

**UNIVERSIDAD INTERNACIONAL DE LAS AMERICAS
ESCUELA DE INGENIERÍA INFORMÁTICA**

TRABAJO FINAL DE GRADUACIÓN

Para optar por el grado de Bachillerato en
Ingeniería de Sistemas de Información

**Prototipo Funcional para Centralización de Sistema de Punto de Venta para la cadena de
Minisúper La Bodeguita, ubicadas en el área de Desamparados y San Sebastián**

Gustavo Alonso Hidalgo Sánchez

AUTOR

Fabián Rodríguez Sibaja

TUTOR

MBD. Olman Núñez Peralta

LECTOR

San José, Costa Rica

AGOSTO, 2020

Dedicatoria

Dedico mi proyecto de graduación a Dios, gracias a él puedo concluir esta etapa tan importante para mí, una etapa llena de conocimiento y buenos momentos en compañía de seres irremplazables.

A mi familia, que siempre han estado para mí, con sus palabras de aliento y su apoyo incondicional durante toda mi etapa universitaria. Gracias a ellos pude seguir adelante teniendo siempre su amor y cariño en todo momento. A mi padre y a mi madre, que me motivaron a salir adelante mediante el estudio, y todos los días trabajaron fuertemente para que no me faltara ese privilegio.

Por último y muy importante, quiero dedicárselo a mi abuela que lamentablemente falleció hace unos años y siempre me motivó a estudiar, estoy seguro que estará orgullosa en el cielo.

Agradecimientos

Al concluir esta etapa tan importante, agradezco primeramente a Dios por darme salud y permitirme terminar mis estudios, muy agradecido por haber vivido tantas experiencias inolvidables durante esta etapa.

Agradezco a cada persona que fue parte de este proceso de formación, a esos compañeros que siempre brindaron su ayuda, aquellos que hicieron los momentos de la universidad únicos, y en especial a todos aquellos profesores que me inculcaron los mejores consejos y conocimientos.

Para mis padres no existen las palabras para expresar lo agradecido que estoy con ellos, siempre apoyándome, estando a mi lado en los momentos difíciles; este logro no sería posible sin la ayuda de ellos.

Contenido	5
Resumen Ejecutivo	21
CAPÍTULO I: INTRODUCCIÓN	22
Descripción de la organización	22
Descripción del problema	22
Propuesta de módulos a desarrollar	24
Módulo de inventario	24
Módulo de compras	24
Módulo de facturación	24
Módulo de reportes	24
Módulo de planilla	25
Módulo de seguridad	25
Herramientas que se utilizaran para el desarrollo	25
Objetivo general	26
Objetivos específicos	26
Justificación	26
Viabilidad técnica	27
Viabilidad operativa	28
Viabilidad económica	28
Proyecciones	28
Alcance técnico	29
Alcance metodológico	29
CAPÍTULO II: MARCO REFERENCIAL	30
CAPÍTULO III: MARCO METODOLÓGICO	34
Enfoques de la investigación	34

Enfoque de investigación cualitativo	34
Enfoque de investigación cuantitativo	35
Enfoque de investigación mixto	35
Enfoque de investigación seleccionado	36
Métodos de la investigación	36
Método de investigación descriptivo.	37
Método de investigación seleccionado.	37
Fuentes de información	37
Fuentes primarias.	37
Fuentes secundarias.	38
Fuentes terciarias.	38
Variables o unidades de análisis	38
Muestra y población	39
Instrumentos de recolección de datos	40
Proceso para la recolección de análisis de datos	41
CAPÍTULO IV: ANÁLISIS DE RESULTADOS	43
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES	49
Conclusiones	49
Recomendaciones	50
CAPÍTULO VI: PROPUESTA	52
Análisis	52
Análisis detallado del software a desarrollar.	52
Análisis detallado del hardware requerido.	53
Análisis detallado de los elementos relacionados con las telecomunicaciones.	54
Descripción detallada de base de datos.	54

Descripción detallada del personal requerido para el uso del sistema.	55
Casos de uso	55
Diseño	66
<i>Arquitectura de sistema.</i>	66
<i>Arquitectura de software.</i>	67
<i>Diseño de interfaces.</i>	68
<i>Diseño de base de datos.</i>	77
<i>Diseño de procesos.</i>	87
<i>Diseño de salidas.</i>	96
Programación	104
<i>Entradas y salidas.</i>	104
<i>Procesos.</i>	107
<i>Validaciones.</i>	109
<i>Módulos señalados en el alcance.</i>	111
Pruebas	119
Referencias	124
APÉNDICES	125
Apéndice 1. Entrevista	125

Ilustraciones

Ilustración 1. Pregunta 8	45
Ilustración 2. Pregunta 9	46
Ilustración 3. Pregunta 10	46
Ilustración 4. Pregunta 11	47
Ilustración 5. Pregunta 12	48
Ilustración 6. Casos de uso.....	56
Ilustración 7. Arquitectura de Sistema.....	67
Ilustración 8. Arquitectura de Software	68
Ilustración 9. Inicio de Sesión.....	69
Ilustración 10. Facturación.....	70
Ilustración 11. Inventario	71
Ilustración 12. Compras	71
Ilustración 13. Proveedores.....	72
Ilustración 14. Sucursales	72
Ilustración 15. Agentes	73
Ilustración 16. Colaboradores	73
Ilustración 17. Incapacidades	74
Ilustración 18. Vacaciones	75
Ilustración 19. Pagos planilla.....	75
Ilustración 20. Marcas.....	76
Ilustración 21. Reportes	76
Ilustración 22. Diagrama Base de Datos.....	77
Ilustración 23. Diagrama Productos.....	88
Ilustración 24. Diagrama Compras.	89
Ilustración 25. Diagrama Facturación.....	90
Ilustración 26. Diagrama Pago Planilla.	91
Ilustración 27. Diagrama Planilla Vacaciones.....	92
Ilustración 28. Diagrama Proveedores.....	93
Ilustración 29. Diagrama Sucursales.....	94
Ilustración 30. Diagrama Agentes.....	95

Ilustración 31. Diagrama Colaboradores.....	96
Ilustración 32. Salida Incapacidades.....	97
Ilustración 33. Salida Vacaciones.....	97
Ilustración 34. Salida Reportes.....	98
Ilustración 35. Salida Pago de Planilla.....	98
Ilustración 36. Salida Productos.....	99
Ilustración 37. Salida Facturación.....	99
Ilustración 38. Salida Archivo PDF.....	100
Ilustración 39. Diagrama Secuencia 01.....	101
Ilustración 40. Diagrama Secuencia 02.....	102
Ilustración 41. Diagrama Secuencia 03.....	103
Ilustración 42. Diagrama Secuencia 04.....	104
Ilustración 43. Entrada de Cambio de Estado de Carga Social.....	105
Ilustración 44. Salida Cambio de Estado de Carga Social.....	105
Ilustración 45. Entrada Actualización de Agente.....	106
Ilustración 46. Salida Actualización de Agente.....	106
Ilustración 47. Entrada Credenciales Inicio de Sesión.....	106
Ilustración 48. Salida de Datos al Iniciar Sesión.....	107
Ilustración 49. Proceso de Cálculo de Planilla Capa Interfaz.....	107
Ilustración 50. Proceso de Cálculo de Planilla Capa Lógica de Negocio.....	107
Ilustración 51. Proceso de Cálculo de Planilla Capa de Datos.....	108
Ilustración 52. Proceso de Cálculo de Planilla Base de Datos 01.....	108
Ilustración 53. Proceso de Cálculo de Planilla Base de Datos 02.....	109
Ilustración 54. Validación Marca de Salida.....	109
Ilustración 55. Validación producto existen en factura actual.....	110
Ilustración 56. Validación producto a vender no sea mayor al disponible.....	110
Ilustración 57. Validación de envío de factura por correo electrónico.....	110
Ilustración 58. Módulo de Facturación 01.....	111
Ilustración 59. Módulo de Facturación 02.....	111
Ilustración 60. Módulo de Facturación 03.....	112
Ilustración 61. Módulo de Facturación 04.....	113

Ilustración 62. Módulo de Compras 01.....	113
Ilustración 63. Módulo de Compras 02.....	114
Ilustración 64. Módulo de Compras 03.....	114
Ilustración 65. Módulo de Planilla 01.....	115
Ilustración 66. Módulo de Planilla 02.....	115
Ilustración 67. Módulo de Inventario 01.....	116
Ilustración 68. Módulo de Inventario 02.....	117
Ilustración 69. Módulo de Seguridad 01.....	118
Ilustración 70. Módulo de Seguridad 02.....	119
Ilustración 71. Módulo de Reportes.....	119

Tablas

Tabla 1. Variables de investigación	39
Tabla 2. CU01	56
Tabla 3. CU02	58
Tabla 4. CU03	59
Tabla 5. CU04	61
Tabla 6. CU05	62
Tabla 6. Tabla Agentes	77
Tabla 7. Tabla Colaboradores	78
Tabla 8. Tabla Compra Detalle	78
Tabla 9. Tabla Compras	79
Tabla 10. Tabla Factura Detalle	79
Tabla 11. Facturas	80
Tabla 12. Tabla Planilla Pagos	80
Tabla 13. Vacaciones	81
Tabla 14. Tabla Productos	81
Tabla 15. Proveedores	82
Tabla 16. Tabla Roles	82
Tabla 17. Tabla Stock	82
Tabla 18. Tabla Sucursales	83
Tabla 19. Planilla Lote Detalle	83
Tabla 23. Aguinaldo Acumulado	85
Tabla 24. Marcas	86
Tabla 25. Teléfono Agente	86
Tabla 26. Teléfono Colaborador	86
Tabla 24. Personas	87
Tabla 24. Pruebas Módulo Inicio de Sesión 01	120
Tabla 25. Pruebas Módulo Inicio de Sesión 02	120
Tabla 26. Pruebas Módulo Marcas	120
Tabla 27. Pruebas Módulo Lote Planilla	121
Tabla 28. Pruebas Módulo Facturación	122

Tabla 29. Pruebas Módulo Compras.....	122
Tabla 31.Pruebas Módulo Reportes	123

Resumen Ejecutivo

En el presente proyecto llamado Prototipo Funcional para Centralización de Sistema de Punto de Venta para la cadena de Minisúper La Bodeguita, ubicadas en el área de Desamparados y San Sebastián, se tiene como objetivo desarrollar un prototipo funcional que ayude en la empresa con las gestiones que se realizan de facturación, compra y manejo de pagos de planilla.

Como solución a las problemáticas presentes en la empresa, se crea un prototipo capaz de mejorar la ineficiencia en el proceso de facturación de productos, también en el control de compras a los diferentes proveedores que tiene la organización. Dentro de las soluciones a la organización está la de brindar una serie de reportes para la correcta toma de decisiones.

Para el desarrollo del proyecto, se efectuaron las siguientes cuatro fases: análisis, diseño, programación y pruebas. Razón por la cual el documento cuenta con todo el análisis previo, mostrando los estudios de factibilidad, fuentes de información utilizada, análisis de viabilidades, proyecciones y resultados de entrevistas y cuestionarios aplicados en la empresa. En la fase de diseño, se detalla de manera clara el funcionamiento del sistema mediante diagramas, procesos y arquitecturas que proceden a la programación. En la etapa de programación se desarrolló una serie de módulos que entre sí integran las actividades de la empresa. Como fase final se muestran los resultados obtenidos tras la aplicación de pruebas al prototipo.

La principal conclusión del proyecto, consiste en que, mediante la implementación de este prototipo, la empresa puede mejorar sus sistemas de control e incrementar su productividad, esto debido a que no tendrá que estar realizando procesos de forma manual e ineficientes.

CAPÍTULO I: INTRODUCCIÓN

Descripción de la organización

La cadena de Minisúper La Bodeguita abrió su primer local en el año 2007, en el centro del cantón de Desamparados, San José. En sus inicios, la cadena se dedicaba únicamente a la venta de frutas, verduras, legumbres y refrescos naturales, posteriormente, conforme las ventas fueron disminuyendo, se decidió en el año 2014 realizar una fuerte inversión para comenzar a vender productos de la canasta básica, meriendas, refrescos gaseosos, jugos procesados, lácteos, pastillas, helados y productos para la limpieza del hogar, todo esto con el fin de aumentar las ganancias. Actualmente, La Bodeguita vende al por menor en tres locales comerciales, dos de ellos ubicados en Desamparados, San José, y el otro se encuentra en el área de San Sebastián, San José.

La Bodeguita cuenta con seis colaboradores, que son los encargados de recibir mercadería, limpieza de los locales y de facturar manualmente. Se tiene previsto abrir dos locales más en enero del 2021 y, de igual manera, contratar a dos colaboradores para que laboren en esas sucursales.

Descripción del problema

Se enumerarán una serie de problemas encontrados en la organización:

1. Desconocimiento de inventario real: actualmente, en la cadena nadie conoce el dato real de la cantidad de producto que tiene, por lo que a la hora de realizar pedidos a proveedores puede comprarse producto de más y tener pérdidas, ya que las frutas y las verduras se descomponen de manera rápida, así como alimentos que venzan rápido como la leche. También, la persona encargada de realizar los pedidos desconoce los productos que hacen falta en los diferentes locales, aumentando más la imprecisión a la hora de solicitar más mercadería a los proveedores.
2. Pérdida de información vital: las compras y ventas que se van realizando quedan respaldadas en papel únicamente, por lo que, en varias ocasiones, se han perdido las facturas y, con esto, la información de los movimientos del día, lo cual afecta la integridad de la correcta administración de los locales.

En varias ocasiones, se han perdido las facturas que dan los agentes de los proveedores, por lo que no se sabe cuánto se pagó ese día de mercadería y también se desconoce la

cantidad de producto que poseen. Todo esto causa una mala toma de decisiones, debido a que no se cuenta con la información exacta de ventas ni de compras.

3. Falta de reportes para la toma de decisiones: no existen datos exactos de las ventas, tales como: conocimiento de los productos más vendidos, productos menos vendidos, ventas por día, ventas por mes, ventas por local, producto más vendido por local y producto menos vendido por local. Esto perjudica en la toma de decisiones, debido a que, por ejemplo, no se puede impulsar más un producto que está siendo menos vendido en uno de los locales (ofrecerlo más) y mantenerlo en inventario trae pérdidas al negocio. Al igual que se puede potenciar aún más un producto que deja bastante ganancia.

No se tiene conocimiento de cuánto factura cada colaborador, con lo cual se desperdicia la oportunidad de premiar al colaborador que más vende y así potenciar una sana competencia que beneficia la productividad del negocio.

Una mala decisión por falta de información fue la causa de pérdidas por aproximadamente trescientos cincuenta mil colones en noviembre de 2019, ya que se pidió a un agente de un proveedor un producto que no se vendía mucho y este producto se mantuvo en los diferentes locales hasta que expiró.

4. Descontrol en el manejo de las horas laboradas por parte del personal: en la cadena actualmente no se tiene control sobre las horas laboradas por cada colaborador, tampoco se maneja ningún sistema de marcas para dejar en evidencia que un local se abrió por parte de un empleado a la hora establecida, y mucho menos la hora del cierre. Este descontrol propicia que constantemente haya pequeñas discusiones acerca del pago que le corresponde a cada uno, lo que les está produciendo pérdidas en algunos de estos casos, pues se ha pagado de más a alguno de ellos.

De igual manera, el descontento por parte de los colaboradores, debido a una mala paga, por no tener el control adecuado, se ve reflejado en su productividad que, al final, se transforma en menos ventas.

El administrador constantemente hace cálculos mentales de cuantos días de vacaciones tienen sus colaboradores. En una ocasión, se le dio de vacaciones 4 días a un colaborador y tiempo después se dieron cuenta de que, en realidad, disponía únicamente de 3, lo que claramente provoca pérdidas económicas para el minimercado.

En vista de que la cadena abrirá más locales, se aumentará la cantidad de colaboradores, lo que hace completamente necesario llevar un control de ellos.

Propuesta de módulos a desarrollar

Módulo de inventario.

En este módulo, los colaboradores con acceso a este podrán agregar, eliminar, modificar y consultar productos según la sucursal en la que trabajan. El colaborador con permisos básicos tendrá la opción de consultar el inventario de otra de las sucursales, sin embargo, solo tendrá permisos de consulta. El usuario administrador podrá consultar, eliminar, modificar y crear productos en el inventario de cualquiera de las sucursales.

Módulo de compras.

En este módulo se podrá agregar, consultar, modificar y eliminar las compras realizadas a los diferentes proveedores. Una vez que se agregue el proveedor, el agente del proveedor, el local que recibe, los montos correspondientes y los productos con sus respectivas cantidades, el movimiento se actualizará en el inventario. Las compras se recibirán físicamente desde la sucursal principal y una persona subordinada de la administradora será la que reparta los productos entre las diferentes sucursales.

Módulo de facturación.

El módulo permitirá agregar productos disponibles en inventario a una factura, se podrá eliminar algún producto o disminuir su cantidad, en caso de que el cliente desee llevar menos o simplemente de que el cajero cometa un error. Por pantalla, se mostrará en tiempo real el precio por pagar por parte del cliente. Una vez que se haya realizado la facturación correspondiente, será almacenada en la base de datos. Se podrán eliminar facturas (se puede realizar solo si es usuario administrador), modificarlas (con usuario administrador) o consultarlas (usuario básico).

Módulo de reportes.

En este módulo, se podrá visualizar una variedad de reportes, según lo necesite el usuario administrador (únicamente este usuario puede acceder a él). Los reportes que se pueden visualizar son los siguientes:

1. Cinco productos más vendidos en los últimos 30 días, con su respectiva cantidad y filtrado por sucursal.
2. Cinco productos menos vendidos en los últimos 30 días, con su respectiva cantidad y filtrado por sucursal.
3. Ventas del día por sucursal y con hora de la generación del reporte (fecha a elegir).
4. Ventas por mes, filtrado por sucursal.
5. Ventas globales del mes.
6. Ventas por mes de cada vendedor.
7. Compras por proveedor (en colones) en los últimos 12 meses.

Todos los reportes pueden ser exportados en formato PDF.

Módulo de planilla.

En este módulo, se mostrarán los nombres de los colaboradores de La Bodeguita, se podrá definir el salario base mensual de cada colaborador y calcular cuánto será el salario neto ya con los rebajos correspondientes. Cada mes laborado por colaborador se podrá visualizar el aumento de un día de vacaciones para cada uno de ellos y también se podrán descontar estos días de vacaciones, una vez el colaborador haga uso de este. No se contarán los días feriados ni las horas extra, ya que, por políticas de la organización, estas actividades no se llegan a cabo. Se parametrizará para que únicamente ciertos usuarios puedan tener acceso a este módulo, con el fin de que puedan definir salarios bases y gestionar el tema de vacaciones.

Módulo de seguridad.

Se crearán los usuarios que utilizarán el sistema, se determinarán los perfiles que acceden o no a los diferentes módulos del sistema. Se podrá crear, actualizar, consultar y eliminar perfiles de usuario.

Herramientas que se utilizaran para el desarrollo

- Entorno de desarrollo: Visual Studio 2019.
- Motor de base de datos: Microsoft SQL Server 2019 Developer Edition.
- Lenguaje de programación: ASP.net.

Según lo que se conversó con la dueña de La Bodeguita, se tomó la decisión de desarrollar la aplicación de manera web, ya que en todas las sucursales cuentan con una computadora portátil,

internet vía cable y vía WiFi, por lo que, en caso de que se suspenda el fluido eléctrico, se podrá acceder al sitio sin mayor inconveniente al conectar el computador portátil a una fuente de datos móviles que los mantendrá en línea. El motivo por el cual se determinó desarrollar la aplicación en ASP.net es que la administradora cuenta con una persona a su disposición con conocimientos en el lenguaje y, por temas de mantenimiento posterior, se llegó a la conclusión de que era la mejor opción.

El motor de base de datos SQL Server 2019 Developer Edition se eligió, ya que cuenta con excelente compatibilidad con el lenguaje de ASP.net al ser productos Microsoft, además de ser una tecnología con gran respaldo, confianza y con fácil acceso a conocedores de la tecnología.

Objetivo general

Desarrollar una aplicación web que permita a La Bodeguita manejar su negocio de manera centralizada y eficiente.

Objetivos específicos

1. Analizar las debilidades actuales del negocio para cubrirlas con una solución tecnológica.
2. Diseñar un prototipo funcional para la solución a las distintas debilidades presentes en los procesos del negocio.
3. Desarrollar cada uno de los módulos propuestos para el sistema tecnológico de la empresa.
4. Ejecutar pruebas funcionales al sistema.

Justificación

Con esta asignación, se pretende remediar las diferentes necesidades que se presentan en la cadena de Minimercados La Bodeguita, que fueron solicitadas para mejorar en la parte tecnológica, con el fin de consolidar el negocio en el área de Desamparados y San Sebastián.

Implementar un sistema de punto de venta agilizaría considerablemente los procesos por realizar cuando un cliente compra, cuando los empleados ingresan nueva mercadería, cuando se

facturan los productos, a la hora de llevar un control, por parte de los administradores se pueden generar reportes de manera rápida y oportuna.

Una de las grandes deficiencias del método actual de la empresa es que se maneja la información escrita con papel y lápiz, lo cual hace más fácil su pérdida. Por lo anterior, es evidente que se necesita un repositorio robusto y confiable, donde la información estará salvaguardada de manera correcta.

Viabilidad técnica.

Para lograr el desarrollo y pruebas del prototipo, es fundamental contar con herramientas adecuadas y el equipo justo que permita su correcta utilización. Por la razón mencionada, es importante tomar en cuenta las consideraciones a nivel técnico que se deben tener para lograr el objetivo de este proyecto.

Se requiere contar con conexión a internet para acceder a la herramienta a desarrollar. De igual manera, se debe tener un *router* que genera la señal WiFi en cada local comercial para lograr llevar el internet del proveedor a las computadoras personales.

Algo que se debe tomar en cuenta es que la empresa tiene equipos en cada una de sus sucursales que cuentan con especificaciones que superan el mínimo requerido. Lo que se requiere como mínimo para el desarrollo de la aplicación es:

- Procesador Intel Core i5 de 7th Gen.
- 6 GB de memoria RAM.
- 256 GB de memoria interna.
- Microsoft Visual Studio instalado en el computador.
- Motor de Base de Datos SQL SERVER Developer Edition.
- Microsoft SQL SERVER Management Studio.

Para la utilización del aplicativo, se requiere como mínimo:

- Procesador Intel Core i3 de 8va Gen.
- Almacenamiento de 1 TB.
- Motor de Base de datos SQL SERVER.
- Conexión a internet de 2 MB.

- Navegador de internet.

Observando los requerimientos mínimos y con base en lo expresado por la administradora de la empresa, se concluye que existe la viabilidad técnica, gracias a que se cuenta con todo el equipo necesario, incluidos los elementos de redes y comunicación.

Viabilidad operativa.

Es importante mencionar que, al realizar un sistema de este tipo, es fundamental el apoyo de la empresa beneficiada hacia el encargado del desarrollo. Por parte de la empresa, se cuenta con completo apoyo e interés.

En vista de la puesta en producción del aplicativo, es importante recalcar que todo el personal de la organización cuenta con conocimientos en utilización de computadores, sistemas de puntos de ventas y manejo del sistema operativo Windows.

Con lo mencionado anteriormente, se llegó a la conclusión de que se cuenta con total viabilidad operativa para ser realizado.

Viabilidad económica.

Es importante aclarar que la cadena de minimercados La Bodeguita no pagará el costo del proyecto, lo que incluye la fase de análisis, diseño, programación y pruebas, que no genera remuneración a los encargados del proyecto. La organización cuenta con el equipo suficiente para poner en producción el sistema, por lo que no debe invertir en hardware.

Explicando las condiciones en las que se encuentra el proyecto, se concluye que existe total viabilidad económica.

Proyecciones

En este punto, es importante tener claro que el objetivo más importante del proyecto es satisfacer las necesidades de la organización beneficiada y poder brindarles soluciones a sus deficiencias. Con ello, se busca obtener un rendimiento en las actividades realizadas a diario por parte de los colaboradores, y en que no exista pérdida monetaria por la falta de control de los procesos. El proyecto por realizar consta de un prototipo funcional que incluye las fases de análisis, diseño, programación y pruebas, por lo que no se incluyen las etapas de documentación ni implementación del aplicativo.

Alcance técnico.

El desarrollo del aplicativo está basado en la web, el sistema se publicará en una máquina de alto rendimiento con el que cuenta la organización en una de las sucursales. El desarrollo del sistema se va a realizar en un equipo con las siguientes características y software:

- Lenguaje de programación: ASP.net.
- Motor de base de datos: SQL SERVER 2019 Developer Edition.
- Herramienta de Desarrollo: Microsoft Visual Studio 2019.

Con base en lo anterior, y valorando la posible implementación del aplicativo, se tiene la necesidad de contar con una computadora con las siguientes características:

- Procesador Intel Core i3 de 8va Gen.
- Almacenamiento de 1 TB.
- Motor de Base de datos SQL SERVER 2019 Developer Edition.
- Conexión a internet de 2 MB.
- Navegador de internet.

Alcance metodológico.

El proyecto está compuesto únicamente por las siguientes fases:

Análisis: se realizará un análisis de la problemática que presenta la organización actualmente y con esto tener una idea principal y poder realizar un análisis más a fondo.

Diseño: se determinará mediante diferentes métodos el funcionamiento.

Desarrollo: se desarrollará el sistema con base en las fases anteriores, es decir, una vez se complete el análisis y diseño del aplicativo.

Pruebas: se diseñarán y ejecutarán una serie de casos de prueba, donde se va a garantizar la calidad del producto y que cumpla con lo solicitado por el cliente.

CAPÍTULO II: MARCO REFERENCIAL

En este capítulo, se van a definir una serie de elementos técnicos que son fundamentales para entender de una mejor manera el proyecto. Estos son considerados importantes, ya que ayudarán a poner en contexto al lector.

Como se ha indicado anteriormente, el prototipo funcional requiere de hardware para funcionar, según Pérez y Merino (2008), el “hardware como tal no son solo los componentes físicos internos como el disco duro, tarjeta madre, etc. Sino que también son los periféricos como por ejemplo las impresoras”. Se puede entender con esto que el hardware es todo con lo que se puede interactuar de manera física.

Por su parte, la computadora, de acuerdo con Gardey y Pérez (2008), “es una máquina que es capaz de recibir, procesar y devolver la información; tiene varios medios de entrada y de salida que sirven como medio de comunicación” (párr. 2). La computadora, por sí sola, no es funcional sin los debidos programas informáticos que son necesarios para procesar la información. Esta se complementa con un teclado, un mouse, un CPU, etc. Este último elemento es la principal herramienta para desarrollar sistemas, debido a que permite el diseño, programación y gestión de los datos. El uso del hardware se hace mediante software que, según Pérez (2008), “es todo lo lógico de una computadora, es decir, son todos los programas informáticos” (párr. 2). El software, a diferencia del hardware, es todo lo que se puede manipular de manera virtual.

Para operar el equipo, es necesario contar con un sistema operativo debidamente instalado, el cual brinda el ambiente digital en el que se interactuó con los programas. Silberschatz (2009) indica que “desde el punto de vista de la computadora, el sistema operativo es el programa más íntimamente relacionado con el hardware”. Lo anterior debido a que el sistema operativo será el encargado de administrar los recursos de hardware disponibles, podrá utilizarlos según se requiera.

Para entender el concepto de software como prototipo se puede definir, según Moreno (2010), como “una muestra más pequeña que un sistema informático, que nos va a mostrar cómo será el programa una vez esté completado” (párr. 1). Un prototipo funcional brinda una idea de lo que será el sistema en su versión final, no todas las funciones están disponibles y es probable que aún le falten detalles en su desarrollo visual y operativo.

El hecho de que el prototipo sea un sistema aún sin finalizar por completo no quiere decir que este va a mostrar errores groseros, tales como que el sistema se cierre sin razón aparente, o que haga cálculos incorrectos en caso de que, por ejemplo, se haga una suma de productos para una factura.

El sistema por desarrollar proveerá una serie de reportes con información importante para la organización. Esta información se coleccionará con base en los datos que van siendo almacenados en la base de datos. Según Nevado (2010), “un dato no tiene ningún sentido, se debe procesar para que tenga sentido. Los datos son una fracción pequeña de la información y si los datos no están procesados estos no van a tener ningún sentido” (p. 18). Para aclarar un poco más el tema, Porto y Merino (2012) indican que “la información está compuesta por un grupo de datos ordenados. Los datos procesados por una empresa son muy importantes, hasta el punto de que una posible filtración de ésta puede causar una pérdida monetaria fatal para la organización” (p. 15).

Otro tema que es importante y que va directamente relacionado con el prototipo es el de la base de datos. Esta será el repositorio donde se almacenan los datos ingresados en el sistema, desde las compras realizadas, las ventas, la facturación según cada usuario, entre otras cosas. De la base de datos, se podrá extraer la información necesaria para formar los reportes con los que la administradora de la empresa tomará decisiones acertadas, según su confiabilidad. De acuerdo con Capote (2009), “es el fondo común almacenada en una computadora para que cualquier persona o programa autorizado pueda acceder a ella, independientemente de su procedencia y el uso que haga” (p. 7).

La base de datos es similar a la función de un almacén, donde los estantes son las tablas de la base de datos y los objetos en los estantes son los datos almacenados. Para poder administrar la base de datos, se necesita un gestor de base de datos, que es el sistema que facilita este manejo.

María Reffino (2018) citó en su texto *Seguridad* que “la seguridad refiere a la ausencia del peligro, miedo y riesgos”. Lo anterior se puede interpretar en el sentido de que ninguna persona ajena a la organización puede tener acceso ilimitado a los datos que posee la empresa en sus bases de datos, incluso dentro de la organización puede estar limitado su acceso por parte de algunos funcionarios.

Es importante considerar en el desarrollo de este capítulo el lenguaje de programación. En el texto llamado *Lenguaje de Programación*, Raffino (2018) dice que “un lenguaje de programación se le conoce como un programa capaz de diseñar otro programa, esto se debe porque comprende un lenguaje” (párr. 1). Los lenguajes de programación brindan la posibilidad de desarrollar sistemas para satisfacer las necesidades de los clientes. Existen diversos tipos de lenguajes de programación, a simple vista, la diferencia entre ellos es la sintaxis, es decir, la estructura en la que se digita el código de programación. El código de programación es el conjunto de instrucciones que se digita para que el sistema realice una acción determinada.

ASP.net es un lenguaje de programación orientado a la web, es decir, permite publicarlo en una página web donde se podría dar acceso público a este. Según Microsoft (2015), “ASP .net es un marco de trabajo web gratuito para crear sitios web”. Para aclarar el concepto de marco de trabajo, se puede mencionar que, de acuerdo con Suárez (2016), “el marco de trabajo es una estructura con módulos de software, programas en lenguaje de programación que se utiliza para desarrollar software, esto se utiliza a la hora de realizar un programa pequeño”.

Para poder desarrollar código y cumplir con las necesidades del cliente, se utiliza un aplicativo llamado Visual Studio, el cual permite desarrollar aplicaciones en diversos lenguajes de programación como Visual Basic, C# y ASP .NET. Según Microsoft (2016), Visual Studio es un completo entorno de desarrollo integrado para crear aplicaciones para Windows, iOS y Android, además de aplicaciones web y servicios de nube innovadores. Es importante mencionar que no todos los lenguajes de programación pueden ser desarrollados con la herramienta Visual Studio.

El sistema por desarrollar cuenta con un módulo de planilla que, de acuerdo con Chambi (2011), “es un documento que expone la información periódica mensual en valores monetarios, con un conjunto de aportes, descuentos laborales y otros beneficios” (p. 37).

El sistema será utilizado por colaboradores de la organización que cuentan con un conocimiento básico, pero suficiente para utilizar la herramienta por desarrollar. Los colaboradores son las personas encargadas de realizar diferentes funciones dentro del ámbito laboral, pueden tener todos diferentes funciones o realizar similares acciones dentro del trabajo. Todos se recompensan monetariamente por realizar sus labores. Montalvo (2011) indica que “un colaborador es una persona que interactúa en todo el proceso donde se involucre el trabajo de varias personas en

equipo o en grupos” (párr. 1). Es decir, un colaborador lleva a cabo sus funciones en un lugar de trabajo, con una remuneración salarial.

El tipo de negocio presentado en este proyecto es el de venta al detalle de una amplia variedad de productos alimenticios y del hogar, donde todo inicia cuando se realiza una compra de productos a un proveedor, posteriormente, un cliente le compra estos productos a la organización. La organización, mediante un sistema informático, gestionará la compra del cliente, dejando registro de todos los movimientos.

El módulo de compras de un sistema como el que se va a desarrollar es el apartado en el que un colaborador de la organización registra un movimiento de compra a un proveedor de productos o de servicios. El colaborador también tiene el poder de modificar, consultar y eliminar los registros de compras.

En lo que respecta a la seguridad, un módulo de seguridad permite gestionar el control de roles y accesos que va a tener cada usuario de un sistema. Existen diversos tipos de roles, por ejemplo, roles de administrador, donde, por lo general, cuentan con acceso a todos los demás módulos. Otros usuarios tendrán menos accesos, depende de su función dentro de la organización.

Cuando se habla del módulo de inventario, se hace referencia al apartado donde están registrados todos los productos que se venden en la organización. Una vez que se realiza una compra a un proveedor, el inventario va creciendo, debido a que se van obteniendo productos disponibles para vender. En un módulo de estos, se puede gestionar todo lo que tenga que ver con el inventario, desde consultar disponibilidad de productos, modificarlo y eliminarlo.

El módulo de planilla es el apartado donde se muestra el control de pagos a los colaboradores, los salarios, los días de vacaciones correspondientes por ley, así como la cantidad de días laborados. En este módulo, se apoya la organización para gestionar temas de índole laboral de sus colaboradores.

CAPÍTULO III: MARCO METODOLÓGICO

Enfoques de la investigación

Primeramente, el enfoque cuantitativo, según Hernández (2014), “es secuencial y probatorio. El enfoque cuantitativo tiene diferentes etapas que no pueden saltarse, el orden es importante, pero desde luego se puede redefinir alguna fase” (p. 4). Este enfoque parte de la idea que se va a utilizar y, una vez delimitada, se derivan objetivos y preguntas de investigación. Posteriormente, se establecen hipótesis y determina variables, se traza un plan, se miden las variables, se analizan las mediciones obtenidas utilizando métodos estadísticos y se extrae una serie de conclusiones.

Por su parte, el enfoque cualitativo, de acuerdo con Hernández (2014), “se guía por áreas o temas significativos de investigación. La diferencia con el enfoque cuantitativo es que cada vez que la claridad de las preguntas de la investigación e hipótesis proceda con la recolección y análisis de los datos, estos estudios cualitativos pueden desarrollar preguntas antes, durante o después de la recolección y el análisis de los datos” (p. 7). El investigador, conforme va examinando la realidad y va recolectando datos, irá desarrollando una teoría coherente con lo que observa que ocurre.

Enfoque de investigación cualitativo

Los tipos de datos que se recolectan pueden ser de distintas clases: orales, visuales, escritos y simbólicos. El fin de la aproximación cualitativa es descubrir patrones, anomalías en los eventos estudiados y analizados. La gran mayoría de hipótesis surgen durante la indagación y van depurándose conforme se obtienen más datos.

Según Sampieri (2017), algunas de las características de la investigación cualitativa son:

1. No se sigue un camino definido, una sola ruta preestablecida.
2. El investigador comienza examinando la realidad y va recolectando datos y desarrollando una teoría coherente con lo que observa que ocurre.
3. Se utiliza más una lógica y un proceso inductivo.
4. La “realidad” se define por las interpretaciones de los participantes en la investigación respecto de su situación. Es así como convergen varias realidades que van modificándose conforme transcurre el estudio.
5. La recolección y el análisis de los datos no son completamente estandarizados (p. 115).

Enfoque de investigación cuantitativo

Hernández (2014) expresa que el enfoque de investigación cuantitativo es secuencial y probatorio. Este tiene diferentes etapas, las cuales no se pueden “brincar”, el orden es importante, pero, desde luego, se puede redefinir alguna fase. Parte de una idea que se va a utilizar y, una vez delimitada, se derivan objetivos y preguntas de investigación. De las preguntas se establecen hipótesis y determina variables donde se traza un plan, se miden las variables, se analizan las mediciones obtenidas utilizando métodos estadísticos, y se extrae una serie de conclusiones. El orden de las fases del enfoque cualitativo es el siguiente:

1. Idea.
2. Planteamiento del problema.
3. Revisión de la literatura y desarrollo del marco teórico.
4. Visualización del alcance del estudio.
5. Elaboración de hipótesis y definición de variables.
6. Desarrollo del diseño de investigación.
7. Definición y selección de la muestra.
8. Recolección de los datos.
9. Análisis de los datos.
10. Elaboración del reporte de resultados (¿Fuente?).

Enfoque de investigación mixto

El enfoque mixto, según Ruiz (2013), es la combinación de los dos enfoques anteriores, ya que tiene características similares. Ruiz cita a Hernández (2013, p. 5) y señala que estos enfoques utilizan 5 características similares, las cuales son las siguientes:

1. Llevan a cabo observación y evaluación.
2. Establecen ideas como consecuencia de la observación y evaluación realizadas.
3. Prueban el grado en las ideas que tienen fundamento.
4. Revisan las ideas sobre la base del análisis.
5. Propone nuevas observaciones y evaluaciones para esclarecer, modificar y fundamentar las suposiciones o ideas.

Ruiz (2013) añade lo siguiente:

Al utilizar el enfoque mixto, se entremezclan los enfoques cualitativos y cuantitativos en la mayoría de sus etapas, por lo que es conveniente combinarlos para obtener información que permita triangularla. Esta triangulación aparece como alternativa a fin de tener la posibilidad de encontrar diferentes caminos para conducirlo a una comprensión e interpretación lo más amplia del fenómeno en estudio (p. 11).

Enfoque de investigación seleccionado

Para la investigación, se seleccionó el enfoque mixto, ya que mezcla las mayores fortalezas de los enfoques mencionados anteriormente y se requiere contar con características de ambos para esta investigación. Mediante una serie de entrevistas al personal que labora en La Bodeguita, se obtendrá información que, luego, será analizada para determinar las soluciones para la problemática presente.

Métodos de la investigación

La selección del método a utilizar, según Sampieri (2017), “implica penetrar en la realidad que se examina desde una óptica peculiar, que depende del contexto, la situación, los recursos disponibles, los objetivos y el problema planteado” (p. 97). El método de investigación es la estrategia que se utilizará con el propósito de obtener la información necesaria para responder el planteamiento del problema. Se utilizarán diseños para analizar la certeza de las hipótesis formuladas en un contexto específico.

Este método constituirá la estrategia para confirmar si es o no cierto la hipótesis planteada previamente, de acuerdo con Sampieri (2017), “si el diseño está concebido cuidadosamente, el producto final de un estudio tendrá mayores posibilidades de producir conocimiento” (p. 97). Algunos de los métodos más utilizados son:

1. Investigación exploratoria: se busca examinar un tema que no ha sido estudiado anteriormente.
2. Investigación descriptiva: narra características y propiedades de un objeto o situación en específico, de manera objetiva.
3. Investigación explicativa: orientada a dar respuesta a la causa de eventos y situaciones de índole social o físico, explicando por qué ocurre y las condiciones en las que se da.
4. Investigación histórica: como característica principal, estudia el pasado reconstruyendo sus hechos a partir del uso de evidencias documentales confiables, utiliza fuentes de

información primarias y secundarias, empleando a una crítica interna y externa los datos obtenidos para lograr alto grado de objetividad.

Método de investigación descriptivo.

Se utiliza para describir la realidad de las situaciones, eventos, personas, grupos o comunidades que estén abordando y que se pretende analizar. La cuestión al problema no va más allá del nivel descriptivo, ya que consiste en plantear lo más relevante de un hecho o situación específica. Las principales fases a seguir en la investigación descriptiva son:

1. Examinar características del tema a investigar.
2. Definirlo y formular hipótesis.
3. Seleccionar la técnica para la recolección de datos.
4. Validar las fuentes a consultar.

Método de investigación seleccionado.

En presente proyecto, se utilizará el método descriptivo, donde podremos estudiar las situaciones que se presentan en la organización y formular algunas hipótesis de estas. Se realiza una observación a eventos específicos donde se puede determinar la existencia o no de una problemática y, con base en esta observación, tomar acciones para solucionarlas.

Fuentes de información

Una fuente de información es de dónde se proveen datos para construir la investigación, hechos y bases de conocimiento. Maranto y González (2015) explican que “Las fuentes de información son cuando realizamos revisión de la literatura, esta debe ser de forma selectiva y dinámica, debido a que continuamente está surgiendo publicaciones acerca de los avances en distintos campos del conocimiento humano en torno a un tema determinado” (párr. 1). Existen tres tipos de fuentes de información:

1. Fuentes primarias.
2. Fuentes secundarias.
3. Fuentes terciarias.

Fuentes primarias.

Según Gómez (2006), “constituyen el objetivo de la búsqueda bibliográfica y proporcionan datos de prima mano, directamente del autor” (p. 51). Es la información de primera mano, proviene

del resultado de análisis originales con base en una investigación previa. Viene de libros, documentales oficiales, trabajos que se presentan en conferencias, etc.

Fuentes secundarias.

Maranto y González (2015) hablan sobre las fuentes de información secundarias y expresan que “son las que ya la información viene procesada de una fuente primaria. El procesado de esta información se pudo dar por medio de una interpretación” (p. 78). Esto nos definiría el concepto como la interpretación por una segunda persona de una información proporcionada por una fuente primaria.

Fuentes terciarias.

Las fuentes terciarias se utilizan básicamente para dar una idea general, son la recopilación de fuentes primarias y secundarias. Según Gómez (2006), “se trata de documentos que compendian nombres y títulos de revistas y otras publicaciones periódicas, así como sitios web, empresas, asociaciones, catálogos de libros básicos y nombres de instituciones nacionales e internacionales al servicio de la investigación” (p. 51).

Para la realización de este proyecto, se utilizarán fuentes de primera mano, es decir, fuentes primarias, ya que la información se recolectará directamente con el personal de la empresa mediante una serie de encuestas y entrevistas. De igual manera, se realizará un proceso de observación, con el fin de identificar las necesidades del negocio y así determinar los requerimientos del prototipo funcional.

Se recolectará información de fuentes secundarias, con el objetivo de conocer más al detalle conceptos del negocio necesarios para su correcto funcionamiento, esto será consultado en libros. De la fuente terciaria, se obtendrá una recopilación de sitios web acerca de temas del negocio. Esto como apoyo a las fuentes primarias y secundarias.

Variables o unidades de análisis

Una variable es una propiedad que puede ser susceptible a variaciones y que permite ser medida de alguna manera, puede cambiar su propiedad de forma constante durante el tiempo. La variable puede adquirir diferentes valores, según su naturaleza. Toro y Parra (2006) indican que “una definición conceptual explica el término o variable con otras palabras. Son definiciones de

diccionario o de libros especializados y cuando describen la esencia o las características reales de un objeto o fenómeno se les denomina definiciones reales”.

Tabla 1. Variables de investigación

Objetivo específico	Variable	Variable conceptual	Variable operacional	Variable instrumental
Analizar las debilidades actuales del negocio para cubrir las con una solución tecnológica.	Requerimientos del cliente.	Según Pérez y Gardey (2014), “un requerimiento es el acto y consecuencia de requerir”.	Entrevistas con la administradora del negocio.	Entrevistas.
Diseñar una serie de soluciones tecnológicas para las distintas debilidades presentes en los procesos del negocio.	Diseño de las diferentes soluciones.	Quito (2012) indica que “el diseño es un proceso complejo que abarca decisiones de diferentes niveles. Es más fácil cuando se descompone el problema con diferentes subproblemas y se resuelve utilizando diferentes técnicas”.	Diseño de los diferentes módulos que tendrá el sistema.	SQL SERVER y Management Studio.
Desarrollar cada uno de los módulos propuestos para suplir las necesidades tecnológicas de la empresa.	Prototipo funcional.	De acuerdo con Alegsa (2010), “es una muestra más pequeña que un sistema informático (párr. 1), Nos va a mostrar cómo va a ser el sistema cuando este completado” (párr. 2).	Desarrollo de interfaz de usuario, procesos, consultas, reportes y demás funcionalidades	Visual Studio 2019
Ejecutar pruebas funcionales del sistema para asegurar la calidad de este.	Pruebas funcionales.	Según Mera (2016), “se basan en funciones, presentaciones y se pueden llevar a cabo en todos los niveles de las pruebas” (p. 170).	Implementación de las pruebas.	Documentos de casos pruebas.

Fuente: Elaboración propia.

Muestra y población

En este proyecto, la población serán todos y cada uno de los colaboradores que laboran para la organización, esto debido a que son solamente seis personas aparte de la administradora. Según López (2014), “la muestra es el subconjunto de la población en que se va a llevar a cabo la investigación, hay diferentes tipos de procedimientos con los cuales se pueden obtener la cantidad de la muestra, la muestra es una gran parte de la población”. La población es un conjunto de

elementos de los que se va a conocer algo de la investigación, puede estar constituido por animales, registros, nacimientos, accidentes viales, personas, etc.

La muestra es un extracto de elementos de una población, con el objetivo de ser estudiada. El muestreo es uno de los métodos que se utiliza para seleccionar los componentes de la muestra. En este proyecto, se utilizará la forma cualitativa. Existen diversos tipos de muestreo para la metodología cualitativa, el que se utilizará es el de por conveniencia, ya que se seleccionará la muestra, según su rol en la empresa y así hacer consultas específicas a alguien que tiene conocimiento.

Instrumentos de recolección de datos

El instrumento de recolección de datos es el medio por el cual se extraerá la información de la muestra, según Arias (2012), “un instrumento de recolección de datos es cualquier recurso, dispositivo o formato (en papel o digital), que se utiliza para obtener, registro o almacenar información” (p. 67).

En el proyecto, se utilizarán las entrevistas como instrumento para la recolección de datos, para posteriormente tener información y poder satisfacer los requerimientos que solicite el cliente. Arias (2012) indica lo siguiente:

La entrevista, más que un simple interrogatorio, es una técnica basada en un dialogo o conversación cara a cara, entre el entrevistador y el entrevistado, acerca de un tema previamente determinado, de tal manera que el entrevistador pueda obtener la información requerida (p. 67).

Se entrevistará a la administradora del negocio y a todos los demás colaboradores de la organización, esto con el propósito de obtener información vital para el desarrollo de la solución a brindar. Se deben hacer consultas orientadas a temas fundamentales, los cuales brindarán la información que se necesita, es decir, evitar realizar preguntas por temas que no entran en el alcance del proyecto.

Arias (2012) explica que “las entrevistas se caracterizan por la profundidad él quiere decir que averigua de forma amplia una gran cantidad de aspectos. Realizar una entrevista puede ocupar un tiempo bastante significativo” (p. 73). En una entrevista, se puede extraer información esencial para la investigación, por lo que el entrevistador debe realizar preguntas valiosas, directas, en las

que el entrevistado brinda la mayor cantidad de detalles posibles. La entrevista se clasifica de la siguiente manera:

Según Arias (2012), “las entrevistas se dividen en 3 el habla sobre la entrevista estructurada o formal la cual menciona que se realiza a partir de una guía prediseñada y contiene las preguntas que se realizarán” (p. 73). En una entrevista estructurada o formal, las preguntas a responder por parte del entrevistado son previamente diseñadas con el propósito de extraer información importante. Deben ser cuidadosamente formuladas con el objetivo de obtener lo que se necesita realmente y no información innecesaria.

Asimismo, Arias (2012) expresa que “la entrevista informal no se dispone de una guía ya elaborada. Sin embargo, se deja llevar por unos objetivos que permiten definir el tema de la entrevista” (p. 73). En este caso, el entrevistador debe tener claro cuál es el tema del que va a realizar las preguntas, sin embargo, no lleva una estructura de preguntas previamente elaboradas.

Por último, Arias (2012) habla sobre la entrevista semiestructurada, “aunque exista una guía estructurada de preguntas el entrevistador puede realizar otras que no se encuentren en el documento” (p. 74). Este tipo de entrevista es mixta, se llevan ciertas preguntas previamente formuladas y en el transcurso de la entrevista pueden surgir nuevas consultas de las cuales se puede sacar provecho.

Proceso para la recolección de análisis de datos

Para recolectar la información necesaria, se realizará una visita a la empresa “La Bodeguita”, con el objetivo de entrevistar a la dueña y administradora de los locales. Se abordarán temas relacionados con los procesos que llevan a cabo y que actualmente se realizan de manera rudimentaria, esto brindará pistas de las necesidades que tiene el negocio para ser más eficiente.

El total de colaboradores de La Bodeguita será sometido a una entrevista individual mediante los cuales se obtendrá información de cómo se están realizando sus labores, cuánto tardan en estas, cómo almacenan actualmente la información del negocio, entre otros temas.

Una vez que se recolecte toda la información, se procederá a realizar un análisis de todo lo obtenido. Se evaluarán los tiempos que necesita el personal para realizar sus funciones y se comparará con lo que duraría aproximadamente el sistema. De igual forma, se analizarán cuáles

de las funciones actuales de los colaboradores pueden ser agilizadas mediante una solución tecnológica.

CAPÍTULO IV: ANÁLISIS DE RESULTADOS

Luego de recolectar la información necesaria, se procedió a realizar el tratamiento correspondiente para su análisis, por lo que la información que arroje será la que indique las conclusiones a las cuales llegará esta investigación. Una vez aplicadas las entrevistas a todos los colaboradores, se notó que en las preguntas abiertas que se realizaron, se obtuvieron las mismas respuestas por parte de todos. Esto fue así debido a que los procedimientos que se realizan en la organización son realizados de la misma manera por todos.

1. ¿Cómo se realiza el proceso de facturación de un producto?

En esta pregunta, como se indicó anteriormente, se respondió igual por todos los colaboradores, la respuesta fue que primero se suman en calculadora y de manera manual el precio de los productos que lleva el cliente, posteriormente se cobra y, en la mayoría de las ocasiones, se apunta la venta en un cuaderno. Sin embargo, según mencionan, en algunos casos, cuando el local está muy lleno, no se apunta la venta.

2. ¿Cómo llevan el control de las ventas de productos?

La totalidad de entrevistados respondió que el control que llevan es muy inexacto, debido a que simplemente de manera visual se fijan si tienen algún producto o no, no existe un mecanismo fiable para validar si hay o no algún producto, esto se complica si se trata de productos en los otros locales.

3. ¿Cómo se lleva el control de la mercadería comprada a proveedores?

Todos los entrevistados respondieron que el único control que tienen son las facturas que les da el proveedor, sin embargo, no se cuenta con información exacta de cuando le compran a cada proveedor por semana ni mes.

4. ¿Cómo calculan los pagos a los colaboradores?

Según la respuesta de la administradora y dueña, los cálculos los hacen tomando en cuenta el salario base que tiene cada colaborador y se le rebaja lo correspondiente por ley. Si algún día un trabajador no labora, se le descuenta el equivalente al día que no trabaja.

Según la respuesta del resto de colaboradores, ellos no saben cómo se calculan los pagos en La Bodeguita, no obstante, ellos alegan que básicamente hacen cálculos mentales para determinar si el pago recibido es el correcto.

5. ¿Cómo se lleva el control de los días de vacaciones disponibles por colaborador?

La administradora responde que cada vez que se acuerda actualiza un cuaderno donde lleva apuntado los días que han solicitado los colaboradores de vacaciones, y hace cálculos casi de manera mental de cuántos días disponibles tiene cada uno de los colaboradores.

Los demás colaboradores responden que no llevan un control como tal, sino que hacen cálculos mentales de cuantos días de vacaciones tienen.

6. ¿De qué manera toman decisiones con relación a la compra de productos?

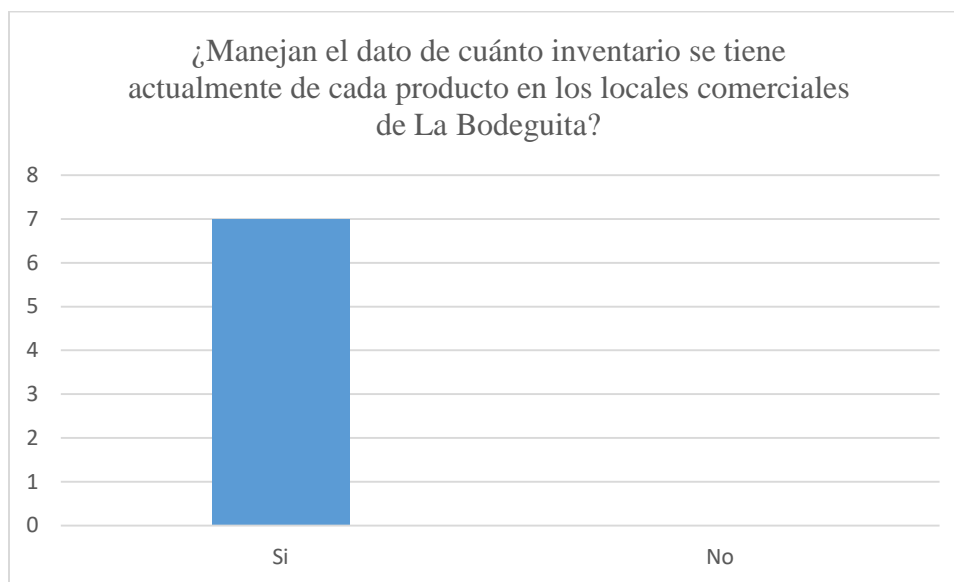
Todos los colaboradores responden que se toma decisiones de compra respecto a la cantidad de producto que visualizan. Algunos productos quedan por fuera de este análisis por el simple hecho de que no observaron el producto y se les olvidó solicitarlo.

7. ¿De qué manera se informan sobre el disponible de productos en los otros locales comerciales de La Bodeguita?

Los colaboradores responden que se realiza una llamada telefónica a otro colaborador que se encuentre en el local deseado y se le pregunta si existe disponible de determinado producto.

8. ¿Manejan el dato de cuánto inventario se tiene actualmente de cada producto en los locales comerciales de La Bodeguita?

Ilustración 1. Pregunta 8



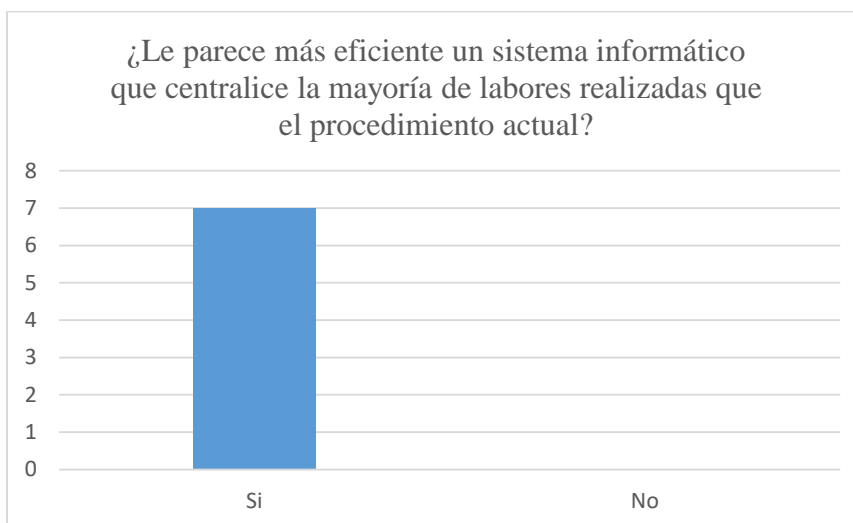
Fuente: Elaboración propia.

Como se puede notar en la gráfica anterior, el 100 % de las entrevistas expresan que no manejan el dato de cuánto inventario se tiene actualmente de cada producto. Para la toma de decisiones, es fundamental tener esta información, ya que, con base en esto, se puede determinar si se necesita comprar más de un producto que se está acabando o, por el contrario, no comprar un producto que hay en exceso, incluso, se podría trasladar un producto de una tienda a otra, en caso de ser necesario.

Con un sistema centralizado, la información de la cantidad de producto en cada una de las sucursales va a contribuir a esa toma de decisiones que puede aumentar las ganancias, mediante una mayor oferta a los clientes de productos de alta demanda.

9. ¿Le parece más eficiente un sistema informático que centralice la mayoría de las labores realizadas que el procedimiento actual?

Ilustración 2. Pregunta 9

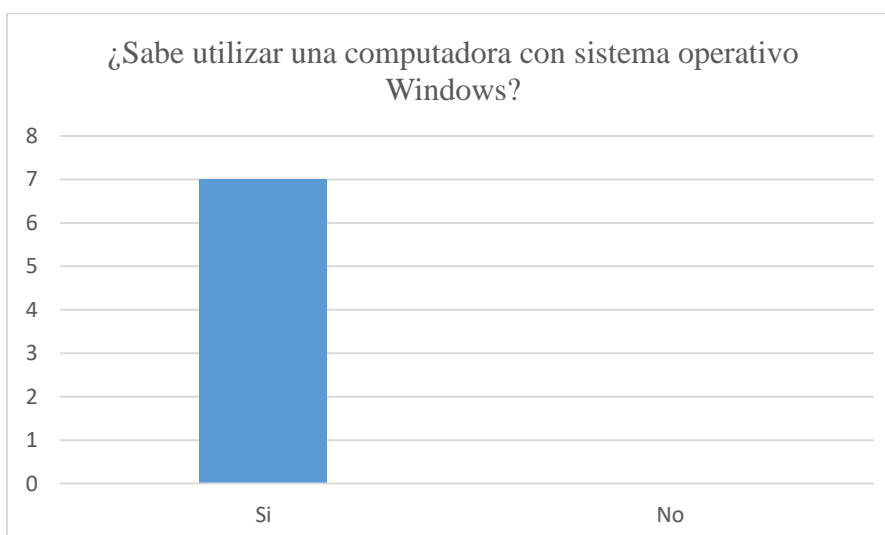


Fuente: Elaboración propia.

Con base en el gráfico anterior, se puede deducir que el 100 % de los encuestados cree que es más eficiente realizar sus funciones mediante un sistema informático que de la manera actual. Es importante destacar que la aprobación de los colaboradores de cara a la posible implementación de un sistema informático es muy positiva.

10. ¿Sabe utilizar una computadora con sistema operativo Windows?

Ilustración 3. Pregunta 10

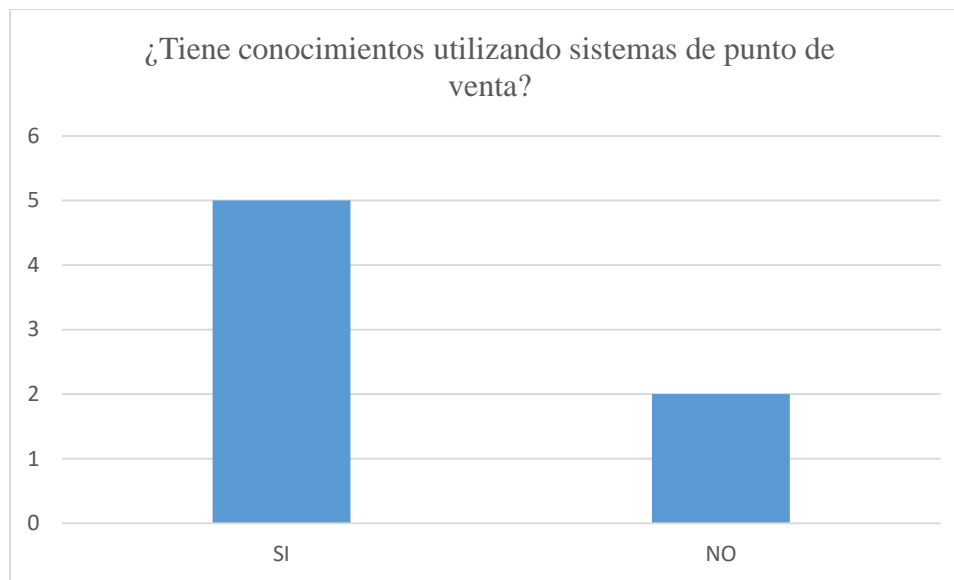


Fuente: Elaboración propia.

Dichosamente, el 100 % de los encuestados tiene conocimiento en el uso de computadoras personales con sistema operativo Windows. Cabe recalcar que el sistema será desarrollado para esta plataforma, lo que facilitará el tema de la capacitación para los colaboradores.

11. ¿Tiene conocimientos utilizando sistemas de punto de venta?

Ilustración 4. Pregunta 11

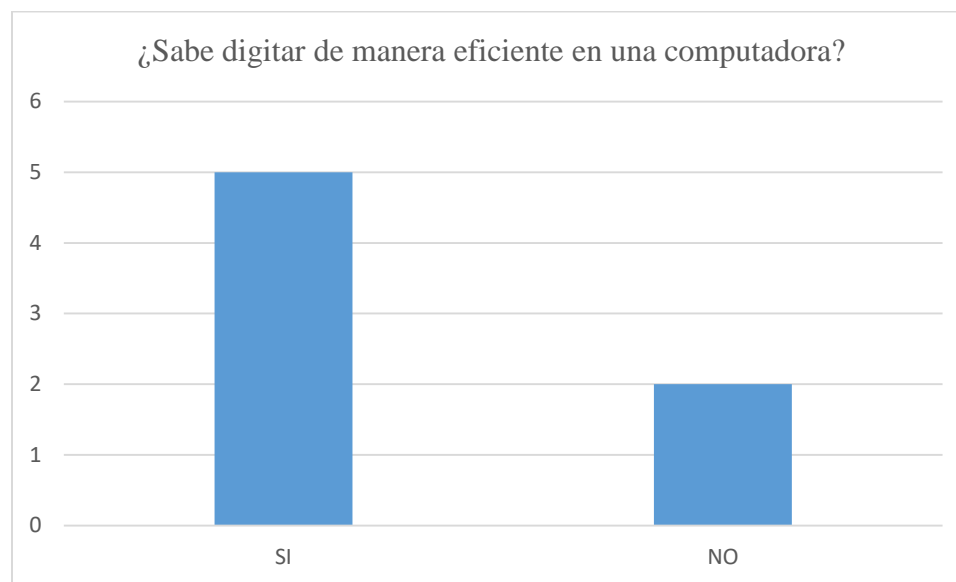


Fuente: Elaboración propia.

Según el gráfico anterior, poco más del 70 % de los encuestados tiene conocimientos utilizando sistema de punto de venta, sin embargo, como se observó en el gráfico anterior a este, todos tienen conocimiento en utilización de computadoras con sistemas operativos Windows, lo que sugiere que no sería mayor problema una capacitación en la utilización del sistema a desarrollar.

12. ¿Sabe digitar de manera eficiente en una computadora?

Ilustración 5. Pregunta 12



Fuente: Elaboración propia.

Como se puede observar, únicamente dos personas consideran que no sabe digitar eficientemente en una computadora, esto, si bien es cierto se trata de un tema ajeno al sistema como tal, sugiere que la gran mayoría de colaboradores son actualmente aptos para utilizar el sistema y lograr un uso eficiente.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

En este apartado, se detallarán las conclusiones y recomendaciones que se lograron obtener a la hora de realizar el prototipo funcional, así, se podrán detallar puntos de mejora para corto o largo plazo, los cuales se obtuvieron gracias a los resultados finales.

Conclusiones

A continuación, se van a especificar las conclusiones que se obtuvieron una vez finalizado el proyecto realizado para La Bodeguita, estas conclusiones van a describir cómo se resolvieron los objetivos que se establecieron al principio del documento. Cada uno de ellos se pudo realizar con éxito en los diferentes capítulos del proyecto.

Como primer objetivo específico tenemos, el análisis de las debilidades actuales del negocio para cubrirlas con una solución tecnológica. Este objetivo se logró cumplir gracias a una serie de mecanismos, donde se evidenció que la organización no estaba realizando ciertos procesos de la mejor manera.

Este análisis permitió tener una idea clara de las diferentes necesidades que tiene la organización, y con base en él se diseña una solución tecnológica que corrija los procesos con amplio margen de mejora. Para este tema se documentaron los casos de uso, con el fin de ahondar un poco más.

Por otra parte, se logra cumplir el segundo objetivo que es diseñar un prototipo funcional para solventar las distintas debilidades presentes en los procesos del negocio. El diseño del prototipo fue clave, ya que es la base para el proceso de desarrollo. En este objetivo se evidencia que un diseño robusto permite dar mayor claridad para el proceso de desarrollo.

Como tercer objetivo, se tiene el desarrollo de cada uno de los módulos propuestos para suplir las necesidades tecnológicas de la empresa, esto hace referencia a la creación de una página web donde se plasmarán todos los diseños especificados en el objetivo anterior. Gracias al desarrollo se logró contar con una herramienta que facilita la ejecución de procesos en la organización, brindando una eficiencia superior a la que se tiene actualmente. De igual manera, se provee de confiabilidad e integridad en la información requerida por La Bodeguita.

El último objetivo del proyecto, tiene como fin el aseguramiento de calidad del producto final. Es por esto, que se realizaron pruebas unitarias, es decir, a un módulo independiente su correcta

funcionalidad y también pruebas de regresión para asegurar que no se afectara la funcionalidad de otros módulos. Es importante aclarar que las pruebas realizadas se hicieron de manera manual.

En las pruebas realizadas se intentó encontrar errores en temas de cálculos de montos, digitación de tipo de carácter no esperado por la base de datos lo cual puede generar un incorrecto funcionamiento del sistema, etc. Gracias a este objetivo se logró comprobar que el sistema cuenta con un alto nivel de calidad en su programación y es apto para manejar posibles usos incorrectos del mismo por parte del usuario.

Fue de gran importancia la implementación de pruebas al sistema, ya que se detectaron potenciales errores y solucionados a tiempo. Queda muy claro que no solamente se trata de desarrollar un sistema y ponerlo en producción, la fase de pruebas va a asegurar un producto de calidad y confiable para el usuario final, no siendo correcto brindar una supuesta solución al cliente que podría hacer incurrir en pérdidas monetarias a la dueña de la organización.

Un gran aporte a nivel profesional es lo que deja un proyecto de esta magnitud, ya que, mediante soluciones tecnológicas, los informáticos pueden brindar ayuda a organizaciones no tan poderosas económicamente y colaborar a dar el paso tecnológico que es tan necesario hoy en día para las organizaciones.

Finalmente, se concluye con la realización de este proyecto que se pudo obtener una experiencia de aprendizaje no solo en desarrollo de una herramienta, sino en el análisis de problemáticas presentes en la cotidianidad de muchas organizaciones. Lograr determinar cómo solucionar un problema fue una de las grandes enseñanzas que dejó este proyecto, muchas veces se tiene claro cuando existe un punto débil en una organización, pero no siempre se sabe cómo solucionarlo, y este proyecto brinda una experiencia enriquecedora en este tema.

Recomendaciones

En esta sección se darán algunas recomendaciones como punto de mejora para la implementación del prototipo funcional. Las recomendaciones que se brindarán a continuación se deben de realizar en un plazo máximo de cuatro meses después de la puesta en producción, esto para, por ejemplo, evitar pérdidas de información en caso de catástrofe y que no exista un respaldo de la base de datos. La responsable de asegurarse que se lleven a cabo las recomendaciones es la administración de La Bodeguita.

Se recomienda la creación de un manual de usuario, donde el colaborador en caso de olvidar la manera de realizar un proceso pueda tener apoyo escrito. De igual manera, en caso de que exista un nuevo colaborador puede capacitarse de manera veraz. Para esta recomendación se estima que debe contratarse un analista de sistemas junior, con experiencia en el análisis de sistemas desarrollados en ASP.NET y SQL SERVER. El tiempo estimado para el análisis y desarrollo del manual de usuario es de 80 horas, donde por cada hora, según la página del Ministerio de Trabajo y Seguridad social (2020), se estima un costo de 1.735 colones, para un total de 139.000 colones.

Se recomienda enérgicamente, la creación de un respaldo de la base de datos en algún otro equipo distinto al actual, para que en caso de un fallo en la base de datos utilizada se logre apuntar al respaldo y así tener un ambiente de alta disponibilidad. Para esta recomendación se debe contar con una persona con amplio conocimiento en base de datos (senior preferiblemente) para realizar esta mejora, se estima que la duración del análisis, creación del procedimiento y su ejecución es de 48 horas. El costo por hora, según la página del Ministerio de Trabajo y Seguridad Social (2020), se estima en 2.835 colones, para una inversión total de 136.000 colones.

Por último, se recomienda enviar mediante correo electrónico el detalle de la planilla pagada individualmente a cada uno de los colaboradores, y así puedan llevar el control personal de cuánto dinero le han pagado cada mes. Para implementar esta mejora se estima un tiempo total de 48 horas, donde se toma en cuenta el proceso de análisis, diseño, desarrollo, pruebas y la puesta en producción. Se debe agregar adicionalmente dos horas de capacitación al personal que va a utilizar la funcionalidad. Esta mejora debe ser realizada por un analista programador junior con conocimiento en asp.net y bases de datos SQL Server. El costo por hora del analista programador se estima, según la página del Ministerio de Trabajo y Seguridad Social (2020), en 1.735 colones, para un total de 83.280 colones.

Los costos estimados para las recomendaciones se basaron en las indicaciones del ministerio de trabajo en cuanto a salarios mínimos, y tomando en cuenta la experiencia adquirida para poder realizar un cálculo de cuanto se debe pagar por hora en cada caso recomendación mencionada anteriormente.

CAPÍTULO VI: PROPUESTA

Análisis

Análisis detallado del software a desarrollar.

El sistema será desarrollado en 4 capas, donde una de ellas es una capa intermedia que comunica con todas. Por lo que lo detallado a continuación (los módulos) hacen referencia a la capa de Interfaz de Usuario, lo ingresado por el usuario en esa capa irá a la capa de Lógica de Negocio y luego a la capa de Datos y la información volverá en ese mismo orden hasta la Interfaz de Usuario. Todo lo anterior se comunica mediante una capa de entidades.

Módulo de inventario:

Se desarrollará un módulo de inventario, donde el usuario podrá consultar productos, actualizarlos, crearlos y eliminarlos. El usuario ingresará a este módulo y mediante una interfaz intuitiva podrá realizar todo el mantenimiento de productos en inventario.

Para la creación de este módulo, se utilizaron diferentes etiquetas con descripción de cada campo (Descripción producto, *stock*, etc.) y, a la par de estos, un campo de texto, en el que el usuario ingresará los campos requeridos, según la acción que va a realizar. Por ejemplo, si el usuario desea consultar un producto se habilitará el campo de texto de ID del producto, caso contrario si desea modificar un producto, ya que el ID no se podrá modificar.

Al final de cada proceso realizado, se deberá presionar un botón que ejecuta un método que procederá a realizar la acción especificada por el usuario.

Módulo de compras:

Este módulo permite realizar todo el mantenimiento de compras. Se podrán ver las compras realizadas, agregar nuevas compras, actualizarlas y eliminarlas. Al igual que todos los módulos, tiene diferentes etiquetas y campos de texto donde se ingresará la información requerida para llevar a cabo las diferentes acciones que permite el sistema. Una vez completados los campos de texto requeridos, se debe presionar el botón correspondiente para que se ejecuten los procesos internos y lograr realizar la acción.

Módulo de facturación:

El módulo de facturación permite ingresar los productos deseados por el cliente, la cantidad que llevan y se visualizará el precio total de la factura en tiempo real. Se podrá modificar la cantidad

de productos que lleva el cliente o eliminar un producto de la factura, siempre y cuando, sea antes de facturarlo. La información de la factura se irá viendo en tiempo real en un DataGridView. Una vez que factura todo, se presiona un botón donde consultará si realmente desea facturarlo, esto insertará en la base de datos la factura (pasando antes por las capas mencionadas al inicio de este capítulo) y rebajará del inventario los productos comprados por el cliente.

Módulo de reportes:

En este módulo, se pueden generar los diferentes reportes mediante un botón, en el que, posterior a presionarlo, se debe decidir dónde guardarlos. Los reportes en su programación irán a consultar unos procedimientos almacenados en la base de datos que serán los que nutren dichos reportes con información requerida para la toma de decisiones.

Módulo de seguridad:

En este módulo, se definirá el rol que tiene cada uno de los colaboradores en el sistema, dependiente de su rol, así tendrá posibilidad de tener más o menos permisos en la navegación. Por ejemplo, un colaborador no administrador no podrá eliminar productos del inventario. Un administrador será el único que pueda acceder al módulo de seguridad.

Mediante un formulario, se llenarán los campos con la información correspondiente de los usuarios que utilizarán el sistema, una vez que se presiona el botón correspondiente, se almacenará en la base de datos. A la hora de iniciar sesión, se obtendrá el tipo de rol que tiene el usuario y, con base en esa información, permitirá o no realizar algunas acciones en el sistema.

Módulo de planilla:

Una vez que se ingrese en el módulo de planilla, se puede observar en un DataGridView todos los colaboradores de la organización, además de datos como los días disponibles de vacaciones, su salario base y se podrá observar un histórico de pagos realizados a los empleados, así como información de los días solicitados de vacaciones. En este módulo, se verá un apartado donde se ingresa un trámite cada vez que un colaborador saca vacaciones.

Análisis detallado del hardware requerido.

Lo que se requiere como mínimo para el desarrollo de la aplicación es:

- Procesador Intel Core i5 de 7th Gen.
- 6 GB de memoria RAM.

- 256 GB de memoria interna.
- Microsoft Visual Studio 2019 instalado en el computador.
- Motor de base de datos SQL SERVER 2019 Developer Edition.
- Microsoft SQL SERVER Management Studio 18.
- Navegador de internet.

Para la utilización del aplicativo, se requiere un computador que haga la labor de una especie de servidor, que cuenta como mínimo con:

- Procesador Intel Core i3 de 8va Gen.
- Almacenamiento de 1 TB.
- Motor de base de datos SQL SERVER 2019 Developer Edition.
- Conexión a internet de 2 MB.
- Navegador de internet.

De manera implícita, se sobreentiende que se requiere contar con un *modem* y un *router* inalámbrico, debido a que las computadoras son portátiles y. para evitar costos en acondicionamiento, se va a utilizar la conexión mediante WiFi.

Análisis detallado de los elementos relacionados con las telecomunicaciones.

Se requiere conexión a internet de un mínimo de 2MB de descarga y 1MB de subida, de igual forma, se necesita un *modem* y un *router* inalámbrico que permita la conexión a un mínimo de 2 dispositivos de manera concurrente.

Descripción detallada de base de datos.

Se utilizará un motor de base de datos SQL SERVER 2019 Developer Edition, manejada mediante el sistema Microsoft SQL SERVER Management Studio 18. Lo anterior con licenciamiento gratuito. La capacidad de almacenamiento será de la totalidad disponible en disco duro en el computador, donde se alojará esta base de datos, es decir, de aproximadamente 1000GB, lo que proporciona un margen sumamente enorme de almacenamiento. Se estima que el almacenamiento soportará varios años.

Descripción detallada del personal requerido para el uso del sistema.

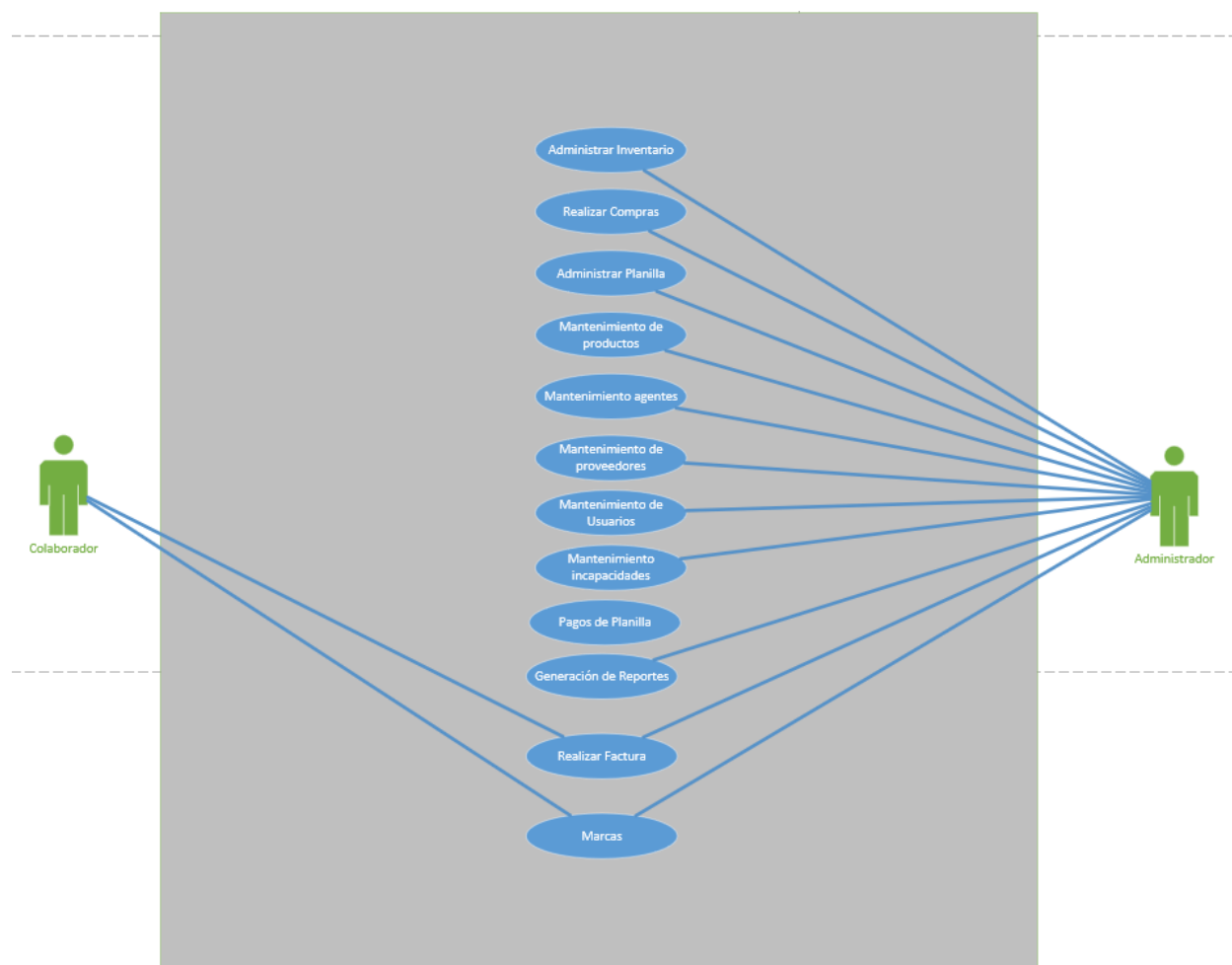
El sistema será utilizado por el 100 % del personal disponible de La Bodeguita. El personal deberá tener conocimientos en utilización del sistema operativo Windows en cualquiera de sus versiones más recientes, también deberán saber digitar eficientemente en la computadora. Un valor agregado debe ser conocer y haber utilizado un sistema de punto de venta, sin embargo, no es crítico este punto. Las capacitaciones necesarias serían:

- Navegación en el sistema de punto de ventas.
- Inicio de sesión en el sistema.
- Proceso para facturar productos.
- Roles y permisos.
- Manejo del módulo de pagos y solicitud de vacaciones a colaboradores.
- Procedimiento para realizar una compra de productos.
- Procedimientos para generar reportes.
- Procedimiento para realizar el mantenimiento en cada uno de los módulos.

Casos de uso

En esta sección, se van a mostrar los distintos casos de uso referente al prototipo del sistema unificado para La Bodeguita. Es importante conocer la comunicación que tienen los actores con el prototipo y sus funcionalidades para así comprender cual es la relación entre ellos y los requisitos funcionales.

Ilustración 6. Casos de uso



Fuente: Elaboración propia.

Tabla 2. CU01

Prototipo: Sistema La Bodeguita	
Número caso de uso: 1	Nombre del caso de uso: Administrar inventario
Fecha elaboración:	24/05/2020.
Descripción caso de uso:	Se realiza el mantenimiento del inventario de La Bodeguita.
Autor caso de uso:	Gustavo Hidalgo Sánchez.
Actores relacionados:	Usuario del sistema.

Precondiciones:	Se tuvo que realizar el <i>Login</i> al sistema con un usuario activo y se tuvo que ingresar al módulo de inventario.
Flujo básico del caso de uso	
Este caso de uso comienza cuando el actor ingresa al módulo de inventario. El sistema muestra un formulario donde para ingresar un nuevo producto debe presionar el botón “Nuevo Producto” (SF-1), si desea actualizarlo debe presionar el botón “Actualizar Producto” (SF-2), si desea eliminar un producto debe presionar el botón “Eliminar Producto” (SF-3) y si desea consultar un producto debe escribir en la barra buscadora (SF-4)	
Sub flujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF-1	Una vez el usuario presione el botón de “Nuevo Producto”, se habilitan los campos necesarios para la creación del producto, cuando los campos requeridos están completos se presiona el botón de Insertar y el sistema valida los datos a enviar (A-1) y luego registra la información en la base de datos.
SF-2	Se habilitará el <i>textbox</i> del ID del producto, para buscarlo se presiona el botón de búsqueda y se trae la información de base de datos. Posteriormente, se ingresa el cambio deseado y se presiona el botón de “Actualizar”, el sistema valida los datos a enviar (A-2).
SF-3	Se habilitará el <i>textbox</i> del ID del producto, para buscarlo se presiona el botón de búsqueda y se trae la información de base de datos. Posteriormente, se presiona el botón de “Eliminar”, el sistema elimina el dato correspondiente.
SF-4	El usuario escribe el nombre (Campo descripción en base de datos) del producto y presiona el botón “Enter”. El sistema trae las coincidencias de la búsqueda.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores.	

A-1	Se validan los campos enviados para la correcta inserción de productos, cada dato debe ser del tipo de dato correspondiente en base de datos, de lo contrario, mostrará mensaje de inconsistencia.
A-2	Se validan los campos enviados para la correcta actualización de productos, cada dato debe ser del tipo de dato correspondiente en base de datos, de lo contrario, mostrará mensaje de inconsistencia.
Requerimientos especiales	
No hay requerimientos especiales para este caso de uso.	
Post-condiciones	
Insertar, eliminar, actualizar o consultar productos del inventario.	

Fuente: Elaboración propia.

Tabla 3. CU02

Prototipo: Sistema La Bodeguita	
Número caso de uso: 2	Nombre del caso de uso: Realizar Factura
Fecha elaboración:	24/05/2020.
Descripción caso de uso:	Se realiza la facturación de productos y también visualización y eliminación de facturas.
Autor caso de uso:	Gustavo Hidalgo Sánchez.
Actores relacionados:	Usuario del sistema.
Precondiciones:	Se tuvo que realizar el <i>Login</i> al sistema con un usuario activo y se tuvo que ingresar al módulo de facturación.
Flujo básico del caso de uso	
Este caso de uso comienza cuando el actor ingresa al módulo de facturación. El sistema muestra en su pantalla principal una barra buscadora para buscar el producto por su descripción y al seleccionarla se agrega a un formulario donde se solicita la cantidad del producto, se presiona el botón agregar y se va acumulando en un <i>gridview</i> , una vez estén todos los productos que lleva el cliente, se presiona el botón “Facturar” el sistema consulta si desea o no factura, si presiona “Sí” (SF-1), y si presiona “No”, el sistema se devuelve. Si el usuario administrador desea eliminar una Factura (SF-2), debe presionar la pestaña de “Facturas”.	

Detallar el paso a paso del flujo básico.	
Sub flujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF-1	El sistema envía los datos a la base de datos y mediante un procedimiento almacenado se insertan los registros en las tablas correspondientes y se modifican en inventario.
SF-2	El sistema muestra un DataGridView con las facturas realizadas y podrá eliminarlas seleccionando el ID de la factura y luego presionando el botón de “Eliminar”.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores.	
No hay.	
Requerimientos especiales	
No existen requerimientos especiales.	
Post-condiciones	
Facturación, visualización y eliminación de facturas.	

Fuente: Elaboración propia.

Tabla 4. CU03

Prototipo: Sistema La Bodeguita	
Número caso de uso: 3	Nombre del caso de uso: Administrar Planilla
Fecha elaboración:	25/05/2020.
Descripción caso de uso:	Se podrá agregar usuarios y todo el mantenimiento correspondiente. También, se podrá realizar trámite de solicitudes de vacaciones y grabar en bitácora el pago de salarios.
Autor caso de uso:	Gustavo Hidalgo Sánchez.

Actores relacionados:	Usuario del sistema.
Precondiciones:	Se tuvo que realizar el <i>Login</i> al sistema con un usuario activo y se tuvo que ingresar al módulo de planilla
Flujo básico del caso de uso	
Este caso de uso comienza cuando el actor ingresa al módulo de planilla, de primera entrada verá un DataGridView con todos los usuarios y sus detalles. Aparece un botón de “Tramitar Vacaciones” (SF-1), otro botón de “Grabar Pago Salario” (SF-2) y otro de “Modificar Usuario (SF-3)”	
Sub flujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF-1	Se despliega una pantalla donde solicita el ID del Colaborador que saca vacaciones y cuantos días sacó. El sistema valida los días disponibles de vacaciones (A-1) y se presiona el botón “Registrar”.
SF-2	Se selecciona el colaborador a pagar, se indica en números enteros, la cantidad de días laborados (por defecto y para cálculos será 30), si faltó 1 día a laborar se coloca 29. Y se presiona el botón “Registrar Pago”.
SF-3	El usuario selecciona mediante el ID, el colaborador que desea modificar, ingresando este dato en el <i>textbox</i> y presionando el botón de “Buscar”. La información del colaborador aparece cargada en los <i>TextBox</i> donde modificará el campo que desee (Excepto el ID). Luego presiona el botón “Modificar Colaborador”.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores	
A-1	El sistema valida si la cantidad de días de vacaciones solicitados supera el disponible, de lo contrario muestra mensaje de advertencia.
Requerimientos especiales	
No hay requerimientos especiales.	
Post-condiciones	
Visualización de usuarios, registro de pagos, visualización de pagos, registro de usuarios y trámite de vacaciones.	

Fuente: Elaboración propia.

Tabla 5. CU04

Prototipo: Sistema La Bodeguita	
Número caso de uso: 4	Nombre del caso de uso: Realizar Compras.
Fecha elaboración:	24/05/2020.
Descripción caso de uso:	Módulo donde se realizan las compras de la organización.
Autor caso de uso:	Gustavo Hidalgo Sánchez.
Actores relacionados:	Usuario del sistema.
Precondiciones:	Se tuvo que realizar el <i>Login</i> al sistema con un usuario activo y se tuvo que ingresar al módulo de compras.
Flujo básico del caso de uso	
Este caso de uso comienza cuando el actor ingresa al módulo de compras. Se visualizarán botones para mantenimiento de “Proveedores” (SF-1), también para mantenimiento de “Agentes” (SF-2) y en la pantalla principal se verá el mantenimiento de “Compras” (SF-3)	
Sub flujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF-1	En esta pantalla se podrá agregar un Proveedor, también eliminar y modificar uno existente. Para agregar se presiona el botón de “Agregar”, para eliminar se presiona “Buscar” y luego el botón “Eliminar” y para Consultar se ingresa en la barra buscadora el Nombre y la tecla “Enter”.
SF-2	En esta pantalla se podrá agregar un Agente, también eliminar y modificar uno existente. Para agregar se presiona el botón de “Agregar” colocando todos los campos requeridos en pantalla, para eliminar se presiona “Buscar” y luego el botón “Eliminar” y para Consultar se ingresa en la barra buscadora el Nombre y la tecla “Enter”.
SF-3	Para las compras, se deben ingresar ciertos datos requeridos con información para la factura interna, ingresar los productos, cantidades, etc. Luego, se presiona el botón “Efectuar compra”.

	Para buscar una compra, se puede visualizar mediante el ID de la misma o ver la fecha efectuada. Para eliminarla será mediante el ID.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores	
No hay.	
Requerimientos especiales	
No hay requerimientos especiales.	
Post-condiciones	
Mantenimiento de compras, proveedores y agentes.	

Fuente: Elaboración propia.

Tabla 6. CU05

Prototipo: Sistema La Bodeguita	
Número Caso de Uso:	Nombre del Caso de Uso: Generar Incapacidades
Fecha elaboración:	24/05/2020
Descripción Caso de Uso:	Módulo donde se generan las incapacidades
Autor caso de uso:	Gustavo Hidalgo Sánchez
Actores relacionados:	Usuario del sistema.
Precondiciones:	Se tuvo que realizar el Login al sistema con un usuario activo y se tuvo que ingresar al módulo de incapacidades.
Flujo Básico del caso de uso	
Este caso de uso comienza cuando el actor ingresa al módulo de incapacidades. Se visualizarán dos calendarios donde se ingresan los parámetros, una vez se ingresan los datos solicitados se presiona el botón “Registrar Incapacidad” (SF-1), también se pueden ver las incapacidades en el link llamado “Ver incapacidades” (SF-2)	
Sub Flujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el Flujo Básico.	
SF-1	Al presionar el botón el sistema internamente toma los datos suministrados y los registra en la base de datos.

SF-2	Al ingresar a link, el sistema despliega una nueva pantalla donde se muestran todas las incapacidades registradas empezando por la última registrada.
Flujos Alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores	
No hay.	
Requerimientos especiales	
No hay requerimientos especiales.	
Post-Condiciones	
Registro de incapacidades o Visualización de incapacidades.	

Fuente: Elaboración propia.

Tabla 7. CU06

Prototipo: Sistema La Bodeguita	
Número Caso de Uso:	Nombre del Caso de Uso: Generar Reportes
Fecha elaboración:	24/05/2020
Descripción Caso de Uso:	Módulo donde se generan los reportes
Autor caso de uso:	Gustavo Hidalgo Sánchez
Actores relacionados:	Usuario del sistema.
Precondiciones:	Se tuvo que realizar el Login al sistema con un usuario activo y se tuvo que ingresar al módulo de reportes.
Flujo Básico del caso de uso	
Este caso de uso comienza cuando el actor ingresa al módulo de reportes. El actor elige el Reporte a generar presionando sobre el link correspondiente (SF-1).	
Sub Flujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el Flujo Básico.	
SF-1	Al ingresar a link, el sistema despliega una nueva pantalla donde se muestran los datos correspondientes al reporte y se presiona el botón de generar reporte para que se descarga un PDF con la información.

Flujos Alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores	
No hay.	
Requerimientos especiales	
No hay requerimientos especiales.	
Post-Condiciones	
Descarga del reporte.	

Fuente: Elaboración propia.

Tabla 8. CU07

Prototipo: Sistema La Bodeguita	
Número Caso de Uso:	Nombre del Caso de Uso: Realizar mantenimiento de Agentes
Fecha elaboración:	24/05/2020
Descripción Caso de Uso:	Módulo donde se da mantenimiento a los agentes
Autor caso de uso:	Gustavo Hidalgo Sánchez
Actores relacionados:	Usuario del sistema.
Precondiciones:	Se tuvo que realizar el Login al sistema con un usuario activo y se tuvo que ingresar al módulo de mantenimiento de agentes.
Flujo Básico del caso de uso	
Este caso de uso comienza cuando el actor ingresa al módulo de mantenimiento de agentes. El actor puede presionar el botón de “Insertar” (SF-1) para agregar los datos del agente y luego presionar “Ejecutar”. O también puede presionar el botón de “Actualizar” (SF-2) y buscar el agente deseado para posterior presionar el botón de “Ejecutar” y realizar la acción.	
Sub Flujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el Flujo Básico.	

SF-1	El actor completa los campos requeridos para insertar un nuevo agente y luego presiona el botón de “Ejecutar” y así hacer efectiva la creación del nuevo agente.
SF-2	El actor busca el agente deseado mediante su Identificador y posteriormente modifica los campos deseados, debe presionar el botón “Ejecutar” para hacer efectivo el cambio.
Flujos Alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores	
No hay.	
Requerimientos especiales	
No hay requerimientos especiales.	
Post-Condiciones	
Creación del agente, modificación del agente.	

Fuente: Elaboración propia.

Tabla 9. CU08

Prototipo: Sistema La Bodeguita	
Número Caso de Uso:	Nombre del Caso de Uso: Realizar mantenimiento de Proveedores
Fecha elaboración:	24/05/2020
Descripción Caso de Uso:	Módulo donde se da mantenimiento a los proveedores
Autor caso de uso:	Gustavo Hidalgo Sánchez
Actores relacionados:	Usuario del sistema.
Precondiciones:	Se tuvo que realizar el Login al sistema con un usuario activo y se tuvo que ingresar al módulo de mantenimiento de proveedores.
Flujo Básico del caso de uso	
Este caso de uso comienza cuando el actor ingresa al módulo de mantenimiento de proveedores. El actor puede presionar el botón de “Insertar” (SF-1) para agregar los datos del proveedor y luego presionar “Ejecutar”. O también puede presionar el botón de “Actualizar” (SF-2) y buscar el proveedor deseado para posterior presionar el botón de “Ejecutar” y realizar la acción. Y, por último, está la opción de eliminar (SF-3) un proveedor.	

Sub Flujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el Flujo Básico.	
SF-1	El actor completa los campos requeridos para insertar un nuevo proveedor y luego presiona el botón de “Ejecutar” y así hacer efectiva la creación del nuevo proveedor.
SF-2	El actor busca el proveedor deseado mediante su Identificador y posteriormente modifica los campos deseados, debe presionar el botón “Ejecutar” para hacer efectivo el cambio.
SF-3	El actor busca el proveedor que se desea eliminar mediante su identificador, posteriormente se presiona el botón de “Ejecutar” para hacer efectiva la eliminación.
Flujos Alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores	
No hay.	
Requerimientos especiales	
No hay requerimientos especiales.	
Post-Condiciones	
Creación del proveedor, modificación del proveedor, eliminación del proveedor.	

Fuente: Elaboración propia.

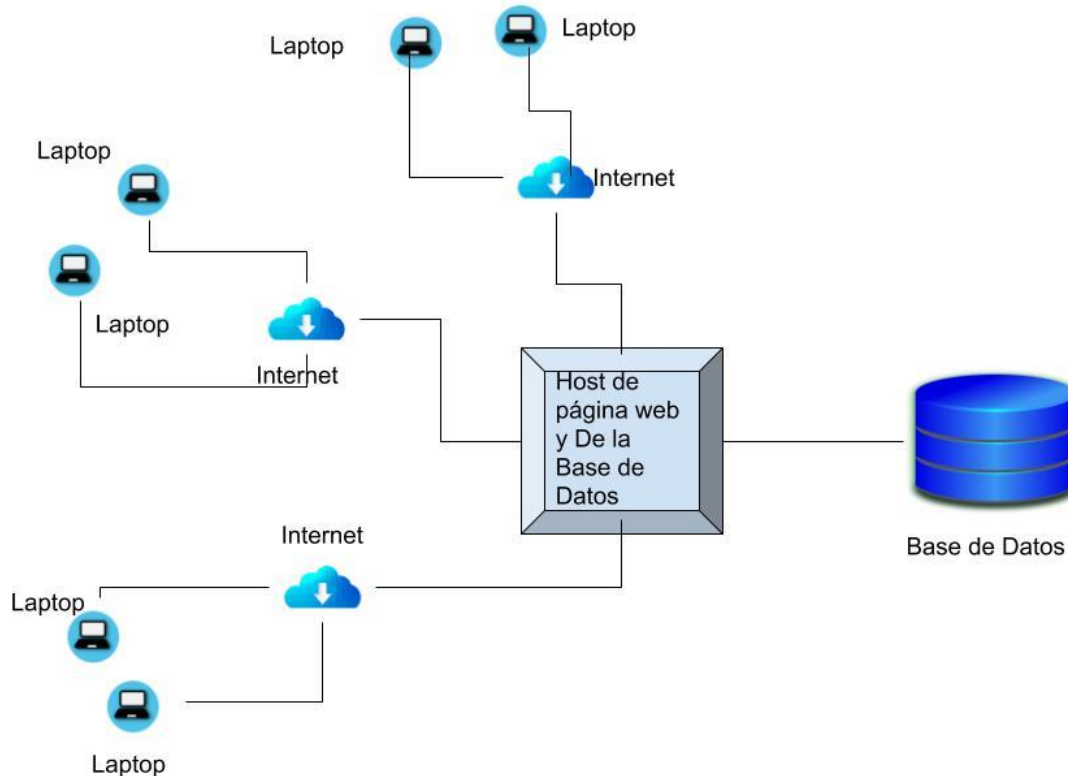
Diseño

En el siguiente apartado, se va a mostrar las arquitecturas que utiliza el prototipo, también los diseños que van a tener las diferentes interfaces, base de datos y los demás diseños para entender el funcionamiento del prototipo.

Arquitectura de sistema.

En la siguiente ilustración, se visualizarán cuáles son los elementos presentes en la arquitectura del sistema, en cuanto a hardware se refiere.

Ilustración 7. Arquitectura de Sistema.



Fuente: Elaboración propia.

Como se puede observar en la arquitectura anterior, cada una de las laptops que utilizan en la organización se va a conectar mediante internet a la computadora que va a simular un servidor, allí estará alojada la página web y la base de datos.

Arquitectura de software.

A continuación, se mostrará el diseño interno del prototipo, en los cuales se puede identificar los principales módulos, procesos y la estructura de capas utilizada:

Ilustración 8. Arquitectura de Software



Fuente: Elaboración propia.

Como vemos en la arquitectura anterior, se encuentra un apartado en el que se está alojado el motor de base de datos, que es donde toda la data de la organización estará reunida. Esta base de datos será gestionada únicamente mediante la capa de datos, pasando por la capa de lógica de negocio donde se aplican las reglas de la organización y posteriormente será mostrada mediante la capa de interfaz gráfica. Todos los procesos y la seguridad ocurren en el servidor de aplicaciones.

En la interfaz gráfica, se encuentran los diferentes módulos del prototipo, dentro de los módulos se encuentran los mantenimientos a las diferentes tablas, consultas, solicitudes de inserción de productos, eliminación, modificación, etc. Esta interfaz gráfica permite al usuario la interacción con la base de datos.

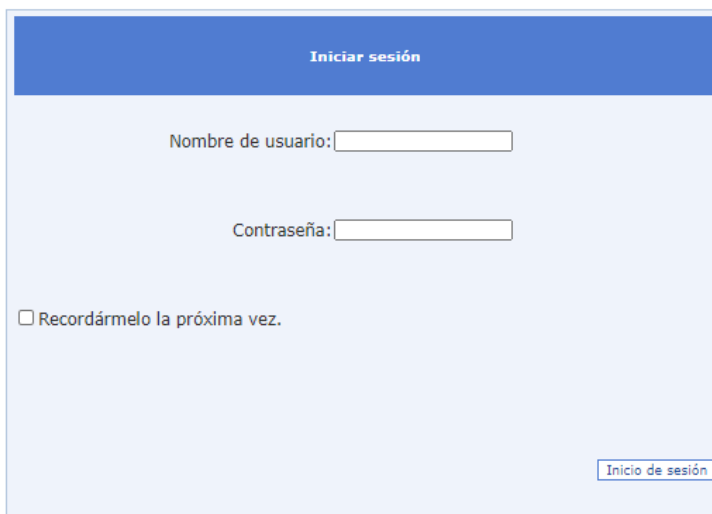
Diseño de interfaces.

A continuación, se muestra el diseño de las interfaces utilizadas en el prototipo funcional de La Bodeguita, en estos formularios, se puede observar el diseño de las pantallas, las cuales van a utilizar los colaboradores, la interfaz gráfica hace que sea fácil la comunicación del usuario y el

computador, ya que se intentó realizar un diseño intuitivo en las pantallas para favorecer su uso, y que los colaboradores no tuvieran problemas a la hora de realizar las gestiones.

Seguidamente, se mostrará la pantalla de Inicio de Sesión:

Ilustración 9. Inicio de Sesión



The image shows a login form with a blue header containing the text "Iniciar sesión". Below the header, there are two input fields: "Nombre de usuario:" and "Contraseña:". Below the password field, there is a checkbox labeled "Recordármelo la próxima vez.". At the bottom right, there is a button labeled "Inicio de sesión".

Fuente: Elaboración propia.

Al iniciar sesión, se podrá observar el marco principal del sistema, compuesto por un encabezado donde aparecen los nombres de los diferentes módulos del sistema y, en el pie de página, se encontrará el nombre de la organización. El colaborador únicamente puede ingresar a los módulos si su rol asociado lo permite.

Una vez que el usuario logre realizar el inicio de sesión correcto, se despliega la pantalla de Facturación de manera inmediata:

Ilustración 10. Facturación

The screenshot shows the 'Facturación' (Billing) module of the 'LA BODEGUITA' system. The header features the logo 'LA BODEGUITA Sistema Unificado' and navigation tabs for 'Facturación', 'Compras', 'Inventario', 'Planilla', and 'A Long Link Text'. The main content area is titled 'FACTURACIÓN' and contains a form with the following fields: 'Nombre Cliente', 'Id Producto', and 'Cantidad'. A 'Agregar Producto' button is located below the 'Cantidad' field. At the bottom of the form, there is a 'Factura Actual' section with a 'Completar Factura' button. The footer of the page displays the copyright notice: 'Copyright © 2020 - All Rights Reserved - LA BODEGUITA'.

Fuente: Elaboración propia.

Si el usuario desea ingresar al módulo de inventario, debe dar clic en la pestaña correspondiente y se despliega la interfaz de inventario, donde el colaborador con rol de administrador tendrá disponible todas las funcionalidades del módulo, es decir, puede crear, actualizar, eliminar y visualizar los productos de la organización. El módulo de inventario es el siguiente:

Ilustración 11. Inventario

	ID PRODUCTO	Descripción	Sucursal	Stock
Seleccionar	22	hielo	1	8,00
Seleccionar	22	hielo	2	0,00
Seleccionar	24	inventado	1	8,00
Seleccionar	24	inventado	2	9,00
Seleccionar	26	inventado	1	0,00
Seleccionar	26	inventado	2	0,00
Seleccionar	27	inventado	1	0,00
Seleccionar	27	inventado	2	0,00
Seleccionar	28	inventado	1	0,00
Seleccionar	28	inventado	2	0,00

Fuente: Elaboración propia.

En la siguiente imagen, se mostrará la interfaz de compras, donde únicamente el administrador tendrá acceso y podrá realizar las compras de la organización a los diferentes proveedores mediante los agentes. El módulo de compras se muestra a continuación:

Ilustración 12. Compras

Fuente: Elaboración propia.

En la imagen que se va a ver a continuación es donde el administrador podrá realizar todo el mantenimiento de proveedores, es decir, podrá crear un proveedor nuevo y actualizarlo:

Ilustración 13. Proveedores



The screenshot shows the 'MANTENIMIENTO PROVEEDORES' (Supplier Maintenance) interface. At the top, there is a navigation bar with 'LA BODEGUITA Sistema Unificado' and a menu with 'Facturación' (highlighted), 'Compras', 'Inventario', 'Planilla', and 'A Long Link Text'. The main content area contains a form with the following elements: 'Insertar' and 'Eliminar' buttons; 'Actualizar' button; 'Id Proveedor' and 'Descripción' input fields; a 'Submit' button; and a search section with 'Buscar' and 'Reiniciar' buttons. The footer contains the text 'Copyright © 2020 - All Rights Reserved - LA BODEGUITA'.

Fuente: Elaboración propia.

En el mantenimiento de sucursales se puede agregar una nueva sucursal en caso de que la organización lo desee y también se puede modificar la descripción de la sucursal:

Ilustración 14. Sucursales



The screenshot shows the 'MANTENIMIENTO SUCURSALES' (Branch Maintenance) interface. It has the same header and navigation bar as the previous screenshot. The main content area contains a form with the following elements: 'Insertar' and 'Eliminar' buttons; 'Actualizar' button; 'Id Sucursal' and 'Descripción' input fields; a 'Submit' button; and a search section with 'Buscar' and 'Reiniciar' buttons. The footer contains the text 'Copyright © 2020 - All Rights Reserved - LA BODEGUITA'.

Fuente: Elaboración propia.

La interfaz para realizar el mantenimiento de los agentes que pertenecen a los diferentes proveedores permitirá crear un agente y ligarlo a un proveedor de la organización, el módulo es el siguiente:

Ilustración 15. Agentes

The screenshot shows a web interface for 'LA BODEGUITA Sistema Unificado'. The main menu includes 'Facturación', 'Compras', 'Inventario', 'Planilla', and 'A Long Link Text'. The current page is titled 'MANTENIMIENTO AGENTES'. It features a form with the following elements:

- Buttons: Insertar, Eliminar, Actualizar, Submit, Buscar, Reiniciar.
- Input fields: Id Agente, Nombre, Primer apellido, Segundo apellido, Id Proveedor (dropdown), and Submit.

Copyright © 2020 - All Rights Reserved - LA BODEGUITA

Fuente: Elaboración propia.

Todos los colaboradores se van a encontrar registrados en una base de datos. Para darle mantenimiento correctamente a esta tabla, se diseñó una interfaz donde un usuario administrador podrá: crear nuevos colaboradores, actualizarlos o eliminarlos según se requiera, la interfaz es la siguiente:

Ilustración 16. Colaboradores

The screenshot shows a web interface for 'LA BODEGUITA Sistema Unificado'. The main menu includes 'Facturación', 'Compras', 'Inventario', 'Planilla', and 'A Long Link Text'. The current page is titled 'MANTENIMIENTO COLABORADORES'. It features a form with the following elements:

- Buttons: Insertar, Eliminar, Actualizar, Submit, Buscar, Reiniciar.
- Input fields: Id Colaborador, Contraseña, Estado, Nombre, Primer Apellido, Segundo Apellido, Vacaciones Disponibles, Id Sucursal, Id Rol, Salario Base, and Submit.

Copyright © 2020 - All Rights Reserved - LA BODEGUITA

Fuente: Elaboración propia.

En el submenú de planilla, se encontrará el *link* hacia la interfaz de registro de incapacidades. En esta pantalla, se puede registrar cuando un colaborador se encuentra incapacitado para que, posteriormente, cuando se realice el pago del salario el sistema pueda calcular el pago de ese día.

En un calendario, se registra la fecha cuando empieza a correr la incapacidad y en el otro calendario se registra cuando termina la incapacidad, si la incapacidad es de un solo día, se registra tanto la fecha inicial como la final el mismo día.

Ilustración 17. Incapacidades

LA BODEGUITA
Sistema Unificado

Facturación Compras Inventario Planilla Organización

INCAPACIDADES

FECHA INICIAL

Junio de 2020

lu.	ma.	mi.	ju.	vi.	sá.	do.
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

FECHA FINAL

Junio de 2020

lu.	ma.	mi.	ju.	vi.	sá.	do.
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Descripción Incapacidad

Id Colaborador 1000

Registrar Incapacidad

Copyright © 2020 - All Rights Reserved - LA BODEGUITA

Fuente: Elaboración propia.

Para el registro de vacaciones, se cuenta con una interfaz que posee dos calendarios, en uno se registra la fecha inicial de las vacaciones y en el otro la fecha final. Existe un campo de texto para agregar un comentario de las vacaciones, se debe definir el identificador del colaborador que goza las vacaciones. Únicamente, un usuario administrador puede ingresar a esta interfaz.

Ilustración 18. Vacaciones

LA BODEGUITA
Sistema Unificado

Facturación Compras Inventario Planilla Organización

REGISTRAR VACACIONES

FECHA INICIAL
Junio de 2020

lu.	ma.	mi.	ju.	vi.	sá.	do.
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

FECHA FINAL
Junio de 2020

lu.	ma.	mi.	ju.	vi.	sá.	do.
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Comentario

Id Colaborador Beneficiario: 1000

Registrar Vacaciones

Copyright © 2020 - All Rights Reserved - LA BODEGUITA

Fuente: Elaboración propia.

Para generar un lote de pagos de planilla, el usuario administrador puede ingresar a la interfaz de pago de planilla, donde se encuentra un campo de texto en el que se debe ingresar un nombre representativo para el lote de planilla, también se visualizan dos calendarios, en uno se registra el inicio del corte de pago y en el otro se registra el último día a pagar en el lote de planilla. En esta misma pantalla, se pueden visualizar los lotes de planilla ya realizados y, de igual forma, se puede ver el detalle del lote deseado.

Ilustración 19. Pagos planilla

LA BODEGUITA
Sistema Unificado

Facturación Compras Inventario Planilla Organización

PAGOS DE PLANILLA

Nombre de Planilla

Fecha Inicial
Junio de 2020

lu.	ma.	mi.	ju.	vi.	sá.	do.
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Fecha Final
Junio de 2020

lu.	ma.	mi.	ju.	vi.	sá.	do.
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Pagar Planilla Ver Lotes de Planilla Ver Detalle de Planilla Reiniciar Página

Copyright © 2020 - All Rights Reserved - LA BODEGUITA

Fuente: Elaboración propia.

Todos los colaboradores, cuando inicien y finalicen su jornada laboral, deben ingresar a la interfaz de “Marcas”, donde se permitirá el registro de estas. El sistema valida si, por ejemplo, cuando se registra una marca de salida, que no exista el mismo día una marca de salida, o si se va a marcar entrada que no exista una marca de entrada previamente. Si no existe una marca de entrada, no se puede marcar salida. La interfaz es la siguiente:

Ilustración 20. Marcas



Fuente: Elaboración propia.

Las diferentes interfaces para generar reportes tienen un estándar, es decir, un detalle estético muy similar, en estas pantallas, se podrá visualizar previo a la generación en PDF del reporte, los datos que tendrá el reporte solicitado. Ejemplo de una interfaz de reporte:

Ilustración 21. Reportes

Id Producto	Descripción	Ventas	Cantidad vendida en su unidad respectiva	Sucursal
89	Queso turnalba	€75,000.00	25,00	Desamparados Centro
91	Hielo 1kg	€22,500.00	12,00	Desamparados Centro

Fuente: Elaboración propia.

Llaves primarias	ID_AGENTE
Llaves secundarias	ID_PROVEEDOR

Fuente: Elaboración propia.

Tabla 8. Tabla Colaboradores.

Nombre de Tabla	COLABORADORES			
Descripción	Se almacenan datos de los colaboradores o usuarios del sistema.			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_COLABORADOR		INT	x	Identificador del colaborador
PASSWORD	50	VARCHAR	x	Contraseña del colaborador o usuario
ID_PERSONA		INT	x	Identificador de persona
VACACIONES_DISPONIBLE		INT	x	Número de días de vacaciones disponibles del colaborador
ID_SUCURSAL		INT	x	Identificador de la sucursal en la que trabaja el colaborador
ID_ROL		INT	x	Identificador del rol al que pertenece el colaborador
SALARIO_BASE		DECIMAL(9,2)	x	Salario base del colaborador
CEDULA		INT		Cédula del colaborador
FECHA_INICIO_CONTRATO		DATETIME		Fecha que inicia el contrato del colaborador
FECHA_FINAL_CONTRATO		DATETIME		Fecha que termina el contrato del colaborador.
Llaves primarias	ID_COLABORADOR			
Llaves secundarias	ID_SUCURSAL, ID_ROL			

Fuente: Elaboración propia.

Tabla 9. Tabla Compra Detalle

Nombre de Tabla	COMPRA_DETALLE			
Descripción	Se almacenan los detalles de las compras.			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_COMPRA		INT	x	Identificador de la compra
ID_PRODUCTO		INT	x	Identificador del producto comprado
CANTIDAD		DECIMAL (6,2)	x	Cantidad del producto comprado
PRECIO_COMPRA		DECIMAL(9,2)	x	Precio pagado al momento de la compra
PRECIO_TOTAL_VENTA		DECIMAL(9,2)	x	Precio para la venta
PRECIO_TOTAL_PRODUCTO		DECIMAL(9,2)		Resultado del precio por la cantidad

Llaves primarias	ID_COMPRA
Llaves secundarias	ID_PRODUCTO

Fuente: Elaboración propia.

Tabla 10. Tabla Compras

Nombre de Tabla	COMPRAS			
Descripción	Se almacenan datos generales sobre una compra realizada a un proveedor			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_COMPRA		INT	x	Identificador de la compra
ID_PROVEEDOR		INT	x	Identificador del proveedor
ID_AGENTE		INT	x	Identificador del agente
FECHA_COMPRA		DATETIME	x	Fecha de la compra
ID_COLABORADOR		INT	x	Identificador del colaborador que realiza la compra
TOTAL_PAGADO		DECIMAL(9,2)	x	Total pagado al proveedor por la compra
ID_SUCURSAL		INT	x	Identificador de la sucursal a la cual se destinará la compra
Llaves primarias	ID_COMPRA			
Llaves secundarias	ID_PROVEEDOR, ID_AGENTE, ID_COLABORADOR, ID_SUCURSAL			

Fuente: Elaboración propia.

Tabla 11. Tabla Factura Detalle

Nombre de Tabla	FACTURA_DETALLE			
Descripción	Se almacenan los detalles de las facturaciones a clientes			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_FACTURA		INT	x	Identificador de la factura
ID_PRODUCTO		INT	x	Identificador del producto
CANTIDAD		DECIMAL(10,3)	x	Cantidad del producto
PRECIO_A_LA_FECHA		DECIMAL(9,2)	x	Precio a la fecha de la compra
PRECIO_TOTAL_POR_PRODUCTO		DECIMAL(9,2)	x	Multiplicación de la cantidad por precio a la fecha
DESCUENTO		DECIMAL(9,2)		Descuento aplicado al producto
Llaves primarias	ID_FACTURA			
Llaves secundarias	ID_PRODUCTO			

Fuente: Elaboración propia.

Tabla 12. Facturas

Nombre de Tabla	FACTURAS			
Descripción	Se almacena los datos generales de las facturas realizadas			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_FACTURA		INT	x	Identificador de la factura
NOMBRE_CLIENTE	50	VARCHAR	x	Nombre del cliente comprador
FECHA		DATETIME	x	Fecha de creación de la factura
TOTAL_PAGADO		DECIMAL(9,2)	x	Total pagado por el cliente
ID_COLABORADOR		INT	x	Identificador del colaborador
ID_SUCURSAL		INT	x	Identificador de la sucursal
CORREO	50	VARCHAR		Correo al que se envía factura.
MONTO_IVA		DECIMAL(9,2)		Monto que corresponde a impuestos de venta.
Llaves primarias	ID_FACTURA			
Llaves secundarias	ID_COLABORADOR, ID_SUCURSAL			

Fuente: Elaboración propia.

Tabla 13. Tabla Planilla Pagos

Nombre de Tabla	PLANILLA_LOTE			
Descripción	Se almacena información de los pagos de planilla.			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_LOTE_PLANILLA		INT	x	Identificador de la planilla pagada
ID_COLABORADOR_APRUEBA		INT	x	Identificador del colaborador que aprueba gestión.
TOTAL_SALARIO_BRUTO		INT	x	Total pagado en salario bruto del lote.
NOMBRE_PLANILLA	50	VARCHAR	x	Nombre de la planilla
FECHA_EFECTUADA		DATETIME	x	Fecha de la gestión
FECHA_INI		DATE	x	Fecha inicio corte de planilla.
FECHA_FIN		DATE	x	Fecha fin del corte de la planilla.
TOTAL_SALARIO_NETO		DECIMAL(9,2)	x	Pago total en salario neto.
TOTAL_CARGAS_SOCIALES		DECIMAL(9,2)		Monto total en cargas sociales.
TOTAL_REBAJO_INCAPACIDAD		DECIMAL(9,2)		Monto rebajado en incapacidades.
TOTAL_MONTO_RENTA		DECIMAL(9,2)		Monto total de renta.

TOTAL_DIAS_INCAPACITADO		INT		Cantidad de días incapacitados del lote.
TOTAL_DIAS_AUSENTE		INT		Cantidad de días ausente sin pago.
TOTAL_DIAS_VACACIONES		INT		Cantidad de días de vacaciones solicitados en el lote.
Llaves primarias	ID_LOTE_PLANILLA			
Llaves secundarias	ID_COLABORADOR_APRUEBA			

Fuente: Elaboración propia.

Tabla 14. Vacaciones

Nombre de Tabla	VACACIONES			
Descripción	Se almacenan datos sobre las vacaciones que solicitan los colaboradores.			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_VACACIONES		INT	x	Identificador de vacaciones
ID_COLABORADOR_BENEFICIADO		INT	x	Identificador del colaborador que se beneficia con las vacaciones
ID_COLABORADOR_APRUEBA		INT	x	Identificador del colaborador que aprueba las vacaciones (realiza el trámite en el sistema)
DESCRIPCION	50	VARCHAR	x	Descripción de las vacaciones.
FECHA_INI		DATE	x	Fecha de inicio de periodo de vacaciones.
FECHA_FIN		DATE	x	Fecha de fin de periodo de vacaciones.
FECHA_SOLICITADO		DATE	x	Fecha de registro del trámite.
Llaves primarias	ID_VACACIONES			
Llaves secundarias	ID_COLABORADOR_BENEFICIADO, ID_COLABORADOR_APRUEBA			

Fuente: Elaboración propia.

Tabla 15. Tabla Productos.

Nombre de Tabla	PRODUCTOS			
Descripción	Se almacenan datos de los productos a la venta en la organización			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_PRODUCTO		INT	x	Identificador del producto
DESCRIPCION	50	VARCHAR	x	Descripción del producto
PRECIO_COMPRA		DECIMAL(9,2)	x	Precio de costo del producto

PRECIO_VENTA		DECIMAL(9,2)	x	Precio para la venta del producto
ESTADO	1	VARCHAR		Estado en el que se encuentra el producto, activo o inactivo.
ID_CATEGORIA	1	VARCHAR		Identificador de la categoría en impuestos del producto.
Llaves primarias	ID_PRODUCTO			
Llaves secundarias	ID_CATEGORIA			

Fuente: Elaboración propia.

Tabla 16. Proveedores

Nombre de Tabla	PROVEEDORES			
Descripción	Se almacenan datos de los proveedores del negocio			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_PROVEEDOR		INT	x	Identificador del proveedor
ID_PERSONA		INT	X	Identificador de tipo persona
Llaves primarias	ID_PROVEEDOR			
Llaves secundarias				

Fuente: Elaboración propia.

Tabla 17. Tabla Roles

Nombre de Tabla	ROLES			
Descripción	Se almacenan datos de los roles del sistema			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_ROL		INT	x	Identificador del rol
DESCRIPCION	50	INT	x	Descripción del Rol
Llaves primarias	ID_ROL			
Llaves secundarias				

Fuente: Elaboración propia.

Tabla 18. Tabla Stock.

Nombre de Tabla	STOCK			
Descripción	Se almacenan datos del stock de productos			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción

ID_PRODUCTO		INT	x	Identificador del producto
ID_SUCURSAL		INT	x	Identificador de la sucursal
STOCK		DECIMAL (6 , 2)	x	Cantidad en stock del producto en la sucursal correspondiente
Llaves primarias	ID_PRODUCTO			
Llaves secundarias	ID_SUCURSAL			

Fuente: Elaboración propia.

Tabla 19. Tabla Sucursales

Nombre de Tabla	SUCURSALES			
Descripción	Se almacenan datos de las sucursales de la organización			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_SUCURSAL		INT	x	Identificador de la sucursal
DESCRIPCION	50	VARCHAR	x	Descripción de la sucursal
Llaves primarias	ID_SUCURSAL			
Llaves secundarias				

Fuente: Elaboración propia.

Tabla 20. Planilla Lote Detalle

Nombre de Tabla	PLANILLA_LOTE_DETALLE			
Descripción	Se almacena el detalle de los PLANILLA_LOTE.			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_LOTE_PLANILLA		INT	x	Identificador del lote planilla
ID_COLABORADOR		INT	x	Identificador del colaborador
SALARIO_BRUTO		DECIMAL(9,2)	x	Salario bruto correspondiente
SALARIO_NETO		DECIMAL(9,2)	x	Salario neto a pagar
CARGAS_SOCIALES		DECIMAL(9,2)	x	Monto rebajado de cargas sociales
REBAJO_INCAPACIDAD		DECIMAL(9,2)	x	Monto por incapacidad
REBAJO_AUSENCIA		DECIMAL(9,2)	x	Rebajo por ausencias
MONTO_RENTA		DECIMAL(9,2)	x	Monto correspondiente a renta
DIAS_INCAPACITADO		INT		Cantidad de días incapacitado

DIAS_AUSENTE		INT		Cantidad de días ausente sin derecho a pago
PORCENTAJE_CARGAS_SOCIALES		FLOAT		Porcentaje descontado correspondiente a cargas sociales
DIAS_VACACIONES_SOLICITADOS		INT		Cantidad de días de vacaciones que tuvo el colaborador.
Llaves primarias				
Llaves secundarias	ID_LOTE_PLANILLA, ID_COLABORADOR			

Fuente: Elaboración propia.

Tabla 20. Incapacidad

INCAPACIDAD				
Nombre de Tabla				
Descripción	Se registran las incapacidades de los colaboradores.			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_INCAPACIDAD		INT	x	Identificador de incapacidad
DESCRIPCION	50	VARCHAR	x	Descripción de la incapacidad
FECHA_INI		DATE	x	Fecha inicio de incapacidad
FECHA_FIN		DATE	x	Fecha finaliza incapacidad
ID_COLABORADOR		INT	x	Identificador de colaborador incapacitado
Llaves primarias	ID_INCAPACIDAD			
Llaves secundarias	ID_COLABORADOR			

Fuente: Elaboración propia.

Tabla 21. Cargas Sociales

CARGAS_SOCIALES				
Nombre de Tabla				
Descripción	Se graban los impuestos de cargas sociales que se cobran en los salarios			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_CARGA_SOCIAL		INT	x	Identificador de carga social
CODIGO_CARGA_SOCIAL	10	VARCHAR	x	Código de carga social
DESCRIPCION_CARGA_SOCIAL	50	VARCHAR	x	Descripción de carga social

VALOR_CARGA_SOCIAL		FLOAT	x	Valor a rebajar de la carga social
ESTADO		INT	x	Estado de la carga social. Activo o inactivo
Llaves primarias	ID_CARGA_SOCIAL			
Llaves secundarias				

Fuente: Elaboración propia.

Tabla 22. Aguinaldos

Nombre de Tabla	AGUINALDOS			
Descripción	Se almacena la información de los aguinaldos pagados			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_AGUINALDO		INT	x	Identificador de aguinaldo
AÑO		DATE	x	Año correspondiente al aguinaldo
MONTO_AGUINALDO		DECIMAL(9,2)	x	Monto correspondiente de aguinaldo
ID_COLABORADOR		INT	x	Identificador del colaborador al que le corresponde el aguinaldo
FECHA_EFECTUADO		DATETIME	x	Fecha que se registra el pago del aguinaldo
Llaves primarias	ID_AGUINALDO			
Llaves secundarias	ID_COLABORADOR			

Fuente: Elaboración propia.

Tabla 21. Aguinaldo Acumulado

Nombre de Tabla	AGUINALDO_ACUMULADO			
Descripción	Se almacena la información de los aguinaldos que se va acumulando por mes			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_COLABORADOR		INT	x	Identificador del colaborador
FECHA		DATE	x	Fecha del pago del salario
MONTO		DECIMAL(9,2)	x	Monto correspondiente de aguinaldo del mes
Llaves primarias	ID_AGUINALDO			
Llaves secundarias	ID_COLABORADOR			

Fuente: Elaboración propia.

Tabla 224. Marcas

Nombre de Tabla	MARCAS			
Descripción	Se registran las marcas de los colaboradores.			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_COLABORADOR		INT	x	Identificador del colaborador
TIPO_MARCA		INT	x	Corresponde al tipo de marca. 1 = Entrada y 2 = Salida
FECHAMMDDYYYY		DATE	x	Corresponde al día, mes y año de la marca
HORA_MARCA		DATETIME		Fecha con la hora del registro
Llaves primarias				
Llaves secundarias	ID_COLABORADOR			

Fuente: Elaboración propia.

Tabla 23. Teléfono Agente

Nombre de Tabla	TELEFONO_AGENTE			
Descripción	Se registran las marcas de los colaboradores.			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_AGENTE		INT	x	Identificador del agente
NUMERO_TELEFONO		INT	x	Número telefónico del agente
Llaves primarias				
Llaves secundarias	ID_AGENTE			

Fuente: Elaboración propia.

Tabla 24. Teléfono Colaborador

Nombre de Tabla	TELEFONO_COLABORADOR			
Descripción	Se registran las marcas de los colaboradores.			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_COLABORADOR		INT	x	Identificador del colaborador
NUMERO_TELEFONO		INT	x	Número telefónico del colaborador
Llaves primarias				
Llaves secundarias	ID_COLABORADOR			

Fuente: Elaboración propia.

Tabla 25. Personas

Nombre de Tabla	PERSONAS			
Descripción	Se registra los datos del tipo persona			
Campo	Tamaño	Tipo de Dato	No nulo	Descripción
ID_PERSONA		INT	x	Identificador del colaborador
NOMBRE	50	VARCHAR	x	Nombre de persona
APELLIDO1	50	VARCHAR		Primer Apellido
APELLIDO2	50	VARCHAR		Segundo Apellido
ESTADO		INT	x	Estado de la persona
Llaves primarias				
Llaves secundarias	ID_COLABORADOR			

Fuente: Elaboración propia.

Diseño de procesos.

Diagrama productos.

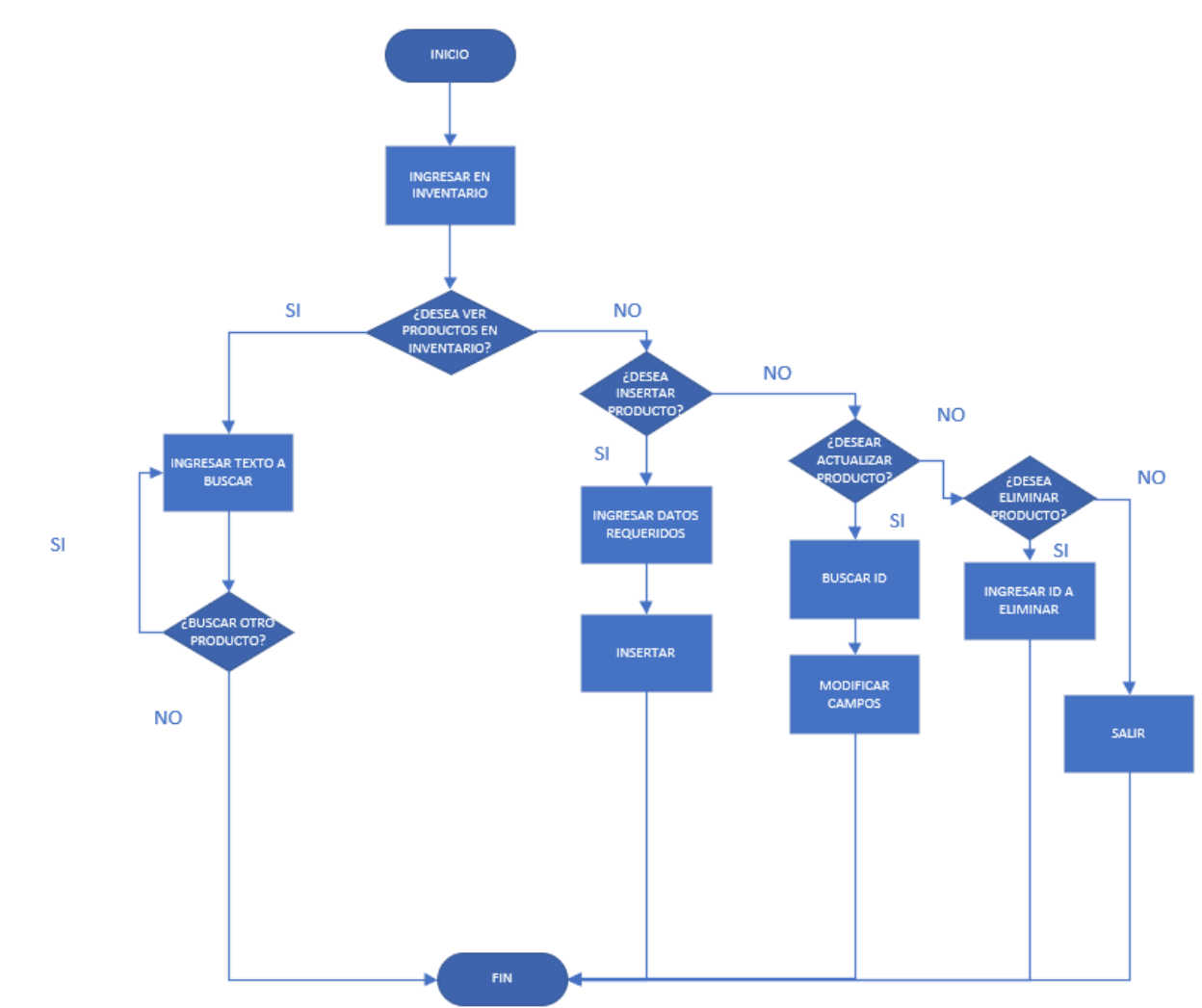
En los procesos relacionados con los productos, se podrá insertar, modificar, eliminar y buscar los productos con los que cuenta la organización. Para realizar una inserción, se requieren todos los datos del producto exceptuando el identificador, el cual es generado automáticamente por el motor de base de datos.

Para una modificación, se quiere ingresar el identificador del producto, el cual se puede buscar en la barra de búsqueda y presionar el botón de “Buscar”. Posteriormente, los datos del producto se cargarán en los campos de texto para realizar la modificación. Una vez que se modifican los campos deseados se presiona el botón “Submit”.

En caso de que se desee eliminar un producto, se debe buscar el producto mediante su identificador, esperar que se carguen los datos en los campos de texto y presionar el botón de “Submit”.

Por último, para buscar un producto se debe poner el nombre del producto en la barra buscadora y presionar el botón de “Buscar”.

Ilustración 23. Diagrama Productos.



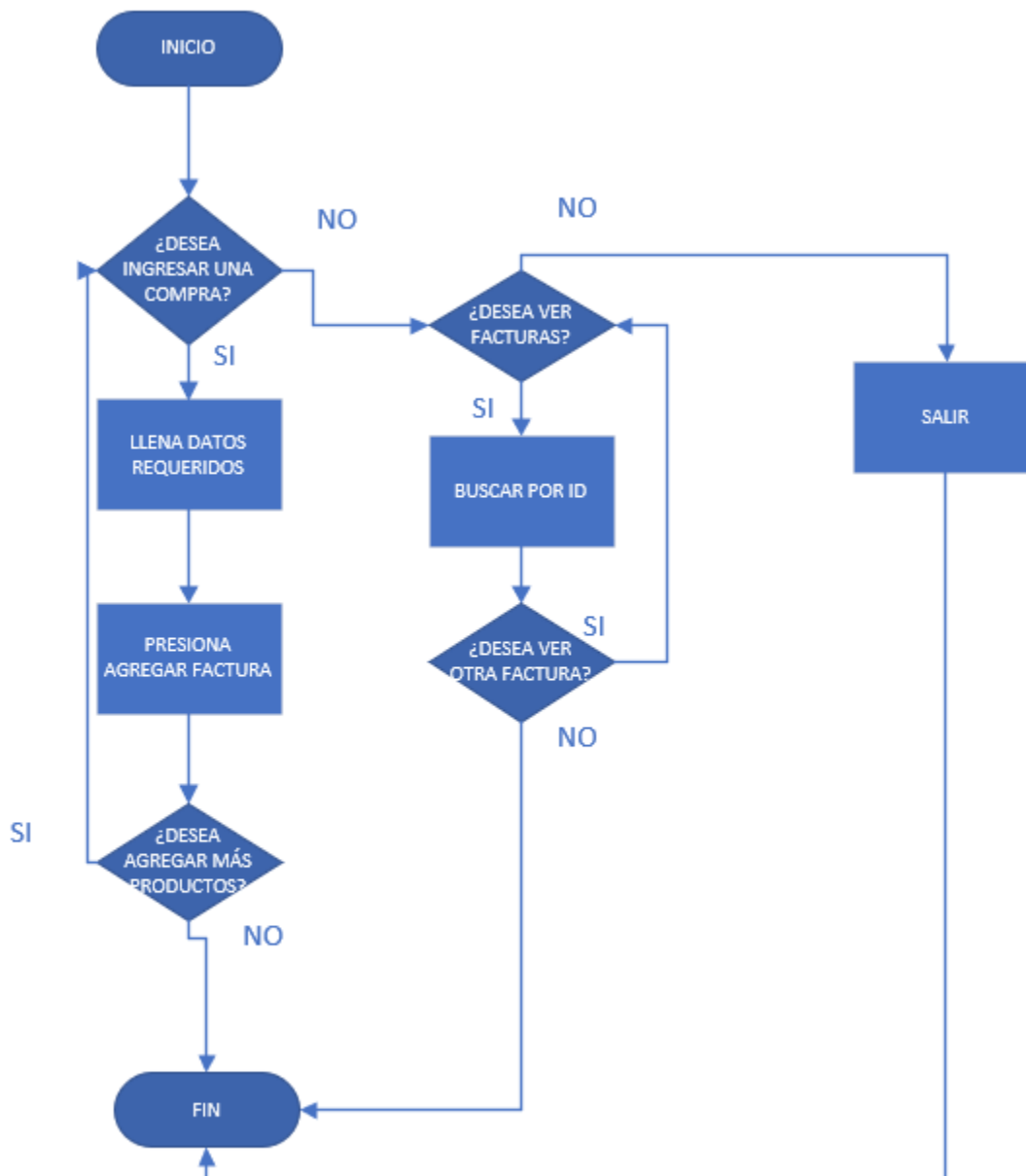
Fuente: Elaboración propia.

Diagrama de compras.

El proceso de compras consiste en buscar los productos mediante su identificador e irlos añadiendo a la factura. Una vez que se definan los productos, sus respectivas cantidades y precios, se efectuará la factura presionando el botón de “Generar Factura”.

En caso de que se desee buscar una factura, se debe introducir el identificador de la factura y, posteriormente, presionar el botón de “Buscar”.

Ilustración 24. Diagrama Compras.

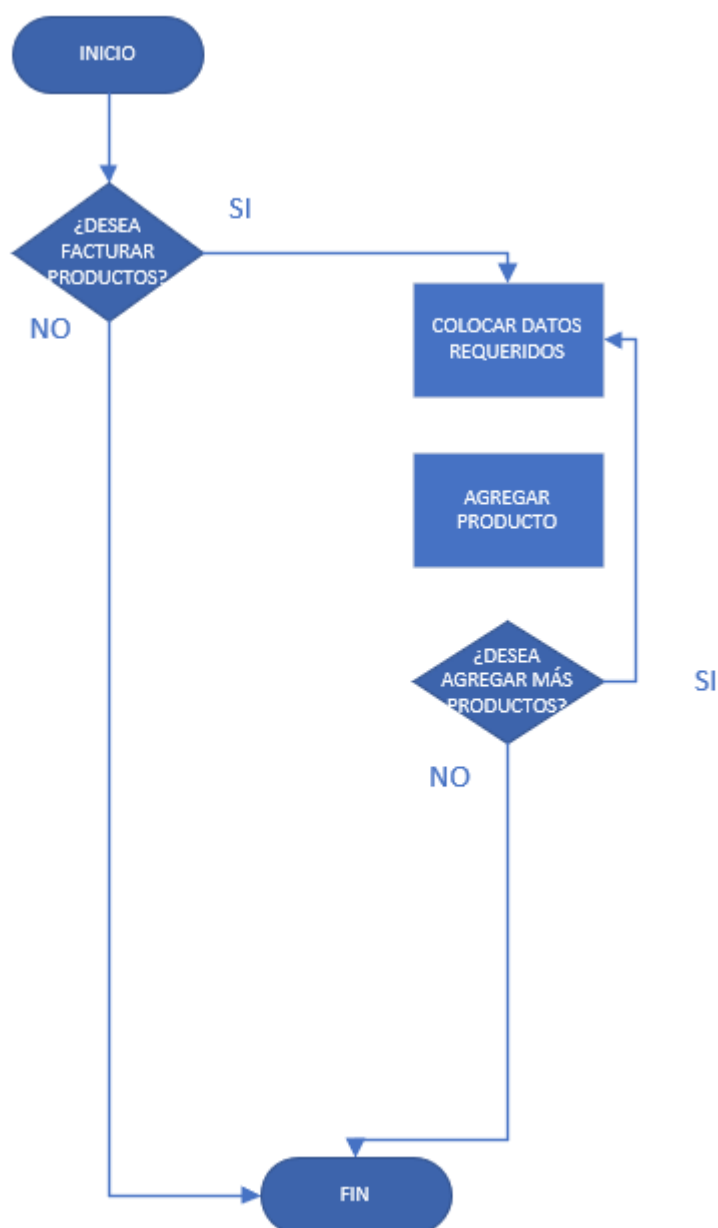


Fuente: Elaboración propia.

Diagrama de facturación.

En el proceso de facturación, el sistema solicitará el nombre del cliente, su correo electrónico en caso de necesitar factura en el correo y los productos que conforman la factura junto con sus respectivas cantidades solicitadas. Una vez que se tengan todos los datos requeridos, se presiona el botón “Facturar” para hacer efectiva la factura.

Ilustración 25. Diagrama Facturación.



Fuente: Elaboración propia.

Diagrama de pagos de planilla.

En el proceso de pago a la planilla, se debe ingresar a la pantalla correspondiente con el mismo nombre y seleccionar el rango de fechas que se van a considerar en el lote y presionar el botón de generar pago de planilla.

Ilustración 26. Diagrama Pago Planilla.



Fuente: Elaboración propia.

Diagrama de planilla vacaciones.

Si el usuario administrador desea registrar las vacaciones de otro colaborador o las de él mismo, se debe ingresar a la pantalla de planilla vacaciones, donde se ingresará el rango de fechas en las que se van a gozar las vacaciones, y definir el identificador del colaborador beneficiado. Posteriormente, se presiona el botón “Registrar Vacaciones” para terminar la operación.

Ilustración 27. Diagrama Planilla Vacaciones.

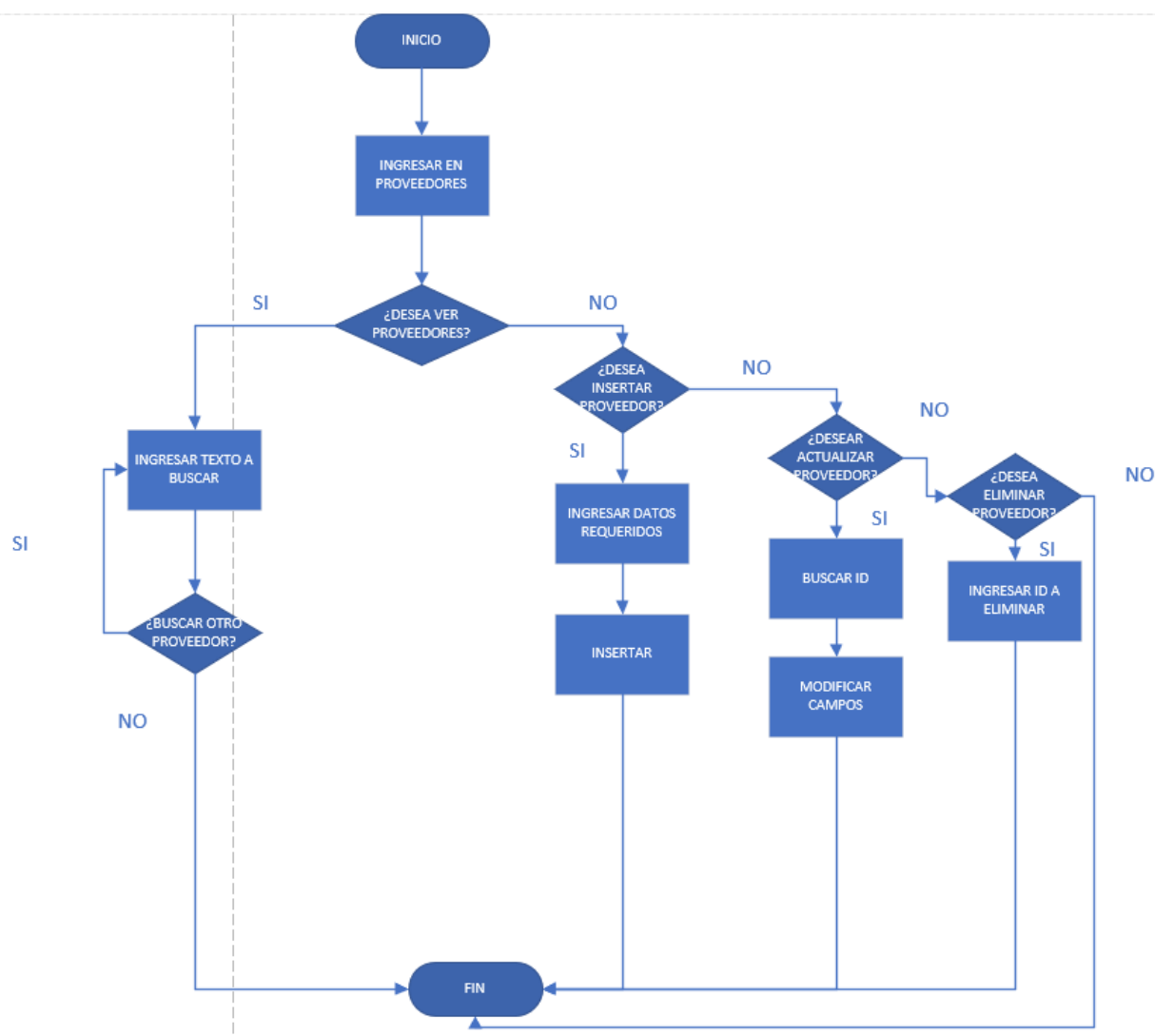


Fuente: Elaboración propia.

Diagrama de proveedores.

Al ingresar al módulo de proveedores, se pueden visualizar los diferentes proveedores que tiene la organización, de igual manera, se puede crear un nuevo proveedor, actualizar uno existente o eliminarlo.

Ilustración 28. Diagrama Proveedores.

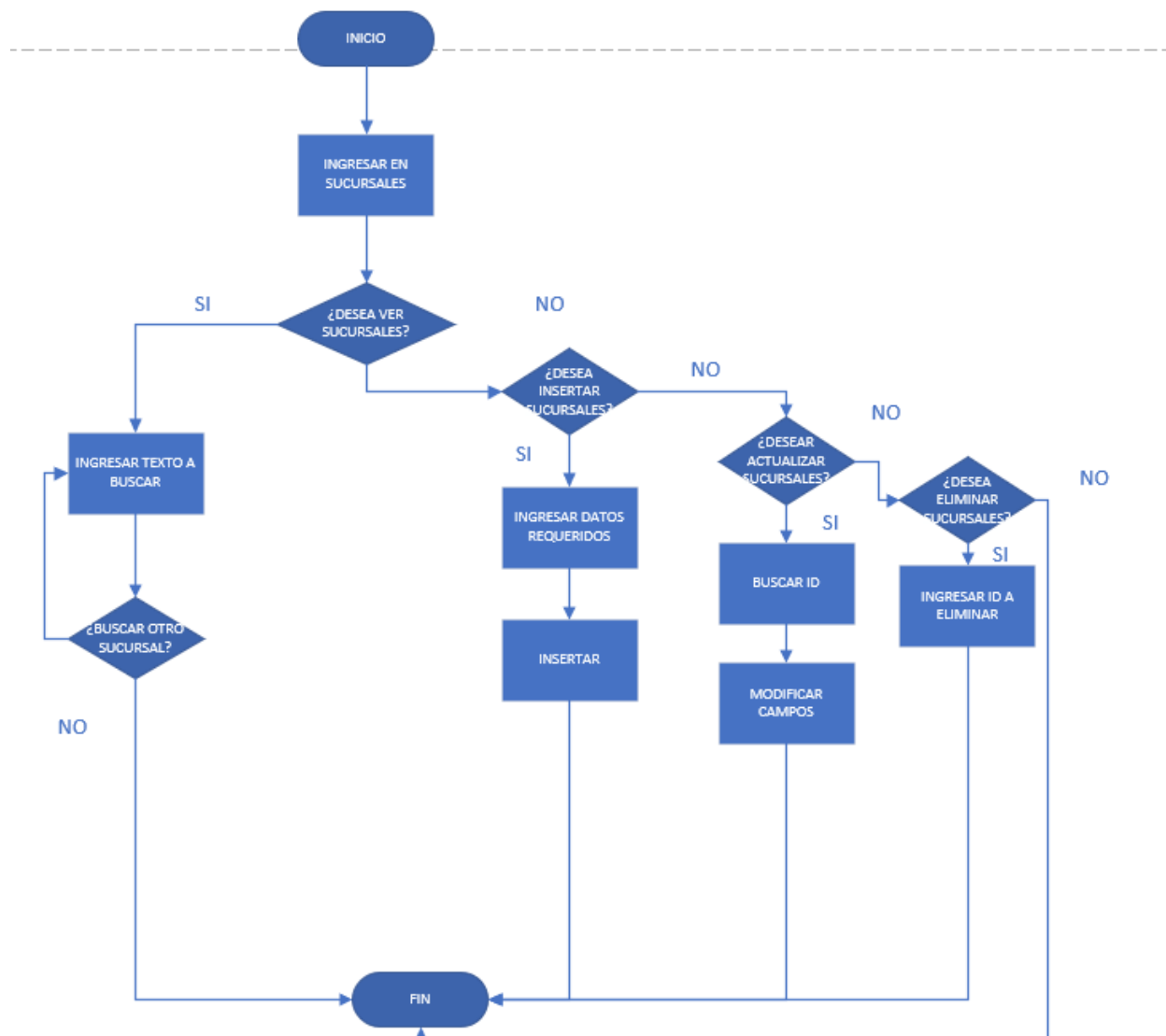


Fuente: Elaboración propia.

Diagrama de sucursales.

El proceso de sucursales, incluye la creación, actualización y eliminación. Asimismo, se podrán visualizar las sucursales con las que cuenta la organización.

Ilustración 29. Diagrama Sucursales.

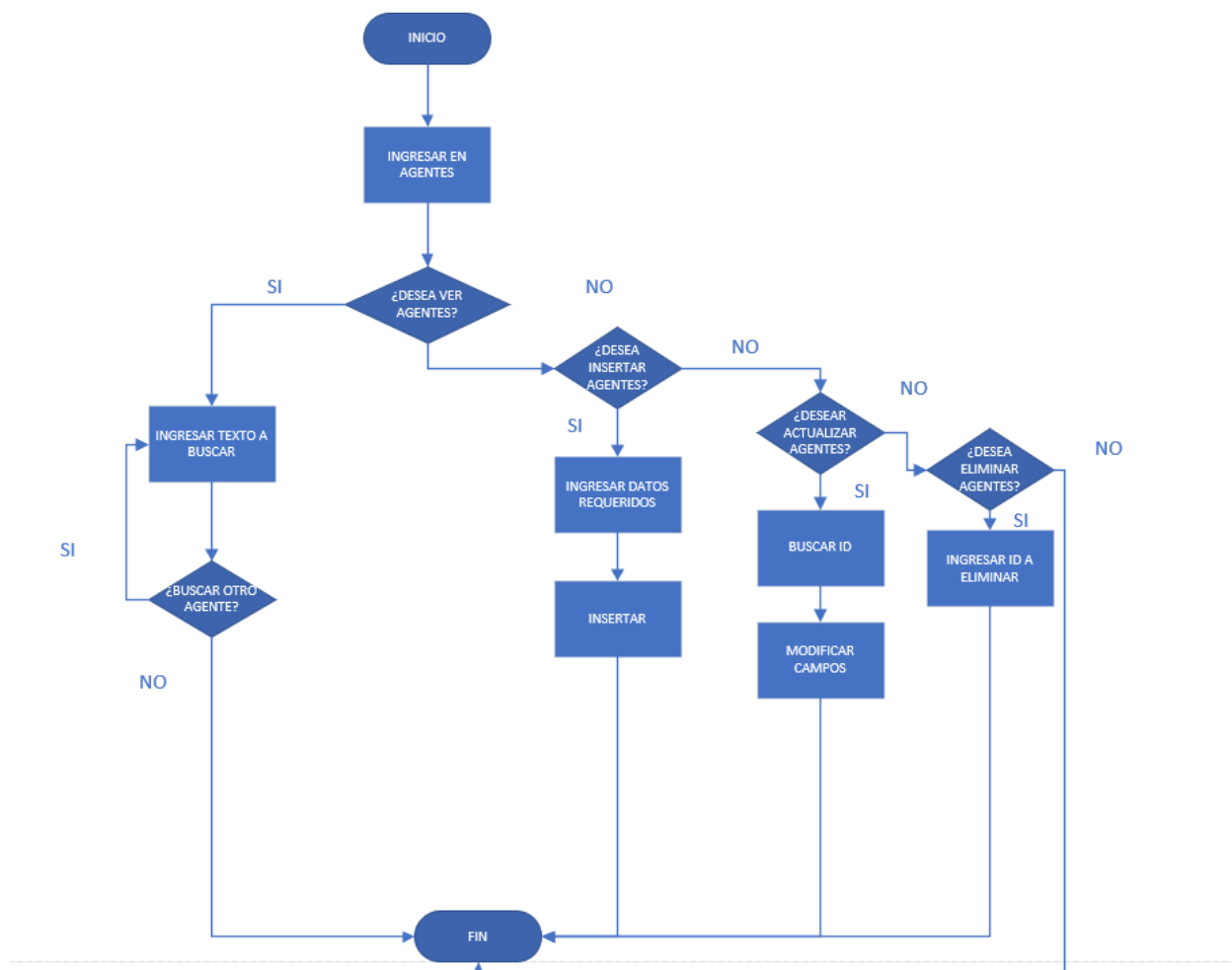


Fuente: Elaboración propia.

Diagrama de agentes.

El mantenimiento de agentes permite crear nuevos agentes y relacionarlos a un proveedor. También permite actualizar un agente ya existen y eliminarlo en caso de ser requerido. En este proceso, también está incluida la opción de ver los agentes con los que trata la organización.

Ilustración 30. Diagrama Agentes.

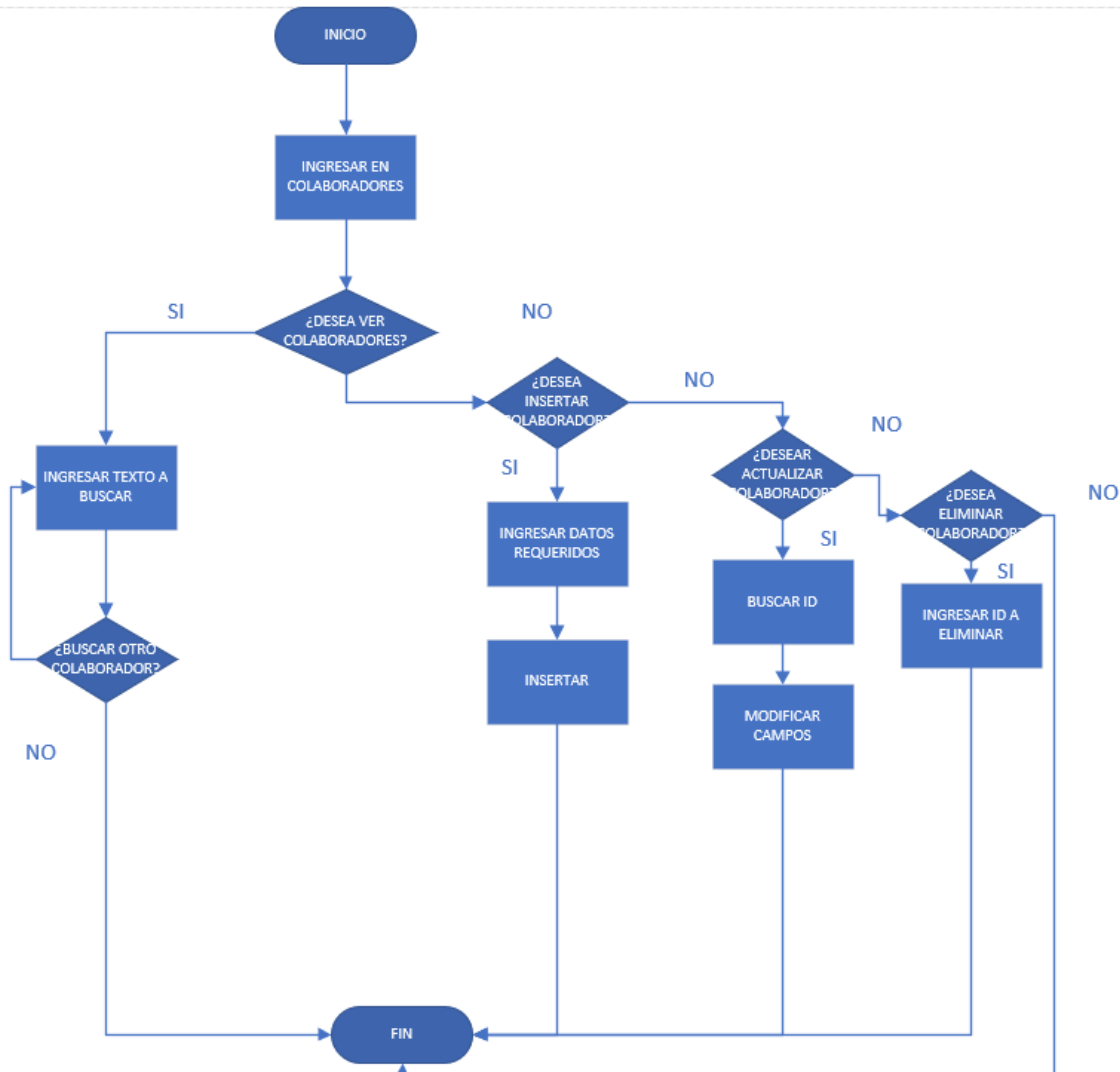


Fuente: Elaboración propia.

Diagrama de colaboradores.

El mantenimiento de colaboradores permite visualizar a todos los empleados de la organización, también facilita la creación de nuevos colaboradores, su actualización y eliminación.

Ilustración 31. Diagrama Colaboradores.



Fuente: Elaboración propia.

Diseño de salidas.

En esta sección, se van a mostrar las principales interfaces donde se encuentran las salidas de datos del prototipo, se van a mostrar los diseños de reportes para mostrar la información y otras pantallas en las que se encuentran salidas de datos.

En el diseño de las interfaces y para facilitar la lectura, se utilizaron vistas de datos, conocidos en la herramienta Visual Studio como un “GridView”, en los que se llenan de información

proveniente de la base de datos para ser mostrada por pantalla. A continuación, se mostrará una imagen del diseño de salidas de la vista de incapacidades:

Ilustración 32. Salida Incapacidades

Id Incapacidad	Descripción	Fecha Inicio	Fecha Final	Id Colaborador
6	GOLFES VARIOS	01/06/2020 0:00:00	01/06/2020 0:00:00	1000

Fuente: Elaboración propia.

Otro de los diseños de salida se puede ejemplificar en la interfaz de vista de vacaciones, donde se muestran todas las vacaciones registradas para los colaboradores de la organización:

Ilustración 33. Salida Vacaciones

Id Vacaciones	Descripción	Fecha Inicio	Fecha Final	Fecha Registro	Colaborador Beneficiario	Colaborador que registró
1	Vacaciones obligatorias	02/06/2020 0:00:00	02/06/2020 0:00:00	27/06/2020 16:27:08	1000	1000
4		29/06/2020 0:00:00	29/06/2020 0:00:00	29/06/2020 21:00:19	1000	1000
5	pruebas sacar vacaciones	19/06/2020 0:00:00	20/06/2020 0:00:00	29/06/2020 21:00:47	1110	1000
6	asdas	25/06/2020 0:00:00	26/06/2020 0:00:00	29/06/2020 21:05:05	1110	1000
7	Viaje a USA	30/06/2020 0:00:00	01/07/2020 0:00:00	30/06/2020 1:07:34	1120	1000
8	45651	13/06/2020 0:00:00	13/06/2020 0:00:00	30/06/2020 1:12:47	1110	1000
9	fx	20/06/2020 0:00:00	21/06/2020 0:00:00	30/06/2020 1:14:25	1110	1000
10	ct	02/06/2020 0:00:00	03/06/2020 0:00:00	30/06/2020 1:14:33	1120	1000
11	prueba	11/06/2020 0:00:00	12/06/2020 0:00:00	30/06/2020 1:14:58	1120	1000

Fuente: Elaboración propia.

La siguiente imagen mostrará el diseño de salida de la información en reporte de los productos más vendidos filtrado por sucursal:

Ilustración 34. Salida Reportes

LA BODEGUITA
Sistema Unificado

Facturación Compras Inventario Planilla Organización

REPORTE PRODUCTOS MÁS VENDIDOS

Sucursal: Desamparados Centro

Junio de 2020

lu.	ma.	mi.	ju.	vi.	sá.	do.
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Junio de 2020

lu.	ma.	mi.	ju.	vi.	sá.	do.
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Generar Reporte

Id Producto	Descripción	Ventas	Cantidad vendida en su unidad respectiva	Sucursal
89	Queso turrilba	€75.000,00	25,00	Desamparados Centro
91	Hielo 1kg	€22.500,00	12,00	Desamparados Centro

Copyright © 2020 - All Rights Reserved - LA BODEGUITA

Fuente: Elaboración propia.

La siguiente ilustración muestra el diseño de salida que tiene la interfaz de pago de planilla, donde se logran visualizar los diferentes lotes de planilla que se han realizado en la organización:

Ilustración 35. Salida Pago de Planilla

LA BODEGUITA
Sistema Unificado

Facturación Compras Inventario Planilla Organización

PAGOS DE PLANILLA

Nombre de Planilla:

Fecha Inicial: Junio de 2020

lu.	ma.	mi.	ju.	vi.	sá.	do.
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Fecha Final: Junio de 2020

lu.	ma.	mi.	ju.	vi.	sá.	do.
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Pagar Planilla Ver Lotes de Planilla Ver Detalle de Planilla Reiniciar Página

ID_LOTE_PLANILLA	TOTAL_SALARIO_BRUTO	NOMBRE_PLANILLA	FECHA_EFECTUADA	FECHA_INI	FECHA_FIN	ID_COLABORADOR_APRUEBA	TOTAL_SALARIO_NETO	TOTAL_CARGAS_SOCIALES	TOTAL_REBAJO_INCAPACIDAD	TOTAL_MONTO_RENTA	TOTAL_DIAS_INCAPACITADO	TOTAL_DIAS_INCAPACITADO
46	8763330	Nombre de Planilla	28/06/2020 21:11:10	01/06/2020 0:00:00	10/06/2020 0:00:00	1000	6135299	-922249	1033333	1725782	1	1
47	8763330	Nombre de Planilla	28/06/2020 21:11:22	01/06/2020 0:00:00	10/06/2020 0:00:00	1000	6135299	-922249	1033333	1725782	1	1
48	8763330	Nombre de Planilla	28/06/2020 21:11:27	01/06/2020 0:00:00	10/06/2020 0:00:00	1000	6135299	-922249	1033333	1725782	1	1
49	0	Nombre de Planilla	28/06/2020 21:24:52	01/01/0001 0:00:00	01/01/0001 0:00:00	1000	0	0	0	0	0	1
50	0	Nombre de Planilla	28/06/2020 21:36:21	01/01/0001 0:00:00	01/01/0001 0:00:00	1000	0	0	0	0	0	1

Copyright © 2020 - All Rights Reserved - LA BODEGUITA

Fuente: Elaboración propia.

Otro ejemplo de un diseño de salida, pero un poco diferente, es el de la interfaz de productos, en esta, al consultar un producto en pantalla, puede realizarse una carga de datos en campos de

texto para su modificación, muy diferente a los anteriores que se carga en un “GridView”.
Ilustración de la salida:

Ilustración 36. Salida Productos

ID PRODUCTO	Descripción	PRECIO COMPRA	PRECIO VENTA	Sucursal	Stock
86	Arroz con pollo	1500	2500	1	8976,00
86	Arroz con pollo	1500	2500	2	10,00
92	Coca Cola 1L Retornable	1000	1250	1	9966,00
92	Coca Cola 1L Retornable	1000	1250	2	0,00
92	Coca Cola 1L Retornable	1000	1250	3	0,00
92	Coca Cola 1L Retornable	1000	1250	4	0,00
92	Coca Cola 1L Retornable	1000	1250	5	0,00
93	Consome de Pollo unid	75	100	1	5499,00
93	Consome de Pollo unid	75	100	2	0,00
93	Consome de Pollo unid	75	100	3	0,00

Fuente: Elaboración propia.

Uno de los diseños de interfaz que mezcla varios tipos de salidas es el módulo de facturación. En la siguiente ilustración, se puede visualizar la salida mediante campos de textos, que son cargados provenientes de la base de datos y también se ve la salida de información por medio del “GridView”.

Ilustración 37. Salida Facturación

Acción	ID_PRODUCTO	DESCRIPCION	PRECIO_VENTA	STOCK	CATEGORIA	PORCENTAJE IMPUESTO
Seleccionar	90	Papa	1500	9911,40	Canasta básica	1,00

Factura
Total a pagar: €6000

Paga con:
Completar Factura

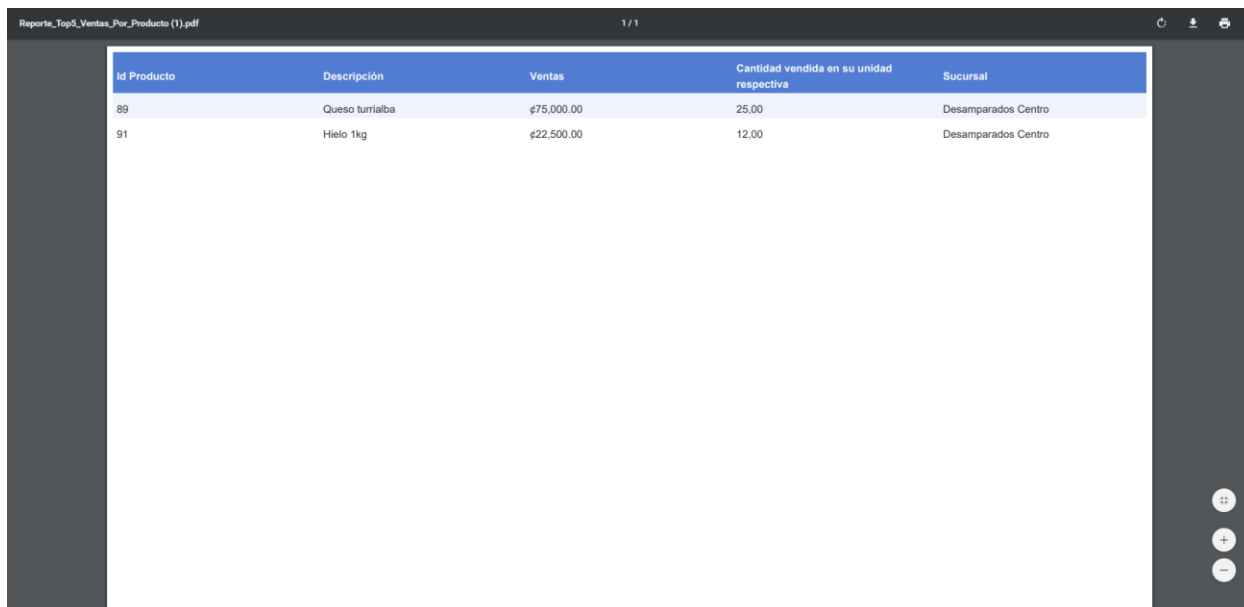
Acción	Id Producto	Descripción	Cantidad	Precio	Subtotal	Categoría IMP	Porcentaje Impuestos
Borrar Registro	89	Queso turrialba	2	3000	6000	Canasta básica	1,00

Copyright © 2020 - All Rights Reserved - LA BODEGUITA

Fuente: Elaboración propia.

El otro tipo de salidas que ofrece el sistema es mediante un archivo .PDF generado en los diferentes reportes, en este formato, se muestra una tabla con la información que se generó a partir de una consulta a base de datos y, posteriormente, se descarga al computador.

Ilustración 38. Salida Archivo PDF



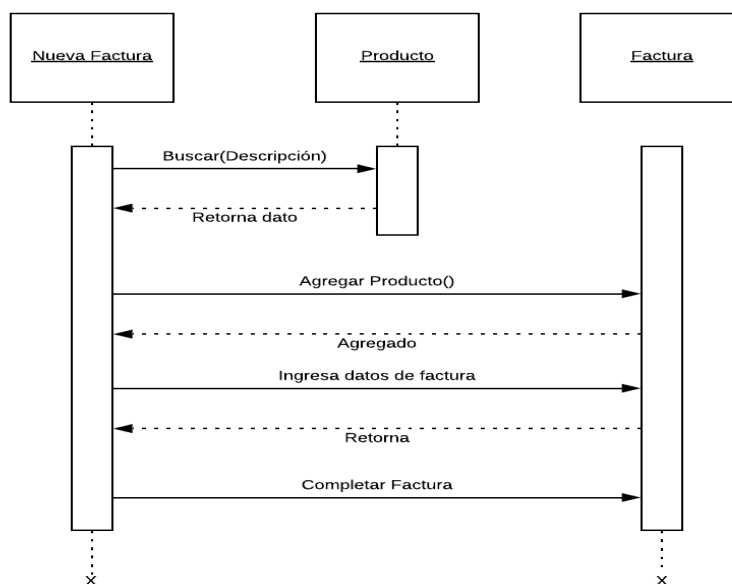
Id Producto	Descripción	Ventas	Cantidad vendida en su unidad respectiva	Sucursal
89	Queso turrialba	₡75.000.00	25,00	Desamparados Centro
91	Hielo 1kg	₡22.500.00	12,00	Desamparados Centro

Fuente: Elaboración propia.

Diagramas UML

A continuación, se muestran los diagramas de secuencia de diferentes procesos que se realizan en el prototipo. La siguiente ilustración muestra el diagrama de secuencia para facturar:

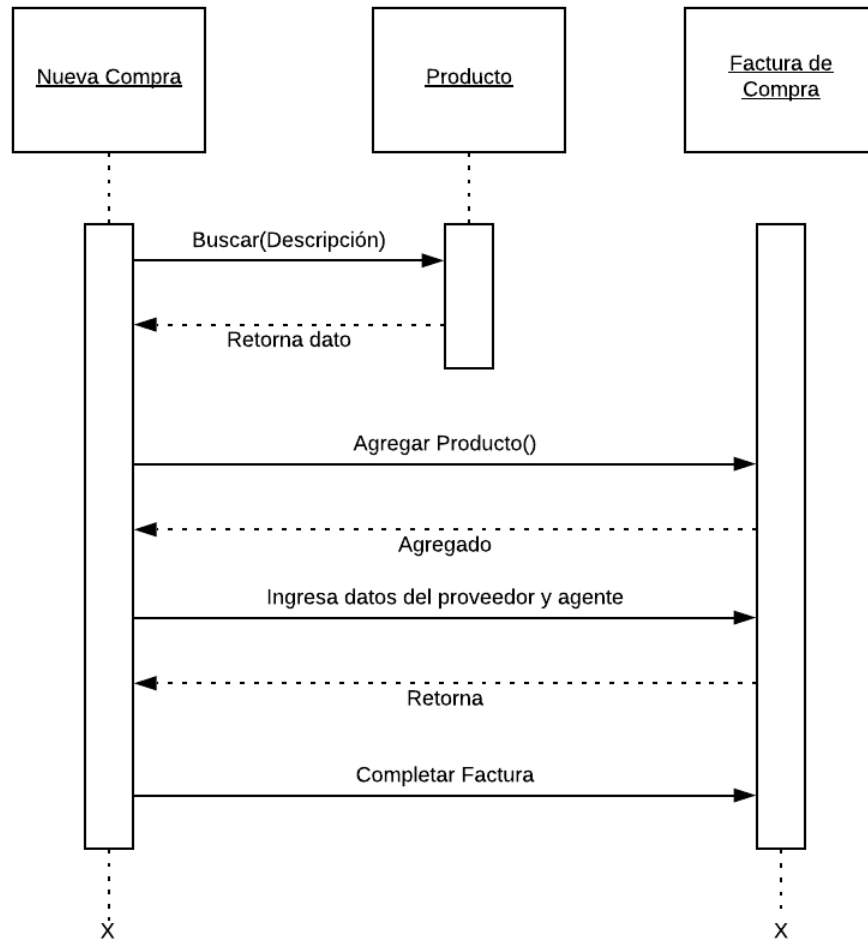
Ilustración 39. Diagrama Secuencia 01



Fuente: Elaboración propia.

La siguiente ilustración representa la secuencia para realizar una nueva compra para la organización.

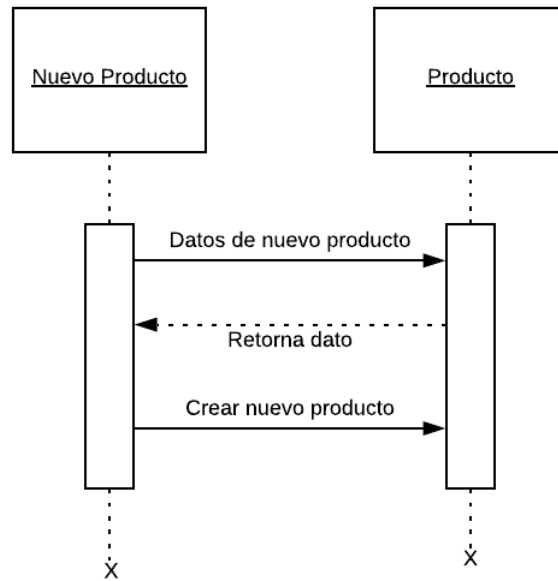
Ilustración 40. Diagrama Secuencia 02



Fuente: Elaboración propia.

Ahora, veremos el diagrama de secuencia para ingresar un nuevo producto al inventario:

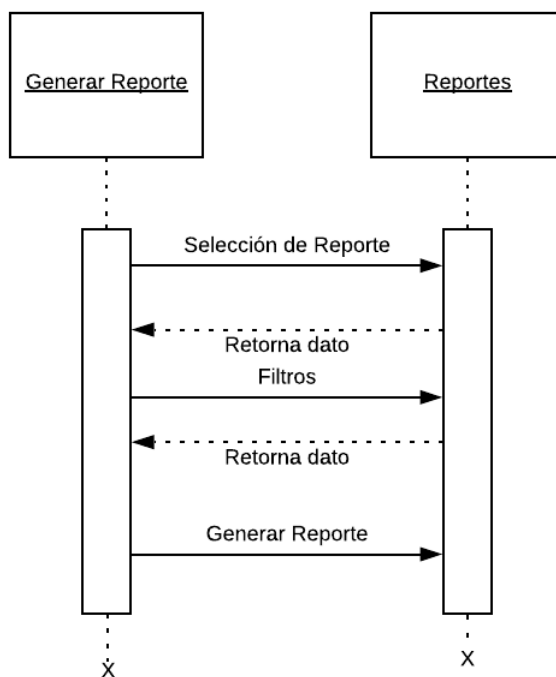
Ilustración 41. Diagrama Secuencia 03



Fuente: Elaboración propia.

Ahora veremos el diagrama de secuencia para generar alguno de los reportes que puede crear el prototipo funcional:

Ilustración 42. Diagrama Secuencia 04



Fuente: Elaboración propia.

Programación

A continuación, se presentarán pequeños ejemplos del código utilizado para las entradas y salidas, los procesos y los módulos que abarcan el alcance del proyecto.

Entradas y salidas.

En este apartado, se mostrarán extractos del código fuente utilizado para las entradas y salidas de la información presente en el prototipo. A continuación, se puede observar la entrada de información para modificar el estado de una carga social:

Ilustración 43. Entrada de Cambio de Estado de Carga Social

```

protected void btnActualizarCargaSocial_Click(object sender, EventArgs e)
{
    lblAccion.Text = "2";
    txtDescripcion.Enabled = true;
    txtCodigo.Enabled = true;
    txtValor.Enabled = true;
    DropDownList1.Enabled = true;
    btnSubmit.Enabled = true;
}

0 referencias
protected void btnSubmit_Click(object sender, EventArgs e)
{
    if (txtCodigo.Text.Trim() == "" || txtDescripcion.Text.Trim() == "" || txtValor.Text.Trim() == "")
    {
        MessageBox.Show("Complete los espacios en blanco.");
    }
    else {
        if (lblAccion.Text == "1") {
            carga.CODIGO_CARGA_SOCIAL = txtCodigo.Text;
            carga.DESCRIPCION_CARGA_SOCIAL = txtDescripcion.Text;
            carga.ESTADO = Convert.ToInt32( DropDownList1.SelectedValue);
            carga.VALOR_CARGA_SOCIAL = (float)Convert.ToDecimal(txtValor.Text);
            _cargaSocialBol.InsertarCargaSocial(carga);
            gridc1.DataBind();
        }
        else {
            if (lblAccion.Text == "2") {
                carga.CODIGO_CARGA_SOCIAL = txtCodigo.Text;
                carga.ID_CARGA_SOCIAL = Convert.ToInt32(txtId.Text);
                carga.DESCRIPCION_CARGA_SOCIAL = txtDescripcion.Text;
                carga.ESTADO = Convert.ToInt32(DropDownList1.SelectedValue);
                carga.VALOR_CARGA_SOCIAL = (float)Convert.ToDecimal(txtValor.Text);
                _cargaSocialBol.ActualizarCargaSocial(carga);
                gridc1.DataBind();
                MessageBox.Show("Registro actualizado correctametne.");
            }
        }
    }
}
}

```

Fuente: Elaboración propia.

La salida correspondiente al proceso anterior, donde el prototipo muestra el cambio al estado de la carga social, se visualiza en la siguiente ilustración:

Ilustración 44. Salida Cambio de Estado de Carga Social

```

gridc1.DataBind();

```

Fuente: Elaboración propia.

Otra de las entradas que se ilustrará a continuación es la de actualización de los agentes de los diferentes proveedores que tiene la organización. En este caso, se reciben varios parámetros que

serán enviados a la base de datos para que los procese y, posteriormente, muestre las salidas respectivas:

Ilustración 45. Entrada Actualización de Agente

```

if (lblTipoAccion.Text == "3")
{
    if (txtIdAgente.Text.Trim() == "")
    {
        MessageBox.Show("Faltan campos por completar.");
    }

    //aquí actualiza
    agente.ID_AGENTE = Convert.ToInt32(txtIdAgente.Text);
    agente.NOMBRE = txtNombre.Text;
    agente.APELLIDO1 = txtApellido1.Text;
    agente.APELLIDO2 = txtApellido2.Text;
    agente.ID_PROVEEDOR = Convert.ToInt32(DDLIdProveedor.Text);
    agente.TELEFONO_AGENTE = Convert.ToInt32(txtTelefonoAgente.Text);

    _agenteBo1.UpdateAgente(agente);
}

```

Fuente: Elaboración propia.

Ilustración 46. Salida Actualización de Agente

```

GridView1.DataBind();
GridView1.Enabled = true;
LimpiarPaginaLoad();

```

Fuente: Elaboración propia.

En este otro caso, se pueden observar las entradas al sistema que serían las credenciales del usuario al intentar hacer inicio de sesión:

Ilustración 47. Entrada Credenciales Inicio de Sesión.

```

protected void txtIngresar_Click(object sender, EventArgs e)
{
    colaborador.ID_COLABORADOR = Convert.ToInt32(txtUser.Text);
    colaborador.PASSWORD = txtPass.Text;
    colaborador.ESTADO = "X";
    colaborador.ID_SUCURSAL = 0;
    colaborador.ID_ROL = 0;
    user = colaborador.ID_COLABORADOR;
    rol = colaborador.ID_ROL;
    sucursal = colaborador.ID_SUCURSAL;
}

```

Fuente: Elaboración propia.

Posterior a la entrada de datos, se manifiesta una salida de información no visible al usuario:

Ilustración 48. Salida de Datos al Iniciar Sesión

```

List<EColaborador> colaboradores = _colaboradorBol.LOGIN(colaborador);

foreach (var a in colaboradores)
{
    user = a.ID_COLABORADOR;
    rol = a.ID_ROL;
    sucursal = a.ID_SUCURSAL;
    estado = a.ESTADO;
}

```

Fuente: Elaboración propia.

Procesos.

En las ilustraciones que se mostrarán a continuación, se podrá ver el proceso que realiza el prototipo para generar un lote de planilla y calcular los montos correspondientes por cada colaborador de la organización:

Ilustración 49. Proceso de Cálculo de Planilla Capa Interfaz

```

protected void btnPagarPlanilla_Click(object sender, EventArgs e)
{
    planillalote.NOMBRE_PLANILLA = txtNombrePlanilla.Text;
    planillalote.FECHA_INI = FechaIni.SelectedDate.Date;
    planillalote.FECHA_FIN = FechaFin.SelectedDate.Date;
    planillalote.ID_COLABORADOR_APRUEBA = Convert.ToInt32(Session["User"].ToString());
    _planillaloteBol.InsertarPlanillaLote(planillalote);

    GridView1.Visible = true;
}

```

Fuente: Elaboración propia.

Seguidamente, el sistema envía el parámetro de usuario que aprueba el registro a la capa de lógica de negocio:

Ilustración 50. Proceso de Cálculo de Planilla Capa Lógica de Negocio

```

2 referencias
public class PlanillaLoteBol {
    private PLANILLA_LOTE_DAL _PlanillaLoteDAL = new PLANILLA_LOTE_DAL();
    public readonly StringBuilder stringBuilder = new StringBuilder();

    1 referencia
    public void InsertarPlanillaLote(EPlanillaLote planillalote)
    {
        _PlanillaLoteDAL.INSERT_PLANILLA_LOTE(planillalote);
    }
}

```

Fuente: Elaboración propia.

Después, la lógica de negocio envía el parámetro a la capa de datos, donde se encarga de prepagar el parámetro para ser enviado a un procedimiento almacenado en la base de datos:

Ilustración 51. Proceso de Cálculo de Planilla Capa de Datos

```

1 referencia
public void INSERT_PLANILLA_LOTE(EPlanillaLote planillalote)
{
    string ConnectionString = "Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=LABODEGUITA;Data Source=LAPTOP-IKPMILKQ\\SQL2019";

    // se crea el string de conexión y se da el nombre del proceso almacenado
    SqlConnection cnn = new SqlConnection(ConnectionString);
    SqlCommand cmd = new SqlCommand("SP_PLANILLA_LOTE", cnn);
    cmd.CommandType = CommandType.StoredProcedure;
    //se abre la conexión
    cnn.Open();

    cmd.Parameters.Add("@NOMBRE_PLANILLA", SqlDbType.Char, 50);
    cmd.Parameters.Add("@FECHA_INI", SqlDbType.Date, 50);
    cmd.Parameters.Add("@FECHA_FIN", SqlDbType.Date, 50);
    cmd.Parameters.Add("@ID_COLABORADOR_APRUEBA", SqlDbType.Int, 50);

    cmd.Parameters["@NOMBRE_PLANILLA"].Value = planillalote.NOMBRE_PLANILLA;
    cmd.Parameters["@FECHA_INI"].Value = planillalote.FECHA_INI;
    cmd.Parameters["@FECHA_FIN"].Value = planillalote.FECHA_FIN;
    cmd.Parameters["@ID_COLABORADOR_APRUEBA"].Value = planillalote.ID_COLABORADOR_APRUEBA;

    //se ejecuta el query
    cmd.ExecuteNonQuery();
    //se cierra la conexión
    cnn.Close();
}

```

Fuente: Elaboración propia.

Ilustración 52. Proceso de Cálculo de Planilla Base de Datos 01

```

9 ALTER PROCEDURE [dbo].[SP_PLANILLA_LOTE]
10 (
11     @NOMBRE_PLANILLA VARCHAR(50),
12     @FECHA_INI DATE,
13     @FECHA_FIN DATE,
14     @ID_COLABORADOR_APRUEBA INT
15 )
16 AS
17
18
19 SET NOCOUNT ON;
20
21 CREATE TABLE #TEMPplanilla (ID_COLABORADOR INT, SALARIO_BRUTO INT)
22 INSERT INTO #TEMPplanilla (ID_COLABORADOR, SALARIO_BRUTO) SELECT ID_COLABORADOR, SALARIO_BASE FROM [dbo].[COLABORADORES] WHERE ESTADO = 'A';
23
24 DECLARE @COUNT_TABLA INT = (SELECT COUNT(*) FROM #TEMPplanilla);
25
26 -----
27
28 Declare @ID_COLABORADOR int;
29 Declare @SALARIO_BASE int;
30 DECLARE @CONTADOR INT = 0;
31

```

Fuente: Elaboración propia.

Ilustración 53. Proceso de Cálculo de Planilla Base de Datos 02

```

--CALCULO DE TEMA DE VACACIONES

--SI VACACIONES ES = 0
IF ((SELECT COUNT(*) FROM [dbo].[VACACIONES] WHERE [ID_COLABORADOR_BENEFICIADO] = @ID_COLABORADOR AND
[FECHA_INI] >= @FECHA_INI AND [FECHA_FIN] <= @FECHA_FIN ) <= 0)
BEGIN
    SET @DIAS_VACACIONES = 0
END
ELSE
BEGIN
    SET @DIAS_VACACIONES = (SELECT SUM (DATEDIFF (DAY,[FECHA_INI] ,DATEADD(DAY ,1 ,[FECHA_FIN] ) )) FROM [dbo].[VACACIONES]
WHERE [ID_COLABORADOR_BENEFICIADO] = @ID_COLABORADOR AND [FECHA_INI] >= @FECHA_INI AND
[FECHA_FIN] <= @FECHA_FIN)
END

--TEMAS DE MARCAS
if (@FECHA_INI = @FECHA_FIN)
begin
    set @diasCortes = 1;
end
else
begin
    --cantidad de dias entre los cortes es decir si la planilla empieza del 01 de enero y termina el 31 de enero, cuenta cuantos dias hay
    set @diasCortes = (select datediff(dd, @FECHA_INI, @FECHA_FIN) - (datediff(wk, @FECHA_INI, @FECHA_FIN)) -
    case when datepart(dw, @FECHA_INI) = 1 then 1 else 0 end +
    case when datepart(dw, @FECHA_FIN) = 1 then 1 else 0 end)
end

```

Fuente: Elaboración propia.

Validaciones.

La imagen que se presenta a continuación muestra una validación, en la que el sistema comprueba que existe una marca de entrada cuando se intenta realizar la marca de salida. Cuando el prototipo envía por parámetro el registro de la marca de salida, devuelve un mensaje que es mostrada por pantalla, en el mensaje se determina si se registró correctamente, o si la marca de entrada no existe. Igualmente, puede retornar un mensaje diciendo que la marca ya existe:

Ilustración 54. Validación Marca de Salida

```

protected void btnSalida_Click(object sender, EventArgs e)
{
    marca.ID_COLABORADOR = Convert.ToInt32(lblUser.Text);
    marca.TIPO_MARCA = 2;

    List<EMarcas> marcas = _marcaBol.InsertarMarca(marca);

    foreach (var a in marcas)
    {
        lblMensaje.Text = (a.MENSAJE);
    }
}

```

Fuente: Elaboración propia.

En la siguiente validación, se comprobará que no exista un producto específico en la actual factura, lo anterior en el módulo de facturación:

Ilustración 55. Validación producto existen en factura actual

```
//aquí se va a validar que el producto no esté ya en el grid view
for (int fila = 0; fila < GridView2.Rows.Count; fila++)
{
    if (txtIdProducto.Text == GridView2.Rows[fila].Cells[1].Text) {
        MessageBox.Show("Producto ya está en la factura, si desea agregar más cantidad de este producto, por favor, elimínelo de la factura y vuelva a ingresarlo con la cantidad correcta.");
        ExisteEnFactura = true;
    }
}
}
```

Fuente: Elaboración propia.

A continuación, se mostrará una imagen de la validación que realiza el sistema para que la cantidad de producto que se pretende vender no sea superior a la disponible en inventario:

Ilustración 56. Validación producto a vender no sea mayor al disponible

```
if (Convert.ToDecimal(txtCantidad.Text) > Convert.ToDecimal(lblCantTemp.Text))
{
    MessageBox.Show("Cantidad ingresada es mayor al disponible en inventario.");
}
```

Fuente: Elaboración propia.

En la siguiente ilustración, se podrá observar cómo el sistema valida el envío o no de factura mediante correo electrónico, al consultar si existe o no un correo al cual enviar:

Ilustración 57. Validación de envío de factura por correo electrónico

```
if (txtCorreo.Text != "") {
    try
    {
        System.Net.Mail.MailMessage mail = new System.Net.Mail.MailMessage();
        mail.To.Add(txtCorreo.Text);
        mail.From = new MailAddress(txtCorreo.Text, "Sistema Automático La Bodeguita", System.Text.Encoding.UTF8);
        mail.Subject = "Factura";
        mail.SubjectEncoding = System.Text.Encoding.UTF8;
        mail.Body = bodyCompleto;
        mail.BodyEncoding = System.Text.Encoding.UTF8;
        mail.IsBodyHtml = true;
        mail.Priority = MailPriority.High;
        SmtplibClient client = new SmtplibClient();
        client.Credentials = new System.Net.NetworkCredential("tavohs354@gmail.com", " ");
        client.Port = 587;
        client.Host = "smtp.gmail.com";
        client.EnableSsl = true;

        try
        {
            client.Send(mail);
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
```

Fuente: Elaboración propia.

Módulos señalados en el alcance.

En las siguientes ilustraciones, se evidencia el código fuente utilizado para desarrollar el módulo de facturación:

Ilustración 58. Módulo de Facturación 01

```

public SqlConnection cn = new SqlConnection("Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=LABODEGUITA;Data Source=LAPTOP-IKPHILKQ\\SQL2019");
DataTable dt = new DataTable();
Referencias
protected void Page_Load(object sender, EventArgs e)
{
    Page.UnobtrusiveValidationMode = System.Web.UI.UnobtrusiveValidationMode.None;
    if (!Page.IsPostBack) {
        if (ViewState["Records"] == null) {
            dt.Columns.Add("Id Producto");
            dt.Columns.Add("Descripción");
            dt.Columns.Add("Cantidad");
            dt.Columns.Add("Precio");
            dt.Columns.Add("Subtotal");
            dt.Columns.Add("Categoría INP");
            dt.Columns.Add("Porcentaje Impuestos");
            dt.Columns.Add("Descuento x unidad de medida");
            dt.Columns.Add("Total Descuento");
            dt.Columns.Add("Monto impuestos");
            ViewState["Records"] = dt;
        }
    }

    if (Page.IsPostBack) {
        lblIdSucursal.Text = Session["ID_SUCURSAL"].ToString();
        lblIdUsuario.Text = Session["User"].ToString();
    }
}
}

```

Fuente: Elaboración propia.

Ilustración 59. Módulo de Facturación 02

```

protected void btnAgregarProducto_Click(object sender, EventArgs e)
{
    Boolean ExisteEnFactura = false;
    if (txtIdProducto.Text.Trim() == "" || txtCantidad.Text.Trim() == "")
    {
        MessageBox.Show("Favor complete campo Id Producto o ingrese una cantidad.");
    }
    else {
        if (Convert.ToDecimal(txtCantidad.Text) > Convert.ToDecimal(lblCantTemp.Text))
        {
            MessageBox.Show("Cantidad ingresada es mayor al disponible en inventario.");
        }
        else {
            //aquí se va a validar que el producto no esté ya en el grid view
            for (int fila = 0; fila < GridView2.Rows.Count; fila++)
            {
                if (txtIdProducto.Text == GridView2.Rows[fila].Cells[1].Text) {
                    MessageBox.Show("Producto ya está en la factura, si desea agregar más cantidad de este producto, por favor, elimínelo de la factura y vuelva a ingresarlo con la cantidad correcta.");
                    ExisteEnFactura = true;
                }
            }
            //validamos que el monto de descuento no sea mayor al precio del producto
            if (txtDescuento.Text == "") {
                txtDescuento.Text = "0";
            }
            if (Convert.ToInt32(txtDescuento.Text) >= Convert.ToInt32(lblPrecioTemp.Text))
            {
                MessageBox.Show("Monto de descuento no puede ser mayor o igual al valor del producto.");
            }
        }
    }
    else {
        if (!ExisteEnFactura) {
            //agrega producto a la factura
            dt = (DataTable)ViewState["Records"];
            if (txtDescuento.Text == "") {
                txtDescuento.Text = "0";
            }
        }
    }
}

```

Fuente: Elaboración propia.

Ilustración 60. Módulo de Facturación 03

```

if (txtDescuento.Text == "") {
    txtDescuento.Text = "0";
}
if (Convert.ToInt32(txtDescuento.Text) >= Convert.ToInt32(lblPrecioTemp.Text))
{
    MessageBox.Show("Monto de descuento no puede ser mayor o igual al valor del producto.");
}
else {
    if (!ExisteEnFactura) {
        //agrega producto a la factura
        dt = (DataTable)ViewState["Records"];
        if (txtDescuento.Text == "") {
            txtDescuento.Text = "0";
        }
        string PRECIOUNI = Convert.ToString(((Convert.ToInt32(lblPrecioTemp.Text) - Convert.ToInt32(txtDescuento.Text))));
        int SUBTOTAL = Convert.ToInt32((Convert.ToDecimal(txtCantidad.Text) * (Convert.ToDecimal(lblPrecioTemp.Text) - Convert.ToDecimal(txtDescuento.Text))));
        int DESCUENTOTOTOPROD = Convert.ToInt32(Convert.ToDecimal(txtDescuento.Text) * Convert.ToDecimal(txtCantidad.Text));
        int montImpuestosPROD = Convert.ToInt32(Convert.ToDecimal(lblPorcentajeImp.Text) * Convert.ToDecimal((Convert.ToDecimal(txtCantidad.Text) * (Convert.ToDecimal(lblPrecioTemp.Text) - Convert.ToDecimal(txtDescuento.
dt.Rows.Add(txtIdProducto.Text, txtDescripcion.Text, txtCantidad.Text, PRECIOUNI, SUBTOTAL, lblIdCategoria.Text, lblPorcentajeImp.Text, txtDescuento.Text ,DESCUENTOTOTOPROD, montImpuestosPROD);
GridView2.DataSource = dt;
GridView2.DataBind();
//suma total para la factura
int totalSalary = 0;
foreach (DataRow dr in dt.Rows)
{
    totalSalary += Convert.ToInt32(dr["Subtotal"]);
}
lblFacturaTotalAPagar.Text = totalSalary.ToString();
GridView2.DataBind();

double totalDESCUENTO = 0;
foreach (DataRow dr in dt.Rows)
{
    totalDESCUENTO += Convert.ToDouble(dr["Total Descuento"]);
}
lblsumDESCUENTOFACT.Text = totalDESCUENTO.ToString();

int totalIVAFACTURA = 0;
foreach (DataRow dr in dt.Rows)
{
    totalIVAFACTURA += Convert.ToInt32(dr["Monto impuestos"]);
}
lblsumIMPUESTOSfact.Text = totalIVAFACTURA.ToString();
// lblSumatorIoIVA.Text = Convert.ToString(totalDESCUENTO)
}

```

Fuente: Elaboración propia.

Ilustración 61. Módulo de Facturación 04

```

protected void EliminarRegistroDeFactura(object sender, GridViewDeleteEventArgs e)
{
    int index = Convert.ToInt32(e.RowIndex);
    DataTable dt = ViewState["Records"] as DataTable;
    dt.Rows[index].Delete();
    ViewState["dt"] = dt;
    GridView2.DataSource = dt;
    GridView2.DataBind();
    //agrega producto a la factura
    dt = (DataTable)ViewState["Records"];
    GridView2.DataSource = dt;
    GridView2.DataBind();
    //suma total para la factura
    int totalSalary = 0;
    foreach (DataRow dr in dt.Rows)
    {
        totalSalary += Convert.ToInt32(dr["Subtotal"]);
    }

    int totalIVA = 0;
    foreach (DataRow dr in dt.Rows)
    {
        totalIVA += Convert.ToInt32(dr["Descuento x unidad de medida"]);
    }

    lblSumatorioIVA.Text = Convert.ToString(totalIVA);
    lblFacturaTotalAPagar.Text = totalSalary.ToString();
    int totalDESCUENTO = 0;
    foreach (DataRow dr in dt.Rows)
    {
        totalDESCUENTO += Convert.ToInt32(dr["Total Descuento"]);
    }
    lblsumDESCUENTOFACT.Text = totalDESCUENTO.ToString();

    int totalIVAFACTURA = 0;
    foreach (DataRow dr in dt.Rows)
    {
        totalIVAFACTURA += Convert.ToInt32(dr["Monto impuestos"]);
    }
    lblsumIMPUESTOSfact.Text = totalIVAFACTURA.ToString();

    GridView2.DataBind();
}

```

Fuente: Elaboración propia.

Ahora, se va a mostrar un extracto del código empleado para desarrollar el módulo de compras del prototipo:

Ilustración 62. Módulo de Compras 01

```

protected void btnBuscar_Click(object sender, EventArgs e)
{
    SqlDataAdapter da = new SqlDataAdapter("SELECT distinct(p.ID_PRODUCTO), p.DESCRIPCION, p.PRECIO_COMPRA, p.PRECIO_VENTA FROM dbo.PRODUCTOS p inner join dbo.STOCK s on p.ID_PRODUCTO = s.ID_PRODUCTO WHERE p.DESCRIPCIO
    DataTable dt = new DataTable();
    da.Fill(dt);
    GridView1.DataSource = (dt);
    //GridView1.DataSource = (dt);
    GridView1.DataBind();

    GridView1.Visible = true;
}

```

Fuente: Elaboración propia.

Ilustración 63. Módulo de Compras 02

```

protected void btnAgregarFactura_Click(object sender, EventArgs e)
{
    txtCantidad.Enabled = false;
    txtPrecioCompra.Enabled = false;
    txtPrecioVenta.Enabled = false;

    Boolean ExisteEnFactura = false;
    if (txtIDProducto.Text.Trim() == "" || txtCantidad.Text.Trim() == "" || txtPrecioCompra.Text.Trim() == "" || txtPrecioVenta.Text.Trim() == "")
    {
        MessageBox.Show("Favor complete campo Id Producto o ingrese una cantidad.");
    }
    else
    {
        //aquí se va a validar que el producto no esté ya en el grid view
        for (int fila = 0; fila < GridView2.Rows.Count; fila++)
        {
            if (txtIDProducto.Text == GridView2.Rows[fila].Cells[2].Text)
            {
                MessageBox.Show("Producto ya está en la factura, si desea agregar más cantidad de este producto, por favor, elimínelo de la factura y vuelva a ingresarlo con la cantidad correcta.");
                ExisteEnFactura = true;
            }
        }

        if (!ExisteEnFactura)
        {
            //agrega producto a la factura
            dt = (DataTable)ViewState["Records"];
            dt.Rows.Add(DDLIDProveedor.Text,txtDescripcion.Text , txtIDProducto.Text, DDLIDAgente.Text, DDLIDSucursal.Text, Convert.ToDecimal(txtCantidad.Text), txtPrecioCompra.Text, txtPrecioVenta.Text, Convert.T
            GridView2.DataSource = dt;
            GridView2.DataBind();

            //suma total para la factura
            double TotalFactura = 0;
            foreach (DataRow dr in dt.Rows)

```

Fuente: Elaboración propia.

Ilustración 64. Módulo de Compras 03

```

protected void EliminarRegistroFactura(object sender, GridViewDeleteEventArgs e)
{
    int index = Convert.ToInt32(e.RowIndex);
    DataTable dt = ViewState["Records"] as DataTable;
    dt.Rows[index].Delete();
    ViewState["dt"] = dt;

    GridView2.DataSource = dt;
    GridView2.DataBind();

    //agrega producto a la factura
    dt = (DataTable)ViewState["Records"];

    GridView2.DataSource = dt;
    GridView2.DataBind();

    //suma total para la factura
    double totalSalary = 0;
    foreach (DataRow dr in dt.Rows)
    {
        totalSalary += Convert.ToDouble(dr["Subtotal"]);
    }

    lblTotalFactura.Text = totalSalary.ToString();

    .
    GridView2.DataBind();
}

```

Fuente: Elaboración propia.

Seguidamente, se observará un poco del código que se utilizó para desarrollar el módulo de planilla, donde se genera el lote de pago de planilla a todos los colaboradores activos de la organización:

Ilustración 65. Módulo de Planilla 01

```
protected void btnPagarPlanilla_Click(object sender, EventArgs e)
{
    planillalote.NOMBRE_PLANILLA = txtNombrePlanilla.Text;
    planillalote.FECHA_INI = FechaIni.SelectedDate.Date;
    planillalote.FECHA_FIN = FechaFin.SelectedDate.Date;
    planillalote.ID_COLABORADOR_APRUEBA = Convert.ToInt32(Session["User"].ToString());
    _planillaloteBol.InsertarPlanillaLote(planillalote);

    GridView1.Visible = true;
}
```

Fuente: Elaboración propia.

Ilustración 66. Módulo de Planilla 02

```
protected void btnVerDetallePlanilla_Click(object sender, EventArgs e)
{
    GridView1.Visible = false;
    GridView2.Visible = false;
    lblValorIdPlanilla.Text = DDLIdDetallePlanilla.SelectedValue;
    GridView3.DataBind();
    GridView3.Visible = true;
}
```

Fuente: Elaboración propia.

Uno de los módulos del prototipo es el de inventario, a continuación, se mostrarán extractos del código fuente de este módulo:

Ilustración 67. Módulo de Inventario 01

```
//toma el valor del idproducto y lo consulta en bd para traer los datos correspondientes
producto_ID_PRODUCTO = Convert.ToInt32(txtIdProducto.Text);
List<EProductos> productos = _productoBol.Select_PRODUCTO_ID(producto);

foreach (var a in productos)
{
    txtDescripcion.Text = Convert.ToString(a.DESCRIPCION);
    txtPrecioCompra.Text = Convert.ToString(a.PRECIO_COMPRA);
    txtPrecioVenta.Text = Convert.ToString(a.PRECIO_VENTA);
}

if (lblTipo_Accion.Text == "1" && txtDescripcion.Text.Length > 0)
{
    txtIdProducto.Enabled = false;
    txtDescripcion.Enabled = true;
    txtPrecioCompra.Enabled = true;
    txtPrecioVenta.Enabled = true;
    DDLSucursales.Enabled = true;
    txtStock.Enabled = true;
    btnSubmit.Enabled = true;
    DDLIdCategoriaProducto.Enabled = true;
}
else {
    if (lblTipo_Accion.Text == "2" && txtDescripcion.Text.Length > 0)
    {
        txtIdProducto.Enabled = false;
        txtDescripcion.Enabled = false;
        txtPrecioCompra.Enabled = false;
        txtPrecioVenta.Enabled = false;
        DDLSucursales.Enabled = false;
        txtStock.Enabled = false;
        btnSubmit.Enabled = true;
        BtnAccActualizar.Enabled = false;
    }
    else
    {
        if (lblTipo_Accion.Text == "3" && txtDescripcion.Text.Length > 0)
        {
            txtIdProducto.Enabled = false;
            txtDescripcion.Enabled = true;
            txtPrecioCompra.Enabled = true;
            txtPrecioVenta.Enabled = true;
            DDLSucursales.Enabled = true;
            txtStock.Enabled = true;
            btnSubmit.Enabled = true;
            DDLIdCategoriaProducto.Enabled = true;
        }
    }
}
```

Fuente: Elaboración propia.

Ilustración 68. Módulo de Inventario 02

```
protected void btnSubmit_Click(object sender, EventArgs e)
{
    if (lblTipo_Accion.Text == "1")
    {
        SUBMIT_Insertar_Producto();
    }

    if (lblTipo_Accion.Text == "3")
    {
        SUBMIT_Update_Producto();
    }

    if (lblTipo_Accion.Text == "2")
    {
        SUBMIT_Delete_Producto();
    }

    if (lblTipo_Accion.Text == "4")
    {
    }

    if (lblTipo_Accion.Text == "5")
    {
    }

    btnResetear.Enabled = false;

    Page_Load(sender, e);
    GridView1.DataBind();
    GridView1.Enabled = true;
    LimpiarTxt();
}
```

Fuente: Elaboración propia.

A continuación, se mostrarán extractos de código fuente que componen el módulo de seguridad presente en todo el prototipo, con el fin de comprobar el correcto acceso por parte de los usuarios a los diferentes módulos:

Ilustración 69. Módulo de Seguridad 01

```
protected void Page_Load(object sender, EventArgs e)
{
    Page.UnobtrusiveValidationMode = System.Web.UI.UnobtrusiveValidationMode.None;
    if (Session["ID_Rol"].ToString() == "2" )
    {
        MessageBox.Show("Su rol no cuenta con acceso a este módulo.");
        Response.Redirect("~/FACTURACION_UI.aspx");
    }
    else {
        if (!Page.IsPostBack) {
            DDIdAgente.Enabled = false;
            if (ViewState["Records"] == null)
            {
                dt.Columns.Add("Id Proveedor");
                dt.Columns.Add("Descripción");
                dt.Columns.Add("Id Producto");
                dt.Columns.Add("Id Agente");
                dt.Columns.Add("Id Sucursal");
                dt.Columns.Add("Cantidad");
                dt.Columns.Add("Precio Compra");
                dt.Columns.Add("Precio Venta");
                //dt.Columns.Add("Total Producto");
                lblidUser.Text = Session["User"].ToString();
                dt.Columns.Add("Subtotal");
                ViewState["Records"] = dt;
            }
        }
    }
}
```

Fuente: Elaboración propia.

Ilustración 70. Módulo de Seguridad 02

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!Page.IsPostBack && Session["ID_Rol"].ToString() == "1")
    {
    }
    else
    {
        MessageBox.Show("Su rol no cuenta con acceso a este módulo.");
        Response.Redirect("~/FACTURACION_UI.aspx");
    }
}
```

Fuente: Elaboración propia.

Por último, se mostrará un extracto del código perteneciente al módulo de reportes:

Ilustración 71. Módulo de Reportes

```
protected void btnGenerarReporte_Click(object sender, EventArgs e)
{
    //string nombrePDF = "Reporte_Top5_Más_Vendidos_entre";
    GridView1.DataBind();
    GridView1.Visible = true;

    Response.ContentType = "application/pdf";
    Response.AddHeader("content-disposition", "attachment;filename=Reporte_Top5_Ventas_Por_Producto.pdf");
    Response.Cache.SetCacheability(HttpCacheability.NoCache);
    StringWriter sw = new StringWriter();
    HtmlTextWriter hw = new HtmlTextWriter(sw);
    GridView1.AllowPaging = false;
    GridView1.DataBind();
    GridView1.RenderControl(hw);
    StringReader sr = new StringReader(sw.ToString());
    Document pdfDoc = new Document(PaperSize.A2, 7f, 7f, 7f, 0f);
    HTMLWorker htmlparser = new HTMLWorker(pdfDoc);
    PdfWriter.GetInstance(pdfDoc, Response.OutputStream);
    pdfDoc.Open();
    htmlparser.Parse(sr);
    pdfDoc.Close();
    Response.Write(pdfDoc);
    Response.End();
}
```

Fuente: Elaboración propia.

Pruebas

En el siguiente apartado, se evidenciarán las diferentes pruebas realizadas al prototipo. Se visualizará un resultado esperado y un resultado obtenido, esto para comprobar la calidad del sistema desarrollado.

Como primer objetivo, se realizarán pruebas a la interfaz de inicio de sesión, donde se probará tanto con un inicio de sesión correcto como intentando ingresar con credenciales incorrectos:

Tabla 26. Pruebas Módulo Inicio de Sesión 01

Casos de pruebas							
Número de prueba	Módulo	Nombre de la prueba	Paso	Descripción	Expectativa del resultado	Tipo	Resultado final
1	Inicio de Sesión	Validación de credenciales incorrectos.	1	Se ingresará un usuario y contraseña que no existen en la base de datos.	El prototipo no debe permitir el ingreso al sistema.	Manual	Exitoso

Fuente: Elaboración propia

Tabla 27. Pruebas Módulo Inicio de Sesión 02

Casos de pruebas							
Número de prueba	Módulo	Nombre de la prueba	Paso	Descripción	Expectativa del resultado	Tipo	Resultado final
2	Inicio de Sesión	Validación de credenciales correctos.	1	Se ingresará un usuario y contraseña válidos.	El prototipo debe permitir el ingreso al sistema.	Manual	Exitoso

Fuente: Elaboración propia

Ahora se realizarán pruebas al módulo de marcas en el sistema, donde algunas de las pruebas intentan realizar marcas existentes:

Tabla 28. Pruebas Módulo Marcas

Casos de pruebas							
Número de prueba	Módulo	Nombre de la prueba	Paso	Descripción	Expectativa del resultado	Tipo	Resultado final
3	Marcas	Validación de ingreso de marca de entrada cuando ya fue realizada	1	Se debe ingresar una marca de entrada por segunda vez el mismo día.	El sistema debe mostrar un mensaje informando que la marca ya existe.	Manual	Exitoso
4	Marcas	Validación de ingreso de marca de salida cuando ya fue realizada.	2	Se debe ingresar una marca de salida por segunda vez el mismo día.	El sistema debe mostrar un mensaje informando que la marca ya existe.	Manual	Exitoso

5	Marcas	Validación de ingreso de marca de salida cuando no existe una marca de entrada en el mismo día.	3	Se debe ingresar una marca de salida cuando aún no existe una marca de entrada en el mismo día.	El sistema debe mostrar un mensaje informando que no existe una marca de entrada.	Manual	Exitoso
---	--------	---	---	---	---	--------	---------

Fuente: Elaboración propia

Seguidamente, se muestra el *script* de pruebas al módulo de generación de Lote de Planillas, parte del *script* de pruebas intenta generar planillas donde la fecha inicial es mayor a la fecha final:

Tabla 29. Pruebas Módulo Lote Planilla

Casos de pruebas							
Número de prueba	Módulo	Nombre de la prueba	Paso	Descripción	Expectativa del resultado	Tipo	Resultado final
6	Planilla Lote Pago	A la hora de seleccionar las fechas, la fecha inicial debe ser mayor a la fecha final	1	Se debe ingresar una fecha inicial y una fecha final en la cual la fecha inicial debe ser mayor a la final y presione el botón de "Generar".	El sistema debe mostrar un mensaje de error, el cual debe informar que la fecha final no puede ser menor a la inicial.	Manual	Exitoso
7	Planilla	El campo de nombre de planilla no debe quedar en blanco	2	Se debe verificar que el nombre de planilla no quede en blanco si es así a la hora de presionar el botón "Generar" muestra un mensaje informando.	El sistema debe mostrar un mensaje informativo donde avise que el campo de nombre de la planilla está en blanco.	Manual	Exitoso

Fuente: Elaboración propia

En el módulo de facturación, se intenta vender un producto en una cantidad superior al disponible en *stock*, también se intenta facturar con el correo en blanco para demostrar la capacidad que tiene el sistema para ignorar este evento.

Tabla 30. Pruebas Módulo Facturación

Casos de pruebas							
Número de prueba	Módulo	Nombre de la prueba	Paso	Descripción	Expectativa del resultado	Tipo	Resultado final
8	Facturación	Validar que el descuento no sea mayor al precio del producto.	1	Ingresar una cantidad de descuento superior al precio del producto.	Que el sistema no permita la acción e informe el hallazgo.	Manual	Exitoso
9	Facturación	Comprobar el comportamiento del sistema al dejar campo de correo en blanco.	2	Se debe generar la factura sin un correo asociado a la factura.	El sistema debe permitir generar la factura sin correo.	Manual	Exitoso
10	Facturación	Validar que cantidad de producto agregado a la factura no sea superior al disponible en inventario.	3	Se debe ingresar una cantidad del producto a vender superior al disponible.	El sistema debe mostrar un mensaje informando que la cantidad a vender es superior al stock.	Manual	Exitoso

Fuente: Elaboración propia

En el módulo de compras se realizan pruebas para validar caracteres no permitidos, y también se intentará agregar productos a la factura con campos vacíos.

Tabla 31. Pruebas Módulo Compras

Casos de pruebas							
Número de prueba	Módulo	Nombre de la prueba	Paso	Descripción	Expectativa del resultado	Tipo	Resultado final
11	Compras	Validar que no se permita ingresar en cantidad un carácter no permitido.	1	Ingresar un carácter diferente a numérico en el campo de cantidad.	Que el sistema no permita agregar productos con un campo diferente a numérico en cantidad.	Manual	Exitoso
12	Compras	Comprobar el comportamiento del sistema al agregar a factura datos incompletos.	2	Intentar agregar producto a la factura con un campo requerido en blanco.	El sistema debe mostrar un mensaje informando que hace falta de completar un campo.	Manual	Exitoso

13	Compras	Validar que el sistema reste al total de la factura cuando se eliminar un registro de la factura.	3	Borrar de factura un registro.	El total de la factura disminuye al borrar registro de la factura.	Manual	Exitoso
----	---------	---	---	--------------------------------	--	--------	---------

Fuente: Elaboración propia

Tabla 32.Pruebas Módulo Reportes

Casos de Pruebas							
Numero de Prueba	Modulo	Nombre de la prueba	Paso	Descripción	Expectativa del resultado	Tipo	Resultado Final
14	Reportes	Validar que se genere el reporte.	1	Se presiona el botón de generar reporte.	Que se descargue el reporte con los datos correspondientes.	Manual	Exitoso
15	Reportes	Comprobar el comportamiento del sistema al generar reporte de fechas que no generen información	2	Buscar un rango de fechas donde no existan registros.	El sistema no debe descargar ningún reporte.	Manual	Exitoso

Fuente: Elaboración Propia

Referencias

- Arias, F. (2012). *El Proyecto de Investigación Introducción a la metodología científica*. Sexta Edición. Caracas: Editorial Episteme.
- Capote, O. P. (2009). *Introducción a las bases de datos: el modelo relacional*. En O. P. Capote, *Introducción a las bases de datos: el modelo relacional*.
- Fé, I. R., & Quito, A. (2012). *Irfeyal wordpress*. Obtenido de BASE DE DATOS: <https://irfeyal.wordpress.com/bases-de-datos/modelamiento-de-bdd/>
- Gómez, M. M. (2006). *Introducción a la Metodología de la Investigación Científica*. En M. M. Gómez, *Introducción a la Metodología de la Investigación Científica*. Editorial Brujas.
- Hernández, D. R., Fernandez, D. C., & Baptista, D. M. (2014). *Metodología de la investigación*. México DF: INTERAMERICANA EDITORES, S.A. DE C.V.
- López, P. L. (2004). *Población muestra y muestreo*. Punto Cero.
- Microsoft. (2019). *Microsoft*. Obtenido de .NET: <https://docs.microsoft.com/es-es/dotnet/framework/get-started/>
- Microsoft. (2019). *Microsoft*. Obtenido de Tablas: <https://docs.microsoft.com/es-es/sql/relational-databases/tables/tables?view=sql-server-2017>
- Moreno, M. (2010). Obtenido de Alesga: <http://www.alesga.com.ar/Diccionario/C/7664.php>
- Porto, J. P., & Gardey, A. (2012). *Información*. Obtenido de <https://definicion.de/informacion/>
- Porto, J. P., & Gardey., A. (2014). *Definición de*. Obtenido de Requerimiento: <https://definicion.de/requerimiento/>
- Porto, J. P., & Merino, M. (2008). *Definición de*. Obtenido de <https://definicion.de/?s=Computadora>
- Porto, J. P., & Merino, M. (2008). *Hardware*. Obtenido de <https://definicion.de/hardware/>
- Porto, J. P., & Merino, M. (2009). *Definición de*. Obtenido de Datos: <https://definicion.de/datos/>
- Raffino, M. (2019). *Concepto de*. Obtenido de <https://concepto.de/seguridad/>

Suárez, M. (2016). *Framework*. Obtenido de <https://es.quora.com/Qu%C3%A9-es-un-framework>

Ministerio de Trabajo y Seguridad Social (2020). Mtss.go.cr. Recuperado de http://www.mtss.go.cr/temas-laborales/salarios/Documentos-Salarios/lista_salarios_2020.pdf

APÉNDICES

Apéndice 1. Entrevista

1. ¿Como se realiza el proceso de facturación de un producto?

2. ¿Cómo llevan el control de las ventas de productos?
3. ¿Cómo se lleva el control de la mercadería comprada a proveedores?
4. ¿Cómo calculan los pagos a los colaboradores?
5. ¿Cómo se lleva el control de los días de vacaciones disponibles por colaborador?
6. ¿De qué manera toman decisiones con relación a la compra de productos?
7. ¿De qué manera se informan sobre el disponible de productos en los otros locales comerciales de La Bodeguita?
8. ¿Manejan el dato de cuanto inventario se tiene actualmente de cada producto en los locales comerciales de La Bodeguita?
9. ¿Le parece más eficiente un sistema informático que centralice la mayoría de labores realizadas que el procedimiento actual?
10. ¿Sabe utilizar una computadora con sistema operativo Windows?
11. ¿Tiene conocimientos utilizando sistemas de punto de venta?
12. ¿Sabe digitar de manera eficiente en una computadora?
13. ¿Qué expectativa tiene si se llega a implementar un sistema centralizado de punto de ventas donde pueda realizar facturas a clientes, llevar control de inventario y de control de planilla?



Máster

Olda Bustillos Ortega

Directora Escuela de Ingeniería Informática

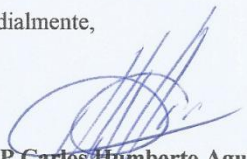
Universidad Internacional de Las Américas

Estimada Directora

En mi calidad de docente del curso Taller Intensivo de Investigación para las carreras de la Escuela, hago constar que he revisado y avalado el contenido del anteproyecto presentado por el estudiante Gustavo Alonso Hidalgo Sánchez, el cual se denomina "Prototipo Funcional para Centralización de Sistema de Punto de Venta para la cadena de Minisúper La Bodeguita, ubicadas en el área de Desamparados y San Sebastián".

El documento reúne los requisitos de fondo y forma para aprobar el taller. Someto a su consideración el documento.

Cordialmente,



MAP Carlos Humberto Aguilar Mora
Docente del curso