

UNIVERSIDAD TÉCNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TEMA:

**DESARROLLO DEL SISTEMA BACKEND WEB PARA EL PROCESO DE
PRECONTRATACIÓN Y CONTRATACIÓN DE LA EMPRESA MASTERCUBOX
S.A. CON EL FRAMEWORK SPRING.**

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO
EN SISTEMAS COMPUTACIONALES

AUTOR:

Jefferson Santiago Ortega Checa

DIRECTOR:

Msc. Xavier Mauricio Rea Peñafiel

Ibarra, 2019

UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

**AUTORIZACIÓN Y USO DE PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD
TÉCNICA DEL NORTE**

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DEL CONTACTO	
CÉDULA DE IDENTIDAD:	100351668-7
APELLIDOS Y NOMBRES:	ORTEGA CHECA JEFFERSON SANTIAGO
DIRECCIÓN:	IBARRA – SALVADOR DALI S/N Y EDUARDO VILLACÍS
EMAIL:	jsortegac@utn.edu.ec
TELÉFONO MÓVIL:	0960552729

DATOS DE LA OBRA	
TÍTULO:	DESARROLLO DEL SISTEMA BACKEND WEB PARA EL PROCESO DE PRECONTRATACIÓN Y CONTRATACIÓN DE LA EMPRESA MASTERCUBOX S.A. CON EL FRAMEWORK SPRING.

AUTOR (ES):	ORTEGA CHECA JEFFERSON SANTIAGO
FECHA:	25-09-2019
PROGRAMA:	PREGRADO
TÍTULO POR EL QUE OPTA:	INGENIERO EN SISTEMAS COMPUTACIONALES
DIRECTOR:	MSc. XAVIER MAURICIO REA PEÑAFIEL

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de esta y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 25 días del mes de septiembre del 2019



Jefferson Santiago Ortega Checa

1003516687



UNIVERSIDAD TÉCNICA EL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

Ibarra, 25 de septiembre del 2019

CERTIFICACIÓN DIRECTOR

El Sr. Jefferson Santiago Ortega Checa, portador de la cédula de identidad número: 100351668-7, ha trabajado en el desarrollo del proyecto de grado **“DESARROLLO DEL SISTEMA BACKEND WEB PARA EL PROCESO DE PRECONTRATACIÓN Y CONTRATACIÓN DE LA EMPRESA MASTERCUBOX S.A. CON EL FRAMEWORK SPRING.”**, previo a la obtención del Título de Ingeniero en Sistemas Computacionales, realizando con interés profesional y responsabilidad, que certifico en honor a la verdad.

.....
MSc. MAURICIO REA
DIRECTOR DE TRABAJO DE GRADO



Febrero 2019

ACTA DE ENTREGA RECEPCIÓN

TEMA: Módulo de precontratación y contratación.

En las instalaciones de la empresa MASTERCUBOX S.A. de la Ciudad de Ibarra, el día 4 de febrero del 2019, la Carrera de Ingeniería de Sistemas Computacionales de la Universidad Técnica del Norte por medio del trabajo de titulación "DESARROLLO DE SISTEMA BACKEND WEB PARA EL PROCESO DE PRECONTRATACIÓN Y CONTRATACIÓN DE LA EMPRESA MASTERCUBOX S.A. CON EL FRAMEWORK SPRING" hace la entrega del sistema web "Precontratación y contratación", al Ing. Miguel Iturralde, Gerente General de la empresa MASTERCUBOX S.A.

El desarrollo del sistema informático lo realizó la Sr. **Jefferson Santiago Ortega Checa** con CI: **1003516687** bajo la tutoría del Msc. Mauricio Rea, conforme a los requisitos solicitados por el Ing. Pedro Román Jefe del Departamento Agrícola de la empresa.

A continuación, se detalla los productos entregados:

- Proyecto de desarrollo de software (Código Fuente).
 - Registros de predios, proveedores y contratos.
 - Gestión de predios y contratos.
 - Reportes.
- Pruebas Funcionales y Aceptación del sistema desarrollado.
- Manuales de Usuario.
- Manuales técnicos.
- Capacitación al personal encargado.

Atentamente, ENTREGA CONFORME


Sr. Jefferson Ortega.
Tesista
UNIVERSIDAD TÉCNICA DEL NORTE


Ing. Pedro Granda, Msc.
Coordinador Carrera Ingeniería
en Sistemas Computacionales



Ing. Mauricio Rea, Msc.
Docente tutor
UNIVERSIDAD TÉCNICA DEL NORTE

RECIBE CONFORME,


Ing. Miguel Iturralde
Gerente General MASTERCUBOX S.A.



Dedicatoria

El presente trabajo de titulación lo dedico principalmente a Dios, por darme fuerza para continuar en los momentos más difíciles en el proceso de obtener uno de los anhelos más deseados.

A mis padres Santos y Virginia, por su amor, trabajo, sacrificio y confianza que me han brindado en todos estos años de estudio, gracias a ustedes he logrado llegar hasta aquí y convertirme en un profesional. Será siempre un orgullo y privilegio ser su hijo, son los mejores padres.

Finalmente quiero dedicar esta tesis a mi persona especial quien siempre me apoyo y comprendió en este proceso Te Amo Danny, y a todos mis amigos, por apoyarme en todo, por extender su mano en momentos difíciles, de verdad mil gracias.

Jefferson Ortega C.

Agradecimientos

Agradezco a todos los docentes de la Carrera de Ingeniería en Sistemas Computacionales de la Universidad Técnica del Norte, por haber compartido sus conocimientos y experiencias a lo largo de la preparación en nuestra carrera, de manera especial, al Ing. Mauricio Rea director de mi trabajo de titulación quien me ha guiado con su experiencia y responsabilidad como docente, a la Ing. Daysi Imbaquingo que gracias a su apoyo y consejos hicieron que pueda crecer día a día como profesional y ganar experiencia en el campo laboral.

También agradezco a la empresa Mastercubox S.A. por confiar en nosotros y abrir las puertas a todo el equipo de desarrollo y permitirnos implementar nuestro sistema.

Jefferson Ortega C.

Tabla de Contenido

Dedicatoria	VI
Agradecimientos	VII
Tabla de Contenido	VIII
Índice de Figuras	XI
Índice de Cuadros.....	XII
Resumen	XIII
Abstract	XIV
INTRODUCCIÓN.....	1
Antecedentes	1
Situación Actual	2
Prospectiva	3
Planteamiento del Problema	4
Objetivos	4
Objetivo General.....	4
Objetivos Específicos	4
Alcance	5
Justificación.....	5
Impacto social.....	5
Impacto Tecnológico	6
Contexto.....	6
CAPÍTULO 1	7
Marco Teórico	7
1.1. Importancia del software en empresas agrícolas	7
1.2. Los backend's en los ERP	7
1.2.1. Backend	7
1.2.2. ERP.....	9

1.2.2.1. Ventajas de un sistema ERP	10
1.3. Spring Framework	10
1.4. Base de Datos PostgreSQL	11
1.5. Metodología Scrum	12
1.5.1. ¿Por qué usar Scrum?	13
CAPÍTULO 2.....	14
Herramientas Utilizadas	14
2.1. Spring Boot	14
2.1.1. Spring Boot.....	14
2.1.2. Dependencias de Spring Boot.....	14
2.1.3. Acceso a datos con Spring Boot	15
2.1.3.1. Patrón DAO	15
2.2. Modelos, vistas y controladores	16
2.2.1. Patrón MVC.....	16
2.3. Base de datos	17
2.4. Servidor de aplicaciones	17
2.4.1. Apache Tomcat	18
CAPÍTULO 3.....	19
Desarrollo.....	19
3.1. Implementación módulo de precontratación y contratación	19
3.2. Definición de Requisitos.....	19
3.3. Definición del Product Backlog.....	22
3.4. Definición de los Roles del Proyecto	23
3.5. Desarrollo del Aplicativo.....	24
3.5.1. Desarrollo de los Sprints	24
3.5.1.1. Sprint 0.....	24
3.5.1.2. Sprint 1	27
3.5.1.3. Sprint 2.....	32

3.5.1.4. Sprint 3.....	36
3.5.2. Productos Entregados.....	42
CONCLUSIONES.....	43
RECOMENDACIONES.....	44
REFERENCIAS.....	45

Índice de Figuras

Fig. 1	Árbol de problemas.....	4
Fig. 2	Arquitectura ERP.....	9
Fig. 3	Dependencias de serie de Spring Boot.	15
Fig. 4	Patrón DAO	15
Fig. 5	Patrón MVC	16
Fig. 6	Arquitectura J2EE.....	18
Fig. 7	Arquitectura del módulo de Precontratación y contratación.....	26
Fig. 8	Herramientas para el desarrollo del sistema I-FOODS.....	27
Fig. 9	Diseño de Registro de Proveedores.	29
Fig. 10	Listado de Registro de Proveedores.....	30
Fig. 11	Diseño de Registro de Predios.	30
Fig. 12	Listado de Registro de Predios.....	31
Fig. 13	Controller de la entidad Predio	31
Fig. 14	Controller de la entidad Proveedor	32
Fig. 15	Diseño de Registro de Contratos.....	34
Fig. 16	Listado de Registro de Contratos.	35
Fig. 17	Controller de la entidad Precontratación.....	35
Fig. 18	Prototipo de Reporte de datos.	38
Fig. 19	Reporte de datos Proveedores.....	39
Fig. 20	Reporte de datos Predios.	39
Fig. 21	Reporte de datos Contrato.....	40
Fig. 22	Instalación del servidor de aplicaciones.	40

Índice de Cuadros

Tabla 1. Lenguajes de programación y SGBD para el desarrollo de backend	8
Tabla 2. Evolución de versiones Spring Framework.....	11
Tabla 3. Historia de Usuario Nro. 1.....	19
Tabla 4. Historia de Usuario Nro. 2.....	19
Tabla 5. Historia de Usuario Nro. 3.....	20
Tabla 6. Historia de Usuario Nro. 4.....	21
Tabla 7. Historia de Usuario Nro. 5.....	21
Tabla 8. Historia de Usuario Nro. 6.....	22
Tabla 9. Lista del Producto	23
Tabla 10. Roles del Proyecto.....	23
Tabla 11. Desarrollo de los Sprints.....	24
Tabla 12. Sprint 0 Backlog.....	24
Tabla 13. Planificación del Sprint 0.....	25
Tabla 14. Seguimiento del Sprint 0.....	25
Tabla 15. Sprint 1 Backlog.....	27
Tabla 16. Planificación del Sprint 1.....	28
Tabla 17. Seguimiento del Sprint 1.....	29
Tabla 18. Plan de mejoras del Sprint 1	32
Tabla 19. Sprint 2 Backlog.....	33
Tabla 20. Planificación del Sprint 2.....	33
Tabla 21. Seguimiento del Sprint 2.....	34
Tabla 22. Plan de mejoras del Sprint 2.....	36
Tabla 23. Sprint 3 Backlog.....	36
Tabla 24. Planificación del Sprint 3.....	36
Tabla 25. Seguimiento del Sprint 3.....	37
Tabla 26. Pruebas de aceptación	41
Tabla 27. Plan de mejoras del Sprint 3.....	42
Tabla 28. Productos entregados.....	42

Resumen

En este trabajo de titulación nombrado “DESARROLLO DEL SISTEMA BACKEND WEB PARA EL PROCESO DE PRECONTRATACIÓN Y CONTRATACIÓN DE LA EMPRESA MASTERCUBOX S.A. CON EL FRAMEWORK SPRING.”, se encuentra conformado por tres capítulos.

En la parte de la Introducción se define el problema, objetivo general y objetivos específicos. También se incluye el alcance que va a tener el proyecto realizado, así como la justificación de la realización de este.

En el primer capítulo, se realizó un marco teórico de gestión y tecnológico y el proceso a automatizar.

En el segundo capítulo, se realizó una recopilación de las herramientas que se utilizarán para el desarrollo del sistema

En el tercer capítulo, se detalla el desarrolló la aplicación web para el módulo de precontratación y contratación de la empresa MASTERCUBOX.S.A. El nombre escogido para esta aplicación es “I-FOODS Mastercubox”, esta aplicación se realizó utilizando Angular para el front-end y Spring boot para el backend, para la comunicación entre estos se utilizaron microservicios REST implementando la Metodología Ágil Scrum.

Abstract

In this degree work called ***“DEVELOPMENT OF THE SYSTEM WEB BACKEND FOR THE PROCESS OF PRE-ENGAGEMENT PROCESS AND RECRUITMENT OF THE COMPANY MASTERCUBOX S.A WITH THE FRAMEWORK SPRING”*** it consists of three chapters.

In the introduction part, it defines the problem, general objective and specific objectives. Also, there is included the scope that the project will have realized, as well as the justification for the realization of this.

In the first chapter, a theoretical basis of technological management and the process to be automated was carried out.

In the second chapter, a compilation of the tools to be used for system development was realized.

In the third chapter, there is detailed the development of the web application for the module of pre-recruitment and recruitment of the company MASTERCUBOX S.A. The name chosen for this application is “I-FOODS Mastercubox”, this application was realized using angular for the front -end and Spring boot for the back-end, for the communication between these REST microservices were used implementing the Agile Scrum methodology.

INTRODUCCIÓN

Antecedentes

“La evolución de las plataformas y herramientas para la creación de sitios Web ha expandido las posibilidades del desarrollo, especializando sus partes y dividiendo las tareas principales del programador en dos grandes áreas: La interfaz visible para los usuarios denominada front-end y las configuraciones backend de un proyecto Web. Un desarrollador backend es un profesional fundamental en los proyectos Web, responsable de la programación y todos sus componentes, coordinando páginas, formularios, funcionalidades, bases de datos y servidores Web.” (CampusMVP, 2015)

En sus inicios las aplicaciones Web fueron muy básicas y aún no existía el término ERP como sistema Web de tal forma que únicamente fueron desarrolladas con un solo lenguaje de programación y hasta podemos decir que fueron muy limitadas en sus capacidades.

“Con la evolución de las plataformas y herramientas se fue expandiendo tanto su eficiencia como calidad de desempeño, el monitoreo de los recursos naturales generalmente se concibe a través de la utilización de herramientas sofisticadas y costosas como son los Sistemas de Información Geográfica y los sistemas de percepción remota. Por otra parte, el conocimiento detallado que tienen de los recursos naturales los principales usuarios de éstos, que muchas veces son campesinos o indígenas, no se valora en su justa dimensión. En la actualidad se puede encontrar aplicaciones Web como SIMA (Sistema Integrado de Monitoreo Agrícola) que se encarga de recolección, análisis y monitoreo de campos agrícolas.” (Torrealba & Laforge, 1998)

“En la Empresa Mastercubox S.A. se viene implementando un proyecto de innovación tecnológica de elaboración de alimentos para animales en base a la producción de alfalfa. Es una empresa privada que se enfoca en el área agrícola, el principal producto que realizan son cubos o croquetas de alfalfa. Para la implementación de este proyecto innovador Mastercubox S.A. firmo un contrato con el Ministerio Coordinador de la Producción Empleo y Competitividad, el mismo que durará 15 años.” (Yachay, 2015)

Situación Actual

Las empresas en la actualidad se apoyan cada vez más en la tecnología para la mejora de sus procesos y productos. Por lo que la adopción de un sistema Web está dejando de ser una alternativa para pasar a ser un requerimiento, y toda herramienta que ayude a ahorrar tiempo, dinero, recursos y mejorar el servicio a clientes permite lograr una generación de valor para la empresa.

En este contexto las empresas emprendedoras, se apoyan cada vez más en la tecnológica. Por ende, en la empresa Mastercubox surgió la necesidad de implementar un sistema Web o ERP para evitar el trabajo con el método tradicional en el que se encuentran varias empresas y cambiar por un nuevo enfoque que aproveche los beneficios que las nuevas tecnologías ofrecen.

Hoy en día, muchas empresas buscan los beneficios de un sistema ERP, el término ERP se refiere a “sistema de planificación de recursos empresariales”; las principales ventajas de un sistema ERP es que automatizan los procesos de las empresas además de lograr una integración de distintas plataformas en un solo sistema y teniendo disponible la información de la empresa en el momento que se requiera.

En la Empresa Mastercubox S.A. actualmente se vienen realizando procesos manuales los mismos que dificultan llevar un buen control y registro de la información relacionada con esta empresa, el proceso de precontratación y contratación es de vital importancia para el análisis de la información que llega a la empresa, trabajo inicial crucial para obtener un producto de calidad.

Para este proyecto se selecciona la plataforma tecnológica Spring, que viene dando pasos concretos con proyectos de gran importancia. entre los más representativos se encuentran Spring Cloud, Spring para Android, Spring Security y Spring Boot.

Prospectiva

Mediante el análisis del lenguaje Java y el Framework Spring se desarrollará sistemas Web de mejor calidad y eficiencia para backend.

Aprovechando la facilidad de integración de plataformas en sistemas ERP la Empresa Mastercubox S.A. permitirá la creación de un sistema Web el mismo que se espera como resultado final que el proceso de precontratación y contratación estén automatizados por completo con una arquitectura muy robusta y herramientas de calidad que faciliten el trabajo de la empresa y aseguren el desempeño del sistema ERP y la integridad de la información de la empresa almacenada en el mismo.

La innovación y los nuevos frameworks seguirán apareciendo en el mercado y una parte importante de los desarrolladores es que se seguirá teniendo la necesidad de integrar estos productos dentro de su soporte y es aquí donde Spring tiene mucho que decir porque es una plataforma que se integra con facilidad.

Con el sistema en funcionamiento el personal que trabaja en la empresa Mastercubox llevará un mejor control de los clientes que llegan a la empresa para obtener los servicios de esta, la empresa también optimizará los resultados de atención a clientes informándoles de las observaciones que la empresa pudo detectar.

Planteamiento del Problema



Fig. 1 Árbol de problemas.
Fuente: Propia

¿Por qué los programadores deben elegir Spring Framework y el Lenguaje Java para el proceso de desarrollo de un ERP o Sistema Web?

Objetivos

Objetivo General

Desarrollar el sistema Backend Web para el proceso de precontratación y contratación de la Empresa Mastercubox S.A. con el Framework Spring

Objetivos Específicos

- Estudiar las bases teóricas necesarias y recolectar información a través de una investigación documental que sustente conceptual y técnicamente la información para ser aplicada en el desarrollo de sistemas backend Web con lenguaje Java y Spring Framework.
- Estudiar el Framework Spring en el desarrollo de aplicaciones Backend Web y validar la información en cada Spring realizado con la metodología ágil Scrum.
- Desarrollar un sistema Web para la Precontratación y Contratación de clientes, sistematizando el proceso de recopilación de información, estudio del suelo para su aprobación y una posterior contratación para producción agrícola, bajo la metodología ágil Scrum.

Alcance

Para el objetivo de este proyecto en el desarrollo del Sistema Precontratación y Contratación de clientes para la empresa Mastercubox S.A. se utilizará la metodología SCRUM y tendrá los siguientes módulos:

- Módulo de Registro de Proveedores y Predios. - Este módulo permitirá tener un registro detallado de los agricultores quienes mediante una solicitud a la empresa serán analizados por el departamento Agrícola, Financiero y Gerencia General, también, permitirá llevar un registro de los supervisores del departamento agrícola que son los encargados en recopilar la información de la propiedad o terreno para su análisis.
- Módulo de Administración de Información (Visita Técnica). - Este módulo será el encargado de almacenar, actualizar, leer y eliminar la información proporcionada por la visita técnica del supervisor encargado a la propiedad o terreno del cliente y presentar a los departamentos Agrícola, Financiero y Gerencia General para su respectivo análisis.
- Módulo de Contrataciones. - Una vez aprobada la contratación del agricultor por los diferentes departamentos avanzan a una etapa de contratación que es la encargada de la contratación y también de llevar un registro de los contratos de los agricultores con toda la información detallada para su seguimiento.

Justificación

Impacto social

Utilizando las bases teóricas recolectadas durante el proceso de investigación se desarrollará un sistema Web de recopilación de información para la Precontratación de clientes, contratación de clientes luego de sus análisis respectivos por los departamentos encargados y registro de clientes para las visitas técnicas de la Empresa Mastercubox S.A. que actualmente no disponen de una automatización en sus procesos de Precontratación, Contratación y Registros.

Impacto Tecnológico

El estudio surge de la necesidad que tienen los nuevos desarrolladores, que buscan ampliar sus conocimientos en desarrollo de aplicaciones backend Web utilizando el lenguaje de programación Java con Spring Framework, con el objetivo de mejorar la calidad del software y el coste de tiempo durante el proceso de desarrollo.

Contexto

En el repositorio de la Biblioteca de la Universidad de Guayaquil encuentro que se ha realizado una investigación sobre “desarrollo de una arquitectura juntamente con el desarrollo del módulo de seguridad a utilizar en el sistema de ESALUD, utilizando las tecnologías JSF, Spring Framework y OpenSource” del autor García Chóez, M. O., & Tomalá Mina, E. C. (2017).

Esta investigación no tiene ninguna relación con este anteproyecto debido a que mi tema se centra en el estudio de Spring Framework para el desarrollo de backend's de sistemas Web, lo que difiere y mi propuesta es novedosa y actual.

CAPÍTULO 1

Marco Teórico

1.1. Importancia del software en empresas agrícolas

A nivel mundial existe un gran número de empresas que por su tamaño y su nivel de producción se ven en la necesidad de implementar o contratar los servicios de software para la producción agrícola, cabe recalcar que la mayoría de estos sistemas no son de uso gratuito ya que las ventajas que nos ofrecen son muy significativas en cuanto a la gestión y automatización de los procesos, mismos que tradicionalmente se los realiza de forma manual exponiéndose a errores humanos tanto de digitación y cálculo.

Uno de los sistemas más completos a nivel mundial es Agroware, un software mexicano creado para la planeación, control y costos de producción agrícola que entre sus principales beneficios están la automatización de procesos administrativos y operativos del campo.

En Ecuador tenemos una gran variedad de empresas dedicadas a la producción agrícola las cuales se ven en la necesidad de contratar o implementar un software para la gestión de procesos, en este aspecto la empresa agrícola Mastercubox S.A. gracias a un convenio existente con la Universidad Técnica del Norte permitió que estudiantes de la carrera de Ingeniería en Sistemas Computacionales desarrollen el ERP que lleva por nombre I-FOODS para la automatización de sus procesos logrando así mejorar la calidad de sus productos y mejorar el servicio que brinda a la ciudadanía del norte de Ecuador y Sur de Colombia.

1.2. Los backend's en los ERP

1.2.1. Backend

“Un desarrollador backend es quien trabaja del lado del servidor, utilizando lenguajes tales como Java, C#, Python, etc; interactuando con bases de datos, verificando sesiones de usuario y montando una página en el servidor.” (Luna, Peña, & Iacono, 2018)

LENGUAJES Y BASES DE DATOS DEL LADO DEL SERVIDOR	
LENGUAJES DE PROGRAMACIÓN	BASES DE DATOS
<p>Python Lenguaje versátil y fácil de aprender.</p>	<p>SQL Server Sistema de manejo de bases de datos del modelo relacional, desarrollado por la empresa Microsoft.</p>
<p>Java Uno de los lenguajes orientado a objetos mas populares.</p>	<p>MySQL Sistema de gestión de bases de datos relacional, desarrollado bajo la licencia dual GPL/Licencia Comercial.</p>
<p>Ruby Sin tanta historia como otras opciones, pero igualmente poderoso.</p>	<p>Oracle Sistema de gestión de bases de datos de tipo objeto relacional desarrollado por oracle.</p>
<p>C# Lenguaje orientado a objetos desarrollado y estandarizado por microsoft como parte de su plataforma .NET</p>	<p>PostgreSQL Sistema de gestión de bases de datos relacional orientado a objetos y libre.</p>

Tabla 1. Lenguajes de programación y SGBD para el desarrollo de backend
Fuente: Propia

El backend es la parte lógica de un sistema web, es decir es la comunicación y conjunto de acciones del sistema web con el servidor para que funcione de manera adecuada dicho sistema, los backend's es lo que llamamos comúnmente como la parte administrativa del sistema web en donde se gestionan los contenidos y ajustes que serán mostrados al usuario final.

“Los sistemas web están estructurados en dos capas claramente diferenciadas, backend y frontend, una división clásica definida en el área del desarrollo de software. En este contexto, el backend es el extremo del servidor de la aplicación, aquél que proporciona la lógica, definición del servicio y acceso al modelo de datos.” (Martínez, 2015).

1.2.2. ERP



Fig. 2 Arquitectura ERP.

Fuente: <https://www.goyasoluciones.com/wp-content/uploads/2018/06/queEsunErp.png>

“Los sistemas de planificación de recursos empresariales o por sus siglas en inglés Enterprise Resource Planning (ERP) tiene como propósito principal de formar una integración perfecta de información interrelacionada en toda la organización. ERP se basa en el concepto de identificar e implementar el conjunto de mejores prácticas, procedimientos y herramientas que las diferentes funciones de una empresa pueden emplear para lograr la excelencia organizativa a través de la integración de la información. El uso efectivo de los sistemas ERP ayuda a las empresas a mejorar el rendimiento de sus negocios.” (Liu, y otros, 2002).

De esta manera los sistemas ERP también tienen como objetivo eliminar la redundancia operacional automatizando los procesos. Con los diferentes módulos que lo conforman permiten el desarrollo y la gestión del negocio de manera unificada. La ventaja que se obtiene a partir del buen funcionamiento del ERP es una información más consistente que proporciona una mejora en la toma de decisiones basándose en datos reales. Las ventajas que proporcionan los ERP a las empresas se los puede definir en grandes ganancias tanto en productividad, rapidez y eficiencia al momento de realizarse el negocio de la empresa.

“Los sistemas ERP surgieron con la promesa de solucionar problemas relacionados con la obtención de informaciones integradas, con calidad y confiables para apoyar la toma de decisiones. De este modo se desarrolló en un único sistema, aportando funcionalidades que soportan las actividades de los diversos procesos de negocio de las empresas.” (Burgos, 2016)

1.2.2.1. Ventajas de un sistema ERP

“Una de las ventajas de la implementación de ese sistema es que las empresas pueden eliminar los sistemas separados y/o ineficientes, por un sistema integrado de aplicaciones. Muchos de aquellos están desactualizados con los procesos actuales de trabajo y no tienen ninguna asistencia técnica de sus desarrolladores originales. Por ello, es extremadamente difícil repararlos cuando dan problemas y es casi imposible adaptarlos a las nuevas necesidades de negocio que puedan ocurrir. Por lo tanto, esas trabas acaban por impedir a las empresas a mantenerse competitivas e innovadoras.” (Valle, Puerta, & Nuñez, 2017)

Un sistema ERP tiene la capacidad de trabajar con sistemas de información asociados que ayudan al mejoramiento de las diferentes necesidades de la empresa en el mercado que se desempeña, además ayuda al mejoramiento de la calidad, rapidez y confiabilidad con la que se obtiene información de la empresa eliminando la redundancia e inconsistencia de los datos.

“Al ofrecer una base de datos integrada, el sistema de ERP ayuda a la mejoría del acceso de los datos para la toma de decisiones operacionales. A través de un conjunto de datos que dan soporte a todas las funciones empresariales, es posible proporcionar un excelente soporte a la toma de decisiones, permitiendo que las empresas ofrezcan servicios más diversificados y específicos.” (Valle, Puerta, & Nuñez, 2017)

1.3. Spring Framework.

Spring framework es un proyecto de código abierto por lo cual cuenta con una extensa comunidad que brinda retroalimentación basándose en una gran tendencia de casos de uso en todo el mundo, gracias a esto se ha logrado mejorar con éxito todo el proyecto de spring framework y sus diferentes módulos.

Fecha	Versión
Marzo de 2004	Versión 1
Abril de 2004 – octubre de 2006	Nuevas versiones posteriores hasta lograr incluso premios en 2006
Octubre de 2006	Versión 2
Noviembre de 2007	Versión 2.5
Diciembre de 2009	Versión 3
Diciembre de 2011	Versión 3.1
Noviembre de 2013	Versión 3.2.5

Diciembre de 2014	Versión 4, ya con soporte para Java SE 8, Groovy, etc.
Julio de 2015	Versión 4.2.0
Septiembre de 2015	Versión 4.2.1, todavía compatible con Java 6,7 y 8
Junio de 2016	Versión 4.3, última versión que permite Java SE 6 y Servlet 2.5 en adelante con soporte hasta 2019
Finales de 2017	Versión 5

Tabla 2. Evolución de versiones Spring Framework.
Fuente: Spring 5 Manual Imprescindible

En el sitio oficial de Spring framework nos dice:

“Spring facilita la creación de aplicaciones empresariales Java, proporcionando todo lo que necesita para adoptar el lenguaje Java en un entorno empresarial, con soporte para Groovy y Kotlin como lenguajes alternativos en la JVM, y con la flexibilidad de crear muchos tipos de arquitecturas según las necesidades de una aplicación.” (Pivotal, 2019)

“Uno de los mayores logros de Spring, sin duda, fue que consiguió desplazar los EJB Enterprise Java Beans, la tecnología oficial que existía a principios de milenio para el desarrollo de aplicaciones empresariales. Los EJB eran mucho más complejos de manejar y el framework Spring les fue ganando terreno hasta ser el líder.” (Pérez Martínez & Altadill Izura, 2018)

1.4. Base de Datos PostgreSQL.

“PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado” (Zae Ordoñez, Molina Ríos, & Redrován Castillo, 2017)

En este caso se ha utilizado PostgreSQL como SGBD para la realización y ejecución del proyecto ya que por ser una de las herramientas más robustas y de licencia gratuita se ajusta a las necesidades del sistema web.

“PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando” (Zae Ordoñez, Molina Ríos, & Redrován Castillo, 2017)

En el sitio oficial de PostgreSQL se puede encontrar las características de este administrador de base de datos, las mismas que se muestran a continuación:

- Tipos de datos

- Integridad de los datos
- Concurrencia, rendimiento
- Confiabilidad, recuperación de desastres
- Seguridad
- Extensibilidad
- Internacionalización, búsqueda de texto

“Hay muchas más características que puede descubrir en la documentación de PostgreSQL. Además, PostgreSQL es altamente extensible: muchas características, como los índices, tienen API definidas para que pueda desarrollarse con PostgreSQL para resolver sus desafíos.” (PostgreSQL, 2019)

1.5. Metodología Scrum

“Scrum es un marco de referencia para crear software complejo y entregarlo a tiempo de una forma mucho más sencilla.

Scrum es realmente fácil de entender, pero puede tomar algunos años antes de dominarlo. Sin embargo, esto no debe ser motivo para desinteresarse; los beneficios superan con creces la curva de aprendizaje.” (Dimes, 2015)

La metodología ágil scrum nos propone una nueva modalidad para crear el software en cortos ciclos repetitivos los mismos que podrían ir desde una semana hasta un mes, estos pequeños períodos de tiempos generalmente son llamados sprint.

“El beneficio de trabajar por iteraciones, es que cada una de ellas tiene como resultado un producto listo para entregar, de tal manera que, si la siguiente iteración genera un software inestable y/o con muchos errores, el equipo de trabajo simplemente revierte al último hito sin necesidad de empezar de cero.” (Dimes, 2015)

En scrum existen equipos de trabajo los cuales desarrollan las tareas designadas mediante ciclos repetitivos o iteraciones, este equipo de trabajo también realiza cortas reuniones en donde se deciden las tareas que se realizarán en cada iteración o repetición.

Todos los involucrados en el equipo informan sobre todas las actividades en las que se encuentran trabajando, los problemas que se han suscitado en el desarrollo de estas, las futuras tareas que asumirán y la duración que tendrán cada actividad para su desarrollo.

Para concluir se programan repeticiones secundarias ya sea para incluir o desechar características del software en desarrollo, todo esto dependerá de la disponibilidad de tiempo para su finalización y de las restricciones que se tengan.

1.5.1. ¿Por qué usar Scrum?

Gran parte de las industrias y empresas de desarrollo de software se enfrentaban a un gran problema al momento de llevar a cabo o entregar un prototipo del sistema en desarrollo, ya que sus productos no eran entregados en el tiempo acordado y si los entregaban no lo hacía satisfactoriamente por lo cual carecía de muchas de las características importantes.

Esto sucede porque mucho de los desarrolladores no tienen tiempos específicos para la entrega de las tareas designadas y hasta se subestiman las tareas de mayor complejidad y sin darles importancia sino hasta cuando el plazo de entrega está por cumplirse, enfrentándose así a problemas de mayor complejidad y entregando a destiempo un código desordenado y en ocasiones hasta inservible por la premura del tiempo y la intención de cumplir el tiempo de entrega.

Actualmente la mayoría de las empresas desarrolladoras de software trabajan bajo metodologías ágiles como son Scrum o XP, las mismas que han ayudado para el desarrollo de un sistema de calidad con todas las características necesarias en el tiempo acordado.

Los desarrolladores de software son más productivos ya que las tareas son divididas en pequeñas partes con un tiempo determinado de entrega y juntamente con el equipo de trabajo se van creando funcionalidades que se perciben como necesidades para el cliente.

CAPÍTULO 2

Herramientas Utilizadas

2.1. Spring Boot

2.1.1. Spring Boot

“Spring Boot es una de las características más útiles de Spring, ya que facilita enormemente la configuración de un proyecto poniendo a punto las dependencias de software y, en definitiva, acelerando la puesta en marcha de proyectos. Hasta la llegada de Spring Boot era necesario configurar los distintos paquetes que necesitaba el proyecto de manera manual, dando lugar a todo tipo de problemas con las versiones de todo el software necesario, ya fuera a través de ficheros de configuración Maven o en otros gestores de proyectos como Gradle.” (Pérez Martínez & Altadill Izura, 2018)

Spring Boot contiene un grupo de dependencias interrelacionadas mismas que ayudan para obtener un gran soporte para varias funcionalidades que vienen asociadas.

En el sitio oficial de Spring Framework nos comenta sobre Spring Boot lo siguiente: “Spring Boot facilita la creación de aplicaciones basadas en Spring, autómatas y del nivel de producción que simplemente se ejecutan. Tomamos una visión dogmática de la plataforma de Spring y de las bibliotecas de terceros para que pueda empezarse con el mínimo revuelo” (Pivotal, 2019)

2.1.2. Dependencias de Spring Boot

Spring Boot ofrece dos dependencias de serie, obviamente se puede agregar otras dependencias a mano, pero esto no resultaría tan cómodo además que hay que tener en cuenta incompatibilidades de las versiones.

“Las dependencias esenciales en proyectos de consola Spring Boot serían *starter* y las dependencias de *testing*.” (Pérez Martínez & Altadill Izura, 2018)

```

<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>

```

Fig. 3 Dependencias de serie de Spring Boot.
Fuente: Spring 5 Manual Imprescindible

2.1.3. Acceso a datos con Spring Boot

Una de las características de las aplicaciones empresariales es que pueden gestionar los datos de la empresa almacenados en un SGBD (Software de Gestión de Base de Datos). Generalmente esto se puede lograr cuando el código tiene acceso a un origen de datos, enviando consultas y procesando los resultados.

“Spring facilita enormemente el acceso a datos adoptando las técnicas de los frameworks ORM como Hibernate, donde los datos de un SGBD se traducen a colecciones de objetos y viceversa. Gracias al uso de interfaces como EntityManager, CrudRepository, etc.” (Pérez Martínez & Altadill Izura, 2018)

2.1.3.1. Patrón DAO

“El patrón DAO se aplica para que un proyecto de software se abstraiga lo más posible de un origen de datos y la lógica de negocio se ocupe, únicamente, de gestionar objetos. Una clase DAO es la encargada de encapsular las operaciones esenciales: buscar por id, buscar todos los registros, insertar, modificar y eliminar es decir las operaciones CRUD.” (Pérez Martínez & Altadill Izura, 2018)

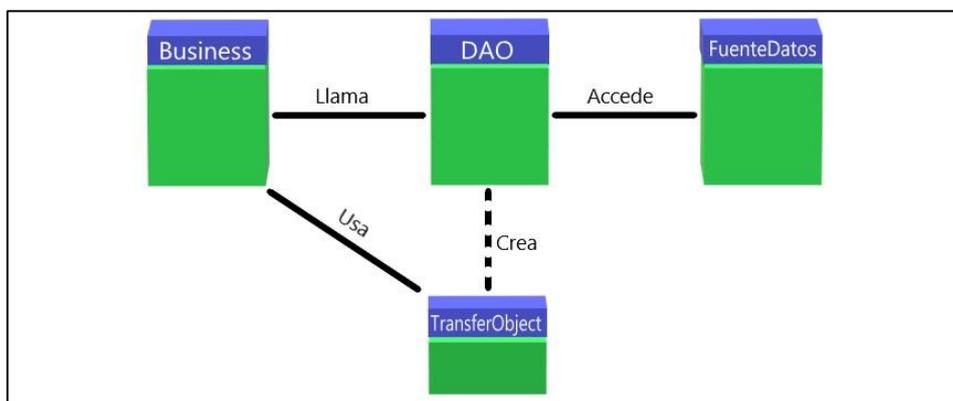


Fig. 4 Patrón DAO
Fuente: Propia

2.2. Modelos, vistas y controladores

2.2.1. Patrón MVC

“Este es el patrón más común que siguen la mayoría de los frameworks para la web, tanto en backend como en frontend. El patrón MVC permite organizar el proyecto de tal manera que se desacoplan todos los elementos que forman parte de este, separando la lógica de negocio de la presentación.” (Pérez Martínez & Altadill Izura, 2018)

Las siglas MVC significan Modelo, Vista, Controlador y son estos cada uno de los componentes que forman parte de este patrón de diseño.

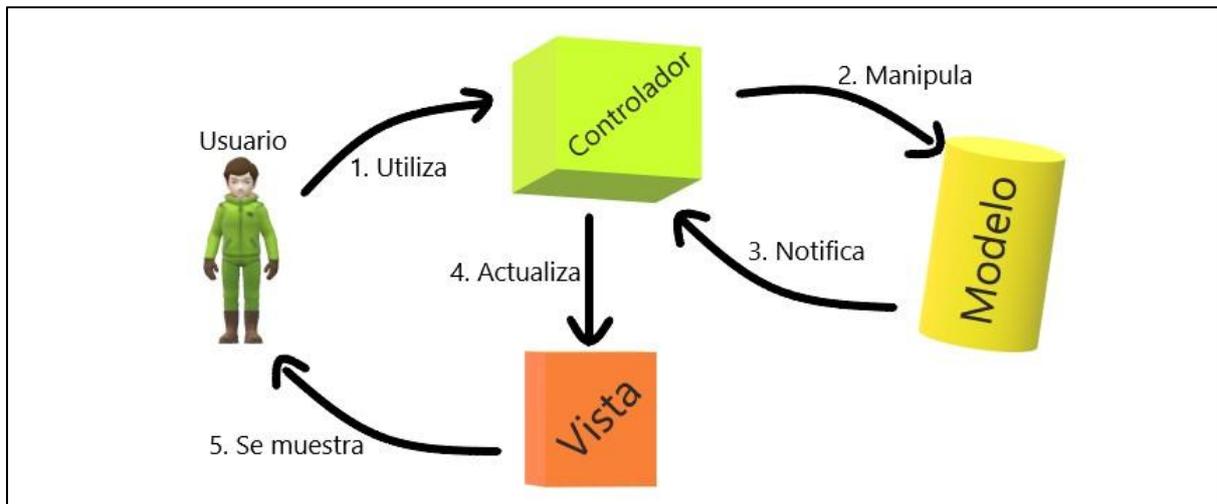


Fig. 5 Patrón MVC
Fuente: Propia

El patrón MVC empieza su funcionamiento cuando el usuario hace una petición, o en el caso de la Web la petición se la realiza por medio de un clic en un enlace o el envío de un formulario, el papel que desempeñan cada uno de estos componentes son los siguientes:

- Controlador: este componente es el encargado de recibir la petición del usuario y analizarla para luego enviar a la lógica de negocio correspondiente para poder manipular al modelo.
- Modelo: en este componente se encuentra la representación de la información, el cuál luego de recibir la petición del controlador envía la información hacia la vista.
- Vista: es el componente encargado de mostrar al usuario final la presentación de la información, la vista se modifica con lo que recibe de respuesta del modelo.

2.3. Base de datos

En la actualidad las bases de datos son de vital importancia para todo el mercado comercial indistintamente del tipo de negocio al que la empresa se dedique, están basadas en la administración de información ya sea desde una o varias tablas relacionadas y con una gran variedad de software para aplicaciones web son utilizadas en todo el mundo mejorando la confiabilidad de los datos y disminuyendo la redundancia de estos.

“Una base de datos es un banco de información, el cual contiene datos relacionados entre sí y se encuentran agrupados o estructurados; además son manipulados por programas conocidos actualmente como Sistema de Gestión de Base de Datos (SGBD).” (Zae Ordoñez, Molina Ríos, & Redrován Castillo, 2017)

“Las bases de datos cumplen una función muy importante en los sistemas de información, la mayoría de las empresas sean públicas o privadas tiene sus procesos automatizados y esto hace que los sistemas manuales queden obsoletos a la hora de realizar búsquedas, modificaciones y cualquier obtención de información de dicha empresa, es por ello que la utilización de una base de datos se hace indispensable al momento de almacenar grandes volúmenes de información con la que cuenta la organización.” (Zae Ordoñez, Molina Ríos, & Redrován Castillo, 2017)

2.4. Servidor de aplicaciones

Un servidor de aplicaciones es aquel que proporciona la lógica de negocio y disponibilidad para una aplicación desplegada, el servidor de aplicaciones comúnmente es parte de una aplicación de tres capas o niveles que consta de interfaz gráfica de usuario (GUI), servidor de aplicaciones (lógica de negocio) y un servidor de base de datos.

En el sitio oficial de la empresa IBM define a los servidores de aplicaciones como “Los servidores de aplicaciones son programas de servidor en una red distribuida que proporciona el entorno de ejecución para un programa de aplicación. Más específicamente, el servidor de aplicaciones es el componente de tiempo de ejecución principal en todas las configuraciones y donde una aplicación se ejecuta. El servidor de aplicaciones colabora con el servidor web para ofrecer una respuesta dinámica y personalizada a una solicitud de cliente.” (IBM, 2015)

De acuerdo con Bravo (2015) menciona que:

“Los servidores de aplicaciones más usados están basados en la tecnología J2EE (Java 2 Enterprise Edition), misma que es un conjunto de tecnologías para desarrollo de aplicaciones web.” (Bravo, 2015)

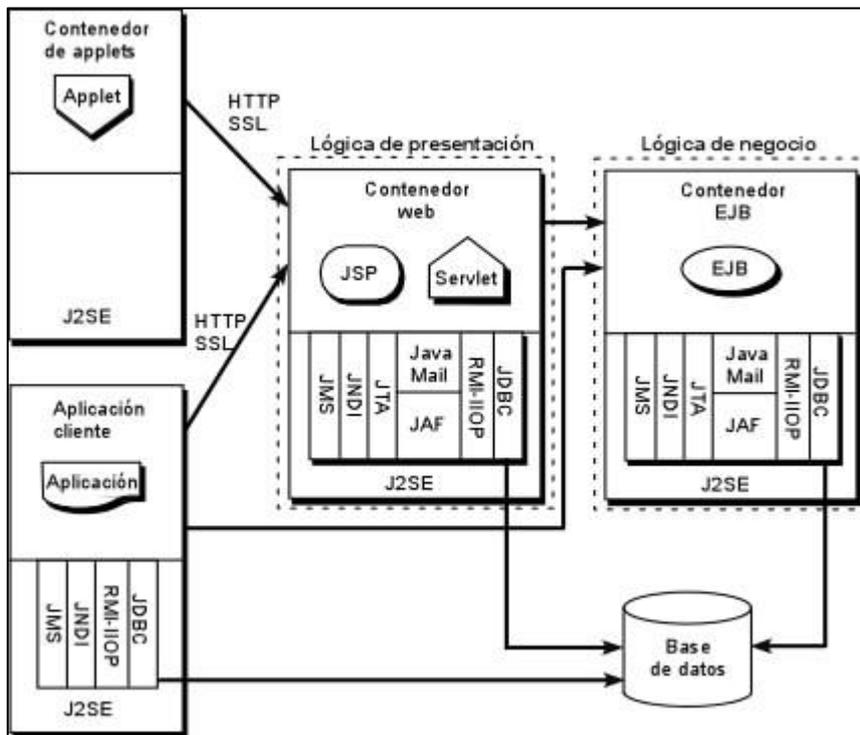


Fig. 6 Arquitectura J2EE.

Fuente: <http://www.jtech.ua.es/j2ee/2003-2004/abierto-j2ee-2003-2004/sa/primera.jpg>

Entre los principales servidores de aplicaciones podemos encontrar los siguientes:

- Servidor de aplicaciones para JAVA EE, como con Weblogic de Oracle, WebSphere de IBM, JBoss, GlassFish.
- En la versión Enterprise Edition de Java contiene clases para el desarrollo de este tipo de aplicaciones, como son Servlets, Java Server Pages y Enterprise Java Beans (EJBs).

(Apache, 2019)

2.4.1. Apache Tomcat

“El Apache Tomcat[®] software es una implementación de código abierto del servlet de Java, Java Server Pages, tecnologías Java WebSocket del lenguaje Java y Expresión. Las especificaciones de Java Servlet, JavaServer Pages, Java Expression Language y Java WebSocket se desarrollan bajo el Proceso de la Comunidad Java.” (Apache, 2019)

CAPÍTULO 3

Desarrollo

3.1. Implementación módulo de precontratación y contratación

Para la realización del módulo de precontratación y contratación en la empresa MASTERCUBOX, se desarrollará un sistema informático para la gestión de los procesos, el módulo se integrará con los módulos de control de materia prima, control de calidad y móvil, mismos módulos que conforman un proyecto denominado I-FOODS dentro de la empresa, estos módulos fueron realizados por estudiantes de la carrera de Ingeniería en Sistemas Computacionales como parte del proyecto integrador realizado para la empresa.

3.2. Definición de Requisitos

Las historias de usuario se obtuvieron luego de varias reuniones realizadas con el Ing. Pedro Román jefe del departamento agrícola de la empresa Mastercubox S.A. y Jefferson Ortega (Tesisista) con el fin de realizar el levantamiento de requerimientos.

Tabla 3. Historia de Usuario Nro. 1

HISTORIA DE USUARIO		
Número: 1	Usuario: Cliente/Administrador	
Nombre: Levantamiento de requerimientos		
Prioridad: Alta	Riesgo: Alto	Estimación: 8 horas
Descripción: Se obtiene los requisitos con una reunión con el dueño del producto		
Pruebas de Aceptación: Levantamiento de requisitos firmados por el dueño del producto		

Tabla 4. Historia de Usuario Nro. 2

HISTORIA DE USUARIO		
Número: 2	Usuario: Cliente/Administrador	
Nombre: Base de datos		
Prioridad: Alta	Riesgo: Alto	Estimación: 10 horas

Descripción: Como cliente necesito una base de datos diseñada de manera sencilla y sobre todo entendible, debe contener todas las tablas que son necesarias para un correcto funcionamiento.

Pruebas de Aceptación:

- Las tablas deben estar muy bien relacionadas con las tablas de los otros módulos dentro de la base de datos.
 - La base de datos debe ser realizada bajo licencia gratuita.
 - Documentar la base de datos completa con los diferentes módulos.
-

Tabla 5. Historia de Usuario Nro. 3

HISTORIA DE USUARIO

Número: 3

Usuario: Administrador

Nombre: Gestión de Proveedores en el sistema

Prioridad: Alta

Riesgo: Alto

Estimación: 12 horas

Descripción: Como administrador del sistema necesito registrar nuevos proveedores para la empresa ingresando los datos personales como: cédula de identidad, nombre de representante, nombre de razón social, dirección de domicilio, correo electrónico, números de teléfonos.

Pruebas de Aceptación:

- Todos los campos de ingreso de datos deben estar validados.
 - Se emitirá mensajes de advertencia antes de guardar los datos ingresados
-

Descripción: Como administrador del sistema necesito crear nuevos contratos con proveedores mismos que se generara con la información del proveedor, predio y la variedad del producto.

Pruebas de Aceptación:

- Se emitirá mensajes de advertencia al momento de crear un nuevo contrato.
 - Se generará un modelo de contrato con las cláusulas predefinidas y listo para su impresión.
-

Tabla 8. Historia de Usuario Nro. 6

HISTORIA DE USUARIO

Número: 6

Usuario: Administrador

Nombre: Reportes

Prioridad: Alta

Riesgo: Medio

Estimación: 10 horas

Descripción: Como administrador del sistema necesito ver un listado de proveedores y predios, mismo listado debe ser completo. También necesito un reporte detallado de los contratos generados.

Pruebas de Aceptación:

- Imprimir reporte general.
-

3.3. Definición del Product Backlog

Cuando se ha generado las historias de usuario se puede obtener la lista del producto (Product backlog) de las tareas a realizarse del proyecto, ver Tabla 9.

Tabla 9. Lista del Producto

ID	PRIORIDAD	HISTORIA	ESTIMACIÓN
HU1	ALTA	Levantamiento de requerimientos	8
HU2	ALTA	Base de datos	10
HU3	ALTA	Gestión de Proveedores en el sistema	12
HU4	ALTA	Gestión de predios en el sistema	12
HU5	ALTA	Gestión de contratos en el sistema	12
HU6	ALTA	Reportes	10

3.4. Definición de los Roles del Proyecto

En el desarrollo del módulo de precontratación y contratación de la empresa Mastercubox S.A. se han designado los siguientes roles de conformidad a la metodología Scrum. Ver tabla 10.

Tabla 10. Roles del Proyecto

PERSONA	DESCRIPCIÓN	ROL
Ing. Mauricio Rea	Director de Trabajo de Grado y Docente de la Carrera de Sistemas de la Universidad	Jefe de Proyecto (Scrum Máster)
Ing. Miguel Iturralde	Gerente General de la empresa Mastercubox S.A.	Propietario del Producto (Product Owner)
Sr. Jefferson Ortega	Tesista	Equipo de Desarrollo (Development Team)

3.5. Desarrollo del Aplicativo

En la realización del sistema se trabajó con la metodología ágil de SCRUM, misma que trabaja por sprints los cuáles son lapsos de tiempo para realizar tareas designadas, en cada sprint se trabaja para cumplir con el objetivo de los requisitos de software teniendo como apoyo las historias de usuario.

Para el correcto desarrollo del sistema se puede observar detalladamente los procesos de diseño, desarrollo e implementación de cada sprint para finalmente obtener un producto terminado.

3.5.1. Desarrollo de los Sprints

En esta fase del desarrollo del proyecto se detalla los sprints realizados hasta lograr obtener un producto terminado, así como también una síntesis de la realización de los sprints.

Tabla 11. Desarrollo de los Sprints

Sprint	Inicio	Finalización
Sprint 0	06/03/2018	06/04/2018
Sprint 1	07/04/2018	04/05/2018
Sprint 2	05/05/2018	01/06/2018
Sprint 3	01/06/2018	29/06/2018

3.5.1.1. Sprint 0

a) Reunión de planificación

Fecha de reunión: 06/03/2018

Asistentes a la reunión: Scrum Máster, Product Owner, Development Team

Fecha de inicio de Sprint: 06/03/2018

Fecha de fin de Sprint: 06/04/2018

Objetivo de Sprint: Definición de base de datos del sistema y arquitectura tecnológica.

- **Sprint Backlog**

Tabla 12. Sprint 0 Backlog

ID	Historia de usuario
HU1	Levantamiento de requerimientos
HU2	Base de datos

En el Sprint inicial se analizó la arquitectura a emplearse para el desarrollo de todos los módulos necesarios del sistema I-FOODS.

Tabla 13. Planificación del Sprint 0

PLANIFICACIÓN DE TRABAJOS DE DESARROLLO SPRINT 0			
Historia de Usuario	Fase Desarrollo	Tarea	Tiempo Estimado (Horas)
HU1	Análisis	Realizar los diagramas de proceso.	2
	Análisis	Wireframe, se realizarán los diseños de las pantallas del sistema	8
HU2	Análisis	Definir base de datos	3
	Análisis	Realizar el modelo entidad – relación, utilizando un programa para diseñar la base de datos	8
	Análisis	Acoplar el modelo en una base de datos consolidada: Se unirán todos los modelos de la base de datos de los diferentes módulos y se entrega una base de datos central	7
	Análisis	Implementar el modelo en la base de datos seleccionada	5
Reuniones	Planificación	Planificación	4
	Revisión	Revisión	2
	Revisión	Retrospectiva	1
Tareas no planificadas			
Total			40

b) Reunión de revisión

Con las respectivas historias de usuario completadas en este sprint inicial se presenta al cliente en forma de incremento del producto final.

Finalizando el sprint inicial se obtiene un cumplimiento de los requerimientos del sistema planteados en el Product Backlog.

Tabla 14. Seguimiento del Sprint 0

SEGUIMIENTO SPRINT 0			
Proyecto: Módulo de Precontratación y contratación			
Historia de Usuario	Desarrollador	Descripción	Estado

HU1	Jefferson Ortega	Reunión para definición de requerimientos	Realizado
	Jefferson Ortega	Realizar los diagramas de procesos	Realizado
	Jefferson Ortega	Wireframes	Realizado
HU2	Jefferson Ortega	Definir base de datos	Realizado
	Jefferson Ortega	Realizar el modelo entidad – relación	Realizado
	Jefferson Ortega	Acoplar el modelo en una base de datos consolidada	Realizado
	Jefferson Ortega	Realizar el modelo en la base de datos	Realizado
Reuniones	TEAM	Planificación	Realizado
	TEAM	Revisión	Realizado
	TEAM	Retrospectiva	Realizado

c) Incremento del producto potencialmente entregable

- **Arquitectura**

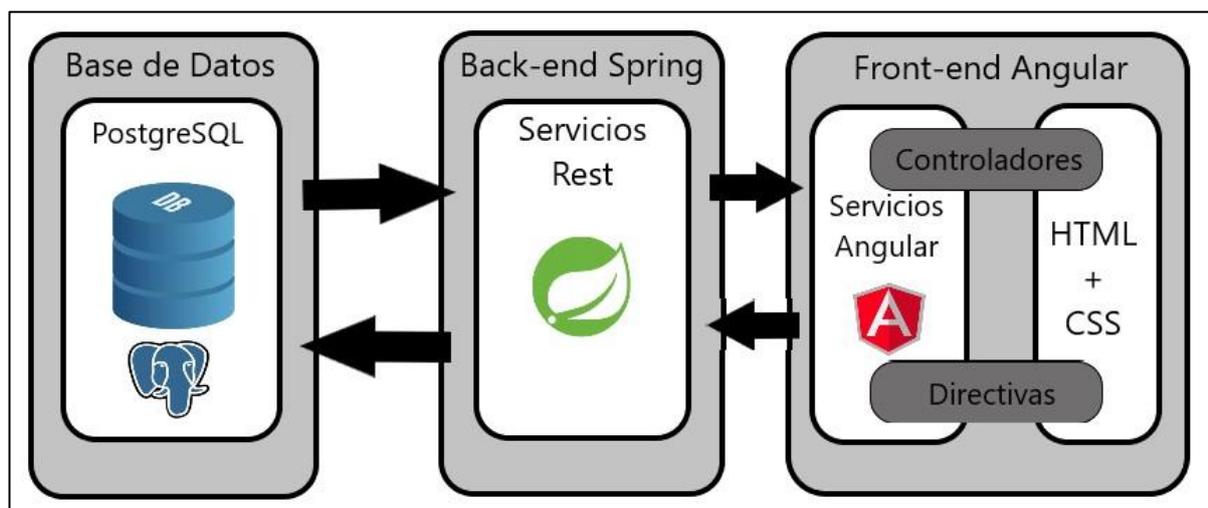


Fig. 7 Arquitectura del módulo de Precontratación y contratación
Fuente: Propia

Esta arquitectura se resolvió entre todo el equipo de trabajo para todos los módulos que conformarán este sistema, se desarrolló con la base de datos PostgreSQL centralizada, para el backend como servidor se desarrolló con el framework Spring y para la parte de la vista o frontend con el framework Angular. Con estas herramientas escogidas se desarrolló el sistema I-FOODS un software de calidad gracias a su fácil acoplamiento ya que tanto el framework del backend como el framework del frontend fueron desarrollados con el lenguaje Java.

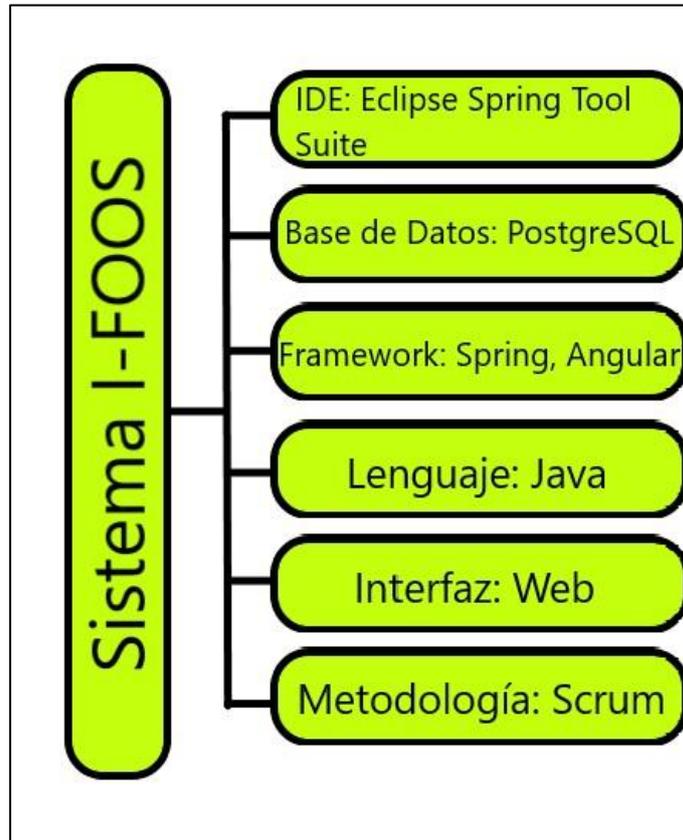


Fig. 8 Herramientas para el desarrollo del sistema I-FOODS.
Fuente: Propia

3.5.1.2. Sprint 1

d) Reunión de Planificación

Fecha de Reunión: 06/03/2018

Asistentes a la reunión: Scrum Máster, Product Owner, Team Development.

Fecha Inicio Sprint: 07/04/2018

Fecha Fin Sprint: 04/05/2018

Objetivo de Sprint: Desarrollo de CRUDS de las tablas proveedores y predios.

- **Sprint Backlog**

Tabla 15. Sprint 1 Backlog

ID	Historia de usuario
HU3	Gestión de Proveedores en el sistema
HU4	Gestión de Predios en el sistema

Tabla 16. Planificación del Sprint 1

PLANIFICACIÓN DE TRABAJOS DE DESARROLLO SPRINT 1				
Proyecto: Módulo de Precontratación y contratación				
Historia de Usuario	Desarrollador	Fase Desarrollo	Descripción	Tiempo Estimado (Horas)
Gestión de Proveedores	Jefferson Ortega	Codificación	Crear Proveedor: Diseño del formulario de registro de proveedores, en donde el usuario ingresa los datos correspondientes que son: CI o RUC, Nombre, Dirección, Teléfono Fijo, Teléfono Móvil, Correo, Fecha de Registro.	4
	Jefferson Ortega	Codificación	Listar Proveedores: Diseño y codificación de la vista de ingresos del sistema.	2
	Jefferson Ortega	Codificación	Editar Proveedores: Diseño y codificación de la opción editar en cada registro de ingresos del sistema.	2
Gestión de Predios	Jefferson Ortega	Codificación	Crear Predio: Diseño del formulario de registro de proveedores, en donde el usuario ingresa los datos correspondientes que son: Accesibilidad, Agua de Riego, Altitud, Distancia a la Fabrica, Superficie, Tipo de Suelo, Pedregosidad, Topografía.	4
	Jefferson Ortega	Codificación	Listar Predios: Diseño y codificación de la vista de ingresos del sistema.	3
	Jefferson Ortega	Codificación	Editar Predio: Diseño y codificación de la opción editar en cada registro de ingresos del sistema.	3
Reuniones		Planificación	Planificación	12
		Revisión	Revisión	6
		Revisión	Retrospectiva	3
Tareas no planificadas				
Total				39

e) Reunión de revisión

Una vez finalizado todas las tareas programadas dentro del tiempo establecido, se determinó que se dio cumplimiento a los requerimientos planteados en el Product Backlog.

Tabla 17. Seguimiento del Sprint 1
SEGUIMIENTO SPRINT 1

Proyecto: **Módulo de Precontratación y contratación**

Historia de Usuario	Desarrollador	Fase Desarrollo	Descripción	Tiempo Estimado (Horas)	Tiempo Real (Horas)	Estado
Gestión de Proveedor	Jefferson Ortega	Codificación	Crear Proveedor	4	6	Realizado
	Jefferson Ortega	Codificación	Listar Proveedor	2	3	Realizado
	Jefferson Ortega	Codificación	Editar Proveedor	2	2	Realizado
Gestión de Predio	Jefferson Ortega	Codificación	Crear Predio	4	6	Realizado
	Jefferson Ortega	Codificación	Listar Predio	3	3	Realizado
	Jefferson Ortega	Codificación	Editar Predio	3	2	Realizado
Reuniones	TEAM	Planificación	Planificación	12	12	Realizado
	TEAM	Revisión	Revisión	6	6	Realizado
	TEAM	Retrospectiva	Retrospectiva	3	3	Realizado
TOTAL				39	43	

f) Incremento del producto potencialmente entregable

Luego de realizar las pruebas necesarias se hace entrega del sprint con el total cumplimiento.

A continuación, imágenes del diseño de prototipo y resultado final.

- Prototipo de la creación de proveedores, muestra un modelamiento del producto final haciendo referencia a la Historia de Usuario Nro. 3

Fig. 9 Diseño de Registro de Proveedores.
Fuente: Propia

- Listado de Proveedores: esta imagen es de la interfaz web en la cual se aprecia el listado de datos y parte de administración de Proveedores misma que puede realizar un nuevo ingreso de proveedor y editar cualquiera de los proveedores almacenados en la base de datos. Este tipo de tabla permite exportar todos los datos a un formato pdf y Excel, así como también imprimir.

Botones	RUC-CI	NOMBRE REPRESENTANTE	RAZON SOCIAL	FECHA REGISTRO	CORREO	DIRECCION	ESTADO	TEL FIJO	TEL MOVIL
	100	Andres Suarez	Andres Suarez	2019-01-29	AndresSuarez@gmail.com	Ibarra	A	909898988	099898989
	1003516687001	Jefferson Ortega	Jefferson Ortega	2019-01-30	jsortegac@utn.edu.ec	Ibarra	A	2631156	0960552729
	1001183589	Miguel Angel Iturralde	Miguel Angel Iturralde Castelo	2019-02-03	miturralde@mastercubox.com	Juan Genaro Jaramillo 4 - 46	A	2951458	0993474029
	1003111877	Xavier Inuca	Xavier Inuca	2019-02-06	masflow@hotmail.com	OTAWA	A	2615234	0987654321

Fig. 10 Listado de Registro de Proveedores.
Fuente: Propia

- Prototipo de la creación de predios, muestra un modelamiento del producto final haciendo referencia a la Historia de Usuario Nro. 4

Fig. 11 Diseño de Registro de Predios.
Fuente: Propia

- Listado de Predios: esta imagen es de la interfaz web en la cual se aprecia el listado de datos y parte de administración de Predios misma que puede realizar un nuevo ingreso de predio y editar cualquiera de los predios almacenados en la base de datos. Este tipo de tabla permite exportar todos los datos a un formato pdf y Excel, así como también imprimir.

Botones	Accesibilidad	Agua de riego	Altitud	Distancia a la fabrica	Superficie	Tipo de suelo	Pedregosidad	Topografia	Estado del predio
	A	SI	2500	12.1	25	Arenoso	Baja	Plano a ligeramente plano	A
	A								
	A								
	A	SI	2700	45	4	Limoso	0 cm (Nula)	4% - 7% (Ligeramente inclinado)	A
	A	SI	240	5	5	Franco Limoso	Piedras de 4cm - 10cm (Media)	4% - 7% (Ligeramente inclinado)	A

Fig. 12 Listado de Registro de Predios.
Fuente: Propia

```

public class PredioController {

    @Autowired
    private PredioRepository predioRepository;
    @Autowired
    private ProveedoresRepository proveedoresRepository;
    @Autowired
    private ParroquiaRepository parroquiaRepository;
    @Autowired
    private FotosRepository fotosRepository;

    @RequestMapping(method = RequestMethod.GET, produces = MediaType.APPLICATION_JSON_VALUE)
    public List<PrePredio> getPredio() {

        return predioRepository.findAll();
    }

    @RequestMapping(value="/{id}",method = RequestMethod.GET, produces = MediaType.APPLICATION_JSON_VALUE)
    public Optional<PrePredio> getPredioporId(@PathVariable("id") Integer id) {
        return this.predioRepository.findByIdPredio(id);
    }

    @RequestMapping(value="prov/{id}",method = RequestMethod.GET, produces = MediaType.APPLICATION_JSON_V
    public List<PrePredio> getPredioporCiRuc(@PathVariable("id") String id) {
        return this.predioRepository.findPredioPorCedula(id);
    }
    //
    //

    @RequestMapping(method = RequestMethod.POST, produces = MediaType.APPLICATION_JSON_VALUE, consumes =
    public PrePredio save(@RequestBody PrePredio predio) {
        return this.predioRepository.save(predio);
    }
}

```

Fig. 13 Controller de la entidad Predio
Fuente: Propia

```

import java.text.ParseException;

@CrossOrigin(origins="*", allowCredentials="true", allowedHeaders="*")
@RestController
@RequestMapping(value = "proveedor")

public class ProveedorController {

    @Autowired
    private ProveedoresRepository proveedoresRepository;
    @Autowired
    private ParroquiaRepository parroquiaRepository;

    @RequestMapping(method = RequestMethod.GET, produces = MediaType.APPLICATION_JSON_VALUE)
    public List<PreProveedor> getProveedoresC() {
        return proveedoresRepository.findAll();
    }

    @RequestMapping(value="/{id}",method = RequestMethod.GET, produces = MediaType.APPLICATION_JSON_VALUE)
    public Optional<PreProveedor> getProveedorporRucCi(@PathVariable("id") String id) {
        return this.proveedoresRepository.findByRucCi(id);
    }

    @RequestMapping(method = RequestMethod.POST, produces = MediaType.APPLICATION_JSON_VALUE, consumes = "application/json")
    public void save(@RequestBody PreProveedor proveedor) {
        this.proveedoresRepository.save(proveedor);
    }

    @RequestMapping(method = RequestMethod.PUT, produces = MediaType.APPLICATION_JSON_VALUE, consumes = "application/json")
    public void editar(@RequestBody PreProveedor proveedor) {

```

Fig. 14 Controller de la entidad Proveedor
Fuente: Propia

g) Reunión de Retrospectiva

Tabla 18. Plan de mejoras del Sprint 1

RETROSPECTIVA		
Fecha: 04/05/2018		
Asistentes a la reunión: Scrum Máster, Development Team		
¿Qué salió bien en la iteración? (aciertos)	¿Qué no salió bien en la iteración? (errores)	¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)
<p>✓ Comunicación entre todos los integrantes del equipo de trabajo</p>	<p>- Desconocimiento de comando de GIT al momento de integrar los diferentes módulos</p>	<p>• Mejorar los conocimientos sobre GIT</p>

3.5.1.3. Sprint 2

h) Reunión de Planificación

Fecha de Reunión: 06/04/2018

Asistentes a la reunión: Scrum Máster, Product Owner, Team Development.

Fecha Inicio Sprint: 05/05/2018

Fecha Fin Sprint: 05/06/2018

Objetivo de Sprint: Desarrollo de la gestión de la tabla contratos.

- **Sprint Backlog**

Tabla 19. Sprint 2 Backlog

ID	Historia de usuario
HU5	Gestión de Contratos en el sistema

Tabla 20. Planificación del Sprint 2

PLANIFICACIÓN DE TRABAJOS DE DESARROLLO SPRINT 2				
Proyecto: Módulo de Precontratación y contratación				
Historia de Usuario	Desarrollador	Fase Desarrollo	Descripción	Tiempo Estimado (Horas)
Gestión de Contratos	Jefferson Ortega	Codificación	Crear Contrato: Diseño del formulario de registro de proveedores, en donde el usuario ingresa los datos correspondientes que son: Seleccionar Proveedor, Seleccionar Predio, Variedad, Cláusulas.	4
	Jefferson Ortega	Codificación	Listar Contratos: Diseño y codificación de la vista de ingresos del sistema.	3
	Jefferson Ortega	Codificación	Impresión del Contrato: Diseño y codificación de la opción para imprimir en cada registro de ingresos del sistema.	2
Reuniones		Planificación	Planificación	12
		Revisión	Revisión	6
		Revisión	Retrospectiva	3
Tareas no planificadas				
Total				30

i) Reunión de revisión

Una vez finalizado todas las tareas programadas dentro del tiempo establecido, se determinó que se dio cumplimiento a los requerimientos planteados en el Product Backlog.

Tabla 21. Seguimiento del Sprint 2
SEGUIMIENTO SPRINT 2

Proyecto: **Módulo de Precontratación y contratación**

Historia de Usuario	Desarrollador	Fase Desarrollo	Descripción	Tiempo Estimado (Horas)	Tiempo Real (Horas)	Estado
Gestión de Contrato	Jefferson Ortega	Codificación	Crear Contrato	4	6	Realizado
	Jefferson Ortega	Codificación	Listar Contrato	3	3	Realizado
	Jefferson Ortega	Codificación	Impresión Contrato	2	3	Realizado
Reuniones	TEAM	Planificación	Planificación	12	12	Realizado
	TEAM	Revisión	Revisión	6	6	Realizado
	TEAM	Retrospectiva	Retrospectiva	3	3	Realizado
TOTAL				30	33	

j) Incremento del producto potencialmente entregable

Luego de realizar las pruebas necesarias se hace entrega del sprint con el total cumplimiento.

A continuación, imágenes del diseño de prototipo y resultado final.

- Prototipo de la creación de Contratos, muestra un modelamiento del producto final haciendo referencia a la Historia de Usuario Nro. 5

Fig. 15 Diseño de Registro de Contratos.
Fuente: Propia

- Listado de Contratos: esta imagen es de la interfaz web en la cual se aprecia el listado de datos y parte de administración de Contratos misma que puede realizar un nuevo ingreso de contrato e imprimir cualquiera de los contratos almacenados en la base de datos. Este tipo de tabla permite exportar todos los datos a un formato pdf y Excel, así como también imprimir.

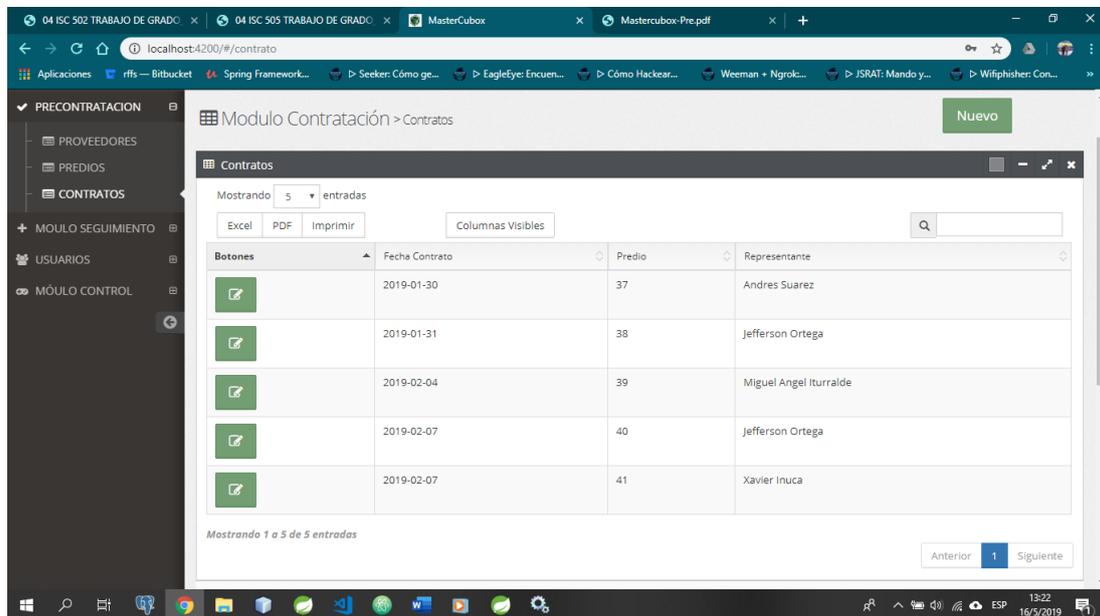


Fig. 16 Listado de Registro de Contratos.
Fuente: Propia

```

import java.util.List;

@CrossOrigin(origins="*", allowCredentials="true", allowedHeaders="*")
@RestController
@RequestMapping(value = "precontratacion")

public class PreContratacionController {

    @Autowired
    private PreContratacionRepository precontratacionRepository;

    @RequestMapping(method = RequestMethod.GET, produces = MediaType.APPLICATION_JSON_VALUE)
    public List<PreContratacion> getContratos() {
        return this.precontratacionRepository.findAll();
    }

    @RequestMapping(value="/{id}",method = RequestMethod.GET, produces = MediaType.APPLICATION_JSON_VALUE)
    public Optional<PreContratacion> getContratoPorId(@PathVariable("id") Integer id) {
        return this.precontratacionRepository.findByIdPrecontratacion(id);
    }

    @RequestMapping(method = RequestMethod.POST, produces = MediaType.APPLICATION_JSON_VALUE, consumes =
    public PreContratacion save(@RequestBody PreContratacion contrato) {
        return this.precontratacionRepository.save(contrato);
    }

    @RequestMapping(method = RequestMethod.PUT, produces = MediaType.APPLICATION_JSON_VALUE, consumes = M
    public void editar(@RequestBody PreContratacion contrato) {
        this.precontratacionRepository.save(contrato);
    }
}

```

Fig. 17 Controller de la entidad Precontratación.
Fuente: Propia

k) Reunión de Retrospectiva

Tabla 22. Plan de mejoras del Sprint 2

RETROSPECTIVA		
Fecha: 05/04/2018		
Asistentes a la reunión: Scrum Máster, Development Team		
¿Qué salió bien en la iteración? (aciertos)	¿Qué no salió bien en la iteración? (errores)	¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)
✓ Mejor manejo de la plataforma GIT	- Ediciones a nivel de Base de datos.	• Mejorar la comunicación con el equipo de trabajo

3.5.1.4. Sprint 3

l) Reunión de Planificación

Fecha de Reunión: 06/05/2018

Asistentes a la reunión: Scrum Máster, Product Owner, Team Development.

Fecha Inicio Sprint: 07/06/2018

Fecha Fin Sprint: 04/07/2018

Objetivo de Sprint: Realizar las pantallas para reporte de datos, integración, instalación y pruebas.

- **Sprint Backlog**

Tabla 23. Sprint 3 Backlog

ID	Historia de usuario
HU6	Reporte de datos, Integración, Instalación y pruebas

Tabla 24. Planificación del Sprint 3

PLANIFICACIÓN DE TRABAJOS DE DESARROLLO SPRINT 3				
Proyecto: Módulo de Precontratación y contratación				
Historia de Usuario	Desarrollador	Fase Desarrollo	Descripción	Tiempo Estimado (Horas)
Gestión de Reportes	Jefferson Ortega	Codificación	Reportes de datos de cada registro.	8

Integración	Jefferson Ortega	Implementación	Unir los módulos del proyecto tanto en servidor como en vista.	10
Instalación	Jefferson Ortega	Instalación	Instalación del proyecto en el servidor de la empresa	6
Pruebas con el usuario	Jefferson Ortega	Pruebas	Verificar si cumple todos los requerimientos planteadas al inicio del proyecto	5
Pruebas con el servidor	Jefferson Ortega	Pruebas	Probar el sistema en el servidor de la empresa	6
Reuniones		Planificación	Planificación	5
		Revisión	Revisión	5
		Revisión	Retrospectiva	3
Tareas no planificadas				
Total				48

m) Reunión de revisión

Con todo el equipo de trabajo se presentó las historias de usuario completamente finalizadas en este sprint.

Una vez finalizada la codificación de todos los módulos se pudo asegurar que cumple con todas las funciones y requerimientos mencionados en el Product Backlog.

Tabla 25. Seguimiento del Sprint 3
SEGUIMIENTO SPRINT 3

Proyecto: **Módulo de Precontratación y contratación**

Historia de Usuario	Desarrollador	Fase Desarrollo	Descripción	Tiempo Estimado (Horas)	Tiempo Real (Horas)	Estado
Gestión de Reportes	Jefferson Ortega	Codificación	Reportes de datos	8	9	Realizado
Integración	Jefferson Ortega	Implementación	Unir los módulos del proyecto tanto en servidor como en vista.	10	11	Realizado
Instalación	Jefferson Ortega	Implementación	Instalación del proyecto en el servidor de la empresa	6	7	Realizado

Pruebas con el usuario	Jefferson Ortega	Pruebas	Verificar si cumple todos los requerimientos o planteadas al inicio del proyecto	5	6	Realizado
Pruebas con el servidor	Jefferson Ortega	Pruebas	Probar el sistema en el servidor de la empresa	6	7	Realizado
Reuniones	TEAM	Planificación	Planificación	5	5	Realizado
	TEAM	Revisión	Revisión	5	5	Realizado
	TEAM	Retrospectiva	Retrospectiva	3	3	Realizado
TOTAL				48	53	

n) Incremento del producto potencialmente entregable

Luego de realizar las pruebas suficientes se hace entrega del sprint con el total cumplimiento.

A continuación, imágenes del diseño de prototipo y resultado final.

- Prototipo de la creación de Reportes, muestra un modelamiento del producto final haciendo referencia a la Historia de Usuario Nro. 6

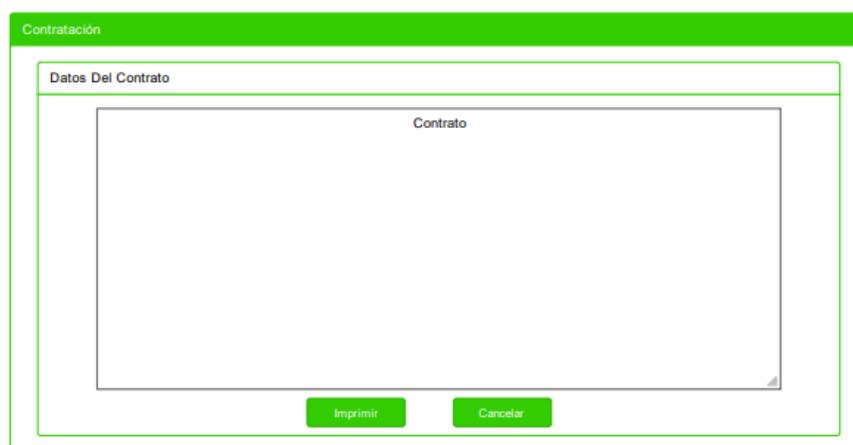


Fig. 18 Prototipo de Reporte de datos.
Fuente: Propia

- Reportes: estas imágenes son de la interfaz web en la cual se aprecia el listado de datos del sistema I-FOOS almacenados en la base de datos.

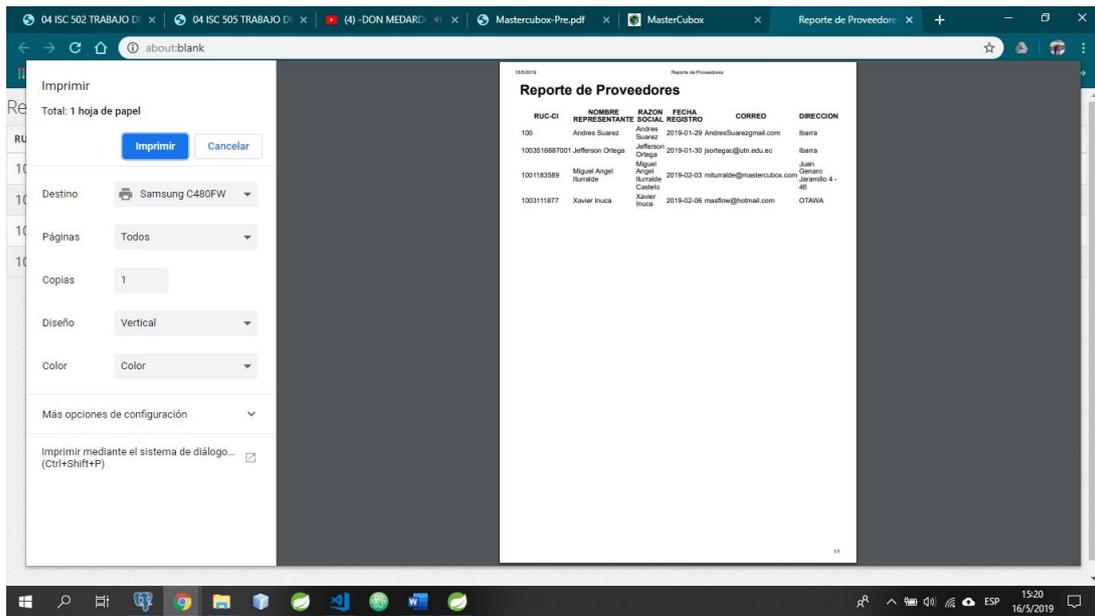


Fig. 19 Reporte de datos Proveedores.
Fuente: Propia

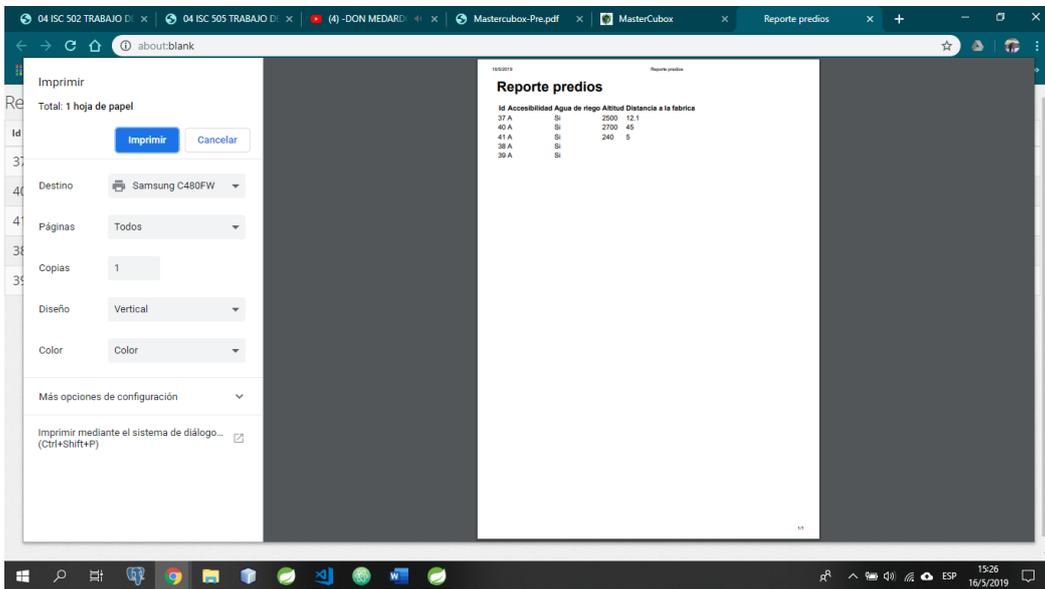


Fig. 20 Reporte de datos Predios.
Fuente: Propia

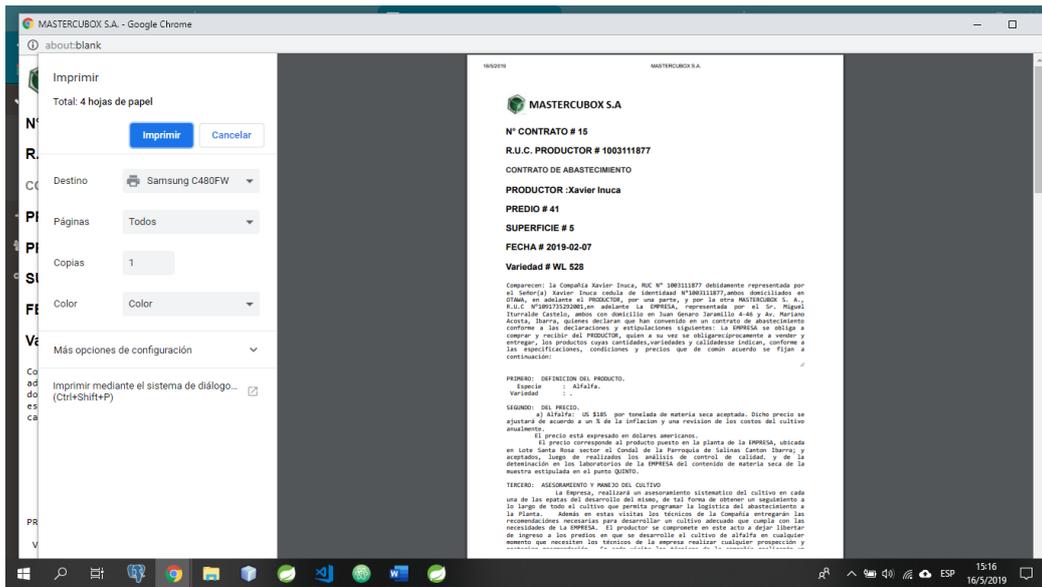


Fig. 21 Reporte de datos Contrato.
Fuente: Propia

- Instalación del servidor de aplicaciones para las pruebas necesarias del backend desarrollado con el framework Spring.

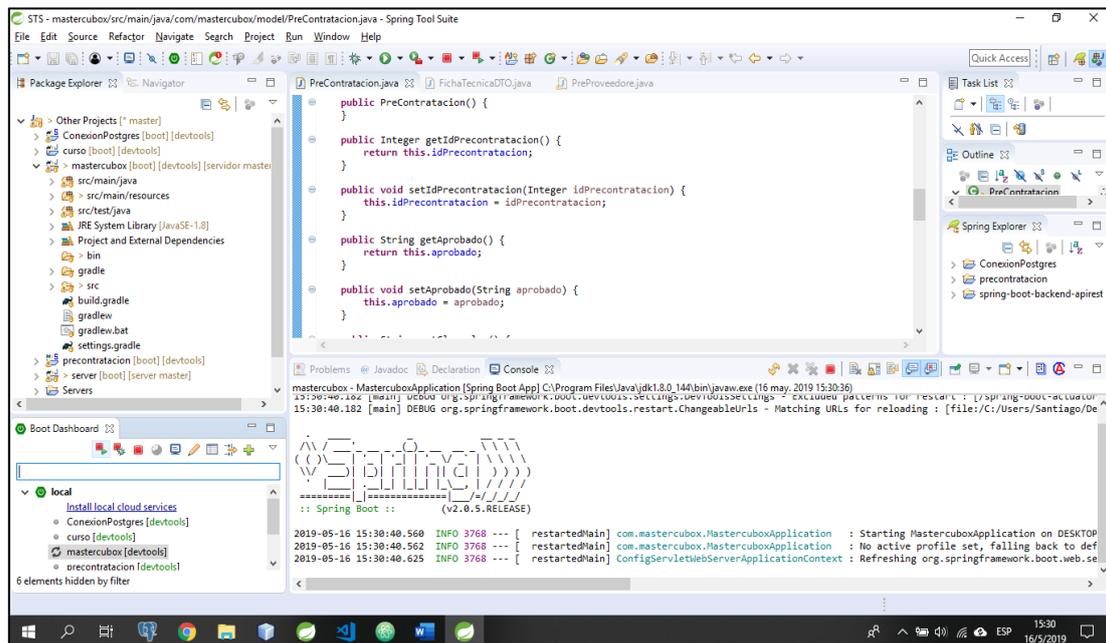


Fig. 22 Instalación del servidor de aplicaciones.
Fuente: Propia

- Pruebas de aceptación realizadas en coordinación con el Product Owner, Scrum Máster, Development Team para la revisión y conformidad del cliente con el cumplimiento total del sistema I-FOODS y se realizaron en un computador como servidor con las siguientes características:
 - 1TB DDR
 - 8GB Ram
 - Procesador Intel Core I

PRUEBAS DE ACEPTACIÓN		
Proyecto de desarrollo de software (Código Fuente)	Registro de proveedores, predios, contratos.	✓
	Gestión de predios pendientes según mejoras y gestión de contratos.	✓
	Reportes	✓
Pruebas funcionales y aceptación del sistema desarrollado		✓
Manuales de usuario		✓
Manuales técnicos		✓
Capacitación al personal encargado		✓

Tabla 26. Pruebas de aceptación
Fuente: Propia

o) Reunión de Retrospectiva

Tabla 27. Plan de mejoras del Sprint 3

RETROSPECTIVA		
Fecha: 05/04/2018		
Asistentes a la reunión: Scrum Máster, Development Team		
¿Qué salió bien en la iteración? (aciertos)	¿Qué no salió bien en la iteración? (errores)	¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)
✓ Correcto funcionamiento del sistema I-FOODS		

3.5.2. Productos Entregados

Gracias al desarrollo del sistema I-FOODS bajo el framework Spring y la implementación de la metodología Scrum se logró cumplir con las metas planteadas en este proyecto, a continuación, se muestran los productos entregados:

PRODUCTO	ESTADO
Proyecto de Desarrollo de Software – Sistema Backend Web para contratación en la empresa Mastercubox S.A. (Código fuente).	ENTREGADO
Pruebas de funcionalidad, operatividad con la tecnología asignada.	ENTREGADO
Manuales técnicos	ENTREGADO
Capacitación al personal técnico de la empresa.	ENTREGADO

Tabla 28. Productos entregados
Fuente: Propia

CONCLUSIONES

- Siguiendo los lineamientos estudiados de la información necesaria de las bases teóricas de spring framework se logró un excelente desarrollo del módulo de precontratación y contratación para la empresa Mastercubox S.A. y se pudo constatar que se tiene un desarrollo más ágil ya que poseen repositorios para la creación de funciones crud como es el repositorio CrudRepository el cual nos permitió generar de manera más fácil funciones de búsqueda filtrando por todos los campos que componen la entidad del proyecto.
- Una vez aplicada la metodología ágil scrum en el desarrollo del sistema I-FOODS se logró una buena creación y sincronización de todos los módulos mediante el manejo adecuado de los sprint.
- La implementación del sistema de precontratación y contratación en la empresa Mastercubox S.A., la mencionada empresa permitió lograr una excelente gestión tanto de sus proveedores como de sus predios para su correcta contratación basándose en la información recolectada por sus técnicos agrícolas.
- Con el estudio del framework Spring como servidor backend en la arquitectura base del sistema I-FOODS se desarrolló el frontend con el framework Angular acoplándose de manera óptima y sin ningún inconveniente.

RECOMENDACIONES

- Realizar el levantamiento de requerimientos juntamente con la persona que tenga mayor conocimiento sobre los procesos que se desean automatizar, y la creación de un acta de constancia firmada por las partes involucradas para evitar cambios no estipulados en el acta ya sea en la fase de desarrollo o en la entrega del producto.
- Diseñar una base de datos centralizada para los diferentes módulos que contendrá el ERP y en lo posible evitar cambios de la base de datos durante el proceso de codificación ya que cualquier tipo de cambio en el diseño nos traería alteraciones en el proceso de desarrollo.
- Mantener una constante comunicación con el equipo de trabajo para poder resolver de mejor manera las dificultades y tener conocimiento sobre los avances que se van realizando en el desarrollo del sistema y no esperar hasta una próxima reunión para comentar los problemas que se han presentado.
- Investigar con mayor énfasis sobre los diferentes repositorios y dependencias para el uso de Spring Framework en especial de Spring Boot ya que nos pueden ahorrar tiempo y facilitar el desarrollo de sistemas bajo esta tecnología.

REFERENCIAS

- Apache. (2019). *Apache Software Foundation*. Obtenido de Apache Software Foundation: tomcat.apache.org
- Bravo, D. M. (2015). *Selección, instalación, configuración y administración de los servidores de transferencia de archivos*. España: Elearning S.L.
- Burgos, R. N. (2016). *Software ERP Análisis y consultoría de software empresarial. 2ª Edición*. Estados Unidos: IT Campus Academy.
- CampusMVP. (25 de Agosto de 2015). *CampusMVP*. Obtenido de CampusMVP: <https://www.campusmvp.es/recursos/post/Desarrollador-web-Front-end-back-end-y-full-stack-Quien-es-quien.aspx>
- Dimes, T. (2015). *Conceptos Básicos de Scrum*. Babelcube Inc.
- IBM. (15 de Agosto de 2015). *IBM Knowledge Center*. Obtenido de IBM Knowledge Center: https://www.ibm.com/support/knowledgecenter/es/ssw_ibm_i_73/rzahg/rzahgebappserv.htm
- Liu, W., Chua, T. J., Larn, J., Yang, F. W., Cai, T. X., & Yin, X. F. (2002). APS, ERP and MES systems integration for semiconductor backend assembly. *7th International Conference on Control, Automation, Robotics and Vision*, 1403-1408.
- Luna, F., Peña, C. M., & Iacono, M. (2018). *PROGRAMADOR WEB Full Stack*. Buenos Aires: Vaccaro Hnos.
- Martínez, A. I. (2015). *Diseño y desarrollo del backend para la explotación de componentes web, empleando Google App Engine*. Madrid.
- Pérez Martínez, E., & Altadill Izura, P. (2018). *Spring 5 Manual Imprescindible*. Madrid: Ediciones Anaya Multimedia.
- Pivotal. (9 de Enero de 2019). *Spring Framework*. Obtenido de Spring Framework: <https://docs.spring.io/spring/docs/5.1.4.RELEASE/spring-framework-reference/overview.html#overview>
- PostgreSQL, S. (2019). *PostgreSQL*. Obtenido de PostgreSQL: www.postgresql.org/about

SIMA. (2019). *SIMA SOFTWARE*. Obtenido de SIMA SOFTWARE:
<https://www.sima.ag/>

Torrealba, P., & Laforge, M. (Octubre de 1998). *Campesinos, Mapas y Computadoras*.

Valle, A., Puerta, A., & Nuñez, R. (2017). *Curso de consultoría TIC. Gestión, Software ERP y CRM*. IT Campus Academy.

Yachay. (Marzo de 2015). *Yachay*. Obtenido de Yachay:
<http://www.yachay.gob.ec/wp-content/uploads/downloads/2015/03/informacion-MASTERCUBOX.pdf>

Zae Ordoñez, M. P., Molina Ríos, J. R., & Redrován Castillo, F. F. (2017). *Administración de base de datos con PostgreSQL*. Alicante: Editorial Área de Innovación y Desarrollo, S.L.