

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

UNIDAD DE TITULACIÓN

DESARROLLO DE UN SISTEMA WEB DE FACTURACIÓN ELECTRÓNICA

PROYECTO PREVIO A LA OBTENCIÓN DE TÍTULO DE TECNÓLOGO EN ANÁLISIS DE SISTEMAS INFORMÁTICOS

PEDRO ANIBAL MUÑOZ BURGOS

pedro.munoz@epn.edu.ec

Director: PhD. Richard Paul Rivera Guevara

richard.rivera01@epn.edu.ec

Co-director: PhD. Myriam Guadalupe Peñafiel Aguilar

myriam.penafiel@epn.edu.ec

Quito, julio 2021

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por Pedro Anibal Muñoz Burgos como requerimiento parcial a la obtención del título de **TECNÓLOGO EN ANÁLISIS DE SISTEMAS INFORMÁTICOS**, bajo nuestra supervisión:

Richard Paul Rivera Guevara
DIRECTOR DEL PROYECTO

Myriam Guadalupe Peñafiel Aguilar
CODIRECTORA DEL PROYECTO

DECLARACIÓN DE AUTORÍA

Yo, Pedro Anibal Muñoz Burgos con CI: 1718602608 declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación – COESC, soy titular de la obra en mención y otorgo una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entrego toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.

Muñoz Burgos Pedro Anibal

ÍNDICE DE CONTENIDO

1. INTRODUCCIÓN	1
1.1 PLANTEAMIENTO DEL PROBLEMA	1
1.2 OBJETIVO GENERAL	2
1.3 OBJETIVOS ESPECÍFICOS.....	2
1.4 ALCANCE.....	2
2. METODOLOGÍA.....	3
2.1. METODOLOGIA DE DESARROLLO	3
2.2. DISEÑO DE PROTOTIPOS DE LAS INTERFACES (MOCKUPS) Y DE LA BASE DE DATOS	5
2.3. DISEÑO DE LA ARQUITECTURA DEL SISTEMA WEB	7
2.4. HERRAMIENTAS DE DESARROLLO	8
3. RESULTADOS Y DISCUSIÓN	9
3.1. DESARROLLO DE LOS SPRINTS	9
3.2. PRUEBAS DEL SISTEMA WEB	21
4. CONCLUSIONES Y RECOMENDACIONES	22
4.1. CONCLUSIONES	22
4.2. RECOMENDACIONES.....	22
5. REFERENCIAS	24
6. ANEXOS.....	27
6.1. MANUAL TÉCNICO.....	27
6.2. MANUAL DE USUARIO.....	27
6.3. MANUAL DE INSTALACIÓN	27

LISTA DE FIGURAS

Figura 1. Registro de usuario	6
Figura 2. Inicio de sesión	7
Figura 3. Arquitectura usada para el proyecto de facturación web [10]	8
Figura 4. Páginas oficiales de Mongodb, nodejs, Angular, Postman	9
Figura 5. Documento producto.....	10
Figura 6. Documento producto modificado.....	10
Figura 7. Diseño de las pantallas del proyecto.....	11
Figura 8. Mensaje exitoso de conexión a la base de datos	12
Figura 9. Estructura backend	13
Figura 10. Creación de usuarios	13
Figura 11. Ingreso de usuarios	13
Figura 12. Proyecto angular.....	14
Figura 13. Crud clientes, productos detalle-factura.....	15
Figura 14. Método para subir archivos a nuestra aplicación.....	15
Figura 15. Métodos para el documento XML	16
Figura 16. Conexión a los servicios del SRI.....	16
Figura 17. Pruebas de envío y recepción.....	17
Figura 18. Ingreso y registro de usuarios.....	18
Figura 19. Pantallas de Producto	19
Figura 20. Pantallas de Clientes	19
Figura 21. Pantallas de emisor	20
Figura 20. Pantalla de facturación.....	20
Figura 21. Configuración JMeter.....	21
Figura 22. Pruebas de carga con 10 peticiones simultaneas	21
Figura 23. Modificación de pruebas	21
Figura 24. Pruebas exitosas con 500 peticiones en 5 segundos	21

LISTA DE TABLAS

Tabla I Equipo de trabajo del proyecto.....	4
Tabla II Historia de Usuario.....	4

RESUMEN

Este proyecto de titulación trata del desarrollo de un sistema web para realizar facturación electrónica, este sistema será capaz de conectarse a los servicios del Servicio de Rentas Internas, y a su vez emitir la autorización del respectivo documento XML de una factura que se ha generado dentro del sistema, y estará dirigido a las personas naturales y jurídicas que usen la aplicación, cuando lo requieran. El sistema se ha desarrollado con la metodología de software Scrum.

El documento está estructurado por cinco secciones estructuradas de la siguiente forma. En la Sección I, se presenta el planteamiento del problema, el objetivo general, los objetivos específicos, y el alcance. La Sección II, muestra el diseño de las interfaces y de la base de datos, su respectiva arquitectura, herramientas en donde se desarrolló el sistema, y la metodología de desarrollo. La Sección III, muestra el desarrollo de los diferentes sprints y las pruebas sobre el sistema web. La Sección IV, detalla los resultados obtenidos y las posibles soluciones para mejora o mantenimiento del mismo. Finalmente la última sección presenta las conclusiones y recomendaciones obtenidas a lo largo del desarrollo de este proyecto.

PALABRAS CLAVE: SCRUM, Facturación electrónica.

ABSTRACT

This degree project deals with the development of a web system to carry out electronic invoicing, this system will be able to connect to the services of the Internal Revenue Service, and at the same time issue the authorization of the respective XML document of an invoice that has been generated within the system, and will be aimed at natural and legal persons who use the application, when required. The system has been developed with the Scrum software methodology.

The document is structured in five sections structured as follows. In Section I, the problem statement, the general objective, the specific objectives, and the scope are presented. Section II shows the interface and database design, their respective architecture, tools where the system was developed, and the development methodology. Section III shows the development of the different sprints and the tests on the web system. Section IV details the results obtained and the possible solutions for its improvement or maintenance. Finally, the last section presents the conclusions and recommendations obtained throughout the development of this project.

KEYWORDS: SCRUM, Electronic billing.

1. INTRODUCCIÓN

1.1 Planteamiento del problema

Por medio del avance tecnologico, las empresas se han obligado a evolucionar la forma de comercio para generar ganancias, con esto mencionamos lo que son las compras en línea, el intercambio de dinero ya sea por un bien o un servicio, se creó una necesidad de establecer un método seguro para el intercambio de estos, como son firmas electrónicas y certificados digitales, que nos servirán para la identificación de la transacción realizada entre el respectivo comercio y el usuario. La implementación de la factura electrónica se ha visto beneficiada por algunos aspectos como la competitividad, actualización de la tecnología dentro de las empresas, y se recalca que al momento de realizar algunos procesos administrativos dentro de la empresa, estos se reducen y facilitan el cumplimiento de la empresa con el estado. [1]

La principal problemática de la facturación electrónica en el país es debido al sistema gratuito que provee dicho organismo público, este sistema tiene limitaciones de compatibilidad en lo que se refiere a algunos sistemas operativos como Windows 7, MacOS x, Linux, con esto al momento de la instalación se necesitan controladores de los respectivos *tokens* de seguridad, para que el sistema funcione de manera correcta, este sistema al momento de su utilización se ha evidenciado de algunos problemas ya sea que no emite las facturas o genera como claves generadas con lo que el usuario procede a corregir y persiste el error.

La facturación electrónica, en el Ecuador tiene presencia desde el 2012, donde se buscaba aumentar el número de contribuyentes y con esto las obligaciones al estado, esta forma de facturación pretende que al usuario le sea más fácil la deducción del impuesto a la renta. Con esto también será menor el uso del papel al momento de dar las facturas. [2] [3]

Teniendo en cuenta la tesis anterior sobre la facturación electrónica, se propone desarrollar un sistema de facturación electrónica para que los negocios pequeños, medianos o grandes, puedan utilizarlo. Como beneficios, el sistema será capaz de conectarse a los servicios del Servicio de Rentas Internas (SRI.)

1.2 Objetivo general

Desarrollar el sistema de gestión y monitoreo de seguridad y control de acceso de las aulas de la ESFOT.

1.3 Objetivos específicos

- Determinar los requerimientos del sistema web.
- Diseñar la arquitectura de la solución y de la base de datos.
- Desarrollar el sistema de acuerdo con los requerimientos establecidos.
- Realizar pruebas de funcionamiento del sistema.

1.4 Alcance

El sistema web propuesto, consiste en la realización de un sistema de facturación electrónica por el cual podremos generar facturas y este a su vez se conectara al Servicio de Rentas Internas (SRI), donde se autorizara o rechazara el documento, dependiendo de los lineamientos ya establecidos por la entidad.

Esto permitira al usuario que use dicha aplicación, disminuir los procesos administrativos y cumplir con las obligaciones de dicha entidad, ser amigable con el ambiente no usando el papel para facturar, con esto tambien el usuario experimentara mas agilidad a la hora de facturar y un manejo mas amigable de una aplicación web.

La aplicación web tendra la capacidad de realizar facturas y estas a su vez ser aprobadas por el Servicio de Rentas Internas, el usuario podrá registrar su nombre comercial como su firma electrónica, requisito para poder realizar la facturación electrónica, dentro del sistema se especificara la firma que deberá subir al mismo, y una vez completado el perfil del cliente emisor de la aplicación, podremos dirigirnos a la sección de facturación, donde podremos enviar firmado al Servicio de Rentas Internas o solo guardar el documento sin enviar, ni firmar.

2. METODOLOGÍA

2.1. Metodología de desarrollo

La metodología que se adoptó para el proyecto es SCRUM, la metodología SCRUM, agrupa un conjunto de buenas prácticas para el desarrollo colaborativo (en equipo), para así tener un buen resultado del equipo de trabajo. Esta metodología es capaz de adaptarse a cambios que se den en el transcurso del tiempo y mejora de la comunicación entre el equipo de trabajo y el cliente, para un mejor entendimiento del producto final requerido. Las entregas se las realiza mediante SPRINTS, que son limitados periodos de tiempo en realizar una parte de las actividades ya establecidas por el equipo de trabajo.

Roles

Para que un proyecto se construya de manera que sea un software de calidad, depende del equipo con quien se trabaja, para esto la metodología SCRUM, se gestiona un proyecto de manera que define las propiedades que debe tener el software a desarrollarse y superar cualquier dificultad que se pueda presentar en el desarrollo, por lo que el equipo se conforma de la siguiente manera: [4]

- **Scrum Master:** Es la persona encargada del equipo de desarrollo, verifica si se cumplen las reglas y procesos de SCRUM y también trabaja en la reducción de dificultades que se presenten dentro del desarrollo del proyecto. [4]
- **Product Owner (PO):** Es la persona que prioriza las actividades, es decir, toma prioridad de las tareas más importantes o que se necesita que se realice con urgencia. [4]
- **Team (Equipo de desarrollo):** El equipo de desarrollo está integrado por personas (programadores), que tienen el conocimiento necesario para la realización de las respectivas tareas programadas que se comprometen a realizar en cada sprint (periodo de tiempo para completar una tarea de trabajo), con esto el grupo de desarrollo entrega avances del proyecto que se está realizando hasta finalizarlo, el equipo de trabajo se conformara de la manera que se presenta en la **TABLA I**.

Tabla I Equipo de trabajo del proyecto

ROL	PERSONA
SCRUM Master	PhD. Richard Rivera Guevara
Product Owner (PO)	PhD. Richard Rivera Guevara
Team	Pedro Muñoz Burgos

Artefactos o productos esenciales

Estos artefactos o productos esenciales son varios componentes que aseguran el registro de información importante y la transparencia del mismo, estos elementos son el fundamento para la productividad y calidad del proyecto, para los cuales tenemos los siguientes: [5] [6] [7]

- **Recopilación de requerimientos:** La toma de requerimientos es la primera etapa para el comienzo de un desarrollo que se va a comenzar, esto quiere decir que el líder del proyecto se encargara de coordinar reuniones con el cliente ya sea presenciales o por vía telemática; para el proyecto de facturación se mantuvo reuniones vía telemática, por lo tanto, los requerimientos para la elaboración de dicho sistema se han listado y se podrá encontrar en el Manual Técnico.
- **Historias de usuario:** las historias de usuario es información que se obtiene a partir de los clientes mediante reuniones donde el cliente aportara con todos los datos que requiera saber el encargado del proyecto para poder realizar el mismo.

A continuación, se presenta un ejemplo de la historia de usuario, que se usa en la metodología scrum [5] [6] [7].

Tabla II Historia de Usuario

Historia de Usuario	
Identificador: HU1	Usuario: Administrador
Nombre historia: Registro de usuario.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Días estimados: 1	Iteración asignada: 1
Programador responsable: Pedro Muñoz Burgos	

<p>Descripción:</p> <p>El usuario tendrá que registrarse en la aplicación con su nombre, correo electrónico, número de teléfono y su respectivo password que el usuario considere necesario.</p>
<p>Observaciones:</p> <ul style="list-style-type: none"> La ventana de registro contara con validaciones para que al momento de llenar los datos del usuario no tenga ningún campo vacío.

- **Product Backlog:** es una lista de requerimientos que el cliente necesita que se implemente, en el desarrollo del producto, esta lista de requerimientos se realizó cuando el *Product Owner* y el *Cliente*, en las reuniones que tuvieron, armaron el esquema del funcionamiento del producto que se va a realizar.

En las reuniones que se tuvo con el dueño del producto se realizó esta lista de requerimientos, que se encuentra en el Manual de Técnico [5] [6] [7].

- **Sprint Backlog:** es la reunión de la lista de requerimientos realizados en el *Product Backlog* donde son elegidos y presentados con una planeación de tiempo, lo que quiere decir que una tarea o desarrollo determinado puede ser realizado en un periodo de tiempo y mientras se vayan cumpliendo dichas tareas, van surgiendo las siguientes hasta terminar el producto.

En el Manual Técnico se especifica los *sprints* planificado con el respectivo tiempo que se ha de demorar cada tarea, para poder así, tener un mejor control sobre cualquier inconveniente que se presente mientras se esté realizando el desarrollo [5] [6] [7].

2.2. Diseño de prototipos de las interfaces (Mockups) y de la base de datos

- **NINJAMOCK**

Ninjamock es una pequeña *startup* con sede en Dinamarca, una ciudad famosa por su fuerte compromiso en la calidad y diseño. Son un grupo de personas que se han permitido transformar muchas ideas que se ha tenido en una acción sólida. Este programa se ha usado en el diseño de las interfaces ya que es muy amigable para realizar mockups o diseños ya sea para web o móvil, para nuestro caso se lo uso para el diseño de la aplicación web de facturación. [8]

A continuación, se presentan ejemplos del diseño de las interfaces realizadas en ninjamock, los diseños restantes se encuentran en el Manual Técnico.

The image shows a wireframe of a web browser window. The browser's address bar is empty. The main content area contains a registration form titled "CREAR CUENTA!". To the left of the form is a large square placeholder with an 'X' inside. The form itself has the following fields and elements:

- Form Title: CREAR CUENTA!
- Input Fields:
 - Names (Nombre)
 - Telephone (Teléfono)
 - Electronic Mail (Correo Electrónico)
 - Password (Contraseña)
 - Confirm Password (Confirmar Contraseña)
- Registration Button: Registrar Cuenta
- Link: [¿Ya tienes una cuenta?, Entrar](#)

Figura 1. Registro de usuario

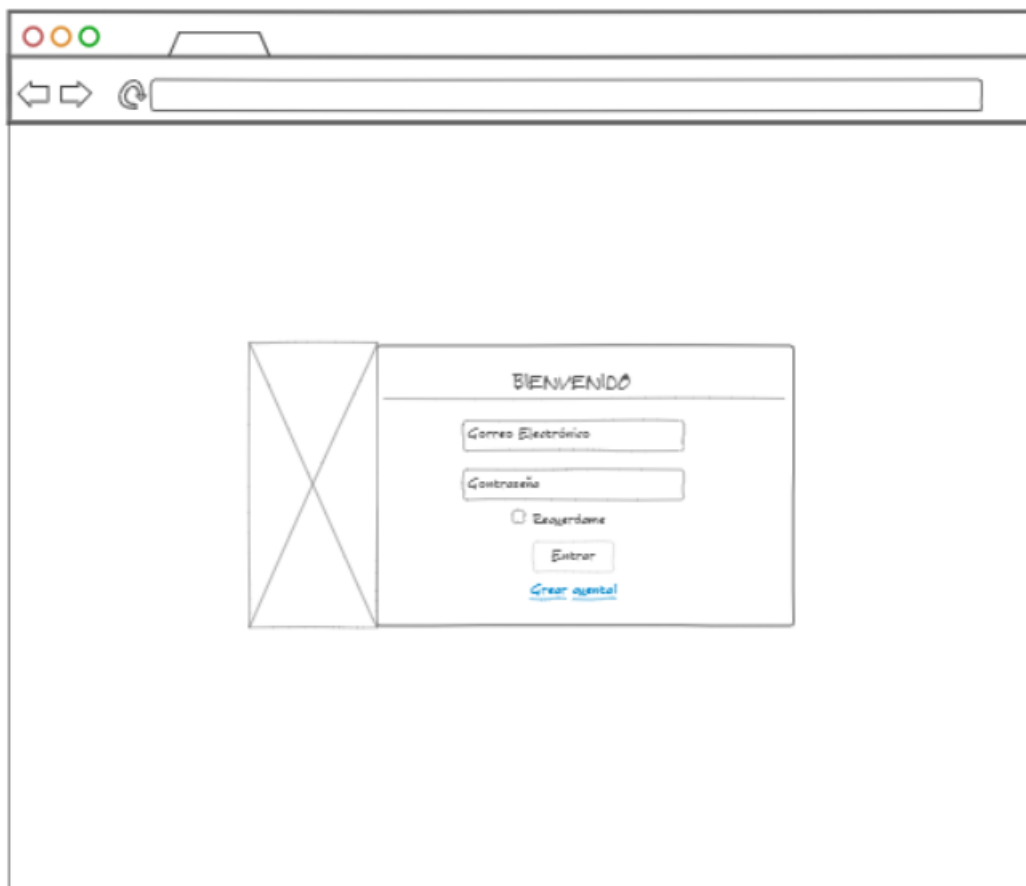


Figura 2. Inicio de sesión

2.3. Diseño de la arquitectura del sistema web

La arquitectura empleada para este proyecto es MVC (Modelo Vista Controlador), la arquitectura MVC, esta arquitectura hace que el desarrollo de una aplicación web o móvil sea más eficiente, ya que se puede modificar cualquier elemento del modelo, vista o controlador de la aplicación sin cambiar o modificar ninguno otro, otra ventaja de esta arquitectura es que hace que las aplicaciones sean escalables de tal forma que se pueda añadir o quitar elementos, funcionalidades a dicho sistema web o aplicación móvil.

Para el proyecto de facturación la arquitectura que se va a usar, se muestra en la Figura 3, con esto se muestra la independencia de componentes, al momento de la realización del proyecto, lo cual si se realiza un cambio ya sea en las interfaces este no afectara al resto de componentes o funcionamiento de la aplicación web [9].

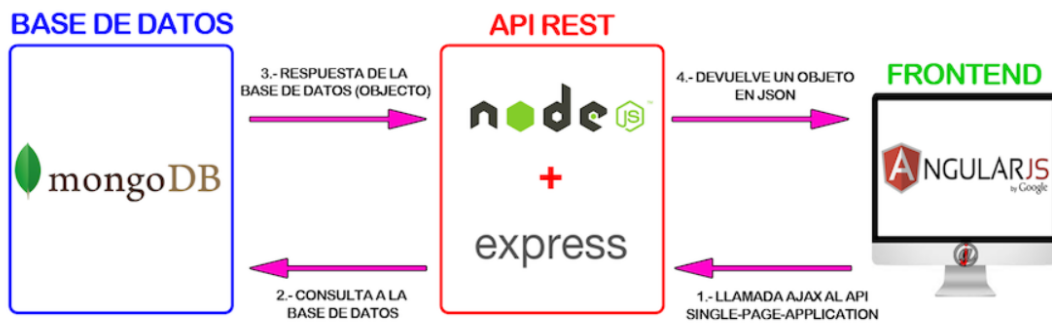


Figura 3. Arquitectura usada para el proyecto de facturación web [10]

2.4. Herramientas de desarrollo

MONGODB

Mongodb es una base de datos NoSQL basada en documentos, esto quiere decir que los datos los guarda en formato binario BSON, que es similar al formato JSON, con esto hace que las consultas sean flexibles y escalables. También puede realizar combinaciones en las consultas, se puede realizar relaciones entre documentos dependiendo del diseño de base de datos que se desea implementar, también ofrece escalabilidad y flexibilidad de los datos que quiere decir que mientras crezca la aplicación esta base de datos se adapta y no va a existir problemas al momento de realizar una consulta. [11] [12] [13]

NODEJS + EXPRESS: NODEJS es un entorno de ejecución de JavaScript, que permite la comunicación entre la base de datos y las interfaces de usuario, es escalable y tiene alto rendimiento para poder conectarse de una manera más eficaz y soporta varias conexiones al mismo tiempo, sin tener ningún problema de latencia. [14] [10]

ANGULAR: es una plataforma de desarrollo construida con *Typescript (lenguaje de tipado)*, es un framework basado en JavaScript, que fue creado por google, este framework posee herramientas para permitir al desarrollador trabajar de una manera más óptima en lo que es las vistas o frontend de la aplicación que se está desarrollando, también la ventaja que posee es que se puede realizar aplicaciones de una sola página, y también aplicaciones empresariales más robustas. [15]

BOOTSTRAP: Es un framework donde se usa JavaScript y CSS creado por dos desarrolladores de twitter donde al principio se llamaba Twitter Blueprint, pero desde el 2011 paso a ser de código abierto donde cambio su nombre a Bootstrap, Bootstrap proporciona que el manejo de una página web sea de fácil navegación para el usuario y también es *mobile first (que quiere decir que también se adecua a los teléfonos móviles)*. [16].

3. RESULTADOS Y DISCUSIÓN

3.1. Desarrollo de los sprints

Para el desarrollo de los sprints se debe verificar el *product backlog*, que se encuentra en el Manual Técnico, donde se detalla el tiempo y las respectivas funcionalidades que debe estar implementada al final de cada *sprint*.

Sprint 0. Comienzo del desarrollo del proyecto

Para este *sprint*, se ha comenzado con la configuración del ambiente de desarrollo del proyecto web, con sus respectivas instalaciones de las herramientas a usar, las actividades a realizar para este proyecto se ha definido las siguientes actividades:

- Instalación de *Mongodb*, *Nodejs*, *Angular*, *Postman* en ambiente local
- Diseño de la base de datos.
- Diseño de interfaces de la aplicación.

Instalación de Mongodb, Nodejs, Angular, Postman

Para el desarrollo del proyecto debemos instalar las herramientas mencionadas anteriormente, para esto nos iremos a las páginas oficiales de cada una de las herramientas como se muestra en la Figura 4.

MongoDB Community Server

MongoDB offers both an Enterprise and Community version of its powerful distributed document database. The community version offers the flexible document model along with ad hoc queries, indexing, and real time aggregation to provide powerful ways to access and analyze your data. As a distributed system you get high availability through built-in replication and failover along with horizontal scalability with native sharding.

The MongoDB Enterprise Server gives you all of this and more. Review the Enterprise Server tab to learn what else is available.

Available Downloads

Version
4.4.6 (current)




Platform
Windows

Package
msi

Download
Copy Link

Versión actual: 14.17.3 (includes npm 6.14.13)


Descargue el código fuente de Node.js o un instalador pre-compilado para su plataforma, y comience a desarrollar hoy.

LTS Recomendado para la mayoría	Actual Últimas características	
 Instalador Windows <small>node-v14.17.3-x64.msi</small>	 Instalador macOS <small>node-v14.17.3.pkg</small>	 Código Fuente <small>node-v14.17.3.tar.gz</small>

Instalador Windows (.msi)	32-bit	64-bit
Binario Windows (.zip)	32-bit	64-bit
Instalador macOS (.pkg)	64-bit	
Binario macOS (.tar.gz)	64-bit	
Binario Linux (x64)	64-bit	
Binario Linux (ARM)	ARMv7	ARMv8
Código Fuente	node-v14.17.3.tar.gz	

The Postman app

The ever-improving Postman app (a new release every two weeks) gives you a full-featured Postman experience.


Download the App

By downloading and using Postman, I agree to the [Privacy Policy](#) and [Terms](#).

Version 8.7.0 | [Release Notes](#) | [Product Roadmap](#)

Not your OS? Download for Mac ([macOS](#)) or Linux ([x64](#))

Figura 4. Páginas oficiales de Mongodb, nodejs, Angular, Postman

Para el caso de angular debemos instalar Angular CLI, de la siguiente forma:

```
npm install -g @angular/cli
```

Diseño de la base de datos

La base de datos como es por documentos se comenzó a identificar las respectivas tablas que se va a realizar en el proyecto, para esto primero se realizó un análisis de otros sistemas de facturación que se pueda estudiar las posibles tablas que entren en el proyecto, por lo que se estableció primero en un documento JSON, en la Figura 5 y en la Figura 6 observamos una relación entre la tabla usuarios y la tabla producto

```
{
  "usuario": {
    "type": "Schema.Types.ObjectId",
    "ref": "Usuario"
  },
  "codigo_principal": "",
  "tipo_producto": "",
  "nombre": "",
  "valor_unitario": "",
  "iva": "",
  "ice": "",
  "irbpnr": "",
  "info_adicional": ""
}
```

Figura 5. Documento producto

```
const ProductoSchema = Schema({
  codigo_principal: {
    type: String,
    required: true
  },
  codigo_auxiliar: {
    type: String,
  },
  tipo_producto: {
    type: String,
  },
  descripcion: {
    type: String,
    required: true
  },
  valor_unitario: {
    type: String,
    required: true
  },
  // Impuestos
  iva: {
    type: Number,
  },
  ice: {
    type: Number,
  },
  irbpnr: {
    type: Number,
  },
  // Info adicional - ANALIZAR
  info_adicional: {
    type: String,
  },
  //
  usuario: {
    type: Schema.Types.ObjectId,
    ref: "Usuario",
    required: true
  }
});
```

Figura 6. Documento producto modificado

Diseño de interfaces de la aplicación

Para esto se usó la herramienta *ninjamock*, donde se diseñaron todas las interfaces que el usuario tiene al momento de ingresar a la aplicación por lo tanto en la Figura 7, se observara los diseños de cada una de las pantallas de este proyecto.

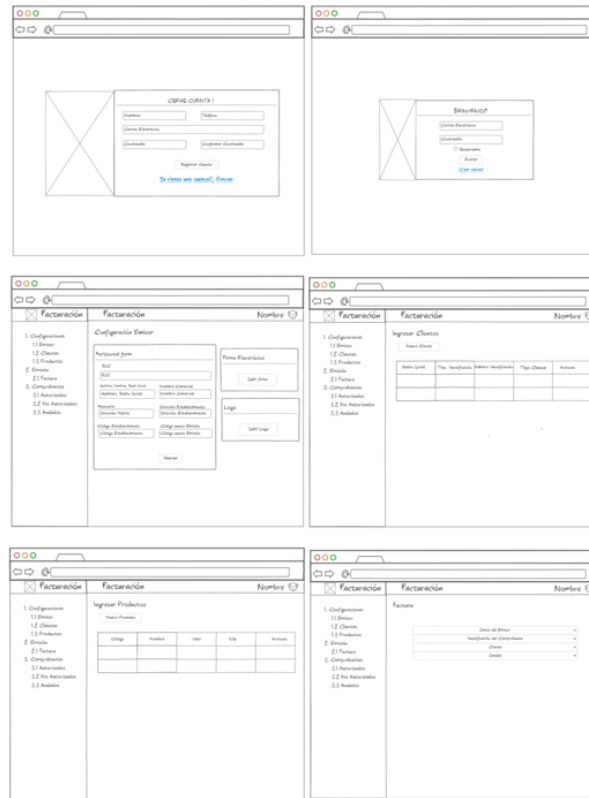


Figura 7. Diseño de las pantallas del proyecto

Sprint 1.

En este *sprint* se comenzó a construir el *backend*, comenzando con la configuración de *express* con *nodejs*, con esto también se definió la estructura del proyecto, la conexión a la base de datos, con esto las tareas son las siguientes:

- Conexión *backend* con la base de datos *mongodb*
- Definición de estructura del proyecto
- Creación de servicio de inicio de sesión y registro de usuario

Conexión *backend* con la base de datos *MongoDB*

Para esto usaremos la terminal, para visualizar el mensaje de que esta levantada la base de datos, el mensaje se muestra en la Figura 8:

```
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Server corriendo en el puerto: 3001
DB Online
```

Figura 8. Mensaje exitoso de conexión a la base de datos

Definición de estructura del proyecto

La estructura del proyecto backend para más detalle se muestra en la Figura 9, por lo que se explicara cada una de ellas:

Controllers: aquí van a estar los métodos CRUD (Create, Read, Update, Delete), para los diferentes modelos de la base de datos.

Db: en esta carpeta se tendrá la conexión a la base de datos, donde se pondrán mensajes de error en un caso no se pueda conectar

Helpers: En esta carpeta estarán los métodos para poder generar un token (identificador), para cuando el usuario se registre, tenga ese token una duración determinada

Middlewares: aquí definiremos las validaciones cuando los campos estén vacíos o estén incorrectos o repetidos, y también una validación de token (identificador)

Models: Aquí se definirá los campos que van dentro de cada documento o tabla

Routes: aquí se tendrá definidas las rutas para poder acceder a los documentos creados

Uploads: en este directorio contendrán las firmas electrónicas y logos que subirán los usuarios que deseen facturar.

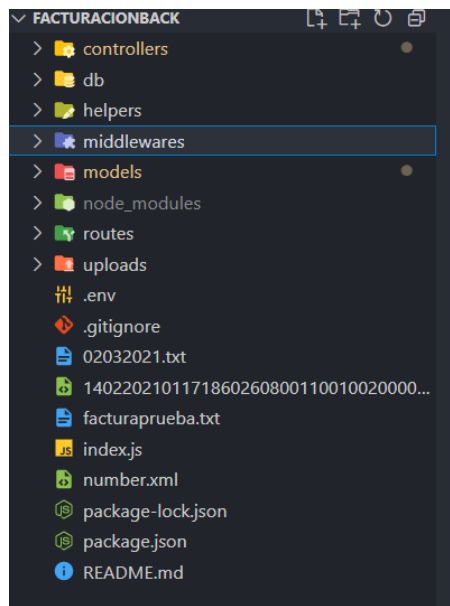


Figura 9. Estructura backend

Creación de servicio de inicio de sesión y registro de usuario

En el backend que es donde vamos a poner la lógica de negocio, vamos a crear el ingreso y registro de usuario con las respectivas validaciones que conlleva el backend, en la Figura 10 podemos visualizar que la creación de usuarios, es el método crearUsuarios, para el servicio de login se puede visualizar la Figura 11.

```
const getUsuarios = async(req, res = response) => { ...
}

const crearUsuarios = async(req, res = response) => { ...
}

const actualizarUsuarios = async(req, res = response) => { ...
}
```

Figura 10. Creación de usuarios

```
> const login = async(req, res = Response) => { ...
}
```

Figura 11. Ingreso de usuarios

Sprint 2.

Para este sprint se ha continuado con la construcción del *backend* de la aplicación, construyendo los demás servicios, y validaciones, por lo que las actividades definidas para este sprint son las siguientes:

- Creación del proyecto frontend en angular
- Servicio de clientes, productos, detalle-factura, factura (Create, Read, Update, Delete)
- Servicio para subir archivos a la aplicación

Creación del proyecto frontend en angular

Para la creación del proyecto en angular se utilizó el siguiente comando:

ng new facturacionfront

Con esto se ha creado la carpeta que se muestra en la Figura 12, y con esto la estructura también del proyecto en angular, que es parte de las vistas que se mostrará al usuario.

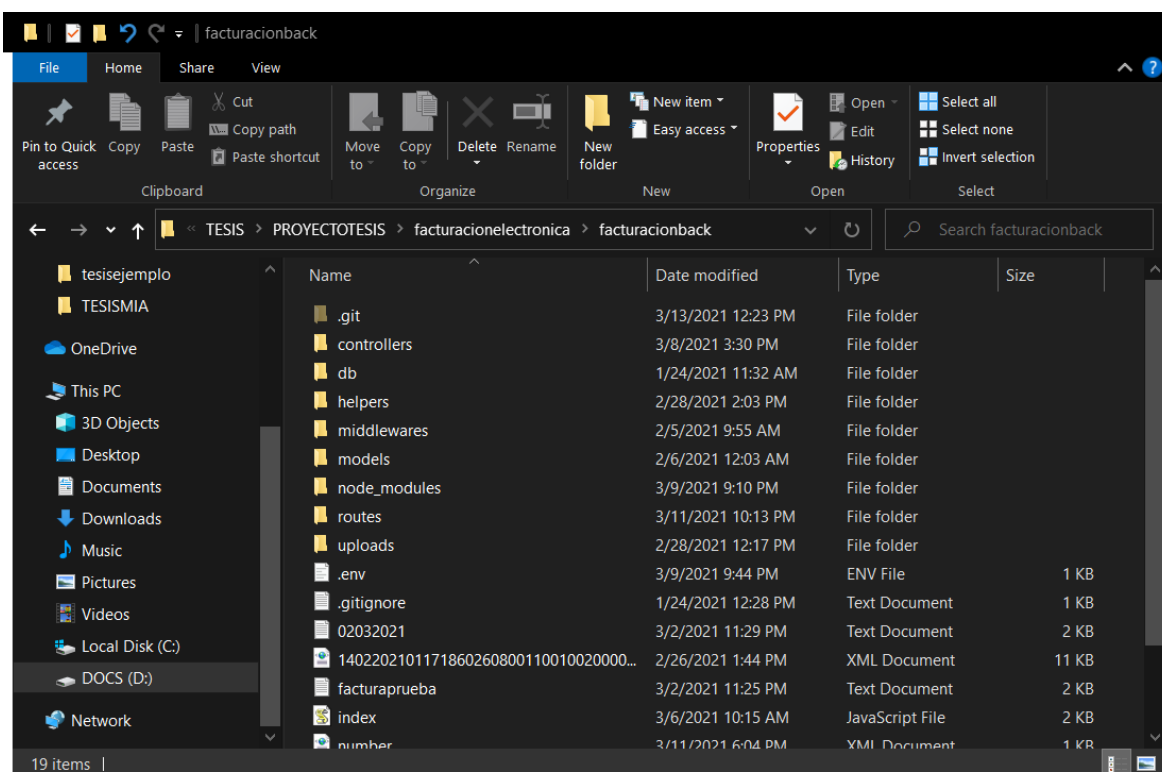


Figura 12. Proyecto angular

Servicio de clientes, productos, detalle-factura (Create, Read, Update, Delete)

Para la creación de estos servicios se visualizó el modelo que se ha creado por lo tanto, los métodos CRUD, se visualizan en la Figura 13.

```
const getCientes = async(req, res = response) => { ...  
}  
  
const crearCliente = async(req, res = response) => { ...  
}  
  
const actualizarCliente = async(req, res = response) => { ...  
}  
  
const borrarCliente = async(req, res = response) => { ...  
}  
  
const getProducto = async(req, res = response) => { ...  
}  
  
const crearProducto = async(req, res = response) => { ...  
}  
  
const actualizarProducto = async(req, res = response) => { ...  
}  
  
const borrarProducto = async(req, res = response) => { ...  
}  
  
const getDetalle = async(req, res = response) => { ...  
}  
  
const crearDetalle = async(req, res = response) => { ...  
}  
  
const actualizarDetalle = async(req, res = response) => { ...  
}  
  
const borrarDetalle = async(req, res = response) => { ...  
}
```

Figura 13. Crud clientes, produtos detalle-factura

Servicio para subir archivos a la aplicación

Para este servicio se utilizó la librería express fileupload, por lo que esto facilita el proceso de subir archivos ya sean firmas electrónicas, logos, documentos en Word, PDF o Excel. En la Figura 14 se visualiza los métodos involucrados al momento de realizar esta funcionalidad.

```
const fileUpload = (req, res = response) => { ...  
}  
  
const retornaFirma = (req, res) => { ...  
}
```

Figura 14. Método para subir archivos a nuestra aplicación

Sprint 3.

En este sprint se siguió con el backend pero exactamente centrado en dos servicios que son la creación del documento XML y conectarse a los servicios del SRI, que presentaron un nivel de complejidad más alto. Las actividades se definen de la siguiente manera:

- Creación del documento XML
- Conexión a los servicios del SRI
- Pruebas de envío y recepción de estructuras XML

Creación del documento XML

Para la creación del documento XML, se tuvo varios retos, por ejemplo, la documentación sobre la creación del mismo no se encontraba para JavaScript, por otro lado, se usó varias librerías de formación de documentos por lo que al final, se decidió hacerla de la forma que se muestra en la Figura 15.

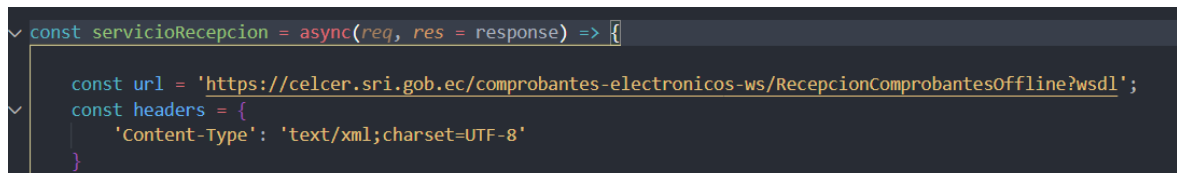


```
> const getXML = async(req, res = response) => { ...  
  }  
  
> const crearXML = async(req, res = response) => { ...  
  }
```

Figura 15. Métodos para el documento XML

Conexión a los servicios del SRI

En la Figura 16 se muestra los servicios que se crearon para poder conectarnos al servicio soap del Servicio de Rentas Internas, con esto se tuvo que armar query para poder aprobar o rechazar facturas.



```
✓ const servicioRecepcion = async(req, res = response) => {  
  const url = 'https://celcer.sri.gob.ec/comprobantes-electronicos-ws/RecepcionComprobantesOffline?wsdl';  
  ✓ const headers = {  
    'Content-Type': 'text/xml; charset=UTF-8'  
  }  
}
```

Figura 16. Conexión a los servicios del SRI

Pruebas de envío y recepción

En las pruebas de envío y recepción de facturas se las ha realizado como se muestra en la Figura 17, para que esto funcionara correctamente se hicieron varias pruebas de funcionamiento.

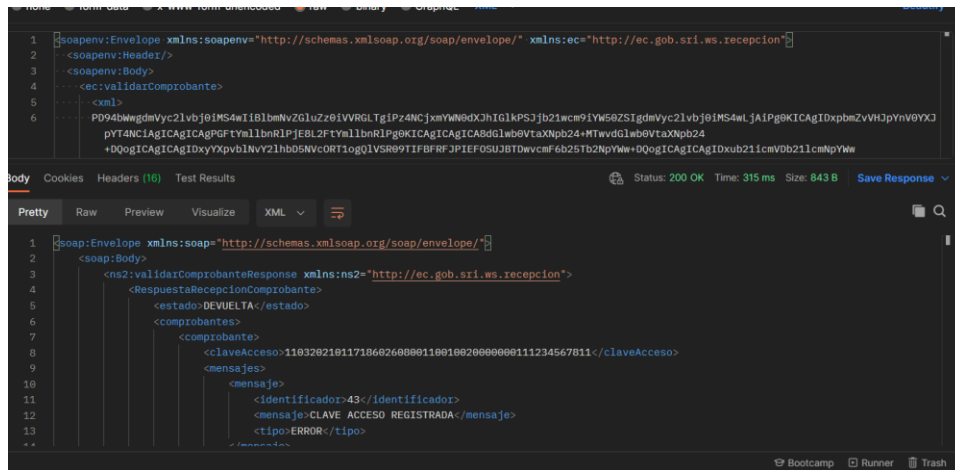


Figura 17. Pruebas de envío y recepción

Sprint 4.

Este sprint se ha basado netamente en crear la parte de las vistas de usuario, a continuación se definen las actividades:

- Creación de las pantallas de login y registro de usuario
- Creación de pantallas de emisor, productos, clientes
- Creación de pantalla de factura

Creación de pantallas de login y registro de usuario

Cuando creamos las pantallas de ingreso y de registro se ha basado en los mockups que se realizaron anteriormente, por lo tanto, en la Figura 18 se muestra las pantallas de login y registro.

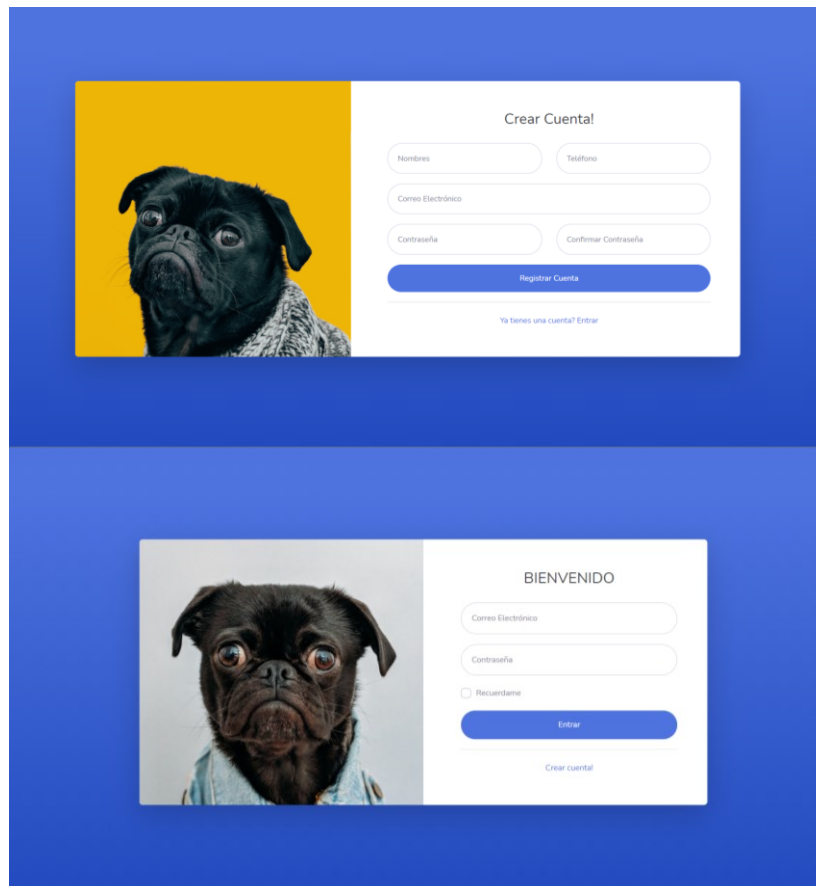


Figura 18. Ingreso y registro de usuarios

Creación de pantallas de emisor, productos, clientes

Las pantallas que siguen en el diseño son las del emisor que es para configurar la firma electrónica, productos y clientes, dependiendo de la actividad económica que se realice, en las Figuras 19, Figura 20 y Figura 21 se visualiza dichas pantallas.

Nuevo Producto

×

Catálogo INEC

Catálogo INEC

Nombre

Nombre

Valor

Valor

Información adicional

Información adicional

IVA

0%

▼

ICE

Seleccione

▼

Cerrar

Guardar

Figura 19. Pantallas de Producto

Nuevo Cliente

×

Apellidos, Nombres/ Razón Social

Apellidos Nombres / Razón Social

Tipo Identificación

CÉDULA

▼

Ingrese su Identificación

Tipo Cliente

CLIENTE

▼

Dirección

Dirección

Teléfono

Teléfono

Correo Electrónico

Correo Electrónico

Cerrar

Guardar

Figura 20. Pantallas de Clientes

Configuración Emisor

Emisor

RUC

Apellidos, Nombres, Razón Social

Nombre Comercial

Dirección Matriz

Dirección Establecimiento

Código Establecimiento

Código punto Emisión

☐ Obligado a llevar contabilidad
 ☐ Contribuyente Régimen Microempresas

[Guardar](#)

Firma Electrónica

Tamaño máximo 50 KB y Archivo con extensión p12

[Subir Firma Electrónica](#)

Logo

Tamaño máximo 100 KB, extensión .jpg o .png y ser menor a 200px por 200px.

[Subir logo](#)

Figura 21. Pantallas de emisor

Creación de pantalla factura

La pantalla de facturación se ha realizado como se muestra en la Figura 22, esto definiendo cada parte de la factura y traer todos los componentes para realizar la facturación.

Datos del Emisor

RUC

Nombre Comercial

Razón Social / Apellidos y nombres

Nombre Comercial

Contribuyente Especial

Dirección Matriz

Dirección Establecimiento

Identificación del Comprobante

Cliente

Figura 22. Pantalla de facturación

3.2. Pruebas del Sistema Web

Pruebas de carga

Para estas pruebas se ha usado JMeter que es un programa para realizar pruebas de carga a los servicios que fueron creados para este proyecto de facturación, estas pruebas se las realiza para observar que es lo que pasa cuando se sobrecarga o si hay varias conexiones al mismo tiempo, con esto se ve como trabaja la aplicación en este tipo de condiciones.

Para esto se ha configurado como se muestra en la Figura 23.

The screenshot shows the JMeter configuration interface with the 'Basic' tab selected. The 'Web Server' section is configured with Protocol [http], Server Name or IP: localhost, and Port Number: 3001. The 'HTTP Request' section is configured with Method: POST and Path: /api/usuarios. The 'Content encoding' field is empty. At the bottom, there are checkboxes for 'Redirect Automatically' (unchecked), 'Follow Redirects' (checked), 'Use KeepAlive' (checked), 'Use multipart/form-data' (unchecked), and 'Browser-compatible headers' (unchecked).

Figura 23. Configuración JMeter

Una vez configurado vamos a observar en la Figura 24. Al momento de realizar pruebas de usuario, por lo que las pruebas fueron exitosas.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	10	80	76	85	2.55	0.00%	2.2/sec	1.66	0.62	781.0
TOTAL	10	80	76	85	2.55	0.00%	2.2/sec	1.66	0.62	781.0

Figura 24. Pruebas de carga con 10 peticiones simultaneas

Estas pruebas de estrés tambien se puede realizar con mas peticiones simultaneas, con esto se va a modificar el servicio y el tiempo que se puede demorar cada uno, como se muestra en las Figura 25 y Figura 26

The screenshot shows the 'Thread Properties' dialog box in JMeter. The 'Number of Threads (users):' field is set to 500. The 'Ramp-up period (seconds):' field is set to 5. The 'Loop Count:' section has the 'Infinite' checkbox unchecked and the value '1' entered in the adjacent field.

Figura 25. Modificación de pruebas

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	510	2669	76	5240	1424.86	0.00%	2.0/min	0.03	0.01	781.0
TOTAL	510	2669	76	5240	1424.86	0.00%	2.0/min	0.03	0.01	781.0

Figura 26. Pruebas exitosas con 500 peticiones en 5 segundos

4. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- Mediante las investigaciones realizadas para este proyecto se identificó algunos problemas al momento de realizar la facturación por medio de la aplicación de escritorio que tiene el Servicio de Rentas Internas. El sistema al momento de enviar una factura se demoraba en autorizarse, la configuración se volvió un poco tediosa, por lo que el sistema trata de mejorar la interacción con el usuario y la interacción del sistema con los servicios web de la institución antes mencionada.
- El uso de las nuevas tecnologías, hace que cada empresa se modernice conforme pase el tiempo, ya que la renovación de tecnologías o migración de sistemas es necesario ya que con la modernización se puede evitar ataques informáticos en la empresa.
- La base de datos NoSQL basado en documentos *Mongodb*, es una base de datos no relacional, que para nuestro proyecto se analizó y se adaptaba muy bien al modelo de base de datos propuesto, cabe mencionar que *Mongodb* por el hecho de ser una base no relacional ofrece consultas tipo SQL y agregaciones que en este caso son JOINS (uniones de tablas) en SQL, con el lenguaje JavaScript se puede realizar las diferentes consultas.
- Dentro de las pruebas realizadas al sistema, se demostró el funcionamiento al momento de subir la firma electrónica, el registro de usuario el ingreso de usuario, y al momento de realizar la facturación con el consumo de los web services de la institución que se ha venido mencionando.

4.2. Recomendaciones

- Se recomienda siempre revisar el código para optimizarlo de mejor manera, ya que si se tiene un código bien estructurado, los programadores que sigan con el proyecto podrán entender lo que se realizó con el proyecto.
- Cuando el sistema se encuentre en producción y pruebas, es recomendable que el enlace de acceso sea solo para quienes vayan a probar ya que las seguridades aún

no se han implementado como ingresar una contraseña robusta, por lo que se plantea que en un futuro se realice este cambio a la aplicación web.

- Si la plataforma web crece se debe realizar un cambio en la base de datos, ya que tiene un número determinado de peticiones, y una vez pasado ese número de peticiones, los servidores de base de datos tienen un costo anual o mensual, esto será analizado por el desarrollador y también por el cliente donde estará funcionando la aplicación web.
- Es necesario que según la escalabilidad del programa también se analice tener un equipo de soporte técnico para cualquier inconveniente dentro del programa.

5. REFERENCIAS

- [1] D. V. D. CAROLINA y V. I. YUQUI BARRETO, «<http://repositorio.ug.edu.ec/>,» Septiembre 2019. [En línea]. Available: <http://repositorio.ug.edu.ec/bitstream/redug/46338/1/Tesis%20Facturacion%20electronica.pdf>. [Último acceso: 18 08 2020].
- [2] V. Heredia, «La facturación electrónica en Ecuador va en aumento,» *EL COMERCIO*, 02 Abril 2017.
- [3] P. Alvarado, «Conozca cómo acceder a servicios de facturación electrónica y de cobro con tarjetas para atender con más seguridad en la emergencia,» *EL COMERCIO*, 29 Abril 2020.
- [4] SOFTENG, «<https://www.softeng.es/>,» [En línea]. Available: <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html>. [Último acceso: 17 06 2021].
- [5] J. Roche, «<https://www2.deloitte.com/>,» [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/artefactos-scrum.html>. [Último acceso: 18 06 2021].
- [6] I. IT, «<https://integrait.com.mx/>,» [En línea]. Available: <https://integrait.com.mx/blog/roles-de-scrum/>. [Último acceso: 18 06 2021].
- [7] M. Bara, «<https://www.obsbusiness.school/>,» [En línea]. Available: <https://www.obsbusiness.school/blog/roles-eventos-y-artefactos-en-la-metodologia-scrum>. [Último acceso: 18 06 2021].
- [8] ninjamock, «<https://ninjamock.com/>,» [En línea]. Available: <https://ninjamock.com/about-us>. [Último acceso: 18 06 2021].
- [9] U. d. Alicante, «<https://si.ua.es/es/>,» [En línea]. Available: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>. [Último acceso: 21 06 2021].
- [10] R. Moya, «<https://jarroba.com/author/ricardomoya/>,» [En línea]. Available: <https://jarroba.com/mean-mongo-express-angular-node-ejemplo-de-aplicacion-web-parte-ii/>. [Último acceso: 19 06 2021].
- [11] mongodb, «<https://docs.mongodb.com/manual/>,» [En línea]. Available: <https://docs.mongodb.com/manual/>. [Último acceso: 20 06 2021].
- [12] mongodb, «<https://www.mongodb.com/>,» [En línea]. Available: <https://www.mongodb.com/>. [Último acceso: 20 06 2021].

- [13] mongodb, «[https://www.mongodb.com/es](https://www.mongodb.com/es,» [En línea]. Available: <a href=). [Último acceso: 20 06 2021].
- [14] <https://nodejs.org>, «[https://nodejs.org/es/about/](https://nodejs.org,» [En línea]. Available: <a href=). [Último acceso: 15 09 2020].
- [15] angular, «[https://angular.io/guide/what-is-angular](https://angular.io,» [En línea]. Available: <a href=). [Último acceso: 20 06 2021].
- [16] bootstrap, «[https://getbootstrap.com/](https://getbootstrap.com/,» [En línea]. Available: <a href=). [Último acceso: 22 06 2021].
- [17] P. Valdés, «[https://www.inboundcycle.com/inbound-marketing-que-es](https://www.inboundcycle.com,» [En línea]. Available: <a href=). [Último acceso: 06 10 2019].
- [18] E. UNIVERSO, «Facturas ahora se envían por e-mail,» *EL UNIVERSO*, 13 Abril 2011.
- [19] I. H. R. Tückler, «[https://www.laprensa.com.ni/2014/05/17/economia/194648-tecnologia-uw](https://www.laprensa.com.ni/2014/05/17/economia/194648-tecnologia-uw,» [En línea]. Available: <a href=). [Último acceso: 31 10 2019].
- [20] J. J. Y. Torres, «[http://bibing.us.es/proyectos/abreproy/11138/fichero/memoria%252FCap%C3%ADtulo+3.pdf+](http://bibing.us.es,» [En línea]. Available: <a href=). [Último acceso: 02 03 2020].
- [21] C. Samsing, «[https://blog.hubspot.es/marketing/que-es-inbound-marketing-slide-share](https://blog.hubspot.es,» [En línea]. Available: <a href=). [Último acceso: 06 10 2019].
- [22] M. A. G. Sagñay, «[https://es.slideshare.net/magavilanes/protocolo-80211a](https://es.slideshare.net/magavilanes/protocolo-80211a,» [En línea]. Available: <a href=). [Último acceso: 02 03 2020].
- [23] A. Robledano, «[https://openwebinars.net/blog/que-es-mongodb/](https://openwebinars.net,» [En línea]. Available: <a href=). [Último acceso: 20 06 2021].
- [24] proyectosagiles, «[https://proyectosagiles.org/que-es-scrum/](https://proyectosagiles.org,» [En línea]. Available: <a href=). [Último acceso: 23 08 2020].
- [25] pcihispano, «[https://www.pcihispano.com/como-afecta-la-nueva-version-del-top-ten-de-owasp-el-cumplimiento-de-pci-dss-v3-2/](https://www.pcihispano.com,» [En línea]. Available: <a href=).
- [26] nodejs, «[https://nodejs.org/es/about/](https://nodejs.org,» [En línea]. Available: <a href=). [Último acceso: 20 06 2021].

- [27] R. LÍDERES, «<https://www.revistalideres.ec/lideres/>,» [En línea]. Available: <https://www.revistalideres.ec/lideres/negocio-tecnologia-pymes-bancaprivada-xchange.html>. [Último acceso: 10 11 2019].
- [28] J. F. M. Lendech, «<https://core.ac.uk/>,» [En línea]. Available: <https://core.ac.uk/download/pdf/154797518.pdf>. [Último acceso: 30 10 2019].
- [29] S. d. R. Internas, «Manual de Usuario offline Facturador electrónico gratuito,» Quito.
- [30] S. d. R. Internas, «Facturador electrónico gratuito,» *Manual de usuario offline*, p. 108.
- [31] <https://www.kushkipagos.com/>, «<https://www.kushkipagos.com/>,» [En línea]. Available: <https://www.kushkipagos.com/>. [Último acceso: 10 11 2019].
- [32] <https://si.ua.es>, «<https://si.ua.es>,» [En línea]. Available: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>. [Último acceso: 19 06 2021].
- [33] <https://introbay.com>, «<https://introbay.com/es/blog/2016/07/05/>,» [En línea]. Available: <https://introbay.com/es/blog/2016/07/05/https-que-es-y-para-que-sirve>. [Último acceso: 10 11 2019].
- [34] J. Francia, «<https://www.scrum.org/>,» [En línea]. Available: <https://www.scrum.org/resources/blog/que-es-scrum>. [Último acceso: 23 08 2020].
- [35] C. DRUMOND, «<https://www.atlassian.com/es>,» [En línea]. Available: <https://www.atlassian.com/es/agile/scrum>. [Último acceso: 23 08 2020].
- [36] P. Alvarado, *EL COMERCIO*, 29 Abril 2020.

6. ANEXOS

Los anexos del proyecto se encuentran en repositorio de GitHub del siguiente enlace:

<https://github.com/EsfotEPN/Proyecto-Titulacion-Pedro-Munoz>

6.1. Manual Técnico

- Recopilación de Requerimientos
- Historias de Usuario
- Product Backlog
- Sprint Backlog
- Diseño de interfaces
- Pruebas

6.2. Manual de Usuario

- Manual en video

6.3. Manual de Instalación

- Despliegue del Sistema Web
- Despliegue de la Aplicación Móvil
- Credenciales de acceso
- Repositorio del código fuente