

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



**Universidad de las Ciencias
Informáticas**

Facultad 1

**Portal Web de la Dirección Municipal del Trabajo y Seguridad
Social del municipio Julio A. Mella**

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

Autor:

Ernesto Pupo Palacios

Tutor:

MSc. Aylin Estrada Velazco

La Habana, junio de 2021

“Año 63 de la Revolución”

DECLARACIÓN DE AUTORÍA

Yo **Ernesto Pupo Palacios**, con carné de identidad **97082618664**, declaro por este medio que soy el autor principal del trabajo titulado “Portal de La Dirección Municipal del Trabajo y Seguridad Social del municipio Julio A. Mella” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firmamos la presente a los _____ días del mes de _____

Del año _____

Ernesto Pupo Palacios
Autor

Msc. Aylin Estrada Velazco
Tutor

Índice

Introducción	6
Capítulo 1. Referentes Teórico-Methodológicos sobre la gestión de los procesos de la DMTSS 10	
1.1 Generalidades sobre la dirección del trabajo y la seguridad social	10
1.2 Análisis de organismos/instituciones que gestionan y controlan el trabajo y la seguridad social	11
1.2.1 Dirección General de la Ordenación de la Seguridad Social de España	11
1.2.2 Ministerio de Trabajo y Seguridad Social de Cuba	12
1.2.3 Instituto Nacional de Seguridad Social de Cuba	13
1.3 Metodología de desarrollo de software	14
1.4 Descripción de las herramientas y tecnologías para el desarrollo del portal web	21
1.4.1 Frameworks	21
1.4.2 Lenguajes de programación	22
1.4.3 Gestor de base de datos	25
1.4.4 Herramientas	25
1.5 Conclusiones Parciales	26
Capítulo 2. Diseño de la Propuesta de Solución	28
2.1 Caracterización del entorno de desarrollo	28
2.1.1 DMTSS del municipio Julio A. Mella	28
2.1.2 Modelo Conceptual	31
2.2 Características del sistema	32
2.2.1 Usuarios del sistema	32
2.2.2 Especificación de requisitos	32
2.2.3 Historias de Usuarios	35
2.2.4 Plan de Iteraciones	36
2.2.5 Plan de Entrega	37
2.3 Descripción Propuesta de solución	37
2.3.1 Propuesta de solución para la DMTSS del municipio Julio A. Mella	37
2.3.2 Descripción de la arquitectura	38
2.3.3 Patrones de Diseño	39
2.3.4 Diagrama de Clase	39
2.3.5 Modelo Entidad-Relación	41
2.3.6 Modelo de despliegue	42
2.4 Conclusiones Parciales	43

Capítulo 3 Implementación y Pruebas del Sistema	44
3.1 Justificación del estándar de codificación.	44
3.2 Implementación y codificación de la propuesta de solución	45
3.2.1 Codificación	46
3.3 Pruebas y resultados	48
3.3.1 Desarrollo dirigido por pruebas (TDD)	48
3.3.2 Pruebas de Regresión.....	52
3.3.3 Pruebas de Usabilidad	52
Conclusiones parciales del Capítulo	55
Conclusiones Generales	56
Bibliografía	57
Anexos.....	62
Anexo 1: Historias de Usuario	62
Anexo 2 Representación de la Metodología XP.....	64
Anexo 3 Representación del Modelo Vista Controlador.....	65

Resumen

La Dirección Municipal del Trabajo y Seguridad Social es un eslabón esencial para la correcta planificación, administración y control de los gastos de seguridad social de cualquier municipio. Lograr que esta institución tenga un alto nivel de difusión es de vital importancia en la actualidad. Por lo que tener un portal web que complemente a esta entidad sería un gran avance en ese sector. En la actualidad La Dirección Municipal del Trabajo y Seguridad Social del municipio Julio A. Mella no presenta un sitio web que le permita difundir su labor y funcionamiento. Dicho portal dará la posibilidad que La Dirección Municipal del Trabajo y Seguridad Social tenga una mayor presencia en la red y lograr un mayor nivel de difusión. Para ello se muestran las distintas facetas del desarrollo, como el proceso de levantamiento de requisitos, análisis y diseño de la propuesta de solución a la problemática y por último la realización de pruebas de software que validen la calidad del producto final.

Abstract

The Municipal Directorate of Labor and Social Security is an essential link for the correct planning, administration and control of social security expenses in any municipality. Achieving that this institution has a high level of diffusion is of vital importance today. Therefore, having a web portal that complements this entity would be a great advance in this sector. At present, the Municipal Directorate of Labor and Social Security of the Julio A. Mella municipality does not present a website that allows it to disseminate its work and operation. This portal will allow the Municipal Directorate of Labor and Social Security to have a greater presence on the network and achieve a higher level of dissemination. The different facets of development are shown, such as the process of raising the requirements, analysis and design of the proposed solution to the problem and finally the performance of software tests that validate the quality of the final product.

Palabras claves: planificación, administración, seguridad social, portal web, difusión

Introducción

Antes de 1959 no existía en Cuba un sistema de seguridad social, estaba presente una anarquía institucional en cuanto a la atención de esta importante materia del derecho laboral. Con el triunfo revolucionario del 1ro de Enero de 1959, se inició una etapa de transformaciones encaminadas a solucionar los problemas existentes, transformando la estructura económica, la conciencia social y las condiciones de vida del pueblo. Estas transformaciones se han desarrollado gracias a la voluntad política del gobierno de realizar junto al desarrollo económico un vasto programa de política social, donde la seguridad social es un eslabón fundamental (1).

Al triunfar la Revolución los principales esfuerzos estuvieron concentrados en revertir la precaria protección social que impactaba al país y comenzaron a darse los primeros pasos hacia la transformación organizativa y financiera de los seguros sociales en Cuba, con el Ministerio de Trabajo y Seguridad Social como principal gestor de la aplicación de ese sistema(2).

El Sistema de Seguridad Social cubano ofrece de manera gratuita prestaciones en servicios y en especies a personas que no estén en condiciones de trabajar, no tengan familiares que los puedan ayudar o no tengan recursos para su sustento.

En la actualidad en nuestro país la Dirección Municipal del Trabajo y Seguridad Social (DMTSS) tiene la misión de dirigir, controlar y asesorar La Política Laboral, de Prevención, Asistencia y Trabajo Social en cada municipio. Esta institución vela por el logro, de manera eficiente en el uso de la fuerza de trabajo con el avance del nuevo modelo de gestión económica, proteger a la población que lo requiera, para contribuir al aumento constante de la productividad del trabajo y al fortalecimiento del sistema de justicia laboral.

La DMTSS del municipio Julio A. Mella apuesta por la mejora continua del desarrollo de sus procesos con el fin de garantizar un elevado índice de calidad y la satisfacción de los clientes, además cuenta con ocho departamentos para brindar diferentes servicios a la población como por ejemplo el departamento de empleo no estatal, que tiene como objetivo tramitar y decidir el otorgamiento o no de la autorización para el ejercicio del trabajo por cuenta propia. Sin embargo, es contraproducente entonces que en la entidad no se disponga de un mecanismo centralizado que gestione sus documentos rectores, lo que implica que se duplique información entre los departamentos, máxime cuando el funcionamiento de algunos depende de la información que se genera en otros. Por otra parte, la tenencia de la información en formato duro y dispersa por cada departamento, para la dirección de la entidad resulta engorroso realizar un seguimiento constante del avance del proceso de atención a la población, lo que provoca además altos niveles de consumo de materiales de oficina y dificulta la búsqueda de manera rápida.

En este último año el número de clientes que acceden a la DMTSS del municipio Julio A. Mella para informarse o para realizar algún trámite ha aumentado. A partir de una entrevista realizada a dirección de la entidad se pudo constatar que alrededor de 50 personas acceden diariamente y que en su mayoría requieren asesoría e información sobre las reglamentaciones de los servicios que se prestan, los cuales se atienden presencialmente, que si bien es un método muy eficiente también puede provocar pérdida de tiempo e inconformidades en los clientes por las largas horas de espera. Actualmente, no existe un mecanismo para que el cliente pueda estar informado del avance de los procesos que tiene en curso.

El análisis del funcionamiento del proceso de la DMTSS del municipio Julio A. Mella permitió identificar las siguientes deficiencias:

- La entidad no dispone de un mecanismo centralizado que gestione los procesos que se llevan a cabo.
- La atención a la población de forma presencial genera un gran volumen de información en formato duro.
- La gestión de un proceso entre dos o más departamentos ocasiona pérdida de información.

A partir de lo planteado como situación problemática se formula el siguiente **problema de Investigación**: ¿Cómo contribuir a la gestión de los procesos de la Dirección Municipal del Trabajo y Seguridad Social del municipio Julio A. Mella para favorecer la calidad en la atención a la población?

El diseño de la investigación permite declarar como **objeto de estudio** el proceso de gestión de la DMTSS, enmarcado en el **campo de acción**: informatización de los procesos de la Dirección Municipal del Trabajo y Seguridad Social del municipio Julio A Mella.

Se define como **objetivo general** desarrollar un sistema informático que permita la gestión de los procesos que se llevan a cabo en la Dirección Municipal del Trabajo y Seguridad Social del municipio Julio A. Mella para favorecer la calidad en la atención a la población.

Para dar cumplimiento al objetivo general se plantean los siguientes **objetivos específicos**:

- Construir un marco teórico que sustentan la investigación relacionada con la gestión de los procesos de la DMTSS del municipio Julio A. Mella.
- Diseñar las funcionalidades del sistema para la gestión y divulgación de los procesos de la DMTSS del municipio Julio A. Mella.
- Implementar las funcionalidades para gestionar y controlar los procesos de la DMTSS del municipio Julio A. Mella.
- Validar las funcionalidades para gestionar y controlar los procesos de la DMTSS del municipio Julio A. Mella

Como **hipótesis** se plantea: Al desarrollar una solución informática que contribuya con la gestión y control de los procesos de la DMTSS del municipio Julio A. Mella se favorecerá la calidad en la atención a la población.

Para el desarrollo de la investigación se emplearon los siguientes **métodos de la investigación científica**:

Métodos Empíricos

El método empírico es un modelo de investigación que pretende obtener conocimiento a partir de la observación de la realidad. Por ende, está basado en la experiencia (3).

- **Observación**: para analizar y registrar, cómo se desarrollan los procesos de la DMTSS del municipio Julio A. Mella.

- **Entrevista:** para obtener los criterios de los especialistas que gestionan la DMTSS del municipio Julio A. Mella aplicando una encuesta al personal que interactúa con la entidad.

Métodos Teóricos

Permiten descubrir en el objeto de investigación, las relaciones esenciales y las cualidades fundamentales, no detectables de manera sensorial. Por ello se apoya básicamente en los procesos de abstracción, análisis, síntesis, inducción y deducción (4).

- **Método Hipotético–Deductivo:** el uso de este método permitió arribar a conclusiones particulares a partir de la hipótesis.
- **Métodos de Análisis Histórico y Lógico:** se utilizó para conocer la evolución y desarrollo del objeto o fenómeno de investigación, para ello fue necesario revelar su historia.
- **Analítico-sintético:** se utilizó para el estudio de trabajos similares e investigaciones que abordan el tema del sistema informático.

El documento está compuesto por tres capítulos que responden a los objetivos específicos anteriormente descritos. Cada uno de ellos contiene los elementos necesarios que conforman la solución final, dividiendo el estudio del problema planteado.

En el capítulo 1 se aborda sobre sistemas anteriormente usados para la gestión de los procesos en las direcciones semejantes a la DMTSS del municipio Julio A. Mella, así como los principales conceptos de esta actividad. Se describen además las tecnologías, metodología, herramientas y el lenguaje de programación utilizado en el desarrollo de la solución.

En el capítulo 2 se exponen los elementos que permiten describir la propuesta de solución, tales como: requisitos para la implementación de un sistema para la DMTSS del municipio Julio A. Mella. Se modelan los aspectos del diseño basados en el diagrama de clases de diseño, también se muestra el modelo entidad relación donde se especifican las clases principales con las que interactúa el sistema.

En el capítulo 3 se realiza la implementación de la propuesta de solución, obteniendo un sistema de gestión para administrar y gestionar la DMTSS del municipio Julio A. Mella. Se describen un conjunto de pruebas realizadas al sistema una vez que concluye la implementación, asegurando que este cumple con las especificaciones requeridas.

Capítulo 1. Referentes Teórico-Metodológicos sobre la gestión de los procesos de la DMTSS

En este capítulo se desarrollan las bases teóricas y metodológicas para gestionar los procesos que tienen lugar en la DMTSS del municipio Julio A. Mella. Para ello se analizan los principales conceptos más relevantes relacionados con la temática, así como su relación con las demás entidades en Cuba y en el Mundo.

1.1 Generalidades sobre la dirección del trabajo y la seguridad social

La seguridad social y el trabajo han sido abordadas en nuestro país desde diferentes aristas, siendo los aspectos más trascendentales lo concerniente a: exposición a riesgos laborales, el estudio de los accidentes de trabajo y el ambiente laboral, cuestión corroborada a través del estudio de las tesis defendidas de salud ocupacional, en el periodo comprendido entre 2000-2006, lo que nos permitió comprender que estas investigaciones precedentes realizaron importantes aportes a la seguridad social y salud ocupacional, laboral o del trabajo pero no se propusieron como objetivo directo el análisis de los sistemas de gestión de la dirección de trabajo y seguridad social.

Para poder determinar a que nos referimos con dirección trabajo y seguridad social es importante establecer primero la definición de dirección de trabajo y seguridad social cada uno por separado, con la finalidad de tener un contexto más claro de estos vocablos y he ir de lo general a lo particular.

La **dirección del trabajo** está encaminada al cumplimiento de la legislación laboral, así como fijar el sentido y alcance de las leyes laborales mediante oficios o peticiones a través de dictámenes, para dar a conocer los principios técnicos y sociales de las leyes laborales (5).

La **seguridad social** es la protección que una sociedad proporciona a los individuos y los hogares para asegurar el acceso a la asistencia médica y garantizar la seguridad del ingreso, en particular en caso de vejez, desempleo, enfermedad, invalidez, accidentes del trabajo, maternidad o pérdida del sostén de familia (5).

En Cuba la DMTSS está encaminada a resolver los principales problemas existentes en la población, así como dirigir y controlar la política para el empleo, el trabajo en el sector no estatal, la planificación de la fuerza de trabajo, especialmente la fuerza de trabajo calificada y la asistencia y trabajo social.

La **política para el empleo** en Cuba tiene como objetivo crear empleos productivos sostenibles. El tipo de empleo al que deberá aspirar una política depende de las condiciones nacionales. Por ejemplo, el que el tipo de empleo necesario a corto plazo sea altamente calificado o no sea depender de condiciones nacionales tales como el nivel de desarrollo económico, el nivel de educación y de capacitación y así la proporción entre personas sin calificación y personas calificadas y el tipo de ventajas comparativas con que cuenta o que podría desarrollar el país(6).

La **planificación de la fuerza de trabajo** es el proceso mediante el cual se alinean los elementos operativos y estratégicos de la plantilla de empleados con los objetivos de la organización. La planificación de la fuerza de trabajo operativa considera las necesidades de los empleados, incluida la programación, los conjuntos de habilidades, la capacitación y la satisfacción en el trabajo (7).

1.2 Análisis de organismos/instituciones que gestionan y controlan el trabajo y la seguridad social

Actualmente en el mundo existen un gran número de organizaciones e instituciones encargadas de controlar y gestionar las direcciones laborales y la seguridad social las cuales tienen por finalidad contribuir al desarrollo del país, impulsando políticas públicas que promuevan el trabajo decente, la formación para el trabajo, la seguridad y salud laboral, una mayor integración de grupos vulnerables en el mercado del trabajo así como los cambios normativos necesarios para la ampliación y ejercicio de los derechos de los trabajadores, especialmente los derechos colectivos.

1.2.1 Dirección General de la Ordenación de la Seguridad Social de España

La Dirección General de Ordenación de la Seguridad Social (DGOSS) es un órgano directivo dependiente de la Secretaría de Estado de Seguridad Social y Pensiones, al cual le corresponde entre otras el desarrollo de las funciones de ordenación jurídica y económico-financieras de la Seguridad Social, planificación y realización de estudios económico-financieros y demográficos, elaboración del anteproyecto de presupuestos de la Seguridad Social, el seguimiento económico y presupuestario de las Entidades Gestoras y Servicios Comunes, conocimiento y evaluación de la gestión y situación económico-financiera de las Mutuas Colaboradoras con la Seguridad Social (8).

Entre sus principales funciones se encuentran: La elaboración de los informes económicos preceptivos en relación con las disposiciones que incidan en la financiación y en el gasto de la Seguridad Social; El diseño, desarrollo y mantenimiento del sistema estadístico de la Seguridad Social, sin perjuicio de las competencias atribuidas, en el ámbito estadístico-contable, a la Intervención General de la Seguridad Social; La realización del seguimiento en el orden económico y presupuestario de las Entidades Gestoras, Servicios Comunes y entidades colaboradoras de la Seguridad Social (9).

Esta entidad cuenta con sitio web oficial el cual se puede acceder de cualquier parte del mundo. El sitio está estructurado por una página principal que ofrece un alto nivel de información sobre la seguridad social a nivel mundial, además cuenta con una sección para la gestión y control de los pensionados en el país, así como el presupuesto que es asignado a este personal. También cuenta con una sección dedicada a los empleadores, para divulgar las ofertas de trabajo del país.

1.2.2 Ministerio de Trabajo y Seguridad Social de Cuba

Ministerio de Trabajo y Seguridad Social (MTSS), es un Organismo de la Administración Central del Estado de la República de Cuba, según dispone el Decreto Ley 147 de la reorganización de los organismos de la Administración Central del Estado, del 21 de abril de 1994, en su artículo 10. El MTSS se encarga de dirigir el trabajo y la seguridad de los trabajadores en la Isla caribeña (10).

El Ministerio de Trabajo y Seguridad Social queda encargado de proponer, dirigir, controlar y evaluar sistemáticamente la política del Estado y el Gobierno en materia laboral, salarial, seguridad y protección en el trabajo y de prevención, atención y seguridad social, con las atribuciones y funciones específicas que se otorgan en el propio acuerdo (10).

Este ministerio tiene como función fundamental desarrollar el pleno empleo, la Gestión Estratégica de los Recursos Humanos para el logro de un alto desempeño humano, económico y social, y el nuevo modelo de gestión de la Seguridad Social (10).

Esta entidad cuenta con un portal web el cual permite la interacción de los usuarios con la entidad. Este sitio está compuesto en primer lugar por una sección principal donde se gestiona el funcionamiento del ministerio, estructura y funciones de trabajo. Cuenta con un espacio de descargas, para que los usuarios descarguen las aplicaciones que trabajan directamente con la entidad. Presenta también una sección de atención a la población donde los usuarios pueden realizar una queja o inquietud. Entre las funcionalidades más importantes del sitio se encuentra gestionar algún trámite, donde el usuario puede solicitar un empleo o acudir a la protección social.

1.2.3 Instituto Nacional de Seguridad Social de Cuba

El Instituto Nacional de Seguridad Social tiene como misión Garantizar el servicio a la población beneficiaria de la seguridad social, velar por la correcta planificación, administración y control de los gastos de seguridad social, así como de los recursos materiales y humanos destinados a ese fin (11).

El INASS administra el presupuesto que anualmente el Estado destina a la seguridad social, es una entidad de carácter nacional, cuenta con una sede central integrada por cinco direcciones, una subdirección, dos departamentos independientes, una sección y un centro, como límite máximo de unidades organizativas (11).

El INASS cuenta con representaciones territoriales (filiales) en todas las provincias y municipios, con excepción de las provincias de Mayabeque y Artemisa, donde la función es asumida por las direcciones de Trabajo (11).

Esta entidad cuenta con una página disponible donde acceden un gran número de usuarios desde cualquier parte del país con el objetivo fundamental de divulgar la los resultados obtenidos por la INASS cada semestre.

Con el estudio de estos sistemas similares se demostró la presencia de aplicaciones web homólogas con el sistema informático que se desea desarrollar, pero ninguna de las mencionadas anteriormente resuelve la situación problemática planteada en la introducción, pero sirvieron como guía para el desarrollo del proyecto porque permitieron identificar algunas funcionalidades que se pudieran añadir a la aplicación web de la DMTSS del municipio Julio A. Mella como, por ejemplo, el mecanismo para el control y la gestión de los usuarios.

1.3 Metodología de desarrollo de software

Uno de los aspectos fundamentales en la elaboración de un software, es seleccionar las tecnologías que mayores beneficios aporten a la solución informática. Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. En el campo del desarrollo de software, existen dos grupos de metodologías, las denominadas tradicionales (formales) y las ágiles (12).

Ambas propuestas tienen sus propias ventajas y desventajas; de cualquier manera, las metodologías de desarrollo nos dicen el ¿Qué hacer? más no el ¿Cómo hacer?, esto significa que la metodología que elijamos, debe ser adaptada al contexto del proyecto, teniendo en cuenta los recursos técnicos y humanos; tiempo de desarrollo y tipo de sistema. Actualmente, los ambientes de negocio son muy variables, donde las peticiones del cliente varían con facilidad. Esto hace necesario minimizar el tiempo en las entregas de los productos, por lo que el enfoque ágil prevalece sobre el tradicional, al ser este más dinámico (12).

Para la elección del enfoque se utilizó como punto de partida el modelo de Barry Boehm y Richard Turner debido a su fiabilidad a la hora de definir el enfoque del proyecto tomando en cuenta sus características, además de ser un método que solo necesita cinco criterios para lograr este objetivo (13).

Figura 1: Estrella de Boehm-Turner



Para seleccionar la metodología se necesita analizar y medir diversos factores y riesgos asociados al escenario en causa. En el presente trabajo se propone evaluar los siguientes criterios de calidad (Tabla 1) propuesto por Boehm y Turner.

Tabla 1: Criterios para la selección apropiada de la metodología.

Factor	Discriminadores ágiles	Discriminadores formales
Tamaño	Escalabilidad y conocimiento. Dependencia y escalabilidad limitada por el porcentaje alto de conocimiento tácito. Apropiado para equipos y productos pequeños.	Escalabilidad y conocimiento explícito. Apropiado para productos y equipos grandes. Difícil de mantener en pequeños proyectos.
Críticidad	La simplicidad en la documentación y el diseño dificulta los planes de pruebas. No aconsejado para sistemas con niveles de criticidad altos.	Rigor de requisitos y diseño adecuados para procesos de pruebas, verificación y validación. Difícil de gestionar en proyectos de escasa criticidad.
Dinamismo	“Refactorizar” desde un diseño básico hasta el producto final es un método ideal para entornos dinámicos e innovadores, pero muy caro por el “retrabajo” para entornos estables o conocidos.	En sistemas estables y conocidos, a partir de requisitos completos y diseños detallados permite trazar y seguir un plan completo y “hacerlo bien a la primera”.
Personal	Los métodos de trabajo ágiles requieren una masa crítica de técnicos con niveles de experiencia medios o altos, capaces de comprender y adaptar los métodos y las técnicas empleadas. Consiste en grupos pequeños y trabajando en el mismo local.	Aunque es aconsejable contar con personas expertas en las fases de definición del proyecto, luego pueden ejecutarse con menor masa crítica de expertos. Consiste en grupos grandes y pueden trabajar de forma distribuidas.

Cultura	Más apropiado para culturas de “ <i>empowerment</i> ” responsabilidad y horquilla de decisión y libertad personal.	Más apropiado en culturas en las que las personas se sienten seguras con un marco de tareas y responsabilidades bien definido.
---------	--	--

Para el desarrollo del proyecto se cuenta con un equipo constituido por una persona no experta, con conocimientos básicos de las tecnologías a emplear en la solución. El mismo intervendrá en un ambiente dinámico donde las especificaciones y las necesidades del cliente varían. Por otra parte, la criticidad o impacto social del proyecto este se encuentra dentro de la zona de confort donde no existen posibilidades de pérdidas económicas ni de vidas humanas por el uso futuro del sistema, por tanto, se decide utilizar un enfoque ágil.

Existen varias metodologías con enfoque ágil. Analizando las más representativas y utilizadas en el mundo para el proceso de desarrollo de software, a continuación, se exponen alguna de estas.

Proceso Unificado Ágil (AUP)

Desarrollada por *Scott Ambler*, constituye una versión simplificada del Proceso Unificado Racional (RUP, por las siglas de Rational Unified Process). Es una metodología que combina procesos propios del concepto unificado tradicional con técnicas ágiles, con el objetivo de mejorar la productividad. Permitiendo así describir de manera simple y fácil de entender cómo desarrollar aplicaciones de software de negocio (14). El AUP aplica técnicas ágiles incluyendo:

- Desarrollo Dirigido por Pruebas (test driven development - TDD en inglés)
- Modelado ágil
- Gestión de Cambios ágil
- Refactorización de Base de Datos para mejorar la productividad.

Crystal Methodologies

La familia Crystal es una colección de metodologías ágiles de desarrollo de software que pueden utilizarse para diferentes proyectos de software dependiendo del tamaño, la complejidad, la criticidad y el número de personas involucradas. Fue desarrollada por Alistair Cockburn a principios de 1990 mientras trabajando en IBM. Según la filosofía de Cockburn "En la medida en que puedas sustituir la documentación escrita con la interacción cara a cara, se puede mejorar la probabilidad de entregar el sistema". Los métodos Crystal se centran en las personas y en la comunicación entre las personas más que en los procesos para entregar con frecuencia un software que funcione (15). La familia Crystal incluye una serie de métodos representados por diferentes colores ordenados en opacidad ascendente.

SCRUM

Scrum fue desarrollado inicialmente para gestionar y desarrollar productos. Desde principios de los años 90 Scrum se ha usado ampliamente en todo el mundo para (16):

- Investigar e identificar mercados viables, tecnologías y capacidades de productos.
- Desarrollar productos y mejoras.
- Liberar productos y mejoras tantas veces como sea posible durante el día.
- Desarrollar y mantener ambientes en la Nube (en línea, seguros, bajo demanda) y otros entornos operacionales para el uso de productos.
- Mantener y renovar productos.

SCRUM se ha usado para desarrollar software, hardware, software embebido, redes de funciones interactivas, vehículos autónomos, escuelas, gobiernos, mercadeo, también para gestionar la operación de organizaciones y casi todo lo que usamos en nuestra vida diaria, como individuo y como sociedad (16).

Extreme Programming (XP)

XP es una metodología ágil defendida por Kent Beck, Ron Jeffries, Ward Cunningham entre otros. Se compone de un conjunto de valores y prácticas que están en camino desde mediados de los 90s. Muchos equipos y empresas buscan entender el desarrollo del software desde otra perspectiva. Desde su implementación provocó un cambio en la manera de gestionar proyectos ya que se basa en la perspectiva de la gestión técnica utilizando cada vez menos las prácticas de ingeniería (17).

La programación extrema es una metodología ágil y ligera de desarrollo de software que tiene la capacidad de responder rápidamente a requisitos cambiantes. Además, sugiere conseguir diseños simples con el objetivo de hacer todo lo menos complicado para el usuario, haciéndolo entendible e implementable. Es una metodología ágil debido a que permite priorizar y adoptar control de procesos ya sea de manera empírica o de manera definida. La misma se centra en potenciar las relaciones interpersonales del equipo de desarrollo como clave del éxito mediante el trabajo en equipo, el aprendizaje continuo y el buen clima de trabajo(17) .

Esta metodología, trata de realizar ciclos de desarrollo cortos llamados iteraciones, con entregables funcionales al finalizar cada ciclo. En cada iteración se realiza un ciclo completo de análisis, diseño, desarrollo y pruebas, pero utilizando un conjunto de reglas y prácticas que la caracterizan. La misma propone técnicas como diseño simple, pruebas intensivas, refactorización y programación en pareja (18).

Ciclo de vida de la Metodología XP

Exploración: Esta fase determina donde se gana confianza las herramientas que se utilizarán, los miembros del ED (Equipo de desarrollo) se conocen y aprenden a confiar unos en otros. Durante esta fase los programadores deben estimar cada tarea de programación en la que hayan sido divididas las historias de usuario, mientras los clientes van escribiendo las historias de usuarios.

Planificación: El objetivo de la fase de planificación es que el cliente y el ED lleguen a una serie de acuerdos y se elabore el plan de entrega. En esta fase el cliente asigna prioridad a las historias de usuario, se estima el costo total del sistema, el alcance de las entregas y para qué fecha culminará.

Iteraciones: Esta fase incluye varias iteraciones antes de que se libere la primera versión del software. El plan de entrega se divide en diferentes planes de iteración. Cada iteración puede tener de una a cuatro semanas de duración. De la primera iteración se obtendrá la arquitectura del futuro sistema.

Producción: Requiere de pruebas y chequeos de funcionamiento y rendimiento del sistema antes de ser liberado a los clientes. Durante esta fase las iteraciones deben dinamizarse, duran entre una y tres semanas. Las nuevas ideas y sugerencias se posponen y se documentan para ser implementadas en la fase de mantenimiento.

Mantenimiento: En esta fase, después de que el sistema pasó la fase de producción, la velocidad de desarrollo tiende a disminuir. Esta fase requiere de un cambio en el equipo desarrollo y en su estructura.

Muerte: En esta fase el cliente debe quedar satisfecho con el trabajo realizado y no escribir ninguna otra historia de usuario. Se procede a escribir la documentación del sistema si no va a haber más cambios en la arquitectura, ni en el código.

Metodología Seleccionada

Para la presente propuesta de solución se selecciona la metodología XP, esto debido a su tasa de errores muy pequeña. Además, fomenta la comunicación entre el desarrollador y el cliente. Posee también una alta eficiencia en el proceso de prueba y planificación, lo que mejora la experiencia de trabajo de este proyecto que solo cuenta con un desarrollador. (Ver anexo 2)

1.4 Descripción de las herramientas y tecnologías para el desarrollo del portal web

Los modelos y tecnologías de desarrollo web han evolucionado mucho en la última década, existen multitud de aplicaciones, frameworks, librerías, arquitecturas y sistemas de publicación en diferentes versiones que a su vez reciben cambios o mejoran con el tiempo. Para ello en este epígrafe se detallarán las tecnologías a utilizar en el desarrollo de la solución informática.

1.4.1 Frameworks

Un **framework** es un marco o esquema de trabajo generalmente utilizado por programadores para realizar el desarrollo de software. Utilizar un framework permite agilizar los procesos de desarrollo ya que evita tener que escribir código de forma repetitiva, asegura unas buenas prácticas y la consistencia del código. Un framework es por tanto un conjunto de herramientas y módulos que pueden ser reutilizados para varios proyectos (19).

Utilizar un framework implica una gran ventaja para el desarrollo de software debido a que el programador ahorra tiempo ya que dispone ya del esqueleto sobre el que desarrollar una aplicación. Además, facilita los desarrollos colaborativos, al dejar definidos unos estándares de programación. También al estar ampliamente extendido, es más fácil encontrar herramientas, módulos e información para utilizarlo (20).

Estas son las 3 principales ventajas de utilizar frameworks web:

1. Fácil depuración y mantenimiento: La mayoría de los lenguajes de programación no dan importancia a la legibilidad y el mantenimiento del código. Pero muchos marcos de desarrollo de sitios web populares lo hacen. Los frameworks se recomiendan para el desarrollo web a medida porque tienen mucho que ver con la facilidad de depuración y soporte.

2. Reducir la longitud del código: Si se utilizan frameworks, no es necesario escribir largas líneas de código para añadir funcionalidades estándar a un sitio web. Los frameworks vienen con características de generación de código para promover la simplicidad y concisión, reduciendo el tiempo y el esfuerzo requerido por los desarrolladores. Además, los frameworks proporcionan herramientas y funciones que ayudan a los desarrolladores a automatizar tareas habituales como la autenticación, la asignación de URL, el almacenamiento en caché, etc.

3. Seguridad mejorada: Los frameworks proporcionan características y mecanismos de seguridad integrados que ayudan a los desarrolladores a proteger los sitios web de las amenazas de seguridad presentes y futuras. Mediante el uso de frameworks, los programadores pueden proteger los sitios web de diversos ciberataques como la manipulación de datos, DDoS, inyecciones SQL, etc. Además, se pueden utilizar marcos web de código abierto para crear especificaciones de seguridad personalizadas para los sitios web.

Laravel 8

Laravel es una herramienta de código abierto, desarrollada con todas las opciones disponibles para elaborar tus proyectos web sin perder el tiempo creando todo desde cero. Requiere menos código que escribir y tienes un apartado de bibliotecas diseñado para tareas sencillas y comunes. Laravel tiene muchas funcionalidades, con las cuales puedes contar y que te harán preferir esta herramienta por encima de otras. Laravel usa muchas herramientas o utilidades con distintos fines (21).

Características de Laravel:

Su motor de plantilla, llamado **Blade**, da numerosas posibilidades para hacer unas páginas visualmente muy potentes y eficaces, capaz de utilizar sus propias variables y reutilizarlas.

Su arquitectura es conocida como **MVC (Modelo-Vista-Controlador)** que da muchas facilidades para relacionar de manera clara y sencilla todas las partes de una aplicación. Esta arquitectura es muy usada en el mundo del software, otros framework pueden distintos de Laravel pueden resultar muy similares gracias a compartir la misma arquitectura MVC.

1.4.2 Lenguajes de programación

En este epígrafe se especificarán los tipos de lenguajes de programación a usar para el desarrollo del sistema informático.

HTML5

El HTML es el principal componente de una página web, sin un documento HTML no existe una página web. Es el lenguaje que le da la estructura y organización a una página web. Es el esqueleto de una página. Dentro del documento HTML se definen todos los elementos que componen una página, cualquier información adicional o recurso externo que esta vaya a utilizar(22)

Ventajas:

- Es nativo, y por tanto independiente de plugins de terceros. Es decir, no pertenece a nadie, es opensource.
- Es más semántico, con etiquetas que permiten clasificar y ordenar en distintos niveles y estructuras el contenido. Además, incorpora metadatos de manera más formal, favoreciendo el posicionamiento SEO y la accesibilidad.
- El código es más simple lo que permite hacer páginas más ligeras que se cargan más rápidamente favoreciendo la usabilidad y la indexación en buscadores.
- Ofrece una compatibilidad mayor con los navegadores de dispositivos móviles.

CSS3

CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

Ventajas:

- El uso de las tablas nos permitía crear diseños complejos de forma mucho más sencilla que utilizando CSS, aunque CSS3 está intentando facilitar dicho trabajo.
- Nuevos modelos de color que incluyen translucidez y degradado sin imágenes.
- Efectos visuales como sombras en textos y cajas, bordes redondeados y con imágenes, múltiples fondos para un mismo elemento.

JavaScript ES2017

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico (23).

Ventajas:

- JavaScript tiende a ser muy rápido porque a menudo se ejecuta inmediatamente en el navegador. Entonces mientras no requiera recursos externos, JavaScript no tiene permitido retrasarse por llamados del servidor backend.

- La sintaxis de JavaScript está inspirada por Java y es relativamente sencillo de aprender comparado a otros lenguajes de programación populares como C++.
- A diferencia de PHP u otros lenguajes scripting, JavaScript puede ser usado en cualquier página web. JavaScript puede ser usado en diferentes tipos de aplicaciones gracias al soporte en otros lenguajes como Pearl y PHP.

UML 2.5

El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento (24).

Ventajas:

- UML Se puede usar para diferentes tipos de sistemas.
- UML consolida muchas de las notaciones y conceptos más usadas orientados a objetos.
- UML es fácilmente entendible.

PHP 8

Es un lenguaje de código abierto muy popular en el mundo, adecuado para desarrollo web y que puede ser incrustado de manera muy flexible con HTML5. Por lo tanto, si vemos el internet un gran número de páginas y portales web están usando PHP de manera abierta.

Ventajas:

- Lenguaje totalmente libre y abierto.
- Posee una curva de aprendizaje muy baja.
- Los entornos de desarrollo son de rápida y fácil configuración.
- Fácil de instalar: existen paquetes autoinstalables que integran PHP rápidamente.
- Fácil acceso e integración con la base de datos.

Desventajas:

- El inconveniente es que el código fuente no pueda ser ocultado de una manera eficiente. La ofuscación es una técnica que puede dificultar la lectura del código y, en ciertos aspectos representa tiempos de ejecución.
- Si no lo configuras y/o proteges correctamente dejas abiertas muchas brechas de seguridad que a la larga tendremos problemas.
- Solo se ejecuta en un servidor y se necesita un servidor web para que funcione.

1.4.3 Gestor de base de datos

PostgreSQL es un poderoso sistema de base de datos relacional de código abierto. Usa el lenguaje SQL combinado con muchas características que almacenan y escalan de manera segura las cargas de trabajo de datos más complicadas. Los orígenes de PostgreSQL se remontan a 1986 como parte del proyecto POSTGRES en la Universidad de California Berkeley y tiene más de 30 años de desarrollo activo en la plataforma central (25).

PostgreSQL provee muchas características destinadas a ayudar a los desarrolladores a crear aplicaciones. También permite a los administradores proteger la integridad de los datos y a crear entornos tolerantes a fallos. Además, proporciona administración de sus datos sin importar cuán grande o pequeño sea el conjunto. Es gratuito, de código abierto y altamente extensible. Intenta ajustarse al estándar SQL donde tal conformidad no contradice las características tradicionales. Se admiten muchas de las características requeridas por dicho estándar, aunque a veces con una sintaxis o función ligeramente diferente (26).

1.4.4 Herramientas

A continuación, se definen las principales herramientas a utilizar para la implementación de la aplicación web.

WebStorm 2021.2.2

WebStorm es un entorno de desarrollo integrado para codificar en JavaScript y sus tecnologías relacionadas. Al igual que IntelliJ IDEA y otros IDE de JetBrains, WebStorm hará que su experiencia de desarrollo sea más agradable, automatizando el trabajo de rutina y ayudándole a manejar tareas complejas con facilidad (27).

WebStorm es directamente compatible con JavaScript, TypeScript, React, React Native, Electron, Vue, Angular, Node.js, HTML, hojas de estilo y muchas otras tecnologías. Puede comenzar a programar directamente, sin tener que pensar en instalar ni mantener ningún complemento.

WebStorm comprende perfectamente la estructura de su proyecto y le puede ayudar en todos los aspectos de la escritura de código. Autocompletará su código, detectará y sugerirá correcciones para los errores y redundancias, y le ayudará a refactorizar el código con seguridad.

Google Chrome 93

Es un navegador web desarrollado por la empresa Google, la fase beta fue lanzada el 2 de septiembre del 2008 y la primera versión estable vio la luz el 11 de diciembre del mismo año. Google Chrome se ha expandido no sólo a ser un navegador para pc, sino que ha sabido aprovechar las diferentes tecnologías en crecimiento, como es el caso de tabletas y teléfonos inteligentes entre otros, haciendo de Google Chrome el navegador web con una cuota de mercado del 42%, posicionándolo como el navegador web más usado (28).

Visual Paradigm 15.2

Es una herramienta de Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación (29).

Permite a su equipo gestionar la complejidad de la transformación empresarial para hacer frente a los mercados, las tecnologías y los requisitos normativos que cambian rápidamente. Es una solución integral ideal para la planificación de la arquitectura empresarial y la transformación empresarial, la gestión de proyectos y el desarrollo ágil de software, para que su empresa pueda mantener el control y fomentar el crecimiento (29).

1.5 Conclusiones Parciales

Al concluir con el estudio de este capítulo es posible arribar a las siguientes conclusiones:

- Se ha decidido usar la metodología de desarrollo de software Extreme Programming pues las continuas pruebas generan sistemas muy estables y el procedimiento iterativo en colaboración con el enfoque minimalista garantizan que solo se creen funciones que realmente son importantes para el proyecto.
- Se analizaron distintos marcos de trabajo, seleccionando Laravel para el desarrollo de la solución, y como gestor de base de datos PostgreSQL.

- Para el desarrollo de la solución se selecciona WebStorm como IDE, como lenguaje de modelado UML y la herramienta CASE Visual Paradigm en su versión 15.1.

Capítulo 2. Diseño de la Propuesta de Solución

En el capítulo en primer lugar, se realizará una breve caracterización del entorno de desarrollo desglosando el funcionamiento de la DMTSS del municipio Julio A. Mella en específico y representando el negocio mediante un modelo conceptual. Luego se realizará caracterización del sistema describiendo los usuarios de sistema, la especificación de los requisitos, las historias de usuario, el plan de iteración y el plan de entrega del proyecto. Posteriormente, se detallará la propuesta de solución, para ello se describirá la arquitectura que se desea utilizar unido a los patrones de diseños y se definirán los diagramas de clase, modelo entidad-relación y modelo del despliegue.

2.1 Caracterización del entorno de desarrollo

Para la caracterización del entorno de desarrollo se realizará una breve reseña de las funciones específicas en la DMTSS de municipio Julio A. Mella y se representará el mapa de proceso mediante la representación del modelo conceptual. Además, se describirá cada clase del modelo.

2.1.1 DMTSS del municipio Julio A. Mella

La institución radicada en el municipio Julio A. Mella se vinculan como misión fundamental de ser el organismo de la Administración Central del Estado que tiene la misión de proponer, dirigir y controlar la política del Estado y el Gobierno en materia de trabajo, seguridad y salud en el trabajo, seguridad social y prevención, asistencia y trabajo social, en virtud de los dispuesto en el Acuerdo No. 7335 del Consejo de Ministros de 19 de diciembre del 2012.

La entidad tiene como funciones específicas las siguientes:

- Proponer, dirigir y controlar la política para el empleo, el trabajo en el sector no estatal y la planificación de la fuerza de trabajo, especialmente la fuerza de trabajo calificada.
- Proponer, dirigir y controlar la política salarial y de organización, protección, seguridad e higiene del trabajo y verificar la calidad protectora de los equipos de protección personal.
- Proponer, dirigir y controlar la política de prevención, asistencia y trabajo social.
- Proponer, dirigir y controlar la política de seguridad social; así como planificar, administrar y controlar los gastos del presupuesto de la seguridad social.
- Comprobar el cumplimiento de la legislación de trabajo y seguridad social.

- Intervenir, en su condición de autoridad gubernamental, en la solución de los conflictos derivados del incumplimiento, por alguna de las partes, del Convenio Colectivo del Trabajo.
- Dirigir y controlar técnica, normativa y metodológicamente, los órganos que en las entidades conocen y resuelven los conflictos laborales de los trabajadores.
- Representar al Gobierno ante la Organización Internacional del Trabajo y demás organismos internacionales especializados en materia de trabajo, protección, seguridad e higiene del trabajo, seguridad social y de prevención, asistencia y trabajo social.
- Actuar como órgano de relación con las asociaciones que le competen.

La entidad está estructurada con una: Dirección, una subdirección de Empleo, Subdirección del PATS, Una Unidad de Trabajo Social, un departamento de TPCP, un Departamento Económico un departamento de Administrativo y de Servicio.

Funciones de los Departamentos

Dirección

Dirigir y Controlar la aplicación, las políticas de asistencia social de los trabajadores y su familia y de preservación, atención e integración social de los grupos vulnerables de la población. Garantizar un adecuado y eficaz servicio a la población beneficiaria de la seguridad social, así como el control de los recursos financieros, materiales y humanos destinados a ese fin. Dirigir y controlar la aplicación d la política de seguridad y salud en el trabajo, para prevenir los riesgos que provoquen accidentes de trabajo y enfermedades profesionales.

Subdirector

Asistir al director, ejerciendo las atribuciones y obligaciones específicas que este le asigne; sustituir al director de manera temporal, cuando fuere necesario por razón de ausencia, incapacidad legal, enfermedad o fallecimiento otros, dirige y controla el proceso de ratificación o renovación de los miembros de los Órganos de Justicia Laboral de Base cada tres años y que el funcionamiento de estos esta correspondencia con la legislación vigente.

Departamento de Empleo Estatal y Fuerza de Trabajo Calificada

Controlar y evaluar el cumplimiento en el territorio del balance, disponibilidad y necesidad de la fuerza de trabajo previsto en el plan de la economía; la incorporación de las personas a más de un empleo y de los jubilados reincorporados; los procesos de disponibilidad que se aprueben, de conjunto con los OACE, la Dirección de Trabajo Provincial y las Municipales.

Departamento de Empleo no Estatal

Tramitar y decidir el otorgamiento o no de la autorización para el ejercicio del trabajo por cuenta propia. Notificar a los autorizados de las normas jurídicas y los aspectos que debe cumplir en el ejercicio de la actividad y mantener actualizado el registro de control de las personas autorizadas a ejercer el trabajo por cuenta propia.

Departamento de Asistencia Social

Dirigir y controlar el estricto cumplimiento de la política trazada por el Estado y Gobierno Cubano en materia de prevención, asistencia y trabajo social, auxiliar al presidente del CAP en el logro de estos objetivos y encaminar la labor fundamental a las actividades preventivas que conlleven a una disminución progresiva de las causas que originan la comisión de indisciplinas, ilegalidades y conductas denigrantes.

Departamento de Prevención y Trabajo Social

Departamento de Prevención, Asistencia y Trabajo Social, en el municipio atiende núcleos beneficiarios de la Asistencia Social que reciben prestaciones monetarias, servicios en especies, se atiende a madres de hijos con discapacidad severa, servicio de asistentes social a domicilio, pago de servicios de alimentación, servicios de hogar de ancianos.

Departamento de Seguridad Social

Dirigir y controlar el estricto cumplimiento de la política trazada por el Estado y Gobierno Cubano en materia de seguridad social, logrando la protección oportuna y eficaz de los trabajadores y sus familias en casos de enfermedad o fallecimiento del trabajador, la protección a los jubilados y pensionados.

Departamento de Organización del Trabajo y los Salario

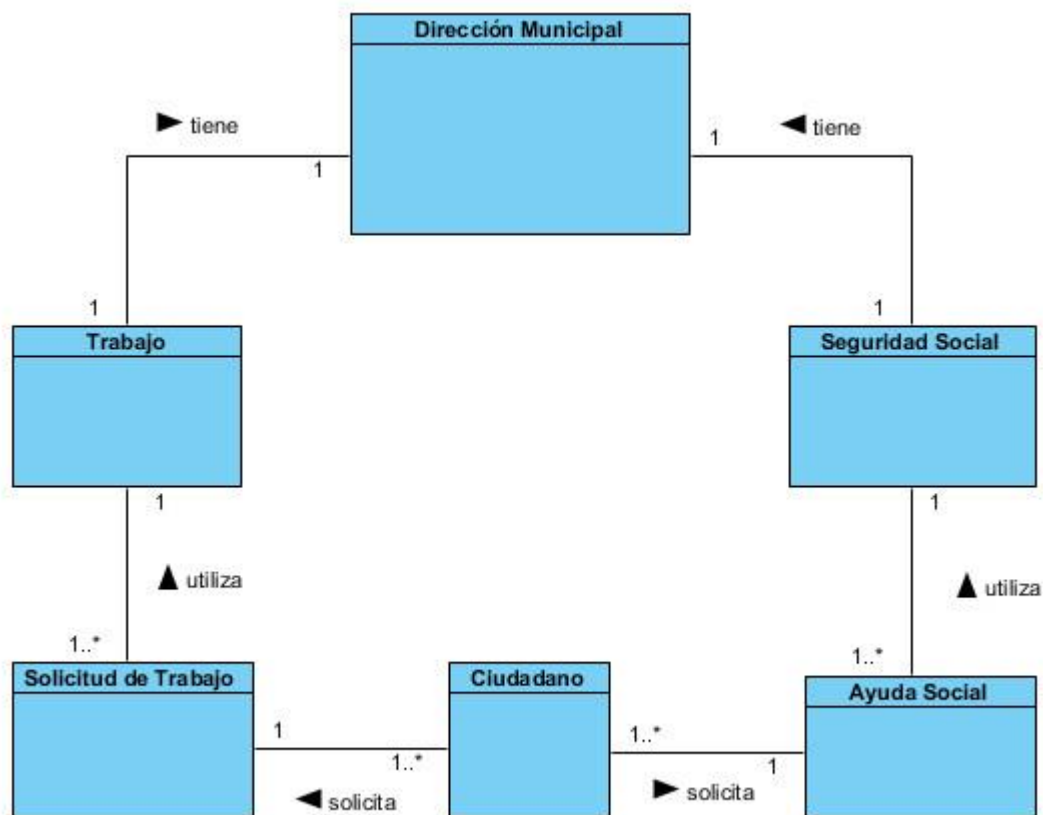
Dirigir y controlar el estricto cumplimiento de la política trazada por el Estado y Gobierno Cubano en materia de la Organización del Trabajo y los Salario y de seguridad y Salud del trabajo, logrando estimular el crecimiento continuo de la productividad y la disminución de los accidentes de trabajo.

2.1.2 Modelo Conceptual

Un diagrama conceptual del negocio no es más que un artefacto construido bajo las reglas de UML durante la concepción de un proyecto informático. Este modelo puede ser utilizado para capturar y expresar el entendimiento ganado sobre el negocio (30).

Se determinó necesario generar este artefacto para mejor entendimiento del negocio. Así identificando los principales conceptos y como se relacionan entre ellos facilita el desarrollo del de la aplicación web.

Figura 2: Modelo conceptual



Este modelo conceptual es una forma sencilla de entender el negocio. La entidad esta dividida en dos sectores fundamentales la dirección de trabajo y la seguridad social. En la dirección de trabajo es la encargada de controlar y asignar las ofertas de empleo que el municipio tiene disponible. Por otra parte, la dirección de seguridad social se encarga de gestionar todas las solicitudes de ayudas sociales en el municipio. De este modo cualquier ciudadano del municipio podrá solicitar una oferta de empleo o solicitar una ayuda social.

La clase Dirección Municipal es la encargada de controlar todo el proceso. La clase Trabajo tendrá como funcionalidad gestionar todas las ofertas de trabajo de la entidad y la clase Seguridad Social controlará todos los procesos sociales de la entidad. Por otro parte las clases Solicitud de Trabajo y Ayuda Social gestionaran las solicitudes de trabajo y de ayuda social respectivamente. La clase usuario será usada para administrar todos los usuarios que acceden al sistema.

2.2 Características del sistema

Para la caracterización del sistema se definirán los usuarios que intervienen en el negocio, se listarán una serie de requisitos funcionales y no funcionales, se definirán las historias de usuario y el plan de iteraciones.

2.2.1 Usuarios del sistema

Los usuarios del sistema son aquellas personas que interactúan con la aplicación. Para ellos existen algunas restricciones específicas. A continuación, se describen los usuarios del sistema con sus responsabilidades.

- **Administrador:** Es el encargado de la gestión de los usuarios, las ofertas de trabajo, las solicitudes de trabajo y las solicitudes de ayuda social.
- **Usuario:** son aquellas personas que utilizarán la aplicación informática de forma habitual.

2.2.2 Especificación de requisitos

El proceso de requisitos es una etapa fundamental en todo proyecto de desarrollo de software pues, junto con la gestión de requisitos, garantizan que el producto desarrollado satisfaga efectivamente las necesidades que realmente tiene el cliente. Para lograrlo, la Ingeniería de Requisitos plantea la ejecución disciplinada de una serie de actividades entre las que adquiere gran relevancia, por su papel en la consecución de la calidad, la especificación o definición de los requisitos (31).

Requisitos funcionales

Los requisitos son, por definición “una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo”, y que “expresan las necesidades y limitaciones impuestas a un producto de software que contribuyen a la solución de algún problema en el mundo real”. La Ingeniería de Requisitos es tratada como una disciplina que tiene por propósito “desarrollar una especificación de requerimientos completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en dónde se describen las funciones que realizará el sistema” (32).

RF1 Gestionar Ofertas de Trabajo

RF1.1 Añadir Ofertas de Trabajo

RF1.2 Listar Ofertas de Trabajo

RF1.3 Modificar Ofertas de Trabajo

RF1.4 Eliminar Ofertas de Trabajo

RF2 Gestionar solicitud de Trabajo

RF2.1 Añadir Solicitud de Trabajo

RF2.2 Listar Solicitud de Trabajo

RF2.3 Modificar Solicitud de Trabajo

RF2.4 Eliminar Solicitud de Trabajo

RF3 Gestionar Solicitud de Ayuda

RF3.1 Añadir Solicitud de Ayuda

RF3.2 Listar Solicitud de Ayuda

RF3.3 Modificar Solicitud de Ayuda

RF3.4 Eliminar Solicitud de Ayuda

RF4 Gestionar Ayuda Solicitada

RF4.1 Añadir Ayuda Solicitada

RF4.1 Listar Ayuda Solicitada

RF4.1 Modificar Ayuda Solicitada

RF4.1 Eliminar Ayuda Solicitada

RF5 Gestionar Usuario

RF5.1 Añadir Usuario

RF5.2 Listar Usuario

RF5.3 Modificar Usuario

RF5.4 Eliminar Usuario

RF6 Gestionar Departamento

RF6.1 Añadir Departamento

RF6.4 Listar Departamento

RF6.3 Modificar Departamento

RF6.4 Eliminar Departamento

Requisitos no funcionales

Los requisitos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación (33).

Definición de requisitos no funcionales tomando como guía el Estándar de Calidad Internacional ISO/IEC 9126-1.

Usabilidad

RNF1 Interfaz amigable fáciles de usar por el usuario.

RNF2 Fácil navegación dentro del sistema.

Seguridad

RF3 El acceso debe estar restringido por usuario y contraseña.

RF4 Cada usuario solo puede realizar acciones de acuerdo con su rol.

Software

RNF5 El navegador debe estar actualizado con versión Google Chrome 93.

Apariencia

RNF6 El sistema debe contener los colores institucionales.

2.2.3 Historias de Usuarios

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (34).

Tabla 2: HU1-Gestionar Ofertas de Trabajo.

Historia de Usuario	
Número: 1	Nombre: Gestionar Ofertas de Trabajo
Usuarios: Administrador	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Estimación: 4 días	Iteración Asignada: 1
Descripción: El administrador gestiona todas las ofertas trabajos que están disponibles en el municipio	
Observaciones: El administrador se encargará de adicionar, listar, modificar y eliminar una oferta de trabajo	

Tabla 3: HU2- Gestionar Solicitud de Trabajo.

Historia de Usuario	
Número: 2	Nombre: Gestionar Solicitud de Trabajo
Usuarios: Administrador	
Prioridad en el negocio: Alta	Nivel de complejidad: Media
Estimación: 3 días	Iteración Asignada: 1
Descripción: El administrador gestiona todas las solicitudes de trabajo del municipio	
Observaciones: El administrador se encargará de adicionar, listar, modificar y eliminar una solicitud de trabajo	

Para más historias de usuario ver Anexo 1.

2.2.4 Plan de Iteraciones

Todo proyecto que se guía por la metodología XP, se planifica en iteraciones de aproximadamente tres semanas de duración. Para cada iteración se define un conjunto de HU que se van a implementar.

Tabla 4: Plan de Entrega.

Iteraciones	Orden de las HU a Implementar	Tiempo de trabado
1	Gestionar Ofertas de Trabajo, Gestionar Solicitud de Trabajo	1 semanas
2	Gestionar Solicitud de Ayuda, Gestionar Ayuda Solicitada	1 semanas
3	Gestionar Usuario, Gestionar Departamento	1 semanas

2.2.5 Plan de Entrega

El plan de entregas es el compromiso final del equipo de trabajo con los clientes. Esta es una cuestión de vital importancia para ambas partes, ya que la entrega tardía o temprana de la solución, repercute notablemente en la economía y la moral de los implicados.

Tabla 5: Plan de Entrega.

Plan de Entrega		
Iteración 1	Iteración 2	Iteración 3
1 de septiembre de 2021	8 de septiembre de 2021	15 de septiembre de 2021

2.3 Descripción Propuesta de solución

En la descripción de la propuesta de solución se realizará un breve resumen de la solución informática que se desea obtener, la descripción de la arquitectura con su patrón arquitectónico y los patrones de diseños que se desean utilizar. También se diseñarán los diagramas de clase, de entidad-relación y modelo de despliegue.

2.3.1 Propuesta de solución para la DMTSS del municipio Julio A. Mella

Para dar solución a la problemática presente en la DMTSS del municipio Julio A. Mella se propone desarrollar un portal web con el objetivo de fortalecer el funcionamiento de la entidad y la gestión de los procesos que se llevan a cabo en la misma. De esta manera logrará una mayor conformidad a los clientes que son atendidos.

El sistema informático estará compuesto por una página de bienvenida accesible para todos los usuarios donde se muestran las principales características de la entidad como son el nombre, la función la misión, la visión y la historia. Además, esta página principal muestreará cuatro enlaces principales para acceder a los departamentos, a las ofertas de trabajos, a solicitar un trabajo y a solicitar una ayuda social. También la aplicación brindara la opción de registrarse en el sistema o de autenticarse en el mismo. Para que un usuario realice una solicitud de trabajo o solicite una ayuda social deberá estar autenticado.

2.3.2 Descripción de la arquitectura

Los sistemas de software dependen directamente de su diseño arquitectónico para garantizar su uso a largo plazo, un mantenimiento eficiente y una evolución adecuada en un entorno de ejecución. Diversos autores afirman que las arquitecturas son el factor crítico en la capacidad de los sistemas de software para perdurar y evolucionar (35). La creatividad de los arquitectos de software puede llevar a diferentes diseños o arquitecturas de soluciones que sirvan para el mismo propósito. Los diagramas que ilustran los diseños propuestos pueden valer miles de palabras (36).

Patrón arquitectónico Modelo Vista Controlador

El modelo–vista–controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento. De manera genérica, los componentes de MVC se podrían definir como sigue (37): (**Ver Anexo 3**)

El Modelo: Es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos a dicha información, tanto consultas como actualizaciones. Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.

El Controlador: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud de información (por ejemplo, editar un documento o un registro en una base de datos). Por tanto, se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.

La Vista: Presenta el 'modelo' y los datos preparados por el controlador al usuario de forma visual. El usuario podrá interactuar con la vista y realizar otras peticiones que se enviarán al controlador.

2.3.3 Patrones de Diseño

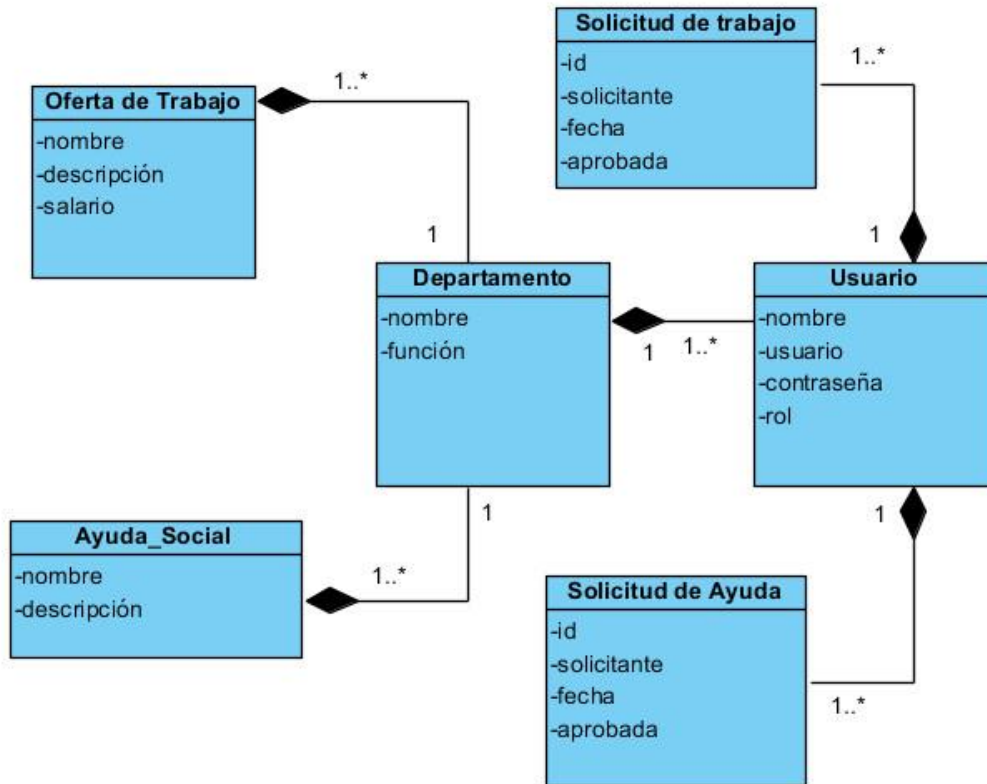
Patrones generales de asignación de responsabilidades (GRASP)

- **Experto:** Básico para asignación de responsabilidades. Es utilizado en la clase controladora Ofertas de Trabajo la cual es encargada de crear un objeto que conoce todos sus métodos y atributos. De este modo obtendremos mayor cohesión, seguridad y encapsulamiento.
- **Bajo Acoplamiento:** usar la menor interacción o ligamiento entre clases, para crear independencia, así, en caso de modificar alguna clase, esto genera la menor afectación posible en la estructura del sistema.
- **Alta Cohesión:** Es utilizado en todas las clases implementadas en el proyecto para que la información almacenada por cada clase debe estar en correspondencia con esta.

2.3.4 Diagrama de Clase

El diagrama de clases describe gráficamente las especificaciones de las clases de software en una aplicación. A diferencia del modelo conceptual, un diagrama de este tipo contiene las definiciones de las entidades del software en vez de conceptos del mundo real. El UML no define concretamente un elemento denominado "diagrama clases del diseño", sino que se sirve de un término más genérico: "diagrama de clases". Se trata de una perspectiva desde el punto de vista del diseño de las entidades de software y no de una concepción analítica sobre los conceptos del dominio (38).

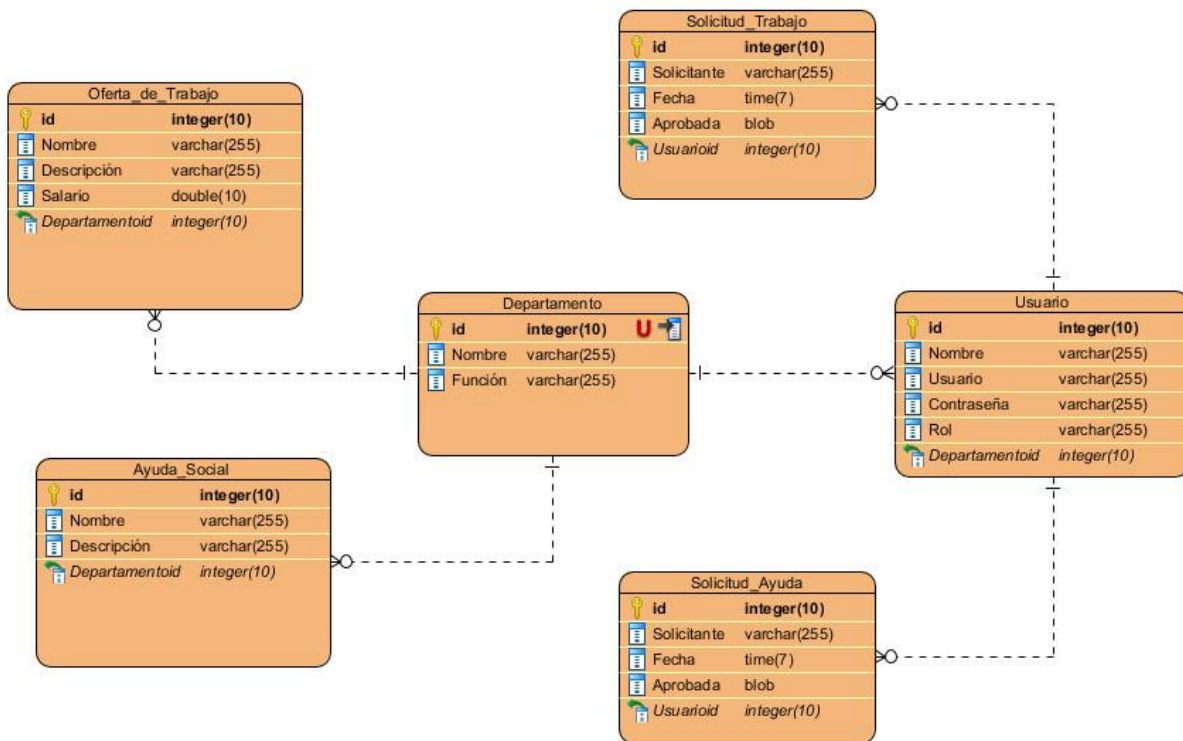
Figura 3: Diagrama de clase



2.3.5 Modelo Entidad-Relación

El Modelo Entidad Relación (ER) permite desarrollar un diseño de base de datos en un esquema de alto nivel conceptual sin considerar los problemas de bajo nivel como la eficiencia, el modelo implícito del administrador de base de datos o las estructuras físicas de los datos. El Modelo Entidad Relación se hizo muy popular para el diseño de base de datos y es usado extensivamente. Para aumentar su poder de expresión, muchos investigadores han introducido o propuesto ciertas extensiones a este modelo. Algunas de estas extensiones son importantes, mientras que otras agregan poco poder de expresión, pero proveen características auxiliares. Puesto que el Modelo Entidad Relación es ampliamente usado, es importante conocer qué extensiones han sido propuestas para este modelo y qué ofrecen estas extensiones a los usuarios (39).

Figura 4: Modelo Entidad-Relación



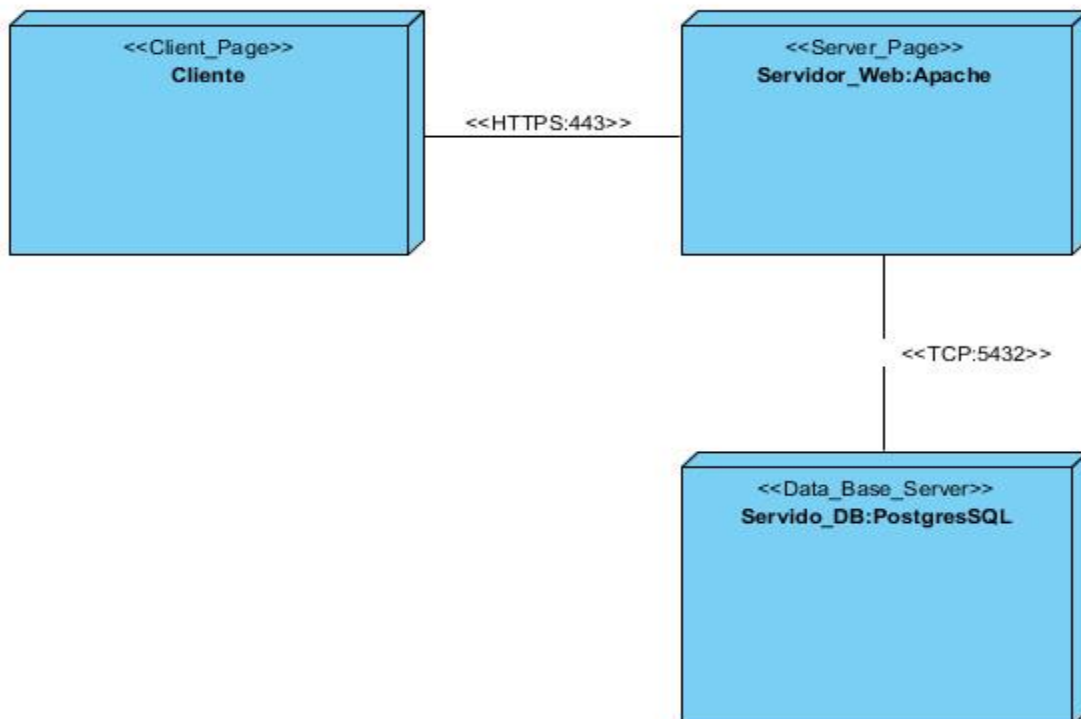
2.3.6 Modelo de despliegue

Diagrama de Despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos (40).

Los diagramas de despliegue son los complementos de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema. Describen la topología del sistema la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. Los diagramas de despliegue representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP (41).

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 la red, el tiempo de respuesta (42).

Figura 5: Modelo de Despliegue



2.4 Conclusiones Parciales

- En la caracterización del entorno de desarrollo se obtuvo el Modelo Conceptual como apoyo para una mejor comprensión del negocio, además se realizó el Diagrama de Clases y el Modelo Entidad-Relación.
- En la fase de exploración se realizó la detección de los usuarios a interactuar con el sistema, además se levantaron los requisitos funcionales y no funcionales. Se confeccionaron también las Historias de Usuarios.
- Fue necesario el diseño del diagrama de clase, entidad-relación y modelo del despliegue para un mayor entendimiento del desarrollo del proyecto, lo cual servirá de guía para la implementación posteriormente.

Capítulo 3 Implementación y Pruebas del Sistema

La implementación del sistema es una de las fases imprescindibles dentro del proceso de desarrollo de software. Esta fase comprende la materialización, en forma de código, de todos los artefactos, descripciones y arquitectura propuestos en la etapa de análisis y diseño; con el objetivo de conformar el producto final requerido por el cliente.

Aparejado al proceso de implementación, el software que se construye debe ser sometido a determinadas pruebas que corroboren la correspondencia entre el producto y los requisitos definidos en las etapas anteriores. A esta etapa se le conoce como validación del sistema y en ella, pueden realizarse diferentes tipos de pruebas en función de los objetivos de las mismas.

En este capítulo se cuantifica el nivel de fiabilidad de los resultados obtenidos y la calidad en el desarrollo de la propuesta de solución. Para lograr esta valoración se definen estrategias de pruebas de software en correspondencia con los estándares actuales, aplicando distintas disciplinas de pruebas internas y de aceptación que se definen en la metodología que se empleó como guía durante todo el desarrollo de la propuesta de solución.

3.1 Justificación del estándar de codificación.

Los estándares de codificación son especificaciones o estilos que establecen la forma de generar del código funcional de las aplicaciones informáticas. Puesto que, en muchas ocasiones, los sistemas de cómputo son implementados por varios programadores, la adopción inicial de un único estilo de codificación constituye uno de los factores de mayor peso en la calidad, rendimiento, legibilidad y capacidad de mantenimiento del producto final (43).

XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios (44).

Entre los estándares de codificación o programación utilizados en la aplicación informática tenemos:

- Utilizar 4 espacios para la sangría.
- Cada línea de código no debe exceder los 80 caracteres en la medida de lo posible.
- La declaración de importación debe escribirse en líneas separadas.
- Poner sufijo *Exception* a todas las excepciones.
- Poner sufijo *View* a las vistas.

- Poner sufijo *Model* a los modelos.
- Poner sufijo *Controller* a las clases controladoras.
- Las variables usaran las letras en minúscula y separando las palabras con un guion bajo.
- Usar verbos para los métodos y sustantivos para atributos.
- Escribir comentarios para encontrar código más fácilmente.
- Los estilos van en archivos separados del código HTML.
- Usar nombres de variable significativos como *usuario* o *equipo* en lugar de *u* o *e*.

3.2 Implementación y codificación de la propuesta de solución

Se implementaron las Historias de Usuario en tres iteraciones, repasando y corrigiendo los detalles con cada una. Al culminar el proceso se realizó la revisión del plan de iteraciones y se corrigieron los cambios identificados. Específicamente las pruebas de software permiten evaluar las soluciones y determinar el nivel de calidad que poseen.

- **Iteración No.1:** Se implementó la HU 1 Gestionar Ofertas de Trabajo y la HU2 Gestionar Solicitud de Trabajo.
- **Iteración No.2:** Se implementó la HU 3 Gestionar Solicitud de Ayuda y la HU4 Gestionar Ayuda Solicitada
- **Iteración No.3:** Se implementó la HU 5 Gestionar Usuario y la HU6 Gestionar Departamento.

Como resultado de las iteraciones anteriores, se realizó la siguiente tabla que muestra el tiempo estimado y real de cada Historia de usuario y su respectiva iteración.

Tabla 6: Iteraciones por Historia de Usuario.

Historia de Usuario	Tiempo Estimado	Tiempo Real	Iteración
Gestionar Ofertas de Trabajo	4 días	3 días	1
Gestionar Solicitud de Trabajo	3 días	3 días	1
Gestionar Solicitud	4 días	3 días	2
Gestionar Ayuda Solicitada	3 días	3 días	2
Gestionar Usuario	4 días	4 días	3
Gestionar Departamento	3 días	2 días	3

3.2.1 Codificación

A continuación, se muestran fragmentos de códigos que son considerado de vital importancia para el desarrollo del proyecto.

Imagen 1: Implementación de la vista Ofertas de Trabajo

```
<x-app-layout>
  <div class="mt-10">
    <div class="container mt-10">
      <h1 class="pt-4 tracking-widest font-bold border-b-2 pb-3">Ofertas de Trabajo</h1>
      <div class="grid grid-cols-3 gap-3 pt-10 pb-20">
        @foreach($ofertas as $oferta)
          <div class="flex flex-column shadow p-4 rounded col-span-1 bg-white dark:bg-gray-800">
            <h1 class="text-2xl">{{ $oferta['nombre'] }}</h1>
            <div class="flex flex-col mb-2">
              <span class="text-sm"><b>Nombre: </b>{{ $oferta['contacto']->nombre }}</span>
              <span class="text-sm"><b>Num. telefono: </b>{{ $oferta['contacto']->num_telefono }}</span>
              <span class="text-sm"><b>Correo: </b>{{ $oferta['contacto']->correo }}</span>
            </div>
            <p class="font-sans truncate">
              {{ $oferta['descripcion'] }}
            </p>
            <div class="w-full flex justify-end">
              <a href="" style="...">Leer más →</a>
            </div>
          </div>
        @endforeach
      </div>
    </div>
  </div>
</x-app-layout>
```

Imagen 2: Implementación del controlador Ofertas de Trabajo

```
use App\Models\OfertaTrabajo;
use Illuminate\Contracts\Foundation\Application;
use Illuminate\Contracts\View\Factory;
use Illuminate\Contracts\View\View;
use Illuminate\Http\Request;
use Illuminate\Http\Response;

class OfertaTrabajoController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return Application|Factory|View
     */
    public function index()
    {
        $ofertas = OfertaTrabajo::query()
            ->orderByDesc('created_at')
            ->take(12)->get();
        return view('pages.ofertas-trabajo', compact('ofertas'));
    }
}
```

Imagen 3: Implementación del modelo Ofertas de Trabajo

```
namespace App\Models;

use ...

class OfertaTrabajo extends Model
{
    use HasFactory;

    protected $table = 'ofertas_trabajos';

    protected $fillable = [
        'nombre',
        'descripcion',
        'contacto',
    ];

    protected $casts = [
        'created_at' => 'datetime',
        'updated_at' => 'datetime',
        'contacto' => 'object'
    ];
}
```

3.3 Pruebas y resultados

La prueba es un proceso que se enfoca sobre la lógica interna del software y las funciones externas. Es un proceso de ejecución de un programa con la intención de descubrir un error, no puede asegurar la ausencia de defectos; sólo puede demostrar que existen defectos en el software (45).

La metodología XP enfatiza en la realización de pruebas a lo largo de todo el desarrollo del software, con el fin de lograr un producto con calidad, reduciendo el número de errores y disminuyendo el tiempo transcurrido entre la aparición de un error y su corrección. En este proceso no solo participa el equipo de desarrollo, también es importante la colaboración del cliente, sobre todo en las pruebas de aceptación (46).

3.3.1 Desarrollo dirigido por pruebas (TDD)

Desarrollo Orientado a Pruebas es el desarrollo de software orientado a objetos que incluye un ciclo de pruebas constantes al software, se emplea en las metodologías ágiles, en las cuales se escribe código para superar las pruebas que se han especificado con anterioridad, y en algunas ocasiones sustituyen la especificación de requisitos(47).

Los pasos dentro del enfoque TDD son los siguientes (48):

1. Escribir una prueba para la funcionalidad que se va a implementar.
2. Ejecutar la prueba (la nueva prueba debe fallar, porque no hay código para ella) - este paso verifica que las pruebas están escritas correctamente.
3. Implementar la cantidad mínima de código para que todas las pruebas pasen - este paso verifica que el código implementa la funcionalidad prevista para una iteración determinada. En caso de fallo modificar el código hasta que se superen todas las pruebas.
4. Refactorizar el código para mejorar su legibilidad y mantenimiento.
5. Vuelva al paso 1.

3.3.2 Pruebas Funcionales

Prueba de Caja Negra

La técnica de caja negra puede utilizarse para lograr objetivos de cobertura de entrada y salida, con entradas humanas, vía interfaces a un sistema, o parámetros de interfaz de las pruebas de integración”. En esta técnica es importante identificar las clases de equivalencia, por ejemplo, rango de valores entre 1 y 10 serán las clases de equivalencia, es decir que todo valor menor a 1 y todo valor mayor a 10 serán valores inválidos. Luego se generan los casos de prueba con diferentes valores para asegurar que la aplicación solo acepte valores entre 1 y 10 (49).

Se emplearon las pruebas de caja negra con el objetivo de encontrar errores en las siguientes categorías:

- Funciones incorrectas o faltantes.
- Errores de interfaz.
- Errores en las estructuras de datos.

Para realizarle pruebas a las funcionalidades del sistema se utilizó el nivel de pruebas de aceptación utilizando la técnica de **caja negra**. Las pruebas de aceptación son las especificaciones para el comportamiento deseado y la funcionalidad de un sistema. Estas pruebas fueron escritas por el equipo de desarrollo en conjunto con el cliente antes de desarrollar cada HU y aplicadas al final de cada iteración. A continuación, se presentan los casos de pruebas de aceptación pertenecientes a las HU mostradas en el Capítulo 2.

Tabla 1: Prueba de aceptación 1

Caso de prueba 1: RF1_ Añadir Oferta de trabajo							
Condiciones de ejecución: El usuario debe estar autenticado en el sistema con el rol de administrador							
Escenario	Descripción	Nombre del trabajo	Descripción de la Oferta de trabajo	Nombre del administrador	Contacto del admin	Respuesta del Sistema	Flujo Central
EC 1.1 Añadir una oferta de trabajo al sistema de forma correcta	Se muestra una interfaz para rellenar los datos para añadir una oferta de trabajo, si todos los datos son correctos la oferta es añadida al sistema.	Programador en el centro de desarrollo Desowft	!!!!Es un trabajo honrado!!!!	Pérez	56358722	Se agrega la oferta de trabajo de forma correcta y se muestra una notificación.	Acceder al menú de enlace Ofertas de Trabajos, click en añadir ofertas de trabajo, rellenar los datos y aceptar
EC 1.2 Añadir una oferta de trabajo con campos vacío	Interfaz para llenar los datos de la oferta de trabajo, si existe algún campo vacío, se muestra un mensaje pidiendo rellenar el campo			Pérez	56358722	El sistema no agrega la oferta de trabajo y muestra un mensaje de error	Acceder al menú de enlace Ofertas de Trabajos, click en añadir ofertas de trabajo, rellenar los datos y dejar algunos campos vacíos y aceptar
EC 1.3 Añadir una oferta de trabajo con caracteres extraños	Interfaz para llenar los datos de la oferta de trabajo, si existe algún carácter extraño, se muestra un mensaje de error	Programador en el centro de desarrollo Desowft	*#\$%&/()	&&&/	56*****	El sistema no agrega la oferta de trabajo y muestra un mensaje de error	Acceder al menú de enlace Ofertas de Trabajos, click en añadir ofertas de trabajo, rellenar los datos con algunos caracteres extraños y aceptar

Tabla 8: Prueba de aceptación 2

Caso de prueba 2: RF2_ Listar Oferta de trabajo							
Condiciones de ejecución: El usuario debe estar autenticado en el sistema con el rol de administrador							
Escenario	Descripción	Nombre del trabajo	Descripción de la Oferta de trabajo	Nombre del administrador	Contacto del admin	Respuesta del Sistema	Flujo Central
EC 1.1 Listar una oferta de trabajo al sistema de forma correcta	Se muestra una vista con todas las ofertas de trabajo listadas	Programador en el centro de desarrollo Desowft	!!!!Es un trabajo honrado!!!!	Pérez	56358722	Se muestra la oferta de trabajo en una vista de forma correcta	Acceder al menú de enlace Ofertas de Trabajos, click en listar ofertas de trabajo, seleccionar la oferta de trabajo y aceptar

Tabla 9: Prueba de aceptación 3

Caso de prueba 3: RF3_ Actualizar Oferta de trabajo							
Condiciones de ejecución: El usuario debe estar autenticado en el sistema con el rol de administrador							
Escenario	Descripción	Nombre del trabajo	Descripción de la Oferta de trabajo	Nombre del administrador	Contacto del admin	Respuesta del Sistema	Flujo Central
EC 1.1 Actualizar una oferta de trabajo en el sistema de forma correcta	Se muestra una vista con todas las ofertas de trabajo actualizadas	Programador en el centro de desarrollo Desowft	!!!!Es un trabajo honrado!!!!	Pérez	56358722	Se muestra en una vista las ofertas de trabajo actualizada	Acceder al menú de enlace Ofertas de Trabajos, click en actualizar ofertas de trabajo, seleccionar la oferta de trabajo, actualizar campos y aceptar

3.3.2 Pruebas de Regresión

Las pruebas de regresión son pruebas de un programa previamente probado que ha sufrido modificaciones, para asegurarse que no se han introducido o descubierto defectos en áreas del software que no han sido modificadas como resultado de los cambios realizados. Se realiza cuando el software o su entorno han sido modificados (50).

En el caso de la aplicación se realizaron las pruebas de regresión descritas a la HU Gestionar Ofertas de Trabajo y se arrojaron distintos resultados en tres distintas iteraciones:

Tabla 10: Especificación de Pruebas de Regresión.

Iteración	Error a Resolver	Problemas Encontrados
Iteración 1	No era posible añadir una oferta de trabajo	Al eliminar una oferta de trabajo no se eliminaba el identificador correspondiente a la oferta de trabajo eliminada.
Iteración 2	Se corrigió el error de la Iteración 1	La interfaz de eliminación de ofertas de trabajo mostraba errores en el <i>responsive design</i>
Iteración 3	Se corrigió el error de la Iteración 2	No se encontraron problemas

3.3.3 Pruebas de Usabilidad

Las pruebas de usabilidad son un servicio de aseguramiento de calidad que consiste en probar un producto y detectar no conformidades sobre su facilidad de uso. El objetivo principal de estas pruebas es identificar los problemas de usabilidad, para mejorar el producto y la satisfacción de sus clientes. Proporcionar retroalimentación para mejorar el diseño. Valorar en qué medida se cumplen los objetivos de los usuarios y la organización y poder exponer cuantitativamente el grado de usabilidad que presenta el producto probado (51).

Las subcaracterísticas de usabilidad que define la ISO/IEC 25000 son:

- Reconocimiento apropiado: Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- Aprendibilidad: Capacidad del producto que permite al usuario aprender su aplicación.
- Operatividad: Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.

- Protección contra errores de usuario: Capacidad del sistema para proteger a los usuarios de cometer errores.
- Estética de la interfaz de usuario: Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
- Accesibilidad: Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

Tabla 11: Lista de Chequeo de Pruebas de Usabilidad.

Elementos definidos por la metodología				
No	Indicador a evaluar	Evaluación	NP	Observación
Visibilidad del sistema				
1	¿La página refleja la identidad de la empresa (logos, compañía...)?	Si		
2	¿El menú de navegación aparece en un lugar destacado?	Si		
3	¿No utiliza más de siete opciones principales en el menú de navegación?	Si		
4	¿El nombre de los enlaces es el mismo que el título de la página a la que dirige?	Si		
Lenguaje común entre sistema y usuario				
5	¿La información es de rápida lectura, y con una disposición asequible?	Si		

6	¿Utiliza un texto específico y descriptivo en los vínculos?	Si		
7	¿Si se desea incluir un enlace de correo electrónico, se muestra el correo y no el nombre de la persona?	Si		
8	¿Evita el lenguaje técnico: términos informáticos o propios de Internet?	Si		
Libertad y control por parte del usuario				
9	¿Cuándo es necesaria la descarga de un plugin, hay un enlace a la página donde obtenerlo?	No		La opción de buscar plugin se realiza de forma manual
10	¿Los usuarios pueden ordenar y filtrar los resultados?	No		No existe la implementación de un buscador.
Estética y diseño minimalista				
14	¿Los tipos y tamaños de letra son legibles y distinguibles?	Si		
15	¿Existe suficiente contraste entre el color del fondo y el del texto?	Si		
16	¿Poseen las páginas animaciones innecesarias?	Si		
Prevención de errores				

17	¿Se dan indicaciones para completar campos problemáticos?	Si		
Ayuda y documentación				
18	¿Incluye un vínculo a los datos legales en todas las páginas?	Si		

Conclusiones parciales del Capítulo

- Con el desarrollo del presente capítulo se ha demostrado que se requieren pruebas de software para señalar los errores que ocurren durante las fases de desarrollo.
- Mediante las pruebas se asegura de que las actuaciones de la aplicación sean adecuadas y de que los clientes estén satisfechos con ella.
- Cuando el producto entregado es de calidad, ayuda a ganar la confianza de los clientes.
- Las pruebas de software con una ejecución de prueba estricta aseguran un menor costo de mantenimiento.

Conclusiones Generales

- Con el desarrollo del presente documento se pudo arribar a la conclusión de que la elaboración del marco teórico referencial es de suma importancia para cualquier trabajo escrito ya que permite realizar el análisis teórico del problema. Amplía el horizonte del estudio y guía al investigador para que se centre en su problema evitando desviaciones del planteamiento original.
- La definición de la metodología de desarrollo de software, definió las pautas para el proceso ordenado y detallado de la propuesta de solución.
- La definición de las tecnología y herramientas que formaron parte del entorno de desarrollo, están acorde con los requerimientos actuales del mercado y apuntan a la soberanía tecnológica del país.
- Se confirmó que la validación de pruebas de software es una de las actividades más importantes y fundamentales en el desarrollo de un proyecto, ya que posibilita los procesos, métodos de trabajo y herramientas necesarias para garantizar la calidad de cualquier desarrollo.

Bibliografía

1. Evolución de la Seguridad Social en Cuba - Monografias.com. [online]. [Accessed 10 November 2021]. Available from: <https://www.monografias.com/trabajos106/evolucion-seguridad-social-cuba/evolucion-seguridad-social-cuba.shtml>
2. Un antes y un después de la Seguridad Social en Cuba › Cuba › Granma - Órgano oficial del PCC. [online]. [Accessed 10 November 2021]. Available from: <https://www.granma.cu/cuba/2018-06-14/un-antes-y-un-despues-de-la-seguridad-social-en-cuba-14-06-2018-23-06-32>
3. TAMAYO, Mario Tamayo y. *Metodología formal de la investigación científica*. Editorial Limusa, 1980. ISBN 978-968-18-1186-0. Google-Books-ID: RH_v8jDiHIQC
4. Repositorio IIBI UNAM: Las necesidades de información: fundamentos teóricos y métodos. [online]. [Accessed 10 November 2021]. Available from: https://ru.iibi.unam.mx/jspui/handle/IIBI_UNAM/L110
5. RAMOS-ÁLVAREZ, Orlando, GARCÍA-LEYVA, Librada and ALMAGUER-MARTÍ, Adrián. Algunas tendencias históricas de la dirección del trabajo metodológico en el nivel municipal de educación en el contexto cubano. *Maestro y Sociedad*. 30 November 2017. Vol. 14, no. 4, p. 611–622.
6. LÓPEZ DIAZ, Nolibio. LA POLÍTICA DE EMPLEO EN CUBA. [online]. [Accessed 26 November 2021]. Available from: <https://www.eumed.net/cursecon/ecolat/cu/2008/cgld.htm>
7. ¿Qué es la planificación de la fuerza de trabajo? Genesys [online]. [Accessed 26 November 2021]. Available from: <https://www.genesys.com/es-mx/definitions/what-is-workforce-planning> La planificación de la fuerza de trabajo es el proceso mediante el cual se alinean los elementos operativos y estratégicos de la fuerza de trabajo con los objetivos de la organización.
8. Seguridad Social: Who are we? [online]. [Accessed 10 November 2021]. Available from: <https://www.seg-social.es/wps/portal/wss/internet/Conocenos/QuienesSomos/29429>
9. - Directorio y organigramas - España y su administración - Inicio. [online]. [Accessed 24 November 2021]. Available from: <https://administracion.gob.es/pagFront/espanaAdmon/directorioOrganigramas/fichaUnidadOrganica.htm?idUnidOrganica=129333&origenUO=gobiernoEstado&volver=backFicha>
10. Ministerio de Trabajo y Seguridad Social (Cuba) - EcuRed. [online]. [Accessed 10 November 2021]. Available from: [https://www.ecured.cu/Ministerio_de_Trabajo_y_Seguridad_Social_\(Cuba\)](https://www.ecured.cu/Ministerio_de_Trabajo_y_Seguridad_Social_(Cuba))
11. Instituto Nacional de Seguridad Nacional. (2020). MTSS. - Buscar con Google. [online]. [Accessed 24 November 2021]. Available from: [https://www.google.com/search?q=Instituto+Nacional+de+Seguridad+Nacional.++\(2020\).+MTSS.&aq=Instituto+Nacional+de+Seguridad+Nacional.++\(2020\).+MTSS.&aqs=chrome..69i57j0i333.510j0j9&sourceid=chrome&ie=UTF-8](https://www.google.com/search?q=Instituto+Nacional+de+Seguridad+Nacional.++(2020).+MTSS.&aq=Instituto+Nacional+de+Seguridad+Nacional.++(2020).+MTSS.&aqs=chrome..69i57j0i333.510j0j9&sourceid=chrome&ie=UTF-8)

12. ENRÍQUEZ RUIZ, José Luis, FARIÁS PALACÍN, Elías and FLORES FLORES, Eder. *METODOLOGÍA DE DESARROLLO DE SOFTWARE*. 2017.
13. LEYVA-VÁZQUEZ, Maikel and FEBLES, Pedro. Métodos Ágiles en el Desarrollo de Software Para la Bioinformática. *Serie Científica*. 1 January 2008. Vol. 1.
14. Metodología de Desarrollo de Software Variación de AUP para la UCI - EcuRed. [online]. [Accessed 27 November 2021]. Available from: https://www.ecured.cu/Metodolog%C3%ADa_de_Desarrollo_de_Software_Variaci%C3%B3n_de_AUP_para_la_UCI
15. COCKBURN, Alistair. *Crystal Clear: A Human-Powered Methodology for Small Teams: A Human-Powered Methodology for Small Teams*. Pearson Education, 2004. ISBN 978-0-13-270249-2. Google-Books-ID: O_cMM5ztyMIC
16. MAHALAKSHMI, M. and SUNDARARAJAN, Dr M. *Traditional SDLC Vs Scrum Methodology – A Comparative Study*.
17. IZQUIERDO TORRES, Isabel Alejandra and PAGUAY GUEVARA, Ángel. Análisis de XP (Extreme Programming) desde el punto de vista del estándar ISO/IEC/IEEE 29119. [online]. July 2020. [Accessed 27 November 2021]. Available from: <https://oa.upm.es/64756/>
18. PILATAXI ALBA, Edwin Rolando. E-portafolio y su incidencia en los procesos de evaluación de docentes en la Unidad Educativa Ibarra, utilizando la metodología Extreme Programming. [online]. 2018. [Accessed 27 November 2021]. Available from: <http://repositorio.utn.edu.ec/handle/123456789/8125>Accepted: 2018-04-12T19:14:40Z
19. Qué es Framework - Definición, significado y ejemplos. [online]. [Accessed 24 November 2021]. Available from: <https://www.arimetrics.com/glosario-digital/framework>
20. RODRÍGUEZ, ELÍAS. ¿Qué es un framework?: Descubre todas sus ventajas. [online]. [Accessed 24 November 2021]. Available from: <https://www.seoestudios.es/blog/que-es-un-framework/>
21. RAMA ANDIKA, Jorgie. Sistem Diagnosa Penyakit Pada Ayam Berbasis Web Laravel 8 | Jurnal Informatika dan Sistem Informasi (JIFoSI). [online]. [Accessed 27 November 2021]. Available from: <http://jifosi.upnjatim.ac.id/index.php/jifosi/article/view/305>
22. CORREA, Buenaventura and Carlos José. *Herramienta gráfica para la creación de prototipos web*. 2019.
23. ACERVO LIMA. Ventajas y desventajas de JavaScript. [online]. [Accessed 27 November 2021]. Available from: <https://es.acervolima.com/2021/09/12/ventajas-y-desventajas-de-javascript/>
24. BOOCH, Grady. *El lenguaje unificado de modelado :guía del usuario*. [no date]. : 005.117 B724i2 2006Durante casi una década, el lenguaje unificado demodelado (UML) ha sido el estándar de la industria paravisualizar, especificar, construir y documentar los artefactos delos sistemas software.

25. GROUP, PostgreSQL Global Development. PostgreSQL. *PostgreSQL* [online]. 27 November 2021. [Accessed 27 November 2021]. Available from: <https://www.postgresql.org/The world's most advanced open source database>.
26. LOCKHART, Thomas. *Guía del Programador de PostgreSQL*. 2020.
27. DASCALU, Sergiu. WebStorm: Web based support tool for organization of requirements modeling - ProQuest. [online]. [Accessed 27 November 2021]. Available from: <https://www.proquest.com/openview/3ebfd66a4db0d7e020c7c21fec81bbf6/1?pq-origsite=gscholar&cbl=18750>
28. CARLIN, Nicholas, PORTER FELT, Adrienne and WAGNER, David. An Evaluation of the Google Chrome Extension Security Architecture | USENIX. [online]. [Accessed 27 November 2021]. Available from: <https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/carlini> University of California, Berkeley
29. KEFALAKIS, Nikos and SOLDATOS, John. A Visual Paradigm for IoT Solutions Development | SpringerLink. [online]. [Accessed 27 November 2021]. Available from: https://link.springer.com/chapter/10.1007/978-3-319-16546-2_4
30. PÉREZ ACOSTA, Michelle and DE LEÓN MARTÍNEZ, Claudia Milagros. Sistema informático de apoyo a los procesos “Evaluación del estudiante” y “Censo, capacidad y matrícula” en la Dirección de la Residencia Estudiantil 2 de la Universidad de las Ciencias Informáticas. [online]. June 2015. [Accessed 27 November 2021]. Available from: <https://repositorio.uci.cu/jspui/handle/123456789/7534> Accepted: 2017-09-04T18:11:51Z
31. LAZO, Alonso Toro and VALENCIA, Luis Eduardo Peláez. Ingeniería de Requisitos: de la especificación de requisitos de software al aseguramiento de la calidad. Cómo lo hacen las Mipymes desarrolladoras de software de la ciudad de Pereira. *Entre Ciencia e Ingeniería*. 2016. Vol. 10, no. 20, p. 117–123.
32. DÍAZ, Isabel and SÁNCHEZ2, Juan. *Metamorfosis: Un Marco para el Análisis de Requisitos Funcionales*. 2020. Universidad Central de Venezuela – Laboratorio TOOLS – Escuela de Computación.
33. SERNA-MONTOYA, Édgar. Estado actual de la investigación en requisitos no funcionales Ingeniería y Universidad. . June 2012.
34. GARCIA, Roman. *Métodologías Ágiles en el Desarrollo de Software*. . 2020. DSIC - Universidad Politécnica de Valencia
35. A, Oiver Andrés Pérez. Cuatro enfoques metodológicos para el desarrollo de Software RUP – MSF – XP - SCRUM. *INVENTUM*. 1 February 2011. Vol. 6, no. 10, p. 64–78. DOI 10.26620/uniminuto.inventum.6.10.2011.64-78.
36. FIDEL, José, YLLESCAS, Urquiza, MARTÍNEZ, Alfonso and IBARGUENGOITIA-G, Guadalupe. Proceso General Para el Desarrollo de la Arquitectura de Software en Metodologías Ágiles. . 1 April 2011.

37. GONZÁLEZ, Yanette Díaz and ROMERO, Yenisleidy Fernández. Patrón Modelo-Vista-Controlador. *Telemática*. 12 June 2012. Vol. 11, no. 1, p. 47–57.
38. BLANCO HERNÁNDEZ, Nancy. SISTEMAS GROUPWARE PARA EL DISEÑO DE DIAGRAMA DE CLASES UML EN AMBIENTES TÁCTILES. [online]. [Accessed 27 November 2021]. Available from: <http://www.itc.mx/ojs/index.php/pistas/article/view/1081>
39. BARKER, Richard. *El modelo entidad-relación CASE*methodtm*. Ediciones Díaz de Santos, 1994. ISBN 978-0-201-60111-4. Google-Books-ID: hbOTo05ddxAC
40. Artefacto: Modelo de despliegue. [online]. [Accessed 27 November 2021]. Available from: https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/workproducts/rup_deploymen_t_model_57DF1DF5.html
41. Diagrama de despliegue - EcuRed. [online]. [Accessed 27 November 2021]. Available from: https://www.ecured.cu/Diagrama_de_despliegue
42. CILLERO, Manuel. Diagrama de Despliegue. *manuel.cillero.es* [online]. [Accessed 27 November 2021]. Available from: <https://manuel.cillero.es/doc/metrica-3/tecnicas/diagrama-de-despliegue/>
43. MOSQUERA, Nilson Machado. Estándares de Codificación. *Scribd* [online]. [Accessed 27 November 2021]. Available from: <https://es.scribd.com/document/410185051/ESTANDARES-DE-CODIFICACION-docx>
44. TAPIA ARPASI, Mirian Noemi. Sistema de administración y evaluación integral online para el proceso de contrataciones en instituciones públicas y privadas. *Universidad Nacional del Altiplano* [online]. 16 October 2019. [Accessed 27 November 2021]. Available from: <http://repositorio.unap.edu.pe/handle/UNAP/12412>Accepted: 2019-12-17T17:54:38Z
45. FERNÁNDEZ SANZ, Luis. Un sondeo sobre la práctica actual de pruebas de software en España. . December 2005. Vol. REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software, 1, núm. 2, pp. 43–54. Asociación de Técnicos de InformáticaMadrid, España
46. J. J. GUTIÉRREZ, M. J. ESCALONA and MEJÍAS, J. TORRES. PRUEBAS DEL SISTEMA EN PROGRAMACIÓN EXTREMA. . Vol. Department de Lenguajes y Sistemas Informáticos University of Sevilla.
47. SUAZA, Katerine Villamizar, GARCÍA, John Jairo Tabares and JARAMILLO, Carlos Mario Zapata. Mejora de historias de usuario y casos de prueba de metodologías ágiles con base en TDD. *Cuaderno Activa*. 2015. Vol. 7, p. 41–53.
48. ROMAN, Adam and MNICH, Michal. Test-driven development with mutation testing – an experimental study. *Software Quality Journal*. 1 March 2021. Vol. 29, no. 1, p. 1–38. DOI 10.1007/s11219-020-09534-x.

49. GARRIDO TEJERO, Antonio. Pruebas de caja negra. Técnica de partición equivalente. [online]. 1 July 2016. [Accessed 27 November 2021]. Available from: <https://riunet.upv.es/handle/10251/66886>Accepted: 2016-07-01T08:55:18Z
50. BORIO, Juan Ignacio and PATERNO, Ricardo Javier. *Automatización de pruebas de regresión* [online]. Tesis. Universidad Nacional de La Plata, 2021. [Accessed 27 November 2021]. Available from: <http://sedici.unlp.edu.ar/handle/10915/119477>
51. FLORIÁN, BEATRIZ E, SOLARTE, OSWALDO and REYES, JAVIER M. PROPUESTA PARA INCORPORAR EVALUACIÓN Y PRUEBAS DE USABILIDAD DENTRO DE UN PROCESO DE DESARROLLO DE SOFTWARE. *Escuela de Ingeniería de Antioquia Envigado, Colombia*. 13 July 2010.

Anexos

Anexo 1: Historias de Usuario

Tabla 12: HU3- Gestionar Solicitud de ayuda.

Historia de Usuario	
Número: 3	Nombre: Gestionar Solicitud de Ayuda
Usuarios: Administrador	
Prioridad en el negocio: Alta	Nivel de complejidad: Alta
Estimación: 4 días	Iteración Asignada: 2
Descripción: El administrador gestiona todas las solicitudes de ayuda	
Observaciones: El administrador se encargará de adicionar, listar, modificar y eliminar una solicitud de ayuda	

Tabla 13: HU4- Gestionar Ayuda Solicitada.

Historia de Usuario	
Número: 4	Nombre: Gestionar Ayuda Solicitada
Usuarios: Administrador	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Estimación: 3 días	Iteración Asignada: 2
Descripción: El administrador gestiona todas las ayudas existentes en el sistema	
Observaciones: El administrador se encargará de adicionar, listar, modificar y eliminar una ayuda solicitada	

Tabla 14: HU5-Gestionar Usuario.

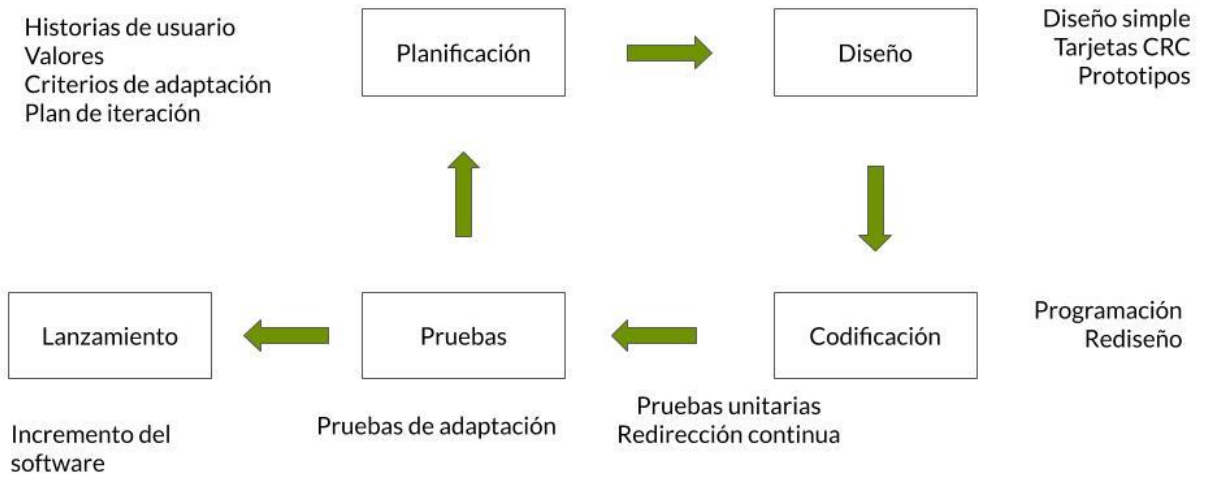
Historia de Usuario	
Número: 5	Nombre: Gestionar Usuario
Usuarios: Administrador	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Estimación: 4 días	Iteración Asignada: 3
Descripción: El administrador gestiona todo lo referente al trabajo con los usuarios	
Observaciones: El administrador se encargará de adicionar, listar, modificar y eliminar usuarios	

Tabla 15: HU6-Gestionar Departamento.

Historia de Usuario	
Número: 6	Nombre: Gestionar Departamento
Usuarios: Administrador	
Prioridad en el negocio: Alta	Nivel de complejidad: Alto
Estimación: 3 días	Iteración Asignada: 3
Descripción: El administrador gestiona todos los departamentos del sistema	
Observaciones: El administrador se encargará de adicionar, listar, modificar y eliminar un departamento	

Anexo 2 Representación de la Metodología XP

Metodología XP o Programación Extrema



Anexo 3 Representación del Modelo Vista Controlador.

