



INSTITUTO TECNOLÓGICO SUPERIOR DE HUATUSCO

SISTEMA WEB PARA LA GENERACION DE REPORTE DE FACTURAS RECIBIDAS, PARA LA EMPRESA ABARROTES CORDOBESES SA DE CV.

INFORME TECNICO DE RESIDENCIAS

PARA OBTENER EL TITULO DE:
ING. EN SISTEMAS COMPUTACIONALES

PRESENTA:
CELESTINO DELGADO NOGALES

DIRECTOR DE TRABAJO RECEPCIONAL
MCI. GASPAS CASTILLO DOMÍNGUEZ

HUATUSCO, VER.

FEBRERO 2024

DEDICATORIAS

El presente trabajo lo dedico principalmente a Dios, a mi familia, a mi esposa y a mis hijos quienes han sido parte fundamental en todo este proceso.

A mi padre, el cual, a pesar de haberlo perdido a muy temprana edad, ha estado siempre guiándome, llevándome por un buen camino y cuidándome desde el cielo. Te amo papá.

Y a todas las personas que me han apoyado y han hecho que el trabajo se realice con éxito y en especial a aquellos que me abrieron las puertas y compartieron sus conocimientos.

AGRADECIMIENTOS

Le agradezco mucho a mi asesor el MCI. Gaspar Castillo Domínguez por su dedicación y paciencia, sin sus palabras y correcciones precisas no hubiese podido lograr esta meta. Gracias por su guía y todos sus consejos, los llevaré grabados para siempre en la memoria de mi futuro profesional.

A todos los profesores del Tecnológico Superior de Huatusco, por su dedicación y esfuerzo, quienes me han aportado todos los conocimientos profesionales y personales que me han servido para llegar a finalizar con éxito esta gran meta.

Quiero agradecer también a todos mis compañeros los cuales muchos de ellos se han convertido en mis amigos, cómplices y hermanos. Gracias por las horas compartidas, por los trabajos realizados en conjunto y principalmente por las historias vividas.

Agradezco también a cada directivo por su trabajo y por su gestión ya que sin lo cual no estarían las bases ni las condiciones para forjar conocimientos.

Por último, agradecer a la institución que me ha exigido tanto, pero al mismo tiempo me ha permitido realizarme profesionalmente.

Índice de Contenido

DEDICATORIAS	1
AGRADECIMIENTOS	2
I. Resumen.....	1
II. Abstract	3
III. Introducción.....	5
IV. Definición del Problema	7
V. Objetivo General	8
VI. Objetivos Específicos.....	9
VII. Justificación	10
VIII. Alcances y Limitaciones.....	11
IX. Marco teórico	12
Capítulo 1 - Antecedentes de la Empresa.....	38
1.1 Historia.....	38
1.2 Ubicación	39
1.3 Misión.....	40
1.4 Visión	40
1.5 Política de Calidad	40
1.6 Organigrama	41
1.7 Plano de la Empresa.....	42
1.8 Caracterización del Área.....	43
Capítulo 2 - Metodología de Desarrollo.....	44
2.1 Selección de la Metodología	75
2.2 Implementación de la metodología	83
Capítulo 3 - Análisis de Requerimientos	93
3.1 Requerimientos Funcionales.....	93
3.2 Requerimientos No Funcionales	94
3.2.1 Requisitos de Rendimiento	94
3.2.2 Requisitos de Seguridad.....	95
3.2.3 Requisitos de Fiabilidad.....	96
3.2.4 Requisitos de Disponibilidad	96

3.2.5 Requisitos de Mantenibilidad	97
3.2.6 Requisitos de Portabilidad	97
3.3 Otros Requerimientos	98
3.4 Estudio de Factibilidad	98
3.4.1 Factibilidad Técnica	98
3.4.2 Factibilidad Económica	100
3.4.3 Factibilidad Operacional	101
Capítulo 4 - Desarrollo del proyecto	102
4.1 Casos de Uso.....	102
4.2 Diagramas de Secuencia	108
4.3 Diagramas de Clases.....	113
4.4 Diagrama de Contexto	114
4.5 Diseño de la Base de Datos.....	115
4.5.1 Modelo Entidad-Relación.....	115
4.5.2 Normalización del Modelo Entidad-Relación	116
4.5.3 Modelo Relacional	118
4.5.4 Diccionario de Datos.....	119
4.6 Diseño de la Interfaz Gráfica de Usuario	121
Capítulo 5 - Pruebas y Resultados.....	125
5.1 Descripción del Funcionamiento del Sistema	125
5.2 Pruebas del Sistema.....	126
5.2.1 Metodología para las Pruebas del Sistema	126
5.2.2 Pruebas Unitarias	126
5.2.3 Observaciones Durante las Pruebas del Sistema.....	128
5.3 Implementación del Sistema	128
5.4 Manual del Usuario	132
5.5 Mejora Continua.....	132
Capítulo 6 - Conclusiones	133
6.1 Competencias Desarrolladas o Aplicadas.....	133
6.2 Conclusiones.....	135
6.3 Recomendaciones	136
Bibliografía	137

Índice de Términos.....	144
Anexos	146
Anexo 1 - Manual del Usuario	146
2. Menú Principal.....	151
3. Módulo de Consulta de CFDI.....	152
3.1.-Consulta por Proveedor	153
3.2.-Consulta por Folio de Factura.....	154
3.3.-Consulta por Folio Fiscal-UUID.....	155
3.4.-Descarga de CFDI	156
4.-. Módulo de Reportes.....	158
4.1.-Consulta de Reporte	159
4.2.-Exportar Reporte.....	160
5.- Cerrar Sesión.....	162
Anexo 2 - Ejemplos Varios	163

Índice de Figuras

Figura 1. Cláusulas de SQL	35
Figura 2. Tipos de términos de SQL.	36
Figura 3. Lenguaje de consulta estructurada SQL.....	37
Figura 4. Edificio y ubicación Abarrotes Cordobeses Sa de CV.....	39
Figura 5. Organigrama de Abarrotes Cordobeses Sa de CV	41
Figura 6. Plano de la Empresa Abarrotes Cordobeses	42
Figura 8 . Metodología SCRUM	48
Figura 9. Capas de la programación extrema.....	56
Figura 10. Variables de la Programación Extrema.	57
Figura 11. Familia de Metodologías Crystal.....	61
Figura 12. Tablero Kanban.	66
Figura 13. Modelo del ciclo de vida en Cascada (Waterfall).	68
Figura 14. Modelo del ciclo de vida Prototipo	71
Figura 15. Modelo del Ciclo de vida en Espiral.	72
Figura 16. Modelo del Ciclo de vida Incremental.....	75
Figura 17. Resultado de Pregunta 1.....	88
Figura 18. Resultado de Pregunta 2.....	88
Figura 19. Resultado de Pregunta 3.....	89
Figura 20. Resultado de Pregunta 4.....	89
Figura 21 . Resultado de Pregunta 5.....	90
Figura 22. Resultado de Pregunta 6.....	91
Figura 23. Resultado de Pregunta 7.	91
Figura 24. Resultado de Pregunta 8.....	92
Figura 25. Caso de uso General	102
Figura 26. Caso de uso 1. Ingresar al sistema.	103
Figura 27. Caso de uso 2. Consultar Factura	104
Figura 28. Caso de uso 3. Consultar Proveedor.	105
Figura 29. Caso de uso 4. Generar Reporte Facturas.	106
Figura 30 . Diagrama de secuencia 1. Ingresar al Sistema.....	108
Figura 31. Diagrama de secuencia 2. Consultar Factura.....	109
Figura 32 . Diagrama de secuencia 3. Consultar Proveedor.	110
Figura 33 . Diagrama de secuencia 4. Generar Reporte Facturas.....	111
Figura 34. Diagrama de secuencia 5. Generar Reporte Proveedores.....	112
Figura 35. Diagrama de clases.	113
Figura 36. Diagrama de contexto.....	114
Figura 37. Modelo E-R del sistema general de Abarrotes Cordobeses.	115
Figura 38 . Modelo Relacional.....	118
Figura 39. Prototipo de la GUI. Inicio de Sesión.	121
Figura 40. Prototipo de la GUI. Panel Principal.....	122
Figura 41. Prototipo de la GUI. Pantalla de consultar Factura.....	123
Figura 42. Prototipo de la GUI. Pantalla Generar Reporte de Facturas.....	124
Figura 43 . Prueba Unitaria. ReporteFacturasValidasPantalla.....	127
Figura 44 . Ventana de Inicio de Sesión.....	129

Figura 45. Ventana de CFDI-Consulta de Facturas.....	130
Figura 46. Ventana de Reporte-Facturas.	131

Índice de Tablas

Tabla 1. Metodologías De Desarrollo.....	45
Tabla 2. Requerimiento Funcional 1. Ingresar al Sistema	93
Tabla 3. Requerimiento Funcional 2. Consulta Proveedor	93
Tabla 4. Requerimiento Funcional 3. Consultar Comprobante CFDI.....	93
Tabla 5. Requerimiento Funcional 4. Generar Reporte	94
Tabla 6. Requerimiento Funcional 5. Generar archivo PDF.....	94
Tabla 7. Requerimiento Funcional 6. Generar archivo XML	94
Tabla 8. Requerimiento No Funcional 1. Consultas eficientes	94
Tabla 9. Requerimiento No Funcional 2. Uso del Sistema	94
Tabla 10. Requerimiento No Funcional 3. Transacciones	95
Tabla 11. Requerimiento No Funcional 4. Usuarios con nombre	95
Tabla 12. Requerimiento No Funcional 5. Contraseñas	95
Tabla 13. Requerimiento No Funcional 6. Privilegios.....	95
Tabla 14. Requerimiento No Funcional 7. Bitácoras.	95
Tabla 15. Requerimiento No Funcional 8. Excepciones	96
Tabla 16. Requerimiento No Funcional 9. Recuperación de fallos	96
Tabla 17. Requerimiento No Funcional 10. Disponibilidad	96
Tabla 18. Requerimiento No Funcional 11. Inicial.....	96
Tabla 19. Requerimiento No Funcional 12. Reportes Generados.....	97
Tabla 20. Requerimiento No Funcional 14. Sistemas Operativos.....	97
Tabla 21. Requerimiento No Funcional 14. Lenguaje de programación.	97
Tabla 22. Requerimiento No Funcional 15. Bases de datos.	97
Tabla 23. Otro requerimiento 1. Legalidad de datos	98
Tabla 24. Otro requerimiento 2. Interfaz	98
Tabla 25. Especificaciones del Hardware.	99
Tabla 26. Especificaciones del Software.....	99
Tabla 27. Costo Recursos Tecnológicos.....	100
Tabla 28. Costo Recursos Humanos.	100
Tabla 29. Caso de Uso 1. Ingresar al sistema.....	103
Tabla 30. Caso de Uso 2. Consultar Factura.....	104
Tabla 31. Caso de Uso 3. Consultar Proveedor.....	105
Tabla 32. Caso de Uso 4. Generar Reporte Facturas.....	107
Tabla 33. Normalización del Modelo Entidad-Relación.....	117
Tabla 34. Diccionario de Datos Factura.	119
Tabla 35. Diccionario de Datos Proveedor.....	120
Tabla 36. Ocurrencias de los gestos.....	163

I. Resumen

Superlberia es una empresa dedicada a la venta de productos de Abarrotes, vinos y licores; en sus inicios empezó como una tienda de mostrador ubicada en la colonia centro de la ciudad de Córdoba Veracruz con la cantidad mínima de empleados, gracias a los avances tecnológicos y sociales Superlberia cuenta actualmente cuenta con 21 tiendas Vecinos y con más de 500 empleados.

“Hoy en día muchos negocios requieren de un software que facilite sus operaciones ya existen muchos en el mercado; sin embargo, éstos no cubren todos los procesos que una empresa requiere para optimizar” (Torres Muñoz P. &., 2014)

Un gran beneficio de implementar un sistema web a medida es que está programado y optimizado para dar solución, es decir; se amolda a las necesidades reales de la empresa y que puede utilizarse para las diferentes operaciones y/o funciones que se llevan a cabo en todas áreas tanto administrativas, contables y operacionales. (Guerrero-Luzuriaga, 2018)

En tiempos actuales la mejor forma para hacerlo es mediante la implementación de un sistema web, que brinde soporte y pueda facilitar las operaciones internas de la empresa.

Una de las problemáticas de la empresa Superlberia, que se detectó fue dentro de las operaciones administrativas que llevaba a cabo el departamento de conciliaciones muy puntualmente en la de revisión de Estados de cuenta, se visualizó que los usuarios contaban con demasiadas deficiencias en la obtención de información ya que no tenían acceso completo a la información del repositorio de documentos fiscales digitales (CFDI'S) es decir que únicamente tenían acceso a la visualización de la información y no a la manipulación de dicha información.

Ya que toda esta información se tenía que solicitar mediante un correo electrónico al departamento de sistemas quienes eran la única área que podían manipular la información del repositorio de facturas validadas.

Como una mejora a esta situación se llevó a cabo la implementación de un sistema web, que les permitiera generar un reporte de todos los documentos en una hoja de cálculo (Excel), con la finalidad de que se pudiera manipular, esto para facilitar a los usuarios su operación en las revisiones pertinentes de estados de cuenta. (Domínguez, 2018).

Esto va a permitir a los usuarios cumplir en tiempo y forma con la entrega de reportes, programación de pagos a proveedor, pagos vencidos y mayor visibilidad de documentos o comprobantes (facturas) que no se tienen registrados en la base de datos o que el proveedor tiene en proceso de cancelación ante el SAT. (Chaves, 2005).

Llevando a los usuarios del departamento de conciliaciones de la Empresa a una mejora en la administración de documentos y facturas que reciben en su repositorio de facturas validadas.

II. Abstract

Superiberia is a company dedicated to the sale of grocery products and others; In its beginnings, it began as a counter store located in the center of the city of Córdoba Veracruz with the minimum number of employees, thanks to technological and social advances Superiberia currently has 21 Vecinos stores and more than 500 employees.

Today many businesses require software that facilitates their operations and there are many on the market; however, these do not cover all the processes that a company requires to optimize.

A great benefit of implementing a custom web system is that it is made to provide a solution, that is; It adapts to the real needs of the company and can be used for the different operations and/or functions that are carried out in all areas, both administrative and operational.

In current times the best way to do it is through a web system, to which all users have access.

One of the problems of the company Superiberia, which was detected was within the administrative operations carried out by the conciliation department very punctually in the review of account statements, it was seen that the users had too many deficiencies in obtaining information as they did not have full access to future repository information ie; that they only had access to the visualization of the information and not to the manipulation of said information.

Since all this information had to be requested by email to the systems department, who were the only area that could manipulate the information from the repository of validated invoices.

As an improvement to this situation, the implementation of a web system was carried out, which would allow them to generate a report of all the documents in a spreadsheet (Excel), so that it could be manipulated, this to facilitate the users their operation in the pertinent reviews of account statements.

This will allow users to comply in a timely manner with the delivery of reports, supplier payment scheduling, overdue payments and greater visibility of documents or receipts (invoices) that are not registered in the database or that the supplier is in the process of cancellation before the SAT.

Taking the users of the reconciliation department to an improvement in the administration of documents and invoices that they receive in their repository of validated invoices.

III. Introducción

“Un Sistema de Información es un conjunto de componentes interrelacionados que trabajan juntos para recopilar, procesar, almacenar y difundir información para apoyar la toma de decisiones de una Empresa o negocio”. (Trasobares, 2003)

“Todas las Empresas en esta época de la Tecnología, necesitan de un software o Sistema que administre su negocio o en pocas palabras que facilite sus operaciones diarias”. (Castellanos, 2003)

De acuerdo con Larrocha (2017) “Es importante mencionar que existen 3 tipos de sistemas en el mercado.

- Sistemas empacados
- Sistemas de uso específico
- Desarrollos personalizados”

“Los sistemas de usos específico que están dirigidos y hechos para empresas de ciertos giros particulares como hoteles, farmacias y hospitales; empresas que necesitan herramientas tecnológicas para objetivos muy particulares” (Mazzarella, 2018).

Las ventajas de este tipo de software son que se ajustan a un mercado de nicho, por la misma razón poseen mejoras específicas basadas en la experiencia de necesidades de este tipo particular de empresas. Sin embargo, una de sus desventajas es que normalmente requieren de una mayor inversión y termina siendo poco personalizable, ya que las mejoras que se le hagan dependerán de que las empresas o negocios de dicha industria que los soliciten. (Vargas, 2008)

“Los softwares empacados pueden ser útiles para una gran mayoría de empresas de diversos giros e industrias, además que pueden comenzar a operar

en pocos minutos, tiene mejoras masivas para todos y es más económico” (López, 2017).

Aunado a que una de las partes más importantes con las que debe contar un sistema administrativo es la consultoría, y esta suele ser especializada y completa para este tipo de softwares.

“Una de las desventajas de este tipo de software es que no es personalizable y por lo mismo, también habrá que ajustarse al sistema y su funcionamiento” (Cabello, 2015)

Por lo anterior es de gran importancia implementar un sistema para la Empresa ABARROTES CORDOBESES SA DE CV, capaz de emitir un reporte de toda la información de facturas Digitales que la empresa tiene almacenada en el repositorio del Sistema de facturas validadas.

Donde uno de los principales objetivos de este sistema es que permita a los usuarios generar un reporte con facilidad desde el Sistema Web, sin tener que solicitarlo al Departamento de sistemas siempre y cuando se tengan los privilegios para acceder.

“De esta forma se puede explotar dicha información en donde se busca agilizar procesos internos dentro de la empresa” (Muñoz, 2021).

“Por lo tanto, elegir correctamente los recursos y las herramientas de apoyo como son los sistemas de gestión administrativa hará una diferencia competitiva y harán crecer nuestro negocio tanto administrativamente como en clientes” (Ardiles, 2022).

IV. Definición del Problema

Abarrotes Cordobeses SA de CV (Superlberia) es una empresa de giro comercial, que se dedica a la venta de Abarrotes, poniendo a la venta un gran surtido de productos especialmente de canasta básica.

Al ser una mediana Empresa requiere de un Sistema que pueda administrar las facturas que le emiten sus proveedores.

Por lo que este proyecto surge de la necesidad del departamento de conciliaciones para agilizar sus procesos de revisión y conciliación, el objetivo es implementar un sistema que pueda proporcionarles un reporte de la información, que se carga en el repositorio de facturas digitales con la funcionalidad de que los usuarios puedan visualizar y tener acceso a la información de manera rápida y eficaz.

Por tal motivo que de NO contar en tiempo y forma con esta información los usuarios se ven retrasados, sus procesos administrativos y de revisión que deben de cumplir en su operación diaria.

Actualmente se trabaja de forma visual, ya que solo se cuenta con la información digital y nos hay forma de manipular la información que se encuentra en el repositorio de facturas, dicha información solo la puede proporcionar el departamento de Sistemas de la empresa, quienes son los que tienen acceso a la base de datos del sistema donde se almacena esta información.

Sin embargo, con la implementación de este Software podrán tener acceso a toda la información digital, que exista en el repositorio de facturas recibidas, además que los usuarios podrán acceder con gran facilidad a dicho sistema, ingresando con la misma cuenta de acceso que la empresa le designo para sus operaciones administrativas.

V. Objetivo General

Analizar, diseñar, desarrollar e implementar un Sistema Web, mediante el lenguaje de programación Java y NetBeans, que permita generar un Reporte detallado en formato Excel, para facilitar las operaciones diarias, de los usuarios del Área de Conciliaciones de la Empresa SuperIberia Abarrotes Cordobeses SA de CV.

VI. Objetivos Específicos

- ❖ Identificar los procesos internos de recepción de archivos electrónicos y la validación en el sistema.
- ❖ Analizar e identificar las operaciones administrativas que llevan a cabo las demás áreas de la empresa que harán uso del sistema.
- ❖ Organizar la información obtenida a partir de la recepción de archivos digitales y realizar un análisis, para llevar a cabo la funcionalidad del sistema.
- ❖ Realizar la validación de Comprobantes Fiscales, a partir del dato RFC de cada uno de los proveedores.
- ❖ Diseñar el sistema de acuerdo con las necesidades de los usuarios con las funcionalidades requeridas.
- ❖ Crear una herramienta idónea para mejorar la seguridad informática de los archivos electrónicos.
- ❖ Analizar los requerimientos de usuario.

VII. Justificación

El desarrollo de este proyecto está enfocado en la programación de un Sistema Administrativo, que permita llevar a cabo los procesos administrativos de la Empresa “Abarrotes Cordobeses Sa de CV”, el cual mediante la extracción de un Reporte pueda mejorar la administración de las facturas ya que Abarrotes Cordobeses recibe a diario una gran cantidad de facturas de todos los proveedores que tiene a cargo en cartera, por lo que administrar toda esta información digital de forma manual, le implica tiempo y costos en su administración.

Con la implementación de este Sistema, facilitará a los usuarios la obtención de esta información en una hoja de cálculo Excel, que mediante un reporte les permita realizar las revisiones pertinentes, consultas y ajustes de información. Así mismo podrán tener mayor visibilidad y seguimiento a las facturas recibidas en el repositorio de recepción de comprobantes fiscales tales como (egresos, ingresos, complementos de pago etc.) de la misma forma dichos usuarios podrán llevar a cabo en tiempo y forma sus operaciones administrativas, reduciendo tiempos puntualmente en la revisión de los Estados de Cuenta y programación de pago a proveedores.

Ya que anteriormente toda esta información se tenía que solicitar mediante un correo electrónico al departamento de sistemas ya que es la única área de la Empresa que tiene acceso para poder manipular la información de la base de datos, donde se encuentra almacenada esta información.

Además, que el sistema cumplirá con los requerimientos que solicita el SAT (Servicio de Administración Tributaria) de validación y recepción de documentos Fiscales Digitales (CFDI'S) cumpliendo así, con las obligaciones fiscales que se requieren.

VIII. Alcances y Limitaciones

El sistema será instalado en todos los equipos del área de conciliaciones a la cual solo podrán tener acceso los administradores.

- Lograr la funcionalidad del sistema para poder facilitar el Reporte de la información para la consulta de los usuarios de la Empresa.
- Ingresar con facilidad al sistema.
- Reducir tiempo y dedicación para poder obtener la información.
- Fiabilidad de los datos: el sistema debe funcionar perfectamente sin fallos y debe mantener la integridad de los datos.
- Generar reportes en lapsos de tiempo determinados para lograr así mejorar el tratamiento de la información.

Como limitantes:

- Depender de una conexión a internet: El servicio requiere de una buena conexión a internet todo el tiempo.
- Requiere soporte técnico.
- El proyecto se desarrollará únicamente para Abarrotes Cordobeses SA de CV.
- El sistema solo estará disponible en un servidor local de la empresa.
- El software solo funciona en el sistema operativo Windows.
- Se recomienda el funcionamiento del sistema en el Navegador Firefox o Chrome
- El programador no será responsable del mal uso o manejo que el usuario final le dé al software.

IX. Marco teórico

Hoy en día, quedan pocos negocios que no dispongan de un sitio web con el que presentarse a los clientes, es más, según la plataforma Live Stats, actualmente hay más de 1.500 millones de sitios web en la red mundial. De estos, menos de 200 millones están activos. El hito de 1000 millones de sitios web se alcanzó por primera vez en septiembre de 2014, según lo confirmó NetCraft en su *Encuesta de servidores web* de octubre de 2014 y lo estimó y anunció por primera vez Internet Live Stats (ver el tweet del inventor de la World Wide Web, Tim Berners-Lee). (Grande, 2001)

Como afirma Grande (2001) “Posteriormente, el número disminuyó, volviendo a un nivel por debajo de los mil millones (debido a las fluctuaciones mensuales en el recuento de sitios web inactivos) antes de volver a alcanzar y estabilizarse por encima de la marca de los mil millones a partir de marzo de 2016.”

Durante 2016, el número total de sitios ha crecido significativamente, de 900 millones en enero de 2016 a 1700 millones en diciembre de 2016. De 2016 a 2018, el nivel se ha mantenido prácticamente sin cambios. Desde 1 sitio web en 1991 hasta mil millones en 2014, y es que, en la actualidad el ser humano está constantemente dentro de internet y es de vital importancia para las empresas tener un buen posicionamiento e imagen digital a través del desarrollo sistemas web. (Acosta, 2017)

El primer sitio web (info.cern.ch) fue publicado el 6 de agosto de 1991 por el físico británico Tim Berners-Lee mientras estaba en el CERN, en Suiza. El 30 de abril de 1993, el CERN puso la tecnología World Wide Web ("W3" para abreviar) a disposición del dominio público sin regalías, lo que permitió que la Web floreciera. (Benlliure, 1995)

La World Wide Web fue inventada en marzo de 1989 por Tim Berners. También presentó el primer servidor web, el primer navegador y editor (el "WorldWideWeb.app"), el Protocolo de transferencia de hipertexto (HTTP) y, en

octubre de 1990, la primera versión del "Lenguaje de marcado de hipertexto" (HTML). (Tabarés Gutiérrez, 2012)

“Solo en 2013, la web creció más de un tercio: de unos 630 millones de sitios web a principios de año a más de 850 millones en diciembre de 2013 de los cuales 180 millones estaban activos” (Marín, 2006).

Según Rojas (2016) “El número de sitios web casi se duplicó: de 900 millones a 1.700 millones. Sin embargo, el recuento de sitios web activos más confiables se mantuvo estable en alrededor de 170 millones durante todo el año”.

Según Palma Pérez (2019) “Más del 50% de los sitios web actuales están alojados en Apache o Ngin, ambos servidores web de código abierto”.

A partir de junio de 2014, Microsoft se ha acercado mucho a Apache en términos de cuota de mercado (solo una diferencia del 0,15 % los separa). Si la tendencia continúa, Microsoft pronto podría convertirse en el principal desarrollador de servidores web por primera vez en la historia.

Vivimos en un mundo de objetos. Estos objetos existen en la naturaleza, en entidades y en los productos que usamos. Los objetos pueden ser clasificados, descritos, organizados, combinados y creados. Es por esto que se propuso un análisis de desarrollo orientado a objetos, que nos permita aprovechar las características, individualidad y facilidad de manipulación que nos proporcionan los objetos. (Moreno, 2014)

Objeto: Un objeto del mundo real es cualquier cosa que vemos a nuestro alrededor. Digamos que para leer este artículo lo hacemos a través del monitor y una computadora, ambos son objetos, al igual que nuestro teléfono celular, un árbol o un automóvil.

Clase: La clase es un modelo o prototipo que define las variables y métodos comunes a todos los objetos de cierta clase. También se puede decir que una clase es una plantilla genérica para un conjunto de objetos de similares características.

Método: Un método en Java es una porción de código que realiza tareas asociadas a un objeto, básicamente un método es una función que pertenece a un objeto o clase.

Peticiones: Una petición solicita una operación específica debe ser invocada usando uno o varios objetos como parámetros. (Aguirre, 2020)

Java

Java es un lenguaje de programación ampliamente utilizado para codificar aplicaciones web. Ha sido una opción popular entre los desarrolladores durante más de dos décadas, con millones de aplicaciones Java en uso en la actualidad. Java es un lenguaje multiplataforma, orientado a objetos y centrado en la red que se puede utilizar como una plataforma en sí mismo. (Arjona, 2004)

Como plantea Arjona (2004) “Java, es un lenguaje de programación rápido, seguro y fiable para codificar todo, desde aplicaciones móviles y software empresarial hasta aplicaciones de macrodatos y tecnologías del lado del servidor”

Algunos usos comunes de Java incluyen:

1. Desarrollo de juegos: muchos juegos móviles, de ordenador y de video populares se crearon en Java. Incluso los juegos modernos que integran tecnología avanzada, como el machine learning o la realidad virtual, se crean con tecnología Java.

2. Computación en la nube: Java a menudo se conoce como WORA: escribir una vez y ejecutar en cualquier lugar (por sus siglas en inglés “Write Once and Run Anywhere”), lo que lo hace perfecto para aplicaciones descentralizadas basadas en la nube. Los proveedores de la nube eligen el lenguaje Java para ejecutar programas en una amplia gama de plataformas subyacentes.

3. Macrodatos: Java se usa para motores de procesamiento de datos que pueden trabajar con conjuntos de datos complejos y cantidades masivas de datos en tiempo real.

4. Inteligencia artificial: Java es una fuente inagotable de bibliotecas de machine learning. Su estabilidad y velocidad lo hacen perfecto para el desarrollo de aplicaciones de inteligencia artificial como el procesamiento del lenguaje natural y el aprendizaje profundo.

5. Internet de las cosas: Java se ha utilizado para programar sensores y hardware en dispositivos de borde que pueden conectarse de forma independiente a Internet. (JAVIER, 2007)

Como afirma Mestras (1997) **Java** define la sintaxis y la semántica del lenguaje de programación Java. Esto incluye el vocabulario básico y las reglas utilizadas para escribir algoritmos, como tipos de datos inmutables, bloques if/else, bucles, etc. Las API son componentes de software importantes que se incluyen con la plataforma Java. Estos son programas Java prescritos que pueden conectar y reproducir la funcionalidad existente en su propio código. Por ejemplo, podría usar las API de Java para obtener la fecha y la hora, realizar operaciones matemáticas o manipular texto. Cualquier código de aplicación de Java escrito por un desarrollador generalmente combinará código nuevo y preexistente de las API de Java y las bibliotecas de Java.

La edición Java Standard es la plataforma central de programación de Java. Contiene todas las bibliotecas y API que cualquier programador necesita para el desarrollo de Java. Open Java Development Kit (OpenJDK) es la implementación gratuita y de código abierto de Java SE.

La edición Java Enterprise incluye todas las funciones de Java SE y funcionalidades adicionales para implementar software Java tolerante a errores, distribuido y de varios niveles. Las aplicaciones que exigen una escala inmensa, un sistema distribuido, acceso a bases de datos y servicios web generalmente se desarrollan en Java EE.

Java Micro Edition es la plataforma para desarrollar aplicaciones para dispositivos móviles y sistemas integrados como decodificadores. Java ME

proporciona un subconjunto de la funcionalidad de Java SE, pero también presenta bibliotecas específicas para dispositivos móviles. (Deitel, 2003).

El consultor David Chappell dijo: "Java desempeña un papel importante en el soporte del desarrollo de aplicaciones empresariales porque puede ejecutarse en cualquier plataforma sin necesidad de compilación, y el ecosistema de herramientas y marcos está maduro para que Java pueda ejecutarse sin cambios en diferentes entornos.

Es particularmente útil para aplicaciones que necesitan ejecutarse en muchos sistemas operativos y dispositivos". Una característica que es importante Java también juega un papel importante en aplicaciones grandes debido a su estabilidad y seguridad, como se indica en el informe de Red Hat, "El uso de Java de escritura estática patrones. y la gestión automática de la memoria ayuda a reducir los errores operativos que podrían comprometer la seguridad de las aplicaciones".

Los IDE son aplicaciones informáticas que proporcionan servicios integrales para facilitar el trabajo a un desarrollador de software. Suelen integrar un editor de código, herramienta de construcción automática, un depurador junto con funciones de autocompletado de código y un intérprete. Entre los más conocidos y extendidos se encuentra NetBeans.

Este ID es el más usado ya que gracias a su enfoque podemos desarrollar nuestras aplicaciones como módulos (como un componente de un software). Lo cual resulta sumamente cómodo, pues podemos reutilizar nuestro código para diferentes proyectos las veces que queramos sin mayor complicación.

Otra de sus ventajas es que no solo podemos desarrollar en JAVA, también podemos trabajar desarrollando software bajo otros lenguajes como PHP, C, C++ e incluso HTML 5.

NetBeans es el IDE oficial de JAVA y uno de los programas más importantes que todo programador Java y de tecnologías web utiliza con regularidad (Mendoza González, 2015).

Hoy en día, muchas empresas relacionadas con la tecnología tienen la necesidad de medir la productividad de sus empleados, con la finalidad de mejorar su rendimiento en la industria. En el ámbito académico, también se presenta este tipo de necesidad, pero enfocada al desarrollo técnico de los estudiantes. Ureta Muñoz, R. A. (2020)

Sobre todo, en carreras como Ingeniería Civil en Computación o Informática. Concretamente, en la carrera de Ingeniería Civil en Computación de la Universidad de Talca, existen cursos elementales de programación orientados a desarrollar el paradigma de la programación orientada a objetos mediante el trabajo individual o por equipos. En estos cursos por preferencia de los estudiantes se suele utilizar el entorno de desarrollo NetBeans. Ureta Muñoz, R. A. (2020)

Como afirma Aumaille (2002) “Un servidor de aplicaciones es una plataforma que proporciona los servicios definidos en la especificación de Java Enterprise. Arquitectura J2EE”.

A continuación, el autor Cobo, Á. (2005) describe los 5 Servidores de Aplicación Opensources

WidFly – JBoss Application Server

WidFly inicialmente conocida como JBoss Application Server, es un software de aplicación creado por JBoss, ahora desarrollado por Red Hat. WidFly está escrito en Java y se puede ejecutar en las mejores plataformas Java EE, se encuentra disponible en multi plataformas.

En terminos de API, WidFly soporta el último estándar en Java Enterprise (Java EE 7) y está basado en los mejores proyectos de software abierto tales como:

- Undertow (servidor web)
- Apache CXF (web services basado en SOAP)
- RestEasy (web services basado en REST)
- Infinispan (plataforma de Data Grid y Cluster)
- HornetQ (proveedor de JMS)
- JGroups (framework de comunicación Multicast)

En términos de arquitectura interna, WildFly (tal cual JBoss AS 7) está diseñado en una arquitectura modular basado en dos proyectos principales:

JBoss Module, maneja la carga de los recursos de clase en el contenedor

Modular Service Container (MSC), que proporciona una forma de instalar, des-instalar y administrar los servicios utilizados por el contenedor.

WildFly se puede ejecutar de dos formas: con un simple JVM y multi-JVM o modo domain.

- **GlassFish**

GlassFish es un servidor de aplicaciones de código abierto iniciado por Sun Microsystems para la plataforma Java EE y ahora patrocinado por la corporación Oracle.

El servidor GlassFish 4.0 está desarrollado a través de un proyecto de la comunidad de software en <http://glassfish.java.net/>. El proyecto GlassFish proporciona un proyecto estructurado para el desarrollo el cual permite tener disponibles las nuevas características de la plataforma Java EE en forma más rápido que en otros servidores.

GlassFish está construido en un núcleo modular basado en OSGi, GlassFish se ejecuta sobre la aplicación de Apache Felix. También se ejecuta con Equinox OSGi o Knopflerfish OSGi.

- **Apache Geronimo**

Apache Gerónimo es un servidor de software abierto que integra los mejores proyectos de software abierto creados para Java/OSGi el cual conoce las necesidades de programadores y administradores de sistema. Entonces la característica más importante de este servidor es que contiene una colección de los mejores proyectos de software abierto para hacer un servidor complementa funcional. Los componentes del núcleo de Jerónimo incluyen:

- Apache Tomcat (Servidor HTTP y contenedor de servlet)
- Jetty (Servidor HTTP y contenedor de servlet, alternativo a Tomcat)
- Apache ActiveMQ (proveedor de JMS 1.1)
- Apache OpenEJB (contenedor EJB)
- Apache OpenJPA (Implementación de Java Persistence API – JPA 1.0)
- Apache ServiceMix (Enterprise Services Bus)
- Apache CXF (framework de web services con una variedad de protocolos)
- Apache Derby (administrador de base de datos relacional con un soporte nativo para JDBC)
- Apache WADI (solución cluster y de balanceo de carga)

El núcleo de Geronimo es un framework para los servicios del núcleo y control de los componentes básicos del servidor. Todo en Geronimo es un GBean: contenedor, conector, adaptador, aplicación y más. Un GBean es la unidad mínima de Geronimo, el cual consiste en un conjunto de clases (.jar) y el plan de implementación.

- **JOnAS**

JOnAs es una implementación de software libre de la especificación de JEE para servidores de aplicación, desarrollada y alojada el consorcio ObjectWeb.

Las características de JOnAS lo certifican como un contenedor Java EE 6. Proporciona un contenedor de EJB a través de EasyBeans y está disponible con un servidor Tomcat y un servidor Jetty. JVM 6 esta soportado y los intentos de ejecutar en una pila libre con el classpath GNU son prometedores. Además de la pila de Java EE, JOnAS proporciona mecanismos de alto nivel para cluster y así proporcionar escalabilidad y alta disponibilidad.

Apache TomEE

Apache TomEE es la edición Java EE de Apache Tomcat, el cual combina varios proyectos Java incluyendo Apache OpenEJB, ApacheOpenWebBeans, Apache OpenJPA, ApacheMyFaces y otros. El proyecto tiene una certificación de la corporación Oracle como un servidor compatible del perfil web Java EE 6, que es una parte de toda la especificación Java EE. Básicamente ApacheTomEE es ensamblada de vanilla Apache Tomcat.

Como afirma, Serra Manchado (2013) **GlassFish** es un servidor de aplicaciones de software de código abierto, libre (bajo licencias CDDL* y GNU GPL) y gratuito que fue desarrollado por Sun Microsystems (ahora Oracle). Implementa tecnologías definidas en la plataforma Java EE de esta compañía, y permite ejecutar apps que soportan dicha especificación. (Serra Manchado, 2013)

Existe una versión comercial de GlassFish, es la denominada Oracle GlassFish Enterprise Server. Además, Oracle ha aportado el módulo de persistencia TopLink. Por otro lado, esta plataforma usa como base Sun Java System Application Server, un derivado del famoso Apache Tomcat con componentes adicionales como Grizzly, que usa Java NIO para proporcionar escalabilidad y mejor rendimiento (Serra Manchado, 2013).

Estos servicios van desde la ejecución de aplicaciones web sencillas, basadas en Servlets/JSP hasta aplicaciones distribuidas con soporte transaccional, basadas en Enterprise Java Beans (EJB) o servicios Web. Hasta ahora hemos estado trabajando con Apache Tomcat, pero ¿es un servidor de aplicaciones? Tomcat implementa un contenedor Web que en su versión 7 cumple las especificaciones Servlet 3.0, JSP 2.2, y EL 2.2, pero no incluye un contenedor de EJB's o da soporte a otras APIs como JMS, por tanto, no se puede considerar como un servidor de aplicaciones completo (MUÑOZ MUÑOZ, 2015).

“Lo que sí es cierto es que de la suma de Tomcat más Spring, junto con las librerías de terceros adecuadas, resulta algo parecido a un servidor de aplicaciones, cambiando los beans de Spring por EJB's” (Lahiguera Hervás, 2021).

Como afirma Marañón Martín (2014) La ventaja de un servidor de aplicaciones es que aglutina todos los componentes necesarios para empezar a trabajar, permitiendo al programador centrarse en el desarrollo de aplicaciones. Esto no quiere decir que no podamos utilizar las librerías que creamos oportunas, pero el producto ya nos ofrece una solución válida en conjunto.

López J. G (2022) Señala que Para un administrador de sistemas también es más sencillo pues al ser un producto estándar, puede trabajar sobre el servidor sin necesidad de conocer los entresijos de las aplicaciones que ejecutan. ¿Debemos entonces suponer que un servidor de aplicaciones completo es la mejor solución para todos los casos?

En absoluto, por un lado, tenemos el coste de licencias (si se opta por una solución comercial), y por otro la ligereza: un contenedor Web como Tomcat consume menos recursos y es mucho más sencillo de administrar que cualquier servidor de aplicaciones, y una vez desarrollada la infraestructura necesaria para trabajar con un framework como Spring, tenemos la base para desplegar aplicaciones empresariales de gran complejidad. Lo que nos debe llevar a pensar en utilizar un servidor de aplicaciones es considerar que cuestiones como la escalabilidad, la alta disponibilidad, la monitorización del

rendimiento, transaccionalidad y seguridad son tan importantes en nuestros desarrollos que merece la pena invertir en una infraestructura que potencie estas características y sea una solución probada y estandarizada. (López J. G.)

De acuerdo con Niño (2011) En la actualidad, los servidores de aplicaciones más utilizados son los siguientes:

- BEA WebLogic (JEE 5.0 y JEE 6.0 en versión 12c, Comercial, Oracle)
- JBoss (JEE 6.0 Open Source, RedHat)
- GlassFish (JEE 6.0 Open Source y Comercial, Antes Sun, ahora Oracle)
- Websphere (JEE 6.0, Comercial, IBM)

“GlassFish además de ser multiplataforma, pudiéndose instalar en Linux, Windows o Mac Os X. Se caracteriza por ser la implementación de referencia de Java EE y un servidor de aplicaciones relativamente rápido, ligero y fácil de administrar” (Serra Manchado, 2013).

El servidor GlassFish soporta tecnologías como JSP, JSF, Serverlets, EJBs, Java API, JAXB, JPA, RMI, etc. Eso permite a los desarrolladores contar con una gran plataforma para desarrollar apps escalables y portables. Además, no resulta difícil configurar un cluster con carga balanceada para comenzar a exprimir este proyecto en tu propio servidor local o contratado.

GlassFish implementa la plataforma Java EE (Enterprise Edition es una estandarización para el desarrollo de aplicaciones Java empresariales que sean portables, es decir, que pueden ser usadas en cualquier servidor sin importar la plataforma y sin modificaciones. (Goncalves, 2009)

“Actualmente, GlassFish se encuentra en su versión 5.x, que es la última estable. No obstante, tienes versiones más recientes en la Eclipse Foundation (actual desarrolladora)” (Heffelfinger, 2014).

Las aplicaciones web se han hecho populares debido a las diferentes ventajas que generan para las entidades que las utilizan, dando lugar al mejoramiento de la calidad de los procesos en los que se ven involucradas.

Actualmente las Unidades Tecnológicas de Santander cuenta con un sistema para la gestión de funciones docentes que es basado en un documento Excel R-DC-54 en el cual los docentes definen sus actividades desarrolladas durante el curso, el número de horas que se dedican a cada actividad, su información personal y una lista de productos de acuerdo con el número de actividades que desarrollan en función de su tipo de vinculación “(Sánchez, 2020)”.

Roles Java Enterprise

“La especificación Java Enterprise define una arquitectura modular diferenciando claramente entre Servicios (App Server) y Componentes de Negocio (EJB, Servicios Web, etc). Esta diferenciación fomenta la especialización de los distintos grupos de trabajo implicados en un proyecto Java Empresarial” (Groussard, 2010)

De hecho, la especificación define una serie de roles de trabajo, que pueden ser adoptados por una persona o grupo de trabajo:

- ❖ **Tool Provider** Su función es la de seleccionar las herramientas que se utilizarán en la construcción del software (Platform independent tools) y las herramientas específicas de despliegue, administración y monitorización de aplicaciones (Platform dependent tools).

- ❖ **Application Component Provider** Es el encargado de construir los diferentes módulos de las aplicaciones Java Enterprise. Debe tener conocimientos de negocio y experiencia en desarrollar componentes reutilizables. Estos proveedores no tienen por qué tener conocimiento técnico de la arquitectura del sistema que sustentará sus aplicaciones. Dentro de este rol nos encontramos a los programadores, diseñadores web, desarrolladores de

objetos de negocio, etc. Los proveedores, para su trabajo utilizarán herramientas suministradas por el Tool Provider.

- ❖ **Application Assembler** Su labor es la de empaquetar los distintos componentes que conforman una aplicación, generando los descriptores de despliegue correspondientes, censando librerías o dependencias externas, y definiendo las instrucciones necesarias para su correcto despliegue y funcionamiento. De igual forma que el anterior rol, no precisan de conocimientos técnicos concretos sobre la plataforma donde se va a ejecutar.

- ❖ **Deployer** Es un experto en el entorno operativo y es el encargado de desplegar los componentes y aplicaciones J2EE partiendo de las instrucciones proporcionadas por el Application Assembler. Asimismo, es el responsable de configurar cada componente/aplicación y sus dependencias.

- ❖ **System Administrator** Es el responsable de la configuración y administración de la arquitectura hardware, software y comunicaciones. Asimismo, es el encargado de monitorizar el correcto funcionamiento de las aplicaciones J2EE, en el ámbito de la técnica de sistemas.

- ❖ **Product Provider** Bajo este rol encontramos los distintos proveedores de software: sistemas operativos, bases de datos, servidores de aplicaciones, etc. que conforman el ecosistema de nuestra organización. Junto con estos productos suministran las herramientas necesarias para el despliegue y monitorización de aplicaciones. (Franky, (2010, April))

El proyecto GlassFish nace en 2005 de la mano de Sun con el objetivo de construir un servidor de aplicaciones que cumpliera con la especificación de Java Enterprise del momento. Pero no se parte desde cero, se utiliza como base el proyecto Sun Java System Application Server, a su vez basado en Tomcat. (Vasiliev, 2008)

La versión 1.0 sale a la luz en 2006, ofreciendo soporte a la especificación JEE 5.0. Más tarde, en septiembre de 2007 llegaría la versión 2 con capacidades de clúster siendo ésta una de las versiones más extendidas hasta la fecha.

Esta versión resulta interesante para las empresas ya que se ofrece la misma aplicación con dos licencias distintas, la Open Source Edition y la comercial Enterprise Server. Este doble licenciamiento se mantiene hasta la actualidad. (Serra Manchado, 2013)

En diciembre de 2009 Sun presenta la especificación JEE 6.0 y su implementación de referencia GlassFish v3. Al margen de las novedades que presenta la especificación, abandona su antigua arquitectura, heredada de Tomcat, apoyándose en un núcleo modular, que implementa la tecnología OSGi. (estándar para diseñar plataformas que proporcionen múltiples servicios) (Serra Manchado, 2013)

Lamentablemente el esfuerzo que conlleva implementar la nueva especificación y arquitectura provoca que características como la configuración en clúster no se incluyan en esta versión.

“Tras la adquisición de Sun por parte de Oracle se establece una nueva hoja de ruta para GlassFish en la que se puede observar una clara alineación entre Glasfish y la suite comercial de Oracle, la Oracle Fussion MiddleWare” (Almeida Moyano, 2009)

En febrero de 2011 se presenta la versión 3.1 que recupera las características de clustering y alta disponibilidad, y actualmente se está trabajando en la versión 4.0 con el objetivo principal de ser la implementación de referencia de JEE 7, así como dar soporte a la virtualización y a la denominada Platform as a Service (PaaS). (Moreira Zambrano, 2017).

El servidor web permite la comunicación distribuida entre los clientes y el servidor utilizando los protocolos HTTP/HTTPS. Por su parte, el Object Request Broker (ORB) integrado en GlassFish, hace posible invocar métodos de objetos remotos mediante los protocolos RMI/IIOP y RMI/IIOP/SSL, serializando los parámetros de entrada y la salida del método. (Cabello A. L., 2015)

La selección del mejor Servidor de Aplicaciones es un importante aspecto para los desarrolladores, quienes han de tomar en cuenta su uso fácil,

velocidad, y el soporte técnico que estos ofrecen. El auge en el desarrollo de aplicaciones utilizando tecnología J2EE, debido a su apertura de código y su libre distribución, hacen que los desarrolladores elijan un servidor de aplicaciones que se adapte a esta tendencia, por lo que JBOSS es el servidor de más aceptación.

Por otra parte, GLASSFISH es un Servidor de Aplicaciones que posee características similares a las de JBOSS, por lo tanto, es necesario determinar cuál de estos se adapta para obtener un mejor desempeño en las aplicaciones Web. (Moyano & López, 2009)

A su mismo nivel se encuentra el soporte para Presentación (Java Server Faces), Persistencia (JPA) Servicios Web (Metro) y mensajería (JMS).

En la siguiente capa nos encontramos el núcleo GlassFish, basado en el estándar OSGi, junto con sus servicios más importantes. (Palacios, 2018)

Estos servicios son accesibles tanto desde el servidor web como desde el contenedor EJB. Entre ellos destacan

- ❖ **Servicio de nombres:** un servicio de nombres asocia (binds) objetos java a nombres lógicos. Estos nombres pueden ser utilizados por las aplicaciones para obtener acceso a los objetos. El servicio que permite realizar este proceso se denomina JNDI (Java Naming and Directory Interface).

- ❖ **Seguridad:** La tecnología de seguridad definida por el estándar Java EE es la denominada JACC (Java Authorization Contract for Containers). Dependiendo de la identidad del cliente, se permite el acceso a los distintos recursos de los contenedores y de los servicios.

- ❖ **Gestión de transacciones:** el servicio JTA (Java Transaction API) da soporte a la utilización de transacciones distribuidas que pueden englobar distintas bases de datos gestionadas por distintos servidores de aplicaciones.

- ❖ **Clustering (asociación):** Los clusters permiten asociar máquinas y servidores para que actúen de forma conjunta como una única instancia. La

creación de un cluster va a permitir el balanceo de carga y la recuperación frente a fallos.

- ❖ **Consola de administración:** GlassFish proporciona una completa consola de administración (aplicación web) desde la que controlar y configurar el servidor de aplicaciones. Otros conceptos que utilizaremos a lo largo del bloque:

- ❖ **Servidor proxy:** Centraliza peticiones de los clientes y las reenvía hacia otras máquinas. Puede servir como nivel de indirección (desacomplando aplicaciones clientes de servidores) y como control de seguridad. También puede ser usado para realizar balanceo de carga.

- ❖ **Cortafuegos (firewall):** Proporciona servicios de filtrado, autorización y autenticación. Puede actuar como proxy y ayuda a manejar los ataques de los hackers.

- ❖ **Máquina:** Representa una unidad física donde reside un servidor. Una máquina se define como tipo Unix o no Unix (Windows NT, etc.).

- ❖ **Servidor:** Un servidor es una instancia de la clase Server ejecutándose dentro de una máquina virtual de Java. Un servidor está alojado en una máquina, pero una máquina puede contener varios servidores.

- ❖ **Balanceo de carga:** Es una técnica utilizada para distribuir las peticiones entre varios Servidores, configurados en clúster, de tal forma que todos los servidores respondan al mismo número de peticiones.

- ❖ **Recuperación ante fallos (failover):** Permite evitar la caída de un sistema cuando una máquina deja de funcionar o funciona incorrectamente.

- ❖ **Puerto de escucha:** Un servidor tiene varios puertos por los que puede "escuchar" las peticiones. Existen puertos ya asignados a aplicaciones concretas, como por ejemplo el puerto de http que suele ser el 80. Los puertos permiten que varias aplicaciones puedan atender distintas peticiones en la misma máquina. (Forero Cano, 2011)

Un puerto en una dirección se especifica de la siguiente manera: `http://localhost:7001/direc`. En este caso 7001 es el puerto al que estamos atacando. Los puertos del 0 al 1023 son reservados por el sistema. Podemos

disponer de los puertos del 1024 al 65536. Hay que tener en cuenta que dos servicios no pueden estar escuchando en el mismo puerto. (Luján-Mora)

Desde el punto de vista de Vibes, F. P. (2003) “**Un dominio** define un conjunto de propiedades, recursos e instancias de servidores de aplicaciones. La definición de dominios permite flexibilizar y organizar la instalación de servidores de aplicaciones, ya que es posible asignar distintos dominios dentro de un mismo host a distintas organizaciones y administradores.”

El instalador de GlassFish crea un dominio por defecto llamado domain1 y en él crea un servidor llamado server. Con un servidor en el dominio ya podemos empezar a trabajar desplegando aplicaciones, pero en instalaciones relativamente complejas es habitual encontrarnos con que un dominio alberga varios servidores que pueden estar repartidos a su vez por distintas máquinas. Las aplicaciones y recursos instaladas en un dominio son independientes de los otros dominios. Para permitir que distintos dominios puedan estar en marcha al mismo tiempo, cada dominio utiliza distintos puertos de servicio y administración. El servidor por defecto también tiene una serie de funciones administrativas especiales, y por ese motivo también se le denomina Domain Administration Server (DAS). Cuando se inicia un dominio, el DAS es el encargado de desplegar la consola de administración y de sincronizar la configuración entre todos los servidores administrados. (Serra Manchado,2013)

Toda la configuración del dominio se guarda en el fichero domain.xml situado en el directorio domains/_dominio_/config. En el fichero se encuentran todos los detalles sobre los distintos elementos definidos en el dominio (como servidores, agentes de nodo, aplicaciones desplegadas, recursos, servicios, etc.) así como características de la configuración básica (puertos y hosts en los que se ejecutan los distintos servicios, ficheros de contraseñas, etc. (Serra Manchado,2013)

La Consola de Administración es una aplicación Web que permite realizar gran parte de administración de GlassFish de forma visual, sin tener que editar complejos ficheros de configuración y con ayuda en línea bastante

útil. Para que funcione esta consola, el DAS debe estar en ejecución. Para acceder a esta consola basta con ejecutar en el navegador `http://hostname:port` siendo hostname el nombre de la máquina en la que reside GlassFish, y port el puerto del DAS (habitualmente 4848). Por defecto, la cuenta de administrador no tiene contraseña, pero se puede establecer una desde la propia consola. (Aumaille,2002).

“asadmin es una herramienta de administración en modo línea de comandos, por tanto, será la preferida de los administradores de sistemas” (Heffelfinger D. , 2014)

Tiene dos formas de trabajar:

Comandos individuales: (Single Mode) en la que la por cada llamada se ejecuta un comando. Ejemplo:

```
$ asadmin list-jdbc-resources
```

Con esta instrucción mostraremos una lista de recursos e inmediatamente regresaremos al Shell. Encadenando comandos de esta forma podremos automatizar tediosas tareas de configuración en un sencillo Shell script.

- **Modo interactivo:** desde línea de comandos ejecutaremos asadmin y accederemos a la herramienta en modo interactivo. Nos presentará un prompt y nos permitirá ejecutar todos los comandos de administración que necesitemos. Ejemplo:

```
$ asadmin Utilice "exit" para salir y "help" para obtener ayuda en línea  
asadmin>
```

- **Servicios REST** Glasfish proporciona una serie de servicios REST que permiten monitorizar el sistema y acceder a su configuración. Este acceso puede ser muy útil para ser invocado desde otras aplicaciones.

- **Update Tool** Esta herramienta es la encargada de gestionar los módulos instalados o las actualizaciones pendientes de aplicar en GlassFish. Se puede invocar desde línea de comandos, o bien desde la propia consola.

- **Java Management Extensions (JMX)** GlassFish es capaz de publicar sus componentes internos mediante objetos denominados MBeans y exponerlos mediante el API estándar de Java, que es JMX. Mediante este API podremos acceder a la configuración y monitorizar el funcionamiento del servidor.

- **IDE** No es una herramienta de GlassFish en sí, pero tanto Eclipse como NetBeans se integran perfectamente con GlassFish y permiten arrancarlo/pararlo, desplegar aplicaciones y establecer algunas propiedades de configuración. (Heffelfinger D. R., 2014).

Administración básica de dominios en GlassFish

1. Iniciar dominio

```
$ asadmin start-domain dominio1
```

Este comando iniciará el dominio1 con todos sus servidores

2. Parar dominio

```
$ asadmin stop-domain dominio1
```

Tras ejecutarlo, el dominio que especifiquemos dejará de estar activo, finalizando sus correspondientes procesos.

3. Crear un dominio

```
$ asadmin create-domain --adminport 5000 dominio2
```

Crea un dominio con perfil de desarrollador con el nombre dominio2 administrado en el puerto 5000. El script solicitará el login de administrador y la contraseña para acceder posteriormente a la consola de administración del dominio.

4. Eliminar dominio

```
$ asadmin delete-domain domain1
```

Con este comando eliminaremos la configuración completa de un dominio. Para que funcione, el dominio no debe tener ningún servidor en marcha.

5. Copia de seguridad de un dominio

```
$ asadmin backup-domain domain1
```

Este comando genera una copia de seguridad del dominio especificado, para poder restaurarla con posterioridad ante cualquier problema, o bien trasladar el dominio de una máquina a otra. Concretamente se genera un fichero ZIP que contiene toda la información del dominio en el directorio domain1/backups.

6. Restaurar un dominio

```
$ asadmin restore-domain domain1
```

Mediante este comando, podremos restaurar el dominio especificado a partir de copia de seguridad un (backup). El fichero de backup debe encontrarse en el directorio backups del dominio.

También es posible restaurar un dominio que se ha borrado completamente del directorio de dominios (o copiar un dominio en otra instalación del servidor de aplicaciones). Para ello hay que indicar el fichero ZIP con el domino comprimido y el nombre del dominio. En este caso se restauraría con la instrucción:

```
$ asadmin restore-domain --filename archive.zip dominio
```

Por último, hay que comentar brevemente que mediante los comandos create-jvm-options, delete-jvm-options podremos alterar la configuración de la máquina virtual de java en función de las capacidades de nuestra máquina y los requerimientos de las aplicaciones Java desplegadas en GlassFish. (Serra Manchado, 2013)

SQL

Como afirma García (2003). El Lenguaje de Consulta Estructurada (SQL) es un lenguaje gestor para el manejo de la información en las bases de datos relacionales. Este tipo de lenguaje de programación permite comunicarse con la base de datos y realizar operaciones de acceso y manipulación de la información almacenada.

El lenguaje de programación SQL no solo permite realizar operaciones de selección, inserción, actualización y eliminación de datos sino también operaciones administrativas sobre las bases de datos. Por tanto, se trata de un lenguaje completo de bases de datos que va más allá de la recuperación de la información. (Romero Martínez, 1999)

El SQL se usa para controlar todas las funciones que un sistema gestor de base de datos brinda a sus usuarios, proporcionando además un marco para crear la propia base de datos, gestionar su seguridad, actualizar sus contenidos, recuperar los datos y compartirlos entre diferentes usuarios. (Pérez Montoya, 2008).

Según González Menéndez, A. (2022). la mayoría de los sistemas de gestión de base de datos relacionales usan el lenguaje de programación SQL para interactuar con la base de datos. Oracle es una de las herramientas para la gestión de bases de datos más conocida y usada, pero también se encuentran Microsoft SQL, MySQL y Access, entre otros.

Como afirma Escofet (2002). El lenguaje SQL sirve para el acceso a la información almacenada en las bases de datos. Es un lenguaje sencillo de consulta, que permite realizar operaciones de selección, inserción, actualización y borrado de datos, así como operaciones administrativas sobre las bases de datos.

Como afirma Reinosa (2012) “SQL es un lenguaje orientado al manejo de bases de datos relacionales, relational database management system en inglés (RDBMS)”.

Como afirma Reinosa (2012). SQL es un lenguaje de computación para trabajar con conjuntos de datos y las relaciones entre ellos. Los programas de bases de datos relacionales, como Microsoft Office Access, usan SQL para trabajar con datos. A diferencia de muchos lenguajes de computación, SQL no es difícil de leer y entender, incluso para un usuario inexperto. Al igual que muchos lenguajes de computación, SQL es un estándar internacional reconocido por organismos de estándares como ISO y ANSI.

SQL se usa para describir conjuntos de datos que pueden ayudarle a responder preguntas. Al usar SQL, debe usar la sintaxis correcta. La sintaxis es el conjunto de reglas mediante las que se combinan correctamente los elementos de un idioma. La sintaxis SQL se basa en la sintaxis del idioma inglés y usa muchos de los mismos elementos que la sintaxis de Visual Basic para Aplicaciones (VBA) (Reinosa, 2012).

Diseñado para administrar y recuperar información de los sistemas de gestión de bases de datos relacionales, el lenguaje de programación SQL se caracteriza por el uso del álgebra y el cálculo relacional para realizar las consultas (Reinosa, 2012).

No obstante, lo que ha hecho que se popularice y consolide.

Lenguaje estandarizado. Estandarizado en 1986, el lenguaje de programación SQL permite desplegar implementaciones de este en diferentes sistemas, usando prácticamente el mismo código para todos ellos. Por ejemplo, el lenguaje SQL de Oracle no es idéntico al de Microsoft SQL Server, pero las diferencias son mínimas, una gran ventaja porque los comandos para crear la estructura de base de datos o manipular la información comparten la misma forma. (Ruiz, 2002).

Integridad de los datos. El SQL tiene la función de preservar la integridad de los datos, asegurándose de que sean válidos, correctos y completos. También protege la base de datos de corrupciones provocadas fallos del sistema o actualizaciones inconsistentes. (Ruiz, 2002).

Flexibilidad. Uno de los principales motivos del éxito del lenguaje de programación SQL es su flexibilidad. Es un lenguaje muy versátil a la hora de implantar soluciones y permite definir diferentes formas de ver los datos para satisfacer las demandas específicas de los usuarios. (Ruiz, 2002).

Sencillez y claridad. SQL es un lenguaje unificado, claro y de fácil comprensión. Su estructura incluye un elevado nivel de inglés, de manera que las sentencias en SQL son fáciles de entender y aprender. (Ruiz, 2002).

Instrucciones SQL

SQL

SELECT

Gabillaud (2015) Menciona que “Una instrucción SELECT contiene una descripción completa de un conjunto de datos que quiere obtener de una base de datos”.

Gabillaud (2015) Menciona que “Para describir un conjunto de datos con SQL, se escribe una instrucción SELECT. Se incluye lo siguiente:”

- Qué tablas contienen los datos.
- Cómo se relacionan los datos de orígenes diferentes.
- Qué campos o cálculos proporcionarán los datos.
- Criterios que los datos deben cumplir para ser incluidos.
- Si se deben ordenar los datos y, en caso de ser así, cómo deben ordenarse.

Cláusulas

“Como una frase, una instrucción SQL tiene cláusulas. Cada cláusula realiza una función de la instrucción SQL. Algunas cláusulas son necesarias en una instrucción SELECT”. (Gabillaud, 2015).

Comando	Descripción
FROM	Utilizada para especificar la tabla de la cual se van a seleccionar los registros.
WHERE	Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar.
GROUP BY	Utilizada para separar los registros seleccionados en grupos específicos.
HAVING	Utilizada para expresar la condición que debe satisfacer cada grupo.
ORDER BY	Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico.

Figura 1. Cláusulas de SQL

“En la figura anterior se enumeran las cláusulas SQL más comunes. Según” (Gabillaud, 2015).

Términos de SQL

“Cada cláusula SQL se compone de términos, comparables a elementos de la oración. En la siguiente Figura, se enumeran los tipos de términos de SQL”. (García de Figuerola Paniagua, 2009)

CLÁUSULAS DE SQL

CLÁUSULA SQL	FUNCIÓN	OBLIGATORIA
SELECT	Muestra una lista de los campos que contienen datos de interés.	Sí
FROM	Muestra las tablas que contienen los campos de la cláusula SELECT.	Sí
WHERE	Especifica los criterios de campo que cada registro debe cumplir para poder ser incluido en los resultados.	No
ORDER BY	Especifica la forma de ordenar los resultados.	No
GROUP BY	En una instrucción SQL que contiene funciones de agregado, muestra los campos que no se resumen en la cláusula SELECT.	Sólo si están estos campos
HAVING	En una SQL instrucción que contiene funciones de agregado, especifica las condiciones que se aplican a los campos que se resumen en la instrucción SELECT.	No

Figura 2. Tipos de términos de SQL.

Tal como lo expresa García, A. (2003). “El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos”.

Cláusula SELECT

Esta es la cláusula SELECT. Se compone de un operador (SELECT) seguido de dos identificadores ([Dirección de correo electrónico] y Empresa). Si un identificador contiene espacios o caracteres especiales (como "Dirección de correo electrónico"), se debe escribir entre corchetes. Una cláusula SELECT no tiene que mencionar las tablas que contienen los campos y no puede especificar las condiciones que deben cumplir los datos que se van a incluir. La cláusula SELECT siempre aparece antes que la cláusula FROM en una instrucción SELECT. (García, 2003).

Cláusula FROM

Esta es la cláusula FROM. Se compone de un operador (FROM) seguido de un identificador (Contactos). Una cláusula FROM no enumera los campos que se van a seleccionar. (Vegas, 1998).

Clausula WHERE

WHERE City="Seattle" Esta es la clausula WHERE. Se compone de un operador (WHERE) seguido de un identificador (Ciudad="Seattle").

Nota: A diferencia de las cláusulas SELECT y FROM, la cláusula WHERE no es un elemento obligatorio en una instrucción SELECT. Puede llevar a cabo muchas de las acciones que le permite realizar SQL con las cláusulas SELECT, FROM y WHERE. (Moreno F. J., 2005)



Figura 3. Lenguaje de consulta estructurada SQL

Capítulo 1 - Antecedentes de la Empresa

1.1 Historia

Superlberia (Abarrotes Cordobeses SA de CV) funda el negocio hace más de 100 años, primero como una pequeña tienda tradicional de mostrador, en el cual los clientes pedían su mercancía a la persona encargada del mostrador que despachaba cada uno de los productos solicitados en ese momento.

Con el paso de los años, Superlberia, por el mismo cambio que se da y el avance en todos los aspectos técnicos económicos y sociales, empieza a crecer aumentando el tamaño de tienda y ya con empleados; para poder satisfacer la demanda de los clientes, todo esto aunado a la visión, conocimiento y experiencia de su dueño el Sr. Fructuoso Solana González.

Cabe resaltar que Superlberia se encuentra en un plano competitivo contra las sucursales instaladas de las grandes cadenas de autoservicio que se encuentran en esta ciudad, ya que ha sabido satisfacer las necesidades de los clientes, mediante la calidad en su servicio, variedad de productos y principalmente sus precios bajos, ya que desde hace 100 años ha subsistido ante las variables económicas que se han presentado en este lapso de tiempo.

Gracias a los avances y el buen trato a sus clientes ya cuenta con sucursales Iberias VECINOS Ubicadas en los municipios de Córdoba, Fortín de las Flores, Amatlán, Orizaba, Mariano Escobedo e Ixtaczoquitlán.

1.2 Ubicación

El presente proyecto se realizará en la Empresa “SuperIberia - Abarrotes Cordobeses SA DE CV” Sucursal Matriz.

Las instalaciones de la Empresa se encuentran ubicadas en Calle 7 No. 609, Col. Centro, de la ciudad de Córdoba, Veracruz. C.P 94500.

(Véase ilustración 1)

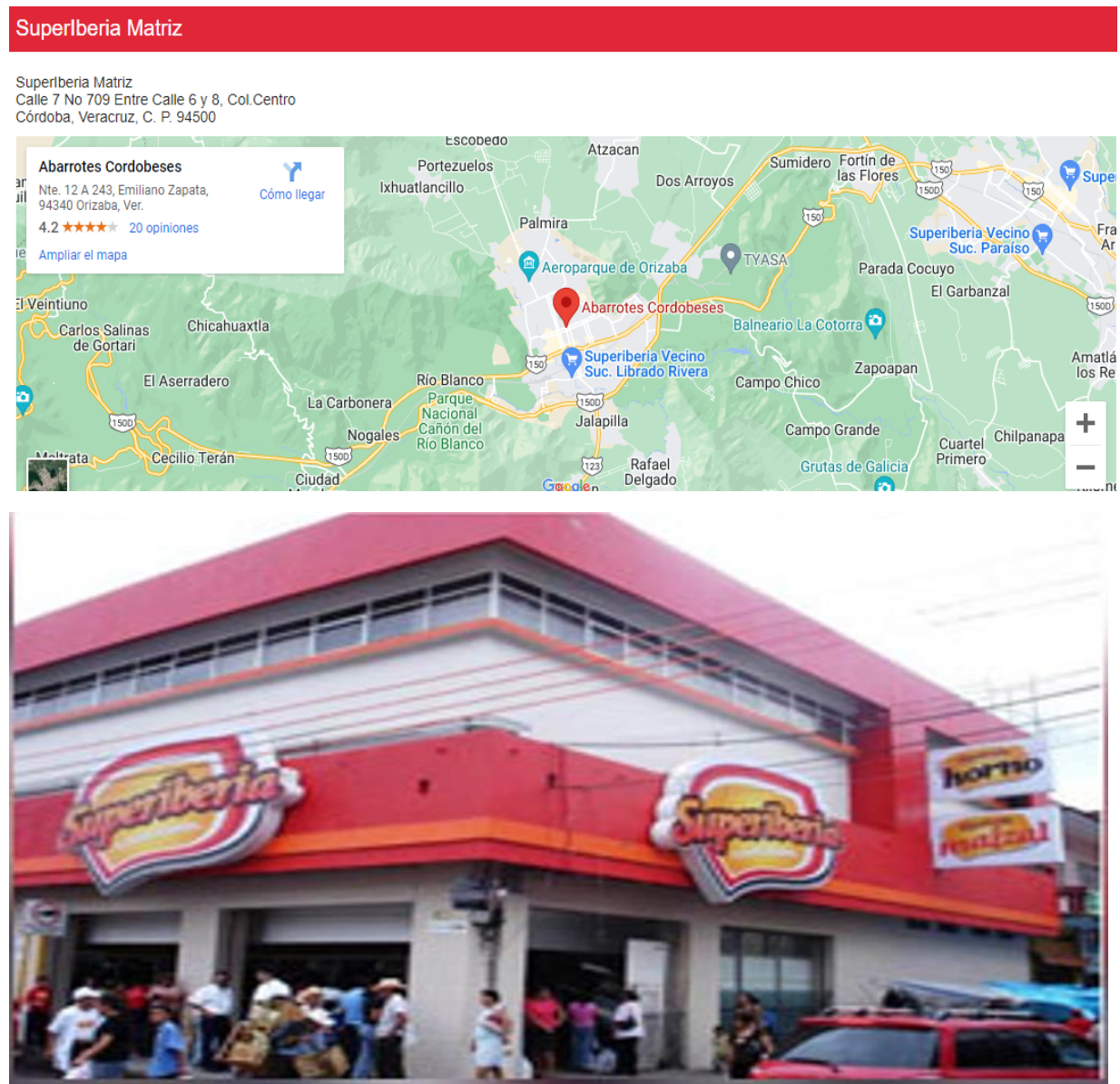


Figura 4. Edificio y ubicación Abarrotes Cordobeses Sa de CV.

1.3 Misión

Nuestra misión es la actualización continua de técnicas de comercio, en la búsqueda creativa que gestione rentabilidad creciente y sostenible, desarrollando con pasión nuestro talento humano.

1.4 Visión

Queremos ser la mejor cadena de venta de abarrotes que pueda ofrecer los mejores precios de la región.

1.5 Política de Calidad

SuperIberia

Es una compañía comprometida con todos sus clientes, uno de sus compromisos es brindar una buena calidad y excelencia en sus productos con el fin de servir, satisfacer y mantener una buena relación con todos sus clientes, además de ofrecer precios bajos todo el año, esto nos posiciona dentro del mercado como una de las mejores cadenas de autoservicios en ofrecer precios bajos todo el año cuyas acciones se están llevando a cabo basándonos en los siguientes valores.

-Efectividad

-Liderazgo

-Equipo

-confianza

-Pasión

-Rentabilidad

-Servicio al cliente con valor agregado.

1.6 Organigrama

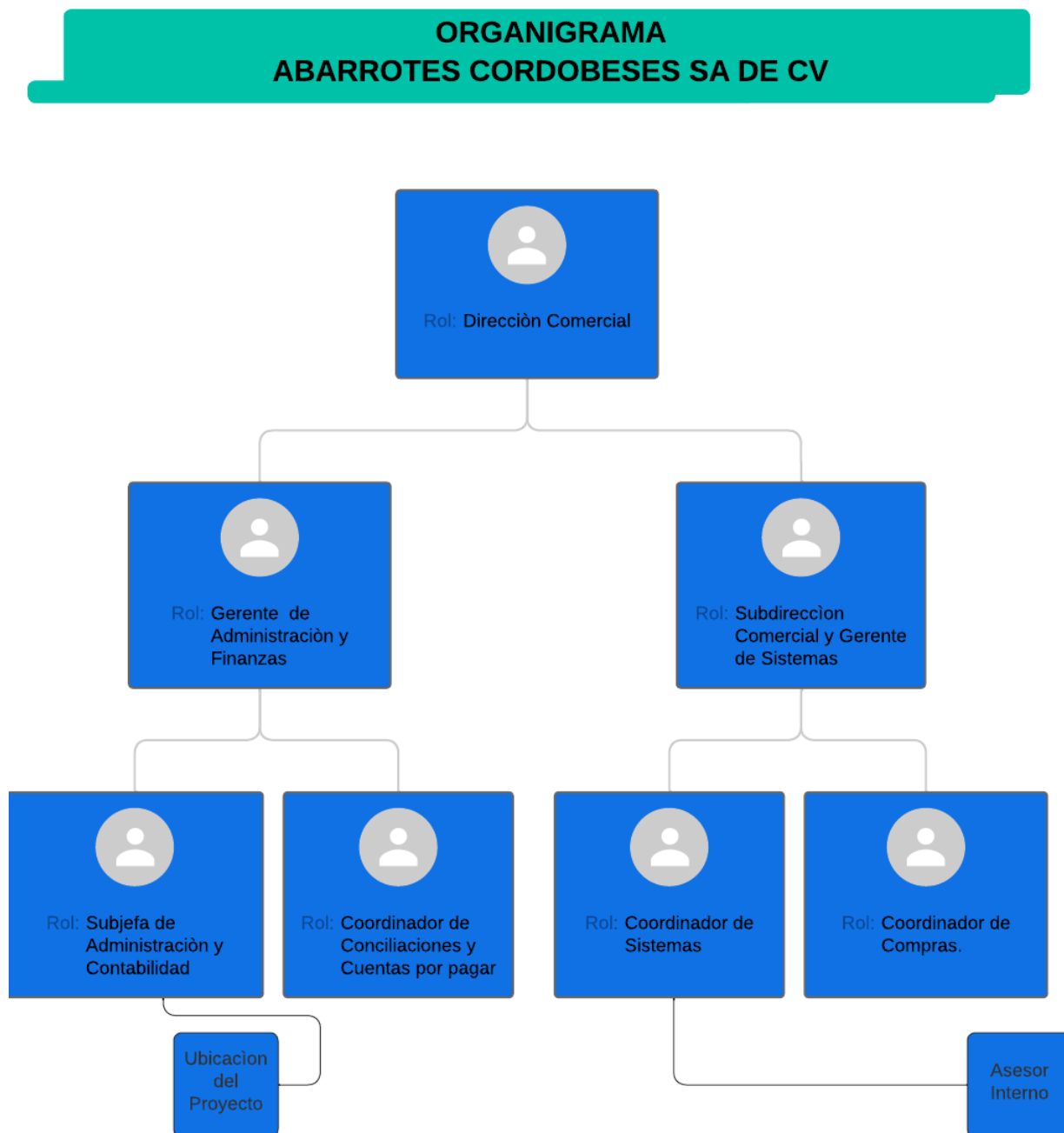


Figura 5. Organigrama de Abarrotes Cordobeses Sa de CV

1.7 Plano de la Empresa

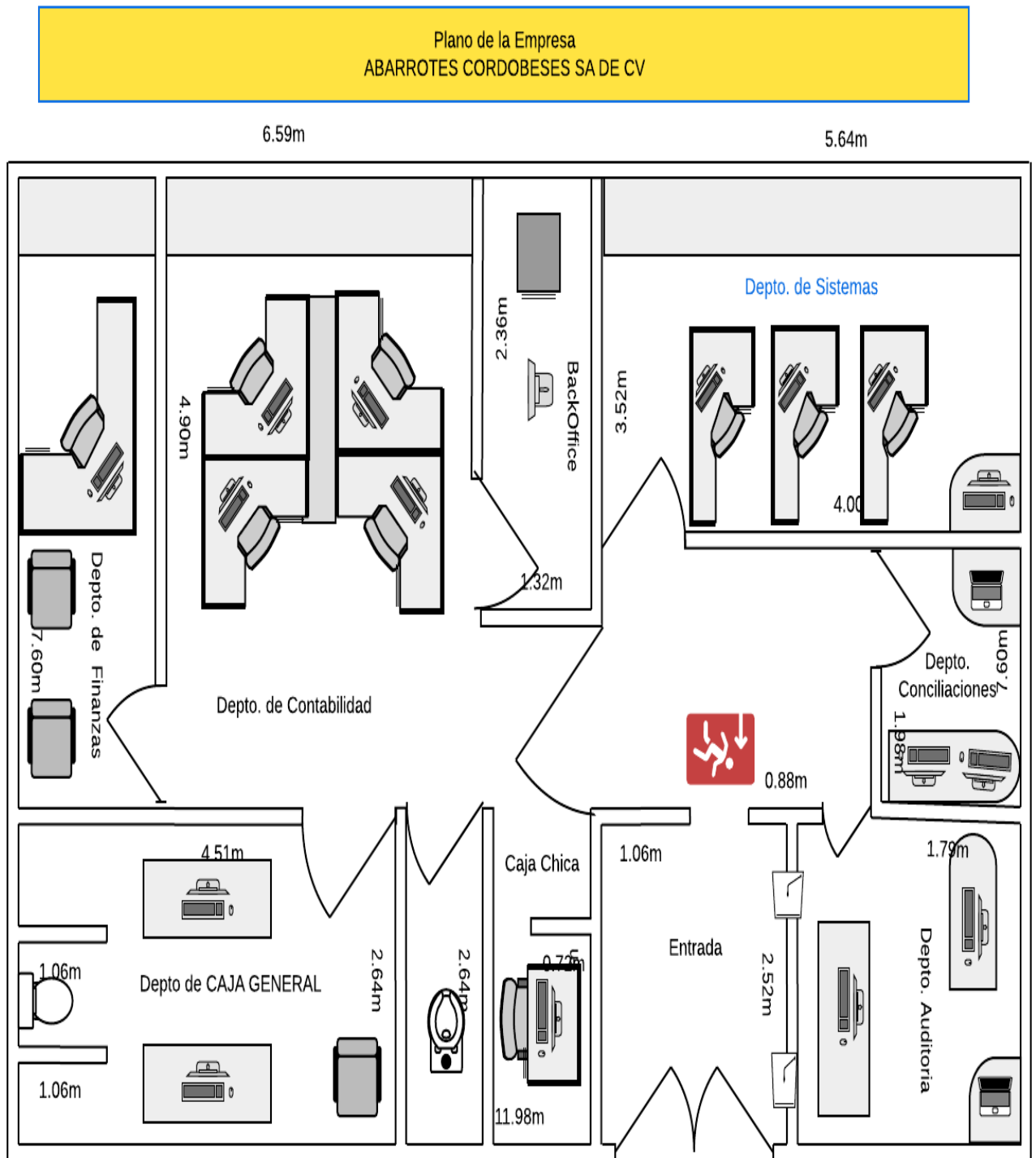


Figura 6. Plano de la Empresa Abarrotes Cordobeses

1.8 Caracterización del Área

Dentro de las funciones principales que se realizan en el departamento de Sistemas es brindar soporte técnico y mantenimiento a todos los equipos de cómputo de las áreas de la empresa además de dar soporte a los diferentes sistemas de trabajo, entre ellos el sistema administrativo y de operaciones como lo es QUALIS.

El equipo de trabajo “sistemas” cuenta con dos áreas; una de soporte y otra de desarrollo.

Para las operaciones en puntos de venta directamente para la recepción de Mercancía y devoluciones cuentan con el Sistema (SOI) que está basado en C# y .NET y como lenguaje de manejo de datos y sus bases de datos están en SQL Server.

Y QUALIS basado Visual Basic, que se utiliza para todas las operaciones administrativas y como lenguaje de manejo de datos y sus bases de datos están en SQL Server.

Además de contar con un excelente equipo de trabajo cuenta con un buen ambiente laboral.

Capítulo 2 - Metodología de Desarrollo

En la actualidad la rapidez y el dinamismo en la industria del software han hecho replantear los cimientos sobre los que se sustenta el desarrollo de software tradicional. Estudios recientes y el mismo mercado actual está marcando la tendencia en la ingeniería del software teniendo como características principales atender a las necesidades de rapidez, flexibilidad y variantes externas que hacen de nuestro entorno una ventaja más competitiva al aumentar la productividad y satisfacer las necesidades del cliente en el menor tiempo posible para proporcionar mayor valor al negocio. (Maida, 2015)

Ante esta situación, el grado de adaptación de las metodologías tradicionales a estos entornos de trabajo no eran del todo eficientes y no cubrían las necesidades del mercado actual. En la actualidad existen una gran cantidad de metodologías para el desarrollo de software, separadas en dos grandes grupos; las metodologías tradicionales o pesadas y las metodologías ágiles. Las metodologías tradicionales se basan en las buenas prácticas dentro de la ingeniería del software, siguiendo un marco de disciplina estricto y un riguroso proceso de aplicación. (Maida, 2015)

Las metodologías ágiles, en cambio, representan una solución a los problemas que requieren una respuesta rápida en un ambiente flexible y con cambios constantes, haciendo caso omiso de la documentación rigurosa y los métodos formales. El objetivo de esta investigación es presentar e introducir sobre las existentes metodologías para el desarrollo de software y los paradigmas que marcan la diferencia entre un método estructurado y un método ágil para así poder identificar cual se adapta de manera más eficiente a un proyecto determinado. (Maida, 2015)

La Metodología Scrum nos permite que se vaya desarrollando de manera ordenada y secuencial cada una de las funcionalidades del sistema, ya que día a día se va haciendo una pequeña programación de las actividades que se van a realizar y al final se analiza lo que se ha desarrollado. (Maida, 2015)

Según Darniame (1999), un modelo de procesos es una representación del mundo real, que captura el estado de actual de las actividades para guiar, reforzar o automatizar partes de la producción de los procesos. En el artículo de E. Georgiadou Software Procesos and Producto Improvement: A Historical Perspective, nos hablan de los modelos; V-Model, WModel, X-Model, RAD y Orientado a Objetos, sin embargo, los más conocidos son:

Modelo secuencial. Representado por metodologías tan famosas como Waterfall. Se inicia con un completo análisis de los requisitos de los usuarios. En el siguiente paso, los programadores implementan el diseño y finalmente, el completado y perfecto sistema es probado y enviado.

Desarrollo incremental. Su principal objetivo es reducir el tiempo de desarrollo, dividiendo el proyecto en intervalos incrementales superpuestos. Del mismo modo que con el modelo waterfall, todos los requisitos se analizan antes de empezar a desarrollar, sin embargo, los requisitos se dividen en incrementos independientemente funcionales.

Desarrollo iterativo. A diferencia del modelo incremental se centra más en capturar mejor los requisitos cambiantes y la gestión de los riesgos. En el desarrollo iterativo se rompe el proyecto en iteraciones de diferente longitud, cada una de ellas produciendo un producto completo y entregable.

Modelo en espiral. Comprende las mejores características de ciclo de vida clásico y el prototipado (desarrollo iterativo). Además, incluye el análisis de alternativas, identificación y reducción de riesgos.

Tabla 1. Metodologías De Desarrollo.

1. Modelo en Cascada (Waterfall)
• Descripción: Este es un enfoque secuencial donde cada fase del proyecto debe completarse antes de pasar a la siguiente. Las fases típicas son: requisitos, diseño, implementación, pruebas, y mantenimiento.
• Ventajas: Claridad y estructura, fácil de entender y gestionar.
• Desventajas: Rigidez ante cambios, dificultad para adaptarse a requisitos nuevos que surgen durante el desarrollo.

2. Desarrollo Ágil
• Descripción: Un conjunto de metodologías que promueven la flexibilidad, la colaboración continua con el cliente, y la entrega incremental del producto. En lugar de un plan rígido, se adaptan a los cambios y se enfocan en entregas frecuentes de valor.
• Ventajas: Adaptabilidad, alta colaboración, entrega continua de valor.
• Desventajas: Puede ser difícil de implementar en grandes equipos o proyectos con requerimientos muy específicos.
❖ Scrum: Utiliza iteraciones llamadas sprints, donde el trabajo se organiza en ciclos cortos y revisables. Cada sprint culmina en una entrega funcional del producto.
❖ Kanban: Se centra en la gestión visual del trabajo y en la mejora continua del flujo de trabajo. Usa tableros Kanban para gestionar el flujo de tareas.
❖ Extreme Programming (XP): Se enfoca en prácticas de desarrollo rigurosas como la programación en pareja y el desarrollo basado en pruebas (TDD) para mejorar la calidad del software.
3. Modelo en Espiral
• Descripción: Combina elementos del modelo Cascada y del desarrollo iterativo. Se basa en ciclos de desarrollo (espirales) que incluyen fases de planificación, análisis de riesgos, desarrollo, y evaluación. Se enfoca en la gestión de riesgos y la iteración continua.
• Ventajas: Adaptabilidad a cambios y enfoque en la gestión de riesgos.
• Desventajas: Puede ser costoso y complejo, requiere una planificación cuidadosa y análisis continuo.
4. DevOps
• Descripción: Un enfoque que integra el desarrollo de software (Dev) con las operaciones (Ops). Se centra en la automatización, la integración continua (CI), y la entrega continua (CD), con el objetivo de mejorar la colaboración entre desarrolladores y equipos de operaciones.
• Ventajas: Mejora la colaboración, acelera la entrega de software, y promueve la automatización.
• Desventajas: Requiere una inversión en herramientas y prácticas, y puede enfrentar resistencia cultural.
5. Lean Software Development
• Descripción: Basado en los principios del Lean Manufacturing, busca eliminar desperdicios y mejorar la eficiencia en el proceso de desarrollo. Se enfoca en entregar valor al cliente de manera continua y rápida.
• Ventajas: Reducción de desperdicios, enfoque en la eficiencia y el valor para el cliente.
• Desventajas: Puede ser desafiante implementar en organizaciones con procesos establecidos, requiere una cultura de mejora continua.
6. Modelo V
• Descripción: Una extensión del modelo Cascada que pone énfasis en la validación y

verificación a lo largo del proceso de desarrollo. Cada fase de desarrollo tiene una fase de prueba correspondiente.
• Ventajas: Mejora la calidad al integrar pruebas desde las primeras fases del desarrollo. • Desventajas: Menos flexible y adaptativo a cambios durante el proceso de desarrollo.
7. Rapid Application Development (RAD)
• Descripción: Un enfoque que enfatiza el desarrollo rápido del software mediante prototipos y la participación continua del usuario. Utiliza herramientas y técnicas para acelerar el desarrollo y obtener feedback temprano. • Ventajas: Entrega rápida y mejora continua basada en el feedback del usuario. • Desventajas: Puede ser difícil de gestionar en proyectos grandes, y el enfoque en prototipos puede llevar a un alcance mal definido.
8. Modelo Incremental
• Descripción: El desarrollo del software se realiza en incrementos o módulos, con cada incremento proporcionando una parte funcional del sistema. Los incrementos se desarrollan y se integran en fases. • Ventajas: Permite una entrega temprana y continua de partes funcionales del sistema. • Desventajas: La integración de incrementos puede ser compleja y requiere una planificación cuidadosa.

Metodologías Ágiles

En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término “ágil” aplicado al desarrollo de software.

En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto.

Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. Tras esta reunión se creó The Agile Alliance, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida es fue el Manifiesto Ágil, un documento que resume la filosofía “ágil”. (Canós, 2003).

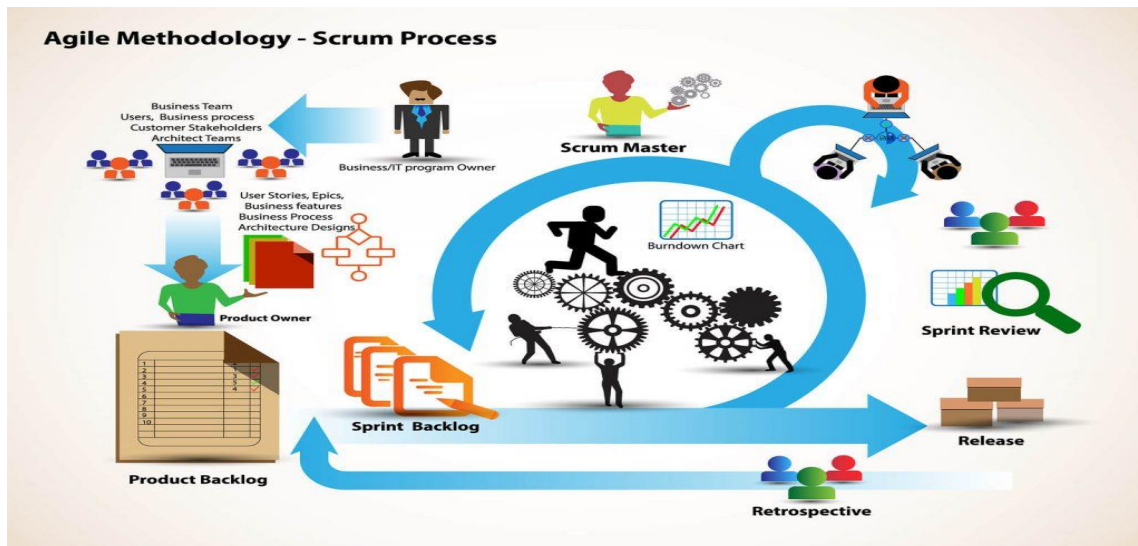


Figura 7 . Metodología SCRUM

Como afirma Duarte, A. O., & Rojas, M. (2008). “La metodología Scrum es un proceso para llevar a cabo un conjunto de tareas de forma regular con el objetivo principal de trabajar de manera colaborativa, es decir, para fomentar el trabajo en equipo”.

Lo que se pretende la Metodología SCRUM es alcanzar el mejor resultado de un proyecto determinado. Las prácticas que se aplican con esta metodología Scrum se retroalimentan unas con otras y la integración de las mismas tiene su origen en un estudio de cómo hay que coordinar a los equipos para ser potencialmente competitivos. (Duarte, 2008).

Uno de los motivos que considere para implementar esta metodología “Scrum” es que se van realizando entregas regulares y parciales del trabajo final, de manera prioritaria y en función del beneficio que aportan dichas entregas a los receptores del proyecto. Y por este motivo, es una metodología especialmente indicada para proyectos complejos, con requisitos cambiantes y en los que la innovación y la flexibilidad son protagonistas. (Fernandez Ardiles, 2022)

Además de que este método se lleva a cabo en proyectos donde la obtención de resultados a corto plazo es necesaria y en aquellos en los que existen situaciones de incertidumbre y tareas poco definidas. (Duarte, 2008)

SCRUM como modelo de trabajo será de gran ayuda con la implementación del proyecto, ya que se ha venido trabajando con las funcionalidades que aportan mayor valor a la Empresa.

El proyecto por cual estoy trabajando requiere de cooperación, de rapidez en los resultados, además que lleva a cabo demasiadas horas de capacitación y reuniones para verificar los avances del proyecto.

La Metodología **SCRUM** nos apoya a resolver aquellas partes del proyecto las cuales no se le está entregando en tiempo y forma a nuestros clientes y pues dichas tareas se dilatan demasiado en resolver o la calidad de nuestro proyecto es muy baja. (Duarte, 2008)

Además de que el método Scrum nos permite identificar y solucionar ineficiencias de forma sistemática o cuando un cliente demanda un proceso altamente especializado para el desarrollo de un producto determinado. (Duarte, 2008)

Como afirma Figueroa (2008) “La Metodología SCRUM lleva a cabo diferentes fases que hacen posible llevar a cabo nuestro proceso de Desarrollo de Software”.

1. Planificación: Product Backlog

- Análisis de los requisitos
- Diseño del sistema
- Implementación
- Verificación
- Mantenimiento

La Empresa Abarrotes Cordobeses es una empresa de giro comercial que se dedica a la venta de productos de canasta básica.

Al ser una PYME requiere de un Sistema que pueda administrar las facturas que le emiten sus proveedores.

Por lo que este proyecto surge de la necesidad del departamento de conciliaciones para agilizar sus procesos de revisión, el objetivo es implementar un sistema que pueda proporcionarles un reporte con la información que se carga en el repositorio de facturas digitales con la intención de que los usuarios la puedan tener obtener de forma rápida y en tiempo y forma.

Además, de que al no contar en tiempo y forma con esta información se ven afectados los procesos administrativos y de revisión que deben de cumplir los usuarios día a día en su operación diaria.

Anteriormente, este proceso de información solo lo podía proporcionar el departamento de Sistemas de la empresa, quienes son quienes cuentan con acceso a la base de datos donde se almacena esta información.

Sin embargo, con la implementación de este Software se podrá tener acceso a toda la información que se tenga en el repositorio de facturas recibidas, además que los usuarios podrán acceder con la misma cuenta que la empresa le fue asignada para sus operaciones administrativas.

2. Ejecución: Sprint

Cada día se realiza una reunión de sincronización (15min). Cada miembro del equipo responde a tres preguntas:

1. ¿Qué he hecho desde la última reunión de sincronización?
2. ¿Qué voy a hacer a partir de este momento?
3. ¿Qué impedimentos tengo o voy a tener?

Existe un Facilitador (Scrum Master) el cual se encarga de eliminar los obstáculos que el equipo no puede resolver por sí mismo y protege al equipo de interrupciones externas.

Durante la iteración, el cliente y el equipo refinan la lista de requisitos y, si es necesario, cambian o re-planifican los objetivos del proyecto.

3. Control: Burn Down

En este día se realiza una reunión de revisión en la que se realiza lo siguiente:

1. Demostración (4hrs): el equipo presenta al cliente los requisitos completados (cubiertos) y el cliente realiza las observaciones y adaptaciones necesarias.

2. Retrospectiva (4hrs): el equipo analiza cómo ha sido su manera de trabajar hasta el momento e identifica cuáles son los problemas que podrían impedirle progresar. El Facilitador elimina los obstáculos identificados.

Además, en la etapa de retrospectiva, se contesta a las siguientes preguntas:

- a. ¿Qué se hizo bien?
- b. ¿Qué procesos serían mejorables?
- c. ¿Cómo perfeccionarlos?

Roles y responsabilidades de las personas involucradas en la metodología

1. Cliente (Product Owner) Las responsabilidades del Cliente son:

- Ser el representante de todas las personas interesadas en los resultados del proyecto (internas o externas a la organización, promotores del proyecto y usuarios o consumidores finales del

producto) y actuar como interlocutor único ante el equipo, con autoridad para tomar decisiones.

- Definir los objetivos del producto o proyecto.
 - Dirigir los resultados del proyecto y maximizar su ROI.
 - Es el propietario de la planificación del proyecto. Crea y mantiene la lista priorizada con los requisitos necesarios para cubrir los objetivos del producto o proyecto, conoce el valor que aportará cada requisito y calcula el ROI a partir del coste de cada requisito que le proporciona el equipo.
 - Reparte los objetivos/requisitos en iteraciones y establece un calendario de entregas.
 - Antes de iniciar cada iteración replanifica el proyecto en función de los requisitos que aportan más valor en ese momento, de los requisitos completados en la iteración anterior y del contexto del proyecto en ese momento
 - Colaborar con el equipo para planificar, revisar y dar detalle a los objetivos de cada iteración. Metodologías de Desarrollo de Software
- Tesis Final – Cátedra: Seminario de Sistemas Carrera:
- Participar en la reunión de planificación de iteración, proponiendo los requisitos más prioritarios a desarrollar, respondiendo a las dudas del equipo y detallando los requisitos que el equipo se comprometer a hacer.
 - Estar disponible durante el curso de la iteración para responder a las preguntas o dudas que puedan surgir.
 - No cambiar los requisitos que se están desarrollando en una iteración, una vez ésta esta iniciada.

2. Facilitador (Scrum Master) Se encarga de liderar al equipo llevando a cabo las siguientes responsabilidades:

- Controlar que todos los participantes del proyecto sigan los valores y principios ágiles, las reglas y proceso de Scrum y guiar la colaboración dentro del equipo y con el cliente de manera que las sinergias sean máximas.
- Hay que asegurar que exista una lista de requisitos priorizada y que esté preparada antes de la siguiente iteración.
- Facilitar las reuniones de Scrum (planificación de la iteración, reuniones diarias de sincronización del equipo, demostración, retrospectiva), de manera que sean productivas y consigan sus objetivos.
- Enseñar al equipo a auto gestionarse. No da respuestas, si no que guía al equipo con preguntas para que descubra por sí mismo una solución.
- Quitar los impedimentos que el equipo tiene en su camino para conseguir el objetivo de cada iteración, proporcionar un resultado útil al cliente de la manera más efectiva y así poder finalizar el proyecto con éxito. Estos obstáculos se identifican de manera sistemática en las reuniones diarias de sincronización del equipo y en las reuniones de retrospectiva.
- Proteger y aislar al equipo de interrupciones externas durante la ejecución de la iteración (introducción de nuevos requisitos, desvinculación no prevista de un miembro del equipo, etc.). De esta manera, el equipo puede mantener su productividad y el compromiso que adquirió sobre los requisitos que completaría en la iteración.

3. Equipo (Team)

Es el grupo de personas que de manera conjunta desarrollan el producto del proyecto. Tienen un objetivo común y comparten la responsabilidad del trabajo que realizan.

El tamaño del equipo está entre 5 y 9 personas. Por debajo de 5 personas cualquier imprevisto o interrupción sobre un miembro del equipo compromete seriamente el compromiso que han adquirido y, por tanto, el

resultado que se va a entregar al cliente al finalizar la iteración. Por encima de 9 personas, la comunicación y colaboración entre todos los miembros se hace más difícil y se forman subgrupos.

Este equipo está constituido por los siguientes perfiles:

- Desarrollador
- Analista funcional
- Analista Técnico
- Tester
- Arquitecto
- Project Manager
- Team Leader

Es un equipo autoorganizado, que comparte información y cuyos miembros confían entre ellos.

Realiza de manera conjunta las siguientes actividades:

- Seleccionar los requisitos que se compromete a completar en una iteración, de forma que estén preparados para ser entregados al cliente.
- Estimar la complejidad de cada requisito en la lista de requisitos priorizada del producto o proyecto.
- En la reunión de planificación de la iteración decide cómo va a realizar su trabajo:
 - ✓ Seleccionar los requisitos que pueden completar en cada iteración, realizando al cliente las preguntas necesarias.
 - ✓ Identificar todas las tareas necesarias para completar cada requisito.
 - ✓ Estimar el esfuerzo necesario para realizar cada tarea.
 - ✓ Cada miembro del equipo se auto asigna a las tareas.

- Durante la iteración, trabajar de manera conjunta para conseguir los objetivos de la iteración. Cada especialista lidera el trabajo en su área y el resto colaboran si es necesario para poder completar un requisito.
- Al finalizar la iteración: o Demostrar al cliente los requisitos completados en cada iteración. o Hacer una retrospectiva la final de cada iteración para mejorar de forma continua su manera de trabajar. (Figueroa, 2008)

Programación Extrema

Como dice Letelier, P. (2006) La programación extrema (XP) puede que marque un antes y un después en la ingeniería del software.

Ésta nueva forma de trabajo fue creada por Kent Beck, Ward Cunningham y Ron Jeffries a finales de los noventa. La programación extrema ha pasado de ser una simple idea para un único proyecto a inundar todas las factorías de software.

Algunos la definen como un movimiento social de los analistas del software hacia los hombres y mujeres de negocios, de lo que debería ser el desarrollo de soluciones en contraposición a los legalismos de los contratos de desarrollo.

Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

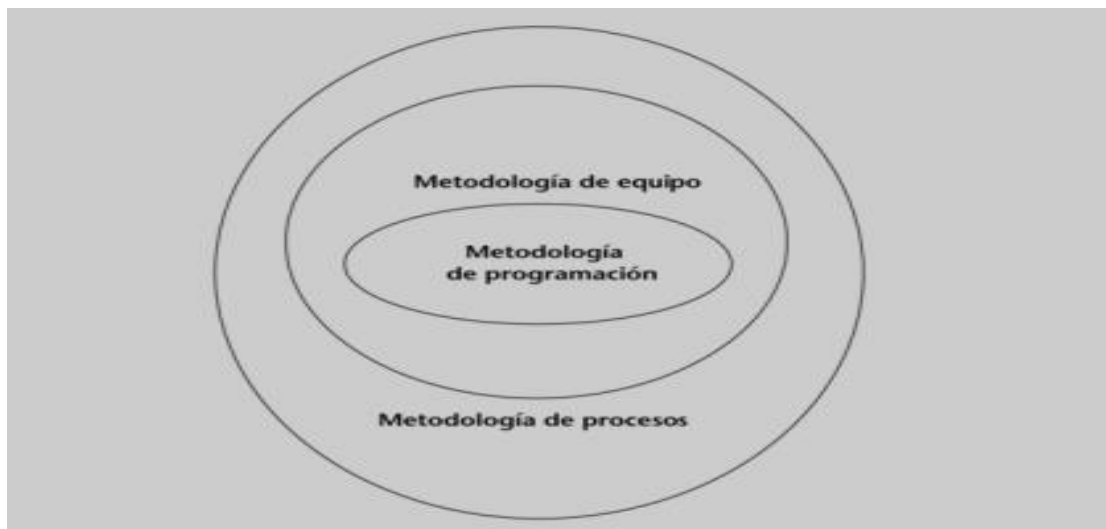


Figura 8. Capas de la programación extrema.

1. Metodología de programación: diseño sencillo, testing, refactorización y codificación con estándares.
2. Metodología de equipo: propiedad colectiva del código, programación en parejas, integración continua, entregas semanales e integridad con el cliente.
3. Metodología de procesos: cliente in situ, entregas frecuentes y planificación.

Lo que XP mantiene es que esta curva ha perdido vigencia y que con una combinación de buenas prácticas de programación y tecnología es posible lograr que la curva no crezca siempre de forma exponencial.

XP pretende conseguir una curva de coste del cambio con crecimiento leve, que en un inicio es más costosa, pero que a lo largo del proyecto permite tomar decisiones de desarrollo lo más tarde posible sin que esa nueva decisión de cambio implique un alto coste en el proyecto.

La programación extrema, propone una ecuación de equilibrio entre el coste, el tiempo de desarrollo, la calidad del software y el alcance de funcionalidades del mismo.

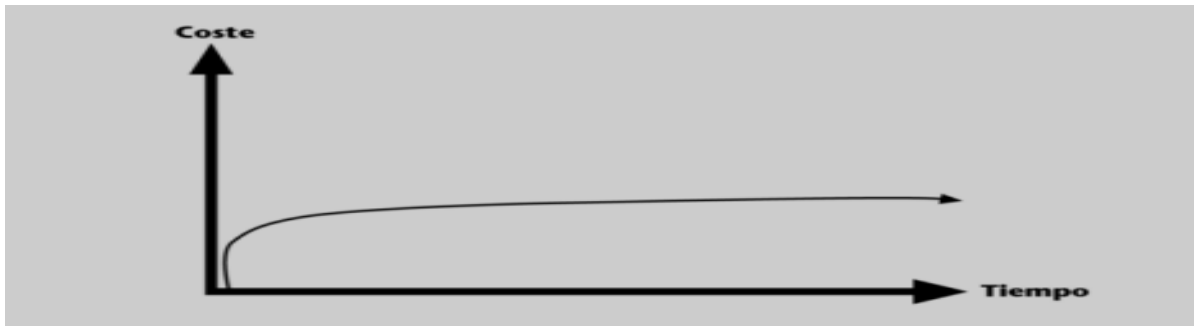


Figura 9. Variables de la Programación Extrema.

Las variables: coste, tiempo, calidad y alcance.

El punto de partida de la metodología XP son las variables que utiliza para cada proyecto:

- Coste (la inversión económica y en recursos).
- Tiempo (el tiempo empleado, determinado por la fecha de entrega final).
- Calidad (del código y del aplicativo desarrollado).
- Alcance (Conjunto de funcionalidades).

De estas cuatro variables, tres podrán ser fijadas por el cliente y/o por el jefe de proyectos, la cuarta es responsabilidad del equipo de desarrollo y se establecerá en función de las otras tres.

Los diferentes roles dentro de la Programación Extrema Cada rol tiene unas funciones claras dentro de la metodología.

Cada persona del equipo puede ejecutar uno o varios roles, o incluso cambiar de rol durante las diferentes fases del proyecto.

1. Programador:

- Escribe las pruebas unitarias.
- Produce el código del programa.

2. Cliente:

- Escribe las historias de usuario.
- Diseña las pruebas de aceptación.
- Prioriza las historias de usuario.
- Aporta la dimensión de negocio al equipo de desarrollo.
- Representa al colectivo de usuarios finales.
- Está siempre disponible para consultas.

3. Encargado de pruebas (tester):

- Ejecuta las pruebas de aceptación.
- Ayuda al cliente a diseñar pruebas de aceptación.
- Ejecuta las pruebas de integración.
- Difunde los resultados entre el equipo de desarrollo y el cliente.
- Es el responsable automatizar los casos de prueba.
- Encargado de seguimiento de los errores y sus diferentes estados.

- Se encarga de realimentar todo el proceso de XP, midiendo las desviaciones con respecto a las estimaciones y comunicando los resultados para mejorar las siguientes estimaciones.
- Realiza el seguimiento de cada iteración del proceso de XP tanto en la etapa de iteraciones como en la de producción.
- Revalúa la posibilidad de incorporar o eliminar historias de usuario.

4. Líder Técnico

- Se encarga del proceso global.
- Garantiza que se sigue la filosofía de XP.
- Conoce a fondo la metodología.
- Provee guías y ayudas a los miembros del equipo a la hora de aplicar las prácticas básicas de XP. a

5. Consultor:

- No forma parte del equipo.
- Tiene un conocimiento específico de un área en concreto.
- Ayuda a resolver un problema puntual, ya sea de spike tecnológico o de valor de negocio.
- PM (Project Manager):
- Es el máximo responsable del proyecto.
- Hace de enlace con los clientes.
- Se encarga de coordinar y de garantizar las condiciones necesarias para el desarrollo del trabajo.

CRYSTAL

Crystal no es solo una metodología de desarrollo de software ágil, ya que se la considera una familia de metodologías, debido a que se subdivide en varios tipos en función a la cantidad de persona involucradas en el proyecto. Esta nueva serie de metodologías fue creada por el antropólogo Alistair

Cockburn, el cuál tomo como base el análisis de distintos proyectos de desarrollo de software y su propia experiencia, lo cual fusionando ambos aspectos dio lugar este nuevo método de trabajo. Estas metodologías presentan un enfoque ágil, con gran énfasis en la comunicación y con cierta tolerancia que la hace ideal en los casos en que sea inaplicable la disciplina requerida por Extreme Programming.

Crystal Clear es la encarnación más ágil de la serie y de la que más documentación se dispone. La misma se define con mucho énfasis en la comunicación y de forma muy liviana en relación a los entregables. Crystal maneja iteraciones cortas con feedback frecuente por parte de los usuarios/clientes, minimizando de esta forma la necesidad de productos intermedios. Otra de las cuestiones planteadas es la necesidad de disponer de un usuario real, aunque sea de forma part time para realizar validaciones sobre la UI (Interface de Usuario) y para participar en la definición de los requerimientos funcionales y no funcionales del software (Figuerola, 2008).

La familia Crystal dispone de un código de color para marcar la complejidad de una metodología, ya que cuanto más oscuro es el color, más pesado es el método y cuanto más crítico es un sistema, más rigor se requiere.

El código cromático se aplica a una forma tabular que se usa para situar el rango de complejidad al cual se aplica una metodología.

El nombre Crystal deriva de la caracterización de los proyectos según 2 dimensiones, tamaño y complejidad (Ortegón, 2005).

En la siguiente imagen se puede apreciar la escala de colores dependiendo de la complejidad del sistema:

Criticalidad del Sistema				
	L6	L20	L40	L80
	E6	E20	E40	E80
	D6	D20	D40	D80
	C6	C20	C40	C80
	Claro	Amarillo	Naranja	Rojo
	Tamaño del Proyecto			

Figura 10. Familia de Metodologías Crystal.

- Claro (Clear) es para equipos de hasta 8 personas o menos.
- Amarillo para equipos entre 10 a 20 personas.
- Naranja para equipos entre 20 a 50 persona.
- Roja para equipos de más de 50 personas.

Propiedades de Crystal Clear:

- Entrega frecuente: consiste en entregar software a los clientes con frecuencia, no solamente en compilar el código. La frecuencia dependerá del proyecto, pero puede ser diaria, semanal o mensual.
- Comunicación osmótica: todos juntos en el mismo cuarto. Una variante especial es disponer en la sala de un experto diseñador senior y discutir respecto del tema que se trate.
- Mejora reflexiva: tomarse un pequeño tiempo diariamente para pensar bien qué se está haciendo, cotejar notas, reflexionar, discutir.
- Seguridad personal: hablar con los compañeros cuando algo molesta dentro del grupo.

- Foco: saber lo que se está haciendo y tener la tranquilidad y el tiempo para hacerlo.
- Fácil acceso a usuarios expertos: tener alguna comunicación con expertos desarrolladores.

Roles

- Patrocinador: produce la Declaración de Misión con Prioridades de Compromiso (Tradeoff). Consigue los recursos y define la totalidad del proyecto.
- Usuario Experto: junto con el Experto en Negocios produce la Lista de Actores-Objetivos y el Archivo de Casos de Uso y Requerimientos. Debe familiarizarse con el uso del sistema, sugerir atajos de teclado, modos de operación, información a visualizar simultáneamente, navegación.
- Diseñador Principal: produce la Descripción Arquitectónica. Se supone que debe ser al menos un profesional de Nivel 3. En Metodologías Ágiles se definen tres niveles de experiencia:
 - Nivel 1 es capaz de seguir los procedimientos.
 - Nivel 2 es capaz de apartarse de los procedimientos específicos y encontrar otros distintos.
 - Nivel 3 es capaz de manejar con fluidez, mezclar e inventar procedimientos.

(Duarte A. O., 2008)

Según El Diseñador Principal Duarte A. O (2008) tiene roles de coordinador, arquitecto, mentor y programador más experto.

- Diseñador/Programador: produce, junto con el Diseñador Principal, los Borradores de Pantallas, el Modelo Común de Dominio, las Notas y Diagramas de Diseño, el Código Fuente, el Código de Migración, las Pruebas y el Sistema Empaquetado.

- Experto en Negocios: junto con el Usuario Experto produce la Lista de Actores/Objetivos y el Archivo de Casos de Uso y Requerimientos. Debe conocer las reglas y políticas del negocio.
- Coordinador: con la ayuda del equipo, produce el Mapa de Proyecto, el Plan de Entrega, el Estado del Proyecto, la Lista de Riesgos, el Plan y Estado de Iteración y la Agenda de Visualización.
- Verificador: produce el Reporte de errores. Puede ser un programador en tiempo parcial, o un equipo de varias personas.
- Escritor: produce el Manual de Usuario. El Equipo como Grupo es responsable de producir la Estructura y Convenciones del Equipo y los Resultados del Taller de Reflexión.

KANBAN

Esta metodología tiene como base de su origen la aplicación de los procesos de producción JIT (Just inTime) ideados por la empresa automotriz Toyota, en la cual utilizaban tarjetas visuales para identificar necesidades de material en la cadena de producción. Kanban se basa en la idea de que el trabajo en curso debería limitarse, y sólo deberíamos empezar con algo nuevo cuando un bloque de trabajo anterior haya sido entregado o ha pasado a otra función posterior de la cadena. (Maida, 2015)

La metodología Kanban utiliza un mecanismo de control visual para hacer seguimiento del trabajo conforme este viaja a través del flujo de valor. Normalmente, se emplea un panel o pizarra con notas adhesivas o un panel electrónico de tarjetas para gestionar el flujo de trabajo y las asignaciones. Las mejores prácticas apuntan al uso de ambos métodos. (Maida, 2015).

“El Kanban (o tarjeta visual) implica que se genera una señal visual para indicar que hay nuevos bloques de trabajo que pueden ser comenzados porque el trabajo en curso actual no alcanza el máximo acordado”. (Maida, 2015)

El aporte principal de Kanban a las metodologías ágiles es que, a través de la implementación de tarjetas visuales, proporciona transparencia al proceso, ya que su flujo de trabajo expone los cuellos de botella, colas, variabilidad y desperdicios a lo largo del tiempo y todas las cosas que impactan al rendimiento de la organización en términos de la cantidad de trabajo entregado y el ciclo de tiempo requerido para entregarlo. (Maida, 2015)

Proporciona a los miembros del equipo y a las partes interesadas visibilidad sobre los efectos de sus acciones o falta de acción. De esta forma, cambia el comportamiento y motiva a una mayor colaboración en el trabajo.

La visibilidad de los cuellos de botella, desperdicios y variabilidades y su impacto también promueve la discusión sobre las posibles mejoras, y hace que los equipos comiencen rápidamente a implementar mejoras en su proceso. Como resultado, Kanban propicia la evolución incremental de los procesos existentes, una evolución que generalmente está alineada con los valores de las metodologías ágiles.

Kanban no genera una revolución radical de la forma en la que las personas trabajan, sino que sugiere un cambio gradual. Es un cambio que surge del entendimiento y del consenso de entre todos los participantes del proyecto. (Maida, 2015)

Las principales ventajas de esta metodología es que es muy fácil de aplicar, actualizar y asumir por parte del equipo. Además, se destaca por utilizar una técnica de gestión de las tareas muy visual y práctica, lo que permite ver a simple vista el estado de los proyectos, así como también pautar el desarrollo del trabajo de manera efectiva. (Úbeda González, 2009).

De acuerdo con las Heras del Dedo (2018) los principales Principios del método **Kanban** son:

- Calidad garantizada: todo lo que se hace debe salir bien en la primera instancia, no hay margen de error. Para lograr esto no se premia la rapidez, sino la calidad final de las tareas realizadas. Esto se basa en el hecho que muchas veces cuesta más arreglarlo después que hacerlo bien de entrada. (las Heras del Dedo, 2018)
- Reducción del desperdicio: se basa en hacer solamente lo justo y necesario, para garantizar que se haga bien. Esto supone la reducción de todo aquello que es superficial o secundario.
- Mejora continua: Se acuerda perseguir el cambio incremental y evolutivo. No es simplemente un método de gestión, sino también un sistema de mejora en el desarrollo de proyectos, según los objetivos a alcanzar. El equipo debe estar de acuerdo que el cambio continuo, gradual y evolutivo es la manera de hacer mejoras en el sistema y debe apegarse a ello. Los cambios radicales pueden parecer más eficaces, pero tienen una mayor tasa de fracaso debido a la resistencia y el miedo en la organización.
- Flexibilidad: es necesario poder priorizar aquellas tareas entrantes según las necesidades del momento y tener la capacidad de dar respuesta a estas tareas imprevistas.

Las tres reglas de Kanban

1. Mostrar el proceso
2. Limitar el trabajo en curso
3. Optimizar el flujo de trabajo

Un tablero Kanban, se divide en columnas las cuales representan un proceso de trabajo. Un ejemplo clásico de columnas para dividir un tablero Kanban, sería el siguiente:

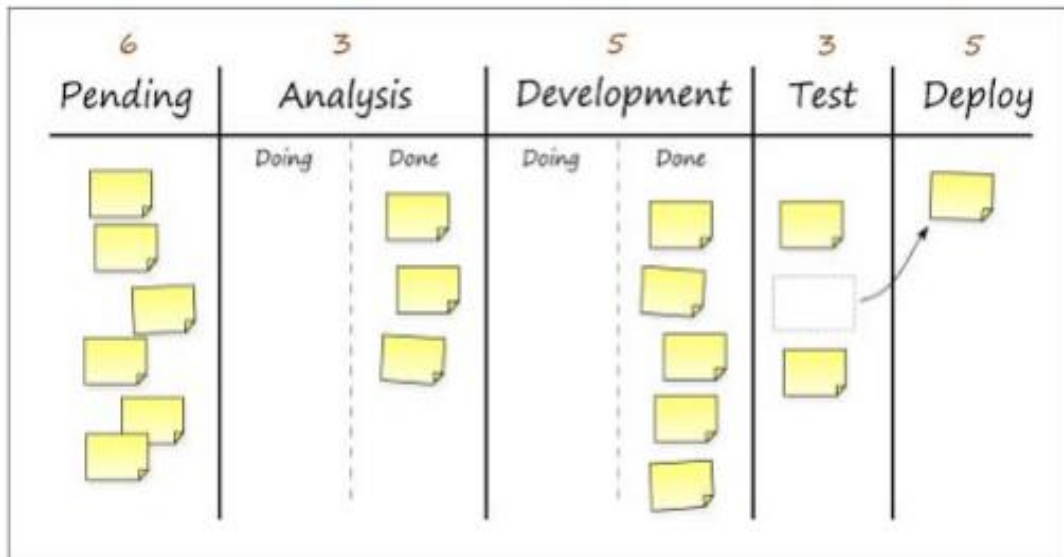


Figura 11. Tablero Kanban.

“La cantidad y nombre de las columnas, varía de acuerdo con las necesidades de cada equipo y en la mayoría de los casos, éstas, son subdivididas en dos columnas: cola de espera y en curso”. (Letelier Torres, JULY 2015)

Metodologías Clásicas

Desarrollar un buen software depende de un gran número de actividades y etapas, donde el impacto de elegir la metodología para un equipo en un determinado proyecto es trascendental para el éxito del producto. Las metodologías tradicionales son denominadas, a veces, de forma despectiva, como metodologías pesadas.

Centran su atención en llevar una documentación exhaustiva de todo el proyecto, la planificación y control de este, en especificaciones precisas de requisitos y modelado y en cumplir con un plan de trabajo, definido todo esto, en la fase inicial del desarrollo del proyecto.

Estas metodologías tradicionales imponen una disciplina rigurosa de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente.

Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software.

Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar (Pacienza, 2015).

WATERFALL (CASCADA)

En Ingeniería de software el desarrollo en cascada es denominado así por la posición de las fases en el desarrollo de esta, que parecen caer en cascada “por gravedad” hacia las siguientes fases. Es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior.

Al final de cada etapa, el modelo está diseñado para llevar a cabo una revisión final, que se encarga de determinar si el proyecto está listo para avanzar a la siguiente fase.

Este modelo fue el primero en originarse y es la base de todos los demás modelos de ciclo de vida. (Pacienza, 2015).

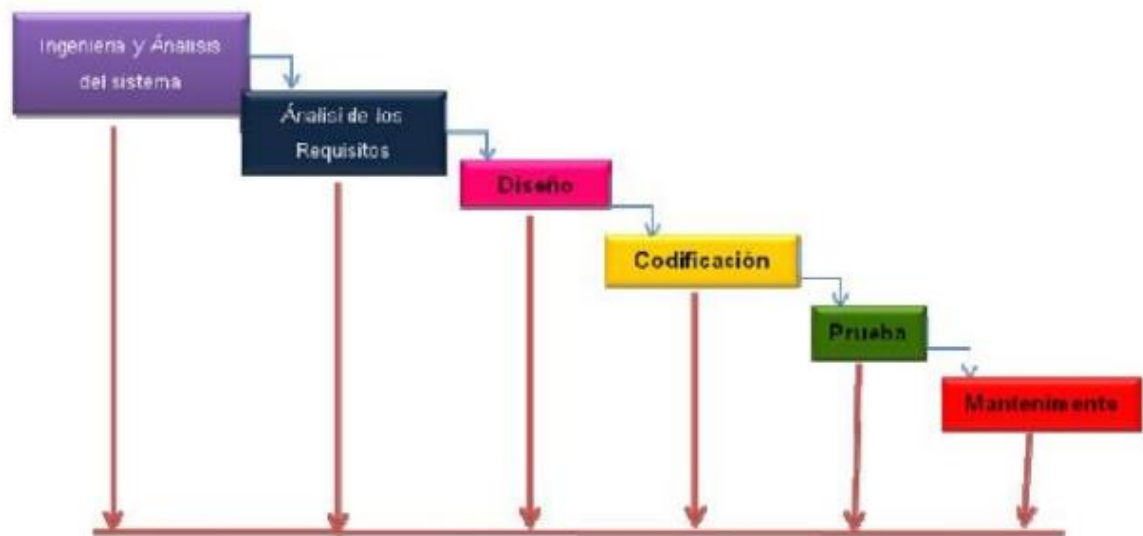


Figura 12. Modelo del ciclo de vida en Cascada (Waterfall).

Ingeniería y Análisis del Sistema: debido que el software es siempre parte de un sistema mayor, el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.

Análisis de los requisitos del software: el proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El ingeniero de software (Analistas) debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.

Diseño: el diseño del software se enfoca en cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.

Codificación: el diseño debe traducirse en una forma legible para la máquina. El paso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada la codificación puede realizarse mecánicamente.

Prueba: una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.

Mantenimiento: el software sufrirá cambios después de que se entrega al cliente. Los cambios ocurrirán debido a se han encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos), o debido a que el cliente requiera ampliaciones funcionales o del rendimiento.

Desventajas:

- Los proyectos reales raramente siguen el flujo secuencial que propone el modelo, siempre hay iteraciones y se crean problemas en la aplicación del paradigma.
- Normalmente, es difícil para el cliente establecer explícitamente al principio todos los requisitos. El ciclo de vida clásico lo requiere y tiene dificultades en acomodar posibles incertidumbres que pueden existir al comienzo de muchos productos.
- El cliente debe tener paciencia. Hasta llegar a las etapas finales del proyecto, no estará disponible una versión operativa del programa. Un error importante no detectado hasta que el programa esté funcionando puede ser desastroso. (Pacienza, 2015)

PROTOTYPING

Un prototipo es una versión preliminar de un sistema de información con fines de demostración o evaluación. El prototipo de requerimientos es la creación de una implementación parcial de un sistema, para el propósito explícito de aprender sobre los requerimientos del sistema. Un prototipo es construido de una manera rápida tal como sea posible.

Esto es dado a los usuarios, clientes o representantes de ellos, posibilitando que ellos experimenten con el prototipo. Estos individuos luego proveen la retroalimentación sobre lo que a ellos les gustó y no les gustó acerca del prototipo proporcionado, quienes capturan en la documentación actual de la especificación de requerimientos la información entregada por los usuarios para el desarrollo del sistema real.

El prototipo puede ser usado como parte de la fase de requerimientos (determinar requerimientos) o justo antes de la fase de requerimientos (como predecesor de requerimientos). En otro caso, el prototipo puede servir su papel inmediatamente antes de algún o todo el desarrollo incremental en modelos incremental o evolutivo. El prototipo ha sido usado frecuentemente en los 90, porque la especificación de requerimientos para sistemas complejos tiende a ser relativamente dificultoso de cursar. Muchos usuarios y clientes encuentran que es mucho más fácil proveer retroalimentación convenientemente basada en la manipulación, leer una especificación de requerimientos potencialmente ambigua y extensa. (Pacienza, 2015).

Ventajas:

- Útiles cuando los requerimientos son cambiantes.
- Cuando no se conoce bien la aplicación.
- Cuando el usuario no se quiere comprometer con los requerimientos.
- Cuando se quiere probar una arquitectura o tecnología.
- Cuando se requiere rapidez en el desarrollo.

Desventajas:

- No se conoce cuando se tendrá un producto aceptable.
- No se sabe cuántas iteraciones serán necesarias.
- Da una falsa ilusión al usuario sobre la velocidad del desarrollo.
- Se puede volver el producto aún y cuando no esté con los estándares.

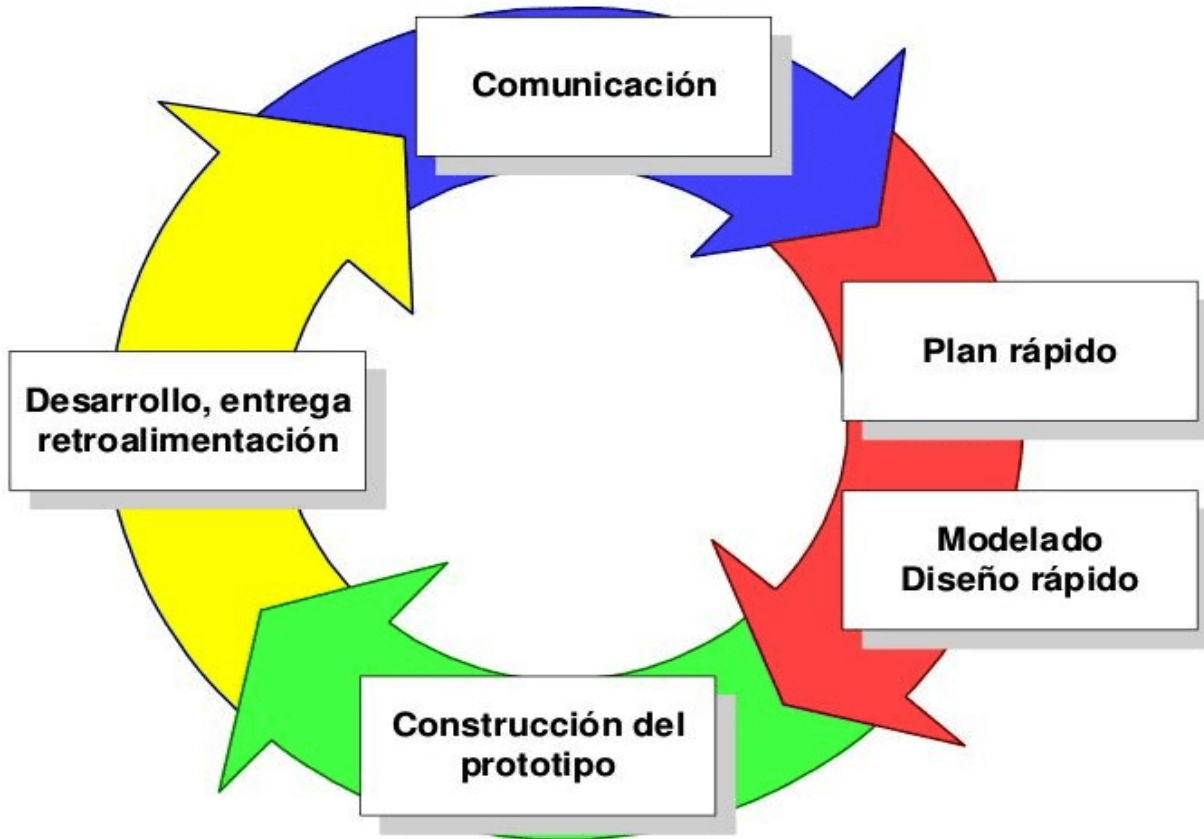


Figura 13. Modelo del ciclo de vida Prototipo

SPIRAL

Toma las ventajas del modelo de desarrollo en cascada y el de prototipos añadiéndole el concepto de análisis de riesgo.

Se definen cuatro actividades:

Planificación: en la que se recolectan los requisitos iniciales o nuevos requisitos a añadir en esta iteración.

Análisis de riesgo: basándonos en los requisitos decidimos si somos capaces o no de desarrollar el software y se toma la decisión de continuar o no continuar.

Ingeniería: en el que se desarrolla un prototipo basado en los requisitos obtenidos en la fase de planificación.

Evaluación del cliente: el cliente comenta el prototipo. Si está conforme con él se acaba el proceso, si no se añaden los nuevos requisitos en la siguiente iteración.

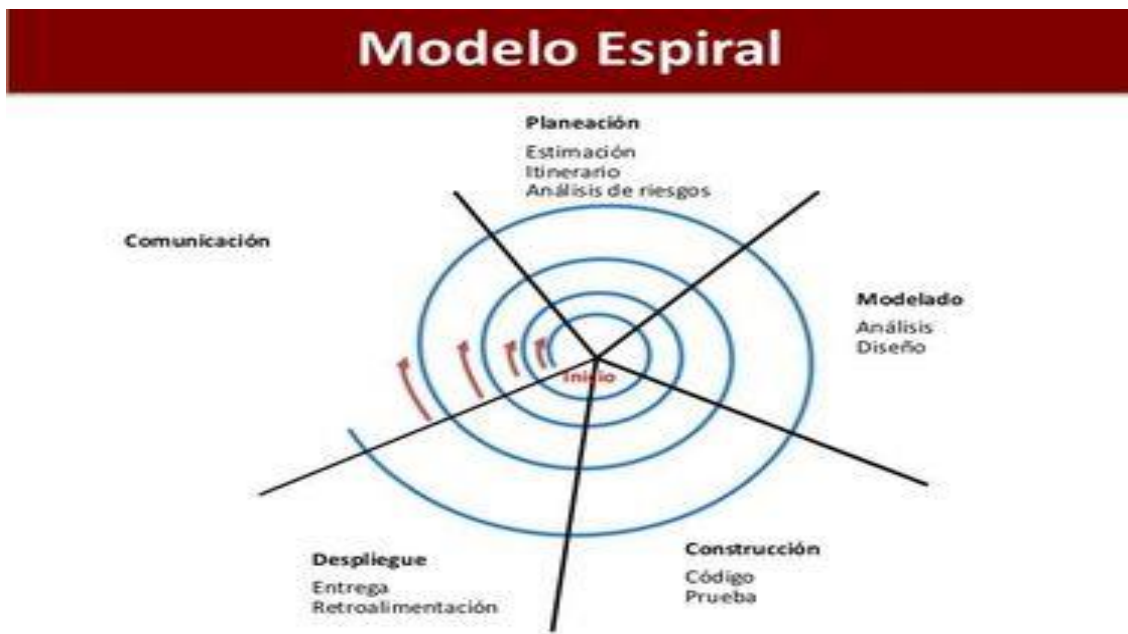


Figura 14. Modelo del Ciclo de vida en Espiral.

El esquema del ciclo de vida para estos casos puede representarse por un bucle en espiral, donde los cuadrantes son, habitualmente, fases de

especificación, diseño, realización y evaluación (o conceptos y términos análogos).

En cada fase el producto gana “madurez” (aproximación al final deseado) hasta que en una iteración se logre el objetivo deseado y este sea aprobado se termina las iteraciones. (Meaurio, 2013).

INCREMENTAL

Como afirma González (2007) La Metodología Incremental Permite construir el proyecto en etapas incrementales en donde cada etapa agrega funcionalidad. Estas etapas, consisten en requerimientos, diseño, codificación, pruebas y entrega. Permite entregar al cliente un producto más rápido en comparación del modelo en cascada.

- Reduce los riesgos ya que provee visibilidad sobre el progreso de las nuevas versiones.
- Provee retroalimentación a través de la funcionalidad mostrada.
- Permite atacar los mayores riesgos desde el inicio.
- Se pueden hacer implementaciones parciales si se cuenta con la suficiente funcionalidad.
- Las pruebas y la integración son constantes.
- El progreso se puede medir en periodos cortos de tiempo.
- Resulta más sencillo acomodar cambios al acortar el tamaño de los incrementos.
- Se puede planear en base a la funcionalidad que se quiere entregar primero.
- Por su versatilidad requiere de una planeación cuidadosa tanto a nivel administrativo como técnico.

Ventajas:

- La solución se va mejorando en forma progresiva a través de las múltiples iteraciones, incrementa el entendimiento del problema y de la solución por medio de los refinamientos sucesivos.
- Los clientes no esperan hasta el fin del desarrollo para utilizar el sistema. Pueden empezar a usarlo desde el primer incremento.
- Los clientes pueden aclarar los requisitos que no tengan claros, conforme ven las entregas del sistema.
- Se disminuye el riesgo de fracaso de todo el proyecto, ya que se puede distribuir en cada incremento.
- Las partes más importantes del sistema son entregadas primero, por lo cual se realizan más pruebas en estos módulos y se disminuye el riesgo de fallos.

Desventaja:

- Requiere de mucha planeación, tanto administrativa como técnica
- Requiere de metas claras para conocer el estado del proyecto.
- Es un proceso de desarrollo de software, creado en respuesta a las debilidades del modelo tradicional de cascada.

Para apoyar el desarrollo de proyectos por medio de este modelo se han creado Framework (entornos de trabajo), de los cuales los dos más famosos son el Rational Unified Process (RUP) y el Dynamic Systems Development Method.

El desarrollo incremental e iterativo es también una parte esencial de un tipo de programación conocido como Extreme Programming y los demás frameworks de desarrollo rápido de software. (Úbeda González, 2009).

Modelo Incremental

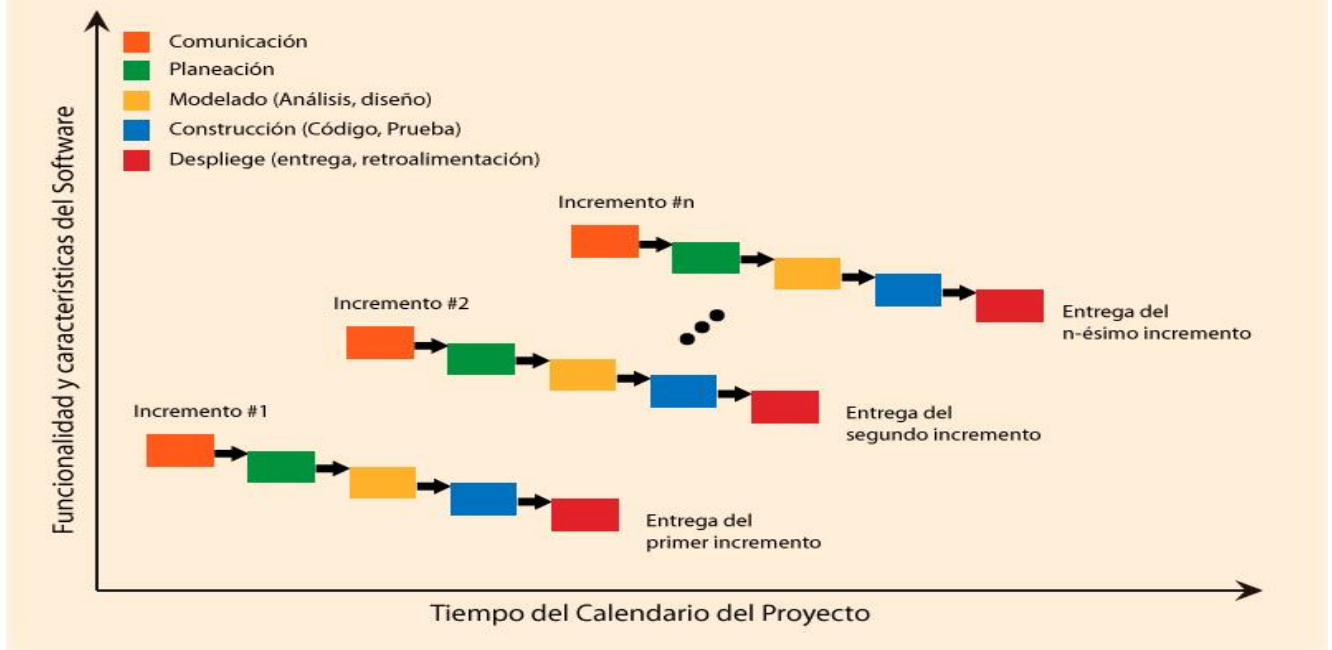


Figura 15. Modelo del Ciclo de vida Incremental.

2.1 Selección de la Metodología

La metodología elegida para trabajar fue Scrum, ya que nos permite que se vaya desarrollando de manera ordenada y secuencial cada una de las funcionalidades del sistema, ya que día a día se va haciendo una pequeña programación de las actividades que se van a realizar y al final se analiza lo que se ha desarrollado. (Ramírez, 2019)

En general, Scrum es un marco de trabajo para el desarrollo de productos software complejos en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente. Es ligero, fácil de entender, pero muy difícil de dominar. (Rodríguez, 2015)

Dentro de Scrum se pueden emplear varias técnicas y procesos que faciliten el desarrollo del software. En él se realizan entregas parciales y

regulares del producto final. Un proyecto se ejecuta en bloques temporales cortos y fijos (iteraciones de un mes y hasta de dos semanas) y cada iteración tiene que proporcionar un resultado completo. (Rodríguez, 2015)

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos. (Rodríguez, 2015) (Mariño, 2014)

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales. (Mariño, 2014)

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto. (Rodríguez, 2015)

La metodología Scrum es un proceso para llevar a cabo un conjunto de tareas de forma regular con el objetivo principal de trabajar de manera colaborativa, es decir, para fomentar el trabajo en equipo. (Godoy, 2015)

Con este método de trabajo lo que se pretende es alcanzar el mejor resultado de un proyecto determinado. Las prácticas que se aplican con la metodología Scrum se retroalimentan unas con otras y la integración de las mismas tiene su origen en un estudio de cómo hay que coordinar a los equipos para ser potencialmente competitivos. (Godoy, 2015).

El Product Backlog es la fase en la que se establecen las tareas prioritarias y donde se obtiene información breve y detallada sobre el proyecto que se va a desarrollar. (Cadavid, 2013)

Con el método Scrum no es necesario definir todos los objetivos al comienzo del proyecto. El Product Owner, de forma conjunta con el equipo de trabajo comienzan a listar lo más importante para el Product Backlog. (Godoy, 2015)

El Product Backlog es necesario para poder arrancar con el primer sprint, tiene permitido cambiar y crecer tantas veces como sea necesario en función del aprendizaje adquirido en el desarrollo del producto. (Estayno, 2014)

Ejecución: Sprint

Dentro del método Scrum, el Sprint es el corazón, un intervalo de tiempo que como máximo tiene una duración de un mes y en donde se produce el desarrollo de un producto que es entregable potencialmente.

Para entenderlo mejor, si el Product Owner solicita el producto se requiere un mínimo esfuerzo para su entrega al cliente.

También se puede definir el Sprint como un mini proyecto en donde el equipo de trabajo se focaliza en el desarrollo de tareas para alcanzar el objetivo que se ha definido previamente en el Sprint planning

Control: Burn Down

El Burn Down es la fase en la que se mide el progreso de un determinado proyecto Scrum. En ella, el Scrum Master será el encargado de actualizar los gráficos cuando se finalice cada uno de los Sprint. (Godoy, 2015)

Roles

En Scrum, el equipo se focaliza en construir software de calidad. La gestión de un proyecto Scrum se centra en definir cuáles son las características que debe tener el producto a construir (qué construir, qué no y en qué orden) y en vencer

cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo. (Estayno, 2014).

Según Estayno (2014) El equipo Scrum está formado por los siguientes roles:

1. Cliente (Product Owner)

Las responsabilidades del Cliente son:

- Ser el representante de todas las personas interesadas en los resultados del proyecto (internas o externas a la organización, promotores del proyecto y usuarios o consumidores finales del producto) y actuar como interlocutor único ante el equipo, con autoridad para tomar decisiones.
- Definir los objetivos del producto o proyecto.
- Dirigir los resultados del proyecto y maximizar su ROI.
- Es el propietario de la planificación del proyecto. Crea y mantiene la lista priorizada con los requisitos necesarios para cubrir los objetivos del producto o proyecto, conoce el valor que aportará cada requisito y calcula el ROI a partir del coste de cada requisito que le proporciona el equipo.
- Reparte los objetivos/requisitos en iteraciones y establece un calendario de entregas.
- Antes de iniciar cada iteración replanifica el proyecto en función de los requisitos que aportan más valor en ese momento, de los requisitos completados en la iteración anterior y del contexto del proyecto en ese momento
- Colaborar con el equipo para planificar, revisar y dar detalle a los objetivos de cada iteración.
- Participar en la reunión de planificación de iteración, proponiendo los requisitos más prioritarios a desarrollar, respondiendo a las dudas del equipo y detallando los requisitos que el equipo se comprometer a hacer.

- Estar disponible durante el curso de la iteración para responder a las preguntas o dudas que puedan surgir.
- No cambiar los requisitos que se están desarrollando en una iteración, una vez ésta esta iniciada.

2. Facilitador (Scrum Master)

Se encarga de liderar al equipo llevando a cabo las siguientes responsabilidades:

- Controlar que todos los participantes del proyecto sigan los valores y principios ágiles, las reglas y proceso de Scrum y guiar la colaboración dentro del equipo y con el cliente de manera que las sinergias sean máximas.
- Asegurar que exista una lista de requisitos priorizada y que esté preparada antes de la siguiente iteración.
- Facilitar las reuniones de Scrum (planificación de la iteración, reuniones diarias de sincronización del equipo, demostración, retrospectiva), de manera que sean productivas y consigan sus objetivos.
- Enseñar al equipo a auto gestionarse. No da respuestas, si no que guía al equipo con preguntas para que descubra por sí mismo una solución.
- Quitar los impedimentos que el equipo tiene en su camino para conseguir el objetivo de cada iteración, proporcionar un resultado útil al cliente de la manera más efectiva y así poder finalizar el proyecto con éxito. Estos obstáculos se identifican de manera sistemática en las reuniones diarias de sincronización del equipo y en las reuniones de retrospectiva.
- Proteger y aislar al equipo de interrupciones externas durante la ejecución de la iteración (introducción de nuevos requisitos, desvinculación no prevista de un miembro del equipo, etc.). De esta manera, el equipo puede mantener su productividad y el compromiso que adquirió sobre los requisitos que completaría en la iteración.

3. Equipo (Team)

Es el grupo de personas que de manera conjunta desarrollan el producto del proyecto. Tienen un objetivo común y comparten la responsabilidad del trabajo que realizan. El tamaño del equipo está entre 5 y 9 personas. Por debajo de 5 personas cualquier imprevisto o interrupción sobre un miembro del equipo compromete seriamente el compromiso que han adquirido y, por tanto, el resultado que se va a entregar al cliente al finalizar la iteración. Por encima de 9 personas, la comunicación y colaboración entre todos los miembros se hace más difícil y se forman subgrupos.

Este equipo está constituido por los siguientes perfiles:

- Desarrollador
- Analista funcional
- Analista Técnico
- Tester
- Arquitecto
- Project Manager
- Team Leader

Herramientas que se utilizan en Scrum

- Lista de objetivos / requisitos priorizada (Product Backlog)

La lista de objetivos/requisitos priorizada representa la visión y expectativas del cliente respecto a los objetivos y entregas del producto o proyecto. El cliente es el responsable de crear y gestionar la lista con la ayuda del Facilitador y del equipo, quien proporciona el coste estimado de completar cada requisito.

- Lista de tareas de la iteración (Sprint Backlog)

Es una lista de tareas que el equipo elabora en la reunión de planificación de la iteración (Sprint planning) como plan para completar los objetivos/requisitos seleccionados para la iteración y que se compromete a demostrar al cliente al finalizar la iteración, en forma de incremento de producto preparado para ser entregado. A cada tarea se la suele llamar Historia (History)

- Gráficos de trabajo pendiente (Burndown charts)

Consiste en un gráfico que indica el trabajo pendiente a lo largo del tiempo y muestra la velocidad a la que se están completando los objetivos/requisitos. Permite predecir si el equipo podrá completar el trabajo en el tiempo estimado. (Estayno, 2014)

Ventajas de utilizar Scrum

- Entregas parciales a corto plazo de resultados
- Gestión regular de las expectativas del cliente y basada en resultados tangibles.
- Resultados anticipados.
- Flexibilidad y adaptación respecto a las necesidades del cliente, cambios en el mercado, etc.
- Gestión sistemática del Retorno de Inversión (ROI).
- Mitigación sistemática de los riesgos del proyecto.
- Productividad y calidad.
- Alineamiento entre el cliente y el equipo de desarrollo.
- Equipo motivado.

Scrum tiene muchas ventajas sobre otras metodologías de desarrollo ágil. Actualmente es el marco de referencia más utilizado y fiable en la industria del software.

Fácilmente escalable: Los procesos de Scrum son iterativos y se manejan dentro de períodos de trabajo específicos, lo que facilita que el equipo se concentre en funcionalidades definidas para cada período. Esto no sólo tiene el beneficio de lograr mejores entregables en línea con las necesidades del usuario, sino que también da la capacidad a los equipos de escalar los módulos en términos de funcionalidad, diseño, alcance y características de una manera ordenada, transparente y sencilla.

Cumplimiento de expectativas: El cliente establece sus expectativas indicando el valor que aporta cada requisito/historia del proyecto, el equipo las estima y con esta información el Propietario de Producto establece su prioridad. De forma periódica, en las demos de los sprints, el Propietario de Producto verifica que se han cumplido los requisitos y transmite el feedback al equipo.

Flexibilidad ante los cambios: Reacción rápida a los cambios en los requisitos generados por las necesidades del cliente o la evolución del mercado. La metodología está diseñada para adaptarse a los requisitos cambiantes que conllevan los proyectos complejos.

Reducción del tiempo de comercialización: El cliente puede empezar a utilizar las funcionalidades más importantes del proyecto antes de que el producto esté completamente listo.

Mayor calidad del software: El método de trabajo y la necesidad de obtener una versión funcional después de cada iteración, ayuda a obtener un software de mayor calidad.

Predicción oportuna: Con esta metodología se conoce la velocidad media del equipo por sprint (puntos de historia), con lo que, en consecuencia, se puede estimar cuándo estará disponible una determinada funcionalidad que aún está en el backlog.

Reducción de riesgos: El hecho de realizar las funcionalidades más valiosas en primer lugar y de conocer la velocidad con la que el equipo avanza en el proyecto, permite despejar los riesgos de forma eficaz y anticipada. (Duarte A. O., 2008).

2.2 Implementación de la metodología

Para este Proyecto de Software realizaré la implementación y verificación de información cualitativa, se llevará a cabo el desarrollo de un sistema de Reportes para la Empresa de ABARROTOS CORDOBESES SA DE CV, este sistema se utilizará para la consulta de información alojada en el repositorio de facturas validadas la cual se extraerá mediante un reporte, para la manipulación y operaciones del departamento de conciliaciones.

Como herramienta de recolección de información lleve a cabo una encuesta por medio de GOOGLE FORMS de una serie de preguntas.

La encuesta fue realizada al personal de SUPERIBERIA, específicamente a los usuarios del área de Contabilidad que involucra al área de conciliaciones y contabilidad.

La finalidad de esta encuesta es saber cuántos usuarios hacen uso del sistema de recepción de Facturas y cuan beneficioso seria darle la funcionalidad para que les pueda facilitar la información mediante un reporte en formato Excel.

De la misma manera poder identificar las operaciones de los usuarios, necesidades y puntos de vista sobre este proyecto.

1. Identificar las características y funcionalidades del sistema para las operaciones administrativas de los diferentes departamentos administrativos que tiene la empresa, para elaborar un sistema a medida.

2. Implementar un modelo de Reportes en función de los requerimientos del y los usuarios.
3. Diseñar un sistema con una interfaz intuitiva, confiable y sobre todo que el usuario pueda controlar efectivamente las acciones de la máquina.
4. Diseñar una base de datos del sistema, confiable y segura para la información que permita ahorrar espacio, mediante el diseño de base de datos optimizadas y sin datos duplicados.

En este tipo de Proyecto de Software se empleó el método de investigación Cualitativa ya que el propósito de esta investigación se trata de determinar el punto de vista de las principales áreas de la empresa, donde harán uso de este sistema, explorar las causas del problema planteado y observar y determinar si es viable implementar el software.

PROCEDIMIENTO DE RECOLECCION

(DISEÑO DEL EXPERIMENTO, TRABAJO DE CAMPO, ENCUESTA)

Se llevó a cabo una recolección de datos, con una serie de encuestas a los Empleados del Área de conciliaciones, específicamente sobre el uso y utilización que le dan al repositorio de Facturas con el que cuenta la Empresa de Abarrotes Cordobeses, y estos fueron los resultados:

Sistema de Reportes para la Empresa de ACO

Proyecto Residencia Profesionales, para la Empresa SUPERIBERIA.



esc.delgadonogales@gmail.com (no compartidos)

[Cambiar de cuenta](#)



¿Sabias que tu empresa , cuenta con un sistema Repositorio de facturas, validadas?

- ☐ Sí
- ☐ No
- ☐ Tal vez

¿Haces uso de la información que contienen el "Repositorio de facturas recibidas" que administra tu empresa?

- ☐ Sí
- ☐ No
- ☐ Tal vez
- ☐ Algunas veces

¿En tus operaciones diarias, requieres de consultas que estén relacionadas con el Repositorio de facturas, es decir con el buzón de facturas ?

- ☐ Sí
- ☐ No
- ☐ Tal vez

¿ Tienes acceso al sistema Repositorio de facturas con el que cuenta tu Empresa?

- ☐ Sí
- ☐ No
- ☐ Tal vez

¿ A qué área solicitan el reporte de facturas, cuando se requiere?

- ☐ Sistemas TI
- ☐ Administración y Contabilidad
- ☐ Pagos y conciliaciones
- ☐ Otro

¿ Si no cuentas con acceso a esta información, que tan viable seria contar con un sistema que pueda facilitar esta información en un reporte en formato Excel?

- ☐ Quizás
- ☐ SI
- ☐ NO
- ☐ Tal vez

¿Te facilitaría en tus actividades diarias , el reporte de la información del repositorio de facturas?

- ☐ Sí
- ☐ No
- ☐ Tal vez
- ☐ Definitivamente si

¿ Contar con un sistema Web, te facilitaría realizar tus actividades de revisión con mayor facilidad ?

- ☐ Sí
- ☐ No
- ☐ Tal vez
- ☐ Haría mas productivas mis operaciones diarias.

Resultado sobre la recolección de datos.

En la primera pregunta obtuvimos un 78.6% de respuesta de los usuarios de la empresa, e indicaron que tenían conocimiento de que la Empresa contaba con un sistema repositorio de facturas.

¿Sabías que tu empresa , cuenta con un sistema Repositorio de facturas, validadas?

14 respuestas

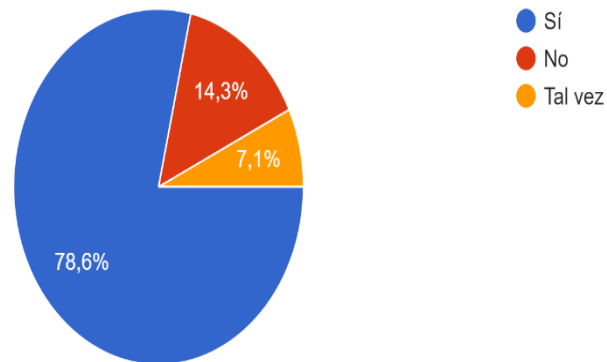


Figura 16. Resultado de Pregunta 1.

En la segunda pregunta obtuvimos que el 78.6% de los usuarios de la empresa, hacen uso de la información que se encuentra alojada en el repositorio de facturas.

¿Haces uso de la información que contienen el "Repositorio de facturas recibidas" que administra tu empresa?

14 respuestas

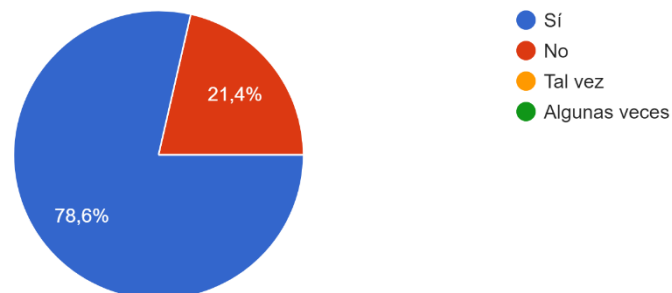


Figura 17. Resultado de Pregunta 2.

En la Tercera pregunta obtuvimos que el 78.6% de los usuarios de la empresa, requieren de consulta y obtención de información alojada en el repositorio de facturas.

¿En tus operaciones diarias, requieres de consultas que estén relacionadas con el Repositorio de facturas, es decir con el buzón de facturas ?

14 respuestas

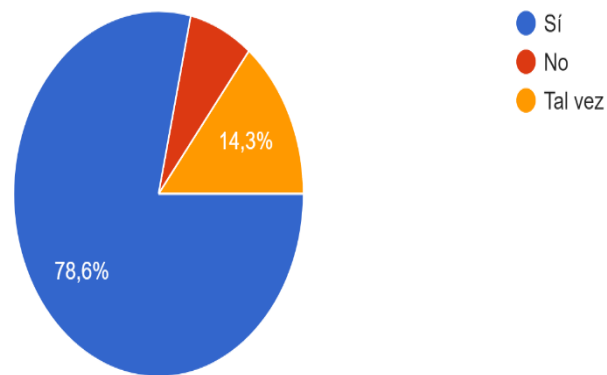


Figura 18. Resultado de Pregunta 3.

En la cuarta pregunta obtuvimos que el 71.4% de los usuarios de la empresa, contestaron que, si tienen acceso al repositorio de facturas, pero solo para consulta de información.

¿ Tienes acceso al sistema Repositorio de facturas con el que cuenta tu Empresa?

14 respuestas

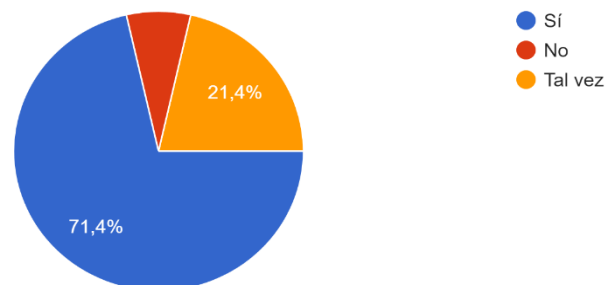


Figura 19. Resultado de Pregunta 4.

En la quinta pregunta obtuvimos que el 71.4% de los usuarios de la empresa, contestaron que el área, a la cual solicitan la información y/o reporte es a Sistemas TI.

¿ A qué área solicitan el reporte de facturas, cuando se requiere?

14 respuestas

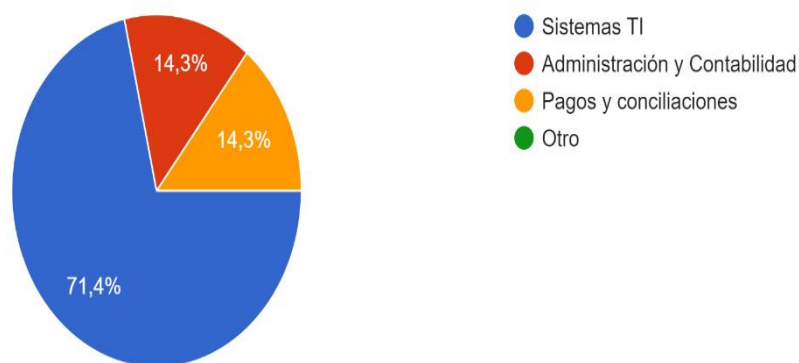


Figura 20 . Resultado de Pregunta 5.

En la sexta pregunta obtuvimos que el 42.9% de los usuarios de la empresa, contestaron que el área, a la cual solicitan la información y/o reporte es a Sistemas TI.

¿ Si no cuentas con acceso a esta información, que tan viable seria contar con un sistema que pueda facilitar esta información en un reporte en formato Excel?

14 respuestas

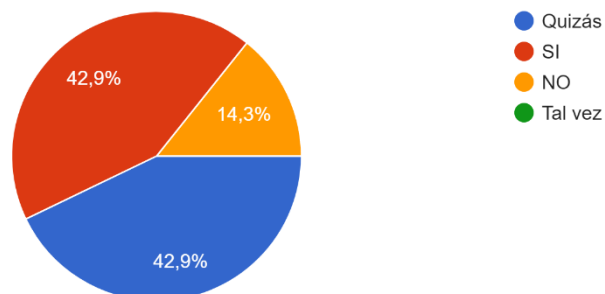


Figura 21. Resultado de Pregunta 6.

En la sexta pregunta obtuvimos que el 42.9% de los usuarios de la empresa, contestaron que sería de gran utilidad contar con un sistema de que emita un reporte del repositorio de facturas, para facilitar sus operaciones diarias.

¿Te facilitaría en tus actividades diarias , el reporte de la información del repositorio de facturas?

14 respuestas

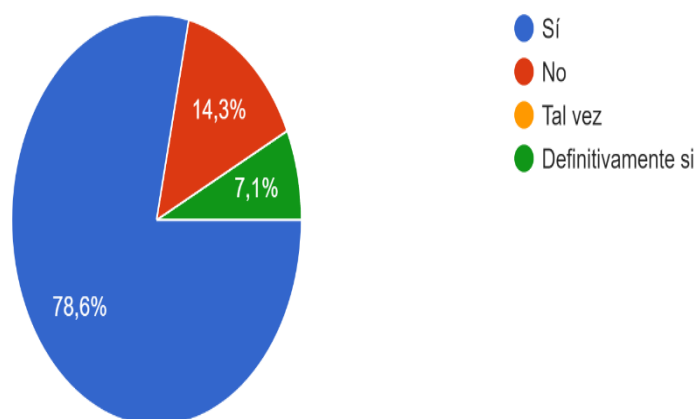


Figura 22. Resultado de Pregunta 7.

En la séptima pregunta obtuvimos que el 78.6% de los usuarios de la empresa, contestaron que sería de gran utilidad contar con un sistema Web de validación de facturas, para facilitar sus operaciones diarias.

¿ Contar con un sistema Web, te facilitaría realizar tus actividades de revisión con mayor facilidad ?

14 respuestas

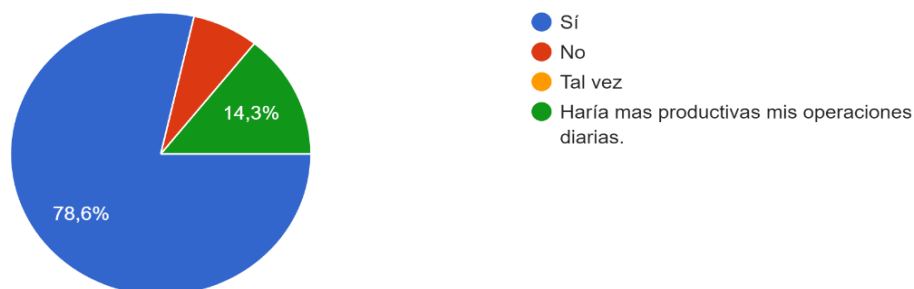


Figura 23. Resultado de Pregunta 8.

En base a la encuesta realizada dando un resultado de más del 75% de los usuarios del área de conciliaciones, pagos y toda el área administrativa indican que es una respuesta fue satisfactoria para poder contar e implementar un sistema de reportes para la empresa.

Por lo que si es viable el desarrollo de un sistema Web, que les permita a los usuarios de la Empresa de Abarrotes Cordobeses, obtener la información del repositorio de facturas mediante un reporte en formato Excel.

Capítulo 3 - Análisis de Requerimientos

Los requerimientos funcionales son los que definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

Es por ello que en base a entrevistas, cuestionarios y detallado de reportes se ha llegado a la reunión de los siguientes requerimientos

3.1 Requerimientos Funcionales

Tabla 2. Requerimiento Funcional 1. Ingresar al Sistema

RF 1	Ingresar al Sistema.		
Tipo: Requisito <u>Restricción</u>	Prioridad: <u>Esencial</u> Deseado Opcional		
Para que el administrador tenga acceso al sistema, deberá ingresar un nombre de usuario valido y una contraseña alfanumérica, en el caso de que estos no sean correctos, recibirá un mensaje de error, denegando el servicio de esta manera.			

Tabla 3. Requerimiento Funcional 2. Consulta Proveedor

RF 2	Consultar Proveedor		
Tipo: <u>Requisito</u> Restricción	Prioridad: <u>Esencial</u> Deseado Opcional		
El sistema será capaz de realizar la búsqueda de registros del proveedor y mostrar en pantalla los resultados obtenidos.			

Tabla 4. Requerimiento Funcional 3. Consultar Comprobante CFDI

RF 3	Consultar Factura CFDI		
Tipo: <u>Requisito</u> Restricción		Prioridad: <u>Esencial</u> Deseado Opcional	
Se presenta cuando se desea obtener información detallada sobre la factura o CFDI a partir de un identificador.			

Tabla 5. Requerimiento Funcional 4. Generar Reporte

RF 4	Generar Reporte		
Tipo: Requisito Restricción		Prioridad: Esencial Deseado <u>Opcional</u>	
El sistema será capaz de generar reporte de los comprobantes emitidos de parte de los proveedores, a partir del nombre de proveedor.			

Tabla 6. Requerimiento Funcional 5. Generar archivo PDF

RF 5	Generar archivo PDF		
Tipo: <u>Requisito</u> Restricción	Prioridad: <u>Esencial</u>	Deseado	Opcional
El sistema será capaz de generar un archivo PDF del comprobante CFDI a partir de la extracción de su UUID.			

Tabla 7. Requerimiento Funcional 6. Generar archivo XML

RF 6	Generar archivo PDF		
Tipo: <u>Requisito</u> Restricción	Prioridad: <u>Esencial</u>	Deseado	Opcional
El sistema será capaz de generar un archivo del comprobante CFDI a partir de la extracción de su UUID.			

3.2 Requerimientos No Funcionales

3.2.1 Requisitos de Rendimiento

Tabla 8. Requerimiento No Funcional 1. Consultas eficientes

RN 1	Consultas eficientes.		
Tipo:	<u>Requisito</u> Restricción	Prioridad:	<u>Esencial</u> Deseado Opcional
La información del sistema será almacenada en servidor de base de datos, lo cual incrementa la eficiencia por cada consulta, hasta en un 90%.			

Tabla 9. Requerimiento No Funcional 2. Uso del Sistema

RN 2	Uso del Sistema.		
Tipo:	<u>Requisito</u> Restricción	Prioridad:	<u>Esencial</u> Deseado Opcional
Por cada transacción que se efectué, esta no deberá consumir más del 50% de la capacidad del CPU.			

Tabla 10. Requerimiento No Funcional 3. Transacciones

RN 3	Transacciones.
Tipo: <u>Requisito</u> Restricción	Prioridad: <u>Esencial</u> Deseado Opcional
El 95% de las transacciones se deben realizar en menos de 60 segundos.	

Tabla 11. Requerimiento No Funcional 4. Usuarios con nombre

RN 4	Usuarios con nombre.
Tipo: <u>Requisito</u> Restricción	Prioridad: <u>Esencial</u> Deseado Opcional
El sistema será capaz de soportar varios usuarios a la vez, a manera de súper usuario privilegiado (administrador).	

3.2.2 Requisitos de Seguridad

Tabla 12. Requerimiento No Funcional 5. Contraseñas

RN 5	Contraseñas.
Tipo: <u>Requisito</u> Restricción	Prioridad: <u>Esencial</u> Deseado Opcional
Uso de contraseñas para cada usuario o administrador, con la finalidad de restringir el acceso al sistema a personas ajenas o inclusive propias de la empresa, pero con privilegios no autorizados.	

Tabla 13. Requerimiento No Funcional 6. Privilegios

RN 6	Privilegios.
Tipo: <u>Requisito</u> Restricción	Prioridad: <u>Esencial</u> Deseado Opcional
El sistema tendrá diferentes niveles de jerarquía entre sus usuarios, otorgándoles diferentes privilegios de acuerdo a su nivel de accesibilidad.	

Tabla 14. Requerimiento No Funcional 7. Bitácoras.

RN 7	Bitácoras.
Tipo: <u>Requisito</u> Restricción	Prioridad: <u>Esencial</u> Deseado Opcional
El sistema registrará en sus bitácoras la total actividad que realice el usuario, con la finalidad de identificar fácilmente errores cometidos principalmente por el administrador y demás usuarios que pueda haber ingresado al sistema.	

3.2.3 Requisitos de Fiabilidad

Tabla 15. Requerimiento No Funcional 8. Excepciones

RN 8	Excepciones.		
Tipo: <u>Requisito</u> Restricción		Prioridad: <u>Esencial</u> Deseado Opcional	
El sistema debe ser capaz de identificar y evadir o solucionar en el menor tiempo posible, todas y cada una de las excepciones que pudieran presentarse durante			

Tabla 16. Requerimiento No Funcional 9. Recuperación de fallos

RN 9	Recuperación de fallos.		
Tipo: <u>Requisito</u> Restricción		Prioridad: <u>Esencial</u> Deseado Opcional	
El sistema debe ser capaz de recuperase ante un fallo lógico y la posibilidad de dar marcha atrás en la definición del perfil de cada usuario.			

3.2.4 Requisitos de Disponibilidad

Tabla 17. Requerimiento No Funcional 10. Disponibilidad

RN 10	Disponibilidad.		
Tipo: <u>Requisito</u>	Restricción	Prioridad: <u>Esencial</u>	Deseado Opcional
El sistema estará disponible, el 100% del horario laboral de los administradores y demás usuarios.			

Tabla 18. Requerimiento No Funcional 11. Inicial

RN 11	Inicial.		
Tipo: <u>Requisito</u> Restricción		Prioridad: <u>Esencial</u> Deseado Opcional	
Una vez que el sistema acepto el nombre de usuario y contraseña, no deberá demorar más de 6 segundos para mostrar la siguiente ventana.			

Tabla 19. Requerimiento No Funcional 12. Reportes Generados.

RN 12	Reportes Generados
Tipo: <u>Requisito</u> Restricción	Prioridad: <u>Esencial</u> Deseado Opcional
El sistema deberá generar el 99.9% del reporte en menos de 15 segundos.	

3.2.5 Requisitos de Mantenibilidad

Se llevó a cabo un análisis del funcionamiento del sistema, con el fin de detectar las fallas que este pueda presentar, para así estar prevenidos a los errores que se manifiesten en la ejecución y lograr el rendimiento más óptimo.

3.2.6 Requisitos de Portabilidad

Tabla 20. Requerimiento No Funcional 14. Sistemas Operativos.

RN 13	Sistemas Operativos.
Tipo: <u>Requisito</u> Restricción	Prioridad: <u>Esencial</u> Deseado Opcional
El sistema será compatible con sistemas operativos propios de Windows, específicamente en las ediciones, Windows 8, Windows 10 y Windows 11.	

Tabla 21. Requerimiento No Funcional 14. Lenguaje de programación.

RN 14	Lenguaje de programación.
Tipo: <u>Requisito</u> Restricción	Prioridad: <u>Esencial</u> Deseado Opcional
Para mayor portabilidad se hace uso del lenguaje de programación Java, por su compatibilidad entre dispositivos.	

Tabla 22. Requerimiento No Funcional 15. Bases de datos.

RN 14	Bases de Datos.
Tipo: <u>Requisito</u> Restricción	Prioridad: <u>Esencial</u> Deseado Opcional
LA base de datos es gestionada en SQL, y este a su vez, puede funcionar tanto en Windows como en Linux.	

3.3 Otros Requerimientos

Tabla 23. Otro requerimiento 1. Legalidad de datos

RO 1	Legalidad de datos.		
Tipo: Requisito Restricción		Prioridad: Esencial Deseado Opcional	
La empresa se compromete a mantener la privacidad e integridad de los datos proporcionados e ingresados en el Software.			

Tabla 24. Otro requerimiento 2. Interfaz

RO 2	Interfaz.		
Tipo: Requisito Restricción		Prioridad: Esencial Deseado Opcional	
El sistema carece de facilidad para interactuar con algún otro sistema externo, los datos son almacenados en una base de datos y están a disposición del administrador.			

3.4 Estudio de Factibilidad

Con la implementación de un nuevo software dentro de la Empresa, se da la oportunidad de hacer un estudio de factibilidad que comprende tres diferentes ámbitos, este es un indicador de costos, beneficios y grado de aceptación por y para el departamento de conciliaciones.

3.4.1 Factibilidad Técnica

Se analizó la información correspondiente a los recursos materiales con los que cuenta el departamento de conciliaciones, para que a partir de este se identificaran totalmente las especificaciones que deben cumplir los recursos, así como las características de cada uno de ellos, y de esta manera ver si existe la posibilidad de hacer uso de los mismos en el desarrollo e implementación del sistema implementado y en caso de no contar con ellos, cotizar aquellos que deben ser adquiridos para el desarrollo y puesta en marcha total del sistema.

De acuerdo a los recursos necesarios para la implementación del sistema, se analizaron los recursos tanto de hardware como de software. De los cuales se

determinó que no era necesario adquirir o actualizar los ya disponibles en el departamento, pues el hardware (Tabla 25) está en óptimas condiciones y por parte del software (Tabla 26), no hay mayor problema debido a que las herramientas de desarrollo son de licencia libre.

Tabla 25. Especificaciones del Hardware.

Modelo	
Marca	Lenovo
Serie	320-14IKBN
Procesador	
Marca	AMD
Línea	e
Modelo	3020
Velocidad	1.20 GHz
Memoria	
Capacidad	8.0 GB
Tipo	DDR4-SDRAM
Disco Duro	
Capacidad	932GB
Pantalla	
Tipo	LED
Tamaño	35.6 cm (14")
Resolución	1366 x 768 Pixeles
Gráficos	
Tarjeta de Vídeo	No disponible
Modelo de gráficos en tarjeta	Intel® HD Graphics 620
Adaptador gráfico en tablero	✓
Frecuencia base de gráficos	300 MHz
Frecuencia dinámica máxima de gráficos	1000 MHz
Placa Madre	
Interfaz	USB 2.0
Tarjeta de red	Conexión RJ-45

Tabla 26. Especificaciones del Software.

Software	Versión	Licencia
Sistema Operativo	Windows 10 64 bits	Propietario
Lenguaje de Desarrollo	Java EE 6	Libre
Entorno de desarrollo de interfaz	Netbeans7.3 Windows	Libre
Administrador de Base de datos	SQL Server	Libre
Servidor de Aplicaciones	GlassFish ogs-3.1.2.2-windows	Libre

3.4.2 Factibilidad Económica

Estudio económico donde se determinan las condiciones financieras necesarias para la implantación del sistema. (Retorno de inversión inicial y punto de equilibrio, etc).

Costos de recursos Tecnológicos.

Como podemos observar en la (Tabla 26), el costo total de los recursos tecnológicos necesarios asciende a \$13,899, sin embargo, dichos recursos ya están en existencia dentro del departamento, de conciliaciones, por lo tanto, el área de finanzas no deberá contemplar esta inversión.

Tabla 27. Costo Recursos Tecnológicos.

Recurso	Descripción	Costo
Equipo PC	Laptop- Lenovo320-14IKBN AMD 3020e 1.20 GHz RAM 8.0 GB DDR4-SDRAM HDD 932GB	\$ 10,300.00
Sistema Operativo	Windows 10 64 bits	\$ 3,599.00
Lenguaje de Desarrollo	Java EE 6	\$ -
Entorno de desarrollo de interfaz	Netbeans7.3 Windows	\$ -
Administrador de Base de datos	SQL Server	\$ -
Servidor de Aplicaciones	GlassFish ogs-3.1.2.2-windows	\$ -
Total		\$ 13,899.00

Costos de recursos humanos: En la siguiente Tabla (28), los costos por los recursos humanos fundamentales para el desarrollo de este software, ascienden a por lo menos \$55,300 pesos, sin embargo, al tratarse de un proyecto de residencias profesionales, se maneja bajo convenio en el cual se establece no recibir remuneración económica, es por ello que el área de finanzas no debe contemplar este gasto.

Tabla 28. Costo Recursos Humanos.

Personal	Tiempo de Trabajo	Pago Mensual	Costo Total
Desarrollador	5	\$ 7,600.00	\$ 38,000.00
Analista	1	\$ 8,400.00	\$ 8,400.00
Diseñador	1	\$ 8,900.00	\$ 8,900.00
Total			\$ 55,300.00

3.4.3 Factibilidad Operacional

El objetivo a nivel operacional, fue que los usuarios tengan la capacidad para la utilización del sistema, logrando que diversos de los procesos realizados en él se lleven a cabo de manera eficiente y rápida, sin ser difícil de operar.

Para el manejo de este sistema, no se requiere una capacitación intensiva, ya que se buscó la manera en que el usuario tenga facilidad para interactuar con el sistema y cuente con una interfaz sencilla que facilite su uso diario, para que el usuario pueda disponer de la información cuando él lo requiera.

Además, los usuarios cuentan con el buen funcionamiento del sistema, presentando de igual manera una interfaz amigable, la cual fue desarrollada haciendo que todas las pantallas presentaran una estructura estática lo que se traduce en una herramienta de fácil manejo y comprensión.

Capítulo 4 - Desarrollo del proyecto
4.1 Casos de Uso

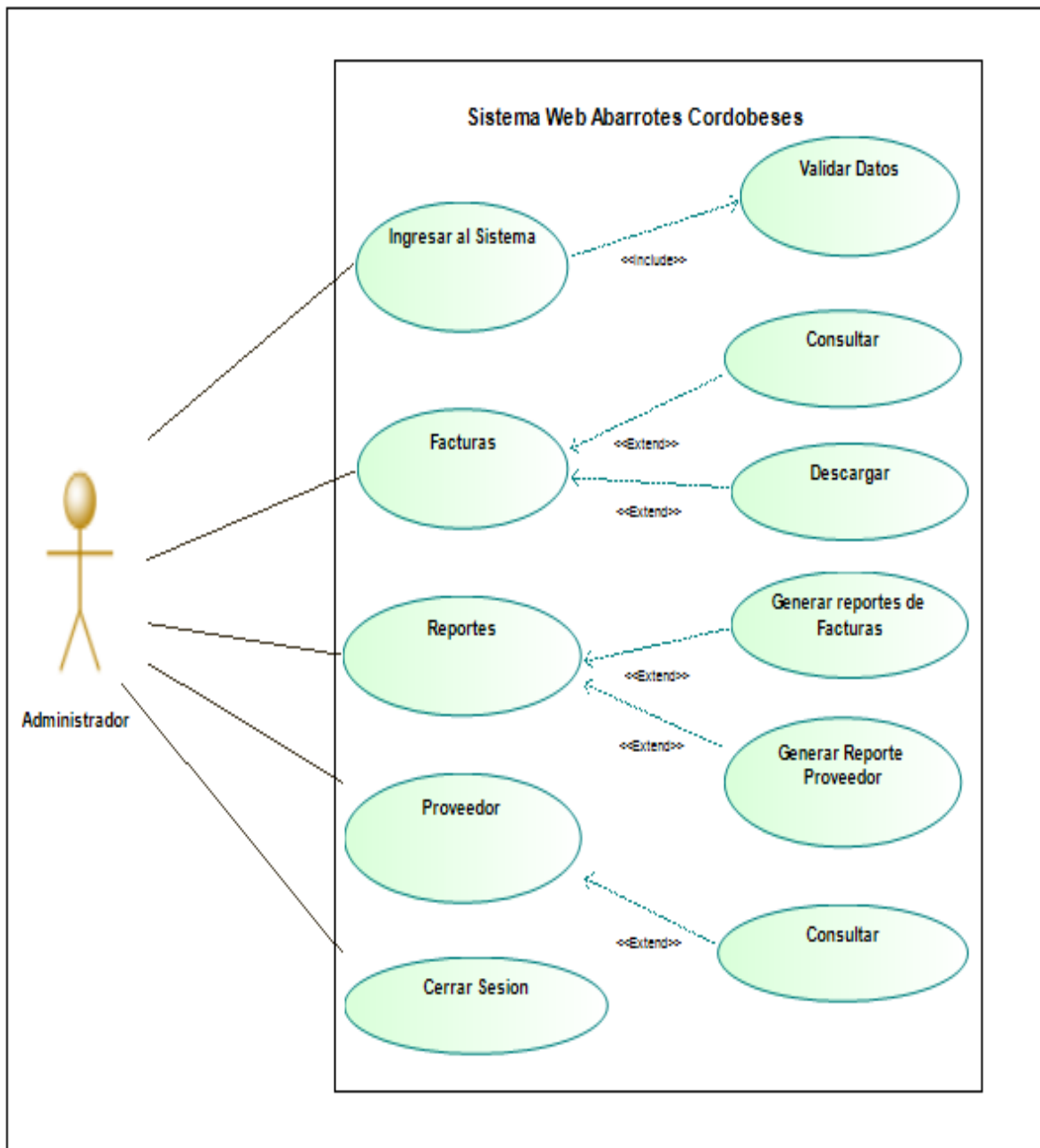


Figura 24. Caso de uso General

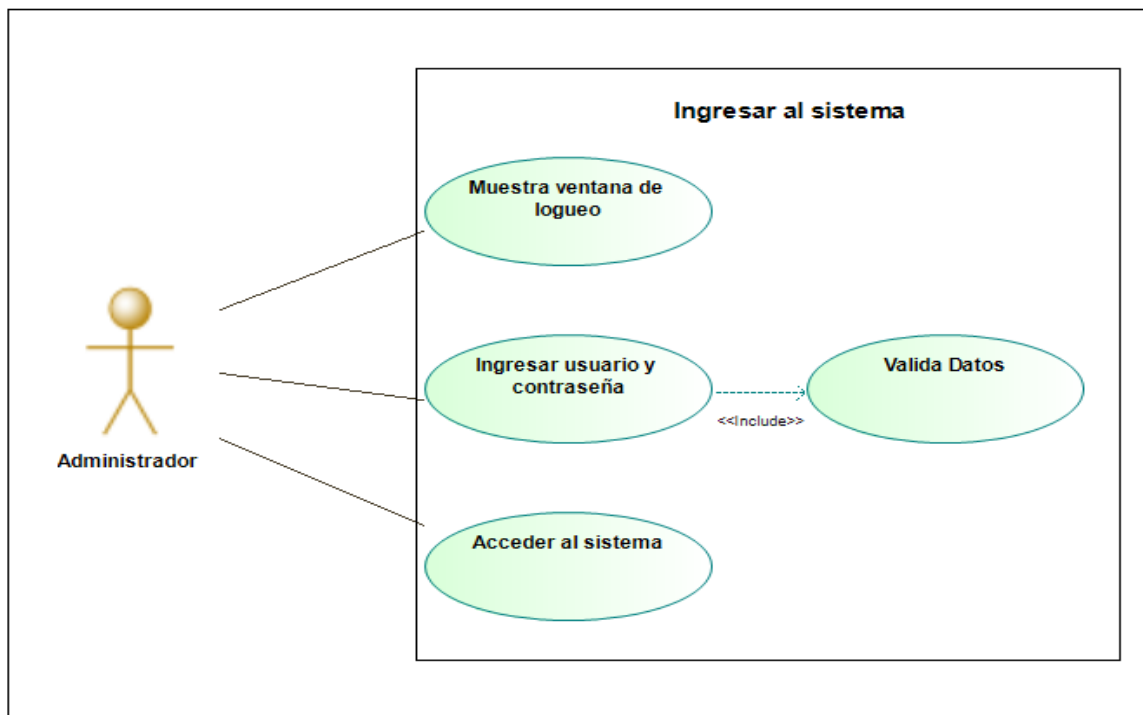


Figura 25. Caso de uso 1. Ingresar al sistema.

Tabla 29. Caso de Uso 1. Ingresar al sistema.

CU 1	Ingresar al sistema	
Descripción	El usuario tendrá acceso al sistema, solo si está registrado dentro de este de alguno u otra manera.	
Actor Principal	Administrador	Frecuencia de uso: Baja Media Alta
Precondiciones	Se requiere que el usuario este registrado o cuente con derechos en el sistema para poder ingresar.	
Postcondiciones	El usuario tendrá acceso a las diferentes herramientas disponibles en la gama de tareas posibles a realizar.	
Disparador	Inicializar el software.	
Curso básico de eventos	1.- Inicializar el software 2.- Ingresar usuario correcto 3.- Ingresar la contraseña correcta 4.- Oprimir el botón ingresar 5.- Acceder al sistema	
Extensiones de eventos		
Excepciones	Si el usuario y contraseña es incorrecto mandará mensaje de error ("Usuario o contraseña incorrectos") Regresa al paso 2.	

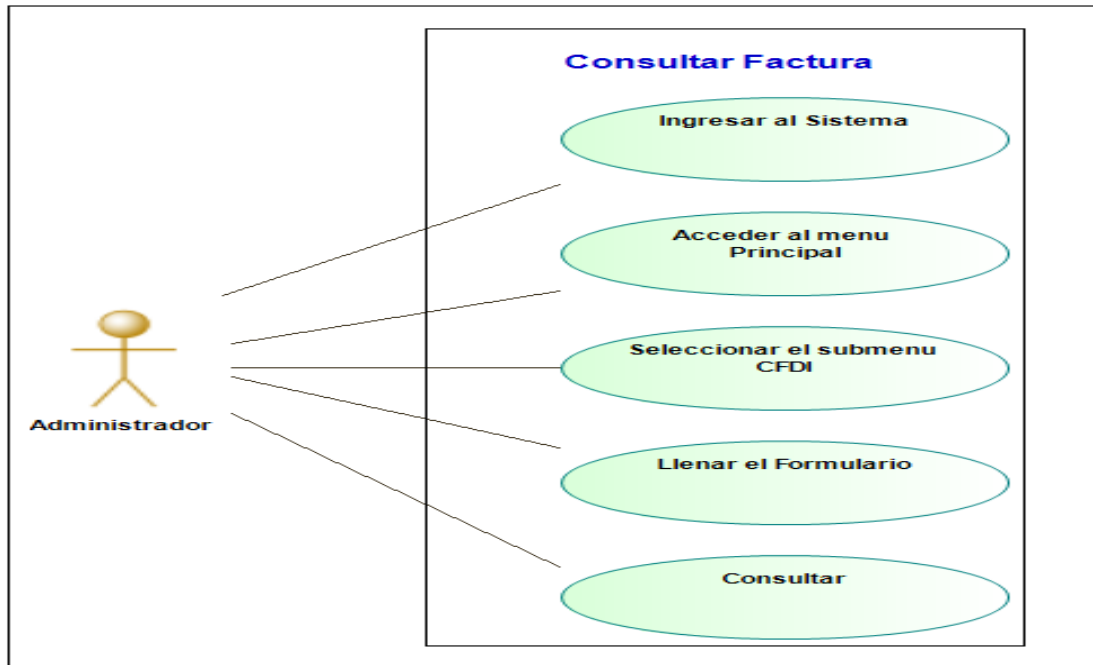


Figura 26. Caso de uso 2. Consultar Factura

Tabla 30. Caso de Uso 2. Consultar Factura.

CU 2	Consultar Factura	
Descripción	El Administrador o empleado podrá consultar la información general del registro de alguna Factura que este desee consultar y así obtener los datos del documento para su visualización.	
Actor Principal	Administrador	Frecuencia de uso: Baja Media <u>Alta</u>
Precondiciones	Se requiere que el folio de registro de la factura a consultar este activa en el sistema para poder obtener los datos e información.	
Postcondiciones	El usuario podrá visualizar la información general de la factura y podrá obtener la factura digital.	
Disparador	Inicializar el software.	
Curso básico de eventos	1.- Ingresar al sistema. 2.- Ingresar el usuario correcto. 3.- Ingresar contraseña correcta 4.- Mostrar un submenú de CFDI. 5.- Ingresar folio de factura. 6.-Consultar los datos generales.	
Extensiones de eventos		
Excepciones	Si el usuario y contraseña es incorrecto mandará mensaje de error ("Usuario o contraseña incorrectos") Regresa al paso 3.	

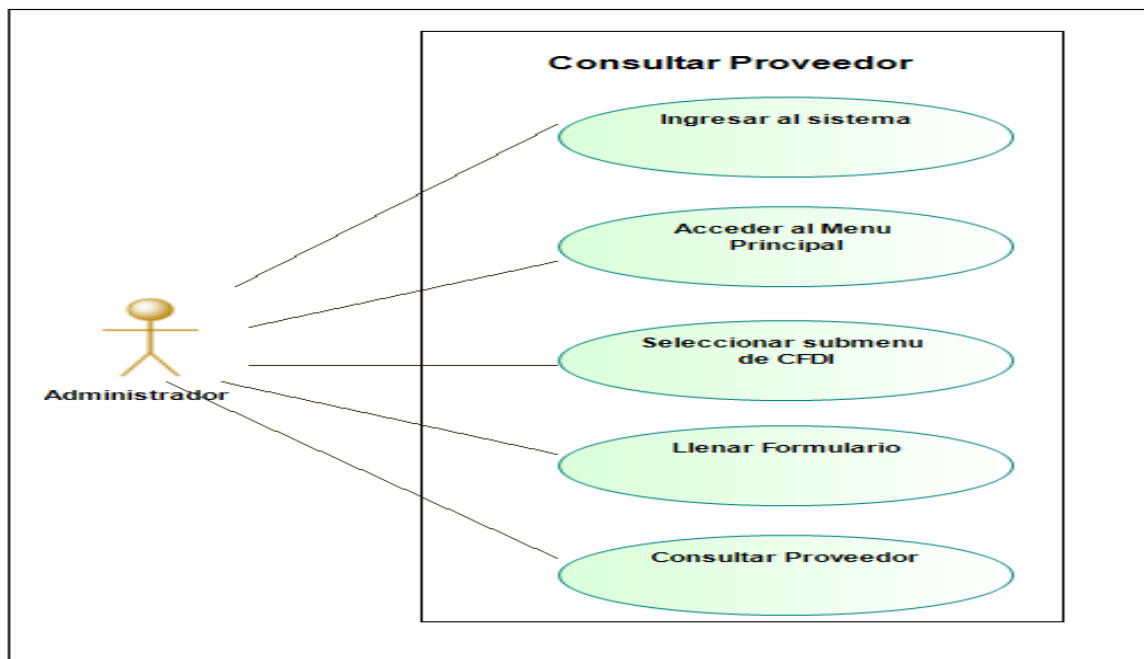


Figura 27. Caso de uso 3. Consultar Proveedor.

Tabla 31. Caso de Uso 3. Consultar Proveedor.

CU 3	Consultar Proveedor	
Descripción	El Administrador o empleado podrá consultar la información general del registro de alguna Factura que este desee consultar y así obtener los datos del documento para su visualización.	
Actor Principal	Administrador	Frecuencia de uso: Baja Media <u>Alta</u>
Precondiciones	Se requiere que el folio de registro de la factura a consultar este activa en el sistema para poder obtener los datos e información.	
Postcondiciones	El usuario podrá visualizar la información general de la factura y podrá obtener la factura digital.	
Disparador	Iniciar el software.	
Curso básico de eventos	1.- Ingresar al sistema. 2.- Ingresar el usuario correcto. 3.- Ingresar contraseña correcta 4.- Mostrar un submenú de CFDI. 5.- Ingresar folio de factura. 6.-Consultar los datos generales.	
Extensiones de eventos		
Excepciones	Si el usuario y contraseña es incorrecto mandará mensaje de error ("Usuario o contraseña incorrectos") Regresa al paso 3.	

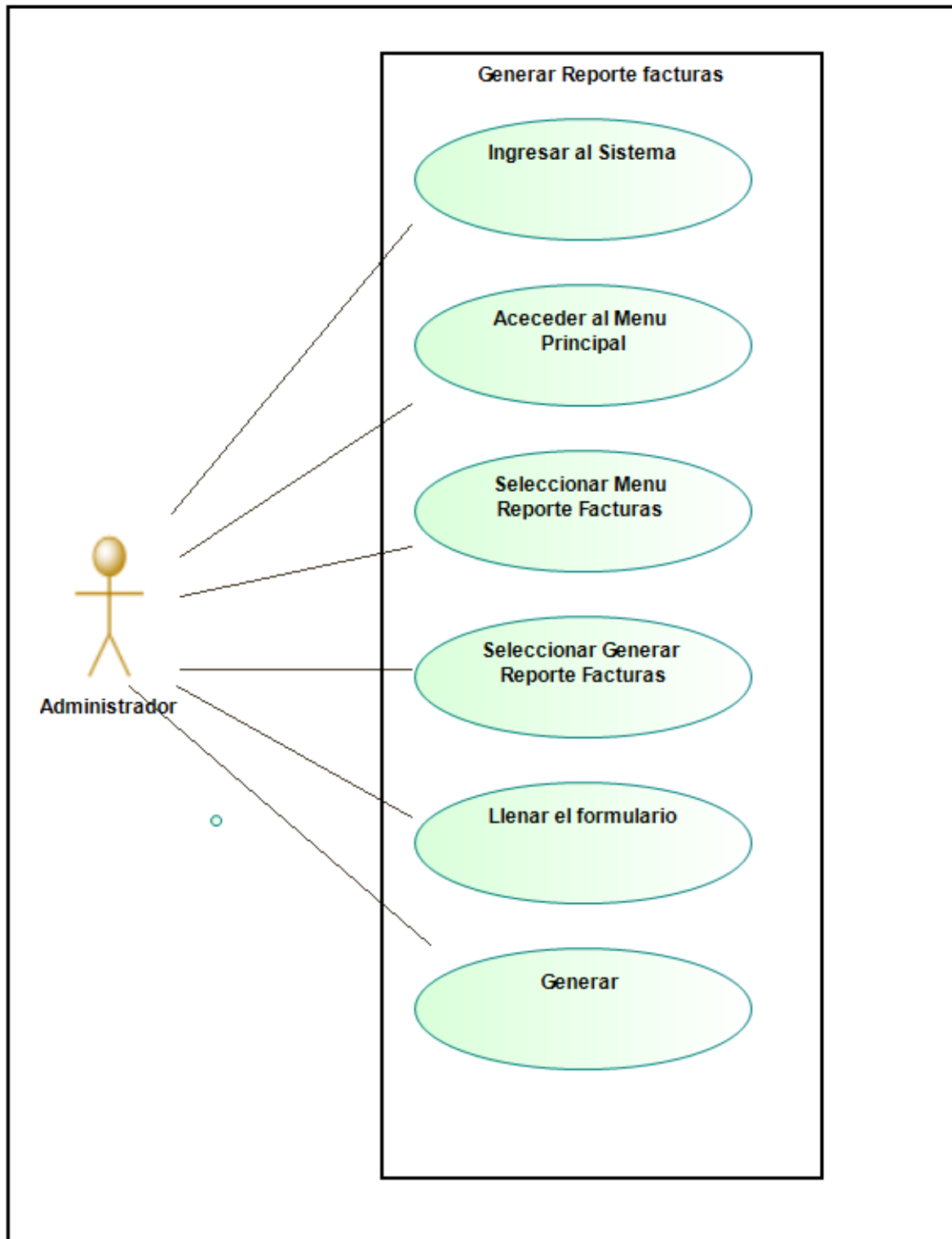


Figura 28. Caso de uso 4. Generar Reporte Facturas.

Tabla 32. Caso de Uso 4. Generar Reporte Facturas.

CU 4	Generar Reporte Facturas	
Descripción	EL administrador podrá generar reportes de todas las facturas que los han enviado al buzón de proveedores durante un determinado periodo, esto con la finalidad de llevar un buen control en las operaciones administrativas.	
Actor Principal	Administrador	Frecuencia de uso: Baja <u>Media</u> Alta
Precondiciones	Se requiere que el administrador tenga el privilegio para poder ingresar a este módulo y la fecha sea correcta para que esto pueda dar la generación de los datos correctos.	
Postcondiciones	El administrador tendrá los privilegios para exportar e imprimir el reporte generado.	
Disparador	Presionar botón aceptar del apartado generar reporte de Facturas.	
Curso básico de eventos	1.- Ingresar al sistema. 2.- Ingresar usuario. 3.- Ingresar contraseña. 4.- Mostrar un menú principal. 5.- Seleccionar reportes. 6.- Seleccionar reporte facturas 7.- Llenar parámetros. 8.- Mostrar reporte de acuerdo a la fecha solicitada. 9.- Impresión y exportación del reporte.	
Extensiones de eventos		
Excepciones	Si el usuario y contraseña es incorrecto manda mensaje de error ("Usuario o contraseña incorrectos, vuelva a intentarlo") Regresa al paso 3.	

4.2 Diagramas de Secuencia

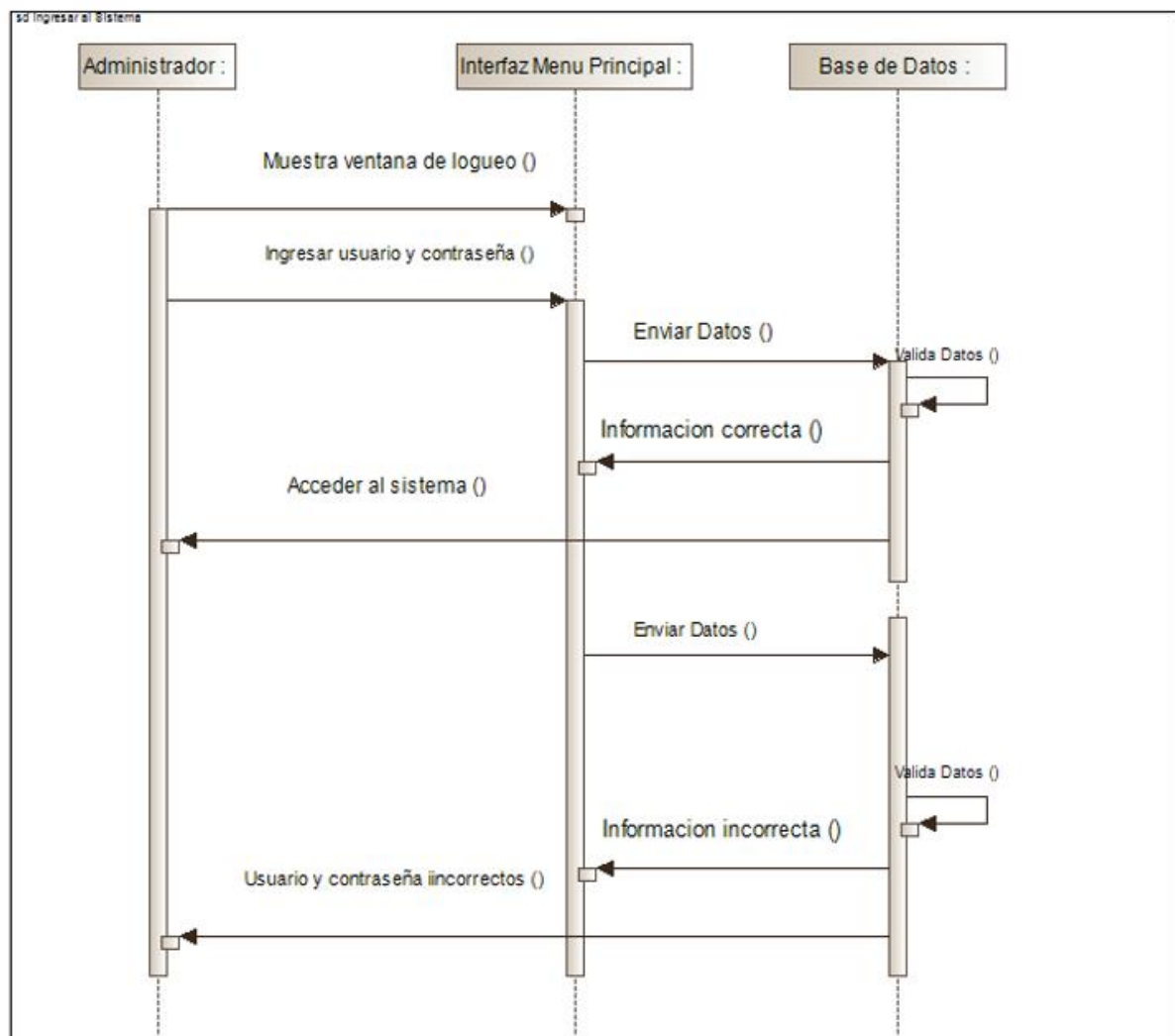


Figura 29 . Diagrama de secuencia 1. Ingresar al Sistema.

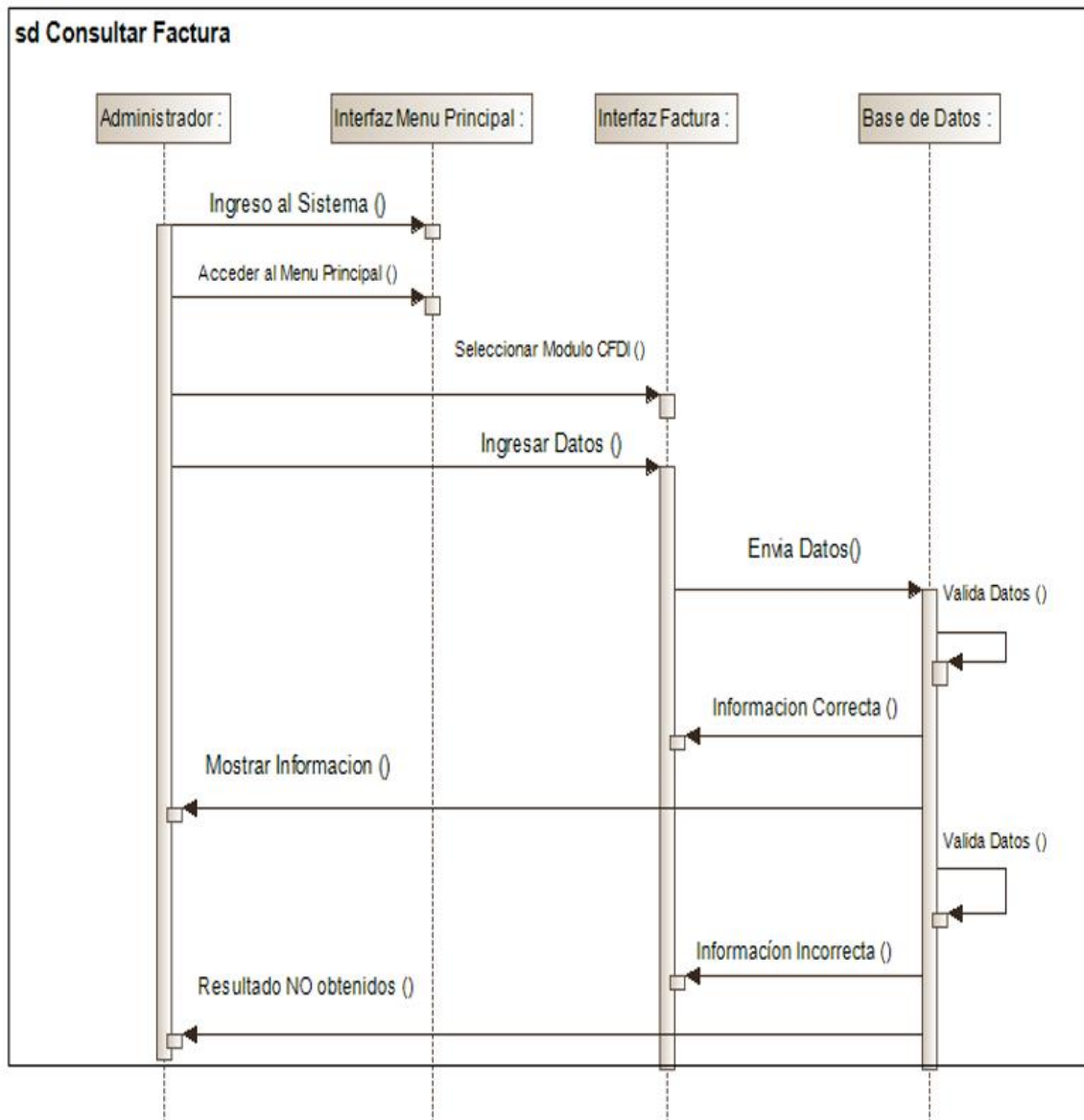


Figura 30. Diagrama de secuencia 2. Consultar Factura

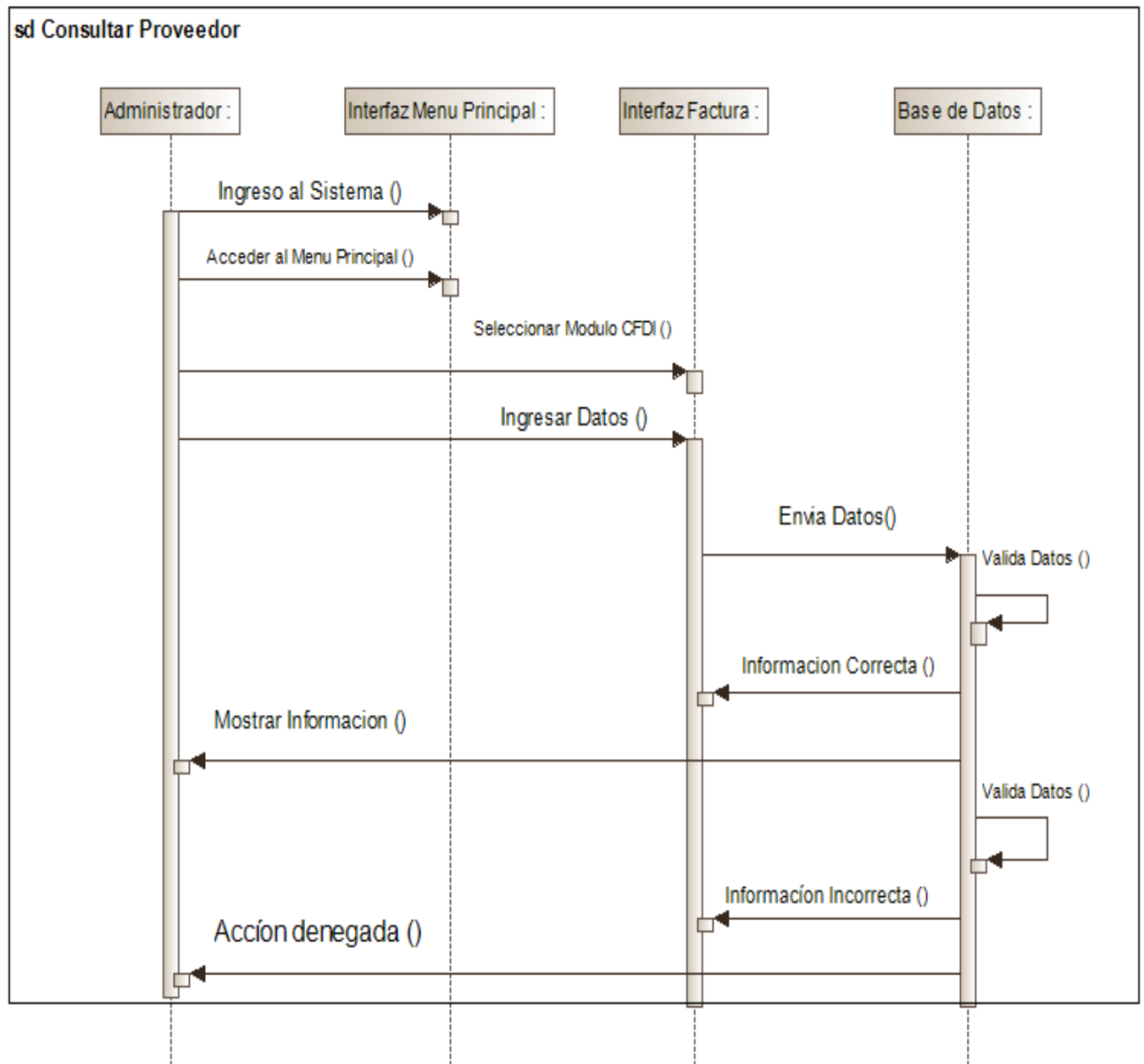


Figura 31 . Diagrama de secuencia 3. Consultar Proveedor.

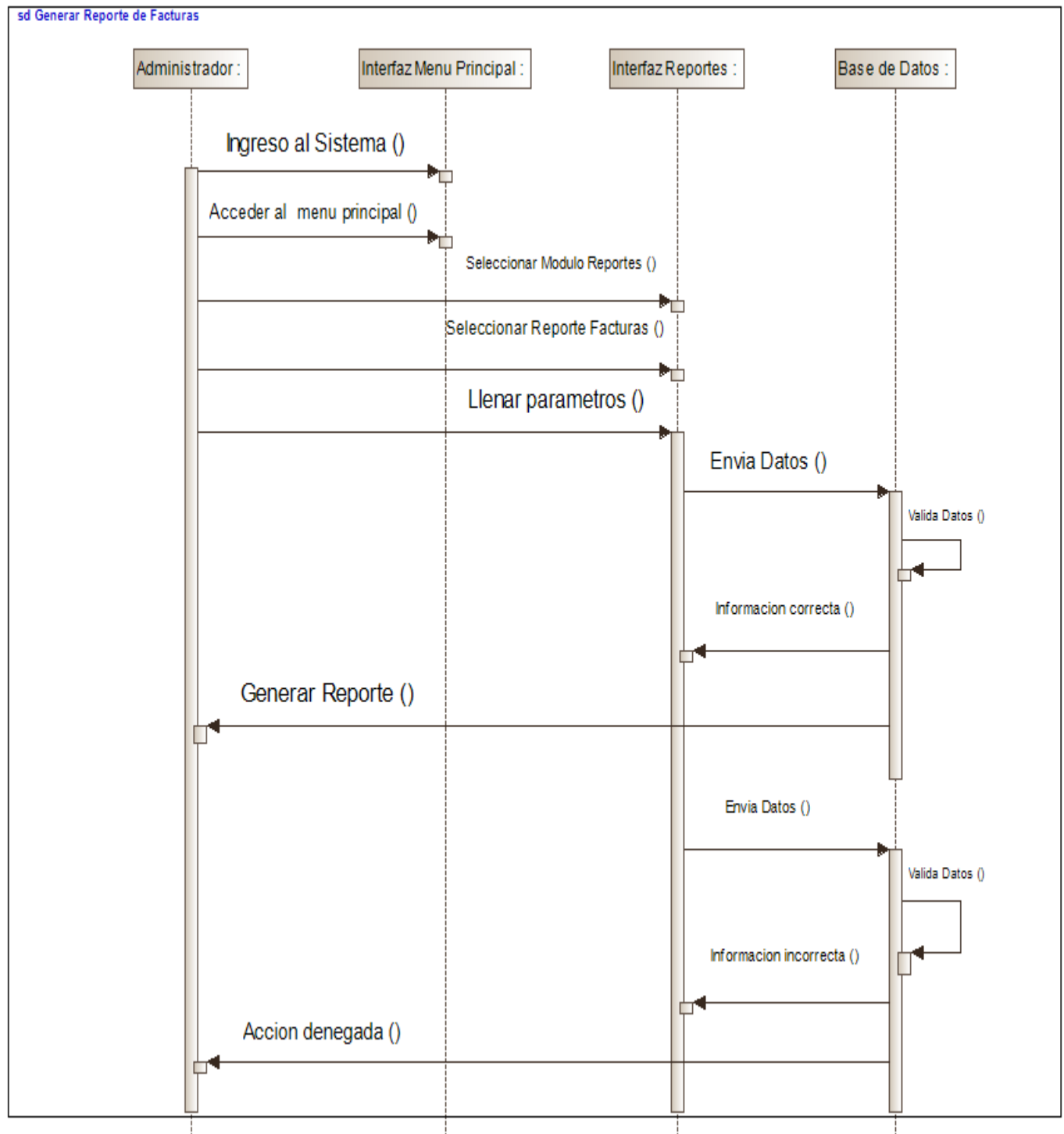


Figura 32 . Diagrama de secuencia 4. Generar Reporte Facturas.

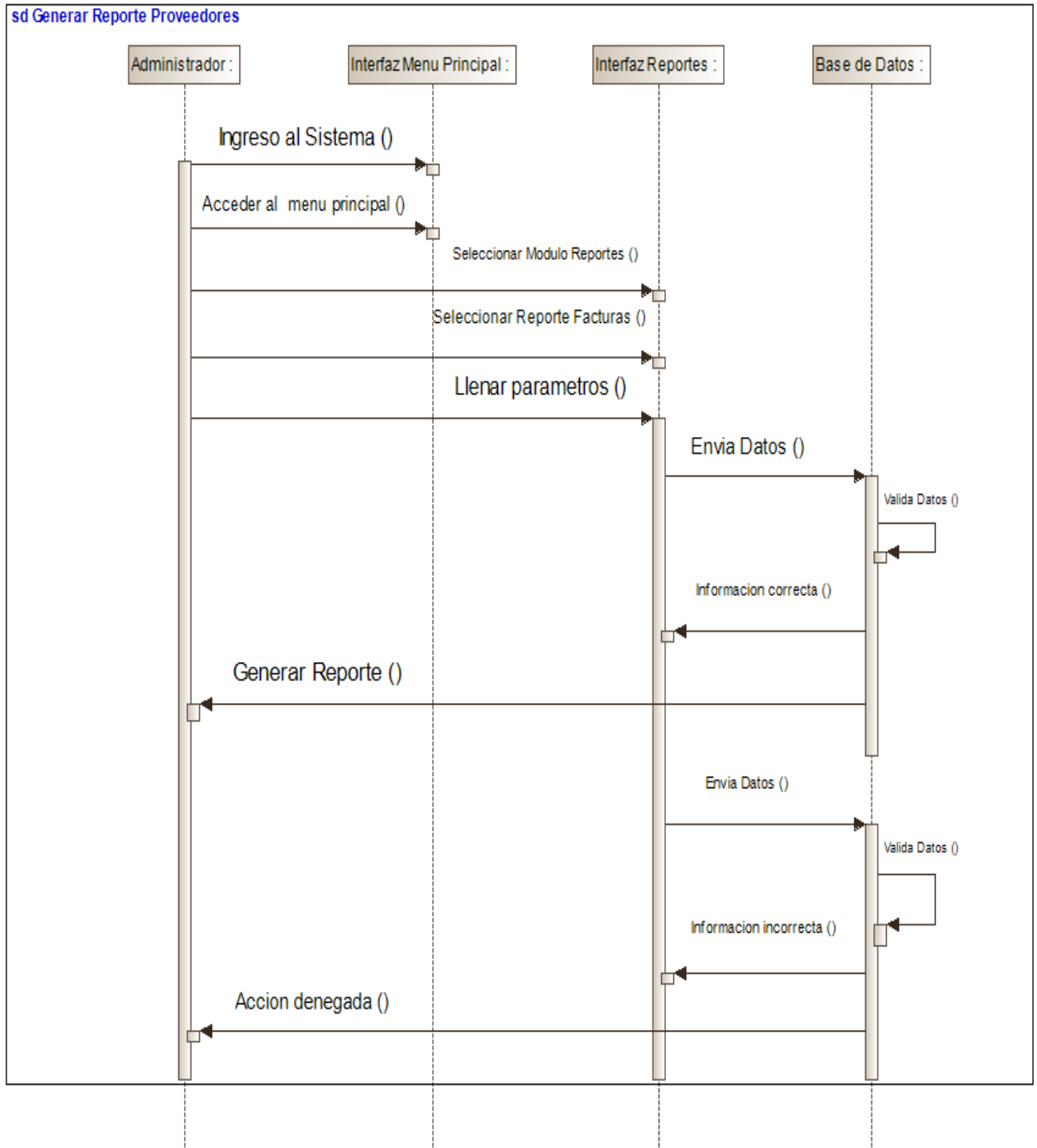


Figura 33. Diagrama de secuencia 5. Generar Reporte Proveedores.

4.3 Diagramas de Clases

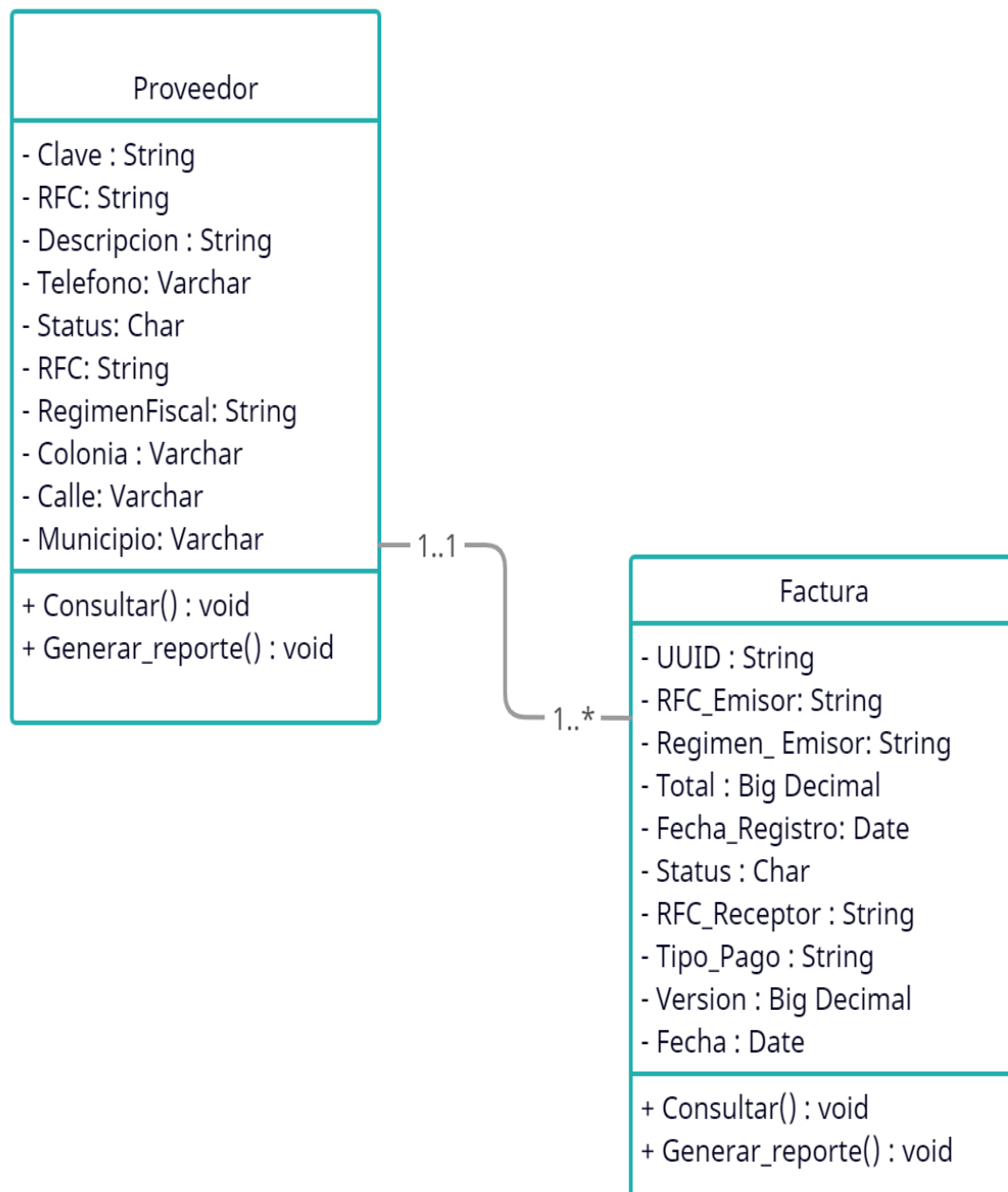


Figura 34. Diagrama de clases.

4.4 Diagrama de Contexto

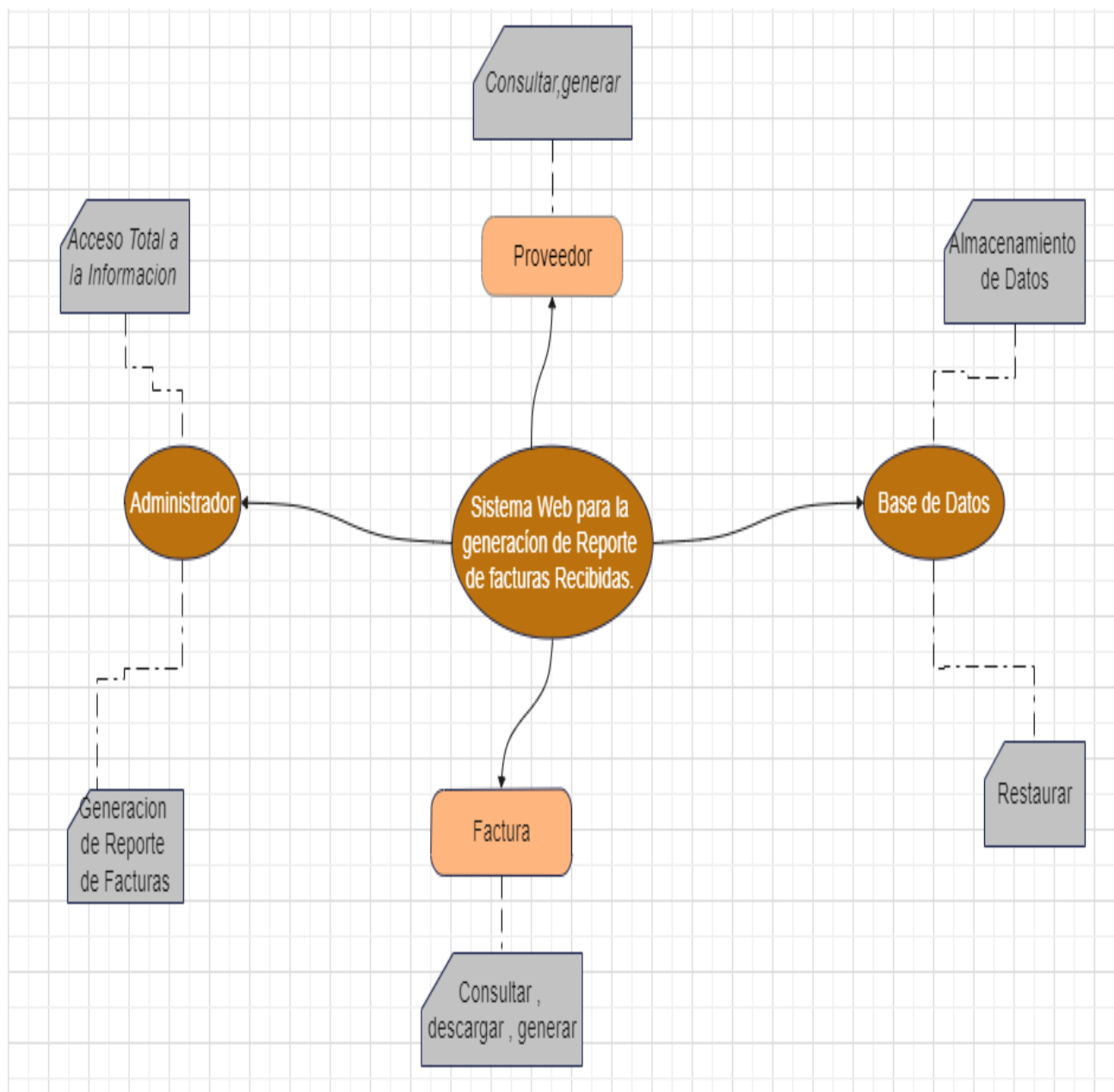


Figura 35. Diagrama de contexto.

4.5 Diseño de la Base de Datos

4.5.1 Modelo Entidad-Relación

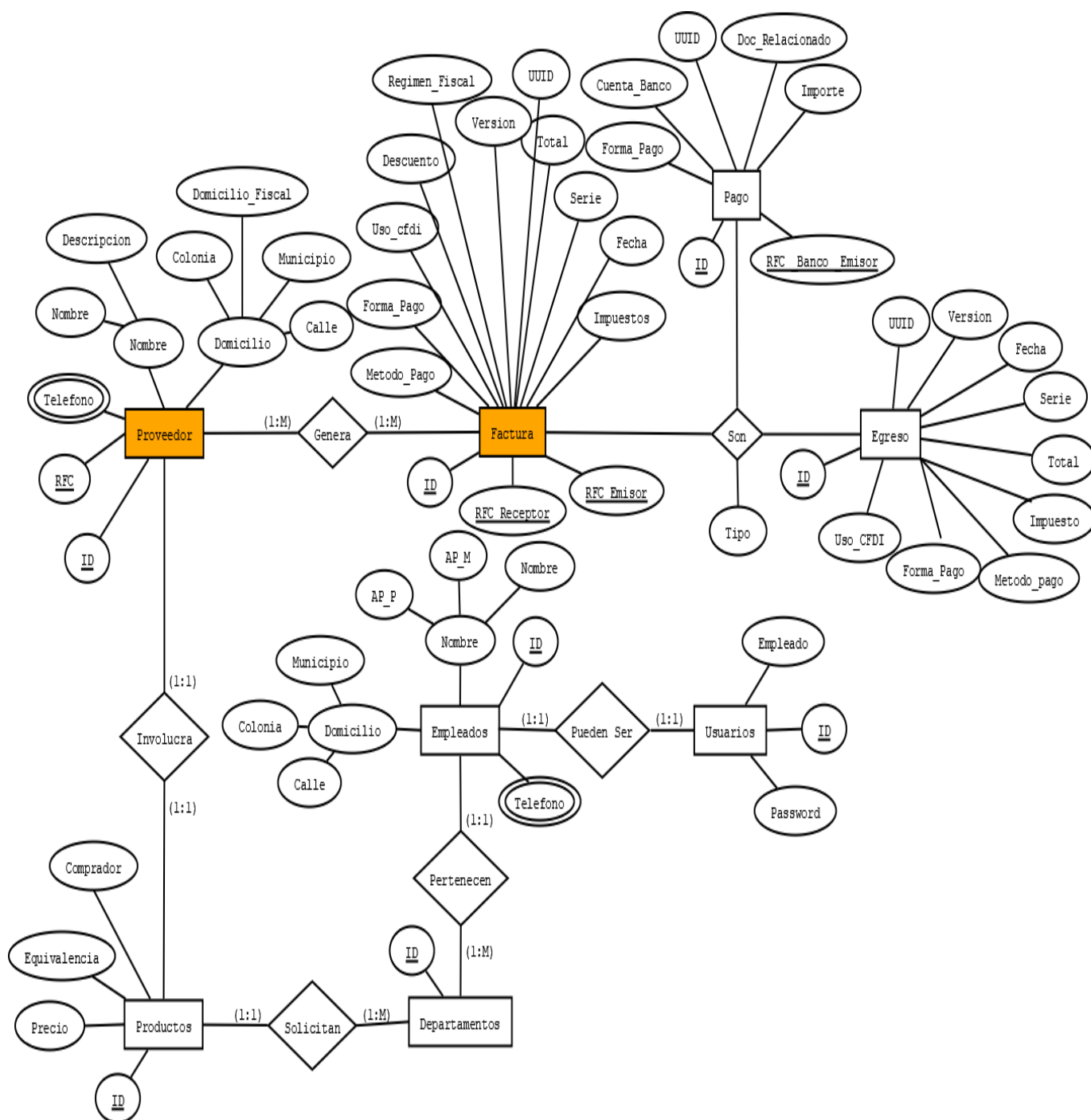


Figura 36. Modelo E-R del sistema general de Abarrotes Cordobeses.

4.5.2 Normalización del Modelo Entidad-Relación

PRIMERA FORMA NORMAL

FACTURA

ID	Metodo_Pago	Forma_Pago	Uso_CFDI	Descuento	Versión
Fecha	UUID	Total	RFC_Emisor	RFC_Receptor	

PROVEEDOR

ID	RFC	Nombre	Domicilio	Teléfono	Regimen_Fiscal
----	-----	--------	-----------	----------	----------------

SEGUNDA FORMA NORMAL

FACTURA

ID	Metodo_Pago	Forma_Pago	Uso_CFDI	Descuento	Versión
Fecha	UUID	Total	RFC_Emisor	RFC_Receptor	Mes
Año	Día				

PROVEEDOR

ID	RFC	Nombre	Domicilio	Telefono	Regimen_Fiscal
Domicilio_Fiscal	Municipio	Colonia	Calle		

TERCERA FORMA NORMAL

FACTURA

ID	Metodo_Pago	Forma_Pago	Uso_CFDI	Descuento	Version
Fecha	UUID	Total	RFC_Emisor	RFC_Receptor	Mes
Año	Día				

FACTURA_IMPLICA_PROVEEDOR

ID_FACTURA	ID_PROVEEDOR
------------	--------------

PROVEEDORES

ID	RFC	Nombre	Domicilio	Teléfono	Regimen_Fiscal
Domicilio_Fiscal	Municipio	Colonia	Calle		

PROVEEDOR_EMITE_FACTURA

ID_PROVEEDOR	ID_FACTURA
--------------	------------

Tabla 33. Normalización del Modelo Entidad-Relación.

4.5.3 Modelo Relacional

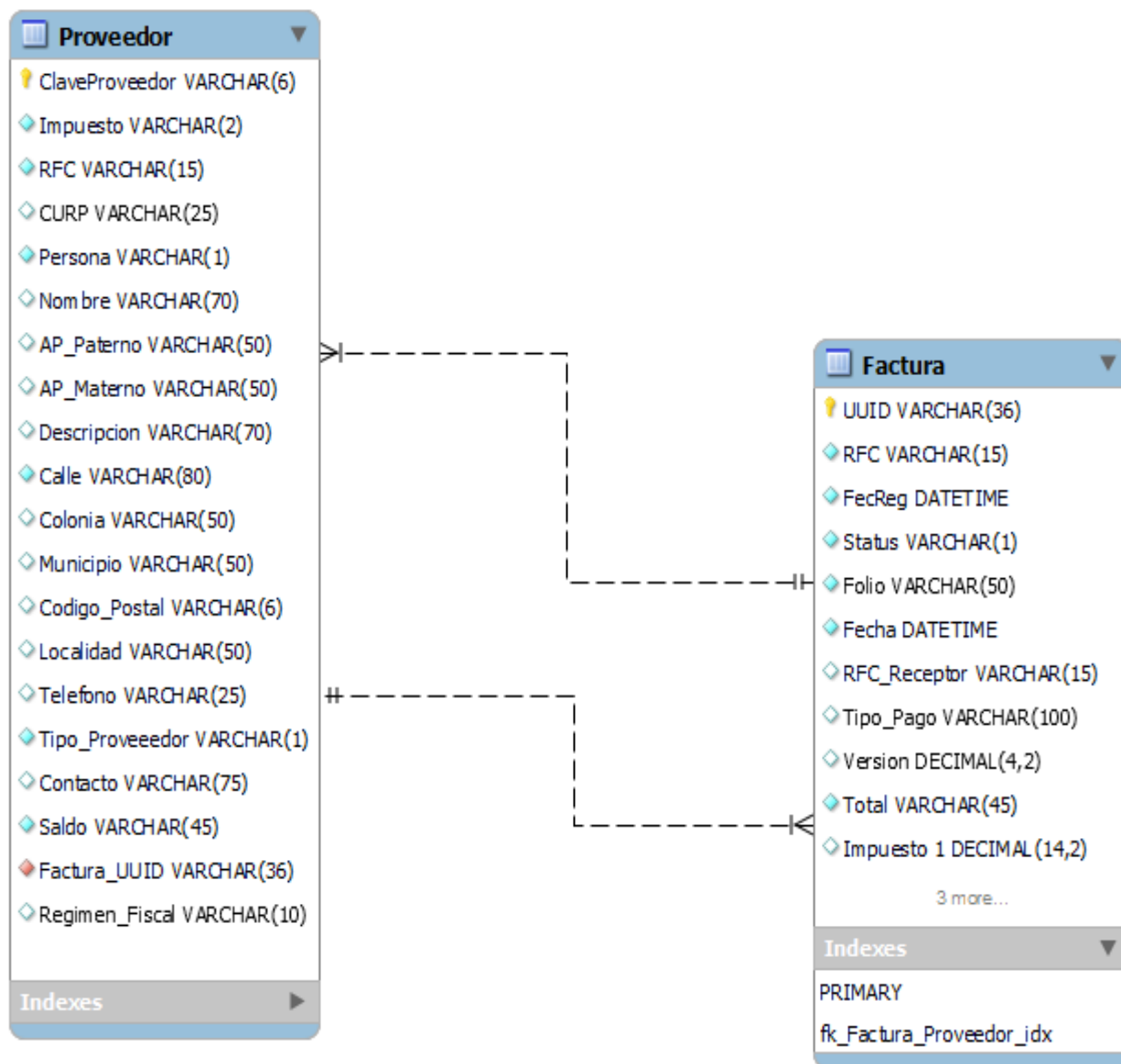


Figura 37 . Modelo Relacional

4.5.4 Diccionario de Datos

FACTURA							
PK	FK	NN	AI	CAMPO	TIPO	LONGITUD	DESCRIPCION
✓		✓	✓	UUID	VARCHAR	36	Identificador numérico del documento (Factura)
		✓		RFC	VARCHAR	15	RFC o razón social del Proveedor
		✓		FECHAREG	DATETIME		Fecha en que se registró el documento (Factura)
		✓		STATUS	VARCHAR	1	Status del documento (Factura)
		✓		FOLIO	VARCHAR	50	Folio de referencia del documento(factura)
		✓		FECHA	VARCHAR		Fecha de emisión del documento(factura)
				RFC_RECEPTOR	VARCHAR	15	RFC o razón social del Cliente o Receptor
				TIPO_PAGO	VARCHAR	100	Identificador numérico que designa el SAT de acuerdo al método de pago empleado por el cliente o receptor
				VERSION	DECIMAL	4,2	Identificador del SAT que determina la versión del comprobante de acuerdo a las características.
		✓		TOTAL	DECIMAL	45	Monto total que debe cubrir el cliente por diferentes conceptos.
				IMPUESTO 1	DECIMAL	14,2	Impuesto que debe cubrir el cliente por diferentes conceptos
				IMPUESTO 2	DECIMAL	14,2	Impuesto que debe cubrir el cliente por diferentes conceptos
				IMPUESTO 3	DECIMAL	14,2	Impuesto que debe cubrir el cliente por diferentes conceptos

Tabla 34. Diccionario de Datos Factura.

PROVEEDOR							
PK	FK	NN	AI	CAMPO	TIPO	LONGITUD	DESCRIPCION
✓		✓	✓	CLAVE_PROVEEDOR	VARCHAR	6	Identificador numérico del Proveedor.
		✓		IMPUESTO	VARCHAR	2	Impuesto que debe cubrir el proveedor que debe cubrir por la emisión de Factura.
		✓		RFC	VARCHAR	15	RFC o razón social del Proveedor
		□		CURP	VARCHAR	25	Identificador numérico del Proveedor.
		✓		PERSONA	VARCHAR	1	Identificador impuesto por el SAT, que si determina si el proveedor es (FISICO O MORAL)
		□		NOMBRE	VARCHAR	70	Nombre del proveedor.
				AP_PATERNO	VARCHAR	50	Apellido Paterno del proveedor.
				AP_MATERNO	VARCHAR	50	Apellido Materno del Proveedor.
				DESCRIPCION	VARCHAR	70	Descripción general del Proveedor
		✓		CALLE	VARCHAR	80	Calle donde reside el proveedor.
				COLONIA	VARCHAR	50	Colonia donde reside el proveedor
		✓		MUNICIPIO	VARCHAR	50	Municipio donde reside el proveedor
		□		CODIGO_POSTAL	VARCHAR	6	Código postal de la localidad donde vive el proveedor
		□		REGIMEN_FISCAL	VARCHAR	10	Régimen Fiscal al que está incorporado el proveedor.
		□		LOCALIDAD	VARCHAR	50	Localidad donde vive el proveedor.
		✓		TELEFONO	VARCHAR	25	Número Telefónico del Proveedor.
		✓		TIPO_PROVEEDOR	VARCHAR	1	Identificador del Tipo de Proveedor.
		□		CONTACTO	VARCHAR	75	Datos de contacto del proveedor
		✓		SALDO	VARCHAR	45	Monto del Saldo del proveedor.

Tabla 35. Diccionario de Datos Proveedor.

4.6 Diseño de la Interfaz Gráfica de Usuario

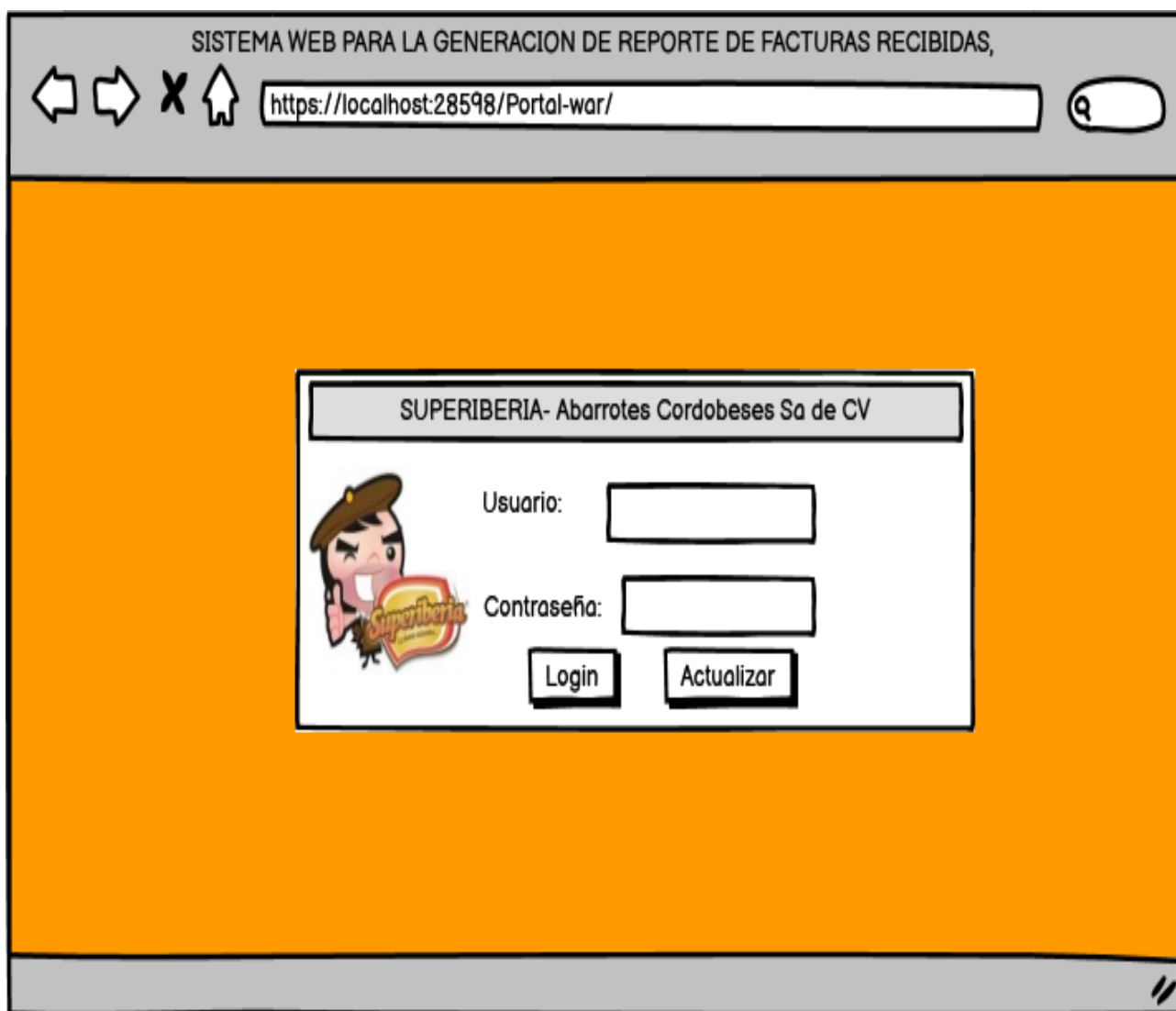


Figura 38. Prototipo de la GUI. Inicio de Sesión.



Figura 39. Prototipo de la GUI. Panel Principal.

SISTEMA WEB PARA LA GENERACION DE REPORTE DE FACTURAS RECIBIDAS,

https://localhost:28598/Portal-war/faces/consultafacturasvalidas.xhtml

 Portal Web - Abarrotes Cordobeses SA DE CV.

Consultas

Logout

Usuario Actual: 003602 Cdelgado Celestino Delgado Nogales

Consultar Facturas Validas

Proveedor:

Tipo de Documento: ☐ Todos ☐ Ingreso ☐ Egreso ☐ Traslado ☐ Pago

Fechas:  

Folio:

☐ Folio Fiscal *

UUID

Consultar

Version	fecha	Importe	Folio	UUID	Tipo de Documento	PDF	XML
---------	-------	---------	-------	------	-------------------	-----	-----

No se encontraron documentos

SUPERIBERIA-Abarrotes Cordobeses SA DE CV.

Figura 40. Prototipo de la GUI. Pantalla de consultar Factura.

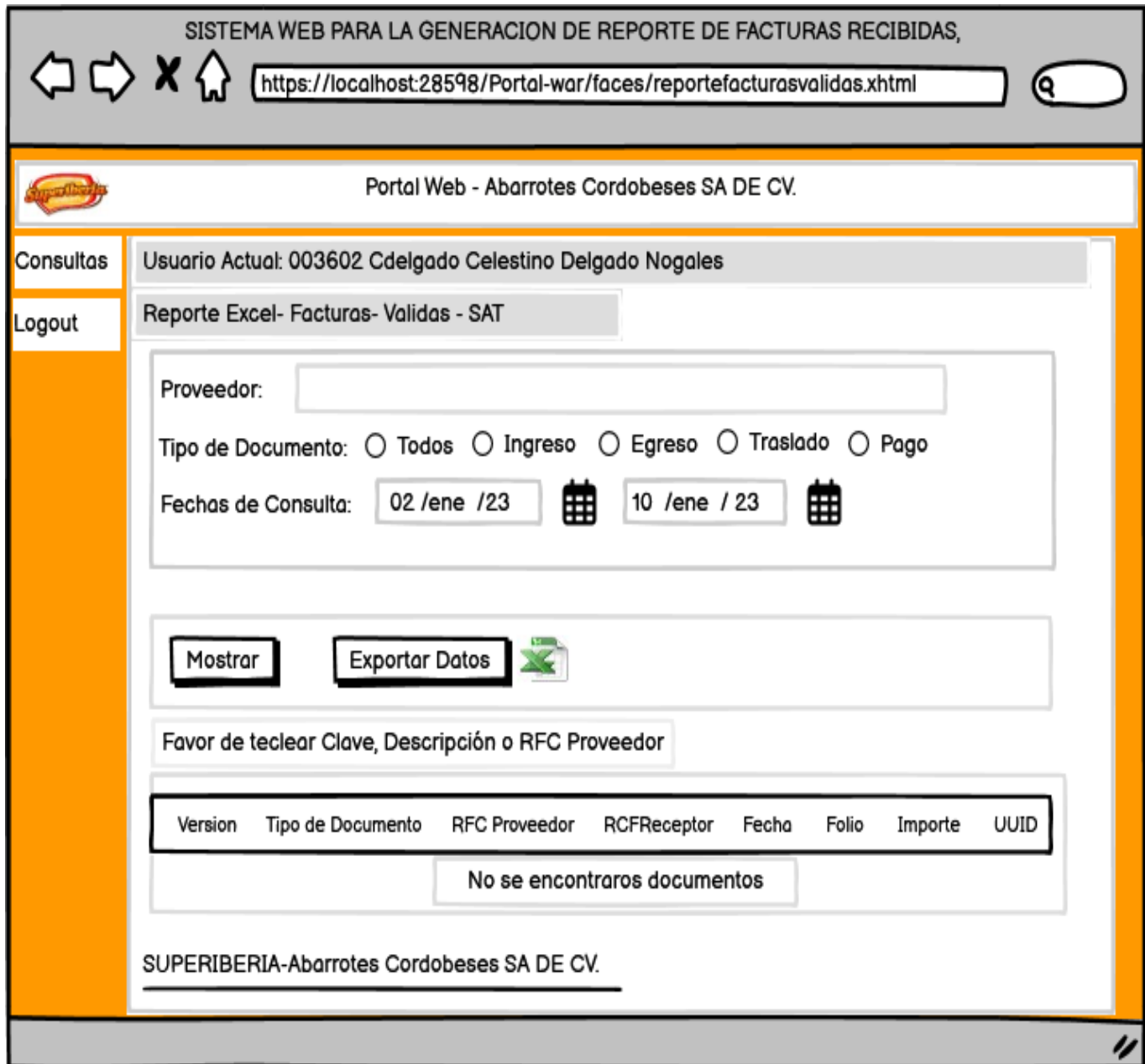


Figura 41. Prototipo de la GUI. Pantalla Generar Reporte de Facturas.

Capítulo 5 - Pruebas y Resultados

5.1 Descripción del Funcionamiento del Sistema

El sistema contiene un menú principal en el cual se tiene un inicio de sesión, solicita dos datos importantes el nombre de usuario y contraseña al ser correctos estos datos se ingresa correctamente al sistema principal en donde el administrador podrá realizar las operaciones o dará uso al sistema, además de que es quien tiene el acceso total del sistema y realiza las diferentes operaciones que se requieren para el área de conciliaciones de la Empresa Abarrotes Cordobeses.

Al ingresar al sistema se cuenta con un menú o panel principal en donde se pueden visualizar los diferentes módulos con los que cuenta el sistema, los módulos son los siguientes:

Módulo de Consulta de CFDI; es el módulo donde el usuario podrá, consultar, descargar y visualizar en formato PDF y XML cada una de las facturas que se tienen almacenadas en el sistema, además que el usuario podrá realizar la consulta de facturas por; proveedor, folio de documento, tipo de documento y por folio fiscal o UUID.

Módulo de Reportes de CFDI; es el módulo donde el usuario podrá consultar y visualizar cada uno de los comprobantes por proveedor además que, podrá realizar la exportación de un Reporte de documentos (Facturas) en formato Excel.

5.2 Pruebas del Sistema

Para poder brindarle a los usuarios de la Empresa Superlberia, un sistema de calidad se realizaron un conjunto de pruebas que enlisto a continuación:

- Se instaló correctamente el sistema en los equipos solicitado por el área administrativa.
- El usuario interactuó con el sistema.
- Se realizaron pruebas de validación de datos resultando exitosas las operaciones

El usuario realizó pruebas de todo el sistema en los diferentes módulos con los que cuenta, obteniendo un resultado exitoso al momento en que el usuario realizaba cada consulta de información.

5.2.1 Metodología para las Pruebas del Sistema

5.2.2 Pruebas Unitarias

Las pruebas unitarias son realizadas para verificar el correcto funcionamiento del sistema, Para esto es necesario revisar el diseño, codificar, y revisar el código de forma continua.

Las pruebas unitarias serán la validación de que se ha construido un software de calidad, detectando y corrigiendo durante las mismas.

Utilizamos la herramienta “JUnit”, que es un conjunto de bibliotecas creadas por Erich Gamma y Kent Beck que son utilizadas en programación para hacer pruebas unitarias de aplicaciones Java.

El framework provee al usuario de herramientas, clases y métodos que le facilitan la tarea de realizar pruebas en su sistema y así asegurar su consistencia y funcionalidad.

A continuación, se observa la realización de dicha prueba en este caso la se realizó a nuestra clase principal, por medio de la ejecución de clase ReporteFacturasValidasPantalla, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase, se comporta como se espera.

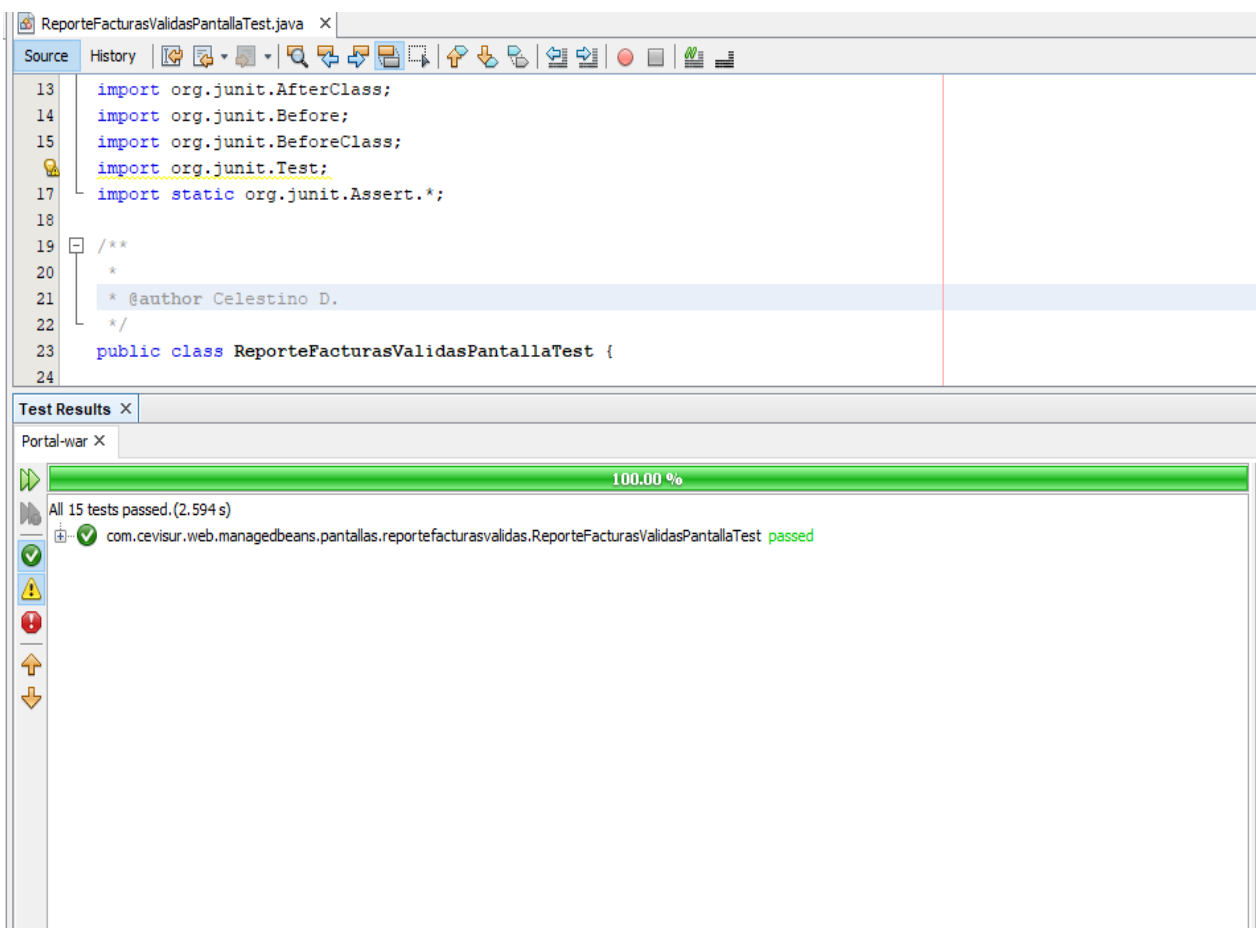


Figura 42 . Prueba Unitaria. ReporteFacturasValidasPantalla.

5.2.3 Observaciones Durante las Pruebas del Sistema

En el proceso de pruebas que se han llevado a cabo al sistema, todo se ha comportado de manera estable, los datos ingresados y consultas realizadas se han registrado correctamente en la base de datos, obteniendo una correcta conexión de información con el sistema de la misma manera se realizaron las pruebas generales sobre los diferentes módulos que contiene el sistema:

- En el módulo de CFDI, se realizó la consulta de facturas por descripción de proveedor, por tipo de documento, por folio de identificación de factura, folio Fiscal (UUID), ETC. Además, en este módulo los usuarios podrán descargar sus facturas en formato PDF Y XML.
- En el módulo de reportes, se realizó la consulta de facturas por descripción de proveedor, por tipo de documento y se exporto correctamente la información a documento de tipo Excel.

Se realizaron estas tareas en base a los módulos, en cuanto al administrador mostró una completa satisfacción por el desempeño del sistema ya que beneficio al departamento de Conciliaciones haciendo más eficaz y productiva su operación diaria de revisión y control diario de la documentación e información.

Al inicio se generaron algunas dificultades en cuanto a la validación de datos del proveedor, el cual no estaba pasando el dato del RFC, posteriormente con el tiempo se corrigió el problema que están generando el conflicto en la base de datos.

A pesar de estas observaciones que fueron notificadas por el usuario, el sistema cumple con las funcionalidades, para las que fue programado cumpliendo así con el objetivo principal.

5.3 Implementación del Sistema

Se llevó a cabo la implementación del sistema, dando acceso al sistema al departamento de Conciliaciones

Utilizando un equipo (Laptop) LENOVO con procesador AMD 3020e with Radeon Graphics 1.20 GHz Sistema operativo de 64 bits, procesador x64 Windows 10 HOME en el cual se realizaron las pruebas necesarias del sistema para ver su correcto funcionamiento.

Para comprobar el correcto funcionamiento del sistema en el equipo, se visualiza la ventana del log del proyecto, que da respuesta de que el sistema web está ejecutándose como se esperaba.

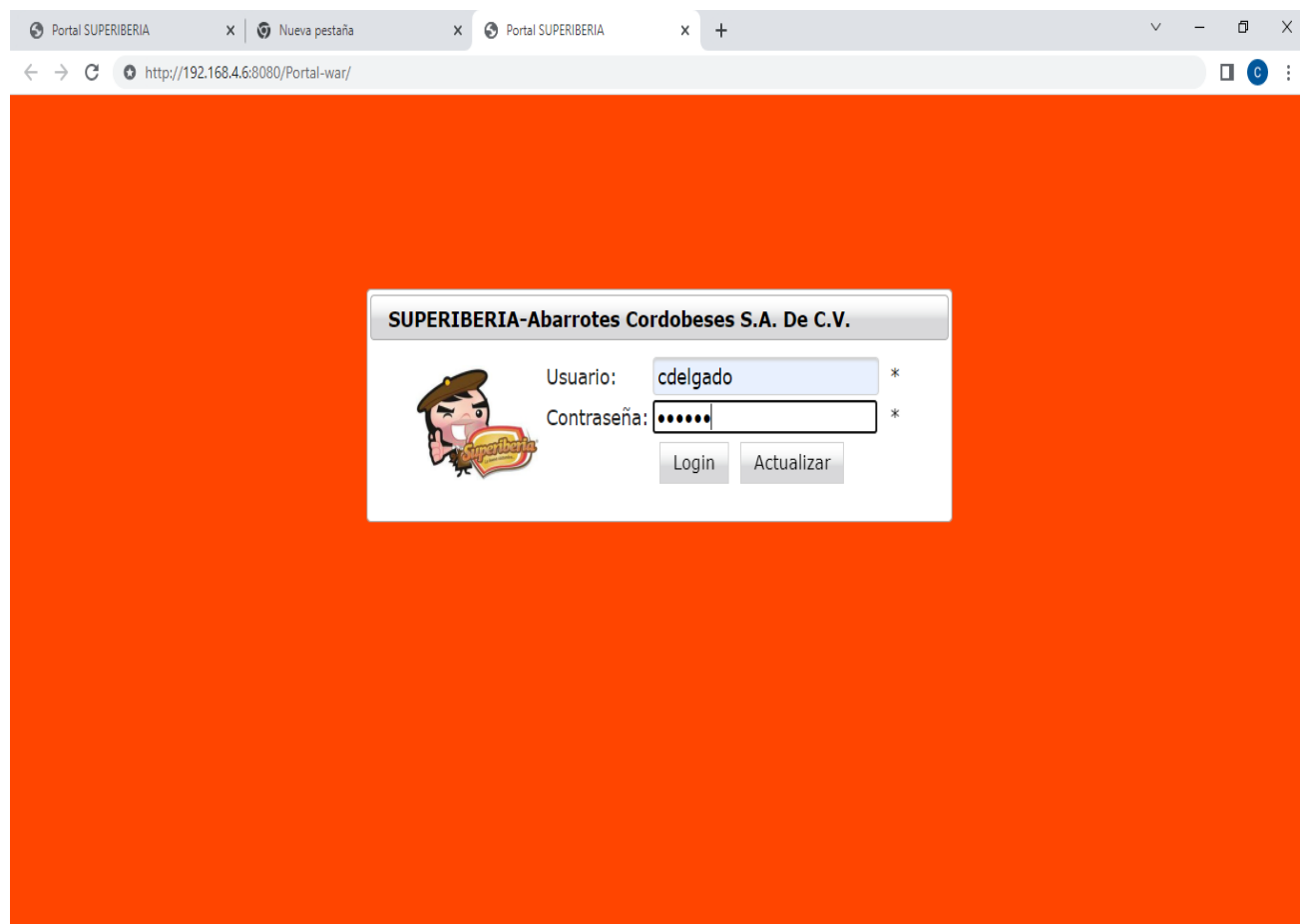


Figura 43 . Ventana de Inicio de Sesión

Consulta Facturas Validas

Consulta Facturas Validas

+

←

→

↻

http://192.168.4.6:8080/Portal-war/faces/consultafacturasvalidas.xhtml

≡

□

Ⓢ

⋮


Abarrotes Cordobeses S.A. De C.V.

Consultas ▾

Logout ▾

Usuario Actual : 003602 cdelgado Celestino Delgado Nogales

Consultar Facturas Validas

Proveedor:

0PD092_DANONE DE MEXICO, S.A. DE C.V._DME-761202-CP9

Tipo de Documento:

☒ Todos
 ☐ Ingreso
 ☐ Egreso
 ☐ Traslado
 ☐ Pago

Fechas:

13/ene/2023

18/ene/2023

Folio:

97070

☐ Folio Fiscal *

UUID:

Consultar

Proveedor : DANONE DE MEXICO, S.A. DE C.V.

Versión	Fecha	Importe	Folio	UUID	Tipo de Documento	PDF	XML
3.30	14-ene-2023	\$231.91	00097070	D16C7E2C-9456-11ED-8E13-092CD7795AB1	EGRESO	Descargar	Descargar
3.30	13-ene-2023	\$7,301.88	00199501	5FE1F0AF-93C0-11ED-B472-C1998B0E8FEA	INGRESO	Descargar	Descargar
3.30	13-ene-2023	\$6,040.70	00199486	58224C3A-93C0-11ED-AF1D-6FDE79F2136D	INGRESO	Descargar	Descargar

SUPERIBERIA-Abarrotes Cordobeses S.A. De C.V.

Figura 44. Ventana de CFDI-Consulta de Facturas.

130 / 163

Consulta Facturas Validas

Reporte - Facturas - Validas

+

192.168.4.6:8080/Portal-war/faces/reportefacturasvalidas.xhtml


Abarrotes Cordobeses S.A. De C.V.

Consultas

Logout

CFDI

Reporte Facturas

Reporte EXCEL - Facturas - Validas- SAT

Proveedor:

OPD092_DANONE DE MEXICO, S.A. DE C.V._DME-761202-CP9

Tipo de Documento:

☒ Todos
 ☐ Ingreso
 ☐ Egreso
 ☐ Traslado
 ☐ Pago

Fechas de Consulta:

13/ene/2023

18/ene/2023

Mostrar

Exportar Datos 

Datos Proveedor :DANONE DE MEXICO, S.A. DE C.V._REGIMEN FISCAL_601

VERSIÓN	TIPO DOCUMENTO	RFC PROVEEDOR	RFC RECEPTOR	FECHA	FOLIO	IMPORTE	UUID
3.30	EGRESO	DME761202CP9	ACO6405017L0	14-ene-2023	00097070	\$231.91	D16C7E2C-9456-11ED-8E13-092CD7795AB1
3.30	INGRESO	DME761202CP9	ACO6405017L0	13-ene-2023	00199501	\$7,301.88	5FE1F0AF-93C0-11ED-B472-C1998B0E8FEA
3.30	INGRESO	DME761202CP9	ACO6405017L0	13-ene-2023	00199486	\$6,040.70	58224C3A-93C0-11ED-AF1D-6FDE79F2136D

SUPERIBERIA-Abarrotes Cordobeses S.A. De C.V.

localhost:28598/Portal-war/faces/reportefacturasvalidas.xhtml

Figura 45. Ventana de Reporte-Facturas.

5.4 Manual del Usuario

Anexo 2-Manual de usuario

5.5 Mejora Continua

La mejora continua se logra a partir de lo que ya se planteó anteriormente, considerando las funcionalidades que realiza actualmente el sistema y de las herramientas que emplea en cada tarea o consulta que lleva a cabo.

- 1.-Poder realizar la validación de comprobantes fiscales, STATUS (Vigente o cancelado); cuando el proveedor en un momento dado haya realizado la cancelación del comprobante fiscal. El sistema pueda mostrar el status del documento V, para mostrar vigente y/o C para cancelado.
- 2.-Contar con la impresión de reportes: Conectar la impresora al sistema para que, mediante una conexión de red, al momento de que el usuario de clic en el botón de reportes, pueda imprimir directamente la información.
- 3.-Plantear nuevamente la interfaz: Esto con la finalidad de que los usuarios de los demás departamentos de la Empresa, puedan conocer, interactuar y entender con mayor facilidad el uso y funcionalidad del sistema, para un correcto entendimiento de las principales funciones que realiza el sistema.
- 4.-Llevar a cabo la validación de comprobantes de tipo ingresos- Facturas con versión del CFDI 4.0, ya que actualmente el sistema solo esta programado para realizar la validación de CFDI'S con versión 3. 0..

Capítulo 6 - Conclusiones

6.1 Competencias Desarrolladas o Aplicadas

- ❖ **Taller de base de datos:** Construye esquemas de base de datos para proyectar las necesidades de un cliente en base a las reglas sintácticas del lenguaje de definición de datos.
-Esta asignatura me aportó un amplio conocimiento, para la implementación de bases de datos, así como el diseño, modelado y manejo de SQL.
- ❖ **Programación Orientada a Objetos:** Comprende y aplica los conceptos del paradigma de programación orientada a objetos para modelar situaciones de la vida real.
-Me aportó un amplio conocimiento para la creación de aplicaciones basadas en clases y objetos así también en la reutilización de código.
- ❖ **Estructura de datos:** Conoce y comprende las diferentes estructuras de datos, su clasificación y forma de manipularlas para buscar la manera más eficiente de resolver problemas.

-Esta asignatura me aportó gran apoyo y conocimiento para poder estructurar y organizar datos haciéndolos más eficientes.
- ❖ **Fundamentos de base de datos:** Conoce y comprende los conceptos básicos de base de datos para proponer soluciones en el tratamiento de información.
-Esta asignatura me aportó el conocimiento para poder analizar, diseñar bases de datos para poder garantizar el manejo de información de un sistema.
- ❖ **Fundamentos de Ingeniería de software:** Identifica y comprende el entorno de aplicación de la ingeniería de software.

-Esta asignatura me aporó el conocimiento para poder crear interfaces, de la misma manera las bases, técnicas y uso de metodologías para la elaboración de un proyecto de software.

- ❖ **Administración de base de datos:** Comprende las actividades de la administración de bases de datos, identifica las funciones del DBA y las características de los diferentes SGBD con el fin de dimensionar su importancia en las organizaciones.

-Esta asignatura me aporó la competitividad para crear grandes bases de datos, con mayor seguridad de la información usando Sistemas Gestores de Base de datos.

- ❖ **Fundamentos de programación:** Comprende y aplica los conceptos básicos, nomenclatura y herramientas para el diseño de algoritmos orientado a la resolución de problemas.

-Esta asignatura me aporó conocimientos principales para el modelado y desarrollo de problemas computarizados.

- ❖ **Tópicos avanzados de programación:** Desarrolla programas para interactuar con el usuario de una manera amigable, utilizando GUI (Interfaz Gráfica de Usuario) manipuladas a través de eventos.

-Esta asignatura me aporó las herramientas necesarias para crear interfaces amigables para el usuario, creación de eventos etc....y por su puesto teniendo siempre presente que la interfaz de usuario determina la usabilidad de nuestra aplicación.

6.2 Conclusiones

La falta de personal y tiempo, entre otros factores, ha impedido que la información sea almacenada y distribuida de forma que todos los departamentos de la Empresa puedan fácilmente tener acceso a ella.

Con la implementación de este Sistema Web, permitirá a la Empresa tener mayor contacto con las áreas de Cuentas por Cobrar de los proveedores, permitirá almacenar y visualizar todos los documentos que emiten los proveedores y de la misma manera facilitará a los usuarios de conciliaciones mejorar la revisión de los documentos.

Además, con la implementación de este sistema se tuvo como resultado la disminución en el tiempo de revisión y consulta de la información, se agilizó el proceso de generación de reportes que se requerían para el proceso de revisión de estados de cuenta

También se podrá cumplir en tiempo y forma el pago a proveedores de la misma manera se logró disminuir el riesgo de pérdida de información dando un buen manejo de información al usuario.

Y, por último, llevar a cabo un análisis detallado como el que se realizó en este proyecto incrementa en gran proporción las probabilidades de tener éxito ya que de ante mano se conoce lo que se quiere lograr y cómo se va a hacer para lograrlo.

6.3 Recomendaciones

Para hacer uso del sistema el usuario llevara a cabo las siguientes recomendaciones:

- Manual de usuario: Se recomienda leer el manual de usuario la primera vez a utilizar el sistema ya que contiene información detallada de cada uno de los módulos con lo que cuenta el sistema y especifica correctamente cada una de sus funcionalidades, si es necesario el usuario podrá consultar el manual de usuario cada vez que lo requiera.
- Llenar el formulario correctamente: Cuando el usuario requiera hacer uso de los módulos, deberá llenar el formulario con los datos de manera completa.
- Ingresar la información necesaria en el formulario de reportes: Se le recomienda al usuario incluir toda la información o llenar los campos correctamente de acuerdo al tipo de dato que requiere, ya que, al no deshabilitar los botones de selección, el sistema automáticamente exportara toda la información que tiene registrada en la tabla.
- Equipo de trabajo: Se le recomienda al usuario montar el sistema en un equipo que cuente con las especificaciones mínimas propuestas por el sistema.

Bibliografía

1. A., R., & Beichner, R. J. (s.f.). *Física para Ciencias e Ingeniería* (Quinta ed., Vol. II). McGraw-Hill.
2. Acosta, T. &.-M. (2017). Análisis de la accesibilidad de los sitios web de las universidades ecuatorianas de excelencia. *Enfoque UTE*, 46-61.
3. Aguirre, S. (2020). *PHP-Programación Orientada a Objetos–Vol 1 Clases, objetos y métodos (Vol. 1)*.
4. Almeida Moyano, W. R. (2009). *Benchmark para el uso de tecnologías relacionadas a servidor de aplicaciones Glassfish y Jboss*.
5. Arjona, J. L. (2004). *Notas de Introducción al Lenguaje de Programación Java*.
6. Aumaille, B. (2002). *2EE: Desarrollo de aplicaciones Web. Ediciones ENI*.
7. Aumaille, B. (2002). *Desarrollo de aplicaciones Web. Ediciones ENI*.
8. Aumaille, B. (2002). *Desarrollo de aplicaciones Web. Ediciones ENI*.
9. Benlliure, J. L. (1995). Una mirada al World Wide Web. *Métodos de información*, 35-40.
10. Cabello, A. L. (2015). *Implantación de aplicaciones web en entornos internet, intranet y extranet. IFCD0210*. Ic Editorial.
11. Cabello, A. L. (2015). *Implantación de aplicaciones web en entornos internet, intranet y extranet. IFCD0210*. Ic Editorial.
12. Cadavid, A. N. (2013). Revisión de metodologías ágiles para el desarrollo de software. *Prospectiva*, 30-39.
13. Canós, J. H. (2003). Metodologías ágiles en el desarrollo de software. *Universidad Politécnica de Valencia, Valencia*, 1-8.

14. Castellanos, J. G. (2003). Pymes innovadoras. Cambio de estrategias e instrumentos. *Revista escuela de Administración de Negocios*, (47), 10-33.
15. Chaves, M. A. (2005). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *InterSedes: Revista de las Sedes Regionales*, 6(10), 1-13.
16. Deitel, H. M. (2003). *Cómo programar en Java*. Pearson educación.
17. Domínguez Domínguez, J. M. (2018). Prototipo de un sistema de registro escolar. (Tesis doctoral). Universidad Autónoma de Chihuahua, Chihuahua..
18. Duarte, A. O. (2008). Las metodologías de desarrollo ágil como una oportunidad para la ingeniería del software educativo. . *Revista Avances en Sistemas e Informática*, 159-171.
19. Duarte, A. O. (2008). Las metodologías de desarrollo ágil como una oportunidad para la ingeniería del software educativo. . *Revista Avances en Sistemas e Informática*, 159-171.
20. Escofet, C. M. (2002). *El lenguaje SQL*. UOC, la universidad virtual.
21. Estayno, M. G. (2014). El desafío de ser un Product Owner. *In XX Congreso Argentino de Ciencias de la Computación*.
22. Fernandez Ardiles, J. E. (2022). Gestión por procesos y ventaja competitiva en la COOPAC Grupo Orienta, Arequipa.
23. Figueroa, R. G. (2008). Metodologías tradicionales vs. metodologías ágiles. Universidad Técnica Particular de Loja. *Escuela de Ciencias de la Computación*, 9(1), 1-10.
24. Forero Cano, M. P. (2011). Solución de integración entre bundles OSGi a través de la tecnología Web Services.
25. Franky, M. C. ((2010, April)). *Módulo de seguridad basado en roles finos y en perfiles de seguridad para aplicaciones Java EE 5*. Cartagena-Colombia.: In Quinto Congreso Colombiano de Computación (5CCC).

26. Gabillaud, J. (2015). *SQL Server 2014: SQL, Transact SQL, diseño y creación de una base de datos (con ejercicios prácticos corregidos)*. Ediciones ENI.
27. García de Figuerola Paniagua, L. C. (2009). Un motor de búsqueda para la investigación experimental en Recuperación de la Información mediante SQL.
28. García, A. (2003). *Manual práctico de sql*. OMEGA.
29. Godoy, D. A. (2015). *Diseño de un Simulador Dinámico de Proyectos de Desarrollo de Software que utilizan metodología Scrum*.
30. Goncalves, A. (2009). *Beginning Java EE 6 Platform with GlassFish 3: from novice to professional*. Apress.
31. González Menéndez, J. A. (2022). Utilización de las bases de datos relacionales en el sistema de gestión y almacenamiento de datos. Editorial Paraninfo. González, A. J. (2007). *Ingeniería de software: Metodologías*.
32. Grande, J. I. (2001). Definiendo la e-Administración: las páginas web de las Administraciones públicas. *Gestión y Análisis de Políticas Públicas*, 123-138.
33. Groussard, T. (2010). *Java enterprise edition: desarrollo de aplicaciones web con JEE 6*. Ediciones Eni.
34. Guerrero-Luzuriaga, A. M.-G.-J. (2018). "ERP como alternativa de eficiencia en la gestión financiera de empresas." *Revista Lasallista de Investigación*, 13(2), 182-193..
35. Heffelfinger, D. (2014). *Java EE 7 con servidor de aplicaciones GlassFish 4*. Birmingham, Bombay: Packt Publishing.
36. Heffelfinger, D. R. (2014). *Java EE 7 with GlassFish 4 Application Server*. Birmingham, Mumbai: Packt Publishing.

- 37.JAVIER, D. M. (2007). *Programación orientada a objetos con Java*. Paraninfo.
- 38.Jiménez, C. &. (s.f.). *El rol del lenguaje SQL en los SGBDR y en la implementación del Modelo Relacional*.
- 39.Lahiguera Hervás, A. (2021). "Desarrollo de material didáctico para el aprendizaje de Spring" (Tesis doctoral, Universitat Politècnica de València).
- 40.Larrocha, E. R. (2017). *Nuevas tendencias en los sistemas de información*. Centro de Estudios Ramon Areces SA.
- 41.Letelier Torres, P. (julio de 2015). "Una actividad para enseñar el uso de tableros Kanban y diagramas de flujo acumulado." En Actas de las XXI Jornadas sobre la Enseñanza Universitaria de la Informática (pp. 288-295).
- 42.Letelier, P. (2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).
- 43.López, D. &. (2017). *Arquitectura de Software basada en Microservicios para Desarrollo de Aplicaciones Web*.
- 44.López, J. G. (s.f.). *Administración de sistemas GNU/LINUX®*. Ediciones de la U.
- 45.Luján-Mora, S. (s.f.). "Programación de aplicaciones web: historia, principios básicos y clientes web." Editorial Club Universitario.
- 46.Maida, E. G. (2015). *Metodologías de desarrollo de software*.
- 47.Marañón Martín, J. M. (2014). "Implementación de una plataforma de servidores de aplicaciones."
- 48.Marín, A. L. (2006). Las tecnologías de la información y la comunicación: Cambios en el modelo de comunicación. *Hologramática*,, 15-33.
- 49.Mariño, S. I. (2014). Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación. *Scientia et technica*,, 413-418.

50. Mazzarella, C. (2018). Desarrollo de habilidades metacognitivas con el uso de las TIC. *Investigación y postgrado*, 175-204.
51. Meaurio, V. S. (2013). La arquitectura de software en el proceso de desarrollo,. *Revista Latinoamericana de Ingeniería de Software*, , 142-146.
52. Mendoza González, G. (2015). "Herramienta de Desarrollo NetBeans." *Herramienta de Desarrollo NetBeans*, 1.
53. Mestras, J. P. (1997). El lenguaje Java en la asignatura de Programación Concurrente. *Actas de las III Jornadas de Enseñanza Universitaria de la Informática*,, 455-462.
54. Moreira Zambrano, C. A.-R.-Q. (2017). "Soluciones de bajo costo usando el modelo infraestructura como servicio con alta disponibilidad y virtualización." *Enfoque UTE*, 8(17), 186-200.
55. Moreno, F. J. (2005). Desempeño de consultas SQL relacionales y objeto-relacionales en Oracle. *Ingeniería e Investigación*, 4-12.
56. Moreno, F. J. (2014). Una propuesta para la clasificación de la programación reflexiva orientada al desarrollo de sistemas autónomos. *Ingeniería y competitividad*,, 91-104.
57. Muñoz Muñoz, J. G. (2015). "Estudio comparativo de capacidad transaccional entre servidores de aplicaciones web." (Tesis doctoral, Universidad de Guayaquil, Facultad de Ciencias Matemáticas y Físicas, Carrera de Ingeniería en Sistemas Computacionales).
58. Muñoz, G. A. (2021). La Competitividad como factor de crecimiento para las organizaciones. *INNOVA Research Journal*, 6(1),, 145-161.
59. Niño, J. (2011). *Servidores de aplicaciones web (Aplicaciones web)*. Editex.
60. Ortégón, E. P. (2005). *Metodología del marco lógico para la planificación, el seguimiento y la evaluación de proyectos y programas*. Cepal.
61. Pacienza, J. &. (2015). *Metodologías de desarrollo de software*.

62. Palacios, D. G. (2018). Análisis del rendimiento de librerías de componentes Java Server Faces en el desarrollo de aplicaciones web. *Revista Digital Novasinergia*, 1(2), 54-59.
63. Palma Pérez, N. (2019). "Solución informática para la selección de Apache 2 y Nginx durante la migración a código abierto." (Tesis de maestría, Universidad de las Ciencias Informáticas, Facultad 1, Centro de Software Libre).
64. Pérez Montoya, L. M. (2008). "Una comparación de dos lenguajes de consulta espacio-temporales: SQL ST y el de Güting." *Ingeniería e Investigación*, 28(1), 138-144.
65. Ramírez, M. R. (2019). Metodología SCRUM y desarrollo de Repositorio Digital. *Revista Ibérica De Sistemas e Tecnologías De Informação*, , 1062-1072.
66. Reinos, E. M. (2012). *Bases de datos*. Alpha Editorial.
67. Rodríguez, C. &. (2015). ¿ Por qué implementar Scrum?. *Revista Ontare*,, 125-144.
68. Rojas, E. F. (2016). *Estado de la banda ancha en América Latina y el Caribe 2016*.
69. Romero Martínez, M. (1999). "Lenguaje de Consultas para una Multibase de Datos." Repositorio Nacional CONACYT.
70. Ruiz, I. L.-N. (2002). *Bases de Datos Desde Chen Has Codd Con Oracle*. Alfaomega Grupo Editor.
71. Serra Manchado, D. (2013). "Estudio del servidor de aplicaciones Glassfish y de las aplicaciones J2EE."
72. Tabarés Gutiérrez, R. (2012). "El inicio de la Web: historia y cronología del hipertexto hasta HTML 4.0", 57-82.

73. Torres Muñoz, G., & Klenner Meixner, A. (2014). "Implementación de Sistemas ERP en PYMES a nivel nacional como herramienta para el control de gestión y toma de decisiones en procesos de negocio". Tesis doctoral.
74. Torres Muñoz, P. (2014). "Implementación de Sistemas ERP en PYMES a nivel nacional como herramienta para el control de gestión y toma de decisiones en procesos de negocio" (Tesis doctoral, Universidad Academia de Humanismo Cristiano).
75. Trasobares, A. H. (2003). Los sistemas de información: evolución y desarrollo. *Revista de relaciones laborales*, (10), 149-165.
76. Úbeda González, R. (2009). "Métodos Ágiles para el Desarrollo de Software".
77. Vargas, B. &. (2008). Competitividad sostenible de la pequeña empresa: un modelo de promoción de capacidades endógenas para promover ventajas competitivas sostenibles y alta productividad. *Journal of Economics, Finance and Administrative Science*, , 13(24), 59-80.
78. Vasiliev, Y. (2008). Getting Started with GlassFish. Beginning Database-Driven Application Development in Java™. *Using GlassFish™*, 27-48.
79. Vegas, J. (1998). "Introducción al SQL." Recuperado de <http://www.infor.uva.es/~jvegas/cursos/bd/sqlplus/sqlplus.html#vistas>.
80. García López, M. (2015). "Desarrollo de Aplicaciones Móviles: Principios y Prácticas." Tesis de maestría, Universidad Tecnológica del Sur.
81. Martínez Ruiz, A., & Sánchez Jiménez, R. (2020). "Big Data y Analítica Predictiva en la Gestión Empresarial." Recuperado de <http://www.empresauni.edu/biblioteca/bigdata/analytics.html>

Índice de Términos

XHTML: Siglas del inglés eXtensible HyperText Markup Language. XHTML es básicamente HTML expresado como XML válido. Es más estricto a nivel técnico, pero esto permite que posteriormente sea más fácil al hacer cambios, buscar errores, etc.

Java: es un lenguaje de programación ampliamente utilizado para codificar aplicaciones web. Ha sido una opción popular entre los desarrolladores durante más de dos décadas, con millones de aplicaciones Java en uso en la actualidad. Java es un lenguaje multiplataforma, orientado a objetos y centrado en la red que se puede utilizar como una plataforma en sí mismo.

Es un lenguaje de programación rápido, seguro y confiable para codificarlo todo, desde aplicaciones móviles y software empresarial hasta aplicaciones de macrodatos y tecnologías del servidor.

Programación orientada a objetos: Técnica de programación que aumenta la velocidad de desarrollo de los programas y hace que su mantenimiento sea más fácil al volver a utilizar "objetos" que tienen comportamientos, características y relaciones asociadas con el programa.

Los objetos son organizados en grupos que están disponibles para la creación y mantenimiento de aplicaciones.

Diseño: Es una actividad que consiste en la planificación, diseño e implementación de sitios web y páginas web. No es simplemente una aplicación del diseño convencional, ya que requiere tener en cuenta cuestiones tales como navegabilidad, interactividad, usabilidad, arquitectura de la información y la interacción de medios como el audio, texto, imagen y vídeo.

Requerimientos: Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal.

Formulario: Casillas que se puede añadir a una ventana y que permite recoger datos que introduce el usuario para luego procesarlos.

SQL: Lenguaje de programación para el manejo de bases de datos.

Base de datos: Es un “almacén” que permite guardar cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.

Datos almacenados en una computadora que pueden ser accedidos y manipulados fácilmente; piense en un gran gabinete físico con muchas etiquetas con archivos de información relevante.

Software: Es un conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en una computadora con un fin en común.

Sistema Operativo. Es el software básico de una computadora que provee una interfaz entre el resto de programas del computador.

Interfaz gráfica de usuario GUI: Conocida también como GUI es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

Módulos: Es una porción de un programa de ordenador. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará, comúnmente, una de dichas tareas (o varias, en algún caso).

Portal: Punto de entrada en la Web que ofrece determinados servicios para operar en internet.

Servidor: Es el computador en el que se ejecutan los programas que realizan alguna tarea en beneficio de otras computadoras.

Anexos

Anexo 1 - Manual del Usuario



ABARROTES CORDOBESES SA DE CV
SuperIberia



Manual de Usuario Sistema Web S.I- A.C.O.

Versión 1.0



Alcance

En el presente documento se describen cada una de las funcionalidades correspondientes del Sistema - Validador - Facturas, en donde se contemplan cada uno de los módulos como lo son: módulo de CFDI, módulo de Reportes. Implementado para la Empresa de Abarrotes Cordobeses SA DE CV.



Índice de contenido

1	. Ingreso al Sistema.	149
2	. Menú Principal	151
3	. Modulo de Consulta de CFDI.	152
3.1	Consulta por Proveedor.....	153
3.2	Consulta por Folio de Factura	154
3.3	Consulta por Folio Fiscal-UUID.....	155
3.4	Descarga de CFDI.....	156
4	. Módulo de Reportes.	158
4.1	Consulta de Reporte	159
4.2	Exportar Reporte	160
5	. Cerrar Sesión.	162

1. Ingreso al Sistema.

Para acceder al sistema, el usuario debe ingresar a la siguiente dirección <http://192.168.4.6:8080/Portal> , en este se observará una pantalla en donde el usuario debe ingresar sus credenciales de acceso (usuario y contraseña), que el administrador debió entregarle cuando ingreso a la empresa.

← → ↻ http://192.168.4.6:8080/Portal-war/

SUPERIBERIA-Abarrotes Cordobeses S.A. De C.V.



Usuario: *

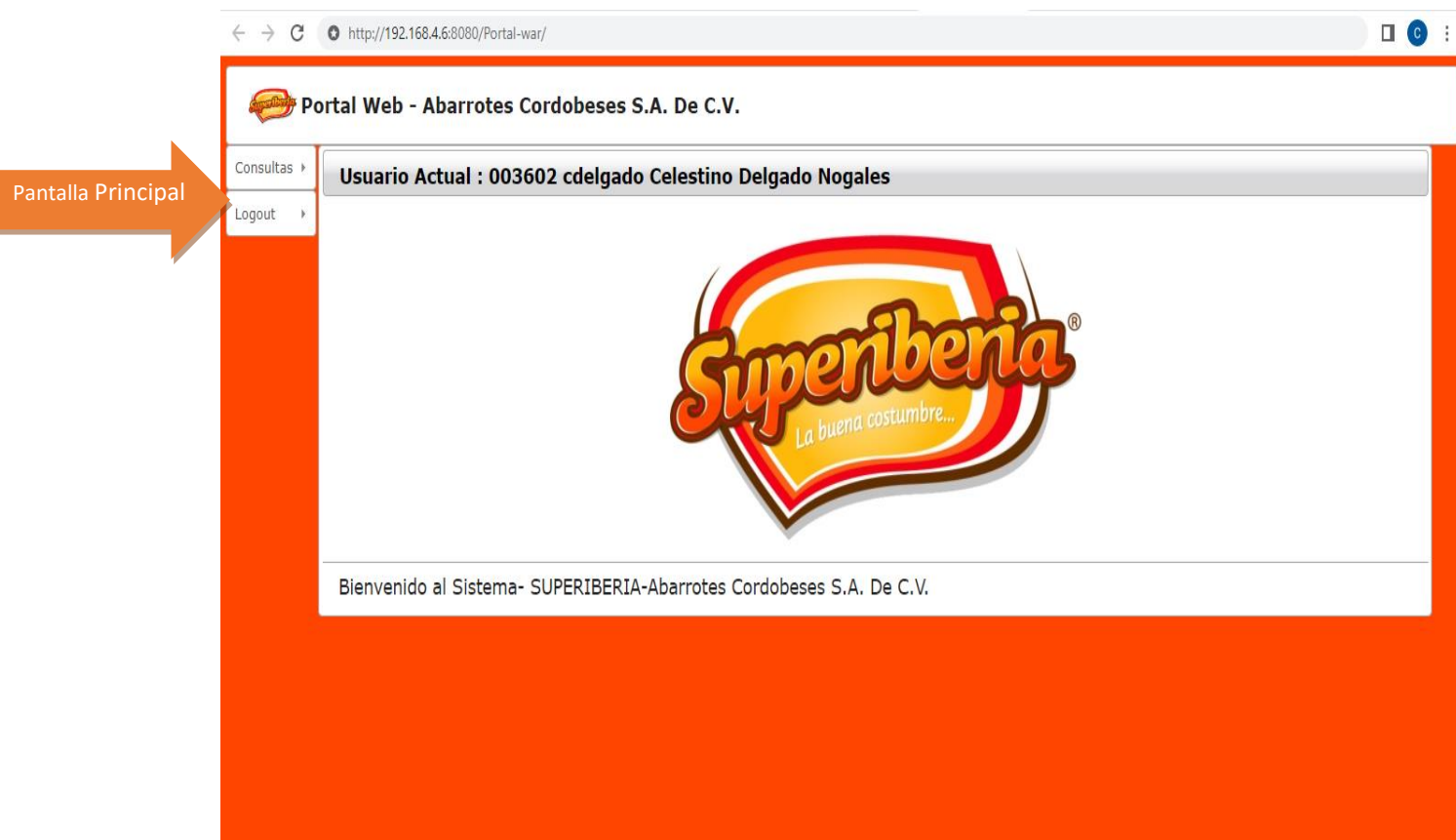
Contraseña: *

Ingreso al Sistema

Ingresar: Una vez introducido el usuario y la contraseña, se debe de dar clic en el botón Log para acceder al menú principal.

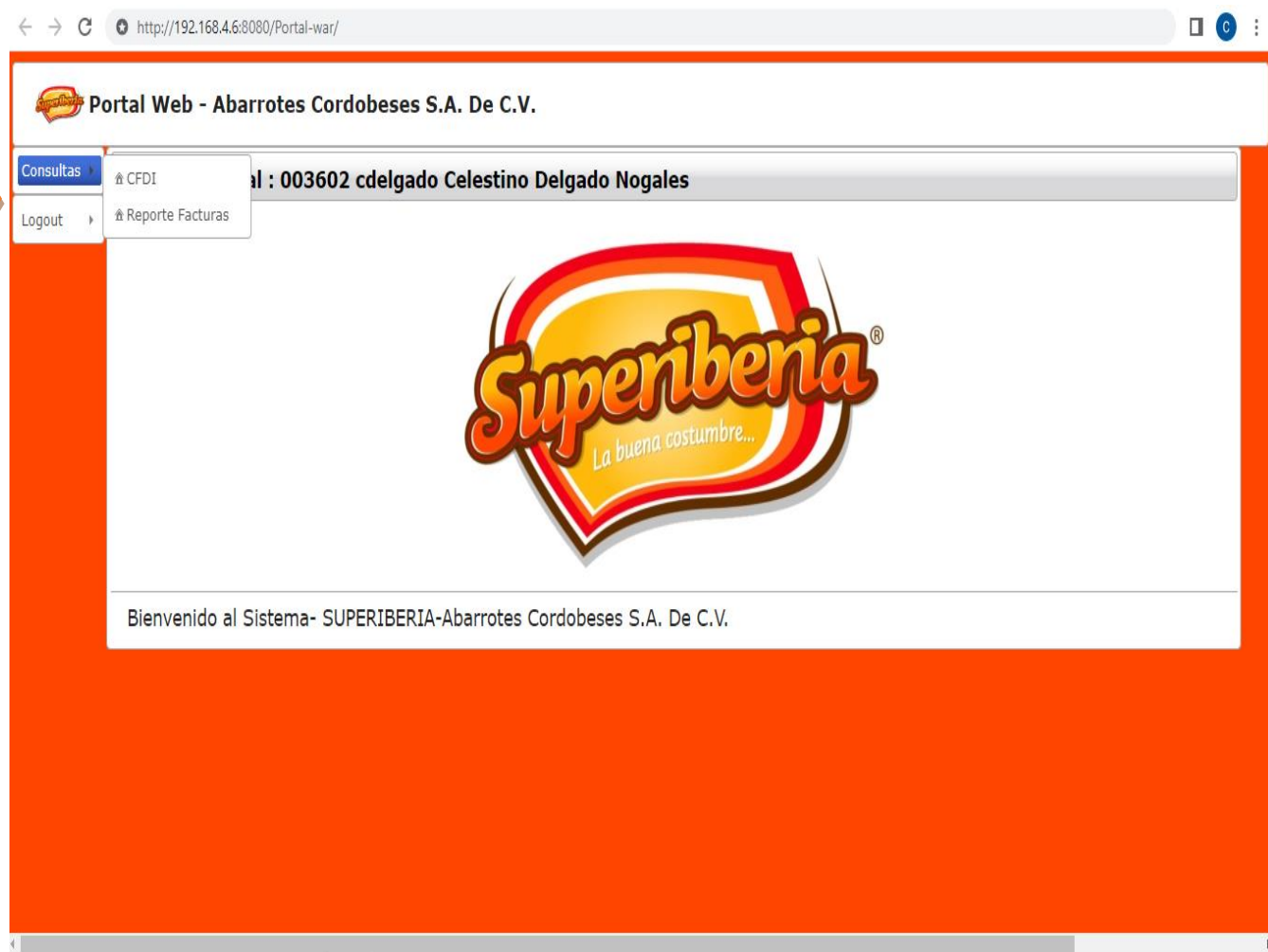
Posteriormente de haber ingresado, se observará la pantalla siguiente pantalla que muestra el inicio del sistema.

Para tener acceso a los módulos se debe de seleccionar las diferentes opciones que se encuentran ubicadas en el menú de la parte superior izquierda.



2. Menú Principal

Para poder empezar a navegar en el Sistema, lo primero que se debe realizar es dirigirse al módulo de Consultas el cual contiene los maestros de los diferentes módulos del sistema como lo muestra la siguiente imagen.



En este se encuentran las diferentes gestiones de los diferentes maestros, así como el de cerrar sesión.

3. Módulo de Consulta de CFDI.

Para gestionar los Proveedores que han enviado documentos - facturas al aplicativo se debe dirigir al módulo de consultas- CFDI y en este dar clic, en él se encontraran varias opciones como se muestra a continuación:

← → ↻ <http://192.168.4.6:8080/Portal-war/faces/consultafacturasvalidas.xhtml>

Abarrotes Cordobeses S.A. De C.V.

Consultas ▶
Logout ▶

Usuario Actual : 003602 cdelgado Celestino Delgado Nogales

Consultar Facturas Validas

Proveedor:

Tipo de Documento: ☒ Todos ☐ Ingreso ☐ Egreso ☐ Traslado ☐ Pago

Fechas:

Folio:

☐ **Folio Fiscal ***

UUID:

Versión	Fecha	Importe	Folio	UUID	Tipo de Documento	PDF	XML
No se encontraron documentos							

SUPERIBERIA-Abarrotes Cordobeses S.A. De C.V.

La pantalla anterior nos muestra los diferentes tipos de consulta que el usuario pueden realizar en el sistema con respecto a los comprobantes- Facturas.

3.1.-Consulta por Proveedor

Para consultar un proveedor en este apartado, debe realizar lo siguiente:

1. En el campo “Proveedor” deberá ingresar el Nombre de Proveedor, descripción, clave interna o RFC).
2. Se desactiva o activa la función por Tipo de documento.
3. Se activan las fechas de consulta dependiendo la fecha que muestre el documento a consultar.
4. Se da clic en el botón “Consultar” y el sistema mostrará la información

Tal como lo muestra la siguiente imagen.

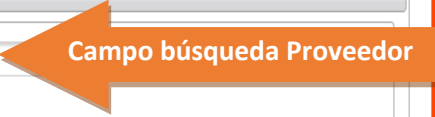
← → ↻ <http://192.168.4.6:8080/Portal-war/faces/consultafacturasvalidas.xhtml>

 **Abarrotes Cordobeses S.A. De C.V.**

Consultas ▶
Logout ▶

Usuario Actual : 003602 cdelgado Celestino Delgado Nogales

Consultar Facturas Validas

Proveedor: 

Tipo de Documento: ☒ Todos ☐ Ingreso ☐ Egreso ☐ Traslado ☐ Pago

Fechas:  

Folio:

☐ Folio Fiscal *

UUID:

Proveedor : COMERCIALIZADORA AUTENTICOS JEBMAR SA DE CV

1 2

Versión	Fecha	Importe	Folio	UUID	Tipo de Documento	PDF	XML
4.00	21-ene-2023	\$6,244.85	11022	39E8A352-32B9-4258-A85A-09C90B6214A2	INGRESO	Descargar	Descargar
4.00	21-ene-2023	\$6,349.25	11020	3F7B8D83-5C5C-40A9-B511-C32BBD37E2AA	INGRESO	Descargar	Descargar
4.00	21-ene-2023	\$6,048.40	11019	BD198968-59D6-4C90-A64D-1F66B960C494	INGRESO	Descargar	Descargar
4.00	21-ene-2023	\$6,882.20	11017	18A8D805-E0BD-4E6C-82D4-2CD60AD164D6	INGRESO	Descargar	Descargar
4.00	21-ene-2023	\$5,320.90	11015	4B902F18-C741-499F-9A2C-CC7F1FC4705C	INGRESO	Descargar	Descargar
4.00	21-ene-2023	\$5,074.10	11014	F4E943E0-8B58-4B47-BD8B-FA29BBD192A0	INGRESO	Descargar	Descargar
4.00	21-ene-2023	\$11,607.85	11013	8AC4BCF8-7300-4A69-A490-687FD314CF5B	INGRESO	Descargar	Descargar
4.00	21-ene-2023	\$12,782.48	11012	7932452D-4F91-48D0-88FC-1267817DF9BA	INGRESO	Descargar	Descargar

Como se muestra en la pantalla anterior la información fue consultada con éxito, y ya podrá hacer uso de la información.

3.2.-Consulta por Folio de Factura

Para realizar o consultar por Folio de Documento deberá realizar lo siguiente:

1. En el campo “Proveedor” deberá ingresar el Nombre de Proveedor, descripción, clave interna o RFC).
2. Se desactiva o activa la función por Tipo de documento
3. Se activan las fechas de consulta dependiendo la fecha que muestre el documento a consultar.
4. En el campo “Folio” deberá ingresar el dato o folio de factura.
5. Se da clic en el botón “Consultar” y el sistema mostrará la información

Tal como lo muestra la siguiente imagen.

← → ↻ http://192.168.4.6:8080/Portal-war/faces/consultafacturasvalidas.xhtml

Abarrotes Cordobeses S.A. De C.V.

Consultas >
Logout >

Usuario Actual : 003602 cdelgado Celestino Delgado Nogales

Consultar Facturas Validas

Proveedor: OPC651_COMERCIALIZADORA AUTENTICOS JEBMAR SA DE CV_CAJ-150219-LD6 **Campo búsqueda Proveedor**

Tipo de Documento: ☒ Todos ☐ Ingreso ☐ Egreso ☐ Traslado ☐ Pago

Fechas: 21/ene/2023 21/ene/2023

Folio: 11023 **Campo búsqueda Folio**

☐ Folio Fiscal *
UUID:

Consultar

Proveedor : COMERCIALIZADORA AUTENTICOS JEBMAR SA DE CV

Versión	Fecha	Importe	Folio	UUID	Tipo de Documento	PDF	XML
4.00	21-ene-2023	\$5,380.85	11023	543A9D0F-6F02-4BA3-9D62-FC2C035AD439	INGRESO	Descargar	Descargar

SUPERIBERIA-Abarrotes Cordobeses S.A. De C.V.

Como se muestra en la pantalla anterior la información fue consultada con éxito, y ya podrá hacer uso de la información.

3.3.-Consulta por Folio Fiscal-UUID

Para realizar o consultar por Folio Fiscal (UUID) deberá realizar lo siguiente:

1. Se activa o selecciona la casilla Folio fiscal.
2. Posteriormente se ingresa el dato del folio fiscal del documento-factura.
3. Se da clic en el botón “Consultar” y el sistema mostrará la información

Tal como lo muestra la siguiente imagen.

← → ↻ http://192.168.4.6:8080/Portal-war/faces/consultafacturasvalidas.xhtml

Abarrotes Cordobeses S.A. De C.V.

Consultas ▶
Logout ▶

Usuario Actual : 003602 cdelgado Celestino Delgado Nogales

Consultar Facturas Validas

Proveedor:

Tipo de Documento: ☒ Todos ☐ Ingreso ☐ Egreso ☐ Traslado ☐ Pago

Fechas:

Folio:

☒ **Folio Fiscal ***

UUID:

Campo búsqueda Folio Fiscal

Versión	Fecha	Importe	Folio	UUID	Tipo de Documento	PDF	XML
4.00	21-ene-2023	\$6,244.85	11022	39E8A352-32B9-4258-A85A-09C90B6214A2	INGRESO	Descargar	Descargar

SUPERIBERIA-Abarrotes Cordobeses S.A. De C.V.

Como se muestra en la pantalla anterior la información fue consultada con éxito, y ya podrá hacer uso de la información.

3.4.-Descarga de CFDI

En este Modulo; el sistema también permite la opción al usuario para descargar los archivos electrónicos del comprobante - factura XML y PDF. Para ello deberá realizar lo siguiente:

1. En el campo “Proveedor” deberá ingresar el Nombre de Proveedor, descripción, clave interna o RFC).
2. Se desactiva o activa la función por Tipo de documento.
3. Se activan las fechas de consulta dependiendo la fecha que muestre el documento a consultar.
4. Se da clic en el botón “Consultar” y el sistema mostrará la información
5. En la tabla que muestra el sistema con la información detallada, en el encabezado “PDF” dará clic en Descargar y automáticamente el archivo de guardará y se almacenará en la carpeta de descargas para su visualización.

Tal como lo muestra la siguiente imagen

← → ↻ http://192.168.4.6:8080/Portal-war/faces/consultafacturasvalidas.xhtml

**Abarrotes Cordobeses S.A. De C.V.**

Consultas >
Logout >

Usuario Actual : 003602 cdelgado Celestino Delgado Nogales

Consultar Facturas Validas

Proveedor: OPC651_COMERCIALIZADORA AUTENTICOS JEBMAR SA DE CV_CAJ-150219-LD6

Tipo de Documento: ☒ Todos ☐ Ingreso ☐ Egreso ☐ Traslado ☐ Pago

Fechas: 21/ene/2023 21/ene/2023

Folio:

☐ Folio Fiscal *
UUID:

Consultar

Proveedor : COMERCIALIZADORA AUTENTICOS JEBMAR SA DE CV

Versión	Fecha	Importe	Folio	UUID	Tipo de Documento	PDF
4.00	21-ene-2023	\$6,244.85	11022	39E8A352-32B9-4258-A85A-09C90B6214A2	INGRESO	Descargar
4.00	21-ene-2023	\$6,349.25	11020	3F7B8D83-5C5C-40A9-B511-C32BBD37E2AA	INGRESO	Descargar
4.00	21-ene-2023	\$6,048.40	11019	BD198968-59D6-4C90-A64D-1F66B960C494	INGRESO	Descargar
4.00	21-ene-2023	\$6,882.20	11017	18A8D805-E0BD-4E6C-82D4-2CD60AD164D6	INGRESO	Descargar
4.00	21-ene-2023	\$5,320.90	11015	4B902F18-C741-499F-9A2C-CC7F1FC4705C	INGRESO	Descargar
4.00	21-ene-2023	\$5,074.10	11014	F4E943E0-8B58-4B47-BD8B-FA29BBD192A0	INGRESO	Descargar
4.00	21-ene-2023	\$11,607.85	11013	8AC4BCF8-7300-4A69-A490-687FD314CF5B	INGRESO	Descargar
4.00	21-ene-2023	\$12,782.48	11012	7932452D-4F91-48D0-88FC-1267817DF9BA	INGRESO	Descargar

Campo PDF descarga

Como se muestra en la pantalla anterior la información fue consultada con éxito, y ya podrá hacer uso de la documentación.

Imagen que muestra la descarga del documento en formato PDF.

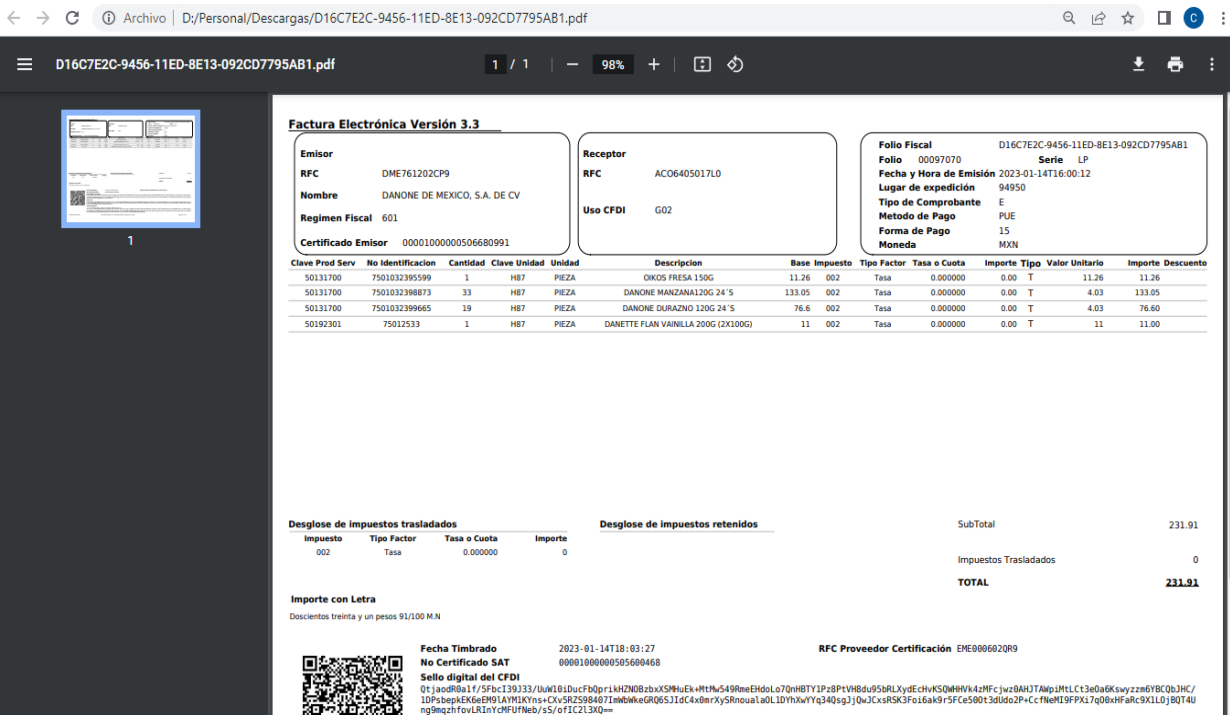


Imagen que muestra la descarga del documento en formato XML (txt).



4.- Módulo de Reportes.

Para gestionar los reportes de facturas que se encuentran almacenadas en el aplicativo se debe dirigir al módulo de Consultas-Reporte Facturas y en este dar clic, en él se encontrarán varias opciones como se muestra a continuación

← → ↻ http://192.168.4.6:8080/Portal-war/faces/reportefacturasvalidas.xhtml

**Abarrotes Cordobeses S.A. De C.V.**

Consultas ▶
Logout ▶

Usuario Actual :cdelgado Celestino Delgado Nogales

Reporte EXCEL - Facturas - Validas- SAT

Proveedor:

Tipo de Documento: ☒ Todos ☐ Ingreso ☐ Egreso ☐ Traslado ☐ Pago

Fechas de Consulta:

Mostrar

Exportar Datos 

TIPO PAGO	VERSIÓN CFDI	TIPO DOCUMENTO	RFC PROVEEDOR	RFC RECEPTOR	FECHA EMISION	FOLIO	IMPORTE	UUID
No se encontraron documentos								

SUPERIBERIA-Abarrotes Cordobeses S.A. De C.V.

4.1.-Consulta de Reporte

Para obtener el reporte de facturas del sistema, deberá realizar lo siguiente:

1. En el campo “Proveedor” deberá ingresar el Nombre de Proveedor, descripción, clave interna o RFC).
2. Se desactiva o activa la función por Tipo de documento “Todos”.
3. Se activan las fechas de consulta dependiendo la fecha que requiera que el sistema de proporcione el reporte.
4. Se da clic en el botón “Mostrar” y el sistema mostrará la información

← → ↻ http://192.168.4.6:8080/Portal-war/faces/reportefacturasvalidas.xhtml

Abarrotes Cordobeses S.A. De C.V.

Consultas ▶
Logout ▶

Usuario Actual :cdelgado Celestino Delgado Nogales

Reporte EXCEL - Facturas - Validas- SAT

Proveedor: OPC651_COMERCIALIZADORA AUTENTICOS JEBMAR SA DE CV_CAJ-150219-LD6 **Campo búsqueda Proveedor**

Tipo de Documento: ☒ Todos ☐ Ingreso ☐ Egreso ☐ Traslado ☐ Pago

Fechas de Consulta: 21/ene/2023 21/ene/2023 **Fecha de Consulta**

Mostrar Datos

Datos Proveedor : _COMERCIALIZADORA AUTENTICOS JEBMAR SA DE CV_REGIMEN FISCAL_626

TIPO PAGO	VERSIÓN CFDI	TIPO DOCUMENTO	RFC PROVEEDOR	RFC RECEPTOR	FECHA EMISION	FOLIO	IMPORTE	UUID
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11024	\$2,078.60	C62B62C1-9604-4D35-B872-5AF19822BCBC
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11023	\$5,380.85	543A9D0F-6F02-4BA3-9D62-FC2C035AD439
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11022	\$6,244.85	39E8A352-32B9-4258-A85A-09C90B6214A2
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11020	\$6,349.25	3F7B8D83-5C5C-40A9-B511-C32BBD37E2AA
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11019	\$6,048.40	BD198968-59D6-4C90-A64D-1F66B960C494
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11017	\$6,882.20	18A8D805-E0BD-4E6C-82D4-2CD60AD164D6
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11015	\$5,320.90	4B902F18-C741-499F-9A2C-CC7F1FC4705C

Como se muestra en la pantalla anterior la información fue consultada con éxito, y ya podrá hacer uso de información.

4.2.-Exportar Reporte

Para Exportar el reporte de facturas del sistema, deberá realizar lo siguiente:

1. En el campo "Proveedor" deberá ingresar el Nombre de Proveedor, descripción, clave interna o RFC).
2. Se desactiva o activa la función por Tipo de documento "Todos".
3. Se activan las fechas de consulta dependiendo la fecha que requiera que el sistema de proporcione el reporte.
4. Se da clic en el botón "Mostrar" y el sistema mostrará la información
5. Se da clic en el botón Exportar Datos.

← → ↻ <http://192.168.4.6:8080/Portal-war/faces/reportefacturasvalidas.xhtml>

 **Abarrotes Cordobeses S.A. De C.V.**

Consultas ▾
Logout ▾

Usuario Actual : cdelgado Celestino Delgado Nogales

Reporte EXCEL - Facturas - Validas- SAT

Proveedor:

Tipo de Documento: ☒ Todos ☐ Ingreso ☐ Egreso ☐ Traslado ☐ Pago

Fechas de Consulta:



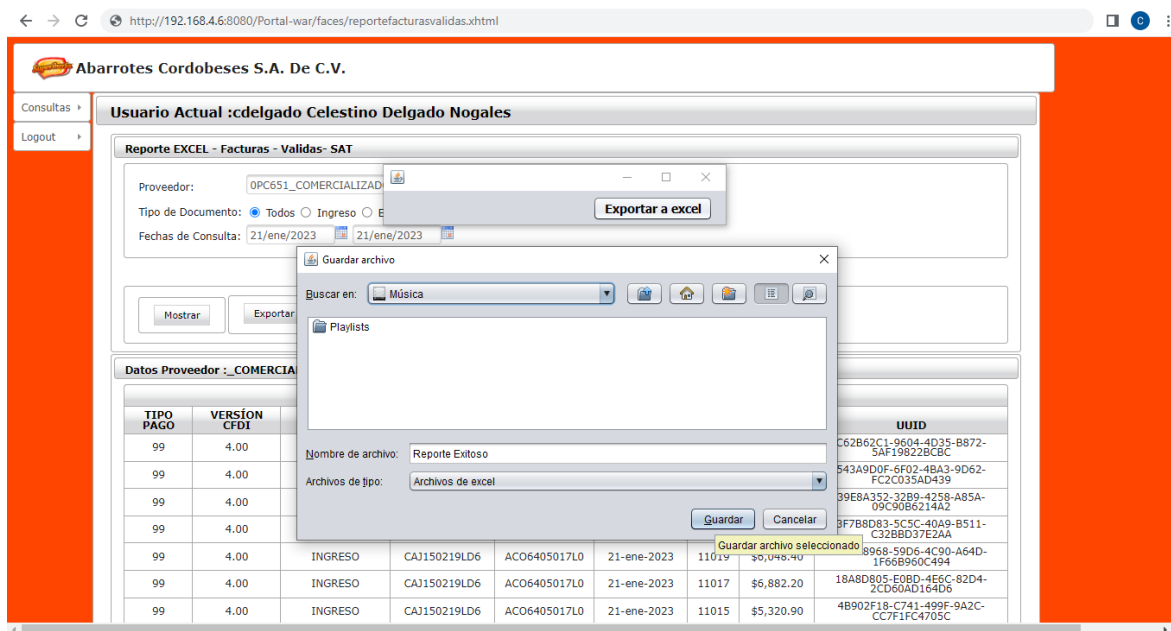
Exportar Datos.

Datos Proveedor : _COMERCIALIZADORA AUTENTICOS JEBMAR SA DE CV_REGIMEN FISCAL_626

TIPO PAGO	VERSIÓN CFDI	TIPO DOCUMENTO	RFC PROVEEDOR	RFC RECEPTOR	FECHA EMISION	FOLIO	IMPORTE	UUID
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11024	\$2,078.60	C62B62C1-9604-4D35-B872-5AF19822BCBC
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11023	\$5,380.85	543A9D0F-6F02-4BA3-9D62-FC2C035AD439
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11022	\$6,244.85	39E8A352-32B9-4258-A85A-09C90B6214A2
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11020	\$6,349.25	3F7B8D83-5C5C-40A9-B511-C328BD37E2AA
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11019	\$6,048.40	BD198968-59D6-4C90-A64D-1F66B960C494
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11017	\$6,882.20	18A8D805-E0BD-4E6C-82D4-2CD60AD164D6
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11015	\$5,320.90	4B902F18-C741-499F-9A2C-CC7F1FC4705C

El sistema mostrará la siguiente ventana y dará clic en exportar a Excel y guardará el archivo.

Tal como se muestra a continuación:



Descargado el archivo, procedera a abrir el documento y le mostrara la informacion,

Tal como se muestra en la siguiente imagen.

ABARROTES CORDOBESES SA DE CV

Reporte de Facturas, relacionadas en el Sistema Validador de Facturas

Proveedor: COMERCIALIZADORA AUTENTICOS JEBMAR SA DE CV

De: 21/01/2023 a: 21/01/2023

TIPO PAGO	VERSION CFDI	TIPO DOCUMENTO	RFC PROVEEDOR	RFC RECEPTOR	FECHA EMISION	FOLIO	IMPORTE	UUID
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11024	\$2,078.60	C62B62C1-9604-4D35-B872-5AF198228CBC
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11023	\$5,380.85	543A9D0F-6F02-4BA3-9D62-FC2C035AD439
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11022	\$6,244.85	39E8A352-32B9-4258-A85A-09C9086214A2
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11021	\$8,555.20	18F1C353-4809-4BAE-AE10-F75041298E0E
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11020	\$6,349.25	3F7B8D83-5C5C-40A9-B511-C32B8D37E2AA
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11019	\$6,048.40	BD198968-59D6-4C90-A64D-1F668960C494
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11018	\$11,315.20	E5C666BA-1140-4C13-B0F4-61B612C35116
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11017	\$6,882.20	18A8D805-E0BD-4E6C-82D4-2CD60AD164D6
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11016	\$8,264.50	7012DF89-1AE8-45BF-A4D0-D81E7AF18B7D
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11015	\$5,320.90	4B902F18-C741-499F-9A2C-CC7F1FC4705C

5.- Cerrar Sesión.

Para cerrar sesión deberá dar clic en el botón Logout.

← → ↻ http://192.168.4.6:8080/Portal-war/faces/reportefacturasvalidas.xhtml

Abarrotes Cordobeses S.A. De C.V.

Consultas ▶
Logout ▶

Usuario Actual :cdelgado Celestino Delgado Nogales

Reporte EXCEL - Facturas - Validas- SAT

Proveedor: 0PC651_COMERCIALIZADORA AUTENTICOS JEBMAR SA DE CV_CAJ-150219-LD6

Tipo de Documento: ☒ Todos ☐ Ingreso ☐ Egreso ☐ Traslado ☐ Pago

Fechas de Consulta: 21/ene/2023 21/ene/2023

Mostrar Exportar Datos Excel

Datos Proveedor : _COMERCIALIZADORA AUTENTICOS JEBMAR SA DE CV_REGIMEN FISCAL_626

TIPO PAGO	VERSIÓN CFDI	TIPO DOCUMENTO	RFC PROVEEDOR	RFC RECEPTOR	FECHA EMISION	FOLIO	IMPORTE	UUID
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11024	\$2,078.60	C62B62C1-9604-4D35-B872-5AF19822BCBC
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11023	\$5,380.85	543A9D0F-6F02-4BA3-9D62-FC2C035AD439
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11022	\$6,244.85	39E8A352-32B9-4258-A85A-09C90B6214A2
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11020	\$6,349.25	3F7B8D83-5C5C-40A9-B511-C32BBD37E2AA
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11019	\$6,048.40	BD198968-59D6-4C90-A64D-1F66B960C494
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11017	\$6,882.20	18A8D805-E0BD-4E6C-82D4-2CD6AD164D6
99	4.00	INGRESO	CAJ150219LD6	ACO6405017L0	21-ene-2023	11015	\$5,320.90	4B902F18-C741-499F-9A2C-CC7F1FC4705C

Posteriormente le mostrará la siguiente ventana. Y habrá cerrado sesión.

Nueva pestaña x 192.168.4.6:8080/Portal-war/ x Portal SUPERIBERIA x +

← → X No seguro | 192.168.4.6:8080/Portal-war/

Ha terminado su sesion

Anexo 2 - Ejemplos Varios

Este anexo contiene ejemplos sobre cómo debes utilizar imágenes, tablas y otros elementos en tu documento.

Ejemplo sobre como incluir imágenes. Las imágenes, como **¡Error! No se encuentra el origen de la referencia.**, también deben estar correctamente referenciadas, y solo se deben incluir imágenes a las que se haga referencia en el texto. La numeración se coloca por debajo.

Ejemplo sobre cómo se deben marcar correctamente las citas, “Esta es una cita” (A. & Beichner), y cómo introducir notas al pie¹.

Ejemplo sobre como insertar una tabla, y hacer referencia a ella. Véase la **¡Error! No se encuentra el origen de la referencia.**. Solo se deben incluir tablas si se hace referencia a ellas en el texto. El título de las tablas se coloca arriba.

Gesto	Ocurrencias
Piedra	10
Papel	6
Tijeras	12

Tabla 36. Ocurrencias de los gestos

¹ Es muy fácil utilizar las notas al pie.