a través del espacio de navegación. El resultado son diagramas de clases que representan estos modelos.
- Diseño de Presentación: Representa las vistas del interfaz del usuario mediante modelos estándares de interacción UML. disponible en: (www10)

### **2.8.2. UML – Based web engineering (UWE)**

Los antecedentes de UML (Lenguaje de modelado unificado) se sitúan en la década de los 90 con distintos estándares para modelado de software, no obstante podemos hablar de dos grandes versiones:

UML 1.X (comprende UML 1.1, 1.2, 1.3, 1.4, 1.5): desde finales de los 90 se empezó a trabajar con el estándar UML. En los años sucesivos fueron apareciendo nuevas versiones que introducían mejoras o ampliaban a las anteriores.

UML 2.X (comprende UML 2.1 hasta UML 2.5, 2.6, etc.): en torno a 2005 se difundió una nueva versión de UML a la que podemos denominar UML 2.X. Comprenden varias revisiones.

Hay que tener en cuenta que UML es un conjunto muy amplio de normas. Prácticamente nadie las conoce todas. Según la empresa o universidad, institución o centro de trabajo se usan determinados programas para crear diagramas y se conocen ciertas partes de UML, pero no el conjunto de UML. (Autor: Martin Fowler, Kendall Scott: 2009, disponible en: [www11](http://www11))

### **2.8.3. Tipos de diagrama en UML**

Los diagramas UML son representaciones gráficas que muestran de forma parcial una información referente a un sistema o software, que este siendo desarrollada o ya haya sido implementada.

También es acompañada de documentación que les sirve de apoyo, adoptando las múltiples formas. UML no excluye la posibilidad de mezclar diagramas, que por lo general es muy común.

#### **2.8.3.1. Modelo de caso de uso**

Un caso de uso especifica el comportamiento de un sistema o una parte del mismo y es una descripción de un conjunto de secuencias de acciones, donde cada secuencia representa la interacción de los elementos externos del sistema (sus actores) con el propio sistema. Un caso de uso representa un requerimiento funcional de sistema.

- Identificar actores: Los actores son mejor dicho, los roles que un usuario o usuarios del sistema llevan a cabo en algún momento del tiempo. También pueden ser otros sistemas con los que el “sistema” en proceso de modelado tiene interacción.

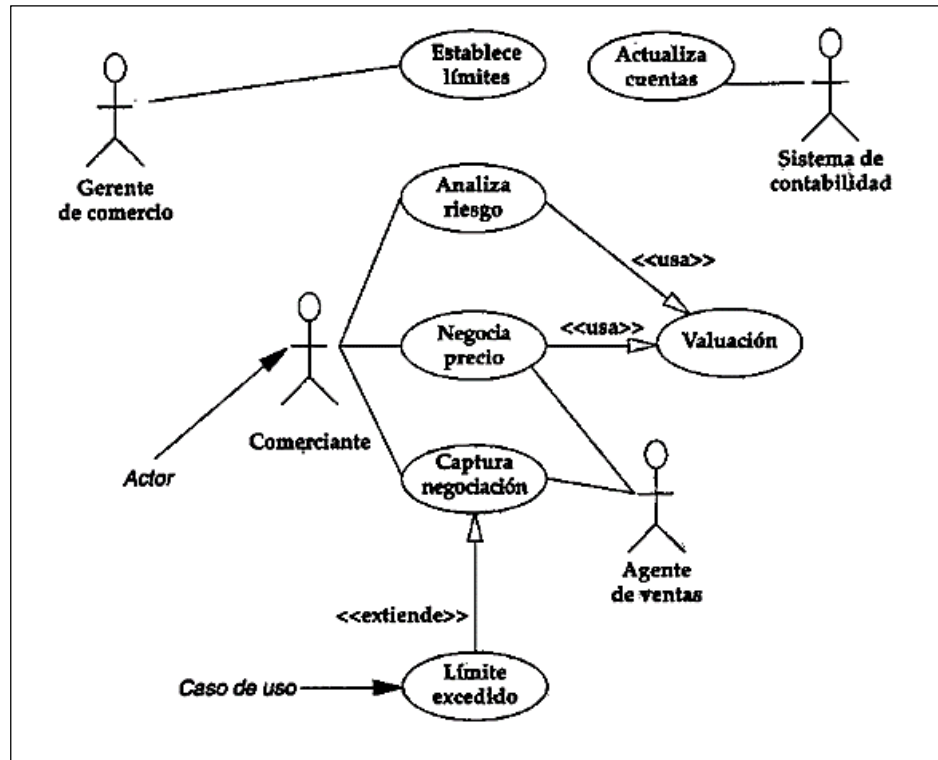
- Identificar Metas (Metas, objetivos generales o responsabilidades): Todos los actores en el entorno a modelar tienen metas u objetivos, o en su defecto responsabilidades, o en su defecto, acciones que desean realizar u obtener del sistema.
- Obtener o identificar los Casos de Uso a partir de las Metas: Las metas son importantes porque a partir de su identificación pasamos a realizarlas y estas se convierten en los Casos de Uso, de esta forma tan sencilla obtenemos la información de funcionalidad que requiere nuestro sistema.
- Especificar cada Caso de Uso: Una vez identificados seguimos a especificarlos uno a uno, es probable que en el inter, de esos sencillos pasos algunos casos de uso desaparezcan o se fusionen con otros, por ambigüedades detectadas o por detalles que se hayan escapado durante el proceso.

### **2.8.3.2 Diagramas de casos de uso**

Los Casos de Uso no forma parte de la llamada Fase de Diseño, sino parte de la fase de Análisis, respondiendo. Estos diagramas muestran operaciones que se esperan de una aplicación o sistema y como se relaciona con su entorno, es por ello que se ve desde el punto de vista del usuario. Describen un uso del sistema y como éste interactúa con el usuario. Los casos de usos se representan en el diagrama por una elipse la cual denota un requerimiento solucionado por el sistema.

El conjunto de casos de usos representa la totalidad de operaciones que va a desarrollar el sistema.

Figura No 2.1 diagrama de casos de uso



[Fuente: www11]

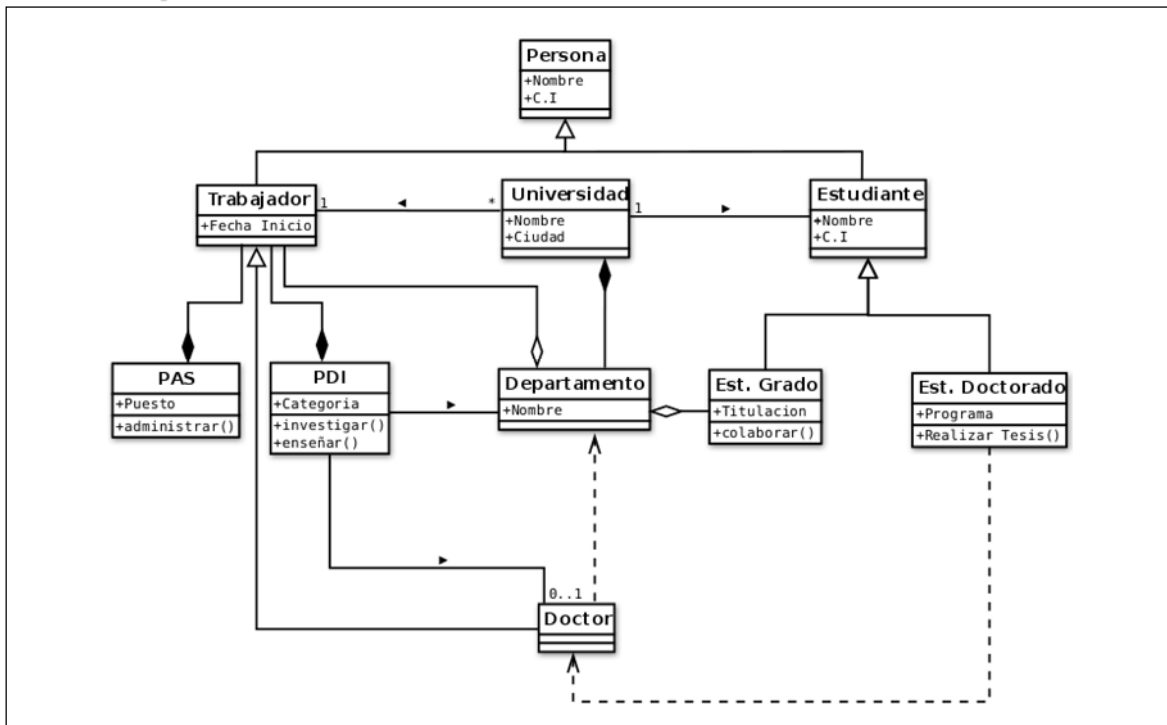
### 2.8.3.3. Diagrama de clases

En UML el diagrama de clases es uno de los tipos de diagramas o símbolo estático y tiene como fin describir la estructura de un sistema mostrando sus clases, atributos y relaciones entre ellos.

Estos diagramas son utilizados durante el proceso de análisis y diseño de los sistemas informáticos, en donde se intentan conformar el diagrama conceptual de la información que se manejará en el sistema. Como ya sabemos UML es un modelado de sistema Orientados a Objetos, por ende los conceptos de este paradigma se incorporan a este lenguaje de modelado lenguaje unificado.



Figura No 2.2 diagrama de clases



[Fuente: www12]

Los diagramas de clases tienen las siguientes características:

- Las clases definen el ámbito de definición de un conjunto de objetos.
- Cada objeto pertenece a una clase.

Los objetos se crean por instanciación de las clases.

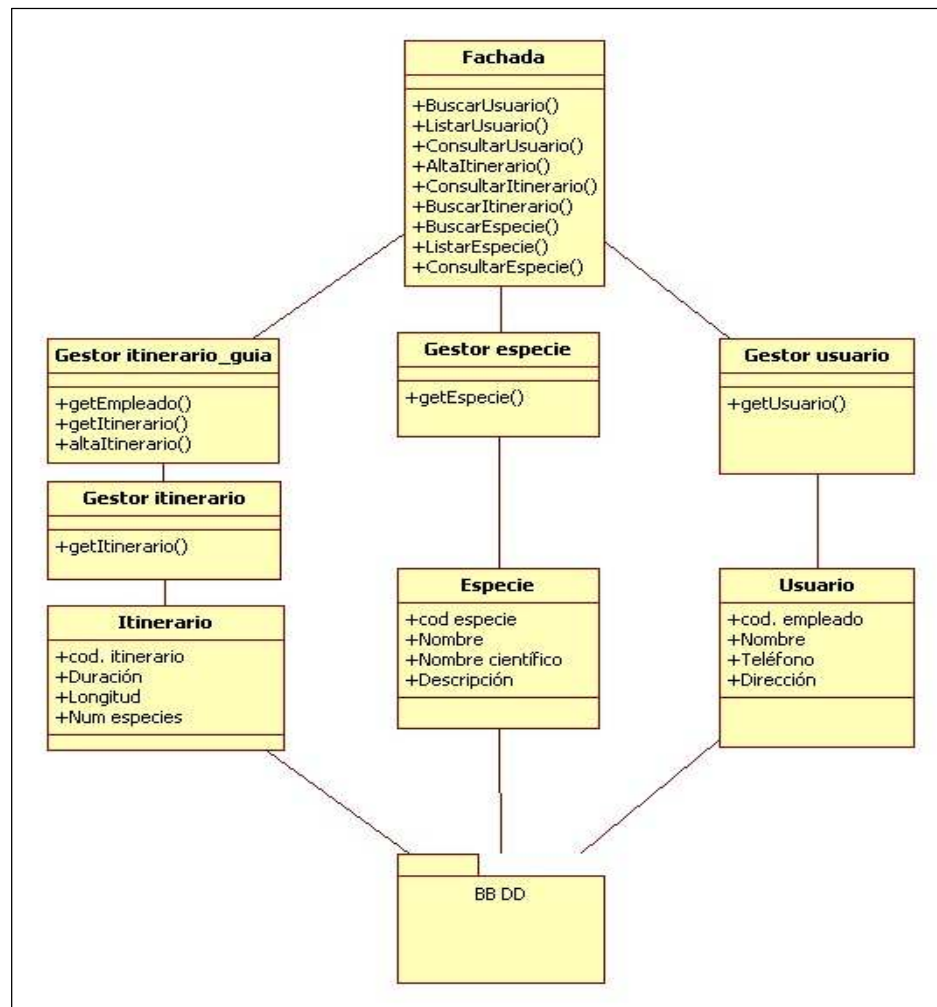
#### 2.8.3.4. Modelo de contenido

Este modelo especifica cómo se encuentran relacionados los contenidos del sistema o del producto, es decir, define la estructura de los datos o de los objetos que se encuentran alojados en el sitio web.

### 2.8.3.5. Diagrama de objetos

Forma parte de la vista estática del sistema. En este diagrama se modelan las instancias de las clases del diagrama de clases. Este diagrama cabe aclarar que cuenta con objetos y enlaces. En estos diagramas también es posible encontrar las clases para tomar como referencia su instanciación. En otras palabras el Diagrama de Objetos muestra un conjunto de objetos y sus relaciones en un momento concreto. .

Figura No 2.3 diagrama de objeto



[Fuente: www13]

### **2.8.3.6. Modelo de navegación**

Este modelo indica como el sistema de páginas web del sitio está relacionado internamente, esto quiere decir que es como se enlazan los elementos de navegación. Para ello se utilizan unidades de navegación llamados “nodos” conectadas por enlaces de navegación. Estos nodos pueden ser representados en la misma página web, no tienen por qué estar en páginas diferentes. Los elementos para representar la estructura del modelo

### **2.8.3.7. Modelo de presentación**

El Modelo de Navegación no indica cuáles son las clases de navegación y de proceso que pertenecen a una página web, si deseamos tener esta información se usa el diagrama de presentación. Agrega una presentación Page class y agrega las propiedades con los estereotipos de UWE en ellos para expresar, que el elemento está ubicado en una página web. Las propiedades pueden anidarse, por ejemplo cada contacto (presentación Group-property) cubre diferentes textos y botones. Ello significa, que para cada contacto la correspondiente dirección de correo y los correspondientes campos de teléfonos y direcciones serán visualizados en la página.

### **2.8.3.8. Modelo de flujo de proceso**

Un flujo del proceso (flujo de trabajo) es representado como un diagrama de actividades, que lo que hacen es describiendo el comportamiento de una clase de proceso, por ejemplo que sucede en detalle, cuando el usuario navega a una clase de proceso, podemos seleccionar nuestro diagrama de navegación y ejecutar la transformación de navegación.

### **2.8.3.9. Diagramas de comportamientos**

Un estado es una condición durante la vida de un objeto, de forma que cuando dicha condición se satisface, se lleva a cabo alguna acción o se espera por un evento. El estado de un objeto se puede caracterizar por el valor de uno o varios de los atributos de su clase.

El diagrama de estados engloba todos los mensajes que un objeto puede enviar o recibir, en otras palabras es un escenario que representa un camino dentro de un diagrama.

Como característica de estos diagramas siempre cuentan con dos estados especiales, el inicial y el final, con la particularidad que este diagrama puede tener solo un estado inicial pero varios estados finales.

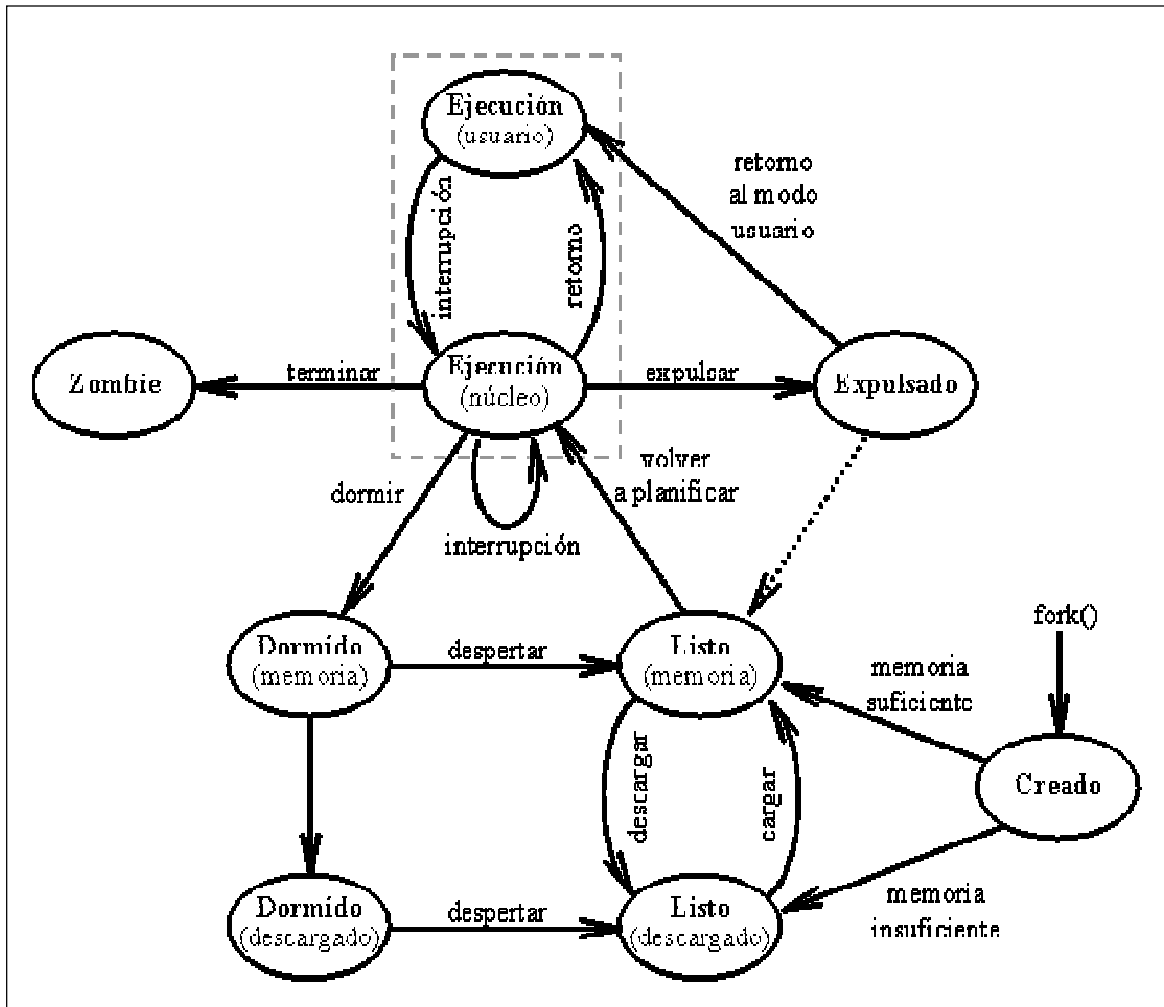
Una transición entre estados representa un cambio de un estado origen a un estado sucesor destino que podría ser el mismo que el estado origen, dicho cambio de estado puede estar aparejado con alguna acción. Además las acciones se asocian a las transiciones y se consideran que ocurre de forma rápida e ininterrumpida durante el proceso de análisis y diseño del producto en esta fase del desarrollo del producto se debe tomar en cuenta los estados por los cuales pasan los objetos.

Los elementos que componen estos diagramas son:

- Círculo lleno, apuntando el estado inicial.
- Círculo hueco que contiene un círculo lleno más pequeño en el interior, indicando el estado final.

- Rectángulo redondeado dividido por una línea horizontal, indicado los estados, en la parte de arriba se encuentra el nombre del estado y abajo se indica la actividad que realiza.
- Flecha, la cual denota la transición, el nombre del evento que causa esta transición etiqueta el cuerpo de la flecha.

Figura No 2.4 diagrama de componentes



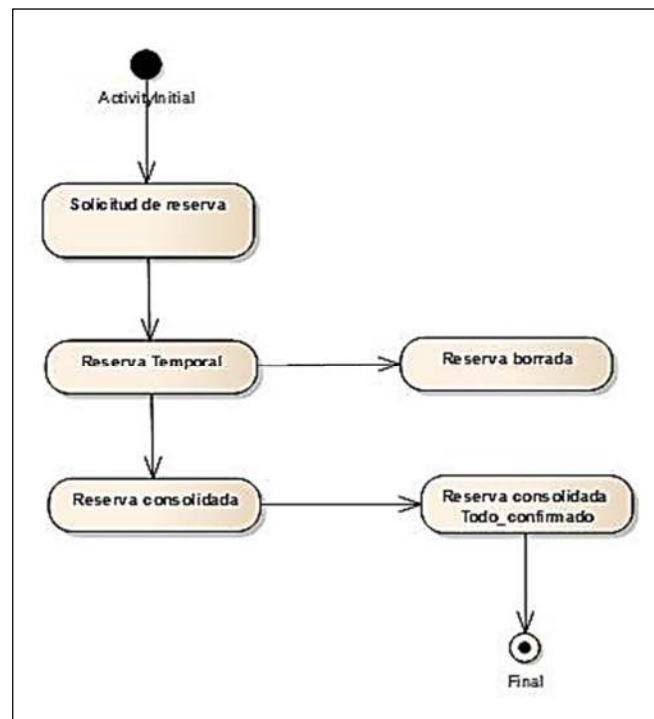
[Fuente: www15]

### 2.8.3.10. Diagrama de estados

Los diagramas de estado muestran el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación en respuesta a eventos (por ejemplo, mensajes recibidos, tiempo rebasado o errores), junto con sus respuestas y acciones. También ilustran qué eventos pueden cambiar el estado de los objetos de la clase. Normalmente contienen: estados y transiciones. Como los estados y las transiciones incluyen, a su vez, eventos, acciones y actividades, vamos a ver primero sus definiciones.

El nombre de un evento tiene alcance dentro del paquete en el cual está definido y puede ser usado en los diagramas de estado por las clases que tienen visibilidad dentro del paquete.

Figura No 2.5 diagrama de estados



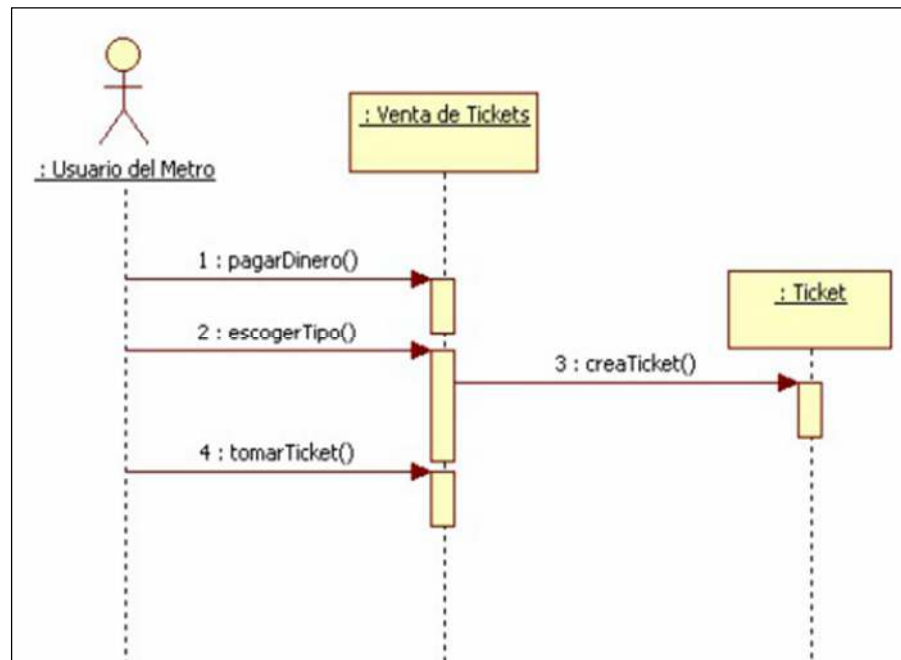
[Fuente: www17]

### 2.8.3.11. Diagrama de secuencia

Un diagrama de secuencia es un tipo de diagrama de interacción porque describe cómo y en qué orden un grupo de objetos funcionan en conjunto. Tanto los desarrolladores de software como los profesionales de negocios usan estos diagramas para comprender los requisitos de un sistema nuevo o documentar un proceso existente. A los diagramas de secuencia en ocasiones se los conoce como diagramas de eventos o escenarios de eventos.

Observa que hay dos tipos de diagramas de secuencia: los diagramas UML y los diagramas que se basan en código. Los últimos se obtienen de un código de programación y no serán cubiertos en esta guía. El software de diagramas UML de Lucidchart está equipado con todas las figuras y funciones que necesitarás para modelar ambos.

Figura No 2.6 diagrama de secuencia



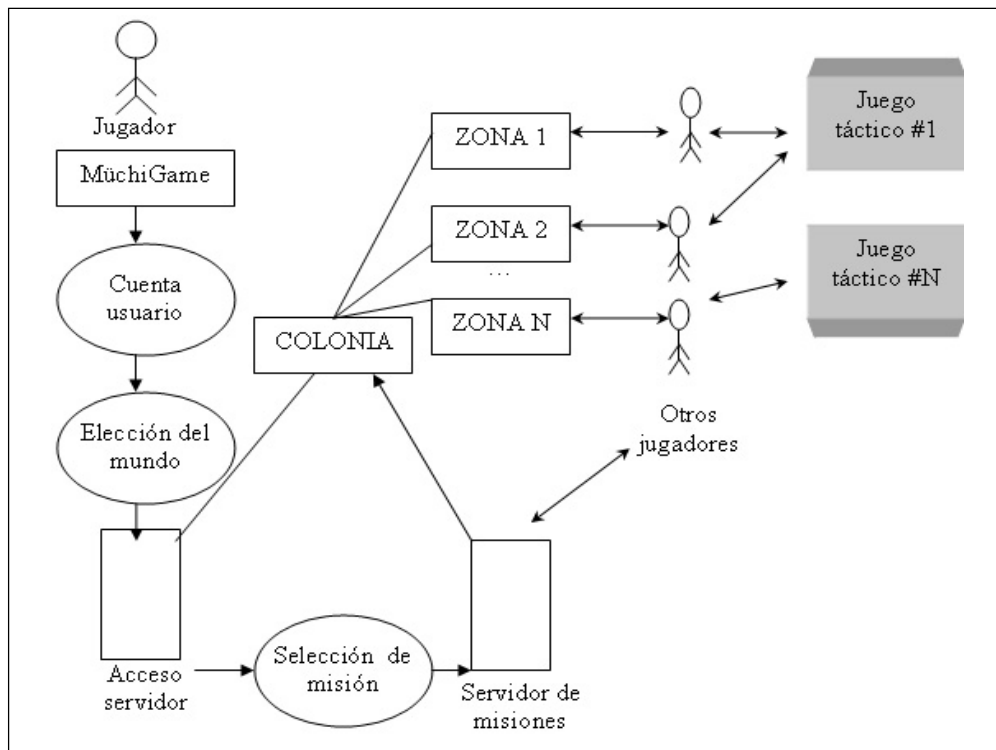
[Fuente: www16]

### 2.8.3.12. Diagrama de iteración

Un Diagrama de Secuencias muestra una interacción ordenada según la secuencia temporal de eventos y el intercambio de mensajes. Los diagramas de secuencia ponen especial énfasis en el orden y el momento en el que se envían los mensajes a los objetos. En los diagramas de Secuencias los elementos están representados por líneas intermitentes verticales, con el nombre del objeto en la parte más alta.

Los mensajes pueden ser o bien síncronos, el tipo normal de llamada del mensaje donde se pasa el control a objeto llamado hasta que el método finalice, o asíncronos donde se devuelve el control directamente al objeto que realiza la llamada.

Figura No 2.7 Diagrama de iteraciones



[Fuente: www18]

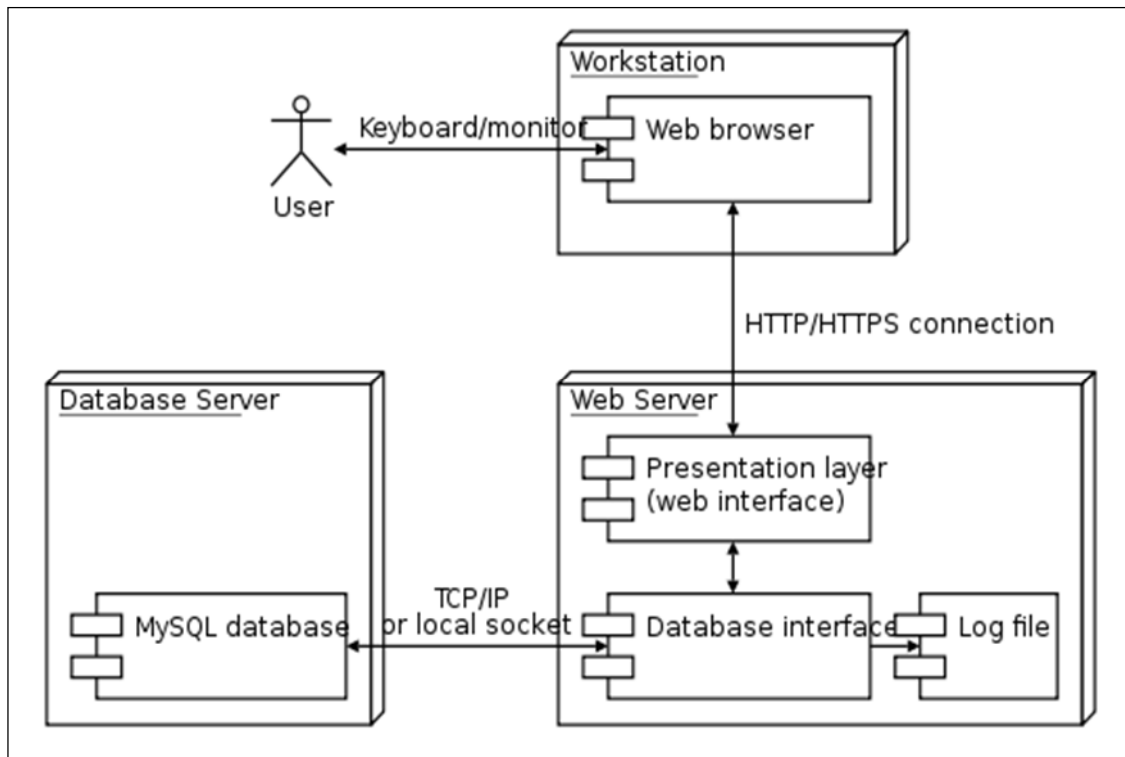


### 2.8.3.13. Diagrama de implementación

Lo que distingue el Diagrama de Componentes de otro tipo de diagramas es sin duda su contenido. Normalmente contiene componentes, interfaces y relaciones entre ellos. Los componentes pertenecen a un mundo físico, es decir, representan a un bloque de construcción al modelar aspectos físicos de un sistema.

Además los Diagramas de Despliegue muestran la configuración en funcionamiento del sistema incluyendo su software y su hardware. Para cada componente de un diagrama es necesario que se deba documentar las características técnicas requeridas, el tráfico de red, el tiempo de respuesta.

Figura No 2.8 Diagrama de implementación



[Fuente: www19]

## **2.9. METODOLOGIA DE DESARROLLO UWE**

Es un sistema de información donde una gran cantidad de datos volátiles, altamente estructurados, es decir un conjunto de datos que por sí solos no dicen nada, van a ser consultados, procesados y analizados mediante navegadores. Una de las principales características va a ser su alto grado de interacción con el usuario, y el diseño de su interfaz debe ser claro, simple y debe estar estructurado de tal manera que sea orientativo para cada tipo de usuario dentro del sistema propuesto. (Autor: Daniel Mínguez Sanz. Emilio José García Morales; Pág. 3; Disponible en: [www6](#))

### **2.9.1. Tipos de aplicación web**

A lo hora de establecer una clasificación la podemos realizar atendiendo a criterios como pueden ser la complejidad de los datos, de la propia aplicación, la volatilidad, la estructuración de los datos o la intencionalidad de la aplicación. De entre todas las posibles clasificaciones, la que aparece a continuación está hecha en base a la intencionalidad de la aplicación:

Los principales problemas que nos encontramos son la falta de fiabilidad, seguridad, escalabilidad, mantenimiento, integración y la alta dependencia. En esta nueva disciplina las necesidades de evolución, mantenimiento, la adaptación a nuevos dispositivos de acceso y la migración a nuevas plataformas y entornos de desarrollo deben dirigir el proceso del ciclo de vida. Para todo esto se han desarrollado metodologías que permiten estructurar comunicar, entender, simplificar y formalizar tanto el dominio como las decisiones de diseño, así como disponer de documentación detallada para posibles cambios del software.

UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en las características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario que son necesarias para la construcción del sistema.

La propuesta de Ingeniería Web basada en UML (UWE (Koch, 2000)) es una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado. Este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el Proceso Unificado de Modelado.

Otras características relevantes del proceso y método de autoría de UWE son el uso del paradigma orientado a objetos, su orientación al usuario, la definición de un meta-modelo (modelo de referencia) que da soporte al método y el grado de formalismo que alcanza debido al soporte que proporciona para la definición de restricciones sobre los modelos. (Autor: Daniel Mínguez Sanz y Emilio José García Morales, Pág. 3 Disponible en: [www7](http://www7))

### **2.9.2. Características de una Aplicación Web**

Las Aplicaciones Web tienen una serie de rasgos comunes que diferencia a unos tipos de aplicaciones software de otros, y que son: Desde el punto de vista del usuario, se ha universalizado su accesibilidad. Actualmente un usuario experto y un usuario con habilidad limitada en el uso de aplicaciones informáticas acceden al mismo tipo de aplicación. Aún más, el número y tipo de usuario de las Aplicaciones Web.

Es predecible, lo que obliga a tener el concepto de facilidad de uso aún más presente que en otros tipos de aplicaciones. Desde el punto de vista de la plataforma se realiza un uso intensivo de la red y la conexión se establece desde distintos tipos de dispositivo de acceso. Desde el punto de vista de la información, asistimos en la actualidad a una disponibilidad global de fuentes heterogéneas de información, estructurada y no estructurada, perteneciente a distintos dominios y que colaboran en el cumplimiento de los objetivos de la aplicación.

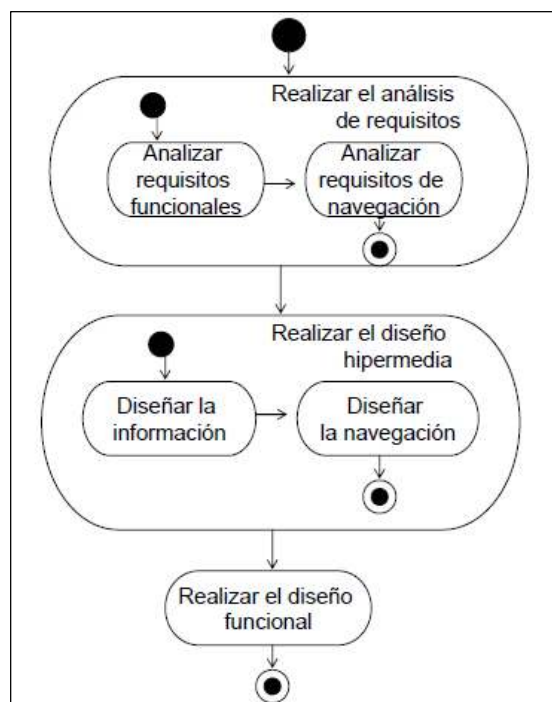
- Informativas: Orientadas a la difusión de información personalizada o no, y con acceso a la BD.
- Orientados a la descarga de datos: Servidores de material didáctico, servidores de canciones,
- Interactivas: Orientadas a la interacción con el usuario.
- Entornos de Trabajo Colaborativo: Herramientas de diseño colaborativo, sistemas de autoría distribuidos. (Autor: Daniel Mínguez Sanz. Emilio José García Morales; Pág. 5; Disponible en: [www8](#))

### **2.9.3. Fases del desarrollo web**

Por lo que respecta al proceso de autoría de la aplicación, UWE hace un uso exclusivo de estándares reconocidos como UML y el lenguaje de especificación de restricciones asociado OCL. Para simplificar la captura de las necesidades de las aplicaciones web, UWE propone una extensión que se utiliza a lo largo del proceso de autoría. Este proceso de autoría está dividido en cuatro pasos o actividades:

- Diseño Conceptual: Materializado en un modelo de dominio, considerando los requisitos reflejados en los casos de uso.
- Análisis de Requisitos: Fija los requisitos funcionales de la aplicación Web para reflejarlos en un modelo de casos de uso.
- Diseño Lógico: Materializado en un modelo de dominio, considerando los requisitos funcionales y no funcionales que son reflejados en los casos de uso que se deben proponer para el desarrollo del producto.

Figura No 2.9 Fases de UWE



Fuente: [Universidad de Sevilla, p. 10]

- Diseño Navegacional : Lo podemos subdividir en
  - Modelo del Espacio de Navegacional.
  - Modelo de la Estructura de navegación: Muestra la forma de navegar ante el espacio de navegación.

- Diseño de Presentación: Representa las vistas del interfaz del usuario mediante modelos estándares de interacción UML. (Daniel Mínguez Sanz. Emilio José García Morales, Pág. 7-10, Disponible en: [www9](#))

## **2.11. CICLO DE VIDA DE LA INGENIERÍA DE SOFTWARE**

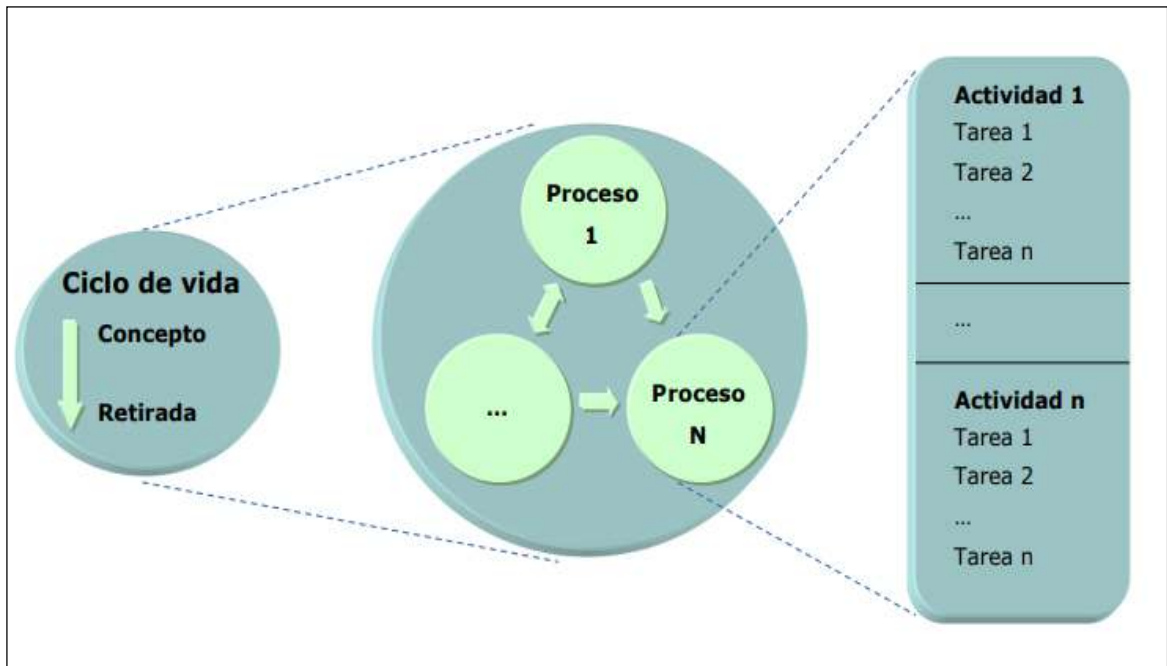
El ciclo de vida es el conjunto de fases por las que pasa el sistema que se está desarrollando desde que nace la idea inicial hasta que el software es retirado o remplazado (muere). También se denomina a veces paradigma.

La ingeniería de software requiere llevar a cabo numerosas tareas entre ellas:

- Análisis de los requisitos de software: Este proceso de reunión de requisitos se intensifica y se centra especialmente en el software. Para comprender la naturaleza de los programas a construirse.
- Diseño: El diseño del software es realmente un proceso de muchos pasos que se centra en cuatro atributos distintos de programa: estructura de datos, arquitectura de software, representaciones de interfaz y detalle procedimental (algoritmo). El proceso del diseño traduce requisitos en una representación del software donde se pueda evaluar su calidad antes de que comience la codificación.
- La Generación de códigos o codificación: El diseño se debe traducir en una forma legible por la máquina. El paso de generación de código lleva a cabo esta tarea. Tomando en cuenta que si se lleva a cabo el diseño de una forma detallada, la generación de código se realiza mecánicamente.

- Las Pruebas: Una vez que se ha generado el código, comienza las pruebas del programa, en la cual se centra en los procesos lógicos internos del software, asegurando de todas las sentencias se han comprobado, y en los procesos externos funcionales; es decir, realizar las pruebas para la detección de errores y garantizar el producto.
- El Mantenimiento: El software indudablemente sufrirá cambios después de ser entregado al cliente (una excepción posible es el software empotrado). Se producirán cambios porque el software debe adaptarse para acoplarse a los cambios de su entorno externo (ejemplo: se requiere un cambio debido a un sistema operativo o dispositivo periférico nuevo), o porque el cliente requiere mejoras funcionales o de rendimiento.. (Roger Pressman, Ingeniería del software, 6th ed. p 77)

Figura No 2.10 Ciclo de vida del proceso



[Fuente: Juan Palacio; 2014]

## **2.12. HERRAMIENTAS**

### **2.12.1. Php**

PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo unas reglas. PHP se utiliza para generar páginas web dinámicas situándose en los lenguajes más populares utilizados hoy por hoy.

Recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc.

¿Cómo trabaja PHP? El lenguaje PHP se procesa en servidores, que son potentes ordenadores con un software y hardware especial, en un navegador web como Internet Explorer, Firefox o Chrome, etc ¿qué ocurre? Se envían los datos de la solicitud al servidor que los procesa, reúne los datos y el servidor lo que devuelve es una página HTML como si fuera estática.

El esquema es: Petición de página web al servidor. El servidor recibe la petición, reúne la información necesaria consultando a bases de datos o a otras páginas webs, otros servidores, etc. El servidor responde enviando una página web pero cuya creación ha sido dinámica.



En un sitio dinámico, la información generalmente está contenida en una base de datos. Cada vez que mostramos la página, como por ejemplo una página de noticias, buscamos en la base de datos las últimas noticias que tenemos ingresadas para mostrar en el navegador del visitante. Ahora bien, ¿cómo se consigue que la página vaya mostrando noticias nuevas? Simplemente cargando las mismas en la base de datos, por ejemplo, a través de un formulario que llena una persona y que una vez relleno cuando pulsa “Enviar” implica que lo que haya escrito se guarde en nuestra base de datos.

De esta manera, cuando solicitamos la página web y el servidor consulte la base de datos, encontrará esta nueva información y mostrará nuestra página de forma distinta a como se veía anteriormente.

Lo que distingue a PHP de JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente como si fuera una página web estática. El cliente recibirá los resultados que el servidor devuelve después de interpretar el código PHP, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido. Es decir, a través de nuestro navegador podríamos ver el código HTML, pero nunca el código PHP que dio lugar al resultado HTML. El servidor web puede ser incluso configurado para que los usuarios no puedan saber si estás o no utilizando PHP.

Lo mejor de usar PHP es que es extremadamente simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales y más avanzados. Con PHP puedes procesar la información de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies, entre muchas más cosas. (Fuente: [www20](http://www20))

### **2.12.2. JavaScript**

JavaScript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con JavaScript podemos crear diferentes efectos e interactuar con nuestros usuarios.

Este lenguaje posee varias características, entre ellas podemos mencionar que es un lenguaje basado en acciones que posee menos restricciones. Además, es un lenguaje que utiliza Windows y sistemas X-Windows,

Es necesario resaltar que hay dos tipos de JavaScript: por un lado está el que se ejecuta en el cliente, este es el JavaScript propiamente dicho, aunque técnicamente se denomina Navigator (navegador) JavaScript. Pero también existe un JavaScript que se ejecuta en el servidor, es más reciente y se denomina LiveWire<sup>3</sup> JavaScript que recientemente se está aplicando en los procesos de desarrollo. (Fuente: <http://www.maestrosdelweb.com/que-es-javascript/>).

### **2.12.3. MySql**

MySQL es un sistema de gestión de base de datos relacional (RDBMS) de código abierto, basado en lenguaje de consulta estructurado (SQL) cabe mencionar que es software de licencia libre.

---

<sup>3</sup> LiveWire: es un marco de construcción de software en el front end simple y sencillo compuesto de código php ([www.sise-tecnologi.com/mysql-livewire/](http://www.sise-tecnologi.com/mysql-livewire/))

MySQL se ejecuta en prácticamente todas las plataformas, incluyendo Linux, UNIX y Windows. A pesar de que se puede utilizar en una amplia gama de aplicaciones, MySQL se asocia más con las aplicaciones basadas en la web y la publicación en línea y es un componente importante de una pila empresarial de código abierto llamado LAMP.

LAMP es una plataforma de desarrollo web que utiliza Linux como sistema operativo, Apache como servidor web, MySQL como sistema de gestión de base de datos relacional o modelo de entidad relacion y PHP como lenguaje de programación orientado a objetos.

Drizzle - un sistema de gestión de base de datos ligero de código abierto bifurcado de mysql en el desarrollo por un grupo de colaboradores que incluye miembros del personal de caconical Ltd, Six y otros basado en MySQL 6.0.

MariaDB - un reemplazo popular "drop-in" desarrollado en la comunidad para MySQL que utiliza las API y los comandos de MySQL. (Fuente: [www21](http://www21)).

#### **2.12.4. Codeigniter**

Codeigniter es simplemente un framework para el desarrollo de aplicaciones escritas en php que utiliza el MVC, que permite a los desarrolladores Web mejorar la forma de trabajar y también agregar mayor velocidad a la hora de crear una página Web. Codeigniter no es el único framework del mercado, también están zend, cakephp, symphony, entre otros. Pero CI (Codeigniter) se diferencia al resto por lo rápido, lo extensible, lo sencillo de aprender y lo poderoso que puede llegar a ser si lo sabemos usar bien. Codeigniter fue desarrollado por Ellis Lab y hasta la versión 1.7.3 soportaba php 4, ya para la versión 2.0 solo soporta php 5.

El MVC es el acrónimo de Modelo Vista Controlador, que es un patrón para el desarrollo de software que separa la lógica de control, la interfaz del usuario y los datos del sistema. En este tipo de arquitectura del software existe un sistema central o controlador que controla las entradas y salida del sistema, un o unos modelo que se encargan de buscar los datos e información que sea necesaria y una interfaz gráfica.

El MVC es muy usado en el desarrollo Web porque a diferencia de la creación de programas locales donde se usa por lo general un mismo lenguaje para desarrollar toda un aplicación en el desarrollo Web se requiere por lo menos usar 2 lenguajes (html y php) para crear una página simple, y todo esto en una sola página aunado a las consultas en la base de datos que son imprescindibles pues crean hasta páginas de 300 líneas que después para depurar, mejorar o saber dónde nos estamos equivocando es un poco difícil.

- Modelo: es el que se encarga de la gestión de la base de datos y toda la lógica del negocio, el modelo permite abstraer al resto de la aplicación de cómo se obtienen o gestionan los datos.

Lo cual nos da una ventaja ya que si después se requiere migrar la base de datos o cambiar algo, no va a ser necesario modificar ni la vista ni los controladores, Controlador: es el centro de control de nuestra aplicación, se encarga de recibir la información del usuario procesarla, validarla, sacar o introducir la información lo que sea necesario con la ayuda de los modelos y renderizar<sup>4</sup> una nueva vista acorde a la necesidad del usuario.

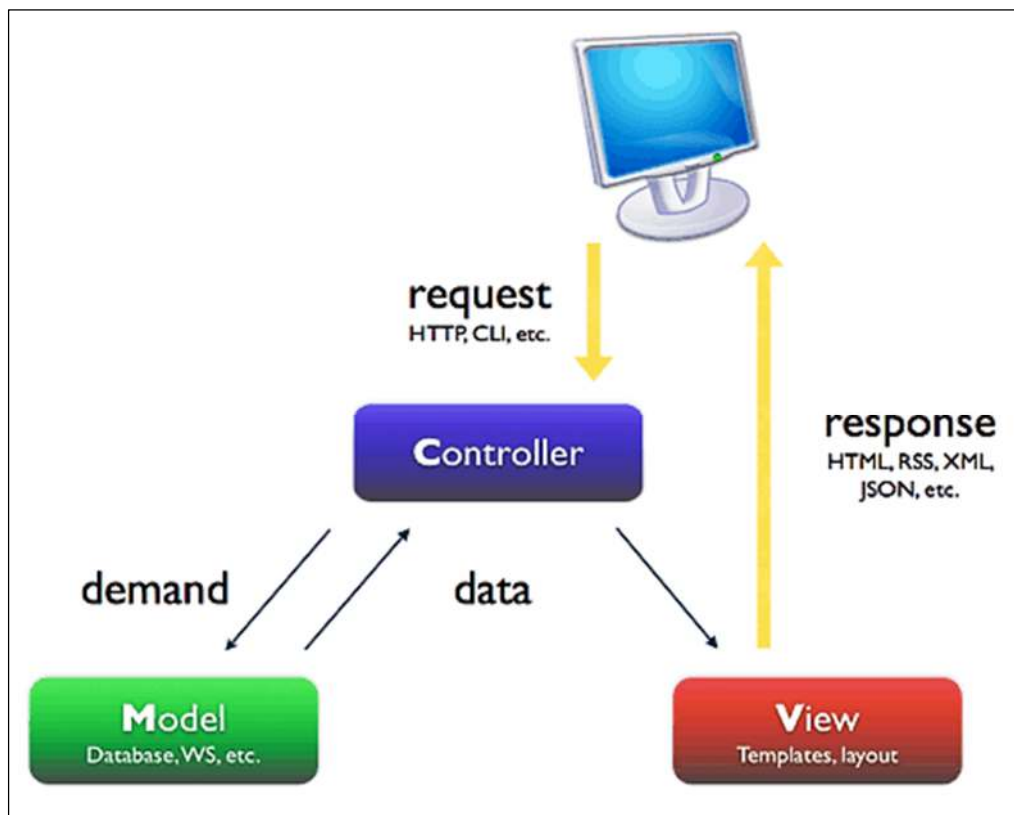
---

<sup>4</sup> Renderizar: Se entiende como el cambio de formato, es aplicar todos los efectos de modo simultáneo a la reproducción. Eso es lo "ideal" y a lo que todos aspiramos. Pero, a menudo, verás que con ciertos filtros, ([videoedicion.org/foro/edicion-de-video-\(general\)/que-es-renderizar/](http://videoedicion.org/foro/edicion-de-video-(general)/que-es-renderizar/))

- Vista: como ya dije anteriormente la vista va a ser la interfaz para nuestro usuario, en nuestro sistema siempre vamos a tener diversidad de vistas que deberán ser cargadas por el control de acuerdo a las necesidades, en desarrollo Web para crear una vista se requiere de html y php, sin contar con que se puede agregar css y JavaScript para darle más dinámica a nuestro sitio.

Codeigniter usa este patrón y además agrega librerías, ayudantes y drivers para ayudarnos a simplificar las tareas en el desarrollo de nuestro sistema, con funciones ya predefinidas que nos facilitan el trabajo. (Fuente: www22)

Figura No 2. 11 modelo vista controlador



(Fuente: www23).

## **2.13. TEORIA DE REDES**

Se han resuelto exitosamente muchos problemas administrativos de diseño de sistema de transporte, sistemas de información y programación de proyectos, con ayuda de los modelos de redes y con técnicas de análisis de redes. Entre los adelantos en el campo de la investigación operativa puede decirse que la teoría de redes está entre las más simples y elegantes que poseen una amplia variedad de aplicaciones.

La estructura topológica de las redes puede ser representada por un gráfico con vértices o nodos y ramas o arcos, correspondientes a las estaciones y enlaces respectivamente.

### **2.13.1. Topología de redes**

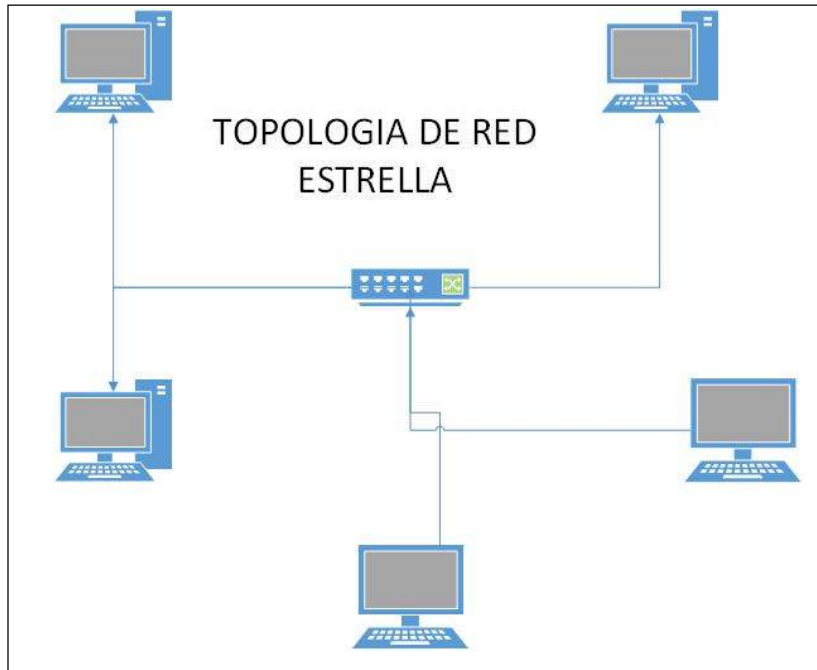
Topología de red, es el arreglo físico o lógico en el cual los dispositivos o nodos de una red (computadoras, impresoras, servidores, hubs, switches, enrutadores, etc.) se interconectan entre sí sobre un medio de comunicación.

Está compuesta por dos partes, la topología física, que es la disposición real de los cables (los medios) y la topología lógica<sup>5</sup>, que define la forma en que los hosts acceden a los medios. Las topologías físicas que se utilizan comúnmente son de bus, de anillo, en estrella, en estrella extendida, jerárquica y en malla.

---

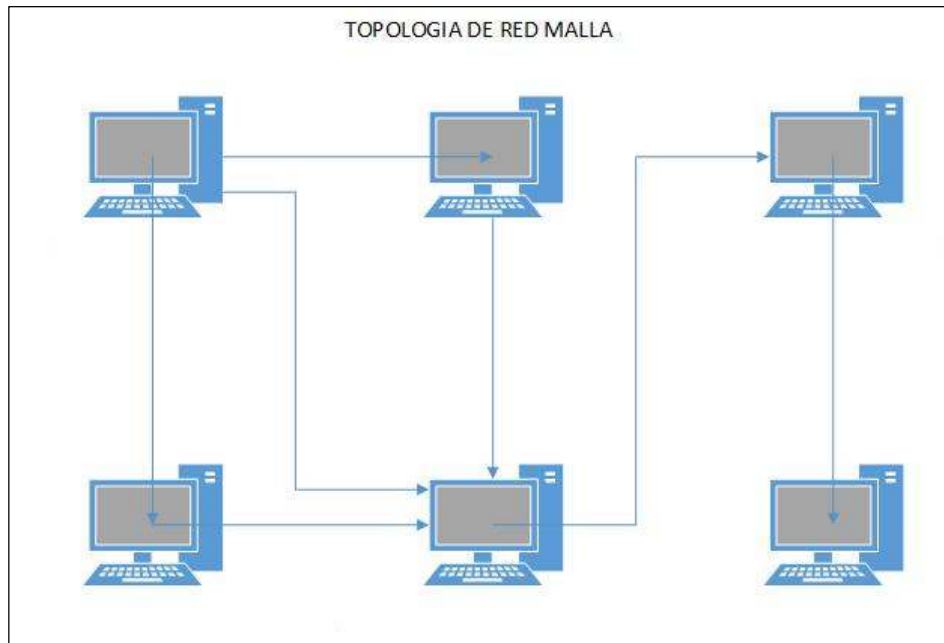
<sup>5</sup> Topología Lógica: describe el flujo de datos a través de la red. Los dos tipos de topologías lógicas son de anillo y de bus. ([www.siles.google.com/site/topologia-logica](http://www.siles.google.com/site/topologia-logica))

Figura No 2.12 Topología de red estrella



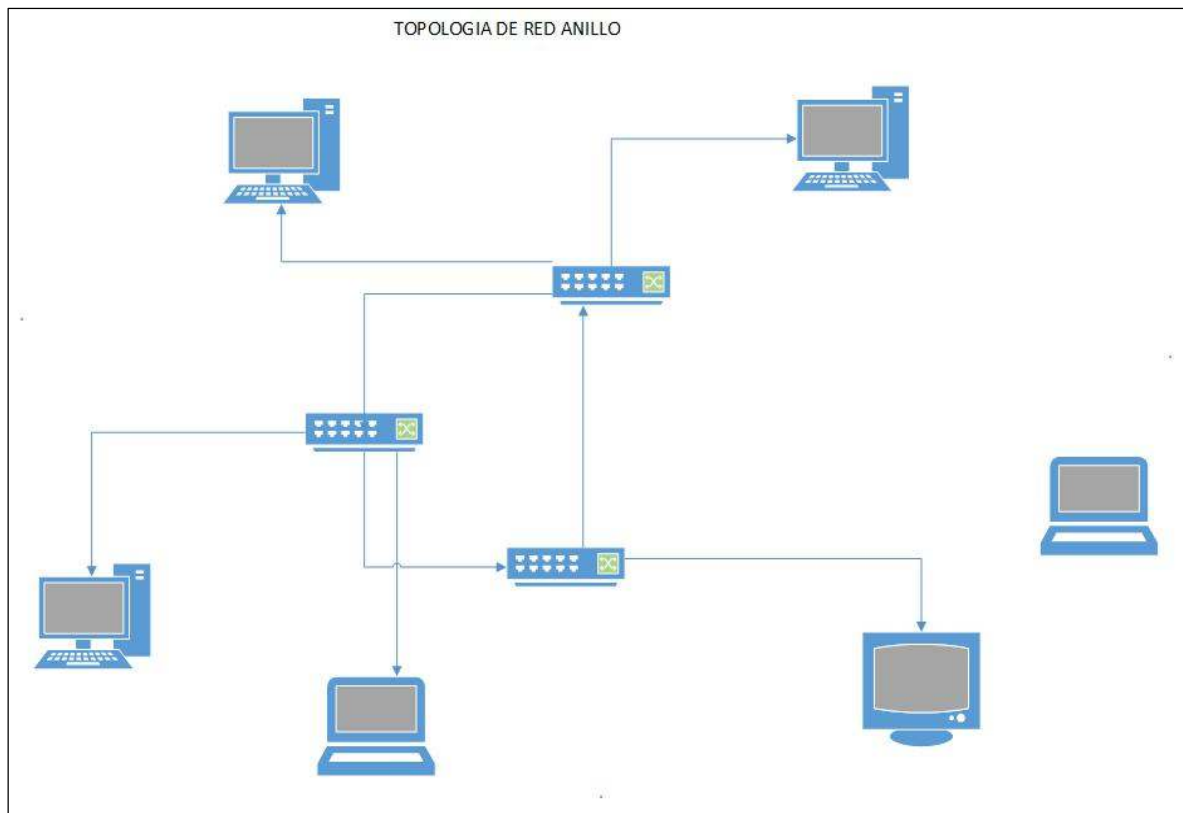
[Fuente: www24]

Figura No 2.13 Topología de red en Malla



[Fuente: www24]

Figura No 2 14 Topología de red Anillo



[Fuente: www24]

### 2.13.2 Modelo cliente servidor

El modelo cliente-servidor representa la forma en la que se producen las comunicaciones entre dos nodos de una red. En este modelo, uno de los nodos que forma parte de la comunicación tiene el rol de cliente, y otro tiene el rol de servidor.

Sin embargo, en la mayor parte de los casos, los elementos que hacen uso de este modelo son realmente aplicaciones/programas que se ejecutan dentro de los nodos. Por tanto hablaremos realmente de aplicaciones cliente y aplicaciones servidor.

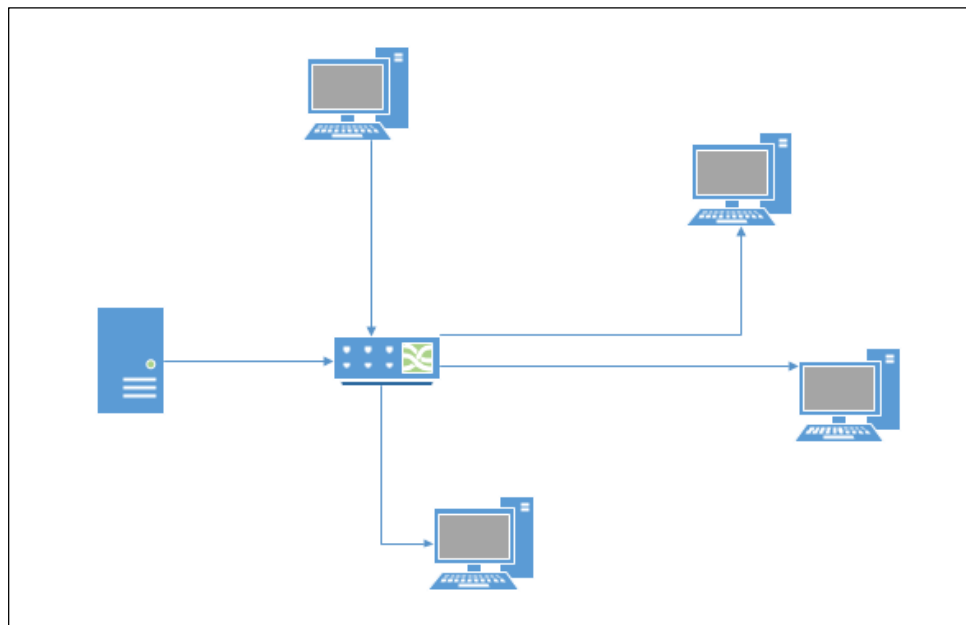


Una aplicación cliente es el elemento de la comunicación que pide o solicita un servicio de red, por ejemplo, el acceso a una página web, o la descarga de un archivo, o el envío de un email.

Una aplicación servidor es el elemento de la comunicación que responde a las peticiones de los clientes, proporcionando el servicio requerido, es decir, enviando la página web o el archivo solicitado o el email.

Lo que ocurre es que en muchas ocasiones, cuando las principales aplicaciones que están funcionando en un equipo son aplicaciones de tipo servidor, el propio equipo es llamado servidor. Sin embargo, de forma general, en un equipo, puede haber aplicaciones cliente como aplicaciones servidor funcionando al mismo tiempo. (Disponible en: [www25](#))

Figura No 2. 15 Topología de red estrella



[Fuente: [www24](#)]

## **2.13. INGENIERIA DE SOFTWARE**

Es un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos, el pilar fundamental a la hora de crear una aplicación. El ingeniero de software se encarga de toda la gestión del proyecto para que este se pueda desarrollar en un plazo determinado y con el presupuesto previsto. (Fuente: www26)

### **2.13.1. Las Métricas y la calidad de software**

El objetivo primordial de la ingeniería del software es producir un sistema, aplicación o producto de alta calidad. Para lograr este objetivo, los ingenieros de software deben emplear métodos efectivos junto con herramientas modernas dentro del contexto de un proceso maduro de desarrollo del software. Al mismo tiempo, un buen ingeniero del software y buenos administradores de la ingeniería del software deben medir si la alta calidad se va a llevar a cabo. A continuación se verá un conjunto de métricas del software que pueden emplearse a la valoración cuantitativa de la calidad de software.

El American Heritage Dictionary define la calidad como “Una característica o atributo de algo.” La definición estándar de calidad en ISO-8402. (Pressman; 2012)

### **2.13.2. Medidas de fiabilidad y de disponibilidad**

Los trabajos iniciales sobre fiabilidad buscaron extrapolar las matemáticas de la teoría de fiabilidad del hardware a la predicción de la fiabilidad del software.

Desgraciadamente, para el software lo que ocurre es lo contrario. De hecho, todos los fallos del software, se producen por problemas de diseño o de implementación. Considerando un sistema basado en computadora, una medida sencilla de la fiabilidad es el tiempo medio entre fallos (TMEF) [Mayrhauser'91], donde:

$$\text{TMEF} = \text{TMDF} + \text{TMDR}$$

(TMDF (tiempo medio de fallo) y TMDR (tiempo medio de reparación)).

Muchos investigadores argumentan que el TMDF es con mucho, una medida más útil que los defectos/KLDC, simplemente porque el usuario final se enfrenta a los fallos, no al número total de errores. Como cada error de un programa no tiene la misma tasa de fallo, la cuenta total de errores no es una buena indicación de la fiabilidad de un sistema. Por ejemplo, consideremos un programa que ha estado funcionando durante 14 meses. Muchos de los errores.

### **2.13.3. Modelo (Cocomo II)**

Cocomo II permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de coste y de escala, en los factores de coste se incluyen aspectos relacionados con la naturaleza del sistema, equipo y características propias del proyecto.

Cocomo II posee tres fases y cada uno de estos está orientado a sectores específicos del mercado de desarrollo de software y sus distintas etapas de desarrollo de software. Puntos objeto, puntos función no ajustados y líneas de código fuente, además se emplean otros parámetros relativos la reutilización, reingeniería, conversión y mantenimiento.

El procedimiento para determinar Puntos Objeto en un proyecto software se resume en:

- I. Determinar cantidad de objetos estimar la cantidad de pantallas, reportes componentes que contendrá la aplicación.
- II. Clasificar cada instancia de un objeto según sus niveles de complejidad (simple, medio o difícil).

Indirectas hacen referencia a la funcionalidad, calidad, funcionalidad, complejidad, eficiencia, fiabilidad, mantenimiento y dominio.

#### **2.13.4. Factores de calidad de estándar ISO 9126**

El estándar ISO 9126 se desarrolló como un intento por identificar los atributos de calidad para el software de computadora. El estándar identifica seis atributos clave de la calidad.

- Funcionalidad. Se aprecia evaluando el conjunto de características y capacidades del programa, la generalidad de las funciones entregadas y la seguridad del sistema global.
- Usabilidad (facilidad de empleo o uso) Se valora considerando factores humanos, la estética, consistencia y documentación general.
- Fiabilidad. Se evalúa midiendo la frecuencia y gravedad de los fallos, la exactitud de las salidas (resultados), el tiempo medio entre fallos (TMEF), la capacidad de recuperación de un fallo y la capacidad de predicción del programa.
- Rendimiento. Se mide por la velocidad de procesamiento, el tiempo de respuesta, consumo de recursos, rendimiento efectivo total y eficacia.

- Capacidad de soporte. Combina la capacidad de ampliar el programa (extensibilidad), adaptabilidad y servicios (los tres representan mantenimiento), así como capacidad de hacer pruebas, compatibilidad, capacidad de configuración, la facilidad de instalación de un sistema y la facilidad con que se pueden localizar los problemas.

### **2.13.5. Métricas basadas en la función**

La métrica de punto de función (PF) se puede usar como medio para predecir el tamaño de un sistema que se va a obtener de un modelo de análisis. Para instruir el empleo de la métrica PF, se considerará una sencilla representación del modelo de análisis mostrado en donde se representa un diagrama de flujo de datos, de una función de una aplicación de software llamada Hogar Seguro. La función administra la interacción con el usuario, aceptando una contraseña de usuario para activar/ desactivar el sistema y permitiendo consultas sobre el estado de las zonas de seguridad y varios sensores de seguridad. La función muestra una serie de mensajes de petición y envía señales apropiadas de control a varios componentes del sistema de seguridad.

El diagrama de flujo de datos se evalúa para determinar las medidas clave necesarias para el cálculo de la métrica de PF.

- Número de entradas de usuario.
- Número de salidas de usuario.
- Número de consultas de usuario.
- Número de archivos.
- Número de interfaces externas.

Una métrica de diseño arquitectónico propuesta por Henry y Kafura [Hamdi '99] también emplea la expansión y la concentración. Los autores definen una métrica de complejidad de la forma:

$$MHK = Longitud(i) \times (F_{in} + F_{out})^2$$

Donde la longitud (i) es el número de sentencias en lenguaje de programación en el módulo (i) y fin (i) es la concentración del módulo i. Henry y Kafura amplían la definición de concentración y expansión no sólo el número de conexiones de control del módulo (llamadas al módulo), sino también el número de estructuras de datos del que el módulo i recoge (concentración) o actualiza (expansión) datos.

Para calcular el MHK durante el diseño, puede emplearse el diseño procedimental para estimar el número de sentencias del lenguaje de programación del módulo i. Como en las métricas de Card y Glass mencionadas anteriormente, un aumento en la métrica de Henry-Kafura conduce a una mayor probabilidad de que también aumente el esfuerzo de integración y pruebas del módulo. (Fuente: Pressman, Ingeniería del software, 2012)

#### **2.13.6. Modelo de construcción de costos**

El modelo original COCOMO se publicó por primera vez en 1981 por Barry Boehm y reflejaba las prácticas en desarrollo de software de aquel momento. En la década y media siguiente las técnicas de desarrollo software cambiaron drásticamente la forma de realizar sistemas la necesidad de realizar sistemas más eficientes hicieron que la gente busque formas más eficientes para lograr metas y objetivos.

Estos cambios incluyen el gasto de tanto esfuerzo en diseñar y gestionar el proceso de desarrollo software como en la creación del producto software, un giro total desde los mainframe que trabajan con procesos batch nocturnos hacia los sistemas en tiempo real y un énfasis creciente en la reutilización de software ya existente y en la construcción de nuevos sistemas que utilizan componentes software a medida.

Estos y otros cambios hicieron que la aplicación del modelo COCOMO original empezara a resultar problemática. La solución al problema era reinventar el modelo para aplicarlo a los 90. Después de muchos años de esfuerzo combinado entre USC-CSE<sup>1</sup>, IRUS y UC Irvine<sup>22</sup> y las Organizaciones Afiliadas al Proyecto COCOMO II, el resultado es COCOMO II, un modelo de estimación de coste que refleja los cambios en la práctica de desarrollo de software profesional que ha surgido a partir de los años 70. Este nuevo y mejorado COCOMO resultará de gran ayuda para los estimadores profesionales de coste software. (Ana M. Moreno; Estimación de Proyectos Software; disponible en: [www27](#))

## **CAPITULO III**

### **3. MARCO APLICATIVO**

Este capítulo se enmarca en la explicación del análisis y diseño del producto, desarrollado bajo la metodología UWE, la cual se implementó en el SLIM-DNA Tiahuanacu, se tiene que explicar a los lectores que la adaptación de dicha metodología logro el resultado óptimo del producto planteado.

#### **3.1. DESCRIPCIÓN GOBIERNO AUTÓNOMO MUNICIPAL DE TIAHUANACU**

El municipio de Tiahuanacu se halla 70 km al noroeste de la ciudad de La Paz, por carretera, y a una altura de 3.885 msnm a 15 km del Lago Titicaca, cuenta con una población de 1127 habitantes aproximadamente.

El municipio de Tiahuanacu se encuentra ubicado dentro la provincia Ingavi tercera sección, está conformado por las subcentrales Huancollo, Achaca, Causaya, Chusicani, Pircuta, Pillapi y Huacullani cada una de estas subcentrales está representada por un Mallku que se encarga de velar los intereses de su región, está constituido por un alcalde y cinco concejales.

#### **3.2. INGENIRIA DE REQUERIMIENTOS**

Las funciones del sistema son una descripción de las cosas que el sistema realizará, se agrupa en paquetes con grupos lógicos el cual nos ayuda a comprender a detalle el análisis y diseño, cada paquete representa un subsistema.



Un subsistema es un entorno único y predefinido a través del cual el sistema coordina el flujo de trabajo y la utilización de recursos.

El usuario gestor, se utiliza para controlar el acceso de usuarios al sistema, que brinda privilegios para los diferentes módulos y proporcionar un estado de sesión inicial para cada usuario lanzado a un interfaz necesario para el acceso a la aplicación.

Tabla No 3.1 Ingeniería de Requerimientos

| N° | Descripción  | Estado    |
|----|--|-----------|
| 1  | Desarrollar el Modelo Web con UWE (General).                                     | Realizado |
| 2  | Modelar la Base de Datos.  | Realizado |
| 3  | Generar entidades en MySQL de acuerdo al modelado de la Base de Datos.           | Realizado |
| 4  | Configurar PHP y MySQL para desarrollo del sistema.                              | Realizado |
| 5  | Realizar pruebas en las entidades generadas.                                     | Realizado |
| 6  | Mejorar los modelos de UWE para el Registro de Beneficiarios.                    | Realizado |
| 7  | Desarrollar el Modulo de Gestión de Beneficiarios.                               | Realizado |
| 8  | Realizar pruebas del Módulo de Gestión de Beneficiarios.                         | Realizado |
| 9  | Integración del Sistema.   | Realizado |
| 10 | Desarrollar el módulo de Registro de Tutores.                                    | Realizado |
| 11 | Realizar pruebas al módulo de Registro de Tutores.                               | Realizado |
| 12 | Integración del Sistema.   | Realizado |
| 13 | Mejorar los modelos UWE para el módulo de Registro de pagos a los beneficiarios. | Realizado |
| 14 | Desarrollar el módulo de Registro de pagos.                                      | Realizado |
| 15 | Realizar pruebas al módulo de Registro de pagos.                                 | Realizado |
| 16 | Integración del Sistema.   | Realizado |
| 17 | Desarrollo del Módulo de reportes.   | Realizado |
| 18 | Corrección de Bugs.  | Realizado |
| 19 | Puesta en Marcha Final del Sistema.  | Realizado |

Fuente: [Elaboración propia]

### **3.3. GAME**

Durante esta etapa del proyecto se desarrollaron 4 iteraciones, cada una de ellas corresponde a un elemento del Sistema: Cultura e Información, enmarcada a recibir información necesaria como ser dirección tipo de discapacidad y porcentaje de inhabilidad ocupación lugar de residencia y ocupación de beneficiarios y también de tutores ya que ellos también forman parte del subsistema llamado ley de inserción laboral y de ayuda a personas con discapacidad, coadyuvando a el Registro y elaboración de Recibos y reportes.

A continuación se desglosan las actividades realizadas en cada una de las etapas la elaboración de estas tareas de manera organizada y secuencial garantiza la culminación del producto con éxito.

La estrategia para el desarrollo de cada iteración es construir en un principio los modelos de la metodología UWE y posteriormente implementarlos utilizando como elemento central la Base de Datos “MySQL”. Las clases y sus métodos fueron implementados en lenguaje PHP. Finalmente se desarrollaron las interfaces Web en base a los modelos de presentación y navegación que se han planificado anteriormente en coordinación con el personal a cargo del servicio legal integral municipal de Tiahuanacu.

#### **3.3.1. Primera iteración**

En esta iteración se desarrollan los módulos planificados y necesarios para la labor y el proceso y seguimiento de las actividades de registro, adición y actualización de los beneficiarios del pago del bono a personas con discapacidad en el municipio de Tiahuanacu.

Tabla No 3. 2 Primera Iteración

| Nro. | Tarea del Sprint                          | Tiempo (Días) | estado    |
|------|---|---------------|-----------|
| 2.1  | Realizar la planificación de la iteración | 3             | Realizado |
| 2.2  | Analizador del backlog del producto       | 4             | Realizado |
| 2.3  | Analizar los casos de uso                 | 2             | Realizado |
| 2.4  | Construir el modelo de contenidos         | 1             | Realizado |
| 2.5  | Construir el modelo de usuario            | 1             | Realizado |
| 2.6  | Construir el modelo de navegación         | 1             | Realizado |
| 2.7  | Construir el modelo de procesos           | 1             | Realizado |
| 2.8  | Construir el modelo presentación          | 1             | Realizado |
| 2.9  | Desarrollar módulo de nuevo Beneficiario  | 4             | Realizado |

Fuente: [Elaboración propia]

En la primera iteración se desarrollaron las siguientes funcionalidades del sistema en reuniones coordinadas con el personal de la unidad del servicio legal integral municipal y defensoría de la niñez y adolescencia:

- Módulo de registro de Beneficiarios.
- Módulo de edición de registro.
- Módulo de adición de Beneficiario.
- Módulo de depuración temporal de tutor.
- Módulo de alerta de caducidad de carnet codepedis.

Figura No 3.1 Módulo Beneficiarios listado

Beneficiarios Listado

+ Agregar Beneficiarios

Mostrar 10 registros por pagina

Buscar: Buscar registros

| #  | Cedula I. | Nombres       | Apellidos       | Vigencia   | Discapacidad         | Tutor                           | Opciones           |
|----|-----------|---------------|-----------------|------------|----------------------|---------------------------------|--------------------|
| 1  | 6910061   | Teodoro       | Bautista Quispe | 2023-01-30 | Multiple-Intelectual | Maria Quispe De Bautista        | [Iconos de acción] |
| 2  | 6047598   | Victoria      | Acarapi Limachi | 2021-10-02 | Intelectual          | Valentina Limachi De Acarapi    | [Iconos de acción] |
| 3  | 6909815   | Justina       | Calle Quispe    | 2022-12-18 | Fisica-Motora        | No Tiene                        | [Iconos de acción] |
| 4  | 8354487   | Asunta        | Choque De Patti | 2018-10-24 | Fisica-Motora        | Policarpio Patti Marin          | [Iconos de acción] |
| 5  | 8426489   | Joel          | Copa Mamani     | 2021-02-17 | Auditiva             | Zacarias Copa Quispe            | [Iconos de acción] |
| 6  | 13787673  | Marco Antonio | Juchani Coronel | 2018-10-13 | Intelectual          | Antonia Coronel Vda. De Juchani | [Iconos de acción] |
| 7  | 6046173   | Patricia      | Mamani Salasar  | 2022-12-07 | Intelectual          | Santos Mamani Mamani            | [Iconos de acción] |
| 8  | 8436778   | Bertha        | Mamani Huanca   | 2019-12-23 | Intelectual          | Calixto Mamani Huanca           | [Iconos de acción] |
| 9  | 9881150   | Dionisio      | Choque Vargas   | 2018-10-13 | Intelectual          | Juana Vargas Choque             | [Iconos de acción] |
| 10 | 6925329   | Gabe          | Apaza Nina      | 2022-12-14 | Intelectual          | Celia Apaza De Condori          | [Iconos de acción] |

Fuente: [Elaboración Propia]

Figura No 3. 2 Módulo Beneficiarios Adicionar

Beneficiario Nuevo

Cedula de identidad:

Nombre:

Apellido:

Género:

Comunidad:

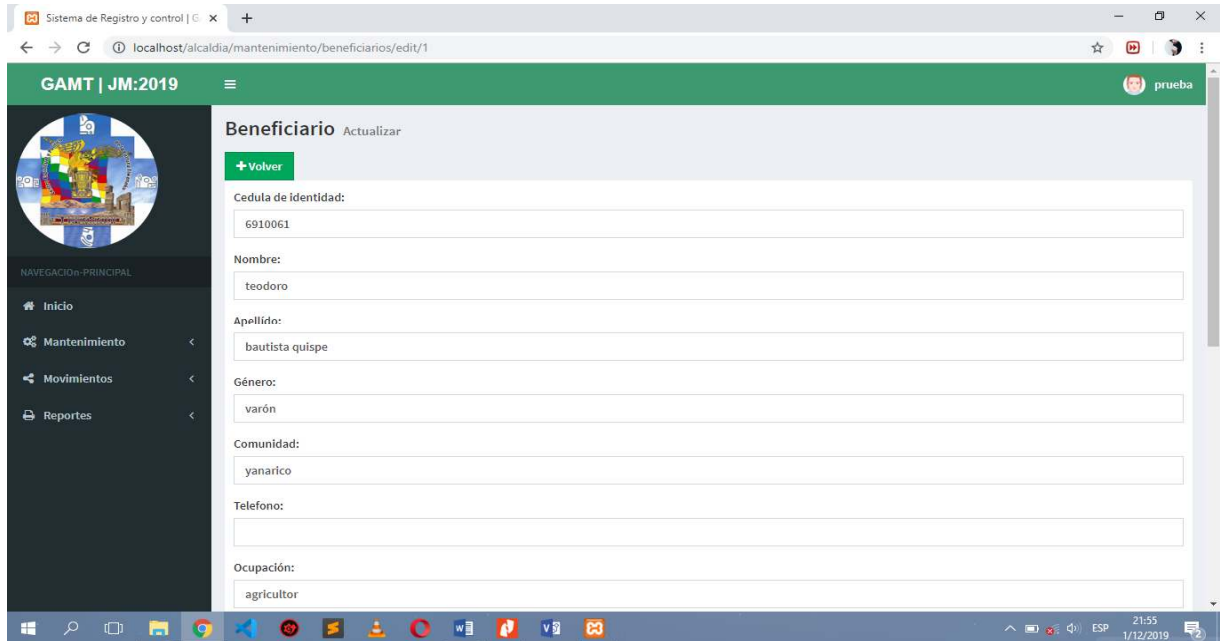
Telefono:

Ocupación:

Fecha de Nacimiento:

Fuente: [Elaboración Propia]

Figura No 3.3 Módulo Beneficiarios Actualizar



Fuente: [Elaboración Propia]

### 3.3.2. Segunda iteración

En la segunda se desarrollan los módulos necesarios para el proceso y seguimiento de labores administrativas de registro adición y edición de los tutores.

Tabla No 3.3 Segunda Iteración

| Nro. | Tarea del Sprint                          | Tiempo (Días) | estado    |
|------|---|---------------|-----------|
| 2.1  | Realizar la planificación de la iteración | 3             | Realizado |
| 2.5  | Construir el modelo de usuario            | 1             | Realizado |
| 2.6  | Construir el modelo de navegación         | 1             | Realizado |
| 2.7  | Construir el modelo de procesos           | 1             | Realizado |
| 2.9  | Desarrollar módulo de nuevo Tutor         | 4             | Realizado |

Fuente: [Elaboración propia]

En la segunda iteración se desarrollaron las siguientes funcionalidades para el sistema:

- Módulo de registro de Tutores.
- Módulo de edición de registro.
- Módulo de adición de Tutores.

Figura No 3.4 Módulo Tutores Listado

The screenshot shows a web application interface for 'Sistema de Registro y control'. The page title is 'Tutores Listado'. There is a green header with 'GAMT | JM:2019' and a user profile 'prueba'. A sidebar on the left contains navigation links: 'Inicio', 'Mantenimiento', 'Movimientos', and 'Reportes'. The main content area features a table of tutors with the following data:

| C.I.    | Nombre    | Apellidos              | Telefono | Dirección                             | Fecha Nacimiento | Opciones        |
|---------|-----------|------------------------|----------|---------------------------------------|------------------|-----------------|
| 432223  | Carmelo   | Apaza Yupanqui         | 72015816 | Chuscani                              | 1947-07-16       | [Edit] [Delete] |
| 432251  | Pedro     | Apaza Huanca           | 71589475 | C/F. Plzarro Nro. 750 Z. Vino Tinto   | 1946-02-22       | [Edit] [Delete] |
| 444429  | Pastora   | Calle Vda. De Choque   |          | Yanarico                              | 1952-03-29       | [Edit] [Delete] |
| 2010592 | Calixto   | Mamani Huanca          | 77592518 | Yanamani                              | 1944-04-16       | [Edit] [Delete] |
| 2124100 | Flora     | Aduviri Vda. De Apaza  | 76297128 | Caluyo                                | 1954-10-26       | [Edit] [Delete] |
| 2125714 | Zacarias  | Copa Quispe            | 71908470 | Huacullani                            | 1956-11-05       | [Edit] [Delete] |
| 2504076 | Mercedes  | Mamani Chinche         | 71598450 | Tiahuanacu                            | 1956-09-24       | [Edit] [Delete] |
| 2512254 | Rene      | Marquez Choque         |          | Tiahuanacu                            | 1965-02-19       | [Edit] [Delete] |
| 2512274 | Benedicta | Choque Pati De Peralta | 76206360 | AV-12 Entre 29 No. 96 Z. Villa Tunari | 1953-05-07       | [Edit] [Delete] |
| 2566907 | Maria     | Quispe De Bautista     |          | Yanarico                              | 1940-12-06       | [Edit] [Delete] |

Fuente: [Elaboración propia]

### 3.3.3. Tercera iteración

En esta iteración se desarrollaron los sistemas para el almacenamiento de información del sistema de registro y control para el pago del bono a personas con discapacidad este módulo es uno de los más importantes porque lo que se busca es querer llevar un control adecuado de registro de pagos,

Tabla No 3.4 Tercera Iteración

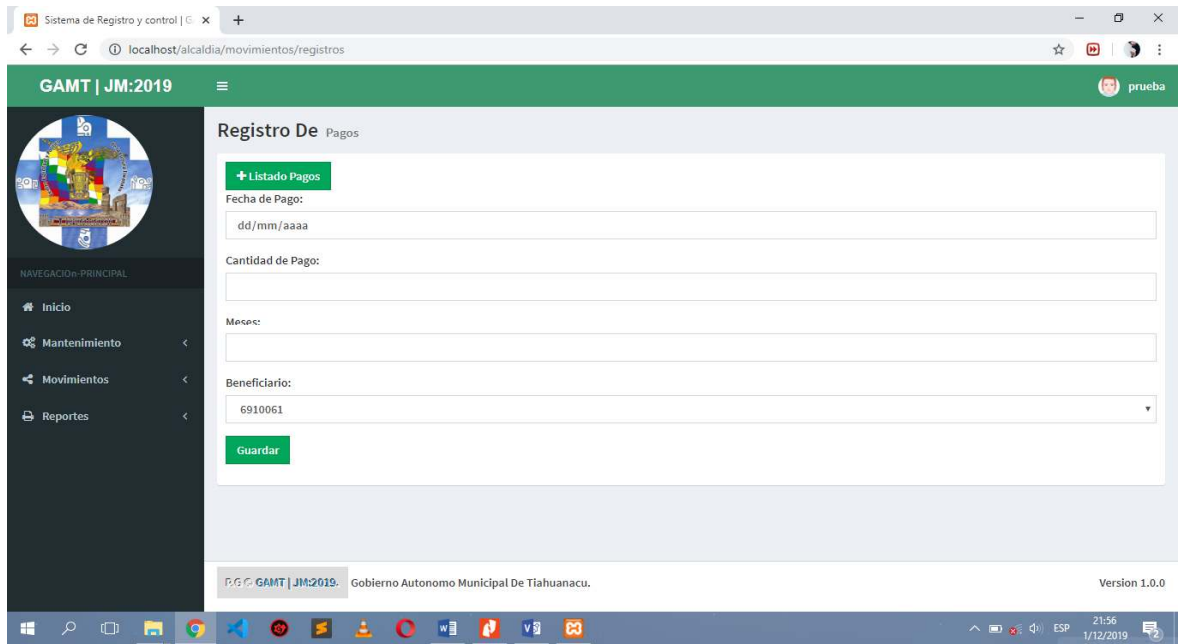
| Nro. | Tarea del Sprint                          | Tiempo | estado    |
|------|---|--------|-----------|
| 3.1  | Realizar la planificación de la iteración | 4      | Realizado |
| 3.2  | Analizador del backlog del producto       | 4      | Realizado |
| 3.3  | Analizar los casos de uso                 | 3      | Realizado |
| 3.4  | Construir el modelo de contenidos         | 1      | Realizado |
| 3.5  | Construir el modelo de usuario            | 1      | Realizado |
| 3.6  | Construir el modelo de navegación         | 1      | Realizado |
| 3.7  | Construir el modelo de procesos           | 1      | Realizado |
| 3.8  | Construir el modelo presentación          | 1      | Realizado |
| 3.9  | Desarrollar módulo de registro de pagos   | 3      | Realizado |

Fuente: [Elaboración propia]

En la tercera iteración las actividades se dedicaron a las funcionalidades que tienen que ver con el registro de pagos y las tareas que implica el desarrollar esta función, dentro de este módulo lo que se busco fue el manejar datos generales, pero también que se pueda especificar datos de registros de algún cliente en específico, se desarrollaron las siguientes funcionalidades para el sistema:

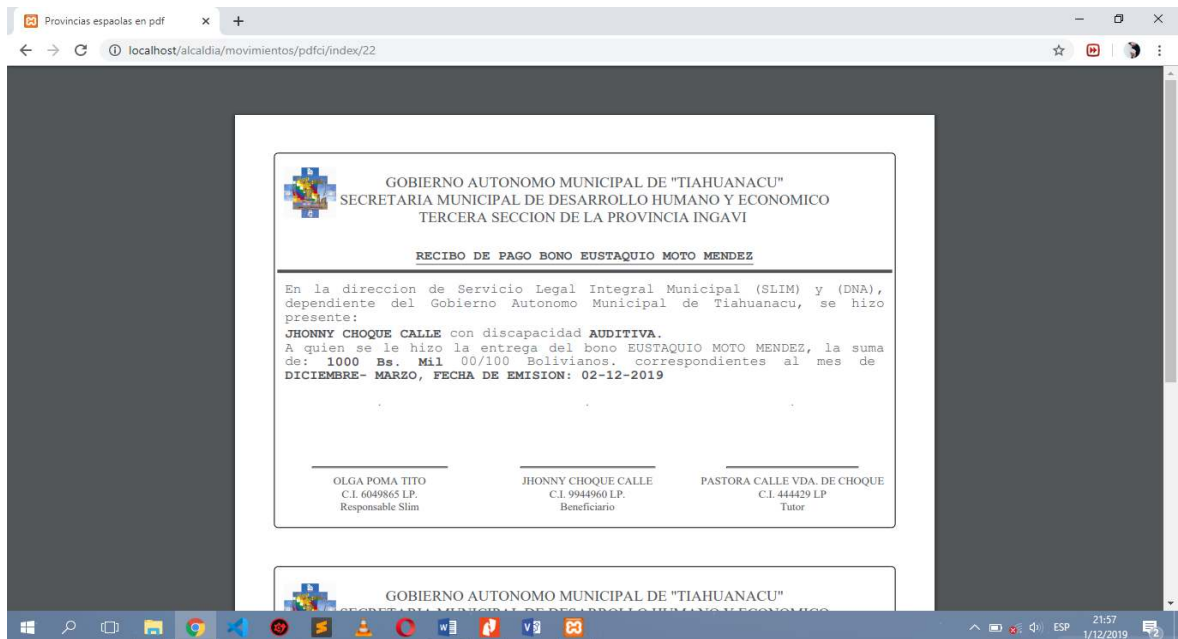
- Módulo de registro de pagos.
- Módulo de filtrado de datos.
- Recibo de pagos.

Figura No 3.5 Módulo Registro de Pagos



Fuente: [Elaboración Propia]

Figura No 3.6 Módulo Recibo de Pagos



Fuente: [Elaboración Propia]



### 3.3.4. Cuarta iteración

Durante la cuarta iteración se desarrollaron los elementos pertenecientes al sistema de control de usuarios y reportes. Las actividades realizadas durante esta iteración se observa en la siguiente tabla, constituye lo siguiente:

Tabla No 3.5 Cuarta Iteración

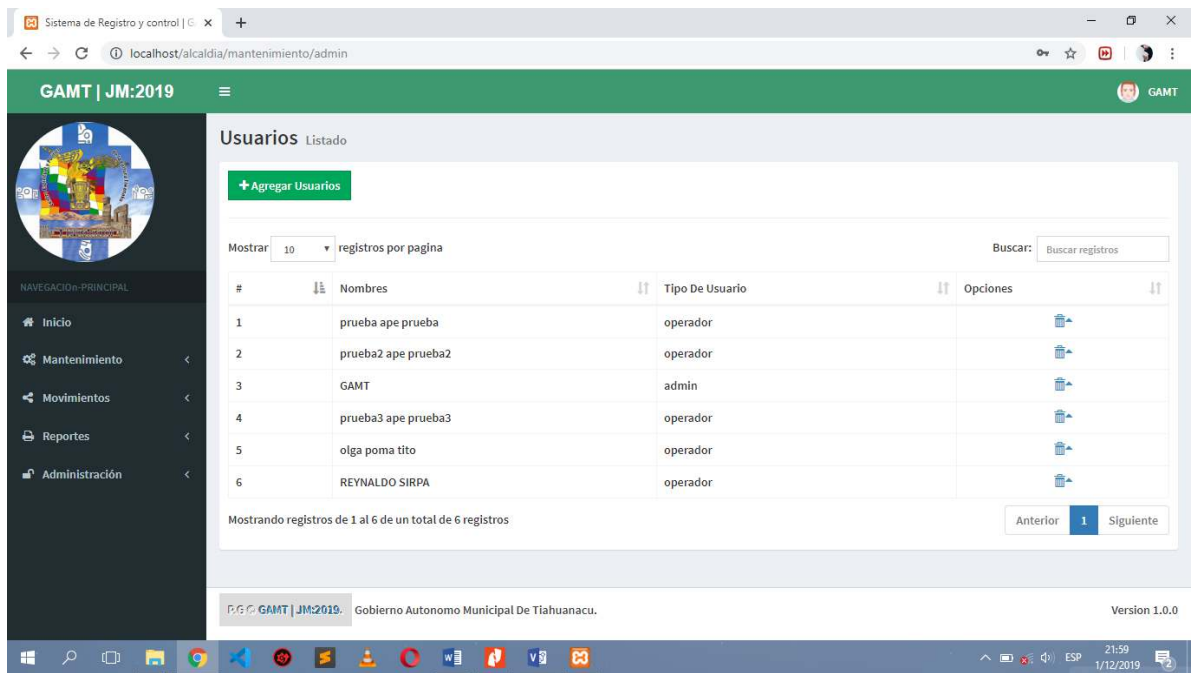
| Nro. | Tarea del Sprint  | Tiempo (Días) | estado    |
|------|---|---------------|-----------|
| 1.1  | Realizar la planificación de la iteración               | 4             | Realizado |
| 1.2  | Analizador del backlog del producto                     | 4             | Realizado |
| 1.3  | Analizar los casos de uso                               | 2             | Realizado |
| 1.4  | Construir el modelo de contenidos                       | 1             | Realizado |
| 1.5  | Construir el modelo de usuario                          | 1             | Realizado |
| 1.6  | Construir el modelo de navegación                       | 1             | Realizado |
| 1.7  | Construir el modelo de procesos                         | 1             | Realizado |
| 1.8  | Construir el modelo presentación                        | 1             | Realizado |
| 1.9  | Desarrollar módulo de control de usuarios según su rol. | 4             | Realizado |

Fuente: [Elaboración propia]

Las funcionalidades realizadas fueron por la importancia que tiene el contar con distintos tipos de usuarios que hagan control y operación, las funcionalidades realizadas son:

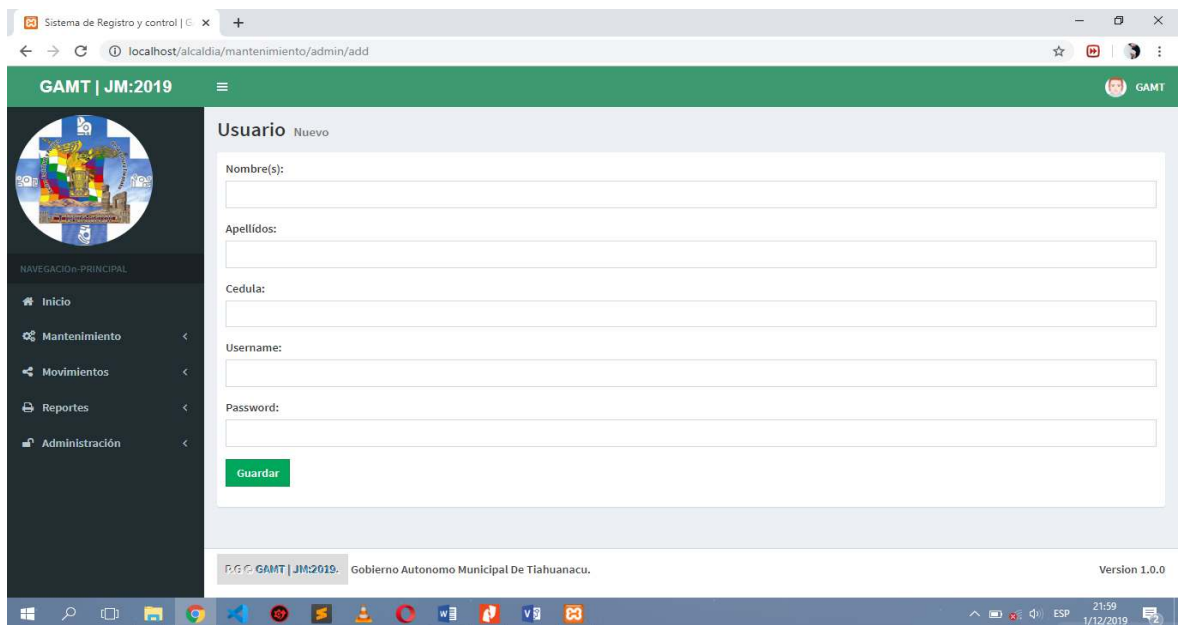
- Módulo de control de usuarios.
- Módulo de copias de seguridad de seguridad.
- Módulos de reportes.

Figura No 3.7 Módulo Usuarios Lista



Fuente: [Elaboración Propia]

Figura No 3.8 Módulo adicionar Usuario



Fuente: [Elaboración propia]

### 3.3.5. Modelo de casos de uso

El modelo de casos de uso es un diagrama del sistema que contiene actores. Casos de uso y sus perspectivas relacionales. A continuación se describen las características de los actores identificados en el manejo e identificación del sistema.

Figura No 3.9 Descripción de Actores

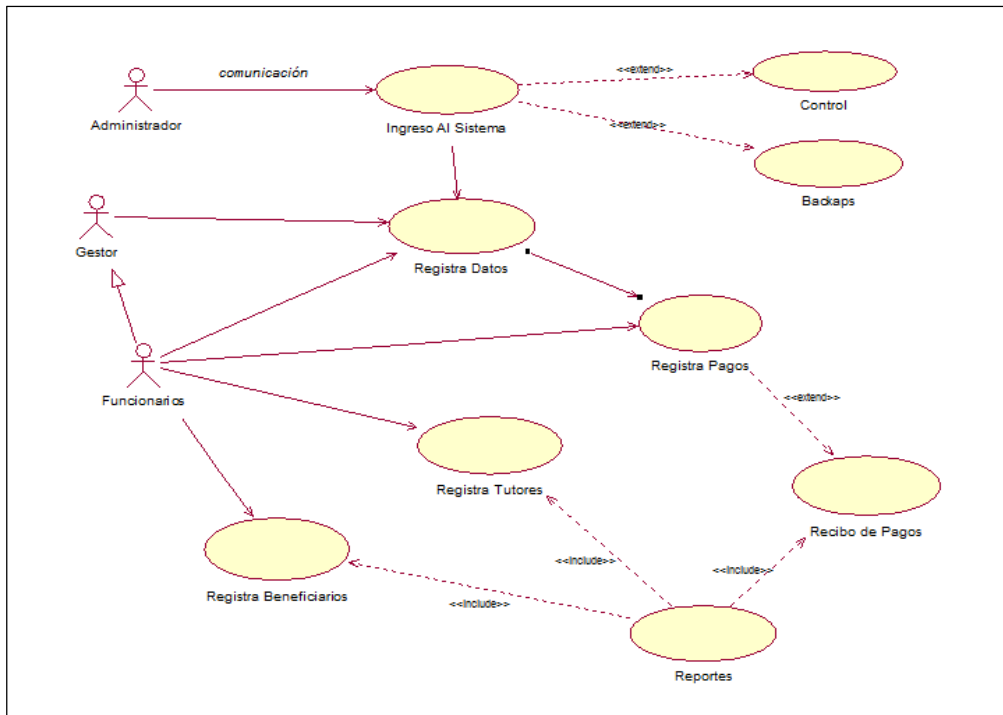
| Actores  | Descripción  |
|--|--|
| <b>Administrador jefe de unidad Slim</b>       | Toma conocimiento de los reportes que se entregan cada mes o cuando se necesita.     |
| <b>Funcionario de la unidad trabajo social</b> | Realiza registro de beneficiarios y tutores si es que los tuviera o fuera necesario. |
| <b>Funcionario de la unidad Psicólogo</b>      | Realiza registro de beneficiarios y tutores si es que los tuviera o fuera necesario. |
| <b>Beneficiario</b>                            | Interesado de la información fechas de pagos.  |
| <b>Tutor de beneficiario</b>                   | Interesado y responsable de beneficiarios si es necesario.                           |

Fuente: [Elaboración propia]

#### 3.3.5.1. Diagrama de casos de uso general

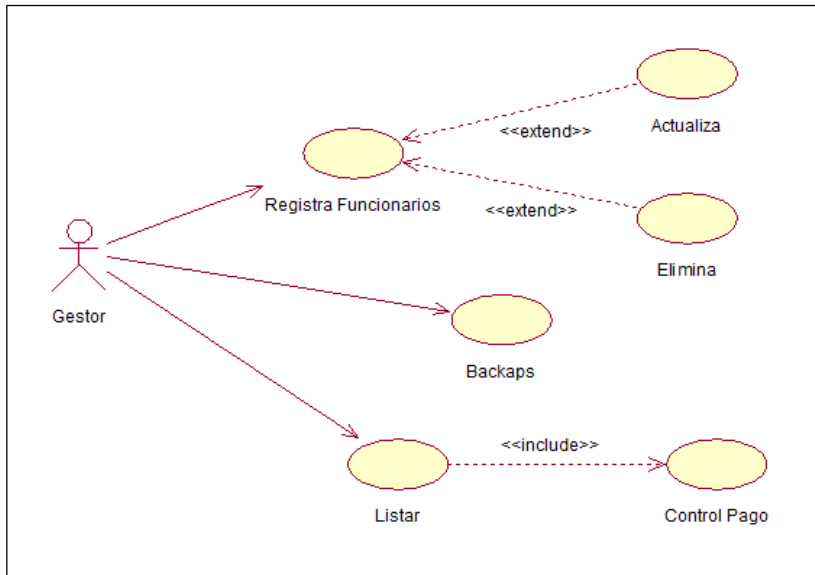
Los casos de uso describirán la secuencia de evento de un actor, es decir es un documento narrativo o imagen narrativa de la interacción que desarrollan los usuarios dentro del sistema los cuales cumplen un rol específico y tareas adaptadas a lo que se enmarca en la ley 977 y el sistema.

Figura No 3.10 Diagrama de Caso de uso General



Fuente: [Elaboración Propia]

Figura No 3.11 Caso de uso Gestor



Fuente: [Elaboración Propia]

### 3.3.5.2. Descripción de caso de uso

Tabla No 3.6 Descripción de Caso de Uso "Registro de Gestor"

| Caso de Uso        | Registro de Gestor   |
|--------------------|--|
| <b>Actor</b>       | Administrador  |
| <b>Descripción</b> | El sistema debe poder almacenar los datos personales del funcionario (C.I., Apellidos, Nombres, dirección, teléfono). Para gestionar de forma adecuada se debe llevar un archivo total del personal. |
| <b>Tipo</b>        | Primario   |

Fuente: [Elaboración propia]

Tabla No 3.7 Descripción de Caso de Uso "Asignar rol al usuario"

| Caso de Uso        | Asignar rol al usuario   |
|--------------------|--|
| <b>Actor</b>       | Administrador  |
| <b>Descripción</b> | El sistema debe poder almacenar los datos de los Usuario (Login, password), para administrar de forma segura los usuarios del sistema. |
| <b>Tipo</b>        | Primario.  |

Fuente: [Elaboración propia]

Tabla No 3.8 Descripción de Caso de Uso "Registro de Beneficiario"

| Caso de Uso        | Registro de Beneficiario  |
|--------------------|---|
| <b>Actor</b>       | Funcionario público   |
| <b>Descripción</b> | El sistema debe poder almacenar los datos personales del Beneficiario y de su tutor si lo requiere (C.I., Apellidos, Nombres, dirección, teléfono, Nro. Codepedis, y fecha en que se registró). Para administrar de forma adecuada. |
| <b>Tipo</b>        | Primario  |

Fuente: [Elaboración propia]

Tabla No 3.9 Descripción de Caso de Uso "Registro de Tutor"

| <b>Caso de Uso</b> | <b>Registro de Tutor</b>  |
|--------------------|---|
| <b>Actor</b>       | Funcionario público   |
| <b>Descripción</b> | El sistema debe poder almacenar los datos personales del Tutor a cargo del beneficiario si lo requiere (C.I., Apellidos, Nombres, dirección, teléfono). Para administrar de forma adecuada se debe llevar un archivo total de los tutores que tienen a cargo a un beneficiario. |
| <b>Tipo</b>        | Primario  |

Fuente: [Elaboración propia]

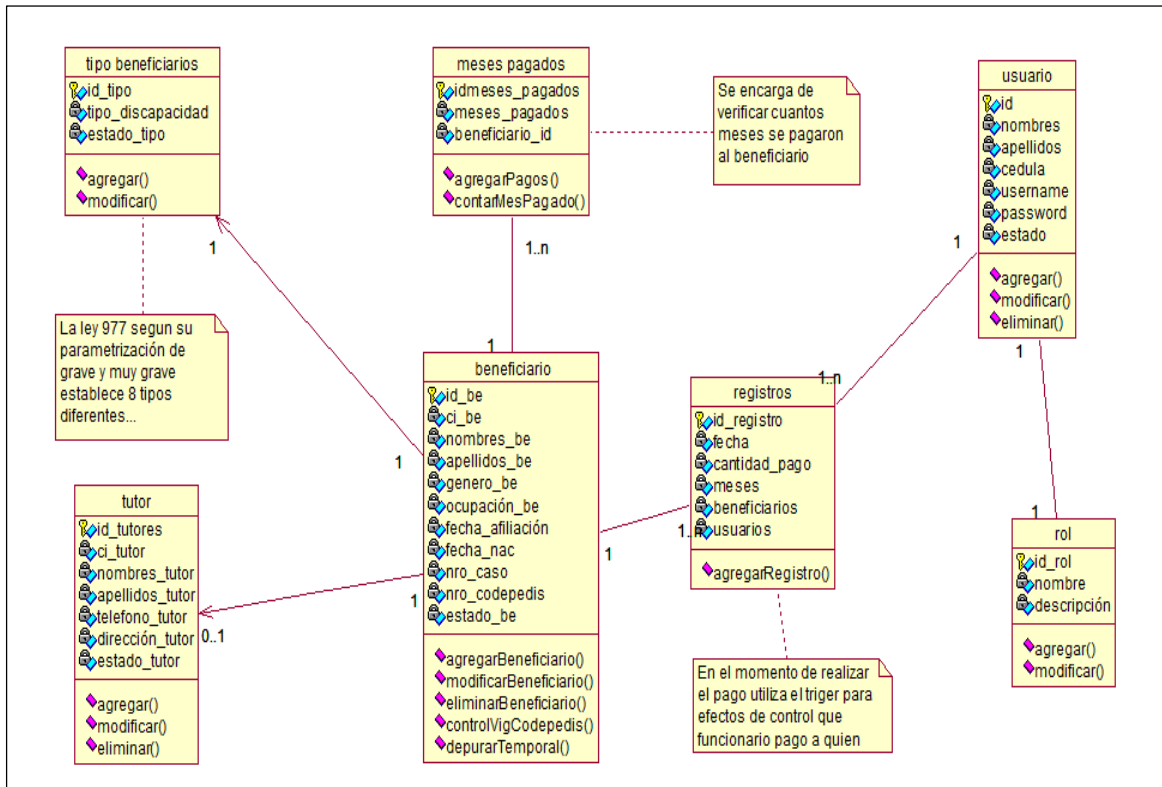
Tabla No 3.10 Descripción de Caso de Uso "Registro de Pagos"

| <b>Caso de Uso</b> | <b>Registro de Pagos</b>  |
|--------------------|---|
| <b>Actor</b>       | Funcionario público   |
| <b>Descripción</b> | El sistema debe poder almacenar los datos que se generan a partir de los pagos que se van realizando trimestralmente (fecha, nombre, apellido, cantidad del pago, meses pagados, beneficiario, encargado de pago). Para administrar de forma adecuada se debe llevar un archivo total de los pagos que se realizan en el municipio de Tiahuanacu. |
| <b>tipo</b>        | Primario  |

Fuente: [Elaboración propia]

### 3.3.6. Diagrama de clases

Figura No 3.12 Diagrama de clases

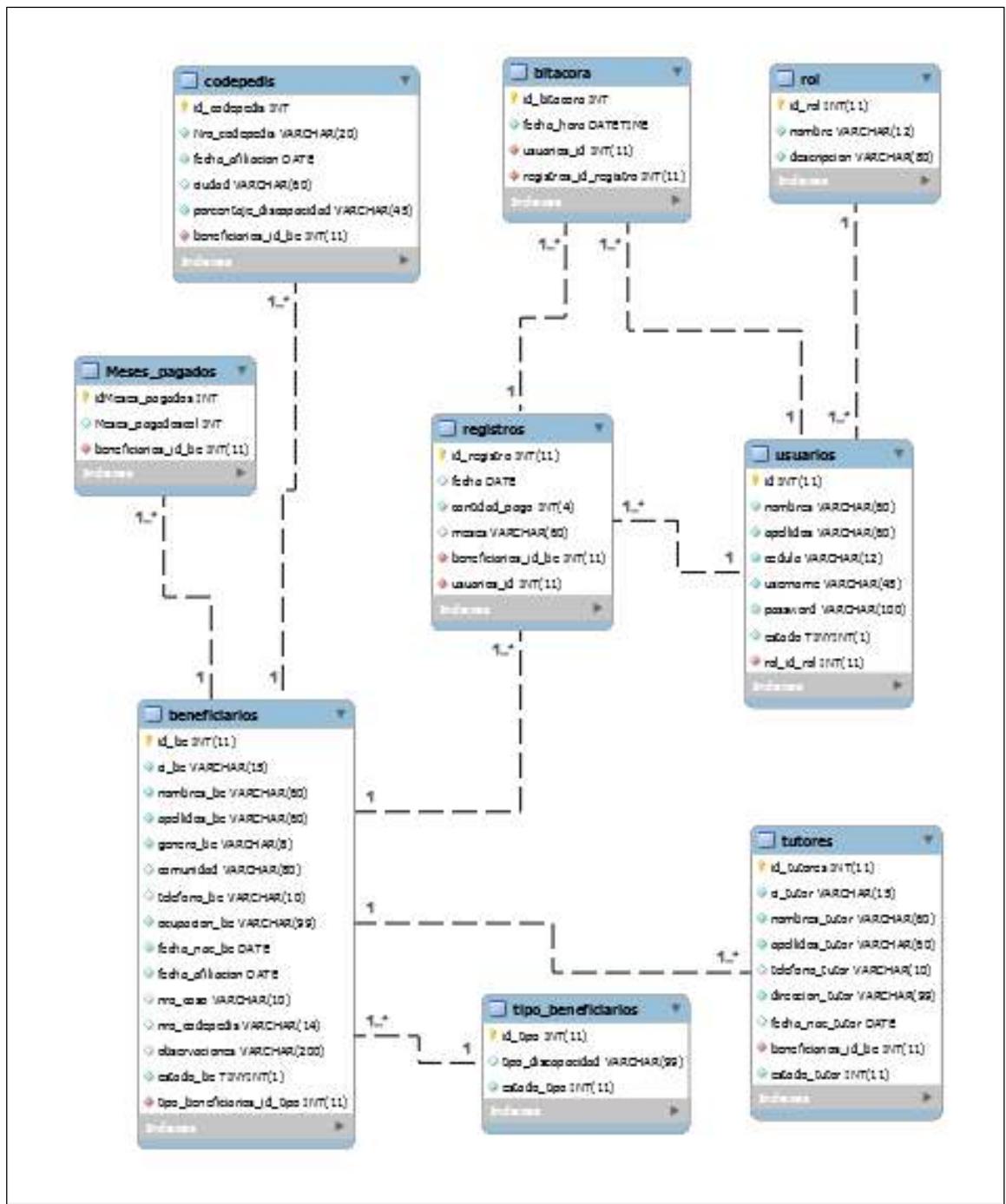


Fuente: [Elaboración propia]

### 3.3.6. Modelado de la base de datos

Una vez revisado los requerimientos para el sistema (Casos de Uso), las entidades encontradas para la Base de Datos del “Sistema de registro y control para el pago del bono a personas con discapacidad: Caso municipio de Tiahuanacu”. Entonces la estructura relacional de la Base de Datos generada para el Sistema se puede observar en la siguiente figura.

Figura No 3.13 Modelo Físico de la Base de Datos



Fuente: [Elaboración Propia]



Tabla No 3.11 Beneficiarios

| Nro | REGISTRO            | TIPO DE DATO | DESCRIPCIÓN  |
|-----|---------------------|--------------|--|
| 1   | Id Beneficiario     | Int(4)       | Se asigna un número único a cada beneficiario.                                       |
| 2   | Ci                  | Int(12)      | El ci es una medida de control para relacionar con el número de codepedis            |
| 3   | Nombres             | Varchar(60)  | Para realizar un seguimiento por persona   |
| 4   | Apellidos           | Varchar(60)  | Para realizar un seguimiento por persona   |
| 5   | Genero              | Varchar(8)   | Obtener el porcentaje de beneficiarios que recibe este bono.                         |
| 6   | Comunidad           | Varchar(80)  | El municipio está compuesto de 8 subcentrales.                                       |
| 7   | Teléfono            | Varchar(10)  | Es necesario para poder realizar seguimiento de los casos.                           |
| 8   | Ocupación           | Varchar(80)  | Para saber si son o no desempleados y poder brindar ayuda.                           |
| 9   | Fecha de nacimiento | Date         | Para cotejar con documentación presentada.   |
| 10  | Fecha de Afiliación | Date         | Para control y mensaje de alerta en caso de que los carnets pasen a ser invalidados. |
| 11  | Nro codepedis       | Varchar(14)  | Llevar registro con relación a plataforma existente (Eustaquio moto mendez).         |
| 12  | Observaciones       | Varchar(90)  | Los motivos por los cuales se realizan determinadas acciones.                        |

Fuente: [Elaboración Propia]

Tabla No 3.12 Tutores

| Nro | Registro            | Tipo de Dato | Descripción                |
|-----|---------------------|--------------|----------------------------|
| 1   | Id Tutor            | Int(4)       | Numero único para un tutor |
| 2   | Ci                  | Int(12)      | Control del tutor          |
| 3   | Nombres             | Varchar(60)  | Seguimiento del tutor      |
| 4   | Apellidos           | Varchar(60)  | Seguimiento del tutor      |
| 5   | Teléfono            | Varchar(10)  | Seguimiento del tutor      |
| 6   | Dirección           | Varchar(80)  | Comunidad donde vive       |
| 7   | Fecha de nacimiento | Date         | Si puede ser o no tutor    |
| 8   | Observaciones       | Varchar(90)  | Registro de actividades    |

Fuente: [Elaboración Propia]

Tabla No 3.13 Registros

| Nro | Registro      | Tipo de Dato | Descripción                        |
|-----|---------------|--------------|------------------------------------|
| 1   | Id registro   | Int(4)       | Numero único de registro           |
| 2   | Fecha de pago | Date         | Fecha en la que se realiza el pago |
| 3   | Meses         | Varchar(60)  | Los meses que se pagaron           |
| 4   | Quien a quien | Int(4)       | Relacionado entre dos tablas       |

Fuente: [Elaboración Propia]

Tabla No 3.14 Mes Pagado

| Nro | Registro         | Tipo de Dato | Descripción                                  |
|-----|------------------|--------------|--|
| 1   | Id Meses Pagados | Int(4)       | Número único de meses pagados durante el año |
| 2   | Mes Pagado       | Date         | Contar los meses que se pagaron              |
| 3   | Beneficiario     | Varchar(60)  | A quien se le va pagar                       |

Fuente: [Elaboración Propia]

Tabla No 3.15 Tipo de Beneficiarios

| Nro | Registro                | Tipo de Dato | Descripción             |
|-----|-------------------------|--------------|-------------------------|
| 1   | Id Tipo de Beneficiario | Int(4)       | Numero único            |
| 2   | Tipo de Discapacidad    | Date         | 8 tipos de discapacidad |
| 3   | Estado                  | Varchar(60)  | Dar de baja un estado   |

Fuente: [Elaboración Propia]

Tabla No 3.16 Usuarios

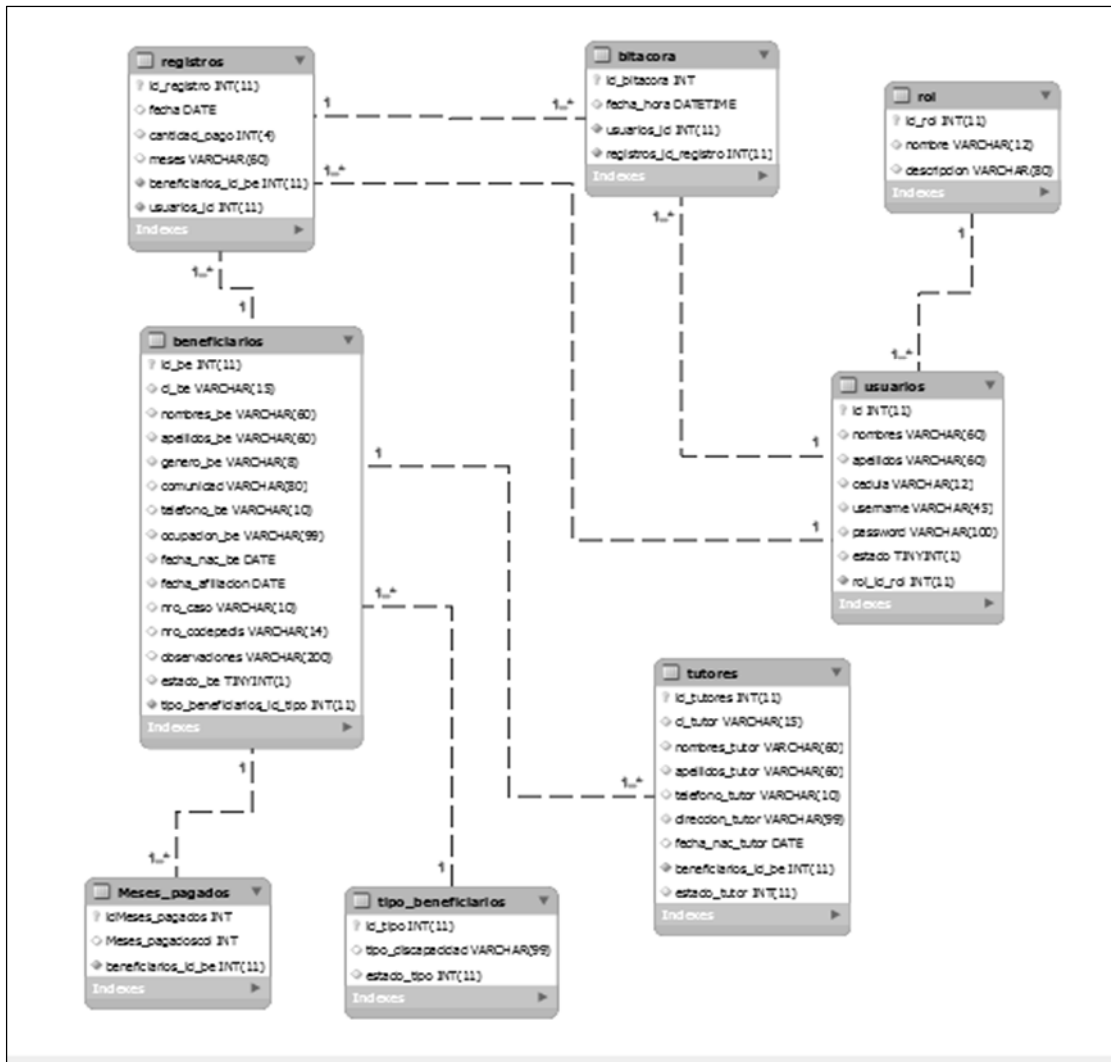
| Nro | Registro            | Tipo de Dato | Descripción                 |
|-----|---------------------|--------------|-----------------------------|
| 1   | Id Usuario          | Int(4)       | Número único                |
| 2   | Cedula de Identidad | Int(12)      | Del funcionario público     |
| 3   | Nombres             | Varchar(60)  | Seguimiento del funcionario |
| 4   | Apellidos           | Varchar(60)  | Seguimiento del funcionario |
| 5   | Username            | Varchar(45)  | Para login                  |
| 6   | Password            | Varchar(100) | Para login                  |
| 7   | Rol de Usuario      | Varchar(90)  | Privilegios de usuarios     |

Fuente: [Elaboración Propia]

### 3.3.7. Modelo de contenidos

Este modelo especifica cómo se encuentra relacionados los contenidos del sistema, es decir, define la estructura de los datos que se encuentran alojados en el sistema, el modelo de contenidos contiene la información relevante almacenada en el sistema, como se estructura y como se relaciona. Lo que se busca específicamente es cómo se encuentra relacionados los contenidos del sistema, es decir, define la estructura de los datos que se encuentran alojados en el sistema.

Figura No 3.14 Modelo de contenidos

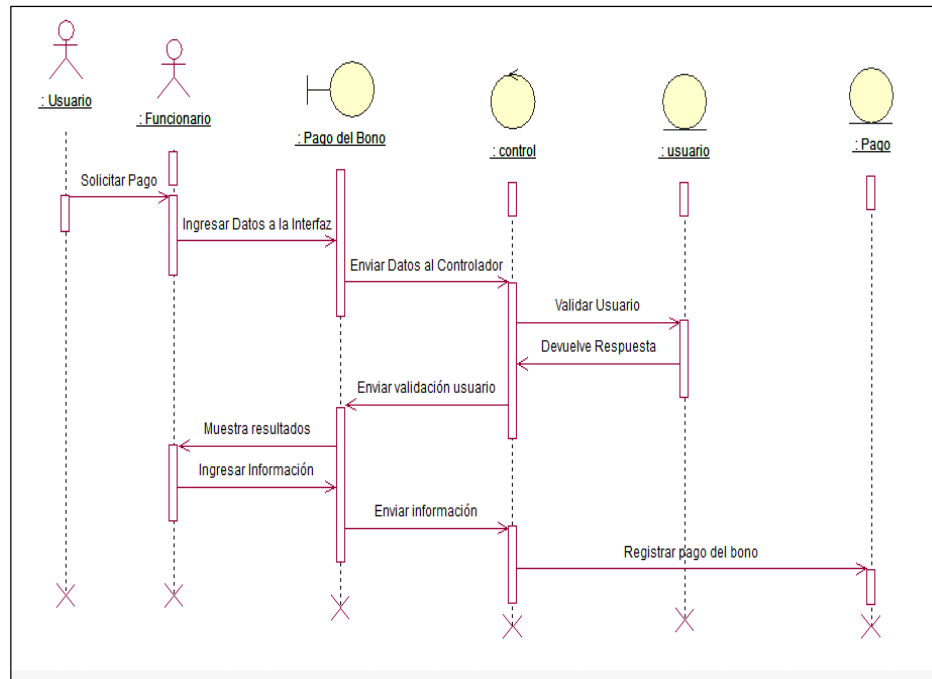


Fuente: [Elaboración Propia]

### 3.3.8. Modelo de Secuencia

En este modelo se especifica la relación interna de un conjunto de objetos, es decir cómo se relaciona cada entidad dentro del sistema con los demás, con la cual en definitiva es cómo se comporta el sistema. El diagrama que se representa a continuación es un modelo simplificado ya que representa la labor de secuencia en el sistema.

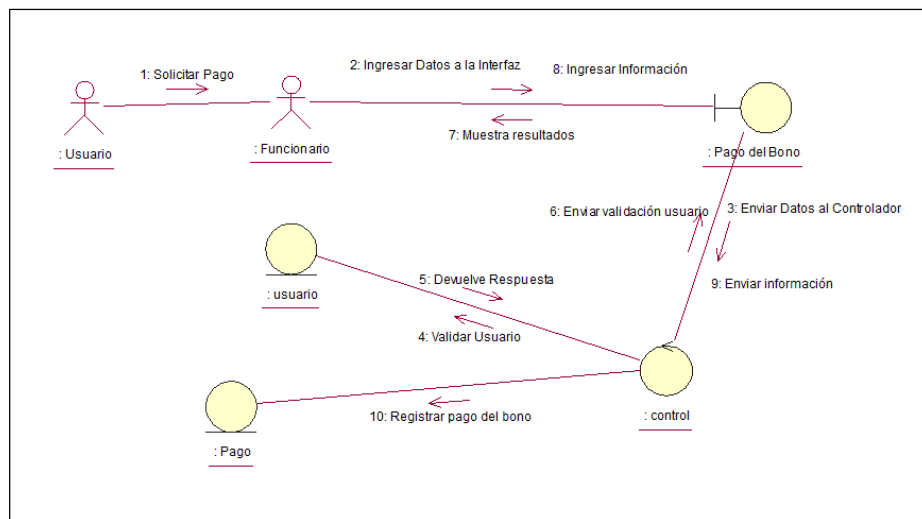
Figura No 3.15 Diagrama de Secuencia



Fuente: [Elaboración Propia]

### 3.3.9. Modelo de colaboración

Figura No 3.16 diagrama de colaboración

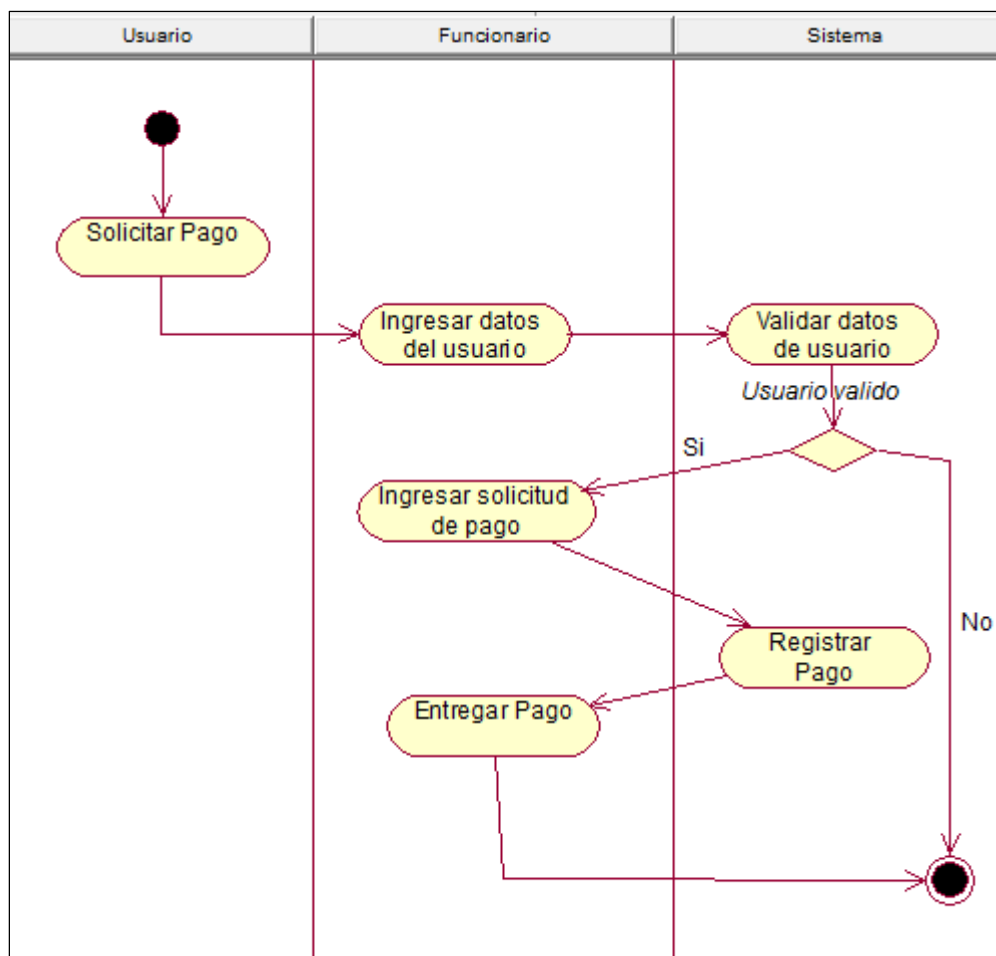


Fuente: [Elaboración Propia]

### 3.3.10. Modelo de actividades

Los diagramas de actividades, junto con los diagramas de casos de uso y los diagramas de máquina de estados, son considerados diagramas de comportamiento porque describen lo que debe suceder en el sistema que se está modelando. Los diagramas de actividades ayudan a que las personas en las áreas de negocios y desarrollo de una organización se integren para comprender el mismo proceso y comportamiento.

Figura No 3.17 Diagrama de actividades

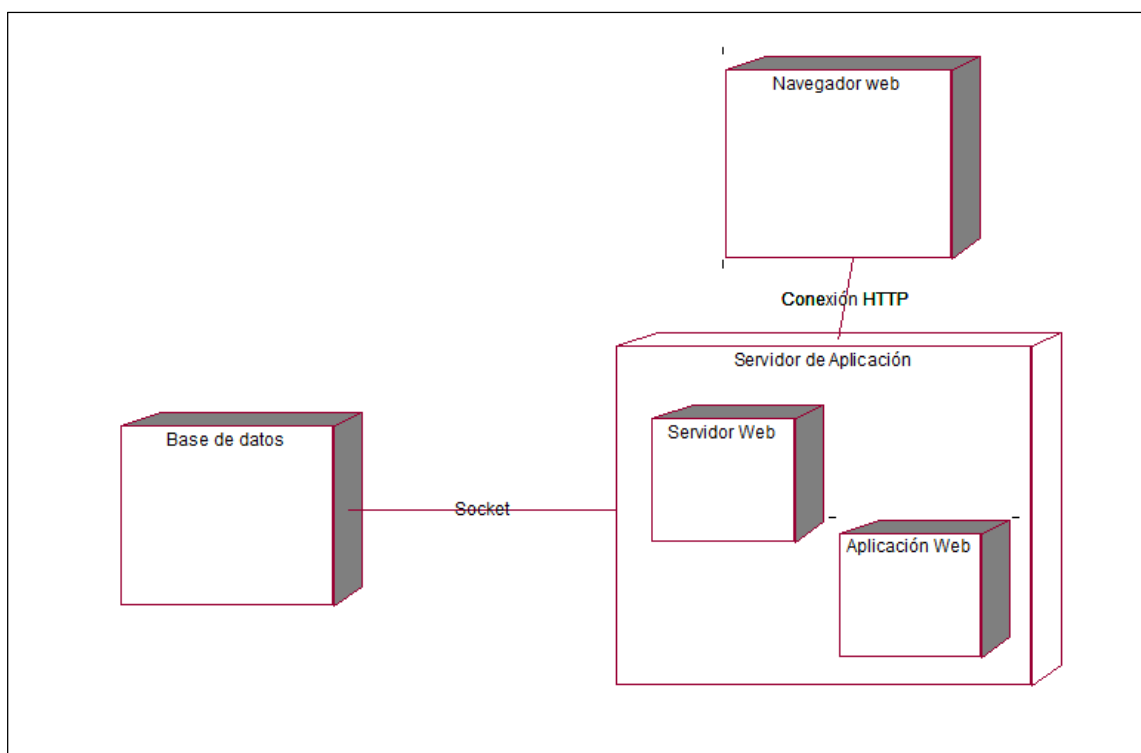


Fuente: [Elaboración Propia]

### 3.3.11. Modelo de despliegue

Se contemplan las clases de navegación y de procesos que pertenecen a cada módulo, las propiedades que contiene por composición se representan como un rectángulo en el que un elemento queda contenido en otro. Esto muestra la configuración de los elementos de hardware y muestra cómo los elementos y artefactos del software se trazan en esos nodos.

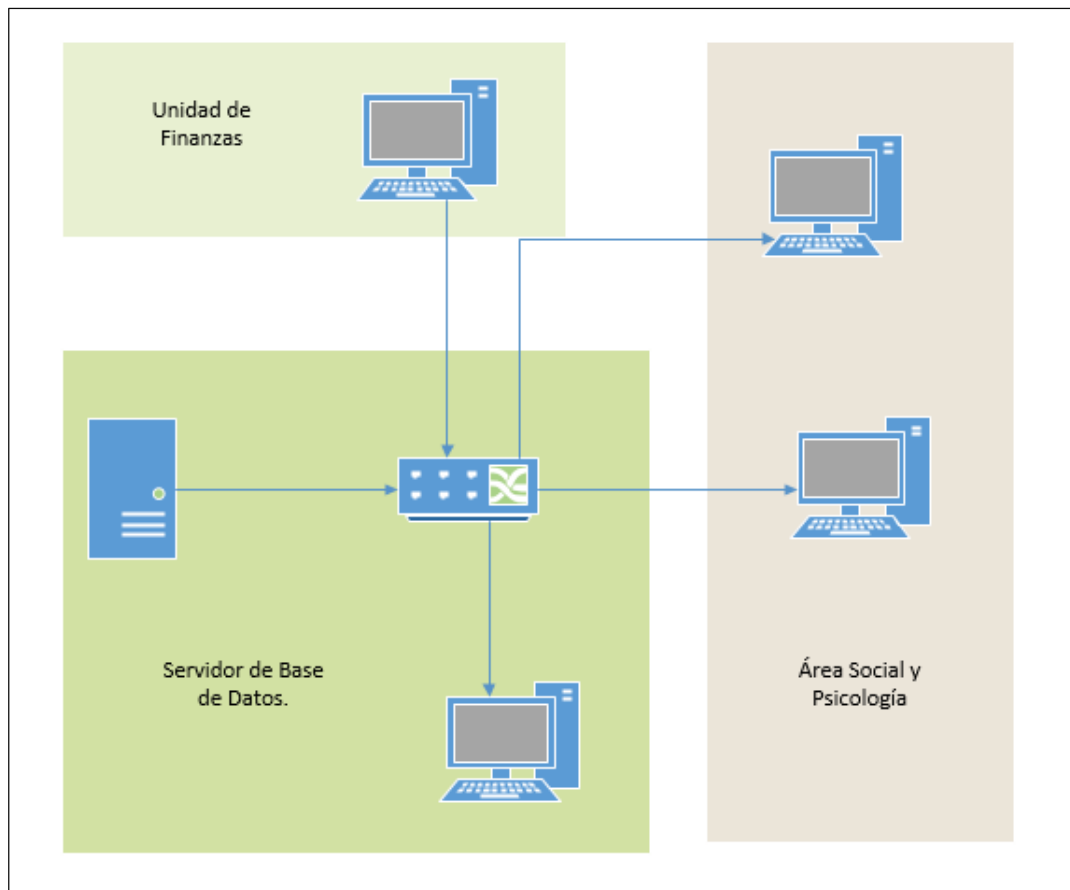
Figura No 3.18 Diagrama de Despliegue



Fuente: [Elaboración Propia]

El diagrama de despliegue anterior nos muestra como está conectado los equipos para el sistema de registro y control para el pago del bono a personas con discapacidad en si el sistema trabaja con un ordenador como servidor del cual se distribuye a las demás.

Figura No 3.19 Topología de red Estrella



Fuente: [Elaboración Propia]

### 3.4. POST-GAME

Durante esta última etapa se realizaron las actividades de prueba del sistema de registro y control para el pago del bono a personas con discapacidad en municipio de Tiahuanacu, se propusieron políticas de seguridad con el objetivo de que este sistema sea de total confianza y así esta pueda coadyuvar en labores cotidianas, se obtuvieron las métricas de calidad y se realizó el diseño el manual de usuario gestor y usuario funcionario para la unidad Slim-DNA del municipio de Tiahuanacu cumpliendo con mucho éxito.



### 3.4.1. Identificación de roles

Conocer tanto a los usuarios como sus objetos es para dar el rumbo que debe seguir el proyecto, con esta información se lograra conocer a quien está destinado el sistema, da a conocer el grado de satisfacción del usuario lo que es el objetivo primario a la hora de desarrollar el software con metodologías ágiles.

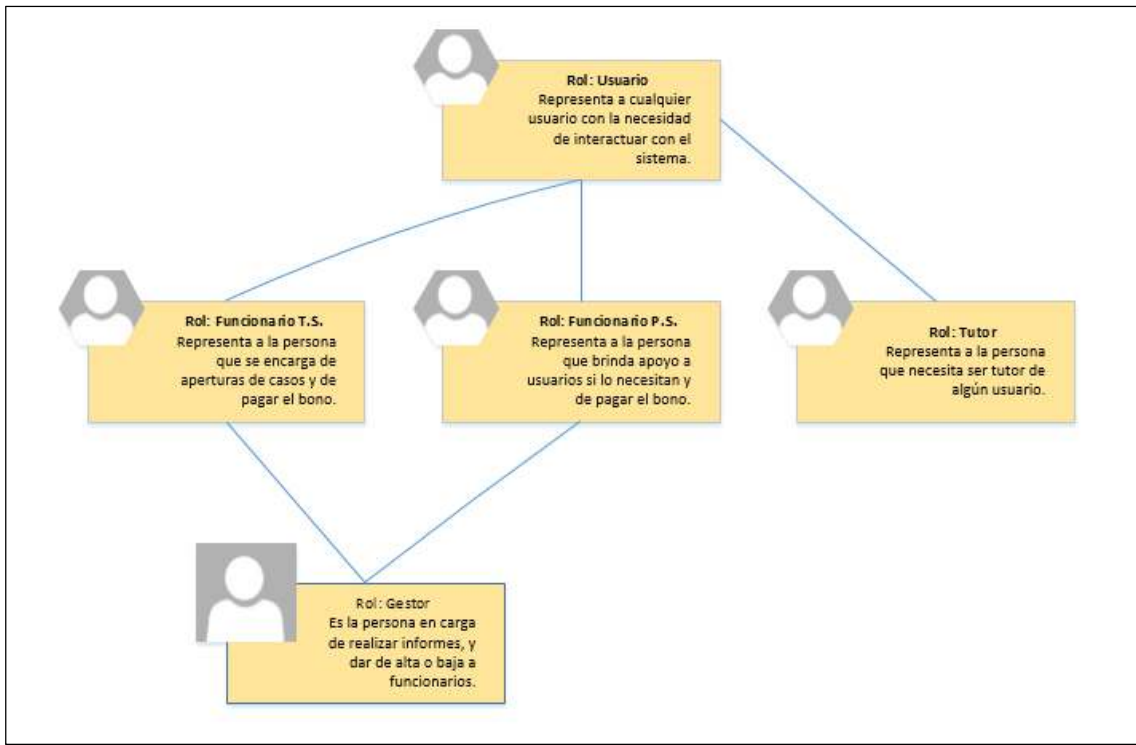
Se identifica los tipos de actores de la aplicación, esto implica identificar a las clases u subclases de actores.

Tabla No 3.17 Descripción de roles del sistema

| Roles   | Descripción   |
|---|---|
| <b>Gestor</b>   | Es la persona encargada de realizar las tareas administrativas responsable del Slim-DNA, dar de alta y baja a los funcionarios nuevos, encargado de realizar reportes de forma general. |
| <b>Funcionario autorizado encargado de trabajo social</b> | Es la persona encargada del registro de los nuevos beneficiarios, de realizar los pagos y de realizar los recibos de pago y reportes específicos o personalizados.                      |
| <b>Funcionario autorizado encargado Psicóloga</b>         | Es la persona encargada del registro de los nuevos beneficiarios, de realizar los pagos y de realizar los recibos de pago y reportes específicos o personalizados.                      |

Fuente: [Elaboración propia]

Figura No 3.20 Clasificación de roles



Fuente: [Elaboración propia]

### 3.5. METRICAS DE CALIDAD

El objetivo de un producto de software es que posea la calidad necesaria y suficiente para que satisfaga las necesidades del usuario tanto explícitas e implícitas. El propósito de la ingeniería de software es construir un software de calidad evaluadas por la iso 9126. Por consiguiente en este capítulo se detalla cuantitativamente la calidad del Sistema (iso 9126).

La determinación de calidad del producto de Software se toma en cuenta las métricas de calidad. Dentro de la tabla tenemos:

factor, criterio, usuario, puntuación, factibilidad de uso, integridad, corrección, fiabilidad, eficiencia, factibilidad de almacenamiento, factibilidad de prueba, flexibilidad, reusabilidad, interoperabilidad, portabilidad. También comprende criterio y detalle respecto al sistema, como ser exactitud, tolerancia de error, eficiencia de ejecución, capacidad de expansión, independencia del Hardware, instrumentación, modularidad, operatividad, seguridad, auto documentación, trazabilidad, e información.

Tabla No 3.18 Métricas de calidad 1.a

| N° | Factor              | criterio                      | Usuario |                        |                          | Puntuación |
|----|---------------------|-------------------------------|---------|------------------------|--------------------------|------------|
|    |                     |                               | Gestor  | Funcionario<br>trabajo | Funcionario<br>Psicóloga |            |
| 1  | Factibilidad de uso | Factibilidad de operación     | 85      | 85                     | 80                       | 83         |
|    |                     | Factibilidad de comunicación  | 90      | 85                     | 85                       | 87         |
|    |                     | Factibilidad de aprendizaje   | 85      | 90                     | 90                       | 88         |
|    |                     | Factibilidad de reutilización | 60      | 60                     | 60                       | 60         |
|    |                     | Factibilidad en uso temporal  | 90      | 60                     | 80                       | 76         |
|    |                     | Factibilidad de respaldo      | 80      | 90                     | 90                       | 87         |
|    |                     | Factibilidad de costo         | 90      | 85                     | 90                       | 88         |
|    |                     | TOTAL                         |         |                        |                          | 81         |
| 2  | Integridad          | Control de acceso             | 80      | 85                     | 85                       | 83         |
|    |                     | Facilidad de auditoria        | 80      | 85                     | 85                       | 83         |
|    |                     | TOTAL                         |         |                        |                          | 83         |
| 3  | Corrección          | Complejidad                   | 90      | 90                     | 85                       | 87         |
|    |                     | Consistencia                  | 85      | 90                     | 85                       | 87         |
|    |                     | Trazabilidad                  | 80      | 85                     | 80                       | 83         |
|    |                     | TOTAL                         |         |                        |                          | 86         |

Fuente: [Elaboración propia]

Tabla No 3.19 Métricas de calidad 1.b

|   |                             | <b>Precisión</b>                       | <b>80</b> | <b>90</b> | <b>90</b> | <b>87</b> |
|---|-----------------------------|--|-----------|-----------|-----------|-----------|
| 4 | Fiabilidad                  | Consistencia                           | 90        | 85        | 85        | 83        |
|   |                             | Tolerancia a fallos                    | 80        | 85        | 80        | 82        |
|   |                             | Modularidad                            | 85        | 90        | 75        | 88        |
|   |                             | Simplicidad                            | 95        | 80        | 80        | 87        |
|   |                             | TOTAL                                  |           |           |           | 85        |
|   |                             |  |           |           |           |           |
| 5 | Eficiencia                  | Eficiencia de ejecución                | 95        | 80        | 80        | 87        |
|   |                             | Eficiencia de almacenamiento           | 90        | 85        | 85        | 88        |
|   |                             | TOTAL                                  |           |           |           | 88        |
| 6 | Facilidad de almacenamiento | Modularidad                            | 90        | 80        | 85        | 85        |
|   |                             | Simplicidad                            | 80        | 85        | 80        | 83        |
|   |                             | Auto descripción                       | 75        | 75        | 90        | 80        |
|   |                             | Consistencia                           | 80        | 75        | 75        | 78        |
|   |                             | Concisión                              | 80        | 85        | 75        | 80        |
|   |                             | TOTAL                                  |           |           |           | 81        |
| 7 | Facilidad de prueba         | Modularidad                            | 95        | 80        | 80        | 85        |
|   |                             | Simplicidad                            | 80        | 85        | 85        | 87        |
|   |                             | Auto descripción                       | 85        | 85        | 90        | 87        |
|   |                             | Instrumentación                        | 80        | 75        | 75        | 78        |
|   |                             | TOTAL                                  |           |           |           | 85        |
| 8 | Flexibilidad                | Auto descripción                       | 95        | 80        | 80        | 87        |
|   |                             | Generalidad                            | 80        | 90        | 85        | 85        |
|   |                             | Modularidad                            | 80        | 85        | 85        | 87        |
|   |                             | Independencia entre sistema y software | 80        | 85        | 85        | 83        |
|   |                             | Independencia de hardware              | 95        | 80        | 80        | 88        |
|   |                             | TOTAL                                  |           |           |           | 86        |

Fuente: [Elaboración propia]

Tabla No 3. 20 Métricas de calidad 1.c

|    |                   |  |           |           |           |           |
|----|-------------------|--|-----------|-----------|-----------|-----------|
| 9  | Reusabilidad      | <b>Auto descripción</b>                | <b>88</b> | <b>90</b> | <b>75</b> | <b>84</b> |
|    |                   | Generalidad                            | 90        | 85        | 85        | 87        |
|    |                   | Modularidad                            | 95        | 80        | 80        | 85        |
|    |                   | Independencia entre sistema y software | 90        | 85        | 85        | 88        |
|    |                   | Independencia de hardware              | 90        | 80        | 80        | 83        |
|    |                   | TOTAL                                  |           |           |           | 86        |
| 10 | Interoperabilidad | Modularidad                            | 90        | 95        | 85        | 90        |
|    |                   | Compatibilidad de comunicación         | 90        | 90        | 85        | 88        |
|    |                   | Compatibilidad de datos                | 85        | 85        | 85        | 85        |
|    |                   | TOTAL                                  |           |           |           | 88        |
| 11 | Portabilidad      | Auto descripción                       | 85        | 90        | 85        | 88        |
|    |                   | Modularidad                            | 95        | 85        | 85        | 88        |
|    |                   | Independencia entre sistema y software | 90        | 80        | 90        | 87        |
|    |                   | Independencia de hardware              | 90        | 85        | 85        | 87        |
|    |                   | TOTAL                                  |           |           |           | 88        |

Fuente: [Elaboración propia]

Usuario 1.- El puntaje asignado por el administrador del sistema. Dra. Olga Poma Tito quien actualmente es directora de la unidad del SLIM-DNA en el municipio de Tiahuanacu y quien tiene cargo a las funcionarias de la unidad.

Usuario 2.- El puntaje asignado fue realizado por la psicóloga Lic. Sara Maldonado del SLIM-DNA de Tiahuanacu quien es una de las encargadas de realizar el pago a las personas con discapacidad.

Usuario 3.- El puntaje asignado fue realizado por la trabajadora social Lic. Ximena Marlene Escobar Mamani del SLIM-DNA de Tiahuanacu quien es una de las encargadas de realizar el pago a las personas con discapacidad.

Una vez habiendo obtenido de los tres usuarios es de sobre las métricas de calidad de. Entonces se concluye que la calidad del sistema de registro y control para el pago del bono a personas con discapacidad, se encuentra entre los niveles aceptados de las métricas de calidad de software.

### **3.6.1. Operación del sistema – facilidad**

- Facilidad de operación: según la encuesta realizada, del presente proyecto de software cumple con este criterio y además se puede apreciar con facilidad en la interfaz gráfica del sistema.
- Facilidad de comunicación: el cuestionario realizado con respecto a este criterio refleja que el producto de software cumple con este criterio en su totalidad.
- Facilidad de aprendizaje: este criterio es dependiente de la facilidad de operación, por lo tanto se cumple con este criterio ya que el producto de software gracias a su facilidad de operación brinda gran facilidad al proceso de transmisión o cambio de sistemas.
- Capacitación y evaluación: Por la sencillez en la operación del sistema el tiempo de capacitación ha sido de las dos horas durante una semana está orientada a operadores habilitados del sistema el mismo que deberá ser evaluado para garantizar un correcto aprendizaje.

La capacitación consiste en el manejo adecuado de las pantallas y formas de misión de reportes a impresora o medios magnéticos. Los operadores deben recibir instrucciones sobre errores comunes que pueden ocurrir como detectarlos, tomar las medidas necesarias cuando se presentan. Recibirán instructivos para la codificación del ingreso de datos al sistema.

- **Mantenimiento:** La fase de mantenimiento es una actividad permanente orientada básicamente a conservar la vigilancia del sistema así como satisfacer los requerimientos emanados según el funcionamiento del mismo.
- **Documentación:** Para lograr una transferencia efectiva del sistema se sustenta la posibilidad de entrega del manual de usuario, así como los programas fuente bajo la recomendación de que toda modificación realizada al sistema debe ir antecedida por la actualización de los documentos correspondientes.
- **Resultados de las pruebas con la aplicación de la prueba de caja negra,** se ha logrado verificar la operatividad de los módulos del sistema. Con la prueba de la caja blanca, se verifico el conocimiento interno (código fuente), con la probabilidad de ejecución. Según la métrica de calidad, nos proporciona un promedio de 91% que se ha realizado a tres usuarios del sistema.

### **3.7. SEGURIDAD**

Podemos entender como seguridad en términos generales e informáticos a un estado que nos indica que el sistema está libre de peligro, daño o riesgo. Se entiende como daño o riesgo todo aquello que pueda afectar su funcionamiento. Para una gran mayoría de los expertos el concepto de seguridad en informática porque no existe un sistema 87% seguro.

Para que un sistema se pueda definir como seguro debe cumplir 4 características:

- **Integridad:** La información solo puede ser modificada por quien está autorizado y de manera controlada.
- **Confidencialidad:** La información solo debe ser visible y legible para los autorizados.
- **Disponibilidad:** debe estar disponible en horario de atención dentro la municipalidad.
- **No refutado:** El uso, modificación o actualización de la información por parte de un usuario debe ser irrefutable, es decir, que el usuario no puede negar determinada acción.

### **3.7.1. Integridad**

La creación de usuarios con sus respectivos roles identificados y sus permisos agregados ayudan a clasificar a los mismos y designar y realizar tareas. La buena administración de los roles de usuario debe estar reflejada en el sistema de acuerdo a las funciones que realizan los funcionarios que serán usuarios del sistema dentro de la unidad del SLIM-DNA.

### **3.7.2. Confidencialidad**

Según lo mencionado anteriormente sobre la definición de roles de usuarios, los cuales se encuentran asociados a un código de servidor público, toda acción realizada por dicho usuario será registrado con el código de funcionario de la unidad de protección que es único y a la vez tendrá acceso a la información que el mismo usuario registro y no así la información que otros usuarios introdujeron al sistema; con esta acción se evitara la pérdida o cambio fraudulenta de la información.



También cabe mencionar que el campo de contraseña de los funcionarios se encuentra encriptado, esto para brindar más seguridad en el sistema y en la tabla de autorización para administradores, habilitados en la unidad.

### **3.7.3. Disponibilidad**

La disponibilidad del sistema como también el acceso a los datos debe realizarse en el momento preciso en que el usuario lo solicite, dentro de la municipalidad. El sistema está desarrollado bajo tecnología web. Conjuntamente se contara un equipo pc al cual se hizo las configuraciones y adaptaciones pertinentes para la implementación del producto, el sistema esta implementado en una red interna pero con la intención de que en un futuro esté disponible en la web como sistema modular de la plataforma EMM.

### **3.7.4. Irrefutabilidad**

Para realizar esta tarea se tiene una tabla llamada bitácora la cual tiene la función de registrar todos los movimientos (altas, Bajas, Modificación y consulta) realizados en el sistema que a su vez almacena la fecha en la que se realizó la operación, quien la realizo, que funcionario fue afectado con el cambio, la tabla que fue afectada.

## **3.8. COSTOS**

El desarrollo del presente proyecto cuenta con una inversión para el desarrollo e implementación del sistema propuesto para la Unidad del Slim-DNA de Tiahuanacu, por lo tanto en relación al costo/beneficio que se obtenga del sistema no se debe exceder en corto, mediano o largo plazo.

Esto implica evaluar cuál es la arquitectura que mejor se adapta para el procesamiento de las aplicaciones que se piensa desarrollar para el futuro servidor.

Según la filosofía para su construcción, el procesamiento puede ser: Centralizado, descentralizado, distribuido o una conjunción de todos ellos y de acuerdo a la envergadura de la institución. Se establece un diseño o arquitectura de construcción de cada dispositivo del nuevo sistema, y en función de esas pautas que determinan el tamaño, capacidad, velocidad y otros para cada componente del sistema.

Por lo expuesto, resulta imprescindible el conocimiento del problema de la institución, antes iniciar todos los aspectos de costos, para que permita definir la nueva filosofía, arquitectura y tipo de procesamiento a emplear para el nuevo sistema.

### **3.8.1. Modelo cocomo**

Para el análisis de costos se utilizó el modelo constructivo de costos en base al modelo jerárquico, se necesita hallar la variable KLDC por lo cual en este proyecto se implementó 9881 líneas de código en el lenguaje PHP.

$$KLDC = \frac{LDC}{1000} \cong \frac{9881}{1000} \cong 9,881 \text{ KLDC}$$

Para el presente análisis se utilizara el modelo semiorgánico y orgánico propio del modelo cocomo como valores constantes en la utilización para los cálculos del sistema para lo cual se cuenta con los siguientes datos.

Tabla No 3.21 Constantes del modelo Semiorgánico

| Modo         | a    | b    | C    | d    |
|--------------|------|------|------|------|
| Orgánico     | 3.20 | 1.05 | 2.50 | 0.38 |
| Semiorgánico | 3.00 | 1.12 | 2.50 | 0.35 |
| Empotrado    | 2.80 | 1.20 | 2.50 | 0.32 |

Fuente: [Elaboración Propia]

### 3.8.1. Estudio de factibilidad

Es necesario conocer la problemática del proyecto a realizar, las particularidades de su implementación y los motivos que la impulsan a estudiar la factibilidad de un proyecto de inversión como sistema de información cliente – servidor con la particularidad de una base de datos con la tecnología MySQL.

Todos estos aspectos y las estimaciones sobre la evolución del nuevo sistema se contemplan en la estrategia global de la información que ha de brindar en el nuevo sistema y es esta la que define las políticas que condicionan dicho proyecto a realizarse.

### 3.8.2. Factibilidad técnica

Con el mismo criterio con que se desarrolla una estrategia de factibilidad y se confecciona un plan de implementación, será necesario establecer una estrategia de sistemas factibles que contemple tanto las necesidades actuales respecto al nuevo sistema como servicio, como de mediano y largo plazo en materia de implementación prueba y resultados. A continuación, se detalla algunos de los aspectos a considerar, si bien se ha mencionado, carecen de universalidad y sólo sirven como referencia para implementar.

- Establecer requerimientos globales.
- Definir arquitectura.
- Grado de sofisticación técnica.
- Establecer alcance del sistema.
- Fijar pautas para el diseño de la red de comunicaciones del sistema

### 3.8.3. Requerimientos de hardware

Tabla No 3.22 Requerimientos mínimos de hardware

| DESCRIPCIÓN   | CANTIDAD |
|---|----------|
| Procesador Pentium Core I3 intel de 2.4 Ghz. Original.<br>Disco duro de 500 Gb. SATA<br>Memoria RAM: de 2 Gb. DDR2 expandible.<br>Sistema operativo: Windows ocho o versiones actuales.<br>Switch TP-Link 120b.<br>Monitor, teclado, mouse y parlantes. | 1        |

Fuente: [Elaboración propia]

Tabla No 3.23 Requerimientos recomendados de hardware

| DESCRIPCIÓN   | CANTIDAD |
|---|----------|
| Procesador Pentium Core I7 6ta. Generación 3.5 Ghz.<br>Disco duro de 500 Gb. SATA<br>Memoria RAM: de 8 Gb. DDR3 o superior expandible.<br>Lector de DVD-RW 52x<br>Sistema operativo: Windows Diez<br>Switch TP-Link 120b.<br>Monitor, teclado, mouse y parlantes. | 1        |

Fuente: [Elaboración propia]

Los equipos informáticos necesarios para la implementación del presente proyecto, es una decisión crítica ya que sin esto, el Software no es factible y no podríamos realizar el desarrollo del proyecto.

Realizando un análisis de recolección de datos referenciales en la institución y el conocimiento empírico adquirido en compra de equipos de hardware podemos llegar a la siguiente conclusión sobre el costo del equipo.

Costo promedio de equipo: CPE = 600 \$us

#### 3.8.4 Requerimientos de software

En el siguiente análisis del costo del software se tomó en cuenta como costo referencial a las licencias del software utilizados para el desarrollo del producto.

Tabla No 3.24 Costos del Software

| Descripción                  | Cantidad | Costo Unitario | Costo \$us |
|------------------------------|----------|----------------|------------|
| Microsoft Windows 8 estándar | 4        | 119            | 476        |
| Base de datos: MySQL6.0      | 1        | 0              | 0          |
| Total \$us                   |          |                | 476        |

Fuente: [elaboración Propia]

#### 3.8.5. Factibilidad operativa

Para la factibilidad operativa se ha tomado en cuenta los siguientes aspectos, el alcance de objetivos y metas, también en el análisis costos de operación y destino de recursos finales en la unidad que satisfagan la utilización o implementación del sistema, garantizar la disponibilidad y usabilidad o utilización del sistema.

Con relación al costo del desarrollo e implementación en la unidad del Slim-DNA dentro la municipalidad de la población de Tiahuanacu.

Tabla No 3.25 Costo Operativo por Año

| DETALLE              | CANTIDAD | PRECIO Bs | COSTO Bs |
|----------------------|----------|-----------|----------|
| Papel bond tam carta | 3        | 20        | 60       |
| Toner                | 1        | 153       | 153      |
| Servicio electrico   | Kw/h     | 250       | 250      |
| Total en Bs          |          |           | 463      |

Fuente: [Elaboración Propia]

### 3.8.6. Costo Laboral en el ciclo de ejecución

El costo laboral determinado por el salario de un programador establecido en las leyes bolivianas, como también por la cantidad de programadores.

Tabla No 3.26 Tarifa del programador para el sistema

| Institución Pública y privada | Salario mensual \$us |
|-------------------------------|----------------------|
| publica                       | 500                  |
| privada                       | 600                  |

Fuente: [Elaboración Propia]

El costo laboral se determina promediando los valores referenciales 550 \$us ese es el sueldo promedio con el cual se trabajó.

Tabla No 3.27 Tarifa del programador para el sistema

| DETALLE       | SUELDO | CANTIDAD | DURACIÓN 4 MESES |
|---------------|--------|----------|------------------|
| Programadores | 550    | 2        | 1100*4           |
| TOTAL \$us    |        |          | 4400             |

Fuente: [Elaboración Propia]

Costo Fijo = Sueldo programadores + Hardware requerido \* Depreciación + Software requerido.

$$\text{Costo Fijo} = 4400 + \left(476 * 0.25 * \frac{4}{12}\right) + 463 = 1183,39 \text{ BS}$$

Para efectos de análisis diseño e implantación del proyecto y el cálculo de se utilizara la siguiente notación.

$$Cs = Pr * D * Sp$$

Dónde:

Cs: Costo del desarrollo del software

Pr: Personas requeridas

D: Tiempo requerido

Sp: Sueldo del personal

$$Cs = 2 * 3.5 * 500 = 3500 \text{ Bs}$$

De donde tenemos el valor de para el factor de ajuste, que es el resultado de la multiplicación de los valores evaluados con anterioridad.

$$FAE = 0.76$$

$$E = KLDC * FAE \left( \frac{Personas}{Mes} \right)$$

$$E = 3.2 * 13.188C^{1.05} * 0.76 \left( \frac{Personas}{Mes} \right)$$

Calculo del tiempo para el desarrollo del producto

$$T = c * Esfuerzo^d (Meses)$$

$$T = 8.5 (Meses)$$

Calculo de la productividad del producto

$$PR = LDC / Esfuerzo \left( \frac{LDC}{Personas} Mes \right)$$

$$PR = LC \left( \frac{LCD}{Personas} Mes \right)$$

Calculo de personal promedio

$$p = \frac{Esfuerzo}{tiempo(personas)}$$

$$p = 3,4 \frac{49}{9,5} (personas)$$

$$p = 4.46 \cong 3 (Personas)$$



Costo por mes

$$c = 4 * 315$$

$$c = (1260 * 8) + 14000 Bs$$

Costo total

$$Costo Total = 24080 Bs$$

El costo final que nos resulta es el que se puede observar en la ecuación anterior, es decir es costo del proyecto.

## **CAPITULO IV**

### **4. CONCLUSIONES Y RECOMENDACIONES**

En este capítulo se dará a conocer algunas observaciones que se establecerán en las conclusiones y recomendaciones a la finalización del sistema.

#### **4.1. CONCLUSIONES**

Una vez finalizado el desarrollo del Sistema de registro y control para el pago del bono a personas con discapacidad para el municipio de Tiahuanacu, se concluye que el sistema coadyuva con la administración de manera eficiente optimizando tareas como el registro, datos centralizados y un mejor control en la atención a las personas con discapacidad del municipio de Tiahuanacu.

Por lo que se concluyó los siguientes módulos y tareas de manera exitosa.

- Se logró desarrollar la base de datos única que permite almacenar la información de las personas con discapacidad y sus tutores, que necesariamente deben ser almacenadas en la base de datos del sistema,
- Se logró desarrollar un sistema seguro al momento de la manipulación de parte de los funcionarios, autenticación de usuario y password encriptado, a la vez que el sistema maneja sesiones automáticamente.
- El módulo de beneficiarios para solicitud de pago del bono, así como de sus tutores si este amerita.

- El módulo de registro de pago donde se lista los pagos realizados y realizar la impresión de recibo para realizar el respectivo informe posteriormente.
- El módulo de depuración temporal y definitiva, esta primera sucede cuando ya no se cumple con los requisitos.
- El módulo del usuario gestor asignada al MAE de la unidad Slim-DNA con la capacidad de controlar las actividades hechas en el sistema.
- Se logró con éxito la función de copias de seguridad del usuario gestor.

Aplicando con éxito normas de calidad y herramientas de programación para que el sistema goce de funcionalidad, usabilidad y fiabilidad dentro de las buenas practicas a las que estas normas se enmarcan.

#### **4.2. RECOMENDACIONES**

En base a las normas de seguridad propuestas y las conclusiones realizadas durante las pruebas se elaboran las siguientes recomendaciones.

- Se recomienda realizar copias de seguridad de la base de datos periódicamente o cada vez que realice la labor de pago del bono a las personas con discapacidad.
- Se recomienda al usuario gestor, al dar de alta a un funcionario este debe tomar en cuenta la cantidad mínima de caracteres y lógica de asignación de password, así como de discreción después de su asignación.
- Se recomienda a la municipalidad la adquisición de un servidor para el alojamiento del sistema o posibles sistemas implementados en el futuro.

## BIBLIOGRAFIA

- Dolado, J (2014). El Modelo COCOMO. Recuperado el 14 de Septiembre, Campus de Gipuzua [http://www.inesad.edu.bo/pdf/wp07\\_2010.pdf](http://www.inesad.edu.bo/pdf/wp07_2010.pdf)
- Kendall, K., & Kendall, J. (2011). Análisis y Diseño de Sistemas (Edición octava). Mexico:Pearson Educación.
- Sommerviller, I. (2005). Ingeniería del Software (Edición Séptima). Madrid – España: Pearson Educación S.A.
- tecnologimerlin. (2018 Diciembre 10). Tecnología es desarrollo. Recuperado el 10 de diciembre de 2018, de [http:// tecnologimerlin.blogspot.com/](http://tecnologimerlin.blogspot.com/).
- Cota, A. (2004). Ingeniería de Software soluciones Avanzadas (Edición Segunda). Wesley – Estados Unidos.
- Yourdon, E. (2003). Análisis Estructurado Moderno (Edición Quinta) Mexico Prentice-hall Hispanoamerica. S.A.
- Leon, G. (1996). Ingeniería de Sistemas de Software (Edición Cuarta) Madrid isdefe S.A
- Yourdon, E. (2003). Análisis Estructurado Moderno (Edición Quinta) Mexico Prentice-hall Hispanoamerica. S.A.

Braude, E (2003) Ingenieria de Software Una Perspectiva Orientada a Objetos (Edicion Cuarta) Mexico RA-MA S.A. Editorial y Publicaciones.

Pantaleo, G. (2008). Calidad en el Desarrollo de Software (Edicion Octava) Colombia Marcombo Editorial.

Gomez, S. &, Moraleda, P, (2010). Calidad de Datos (Edicion Sexta) Ediciones de la U.

Cabello, I. &, Rejas, P, (2019). Aproximacion a la ingenieria del Software (Edicion Primera) Ramon areces. S.A.

Icont, I (2017). Compendio Seguridad de la Información (Edicion Segunda) Icontec Internacional

Palmer, M (2003). Tratamiento Informatico de Datos (Edicion Primera) Ediciones Paraninfo.

Sanches, O & Herrera, D. (2014). Redes Informaticas. Guia Práctica (Edicion Primera) Ediciones Paraninfo.

Lazaro, A & del Rio, J. (20006). Lab view 7.1 Programación Grafica para el Control de datos (Edicion Primera) Ediciones Paraninfo.

Lasa, C & Alvarez, A. & Hrez, R. (2006). Metodos Agiles: Scrum Kanban Lean(Edicion Segunda) Ediciones Paraninfo.

Lasa, C & Alvarez, A. & Hrez, R. (2006). Metodos Agiles: Scrum Kanban Lean (Edicion Segunda) Ediciones Paraninfo.

Aenor, R (2009) Metricas del sistema de Gestion de Calidad (Edición Primera) España Ediciones VV.AA.

Norman, E. (2018). Guia Práctica para la Integración de Sistemas de gestion Iso (Edicion Primera) España Asociación Española de normalización y certificación.

Minguez, D. & Garcia, E. (2000). Metodologia para el Desarrollo Web: UWE (Edicion Primera) Antrix

Prentice, L. (1999). UML y Patrones Introduccion al Analisis y diseño Orientado a Objetos (Edicion Primera) S.E.

Valderrey, P. (2011). Administración de Sistemas y Gestion de Base de Datos (Edicion Primera) Madrid España starbook Editorial.

Espinoza Fuentes, F (2010). Sistemas de Información Para La Gestion de la Empresa. *Serie de Documentos de Trabajo Sobre Desarrollo*,  
[http://www.inesad.edu.bo/pdf/wp07\\_2010.pdf](http://www.inesad.edu.bo/pdf/wp07_2010.pdf)

blanchard, B. (2011). Ingenieria de Sistemas Hispanoamerica. S.A.

Piattini, M. & Vizcaino, A. (2003). Desarrollo Global de Software.

Schwaber, K. & Sutherland, J. (2013) La Guia Definitiva de Scrum: Las reglas del Juego.

Perona, A. (200) Estudios de factibilidad Técnico-Económica de proyectos de ingeniería

Palacio, J. (2014 Mayo 1) Compendio de Ingeniería del Software.

Palacio, J. (2007). Flexibilidad con Scrum. (S:E.).

Moreno, A. (2011). Estimación de Proyectos Software (Ed.). Mexico (p. 84-122).

## DISPONIBLES

- (www1)  
<https://www.aprenderaprogramar.com/index.php-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70&Itemid=193>
- (www2)  
<http://sosinformatico.blogspot.com/2011/02/que-es-codeigniter-y-porque-usarlo-como.html>
- (www3)  
<https://www.incluyeme.com/los-tipos-discapacidad/amp/>; 2019
- (www4)  
<http://www.un.org/es/events/disabilitiesday/>; 2019
- (www5)  
[https://www.ilo.org/dyn/natlex/docs/ELECTRONIC/105530/129060/F1917615840/Ley\\_N\\_977%20BOLIVIA.pdf](https://www.ilo.org/dyn/natlex/docs/ELECTRONIC/105530/129060/F1917615840/Ley_N_977%20BOLIVIA.pdf); 2019
- (www6)  
<https://jorgeportella.files.wordpress.com/2011/11/analisis-diseo-y-desarrollodeaplicacionesweb.pdf>
- (www7)  
<https://jorgeportella.files.wordpress.com/2011/11/analisis-diseo-y-desarrollo-de-aplicacionesweb.pdf>
- (www8)  
<https://jorgeportella.files.wordpress.com/2011/11/analisis-diseo-y-desarrollodeaplicacionesweb.pdf>
- (www9)  
<https://jorgeportella.files.wordpress.com/2011/11/analisis-diseo-y-desarrollodeaplicacionesweb.pdf>
- (www10)



- <https://programacionwebisc.wordpress.com/2-5-metodologias-para-el-desarrollo-de-aplicaciones-web/>
- (www11)  
<https://ingenieriasoftware2011.files.wordpress.com/2011/07/uml-gota-a-gota.pdf>
  - (www12)  
<http://ingenieriadesistemas-shirley.blogspot.com/2012/05/tipos-de-diagramas-uml.html>, 2019
  - (www13)  
<http://ingenieriadesistemas-shirley.blogspot.com/2012/05/tipos-de-diagramas-uml.html>, 2019
  - (www14)  
<http://ingenieriadesistemas-shirley.blogspot.com/2012/06/tipos-de-diagramas-uml.html>, 2019
  - (www15)  
<http://ingenieriadesistemas-shirley.blogspot.com/2012/07/tipos-de-diagramas-uml.html>, 2019
  - (www16)  
<https://www.google.com/diagrama+de+secuencia+en+rational+rose>, 2019
  - (www17)  
<https://www.google.com/diagrama+de+estados+uml+ejemplos&sxsrf=ACYBG>, 2019
  - (www18)  
<http://ingenieriadesistemas-shirley.blogspot.com/2012/05/tipos-de-diagramas-uml.html>, 2019
  - (www19)  
<http://ingenieriadesistemas-shirley.blogspot.com/2012/05/tipos-de-diagramas-uml.html>, 2019
  - (www20)

- [https://www.aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=492:ique-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70&Itemid=193](https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:ique-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70&Itemid=193)
- (www21)  
<https://searchdatacenter.techtarget.com/es/definicion/MySQL>
  - (www22)  
<http://sosinformatico.blogspot.com/2011/02/que-es-codeigniter-y-porque-usarlo-como.html>
  - (www23)  
<http://www.adwe.es/codigo/codeigniter-framework-php-desarrollo-aplicaciones-web>, 2019
  - (www24)  
<https://www.google.com/topologia+de+redes&sxsrfSohxv25ug1bXDM>, 2019
  - (www25)  
[https://www.ecured.cu/Teor%C3%ADa\\_de\\_redes](https://www.ecured.cu/Teor%C3%ADa_de_redes), 2019
  - (www26)  
<https://definicion.de/ingenieria-de-software/>; 2019
  - (www27)  
<https://www.yumpu.com/es/document/view/14635371/estimacion-de-proyectos-software-ana-m-moreno-s-capuchino>
  - (www28)