|  |  |
| --- | --- |
| Resultado de imagen para ingenieria de requerimientos  Especificación de requerimientos de software  Pos-IT | Descripción breve  PoS-it, el punto de venta ideal para tu negocio  Eduardo Canales C.  ADLS-2023 |

**ÍNDICE**

Contenido

[1. INTRODUCCIÓN 3](#_Toc153225439)

[2. DEFINICIÓN DE LA EMPRESA O INSTITUCIÓN 3](#_Toc153225440)

[**2.1.** **Descripción de la empresa** 3](#_Toc153225441)

[**2.2.** **Descripción del área de estudio** 3](#_Toc153225442)

[**2.3.** **Descripción de la problemática** 3](#_Toc153225443)

[3. DEFINICIÓN PROYECTO 4](#_Toc153225444)

[**3.1.** **Objetivos del proyecto** 4](#_Toc153225445)

[4. ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE 4](#_Toc153225446)

[**4.1.** **Objetivos del proyecto. Implementación: Fase de Desarrollo** 4](#_Toc153225447)

[**4.2.** **Objetivos del proyecto** 4](#_Toc153225448)

[**4.3.** **Objetivos del proyecto** 5](#_Toc153225449)

[**4.4.** **Objetivos del proyecto** 5](#_Toc153225450)

[**4.5.** **Objetivos del proyecto** 5](#_Toc153225451)

[5. DISEÑO CONCEPTUAL 6](#_Toc153225452)

[**5.1.** **Modelo de datos Conceptual (MER)** 6](#_Toc153225453)

[6. DISEÑO LÓGICO 7](#_Toc153225454)

[**6.1.** **Modelo Relacional** 7](#_Toc153225455)

[**6.2.** **Validación del modelo utilizando Normalización** 8](#_Toc153225456)

[**6.2.1.** **Primera Forma Normal (1FN)** 8](#_Toc153225457)

[**6.2.2.** **Segunda Forma Normal (2FN)** 8](#_Toc153225458)

[**6.2.3.** **Tercera Forma Normal (3FN)** 9](#_Toc153225459)

[7. DISEÑO FÍSICO 9](#_Toc153225460)

[**7.1.** **Recursos** 9](#_Toc153225461)

[**7.1.1.** **Hardware** 9](#_Toc153225462)

[**7.1.2.** **Software** 9](#_Toc153225463)

[**7.2.** **Tablas y demás estructuras** 9](#_Toc153225464)

[**7.2.1.** **Tablas** 10](#_Toc153225465)

[8. CONCLUSIONES 10](#_Toc153225466)

**HOJA DE CONTROL**

|  |  |  |  |
| --- | --- | --- | --- |
| Organización | LLJM Solutions | | |
| Proyecto | Pos-IT | | |
| Entregable | Especificación de Requerimientos | | |
| Contacto | +56 9 3632 0816 | | |
| e-mail | lbarra22@alumnos.utalca.cl | | |
| Cargo | Director Ejecutivo | | |
| Versión/Edición | 0001 | Fecha Versión | 05/12/2023 |
| Aprobado por | José Cáceres | Fecha Aprobación | 05/12/2023 |

REGISTRO DE CAMBIOS

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Causa del Cambio | Responsable del Cambio | Fecha del Cambio |
| 0001 | Versión inicial | Lucas Barra Roa | 05/12/2023 |
| 0002 | Versión Parcial | Leonardo Cerda Gaete | 07/12/2023 |
| 0003 | Versión Parcial | José Cáceres Aravena | 08/12/2023 |
| 0004 | versión parcial | Martin Carreño Vargas | 12/12/2023 |

CONTROL DE DISTRIBUCIÓN

|  |
| --- |
| Nombre y Apellidos |
| Lucas Barra Roa |
| José Cáceres Aravena |
| Leonardo Cerda Gaete |
| Martin Carreño Vargas |

INTEGRANTES DEL GRUPO DE PROYECTO

|  |  |
| --- | --- |
| Apellidos Nombres | Correo electrónico |
| Barra Roa Lucas | lbarra22@alumnos.utalca.cl |
| Cerda Gaete Leonardo | jocaceres22@alumnos.utalca.cl |
| Cáceres Aravena José | lcerda21@alumnos.utalca.cl |
| Carreño Vargas Martín | macarreno22@alumnos.utalca.cl |

1. **INTRODUCCIÓN**

Este documento presenta la versión inicial de la Especificación de Requerimientos de Software para el proyecto de PoS-IT, siendo este nuestro enfoque a un Punto de Venta. El objetivo principal de esta ERS es definir de manera clara y detallada los requisitos funcionales y no funcionales del software que se desarrollará para el sistema de PoS-IT. Estos requisitos se basan en las necesidades y expectativas del cliente, así como en las mejores prácticas y estándares de la industria. La ERS servirá como guía fundamental para el equipo de desarrollo, proporcionando una visión general de los objetivos del proyecto y estableciendo las bases para el diseño, implementación y prueba del software. A lo largo de este documento, se describirán los diferentes módulos y funcionalidades del sistema de PoS-IT. Este documento está sujeto a cambio con distintas versiones que serán desarrolladas en la transversalidad del avance de nuestro proyecto.

1. **DEFINICIÓN DE LA EMPRESA O INSTITUCIÓN**
   1. **Descripción de la empresa**

Los antecedentes generales de la empresa son:

Nombre: LLJM Solutions.

Dirección: 3466896 Talca, Maule.

Rubro: Desarrollo de software, IT.

* 1. **Descripción del área de estudio**

LLJM Solutions es una empresa emergente fundada a principios de 2023, especializada en el desarrollo de software. Nuestro enfoque se basa en proporcionar soluciones tecnológicas innovadoras y de alta calidad para satisfacer las necesidades de nuestros clientes. Basada en Talca, Chile, y una distribución remota en ciudades como Linares, Pichilemu y Constitución, nos hemos posicionado estratégicamente para atender a una amplia gama de clientes en la región.

* 1. **Descripción de la problemática**

En la actualidad, los softwares de Punto de Venta presentan desafíos para el dueño de tienda de abarrotes, tales como implementaciones complicadas, altos costos, interfaces poco amigables y falta de funciones o módulos necesarios. Estos problemas dificultan su adopción y uso eficiente en los negocios. Es fundamental desarrollar soluciones de PoS que sean fáciles de implementar, asequibles, con interfaces amigables y que ofrezcan una amplia funcionalidad para satisfacer las necesidades comerciales. Además, es importante que estas soluciones sean amigables para adultos mayores y accesibles, teniendo en cuenta la diversidad de usuarios que puedan interactuar con ellas.

1. **DEFINICIÓN PROYECTO**
   1. **Objetivos del proyecto**

El proyecto Pos-it tiene como objetivo desarrollar un software de Punto de Venta (PoS) eficiente y amigable para tiendas de abarrotes. Se busca crear una solución fácil de implementar, con una interfaz intuitiva y amigable, funcionalidades completas y personalizables, y accesibilidad para adultos mayores. Además, se ofrecerá un plan de soporte continuo y actualizaciones regulares para garantizar la estabilidad y seguridad del sistema a lo largo del tiempo. El proyecto busca satisfacer las necesidades de las tiendas de abarrotes, brindando una herramienta eficiente y fácil de usar que optimice las operaciones comerciales y mejore la experiencia del usuario.

En resumen, el proyecto Pos-it busca desarrollar un software de PoS completo y accesible, con una interfaz amigable y funcionalidades personalizables, para tiendas de abarrotes. Se prioriza la simplicidad en la implementación, la usabilidad para adultos mayores y un soporte continuo. El objetivo es proporcionar una solución eficiente que mejore las operaciones comerciales y la experiencia del usuario.

1. **ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE**
   1. **Objetivos del proyecto. Implementación: Fase de Desarrollo**

Después de establecer las bases en el Proyecto Blastoff, se da inicio a la fase de desarrollo del software PoS-IT. En esta etapa crucial, se llevan a cabo las siguientes actividades:

* **Desarrollo Iterativo:**
  + Adoptamos un enfoque iterativo para construir el software, permitiendo ajustes continuos según las necesidades emergentes.
* **Integración de Funcionalidades:**
  + Se incorporan las funcionalidades esenciales identificadas en la fase de captura de requisitos.
* **Pruebas Continuas:**
  + Realizamos pruebas continuas para garantizar la calidad y la conformidad con los requisitos.
  1. **Objetivos del proyecto**

**Refinamiento del Diseño de la Interfaz de Usuario**

Concomitantemente al desarrollo, perfeccionamos el diseño de la interfaz de usuario (UI) para asegurar una experiencia intuitiva y eficiente. Aquí se llevan a cabo las siguientes acciones:

* **Iteraciones de Maquetas:**
  + Se realizan iteraciones en las maquetas del sistema en colaboración con diseñadores de experiencia de usuario (UX).
* **Retroalimentación de Stakeholders:**
  + Presentamos el diseño propuesto a los stakeholders para obtener retroalimentación sobre usabilidad y diseño.
  1. **Objetivos del proyecto**

**Evaluación de Avances y Ajustes**

Durante el desarrollo, realizamos una evaluación constante de los avances y hacemos ajustes según sea necesario:

* **Revisiones Periódicas:**
  + Se llevan a cabo revisiones periódicas para evaluar el progreso del desarrollo y la alineación con los objetivos del cliente.
* **Ajustes Ágiles:**
  + Adoptamos una metodología ágil para realizar ajustes ágiles según los cambios en los requisitos o las preferencias del cliente.
  1. **Objetivos del proyecto**

**Prototipos Funcionales**

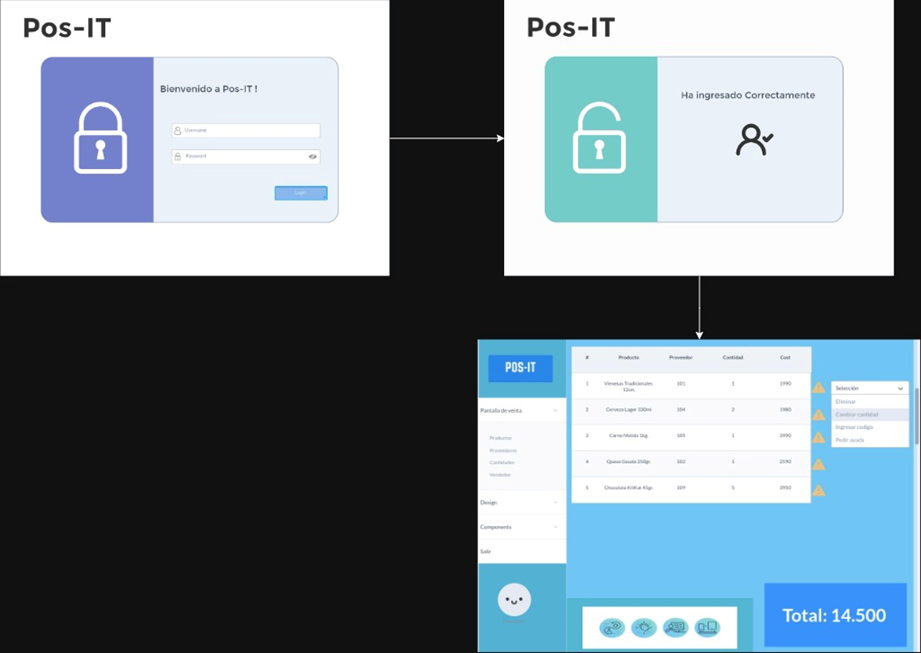
Como parte integral del desarrollo, creamos prototipos funcionales para proporcionar una representación práctica del software en evolución:

* **Validación de Funcionalidades:**
  + Utilizamos prototipos funcionales para validar las funcionalidades clave con los usuarios finales.
* **Refinamiento Iterativo:**
  + La retroalimentación de los usuarios se integra en iteraciones posteriores para un refinamiento continuo.
  1. **Objetivos del proyecto**

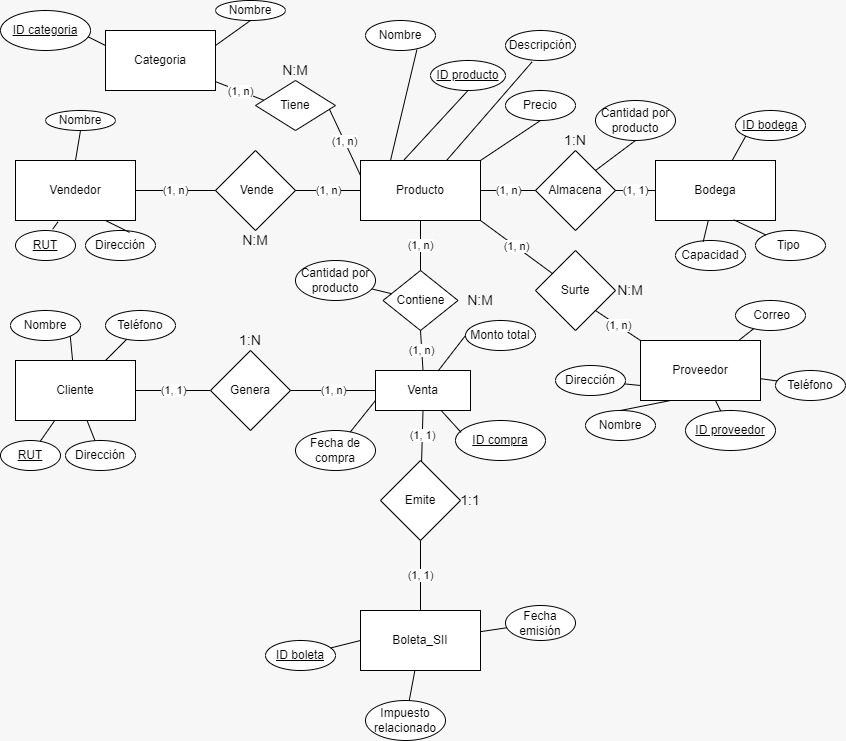
**Documentación del Proceso de Desarrollo**

Se documenta detalladamente el proceso de desarrollo, incluyendo cambios significativos, decisiones clave y lecciones aprendidas. Esta documentación sirve como referencia y guía para futuras fases del proyecto y para el mantenimiento del software.

Mockup:



1. **DISEÑO CONCEPTUAL**
   1. **Modelo de datos Conceptual (MER)**



MER Pos-IT

En este modelo conceptual (Modelo Entidad Relación) se representa de forma gráfica la estructura de nuestra base de datos y sus distintas relaciones:

*VENDEDOR: (RUT, Nombre, Dirección)*

*CLIENTE: (RUT, Teléfono, Nombre, Dirección)*

*PRODUCTO: (ID producto, Nombre, Descripción, Precio)*

*BODEGA: (ID bodega, Tipo, Capacidad)*

*PROVEEDOR: (ID proveedor, Nombre Empresa, Teléfono, Dirección, Correo)*

*COMPRA: (ID de compra, Monto total, Fecha de compra)*

*BOLETA SII: (ID de boleta, Fecha emisión, Impuesto relacionado)*

*CATEGORIA:(ID categoría, nombre)*

Gracias a este modelo se pueden trabajar más fácilmente con las bases de datos, optimizando de la mejor forma el tiempo de desarrollo.

1. **DISEÑO LÓGICO**
   1. **Modelo Relacional**

|  |  |  |
| --- | --- | --- |
| Tablas. | Atributos. | Restricción. |
| [bodegas](http://localhost/phpmyadmin/index.php?route=/sql&table=bodegas) | id(PK), nombre, capacidad | La clave primaria (id) garantiza la unicidad de cada registro en la tabla. |
| [boletassii](http://localhost/phpmyadmin/index.php?route=/sql&table=boletassii) | id(PK), importeImpuesto, fechaHoraCompra, idVenta(FK) | La clave primaria (id) garantiza la unicidad de cada registro en la tabla.  La clave foránea (idVenta) establece una relación con la tabla Venta. |
| [clientes](http://localhost/phpmyadmin/index.php?route=/sql&table=clientes) | rut(PK), nombre, domicilio, celular, correo, activo | La clave primaria (rut) garantiza la unicidad de cada cliente. |
| [existenciasbodega](http://localhost/phpmyadmin/index.php?route=/sql&table=existenciasbodega) | idProducto(FK), idBodega(FK), cantidad, surtido, notas | Las claves foráneas (idProducto y idBodega) establecen relaciones con las tablas Producto y Bodegas, respectivamente. |
| [producto](http://localhost/phpmyadmin/index.php?route=/sql&table=producto) | id(PK), nombre, precio, descripcion, idProveedor(FK) | La clave primaria (id) garantiza la unicidad de cada producto.  La clave foránea (idProveedor) establece una relación con la tabla Proveedores. |
| [productosenventa](http://localhost/phpmyadmin/index.php?route=/sql&table=productosenventa) | idProducto(FK), idVenta(FK), cantidad, | Las claves foráneas (idProducto e idVenta) establecen relaciones con las tablas Producto y Venta, respectivamente. |
| [proveedores](http://localhost/phpmyadmin/index.php?route=/sql&table=proveedores) | id(PK), nombreEmpresa, domicilio, celular, correo, notas, activo | La clave primaria (id) garantiza la unicidad de cada proveedor. |
| [vendedores](http://localhost/phpmyadmin/index.php?route=/sql&table=vendedores) | rut(PK), nombre, domicilio, celular, correo, login, clave, activo | La clave primaria (rut) garantiza la unicidad de cada vendedor. |
| [venta](http://localhost/phpmyadmin/index.php?route=/sql&table=venta) | id(PK), monto, fecha, rutCliente(FK), rutVendedor(FK) | La clave primaria (id) garantiza la unicidad de cada venta.  Las claves foráneas (rutCliente y rutVendedor) establecen relaciones con las tablas Clientes y Vendedores. |
| categoria | id(PK), nombre | La clave primaria (id) garantiza la unicidad de las respectivas categorías. |
| categoriaproducto | idProducto(FK), idCategoria(FK) | Las claves foráneas (idProducto, idCategoria) establecen relaciones entre las tablas categoría y producto. |

* 1. **Validación del modelo utilizando Normalización**

Nuestro objetivo es almacenar información de manera eficiente, evitando redundancias innecesarias y facilitando la recuperación de datos. En este contexto, aplicaremos la normalización como técnica principal para diseñar esquemas que cumplan con las formas normales adecuadas

Por ejemplo, al considerar la tabla 'ExistenciasBodega', aplicaremos la normalización para asegurar que los atributos como 'cantidad', 'surtido' y 'notas' estén adecuadamente relacionados con la clave primaria compuesta 'idProducto' e 'idBodega'. De esta manera, garantizamos una representación eficiente de las existencias en cada bodega, minimizando redundancias y asegurando la consistencia del modelo.

Este enfoque se replicará en otras tablas, como 'Venta', donde la normalización nos permitirá organizar los atributos de manera que mantenga la integridad de la información asociada a cada venta, evitando duplicidad innecesaria y facilitando la gestión de datos a lo largo del tiempo.

Nuestro diseño de base de datos se orienta hacia la eficiencia y consistencia, garantizando que cada tabla esté organizada de acuerdo con las formas normales, lo que optimiza la recuperación de información y facilita la adaptación a futuros cambios en los requisitos del sistema.

* + 1. **Primera Forma Normal (1FN)**

La Primera Forma Normal requiere que cada celda en cada tabla contenga un solo valor, evitando la repetición de grupos de valores.

Todas nuestras tablas cumplen con la 1FN, ya que no hay celdas que contengan múltiples valores o contenido separados por comas

* + 1. **Segunda Forma Normal (2FN)**

La Segunda Forma Normal exige que cada tabla esté en 1FN y que todos los atributos no clave dependan completamente de la clave primaria.

La tabla ExistenciasBodega tiene una clave primaria compuesta por dos atributos: idProducto y idBodega. Además, los atributos cantidad, surtido y notas dependen completamente de la clave primaria, al igual que las demástablas. Por lo tanto, la tabla está en 2FN.

* + 1. **Tercera Forma Normal (3FN)**

La Tercera Forma Normal requiere que una tabla esté en 2FN y que los atributos no clave no dependan transitivamente de la clave primaria.

Nuestras tablas cumplen con la 3FN, ya que no hay dependencias transitivas evidentes en los atributos no clave.

1. **DISEÑO FÍSICO**

* 1. **Recursos**
     1. **Hardware**

El hardware a utilizar para llevar a cabo el presente Proyecto es el siguiente:

Computador con un procesador Pentium Gold 4417U, de 2,3 Ghz (2 MB de caché), 8GB de memoria RAM, Disco Duro 500 GB, 8 MB de memoria de Video.

* + 1. **Software**

Sistema Operativo Unbuntu 18.04.

Herramienta de desarrollo Java SE 6.

Motor de Base de Datos MYSQL 5.0.

* 1. **Tablas y demás estructuras**
     1. **Tablas**

1. **Tabla Cliente**

**Tabla

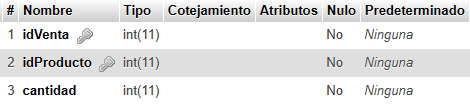
Descripción generada automáticamente**

1. **Tabla Producto**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

1. **Tabla Productosenventa**

****

1. **Interfaz de usuario gráfica, Aplicación

   Descripción generada automáticamenteTabla bodegas**
2. **Tabla existenciasbodega**

**Tabla

Descripción generada automáticamente**

1. **Tabla categoría**
2. **Tabla categoriaproducto**

**Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media**

1. **Tabla venta**

**Tabla

Descripción generada automáticamente**

1. **Tabla boletassii**

**Imagen que contiene Patrón de fondo

Descripción generada automáticamente**

1. **Tabla vendedores**

**Tabla

Descripción generada automáticamente**

1. **Tabla proveedores**

**Tabla

Descripción generada automáticamente con confianza baja**

1. **CONCLUSIONES**

El presente Trabajo ha detallado de manera completa la Especificación de Requerimientos de Software para el proyecto Pos-IT de LLJM Solutions, que se enfoca en la creación de un sistema de Punto de Venta (PoS) diseñado específicamente para tiendas de abarrotes. Hemos recopilado información y problemas varios que tienen las tiendas en torno a sus puntos de ventas y uno de los objetivos fundamentales de PoS-IT es superar los desafíos comunes que enfrentan estas tiendas, como implementaciones complicadas, altos costos, interfaces poco amigables y la carencia de funciones esenciales en los sistemas actuales de PoS.

En resumen, el trabajo establece una base sólida para el desarrollo del proyecto Pos-IT, proporcionando una visión integral desde la definición de la empresa hasta los aspectos técnicos del diseño de software y la implementación práctica. La atención a detalles como la normalización y las formas normales en el diseño de la base de datos demuestra un enfoque riguroso para garantizar la eficiencia y la consistencia en el manejo de la información.