

**DESARROLLO DEL MÓDULO DE CONTABILIDAD EN LENGUAJE  
.NET CORE CUMPLIENDO CON LOS PRINCIPIOS DE  
DESARROLLO SOLID Y DE ARQUITECTURA LIMPIA USANDO  
LOS FRAMEWORK ENTITY, CODE FIRST, DEVEXPRESS Y  
BOOTSTRAP 4 EN UN REPOSITORIO SQL SERVER CON  
CONTROL DE VERSIONES GIT BAJO EL MARCO DE TRABAJO  
SCRUM EN MICROSHIF**

**JAMES ALFRED CUADRADO LÁZARO**



**PROGRAMA DE INGENIERÍA DE SISTEMAS  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA,  
ELECTRÓNICA, SISTEMAS Y TELECOMUNICACIONES  
FACULTAD DE INGENIERIAS Y ARQUITECTURA  
UNIVERSIDAD DE PAMPLONA  
2019**

**DESARROLLO DEL MÓDULO DE CONTABILIDAD EN LENGUAJE  
.NET CORE CUMPLIENDO CON LOS PRINCIPIOS DE DESARROLLO  
SOLID Y DE ARQUITECTURA LIMPIA USANDO LOS FRAMEWORK  
ENTITY, CODE FIRST, DEVEXPRESS Y BOOTSTRAP 4 EN UN  
REPOSITORIO SQL SERVER CON CONTROL DE VERSIONES GIT  
BAJO EL MARCO DE TRABAJO SCRUM EN MICROSHIF**

**JAMES ALFRED CUADRADO LÁZARO**  
Código: 1'098.699.525

**Trabajo de grado presentado como requisito para optar al título de  
INGENIERO DE SISTEMAS**

**Director: Ing. YESID ALEXANDER MADRID**



**PROGRAMA DE INGENIERÍA DE SISTEMAS  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA,  
ELECTRÓNICA, SISTEMAS Y TELECOMUNICACIONES  
FACULTAD DE INGENIERIAS Y ARQUITECTURA  
UNIVERSIDAD DE PAMPLONA**

**2019**

## **Dedicatoria**

A mis padres, que cada día me brindas su apoyo incondicional para que pueda alcanzar mis metas. La educación que recibí va más allá que una institución, y se llama casa. Me hicieron caer en razón , que si haces las cosas bien y con foco y más apoyo de Dios, nada será imposible.

Por ellos, que son la bendición más preciosa que puedo tener, que serán los más felices por ser partícipes de esta meta cumplida. A Yuly y Lisette , mis feos como de cariño les digo, sin importar los problemas , siempre puedo contar con su apoyo incondicional.

## AGRADECIMIENTOS

No puedo comenzar mis agradecimientos sin pasar de alto. ¡Gracias Dios! Gracias por darme la vida y por sus bendiciones que los seres humanos las nombramos “casualidades”. Casualidades que se han aparecido a lo largo de mi existencia y por hacer de mi vida.

A mi familia, que nunca permiten que me detenga, que siempre se tiene que tener la frente en alto, que las cosas se hacen con calidad, responsabilidad y lo mejor, con el todo tu ser.

Gracias al Ingeniero Yesid Madrid, que durante este proceso ha sido más que un director, un amigo y porque las cosas no terminan aquí, vienen muchos proyectos más. A mi compadre Diego Daza, por ese apoyo y sinceridad, lo estimo mucho y tiene mi respeto. A todos los docentes que me apoyaron a cada instante, Omar Portilla, Ailin Orjuela y Sergio Peñaloza, aunque ya no haga parte de la Universidad, pues sus habilidades para transmitir el conocimiento va más allá de lo cotidiano.

A mi equipo gladiador de trabajo, muchos años estuvimos juntos y no bajamos la guardia ante cualquier situación. Admito que la capacidad de retención y entendimiento que tiene Luis Olaya es “letal” como lo dice el pana Víctor Rivera, el del sabor, el músico, el de las niñas. A mi bro, Diego Lozano, su guitarra y ukelele alegraron incontables momentos que compartimos y por último al señor de la plata, el del ganado, al pana Osman Ortega. Gracias equipo por su apoyo y energía.

También, a Nelsón Fernández, por brindarme la oportunidad de ser parte de su excelente grupo de trabajo. Al Ingeniero Sebastián Cortés y al equipo de desarrollo Neptuno Smart System, gracias por la oportunidad de poder realizar las pasantías y descubrir nuevas habilidades.

## RESUMEN EXTENDIDO

### **Desarrollo del módulo de contabilidad en lenguaje .NET Core cumpliendo con los principios de desarrollo SOLID y de arquitectura limpia usando los Framework Entity, Code First, DevExpress y Bootstrap 4 en un repositorio SQL Server con control de versiones Git bajo el marco de trabajo Scrum en Microshif**

James alfred Cuadrado Lázaro

*Universidad de Pamplona, Km 1 Via B/manga, Pamplona, Colombia*

---

#### **Resumen**

Este trabajo de innovación está enfocado en la utilización de los Frameworks Entity, Code First, DevExpress y Bootstrap 4, con el objetivo de crear un módulo enfocado a la contabilidad, aplicando arquitectura limpia basado en principios de desarrollo SOLID, ya que con esto se permite crear nuevas funcionalidades sin tener que modificar en gran medida código antiguo. En dicho modelo se establecen los cinco principios básicos del diseño y la programación orientada a objetos, que está relacionado con los patrones de diseño, en especial, con la alta cohesión y el bajo acoplamiento.

El alcance de este trabajo, es lograr la eficiencia del proceso de contabilidad para los clientes potenciales de la empresa Microshif, utilizando el marco de desarrollo con .NET Core, permitiendo el desarrollo y despliegue en la nube, en diferentes plataformas, llámese Windows, Mac o Linux.

Adicionalmente, el proyecto se llevará a cabo bajo el marco de trabajo Scrum, la cual es una metodología ágil para desarrollar productos innovadores y que, hoy en día es un estándar de la industria, además, hace parte de la filosofía de Microshif, que promueve la colaboración en los equipos de trabajo para lograr el desarrollo de productos complejos.

Y para finalizar, se realiza un manual de buenas prácticas de programación el cual está anexo en un capítulo.

**Palabras Claves:** Frameworks Entity y Code First, SOLID, .NET Core, Scrum, Innovación.

---

## **Abstract**

This innovation work is focused on the use of Frameworks Entity, Code First, DevExpress and Bootstrap 4, with the aim of creating a module focused on accounting, applying clean architecture based on SOLID development principles, since this allows you to create new features without having to modify much old code. This model establishes the five basic principles of design and object-oriented programming, which is related to design patterns, especially high cohesion and low coupling.

The scope of this work, is to achieve the efficiency of the accounting process for potential customers of the company Microshif, using the development framework with .NET Core, allowing the development and deployment in the cloud, on different platforms, whether called Windows, Mac or Linux.

In addition, the project will be carried out under the Scrum framework, which is an agile methodology for developing innovative products and is now an industry standard. It is also part of the Microshif philosophy, which promotes collaboration in work teams to achieve the development of complex products.

And finally, a manual of good programming practices is produced, which is annexed in a chapter.

**Keywords:** Frameworks Entity and Code First, SOLID, NET Core, Scrum, Innovation.

## TABLA DE CONTENIDO

<b>1</b>	<b>INTRODUCCIÓN</b> .....	<b>12</b>
<b>2</b>	<b>JUSTIFICACIÓN</b> .....	<b>13</b>
<b>3</b>	<b>OBJETIVOS</b> .....	<b>14</b>
3.1	Objetivo General .....	14
3.2	Objetivos Específicos.....	14
<b>4</b>	<b>MARCO REFERENCIAL</b> .....	<b>15</b>
4.1	Arquitectura Limpia .....	15
4.2	ASP.NET CORE .....	15
4.3	LINQ.....	16
4.4	C# .....	16
4.5	Principios de desarrollo SOLID .....	17
4.5.1	S - Responsabilidad simple (Single Responsibility) .....	17
4.5.2	O - Abierto/Cerrado (Open/Closed) .....	17
4.5.3	L - Sustitución Liskov (Liskov Substitution).....	18
4.5.4	I - Segregación del Interface (Interface Segregation) .....	18
4.5.5	D - Inversión de dependencias (Dependency inversion) .....	19
4.6	.NET Core .....	19
4.7	Framework Entity .....	20
4.8	DevExpress.....	21
4.9	Bootstrap 4.....	21
4.10	SQL Server .....	22
4.11	Git .....	22
4.12	SCRUM.....	23
4.13	IDE.....	25
<b>5</b>	<b>METODOLOGÍA</b> .....	<b>11</b>
5.1	Conceptualización de las herramientas tecnológicas necesarias para el desarrollo del módulo.....	11
5.2	Desarrollar un estado del arte de las herramientas tecnológicas necesarias para el desarrollo del módulo. ....	12
5.3	Diseñar las funcionalidades de gestión requeridas por el módulo de contabilidad: Plan único de cuentas, Paramétricas, Comprobantes, Conciliaciones. ....	12

5.4	Implementar las funcionalidades de gestión requeridas por el módulo de contabilidad: Plan único de cuentas, Paramétricas, Comprobantes, Conciliaciones.	13
5.5	Validar las funcionalidades de gestión requeridas por el módulo de contabilidad: Plan único de cuentas, Paramétricas, Comprobantes, Conciliaciones.	13
5.6	Documentar las buenas prácticas de las herramientas tecnológicas apropiadas durante el proyecto.	13
5.6.1	Buenas prácticas de desarrollo	14
5.6.2	Herramientas tecnológicas	21
5.6.3	Administración y Gestión de proyectos	12
5.7	Documentar el proceso de implementación de las funcionalidades del módulo de contabilidad.	13
5.8	Esquema del funcionamiento de las herramientas tecnológicas dentro del desarrollo del módulo.	¡Error! Marcador no definido.
<b>6</b>	<b>RESULTADOS</b>	<b>14</b>
6.1	Suite Neptuno	14
<b>7</b>	<b>ANÁLISIS Y DISCUSIÓN DE RESULTADOS</b>	<b>18</b>
7.1	Wireframing	18
7.2	Estructuración	19
<b>8</b>	<b>CONCLUSIONES</b>	<b>21</b>
<b>9</b>	<b>RECOMENDACIONES Y TRABAJOS FUTUROS</b>	<b>22</b>
<b>10</b>	<b>ANEXOS</b>	<b>24</b>
<b>11</b>	<b>REFERENCIAS</b>	<b>27</b>



## LISTA DE FIGURAS

Figura 1 Flujo de fases en GIT .....	23
Figura 2 Definición de roles SCRUM .....	25
Figura 3 Mala definición de variables .....	15
Figura 4 Correcta definición de variables .....	15
Figura 5 Correcta definición de formularios.....	15
Figura 6 Procedimiento para obtener una lista de tipo de identificación.....	17
Figura 7 Componentes Ajax del módulo contabilidad .....	17
Figura 8 Definición de comentarios.....	20
Figura 9 Visual Studio Code (Trakhtenberg, 2019).....	21
Figura 10 Visual Studio (Lui, 2019).....	21
Figura 11 Educative (GitHub, 2019) .....	12
Figura 12 Bootstrap Studio y Canva (GitHub, 2019) .....	12
Figura 13 Iconscout (GitHub, 2019).....	12
Figura 14 One Month (GitHub, 2019).....	12
Figura 15 GitKraken (GitHub, 2019).....	12
Figura 16 Icons8 (GitHub, 2019).....	12
Figura 17 Herramienta Microsoft Teams, tomado de (Microsoft, 2019) .....	11
Figura 18 Herramienta Microsoft Planner, tomado de (Microsoft, 2019).....	11
Figura 19 Pizarra de trabajo Trello, versión web, tomado de (Atlassian, 2019) .....	12
Figura 20 Herramienta Slack, versión web, tomado de (Technologies, 2019).....	13
Figura 21 Vista de Login del aplicativo.....	14
Figura 22 Módulos de administración .....	14
Figura 23 Ventana de inicio del módulo General .....	15
Figura 24 Vista principal de gestión de terceros.....	15
Figura 25 Vista agregar Tercero .....	16
Figura 26 Vista de editar Tercero, información general .....	16
Figura 27 Vista de editar Tercero, pestaña contabilidad.....	17
Figura 28 Vista de eliminar Tercero .....	17
Figura 29 Wireframe de gestión de terceros .....	18
Figura 30 Estructura de carpetas y archivos basado en la Arquitectura Limpia de Terceros .....	19
Figura 31 Referencias y flujos entre capas de la Arquitectura Limpia.....	20
Figura 32 Esquema de herramientas – infraestructura - usuarios .....	23
Figura 33 Estructura de carpetas y archivos basado en la Arquitectura Limpia de Plan contable	24
Figura 34 Módulos de contabilidad y financiero .....	24
Figura 35 Ventana de inicio del módulo contabilidad .....	25
Figura 36 Gestión de Retenciones .....	25
Figura 37 Vista agregar retención.....	25
Figura 38 Gestión de Plan contable .....	26
Figura 39 Vista agregar un plan contable .....	26

## LISTA DE TABLAS

Tabla 1 Prefijos de controles.....	16
Tabla 2 Prefijos de controles, tomado de ( Servicio fitosanitario del estado, 2019) .....	18
Tabla 3 Encabezado de comentarios.....	19

# 1 INTRODUCCIÓN

Esta investigación surge como necesidad de los clientes potenciales que cuenta Microshif, que actualmente necesitan implementar el módulo de contabilidad en su organización. Este proceso anteriormente se llevaba de manera manual; hace ya muchos años atrás, pero gracias a los grandes avances en materia de tecnología y de querer llevarlo a otro nivel por medio de los sistemas de información. Al avanzar los años fue cambiando el enfoque de cómo debería funcionar el sistema de información, en el caso de algunos clientes de la empresa cuentan con software de escritorio, donde cada cliente requiere una copia del aplicativo para cada usuario. El ciclo de vida de ese tipo de software ya está finalizando, pues con la llegada de nuevas tecnologías y herramientas que permiten la fácil implementación de soluciones de desarrollo de software, junto a las buenas prácticas de codificación o programación para la creación de un sin número de funcionalidades sin tener que modificar una cierta parte de código antiguo. Actualmente las empresas de cualquier nivel, se ven en la obligación de mejorar los procesos internos de su organización, además si se trata de la parte contable. Para poder escalar y llegar a las grandes categorías, es necesario pasar por el proceso de maduración y optimización de los procesos dentro de su estructura organizacional. En otras palabras, para estar entre los mejores, es necesario estar constituida con lo mejor, tener implementada funcionalidades con lo último en innovación en el mercado, ya que con ellas las empresas pueden ser más competitivas en el mercado.

Teniendo en cuenta lo planteado anteriormente, el siguiente proyecto busca responder la siguiente pregunta ¿Cómo soportar el proceso de contabilidad en los clientes de Microshif por medio del desarrollo de un módulo en lenguaje .Net Core cumpliendo con los principios de desarrollo SOLID y de arquitectura limpia usando los Frameworks Entity, Code First, DevExpress y Bootstrap 4 en un repositorio SQL Server con control de versiones Git bajo el marco de trabajo Scrum?

## 2 JUSTIFICACIÓN

La pregunta de investigación que se resolvió fue ¿Cómo soportar el proceso de contabilidad en los clientes de Microshif por medio del desarrollo de un módulo en lenguaje .Net Core cumpliendo con los principios de desarrollo SOLID y de arquitectura limpia usando los Frameworks Entity, Code First, DevExpress y Bootstrap 4 en un repositorio SQL Server con control de versiones Git bajo el marco de trabajo Scrum?

Las herramientas utilizadas por los desarrolladores de la empresa cuentan con altos estándares de calidad del mercado, y sin perder el enfoque, este proyecto se realizó en .Net Core, que es una plataforma de desarrollo de código abierto, en dónde una gran parte del proceso de actualización y mantenimiento lo realiza Microsoft y la comunidad de desarrolladores. Cabe recordar que Microsoft es la empresa desarrolladora del sistema operativo más popular en el ecosistema, con una cuota del mercado de alrededor del 90% de los equipos de escritorio. Por otra parte, este proyecto se desarrolló bajo el marco de trabajo Scrum, ya que es una metodología ágil, en donde se entrega un producto funcional en corto tiempo y con este proceso de gestión, se reduce la complejidad en el desarrollo del producto para satisfacer las necesidades de los clientes.

El desarrollo de este proyecto, se basó en el principio de diseño de software de arquitectura limpia, el cual maneja separación de responsabilidades, para la codependencia de procesos, donde cada capa tiene tareas concretas e independientes dentro del módulo. Gracias a esta separación, se permite conseguir un sistema en donde se pueda abordar muy detalladamente la fase de mantenimiento de manera más legible y sencilla.

## **3 OBJETIVOS**

### **3.1 Objetivo General**

Desarrollar el módulo de contabilidad utilizando arquitectura limpia basado en principios de desarrollo SOLID en lenguaje .NET Core implementado los Framework Entity, Code First, DevExpress y Bootstrap 4 en un repositorio SQL server con control de versiones Git bajo el marco de trabajo Scrum en Microshif.

### **3.2 Objetivos Específicos**

- Compilar el estado del arte y la información conceptual necesaria de las herramientas tecnológicas a usar en el desarrollo del módulo.
- Implementar las funcionalidades de gestión requeridas por el módulo de contabilidad.
- Documentar las buenas prácticas de las herramientas utilizadas.

## **4 MARCO REFERENCIAL**

### **4.1 Arquitectura Limpia**

Es un principio de diseño de software, maneja separación de responsabilidades. No existe una única arquitectura limpia, pero todas ellas se basan en lo mismo. Se establece una arquitectura por capas, cada una de ellas con una responsabilidad específica. Gracias a esta separación, se consiguen sistemas independientes de librerías de terceros, se puede realizar Tests, cada capa es independiente de la vista, también, son independientes de la tecnología de base de datos y de agentes exteriores. (Ruben, 2019)

### **4.2 ASP.NET CORE**

Es un marco de trabajo que permite la creación de aplicaciones altamente portables, virtualizables e independientes de la nube, lo que genera una administración adecuada de los costos asociados al despliegue de la aplicación. Así como facilidad para esos clientes que no tengan capacidad para obtener y administrar una arquitectura en la nube. El uso de un marco de trabajo definido como .NET CORE da la facilidad de contar con las tecnologías necesarias para la implementación de elementos transversales a la arquitectura, las cuales se definen a continuación:

- Autenticación - ASP .NET Core Identity
- Autorización - ASP .NET Core Role Authorization
- Auditoria - Entity Framework Audit.NET
- Data Access - Entity Framework Core
- Web Pages - ASP.NET Core Razor Pages
- Logging - ASP .NET Core LoggingFactory

### 4.3 LINQ

Language Integrated Query (LINQ), como su nombre en inglés lo indica, es un lenguaje integrado de consultas. Al igual que SQL para las bases de datos relaciones o XQuery para los archivos XML, LINQ es una herramienta desarrollada para la plataforma .NET por Microsoft cuyo objetivo es sintetizar y simplificar la manera en como se hacen consultas sobre una fuente de datos, por ello, LINQ soporta distintos tipos de fuentes de datos y formatos, dado que su enfoque es para objetos o estructuras de datos que trabajan con objetos y esto permite la integración entre LINQ y Entity Framework para el trabajo continuo entre bases de datos sin que el código fuente se vea afectado de acuerdo al motor de base de datos asociado. (Microsoft, Introduction to LINQ Queries (C#), 2015)

### 4.4 C#

C# es un lenguaje de programación desarrollado y estandarizado por Microsoft que apareció en el año 2000 como parte de la iniciativa .NET y tiene referencias de lenguajes como C y C++. Es multiparadigma, fuertemente orientado a objetos. (Microsoft, Un paseo por el lenguaje C#, 2019)

C# tiene como objetivo ser un lenguaje de programación simple, moderno y de propósito general, donde sus implementaciones provean soporte desde los principios de la ingeniería de software de tal modo que la robustez y durabilidad del software junto con la productividad del programador se reflejen a lo largo del desarrollo de los productos. Asimismo, las aplicaciones desarrolladas bajo C# están destinadas a tener un bajo consumo de memoria y poder de procesamiento. (Microsoft, 2019)

## **4.5 Principios de desarrollo SOLID**

Es un acrónimo inventado por Robert C. Martin para establecer los cinco principios básicos del diseño y la programación orientada a objetos. Este acrónimo tiene bastante relación con los patrones de diseño, en especial, con la alta cohesión y el bajo acoplamiento.

El objetivo de tener un buen diseño de programación es abarcar la fase de mantenimiento de una manera más legible y sencilla, así como conseguir crear nuevas funcionalidades sin tener que modificar en gran medida el código antiguo. Los costes de mantenimiento pueden abarcar el 80% de un proyecto de software por lo que hay que valorar un buen diseño. (RUBIRA, 2019)

### **4.5.1 S - Responsabilidad simple (Single Responsibility)**

Este principio trata de destinar cada clase a una finalidad sencilla y concreta. En muchas ocasiones se coloca un método reutilizable el cual no tiene ninguna relación con la clase porque solo provee un uso específico. El problema surge cuando se necesita utilizar ese mismo método desde otra clase. Si no se refactoriza en ese momento y se crea una clase destinada para dicha finalidad que deberían ser su responsabilidad. (RUBIRA, 2019)

### **4.5.2 O - Abierto/Cerrado (Open/Closed)**

El principio atribuido a Bertrand Meyer que habla de crear clases extensibles sin necesidad de entrar al código fuente a modificarlo. Es decir, el diseño debe ser abierto para poderse extender, pero cerrado para poderse modificar. Aunque dicho parece fácil, lo complicado es predecir por donde se debe extender y que no tengamos que modificarlo. Para conseguir este principio hay que tener muy claro cómo va a funcionar la aplicación, por donde se puede extender y cómo van a



interactuar las clases. (RUBIRA, 2019)

El uso más común de extensión es mediante la herencia y la reimplementación de métodos. Existe otra alternativa que consiste en utilizar métodos que acepten una interface de manera que se pueda ejecutar cualquier clase que implemente la misma. En todos los casos, el comportamiento de la clase cambia sin interacción con el código interno.

#### **4.5.3 L - Sustitución Liskov (Liskov Substitution)**

Este principio fue creado por Barbara Liskov y habla de la importancia de crear todas las clases derivadas para que también puedan ser tratadas como la propia clase base. Cuando se crean clases derivadas se debe asegurar de no re-implementar métodos que hagan que los métodos de la clase base no funcione. (RUBIRA, 2019)

#### **4.5.4 I - Segregación del Interface (Interface Segregation)**

Este principio fue formulado por Robert C. Martin y trata de algo parecido al primer principio. Cuando se definen interfaces estos deben ser específicos a una finalidad concreta. Por ello, si se tiene que definir una serie de métodos abstractos que debe utilizar una clase a través de interfaces, es preferible tener muchos interfaces que definan pocos métodos que tener un Interface con muchos métodos. (RUBIRA, 2019)

El objetivo de este principio es principalmente poder aprovechar las interfaces en otras clases. Si tenemos un Interface que compara y clona en el mismo Interface, de manera más complicada se podrá utilizar en una clase que solo debe comparar o en otra que solo debe clonar. (RUBIRA, 2019)

#### **4.5.5 D - Inversión de dependencias (Dependency inversion)**

Fue definido por Robert C. Martin. El objetivo de este principio, es conseguir desacoplar las clases. En todo diseño siempre debe existir un acoplamiento, pero hay que evitarlo en la medida de lo posible. Un sistema no acoplado no hace nada, pero un sistema altamente acoplado es muy difícil de mantener.

El objetivo de este principio es el uso de abstracciones para conseguir que una clase interactúe con otras clases sin que las conozca directamente. Es decir, las clases de nivel superior no deben conocer las clases de nivel inferior. Dicho de otro modo, no debe conocer los detalles. Existen diferentes patrones como la inyección de dependencias o Service Locator que nos permiten invertir el control. (RUBIRA, 2019)

#### **4.6 .NET Core**

Es una plataforma de desarrollo de uso general de código abierto de cuyo mantenimiento se encargan Microsoft y la comunidad .NET en GitHub. Además, es una implementación de .NET Standard para uso general, modular, multiplataforma y de código abierto. Contiene muchas de las API de .NET Framework (aunque .NET Core es un conjunto más pequeño), e incluye componentes de entorno en tiempo de ejecución, marco de trabajo, compilador y herramientas que admiten diversos sistemas operativos y procesadores. La implementación de .NET Core se debe principalmente a las cargas de trabajo de ASP.NET Core, pero también a la necesidad y la intención de tener una implementación más moderna. Es multiplataforma, admite Windows, macOS y Linux y puede usarse para compilar aplicaciones de dispositivo, nube e IoT.

.NET Core es código abierto y fue presentado a .NET Foundation por Microsoft en 2014. Ahora es uno de los proyectos más activos de .NET Foundation. Todos los individuos y organizaciones pueden adoptarlo libremente, con cualquier fin: personal, académico o comercial. Varias empresas usan .NET Core como parte de aplicaciones, herramientas, nuevas plataformas y servicios de hospedaje. Algunas de estas empresas realizan contribuciones significativas a .NET Core en GitHub y proporcionan una guía sobre la dirección del producto como parte del Technical Steering Group de .NET Foundation. (Microsoft, .NET Core y código abierto, 2019)

#### **4.7 Framework Entity**

El mapeo objeto-relacional o Object-Relational mapping (ORM) Entity Framework es un conjunto de tecnologías que permite el desarrollo de aplicaciones de software orientadas a los datos. Para esto, el framework permite modelar entidades, relaciones y lógica del negocio desde el código que, al ser migrado con el uso de métodos propios de la herramienta, se ve reflejado en el modelo de datos del repositorio de datos configurado. Es decir, con Entity Framework, se hace uso primero de los modelos de datos y sus relaciones y de esto se mapea, por ejemplo, la base de datos o el repositorio de datos que se configure. (L. Martinez, 2015)

Entity Framework permite conectar una aplicación escrita en .NET a cualquier repositorio de datos sin necesidad de cambiar conexiones y conectores según el motor de base de datos al que se deba conectar, o, de llegar a ser necesario, se puede configurar para tener distintas bases de datos conectadas al aplicativo y hacer uso de ellas como se especifique. (Microsoft, Entity Framework overview, 2019)

## **4.8 DevExpress**

Es una de las más completas suites de componentes de UI para el desarrollo en todas las plataformas de .NET como Windows Forms, ASP.NET, MVC, Silverlight y Windows 8 XAML. Una de sus numerosas ventajas se encuentra la reducción de tiempo de desarrollo considerablemente al no tener que escribir todo el código. Es altamente personalizable. Mejora el rendimiento de las aplicaciones al optimizar el código de llenado de las vistas. La creación de los componentes es semi-automática, DevExpress se encarga de realizar todo el código necesario para la visualización y llenado de los componentes según la plataforma utilizada y otras más. (DevExpress, 2019)

## **4.9 Bootstrap 4**

Es un Framework (conjunto de herramientas) de código abierto para desarrollar con HTML, CSS y JS, desarrollado inicialmente (en el año 2011) por Twitter que permite dar forma a un sitio web mediante librerías CSS que incluyen tipografías, botones, cuadros, menús y otros elementos que pueden ser utilizados en cualquier sitio web.

Aunque el desarrollo del Framework Bootstrap fue iniciado por Twitter, fue liberado bajo licencia MIT en el año 2011 y su desarrollo continúa en un repositorio de GitHub.

Bootstrap es una excelente herramienta para crear interfaces de usuario limpias y totalmente adaptables a todo tipo de dispositivos y pantallas, sea cual sea su tamaño. Además, Bootstrap ofrece las herramientas necesarias para crear cualquier tipo de sitio web utilizando los estilos y elementos de sus librerías. (Fontela, 2019)

#### **4.10 SQL Server**

Es un sistema de gestión de base de datos relacional, desarrollado por la empresa Microsoft. Es el cimiento de la completa plataforma de gestión de datos de Microsoft que entrega un nivel de desempeño interesante para aplicaciones de misión crítica a través de tecnologías en-memoria, perspectivas más rápidas proveniente de cualquier tipo de datos con herramientas familiares como Excel y una plataforma para desarrollar, implementar y administrar soluciones que abarquen tanto implementaciones locales como en la nube.

Puede ser configurado para utilizar varias instancias en el mismo servidor físico, la primera instalación lleva generalmente el nombre del servidor, y las siguientes – nombres específicos (con un guion invertido entre el nombre del servidor y el nombre de la instalación). (Sitio, 2019)

#### **4.11 Git**

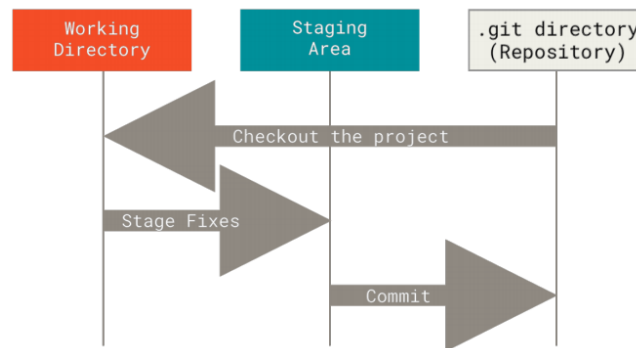
Es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o frontEnd como Cogito o StGIT. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux.

Git se basa en analizar las diferencias entre la versión antigua y la nueva versión del archivo que se ha modificado para evitar tener información duplicada en su sistema y esto lo realiza mediante

snapshots que son como “fotos” de los archivos que fueron modificados al momento de realizar un “commit”.

Un commit, es una de las operaciones que se puede realizar con el repositorio y hace parte de los tres estados que maneja GIT y son mostrados a continuación.



**Figura 1 Flujo de fases en GIT**

Los tres estados que maneja git son: modificado, escenificado (staged) y comprometido (committed).

Con estas tres fases, Git garantiza el control de versiones de los archivos y generar la persistencia de los cambios en un repositorio que puede ser local o remoto si así se configura.

## 4.12 SCRUM

Scrum es un marco de trabajo (framework) para la gestión de proyectos cuyo objetivo principal es mantener una constante relación con el cliente generando un contexto iterativo que permite la rápida adaptación y solución a solicitudes que surjan durante el desarrollo del proyecto.

El término Scrum aparece en 1986 por Hirotaka Takeuchi e Ikujiro Nonoka en su artículo “The New New Product Development” publicado en la Harvard Business Review (Nonoka, 1986) pero

hasta el año 2001 se incluyó en el Manifiesto Ágil como una metodología de desarrollo de software. (Beedle, 2001)

Scrum no es un proceso, una técnica, o un método definido, en lugar de eso es un marco de trabajo dentro del cual se puede emplear varios procesos y técnicas. Scrum muestra la eficacia relativa de las técnicas de gestión de producto y las técnicas de trabajo de modo que se pueda mejorar continuamente el producto, el equipo y el entorno de trabajo.

La metodología de desarrollo de software Scrum se ejecuta desde un equipo (Team Scrum) apoyado por el Scrum Master y el Product Owner junto con el (o los) Stakeholder(s) garantizando la inclusión de todos los participantes del desarrollo del proyecto.

**El Team Scrum** es el equipo de personas encargadas de la construcción del producto. Es importante que todos los miembros de éste equipo sigan un objetivo común, se adhieran a las mismas normas o reglas y muestren respeto el uno al otro. El tamaño ideal de éste equipo es entre 5 y 9 personas pues esto garantiza que sea lo suficientemente pequeño como para permanecer ágil y lo suficientemente grande como para completar una cantidad de trabajo significativa. (Alaimo M. , 2019)

**El Product Owner** representa la voz del cliente y es el encargado de maximizar el valor empresarial para el proyecto y el producto, es una persona que mantiene una visión dual entre entender y apoyar las necesidades e intereses de todos los Stakeholders y comprender las necesidades y el funcionamiento del Team Scrum. Es importante resaltar que este rol lo puede cumplir una única persona, no puede ser un comité.

**El Scrum Master** es un líder que está al servicio del Equipo Scrum y ayuda a las personas externas a entender las interacciones con el Scrum Team que puedan ser útiles y las que no. Es el

responsable de promover, apoyar y ayuda a todos a entender la teoría, práctica, reglas y valores de Scrum, encargándose de eliminar los “impedimentos” que encuentre en el equipo. (Alaimo M. , 2013)

Para que cualquier proyecto tenga éxito, la comunicación es importante, es por ésto que Scrum plantea una serie de eventos o ceremonias que se dan de manera iterativa durante el desarrollo del proyecto con el objetivo de gestionar los equipos y que todas las personas involucradas se encuentren al día de cualquier novedad. Las ceremonias planteadas son:

- Sprint planning meeting.
- Daily scrum meeting.
- Sprint review meeting.
- Sprint retrospective meeting



**Figura 2 Definición de roles SCRUM**

#### **4.13 IDE**

IDE del acrónimo de Integrated Development Environment (Entorno de desarrollo integrado en español), es un software que contiene un “paquete” de lo que necesitamos, por lo cual se le denomina entorno, dentro de sus características está, en contener un editor de código, un



compilador, un depurador y estos pueden trabajar con un lenguaje de programación específico o con más de uno. Los IDEs nos ayudan a poder trabajar de una forma mucho más amigable y fácil, ya que por ejemplo al escribir todo nuestro código en un editor de texto y este contiene un error no sabremos en donde se encuentra. (Nicolas, 2019)

Se dice que un buen IDE debe tener:

- Resaltado de sintaxis.
- Completado de código.
- Búsqueda de código.
- Resaltado de errores y advertencias.
- Refactorización y generación de código.
- Depuración de código.

## 5 METODOLOGÍA

Se realizó un análisis exploratorio con base a unos datos suministrados por la empresa. El módulo de contabilidad optimizará muchos procesos que se llevan de manera manual en los clientes. Se utilizó la metodología mixta, conceptualizando las herramientas tecnológicas Framework Entity, Code First, DevExpress y Bootstrap 4, para así poder realizar un desarrollo de software con estándares de la industria moderna y que por su eficiencia permite abordar cualquier tipo de problemas de manera exploratoria.

### 5.1 Conceptualización de las herramientas tecnológicas necesarias para el desarrollo del módulo

Para poder abordar cualquier proyecto e independientemente el área de enfoque, es necesario realizar una conceptualización de las posibles herramientas tecnológicas requeridas para llevarlo al éxito.

Primero se realizó una capacitación, por tal razón se recibe la socialización de enfoque de trabajo y orientación de desarrollo por parte del equipo y el director de desarrollo encabezaba la lista. Gracias a dicha jornada, se nombran las herramientas tecnológicas de desarrollo, gestión de proyectos y testing; utilizadas actualmente en la empresa, con la respectiva definición y las funciones vitales que desempeñan en los proyectos que están en ejecución. Por medio de múltiples escenarios controlados, se realizan una variedad de ejemplos basados en proyectos de desarrollo vinculando dichas herramientas, en algunos casos, partiendo desde cero hasta llegar a la implementación.

## **5.2 Desarrollar un estado del arte de las herramientas tecnológicas necesarias para el desarrollo del módulo.**

Cumpliendo satisfactoriamente la etapa de conceptualización con todo el equipo de trabajo, se hace más fácil desarrollar el estado del arte de las herramientas. Se recurre a fuentes virtuales, tales como, documentación de los Frameworks en las respectivas páginas oficiales, video tutoriales de implementación de componentes.

Por otra parte y de prioridad media, fue necesario fundamentar terminología de contabilidad. Para que el proyecto cumpliera la meta y el avance fuera significativo era imprescindible tener la fundamentación de conceptos claves para poder desarrollar el aplicativo, buscando en medios digitales y fuentes confiables tales como libros digitales, esta información constó de fundamentación teórica (concepto) para la apropiación del tema a abordar.

Conceptos como plan contable, terceros, retenciones, comprobantes, conciliaciones y paramétricas fueron estudiados detenidamente. Se llevaron a cabo muchas reuniones con el director de desarrollo que diseñó el antiguo módulo y profesionales del área contable, para así poder llevarlo al desarrollo.

## **5.3 Diseñar las funcionalidades de gestión requeridas por el módulo de contabilidad:**

### **Plan único de cuentas, Paramétricas, Comprobantes, Conciliaciones.**

Primero fue indispensable diseñar las vistas de casos de usos por parte del equipo de diseño de la empresa, todas estas contemplaban futuras modificaciones.

Para el desarrollo del módulo se utilizó .Net Core como lenguaje de programación para el proceso de codificación de clases, métodos, interfaces, interactores e implementaciones. Para las funciones se usó javascript y en las vistas se empleó .Net, DevExpress y Bootstrap 4.

#### **5.4 Implementar las funcionalidades de gestión requeridas por el módulo de contabilidad: Plan único de cuentas, Paramétricas, Comprobantes, Conciliaciones.**

Después de terminar satisfactoriamente la etapa de codificación, y por ser un proyecto de desarrollo Code first, es decir, se enfoca primero al código de implementación y luego por medio del Framework Entity, se despliegan los esquemas y entidades dentro de un servidor de base de datos particular, en este caso SQL Server.

Con Entity Framework se realizó la compilación del proyecto y se implementó en el servidor para ser accedido al servicio web por el protocolo HTTP.

#### **5.5 Validar las funcionalidades de gestión requeridas por el módulo de contabilidad: Plan único de cuentas, Paramétricas, Comprobantes, Conciliaciones.**

Para poder entregar un producto con calidad en proyectos de software es necesario someter a pruebas de funcionalidad, pensar como un usuario y ejecutar funciones comunes, en otras palabras, pruebas de caja negra. Para dicha tarea se asignó a una persona de testing el cual se encargó de encontrar fallas. Después que pasó por dichas pruebas, y dar su aprobación, un tercero se encarga de realizar pruebas de seguridad al servidor y de todos módulos que ya se encuentran implementados.

#### **5.6 Documentar las buenas prácticas de las herramientas tecnológicas apropiadas durante el proyecto.**

Después de haber experimentado y ser parte del proyecto Suite Neptuno, construcción del módulo de contabilidad y gestión de terceros en Microshif, todo programador debe tener claro una serie

de buenas prácticas de desarrollo, aplicaciones tecnológicas de desarrollo y administración de proyectos para poder competir con categoría dentro del mercado y ser apetecido en el mismo.

### **5.6.1 Buenas prácticas de desarrollo**

En grandes proyectos, donde suele haber una participación activa de desarrolladores, diseñadores u otros roles, combinados con una errada interpretación y técnicas no efectivas de manejo del código fuente, sumándole la mala utilización de metodologías de estructura de desarrollo de sistemas, aumenta el riesgo de afectar la calidad del producto final, además de presentar incumplimientos en las fechas de entregas y esto causaría gastos excesivos en relación a los costos estimados en la planeación del proyecto e insatisfacción del cliente.

#### **5.6.1.1 Definición de variables**

Sin importar el lenguaje de programación y el nivel de conocimiento, se recomienda tener bien claro la definición de las variables o clases utilizadas en todo el proyecto, cabe recordar que la sugerencia parte de que en materia de mantenimiento y actualizaciones futuras del aplicativo es necesario y que en la actualidad es un estándar de desarrollo que se debe cumplir. ( Servicio fitosanitario del estado, 2019)

Declarar todas las variables ahorra tiempo de programación porque reduce el número de errores debidos a errores. En la Figura 3 se puede observar una mala definición de variables y en la Figura 4 se puede observar la correcta definición de variables. Las variables deben llevar un prefijo para indicar su tipo de dato. Opcionalmente, y en especial para programas largos, el prefijo se puede ampliar para indicar el alcance de la variable.

```

var form1 = $("#btnInsertGeneral").dxForm('instance');
var form1data = form1.option('formData');

var form2 = $("#btnInsertContabilidad").dxForm('instance');
var form2data = form2.option('formData');

```

Figura 3 Mala definición de variables

```

var formGeneral = $("#btnInsertGeneral").dxForm('instance');
var formDataGeneral = formGeneral.option('formData');

var formContabilidad = $("#btnInsertContabilidad").dxForm('instance');
var formDataContabilidad = formContabilidad.option('formData');

```

Figura 4 Correcta definición de variables

### 5.6.1.2 Definición básica de formularios y clases

En el caso de formularios y clases, no se recomienda utilizar sólo el nombre en el identificador (ID). Ejemplo: si el formulario es para visualizar información de saldos, se debe preferir como se muestra en la Figura 5. Como pueden ver, la definición incluye el uso de letras en mayúsculas y minúsculas. Se llama notación camello, es más fácil formViewSaldos, que FORMVIEWSALDOS o formviewsaldos.

```

<div>
  <div>
    @Html.AntiForgeryToken()
    @using (Html.DevExtreme().ValidationGroup())
    {
      @Html.DevExtreme().Form<PlanContables>()
        .ID("formViewSaldos")
        .ReadOnly(true)
        .ColCount(2)
        .ElementAttr("class", "saldosCont")
        .Items(formItems =>
        {

```

Figura 5 Correcta definición de formularios

### 5.6.1.3 Definición de nombres básicas para controles

Cuando se despliegan datos de cualquier tipo de control, el resultado debe tener el mismo nombre. Es decir, si al leer alguna información de una base de datos, ésta retorna un texto, como ejemplo,

apellido paterno, el cuadro se podría llamar apellidoPaterno, con el prefijo “txt”, quedando de esta forma “txtApellidoPaterno”. ( Servicio fitosanitario del estado, 2019)

Para la definición de convenciones de nombres se debe usar un prefijo coherente que facilite la identificación del tipo de objeto, como se muestra en la Tabla 1.

**Tabla 1 Prefijos de controles**

Abbreviations	Standard Control
btn	Button
cb	CheckBox
cbl	CheckBoxList
ddl	DropDownList
fu	FileUpload
hdn	HiddenField
hik	Hyperlink
img	Image
lbl	Label
lbtn	LinkButton
mv	MultiView
pnl	Panel
txt	TextBox
DataGrid	dtg

#### 5.6.1.4 Definición básica para procedimientos

Este es un punto importante en las tareas de definición de nombres. Sin embargo, poco a poco se ha convertido en una buena norma el indicar una descripción precisa (y sin embargo corta) del procedimiento que se refiera. A estos nombres puede antecederse un prefijo sub para distinguir a los procedimientos. No obstante, se deberá anteceder con las letras get y post para indicar si un procedimiento obtendrá información de alguna fuente (get) o la grabará (post). ( Servicio fitosanitario del estado, 2019)

En la Figura 6 , se puede observar un claro ejemplo de la correcta definición del procedimineto para obtener una lista de tipos de indentificación.

```
[HttpGet]
[ValidateAntiForgeryToken]
0 references | James Cuadrado, 23 days ago | 1 author, 2 changes | 0 exceptions
public JsonResult OnGetListTipoIdentificacion(DataSourceLoadOptions options, string tipoPersona)
{
    List<TipoIdentificacion> tipoIdentificacionesList = new List<TipoIdentificacion>();

    try
    {
        tipoIdentificacionesList = _iInteractorTercero.FilterTipoIdentificacionByTipoPersona(tipoPersona);
    }
    catch (Exception ex)
    {
        Logger logger = LogManager.GetCurrentClassLogger();
        logger.Error(ex, "Error al obtener la lista de tipos de identificación");
    }
    return new JsonResult(DataSourceLoader.Load(tipoIdentificacionesList, options));
}
```

Figura 6 Procedimiento para obtener una lista de tipo de identificación

### 5.6.1.5 Implementación de componentes AJAX

Utilización de componentes ajax (acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML)), para el desarrollo de aplicaciones interactivas. Permitiendo que las aplicaciones que utilicen componentes ajax, se ejecuten en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. ( Servicio fitosanitario del estado, 2019)

En la Figura 7 se muestran los componentes Ajax utilizados en el módulo de contabilidad.

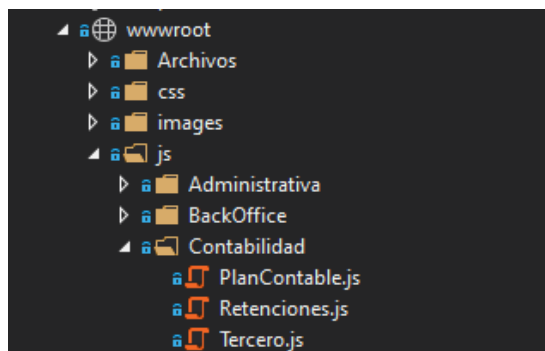


Figura 7 Componentes Ajax del módulo contabilidad



### 5.6.1.6 Nombres descriptivos de procedimientos

El cuerpo de un nombre de procedimiento se debe escribir en mayúsculas y minúsculas y debe tener la longitud necesaria para describir su funcionalidad. Además, los nombres de funciones deben empezar con un verbo, como IniciarNombreMatriz o CerrarDialogo. Para nombres que se usen con frecuencia o para términos largos, se recomienda usar abreviaturas estándar para que los nombres tengan una longitud razonable. En general, los nombres de variables con más de 32 caracteres pueden ser difíciles de leer. Cuando se usen abreviaturas, hay que asegurarse de que sean coherentes en toda la aplicación. Por añadidura, aunque los nombres de variables y procedimientos en las actuales herramientas de desarrollo (IDE), soporta el uso de caracteres acentuados y eñes, estos no deben de ser utilizados. ( Servicio fitosanitario del estado, 2019)

En la Tabla 2 se muestran algunos de los prefijos más importantes y la correcta definición.

**Tabla 2 Prefijos de controles, tomado de ( Servicio fitosanitario del estado, 2019)**

Kind	Rule
Private field	_lowerCamelCase
Public field	UpperCamelCase
Protected field	UpperCamelCase
Internal field	UpperCamelCase
Property	UpperCamelCase
Method	UpperCamelCase
Class	UpperCamelCase
Interface	IUpperCamelCase
Local variable	lowerCamelCase
Parameter	lowerCamelCase

### 5.6.1.7 Convenciones de comentarios al código

Todos los procedimientos y funciones deben comenzar con un comentario breve que describa las características funcionales del procedimiento (qué hace). Esta descripción no debe describir los

detalles de implementación (cómo lo hace), porque a veces cambian con el tiempo, lo que da como resultado un trabajo innecesario de mantenimiento de los comentarios o comentarios erróneos. El propio código y los comentarios de líneas necesarios describirán la implementación. Los argumentos que se pasan a un procedimiento se deben describir cuando sus funciones no sean obvias y cuando el procedimiento espera que los argumentos estén en un intervalo específico. También hay que describir, al principio de cada procedimiento, los valores de retorno de funciones y las variables globales que modifica el procedimiento. ( Servicio fitosanitario del estado, 2019)

En la Tabla 3 se describe las secciones que debe tener un comentario de encabezado en los procedimientos.

**Tabla 3 Encabezado de comentarios**

Encabezado de sección	Descripción del comentario
Finalidad	Lo que hace el procedimiento (no cómo lo hace).
Premisas	Lista de cada variable externa, control, archivo abierto o cualquier otro elemento que no sea obvio.
Efectos	Lista de cada variable externa, control o archivo afectado y el efecto que tiene (sólo si no es obvio).
Entradas	Todos los argumentos que puedan no ser obvios. Los argumentos se escriben en una línea aparte con comentarios de línea.
Resultados	Explicación de los valores devueltos por las funciones.

Se debe tener en cuenta que en cada método declarando se debe incluir un comentario que describa su funcionalidad. Las variables, controles y procedimientos deben tener un nombre bastante claro para que los comentarios sólo sean necesarios en los detalles de implementación complejos, como se puede observar en la **¡Error! No se encuentra el origen de la referencia.** que el nombre es bastante claro y que en la parte superior se encuentra el respectivo summary (comentario) respectivo.

```
/// <summary>
/// Método para buscar un plan contable por medio del id de una básica
/// </summary>
/// <param name="id">Id de la básica</param>
/// <param name="estado">estado en el que se encuentra</param>
/// <returns>Un objeto de tipo plan contable</returns>
2 referencias | James Cuadrado, Hace 5 días | 1 autor, 2 cambios | 0 excepciones
PlanContables FindPlanContableByIdOfBasica(int id, string estado);
```

Figura 8 Definición de comentarios

### 5.6.1.8 Generación de texto al usuario

Mensajes del texto, sobre todo informativo y diálogos de advertencia, necesitan ser llevados de forma apropiada y consistente. ( Servicio fitosanitario del estado, 2019)

Algunas reglas:

- Evite jerga técnica.
- Límite los mensajes a 2-3 líneas.
- Evite redacción que culpe al usuario.
- Evite el uso de abreviaturas.
- Los mensajes se alinearán a la Izquierda cuando sean de líneas múltiples.
- Nunca, incluya faltas de ortografía, errores de gramática o pronunciación incorrecta.
- Uso del conjunto de caracteres apropiado.

### 5.6.1.9 Menús

Su forma y colores deben de ser siempre consistentes en todos los sistemas, los ítems a seleccionar deben tener teclas de acceso rápido, en caso de utilizar íconos o imágenes en los menús estos deben

ser consistentes, en todo el sistema y serán agregados tras previa discusión y puesta en estándares por el grupo de desarrolladores.

Para cualquiera de los tipos de menús, los textos deberán comenzar con un verbo, preferiblemente en infinitivo y que sea suficientemente descriptivo de la acción que realizará. ( Servicio fitosanitario del estado, 2019)

## **5.6.2 Herramientas tecnológicas**

### **5.6.2.1 IDE de desarrollo**

En la actualidad existen muchos entornos de desarrollo que contienen los paquetes necesarios para desarrollar y además brindan múltiples ayudas para trabajar de forma práctica. Dentro de todos ellos se recomienda Visual Studio Code , una herramienta para programar todo tipo de lenguaje excepto .Net, en ese caso se recomienda Visual Studio . Es un IDE completo para programar, depurar, probar e implementar soluciones en cualquier plataforma.



**Figura 9 Visual Studio Code (Trakhtenberg, 2019)**



**Figura 10 Visual Studio (Lui, 2019)**

### **5.6.2.2 Control de versiones**

Una de las herramientas más importante en el desarrollo de software es el de control de versiones. Actualmente, la principal herramienta de control de versiones del mercado GitHub junto con varios “socios y amigos” como lo definen en su página oficial, ofrecen un paquete con las mejores


herramientas para desarrolladores en un sólo lugar, el objetivo es que la comunidad universitaria pueda aprender haciendo software.

“No hay sustituto para la experiencia práctica. Pero para la mayoría de los estudiantes, las herramientas del mundo real pueden ser prohibitivas por los costos.” - Github Education. (GitHub, 2019)

En el sitio oficial GitHub Education (GitHub, 2019) encontrarán herramientas de la industria, disponibles para el sector educativo y en cada una de ellas se muestran los distintos beneficios con sólo acceder con un correo institucional, ejemplo, usuario@unipamplona.edu.co.

En la GitKraken Figura 15, se encuentra la herramienta que se recomienda para control de versiones gráfico llamado GitKraken de Axosoft, el cual ofrece un año en versión Pro.

Por otra parte, los estudiantes que tienen proyecciones en desarrollo web, en la Figura 11, está Educative, donde podrán Obtener 6 meses gratuitos con más de 60 cursos que cubren temas bajo demanda como Desarrollo Web, Python, Java y Machine Learning. y en la Figura 12 se encuentra Bootstrap Studio con una licencia gratuita mientras está en etapa de estudiante, y también Canva, con una suscripción gratuita de 12 meses del nivel Pro. Además, podrá descargar 60 iconos premium al mes, durante 1 año en Iconscout que se encuentra en la Figura 13. En la Figura 14 está One Month, el cual ofrece una suscripción gratuita de 30 días, donde ¡Aprende HTML, CSS, JavaScript y Python en solo 30 días! como lo han denominado y para finalizar se encuentra la Figura 16 con ICONS8, por 3 meses proporcionarán recursos de diseño: iconos, ilustraciones de la interfaz de usuario, fotos.



Sube de nivel las habilidades de codificación de tendencias a tu propio ritmo con cursos interactivos basados en texto.

**Ventaja** Obtenga 6 meses gratuitos de más de 60 cursos que cubren temas bajo demanda como Desarrollo Web, Python, Java y Machine Learning.

Figura 11 Educative (GitHub, 2019)



Bootstrap Studio es una potente aplicación de escritorio para crear sitios web responsivos mediante el marco bootstrap.


**Ventaja** Una licencia gratuita para Bootstrap Studio mientras eres estudiante.



Con Canva, cualquiera puede crear gráficos y diseños de aspecto profesional. Con miles de plantillas y un editor fácil de usar.

**Ventaja** Suscripción gratuita de 12 meses del nivel Pro de Canva.

Figura 12 Bootstrap Studio y Canva (GitHub, 2019)



Iconscout es un mercado de recursos de diseño para obtener iconos de alta calidad, ilustraciones e imágenes de stock.

**Ventaja** Descarga 60 iconos premium al mes durante 1 año.

Figura 13 Iconscout (GitHub, 2019)



El legendario cliente Git multiplataforma que se integra con GitHub.

**Ventaja** Cuenta Pro gratuita durante 1 año.

Figura 15 GitKraken (GitHub, 2019)



¡Aprende HTML, CSS, JavaScript y Python en solo 30 días!

**Ventaja** Suscripción gratuita de 30 días a un mes.

Figura 14 One Month (GitHub, 2019)



Icons8 proporciona recursos de diseño: iconos, ilustraciones de la interfaz de usuario, fotos y software para clasificar sus proyectos.

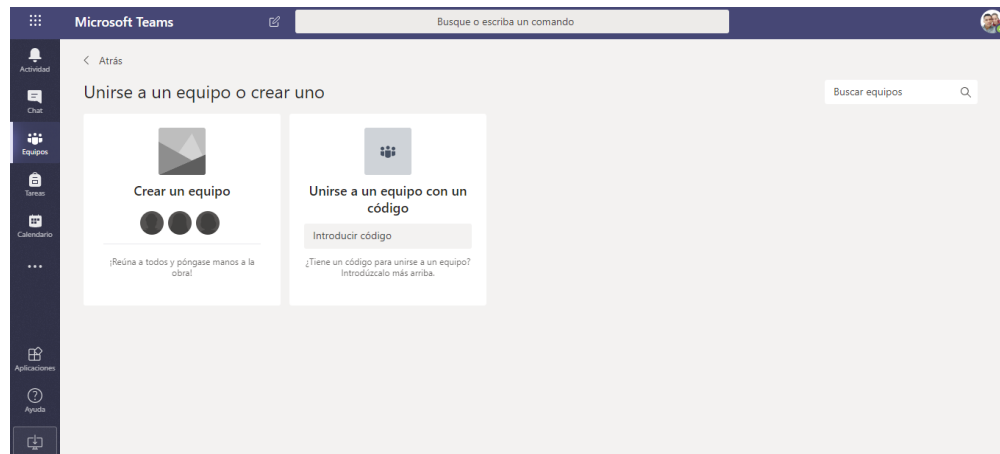
**Ventaja** Suscripción de 3 meses a All Access con iconos, fotos, ilustraciones y música.

Figura 16 Icons8 (GitHub, 2019)

### 5.6.3 Administración y Gestión de proyectos

La Universidad de Pamplona cuenta en el momento con la licencia de Microsoft Office 365, el cual la contiene 28 aplicaciones, donde todo el personal administrativo, estudiantes, y docentes, pueden acceder.

En la Figura 18 Herramienta Microsoft Planner, tomado de Figura 18 está Microsoft Planner la cual se encuentra dentro de la Suite de Office 365 y la Figura 17 está Microsoft Teams, son herramientas que ayudarán en la administración y gestión de proyectos que se desarrollan a lo largo de cada semestre en diferentes asignaturas. Microsoft Teams cuenta con una versión para dispositivos móviles para poder estar más en contacto con los grupos de trabajo.



**Figura 17 Herramienta Microsoft Teams, tomado de (Microsoft, 2019)**



**Figura 18 Herramienta Microsoft Planner, tomado de (Microsoft, 2019)**

### **5.6.3.1 Gestor colaborativo de tareas con Kanban**

“Tengo un montón de cosas por hacer y no sé por dónde empezar”, Esta es una frase que se escucha habitualmente en ámbito laboral y estudiantil. Esta acción requiere de cierta disciplina y organización –y no siempre es fácil-. En la Figura 19 se encuentra Trello, se especializa en

gestionar tareas de manera ágil, cuenta con una versión gratuita para proyectos de aula, además se puede integrar con Microsoft Teams y Slack (herramienta de colaboración) y lo mejor, existe una versión para dispositivos móviles.

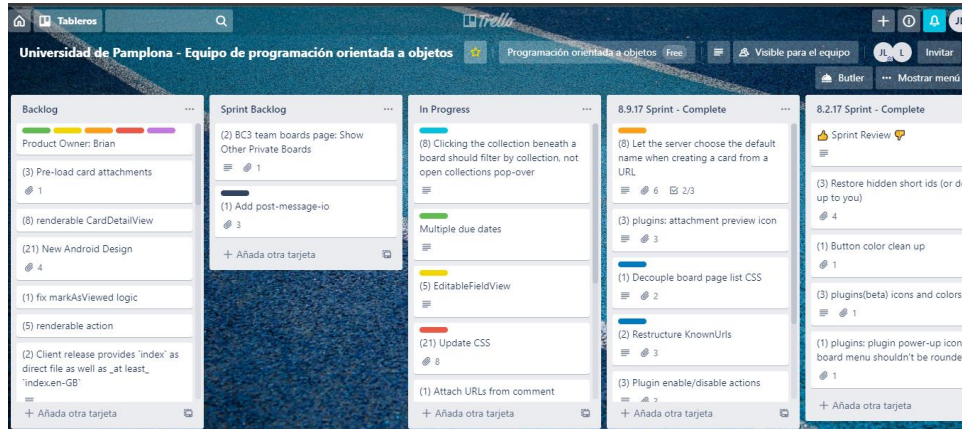


Figura 19 Pizarra de trabajo Trello, versión web, tomado de (Atlassian, 2019)

### 5.6.3.2 Herramienta de colaboración en equipo

En la Figura 20 se puede observar Slack en versión web, su función es ahorrar tiempo, ya que simplifica las tareas cotidianas como consultar el calendario de actividades, colaborar en edición de archivos, informar al equipo de trabajo sobre horarios disponibles y mucho más.

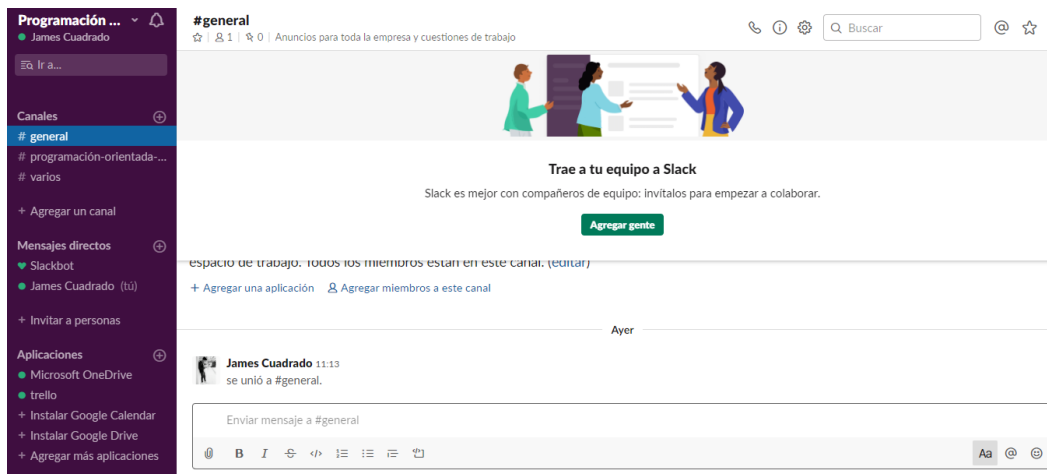


Figura 20 Herramienta Slack, versión web, tomado de (Technologies, 2019)



## **5.7 Documentar el proceso de implementación de las funcionalidades del módulo de contabilidad.**

Teniendo desarrollado e implementado el módulo, se debe documentar las funcionalidades, esto permite optimizar el mantenimiento y futuras implementaciones que se suman al aplicativo. Por medio del Framework Entity, se realizó la extracción de los summary (comentarios de funcionamiento) correspondientes a las distintas interfaces que conforma cada funcionalidad. Luego se realizan los manuales de usuario de flujo de datos en las respectivas vistas en todo el módulo.

## 6 RESULTADOS

### 6.1 Suite Neptuno

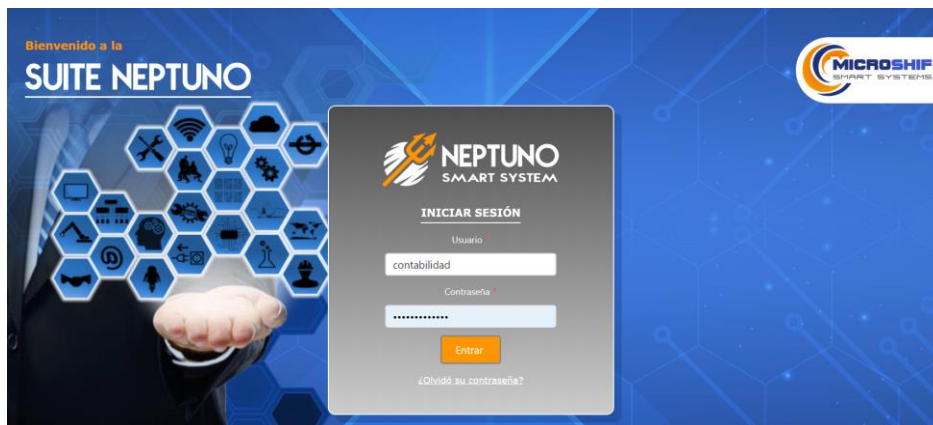


Figura 21 Vista de Login del aplicativo

En la anterior figura, se puede observar la vista principal de la Suite de aplicaciones, donde solicita credenciales de acceso.



Figura 22 Módulos de administración

En la anterior Figura, se puede apreciar después de ingresar los credenciales de acceso, se muestran las categorías y una de ellas es la de administración, dentro de la misma se encuentra el módulo General.



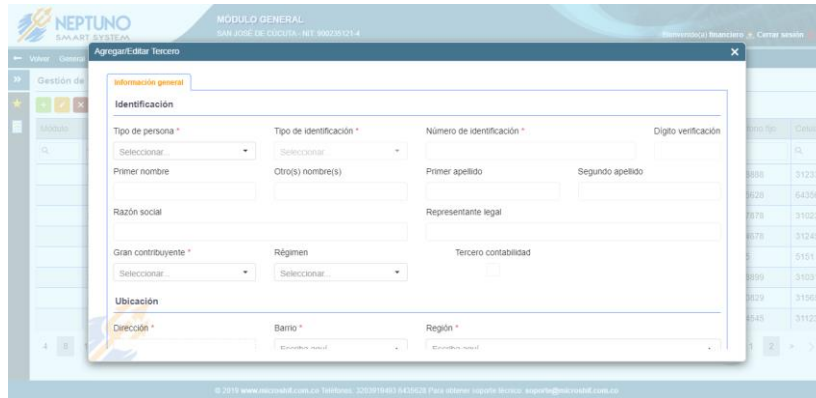
**Figura 23 Ventana de inicio del módulo General**

En la anterior figura se muestra la vista principal del módulo General. Tiene una breve descripción del módulo y las respectivas funciones que se pueden realizar. En la parte superior están las categorías que conforman el módulo.

Módulo	Estado	Tipo de identificación	Número de identificación	Nombre completo	Dirección	Teléfono
	ACTIVO	Cédula de Ciudadanía	96100117646	John Alexander Torres Cañas	Calle falsa 123	6888888
	INACTI...	Cédula de Ciudadanía	13511321	JAVIER ENRIQUE JAGUA ZAMORA	RUITOOQUE CAMPESTRE CABANA 33	6435628
	ACTIVO	Cédula de Ciudadanía	13900988	JAVIER ENRIQUE JAGUA ZAMORA	CONJUNTO RESIDENCIAL CAÑAVERAL CLUB COUNTRY	6787878
	ACTIVO	Cédula de Ciudadanía	24355677	ENRIQUE d NUÑEZ RUEDA	CONJUNTO RESIDENCIAL LOS NARANJOS	6454678
	ACTIVO	Cédula de Ciudadanía	63350559	Carmen Alicia Vasquez Pinto	calle 23#12-39	1515
	ACTIVO	Cédula de Ciudadanía	13455678	JAVIER ENRIQUE JAGUA ZAMORA	APTO 301 CONDOMINIO CAÑAVERAL CAMPESTRE	6778899
	INACTI...	Cédula de Ciudadanía	138669001	JAVIER ENRIQUE JAGUA ZAMORA	CALLE FALSA 123	6789900
	ACTIVO	Cédula de Ciudadanía	24566788	JAVIER ENRIQUE JAGUA ZAMORA	CALLE FALSA 123	6789900

**Figura 24 Vista principal de gestión de terceros**

En la anterior figura se observa la vista principal de la gestión de terceros, dicha gestión se encuentra dentro de la categoría General. En el centro de la vista principal se puede apreciar todos los registros de terceros. En la parte superior izquierda, están los botones de control que permite crear, editar, eliminar y exportar terceros.



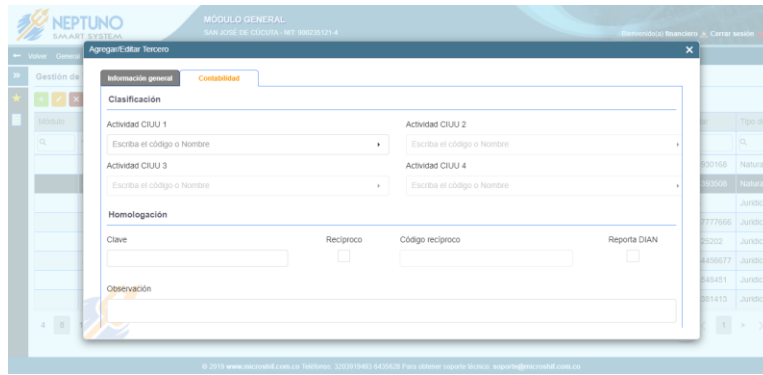
**Figura 25 Vista agregar Tercero**

En la siguiente figura se puede observar el formulario para crear un nuevo tercero.



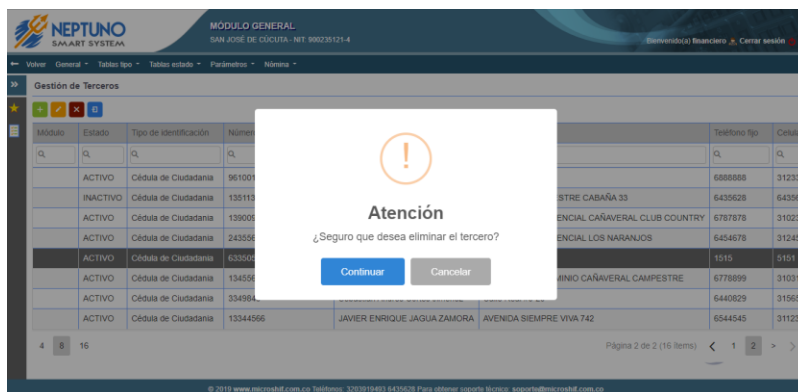
**Figura 26 Vista de editar Tercero, información general**

En la anterior figura se ilustra el formulario de editar en la pestaña de información general. Por medio de un método get se obtiene toda la información del objeto y se carga en el formulario, y así el usuario pueda actualizarlo.



**Figura 27 Vista de editar Tercero, pestaña contabilidad**

En la anterior figura se ilustra el formulario de editar en la pestaña contabilidad. Por medio de un método get se obtiene toda la información del objeto y se carga en el formulario, y así el usuario pueda actualizarlo.



**Figura 28 Vista de eliminar Tercero**

En la anterior figura se observa un mensaje de alerta para eliminar un registro dentro de la gestión. Cabe decir que los registros cambian de estado ACTIVO a INACTIVO, pero no se eliminan de la base de datos.

## 7 ANÁLISIS Y DISCUSIÓN DE RESULTADOS

Por cuestiones de tiempo y algunos inconvenientes que se presentaron a lo largo del desarrollo del módulo, se presentará la **¡Error! No se encuentra el origen de la referencia.** con todas las fases de desarrollo, referencias y flujos entre capas de la arquitectura limpia. La gestión de terceros hace parte del módulo General de Suite Neptuno, está vinculado directamente con la funcionalidad de la gestión Plan contable del módulo de Contabilidad. Se implementa desde un enfoque replicable y escalable, de tal manera se desarrolla aplicando las siguientes fases:

- Wireframing.
- Estructuración.

### 7.1 Wireframing

Antes de iniciar el desarrollo de la gestión, se realiza el proceso de wireframing. El director de desarrollo y el diseñador realizan el prototipado de las vistas y casos de usos con ajustes de usabilidad correspondientes, esto enfocado a que sea un proceso rápido y ágil para el usuario final.

Como se observa en la Figura 29

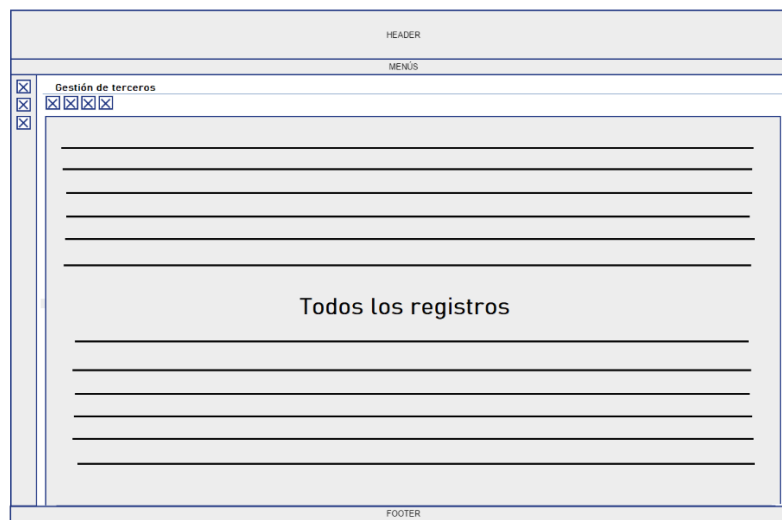


Figura 29 Wireframe de gestión de terceros

## 7.2 Estructuración

Para dar cumplimiento a la arquitectura limpia en la que se encuentra enmarcado el proyecto, se implementa la siguiente estructuración de carpetas, clases y archivos, todo esto, enfocado siempre a que el componente sea replicable en las distintas funcionalidades del proyecto.

Como se observa en la Figura 30 Estructura de carpetas y archivos basado en la Arquitectura Limpia, se enmarca en color rojo, se pueden identificar las tres capas principales propuestas por la Clean Architecture, se tiene una capa de vista (web para este caso), una capa núcleo (core) y una capa de infraestructura.

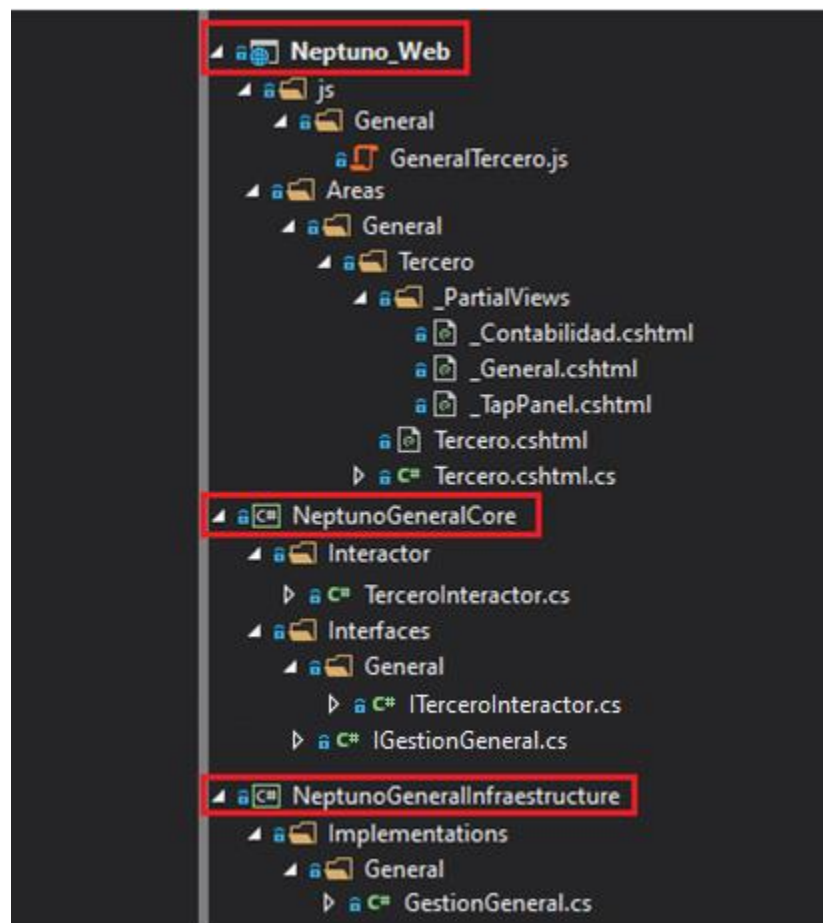


Figura 30 Estructura de carpetas y archivos basado en la Arquitectura Limpia de Terceros

De la capa web se puede observar que se implementa un archivo independiente y específico para lo que el componente requiere llamado “GeneralTercero.js”, por otro lado, dentro del área de General se crea una carpeta llamada “Tercero” esto con el objetivo de plasmar que el componente es compartido por distintas funcionalidades y al estar en una carpeta independiente, éste componente gozará de autonomía para funcionar en el menú donde se ubique. Además se crea un componente de tipo “Razor Page” con el objetivo de eliminar la replicación de los llamados a métodos HTTP en cada funcionalidad.

En este mismo orden de ideas, se crea dentro del core de General, un interactivo y las interfaces tanto de la implementación como del Core específico para la funcionalidad de agregar tercero.

Por último, en la capa de infraestructura se crea una implementación de gestión general. Teniendo definidas las capas, las clases y los archivos, se procede a crear las referencias entre capas lo cual corresponde al último paso de la implementación de la arquitectura limpia.

En la siguiente figura se describen las referencias o conexiones.

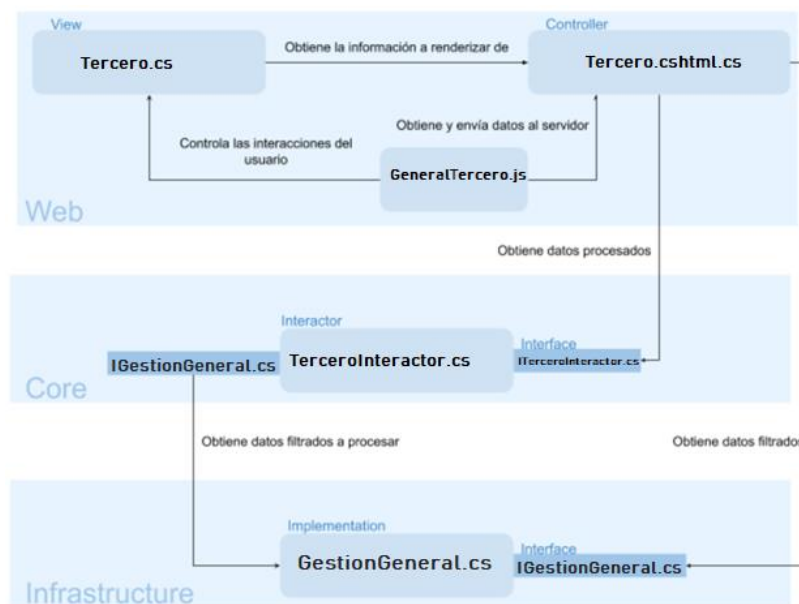


Figura 31 Referencias y flujos entre capas de la Arquitectura Limpia



## 8 CONCLUSIONES

La Arquitectura Limpia (Clean Architecture) garantiza la modularidad del aplicativo y la independencia de la capa de presentación y de la capa de datos, sumado a que con la implementación de los principios SOLID se pueden crear sistemas de información con estándares de integración, seguridad, aptos y válidos por empresas de auditoría.

Al finalizar el periodo de práctica en la empresa Microshif se concluye que la herramienta de gestión de proyectos JIRA es un medio que permite gestionar y llevar control sobre el trabajo realizado por los equipos de trabajo, además que le da la capacidad al Scrum Master de obtener métricas reales sobre la velocidad que se está llevando en el desarrollo del proyecto y tomar los correctivos y ajustes necesarios para cumplir con las metas en los tiempos definidos; esta situación se vio presente en varios de los sprint llevados a cabo por lo que fue necesario llevar jornadas adicionales de trabajo y poder cumplir con lo propuesto a los stakeholder.

Por último, se resalta que si bien al terminar ésta práctica el módulo de Contabilidad no se logra terminar, pues requiere de más tiempo del estimado, también la curvatura de aprendizaje de la arquitectura y algunos eventos que se presentaron a lo largo del desarrollo del módulo donde requería de mayor prioridad.

## 9 RECOMENDACIONES Y TRABAJOS FUTUROS

Se recomienda implementar un sistema de administración de bases de datos de libre distribución (de las cuales existen muchas en la actualidad) en los laboratorios especializados del programa de ingeniería de sistemas. Así mismo, utilizar DBeaver, como herramienta de base de datos universal gratuita, de código abierto para los estudiantes y docentes cuando se necesite gestionar la información en las bases de datos.

Se aconseja al director del programa y docentes del programa, indagar en la plataforma de GitHub (GitHub, 2019), y así concretar el licenciamiento de un año de la herramienta de control de versiones gráfico, GitKraken Pro por ser institución educativa.

Se invita a los docentes y estudiantes utilizar la herramienta de desarrollo, Visual Studio (Microsoft, Visual Studio, 2019) de código abierto, y sacar mayor provecho a las herramientas adquiridas por la institución, es decir, la suite Office 365, enlazado con Slack (Technologies, 2019) y Trello (Atlassian, 2019), en las aulas de clases.

Para realizar prototipos en cualquier plataforma, se recomienda Pencil (Evolution Solutions Co, 2019). Una herramienta de creación de prototipos GUI de código abierto que está disponible para todas las plataformas.

En la siguiente Figura 32, se muestra un esquema con herramientas de libre distribución. Sería una posible alternativa de implementación en el programa de ingeniería de sistemas.

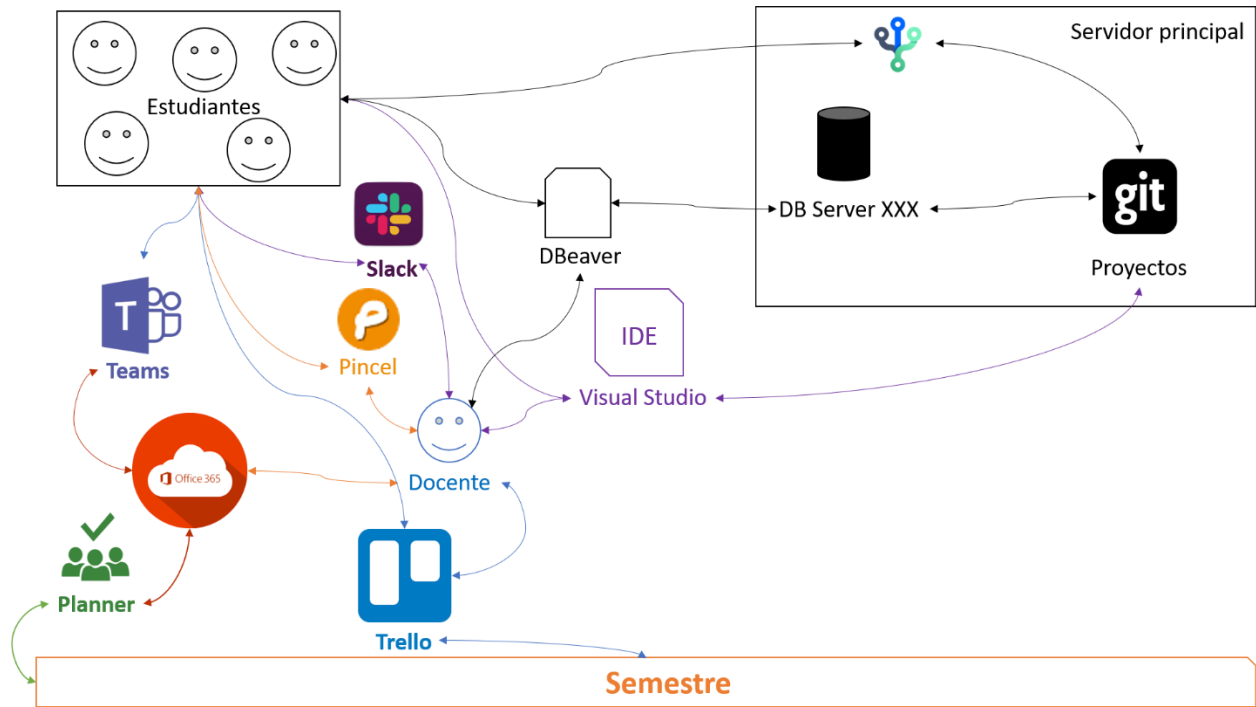


Figura 32 Esquema de herramientas – infraestructura - usuarios

## 10 ANEXOS

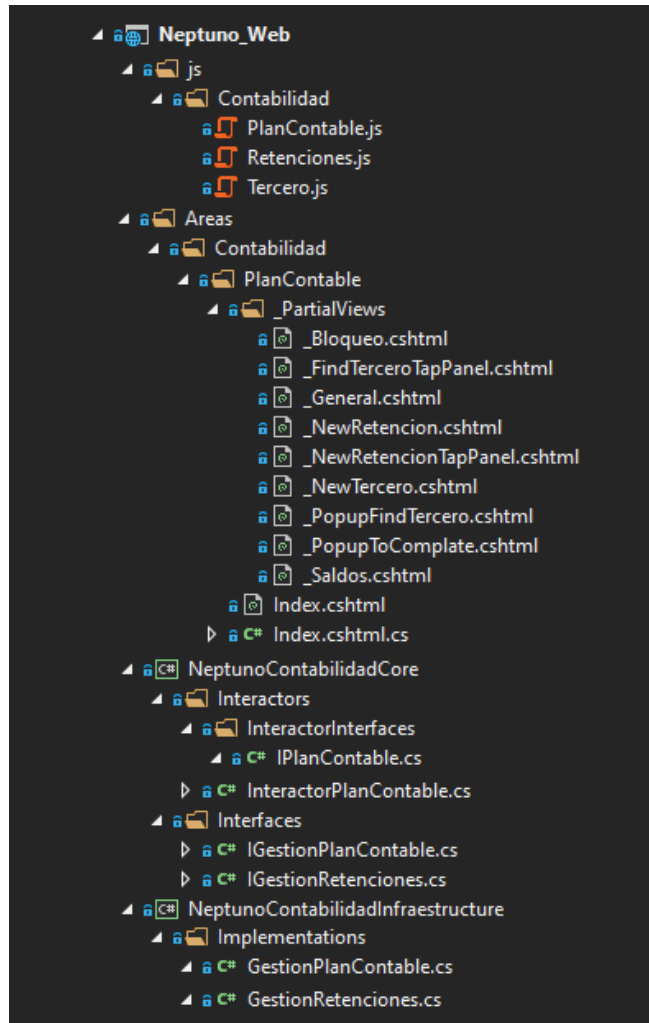


Figura 33 Estructura de carpetas y archivos basado en la Arquitectura Limpia de Plan contable



Figura 34 Módulos de contabilidad y financiero



Figura 35 Ventana de inicio del módulo contabilidad

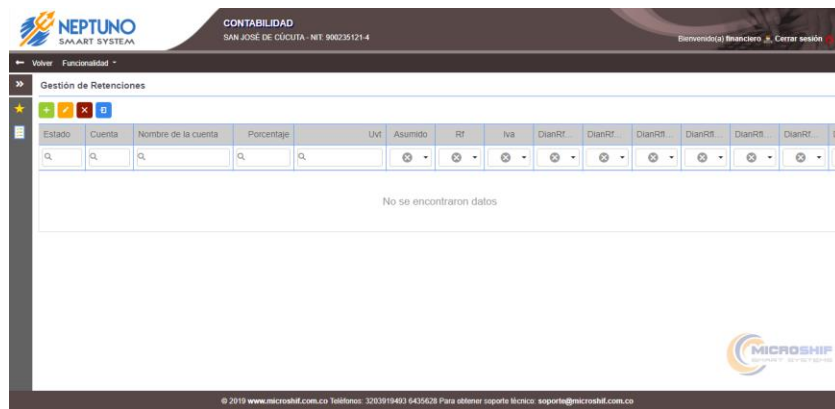


Figura 36 Gestión de Retenciones

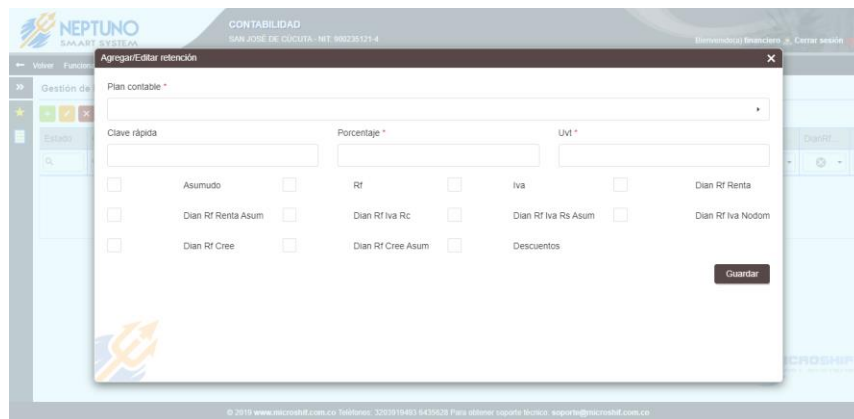
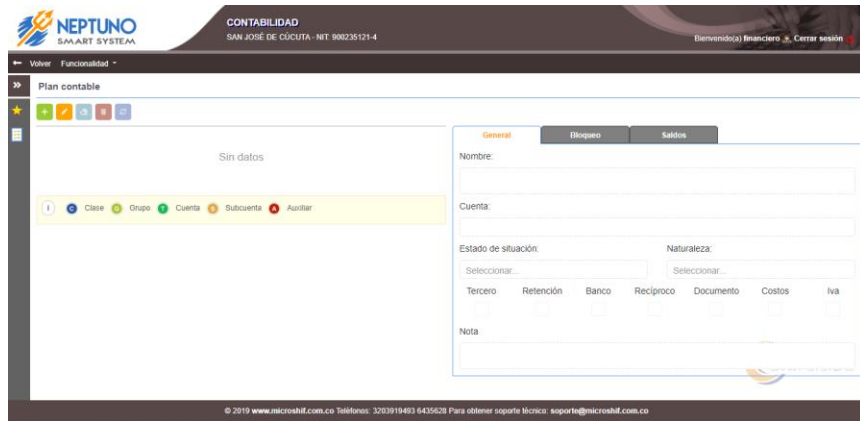
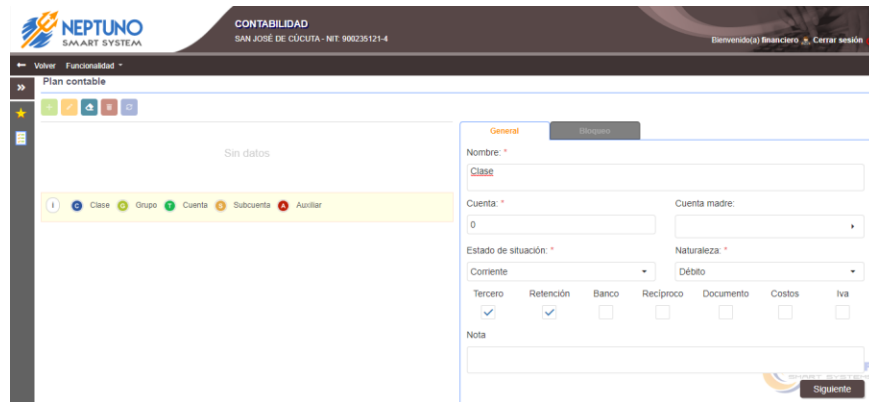


Figura 37 Vista agregar retención



**Figura 38 Gestión de Plan contable**



**Figura 39 Vista agregar un plan contable**

## 11 REFERENCIAS

- Servicio fitosanitario del estado. (15 de 11 de 2019). *Ministerio de agricultura y ganadería Costa Rica*. Obtenido de <http://sistemas.mag.go.cr/SoporteTecnico/Est%C3%A1ndares%20de%20Sistemas.pdf>
- Alaimo, M. (2013). *Proyectos Ágiles con Scrum*. Kleer.
- Alaimo, M. (2019). *Introducción a la Agilidad y Scrum*. Buenos Aires: Kleer.
- Atlassian. (20 de 10 de 2019). *Trello*. Obtenido de <https://trello.com/>
- Beedle, K. B. (01 de 11 de 2001). *Manifiesto for Agile Software Development*. Obtenido de <https://agilemanifesto.org>
- Community, D. (11 de 11 de 2019). *DBeaver*. Obtenido de <https://dbeaver.io/>
- DevExpress. (25 de 07 de 2019). *DevExpress, el mejor aliado en el desarrollo*. Obtenido de <https://dawconsblog.blogspot.com/2014/04/devexpress-el-mejor-aliado-en-el.html>
- Evolution Solutions Co. (11 de 12 de 2019). *PENCIL PROJECT*. Obtenido de <https://pencil.evolus.vn/>
- Fontela, A. (25 de 07 de 2019). *¿Que es Bootstrap?* Obtenido de <https://raiolanetworks.es/blog/que-es-bootstrap/>
- Francia, J. (25 de 07 de 2019). *¿Qué es Scrum?* Obtenido de <https://www.scrum.org/resources/blog/que-es-scrum>
- GitHub. (20 de 11 de 2019). *GitHub Education*. Obtenido de <https://education.github.com/pack>
- L. Martinez, J. C. (2015). *Getting Started with Entity Framework 6 Code First using MVC 5 | Microsoft Docs Getting Started with Entity Framework 6 Code First using MVC 5*.
- Lui, S. (20 de 11 de 2019). *Lifehacker*. Obtenido de Microsoft Adds Java Debugging And Language Support For Android To Visual Studio Update: <https://www.lifehacker.com.au/2015/11/microsoft-adds-java-debugging-and-language-support-for-android-to-visual-studio-update/>
- Martin, R. (01 de 11 de 2019). *The Clean Architecture*. Obtenido de The Clean Code Blog: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-cleanarchitecture.html>
- Microsoft. (19 de 07 de 2015). *Introduction to LINQ Queries (C#)*. Obtenido de <https://docs.microsoft.com/en-us/dotnet/csharp/programmingguide/concepts/linq/introduction-to-linq-queries>
- Microsoft. (19 de 09 de 2019). Obtenido de What's new in C# 8.0: <https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-8>
- Microsoft. (25 de 07 de 2019). *.NET Core y código abierto*. Obtenido de <https://docs.microsoft.com/es-es/dotnet/framework/get-started/net-core-and-open-source>
- Microsoft. (25 de 07 de 2019). *Arquitecturas de aplicaciones web comunes*. Obtenido de <https://docs.microsoft.com/es-es/dotnet/standard/modern-web-apps-azure-architecture/common-web-application-architectures>

Microsoft. (25 de 07 de 2019). *Documentación de Entity Framework*. Obtenido de <https://docs.microsoft.com/es-mx/ef/>

Microsoft. (16 de 10 de 2019). *Entity Framework overview*. Obtenido de <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ef/overview>

Microsoft. (13 de 11 de 2019). *Office 365*. Obtenido de <https://www.office365.com/>

Microsoft. (25 de 07 de 2019). *Sobre .NET Core*. Obtenido de <https://docs.microsoft.com/es-es/dotnet/core/about>

Microsoft. (04 de 04 de 2019). *Un paseo por el lenguaje C#*. Obtenido de <https://docs.microsoft.com/es-es/dotnet/csharp/tour-of-csharp>

Microsoft. (11 de 12 de 2019). *Visual Studio*. Obtenido de Visual Studio: <https://visualstudio.microsoft.com/es/>

Nicolas. (23 de 11 de 2019). *Mi bitácora informática* . Obtenido de <https://ormeno-nicolas.blogspot.com/2012/02/que-es-un-ide.html>

Nonaka, H. T. (1986). The New Product Development. *Harvard Business Review*.

Ruben. (25 de 07 de 2019). *Principios de una arquitectura limpia: mantenible y testeable*. Obtenido de <https://www.genbeta.com/desarrollo/principios-de-una-arquitectura-limpia-mantenible-y-testeable>

RUBIRA, J. (25 de 07 de 2019). *Solid, cinco principios básicos de diseño de clases*. Obtenido de <https://www.genbeta.com/desarrollo/solid-cinco-principios-basicos-de-diseno-de-clases>

Sitio. (25 de 07 de 2019). *SQL Server: ¿Para qué sirve y cuál es la versión que necesito?* Obtenido de <https://www.itsitio.com/ar/sql-server-para-que-sirve-y-cual-es-la-version-que-necesito/>

Technologies, S. (28 de 10 de 2019). *Slack*. Obtenido de Sea cual sea tu trabajo, puedes hacerlo en Slack: <https://slack.com/intl/es-co/>

Trakhtenberg, A. (12 de 12 de 2019). *LaunchDarkly*. Obtenido de <https://launchdarkly.com/blog/launched-version-2-of-the-launchdarkly-visual-studio-code-extension/>

Wikipedia. (25 de 07 de 2019). *Git*. Obtenido de <https://es.wikipedia.org/wiki/Git>

Wikipedia. (25 de 07 de 2019). *Microsoft SQL Server*. Obtenido de [https://es.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://es.wikipedia.org/wiki/Microsoft_SQL_Server)