



UNIVERSIDAD PARTICULAR DE CHICLAYO
FACULTAD DE ARQUITECTURA Y URBANISMO, INGENIERIAS y ARTE
ESCUELA PROFESIONAL DE INGENIERIA INFORMATICA Y DE SISTEMAS

TESIS

“Implementación del Módulo de Envío de Documentos de Ventas del Restaurant Zarandaja de la Ciudad de Chiclayo Consumiendo el servicio SOAP del Operador de Servicios Electrónicos NUBEFACT.”

PARA OPTAR EL TÍTULO PROFESIONAL DE:
INGENIERO INFORMÁTICO Y DE SISTEMAS

PRESENTADO POR:

Bach. Eduardo Anthony Ramírez Benavides

Pimentel, junio de 2021

DEDICATORIA

Mi tesis la dedico con todo mi amor y cariño. A ti Dios que diste la oportunidad de vivir y de regalarme a una familia maravillosa. Con mucho cariño principalmente a mis padres que me dieron la vida y han estado conmigo en todo momento, quienes han sido mi ejemplo de vida y siempre me han apoyado a lo largo de toda mi vida, dándome su cariño y comprensión, los cuales han forjado mis valores, mis principios y todo lo que soy como persona. Este logro hoy representa una gran motivación para seguir adelante y conseguir todo lo que me proponga.

AGRADECIMIENTO

El presente trabajo de tesis en primer lugar me gustaría agradecerle a Dios por bendecirme y hacer realidad este sueño anhelado. A la **UNIVERSIDAD PARTICULAR DE CHICLAYO** por darme la oportunidad de estudiar y ser profesionales. A mis padres quienes en todo momento me brindaron de su apoyo incondicional, sin el cual no hubiera sido posible alcanzar esta meta. También me gustaría agradecer de manera especial a mi asesor de tesis el Ing. Eduardo Arrascue Becerra, por su rectitud en su profesión como docente, por sus consejos, que ayudaron a formarme como persona e investigador. Son muchas las personas que han formado parte de nuestra vida profesional a las que me encantaría agradecerles su amistad, consejos, apoyo, ánimo y compañía en los momentos más difíciles de mi vida. Algunas están aquí con nosotros y otras en nuestros recuerdos y en nuestros corazones, sin importar en donde estén quiero darles las gracias por formar parte de nosotros, por todo lo que me han brindado y por todas sus bendiciones. Para ellos: Muchas gracias y que Dios los bendiga.

DECLARACIÓN DE AUTENTICIDAD

Yo, Bach. Eduardo Anthony Ramírez Benavides, identificado con DNI 77159714 egresado de la Escuela de Ingeniería Informática y de Sistemas de la Facultad de Arquitectura y Urbanismo, a efectos de cumplir con las disposiciones vigentes consideradas en el reglamento de Grados y Títulos de la Universidad Particular de Chiclayo, declaro bajo juramento que el informe del trabajo de suficiencia profesional denominada:

“Implementación del Módulo de Envío de Documentos de Ventas del Restaurant Zarandaja de la Ciudad de Chiclayo Consumiendo el servicio SOAP del Operador de Servicios Electrónicos NUBEFACT.”

Que:

1. El informe es de mi autoría.
2. He respetado y seguido los lineamientos de diseño, reglamento y parámetros establecidos para el desarrollo del informe.
3. El informe no ha sido plagiado, pues no ha sido objeto de publicaciones, ni presentada anteriormente para obtener un grado académico previo o título profesional.
4. Los datos presentados son resultado de visitas, entrevistas y consultas y que éstos no han sido falseados, pues los resultados contribuirán a conocer la realidad problemática con respecto a la investigación.

En tal sentido asumo la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de información aportada por lo cual me someto a lo dispuesto en las normas académicas de la Universidad de Chiclayo.

Pimentel, mayo de 2021

El autor

PRESENTACIÓN

Señores Miembros del Jurado:

De conformidad y cumpliendo lo estipulado en el Reglamento de Grados y Títulos de la Facultad de Arquitectura y Urbanismo e Ingenierías de la Universidad Particular de Chiclayo, para optar el Título Profesional de Ingeniero Informático y de Sistema, someto a vuestra consideración el presente informe Titulado:

“Implementación del Módulo de Envío de Documentos de Ventas del Restaurant Zarandaja de la Ciudad de Chiclayo Consumiendo el servicio SOAP del Operador de Servicios Electrónicos NUBEFACT.”

Gracias

ÍNDICE GENERAL

Contenido

DEDICATORIA.....	2
AGRADECIMIENTO.....	3
DECLARACIÓN DE AUTENTICIDAD	4
PRESENTACIÓN	5
ÍNDICE GENERAL	6
ÍNDICE DE FIGURAS	8
ÍNDICE DE TABLAS.....	10
RESUMEN.....	11
ABSTRACT	13
CAPITULO I: INTRODUCCIÓN.....	16
1.1. Realidad Problemática.....	16
1.2. Formulación del Problema	17
1.3. Hipótesis.....	17
1.4. Objetivos	17
1.4.1. Objetivo General	17
1.4.2. Objetivos Específicos.....	17
CAPITULO II: BASES TEÓRICAS.....	20
2.1. Antecedentes de la Investigación	20
2.2. Marco Teórico	23
2.3. Definición de términos	50
CAPÍTULO III: MARCO METODOLÓGICO	54
3.1. Variables.....	54
3.2. Operacionalización de las variables:	54
3.3. Metodología	55
3.4. Población, muestra y muestreo.....	55
3.5. Técnicas e instrumentos de recolección de datos.	55
3.6. Métodos de análisis de datos	56
3.7. Aspectos éticos.....	56
CAPITULO IV: RESULTADOS	58
CAPITULO V: DISCUSIÓN	121
CAPÍTULO VI: CONCLUSIONES.....	124
CAPÍTULO VII: RECOMENDACIONES	126
CAPÍTULO VIII: PROPUESTA TECNOLÓGICA	128
8.1. Generalidades.....	128
8.2. Análisis.....	130

<i>“Implementación del Módulo de Envío de Documentos de Ventas del Restaurant Zarandaja de la Ciudad de Chiclayo Consumiendo el servicio SOAP del Operador de Servicios Electrónicos NUBEFACT”</i>	
8.3.	Diseño..... 130
8.4.	Implementación del sistema. 131
8.4.	Presupuesto..... 132
CAPITULO IX: BIBLIOGRAFIA 136	
ANEXO 139	

ÍNDICE DE FIGURAS

Imagen No. 1 Estructura SOAP	25
Imagen No. 2 Tipos de datos en Json	42
Imagen No. 3 Flujo del OSE	46
Imagen No. 4 Modelo del Negocio	58
Imagen No. 5 Diagrama de estado Comprobante.	59
Imagen No. 6. Diagrama de estado CDR Sunat	60
Imagen No. 7 Diagrama de Actividades-Generar documento electrónico	61
Imagen No. 8 Diagrama de Actividades- Consultar validez del CPE	62
Imagen No. 9 Diagrama de secuencia Consultar documento electrónico.....	63
Imagen No. 10 Diagrama de secuencia Consultar validez del CPE	64
Imagen No. 11 Modelo relacional parte 1.	65
Imagen No. 12 Modelo relacional parte 2	89
Imagen No. 13 Modelo relacional parte 3	90
Imagen No. 14 Modelo relacional parte 4	91
Imagen No. 15 Tabla tbestadomesa	92
Imagen No. 16 Tabla tbmesa.....	92
Imagen No. 17 Tabla tbdetalle_comanda	92
Imagen No. 18 Tabla tbcomanda.....	93
Imagen No. 19 Tabla tbcategoría	93
Imagen No. 20 Tabla tbproducto	94
Imagen No. 21 Tabla tbpersona	94
Imagen No. 22 Tabla tbtipo_documento_identidad	95
Imagen No. 23 Tabla tbdocpago_ctacobrar	95
Imagen No. 24 Tabla tbtipodoc.....	96
Imagen No. 25 Tabla tbdetalle_docpago_ctacobrar_impreso.....	96
Imagen No. 26 Tabla tbmediopago.....	96
Imagen No. 27 Tabla tbmediopago_tbdocpago_ctacobrar	97
Imagen No. 28 Tabla tbmediopago_tbdocpago_ctacobrar	97
Imagen No. 29 Tabla tbsucursal.	97
Imagen No. 30 Tabla tbsucursalpcs	97
Imagen No. 31 Tabla tbseriesdoc_emitir	98
Imagen No. 32 Tabla tbconfiguracion	98
Imagen No. 33 Tabla tbrol	98

Imagen No. 34 Tabla tbrolpersona	99
Imagen No. 35 Tabla tbctas_cobrar.....	99
Imagen No. 36 Tabla tbdetpago_ctapagar.....	100
Imagen No. 37 Tabla tbfacturacionelectronica.....	100
Imagen No. 38 Texto Json ejemplo con ruc.....	105
Imagen No. 39 Pantalla de inicio de la Aplicación.	106
Imagen No. 40 Menú principal del sistema online Zarandaja Restaurant 2020.	107
Imagen No. 41 Registro general de personas.	108
Imagen No. 42 Mantenimiento de producto	109
Imagen No. 43 Registro de producto.	110
Imagen No. 44 Mapa de mesas para crear comanda.	111
Imagen No. 45 Generar comanda.....	112
Imagen No. 46 Registro de Pago en efectivo	113
Imagen No. 47 Registro de pago Simple.	114
Imagen No. 48 Múltiples Medios de pago.....	115
Imagen No. 49 Venta con factura.	116
Imagen No. 50 Envío de comprobantes de pagos a OSE-SUNAT	117
Imagen No. 51 Retorno de OSE Sunat.....	118
Imagen No. 52 Sistemas de emisión electrónica de comprobantes de Pago	129

ÍNDICE DE TABLAS

Tabla 1 Operacionalización de las variables.....	54
Tabla 2 Presupuesto.....	119
Tabla 3Recuperación - Presupuesto.....	132
Tabla 4 Indicadores de ingreso.....	132
Tabla 5 Flujo de caja parte 1	133
Tabla 6 Flujo de caja Parte 2	134

RESUMEN

El presente informe de tesis denominada “Implementación del Módulo de Envío de Documentos de Ventas del Restaurant Zarandaja de la Ciudad de Chiclayo Consumiendo el servicio SOAP del Operador de Servicios Electrónicos NUBEFACT.” tiene como finalidad explicar y solucionar los inconvenientes que se tiene al generar y enviar los documentos de ventas que exige la SUNAT mediante un operador de servicios electrónicos.

Considerando la estructura otorgada por la Escuela de Informática y de Sistemas de la Universidad de Chiclayo del departamento de Lambayeque, para el desarrollo del informe de las tesis, presento el resumen de capítulos que contiene el presente trabajo de investigación.

El Capítulo I Está conformado, de acuerdo a lo explicado en la guía de la escuela, por la introducción de la investigación, la formulación del problema, hipótesis y objetivos de la presente investigación.

El Capítulo II, redacto las bases teóricas, con las respectivas referencias a sus autores o fuentes. En este capítulo se especifica los antecedentes de la investigación, el marco teórico, los términos, así como las definiciones mas relevantes de este material escrito.

El Capítulo III, especifico el marco metodológico el cual lo constituye la declaración, definición y operacionalización de las variables dependiente e independiente también menciono la metodología de investigación, la población y muestra a tomar en cuenta, las técnicas e instrumentos de recolección de datos incluyendo además el método de análisis de datos y el aspecto ético de la investigación.

En el Capítulo IV se escribe los resultados obtenidos en las encuestas aplicadas a los clientes involucrados en esta investigación especificando los datos obtenidos antes y después de la implementación de la aplicación.

En el Capítulo V se establece la discusión de los resultados utilizando los cuadros en Excel con datos en forma cuantitativa y porcentual obtenidos

del cuestionario aplicado en la encuesta a los usuarios que interactúan con la aplicación antes de la implementación, así como después de la misma.

En el capítulo VI y VII denoto las conclusiones y recomendaciones a la que llevan esta investigación.

En el capítulo VIII está conformado por la propuesta tecnológica mencionándose las generalidades del proyecto, así como el análisis, diseño e implementación de la aplicación. Incluye este capítulo el presupuesto para desarrollar la investigación.

En el capítulo IX se indica las referencias bibliográficas que son todas las fuentes bibliográficas y linkográficas utilizadas y revisadas desde el inicio y hasta el final de esta investigación.

Por último, en el capítulo X se incluye la información anexa a esta investigación.

ABSTRACT

This thesis report called "Implementation of the Module for Sending Sales Documents of the Zarandaja Restaurant of the City of Chiclayo Consuming the SOAP service of the Electronic Services Operator NUBEFACT." Its purpose is to explain and solve the inconveniences encountered when generating and sending the sales documents required by SUNAT through an electronic service operator.

Considering the structure granted by the School of Informatics and Systems of the University of Chiclayo in the department of Lambayeque, for the development of the thesis report, I present the summary of chapters that this research work contains.

Chapter I It is made up, according to what is explained in the school guide, by the introduction of the investigation, the formulation of the problem, hypotheses and objectives of the present investigation.

Chapter II, I write the theoretical bases, with the respective references to their authors or sources. This chapter specifies the background of the investigation, the theoretical framework, the terms, as well as the most relevant definitions of this written material.

Chapter III, specified the methodological framework which is constituted by the declaration, definition and operationalization of the dependent and independent variables, also mentioned the research methodology, the population and sample to take into account, the data collection techniques and instruments, also including the method of data analysis and the ethical aspect of the research.

In Chapter IV the results obtained in the surveys applied to the clients involved in this research are written, specifying the data obtained before and after the implementation of the application.

In Chapter V the discussion of the results is established using the tables in Excel with data in quantitative and percentage form obtained from the questionnaire applied in the survey to the users who interact with the application before implementation, as well as after it.

In chapter VI and VII I denote the conclusions and recommendations to which this research leads.

Chapter VIII is made up of the technological proposal, mentioning the generalities of the project, as well as the analysis, design and implementation of the application. This chapter includes the budget to develop the research.

Chapter IX indicates the bibliographic references that are all the bibliographic and linkographic sources used and reviewed from the beginning and until the end of this research.

Finally, chapter X includes the information attached to this investigation.

CAPÍTULO I

INTRODUCCIÓN

CAPITULO I: INTRODUCCIÓN

1.1. Realidad Problemática

Actualmente, la información suficiente y oportuna constituye el insumo más importante para la toma de decisiones. Para contar con esta información, es necesario acceder y analizar ordenadamente los datos que se generan en los sistemas de la organización y aquellos procedentes de otras fuentes. Hoy casi todas las administraciones tributarias del mundo, bajo una estructura del Centro Interamericano de Administraciones Tributarias (CIAT), están en un amplio desarrollo de la era electrónica, siendo este organismo encargado de proveer asistencia técnica especializada para la actualización modernización de las administraciones tributarias, promoviendo valores de integridad, transparencia y ética, con la disposición de prevenir y combatir todas las formas de fraude, evasión y elusión tributaria facilitando el cumplimiento voluntario.

Todas las empresas formalizadas y aquellas en camino de la formalización están obligadas en emitir documentos electrónicos, así como enviar información a la Sunat respecto a sus comprobantes de pago y otros, según la naturaleza de la empresa, emitidos por las transacciones realizadas por los negocios.

El Operador de Servicios Electrónicos (OSE) es quién se encarga de comprobar informáticamente el cumplimiento de las condiciones de emisión de los comprobantes electrónicos que sean emitidos a través del Sistema de Emisión Electrónica – OSE (SEE-OSE), cuando sean contratados por el emisor electrónico.

El operador no sólo validará la correcta emisión de un comprobante de pago electrónico sino también enviará la información de los comprobantes de pago validados, con sus constancias respectivas, así como de los documentos relacionados a la SUNAT. El OSE debe realizar la comprobación informática de las condiciones de emisión de los documentos que el emisor le envíe, de cumplir con las condiciones el OSE le enviará, además la constancia de recepción, caso contrario le enviará una comunicación de inconsistencias.

Por lo tanto, el restaurant Zarandaja por ser una empresa

formal que emite documentos y comprobantes autorizados por la Sunat requiere que la una empresa Operadora de Servicios Electrónicos autorizados por la Sunat para que permita previamente la validación de dichos comprobantes y documentos emitidos en una transacción del negocio. Para ello se requiere que la aplicación que emita dichos comprobantes o documentos se conecten a un proveedor de OSE para cumplir con la validación y el envío de ésta a la Sunat los documentos validados. El Restaurant contrata los servicios de Nubefact como proveedor de este servicio y por lo tanto se requiere la aplicación que en forma online conecte a dicha operadora, reciba los documentos, valide y envíe a la Sunat constancia de esta validación.

1.2. Formulación del Problema

¿La implementación del Módulo OSE del Restaurant Zarandaja De La Ciudad De Chiclayo Consumiendo El Servicio Soap Del Operador De Servicios Electrónicos Nubefact permitirá la validación online en forma oportuna de los documentos de venta?

1.3. Hipótesis

La implementación del Módulo OSE del Restaurant Zarandaja De La Ciudad De Chiclayo Consumiendo El Servicio Soap Del Operador De Servicios Electrónicos Nubefact permitirá la validación online en forma oportuna de los documentos de venta.

1.4. Objetivos

1.4.1. Objetivo General

Implementar el Módulo de Envío de Documentos de Ventas del Restaurant Zarandaja de la Ciudad de Chiclayo Consumiendo el servicio SOAP del Operador de Servicios Electrónicos NUBEFAC.

1.4.2. Objetivos Específicos

- a. Establecer los requerimientos funcionales para la conexión y emisión remota de los documentos de ventas a la OSE

Nubefact.

- b. Implementar el módulo de envíos de documentos de ventas consumiendo el servicio SOAP del Operador de Servicios Electrónicos NUBEFACT.
- c. Gestionar el almacenamiento de los datos correspondientes a la emisión de documentos electrónicos de ventas.
- d. Emitir los documentos electrónicos solicitados por la Sunat como son Boleta electrónica, Factura electrónica y Nota de Crédito electrónica

Capítulo II: BASES TEÓRICAS

CAPITULO II: BASES TEÓRICAS

2.1. Antecedentes de la Investigación

1.1.1. Tesis: “Emisión de los Comprobantes de Pago Electrónicos para Evitar las Sanciones Tributarias en la Empresa Binresa S.A.C., 2017”.

Autores: Yonirian Karina Chávez Pío.

Isabel Ivonne Requena Maza.

Universidad: UTP- Lima.

Año: 2017

Resumen:

- “• *El estudio realizado trajo consigo la siguiente conclusión, mediante el cual se realizó un análisis en determinar los procedimientos que se debe seguir para la emisión de comprobantes de pago electrónicos que causa las sanciones tributarias, y se observó el grado de incidencia significativa que trajo consigo el incumplimiento de las obligaciones formales, tributarias causando las infracciones, sanciones e intereses.*
- *Se observó que las empresas BINRESA SAC desconoce las características y beneficios de emitir comprobantes electrónicos que causa las sanciones tributarias por lo que durante el 2017 la empresa fue sujeto obligado para emitir comprobantes electrónicos en una fecha prevista, pero no cumplieron con el requerimiento, por lo tanto, la empresa fue sancionada y perjudicó a su proveedor principal.*
- *Se evaluó los diversos sistemas que ha incorporado SUNAT en su portal web y sea elegido el más conveniente la cual fue emisión SOL-SEE para la empresa BINRESA SAC, para así poder emitir los comprobantes electrónicos de manera inmediata, correcta y oportuna facilitando todo el proceso de facturación a la empresa.*
- *Se definió los pasos para emitir los comprobantes electrónicos que evita las sanciones tributarias, de manera que los encargados de realizar este procedimiento deberán cumplir paso a paso para evitar cualquier contingencia que pueda perjudicar a la empresa económicamente.*
- *Se analizó incumplimiento de las obligaciones formales para la emisión de comprobantes electrónicos que evita las sanciones tributarias y se*

determinó que la más perjudicada sería el cliente principal de BINRESA SAC por la falta de cultura tributaria, la empresa EL ESPIRAL SAC tendrá que reparar los gastos obtenidos por los servicios prestados por BINRESA SAC adicionando todo el gastos obtenidos por dicha empresa en la determinación del impuesto a la renta, incrementando así el importe a pagar en el periodo 2017.” (Chávez & Requena, 2017)

2. Tesis: “Sistema de facturación electrónica para la gestión de comprobantes de pago basado en ISO/IEC19845:2015 en Acgenesys S.A.C.”

Autor: Theany Kristy Navarro Flores

Universidad: César Vallejo

Año: 2017

Resumen:

“Este estudio comprende el análisis, desarrollo e implementación de un sistema de facturación electrónica para la gestión de comprobantes de pago en la empresa ACGENESYS S.A.C, la cual actualmente presenta ciertas incidencias y gastos no definidos en la emisión de los comprobantes de pago. El objetivo principal fue determinar los efectos de la implementación de un sistema de facturación electrónica para la gestión de comprobantes de pago y ver su efecto sobre el porcentaje de incidencias en las emisiones de los comprobantes y el porcentaje de gastos operacionales. La muestra estuvo conformada por los registros de comprobantes de pago realizados en un periodo de 30 días. El tipo de estudio es de tipo experimental -aplicado y el diseño de tipo pre-experimental. Como resultados se obtuvo que el porcentaje de incidencias en emisiones antes del sistema de facturación electrónica fue 4.5491 y luego de la implementación fue 0.9111, lo que significa una reducción de 79.97%. Para el indicador porcentaje de gastos operacionales, antes del sistema de facturación se tenía un valor de 0.0278 y luego de la implementación este valor se redujo a 0.0171, presentando una reducción porcentual de 61.551%. Finalmente, la conclusión fue que la implementación de un sistema de facturación tuvo un efecto positivo en la gestión de comprobantes de pago de la empresa ACGENESYS S.A.C.” (Navarro, 2017)

3. Tesis: “Factores que limitan el uso de la factura electrónica por la micro y pequeña empresa (MYPE) del distrito de La Victoria – 2016”.

Autor: César David Matos Guerrero

Lugar: Universidad César Vallejo.

Año: 2017

Resumen:

“La presente investigación denominada factores que limitan el uso de la factura electrónica por la micro y pequeña empresa del distrito de La Victoria, tuvo como objetivo principal identificar cuál es el factor predominante que limita el uso de la factura electrónica, toda vez que existen importantes barreras que impiden a las MYPE del distrito adoptar esta tecnología.

Esta investigación se realizó tomando en consideración al paradigma positivista, con un enfoque cuantitativo, con un estudio del tipo sustantivo, de diseño no experimental y de corte transversal, descriptivo y explicativo, se usó el método hipotético deductivo; la muestra fue de 378 MYPE del distrito de La Victoria; la técnica de la investigación aplicado ha sido la encuesta usando como instrumento un cuestionario de 35 ítems, medidos en la escala de Likert; para la confiabilidad del instrumento se utilizó el Alfa de Cronbach.

En la investigación se identificó que el factor predominante que limita el uso de la factura electrónica, fue el factor tecnológico, llegándose a concluir que es el factor que muestra mayor coeficiente de regresión logística ($B = 0.427$), por tanto, contribuye con mayor limitación al uso de la factura electrónica.”

(Matos, 2017)

2.2. Marco Teórico

2.2.1. SOAP. Para definir el SOAP recorro a las definiciones y explicaciones de las siguientes fuentes:

A. IBM Knowledge Center

(https://www.ibm.com/support/knowledgecenter/es/SSKM8N_8.0.0/com.ibm.etools.mft.doc/ac55770_.htm):

"SOAP es un formato de mensaje XML utilizado en interacciones de servicios web. Los mensajes SOAP habitualmente se envían sobre HTTP o JMS, pero se pueden utilizar otros protocolos. El uso de SOAP en un servicio web específico se describe mediante la definición WSDL.

Hay dos versiones de SOAP en uso común: SOAP 1.1 y SOAP 1.2. Ambas están soportadas en WebSphere Message Broker. SOAP está definido en los siguientes documentos emitidos por World Wide Web Consortium (W3C):

Simple Object Access Protocol (SOAP) 1.1 (nota de W3C).

SOAP Versión 1.2 Part 0: Primer (recomendación de W3C).

SOAP Versión 1.2 Part 1: Messaging Framework (recomendación de W3C).

SOAP Versión 1.2 Part 2: Adjuncts (recomendación de W3C).

El soporte para SOAP en WebSphere Message Broker incluye:

Analizador y dominio SOAP. Consulte el apartado Analizador y dominio SOAP.

Nodos SOAP para enviar y recibir mensajes en formato SOAP.

Definiciones de mensajes suministrados por IBM® para SOAP 1.1 y SOAP 1.2. Estas definiciones de mensajes dan soporte a la validación, a la ayuda de contenido ESQ y a la creación de correlaciones de mensajes para su uso con

mensajes SOAP, en SOAP y otros dominios XML. Consulte Conjuntos de mensajes: mensajes proporcionados por IBM que puede importar.” (IBM, 2019)

B. Joan Vila(Departament d’Informàtica de Sistemes i Computadors - Universitat Politècnica de València)

https://poliformat.upv.es/access/content/group/OCW_6069_2008/T4.-

[%20Integraci%C3%B3n%20de%20aplicaciones%20en%20la%20web/Teor%C3%ADa/soap.pdf](#) nos dice:

“– Es un estándar basado en XML para la mayoría de servicios web públicos

– Usa un lenguaje de definición de servicios denominado WSDL (análogo a IDL), donde se definen los servicios a publicar, los tipos de datos a enviar y retornar. El WSDL es como la URL en XML-RPC.

- *WSDL: Web Services Definition Language*
- *<http://www.w3.org/TR/2001/NOTE-wsdl-20010315>*

– Existe un directorio de servicios web para descubrir de forma dinámica nuevos servicios:

- *UDDI: Universal Description Directory Integration*
- *Estructura de un mensaje SOAP*

– El envoltorio no contiene datos de aplicación, solo empaqueta la cabecera opcional puede contener información de control

– Contenido.

– El cuerpo contiene los datos en XML

- Los datos adjuntos (attachments) pueden contener otros tipos de datos (binarios, uuencoded, etc.)” (Vila, 2019)

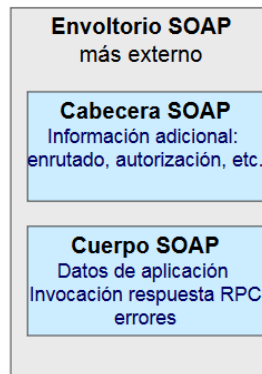


Imagen No. 1 Estructura SOAP

Fuente: https://poliformat.upv.es/access/content/group/OCW_6069_2008/T4.-%20Integraci%C3%B3n%20de%20aplicaciones%20en%20la%20web/Teor%C3%ADa/soap.pdf

Ejemplo simple de Vila:

- Servicio de seguimiento de vuelos web
- Llamada con código de portador y número de vuelo:
 - `public Date getFlightArrival(String carrier, int number)`
- Retorna hora estimada de llegada:
 - `Arrival for AS 25 is Sat Jun 15 00:32:19 PDT 2002”`

En esta misma Fuente (Vila, 2019) nos comenta respecto al formato de la petición SOAP:

```
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle=
"http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
<getFlightArrival>
<op1 xsi:type="xsd:string">AS</op1>
<op2 xsi:type="xsd:int">25</op2>
</getFlightArrival>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Vila también nos detalla el formato de la respuesta SOAP:

```
<SOAP-ENV:Envelope
xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Body>
<getFlightArrivalResponse SOAP-ENV:encodingStyle=
"http://schemas.xmlsoap.org/soap/encoding/">
<getFlightArrivalResult xsi:type="xsd:dateTime">2002-06-
15T07:32:19.403Z
</getFlightArrivalResult>
</getFlightArrivalResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

En <http://www.v-espino.com/~chema/daw2/soap/webservices.htm>
encontramos lo siguiente:

A. S.O.A.P.

"SOAP (siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo deriva de un protocolo creado por David Winer en 1998, llamado XML-RPC. SOAP fue creado por Microsoft, IBM y otros y está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los servicios Web.

SOAP puede formar la capa base de una 'pila de protocolos de web service', ofreciendo un framework de mensajería básica en la cual la web services se puedan construir. Este protocolo basado en XML consiste de tres partes: un sobre (envelope), el cual define qué hay en el mensaje y cómo procesarlo; un conjunto de reglas de codificación para expresar instancias de tipos de datos; y una convención para representar llamadas a procedimientos y respuestas. El protocolo SOAP tiene tres características principales:

Extensibilidad (seguridad y WS-routing son extensiones aplicadas en el desarrollo).

Neutralidad (SOAP puede ser utilizado sobre cualquier protocolo de transporte como HTTP, SMTP, TCP o JMS).

Independencia (SOAP permite cualquier modelo de programación).

Como ejemplo de cómo los procedimientos SOAP pueden ser utilizados, un mensaje SOAP podría ser enviado a un sitio Web que tiene habilitado Web service, para realizar la búsqueda de algún precio en una base de datos, indicando los parámetros necesarios en la consulta. El sitio podría retornar un documento formateado en XML con el resultado, ejemplo, precios, localización, características. Teniendo los datos de respuesta en un formato estandarizado 'parseable', este puede ser integrado directamente en un sitio Web o aplicación externa.

La arquitectura SOAP consiste de muchas capas de especificación: para el formato del mensaje, MEP (Message Exchange Patterns), subyacentes enlaces de protocolo de transporte, modelo de procesamiento de mensajes, y extensibilidad del protocolo. SOAP es el sucesor de XML-RPC, a pesar de que toma el transporte y la neutralidad de la interacción y el envelope / header / body de otra parte (probablemente de WDDX).” (Espino, 2019)

Para enlazar PHP con SOAP se requiere que SOAP esté activo. A través de `phpinfo()` conocemos o nos informamos si tiene soporte para clientes y servidores SOAP.

B. WSDL

“El intercambio de datos se puede hacer mediante un fichero WSDL (Web Services Description Language) un formato XML que se utiliza para describir servicios Web o sin ese fichero.

a. SOAP sin WSDL

- *Se crea una clase con los métodos que se van a ofrecer como servicios `soap_sin_clase.php` (Nota: cuidado que no haya ningún espacio antes de la etiqueta `<?php` en el servidor)*

```
<?php  
class clase {  
    public function multiplica($x,$y) { return $x * $y; }  
    public function suma($x,$y){ return $x+$y; }  
}  
?>
```

- *Se crea el servidor SOAP que ofrece la clase anterior `soap_sin_servidor.php`*

```
<?php  
require_once 'clase.php';  
try {  
    $soap_server = new  
    SoapServer(null,array('uri'=>'urn:webservices'));  
    $soap_server->setClass("clase");  
    $soap_server->handle();  
}
```

```
/*  
    $functions = $soap_server->getFunctions();  
    echo 'Este servidor maneja las siguientes funciones: <br>';  
    foreach($functions as $func) {  
        echo $func . "<br>";  
    }  
*/  
} catch (Exception $e) {  
    echo 'Error --> '. $e->getMessage();  
}  
?>
```

- **Cliente SOAP sin WSDL**

Accede al servidor SOAP y después hace uso de los métodos de la clase como si fueran locales

```
soap_sin_cliente.php  
<?php  
try{  
    $client = @new SoapClient(null,array('location' =>  
        'http://127.0.0.1/soap_server.php','uri' => 'urn:webservices'));  
    $respuesta1=$client->multiplica(4,5);  
    $respuesta2=$client->suma(4,5);  
    echo $respuesta1.' '.$respuesta2;  
} catch (Exception $e) {  
    echo 'Error --> '. $e->getMessage();  
}  
?>
```

b. Acceso a web services públicos (hacerse cliente de web services conocidos)

soap_publico.php

```
<?php
// Acceso a los datos meteorológicos de las grandes ciudades
$parametros = array(
'CityName' => 'Zaragoza',
'CountryName' => 'Spain'
);
$cliente = new
SoapClient('http://www.webservicex.net/globalweather.asmx?W
SDL');
$respuesta = $cliente->GetWeather($parametros);
//print_r($respuesta);
$r=$respuesta->GetWeatherResult;
echo $r.'<br>';

// Acceso a la localizacion de una dirección IP
$client = new
SoapClient('http://www.webservicex.net/geoipservice.asmx?WS
DL');
$result = $client->GetGeoIP(array('IPAddress' =>
'90.165.103.88'));
//print_r($result);
$r=$result->GetGeoIPResult->CountryName;
echo 'Esa direccion esta en '.$r;
?>
```

c. SOAP con WSDL

WSDL es el fichero XML que permite describir y localizar servicios web. Para saber más pulsa [aquí](#). Se crea "a mano" o con alguna de los programas de edición que corren por internet La estructura del WSDL define seis elementos:

tipos de dato: definen los tipos de datos usados en los mensajes. Se utilizan los tipos definidos en la especificación de esquemas XML.

mensajes: definen los elementos de mensaje que forman las comunicaciones. Cada mensaje consiste en una serie de partes lógicas que podrían compararse a los parámetros de la llamada a una función en programación. Un mensaje es una colección de

datos de algún tipo.

tipos de puerto: definen las operaciones permitidas y los mensajes intercambiados. Podría compararse con una librería de funciones o una clase. Una operación podría compararse con una función en programación. Los tipos de puerto funcionan como la especificación de la interfaz software, y se componen de colecciones de operaciones que definen los intercambios de mensajes.

bindings: definen los protocolos de comunicación usados. Los bindings (lazos) indican los protocolos usados por cada puerto.

puertos: definen la dirección de un binding. Un puerto tiene una definición abstracta (tipo de puerto) y una definición concreta (binding, lazo).

d. El archivo WSDL.

soap_con_wsdl.wsdl. CUIDADO con la dirección que aparece al final del mismo, debe apuntar al servidor SOAP

```
<?xml version="1.0"?>
```

```
<definitions name="Servicio"
```

```
targetNamespace="urn:Aritmetica"
```

```
xmlns:tns="urn:Aritmetica"
```

```
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
```

```
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
```

```
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
```

```
xmlns="http://schemas.xmlsoap.org/wsdl/">
```

```
<message name="Entrada">
```

```
<part name="operando1" type="xsd:float" />
```

```
<part name="operando2" type="xsd:float" />
```

```
</message>
```

```
<message name="Salida">
```

```
<part name="respuesta" type="xsd:float" />
```

```
</message>
```

```
<portType name="Operaciones">
```

```
<operation name="sumar">
```

```
<input message="tns:Entrada" />
```

```
<output message="tns:Salida" />
```

```
</operation>
```

```
<operation name="restar">
```

```
<input message="tns:Entrada" />
```

```
<output message="tns:Salida" />
```

```
</operation>
```

```
</portType>
```

```
<binding name="Enlace" type="tns:Operaciones">
```

```
<soap:binding style="rpc"
```

```
transport="http://schemas.xmlsoap.org/soap/http" />
```

```
<operation name="sumar">
```

```
<soap:operation soapAction="urn:Aritmetica#sumar" />
```

```
<input>
```

```
<soap:body use="encoded" namespace="urn:Aritmetica"
```

```
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
```

```
</input>
```

```
<output>
```

```
<soap:body use="encoded" namespace="urn:Aritmetica"
```

```
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
```

```
</output>
```

```
</operation>
```

```
<operation name="restar">
```

```
<soap:operation soapAction="urn:Aritmetica#restar" />
```

```
<input>
```

```
<soap:body use="encoded" namespace="urn:Aritmetica"
```

```
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
```

```
</input>
```

```
<output>
```

```
<soap:body use="encoded" namespace="urn:Aritmetica"
```

```
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
```

```
</output>
```

```
</operation>
```

```
</binding>
```

```
<service name="Servicio">
```

```
<port name="Operaciones" binding="tns:Enlace">
```

```
<soap:address
```

```
location="http://192.168.1.13/soap/soap_con_servidor.php" />
</port>
</service>

</definitions>
```

C. El servidor de objetos SOAP

soap_con_servidor.php

```
<?php
$soap_servidor = new
SoapServer("http://192.168.1.13/soap/soap_con_wsdl.wsdl");
function sumar($operando1,$operando2){ return
$operando1+$operando2; }
function restar($operando1,$operando2){ return $operando1-
$operando2; }
$soap_servidor->AddFunction("sumar");
$soap_servidor->AddFunction("restar");
$soap_servidor->handle();
?>
```

D. El cliente de objetos SOAP

soap_con_cliente.php

```
<?php
try{
$soap_cliente = new
SoapClient('http://192.168.1.13/soap/soap_con_wsdl.wsdl');
$resultado_suma = $soap_cliente->sumar(2.7, 3.5);
$resultado_resta = $soap_cliente->restar(2.7, 3.5);
echo $resultado_suma." --- ".$resultado_resta;
} catch (Exception $e) {
echo 'Error --> '. $e->getMessage();
}
?>
```

El cliente soap puede "ver " las cabeceras y el cuerpo de los mensajes SOAP haciendo una conexion con seguimiento. Algo parecido a:

```
$soap_cliente2 = new
SoapClient('http://127.0.0.1/soap/soap_con_wsdl.wsdl',array('trac
e' => 1));
$resultado2 = $soap_cliente2->sumar(4, 5);
echo "<br>".$resultado2."<br>";
echo $soap_cliente2->__getLastRequestHeaders(). "<br>"; //
Devuelve los encabezados SOAP de la última petición
echo $soap_cliente2->__getLastRequest() . "<br>"; // Ultima
petición SOAP
echo $soap_cliente2->__getLastResponseHeaders(). "<br>"; //
Devuelve los encabezados SOAP de la última respuesta
echo $soap_cliente2->__getLastResponse(). "<br>"; //
Devuelve la última respuesta SOAP
```

2.2.2. PHP

En la fuente oficial de PHP <https://www.php.net/manual/es/intro-whatis.php> nos indica lo siguiente:

“¿Qué es PHP?”

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

¿Qué significa realmente? Un ejemplo nos aclarará las cosas:

Ejemplo #1 Un ejemplo introductorio

```
<!DOCTYPE html>
<html>
<head>
<title>Ejemplo</title>
</head>
<body>

<?php
echo "¡Hola, soy un script de PHP!";
?>

</body>
```

`</html>`

En lugar de usar muchos comandos para mostrar HTML (como en C o en Perl), las páginas de PHP contienen HTML con código incrustado que hace 'algo' (en este caso, mostrar '¡Hola, soy un script de PHP!'). El código de PHP está encerrado entre las etiquetas especiales de comienzo y final `<?php` y `?>` que permiten entrar y salir del 'modo PHP'.

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales. No sienta miedo de leer la larga lista de características de PHP. En unas pocas horas podrá empezar a escribir sus primeros scripts. Aunque el desarrollo de PHP está centrado en la programación de scripts del lado del servidor, se puede utilizar para muchas otras cosas.

¿Qué puede hacer PHP?

Cualquier cosa. PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo que se puede hacer cualquier cosa que pueda hacer otro programa CGI, como recopilar datos de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies. Aunque PHP puede hacer mucho más.

Existen principalmente tres campos principales donde se usan scripts de PHP.

Scripts del lado del servidor. Este es el campo más tradicional y el foco principal. Son necesarias tres cosas para que esto funcione: el analizador de PHP (módulo CGI o servidor), un servidor web y un navegador web. Es necesario ejecutar el servidor con una instalación de PHP conectada. Se puede acceder al resultado del programa de PHP con un navegador, viendo

la página de PHP a través del servidor. Todo esto se puede ejecutar en su máquina si está experimentado con la programación de PHP. Véase la sección sobre las instrucciones de instalación para más información.

*Scripts desde la línea de comandos. Se puede crear un script de PHP y ejecutarlo sin necesidad de un servidor o navegador. Solamente es necesario el analizador de PHP para utilizarlo de esta manera. Este tipo de uso es ideal para scripts que se ejecuten con regularidad empleando cron (en *nix o Linux) o el Planificador de tareas (en Windows). Estos scripts también pueden usarse para tareas simples de procesamiento de texto. Véase la sección Uso de PHP en la línea de comandos para más información.*

Escribir aplicaciones de escritorio. Probablemente PHP no sea el lenguaje más apropiado para crear aplicaciones de escritorio con una interfaz gráfica de usuario, pero si se conoce bien PHP, y se quisiera utilizar algunas características avanzadas de PHP en aplicaciones del lado del cliente, se puede utilizar PHP-GTK para escribir dichos programas. También es posible de esta manera escribir aplicaciones independientes de una plataforma. PHP-GTK es una extensión de PHP, no disponible en la distribución principal. Si está interesado en PHP-GTK, puede visitar su propio» sitio web.

PHP puede emplearse en todos los sistemas operativos principales, incluyendo Linux, muchas variantes de Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, macOS, RISC OS y probablemente otros más. PHP admite la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros. Esto incluye cualquier servidor web que pueda utilizar el binario de PHP FastCGI, como lighttpd y nginx. PHP funciona tanto como módulo como procesador de CGI.

De modo que, con PHP, se tiene la libertad de elegir el sistema operativo y el servidor web. Además, se tiene la posibilidad de utilizar programación por procedimientos o programación orientada a objetos (POO), o una mezcla de ambas.

Con PHP no se está limitado a generar HTML. Entre las capacidades de PHP se incluyen la creación de imágenes, ficheros PDF e incluso películas Flash (usando libswf y Ming) generadas sobre la marcha. También se puede generar fácilmente cualquier tipo de texto, como XHTML y cualquier otro tipo de fichero XML. PHP puede autogenerar estos ficheros y guardarlos en el

sistema de ficheros en vez de imprimirlos en pantalla, creando una caché en el lado del servidor para contenido dinámico.

Una de las características más potentes y destacables de PHP es su soporte para un amplio abanico de bases de datos. Escribir una página web con acceso a una base de datos es increíblemente simple utilizando una de las extensiones específicas de bases de datos (p.ej., para mysql), o utilizar una capa de abstracción como PDO, o conectarse a cualquier base de datos que admita el estándar de Conexión Abierta a Bases de Datos por medio de la extensión ODBC. Otras bases de datos podrían utilizar cURL o sockets, como lo hace CouchDB.

PHP también cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros. También se pueden crear sockets de red puros e interactuar usando cualquier otro protocolo. PHP tiene soporte para el intercambio de datos complejos de WDDX entre virtualmente todos los lenguajes de programación web. Y hablando de interconexión, PHP tiene soporte para la instalación de objetos de Java y emplearlos de forma transparente como objetos de PHP.

PHP tiene útiles características de procesamiento de texto, las cuales incluyen las expresiones regulares compatibles con Perl (PCRE), y muchas extensiones y herramientas para el acceso y análisis de documentos XML. PHP estandariza todas las extensiones XML sobre el fundamento sólido de libxml2, y amplía este conjunto de características añadiendo soporte para SimpleXML, XMLReader y XMLWriter.

Existen otras extensiones interesantes, las cuales están categorizadas alfabéticamente y por categoría.” (Php, 2019)

2.2.3. Visual Basic Net.

En la fuente registrada sobre la definición de Visual Basic Net <http://repositorio.espm.edu.ec/bitstream/42000/86/1/TESIS%20L%C3%93PEZ%20MORA%20WILSON%20FABI%C3%81N%20-.pdf> nos dice:

“Visual Basic.NET es la nueva generación del lenguaje Visual Basic de Microsoft. Con Visual Basic puede generar aplicaciones .NET, incluyendo

servicios Web y aplicaciones Web de ASP.NET, rápida y fácilmente. Las aplicaciones construidas con Visual Basic se generan con los servicios de CommonLanguageRuntime y se benefician de .NET Framework. Visual Basic tiene una gran cantidad de características nuevas y mejoradas como la herencia, las interfaces y la sobrecarga que la convierten en un lenguaje eficaz de programación orientado a objetos. Visual Basic integra .NET Framework y CommonLanguageRuntime completamente, que conjuntamente proporcionan interoperabilidad del lenguaje, recolección de elementos no utilizados, seguridad ampliada y compatibilidad de versiones mejoradas. Visual Basic proporciona prototipos de algunos de los tipos de proyectos más comunes, incluyendo: Aplicación para Windows, Biblioteca de clases, Biblioteca de control de Windows, Aplicación Web ASP.NET, Servicio Web ASP.NET, Biblioteca de control Web, Aplicación de consola, Servicio de Windows. Visual Basic.NET crea aplicaciones de servicio Web XML. Para este propósito, Visual Basic.NET genera código administrado para CommonLanguageRuntime. Esto supone cambios en el propio lenguaje. Los cambios en Visual Basic.NET están proyectados para:

- Simplificar el lenguaje y hacerlo más coherente.*
- Agregar nuevas características solicitadas por usuarios.*
- Hacer el código más sencillo de leer y mantener.*
- Ayudar a los programadores a evitar los errores de programación.*
- Crear aplicaciones más sólidas y más sencillas de depurar.” (Espam, 2019)*

2.2.4. Postgresql.

En la fuente oficial de <https://www.php.net/manual/es/book.pgsql.php> encontramos las siguientes explicaciones y detalles del término en mención:

“La base de datos PostgreSQL es un producto de código abierto disponible sin costo alguno. Postgres, desarrollado originalmente en el Departamento de Informática de la Universidad de California, Berkeley, fue pionera en muchos de los conceptos objeto-relacionales que actualmente están disponibles en algunas bases de datos comerciales. Proporciona soporte para el lenguaje SQL92/SQL99, transacciones, integridad referencial, procedimientos almacenados y extensibilidad de tipos. PostgreSQL es un descendiente de código fuente abierto del código original de Berkeley.

Requerimientos

Para usar el soporte PostgreSQL, necesita PostgreSQL 6.5 o superior, PostgreSQL 8.0 o superior para habilitar todas las características del modulo PostgreSQL. PostgreSQL soporta muchas codificaciones de caracteres incluyendo codificación de caracteres multibyte. La versión actual y más información sobre PostgreSQL está disponible en» <http://www.postgresql.org/> y la » Documentación de PostgreSQL.

Instalación

Con el fin de habilitar el soporte de PostgreSQL, --with-pgsql[=DIR] es requerido cuando se compila PHP. DIR es el directorio base donde está instalado PostgreSQL, por defecto generalmente es /usr/local/pgsql en sistemas linux. Si el módulo de objetos compartidos está disponible, el módulo PostgreSQL puede ser cargado usando la directiva extension en php.ini o la función dl().” (PostgreSql, 2019)

2.2.5. NUBEFACT. (<https://www.nubefact.com>) Se escribe lo que indica la propia empresa en su portal web:

A. *“Definición: NUBEFACT S.A. con RUC 20600695771 es un Proveedor autorizado por la SUNAT para generar: Facturas electrónicas, boletas de ventas electrónica y las notas electrónicas (crédito y débito) conforme lo establece el artículo 1° de la Resolución de Superintendencia N° 097-2012/SUNAT y modificatorias.*

Para la empresa Nuberfact no le importa el lenguaje de programación que se utilizó para implementar la aplicación o módulo. Para ello se puede usar la propia API REST de la OSE para poder emitir documentos electrónicos desde el propio sistema. Se recibe el TRAMA o el ARCHIVO en TXT o JSON y la empresa genera el PDF, el XML, se envía a la Sunat y se almacena la CDR, entre otros procedimientos. El CDR Sunat es la constancia de recepción de la factura electrónica remitida a la Sunat.

B. Versiones De Nubefact Para Usar Nuestra Api.

Esta documentación es para ambas versiones a menos que se indique lo contrario.

- I. VERSIÓN ONLINE: Puede integrar la aplicación usando nuestra WEB SERVICE (API REST). Consumes directamente nuestra WEB SERVICE publicada en www.nubefact.com. Es necesario contar con internet para generar el documento.*
- II. VERSIÓN OFFLINE: Puedes integrar tu aplicación usando nuestra WEB SERVICE (API REST). Instalamos nuestra aplicación en el servidor o PC de la empresa para que consumas nuestra WEB SERVICE vía LOCALHOST. NO es necesario contar con internet para generar el documento.*
- III. VERSIÓN RESELLER: Puedes integrar tu aplicación usando nuestra WEB SERVICE (API REST). Consumes directamente nuestra WEB SERVICE publicada en tu subdominio <https://xxxxxxxx.pse.pe> ingresando a las cuentas de tus clientes y obteniendo la ruta y token.*

C. FORMATOS O ARCHIVOS SOPORTADOS

Actualmente se soporta archivos JSON y TXT, tanto para la solicitud como para la respuesta.

- I. MANUAL para archivo JSON en el link: <https://goo.gl/WHMmSb>*
- II. MANUAL para archivo TXT en el link: <https://goo.gl/Lz7hAq>*

D. AUTENTICACIÓN

El servicio está protegido para evitar intrusos. Lo primero que debe saber es que para usar nuestra API debes autenticarte o loguearte. Para ello necesitas DOS datos: una RUTA y un TOKEN.

- I. RUTA: Cada cliente tiene una RUTA distinta y única.*

EJEMPLO (Versión ONLINE):

<https://www.nubefact.com/api/v1/48239908-7ae7-4353-824d-071765d4>

EJEMPLO (Versión OFFLINE):

<http://localhost:8000/api/v1/48239908-7ae7-4353-824d-071765d4>

II. TOKEN: Puedes tener uno o varios TOKEN.

EJEMPLO:

1c4239064a3f441880d7ced75eea4383b831c0bf26944
169b

Estos datos los podemos conseguir en tu cuenta en NUBEFACT.

E. FORMA DE TRABAJO

- I. Conseguir una RUTA y un TOKEN para trabajar con NUBEFACT (Regístrate o ingresa a tu cuenta en www.nubefact.com).*
- II. Generar un archivo en formato. JSON o .TXT con una estructura que se detalla en este documento.*
- III. Enviar el archivo generado a nuestra WEB SERVICE ONLINE u OFFLINE según corresponda usando la RUTA y el TOKEN.*
- IV. Nosotros a partir de ese archivo generamos el archivo XML y PDF (Según especificaciones de la SUNAT).*
- V. Te devolveremos inmediatamente como respuesta los datos para que imprimas el documento para tu cliente o si lo deseas te enviaremos el PDF y XML generado. La impresión del documento seguirá haciéndose desde tu sistema. Enviaremos el documento por email a tu cliente si así lo indicas en el archivo JSON o TXT. Para ver el documento generado ingresa a www.nubefact.com/login con tus datos de acceso, y luego a la opción "Ver Facturas, Boletas y Notas".*
- VI. Enviaremos el XML generado a la SUNAT y lo almacenaremos junto con el PDF, XML y CDR en la NUBE para que tu cliente pueda consultarlo en cualquier momento, si así lo desea." (Nubefact,2019)*

F. CONSTRUIR UN ARCHIVO JSON

Recomendamos construir el archivo JSON teniendo en cuenta las especificaciones indicadas en <http://json.org>.

“El archivo debe ser construido de tal forma que:

- I. Sea construido en codificación UTF-8.*
- II. No tenga caracteres especiales escondidos en una codificación distinta a UTF-8 o espacios innecesarios.*
- III. Que no tenga comillas dobles (") en los atributos, en la descripción, denominación, dirección, ETC, ejemplo: “clavos 3” pulgadas” (esto interrumpe la estructura del json), se deben usar un escape, como, por ejemplo: “clavos 3\” pulgadas” o eliminar estas comillas.*
- IV. La suma de cada línea del comprobante debe coincidir con los totales.*

G. TIPOS DE DATOS.

TIPO	DESCRIPCIÓN
Integer	Valor entero
String	Valor cadena. Texto de 0 a 250 caracteres.
Text	Valor cadena. Texto de 0 a 1000 caracteres.
Numeric	Valor numérico entero y fraccionario.
Date	Valor fecha dd-mm-yyyy
Boolean	Valor lógico: verdadero, falso.

Imagen No. 2 Tipos de datos en Json

Fuente: Propia autoría

Se pueden hacer 4 tipos de operaciones con nuestra API:

- 1. OPERACIÓN 1: GENERAR FACTURAS, BOLETAS Y NOTAS*
- 2. OPERACIÓN 2: CONSULTA DE FACTURAS, BOLETAS Y NOTAS*
- 3. OPERACIÓN 3: GENERAR ANULACIÓN DE FACTURAS, BOLETAS Y NOTAS*
- 4. OPERACIÓN 4: CONSULTA ANULACIÓN DE FACTURAS, BOLETAS Y NOTAS” (Json, 2019)*

2.2.6. OPERADOR DE SERVICIOS ELECTRÓNICOS(OSE)

En el vínculo que nos vincula a la sección de la SUNAT http://cpe.sunat.gob.pe/informacion_general/operador_servicios_electronicos nos explica lo relacionado al Sistema de Emisión Operador de Servicios Electrónicos:

A. Concepto

“Es parte del sistema de emisión electrónica, que comprueba de manera informática el cumplimiento de los aspectos esenciales para que se considere emitido el documento electrónico que sirve de soporte a los comprobantes de pago electrónicos, a los documentos relacionados, tales como: factura, boleta de venta, notas de crédito, notas de débito, comprobante de retenciones, comprobante de percepciones, guía de remisión, recibos de servicios público.

El emisor desde sus propios sistemas de emisión, contratará los servicios de un Operador de Servicios Electrónicos (SEE- OSE) para la validación de sus comprobantes.

Características:

- *La emisión se realiza desde los sistemas desarrollados por el contribuyente al OSE contratado.*
- *La serie de los comprobantes de pago es alfanumérica, inicia con la letra, y su numeración es correlativa, inicia en 1.*
- *A través del Portal web de la SUNAT, se puede realizar la verificación de autenticidad del archivo digital enviado desde los sistemas del contribuyente a la SUNAT.*

B. Requisitos para ser emisor electrónico

No tener, para efectos del número de RUC, la condición de domicilio fiscal no habido.

No tener en el RUC el estado de suspensión temporal de actividades o baja de inscripción.

Encontrarse afecto en el RUC al impuesto a la renta de tercera categoría, de generar ese tipo de renta, o al Nuevo Régimen Único Simplificado, de corresponder.

¿Cómo se obtiene la calidad de emisor electrónico?

La calidad de emisor electrónico en este Sistema se obtiene por designación de la SUNAT o por elección del contribuyente.

Para ello deberá realizar el siguiente procedimiento:

- *Indicar a través de SUNAT Operaciones en Línea*
- *Cargar un certificado digital y un correo electrónico,*
- *Vincular o autorizar a 1 o más operadores de servicios electrónicos - OSE*

Cabe señalar que dicha autorización surte efecto desde el día calendario siguiente a aquél en que se efectuó el registro.

Excepción: DAE. la calidad se obtiene con la primera emisión

Requisitos para emitir comprobantes electrónicos

- *Para que puedan ser emitidos el contribuyente deberá:*
- *Contar con Clave SOL y haber realizado el procedimiento descrito para obtener la calidad de emisor electrónico.*
- *Tener la condición de domicilio fiscal habido para efectos del RUC.*
- *No encontrarse en estado de suspensión temporal de actividades o de baja de Inscripción.*

C. Tipos de comprobantes a emitir

Se pueden emitir los siguientes comprobantes de pago electrónico, documentos relacionados y otros comprobantes:

- *Factura.*
- *Boleta de venta (Individual o Resumen diario).*
- *Notas de crédito • Notas de débito.*
- *Comprobante de retenciones.*
- *Comprobante de percepciones.*
- *Guía de remisión.*
- *Recibo de Servicio Público.*

D. Otorgamiento

Boletas y notas vinculadas:

- *Entrega o pone a disposición de adquirente o usuario electrónico – medios electrónicos.*
- *Entrega o pone a disposición de adquirente o usuario no electrónico - mediante una representación impresa o medios electrónicos si así lo elige.*

- *Factura y notas vinculadas: Desde que lo entrega o pone a disposición del adquiriente o usuario electrónico o no electrónico, a través de medios electrónicos.*

E. Operatividad

- *Los comprobantes se emiten considerando el formato electrónico establecido por la SUNAT.*
- *El emisor realiza el envío al OSE del XML con los datos del comprobante de pago emitido dentro del plazo establecido (desde el día siguiente de la emisión, hasta 7 días calendarios posteriores)*
- *El OSE debe realizar la comprobación informática de las condiciones de emisión de los documentos que el emisor le envíe, de cumplir con las condiciones el OSE le enviará:*

a. Constancia de Recepción – CDR, la cual puede tener los siguientes estados:

- *Aceptada: Si el comprobante de pago electrónico cumple con las validaciones establecidas. En este caso, el comprobante adquiere total validez tributaria.*
- *Aceptada con observación: Cuando el comprobante de pago electrónico cumple con las validaciones establecidas y, por lo tanto, ya tiene validez tributaria, pero hay datos en el comprobante que, producto de una auditoría, podrían ser reparados.*

b. Comunicación de Inconsistencia

- *Rechazada: Si no cumple con las condiciones establecidas. En este caso, el comprobante de pago electrónico que se hubiera emitido no tiene validez tributaria. El emisor tendrá que emitir un nuevo comprobante electrónico corrigiendo los motivos por los cuales fue rechazado.*
- *Posteriormente, el OSE tiene una (1) hora para remitir la información que realizó en la comprobación informática a la SUNAT (XML + CDR).*

“Implementación del Módulo de Envío de Documentos de Ventas del Restaurant Zarandaja de la Ciudad de Chiclayo Consumiendo el servicio SOAP del Operador de Servicios Electrónicos NUBEFAC”

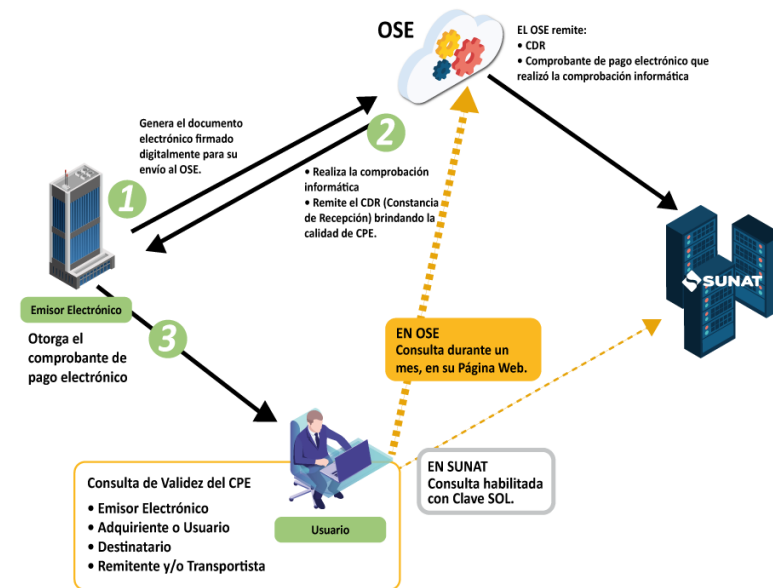


Imagen No. 3Flujo del OSE

Fuente: http://cpe.sunat.gob.pe/informacion_general/operador_servicios_electronicos

Envío a OSE

Puede enviar los comprobantes electrónicos desde la fecha de emisión y hasta un plazo de siete (07) días calendarios, contados desde el día siguiente de la emisión. Si el envío es posterior será rechazado aun cuando hubiere sido entregado al adquiriente o usuario.

Comunicación de baja

El emisor electrónico puede dar de baja la numeración de los documentos no otorgados, si fueron enviados o informados al OSE. A tal efecto, debe considerar lo siguiente:

- a) La baja de la GRE se realiza según la normativa sobre guías electrónicas.
- b) La baja de la factura electrónica, del DAE y de las notas electrónicas vinculadas a aquellos, aun cuando se haya generado respecto de ellos la CDR - comprobantes y nota, se realiza con la comunicación de baja al OSE.

La comunicación de baja puede incluir uno o más documentos, siempre que todos hayan sido generados o emitidos en un mismo día.

En caso lo enviado por el emisor electrónico al OSE no cumpla con la condición antes indicada, el OSE le remite la comunicación de inconsistencias.

La baja de la boleta de venta electrónica o de la nota electrónica vinculada a aquella no otorgada, el emisor electrónico debe incluir la información del documento cuya baja comunica en el formato digital de un resumen diario correspondiente al día de emisión de esa boleta de venta o nota. Ese formato debe ser enviado al OSE de acuerdo a la normatividad, considerando lo siguiente:

- Se puede enviar, como máximo, hasta el séptimo día calendario contado a partir del día calendario siguiente de recibida la CDR - comprobantes y nota o la CDR - resumen diario.

- Tratándose de la boleta de venta electrónica o de las notas electrónicas vinculadas a aquella cuyos ejemplares no se han enviado al OSE o cuya información no se ha incluido en un resumen diario, se puede enviar, a más tardar, hasta el séptimo día calendario contado a partir del día calendario siguiente a la fecha de su generación.

- Se debe indicar, al lado de la información cuya baja se comunica, el estado “anulado”.

Conservación y puesta a disposición del adquirente o usuario

Emisor electrónico: debe almacenar, archivar y conservar.

- 1. Un ejemplar de la boleta de venta electrónica y de la nota electrónica vinculada a aquella cuando opte por enviar a la SUNAT el resumen diario con la información de estas.*
- 2. Un ejemplar del comprobante de pago electrónico y de la nota electrónica que reciba en su calidad de adquirente o usuario electrónico.*

Adquirente o usuario no electrónico: deberá almacenar, archivar y conservar la representación impresa o el ejemplar del comprobante de pago electrónico o la nota electrónica que reciba cuando tenga efecto tributario.

El almacenamiento de los comprobantes de pago electrónicos, notas electrónicas, resúmenes diarios y constancias de rechazo puede ser realizado en medios magnéticos, ópticos, entre otros.

Adicionalmente, el emisor electrónico debe poner a disposición del adquirente o usuario, a través del medio electrónico que él decida, un ejemplar de la boleta de venta electrónica y de la nota electrónica vinculada a aquella que le haya otorgado, cuando no haya optado por enviar sus ejemplares al OSE. El plazo de esa obligación es de un año, contado desde la emisión de esos documentos. Durante ese plazo el adquirente o usuario puede usar ese medio electrónico para leerlos, descargarlos e imprimirlos.

El emisor electrónico debe definir una forma de autenticación que garantice que solo el adquirente o usuario puede acceder a su información, cuando de acuerdo al medio electrónico empleado, ello sea posible.

Consulta de Comprobante de Pago electrónicos

Los comprobantes de pago electrónicos que han sido validados por los Operadores de Servicios Electrónicos (OSE), con constancia de recepción: Aceptado, son válidos para todo efecto.

Los OSE pone a disposición del emisor electrónico, adquirente o usuario, destinatario, remitente y/o transportista la posibilidad de consultar en su página web, el documento electrónico respecto del cual se emitió una CDR y las comunicaciones de inconsistencias que haya emitido. Las OSE deberán mantener por un mes el documento electrónico respecto del cual se emitió el CDR y las comunicaciones de inconsistencias que haya emitido.

La SUNAT mediante SUNAT Virtual pone a disposición del emisor electrónico, el adquirente o usuario, el destinatario, el remitente y/o el transportista, según corresponda, la posibilidad de consultar, para ello deberá contar con Clave SOL.

- *Factura electrónica, el DAE, la nota electrónica vinculada a aquellas y la GRE, la validez, así como la información de las condiciones de emisión y los requisitos mínimos, según corresponda.*
- *Boleta de venta electrónica y la nota electrónica vinculada a aquella, se puede:*
- *Verificar si aquellas fueron emitidas por el emisor electrónico y en qué fecha.*
- *Ver la información relativa a los requisitos mínimos si el emisor electrónico optó por enviar el ejemplar al OSE.*

***Emisores Electronicos del SEE obligados al uso del SEE- OSE y/o
SEE-SOL***

Con fecha 28 de febrero de 2019 se publicó en el Diario Oficial El Peruano la Resolución de Superintendencia N° 044-2019/SUNAT mediante el cual establece el uso obligatorio del Sistema de Emisión Operador de Servicios Electrónicos y/o el Sistema de Emisión Electrónica SUNAT Operaciones en Línea y modifican la Resolución de Superintendencia N° 117-2017/SUNAT.” (Sunat, 2019)

2.3. Definición de términos

Base de datos: Conjunto de estructuras organizadas y vinculadas que contienen datos que son accesados por un gestor de base de datos con las medidas de seguridad de acceso y conservación de los mismos.

Boleta: Documento de venta que se le entrega a un usuario final de la cadena de venta y el cual asume el pago del impuesto general a las ventas.

Consistencia: Cuando los datos almacenados en una tabla son reales y exactos.

Comprobante de pago: Aquel que acredita legal y discalmente una transacción por compra, venta u otros.

Factura: Documento que acredita que una operación se ha realizado entre dos partes, de forma legal y satisfactoria permitiendo además la deducción del impuesto a la venta por no ser usuario final.

Hipótesis: Es la afirmación o negación de una solución ante un problema propuesto

Implementación: Es el desarrollo de un programa en total o en forma modular. Constituye la elaboración del código fuente que el computador procesará.

Integridad: Cuando los datos almacenados en una base de datos pueden ser extraídos o localizados sin dificultades en una tabla de datos.

Lenguaje de programación: Es una conjunta de palabras claves, operadores, reglas y semánticas que permiten escribir código compilable y procesable por un computador.

Modelado: Es la representación visual, gráfica o simbólica de una situación o contexto.

Módulo: Bloque de instrucciones de un programa de computadora que realiza una tarea específica o cumple con una de las funciones en conjunto para alcanzar un objetivo.

OSE: Es la empresa, institución o compañía que se encarga de realizar la comprobación informática del cumplimiento de las condiciones de emisión de los documentos electrónicos enviados por el emisor.

Problema: Es aquello que impulsa el proceso científico y constituye la base de cualquier método de investigación y diseño experimental, desde un experimento verdadero hasta un estudio de caso.

Problemática: Conjunto de incidentes que no permiten lograr o alcanzar metas u objetivos.

Restaurant: Establecimiento en el que se preparan, sirven, venden y distribuyen comidas.

Sistema: Conjunto de elementos que interactúan y ofrecen una solución o cumplen con una función establecida en el mismo entorno donde se desarrolla o existe.

Servicio: Conjunto de acciones que son realizadas para servir a alguien, algo o alguna causa. Proporcionan una respuesta o solución ante una necesidad.

Soap: protocolo estándar que inicialmente permitía la comunicación entre las aplicaciones implementadas en diferentes lenguajes y en diferentes plataformas.

SUNAT: Superintendencia Nacional Tributaria, entidad recaudadora de los impuestos del Estado Peruano.

Tabla de datos: Estructura organizada por columnas y campos que almacenan datos que corresponde a un ente específico.

Variable: Es aquella característica o propiedad que se asume ser la causa del fenómeno o problema estudiado.

Venta: Operación de transacción que involucra el pago mediante una moneda establecida comercialmente y acreditado mediante un documento.

Web: Es el conjunto virtual y enlazadas de páginas por el cual se desplazan en forma libre o restringida usuarios.

Capítulo III:

MARCO METODOLÓGICO

CAPÍTULO III: MARCO METODOLÓGICO

3.1. Variables.

- a) Variable Independiente: Implementación del Módulo OSE del Restaurant Zarandaja De La Ciudad De Chiclayo Consumiendo El Servicio Soap Del Operador De Servicios Electrónicos Nubefact.
- b) Variable dependiente: Validación online en forma oportuna de los documentos de venta.

3.2. Operacionalización de las variables:

Variable	Definición conceptual	Dimensiones	Indicadores
Dependiente	Validación online en forma oportuna de los documentos de venta	Recursos económicos	Reducción de costos
		Rendimiento	Tipo de rendimiento
Independiente	Implementación del Módulo OSE del Restaurant Zarandaja De La Ciudad De Chiclayo Consumiendo El Servicio Soap Del Operador De Servicios Electrónicos Nubefact	Accesibilidad	Seguridad

Tabla 1 Operacionalización de las variables

Fuente: propia autoría

3.3. Metodología

3.3.1. Tipo de estudio:

Investigación aplicada.

3.3.2. Diseño de investigación

- Se define la problemática de la situación a estudiar.
- Se formula el problema, los objetivos y la hipótesis para el trabajo de investigación.
- Se define la variable independiente y dependiente y su operacionalización.
- Se establece la matriz de consistencias.
- Aplicar una encuesta a los actores involucrados antes y después de la implementación de la aplicación.
- Implementar la aplicación
- Evaluar los resultados y establecer el análisis de los datos mediante la discusión de resultados.

3.4. Población, muestra y muestreo.

a. Población:

Trabajadores administrativos y clientes Jurídicos del Restaurant Zarandaja

b. Muestra

60 personas.

c. Muestreo

Muestreo no probalístico

3.5. Técnicas e instrumentos de recolección de datos.

Análisis documental. Se emplearán las fichas textuales y de resumen, considerando como fuentes los informes y documentos.

Encuesta. Se empleará un cuestionario, que será aplicada a los actores involucrados.

3.6. Métodos de análisis de datos

Los datos obtenidos mediante las respuestas a una encuesta serán organizados y procesados en una computadora personal, con el apoyo de programas como Microsoft Excel

3.7. Aspectos éticos

Los datos obtenidos serán el fiel reflejo de lo que se encontró en esta investigación y serán de carácter reservado.

Capítulo IV: RESULTADOS

CAPITULO IV: RESULTADOS

En el presente trabajo de investigación antes de desarrollar la implementación de la aplicación elaboré los diagramas necesarios de análisis y diseño el cual involucra los principales diagramas del UML, así como los diagramas y diccionarios de la base de datos.

A. Diagrama de Casos de uso.

C

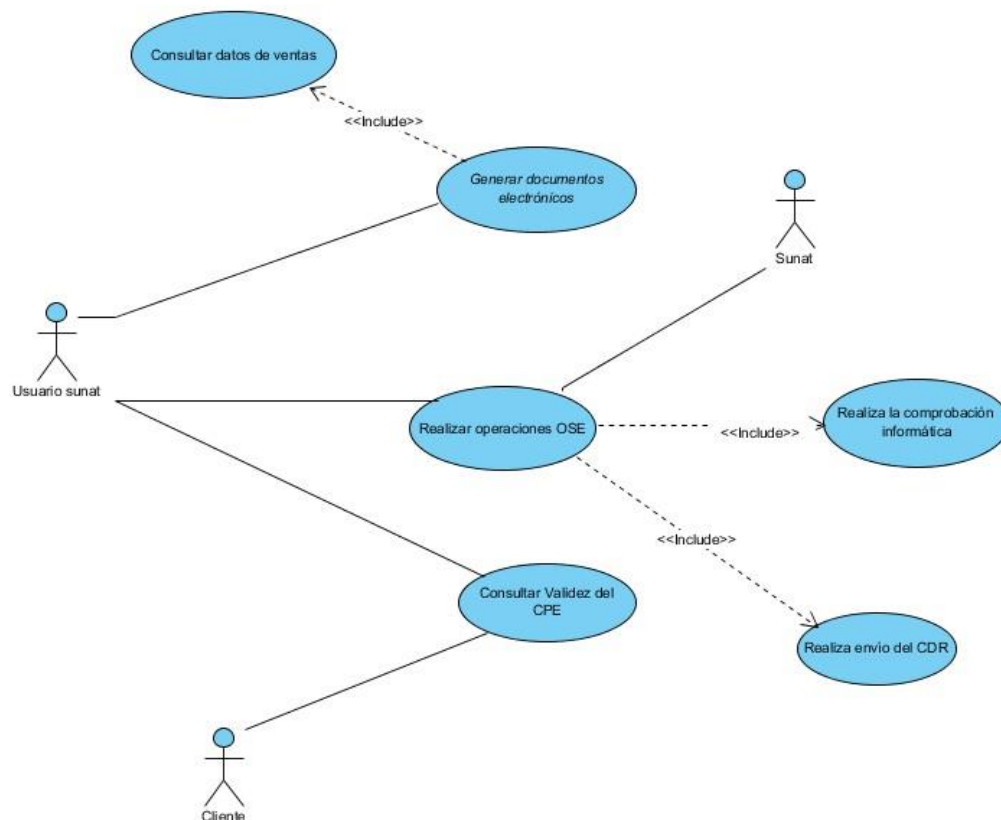


Imagen No. 4 Modelo del Negocio

Fuente: Propia autoría

B. Diagrama de Estados.

1. Comprobante.

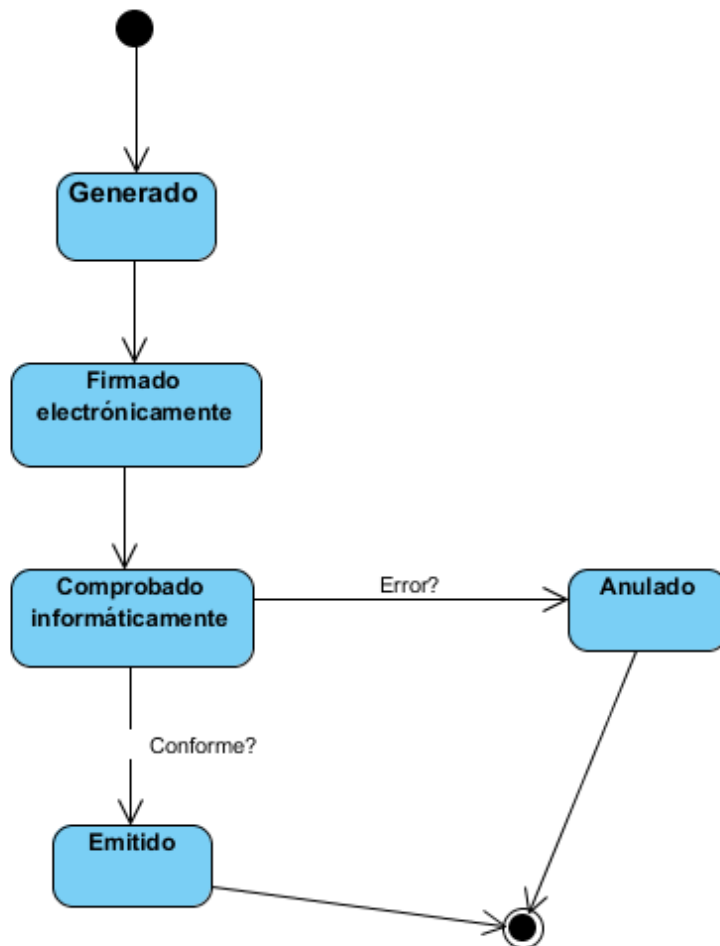


Imagen No. 5 Diagrama de estado Comprobante.

Fuente: Propia autoría

2. CDR SUNAT.

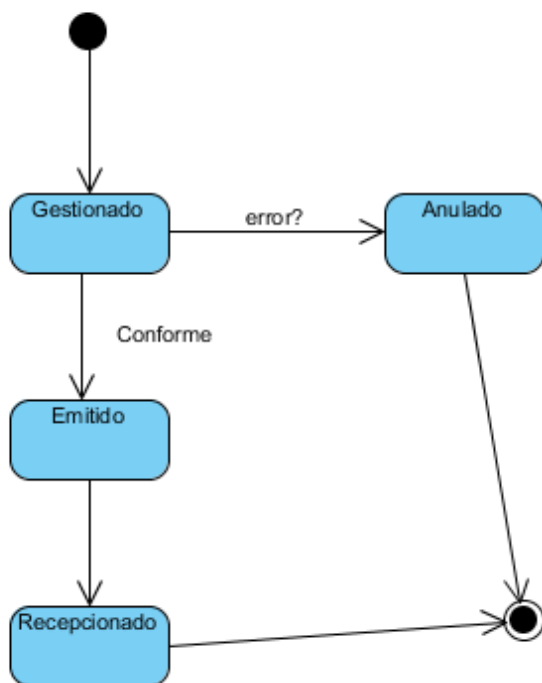


Imagen No. 6. Diagrama de estado CDR Sunat

Fuente: Propia autoría

C. Diagrama de actividades.

1. Consultar documento electrónico.

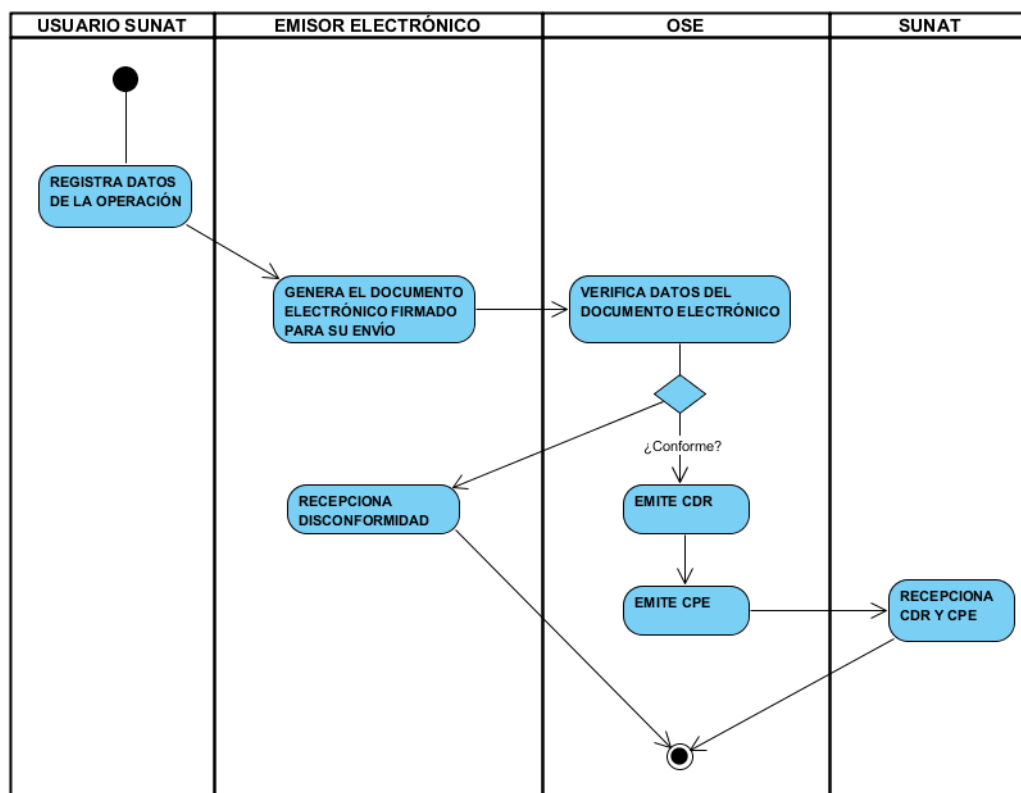


Imagen No. 7 Diagrama de Actividades-Generar documento electrónico

Fuente: Propia autoria

2. Consultar validez del CPE.

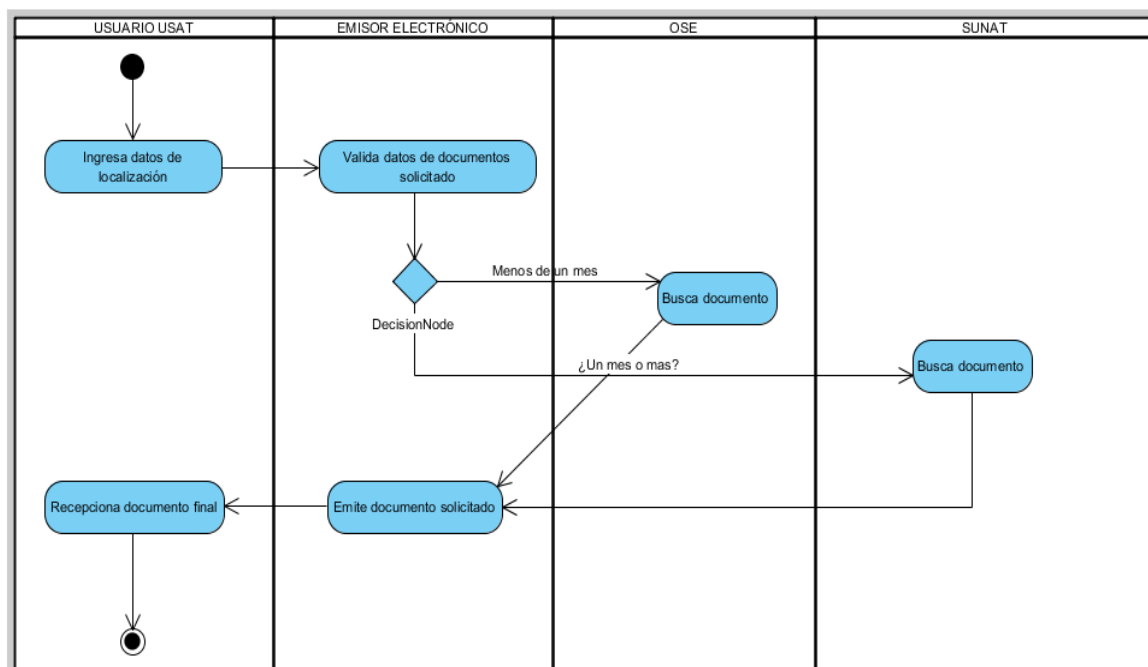


Imagen No. 8 Diagrama de Actividades- Consultar validez del CPE

Fuente: propia Autoria

D. Diagrama de Secuencia.

1. Consultar documento electrónico

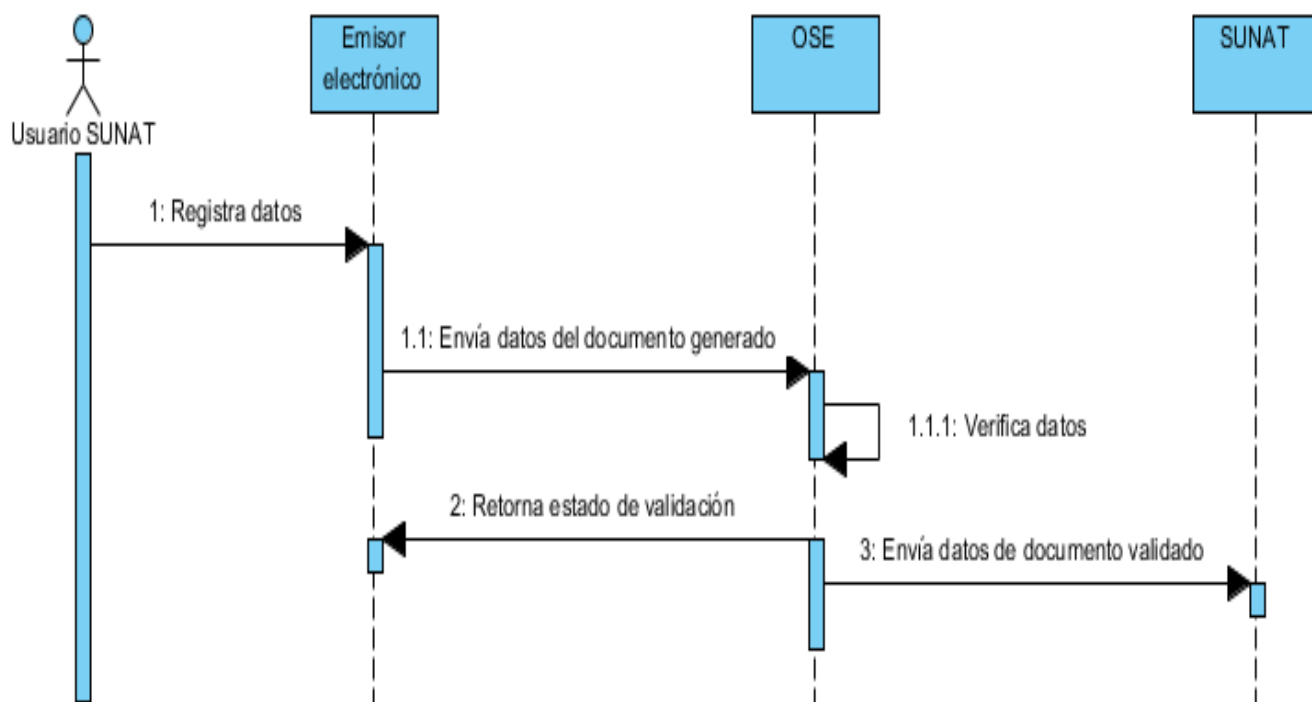


Imagen No. 9 Diagrama de secuencia Consultar documento electrónico

Fuente: Propia Autoria

2. Consultar validez del CPE

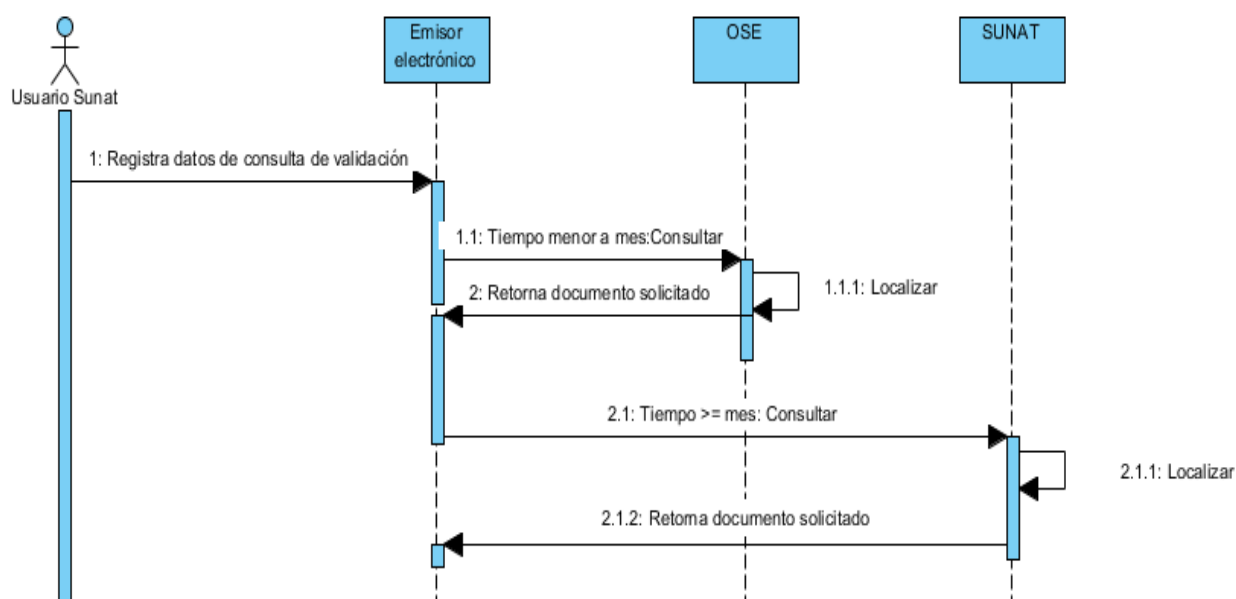


Imagen No. 10 Diagrama de secuencia Consultar validez del CPE

Fuente: Propia Autoria

E. Diseño de la base de datos

1. Diagrama físico de la base de datos.

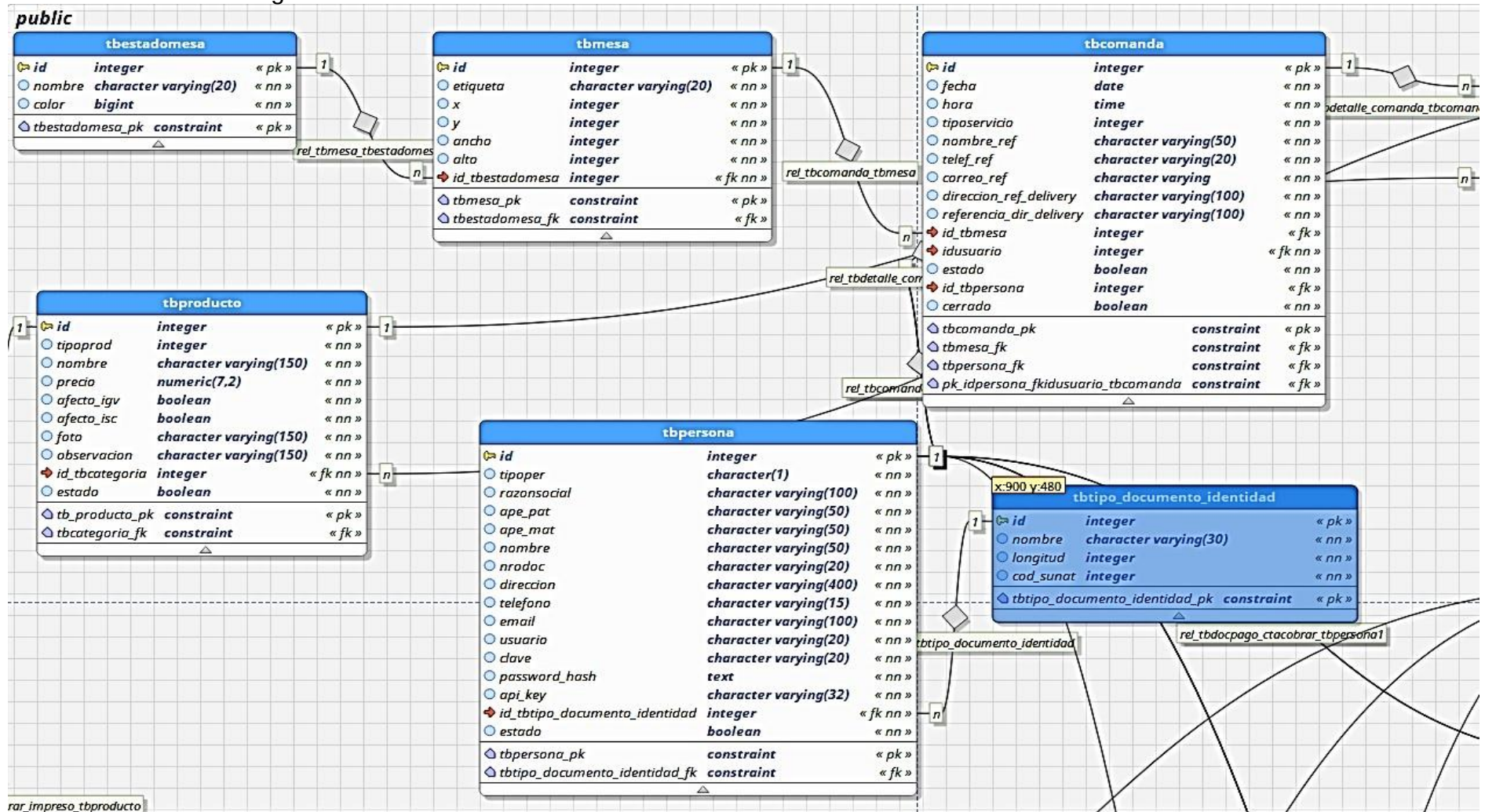


Imagen No. 11 Modelo relacional parte 1.

Fuente: Propia Autoría

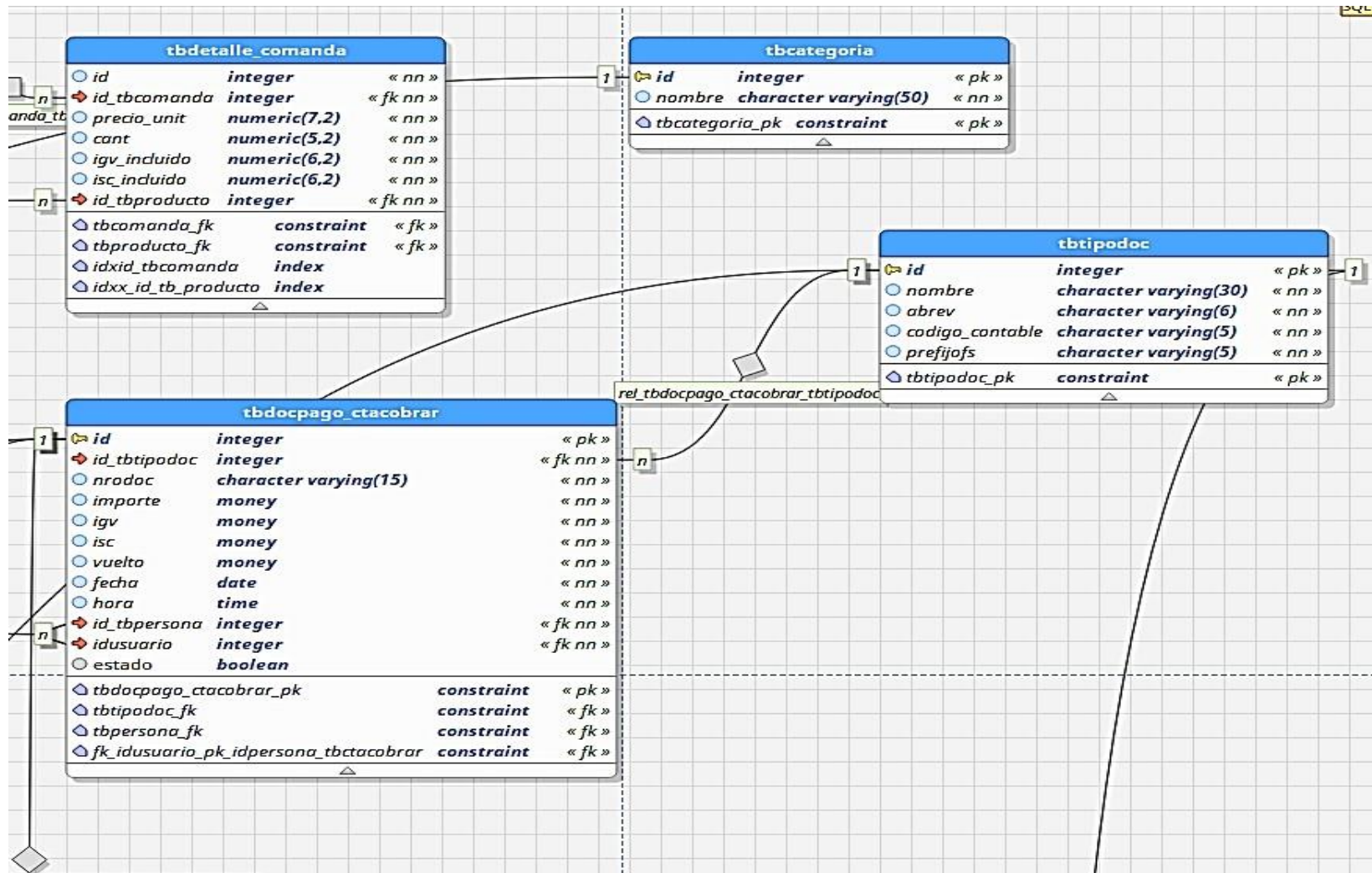


Imagen No. 12 Modelo relacional parte 2

Fuente: Propia Autoría

“Implementación del Módulo de Envío de Documentos de Ventas del Restaurant Zarandaja de la Ciudad de Chiclayo Consumiendo el servicio SOAP del Operador de Servicios Electrónicos NUBEFAC.”

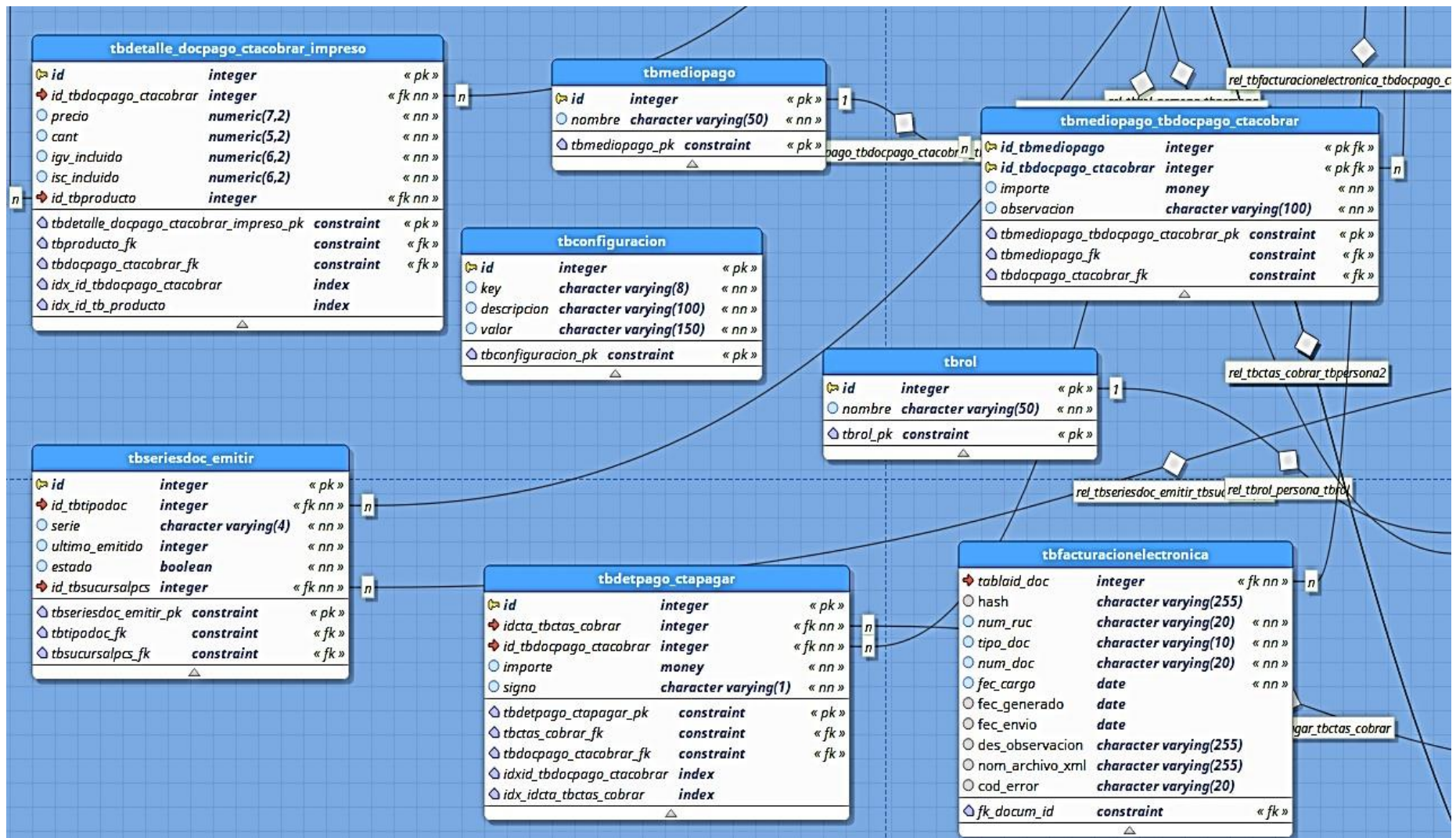


Imagen No. 13 Modelo relacional parte 3

Fuente: Propia Autoría

“Implementación del Módulo de Envío de Documentos de Ventas del Restaurant Zarandaja de la Ciudad de Chiclayo Consumiendo el servicio SOAP del Operador de Servicios Electrónicos NUBEFACT.”

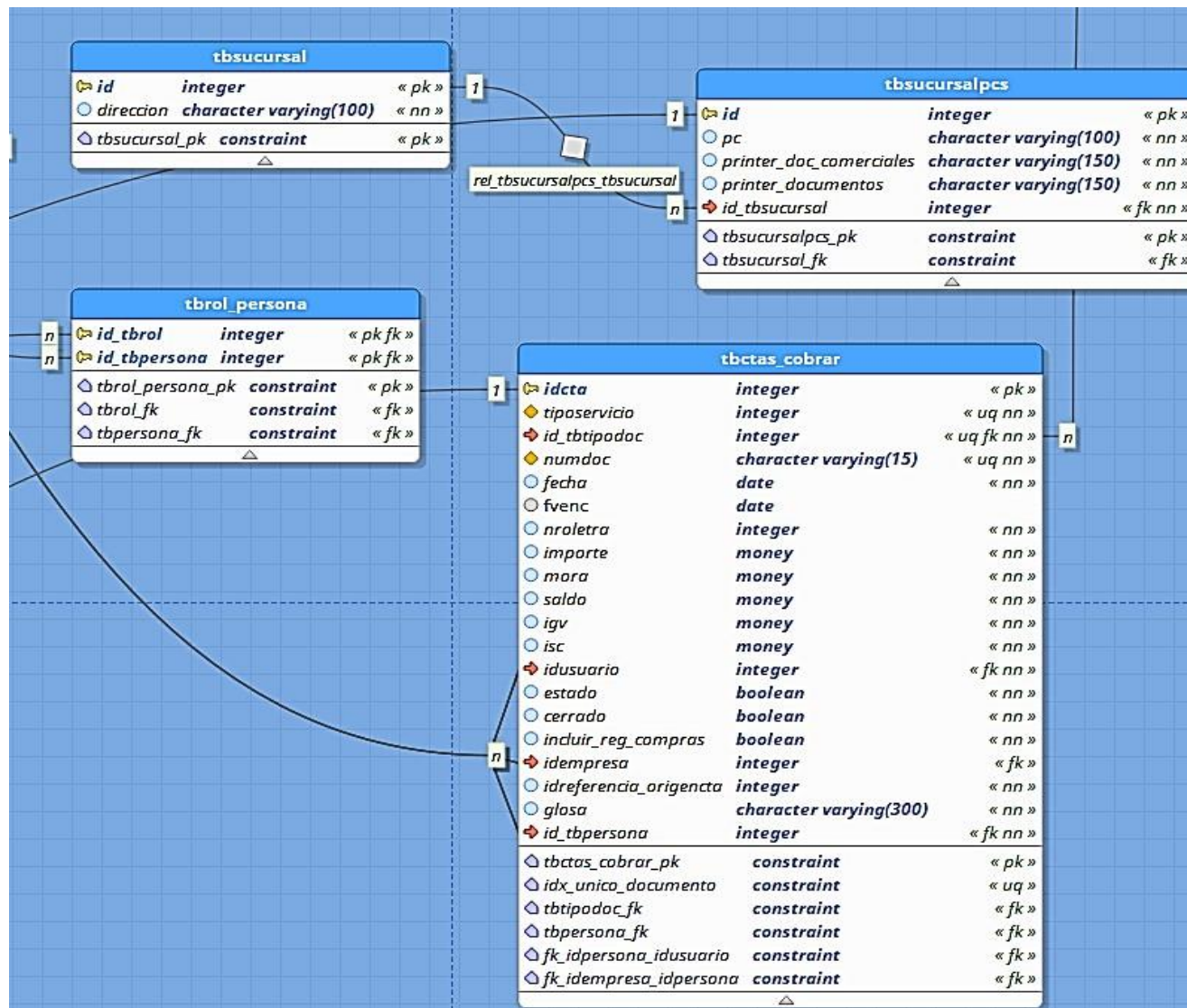


Imagen No. 14 Modelo relacional parte 4

Fuente: Propia Autoría

F. Diccionario de datos de las tablas de la base de datos.

tbestadomesa		
 id	integer	« pk »
 nombre	character varying(20)	« nn »
 color	bigint	« nn »
 tbestadomesa_pk	constraint	« pk »

Imagen No. 15 Tabla tbestadomesa

Fuente: Propia Autoría

tbmesa		
 id	integer	« pk »
 etiqueta	character varying(20)	« nn »
 x	integer	« nn »
 y	integer	« nn »
 ancho	integer	« nn »
 alto	integer	« nn »
 id_tbestadomesa	integer	« fk nn »
 tbmesa_pk	constraint	« pk »
 tbestadomesa_fk	constraint	« fk »

Imagen No. 16 Tabla tbmesa

Fuente: Propia Autoría

tbdetalle_comanda		
 id	integer	« nn »
 id_tbcomanda	integer	« fk nn »
 precio_unit	numeric(7,2)	« nn »
 cant	numeric(5,2)	« nn »
 igv_incluido	numeric(6,2)	« nn »
 isc_incluido	numeric(6,2)	« nn »
 id_tbproducto	integer	« fk nn »
 tbcomanda_fk	constraint	« fk »
 tbproducto_fk	constraint	« fk »
 idxid_tbcomanda	index	
 idxx_id_tb_producto	index	

Imagen No. 17 Tabla tbdetalle_comanda

Fuente: Propia Autoría

tbcomanda		
 id	integer	« pk »
 fecha	date	« nn »
 hora	time	« nn »
 tiposervicio	integer	« nn »
 nombre_ref	character varying(50)	« nn »
 telef_ref	character varying(20)	« nn »
 correo_ref	character varying	« nn »
 direccion_ref_delivery	character varying(100)	« nn »
 referencia_dir_delivery	character varying(100)	« nn »
 id_tbmesa	integer	« fk »
 idusuario	integer	« fk nn »
 estado	boolean	« nn »
 id_tbpersona	integer	« fk »
 cerrado	boolean	« nn »
 tbcomanda_pk	constraint	« pk »
 tbmesa_fk	constraint	« fk »
 tbpersona_fk	constraint	« fk »
 pk_idpersona_fkidusuario_tbcomanda	constraint	« fk »

Imagen No. 18 Tabla tbcomanda

Fuente: Propia Autoría

tbcategoria		
 id	integer	« pk »
 nombre	character varying(50)	« nn »
 tbcategoria_pk	constraint	« pk »

Imagen No. 19 Tabla tbcategoría

Fuente: Propia Autoría

tbproducto		
 id	integer	« pk »
 tipoprod	integer	« nn »
 nombre	character varying(150)	« nn »
 precio	numeric(7,2)	« nn »
 afecto_igv	boolean	« nn »
 afecto_isc	boolean	« nn »
 foto	character varying(150)	« nn »
 observacion	character varying(150)	« nn »
 id_tbcategoria	integer	« fk nn »
 estado	boolean	« nn »
 tb_producto_pk	constraint	« pk »
 tbcategoria_fk	constraint	« fk »

Imagen No. 20 Tabla tbproducto

Fuente: Propia Autoría

tbpersona		
 id	integer	« pk »
 tipoper	character(1)	« nn »
 razonsocial	character varying(100)	« nn »
 ape_pat	character varying(50)	« nn »
 ape_mat	character varying(50)	« nn »
 nombre	character varying(50)	« nn »
 nrodoc	character varying(20)	« nn »
 direccion	character varying(400)	« nn »
 telefono	character varying(15)	« nn »
 email	character varying(100)	« nn »
 usuario	character varying(20)	« nn »
 clave	character varying(20)	« nn »
 password_hash	text	« nn »
 api_key	character varying(32)	« nn »
 id_tbtipo_documento_identidad	integer	« fk nn »
 estado	boolean	« nn »
 tbpersona_pk	constraint	« pk »
 tbtipo_documento_identidad_fk	constraint	« fk »

Imagen No. 21 Tabla tbpersona

Fuente: Propia Autoría

tbtipo_documento_identidad		
 id	integer	« pk »
 nombre	character varying(30)	« nn »
 longitud	integer	« nn »
 cod_sunat	integer	« nn »
 tbtipo_documento_identidad_pk	constraint	« pk »

Imagen No. 22 Tabla tbtipo_documento_identidad

Fuente: Propia Autoría

tbdocpago_ctacobrar		
 id	integer	« pk »
 id_tbtipodoc	integer	« fk nn »
 nrodoc	character varying(15)	« nn »
 importe	money	« nn »
 igv	money	« nn »
 isc	money	« nn »
 vuelto	money	« nn »
 fecha	date	« nn »
 hora	time	« nn »
 id_tbpersona	integer	« fk nn »
 idusuario	integer	« fk nn »
 estado	boolean	
 tbdocpago_ctacobrar_pk	constraint	« pk »
 tbtipodoc_fk	constraint	« fk »
 tbpersona_fk	constraint	« fk »
 fk_idusuario_pk_idpersona_tbtacobrar	constraint	« fk »

Imagen No. 23 Tabla tbdocpago_ctacobrar

Fuente: Propia Autoría

tbtipodoc		
 id	integer	« pk »
 nombre	character varying(30)	« nn »
 abrev	character varying(6)	« nn »
 codigo_contable	character varying(5)	« nn »
 prefijos	character varying(5)	« nn »
 tbtipodoc_pk	constraint	« pk »

Imagen No. 24 Tabla tbtipodoc

Fuente: Propia Autoría

tbdetalle_docpago_ctacobrar_impreso		
 id	integer	« pk »
 id_tbdocpago_ctacobrar	integer	« fk nn »
 precio	numeric(7,2)	« nn »
 cant	numeric(5,2)	« nn »
 igv_induido	numeric(6,2)	« nn »
 isc_induido	numeric(6,2)	« nn »
 id_tbproducto	integer	« fk nn »
 tbdetalle_docpago_ctacobrar_impreso_pk	constraint	« pk »
 tbproducto_fk	constraint	« fk »
 tbdocpago_ctacobrar_fk	constraint	« fk »
 idx_id_tbdocpago_ctacobrar	index	
 idx_id_tb_producto	index	

Imagen No. 25 Tabla tbdetalle_docpago_ctacobrar_impreso

Fuente: Propia Autoría

tbmediopago		
 id	integer	« pk »
 nombre	character varying(50)	« nn »
 tbmediopago_pk	constraint	« pk »

Imagen No. 26 Tabla tbmediopago

Fuente: Propia Autoría

tbmediopago_tbdopago_ctacobrar		
 id_tbmediopago	integer	« pk fk »
 id_tbdopago_ctacobrar	integer	« pk fk »
 importe	money	« nn »
 observacion	character varying(100)	« nn »
 tbmediopago_tbdopago_ctacobrar_pk	constraint	« pk »
 tbmediopago_fk	constraint	« fk »
 tbdopago_ctacobrar_fk	constraint	« fk »

Imagen No. 27 Tabla tbmediopago_tbdopago_ctacobrar

Fuente: Propia Autoría

tbsucursal		
 id	integer	« pk »
 direccion	character varying(100)	« nn »
 tbsucursal_pk	constraint	« pk »

Imagen No. 28 Tabla tbmediopago_tbdopago_ctacobrar

Fuente: Propia Autoría

tbsucursal		
 id	integer	« pk »
 direccion	character varying(100)	« nn »
 tbsucursal_pk	constraint	« pk »

Imagen No. 29 Tabla tbsucursal.

Fuente: Propia Autoría

tbsucursalpcs		
 id	integer	« pk »
 pc	character varying(100)	« nn »
 printer_doc_comerciales	character varying(150)	« nn »
 printer_documentos	character varying(150)	« nn »
 id_tbsucursal	integer	« fk nn »
 tbsucursalpcs_pk	constraint	« pk »
 tbsucursal_fk	constraint	« fk »

Imagen No. 30 Tabla tbsucursalpcs

Fuente: Propia Autoría

tbseriesdoc_emitir		
 id	integer	« pk »
 id_tbtipodoc	integer	« fk nn »
 serie	character varying(4)	« nn »
 ultimo_emitido	integer	« nn »
 estado	boolean	« nn »
 id_tbsucursalpcs	integer	« fk nn »
 tbseriesdoc_emitir_pk	constraint	« pk »
 tbtipodoc_fk	constraint	« fk »
 tbsucursalpcs_fk	constraint	« fk »

Imagen No. 31 Tabla tbseriesdoc_emitir

Fuente: Propia Autoría

tbconfiguracion		
 id	integer	« pk »
 key	character varying(8)	« nn »
 descripcion	character varying(100)	« nn »
 valor	character varying(150)	« nn »
 tbconfiguracion_pk	constraint	« pk »

Imagen No. 32 Tabla tbconfiguracion

Fuente: Propia Autoría

tbrol		
 id	integer	« pk »
 nombre	character varying(50)	« nn »
 tbrol_pk	constraint	« pk »

Imagen No. 33 Tabla tbrol

Fuente: Propia Autoría

tbrol_persona		
 id_tbrol	integer	« pk fk »
 id_tbpersona	integer	« pk fk »
 tbrol_persona_pk	constraint	« pk »
 tbrol_fk	constraint	« fk »
 tbpersona_fk	constraint	« fk »

Imagen No. 34 Tabla tbrolpersona

Fuente: Propia Autoría

tbctas_cobrar		
 idcta	integer	« pk »
 tiposervicio	integer	« uq nn »
 id_tbtipodoc	integer	« uq fk nn »
 numdoc	character varying(15)	« uq nn »
 fecha	date	« nn »
 fvenc	date	
 nroletra	integer	« nn »
 importe	money	« nn »
 mora	money	« nn »
 saldo	money	« nn »
 igv	money	« nn »
 isc	money	« nn »
 idusuario	integer	« fk nn »
 estado	boolean	« nn »
 cerrado	boolean	« nn »
 incluir_reg_compras	boolean	« nn »
 idempresa	integer	« fk »
 idreferencia_origencta	integer	« nn »
 glosa	character varying(300)	« nn »
 id_tbpersona	integer	« fk nn »
 tbctas_cobrar_pk	constraint	« pk »
 idx_unico_documento	constraint	« uq »
 tbtipodoc_fk	constraint	« fk »
 tbpersona_fk	constraint	« fk »
 fk_idpersona_idusuario	constraint	« fk »
 fk_idempresa_idpersona	constraint	« fk »

Imagen No. 35 Tabla tbctas_cobrar

Fuente: Propia Autoría

tbdetpago_ctapagar		
 id	integer	« pk »
 idcta_tbctas_cobrar	integer	« fk nn »
 id_tbdcpago_ctacobrar	integer	« fk nn »
 importe	money	« nn »
 signo	character varying(1)	« nn »
 tbdetpago_ctapagar_pk	constraint	« pk »
 tbctas_cobrar_fk	constraint	« fk »
 tbdcpago_ctacobrar_fk	constraint	« fk »
 idxid_tbdcpago_ctacobrar	index	
 idx_idcta_tbctas_cobrar	index	

Imagen No. 36 Tabla tbdetpago_ctapagar

Fuente: Propia Autoría

tbfacturacionelectronica		
 tablaid_doc	integer	« fk nn »
 hash	character varying(255)	
 num_ruc	character varying(20)	« nn »
 tipo_doc	character varying(10)	« nn »
 num_doc	character varying(20)	« nn »
 fec_cargo	date	« nn »
 fec_generado	date	
 fec_envio	date	
 des_observacion	character varying(255)	
 nom_archivo_xml	character varying(255)	
 cod_error	character varying(20)	
 fk_docum_id	constraint	« fk »

Imagen No. 37 Tabla tbfacturacionelectronica

Fuente: Propia Autoría

A continuación, presento la implementación del sistema con la transcripción del código Json para la consulta del DNI o RUC del cliente ante la Sunat y Reniec.

```
$app->get('/consulta_dni_ruc', function ($request, $response,
$args) {

    $tipo = $request->getQueryParam('tipo');
    $numero = $request->getQueryParam('numero');

    if (!$numero) {
        $numero="";
    }

    $respuesta = array();
    $respuesta['_error'] = false;
    $respuesta['persona'] = array();

    $db=new AccesoDatos_clstbpersona();
    /Buscamos el RUC o DNI en la BD/
    $row = $db->getRowDNIRUC($numero);

    if (is_array($row)) {
        /*Si lo encontramos devolvemos los datos */
        $registro = array();
        $registro['id'] = $row['oid'];
        $registro['tipoper'] = $row['otipoper'];
        $registro['razonsocial'] = $row['orazonsocial'];
        $registro['ape_pat'] = $row['oape_pat'];
        $registro['ape_mat'] = $row['oape_mat'];
        $registro['nombre'] = $row['onombre'];
        $registro['nrodoc'] = $row['onrodoc'];
        $registro['direccion'] = $row['odireccion'];
        $registro['email'] = $row['oemail'];
        array_push($respuesta['persona'], $registro);
    }
    else { /* No existe en la API donde internamente busca en
    RENIEC o SUNAT*/

        $url =
        'https://digitalhipuq/consultadni_ruc.php?usuario=USUARIO&passwo
rd=CLAVE&documento='.$tipo.'&nro_documento='.$numero ;
```

```
$data = array('key1' => 'value1', 'key2' => 'value2');

$options = array(
    'http' => array(
        'header' => "Content-type:
application/json\r\n",
        'method' => 'POST',
        'content' => http_build_query($data)
    )
);
$context = stream_context_create($options);
$result = file_get_contents($url, false, $context);

if ($result === FALSE) { /* Manejar error */
    $registro = array();
    $registro['id'] = "";
    $registro['tipoper'] = "";
    $registro['razonsocial'] = 'ERROR';
    $registro['ape_pat'] = 'ERROR';
    $registro['ape_mat'] = 'ERROR';
    $registro['nombre'] = 'ERROR';
    $registro['nrodoc'] = 'ERROR';
    $registro['direccion'] = 'ERROR';
    $registro['email'] = "";
    array_push($respuesta['persona'], $registro);
}
else {

    $object = json_decode( $result );

    if ($object->success=='False'){
        /El API estuvo activo pero no encontró el
documento/

        $registro = array();
        $registro['id'] = "";
        $registro['tipoper'] = "";
        $registro['razonsocial'] = "";
        $registro['ape_pat'] = "";
        $registro['ape_mat'] = "";
        $registro['nombre'] = "";
        $registro['nrodoc'] = "";
        $registro['direccion'] = "";
        $registro['email'] = "";
    }
}
```

```

                                array_push($respuesta['persona'],
$registro);
                                }
                                else {

                                /Encontró datos en RENIEC o SUNAT/
                                $registro = array();
                                $registro['id'] = 0;
                                if ($tipo == 'DNI') {
                                    /*Si es DNI, es personal Natural
tipoper = 'N' */
                                    $registro['tipoper'] = 'N';
                                    $registro['razonsocial'] = "";
                                    $registro['ape_pat'] = $object->result-
>Paterno;
                                    $registro['ape_mat'] = $object->result-
>Materno;
                                    $registro['nombre'] = $object->result-
>Nombre;
                                    $registro['nrodoc'] = $numero;
                                    $registro['direccion'] = "";
                                    $registro['email'] = "";
                                }
                                else {
                                    /*Si es RUC pero inicia RUC con 10,
es personal natural tipoper = 'N' */
                                    if (substr($numero, 0, 2) == '10'){
                                        $registro['tipoper'] = 'N';
                                    }
                                    else{/Sino Juridica/
                                        $registro['tipoper'] = 'J';
                                    }
                                    $registro['razonsocial'] = $object-
>result->RazonSocial;
                                    $registro['ape_pat'] = "";
                                    $registro['ape_mat'] = "";
                                    $registro['nombre'] = "";
                                    $registro['nrodoc'] = $numero;
                                    $registro['direccion'] = $object->result-
>Direccion;
                                    $registro['email'] = "";
                                }

```

```
                                array_push($respuesta['persona'],
$registro);
                                }
                                }
                                }
                                return $response->withJson($respuesta,200);
});
```

Donde tipo puede ser DNI o RUC y en número, el cliente buscado.

Para probar lo indicado sobre el código anterior accedaremos a la siguiente url asignando un DNI ejemplo el 77159715.

JSON	Datos sin procesar	Cabeceras
Guardar	Copiar	Contraer todo Expandir todo Filtar JSON
<pre>_error: false persona: 0: id: 0 tipoper: "N" razonsocial: "" ape_pat: "CARRASCO" ape_mat: "MUJICA" nombre: "CRISTHIAN ANDY" nrodoc: "77159715" direccion: "" email: ""</pre>		

Imagen No. 52 Texto Json ejemplo con dni.

Fuente: Propia autoría.

Veamos un ejemplo con RUC:

JSON	Datos sin procesar	Cabeceras
Guardar	Copiar	Contraer todo Expandir todo  Filtrar JSON
<pre>{ "_error": false, "persona": { "0": { "id": 2748, "tipoper": "N", "razonsocial": "", "ape_pat": "", "ape_mat": "", "nombre": "UNIVERSIDAD PARTICULAR DE CHICLAYO", "nrodoc": "20209239541", "direccion": "", "email": "" } } }</pre>		

Imagen No. 38 Texto Json ejemplo con ruc.

Fuente: Propia autoría.

Si el campo ID es 0, significa que no estaba grabado anteriormente en la Base de datos y se sustrajo de SUNAT o RENIEC

Para la culminación de la implementación registro en este informe de tesis las pantallas o ventanas implementadas en la aplicación que he desarrollado para solucionar tecnológicamente el problema planteado en este trabajo de investigación.

A. Acceso al sistema



Imagen No. 39 Pantalla de inicio de la Aplicación.

Fuente: Autoría propia.

Descripción: Mediante este formulario el usuario accesa a la aplicación seleccionando el tipo de usuario e ingresando la contraseña. Dichos datos son validados con la base de datos.

A. Menú principal.

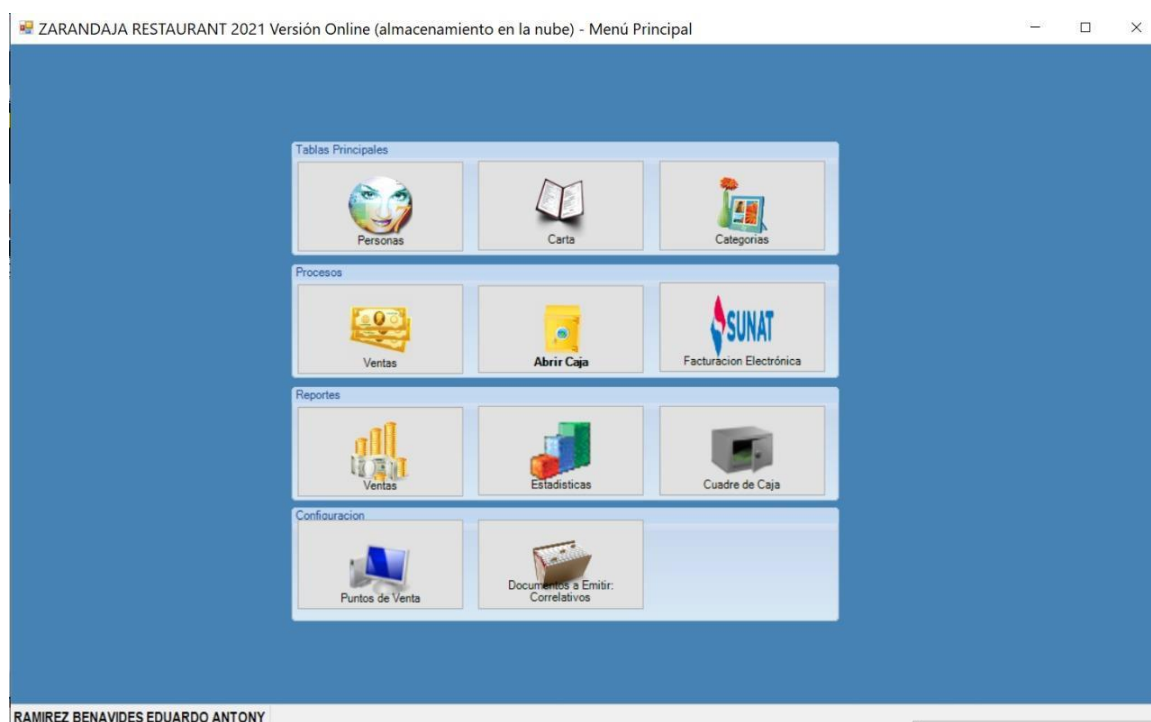


Imagen No. 40 Menú principal del sistema online Zarandaja Restaurant 2020.

Fuente: Propia Autoría.

Descripción: En esta pantalla visualizamos las opciones que se han implementado en el sistema online para el Restaurant Zarandaja. En ella se observa la opción “Facturación Electrónica” el cual es el punto central de este trabajo.

C. Registro general del personas.

ID	tipo	nombres	nrodoc	roles	activo
16	75	J	AVICOLA EVA CECILIA EMPRESA INDIVIDUAL DE RESPONSABILIDAD LI	RUC 20530244084	☒
17	132	J	BEVA BANCO CONTINENTAL	RUC 20100130204	☒
18	103	J	BLENDING S.A.C.	RUC 20519320305	☒
19	108	N	CHONG ARROYO MARTHA CECILIA	DNI 18114095	☒
20	111	N	CHYO HUAMANTINCO TOSHIO	DNI 40727662	☒
21	139	J	CIMENTACIONES PREFABRICADAS TERRATEST PERU SAC	RUC 20566607639	☒
22	49	J	COLEGIO MANUEL PARDO	RUC 20103416982	☒
23	54	J	COLEGIO SAN AGUSTIN	RUC 20103560757	☒
24	83	J	COLEGIO SAN AGUSTIN DE VILLA E.I.R.L.	RUC 20556874981	☒
25	66	J	COLEGIO SANTA MARIA REYNA	RUC 20137382203	☒
26	107	J	CONSORCIO COBRA SCL UA&TC	RUC 20602815995	☒
27	136	J	CONSORCIO P&D	RUC 20603098316	☒
28	119	J	CONSORCIO TESTING SERVICE GROUP S.A.C. - TEPs GROUP S.A.C.	RUC 20603396431	☒
29	131	J	CONSULTUS PERU S.A.C.	RUC 20556569593	☒
30	140	J	COSAPI S A	RUC 20100082391	☒
31	115	J	CPTS DRILLING FLUIDS LIMITED S.A.C.	RUC 20463722472	☒
32	133	J	CREUS THERMOFUSION Y SERVICIOS DE INGENIERIA SAC	RUC 20526067135	☒
33	59	N	CRUZADO PEÑA ALVARO ALBERTO	DNI 47268030	☒
34	47	J	CUBAS CRUZ ELVES ORLANDO	RUC 10081320699	☒
35	67	N	CUBAS CRUZ MARIA ARMANDINA	DNI 42714472	☒
36	101	J	DEMEM S A	RUC 20101759173	☒
37	110	N	DULANTO DEZA JUDITH MYRIAN	DNI 09891875	☒
38	104	J	ENEL GENERACION PERU S.A.A.	RUC 20330791412	☒
39	95	J	ENERGY PROJECT CONSULTING S.A.C	RUC 20566090911	☒
40	59	J	EPPO SOCIEDAD ANONIMA	RUC 20102890508	☒
41	135	J	ESTACION DE SERVICIOS SAN JOSE S.A.C.	RUC 20175642341	☒
42	84	J	ESTUDIO RODRIGUEZ LARRAIN ABOGADOS S.C.R. L.	RUC 20109708063	☒
43	98	J	E Y C METALIKAS SOCIEDAD ANONIMA CERRADA	RUC 20387418882	☒

Imagen No. 41 Registro general de personas.

Fuente: Propia Autoría

Descripción: Formulario del módulo que permite registrar a las personas que solicitan un servicio de atención. En este formulario podemos hacer mantenimiento: Cuando se registra se indica si es cliente, trabajador, etc.

Una persona puede tener uno o muchos roles (cliente, trabajador, etc) y bajo un rol en particular se puede clarificar uno o muchas personas (relación M : N).

Primero se busca en la base de datos al cliente, si no se encuentra lo busca en Reniec o sunat y se agrega a la tabla personas al momento de hacer la cobranza de lo consumido.

D. Mantenimiento de Productos.

ZARANDAJA RESTAURANT 2020 Versión Online (vía internet) - Menú Principal

Registro General de Producto(s)...

Nuevo - F2

Modificar - F3

Eliminar - F4

Cerrar

Producto:

Categoría: [Cualquiera]

200

Ver

200 registros visualizados. Total de coincidencias encontradas 632 según criterios de búsqueda.

Registros Visualizados: 200

id	nombre	P.Vta	estado
25	271 AGUADITO DE CONCHA NEG (F	80.00	☒
26	272 AGUADITO DE CONCHA NEG (P	42.00	☒
27	273 AGUADITO DE FILETE DE PESCADO	40.00	☒
28	274 AGUADITO DE MARISCO (FUE)	75.00	☒
29	275 AGUADITO DE MARISCO (PORC	42.00	☒
30	371 AGUADITO DE MARISCOS PORC FILETE MERO	48.00	☒
31	276 AGUADITO DE MERO (FUENTE)	78.00	☒
32	277 AGUADITO DE MERO (PORCION	42.00	☒
33	278 AGUADITO FILETE MERO	52.00	☒
34	279 AGUADITO FILETE MERO Y CONCHAS NEGRAS	55.00	☒
35	617 AGUA SAN LUIS BOTELLA DE VIDRIO	3.00	☒
36	61 AGUA TONICA	6.00	☒
37	280 ALBONDIGAS PULPA CANGREJO	35.00	☒
38	62 ALGARROBINA COPA CHICA	5.00	☒
39	63 ANIS MONO	20.00	☒
40	2 ANTICUCHO DE LOMO	38.00	☒
41	281 ANTICUCHO DE MERO	48.00	☒
42	372 ANTICUCHO DE PESCADO	38.00	☒
43	3 ANTICUCHO DE POLLO	28.00	☒
44	138 ARROZ A LA CUBANA	14.00	☒
45	378 ARROZ CONCHAS NEGRAS (FUE	78.00	☒
46	379 ARROZ CONCHAS NEGRAS (POR	40.00	☒
47	373 ARROZ CON CONCHAS Y LANGOSTINOS	40.00	☒
48	374 ARROZ CON LANGOSTINO (FUE	78.00	☒
49	375 ARROZ CON LANGOSTINO (POR	40.00	☒
50	376 ARROZ CON MARISCO (FUENTE)	75.00	☒
61	377 ARROZ CON MARISCO (PORCION	39.00	☒

Imagen No. 42 Mantenimiento de producto

Fuente: propia autoría

Descripción: Mediante este formulario se visualiza, se registra y se hace mantenimiento a los productos que ofrece el restaurant. En este formulario se puede aplicar filtros por categoría o por nombre del producto.

E. Registro de producto.

200 registros visualizados. Total de coincidencias encontradas 632 segun criterios de busqueda. Registros Visualizados: 200

id	nombre	P.Vta	estado
91	239 CEBICHE DE LANGOSTINO (FUENT	78.00	☒
92	240 CEBICHE DE LANGOSTINO (PORCI		
93	241 CEBICHE DE MARISCOS (FUENTE)		
94	242 CEBICHE DE MARISCOS (PORCION)		
95	243 CEBICHE DE MERO 3 AÑES		
96	243 CEBICHE DE MERO (FUENTE)		
97	244 CEBICHE DE MERO (PORCION)		
98	246 CEBICHE DE OJO DE UVA		
99	247 CEBICHE DE PESCADO		
100	384 CEBICHE DE PESCADO 3 AÑES		
101	248 CEBICHE DE PESCADO (FUENTE)		
102	249 CEBICHE DE PULPO (FUENTE)		
103	250 CEBICHE DE PULPO (PORCION)		
104	251 CEBICHE MIXTO (FU		
105	253 CEBICHE MIXTO MERO (FUENT	44.00	☒
106	254 CEBICHE MIXTO MERO (PORCI		
107	252 CEBICHE MIXTO (PO	38.00	☒
108	37 CENA	35.00	☒
109	36 CENA	24.00	☒
110	38 CENA NAVIDEÑA	32.00	☒
111	69 CERVEZA CHICA	6.00	☒
112	70 CERVEZA CORONA	7.00	☒
113	71 CERVEZA CRISTAL GRANDE	8.00	☒
114	72 CERVEZA CUSQUEÑA GRANDE	9.00	☒
115	73 CERVEZA CUSQUEÑA NEGRA GRANDE	10.00	☒
116	74 CERVEZA CUSQUEÑA NEGRA CHICA	7.00	☒
117	75 CERVEZA CUSQUEÑA NEGRA CHICA	7.00	☒

Datos Principales

Tipo: **PLATO**

Nombre: CEBICHE DE OJO DE UVA

Categoría: CEBICHES

Precio: S/. 44.00

☒ Afecto IGV? ☐ Afecto ISC?

☒ Activo? **Grabar** Cancelar

Imagen No. 43 Registro de producto.

Fuente: Propia autoría

Descripción: El cuadro emergente registra el producto a la base de datos y luego se actualiza el formulario de mantenimiento de producto.

En este registro se debe indicar el nombre asignar la categoría y establecer si el precio está afecto al impuesto general a las ventas y al impuesto selectivo al consumo. También se establece si es un producto activo (en venta) o no.

F. Mapa de mesas para crear comanda.



Imagen No. 44 Mapa de mesas para crear comanda.

Fuente: Propia Autoría

Descripción: Al pasar el puntero del ratón nos indicará una descripción del estado de la mesa. Se utilizan tres colores: verde (libre), rojo (ocupado) y ámbar (en limpieza). Se pulsa click en cualquiera de los botones para crear una comanda o editar una existente.

G. Generar comanda.

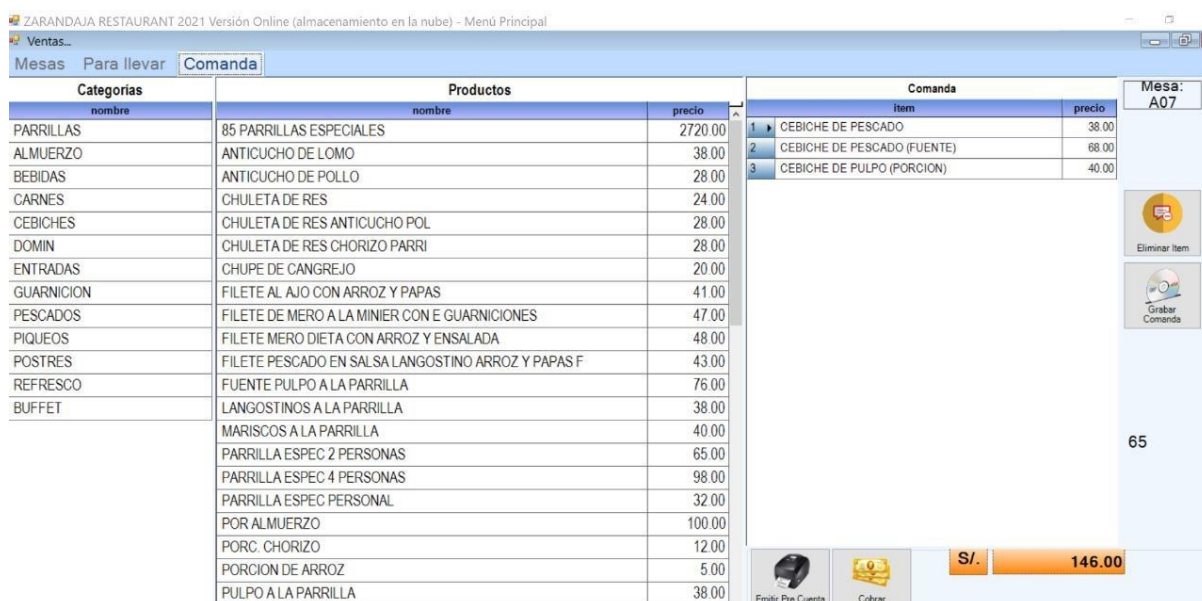


Imagen No. 45 Generar comanda.

Fuente: Propia autoría

Descripción: Utilizando los datos de categoría se ubican los productos solicitados y se registran en la comanda. En la comanda se registra el producto y precio almacenado en la base de datos. Finalmente se debe grabar para poder en algún momento realizar la cobranza.

También podemos emitir una precuenta para conocer el consumo realizado hasta cierto momento por el cliente.

H. Registro de pago en efectivo

Seleccione items para concretar venta...

Consumido: 146.00 Descuento: 0.00

A Pagar: 146.00

Efectivo

Simple

Multiple

Solo pago en Efectivo...

RECIBIDO: 200.00 COBRADO: 146.00 VUELTO: 54.00

0 0.50 1 2 5 10 20 50 100 200

Boleta Factura Ticket Provisional Regresar

Imagen No. 46 Registro de Pago en efectivo

Fuente: Propia autoría

Descripción: En este formulario se registra el pago. El pago en efectivo se registra con el botón “efectivo”, la opción simple permite registrar una venta mediante sólo una tarjeta de crédito. La opción múltiple permite que una cuenta se pague en forma fraccionada en diferentes tarjetas de crédito. Una vez terminado el registro, el usuario podrá seleccionar Boleta o factura según el tipo de cliente que hizo el consumo o la compra. Antes de pagar la comanda podemos emitir un ticket provisional ante el pedido del cliente de saber su cuenta antes de pagar.

I. Registro de pago Simple.

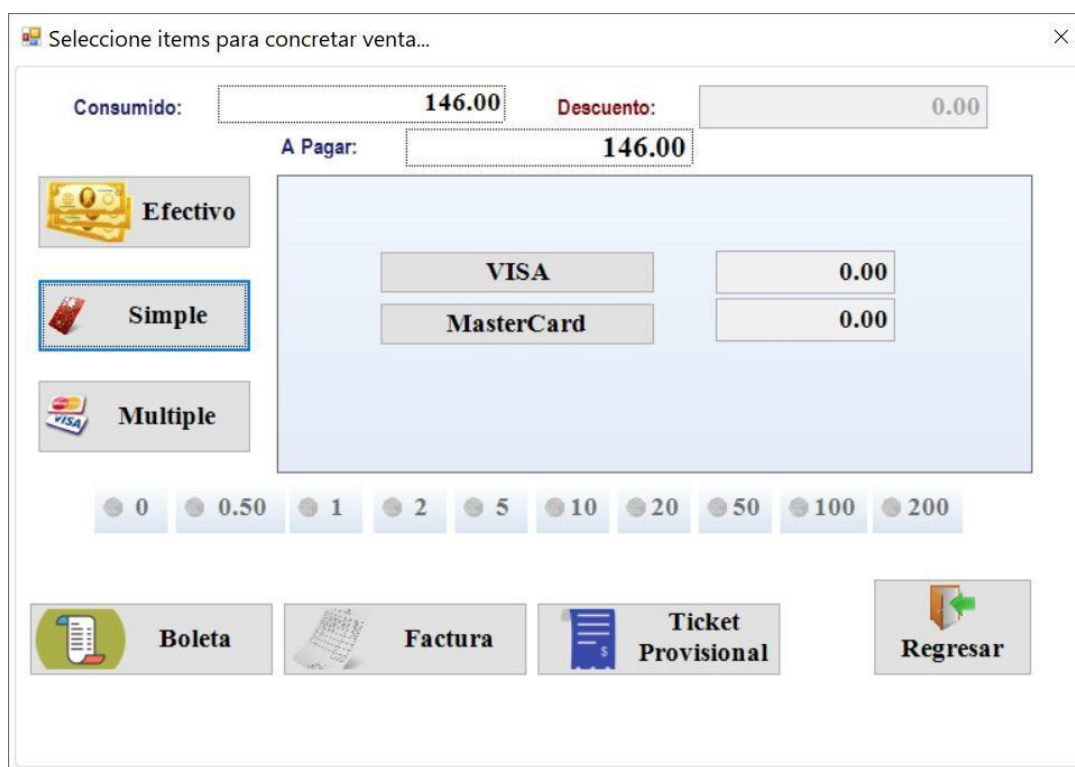


Imagen No. 47 Registro de pago Simple.

Fuente: Propia autoría.

Descripción: En esta opción se registra el pago mediante el uso de una tarjeta Visa o Matercard. Sólo se habilita una caja texto para ingresar el monto a pagar. No se activa para los dos a la vez.

J. Múltiples tipos de pago.

The screenshot shows a software window titled "Seleccione items para concretar venta...". It contains the following elements:

- Consumido:** 146.00
- Descuento:** 0.00
- A Pagar:** 146.00
- Payment Method Selection:**
 - Efectivo:** Represented by a gold coin icon.
 - Simple:** Represented by a red card icon.
 - Multiple:** Represented by a Visa card icon.
- Multiples Medios de Pago... (Sub-window):**
 - Efectivo:** 0.00 (highlighted with a blue bar)
 - VISA:** 0.00
 - MasterCard:** 0.00
 - Canje:** 0.00
 - Cobranza:** 0.00
 - Faltante:** 0.00
- Amount Selection:** A row of buttons for 0, 0.50, 1, 2, 5, 10, 20, 50, 100, and 200.
- Document Type Selection:**
 - Boleta:** Represented by a receipt icon.
 - Factura:** Represented by a document icon.
 - Ticket Provisional:** Represented by a blue document icon.
 - Regresar:** Represented by a green arrow icon.

Imagen No. 48 Múltiples Medios de pago.

Fuente: Propia autoría.

Descripción: Mediante esta opción nos brindá la posibilidad de fraccionar el pago en efectivo una parte y otra en una tarjeta o varias tarjetas. También le da la opción de pagar nada en efectivo y fraccionarlo en varias tarjetas.

K. Venta con factura.

Seleccione items para concretar venta...

Consumido: 146.00 Descuento: 0.00

A Pagar: 146.00

Efectivo

Venta con Factura

RUC: 20480544049

Razón Social: DEPORTE TOTAL SAC

Dirección: CAL. VICENTE LA VEGA NRO. E Mail:

En caso de factura, ingresar nro de RUC y el sistema lo buscará en el padrón SUNAT

Luego para confirmar pulsar en el botón indicado.

Boleta Factura Regresar

CEBICHE DE PULPO (FUENTE)	72.00
CEBICHE DE PULPO (PORCION)	40.00

Imagen No. 49 Venta con factura.

Fuente: Propia autoría

Descripción: Se ingresa el número de RUC. Los datos sobre la razón social, dirección y el email se visualizan en la ficha. Se pulsa el botón check verde para registrar la venta.

L. Envío de comprobantes de pagos a OSE-SUNAT

Envío de Comprobantes de Pago a OSE - SUNAT

De: 14/10/2020 Visualizar Enviar Todos Nuevo

1. Seleccionar día. Luego clic para visualizar los documentos electrónicos para su envío a OSE SUNAT

2. Clic en el botón para el envío en lote.

Total documentos: 83 Boletas - Facturas - Notas de Crédito

sel	rpt	mensaje	hash	iddoc	nrodoc	Fecha	M.Gravado	P	M.IGV	M.Total	M	nrodocmodif	gratuito	nrodoccliente	razonsocialcliente	d
1	☒			96402	B001-0000949	14/10/2020	238.73	1	42.97	281.70	PE	B001-0043147	☐	10373764	SANTACRUZ F. NORM	
2	☒			96403	B001-0043175	14/10/2020	281.36	1	50.64	332.00	PE		☐	00000000	-CLIENTES VARIOS	
3	☒			96404	B001-0043176	14/10/2020	200.00	1	36.00	236.00	PE		☐	00000000	-CLIENTES VARIOS	
4	☒			96405	B002-0045376	14/10/2020	243.31	1	43.79	287.10	PE		☐	00000000	-CLIENTES VARIOS	
5	☒			96406	B001-0043177	14/10/2020	72.46	1	13.04	85.50	PE		☐	00000000	-CLIENTES VARIOS	
6	☒			96407	B001-0000950	14/10/2020	121.27	1	21.83	143.10	PE	B001-0043127	☐	00000000	-CLIENTES VARIOS	
7	☒			96408	B001-0043178	14/10/2020	121.27	1	21.83	143.10	PE		☐	00000000	-CLIENTES VARIOS	
8	☒			96409	B001-0043179	14/10/2020	121.27	1	21.83	143.10	PE		☐	00000000	-CLIENTES VARIOS	
9	☒			96410	B001-0043180	14/10/2020	95.34	1	17.16	112.50	PE		☐	00000000	-CLIENTES VARIOS	
10	☒			96411	B001-0043181	14/10/2020	181.53	1	32.67	214.20	PE		☐	00000000	-CLIENTES VARIOS	
11	☒			96412	B002-0045377	14/10/2020	189.92	1	34.18	224.10	PE		☐	00000000	-CLIENTES VARIOS	
12	☒			96413	B001-0043182	14/10/2020	38.14	1	6.86	45.00	PE		☐	00000000	-CLIENTES VARIOS	
13	☒			96414	B002-0045378	14/10/2020	136.44	1	24.56	161.00	PE		☐	00000000	-CLIENTES VARIOS	
14	☒			96415	B001-0000951	14/10/2020	90.76	1	16.34	107.10	PE	B001-0043133	☐	00000000	-CLIENTES VARIOS	
15	☒			96416	B001-0043183	14/10/2020	245.09	1	44.12	289.20	PE		☐	00000000	-CLIENTES VARIOS	
16	☒			96417	B001-0043184	14/10/2020	117.46	1	21.14	138.60	PE		☐	00000000	-CLIENTES VARIOS	
17	☒			96418	B001-0000952	14/10/2020	159.41	1	28.69	188.10	PE	B001-0043146	☐	00000000	-CLIENTES VARIOS	
18	☒			96419	B001-0043185	14/10/2020	159.41	1	28.69	188.10	PE		☐	00000000	-CLIENTES VARIOS	
19	☒			96420	B001-0043186	14/10/2020	113.64	1	20.46	134.10	PE		☐	00000000	-CLIENTES VARIOS	
20	☒			96421	B001-0043187	14/10/2020	228.81	1	41.19	270.00	PE		☐	00000000	-CLIENTES VARIOS	
21	☒			96422	B002-0045379	14/10/2020	59.32	1	10.68	70.00	PE		☐	00000000	-CLIENTES VARIOS	
22	☒			96423	B002-0045380	14/10/2020	163.98	1	29.52	193.50	PE		☐	00000000	-CLIENTES VARIOS	
23	☒			96424	B002-0045381	14/10/2020	273.81	1	49.29	323.10	PE		☐	00000000	-CLIENTES VARIOS	
24	☒			96426	B002-0045382	14/10/2020	151.78	1	27.32	179.10	PE		☐	00000000	-CLIENTES VARIOS	
25	☒			96427	B001-0043188	14/10/2020	121.19	1	21.81	143.00	PE		☐	00000000	-CLIENTES VARIOS	
26	☒			96428	B002-0045383	14/10/2020	98.39	1	17.71	116.10	PE		☐	00000000	-CLIENTES VARIOS	
27	☒			96429	B001-0043189	14/10/2020	112.29	1	20.21	132.50	PE		☐	00000000	-CLIENTES VARIOS	
28	☒			96430	B001-0043190	14/10/2020	192.37	1	34.63	227.00	PE		☐	00000000	-CLIENTES VARIOS	
29	☒			96431	B002-0045384	14/10/2020	151.78	1	27.32	179.10	PE		☐	00000000	-CLIENTES VARIOS	
30	☒			96437	B002-0045385	14/10/2020	710.66	1	10.44	750.70	PE		☐	00000000	-CLIENTES VARIOS	

Imagen No. 50 Envío de comprobantes de pagos a OSE-SUNAT
Fuente: Propia autoría

Descripción: Se filtra por fecha los documentos electrónicos para su envío a la Ose SUNAT. Una vez visualizado se presiona el botón “Enviar”.

M. Retorno OSE-SUNAT.

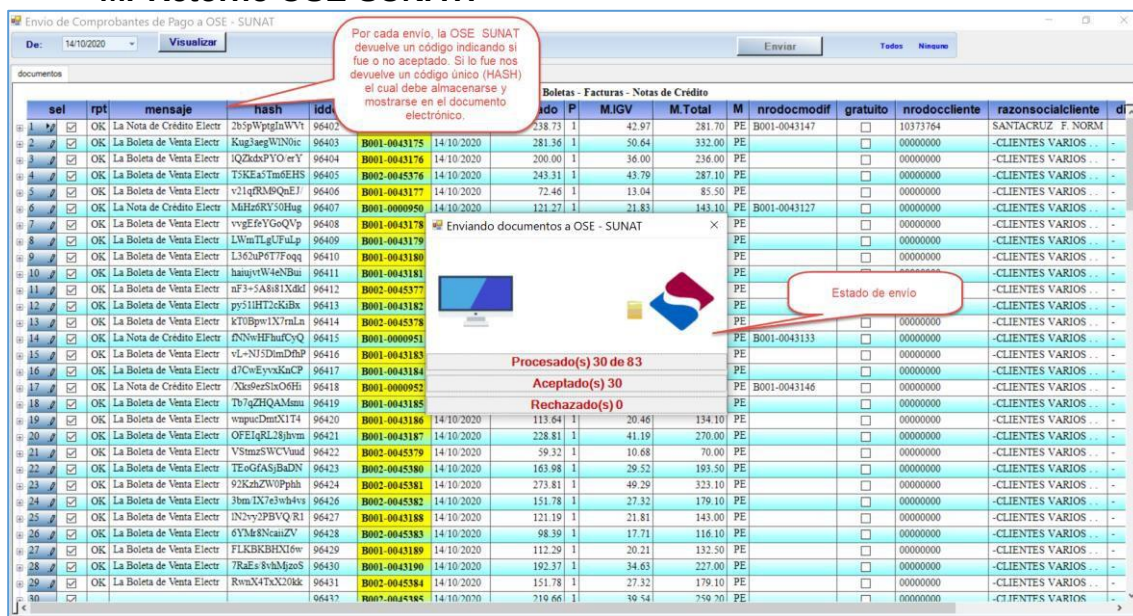


Imagen No. 51 Retorno de OSE Sunat

Fuente: Propia autoría

Descripción: Cuando se presiona el botón “Enviar” el sistema empieza a procesar e enviar cada registro del documento electrónico. Se observa el total a procesar, el total procesado, el total de aceptados y el total de rechazados.

Por último, señalo los costos de inversión desde el inicio del proyecto hasta el final del mismo.

CLASIFICACIÓN	ITEM	CANTIDAD	TIEMPO	TOTAL
EQUIPO DE DESARROLLO	ASESOR	1	4 meses	S/1,300.00
	DESARROLLADOR	1	2 meses	S/1,600.00
SUBTOTAL				S/2,900.00
DOCUMENTACIÓN	FOTOCOPIAS	300		S/15.00
	IMPRESIONES	350		S/245.00
	ESPIRALADOS	5		S/25.00
	EMPASTADOS	5		S/125.00
SUBTOTAL				S/410.00
OTROS	PASAJES		5 meses	S/600.00
	PAQUETE DE DATOS		5 meses	S/350.00
SUBTOTAL				S/950.00
TOTAL				S/4,260.00

Tabla 2 Presupuesto.

Fuente: propia autoría

Capítulo V: DISCUSIÓN

CAPITULO V: DISCUSIÓN

De acuerdo a lo señalado en los antecedentes de este proyecto, así como el marco teórico entonces pongo a discusión lo siguiente enfocados en lo indicado por cada uno de los autores:

- A. En la tesis “Emisión de los Comprobantes de Pago Electrónicos para Evitar las Sanciones Tributarias en la Empresa Binresa S.A.C., 2017” que tiene como autores a Yonirian Karina Chávez Pío y a Isabel Ivonne Requena Maza señalan como conclusión en su resumen la importancia del análisis de los procedimientos que se realizan para la generación de los comprobantes de pago y con ello evitar sanciones tributarias que siempre producen un impacto económico negativo en la empresa. Esto es muy importante porque lo que se busca es en todo momento el bienestar del usuario final pero también el bienestar económico y financiero de la empresa, pero siempre esto debe estar con la realidad y con las disposiciones emitidas por el organismo de control de tributación es por ello que considero importante y lo menciono en uno de los objetivos establecer los requerimientos funcionales, así como la implementación del módulo de envíos de los documentos de ventas hacia el operador de servicios electrónicos NUBEFAC según lo establecido por la Superintendencia de Administración Tributaria-Sunat.
- B. En la presente tesis cuando defino el objetivo “Gestionar el almacenamiento de los datos correspondientes a la emisión de documentos electrónicos de ventas” necesito utilizar un gestor de base de datos de datos que tenga las características suficientes para alcanzar dicho objetivo. Como nos indica en su página oficial del Posgresql este software es de código abierto disponible sin costo alguno, pero sobretodo lo importante es el soporte que brinda este gestor de base de datos.

- C. Sustentado en la experiencia y antecedente con la tesis denominada “Sistema de facturación electrónica para la gestión de comprobantes de pago basado en ISO/IEC19845:2015 en Acgenesys S.A.C” de Theany Kristy Navarro Flores me brinda un soporte para considerar el ámbito del objetivo de emitir los documentos electrónicos solicitados por la Sunat como son Boleta electrónica, Factura electrónica y Nota de Crédito electrónica.

Capítulo VI:

CONCLUSIONES

CAPÍTULO VI: CONCLUSIONES

1. Al establecer los requerimientos funcionales de la empresa permitió establecer las características para la conexión y emisión remota de los documentos de ventas a la OSE Nubefact.
2. Al implementar el módulo implementado de envíos de documentos de ventas consumiendo el servicio SOAP del Operador de Servicios Electrónicos NUBEFAC se cumple con la normativa para la emisión y registro de comprobantes de pago.
3. La gestión del almacenamiento de los datos correspondientes a la emisión de documentos electrónicos de ventas se realizó correctamente utilizando el gestor de base de datos indicado en este informe.
4. La emisión de los documentos electrónicos solicitados por la Sunat como son Boleta electrónica, Factura electrónica y Nota de Crédito electrónica se realizaron sin ningún tipo de inconveniente aplicando la validación y asignación del recurso mediante el módulo implementado.

Capítulo VII

RECOMENDACIONES

CAPÍTULO VII: RECOMENDACIONES

1. Revisar en forma periódica las normativas de la Sunat respecto al proceso de validación de las OSEs debido a que podrían variar.
2. Se debe evaluar en forma trimestral la consistencia de los requerimientos funcionales debido a posibles cambios en las políticas de administración tributaria.
3. Realizar los monitoreos correspondientes mediante reportes y verificación de datos, así como de conectividad con la OSE para verificar la estabilidad de la misma.
4. En forma mensual se debe verificar la consistencia de los datos en la base de datos, así como realizar el mantenimiento respectivo.
5. Siempre estar atentos ante cualquier modificatoria de Ley respecto a la emisión de comprobantes de pago y nota de crédito.

Capítulo VIII

PROPUESTA TECNOLÓGICA

CAPÍTULO VIII: PROPUESTA TECNOLÓGICA

8.1. Generalidades.

La tecnología crece a pasos agigantados y se extiende en todos los contextos de las actividades del ser humano. Estas actividades comprenden desde las domésticas hasta las que desarrollan y realizan las empresas e industrias transnacionales, así como los organismos estatales y el mismo Estado o Gobierno Nacional.

El Estado Peruano formaliza e impulsa la recuadación tributaria permitiendo que las empresas, negocios, industrias y todo aquel negocio formalizado puedan utilizar los diferentes medios y redes informáticas para realizar sus declaraciones de pagos de impuestos, pagos de tributos, así como la emisión y validación de comprobantes como boletas, facturas y notas de créditos que permita el registro automático de las operaciones de ventas de la empresa en forma automática y segura.

Considerando lo último indicado y tomando en cuenta la problemática descrita en este trabajo de investigación es necesario utilizar la tecnología que permita el diseño para exponer operaciones individuales denominados web services, así como un soporte de comunicación de los sistemas legados de una organización y esto lo brinda el SOAP. Con esta tecnología se logra una comunicación fluida, ligera, segura y rápida con el operador de servicios que obliga entidad recaudadora de impuestos del estado y que en esta oportunidad se contrata los servicios de la empresa NUBEFACT.

El restaurant “Zarandaja” dispondrá una aplicación web robusta que cumpla con las disposiciones del Estado, la emisión del documento electrónico de venta y la validación del mismo por lo cual el cliente y el mismo negocio gozarán de la comodidad y la tranquilidad que todo negocio necesita.

La empresa dispondrá en todo momento del CPE (comprobante de pago electrónico) según como lo indica en su portal la sunat:

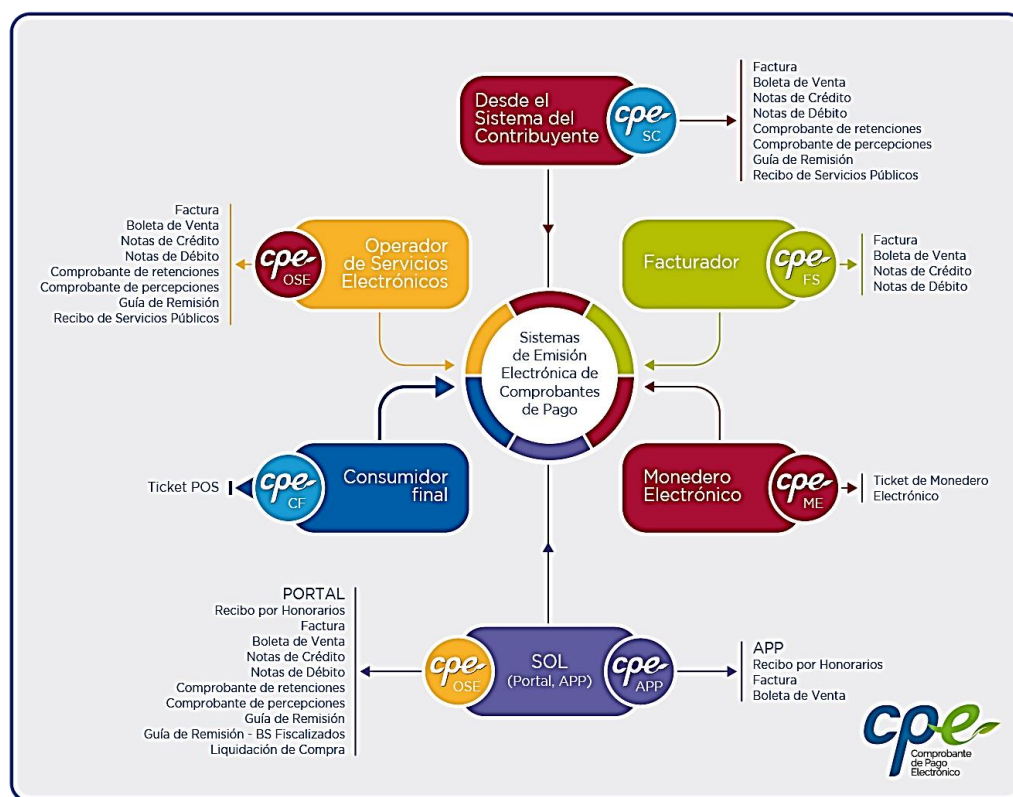


Imagen No. 52 Sistemas de emisión electrónica de comprobantes de Pago

Fuente: https://cpe.sunat.gob.pe/informacion_general/cpe

8.2. Análisis.

El análisis es la etapa en el cual se conoce las características del sistema actual, se establece o se responde a la pregunta ¿Qué sucede? ¿cómo es? Para ello utilicé el Lenguaje Unificado de Modelamiento(UML) para describir gráficamente los procesos fundamentales del sistema es decir las funciones a evaluar mediante el diagrama de casos de uso. También se elaboró los diagramas de estados de los dos documentos que se generan en este sistema, cuyo tratamiento es el núcleo de todos los procesos involucrados y sobretodo son los documentos que definen la continuidad o finalización del proceso. Los documentos son los diagramas de estados de Comprobante y del CDR SUNAT.

También se incluye los diagramas de actividades los cuales no son mas flujogramas que explican en forma secuencial las acciones o actividades que se desarrollan en un proceso específico. Dichos diagramas son básicamente el consultar el documento electrónico y consultar la validez del CPE.

Por último, se muestra los diagramas de secuencia de los dos documentos mencionados anteriormente para establecer las relaciones entre entidades u objetos del sistema y cuando indico entidades u objetos no me refiero a tablas de base de datos sino a componentes del mundo real que tienen sus propias características y comportamiento.

8.3. Diseño.

En la etapa de diseño y cuyas imágenes se encuentran en el capítulo de resultados, se plasma el diagrama físico de la base de datos el cual está conformado por 23 tablas todas ellas nombradas con el prefijo “tb”. Esto motivó a que el diagrama se subdivide en 04 partes para que se pueda apreciar y comprender.

También incluyo en este apartado los diccionarios de datos que está conformado por el detalle de cada tabla y en el cual se especifican el

nombre del campo, tipo de campo, especificación de las llaves principales y foráneas.

La construcción de dichas tablas, así como las relaciones de la base de datos se hicieron a través del gestor de base datos postgresql.

También agrego un segmento del código en Json para la consulta del DNI o Ruc del cliente ante la SUNAT y la RENIEC para identificar y validez al cliente y con ello generar el comprobante de pago con los datos verificados del mismo.

8.4. Implementación del sistema.

Utilizando el lenguaje PHP el cual es un lenguaje a nivel del servidor se implementó los formularios para la navegación web. Es bueno recordar que el PHP nos permite crear aplicaciones a nivel de server por lo cual el código generado en las páginas de los clientes es el resultado de procesar el código inicial de las páginas almacenadas en el servidor. De esta manera se protege el código de personas maliciosas para copiar o plagiar código.

En esta etapa se construyó el sistema y sobre el cual se describen las pantallas principales. Dichas pantallas con sus respectivas descripciones se encuentran en el apartado o capítulo de resultados. También se visualiza un segmento del código en PHP sobre el tratamiento de la aplicación respecto a los datos de Consulta de Datos de Cliente en Sunat y Reniec.

8.4. Presupuesto.

En el capítulo de resultados se definió los costos del proyecto. A continuación, se define la recuperación económica de la empresa

MES	COSTO A RECUPERAR	NUEVOS CLIENTES	PROM. GANANCIA POR CLIENTE	GANANCIA TOTAL PROMEDIO	SALDO A RECUPERAR
1	S/ 4,260.00	25	S/ 35.00	S/ 875.00	S/ 3,385.00
2	S/ 3,385.00	20	S/ 35.00	S/ 700.00	S/ 2,685.00
3	S/ 2,685.00	20	S/ 35.00	S/ 700.00	S/ 1,985.00
4	S/ 1,985.00	20	S/ 35.00	S/ 700.00	S/ 1,285.00
5	S/ 1,285.00	20	S/ 35.00	S/ 700.00	S/ 585.00
6	S/ 585.00	20	S/ 35.00	S/ 700.00	-S/ 115.00

Tabla 3 Recuperación - Presupuesto

Fuente: Propia Autoría

Considerando que por mes el promedio de nuevos clientes es de 25 y la ganancia por cada uno de ellos en promedio es de 35.00 podremos indicar que al 6to mes se recupera lo invertido en este proyecto.

Flujo de Caja.

Nro.	Forma de pago
1.00	Efectivo
2.00	Tarjeta
3.00	Transferencia

Tabla 4 Indicadores de ingreso.

Fuente: Propia Autoría

“Implementación del Módulo de Envío de Documentos de Ventas del Restaurant Zarandaja de la Ciudad de Chiclayo Consumiendo el servicio SOAP del Operador de Servicios Electrónicos NUBEFACT.”

FECHA	CONCEPTO	Cod. Valor	ENTRADAS	SALIDAS	SALDO
1/11/2020	Ventas	1.00	S/ 450.00		S/ 450.00
1/11/2020	Ventas	2.00	S/ 350.00		S/ 800.00
1/11/2020	Pagos de compras directas	1.00		S/ 700.00	S/ 100.00
2/11/2020	Ventas	1.00	S/ 456.00		S/ 556.00
2/11/2020	Ventas	2.00	S/ 340.00		S/ 896.00
3/11/2020	Ventas	1.00	S/ 780.00		S/ 1,676.00
4/11/2020	Ventas	2.00	S/ 890.00		S/ 1,786.00
5/11/2020	Pagos proveedores	3.00		S/ 2,100.00	-S/ 314.00
5/11/2020	Ventas	1.00	S/ 780.00		S/ 466.00
5/11/2020	Ventas	2.00	S/ 700.00		S/ 1,166.00
6/11/2020	Ventas	1.00	S/ 640.00		S/ 1,806.00
6/11/2020	Ventas	2.00	S/ 300.00		S/ 2,106.00
7/11/2020	Pagos de compras directas	3.00		S/ 2,400.00	-S/ 294.00
7/11/2020	Ventas	1.00	S/ 600.00		S/ 306.00
7/11/2020	Ventas	2.00	S/ 350.00		S/ 656.00
8/11/2020	Ventas	1.00	S/ 967.00		S/ 1,623.00
8/11/2020	Ventas	2.00	S/ 600.00		S/ 2,223.00
9/11/2020	Ventas	1.00	S/ 600.00		S/ 2,823.00
9/11/2020	Ventas	2.00	S/ 700.00		S/ 3,523.00
10/11/2020	Ventas	1.00	S/ 780.00		S/ 4,303.00
10/11/2020	Ventas	2.00	S/ 678.00		S/ 4,981.00
11/11/2020	Ventas	1.00	S/ 679.00		S/ 5,660.00
11/11/2020	Ventas	2.00	S/ 909.00		S/ 6,569.00
12/11/2020	Pagos de compras directas	2.00		S/ 2,300.00	S/ 4,269.00
12/11/2020	Ventas	1.00	S/ 567.00		S/ 4,836.00
12/11/2020	Ventas	2.00	S/ 789.00		S/ 5,625.00
13/11/2020	Pagos proveedores	3.00		S/ 4,500.00	S/ 1,125.00
13/11/2020	Ventas	1.00	S/ 600.00		S/ 1,725.00
13/11/2020	Ventas	2.00	S/ 600.00		S/ 2,325.00
14/11/2020	Ventas	1.00	S/ 560.00		S/ 2,885.00
14/11/2020	Ventas	2.00	S/ 568.00		S/ 3,453.00
15/11/2020	Ventas	1.00	S/ 456.00		S/ 3,909.00
15/11/2020	Ventas	2.00	S/ 670.00		S/ 4,579.00
16/11/2020	Ventas	1.00	S/ 560.00		S/ 5,139.00
16/11/2020	Ventas	2.00	S/ 455.00		S/ 5,594.00
17/11/2020	Ventas	1.00	S/ 555.00		S/ 6,149.00
17/11/2020	Ventas	2.00	S/ 420.00		S/ 6,569.00
18/11/2020	Ventas	1.00	S/ 670.00		S/ 7,239.00
18/11/2020	Ventas	2.00	S/ 456.00		S/ 7,695.00
19/11/2020	Ventas	1.00	S/ 560.00		S/ 8,255.00
19/11/2020	Ventas	2.00	S/ 560.00		S/ 8,815.00
19/11/2020	Pago de servicios	3.00		S/ 6,200.00	S/ 2,615.00
20/11/2020	Ventas	1.00	S/ 567.00		S/ 3,182.00
20/11/2020	Ventas	2.00	S/ 456.00		S/ 3,638.00
21/11/2020	Ventas	1.00	S/ 345.00		S/ 3,983.00
21/11/2020	Ventas	2.00	S/ 675.00		S/ 4,658.00
22/11/2020	Ventas	1.00	S/ 700.00		S/ 5,358.00
22/11/2020	Ventas	2.00	S/ 456.00		S/ 5,814.00
23/11/2020	Ventas	1.00	S/ 890.00		S/ 6,704.00

Tabla 5 Flujo de caja parte 1

Fuente: Propia autoría.

“Implementación del Módulo de Envío de Documentos de Ventas del Restaurant Zarandaja de la Ciudad de Chiclayo Consumiendo el servicio SOAP del Operador de Servicios Electrónicos NUBEFAC.T.”

FECHA	CONCEPTO	Cod. Valor	ENTRADAS	SALIDAS	SALDO
23/11/2020	Ventas	2.00	S/ 489.00		S/ 7,193.00
23/11/2020	Pagos proveedores	3.00	S/ 2,900.00		S/ 10,093.00
23/11/2020	Pagos de compras directas	1.00	S/ 1,900.00		S/ 11,993.00
24/11/2020	Ventas	1.00	S/ 800.00		S/ 12,793.00
24/11/2020	Ventas	2.00	S/ 450.00		S/ 13,243.00
25/11/2020	Ventas	1.00	S/ 670.00		S/ 13,913.00
25/11/2020	Ventas	2.00	S/ 780.00		S/ 14,693.00
26/11/2020	Ventas	1.00	S/ 600.00		S/ 15,293.00
26/11/2020	Ventas	2.00	S/ 700.00		S/ 15,993.00
26/11/2020	Pagos sueldos	3.00	S/ 350.00	S/ 12,000.00	S/ 4,343.00
27/11/2020	Ventas	1.00	S/ 550.00		S/ 4,893.00
27/11/2020	Ventas	2.00	S/ 675.00		S/ 5,568.00
28/11/2020	Ventas	1.00	S/ 550.00		S/ 6,118.00
28/11/2020	Ventas	2.00	S/ 660.00		S/ 6,778.00
29/11/2020	Ventas	1.00	S/ 500.00		S/ 7,278.00
29/11/2020	Ventas	2.00	S/ 560.00		S/ 7,838.00
30/11/2020	Ventas	1.00	S/ 700.00		S/ 8,538.00
30/11/2020	Ventas	2.00	S/ 690.00		S/ 9,228.00
30/11/2020	Pagos proveedores	1.00	S/ 6,000.00	S/ 4,800.00	S/ 10,428.00
30/11/2020	Pagos de compras directas	2.00	S/ 2,200.00	S/ 2,200.00	S/ 10,428.00

Tabla 6 Flujo de caja Parte 2

Fuente: Propia autoría.

Capítulo IX:

REFERENCIAS BIBLIOGRÁFICAS

CAPITULO IX: BIBLIOGRAFIA

PHP

- Bravo G. y Gutiérrez A. (2004). PHP4 a través de ejemplos. México. Editorial Alfaomega
- Cibelli C. (2012). PHP programación avanzada para profesionales. Buenos Aires, Argentina. Editorial. Alfaomega.
- Welling, Luke (2017) Desarrollo Web con PHP y MySQL. Editorial Anaya. Quinta Edición. Quinta Edición. España
- Dayle Rees (2016). Lenguaje PHP para todos. Editorial PullHits. EE.UU..
- Welling, L.y Thomson L (2016). Desarrollo Web con PHP y MySQL. Quinta edición. McGrawHalls. España.

Postgresql

- Araya P.(2012). Administración de SGBD Postgresql. Editorial Open libra. Colombia.
- Ginesta Gibert, M. & Mora Pérez, O. (2009), Base de Datos PostgreSQL, España: Autoedición.
- Lardièrre L. (2019) PostgreSQL. Administración y explotación de sus bases de datos. Editorial Epsilon. España.
- Orbegozo B. (2018). Curso Practico Avanzado De Postgresql - La Base De Datos Mas Potente. Editorial Alfaomega. España.

Visual Net

- Groussard T. (2017). Visual Basic 2012 (VB.NET) Los fundamentos del lenguaje - Desarrollar con Visual Studio 2012. Editorial ENI. España
- Ceballos, F. (2007). Visual Basic.Net Curso De Programación. Editorial RA-MA. España-
- Putier S. (2015). VB.NET y Visual Studio 2015 Los Fundamentos del Lenguaje. Ediciones Eni. España.

SOAP

- Aransay, A. L. (2009). Revisión de los Servicios Web SOAP/REST: Características y Rendimiento.
- Arce, S. (2012). Desarrollo e implementación de una API RESTful aplicada a un Sistema Académico Universitario. Mendoza.
- Barbero., P. C. (2012). SERVICIOS WEB BASADOS EN SOAP Y REST . Lima.

General.

- Chávez, Y. & Requena, I. (2017). Emisión de los Comprobantes de Pago Electrónicos para Evitar las Sanciones Tributarias en la Empresa Binresa S.A.C., 2017. Tesis de pregrado. Universidad Tecnológica del Perú. Lima. Perú.
- Marqués, A. M. (2011). Bases de Datos. Castellón de la Plana: Universitat Jaume I.
- Matos, C. (2017). Factores que limitan el uso de la factura electrónica por la micro y pequeña empresa (MYPE) del distrito de La Victoria – 2016. Tesis de pregrado. Universidad César Vallejo. Lambayeque. Perú.
- Navarro, T. (2017). Sistema de facturación electrónica para la gestión de comprobantes de pago basado en ISO/IEC19845:2015 en Acgenesys S.A.C. Tesis de pregrado. Universidad César Vallejo. Lambayeque.
- Rumbaugh J., Jacobson I., Booch G. (2009) "El Lenguaje Unificado De Modelado Manual De Referencia", Editorial Pearson Educación, Madrid.
- Weitzenfeld, Alfredo (2013). Ingeniería Del Software Orientada A Objetos Con Uml, Java E Internet. I Edición. Madrid

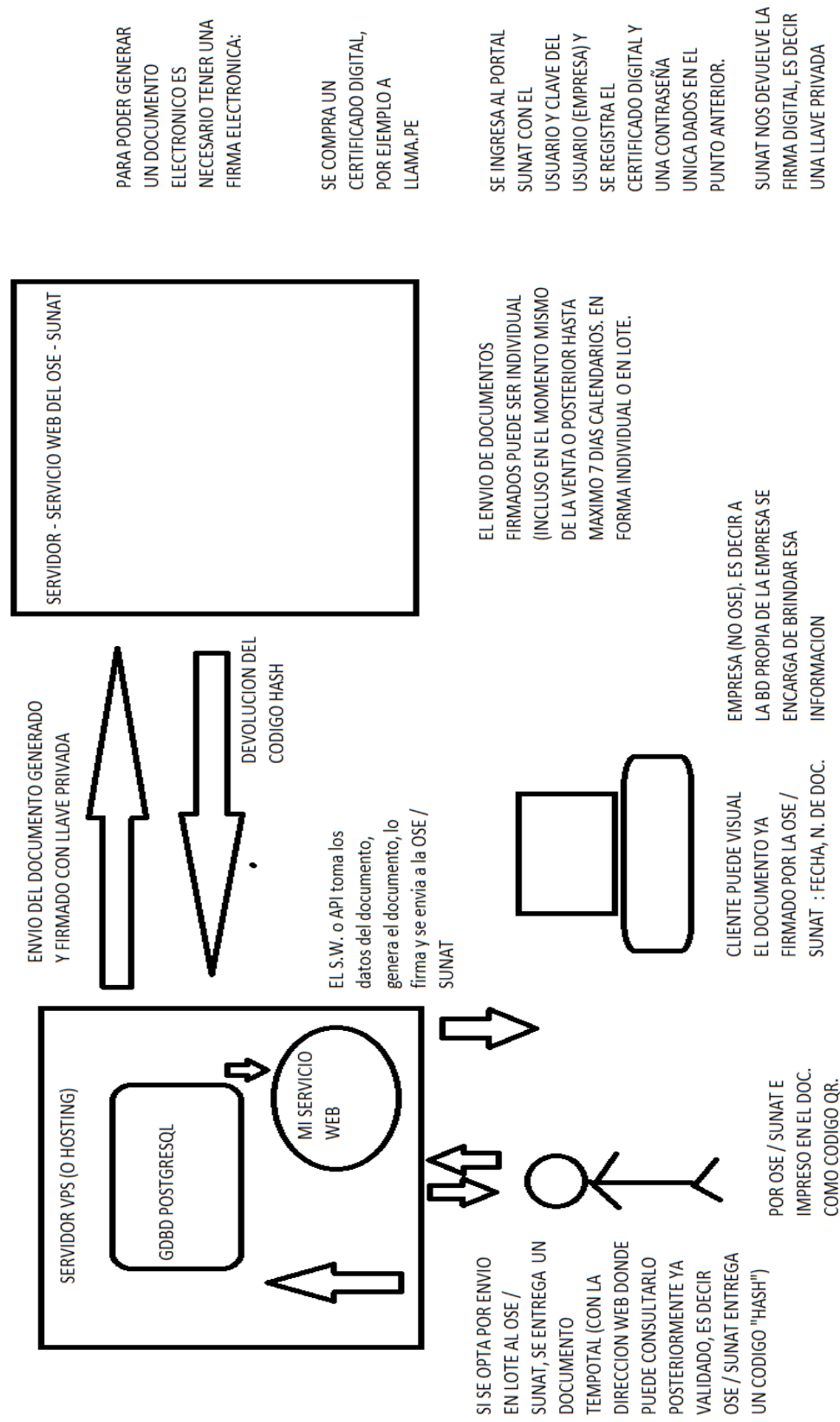
VÍNCULOS:

- IBM Knowledge Center
(https://www.ibm.com/support/knowledgecenter/es/SSKM8N_8.0.0/com.ibm.e.tools.mft.doc/ac55770_.htm, recuperado el 11 30 de octubre de 2019.

- Joan Vila(Departament d'Informàtica de Sistemes i Computadors - Universitat Politècnica de València) en la fuente https://poliformat.upv.es/access/content/group/OCW_6069_2008/T4.-%20Integraci%C3%B3n%20de%20aplicaciones%20en%20la%20web/Teor%C3%ADa/soap.pdf , recuperado el 30 de octubre de 2019.
- <http://www.v-espino.com/~chema/daw2/soap/webservices.htm>. Recuperado el 31 de octubre de 2019.
- <http://json.org>. Recuperado el 20 de octubre de 2020.
- <https://www.nubefact.com>. Recuperado el 20 de octubre de 2019,
- <https://www.php.net/manual/es/intro-what-is.php>. Recuperado el 30 de octubre de 2019.
- <http://repositorio.espm.edu.ec/bitstream/42000/86/1/TESIS%20L%C3%93PEZ%20MORA%20WILSON%20FABI%C3%81N%20-.pdf>. Recuperado el 31 de octubre de 2019.
- <https://www.php.net/manual/es/book.pgsql.php>. Recuperado el 30 de octubre de 2019.
- www.sunat.gob.pe. Recuperado el 30 de octubre de 2019.
- http://cpe.sunat.gob.pe/informacion_general/operador_servicios_electronicos. Recuperado el 31 de octubre de 2019.

ANEXO

Anexo 01. Croquis de proceso de emisión y validación de comprobante electrónico.



Anexo 02: Script de la base de datos.

```
-- Database generated with pgModeler (PostgreSQL Database Modeler).
-- pgModeler version: 0.9.2-alpha1
-- PostgreSQL version: 9.4
-- Project Site: pgmodeler.io
-- Model Author: ---

-- Database creation must be done outside a multicommand file.
-- These commands were put in this file only as a convenience.
-- -- object: bdrestaurante_talara | type: DATABASE --
-- -- DROP DATABASE IF EXISTS bdrestaurante_talara;
-- CREATE DATABASE bdrestaurante_talara
--     ENCODING = 'UTF8'
--     LC_COLLATE = 'Spanish_Peru.1252'
--     LC_CTYPE = 'Spanish_Peru.1252'
--     TABLESPACE = pg_default
--     OWNER = postgres;
-- -- ddl-end --
-- COMMENT ON DATABASE bdrestaurante_talara IS '
-- ';
-- -- ddl-end --
--

-- object: public.tbestadomesa_id_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbestadomesa_id_seq CASCADE;
CREATE SEQUENCE public.tbestadomesa_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
    CACHE 1
    NO CYCLE
    OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbestadomesa_id_seq OWNER TO postgres;
-- ddl-end --

-- object: public.tbestadomesa | type: TABLE --
-- DROP TABLE IF EXISTS public.tbestadomesa CASCADE;
CREATE TABLE public.tbestadomesa (
    id integer NOT NULL DEFAULT nextval('public.tbestadomesa_id_seq'::regclass),
    nombre character varying(20) NOT NULL,
    color bigint NOT NULL DEFAULT 0,
    CONSTRAINT tbestadomesa_pk PRIMARY KEY (id)
);
-- ddl-end --
COMMENT ON TABLE public.tbestadomesa IS '1 = Disponible
2 = Ocupado / Sin Orden Comanda
3 = Ocupado / Con Comanda.';
-- ddl-end --
```

```
ALTER TABLE public.tbestadomesa OWNER TO postgres;
-- ddl-end --

-- object: public.tbmesa_id_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbmesa_id_seq CASCADE;
CREATE SEQUENCE public.tbmesa_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
    CACHE 1
    NO CYCLE
    OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbmesa_id_seq OWNER TO postgres;
-- ddl-end --

-- object: public.tbmesa | type: TABLE --
-- DROP TABLE IF EXISTS public.tbmesa CASCADE;
CREATE TABLE public.tbmesa (
    id integer NOT NULL DEFAULT nextval('public.tbmesa_id_seq'::regclass),
    etiqueta character varying(20) NOT NULL,
    x integer NOT NULL DEFAULT 0,
    y integer NOT NULL DEFAULT 0,
    ancho integer NOT NULL DEFAULT 0,
    alto integer NOT NULL DEFAULT 0,
    id_tbestadomesa integer NOT NULL,
    CONSTRAINT tbmesa_pk PRIMARY KEY (id)
);
-- ddl-end --
ALTER TABLE public.tbmesa OWNER TO postgres;
-- ddl-end --

-- object: public.tbcomanda_id_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbcomanda_id_seq CASCADE;
CREATE SEQUENCE public.tbcomanda_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
    CACHE 1
    NO CYCLE
    OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbcomanda_id_seq OWNER TO postgres;
-- ddl-end --

-- object: public.tbcomanda | type: TABLE --
-- DROP TABLE IF EXISTS public.tbcomanda CASCADE;
CREATE TABLE public.tbcomanda (
    id integer NOT NULL DEFAULT nextval('public.tbcomanda_id_seq'::regclass),
    fecha date NOT NULL,
    hora time NOT NULL,
```

```
tiposervicio integer NOT NULL,
nombre_ref character varying(50) NOT NULL DEFAULT "",
telef_ref character varying(20) NOT NULL DEFAULT "",
correo_ref character varying NOT NULL DEFAULT "",
direccion_ref_delivery character varying(100) NOT NULL DEFAULT "",
referencia_dir_delivery character varying(100) NOT NULL DEFAULT "",
id_tbmesa integer,
idusuario integer NOT NULL,
estado boolean NOT NULL DEFAULT true,
id_tbpersona integer,
cerrado boolean NOT NULL DEFAULT false,
CONSTRAINT tbcomanda_pk PRIMARY KEY (id)
);
-- ddl-end --
COMMENT ON COLUMN public.tbcomanda.tiposervicio IS '1 = Mesa
2 = Para llevar
3 = Delivery';
-- ddl-end --
ALTER TABLE public.tbcomanda OWNER TO postgres;
-- ddl-end --

-- object: public.tbdetalle_comanda_id_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbdetalle_comanda_id_seq CASCADE;
CREATE SEQUENCE public.tbdetalle_comanda_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
    CACHE 1
    NO CYCLE
    OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbdetalle_comanda_id_seq OWNER TO postgres;
-- ddl-end --

-- object: public.tbdetalle_comanda | type: TABLE --
-- DROP TABLE IF EXISTS public.tbdetalle_comanda CASCADE;
CREATE TABLE public.tbdetalle_comanda (
    id integer NOT NULL DEFAULT
nextval('public.tbdetalle_comanda_id_seq'::regclass),
    id_tbcomanda integer NOT NULL,
    precio_unit numeric(7,2) NOT NULL DEFAULT 0.00,
    cant numeric(5,2) NOT NULL DEFAULT 0.00,
    igv_incluido numeric(6,2) NOT NULL DEFAULT 0.00,
    isc_incluido numeric(6,2) NOT NULL DEFAULT 0.00,
    id_tbproducto integer NOT NULL
);
-- ddl-end --
ALTER TABLE public.tbdetalle_comanda OWNER TO postgres;
-- ddl-end --

-- object: public.tbcategoria_id_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbcategoria_id_seq CASCADE;
```

```
CREATE SEQUENCE public.tbcategoria_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
    CACHE 1
    NO CYCLE
    OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbcategoria_id_seq OWNER TO postgres;
-- ddl-end --

-- object: public.tbcategoria | type: TABLE --
-- DROP TABLE IF EXISTS public.tbcategoria CASCADE;
CREATE TABLE public.tbcategoria (
    id integer NOT NULL DEFAULT nextval('public.tbcategoria_id_seq'::regclass),
    nombre character varying(50) NOT NULL,
    CONSTRAINT tbcategoria_pk PRIMARY KEY (id)
);
-- ddl-end --
ALTER TABLE public.tbcategoria OWNER TO postgres;
-- ddl-end --

-- object: public.tbproducto_id_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbproducto_id_seq CASCADE;
CREATE SEQUENCE public.tbproducto_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
    CACHE 1
    NO CYCLE
    OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbproducto_id_seq OWNER TO postgres;
-- ddl-end --

-- object: public.tbproducto | type: TABLE --
-- DROP TABLE IF EXISTS public.tbproducto CASCADE;
CREATE TABLE public.tbproducto (
    id integer NOT NULL DEFAULT nextval('public.tbproducto_id_seq'::regclass),
    tipoprod integer NOT NULL DEFAULT 1,
    nombre character varying(150) NOT NULL,
    precio numeric(7,2) NOT NULL DEFAULT 0.00,
    afecto_igv boolean NOT NULL DEFAULT true,
    afecto_isc boolean NOT NULL DEFAULT false,
    foto character varying(150) NOT NULL DEFAULT "",
    observacion character varying(150) NOT NULL DEFAULT "",
    id_tbcategoria integer NOT NULL,
    estado boolean NOT NULL DEFAULT true,
    CONSTRAINT tb_producto_pk PRIMARY KEY (id)
);
```

```
-- ddl-end --
COMMENT ON COLUMN public.tbproducto.tipoprod IS '1 = Producto Terminado.
2 = Servicio';
-- ddl-end --
ALTER TABLE public.tbproducto OWNER TO postgres;
-- ddl-end --

-- object: public.tbpersona_id_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbpersona_id_seq CASCADE;
CREATE SEQUENCE public.tbpersona_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
    CACHE 1
    NO CYCLE
    OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbpersona_id_seq OWNER TO postgres;
-- ddl-end --

-- object: public.tbpersona | type: TABLE --
-- DROP TABLE IF EXISTS public.tbpersona CASCADE;
CREATE TABLE public.tbpersona (
    id integer NOT NULL DEFAULT nextval('public.tbpersona_id_seq'::regclass),
    tipoper character(1) NOT NULL,
    razonsocial character varying(100) NOT NULL DEFAULT '',
    ape_pat character varying(50) NOT NULL DEFAULT '',
    ape_mat character varying(50) NOT NULL DEFAULT '',
    nombre character varying(50) NOT NULL DEFAULT '',
    nrodoc character varying(20) NOT NULL DEFAULT '',
    direccion character varying(400) NOT NULL DEFAULT '',
    telefono character varying(15) NOT NULL DEFAULT '',
    email character varying(100) NOT NULL DEFAULT '',
    usuario character varying(20) NOT NULL DEFAULT '',
    clave character varying(20) NOT NULL DEFAULT '',
    password_hash text NOT NULL DEFAULT '',
    api_key character varying(32) NOT NULL DEFAULT '',
    id_tbtipo_documento_identidad integer NOT NULL,
    estado boolean NOT NULL DEFAULT true,
    CONSTRAINT tbpersona_pk PRIMARY KEY (id)
);
-- ddl-end --
ALTER TABLE public.tbpersona OWNER TO postgres;
-- ddl-end --

-- object: public.tbtipo_documento_identidad_id_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbtipo_documento_identidad_id_seq CASCADE;
CREATE SEQUENCE public.tbtipo_documento_identidad_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
```

```
CACHE 1
NO CYCLE
OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbtipo_documento_identidad_id_seq OWNER TO postgres;
-- ddl-end --

-- object: public.tbtipo_documento_identidad | type: TABLE --
-- DROP TABLE IF EXISTS public.tbtipo_documento_identidad CASCADE;
CREATE TABLE public.tbtipo_documento_identidad (
    id integer NOT NULL DEFAULT
nextval('public.tbtipo_documento_identidad_id_seq'::regclass),
    nombre character varying(30) NOT NULL,
    longitud integer NOT NULL,
    cod_sunat integer NOT NULL,
    CONSTRAINT tbtipo_documento_identidad_pk PRIMARY KEY (id)
);
-- ddl-end --
ALTER TABLE public.tbtipo_documento_identidad OWNER TO postgres;
-- ddl-end --

-- object: public.tbdocpago_ctacobrar_id_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbdocpago_ctacobrar_id_seq CASCADE;
CREATE SEQUENCE public.tbdocpago_ctacobrar_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
    CACHE 1
    NO CYCLE
    OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbdocpago_ctacobrar_id_seq OWNER TO postgres;
-- ddl-end --

-- object: public.tbdocpago_ctacobrar | type: TABLE --
-- DROP TABLE IF EXISTS public.tbdocpago_ctacobrar CASCADE;
CREATE TABLE public.tbdocpago_ctacobrar (
    id integer NOT NULL DEFAULT
nextval('public.tbdocpago_ctacobrar_id_seq'::regclass),
    id_tbtipodoc integer NOT NULL,
    nrodoc character varying(15) NOT NULL,
    importe money NOT NULL DEFAULT 0.00,
    igv money NOT NULL DEFAULT 0.00,
    isc money NOT NULL DEFAULT 0.00,
    vuelto money NOT NULL DEFAULT 0.00,
    fecha date NOT NULL,
    hora time NOT NULL,
    id_tbpersona integer NOT NULL,
    idusuario integer NOT NULL,
    estado boolean DEFAULT true,
    CONSTRAINT tbdocpago_ctacobrar_pk PRIMARY KEY (id)
```

```
);
-- ddl-end --
ALTER TABLE public.tbdocpago_ctacobrar OWNER TO postgres;
-- ddl-end --

-- object: public.tbtipodoc_id_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbtipodoc_id_seq CASCADE;
CREATE SEQUENCE public.tbtipodoc_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
    CACHE 1
    NO CYCLE
    OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbtipodoc_id_seq OWNER TO postgres;
-- ddl-end --

-- object: public.tbtipodoc | type: TABLE --
-- DROP TABLE IF EXISTS public.tbtipodoc CASCADE;
CREATE TABLE public.tbtipodoc (
    id integer NOT NULL DEFAULT nextval('public.tbtipodoc_id_seq'::regclass),
    nombre character varying(30) NOT NULL,
    abrev character varying(6) NOT NULL DEFAULT "",
    codigo_contable character varying(5) NOT NULL DEFAULT "",
    prefijos character varying(5) NOT NULL DEFAULT "",
    CONSTRAINT tbtipodoc_pk PRIMARY KEY (id)
);
-- ddl-end --
ALTER TABLE public.tbtipodoc OWNER TO postgres;
-- ddl-end --

-- object: public.tbdetalle_docpago_ctacobrar_impreso_id_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbdetalle_docpago_ctacobrar_impreso_id_seq CASCADE;
CREATE SEQUENCE public.tbdetalle_docpago_ctacobrar_impreso_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
    CACHE 1
    NO CYCLE
    OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbdetalle_docpago_ctacobrar_impreso_id_seq OWNER TO postgres;
-- ddl-end --

-- object: public.tbdetalle_docpago_ctacobrar_impreso | type: TABLE --
-- DROP TABLE IF EXISTS public.tbdetalle_docpago_ctacobrar_impreso CASCADE;
CREATE TABLE public.tbdetalle_docpago_ctacobrar_impreso (
```

```
        id integer NOT NULL DEFAULT
nextval('public.tbdetalle_docpago_ctacobrar_impreso_id_seq'::regclass),
        id_tbdocpago_ctacobrar integer NOT NULL,
        precio numeric(7,2) NOT NULL DEFAULT 0.00,
        cant numeric(5,2) NOT NULL DEFAULT 0.00,
        igv_incluido numeric(6,2) NOT NULL DEFAULT 0.00,
        isc_incluido numeric(6,2) NOT NULL DEFAULT 0.00,
        id_tbproducto integer NOT NULL,
        CONSTRAINT tbdetalle_docpago_ctacobrar_impreso_pk PRIMARY KEY (id)

);
-- ddl-end --
ALTER TABLE public.tbdetalle_docpago_ctacobrar_impreso OWNER TO postgres;
-- ddl-end --

-- object: public.tbmediopago_id_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbmediopago_id_seq CASCADE;
CREATE SEQUENCE public.tbmediopago_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
    CACHE 1
    NO CYCLE
    OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbmediopago_id_seq OWNER TO postgres;
-- ddl-end --

-- object: public.tbmediopago | type: TABLE --
-- DROP TABLE IF EXISTS public.tbmediopago CASCADE;
CREATE TABLE public.tbmediopago (
    id integer NOT NULL DEFAULT nextval('public.tbmediopago_id_seq'::regclass),
    nombre character varying(50) NOT NULL,
    CONSTRAINT tbmediopago_pk PRIMARY KEY (id)

);
-- ddl-end --
ALTER TABLE public.tbmediopago OWNER TO postgres;
-- ddl-end --

-- object: public.tbmediopago_tbdocpago_ctacobrar | type: TABLE --
-- DROP TABLE IF EXISTS public.tbmediopago_tbdocpago_ctacobrar CASCADE;
CREATE TABLE public.tbmediopago_tbdocpago_ctacobrar (
    id_tbmediopago integer NOT NULL,
    id_tbdocpago_ctacobrar integer NOT NULL,
    importe money NOT NULL DEFAULT 0.00,
    observacion character varying(100) NOT NULL DEFAULT "",
    CONSTRAINT tbmediopago_tbdocpago_ctacobrar_pk PRIMARY KEY
(id_tbmediopago,id_tbdocpago_ctacobrar)

);
-- ddl-end --
ALTER TABLE public.tbmediopago_tbdocpago_ctacobrar OWNER TO postgres;
```

```
-- ddl-end --

-- object: public.tbsucursal_id_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbsucursal_id_seq CASCADE;
CREATE SEQUENCE public.tbsucursal_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
    CACHE 1
    NO CYCLE
    OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbsucursal_id_seq OWNER TO postgres;
-- ddl-end --

-- object: public.tbsucursal | type: TABLE --
-- DROP TABLE IF EXISTS public.tbsucursal CASCADE;
CREATE TABLE public.tbsucursal (
    id integer NOT NULL DEFAULT nextval('public.tbsucursal_id_seq'::regclass),
    direccion character varying(100) NOT NULL,
    CONSTRAINT tbsucursal_pk PRIMARY KEY (id)
);
-- ddl-end --
ALTER TABLE public.tbsucursal OWNER TO postgres;
-- ddl-end --

-- object: public.tbsucursalpcs_id_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbsucursalpcs_id_seq CASCADE;
CREATE SEQUENCE public.tbsucursalpcs_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
    CACHE 1
    NO CYCLE
    OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbsucursalpcs_id_seq OWNER TO postgres;
-- ddl-end --

-- object: public.tbsucursalpcs | type: TABLE --
-- DROP TABLE IF EXISTS public.tbsucursalpcs CASCADE;
CREATE TABLE public.tbsucursalpcs (
    id integer NOT NULL DEFAULT nextval('public.tbsucursalpcs_id_seq'::regclass),
    pc character varying(100) NOT NULL,
    printer_doc_comerciales character varying(150) NOT NULL DEFAULT "",
    printer_documentos character varying(150) NOT NULL DEFAULT "",
    id_tbsucursal integer NOT NULL,
    CONSTRAINT tbsucursalpcs_pk PRIMARY KEY (id)
);
-- ddl-end --
```

```
ALTER TABLE public.tbsucursalpcs OWNER TO postgres;
-- ddl-end --

-- object: public.tbseriesdoc_emitir_id_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbseriesdoc_emitir_id_seq CASCADE;
CREATE SEQUENCE public.tbseriesdoc_emitir_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
    CACHE 1
    NO CYCLE
    OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbseriesdoc_emitir_id_seq OWNER TO postgres;
-- ddl-end --

-- object: public.tbseriesdoc_emitir | type: TABLE --
-- DROP TABLE IF EXISTS public.tbseriesdoc_emitir CASCADE;
CREATE TABLE public.tbseriesdoc_emitir (
    id integer NOT NULL DEFAULT nextval('public.tbseriesdoc_emitir_id_seq'::regclass),
    id_tbtipodoc integer NOT NULL,
    serie character varying(4) NOT NULL DEFAULT '',
    ultimo_emitido integer NOT NULL DEFAULT 0,
    estado boolean NOT NULL DEFAULT true,
    id_tbsucursalpcs integer NOT NULL,
    CONSTRAINT tbseriesdoc_emitir_pk PRIMARY KEY (id)
);
-- ddl-end --
ALTER TABLE public.tbseriesdoc_emitir OWNER TO postgres;
-- ddl-end --

-- object: public.tbconfiguracion_id_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbconfiguracion_id_seq CASCADE;
CREATE SEQUENCE public.tbconfiguracion_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
    CACHE 1
    NO CYCLE
    OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbconfiguracion_id_seq OWNER TO postgres;
-- ddl-end --

-- object: public.tbconfiguracion | type: TABLE --
-- DROP TABLE IF EXISTS public.tbconfiguracion CASCADE;
CREATE TABLE public.tbconfiguracion (
    id integer NOT NULL DEFAULT nextval('public.tbconfiguracion_id_seq'::regclass),
    key character varying(8) NOT NULL DEFAULT '',
    descripcion character varying(100) NOT NULL DEFAULT '',
    valor character varying(150) NOT NULL DEFAULT '',
```

```
        CONSTRAINT tbconfiguracion_pk PRIMARY KEY (id)

);
-- ddl-end --
ALTER TABLE public.tbconfiguracion OWNER TO postgres;
-- ddl-end --

-- object: public.tbrol_id_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbrol_id_seq CASCADE;
CREATE SEQUENCE public.tbrol_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
    CACHE 1
    NO CYCLE
    OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbrol_id_seq OWNER TO postgres;
-- ddl-end --

-- object: public.tbrol | type: TABLE --
-- DROP TABLE IF EXISTS public.tbrol CASCADE;
CREATE TABLE public.tbrol (
    id integer NOT NULL DEFAULT nextval('public.tbrol_id_seq'::regclass),
    nombre character varying(50) NOT NULL,
    CONSTRAINT tbrol_pk PRIMARY KEY (id)

);
-- ddl-end --
ALTER TABLE public.tbrol OWNER TO postgres;
-- ddl-end --

-- object: public.tbrol_persona | type: TABLE --
-- DROP TABLE IF EXISTS public.tbrol_persona CASCADE;
CREATE TABLE public.tbrol_persona (
    id_tbrol integer NOT NULL,
    id_tbpersona integer NOT NULL,
    CONSTRAINT tbrol_persona_pk PRIMARY KEY (id_tbrol,id_tbpersona)

);
-- ddl-end --
ALTER TABLE public.tbrol_persona OWNER TO postgres;
-- ddl-end --

-- object: public.tbctas_cobrar_idcta_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbctas_cobrar_idcta_seq CASCADE;
CREATE SEQUENCE public.tbctas_cobrar_idcta_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
    CACHE 1
    NO CYCLE
```

```
OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbctas_cobrar_idcta_seq OWNER TO postgres;
-- ddl-end --

-- object: public.tbctas_cobrar | type: TABLE --
-- DROP TABLE IF EXISTS public.tbctas_cobrar CASCADE;
CREATE TABLE public.tbctas_cobrar (
    idcta integer NOT NULL DEFAULT
nextval('public.tbctas_cobrar_idcta_seq'::regclass),
    tiposervicio integer NOT NULL DEFAULT 1,
    id_tbtipodoc integer NOT NULL,
    numdoc character varying(15) NOT NULL DEFAULT "",
    fecha date NOT NULL,
    fvenc date,
    nroletra integer NOT NULL DEFAULT 0,
    importe money NOT NULL DEFAULT 0,
    mora money NOT NULL DEFAULT 0.00,
    saldo money NOT NULL DEFAULT 0,
    igv money NOT NULL DEFAULT 0,
    isc money NOT NULL DEFAULT 0,
    idusuario integer NOT NULL,
    estado boolean NOT NULL DEFAULT true,
    cerrado boolean NOT NULL DEFAULT false,
    incluir_reg_compras boolean NOT NULL DEFAULT false,
    idempresa integer,
    idreferencia_origencta integer NOT NULL DEFAULT 0,
    glosa character varying(300) NOT NULL DEFAULT "",
    id_tbpersona integer NOT NULL,
    CONSTRAINT tbctas_cobrar_pk PRIMARY KEY (idcta),
    CONSTRAINT idx_unico_documento UNIQUE (tiposervicio,id_tbtipodoc,numdoc)
);
-- ddl-end --
COMMENT ON COLUMN public.tbctas_cobrar.tiposervicio IS '1 = Servicio venta comida.
Comanda.
2 = Otros servicios. Para hoteles: Lavanderia, Cochera, ';
-- ddl-end --
ALTER TABLE public.tbctas_cobrar OWNER TO postgres;
-- ddl-end --

-- object: public.tbdetpago_ctapagar_id_seq | type: SEQUENCE --
-- DROP SEQUENCE IF EXISTS public.tbdetpago_ctapagar_id_seq CASCADE;
CREATE SEQUENCE public.tbdetpago_ctapagar_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    START WITH 1
    CACHE 1
    NO CYCLE
    OWNED BY NONE;
-- ddl-end --
ALTER SEQUENCE public.tbdetpago_ctapagar_id_seq OWNER TO postgres;
-- ddl-end --
```

```
-- object: public.tbdetpago_ctapagar | type: TABLE --
-- DROP TABLE IF EXISTS public.tbdetpago_ctapagar CASCADE;
CREATE TABLE public.tbdetpago_ctapagar (
    id integer NOT NULL DEFAULT
nextval('public.tbdetpago_ctapagar_id_seq'::regclass),
    idcta_tbctas_cobrar integer NOT NULL,
    id_tbdopago_ctacobrar integer NOT NULL,
    importe money NOT NULL DEFAULT 0.00,
    signo character varying(1) NOT NULL DEFAULT '+',
    CONSTRAINT tbdetpago_ctapagar_pk PRIMARY KEY (id)
);
-- ddl-end --
ALTER TABLE public.tbdetpago_ctapagar OWNER TO postgres;
-- ddl-end --

-- object: idxid_tbdopago_ctacobrar | type: INDEX --
-- DROP INDEX IF EXISTS public.idxid_tbdopago_ctacobrar CASCADE;
CREATE INDEX idxid_tbdopago_ctacobrar ON public.tbdetpago_ctapagar
    USING btree
    (
        id_tbdopago_ctacobrar
    )
    WITH (FILLFACTOR = 90);
-- ddl-end --

-- object: idxid_tbcomanda | type: INDEX --
-- DROP INDEX IF EXISTS public.idxid_tbcomanda CASCADE;
CREATE INDEX idxid_tbcomanda ON public.tbdetalle_comanda
    USING btree
    (
        id_tbcomanda
    )
    WITH (FILLFACTOR = 90);
-- ddl-end --

-- object: idx_id_tbdopago_ctacobrar | type: INDEX --
-- DROP INDEX IF EXISTS public.idx_id_tbdopago_ctacobrar CASCADE;
CREATE INDEX idx_id_tbdopago_ctacobrar ON
public.tbdetalle_docpago_ctacobrar_impreso
    USING btree
    (
        id_tbdopago_ctacobrar
    )
    WITH (FILLFACTOR = 90);
-- ddl-end --

-- object: idx_id_tb_producto | type: INDEX --
-- DROP INDEX IF EXISTS public.idx_id_tb_producto CASCADE;
CREATE INDEX idx_id_tb_producto ON public.tbdetalle_docpago_ctacobrar_impreso
    USING btree
    (
        id_tbproducto
    )
```

```
)
  WITH (FILLFACTOR = 90);
-- ddl-end --

-- object: idxx_id_tb_producto | type: INDEX --
-- DROP INDEX IF EXISTS public.idxx_id_tb_producto CASCADE;
CREATE INDEX idxx_id_tb_producto ON public.tbdetalle_comanda
  USING btree
  (
    id_tbproducto
  )
  WITH (FILLFACTOR = 90);
-- ddl-end --

-- object: idx_idcta_tbctas_cobrar | type: INDEX --
-- DROP INDEX IF EXISTS public.idx_idcta_tbctas_cobrar CASCADE;
CREATE INDEX idx_idcta_tbctas_cobrar ON public.tbdetpago_ctapagar
  USING btree
  (
    idcta_tbctas_cobrar
  )
  WITH (FILLFACTOR = 90);
-- ddl-end --

-- object: public.tbfacturacionelectronica | type: TABLE --
-- DROP TABLE IF EXISTS public.tbfacturacionelectronica CASCADE;
CREATE TABLE public.tbfacturacionelectronica (
  tablaid_doc integer NOT NULL,
  hash character varying(255),
  num_ruc character varying(20) NOT NULL,
  tipo_doc character varying(10) NOT NULL,
  num_doc character varying(20) NOT NULL,
  fec_cargo date NOT NULL,
  fec_generado date,
  fec_envio date,
  des_observacion character varying(255),
  nom_archivo_xml character varying(255),
  cod_error character varying(20)
);
-- ddl-end --
ALTER TABLE public.tbfacturacionelectronica OWNER TO postgres;
-- ddl-end --

-- object: tbestadomesa_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbmesa DROP CONSTRAINT IF EXISTS tbestadomesa_fk
CASCADE;
ALTER TABLE public.tbmesa ADD CONSTRAINT tbestadomesa_fk FOREIGN KEY
(id_tbestadomesa)
REFERENCES public.tbmesa (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: tbmesa_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbcomanda DROP CONSTRAINT IF EXISTS tbmesa_fk CASCADE;
```

```
ALTER TABLE public.tbcomanda ADD CONSTRAINT tbmesa_fk FOREIGN KEY
(id_tbmesa)
REFERENCES public.tbmesa (id) MATCH FULL
ON DELETE SET NULL ON UPDATE CASCADE;
-- ddl-end --

-- object: tbpersona_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbcomanda DROP CONSTRAINT IF EXISTS tbpersona_fk
CASCADE;
ALTER TABLE public.tbcomanda ADD CONSTRAINT tbpersona_fk FOREIGN KEY
(id_tbpersona)
REFERENCES public.tbpersona (id) MATCH FULL
ON DELETE SET NULL ON UPDATE CASCADE;
-- ddl-end --

-- object: pk_idpersona_fkidusuario_tbcomanda | type: CONSTRAINT --
-- ALTER TABLE public.tbcomanda DROP CONSTRAINT IF EXISTS
pk_idpersona_fkidusuario_tbcomanda CASCADE;
ALTER TABLE public.tbcomanda ADD CONSTRAINT
pk_idpersona_fkidusuario_tbcomanda FOREIGN KEY (idusuario)
REFERENCES public.tbpersona (id) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: tbcomanda_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbdetalle_comanda DROP CONSTRAINT IF EXISTS tbcomanda_fk
CASCADE;
ALTER TABLE public.tbdetalle_comanda ADD CONSTRAINT tbcomanda_fk FOREIGN
KEY (id_tbcomanda)
REFERENCES public.tbcomanda (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: tbproducto_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbdetalle_comanda DROP CONSTRAINT IF EXISTS tbproducto_fk
CASCADE;
ALTER TABLE public.tbdetalle_comanda ADD CONSTRAINT tbproducto_fk FOREIGN KEY
(id_tbproducto)
REFERENCES public.tbproducto (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: tbcategoria_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbproducto DROP CONSTRAINT IF EXISTS tbcategoria_fk
CASCADE;
ALTER TABLE public.tbproducto ADD CONSTRAINT tbcategoria_fk FOREIGN KEY
(id_tbcategoria)
REFERENCES public.tbcategoria (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: tbtipo_documento_identidad_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbpersona DROP CONSTRAINT IF EXISTS
tbtipo_documento_identidad_fk CASCADE;
```

```
ALTER TABLE public.tbpersona ADD CONSTRAINT tbtipo_documento_identidad_fk
FOREIGN KEY (id_tbtipo_documento_identidad)
REFERENCES public.tbtipo_documento_identidad (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: tbtipodoc_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbdocpago_ctacobrar DROP CONSTRAINT IF EXISTS tbtipodoc_fk
CASCADE;
ALTER TABLE public.tbdocpago_ctacobrar ADD CONSTRAINT tbtipodoc_fk FOREIGN
KEY (id_tbtipodoc)
REFERENCES public.tbtipodoc (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: tbpersona_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbdocpago_ctacobrar DROP CONSTRAINT IF EXISTS
tbpersona_fk CASCADE;
ALTER TABLE public.tbdocpago_ctacobrar ADD CONSTRAINT tbpersona_fk FOREIGN
KEY (id_tbpersona)
REFERENCES public.tbpersona (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: fk_idusuario_pk_idpersona_tbtacobrar | type: CONSTRAINT --
-- ALTER TABLE public.tbdocpago_ctacobrar DROP CONSTRAINT IF EXISTS
fk_idusuario_pk_idpersona_tbtacobrar CASCADE;
ALTER TABLE public.tbdocpago_ctacobrar ADD CONSTRAINT
fk_idusuario_pk_idpersona_tbtacobrar FOREIGN KEY (idusuario)
REFERENCES public.tbpersona (id) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: tbproducto_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbdetalle_docpago_ctacobrar_impreso DROP CONSTRAINT IF
EXISTS tbproducto_fk CASCADE;
ALTER TABLE public.tbdetalle_docpago_ctacobrar_impreso ADD CONSTRAINT
tbproducto_fk FOREIGN KEY (id_tbproducto)
REFERENCES public.tbproducto (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: tbdocpago_ctacobrar_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbdetalle_docpago_ctacobrar_impreso DROP CONSTRAINT IF
EXISTS tbdocpago_ctacobrar_fk CASCADE;
ALTER TABLE public.tbdetalle_docpago_ctacobrar_impreso ADD CONSTRAINT
tbdocpago_ctacobrar_fk FOREIGN KEY (id_tbdocpago_ctacobrar)
REFERENCES public.tbdocpago_ctacobrar (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: tbmediopago_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbmediopago_tbdocpago_ctacobrar DROP CONSTRAINT IF
EXISTS tbmediopago_fk CASCADE;
```

```
ALTER TABLE public.tbmediopago_tbdocpago_ctacobrar ADD CONSTRAINT
tbmediopago_fk FOREIGN KEY (id_tbmediopago)
REFERENCES public.tbmediopago (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: tbdocpago_ctacobrar_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbmediopago_tbdocpago_ctacobrar DROP CONSTRAINT IF
EXISTS tbdocpago_ctacobrar_fk CASCADE;
ALTER TABLE public.tbmediopago_tbdocpago_ctacobrar ADD CONSTRAINT
tbdocpago_ctacobrar_fk FOREIGN KEY (id_tbdocpago_ctacobrar)
REFERENCES public.tbdocpago_ctacobrar (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: tbsucursal_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbsucursalpcs DROP CONSTRAINT IF EXISTS tbsucursal_fk
CASCADE;
ALTER TABLE public.tbsucursalpcs ADD CONSTRAINT tbsucursal_fk FOREIGN KEY
(id_tbsucursal)
REFERENCES public.tbsucursal (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: tbtipodoc_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbseriesdoc_emitir DROP CONSTRAINT IF EXISTS tbtipodoc_fk
CASCADE;
ALTER TABLE public.tbseriesdoc_emitir ADD CONSTRAINT tbtipodoc_fk FOREIGN KEY
(id_tbtipodoc)
REFERENCES public.tbtipodoc (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: tbsucursalpcs_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbseriesdoc_emitir DROP CONSTRAINT IF EXISTS
tbsucursalpcs_fk CASCADE;
ALTER TABLE public.tbseriesdoc_emitir ADD CONSTRAINT tbsucursalpcs_fk FOREIGN
KEY (id_tbsucursalpcs)
REFERENCES public.tbsucursalpcs (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: tbrol_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbrol_persona DROP CONSTRAINT IF EXISTS tbrol_fk CASCADE;
ALTER TABLE public.tbrol_persona ADD CONSTRAINT tbrol_fk FOREIGN KEY (id_tbrol)
REFERENCES public.tbrol (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: tbpersona_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbrol_persona DROP CONSTRAINT IF EXISTS tbpersona_fk
CASCADE;
ALTER TABLE public.tbrol_persona ADD CONSTRAINT tbpersona_fk FOREIGN KEY
(id_tbpersona)
```

```
REFERENCES public.tbpersona (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: tbtipodoc_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbctas_cobrar DROP CONSTRAINT IF EXISTS tbtipodoc_fk
CASCADE;
ALTER TABLE public.tbctas_cobrar ADD CONSTRAINT tbtipodoc_fk FOREIGN KEY
(id_tbtipodoc)
REFERENCES public.tbtipodoc (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: tbpersona_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbctas_cobrar DROP CONSTRAINT IF EXISTS tbpersona_fk
CASCADE;
ALTER TABLE public.tbctas_cobrar ADD CONSTRAINT tbpersona_fk FOREIGN KEY
(id_tbpersona)
REFERENCES public.tbpersona (id) MATCH FULL
ON DELETE RESTRICT ON UPDATE CASCADE;
-- ddl-end --

-- object: fk_idpersona_idusuario | type: CONSTRAINT --
-- ALTER TABLE public.tbctas_cobrar DROP CONSTRAINT IF EXISTS
fk_idpersona_idusuario CASCADE;
ALTER TABLE public.tbctas_cobrar ADD CONSTRAINT fk_idpersona_idusuario FOREIGN
KEY (idusuario)
REFERENCES public.tbpersona (id) MATCH FULL
ON DELETE NO ACTION ON UPDATE CASCADE;
-- ddl-end --

-- object: fk_idempresa_idpersona | type: CONSTRAINT --
-- ALTER TABLE public.tbctas_cobrar DROP CONSTRAINT IF EXISTS
fk_idempresa_idpersona CASCADE;
ALTER TABLE public.tbctas_cobrar ADD CONSTRAINT fk_idempresa_idpersona
FOREIGN KEY (idempresa)
REFERENCES public.tbpersona (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: tbctas_cobrar_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbdetpago_ctapagar DROP CONSTRAINT IF EXISTS
tbctas_cobrar_fk CASCADE;
ALTER TABLE public.tbdetpago_ctapagar ADD CONSTRAINT tbctas_cobrar_fk FOREIGN
KEY (idcta_tbctas_cobrar)
REFERENCES public.tbctas_cobrar (idcta) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: tbdocpago_ctacobrar_fk | type: CONSTRAINT --
-- ALTER TABLE public.tbdetpago_ctapagar DROP CONSTRAINT IF EXISTS
tbdocpago_ctacobrar_fk CASCADE;
ALTER TABLE public.tbdetpago_ctapagar ADD CONSTRAINT tbdocpago_ctacobrar_fk
FOREIGN KEY (id_tbdocpago_ctacobrar)
```

```
REFERENCES public.tbdocpago_ctacobrar (id) MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --

-- object: fk_docum_id | type: CONSTRAINT --
-- ALTER TABLE public.tbfacturacionelectronica DROP CONSTRAINT IF EXISTS
fk_docum_id CASCADE;
ALTER TABLE public.tbfacturacionelectronica ADD CONSTRAINT fk_docum_id FOREIGN
KEY (tablaid_doc)
REFERENCES public.tbdocpago_ctacobrar (id) MATCH SIMPLE
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --
```