**UNIVERSIDAD NACIONAL DEL ESTE**

**FACULTAD POLITÉCNICA**

**Proyecto interdisciplinario**

***GESTION DE PEDIDOS***



**ASIGNATURAS:**

**BASE DE DATOS II**

Prof. Aldo Ariel Gómez Ortega

**LENGUAJE DE PROGRAMACION II**

Prof. Luis González

**CIUDAD DEL ESTE - ALTO PARANÁ - PARAGUAY**

**NDICE**

1. [CAPITULO I 3](#_Toc478398924)

[CONTEXTO 4](#_Toc478398925)

[PLANTEAMIENTO DEL PROBLEMA 4](#_Toc478398926)

1. [OBJETIVOS 4](#_Toc478398927)

[OBJETIVO GENERAL 4](#_Toc478398928)

[OBJETIVOS ESPECÍFICOS 5](#_Toc478398929)

1. [CAPITULO II 6](#_Toc478398930)

[PROPÓSITO 7](#_Toc478398931)

[FUNCIONES DEL SISTEMA 7](#_Toc478398932)

[CARACTERÍSTICAS DE LOS USUARIOS 7](#_Toc478398933)

[REQUISITOS 8](#_Toc478398934)

[PROCESO DE NEGOCIO 8](#_Toc478398935)

[REQUISITOS FUNCIONALES 9](#_Toc478398936)

[REQUISITOS NO FUNCIONALES 10](#_Toc478398937)

1. [CAPITULO III 11](#_Toc478398938)

[MODELO LÓGICO DEL SISTEMA 13](#_Toc478398939)

[DIAGRAMA DE ENTIDAD RELACIONAMIENTO 13](#_Toc478398940)

[DIAGRAMA DE CASOS DE USO 16](#_Toc478398941)

1. [CAPITULO IV 19](#_Toc478398942)

[DISEÑO FÍSICO DE LA BASE DE DATOS 20](#_Toc478398943)

[CONFIGURACIÓN DE LA CONEXIÓN A LA BASE DE DATOS 20](#_Toc478398944)

[PERSISTENCIA Y MAPEO OBJETO-RELACIONAL 20](#_Toc478398945)

1. [CAPITULO V 22](#_Toc478398946)

# CAPITULO I

INTRODUCCION

## CONTEXTO

El presente proyecto se plantea el desarrollo de una aplicación que permita agilizar y optimizar las operaciones más significativas llevadas a cabo en la operación de una distribuidora de productos electrodomésticos.

El sistema será un sistema estándar, que contará con las funcionalidades básicas de una distribuidora, específicamente del módulo de pedidos y gestión de existencia de productos.

## PLANTEAMIENTO DEL PROBLEMA

En las empresas que no implementan una solución informatizada para la gestión de la existencia de productos y el mantenimiento de sus pedidos el proceso es generalmente tedioso y engorroso debido a la constante consulta de manera manual o verbal de la existencia de mercaderías o de los datos del cliente en lo que refiere por citar un ejemplo, la dirección de entrega de los pedidos.

# OBJETIVOS

Para dar respuesta a la problemática planteada en el apartado anterior, se han fijado los siguientes objetivos:

## OBJETIVO GENERAL

Desarrollar e implementar una aplicación, , que esté diseñada para gestionar la existencia de artículos y controlar los pedidos que se realizan en una empresa distribuidora de productos electrodomésticos.

## OBJETIVOS ESPECÍFICOS

* ~~.~~

# CAPITULO II

ESPECIFICACIÓN DE REQUISITOS

**ESPECIFICACIÓN DE REQUISITOS**

## PROPÓSITO

La presente especificación de requisitos de software surge con el objetivo primordial de plasmar una descripción detallada del comportamiento del sistema que se desea construir, permitiendo de esta manera contar con un documento que sirva como guía de apoyo al equipo encargado del desarrollo del software, así como también a todas las personas que harán uso del sistema una vez que haya sido terminada.

El propósito del entregable es proporcionar a las empresas dedicadas al rubro de distribución de electrodomésticos un aplicativo que los módulos de gestión y existencia de productos y mantenimiento de productos.

## FUNCIONES DEL SISTEMA

La aplicación a desarrollar permitirá la búsqueda, creación, borrado y actualización dinámica de los datos.

De igual manera, posibilitará generar informes acerca de los históricos de movimientos de pedidos según rangos a establecer; los cuales podrán ser guardados por el usuario en formato PDF O XLS o ser impresos directamente a través del diálogo de pantalla de la impresora predeterminada.

## CARACTERÍSTICAS DE LOS USUARIOS

Con el fin de mantener la seguridad de la información, se han establecido dos perfiles de usuario para acceder a la aplicación: administrador, y vendedor; en el que cada perfil tiene privilegios y permisos distintos de los demás. Dichas características y niveles de accesos se detallan a continuación:

**Administrador:** tendrá acceso y control total sobre todas las funciones del sistema, incluyendo el registro de una nueva cuenta de usuario, así como la alteración del tipo de usuario y del estado de un personal.

**Vendedor:** sólo tendrá el privilegio de acceder al formulario de registro de pedidos.

## REQUISITOS

En base a las necesidades de los usuarios, y teniendo en cuenta el objetivo general del trabajo, en este apartado se definen los requerimientos funcionales y no funcionales que deberán ser satisfechos por el sistema, los cuales se especifican a continuación:

## PROCESO DE NEGOCIO

Las reglas básicas del proceso de negocio de la aplicación a desarrollar son:

**Registro de Usuario**

En este proceso se procederá al registro de usuarios que utilizarán el sistema donde se seleccionarán el rol que van a desempeñar dentro del sistema.

**Registro de Cliente**

En este proceso se procederá al registro de clientes con sus datos necesarios para su intervención en el módulo de pedidos, se solicitará el nombre, números de teléfono y documento y una dirección de correo.

**Registrar Producto**

Se procederá a la creación de productos cuyos parámetros para su realización comprende definir un código identifica torio único según el formato establecido, la descripción, la selección de categoría u subcategoría a la que pertenece y la existencia inicial.

**Mantener Pedidos**

Modulo principal de la aplicación en la que se registran los pedidos con status inicial de presupuesto, siendo necesario los datos de la cabecera: cliente, vendedor y forma de pago.

También se solicita la selección de al menos un ítem de producto en el detalle del producto para poder mantener el pedido para su posterior emisión, cancelación o envió por correo electrónico.

Al seleccionar un producto y la cantidad solicitada se exhibirá la existencia actual del mismo, de ser mayor el pedido a la existencia disponible se indicara con un alerta visual esa situación, y no se podrá pasar del status de presupuesto al de emitido, siendo necesario, ingresar más ítems del producto, eliminar de los pedidos el productos solicitados o ajustar la cantidad solicitada a la existencia.

Al ser emitido un pedido se realiza el correspondiente ajuste de la existencia de los productos con las cantidades solicitadas para cada ítem del detalle.

## REQUISITOS FUNCIONALES

Un requisito funcional define el comportamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica. Son complementados por los requisitos no funcionales, que se enfocan en cambio en el diseño o la implementación.

Como se define en la ingeniería de requisitos, los requisitos funcionales establecen los comportamientos del sistema.

A continuación, se presenta las tablas de los principales requisitos funcionales:

**CASOS DE USOS DE ALTO NIVEL**

**CASOS DE USOS EXPANDIDO**

**DIAGRAMAS DE CASOS DE USOS**

## REQUISITOS NO FUNCIONALES

Los requisitos no funcionales a considerar en el desarrollo de la aplicación son:

* **RNF-01: Performance.** El sistema deberá tener un tiempo de respuesta inmediato o de un máximo de 2 segundos por cada requerimiento. RESTRICCIONES DE DISEÑO
* **RNF-XX: AGREGAR +**

# CAPITULO III

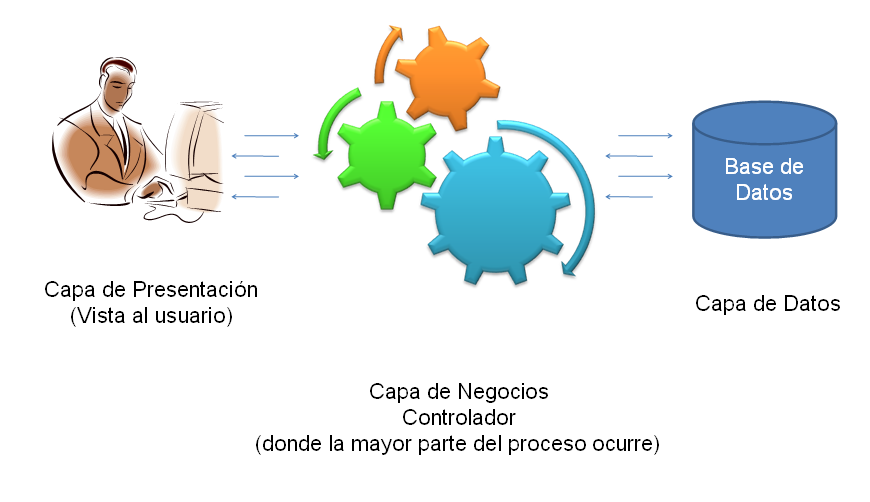
**ANÁLISIS Y DISEÑO DEL SISTEMA**

**ANÁLISIS Y DISEÑO DEL SISTEMA**

El modelo–vista–controlador (MVC) se eligió como el patrón de arquitectura de software para separar los datos y la lógica de negocio de la aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

**MODELO–VISTA–CONTROLADOR (MVC)**

La arquitectura sugiere que la separación en tres capas desarrollo (Modelo, Vista y Control), esta separación facilita el desarrollo y mantenimiento del código. Según Luckow y Melo (2010, p.180) "... el objetivo principal de la MVC no se ajusta a separarse en capas, pero si definen cómo deben interactuar capas”. Usando este concepto, el sistema desmodular, donde el intercambio de tecnología tiene ninguna interferencia importante de una capa a la otra.



La separación lógica de estas partes asegura que la capa de aplicación Modelo no sabe prácticamente nada de lo que se muestra; restringido por representar partes del problema que se resuelve por los componentes aplicación.

Del mismo modo, la capa de presentación sólo se relaciona con mostrar los datos y no aplicar la lógica de negocio que se controla la capa del Modelo. El controlador, como un gestor de tráfico, dirige las presentaciones que se mostrarán y con los cambios necesarios recuperación de datos y capa del modelo que viene.

La arquitectura MVC dice que la capa de Modelo no puede interactuar con la capa de Vista. El controlador actúa como un mediador de estas dos capas y es el único enlace de comunicación entre éstas capas. El foco importante esta arquitectura es disociar la relación entre la visión y el modelo.

## MODELO LÓGICO DEL SISTEMA

Durante el proceso de construcción de un sistema, una de las etapas más relevantes y determinantes de la estructura y calidad del software, radica en el diseño de una serie de diagramas que representen la aplicación a desarrollar. Dada esta situación, a seguir se brinda, desde diferentes enfoques, un modelo del plan a ejecutarse.

## DIAGRAMA DE ENTIDAD RELACIONAMIENTO

Atendiendo a los factores considerados recientemente, en esta primera etapa de modelado de datos se presenta, mediante el diagrama de entidad relacionamiento, un esquema conceptual de la cardinalidad de las interrelaciones existentes entre las distintas entidades del sistema.

**DIAGRAMA DE ENTIDAD RELACIONAMIENTO**

**DIAGRAMA DE CLASES**

## DIAGRAMA DE CASOS DE USO

Una de las partes más importantes en el desarrollo de una aplicación es la identificación de las necesidades de los usuarios. La mayoría de los fracasos en la construcción de un nuevo sistema de información proviene de una mala definición de los requisitos funcionales. Así, el modelado de casos de uso es una notación gráfica que describe la comunicación y el comportamiento del software a partir de la especificación de las funcionalidades que el sistema ha de ofrecer desde la perspectiva de los usuarios y lo que el sistema ha de realizar para satisfacer las peticiones de todas las partes involucradas (clientes y futuros usuarios del sistema).

Dicho en otros términos, un diagrama de casos de uso es un modelo de como distintos tipos de usuarios interactúan con la aplicación para tratar de resolver un problema determinado. Como tal, este describe las metas de los usuarios, las interacciones entre los usuarios y el sistema y el comportamiento requerido del sistema para satisfacer esas metas.

Desde el punto de vista del cliente proveen una visión de “caja negra” de la aplicación, esto es, como se deja apreciar la aplicación desde el exterior sin necesidad de entrar en los detalles de su construcción. Sin embargo, para los desarrolladores representan el punto de partida y el eje sobre el que se apoya todo el proceso de desarrollo de la aplicación en sus procesos de análisis y diseño. En virtud de todo lo expuesto, y para dar una visión gráfica de las funcionalidades del sistema a desarrollar, se presenta a diagrama:

## 

**CONCEPTUALIZACIÓN Y PROTOTIPADO**

Por más que el levantamiento de requerimientos, por sí solo, permita obtener una infinidad de informaciones que deberán estar presentes en el diseño de la interfaz de usuario, es durante la interacción real del usuario con el sistema en el que los detalles son realmente percibidos. Así, para poder proyectar una experiencia perfecta, la utilización de prototipos representa la forma más eficaz de simular esta interacción, debiendo ser esquematizada en la etapa de concepción del sistema, evitando duplicidad de esfuerzos y pérdida de tiempo y dinero.

Un prototipo es una versión inicial de un sistema de software utilizado para demostrar los conceptos, la experiencia de opciones de diseño y encontrar más información sobre el problema y las posibles soluciones. (Sommerville, 2011, p. 31).

Específicamente, su principal objetivo consiste en centrarse en la funcionalidad, la experiencia de usuario y la prioridad de los contenidos de la aplicación. Todo esto hace que, a menudo, los usuarios comprueben que su visión inicial fue incorrecta o incompleta, produciéndose cambios significativos en la estructura y navegabilidad del software, los cuales minimizarán el riesgo y la incertidumbre durante el proceso de codificación del sistema [2].

Con todo lo expuesto hasta ahora, queda claro el hecho de que el prototipado (wireframing) es una de las actividades preliminares más importantes y de mayor impacto futuro dentro de las primeras fases de construcción de un software, puesto que permite validar de forma temprana el alcance y la usabilidad del proyecto, permitiendo corregir las deficiencias que presenten y de esta manera, asegurar la alta calidad del producto final.

Teniendo en cuenta el papel fundamental que desempeña el prototipado en la arquitectura de la información de un sistema informático, para el presente proyecto se ha elaborado el diseño de la estructura visual interna del formulario a construir La razón por la cual no se han elaborado los prototipos correspondientes a las otras partes del software, se debe a que las demás interfaces de la aplicación, además de tener características muy similares a esta, no presentan mayores complicaciones en su comprensión e implementación.

Prototipo del formulario de Mantener Pedido.

Prototipo del formulario de Registrar Producto.

Prototipo del formulario de consultar Pedido.

Prototipo del formulario de consultar Producto.

# CAPITULO IV

**DESARROLLO E IMPLEMENTACIÓN**

**DESARROLLO E IMPLEMENTACIÓN**

Una vez que se han reunido todas las necesidades y/o requerimientos del cliente, y se han llevado a cabo minuciosos y detallados estudios sobre cada uno de ellos, en esta parte del documento se describen los procesos implicados en la construcción y puesta en marcha del sistema, los cuales requieren la realización de una serie de actividades concretas que condicionan la resolución final del problema planteado inicialmente.

## DISEÑO FÍSICO DE LA BASE DE DATOS

Partiendo del esquema lógico obtenido en el capítulo anterior, en esta primera etapa de construcción del sistema se procedió a la creación de la estructura física de la base de datos junto con sus respectivas tablas y relaciones. Para esta finalidad, tal como ha sido especificado previamente, fue necesario la descarga y posterior instalación de la versión 9.1 del [sistema de gestión de bases de datos](http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos) PostgreSQL.

Diseño físico de la base de datos.ingrese captura de pantalla

## CONFIGURACIÓN DE LA CONEXIÓN A LA BASE DE DATOS

## PERSISTENCIA Y MAPEO OBJETO-RELACIONAL

Entorno de Eclipse caprura de pantalla del IDE

Explorador de Proyectos

Pantalla de Inicio de la aplicación

# CAPITULO V

**DICCIONARIO DE DATOS**

|  |
| --- |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| **cliente** | | |
| **Columna** | **Tipo** | **Comentario** |
| ***Id*** | bigint(20) | Clave Primaria de la tabla de cliente. |
| documento\_ruc | varchar(14) | Numero de RUC del cliente |
| Email | varchar(255) | Dirección de correo del cliente |
| Nombre | varchar(100) | Nombre del cliente |
| Tipo | varchar(10) | Tipo del cliente |

Planilla de Certificación – Trabajo Interdisciplinario

|  |  |
| --- | --- |
| Proyecto : | **Integrantes** |
| Nombre del Grupo : |  |
| Email del Grupo : |  |
| Fecha de Inicio : |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  | **Firma** |
| BASE DE DATOS II | Diagrama de Entidad Relacionamiento |  |  |  |  |  |
|
| Diccionario de Datos |  |  |  |  |  |
| Copia de Seguridad (SQL-TXT) |  |  |  |  |  |
| INGENIERIA DE SOFTWARE II | Casos de usos alto nivel |  |  |  |  |  |
| Atributos del Sistema |  |  |  |  |  |
| Casos de usos expandido |  |  |  |  |  |
| Modelo Conceptual |  |  |  |  |  |
| LENGUAJE DE PROGRAMACION II |  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |