

UNIVERSIDAD TÉCNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

**CARRERA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES**

TEMA:

**ESTUDIO DE LA HERRAMIENTA MAVEN COMO GESTOR DE
PROYECTOS SPRING MVC CON EL CASO DE USO APLICACIÓN PARA
COMERCIO ELECTRÓNICO**

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

AUTORA:

Gina Araceli Ulloa Campoverde

DIRECTOR:

MSc. Mauricio Rea

Ibarra – Ecuador

2019



**UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA**

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

1. IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional determinó la necesidad de disponer los textos completos de forma digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DEL CONTACTO	
CÉDULA DE IDENTIDAD	1002870812
APELLIDOS Y NOMBRES	Ulloa Campoverde Gina Araceli
DIRECCIÓN	Guayas y 13 de Abril
EMAIL	ulloa_gina@live.com
TELÉFONO FIJO	062612399
TELÉFONO MÓVIL	0997516456
DATOS DE LA OBRA	
TÍTULO	“ESTUDIO DE LA HERRAMIENTA MAVEN COMO GESTOR DE PROYECTOS SPRING MVC CON EL CASO DE USO APLICACIÓN PARA COMERCIO ELECTRÓNICO”
AUTOR	Ulloa Campoverde Gina Aracelly
FECHA	13 de febrero del 2019
PROGRAMA	<input checked="" type="checkbox"/> Pregrado <input type="checkbox"/> Postgrado
TÍTULO POR EL QUE OPTA	Ingeniería en Sistemas Computacionales
DIRECTOR	Msc. Mauricio Rea

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Gina Araceli Ulloa Campoverde, con cédula de ciudadanía N° 1002870812, en calidad de autora y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago la entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y el uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior artículo 144.

3. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y la desarrollo, sin violar los 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 la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 13 días del mes de febrero del 2019

LA AUTORA:

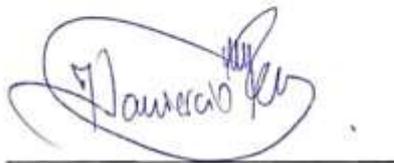

.....

Nombre: Gina Araceli Ulloa Campoverde

C. I.: 1002870812

CERTIFICACIÓN DEL DIRECTOR

Como Director del presente Trabajo de Grado, cuyo tema es: **“ESTUDIO DE LA HERRAMIENTA MAVEN COMO GESTOR DE PROYECTOS SPRING MVC CON EL CASO DE USO APLICACIÓN PARA COMERCIO ELECTRÓNICO”**, Certifico que ha sido realizado en su totalidad por la estudiante Gina Araceli Ulloa Campoverde, demostrando interés y responsabilidad en el desarrollo del trabajo de grado; particular que pongo en su conocimiento en honor a la verdad.

A handwritten signature in blue ink, appearing to read "Mauricio Rea", is written over a horizontal line.

MSc. Mauricio Rea

DIRECTOR DE TESIS



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

Ibarra, a los 13 días del mes de febrero del 2019

ACTA ENTREGA RECEPCIÓN

TEMA: “ESTUDIO DE LA HERRAMIENTA MAVEN COMO GESTOR DE PROYECTOS SPRING MVC CON EL CASO DE USO APLICACIÓN PARA COMERCIO ELECTRÓNICO”.

En las instalaciones de Bazar y Novedades Guayas, el día 09 de Febrero del 2019, la Carrera de Ingeniería en Sistemas Computacionales hace la entrega del “Sistema de Ventas Online”, a la Sra. Bertha Campoverde Gerente Propietaria del mencionado Bazar.

El sistema ofrece las funcionalidades de administración de información de los usuarios y productos que se gestionan en el Bazar Guayas, la cual incluye un carrito de compras online, el proyecto fue desarrollado de acuerdo a los requerimientos de la Gerente Propietaria del Bazar Guayas, el desarrollo del proyecto lo realizó la estudiante Gina Ulloa, con la tutoría del Ing. Mauricio Rea.

Productos entregados:

- Código fuente del Sistema de Ventas Online.
- Pruebas de funcionamiento con las herramientas y software utilizados.
- Manual de usuario.
- Capacitación de funcionamiento del sistema al encargado de manejar el sistema.

Atentamente,

Gina Ulloa

Estudiante – Tesista

MSc. Mauricio Rea

Tutor de Tesis

Sra. Bertha Campoverde

Gerente – Propietaria Bazar y Novedades Guayas

DEDICATORIA

El presente trabajo de investigación está dedicado a las personas que son el pilar fundamental en mi vida: mi Madre, mi ejemplo a seguir, mi Padre que desde el cielo espero esté orgulloso de la hija que formó, mi esposo Roger quien ha sido un gran compañero de vida, mis hijos Martin e Isabella quienes han sido la fuerza para cumplir mis sueños y metas y a toda mi familia que con su apoyo incondicional, sus muestras de ánimo y cariño han permitido la culminación de esta etapa estudiantil.

A mis amigos y amigas que con su sincera amistad y sus constantes consejos me han permitido salir adelante ante las adversidades.

Gina Ulloa

AGRADECIMIENTO

Mi agradecimiento primeramente a Dios quien guía mis pasos cada día y es mi fortaleza para enfrentar las adversidades y dificultades.

A la Facultad de Ingeniería en Ciencias Aplicadas por permitirme adquirir conocimientos importantes que sirven de guía en mi desarrollo como profesional.

A mi familia por su apoyo constante e incondicional brindándome la confianza necesaria para cumplir con mis metas propuestas.

A los docentes de la Carrera de Ingeniería en Sistemas Computacionales por impartir sus conocimientos y de manera especial al Ing. Mauricio Rea Msc., por su apoyo incondicional en el desarrollo del presente trabajo de grado.

Gina Ulloa

RESUMEN

El desarrollo del presente trabajo de investigación tuvo como objetivo realizar el estudio de Maven como gestor de proyectos Spring MVC, como resultado de esta investigación se desarrolló una aplicación de ventas online mediante un catálogo electrónico que será aplicado en el Bazar y Novedades Guayas.

Se cumplió con el objetivo planteado presentando documentación técnica tanto de Maven, Spring MVC y Spring Security, con la finalidad de brindar una guía para el usuario que quiera ampliar su conocimiento en estas herramientas y no encuentre dificultades al momento de elegir las, es importante mencionar que se utilizó la metodología de estudio Scrum en el desarrollo del proyecto ya que es una metodología ágil, la cual nos permite entregas continuas mediante iteraciones para lograr el éxito esperado en el proyecto.

Se aplicó una encuesta a un grupo de treinta profesionales que se dedican al desarrollo del software y que cuentan con la experiencia necesaria en esta área, esto con la finalidad de obtener criterios sobre el conocimiento de las herramientas de gestión de proyectos; con los resultados obtenidos se pudo determinar que actualmente la mayoría de empresas utilizan al menos una herramienta para la gestión de sus proyectos, más de la mitad utilizan a Maven en cuanto a proyectos Java se refiere ya que simplifica sus procesos de desarrollo en todas las fases, desde la construcción hasta el despliegue, facilitando la modularidad y gestión de dependencias y por último se encuentran las conclusiones y recomendaciones resultado del presente estudio.

ABSTRACT

The development of the present research work had as objective to carry the Maven studio as a Spring MVC project, as a result of this research an online sales application was developed through an electronic catalog that will be applied in the Bazar and Guayas News.

The objective was met by presenting technical documentation of both Maven, Spring MVC and Spring Security, in order to provide a guide for the user that wants to expand their knowledge in these tools and not encountering difficulties when choosing them, it is important to mention that the Scrum study methodology was used in the development of the project since it is an agile methodology, which allows us to deliver continuously through iterations to achieve the expected success in the project.

A survey was applied to a group of thirty professionals that are dedicated to software development and who have the necessary experience in this area, this with the purpose of obtaining criteria on the knowledge of the management tools of projects; with the results obtained it was possible to determine that currently most of the companies use Maven, in terms of Java projects, it refers to the fact that it simplifies its development processes in all phases, from construction to deployment, facilitating the modularity and management of dependencies and finally, conclusions and recommendations resulting from the present.

ÍNDICE DE CONTENIDOS

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE.....	ii
CERTIFICACIÓN DEL DIRECTOR	iv
ACTA ENTREGA RECEPCIÓN	v
DEDICATORIA.....	v
AGRADECIMIENTO	vii
RESUMEN	viii
ÍNDICE DE CONTENIDOS.....	x
ÍNDICE DE FIGURAS	xiii
ÍNDICE DE GRÁFICOS	xv
ÍNDICE DE TABLAS.....	xvi
INTRODUCCION.....	1
SITUACION ACTUAL DE LA MICROEMPRESA BAZAR Y NOVEDADES GUAYAS	1
PRESENTACIÓN DE LA EMPRESA	1
UBICACIÓN FÍSICA	1
DEFINICIÓN DEL PROBLEMA.....	2
ANTECEDENTES	3
OBJETIVO GENERAL	4
OBJETIVOS ESPECÍFICOS	4
ALCANCE	4
JUSTIFICACIÓN.....	5
Justificación Económica	5
Justificación Técnica.....	6
BENEFICIARIOS	6
Contexto.....	6
CAPÍTULO I.....	7
MARCO TEÓRICO	7
1.1 MAVEN COMO GESTOR DE PROYECTOS DE SOFTWARE SPRING MVC ...	7
1.1.1 Introducción	7
1.1.2 Instalación y configuración de Maven.....	8
1.1.2.1 Requisitos del sistema	8
1.1.2.2 Configuración de Apache Maven.....	9

1.1.3 Características	11
1.1.4 Ventajas de Maven.....	11
1.1.5 Documentación técnica.....	12
1.2 SPRING MVC.....	30
1.2.1 Introducción	30
1.2.2 Características	32
1.2.3 Patrón de Diseño MVC.....	32
1.2.3.1 Modelo.....	34
1.2.3.2 Vista.....	34
1.2.3.3 Controlador.....	35
1.2.4 Documentación técnica.....	35
1.2.5 Integración de Maven con Spring MVC.....	45
1.3 SPRING SECURITY	45
1.3.1 Características	46
1.3.2 Documentación técnica.....	46
1.4 SERVICIOS DE PAGO ONLINE MÁS SEGUROS EN INTERNET.....	54
CAPITULO II.....	55
DESARROLLO.....	55
2.1 ESPECIFICACIONES TÉCNICAS.....	55
2.1.1 Descripción del proyecto	55
2.1.2 Aspectos funcionales	56
2.1.3 Aspectos operativos	56
2.1.4 Aspectos técnicos.....	56
2.1.5 Aspectos del proyecto.....	56
2.2 PLANIFICACIÓN	57
2.2.1 Plan de desarrollo de software	57
2.2.2 Plan de gestión de requerimientos	57
2.2.3 Plan de aseguramiento de la calidad del software	58
2.2.4 Métricas de calidad de software.....	58
2.2.5 Plan de etapas de entregas	59
2.3 DESARROLLO.....	60
2.3.1 Metodología de desarrollo Scrum.....	60
2.3.2 Características de Scrum.....	62
2.3.3 Principios de Scrum	62
2.3.4 Visión general del proceso.....	62

2.3.5 Fases de la metodología de desarrollo Scrum.....	63
2.3.5.1 Fase 1: Definir el backlog del proyecto.....	63
2.3.5.2 Fase 2: Planificación del sprint.....	66
2.3.5.3 Fase 3: Scrum diario	69
2.3.5.4 Fase 4: Revisión del sprint	70
2.3.5.5 Fase 5: Retrospectiva del sprint.....	72
2.4 ESTABILIZACIÓN DEL PROYECTO	73
2.4.1 Pruebas de software para estabilización del proyecto.....	73
2.5 IMPLEMENTACIÓN DEL PROYECTO	74
2.7 MANUAL DE USUARIO.....	75
CAPITULO III	87
RESULTADOS	87
3.1 VALIDACIÓN	87
3.1.1 Tabulación y análisis de resultados de la encuesta aplicada.....	88
CONCLUSIONES.....	101
RECOMENDACIONES	102
BIBLIOGRAFÍA	103
ANEXOS	104
ANEXO 1: MANUAL TÉCNICO	105
ANEXO 2: ENCUESTA DE SATISFACCIÓN DE USUARIOS	123
ANEXO 3: CERTIFICACIÓN DE LA EMPRESA	127

ÍNDICE DE FIGURAS

Figura N° 1 Arquitectura Tecnológica	5
Figura N° 2 Ciclo de trabajo de Maven	7
Figura N° 3 Extracción del archivo de Maven	9
Figura N° 4 Archivos ejecutables para configuración de Apache Maven	9
Figura N° 5 Agregación de las variables de entorno	10
Figura N° 6 Definición de las variables de entorno	10
Figura N° 7 Ruta de configuración de Variables configuradas	10
Figura N° 8 Información general de un proyecto Maven	17
Figura N° 9 Repositorio central de Maven	17
Figura N° 10 Dependencia Spring JPA del repositorio central de Maven	18
Figura N° 11 Versiones de la dependencia Spring JPA	18
Figura N° 12 Código para añadir al fichero POM	19
Figura N° 13 Proyecto Maven ya incluyendo las dependencias	19
Figura N° 14 Agregando repositorio interno en Maven	20
Figura N° 15 Ejemplo de plugin	22
Figura N° 16 Ejemplo de plugin con una fase asociada	23
Figura N° 17 Fases del ciclo de vida de construcción	24
Figura N° 18 Ejemplo de pase de construcción y objetivos	25
Figura N° 19 Configuración de POM.XML padre	26
Figura N° 20 Configuración POM.XML hijo	27
Figura N° 21 Configuración variables	28
Figura N° 22 Estructura de directorio de un proyecto web con Maven	30
Figura N° 23 Capas de una aplicación Spring MVC	31
Figura N° 24 Funcionamiento del Patrón de diseño MVC	32
Figura N° 25 Ciclo de vida de un proceso en Spring MVC	40
Figura N° 26 Configuración de un proyecto Spring MVC	42
Figura N° 27 Configuración de DispatcherServer	43
Figura N° 28 Librerías utilizadas por Spring MVC	43
Figura N° 29 Página de bienvenida de Spring MVC	44
Figura N° 30 Flujo de Trabajo de Scrum	61
Figura N° 31 Pantalla principal del proyecto	71
Figura N° 32 Pantalla final del proyecto finalizado	71
Figura N° 33 Página principal de la tienda Online	75
Figura N° 34 Descripción de productos de la tienda Online	76
Figura N° 35 Pantalla de logueo para ingresar a la tienda Online	77
Figura N° 36 Pantalla de registro de un nuevo cliente	77
Figura N° 37 Página principal con las opciones del rol Administrador	78
Figura N° 38 Página de administración de información	79
Figura N° 39 Página de edición de información	79
Figura N° 40 Página de gestión de productos	80
Figura N° 41 Página de ingreso de productos	81
Figura N° 42 Página de edición de productos	81
Figura N° 43 Página de edición de productos	82
Figura N° 44 Página de edición de productos	82

Figura N° 45 Página de stock de productos.....	83
Figura N° 46 Página principal para el usuario comprador	84
Figura N° 47 Página de stock de productos.....	85
Figura N° 48 Página de carrito de compras	85
Figura N° 49 Página de administración de información.....	86
Figura N° 50 Página de edición de información.....	86

ÍNDICE DE GRÁFICOS

Gráfico N° 1 Causas y efectos sobre el conocimiento de MVC	2
Gráfico N° 2 Diagrama Burndown para el sprint backlog 1	69
Gráfico N° 3 Diagrama Burndown para el sprint backlog 11	70
Gráfico N° 4 Herramienta para gestión de proyectos de software.....	88
Gráfico N° 5 Uso de herramienta para gestión de proyectos de software	89
Gráfico N° 6 Uso de herramienta Maven	90
Gráfico N° 7 Uso de herramienta Maven sin experiencia	91
Gráfico N° 8 Reducción de tiempos con el uso de la herramienta Maven	92
Gráfico N° 9 Gestor de proyectos Maven.....	93
Gráfico N° 10 Maven como mejor gestor de proyectos	94
Gráfico N° 11 Maven como mejor gestor de proyectos	95
Gráfico N° 12 Grado de satisfacción de Maven	96
Gráfico N° 13 Gestores de proyectos conocidos	97
Gráfico N° 14 Gestores de proyectos utilizados	98
Gráfico N° 15 Tipos de proyectos generados con Maven	99
Gráfico N° 16 Maven como tema de estudio en la preparación académica	100

ÍNDICE DE TABLAS

Tabla N° 1 Principales arquetipos de Maven.....	29
Tabla N° 2 Métricas y características	59
Tabla N° 3 Definición Pila del producto.....	65
Tabla N° 4 Planificación del Sprint	67
Tabla N° 5 Sprint backlog definidos en la primera reunión de planificación.....	68
Tabla N° 6 Estado final de las tareas del historial de usuario del proyecto.....	72
Tabla N° 7 Herramienta para gestión de proyectos de software.....	88
Tabla N° 8 Uso de herramienta para gestión de proyectos de software	89
Tabla N° 9 Uso de herramienta Maven.....	90
Tabla N° 10 Uso de herramienta Maven sin experiencia	91
Tabla N° 11 Reducción de tiempos con el uso de la herramienta Maven	92
Tabla N° 12 Gestor de proyectos Maven	93
Tabla N° 13 Maven como mejor gestor de proyectos.....	94
Tabla N° 14 Maven como gestor de proyectos	95
Tabla N° 15 Grado de satisfacción de Maven	96
Tabla N° 16 Gestores de proyectos conocidos	97
Tabla N° 17 Gestores de proyectos utilizados	98
Tabla N° 18 Tipos de proyectos generados con Maven	99
Tabla N° 19 Maven como tema de estudio en la preparación académica	100

INTRODUCCIÓN

SITUACIÓN ACTUAL DE LA MICROEMPRESA BAZAR Y NOVEDADES GUAYAS

PRESENTACIÓN DE LA EMPRESA

Bazar y novedades Guayas es una microempresa que lleva en el mercado 40 años aproximadamente, está dedicada a la comercialización de artículos de papelería, bazar, cosméticos, ropa y ferretería; esta microempresa familiar durante todo este tiempo ha venido manejando las ventas de modo tradicional, pero al querer aumentar su nivel de ingresos tiene la necesidad de actualizar su manera tradicional de ventas y adoptar nuevas formas de comercialización como es el comercio electrónico.

Una de las ventajas de implementar el comercio electrónico en bazar Guayas es que el cliente puede realizar sus compras a cualquier hora del día y el proveedor tiene la obligación de revisar el buzón de compras para tener listos los pedidos y así de esta manera evitar demoras con el cliente, realizar más ventas, ampliar la cartera de clientes y aumentar el nivel de ingresos para la microempresa, esta innovación tecnológica hace que se incrementen varias oportunidades tanto para la microempresa como para la propietaria de la misma.

En la actualidad el comercio electrónico es una tecnología muy habitual para empresas que quieren innovar sus procesos de facturación y ventas, en Bazar Guayas es necesario incrementar el comercio electrónico porque hay clientes que requieren realizar sus compras de manera rápida y segura sin tener que recurrir al bazar hacer fila o esperar para que le despachen su mercadería.

UBICACIÓN FÍSICA

La microempresa bazar y novedades Guayas está ubicada en la provincia de Imbabura, cantón Ibarra, sector Alpachaca, en las calles Machala y Puyo, en esta microempresa hay un empleado que hace los trabajos de bodega y cuenta con la gerente propietaria para realizar las ventas y cobros de la mercadería.

DEFINICIÓN DEL PROBLEMA

Existen profesionales y desarrolladores que desconocen sobre algunos frameworks de desarrollo que han ido surgiendo a través del tiempo, muchas empresas en el Ecuador y en otros países están usando estos frameworks para el desarrollo y gestión de sus proyectos de software, uno de los principales es Spring MVC, Maven, entre otros, es por esta razón que se hace necesario realizar una investigación sobre estos frameworks para poder ofertar estos servicios en el mercado competitivo, además permitirá utilizar una herramienta que gestione los proyectos; de esta manera se ha propuesto como objetivo principal realizar un estudio profundo para elaborar una guía completa de consulta para desarrolladores de software que se inician en el uso de las herramientas propuestas en el presente estudio.

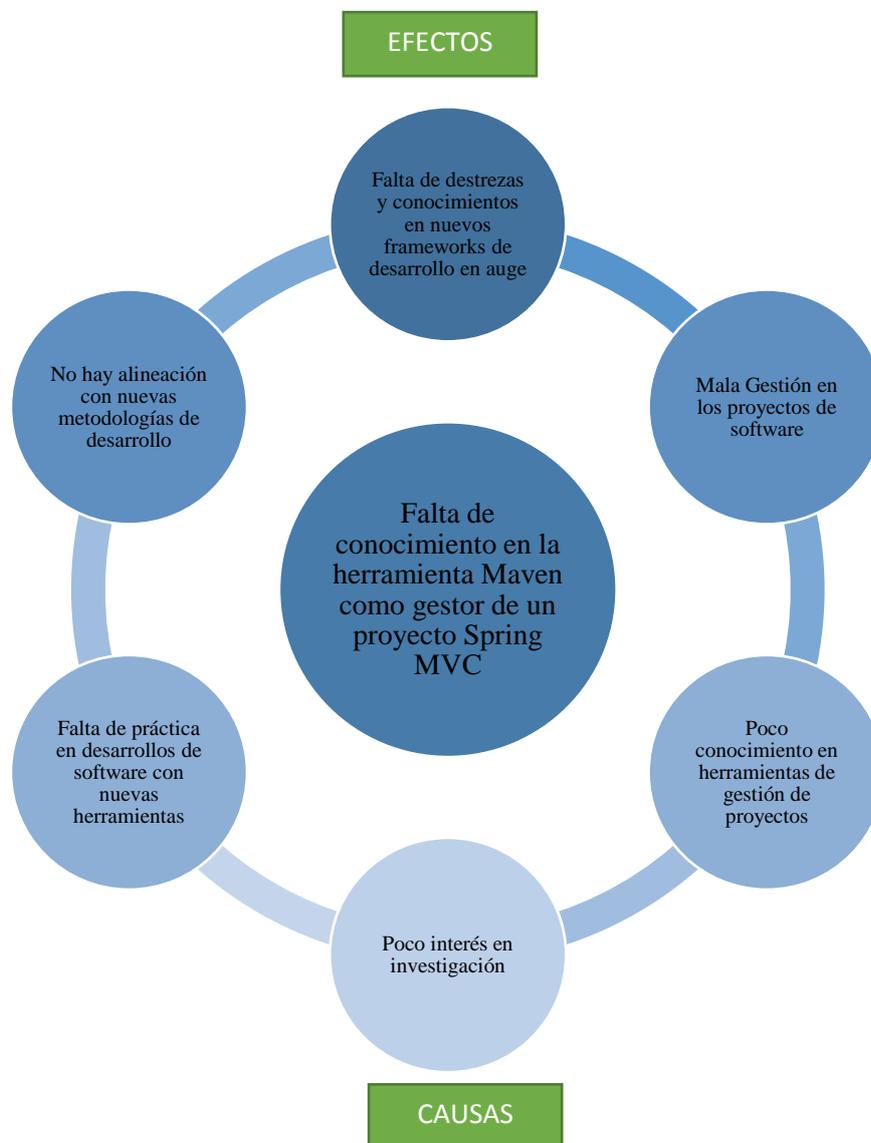


Gráfico N° 1 Causas y efectos sobre el conocimiento de MVC
Elaborado por: Gina Ulloa

ANTECEDENTES

El desarrollo de aplicaciones Web ha ido evolucionando en el mundo empresarial y su forma de hacer negocios en todo el mundo, no obstante para poder llevar a cabo estos avances es necesario adaptarse a nuevas tecnologías, frameworks y herramientas para el desarrollo de software; el objetivo de los frameworks es hacer que el programador se centre en el verdadero problema y no se preocupe por implementar funcionalidades que son de uso común en muchas aplicaciones y procesos que ya han sido desarrollados y probados, lo cual permitirá que la aplicación sea más robusta.

Al utilizar una herramienta que ayude a gestionar los proyectos también permitirá la homogeneización de los mismos sin importar su tamaño o complejidad, así como también la automatización de un buen número de tareas, eliminando fallos humanos y agilizando los procesos de desarrollo, procesos que tradicionalmente costaban al desarrollador tanto tiempo y esfuerzo realizarlos. Hay que tener en cuenta la progresión de Internet, ya que cada vez es más frecuente que las personas realicen transacciones, pagos, compras entre otras cosas, es ahí donde la necesidad se vuelve una oportunidad para los desarrolladores.

En términos generales, el comercio electrónico es la compra-venta de productos o servicios a través de Internet y redes informáticas, esto supone una revolución en la manera en la que las empresas se comunican y desarrollan su actividad comercial. Una de las ventajas de utilizar herramientas que permitan a una empresa mantenerse tecnológicamente actualizada es que siempre genera mayor rentabilidad ya que permite ofrecer tanto productos como servicios dentro y fuera de la ciudad sin necesidad siquiera de tener un lugar físico para la venta.

Toda empresa busca aumentar sus ventas y mejorar sus ganancias, pero al tener un mercado tan competitivo y extenso se vuelve difícil ofrecer productos o servicios y abarcar un mercado más amplio donde la demanda sea aún mayor, esto se vuelve un bloqueante importante para cualquier emprendedor ya que el tener que esperar que los clientes sean los que tengan que movilizarse al lugar de establecimiento, además del ahorro en tiempo y desplazamiento minimiza su mercado simplemente a las personas ubicadas geográficamente cerca.

Adicionalmente las ventas por internet posibilitan tener un horario comercial de 24 horas al día, todos los días del año gracias al entorno online, así que los clientes podrán visitar la página y acceder a los productos disponibles sin problemas, las personas cada vez confían en esta forma de adquirir productos y servicios; además, se han creado métodos de compra que generan confianza entre los usuarios, es por esta razón que las empresas reacias a introducirse en el e-commerce pueden quedarse atrás en relación a sus competidores.

OBJETIVO GENERAL

Realizar el estudio de la herramienta Maven como gestor de proyectos Spring MVC mediante el desarrollo de una aplicación de Comercio Electrónico.

OBJETIVOS ESPECÍFICOS

- Elaborar un estudio sobre la herramienta Maven como gestor de proyectos de software, el framework Spring MVC basado en el patrón Modelo – Vista – Controlador y el framework Spring Security para gestión de la seguridad de la aplicación.
- Desarrollar la aplicación de Comercio Electrónico con la herramienta Maven como gestor de proyectos Spring MVC.
- Realizar la validación de resultados.

ALCANCE

Para el estudio se utilizará la herramienta Maven como gestor de proyectos Spring MVC y se desarrollará una aplicación de Comercio Electrónico para la empresa Bazar y Novedades Guayas.

- Se presentará documentación técnica que servirá como guía para las personas interesadas en Maven como gestor de proyectos de software.
- Se presentará documentación técnica que servirá como guía para las personas interesadas en el framework Spring MVC basado en el patrón Modelo – Vista – Controlador.
- Se presentará documentación técnica que servirá como guía para las personas interesadas en el framework Spring Security para gestión de la seguridad de la aplicación.

El aplicativo tendrá el módulo de front end que es una tienda virtual y gestión del catálogo de productos e inventario, así como también se utilizará un servicio externo para el pago seguro online, el mismo que es parte de la investigación. El IDE de desarrollo será Spring Tool Suite, la base de Datos será Postgresql.

Para el desarrollo del presente proyecto se usará en su mayoría herramientas de software libre para la reducción de costos, evitando de esta manera grandes requerimientos de hardware y costos de licenciamiento, a continuación se mostrará una idea de la arquitectura a utilizarse:

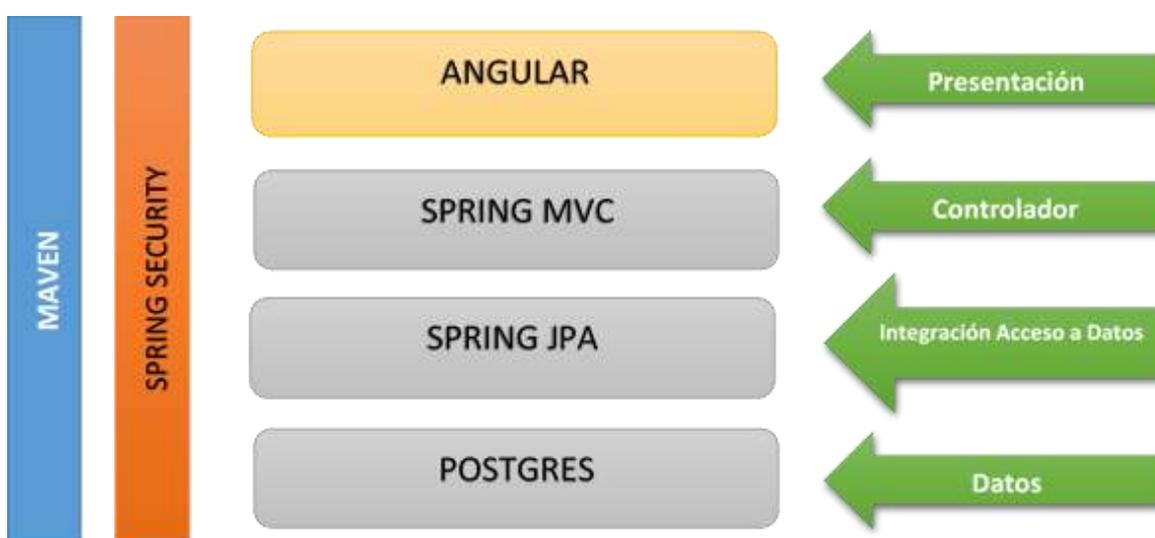


Figura N° 1 Arquitectura Tecnológica
Elaborado por: Gina Ulloa

JUSTIFICACIÓN

Justificación Económica

La utilización de herramientas de software libre y software gratuito permite que los clientes estén más abiertos a contratar a empresas de desarrollo para solventar sus requerimientos, sin que esto signifique que los sistemas no sean robustos y seguros, evitando así de grandes requerimientos de hardware y software.

Para los desarrolladores evita pérdida de tiempo en el desarrollo, reutilización de código, lo que también significa reducción de costos para el cliente. Además, el manejo del comercio electrónico para una empresa que quiere ampliar su mercado y mejorar sus ganancias sin

incurrir en el costo de abrir nuevas sucursales es la mejor solución ya que puede tener ventas las 24 horas del día los 365 días del año.

Justificación Técnica

En la actualidad existen varias empresas importantes tanto dentro y fuera del país que tienen como base estas herramientas y frameworks de desarrollo por lo que se ha vuelto un buen referente para el estudio de las mismas, además en un estudio realizado por Rebellabs el framework Spring MVC es uno de los más importantes de Java, de manera que los desarrolladores tengan un buen nivel de conocimientos respecto al tema y ser profesionalmente competitivos.

Las herramientas y frameworks en estudio nos permitirán elaborar una amplia posibilidad de desarrollos pues son las plataformas para nuevas implementaciones, además conocer una herramienta que permitirá gestionar el ciclo de vida de construcción de las aplicaciones, esto evita pérdida de tiempo en el desarrollo, reutilización de código, utilización de plantillas, manejo de librerías entre muchas otras utilidades. Las herramientas usadas en su mayoría son software libre lo que permitirá tener un amplio soporte e información para la investigación de las mismas.

BENEFICIARIOS

Contexto

Existen proyectos de titulación que se realizaron en diferentes universidades acerca del framework Spring en el desarrollo de Aplicaciones Web, pero no existe ningún proyecto de Maven como gestor de proyectos Spring MVC, como por ejemplo existen dos temas que sobresalen y tienen algo de similitud con el tema investigado:

- Universidad Politécnica de Madrid. Desarrollo de una aplicación Web con Spring Framework para un gestor de un recetario. Desarrollada por Federico Julián Gutiérrez Faraoni en el año 2014.
- Universidad de San Carlos de Guatemala, Spring Framework como base de una arquitectura para aplicaciones web. Desarrollada por Nelson Moris Larin Reyes en el año 2011.

CAPÍTULO I

MARCO TEÓRICO

1.1 MAVEN COMO GESTOR DE PROYECTOS DE SOFTWARE SPRING MVC

1.1.1 Introducción

Maven es una herramienta que usa una arquitectura diseñada en plugins, puede ejecutarse nativamente en el computador o también puede integrarse en la mayoría de IDEs¹, una de las características importantes de Maven es que tiene la capacidad de trabajar en red, permite la creación y gestión de software con librerías incluidas, compilación y empaquetado del código dentro de la estructura del JAR, los proyectos desarrollados en Maven se identifican en un archivo XML, el motor que Maven tiene incluido en su núcleo permite descargar los plugins necesarios de un repositorio.

A continuación se mostrará un gráfico sobre la forma de cómo trabaja de Maven:



Figura N° 2 Ciclo de trabajo de Maven
Elaborado por: Gina Ulloa

Una vez que el proyecto ha sido desarrollado pasa a la fase de validación y compilación pero si surgen errores regresa al área de desarrollo para las respectivas correcciones, si pasa la etapa de validación pasa a la fase de pruebas pero si existen errores en esta fase nuevamente regresa al área de desarrollo, una vez corregido los errores el proyecto se empaqueta en un archivo WAR y se despliega en el servidor de pruebas y pasa nuevamente a la fase de pruebas de integración y aquí también pueden surgir errores de ser el caso regresará nuevamente al área de desarrollo, una vez solventado los errores el archivo WAR sube al servidor de producción, una vez terminado este proceso se procederá a la creación de la página de documentación.

¹ **IDE:** Entorno de Desarrollo Integrado, es un programa informático que brinda servicios completos al desarrollador de software.

De acuerdo a (**Sonatype, 2008**) define a Maven de la siguiente manera: Maven incorpora el concepto al proporcionar desarrollos por defecto razonables para los proyectos. Maven compila el código fuente y luego crear un archivo JAR distribuible. Si bien esto puede parecer trivial, considere el hecho de que la mayoría de las compilaciones deben definir las ubicaciones de estos directorios en cada sub-proyecto. Sin embargo la configuración de Maven va más allá de las simples ubicaciones de directorios; los complementos principales de Maven aplican un conjunto común de aplicaciones para compilar código fuente, empaquetar distribuciones, generar sitios web y muchos otros procesos. Tiene un ciclo de vida definido y un conjunto de complementos comunes que saben cómo crear librerías y aplicaciones web. (pag. 3)

1.1.2 Instalación y configuración de Maven

(**Sonatype, 2008**) menciona que: “Existen grandes diferencias entre los sistemas operativos como Mac OSX y Microsoft Windows y existen diferencias sutiles entre las diferentes versiones de Windows. Afortunadamente, el proceso de instalación de Maven en todos estos sistemas operativos es relativamente sencillo y fácil”. (pag. 14)

Para poder empezar a trabajar con Maven es necesario preparar el ambiente de trabajo, es decir, se comenzará con la descarga de Maven 3.3.9 que está disponible para Windows (.zip) o para Linux (.tar) que es una versión estable, para esto debemos descargarnos el paquete de Maven de <http://maven.apache.org/download.html#installation> (es recomendable elegir un mirror para la descarga), primeramente se procederá a descomprimir el archivo de Maven, para la instalación es necesario extraer el archivo anteriormente descargado pero primero hay que verificar los siguientes requerimientos:

1.1.2.1 Requisitos del sistema

Capacidad en disco: Maven requiere 10MB para su instalación, además necesitará de 500MB para instalar su repositorio local en el equipo.

Capacidad de memoria: No existen requerimientos mínimos.

JDK (Kit de desarrollo Java): Maven requiere de la plataforma JDK 1.7 o una versión superior para su respectiva ejecución.

Sistema operativo que soporta: No existen requerimientos mínimos en el sistema operativo para la instalación de Maven.

En la siguiente figura se muestra el proceso de extracción del archivo Maven:

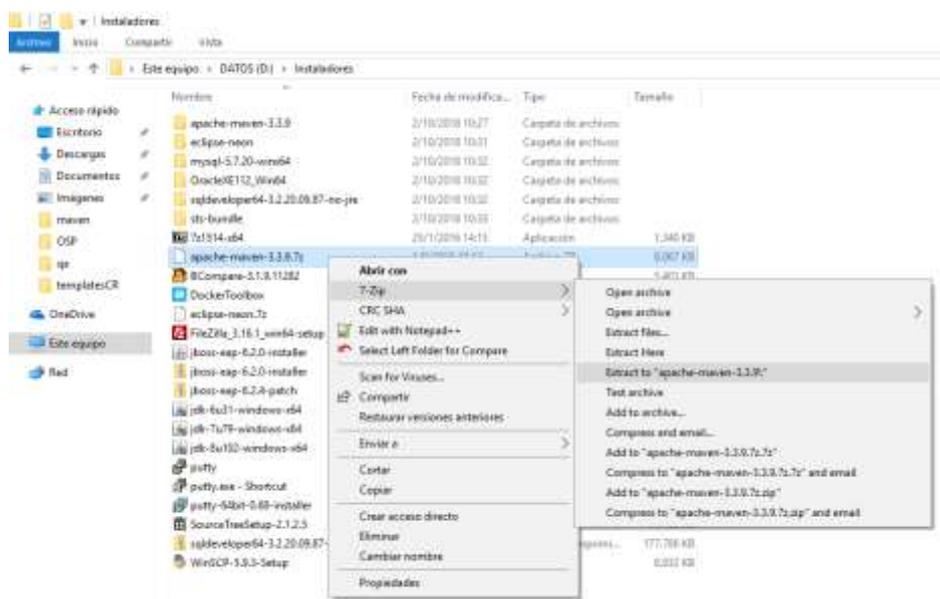


Figura N° 3 Extracción del archivo de Maven
Elaborado por: Gina Ulloa

1.1.2.2 Configuración de Apache Maven

Primeramente se deberá crear una carpeta denominada bin, a continuación se deberá configurar la variable de entorno JAVA_HOME; para el almacenamiento de los proyectos creados se lo hará en la variable de entorno MAVEN_OPTS; en la carpeta M2_HOME se encuentran los archivos de configuración para el uso de Maven, en los archivos maven.config y extensions.xml se encuentran las configuraciones para ejecutar los proyectos en Maven; una vez que se tiene listo el archivo en el computador se procederá a definir la ruta asignada:

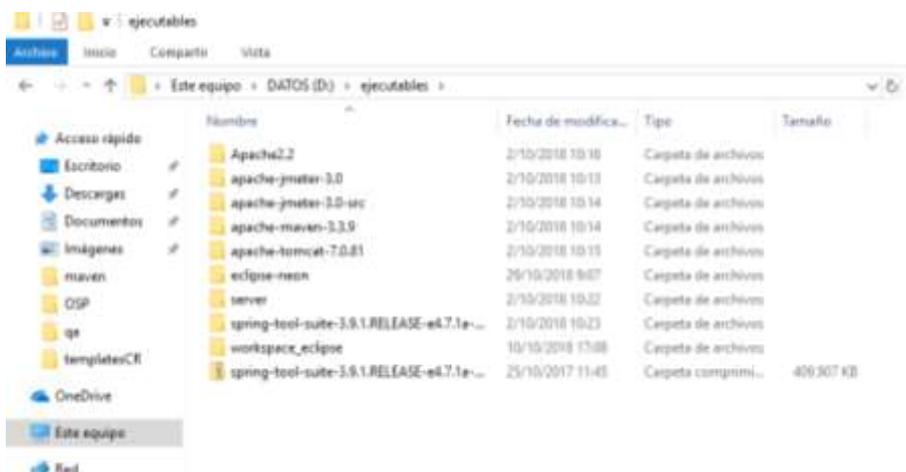


Figura N° 4 Archivos ejecutables para configuración de Apache Maven
Elaborado por: Gina Ulloa

- A continuación, se procederá agregar las variables del sistema, el path de Maven y de la plataforma Java, como se observa en el siguiente gráfico:



Figura N° 5 Agregación de las variables de entorno
Elaborado por: Gina Ulloa

- Luego se procederá a definir las respectivas variables de entorno:

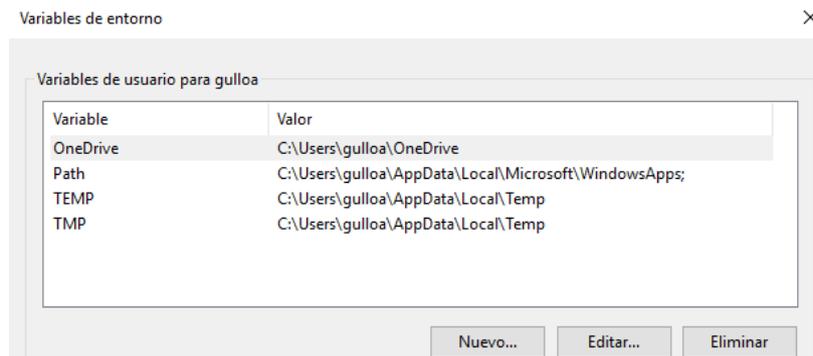


Figura N° 6 Definición de las variables de entorno
Elaborado por: Gina Ulloa

- Las variables de sistema definidas son las siguientes:

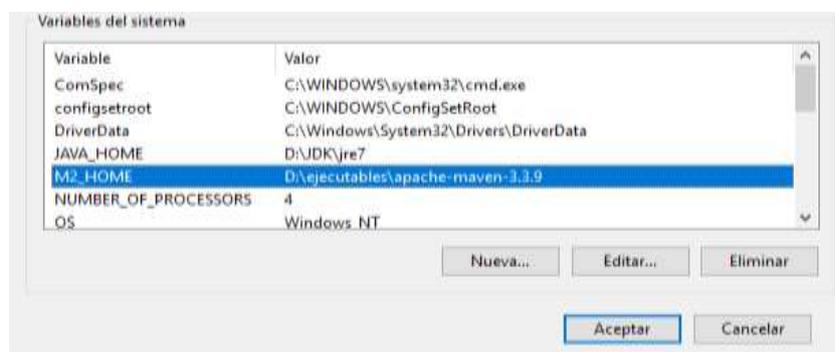


Figura N° 7 Ruta de configuración de Variables configuradas
Elaborado por: Gina Ulloa

Para comprobar que está correctamente instalado podemos probar ejecutando: **mvn – version** en la consola de Windows o en el terminal de Linux.

1.1.3 Características

Entre las características más importantes que tiene Maven son:

- Es posible trabajar en Maven en modo online es decir sin necesidad de estar conectado a internet solo con realizar una pequeña configuración en el fichero POM.
- En el fichero POM.xml se almacena toda la información del proyecto dependencias, plugins, modalidad, etc.
- Los sitios web de los proyectos Apache están contruidos con Maven, además site es una fase adicional de Maven la cual se agrega una vez que se invoca la sentencia mvn site.
- Maven es más amigable si se trabaja en una interfaz gráfica, el uso de consola podría ser un poco más complicado para el programador.

1.1.4 Ventajas de Maven

Entre las ventajas más importantes que tiene Maven son:

- **Estandarización en el sistema de compilación**, es necesario conocimientos previos para evitar pérdidas de tiempo en el aprendizaje.
- **Enfoque en la automatización**, Maven permite automatizar procesos en el proyecto de software.
- **Fácil configuración**, Maven no requiere de configuraciones para construir artefactos.
- **Gestionamiento de dependencias**, esta es una gran ventaja debido a que Maven gestionará las dependencias del programador.
- **Facilitación de pruebas**, Maven tiene la capacidad de ejecutar pruebas como parte del ciclo de vida del proyecto.
- **Disponibilidad de complementos**, Maven dispone de una variedad de complementos y plugins para ejecutar tareas diversas, solo es necesario configurar el archivo POM o en su defecto descargar dichos plugins de acuerdo al requerimiento del programador.
- **Organización y estandarización** en la estructura de ficheros del proyecto.

- **Facilita la modularidad y el versionamiento** de los proyectos lo que permite trabajar con varios grupos de programadores en un mismo desarrollo.

1.1.5 Documentación técnica

A continuación se detallará una guía técnica sobre Maven.

INTRODUCCIÓN A MAVEN
Documentación Técnica

ÍNDICE

1. Objetivo del presente documento.....	16
2. Definición de Maven.....	16
3. Objetivos de Maven.....	16
4. Herramientas de Maven.....	16
5. Estandarización de la estructura de un proyecto en Maven.....	17
5.1 Información general del proyecto.....	17
5.2 Dependencias.....	18
5.3 Plugins.....	22
5.3.1 Plugin del núcleo.....	22
5.3.2 Plugin de empaquetamiento.....	23
5.3.3 Plugin de reportes.....	23
5.4 Automatización de las fases de un proyecto desarrollado en Maven.....	24
6. Definición de POM.....	26
7. Herencia entre proyectos Maven.....	27
8. Proyecto de interpolación y variables.....	28
9. Creación de un proyecto Maven.....	29

Tipo de documento		
Informativo		
Tema del documento	Revisión	Nº paginas
Introducción a Maven 3.3.9 Documentación técnica	1.0	16
Elaborado por:		
Gina Ulloa		
Historial de cambios		
Versión	Fecha	Cambios

1. Objetivo del presente documento

El objetivo de esta documentación técnica de Maven es describir las características y requerimientos técnicos que necesita este gestor para el desarrollo de un proyecto de software y también definir conceptos utilizados en esta herramienta para que el usuario tenga al alcance una guía práctica y entendible.

2. Definición de Maven

Apache Maven es una herramienta estándar de software que utiliza un POM (modelo de objeto del proyecto), permite la compilación, gestión de informes y documentación de un proyecto de software, tiene tareas definidas como el empaquetado y la compilación del código fuente permitiendo definir de manera clara el objetivo del proyecto.

De acuerdo a la (**Apache Software Foundation, 2019**) menciona lo siguiente: Apache Maven es una herramienta de gestión y comprensión de proyectos de software. Basándose en el concepto de un modelo de objeto de proyecto (POM), Maven puede gestionar la compilación, los informes y la documentación de un proyecto a partir de una información central.

3. Objetivos de Maven

Maven cumple con objetivos definidos como:

- Permitir que el proceso de elaboración de un proyecto sea más sencillo.
- Brindar información de calidad del proyecto.
- Brindar un proyecto con elaboración uniforme.
- Ofrecer información transparente para todos los usuarios de Maven
- Ayudar en el flujo de trabajo del proyecto.
- Proporcionar facilidades en las actualizaciones de Maven.

4. Herramientas de Maven

Estas herramientas están disponibles en Maven por defecto a continuación se detalla cada una de ellas:

- **Ant:** Permite generar un archivo de compilación para el proyecto.

- **Antrun:** Permite ejecutar ciertas tareas desde cualquiera de las fase de construcción de proyecto.
- **Archetype:** Permite generar la estructura de un proyecto mediante un patrón.
- **Dependency:** Permite copiar, empaquetar o desempaquetar archivos.
- **Enforcer:** Permite ejecutar reglas personalizadas del usuario.
- **Gpg:** Permite crear firmas para los POM y plugins.
- **Help:** Permite obtener información sobre el entorno de trabajo del proyecto.
- **Pdf:** Permite generar informes en formato PDF.
- **Scm:** Permite ejecutar comandos SCM para el proyecto.
- **Scm – publish:** Permite publicar el sitio web en una ubicación de SCM.

5. Estandarización de la estructura de un proyecto en Maven

Maven permite que las dependencias, plugins, meta datas del sistema puedan ser declaradas con el objeto de estandarizar el proyecto y facilitar su empaquetamiento, en el archivo pom.xml se guarda todas las configuraciones que Maven realiza; cabe aclarar que Maven trabaja bajo el concepto de Modelo de Objetos de Proyecto, a continuación se describirá la estructura básica estándar de un archivo pom.xml:

5.1 Información general del proyecto

Esta sección contiene la información principal que caracteriza al proyecto los ítems son los siguientes:

groupId: Aquí se identifica el nombre del grupo al que pertenece el proyecto, es un campo obligatorio, aunque no es necesario definirlo si se heredan de un padre, normalmente es único entre una organización o un proyecto. Por ejemplo los artefactos de Maven deberían tener el groupId.

artifactId: Normalmente es el nombre con el que se conoce al proyecto, esto junto con el groupId definen como se grabaron los artefactos dentro del repositorio.

version: Los cambios de código deben ser versionados y este elemento mantiene esas versiones en línea. También se usa dentro del repositorio de un artefacto para separar las versiones entre sí.

Estos tres elementos apuntan a una versión específica de un proyecto, lo que le permite a Maven identificar con quién está trabajando, en la figura que se muestra a continuación describe parte de la información de configuración del proyecto Maven:

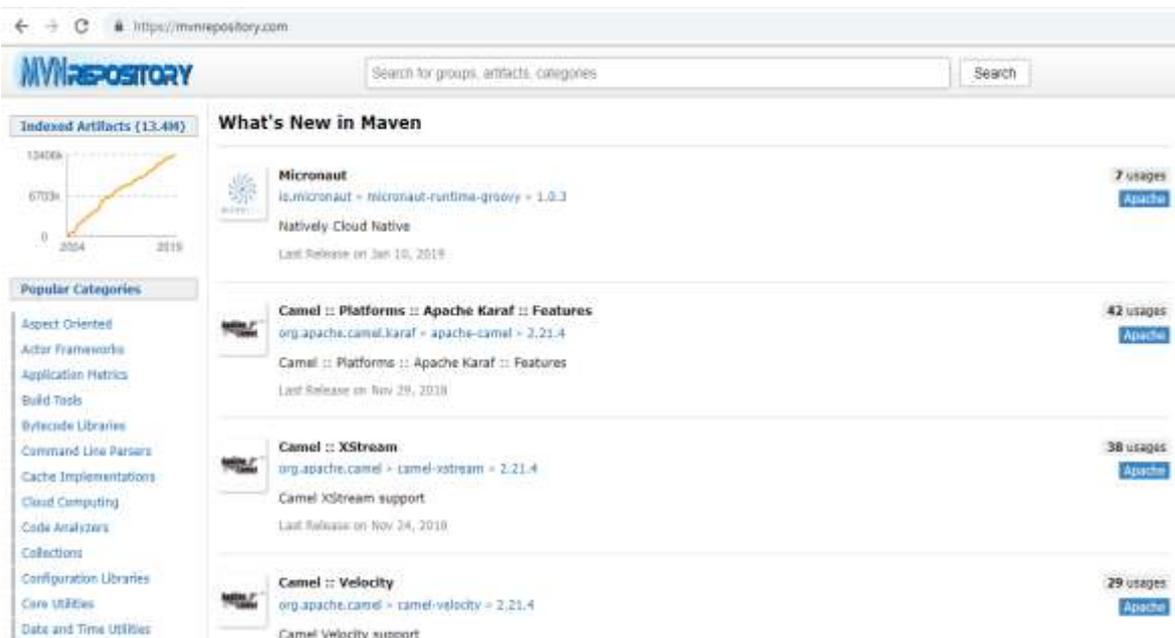
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5     <modelVersion>4.0.0</modelVersion>
6
7     <groupId>com.tienda.online</groupId>
8     <artifactId>tiendaOnline</artifactId>
9     <version>0.0.1-SNAPSHOT</version>
10
11 </project>
```

Figura N° 8 Información general de un proyecto Maven
Elaborado por: Gina Ulloa

5.2 Dependencias

En esta sección se mostrará la forma de agregar las dependencias que necesita un proyecto para su desempeño, para esto se necesitará acceder al repositorio principal de Maven donde se encuentran los artefactos que podemos necesitar con sus versiones específicas.

Para descargar estas dependencias es necesario ir al repositorio de Maven: (MavenRepository, 2006), la url de acceso es: <http://www.mvnrepository.com>



The screenshot displays the Maven Repository website interface. At the top, there is a search bar with the text "Search for groups, artifacts, categories" and a "Search" button. Below the search bar, the page is divided into several sections. On the left, there is a "Popular Categories" sidebar with a list of categories such as "Aspect Oriented", "Actor Frameworks", "Application Metrics", "Build Tools", "Bytecode Libraries", "Command Line Parsers", "Cache Implementations", "Cloud Computing", "Code Analyzers", "Collections", "Configuration Libraries", "Core Utilities", and "Date and Time Utilities". The main content area features a "What's New in Maven" section. This section includes a graph showing "Indexed Artifacts (13,404)" with a line chart showing an upward trend from 2004 to 2019. Below the graph, there are four entries for new releases: 1. "Micronaut" with the artifact "is.micronaut - micronaut-runtime-groovy = 1.0.3", last released on Jan 10, 2019, with 7 usages and an Apache license. 2. "Camel :: Platforms :: Apache Karaf :: Features" with the artifact "org.apache.camel.karaf - apache-camel = 2.21.4", last released on Nov 29, 2018, with 42 usages and an Apache license. 3. "Camel :: XStream" with the artifact "org.apache.camel - camel-xstream = 2.21.4", last released on Nov 24, 2018, with 38 usages and an Apache license. 4. "Camel :: Velocity" with the artifact "org.apache.camel - camel-velocity = 2.21.4", last released on Nov 24, 2018, with 29 usages and an Apache license.

Figura N° 9 Repositorio central de Maven
Fuente: www.mvnrepository.com

Dentro de esta página permite buscar al investigador los artefactos necesarios para empezar a trabajar en el proyecto Maven, para la presente investigación se necesitará de SpringJPA, como se muestra a continuación:

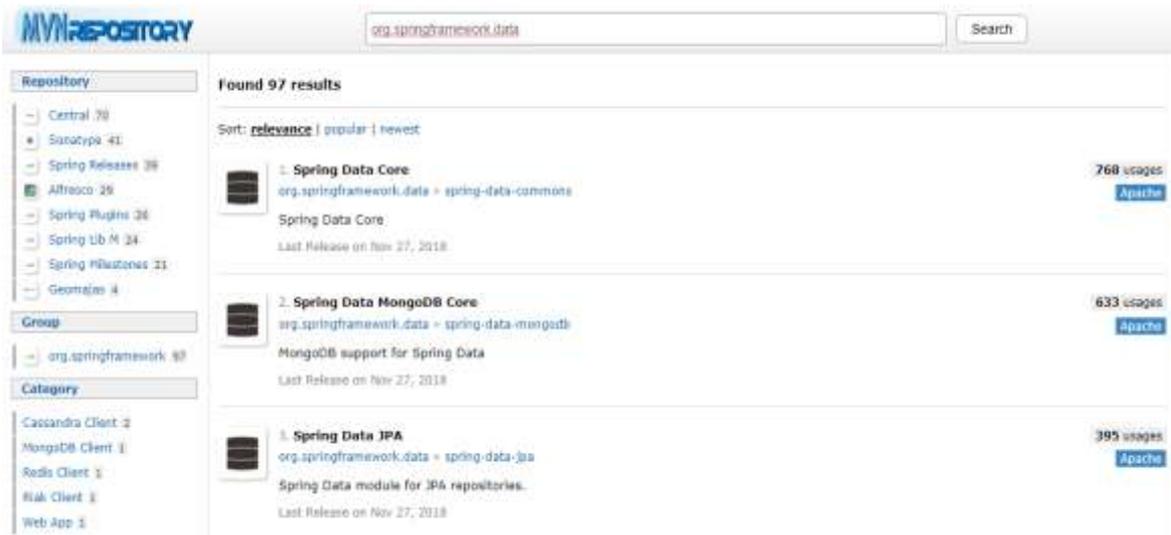


Figura N° 10 Dependencia Spring JPA del repositorio central de Maven
Fuente: www.mvnrepository.com

Luego de pulsar en el enlace **Spring Data JPA** se puede observar las distintas versiones que se encuentran almacenadas en el repositorio central de Maven del artefacto, a continuación se escogerá la versión adecuada para el desarrollo del proyecto en desarrollo, como se muestra a continuación:

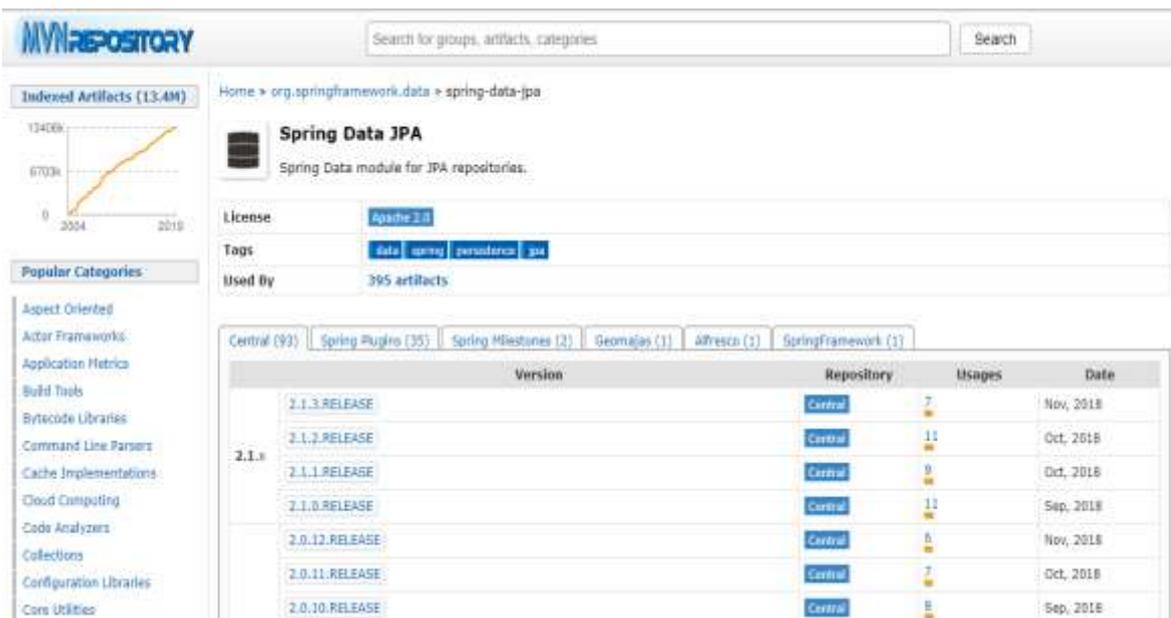
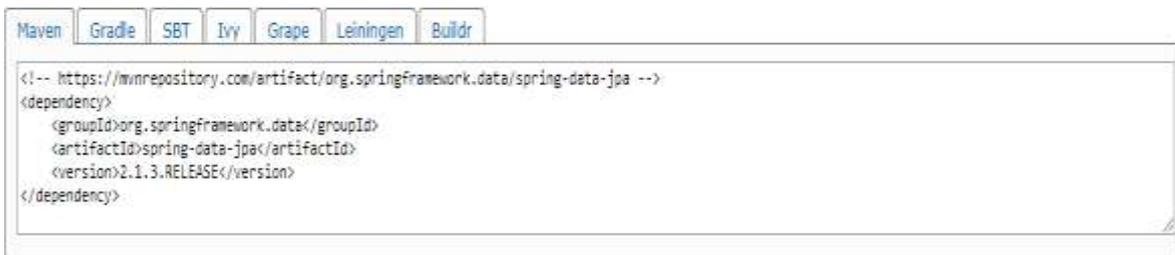


Figura N° 11 Versiones de la dependencia Spring JPA
Fuente: www.mvnrepository.com

Una vez seleccionada la versión, se podrá visualizar el código necesario que se deberá añadir en el fichero POM.XML para utilizar en el proyecto.



```

Maven  Gradle  SBT  Ivy  Grape  Leiningen  Buildr
<!-- https://mvnrepository.com/artifact/org.springframework.data/spring-data-jpa -->
<dependency>
  <groupId>org.springframework.data</groupId>
  <artifactId>spring-data-jpa</artifactId>
  <version>2.1.3.RELEASE</version>
</dependency>

```

Figura N° 12 Código para añadir al fichero POM
Fuente: www.mvnrepository.com

Determinadas las dependencias necesarias para la elaboración proyecto, es necesario empaquetar la aplicación con “mvn package”, este permitirá a Maven bajar de su repositorio central las dependencias necesarias y las añade a la carpeta lib.

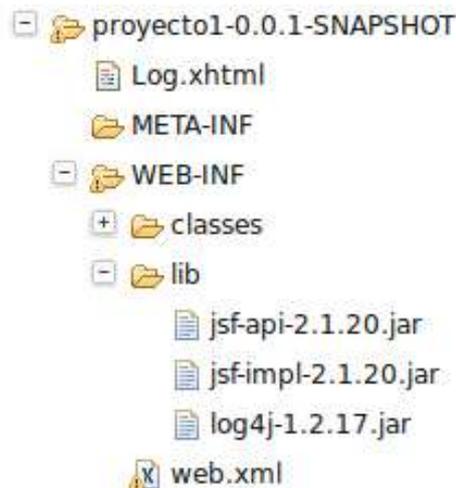


Figura N° 13 Proyecto Maven ya incluyendo las dependencias
Elaborado por: Gina Ulloa

Existen dos tipos de repositorios:

- **Repositorio local:** es una carpeta local en donde se guardan las dependencias de los proyectos (jars, plugins y otros archivos que se descargan por Maven), por defecto esta carpeta se llama .m2 y se encuentra ubicada según el Sistema Operativo en:

- **Linux:** /home/\${usuario}/.m2/
- **Windows 7 o superiores:** C:/Users/\${usuario}/.m2

Si se desea indicar otra ruta para el repositorio local es necesario incluir una línea en el archivo de configuración de Maven settings.xml.

```
<localRepository>${ruta_nueva_repositorio_local}</localRepository>
```

El archivo de configuración se encuentra ubicado en la ruta de instalación de Maven, según el sistema operativo las rutas donde se instala por defecto son las siguientes:

- **Linux:** /opt/apache-maven-\${version}/conf/
 - **Windows:** C:\Program Files (x86)\apache-maven-\${version}\conf
- **Repositorio Remoto:** es cualquier otro tipo de repositorio, al que se accede mediante varios protocolos como file:// y http:// o puede ser un repositorio realmente remoto configurado por un tercero para suministrar sus artefactos para descargar, por ejemplo, repo.maven.apache.org y uk.maven.org también alojan el repositorio central de Maven, también pueden ser repositorios internos configurados en un archivo o servidor HTTP dentro de nuestra empresa los cuales se puede utilizar para compartir artefactos privados para los equipos de desarrollo.

Para usar el repositorio interno es necesario hacer un cambio para agregar un elemento de repositorios:



```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  ...
  <repositories>
    <repository>
      <id>repositorio interno</id>
      <name>Repositorio se ejecuta en repo.mycompany.com </name>
      <url>http://repo.mycompany.com/proxy</url>
    </repository>
  </repositories>
  ...
</project>
```

Figura N° 14 Agregando repositorio interno en Maven
Elaborado por: Gina Ulloa

Estos repositorios están estructurados de la misma manera para que los scripts puedan ejecutarse en cualquier lado, o pueda sincronizarse para usarlos sin conexión, sin embargo, el diseño de los repositorios es transparente para el usuario de Maven.

5.3 Plugins

Maven tiene un motor central que proporciona capacidades básicas de procesamiento de proyectos y gestión de procesos de compilación y una serie de complementos, en esta sección es donde se realiza gran parte de la acción real, estos plugins se utilizan para: crear archivos jar, crear archivos war, compilar código, código de prueba de unidad, crear documentación del proyecto, entre otras. Casi cualquier acción que se piensa realizar en un proyecto se implementa como un complemento de Maven. El comportamiento del complemento se puede personalizar mediante un conjunto de parámetros que se presentan mediante una descripción de cada objetivo del complemento (Mojo²), esta es una de las ventajas que tiene este motor central ya que brinda soporte a toda la gama de plugins como son: plugins de empaquetado, generación de reportes, análisis de código, entre otros, los mismos que se utilizan en la elaboración de proyectos y permiten el correcto funcionamiento de Maven.

5.3.1 Plugins del Núcleo

Estos plugins permiten dar la funcionalidad a los servicios básicos de Maven:

- **Clean:** Permite realizar un mantenimiento o limpieza al proyecto.
- **Compiler:** Este plugin permite la compilación del código en diversas versiones del JDK.
- **Install:** Despliega el proyecto en un repositorio local.
- **Deploy:** Despliega el proyecto en un repositorio remoto.
- **Resources:** Este plugin permite copiar los recursos para que se incluyan en el JAR.
- **Site:** Permite generar un portal web para el proyecto.
- **Surefire:** Ejecuta pruebas unitarias y genera un informe de resultados.
- **Verifier:** Permite pruebas de integración.
- **Cargo:** Permite desplegar el proyecto en varios servidores de aplicaciones como: JBoos, Tomcat, Jetty, Geronimo, OC4J, Weblogic, entre otros.

² **Mojo:** es un objetivo en Maven y los complementos son cualquier número de objetivos (Mojos)

5.3.2 Plugins de empaquetamiento

Entre ellos están los siguientes: ear, ar, ejb, jar, rar, jmod, jlink, shade.

5.3.3 Plugins de Reportes

Entre ellos están los siguientes:

- **Changelog:** Permite crear un documento con los cambios recientes del proyecto.
- **Changes:** Genera un informe de los cambios realizados en el proyecto.
- **Checkstyle:** Permite generar informes de calidad del código, mediante el uso de métricas de calidad de software.
- **Doap:** Permite generar un informe desde el archivo POM.
- **Dock:** Permite la verificación de la documentación.
- **Javadoc:** Permite crear el documento java del proyecto.
- **Pmd:** Permite generar informes una vez analizado posibles fallos en las sentencias y ciclos, código ineficiente, código duplicado, métodos no utilizados.
- **Project – info – reports:** Permite generar informes estandarizados.
- **Surefire – Report:** Permite generar un informe con los resultados de las pruebas JUnit.

Maven incluye sus plugins para usarlos con valores predeterminados, pero también permite configurar el comportamiento de los plugins sobre-escribiendo estos valores predeterminados, permitiendo definir nuevos valores, para ellos es necesario utilizar el elemento de Maven <configuration>, como se muestra a continuación:

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>3.6.1</version>
  <configuration>
    <source>1.8</source>
    <target>1.8</target>
  </configuration> </plugin>
```

Figura N° 15 Ejemplo de plugin
Elaborado por: Gina Ulloa

Un plugin puede tener uno o varios objetivos, donde cada uno indica una función específica del plugin, por ejemplo, el plugin **Compiler** tiene dos objetivos que son:

- **compile** que compila el código fuente principal
- **testCompile** que compila el código fuente de las pruebas

Además puede contener información de la fase del ciclo de vida asociado, los plugins pueden utilizarse en más de una fase o no tener ninguna asociada, como se muestra en la siguiente figura:

```

<plugin>
  <groupId>com.mycompany.example</groupId>
  <artifactId>display-maven-plugin</artifactId>
  <version>1.0</version>
  <executions>
    <execution>
      <phase>process-test-resources</phase>
      <goals>
        <goal>time</goal>
      </goals>
    </execution>
  </executions>
</plugin>

```

Figura N° 16 Ejemplo de plugin con una fase asociada
Elaborado por: Gina Ulloa

5.4 Automatización de las fases de un proyecto desarrollado en Maven

El ciclo de vida de un proyecto en Maven tiene varias fases que se deben cumplir, estas se ejecutan secuencialmente, lo que significa que ejecutar una fase representa ejecutar todas las fases anteriores. Maven se basa en la noción central de un ciclo de vida de construcción, hay tres ciclos de vida de construcción agregados:

- **Defecto (default):** permite manejar la implementación del proyecto.
- **Limpiar (clean):** permite manejar la limpieza del proyecto.
- **Sitio (site):** permite la creación de documentación del sitio del proyecto.

Clean Lifecycle	Default Lifecycle		Site Lifecycle
pre-clean	validate	test-compile	pre-site
clean	initialize	process-test-classes	site
post-clean	generate-sources	test	post-site
	process-sources	prepare-package	site-deploy
	generate-resources	package	
	process-resources	pre-integration-test	
	compile	integration-test	
	process-classes	post-integration-test	
	generate-test-sources	verify	
	process-test-sources	install	
	generate-test-resources	deploy	
	process-test-resources		

Figura N° 17 Fases del ciclo de vida de construcción

Fuente: <https://rubensa.wordpress.com/2016/06/10/maven-lifecycle/>

Cada ciclo de vida se compone de fases, el ciclo de vida **default** es muy importante porque Maven ejecuta las fases de forma secuencial, entre las más importantes se encuentran:

- **Validate:** Validar el proyecto y la información necesaria que está disponible de manera correcta.
- **Compile:** Compilar el código fuente del proyecto.
- **Test:** Probar el código fuente compilado.
- **Package:** Empaquetar el código compilado en un formato conocido.
- **Verify:** Verificar los resultados de las pruebas de integración para garantizar los criterios de calidad.
- **Install:** Instalar los paquetes en el repositorio local.
- **Deploy:** Copiar los paquetes al repositorio remoto para que otros desarrolladores hagan uso de estos archivos.

Para pasar por las fases anteriores solo se deberá llamar al comando **mvn install**, lo que permitirá que se ejecute desde la primera fase de manera secuencial hasta la instalación (install). Una fase de construcción se compone de objetivos de plugins y aunque una fase es responsable de un paso en el ciclo de vida, este puede variar indicando objetivos (goals³) de plugins asociados a las fases de construcción.

³ **Goal:** Es un comando que toma Maven como parámetro para hacer algo

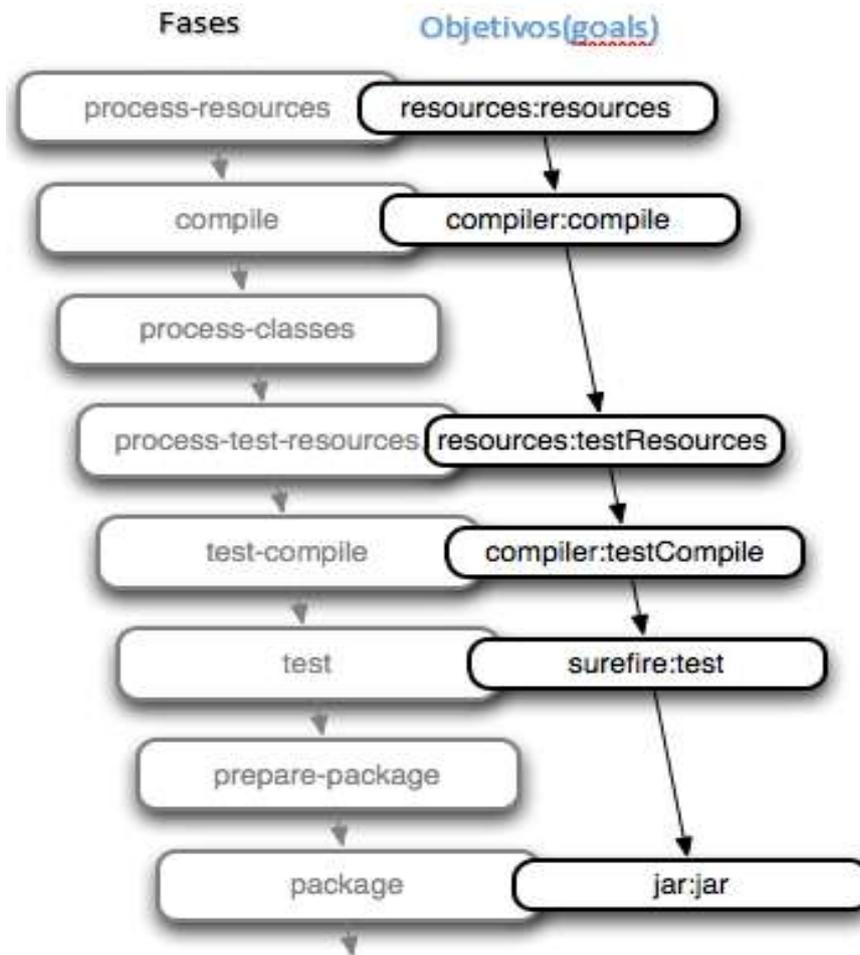


Figura N° 18 Ejemplo de fase de construcción y objetivos
Fuente: <https://rubensa.wordpress.com/2016/06/10/maven-lifecycle/>

6. Definición de POM

Project Object Model es la unidad principal de trabajo en Maven, es un archivo XML donde se encuentra información sobre la configuración, versión, listas de correo del proyecto, contiene configuraciones predeterminadas para los proyectos que se construyen en Maven como por ejemplo se puede citar a los siguientes: target (directorio de compilación), src/main/java (directorio fuente), src/test/java (directorio fuente de prueba); el POM deberá contener los siguientes requisitos:

- Raíz del proyecto
- Modelo o versión de la herramienta
- Id del grupo del proyecto
- Id del artefacto
- Versión del artefacto en el grupo específico

Se mostrará un ejemplo de configuración de un POM simple:

1. <project>
2. <modelVersion> 3.9.9 </modelVersion>
3. <groupId> com.tesis.app </groupId>
4. <artifactId> my-app </artifactId>
5. <version> 1 </version>
6. </project>

El tipo de empaquetamiento que tendrá el proyecto a elaborarse se configura por defecto en Maven y se indica el empaquetamiento en jar; en el caso de aplicaciones web tendremos que especificar empaquetamiento war, también podemos usar empaquetamientos ear o pom. Existe además el Super POM el cuál es el POM predeterminado de Maven, todos los POM se heredan de un elemento principal o predeterminado, es necesario señalar que debe haber un solo archivo POM para cada proyecto. Para conocer las configuraciones predeterminadas del Súper POM se debe ejecutar el siguiente comando en la consola, se deberá ubicar en la carpeta que contiene pom.xml:

```
mvn help:effective-pom
```

7. Herencia entre proyectos Maven

Hay ocasiones en que el programador tendrá más de cinco proyectos con dependencias cruzadas, este entorno se hace inmanejable, pero Maven tiene la solución a este problema la herencia entre proyectos, para esto es necesario crear un pom.xml que controle a todos, es decir, que sea el padre. Este pom.xml se puede colocar en el directorio padre de los proyectos, pero también a la misma altura de los mismos como si fueran hermanos. Este pom.xml tendrá el siguiente aspecto:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>
  <groupId>com.tienda.online</groupId>
  <artifactId>tiendaOnline</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>pom</packaging>
  <modules>
    <module>../tiendaOnlineBackend</module>
    <module>../tiendaOnlineFrontend</module>
  </modules>
</project>
```

Figura N° 19 Configuración de POM.XML padre
Elaborado por: Gina Ulloa

Como se puede observar el elemento packaging es el pom y dentro del elemento modules se listan todos los módulos que forman parte de la aplicación. Al ejecutar un objetivo sobre este POM.XML, Maven se encarga de ejecutarlo sobre todos los módulos configurados, tomando en cuenta las dependencias entre ellos, lo que significa que compila primero la dependencia y luego al dependiente, también se deberá modificar el POM.XML de los módulos del padre para indicar la herencia, para esto se deberá añadir lo siguiente en cada POM.XML hijo:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  ...
  <parent>
    <groupId>com.tienda.online</groupId>
    <artifactId>tiendaOnline</artifactId>
    <version>0.0.1-SNAPSHOT</version>
  </parent>
  ...
</project>
```

Figura N° 20 Configuración POM.XML hijo
Elaborado por: Gina Ulloa

8. Proyecto de interpolación y variables

Maven recomienda no repetir código innecesario pero hay lugares del proyecto en donde es necesario usar el mismo valor en varias ubicaciones diferentes de los poms, para garantizar que este valor solo se especifique una vez. Maven permite usar sus propias variables. Por ejemplo, para acceder a la variable project.version se deberá referenciarla de la siguiente manera:

```
<version>${project.version}</version>
```

Estas variables se resuelven después de la herencia descrita anteriormente, lo que significa que si un proyecto principal usa una variable, su definición en el elemento secundario será la que la utilizará, también se puede hacer referencia a las propiedades determinadas en el proyecto como una variable:

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  ...
  <properties>
    <mavenVersion>2.1</mavenVersion>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.apache.maven</groupId>
      <artifactId>maven-artifact</artifactId>
      <version>${mavenVersion}</version>
    </dependency>
    <dependency>
      <groupId>org.apache.maven</groupId>
      <artifactId>maven-project</artifactId>
      <version>${mavenVersion}</version>
    </dependency>
  </dependencies>
  ...
</project>

```

Figura N° 21 Configuración variables
Elaborado por: Gina Ulloa

9. Creación de un proyecto Maven

Una vez que se ha estudiado todo acerca de las características y funcionalidades de Maven se podrá empezar a crear proyectos, para esto se deberá recordar que Maven usa “arquetipos” que no son más que plantillas, es decir, gracias a un arquetipo Maven tiene la capacidad de generar una estructura de directorios y ficheros.

Una de las ventajas de estas plantillas es que facilita el trabajo para el programador solo es necesario indicarle a Maven que tipo de proyecto se quiere elaborar y automáticamente creará la estructura base del proyecto a elaborarse.

ARQUETIPOS	DESCRIPCIÓN
maven-archetype-archetype	Arquetipo para generar un arquetipo de muestra.
maven-archetype-j2ee-simple	Arquetipo para generar una aplicación J2EE de muestra simplificada.
maven-archetype-plugin	Arquetipo para generar un plugin Maven de muestra.
maven-archetype-plugin-site	Arquetipo para generar un sitio de plugin de Maven de muestra.
maven-archetype-portlet	Arquetipo para generar un Portlet JSR-268 de muestra.
maven-archetype-quickstart	Arquetipo para generar un proyecto maven de muestra.
maven-archetype-simple	Arquetipo para generar un proyecto simple de Maven.
maven-archetype-site	Arquetipo para generar un sitio Maven de muestra que muestra algunos de los tipos de documentos compatibles, como APT, Markdown, XDoc y FML, y muestra cómo configurar su sitio.
maven-archetype-site-simple	Arquetipo para generar un sitio de muestra de Maven.
maven-archetype-site-skin	Arquetipo para generar una muestra de Maven Site.
maven-archetype-webapp	Arquetipo para generar un proyecto de ejemplo de aplicación web Maven.

Tabla N° 1 Principales arquetipos de Maven
Elaborado por: Gina Ulloa

Para este ejemplo se utilizó el arquetipo Web, es decir deberá ejecutar una sentencia en el terminal y escribir lo siguiente:

```
mvn archetype:generate -DarchetypeGroupId=org.apache.maven.archetypes -
DarchetypeArtifactId=maven-archetype-webapp -DarchetypeVersion=1.3
```

Esta instrucción permitirá llenar la información básica del archivo de configuración POM la cual se ha indicado anteriormente, definición de información como es el groupId,

artefactId, versión y package, finalmente el proyecto web creado cuenta con la siguiente estructura de carpetas:

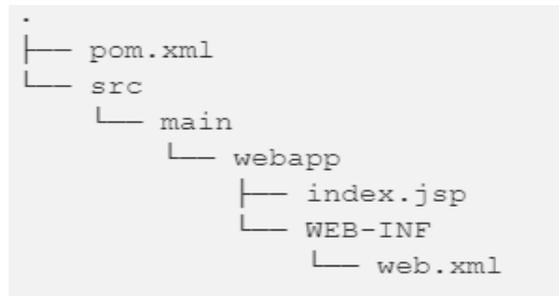


Figura N° 22 Estructura de directorio de un proyecto web con Maven
Elaborado por: Gina Ulloa

1.2 SPRING MVC

1.2.1 Introducción

Este framework que es de código abierto brinda un modelo de programación entendible y estable para la comunidad de desarrolladores Java, Spring MVC es un conjunto de librerías portables que no necesita de un servidor para implementar sus APIs, además sigue el patrón de diseño Modelo Vista Controlador; una de las características importantes de este framework es el soporte que brinda a nivel de aplicación, es decir se centra en la instalación de las aplicaciones para que los equipos trabajen en la lógica del negocio; actualmente el punto fuerte donde interactúa Spring es Big Data, aplicaciones móviles, HTML5, entre otros.

Este framework puede ser configurado mediante los archivos XML o a través de código Java, está basado en peticiones HTTP que define varias interfaces mediante un patrón de diseño que encapsula algoritmos relacionados en subclases para una superclase concreta, con la finalidad de re-utilizar estos algoritmos y mostrar cada una de las interfaces que cumplen funciones y responsabilidades en el proyecto y permiten que para el programador se visualice de manera sencilla y amigable, a continuación se detalla las interfaces más importantes:

- **Controller:** Controla las peticiones que ingresan y las redirige a su destino, estos componentes son programables y se encuentra entre las capas Modelo y Vista.
- **HandlerAdapter:** En esta interface se realiza la invocación al componente central de Spring MVC denominado DispatcherServlet cuya descripción se la definirá más adelante.

- **HandlerInterceptor:** Es la interface encargada de interceptar las peticiones de entrada.
- **HandlerMapping:** Es la encargada de seleccionar los objetos que manejan las peticiones de entrada.
- **LocaleResolver:** Esta interface resuelve y guarda la localización de un usuario.
- **MultipartResolver:** Esta interface permite el fácil trabajo con la carga de archivos.
- **View:** En esta interface se regresan las respuestas al usuario final, dependiendo del tipo de peticiones algunas respuestas irán directo a la capa de vista sin pasar por la capa de modelo.
- **ViewResolver:** Se encarga de identificar el recurso al cual se está haciendo referencia.
- **DispatcherServlet** es un controlador que utiliza el patrón de diseño de software front controller el mismo que consiste en usar un único camino de acceso para gestionar las peticiones HTTP, este controlador además de gestionar comprueba las restricciones de seguridad, se encarga del manejo de errores, mapeo y delegación de peticiones a los diferentes componentes que integran la aplicación para generar la vista que utilizará el cliente final, este archivo es el index.php.
- **Servlet** proviene de la palabra applet es una clase del lenguaje de programación Java, esta clase es usada para aumentar las capacidades de un servidor, su función principal es generar páginas web de manera dinámica mediante parámetros dependiendo de las peticiones que haga el navegador web.

A continuación se muestra un esquema de la arquitectura general de las aplicaciones web con Spring MVC:

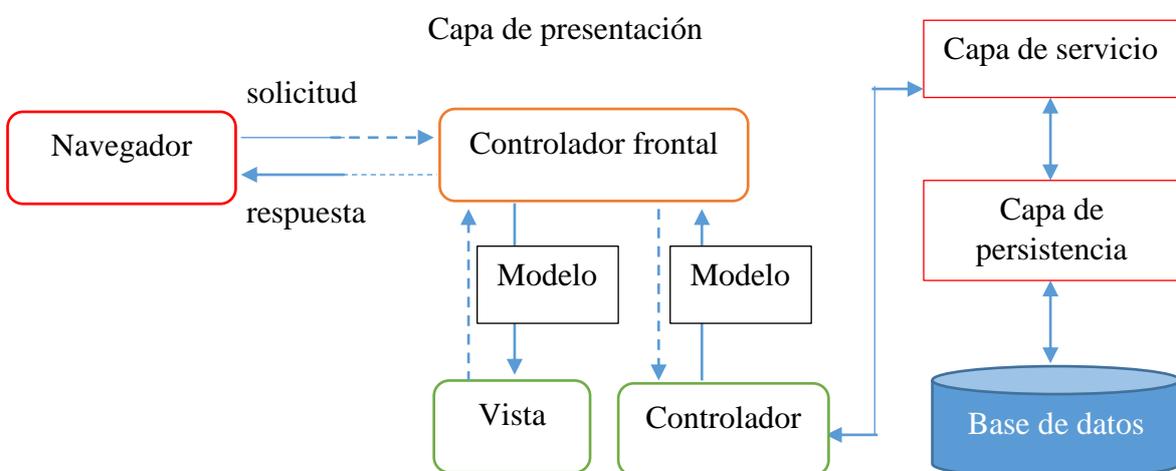


Figura N° 23 Capas de una aplicación Spring MVC
Fuente: (Ganeshman, 2016)

1.2.2 Características

A continuación se describirán algunas características de Spring MVC:

- Spring MVC está orientado a peticiones hacia los controladores con una funcionalidad especial que permite el desarrollo de proyectos web.
- Los controllers se construyen sobre el API de servlets.
- Spring MVC se caracteriza por su política de diseño “Abierto para recibir sugerencias de implementaciones, cerrado para modificación de los métodos”, esto quiere decir que ciertos métodos en las clases core están protegidos contra escritura, así el framework asegura su esencia y comportamiento básico.
- Spring MVC comparte el motor DispatcherServlet con otros frameworks Modelo Vista Controlador.

1.2.3 Patrón de Diseño MVC

Este patrón de diseño tiene la finalidad de separar el código fuente y la lógica del negocio en tres capas distintas cada una con sus responsabilidades, permite desarrollar proyectos de mejor calidad, es decir, que el programador puede hacer cambios en el código y esto no debería de afectar al resto de las capas, este patrón de diseño es muy utilizado en la mayoría de arquitecturas como son: Python, Lavarel, Ruby on Rails, AngularJS, su esencia se centra en la reutilización de código y el principio de diseño con la separación de secciones de un proyecto, a continuación se muestra el funcionamiento de este patrón de diseño:

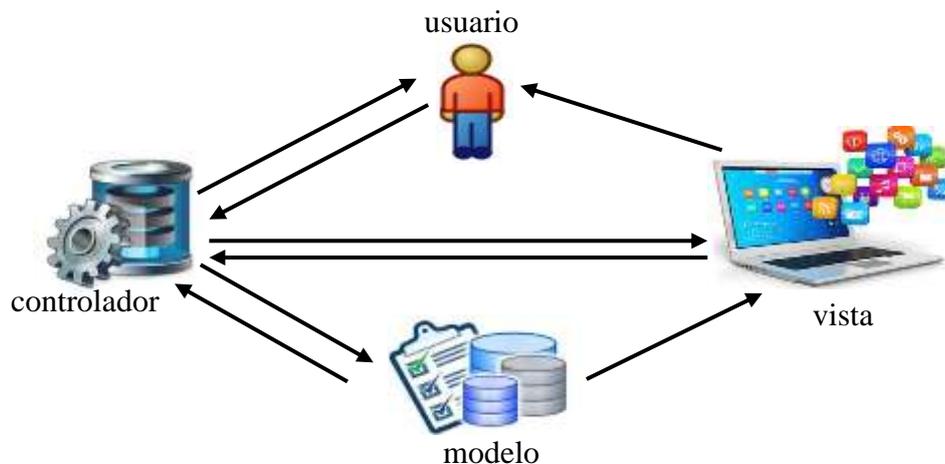


Figura N° 24 Funcionamiento del Patrón de diseño MVC
Elaborado por: Gina Ulloa

Como se puede observar en la figura el usuario hace una petición al servidor, es decir accede al controlador inmediatamente este se comunica con el modelo y la vista, aquí se realiza un proceso interno donde la petición es acceder o actualizar los datos y automáticamente la vista solicita la salida de la petición del usuario aunque en ocasiones la salida puede regresar de nuevo al controlador y este sería el que envía la respuesta al usuario.

Ventajas del patrón de diseño MVC

- Este patrón permite que las aplicaciones posean extensibilidad y mantenibilidad
- Los cambios realizados en la capa vista no afectan a la capa modelo de dominio, solo se modifica la presentación de los datos.
- Realiza actualizaciones constantes; de modo que cuando el programador efectúa cambios no es necesario que realice peticiones de actualización a la capa de vistas, porque esta acción se realiza de manera automática en la capa de modelo de la aplicación.
- La aplicación de este patrón de diseño se la realiza cumpliendo las fases de cada una de las capas.

Desventajas del patrón de diseño MVC

- El patrón de diseño Modelo Vista Controlador está orientado a objetos por lo que su implementación resulta muy costosa.
- Para la implementación de este patrón de diseño es necesaria mayor planificación inicial en el desarrollo.
- Es necesario de una arquitectura para implementar el patrón de diseño, debe existir una clase para la capa modelo donde se realiza las tareas de comunicación, otra clase para la capa vista para las notificaciones y una clase para la capa controlador donde se realizará las actualizaciones que servirán de eje en el desarrollo del proyecto.

De acuerdo a (**Ganeshman, 2016**) menciona lo siguiente: La capa de presentación de cada aplicación de nivel empresarial se puede dividir lógicamente en tres partes principales:

- La parte que gestiona los datos (**Modelo**)

- La parte que crea la interfaz de usuario y las fichas (**Vista**)
- La parte que maneja las interacciones entre el usuario, la interfaz de usuario y los datos (**Controlador**) (pag. 51)

A continuación se describirá cada una de las capas:

1.2.3.1 Modelo

En esta capa se encapsulan los datos y representa la lógica del proyecto, por lo general los datos se encuentran almacenados en una base de datos, esta capa es independiente del resultado de las capas controlador o vista, existe el **modelo de dominio** que contiene un conjunto de clases definidas por el análisis del planteamiento del problema y el **modelo de la aplicación** es un conjunto de clases que son el puente para relacionarse las vistas con el modelo de dominio, este es el coordinador del proyecto.

Las funciones de la capa de modelo son las siguientes:

- Definir la funcionalidad del proyecto.
- Llevar un registro de las capas vistas y controladores del proyecto.
- Notificar a la capa vista los cambios que se producen en los datos por los usuarios.
- Acceder al almacenamiento de datos.

1.2.3.2 Vista

En esta capa se presentan los datos del modelo mediante una interfaz de usuario, pueden existir varias vistas contenidas en un mismo modelo, en esta capa se accede a la información del modelo pero desconoce su funcionalidad frente al usuario, aquí se encuentra el código fuente para mostrar la salida de la información, cada vez que el usuario realiza interacciones que implican modificaciones en la capa de modelo de dominio, automáticamente la vista se actualiza mediante notificaciones generadas por la capa de modelo de la aplicación.

Las funciones de la capa vista son las siguientes:

- Receptar la información de la capa modelo y mostrarla al usuario.
- Realizar un registro del controlador interno.
- Actualizar la información cada vez que el controlador o el modelo lo requiera.

1.2.3.3 Controlador

En esta capa se realiza la interpretación de la información enviada por la vista, es el enlace entre la vista y el modelo para resolver las solicitudes del usuario; su función es actuar en el caso de realizarse cambios o actualizaciones en la vista o en el modelo del proyecto, en esta capa no se manipula la información de manera directa.

Las funciones de la capa controlador son las siguientes:

- Receptar y recibir las acciones de entrada del usuario.
- Ejecutar reglas de gestión de eventos.

1.2.4 Documentación técnica

A continuación se detallará una guía técnica sobre Spring MVC.

INTRODUCCIÓN A SPRING MVC
Documentación Técnica

ÍNDICE

1. Objetivo del presente documento.....	40
2. Definición de Spring MVC.....	40
3. Objetivos de Spring MVC.....	41
4. Funcionalidades de Spring MVC.....	41
5. Controladores de Spring MVC.....	41
6. Instalación y configuración de Spring MVC.....	42
6.1 Ciclo de vida de un proceso en Spring MVC.....	42
6.2 Configuración de Spring MVC.....	43
6.2.1 Configuración de las dependencias de Spring MVC.....	43
6.2.2 Configuración del DispatcherServlet.....	43
6.2.3 Configuración de un proyecto MVC.....	44

Tipo de documento		
Informativo		
Tema del documento	Revisión	Nº paginas
Introducción a Spring MVC Documentación técnica	1.0	7
Elaborado por:		
Gina Ulloa		
Historial de cambios		
Versión	Fecha	Cambios

1. Objetivo del presente documento

El objetivo de esta documentación técnica de Spring MVC es describir las características y requerimientos técnicos que necesita este framework para el desarrollo de un proyecto de software y también definir conceptos utilizados en esta herramienta para que el usuario tenga al alcance una guía práctica y entendible.

2. Definición de Spring MVC

Spring MVC es un framework flexible y de fácil configuración que posee características similares con otros frameworks, pero hay ciertas distinciones que posee Spring MVC:

- En Spring MVC las capas tienen su prioridad, está basado en interfaces, además brinda plugins y controladores que son configurados como los demás objetos de Spring a través de Ioc⁴.
- Spring MVC no exige utilizar JSP como única tecnología de desarrollo.
- Los web tiers son sencillos de configurar.

Spring MVC es un framework popular de desarrollo que es de código abierto y facilita el desarrollo de aplicaciones Java, su componente central es el DispatcherServlet, este envía solicitudes a los controladores con información de la configuración regional, zona horaria, resolución de vistas, soporte para cargar archivos, el WebApplicationInitializer es una interfaz que proporciona Spring MVC para el desarrollo de proyectos, la biblioteca JSTL se puede agregar al proyecto inicialmente ya que permite dar soporte a los archivos de configuración de los beans XML además permite realizar las siguientes funciones:

- **Finalización de código:** Invoca archivos de configuración XML de Spring MVC para Java.
- **Navegación:** Permite la hipervinculación de las clases, sus propiedades, definiciones y referencias de los beans de Spring MVC.
- **Refactorización:** Permite el cambio de nombre de las referencias de las clases Java en archivos de configuración XML.

⁴ **IoC.**- Inversión de control del bean que controla la creación de instancias.

3. Objetivos de Spring MVC

Spring MVC cumple con objetivos definidos como:

- Permitir que el proceso de elaboración de un proyecto sea más sencillo.
- Brindar información de calidad del proyecto.
- Brindar un proyecto con elaboración uniforme.
- Ofrecer información transparente para todos los usuarios de Maven
- Ayudar en el flujo de trabajo del proyecto.
- Proporcionar facilidades en las actualizaciones de Maven.

4. Funcionalidades de Spring MVC

Spring MVC permite las siguientes funcionalidades:

- Ofrecer una separación clara de Modelo, Vista, Controladores.
- Flexibilidad ya que toda la estructura se implementa mediante interfaces.
- Proveer de interceptores y controllers que ayudan a interpretar y adaptar el comportamiento en el manejo de múltiples requests.
- Los controllers se configuran mediante el IoC juntamente con los demás objetos, lo que permite que sean de fácil testeo e integración con otros objetos que maneje Spring MVC.
- Ofrecer una interfaz bien definida para la capa de negocio.
- Spring MVC ofrece una integración más adaptable con tecnologías distintas a JSP.

5. Controladores de Spring MVC

Un controlador es una secuencia de instrucciones que proporciona una interfaz para utilizar un proyecto o un dispositivo, actúa sobre los datos que son el resultado del procesamiento de la capa modelo.

Las tareas que ejecuta un controlador son las siguientes:

- Interceptar las peticiones HTTP del usuario.
- Procesar cada petición en una operación válida para ser ejecutada.
- Seleccionar la vista adecuada para mostrar al usuario.
- Devolver los resultados al cliente.

El analizador de componentes deberá seleccionar automáticamente las clases con la definición de @Controller y será declarado como un bean⁵ para las aplicaciones Spring MVC.

JavaBean.- Son componentes de software reutilizables que contienen atributos y los métodos get y set que sirven para acceder y modificar los atributos de las clases con eventos que permitirán comunicarse con otros JavaBean, deberá tener las siguientes características:

- Todos los atributos son privados.
- Deberá tener al menos un constructor público por defecto.
- Tendrá métodos set y get de atributos privados.
- Tendrán implementación de objetos codificados en un medio de almacenamiento.

6. Instalación y configuración de Spring MVC

6.1 Ciclo de vida de un proceso en Spring MVC

Cada vez que el cliente interactúa en el proyecto internamente se envía una solicitud, esta solicitud realiza varias paradas en cada una entrega información y recoge otro tipo de información, a continuación se ilustra este proceso:

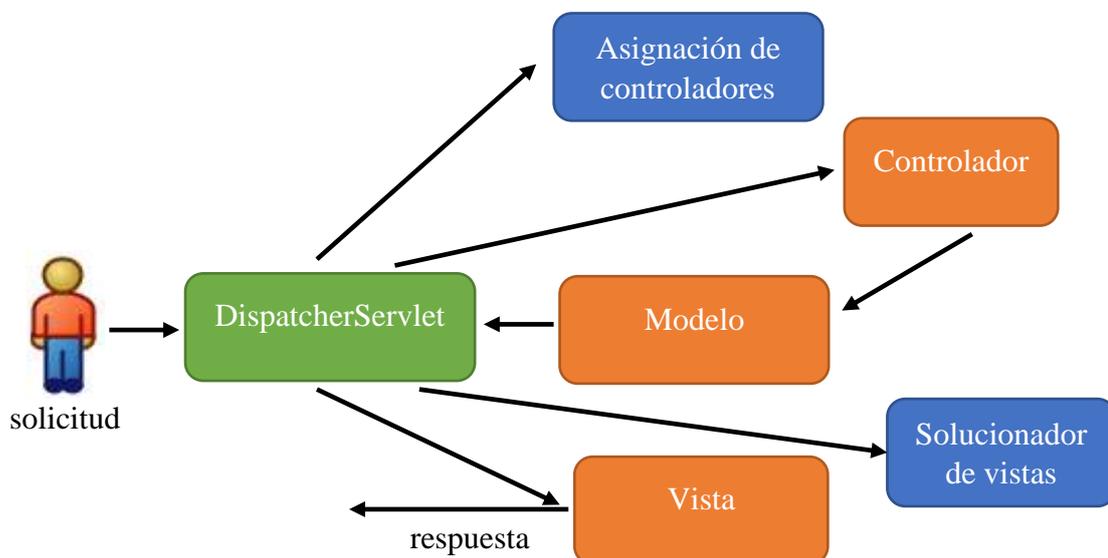


Figura N° 25 Ciclo de vida de un proceso en Spring MVC
Elaborado por: Gina Ulloa

⁵ **Bean.-** Es un objeto que está instanciado, ensamblado y gestionado por un contenedor Spring IoC. Los beans y las dependencias entre ellos se reflejan en los metadatos de configuración utilizados por un contenedor. **Fuente:** [<https://docs.spring.io/spring/docs/4.0.x/spring-framework-reference/htmlsingle/#mvc-features>]

La solicitud es enviada con la información requerida por el cliente, esta solicitud va a incluir una URL o un formulario, la información llega hasta el dispatcherServlet que es el controlador frontal y este deberá tomar la decisión a que controlador enviar la solicitud, éste controlador tomará muy en cuenta la URL y deberá procesar parte de la información que se muestra en el navegador, una de las tareas que también lleva acabo el controlador es empaquetar los datos de la capa de modelo e identificará el nombre de la vista que deberá generar el resultado, una vez completado todo este proceso interno se envía la solicitud juntamente con el modelo y el nombre de la vista de nuevo al dispatcherServlet sin embargo este ya tiene conocimiento que vista va a procesar el resultado, por último la implementación de la vista donde se entrega los datos del modelo, la vista va a utilizar los datos para generar resultados y presentarlos al cliente.

El dispatcherServlet utiliza un viewResolver para asignar el nombre en una vista concreta que puede ser un JSP, XSTL, Tiles, Velocity entre otros, también utiliza la vista para mostrar el modelo al cliente. Spring MVC proporciona implementaciones de algunas interfaces como son: HandlerMapping, Controller, View Resolver, View, del paquete springframework que permiten definir una tabla para asociar URLs con los controladores.

6.2 Configuración de Spring MVC

6.2.1 Configuración de las dependencias de Spring MVC

Antes de empezar es necesario configurar el archivo POM para empezar a usar Spring MVC

```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>${spring-framework.version}</version>
</dependency>
```

6.2.2 Configuración del DispatcherServlet

Este archivo es el encargado de dirigir las solicitudes a través de los numerosos componentes de Spring MVC, el dispatcherServlet crea un entorno de la aplicación de Spring que incluye los bean declarados RootConfig y SecurityConfig toda esta configuración se almacena en el webConfig, Spring MVC tiene el archivo

ContextLoaderListener que es el archivo raíz donde se carga el contenido de la aplicación, `getRootConfigClasses()` es un método que permite cargar el contexto de la aplicación.

Para que se pueda habilitar Spring MVC se deberá utilizar la sentencia `@EnableWebMvc` del paquete `org.springframework.web.servlet.config.annotation` que será configurada en la clase `WebConfig`.

6.3.3 Configuración de un proyecto MVC

Para la configuración de un proyecto Spring MVC es necesario realizar algunos pasos que no son tan complicados, para iniciar se deberá realizar los siguientes pasos:

- Primero deberá dar nombre al proyecto a crear.
- El programador deberá seleccionar el servidor de aplicaciones con el cual se ejecutará el proyecto.
- A continuación deberá escoger el framework en el cual desea trabajar en este caso será Spring Web MVC.
- A continuación deberá seleccionar las librerías para Spring MVC como se muestra en la figura a continuación:

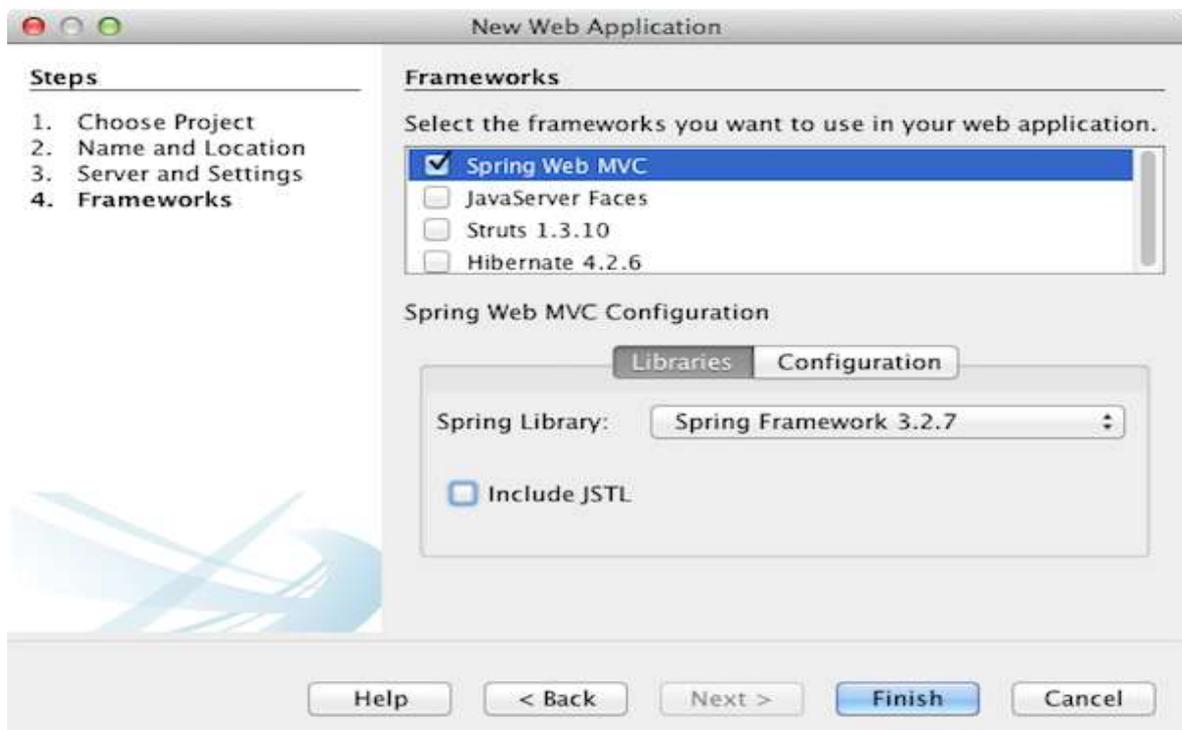


Figura N° 26 Configuración de un proyecto Spring MVC
Elaborado por: Gina Ulloa

A continuación se deberá configurar el nombre y la asignación del DispatcherServer:

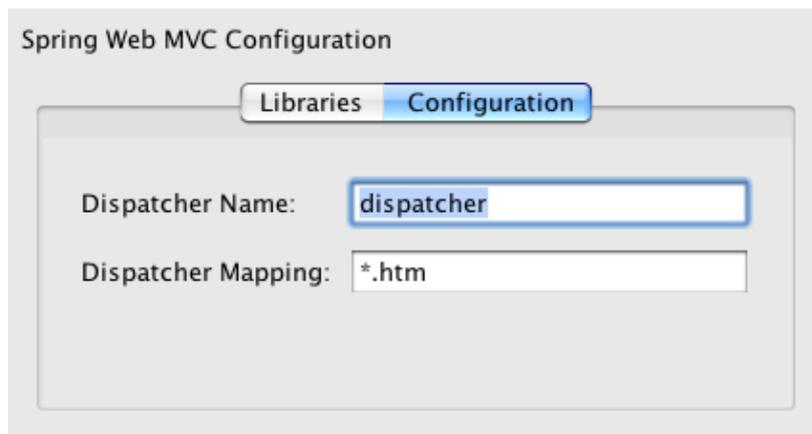


Figura N° 27 Configuración de DispatcherServer
Elaborado por: Gina Ulloa

Una vez finalizado este proceso el IDE crea un proyecto en donde incluye metadatos, scripts de compilación, dispatcherServlet.xml, applicationContext.xml, a continuación se muestra un despliegue de librerías agregadas para que el proyecto trabaje con facilidad:

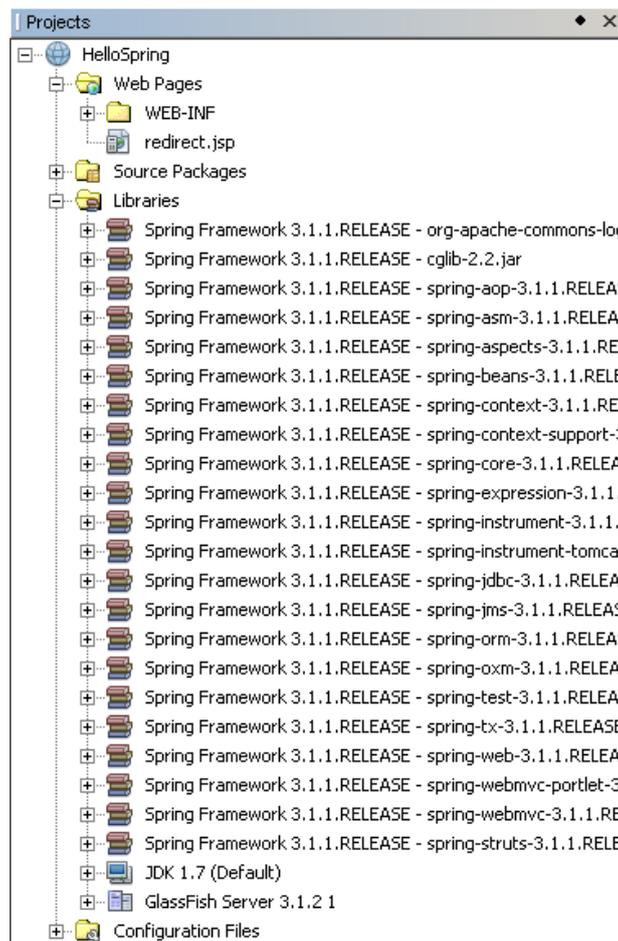


Figura N° 28 Librerías utilizadas por Spring MVC
Elaborado por: Gina Ulloa

Una vez configurado las librerías necesarias se puede ejecutar, el proyecto se compila y se implementa el servidor para probar que el proyecto creado está funcionando correctamente, como se muestra a continuación:

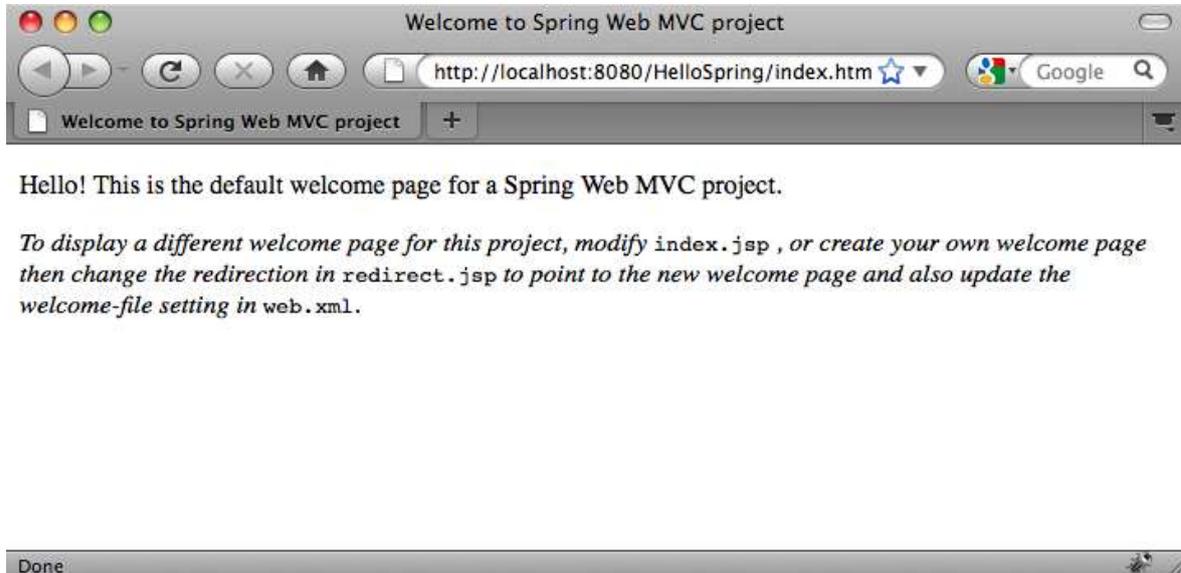


Figura N° 29 Página de bienvenida de Spring MVC
Elaborado por: Gina Ulloa

La aplicación que se ejecuta se compone de dos páginas a las que se puede hacer referencia como vistas en el Spring MVC, la primera vista tendrá un formulario HTML con un campo de entrada que solicitará el nombre del usuario y una contraseña es decir se ejecutará un login, en la segunda vista que es una página donde se mostrará la página principal que contiene las opciones del proyecto; las vistas son administradas por un controlador este recibe solicitudes de la aplicación y decide qué vistas devolver, al modelo pasa la información que se necesitará mostrar una vez que la información haya pasado a las vistas. Para los proyectos web complejos, la lógica de negocios no estará contenida directamente en el controlador, este utilizará otra entidad denominada servicio cuando necesite realizar alguna lógica de negocios, para la configuración de las vistas es necesario configurar un bean que será el encargado de crear de manera automática las asignaciones de URL para todos los controladores registrados.

1.2.5 Integración de Maven con Spring MVC

Actualmente las aplicaciones son escalables, la fusión de varias tecnologías de software permite el desarrollo de proyectos web más robustos y sólidos, para la presente investigación se propone integrar **Maven** que permite la estandarización en la estructuración de los proyectos con **Spring MVC** que es un conjunto de librerías portables y que es muy funcional permite desarrollar proyectos de calidad, el objetivo de esta integración es hacer que los proyectos web sean más flexibles e interoperables, su código reutilizable y entendible.

Para el presente estudio se propone el siguiente entorno:

- Un computador Hp Compaq 8200 Elite, core i5, 1GB RAM, 1TB en disco.
- Windows 10
- Spring Tool Suite 3.9.6
- Maven 3.3.9
- Apache Tomcat
- Spring MVC

1.3 SPRING SECURITY

Spring Security es un framework que brinda autenticación y autorización a las aplicaciones, el objetivo es proporcionar seguridad a los proyectos web mediante la comprobación de identidad del usuario y los respectivos permisos a los roles de usuario para el acceso a los diferentes módulos del proyecto.

De acuerdo a (**Pivotal, 2018**) menciona lo siguiente: Spring Security es un framework que se enfoca en proporcionar tanto autenticación como autorización para aplicaciones Java. Como todos los proyectos de Spring, el verdadero poder de Spring Security se encuentra en la facilidad con que se puede extender para cumplir con los requisitos personalizados.

El autor (**Mularien, 2010**) menciona los beneficios que Spring Security le permite ejecutar al usuario:

- Segmentar a los usuarios del sistema en clases de usuario.
- Asignar niveles de autorización a roles de usuario.

- Asignar roles de usuario a clases de usuario
- Aplicar reglas de autenticación global a través de los recursos de la aplicación.
- Aplicar reglas de autorización en todos los niveles de la arquitectura de la aplicación.
- Prevenir ataques comunes destinados a manipular o robar la sesión de un usuario. (pag. 35)

1.3.1 Características

- Fácil integración con Spring web MVC.
- Protección segura contra hacking, falsificación de identidad, accesos indebidos al proyecto, entre otros.
- Spring security es configurable de manera manual y automática; proporciona canales de seguridad automáticos que se integra con JCapcha.
- Spring security soporta la integración con las tecnologías HTTP, LDAP, JAAS, entre otras.
- Este framework es muy aceptable en los desarrolladores por su flexibilidad en los métodos de autenticación, aporta con autorizaciones en base de solicitudes web, autorizaciones en base a llamadas y autorizaciones en base al acceso a instancias.

Según el autor (**Mularien, 2010**) asegura que: Spring reúne todo lo que necesita para implementar una Solución de seguridad de aplicación inferior de manera concisa y sensible. Además, Spring Security ofrece integración inmediata con muchos sistemas de autenticación Enterprise comunes; por lo que es adaptable a la mayoría de las situaciones con poco esfuerzo por parte del desarrollador. (pag. 35)

1.3.2 Documentación técnica

A continuación se detallará una guía técnica sobre Spring Security.

INTRODUCCIÓN A SPRING SECURITY
Documentación Técnica

ÍNDICE

1. Objetivo del presente documento.....	51
2. Definición de Spring Security.....	51
3. Módulos de Spring Security.....	51
4. Protección de solicitudes web.....	52
4.1 Seguridad del canal.....	53
4.2 Protección de los métodos.....	53
5. Servicios para los usuarios de Spring Security.....	55

Tipo de documento Informativo		
Tema del documento Introducción a Spring Security Documentación técnica	Revisión 1.0	Nº paginas 5
Elaborado por: Gina Ulloa		
Historial de cambios		
Versión	Fecha	Cambios

1. Objetivo del presente documento

El objetivo de esta documentación técnica de Spring Security es describir las características y requerimientos técnicos que necesita este framework para el desarrollo de un proyecto de software y también definir conceptos utilizados en esta herramienta para que el usuario tenga al alcance una guía práctica y entendible.

2. Definición de Spring Security

Spring Security brinda seguridad a los datos que en la actualidad son muy valiosos, en los proyectos desarrollados es necesario seguir lineamientos para proteger la información almacenada, Spring Security brinda seguridad integral para la autenticación, la autorización de roles, solicitudes web y la invocación de métodos, Spring Security utiliza filtros de servlets para realizar la protección de solicitudes web y la restricción de URL, también utiliza AOP de Spring para proteger las invocaciones a los métodos mediante la aplicación de proxy sobre los objetos.

De acuerdo al autor (**Mularien, 2010**) define lo siguiente: Spring Security proporciona una gran cantidad de recursos que permiten que muchas prácticas de seguridad comunes se declaren o configuren de manera directa y darnos confianza de que nuestra aplicación de comercio electrónico es segura. (pag. 35)

3. Módulos de Spring Security

Spring Security contiene once módulos muy importantes, a continuación se describirá los siguientes:

- **ACL:** Este módulo permite el uso de la seguridad en los objetos de dominio mediante Listas de Control de Acceso.
- **Aspectos:** Es un módulo que permite admitir aspectos de AspectJ a la hora de utilizar apuntes de Spring Security.
- **Biblioteca de etiquetas:** En este módulo se encuentra la biblioteca de etiquetas JSP de Spring Security.
- **Cliente CAS:** Permite la integración con el Servicio de Autenticación Central de Jasig.
- **Criptografía:** Este módulo permite utilizar encriptación y codificación mediante contraseñas.

- **Configuración:** Este módulo permite configurar Spring Security con XML y Java.
- **Dispositivos remotos:** Este módulo permite la integración de Spring remoting.
- **Web:** Este módulo permite proporcionar soporte para la seguridad Web que está basada en los filtros de Spring Security.

A continuación se mostrará la configuración del archivo POM con sus respectivos módulos:

```
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-web</artifactId>
    <version>${spring-security.version}</version>
</dependency>
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-config</artifactId>
    <version>${spring-security.version}</version>
</dependency>
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-taglibs</artifactId>
    <version>${spring-security.version}</version>
</dependency>
```

4. Protección de solicitudes web

Spring Security permite la configuración de seguridad del proyecto mediante filtros de servlet, con la finalidad de proteger las peticiones web y solicitudes a nivel de URL, el filtro especial DelegatingFilterProxy verificará las peticiones entrantes en el proyecto y las enviará al bean con el identificador springSecurityFilterChain perteneciente al filtro especial FilterChainProxy permite conectar entre si uno o más filtros adicionales que permiten brindar seguridad al framework Spring; es obligatorio configurar un bean que sea implementado por WebSecurityConfigurer a continuación se mostrará la configuración:

```
package com.ventas.configuracion;
```

```
import org.springframework.security.web.context.AbstractSecurityWebApplicationInitializer;
import org.springframework.security.config.annotation.web.configuration;

public class SecurityWebInitializer extends AbstractSecurityWebApplicationInitializer {
}
```

En un proyecto de software no todas las peticiones que realiza el usuario deberán protegerse de la misma manera, por ejemplo algunas solicitudes no necesariamente necesitan autenticación, otras solicitudes requerirán de un determinado rol de usuario, para definir la seguridad en cada petición que realiza el usuario se deberá configurar el método `configure()`.

4.1 Seguridad del canal

Spring Security brinda otra funcionalidad como es la de reforzar la seguridad del canal por donde viajan los datos, por defecto la información viaja a través de la red sin codificación alguna, de esta manera puede ser susceptible a diversos ataques de haking y robos como por ejemplo los cobros con las tarjetas de crédito, una solución a este problema es enviar la información cifrada a través del canal https, cuando los proyectos son utilizados dentro de la intranet no es tan obligatorio restringir la seguridad del canal.

4.2 Protección de los métodos

Una de las ventajas de Spring Security es brindar seguridad a los métodos de un proyecto que consiste en aplicar reglas de seguridad especiales, existen tres tipos de instrucciones para proteger los métodos:

- **Secured:** Este método acepta una matriz de elementos String, cada uno de estos es una autorización para invocar un método; cuando un usuario no autenticado trata de ingresar al sistema e invoca este método automáticamente se genera una de las excepciones de seguridad que brinda Spring Security, para habilitar este tipo de seguridad es necesario realizar la siguiente configuración:

```
EnableGlobalMethodSecurity(securedEnabled=true)
public class MethodSecurityConfig extends GlobalMethodSecurityConfiguration {
}
```

- **RolesAllowed:** Este método de seguridad es igual al método Secured, la diferencia es que el método RollesAllowed es uno de los métodos característicos de Java, en ciertas ocasiones las restricciones de seguridad dependen de más factores que los privilegios concretos de un usuario, a continuación se muestra la configuración de este método:

```
EnableGlobalMethodSecurity(jsr250Enabled=true)
public class MethodSecurityConfig extends GlobalMethodSecurityConfiguration {
}
```

- **PreAuthorize, PostAuthorize, PreFilter y PostFilter:** La protección con estos métodos permite restricciones completas mediante el uso de SpEL que es un lenguaje de expresión de Spring mediante la invocación de métodos y la funcionalidad básica de plantillas de cadenas, existen otros lenguajes de expresión de Java disponibles como son: OGNL, MVEL, JBossEL, SpEL se creó con la finalidad de proporcionar a la comunidad un lenguaje de expresión único y seguro, a continuación se describirá a cada uno de ellos:

PreAuthorize: Permite restringir el acceso a un método antes de que sea invocado en función del resultado al evaluar una expresión.

PostAuthorize: Permite invocar un método que genera una excepción de seguridad cuando la expresión a evaluar es falso.

PreFilter: Permite invocar un método primero filtra la información de entrada antes de acceder al método.

PostFilter: Permite invocar un método pero primero filtra los resultados en función de una expresión.

Si la expresión ingresada es verdadera se cumple con la regla de seguridad y accede al método requerido, estos métodos de seguridad aceptan expresiones SpEL con su respectivo parámetro de entrada, a continuación se mostrará la configuración:

```
EnableGlobalMethodSecurity(prePostEnabled=true)
public class MethodSecurityConfig extends GlobalMethodSecurityConfiguration {
}
```

5. Servicios para los usuarios de Spring Security

Es obligatorio que exista un repositorio de usuarios donde se almacene nombre de usuarios, contraseñas, decisiones de autenticación y datos alternos, Spring Security es muy flexible y brinda varias opciones:

- **Repositorio de usuarios en memoria.**- muy útil en tareas de depuración y pruebas de desarrollo pero no recomendable en pruebas de producción.
- **Repositorio de usuarios en una base de datos relacional.**
- **Repositorio de usuarios en un directorio LDAP.**- muy útil si se tiene un servicio de directorio y no se quiere conservar las cuentas de usuario y contraseñas adicionales para el acceso.
- **Repositorio de usuarios personalizado.**- muy útil cuando las anteriores opciones no satisfacen las expectativas del cliente en cuanto al acceso web se hace necesario crear y configurar un servicio personalizado en los detalles del usuario.

1.4 SERVICIOS DE PAGO ONLINE MÁS SEGUROS EN INTERNET

Los pagos más seguros online son aquellos que utilizan software, páginas y herramientas de cifrado, el objetivo de esta forma de pago es ofrecer seguridad y confianza tanto para el usuario como para el vendedor, existen diferentes formas de pago seguras sin el uso de tarjetas de crédito a continuación se describe a las siguientes:

PayPal: empresa dedicada al pago y envío de dinero mediante internet, esta plataforma evita compartir la información financiera del usuario, es un método de pago online seguro porque utiliza tecnología de encriptación SSL de 128 bits, una de las ventajas es que brinda un servicio gratuito.

Google Pay Send: es una forma de pago seguro online creada por Google disponible también para dispositivos móviles con sistemas operativos Android, IOS; protege la información del usuario con ciertos niveles de seguridad y utiliza una de las infraestructuras más actuales y avanzadas del mundo, una de las desventajas de esta forma de pago es que solo se encuentra disponible en Estados Unidos y Reino Unido.

Amazon Payments: es una plataforma que cuenta con políticas de protección de datos en los pagos que se realizan a través de internet, posee una tecnología de protección avanzada contra fraudes y estafas online, una de las ventajas de esta plataforma es que el usuario puede hacer uso de las credenciales de Amazon.

CAPÍTULO II

DESARROLLO

Una empresa es productiva cuando optimiza el desarrollo de sus proyectos de software y lleva definida una estructura de trabajo óptima lo que permite brindar productos de calidad, es necesario utilizar una buena metodología en la gestión de los proyectos desarrollados para conseguir resultados óptimos, sin embargo, los usuarios ejecutan procesos a cada instante es por esto la importancia de unificar los procesos y registrarlos con la documentación adecuada.

El desarrollo de un proyecto de software conlleva varias etapas desde su creación hasta su finalización; con la finalidad de que la productividad del proyecto incremente su rendimiento conforme recibe actualizaciones o implementaciones es necesario utilizar estándares de calidad basados en la funcionalidad y el rendimiento centrado principalmente en las necesidades y peticiones del cliente, estas características implementadas permitirán asegurar el éxito del proyecto y de la empresa.

2.1 ESPECIFICACIONES TÉCNICAS

Las especificaciones técnicas representan las características y aspectos más importantes que tiene un proyecto de software, las **especificaciones descriptivas** tratan sobre las características medibles en el proyecto, en cambio las **especificaciones funcionales** o de rendimiento se utilizan cuando se usan prototipos, las especificaciones deben mostrar las características principales del proyecto desarrollado, el desarrollador deberá evitar las especificaciones innecesarias y también deberá redactar las especificaciones de manera clara y precisa.

2.1.1 Descripción del proyecto

Bazar y novedades Guayas requiere la implementación de un sistema de comercio electrónico que permita lo siguiente:

- El proyecto permitirá mostrar un catálogo y el stock de los productos que hay en existencia en el bazar.
- Deberá permitir realizar ventas a través de internet.
- Deberá permitir realizar un inventario de la mercadería existente en el bazar.

- El proyecto permitirá realizar pagos por vía online.

2.1.2 Aspectos funcionales

El proyecto de comercio electrónico deberá contar con las siguientes funcionalidades:

- El software deberá permitir alertar al administrador sobre el stock cero de los productos para su respectivo reaprovisionamiento.
- El software deberá permitir alertar transacciones o compras inusuales.
- El software deberá alertar al administrador del sistema sobre posibles ataques de espionaje o malware.
- El software deberá alertar al administrador sobre posibles riesgos de fraude en los pagos online o con tarjetas de débito y crédito.
- El software deberá ser adaptable, flexible y escalable donde se podrá incrementar módulos adicionales de acuerdo a las necesidades del cliente.

2.1.3 Aspectos operativos

- El software deberá contar con seguridad mediante los perfiles de usuario.
- El usuario contará con un manual de manejo del sistema.

2.1.4 Aspectos técnicos

- El proyecto se podrá instalar en Windows o Linux.
- El proyecto será desarrollado en un entorno web.
- El proyecto deberá utilizar una base de datos sin licenciamiento.

2.1.5 Aspectos del proyecto

- El proyecto es de licenciamiento libre.
- Para la creación e implementación del proyecto se utilizará una metodología de desarrollo rápido.
- El software del proyecto deberá ser mantenible y actualizable.
- El proyecto deberá ser fácil, entendible y sencillo en su manejo.

2.2 PLANIFICACIÓN

Para la planificación de un proyecto de software es necesario definir el ámbito, las expectativas de la planificación, identificar los requerimientos de la empresa y elaborar un plan de desarrollo de software para establecer las estimaciones del proyecto, también es importante identificar las necesidades de la empresa y definir el alcance del proyecto para cuando la empresa necesite incrementar a futuro más módulos en el proyecto y no tenga inconvenientes en implementarlos; la arquitectura utilizada en el desarrollo del proyecto será definida en los requerimientos, el programador desarrollará un prototipo piloto para una futura revisión con los interesados en el proyecto.

Los objetivos de la planificación en los proyectos de software son:

- Elaborar estimaciones antes de empezar a elaborar el proyecto con la finalidad de delimitar alcances y objetivos del software planteado.
- Definir escenarios para limitar los requerimientos del cliente.
- Proporcionar un análisis de los requerimientos del proyecto con la finalidad de que el programador pueda realizar estimaciones de recursos, costos, tiempos, entre otros.
- Plantear un plan de aseguramiento de calidad del software, también es necesario incluir un plan de etapas de entregas, un plan de requerimientos, plan de gestión de cambios.

2.2.1 Plan de desarrollo de software.- Su objetivo principal es recabar toda la información importante para un correcto desenvolvimiento del proyecto, describiendo de esta manera el enfoque que persigue el proyecto, el plan de desarrollo de software es utilizado por:

- **El gestor de proyectos** para la respectiva planificación, supervisión y los requerimientos de recursos y
- **Los integrantes del equipo de desarrollo** que usan este plan para tener conocimiento de los requerimientos, cronograma, fases y actividades del proyecto.

2.2.2 Plan de gestión de requerimientos.- Este plan permite documentar la información más relevante para gestionar los requerimientos de un proyecto de software empezando desde la etapa de inicio hasta la etapa de cierre o entrega del proyecto, este plan de gestión se divide en tres partes que son:

- **Uso del plan de gestión de requerimientos.-** Es importante el uso del plan propuesto para que el proyecto tenga el éxito esperado.
- **Visión general de la gestión de requerimientos.-** Aquí se requiere de organización y designación de responsabilidades para realizar las tareas asignadas en el flujo de requerimientos, aquí es necesario plantear las herramientas de software, el entorno de hardware y la infraestructura que se utilizará para la elaboración del proyecto en todo su ciclo de vida.
- **Administración de los requerimientos.-** Es de vital importancia definir los detalles o restricciones de los requerimientos del cliente, el programador deberá realizar la **trazabilidad de requerimientos** que es la descripción del ciclo de vida de los requerimientos, los **flujos de trabajo y actividades** también se definirán en el plan de gestión de requerimientos y por último los **cambios de gestión** también se deberán documentar los cambios, revisiones, entregas, problemas detectados, todos estos aspectos se deberán tomar en cuenta para que el plan de gestión de requerimientos aplicado en el software propuesto tenga el éxito esperado.

2.2.3 Plan de aseguramiento de la calidad del software.- Este plan es un conjunto de técnicas y métodos que permiten determinar la calidad del software a elaborar, esta herramienta es muy importante en el desarrollo de proyectos, muchas empresas no aplican este plan porque necesitan más recursos humanos, económicos y adaptaciones a nuevos estándares de calidad, el objetivo que persigue la aplicación de este plan es gestionar la calidad en el desarrollo de un proyecto mediante métricas que permiten mejorar los procesos internamente y valorar resultados y así, de esta manera satisfacer las necesidades del cliente.

2.2.4 Métricas de calidad de software.- Permiten ajustar los requerimientos del cliente con criterios de alta calidad ajustables al proyecto elaborado, estas métricas permiten evaluar la calidad del código fuente, los casos de prueba creados, la aplicación de las métricas son en tiempo real y los resultados esperados son: efectividad en las tareas programadas y garantías de calidad en el proyecto elaborado.

Para el presente trabajo de investigación se utilizó las siguientes métricas que se describen a continuación:

MÉTRICA	CARACTERÍSTICA	SUB-CARACTERÍSTICA	DESCRIPCIÓN
Funcionalidad	Las funciones y propiedades del sistema satisfacen las necesidades explícitas del cliente	Adecuación	<ul style="list-style-type: none"> • Contiene las funciones adecuadas para ejecutar las tareas especificadas
		Exactitud	<ul style="list-style-type: none"> • El sistema ejecuta lo acordado con el cliente en forma correcta.
Confiabilidad	El proyecto mantiene los niveles de rendimiento bajo ciertas condiciones y tiempos	Tolerancia a errores	<ul style="list-style-type: none"> • Comportamiento del sistema en caso de suceder una falla
		Entendimiento	<ul style="list-style-type: none"> • El sistema tiene una estructura y diseño entendible
Usabilidad	El proyecto es fácil de aprender y manejar	Aprendizaje	<ul style="list-style-type: none"> • El sistema es de fácil manejo
		Operabilidad	<ul style="list-style-type: none"> • El sistema es controlable con facilidad
		Atracción	<ul style="list-style-type: none"> • El diseño del software es atractivo para el cliente
		Cambiable	<ul style="list-style-type: none"> • El sistema es adaptable y modificable preguntar

Tabla N° 2 Métricas y características

Fuente: (Alberto Largo, 2005)

2.2.5 Plan de etapas de entregas.- Este plan consiste en la planificación de las etapas del ciclo de vida de un proyecto de software, aquí el jefe de proyectos ya debió considerar recursos, actividades, tareas, tiempos de entregas tanto de documentos como del proyecto en sí, este plan consiste de cinco etapas que se describirán a continuación:

- **Fase de inicio:** Esta fase es importante antes de empezar a elaborar un proyecto, aquí se definen los alcances, limitaciones, actividades, tareas, equipo de desarrollo, solo con la definición clara de lo que se quiere obtener se logrará el éxito del proyecto.
- **Fase de planificación:** Es la etapa más difícil en la elaboración del proyecto ya que en esta etapa se toman las decisiones más importantes en relación a presupuestos, tiempos

de entregas, recursos tecnológicos, equipo de desarrollo a utilizar, es decir, la planificación de actividades, comunicados se planifican en esta fase.

- **Fase de ejecución:** En esta fase se ejecutan las decisiones y actividades propuestas en la fase de planificación mediante la ejecución y entrega del proyecto, también es importante administrar los recursos, presupuesto y tiempos de entrega, monitorear cambios, actualizaciones, modificaciones y posibles fallas que llegarán a surgir a futuro.
- **Fase de seguimiento:** En esta fase se realiza el monitoreo, seguimiento y control del sistema con la finalidad de detectar posibles fallas en algún módulo del sistema o realizar cambios sugeridos en la planificación.
- **Fase de cierre:** En esta fase se completa el ciclo de vida del proyecto de software, una vez que se han depurado fallas y se han completado los requerimientos de software, se hacen las entregas respectivas del proyecto propuesto.

2.3 DESARROLLO

El desarrollo de un producto de software puede entenderse como un proceso de producción de un programa informático que cumple los requerimientos del cliente y se desarrolla a través de una metodología, para el presente trabajo de investigación se aplicará la metodología de desarrollo ágil SCRUM.

2.3.1 Metodología de desarrollo Scrum

Esta metodología de desarrollo ágil es de fácil aplicación pero requiere de esmerado trabajo para lograr el objetivo, el programador debe adaptarse a las circunstancias dependiendo de la evolución del proyecto, esta metodología no sigue un plan establecido, uno de los objetivos de Scrum es dividir el desarrollo del proyecto en ciclos de trabajo cortos denominados Sprints.

Para el autor (**Satpathy, 2016**) considera esta metodología de la siguiente manera: Scrum es una de las metodologías ágiles más populares. Es una metodología de adaptación,

iterativa, rápida, flexible y eficaz, diseñada para ofrecer un valor significativo de forma rápida en todo el proyecto. Scrum garantiza transparencia en la comunicación y crea un ambiente de responsabilidad colectiva y de progreso continuo. El marco de Scrum está estructurado de tal manera que es compatible con los productos y el desarrollo de servicio en todo tipo de industrias y en cualquier tipo de proyecto, independientemente de su complejidad. (pág. 2)

De acuerdo a los autores (**Ken Schwaber, 2017**) Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el trabajo en productos complejos desde principios de los años 90. Scrum no es un proceso, una técnica o método definitivo. En lugar de eso, es un marco de trabajo dentro del cual se pueden 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 podamos mejorar continuamente el producto, el equipo y el entorno de trabajo. El marco de trabajo Scrum consiste en los Equipos Scrum y sus roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso. (pág. 3)

Un **sprint** son cortos periodos de tiempo en los cuales se desarrolla el proyecto, es decir son las entregas que realiza el programador para su respectiva revisión.

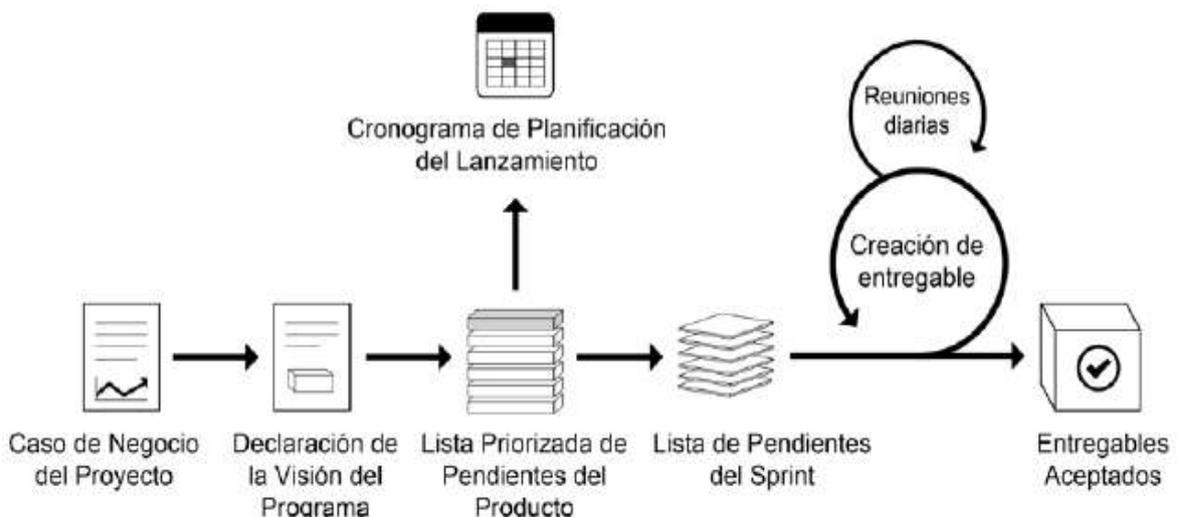


Figura N° 30 Flujo de Trabajo de Scrum
Fuente: Una guía para el Conocimiento de Scrum.pdf

2.3.2 Características de Scrum

Entre las características más relevantes se encuentran las siguientes:

- Adaptabilidad
- Proceso de desarrollo eficiente
- Transparencia
- Resolución de problemas de manera más rápida y eficiente
- Centrado en el cliente
- Una de las desventajas de Scrum es que no genera la documentación necesaria como otras metodologías de desarrollo.

2.3.3 Principios de Scrum

- Control de proceso empírico.- Lo realiza mediante tres ideas principales que son: transparencia, inspección y adaptación.
- Auto organización.- Equipo de desarrollo con compromiso y responsabilidad.
- Colaboración.- Se centra en el trabajo colaborativo de tres dimensiones básicas que son: articulación, conciencia y apropiación.
- Priorización basada en valores.- Enfoque de Scrum para ofrecer el máximo valor del negocio.
- Tiempo asignado.- Restricción limitante en Scrum.
- Desarrollo interactivo.- Enfatiza cómo manejar los cambios y crear productos que cumplan con los requerimientos del cliente.

2.3.4 Visión general del proceso

Los elementos que conforman la metodología de desarrollo Scrum son:

- Reuniones.- que se pueden considerar tres tipos: planificación de sprint, reunión diaria, revisión de sprint.
- Artefactos.- consta de tres tipos: product backlog, print backlog, grafica de progreso.
- Roles.- consta de tres tipos de roles: equipo scrum, scrum master, dueño del producto.

2.3.5 Fases de la metodología de desarrollo Scrum

Scrum está compuesto de 5 fases que se cumplirán en el desarrollo de un proyecto de software:

- **Fase 1:** Definir el backlog del proyecto
- **Fase 2:** Planificar el backlog
- **Fase 3:** Scrum diario
- **Fase 4:** Revisión del sprint
- **Fase 5:** Retrospectiva del sprint

Esta metodología incorpora al cliente como un miembro más del equipo, de este modo los requerimientos aparecen dentro de las denominadas historias de usuario.

2.3.5.1 Fase 1: Definir el backlog del proyecto

El backlog es el corazón de Scrum, conocido también como historial de usuario, para la presente investigación se ha determinado realizar el caso de uso denominado: “Aplicación para comercio electrónico”, el cliente solicitante es Bazar y Novedades Guayas, representado por la Sra. Bertha Campoverde.

a. Descripción del proyecto

Bazar y Novedades Guayas es una microempresa familiar, con el pasar del tiempo ha ido creciendo hasta convertirse en uno de los bazares más concurridos, actualmente las ventas se las realiza de manera tradicional y al contar el local con una amplia variedad de productos es difícil que el consumidor final pueda conocer la amplia disponibilidad de inventario q se proporciona, por esta razón mucha mercadería se encuentra estancada, aunque esta forma de venta permite una mayor comunicación e interacción entre el vendedor y el consumidor disminuye potencialmente el crecimiento de la empresa, reduciendo así las ganancias que podría percibir anualmente.

Debido a ello, la dueña de la empresa desea que se automatice el proceso de venta mediante una aplicación, la misma que cumplirá con el objetivo de ofertar el catálogo de productos online, ya que de esta manera se amplía la cobertura y se extiende el mercado.

b. Definición de la pila del producto

En la primera fase de la metodología Scrum, se definirá el product backlog, la misma que es una lista de requerimientos del usuario con su importancia, la cual es proporcionada por el dueño del producto, a continuación, se mostrará los requerimientos funcionales:

REQUERIMIENTOS FUNCIONALES ROL ADMINISTRADOR

- **Administrar productos**

- Historia de Usuario 1 (HU01): Yo como administrador quiero poder ingresar productos nuevos, eliminar productos o actualizar productos para tener actualizado el catálogo.

- **Visualizar detalles de productos**

- Historia de Usuario 2 (HU02): Yo como administrador quiero poder visualizar todos los productos con sus respectivas características para conocer el catálogo de productos.

- **Administrar los pedidos de los usuarios**

- Historia de Usuario 3 (HU03): Yo como administrador quiero poder administrar los pedidos de los usuarios para su aprobación o cancelación.

- **Administrar el stock de los productos**

- Historia de Usuario 4 (HU04): Yo como administrador quiero poder visualizar el stock de los productos para poder surtir los mismos.

REQUERIMIENTOS FUNCIONALES ROL USUARIO

- **Registrarse en el sistema**

- Historia de Usuario 5 (HU05): Yo como usuario quiero poder registrar datos personales en el sistema para poder realizar compras.

- **Comprar productos**

- Historia de Usuario 6 (HU06): Yo como usuario quiero poder realizar compras del catálogo de productos del sistema para poder satisfacer las demandas.

- **Administrar la información de mi cuenta**
 - Historia de Usuario 7 (HU07): Yo como usuario quiero poder administrar la información registrada en el sistema de mi cuenta para poder actualizar, eliminar o visualizar mi información

- **Conocer el listado de los pedidos realizados**
 - Historia de Usuario 8 (HU08): Yo como usuario quiero poder visualizar los pedidos que he realizado con una descripción detallada para conocer las compras realizadas.

- **Conocer el listado de los productos en mi carrito de compras**
 - Historia de Usuario 9 (HU09): Yo como usuario quiero poder visualizar los productos que he añadido a mi carrito de compras para conocer mis posibles compras.

- **Pagar mis compras**
 - Historia de Usuario 10 (HU10): Yo como usuario quiero poder pagar mis compras de forma segura para evitar que roben mi información

A continuación se muestra la pila del producto con la importancia indicada por el dueño del producto.

PILA DEL PRODUCTO		
ID	Descripción de requerimiento	Importancia
1	Administrar productos	80
2	Visualizar detalles de productos	50
3	Administrar los pedidos de los usuarios	50
4	Administrar el stock de los productos	20
5	Registrarse en el sistema	13
6	Comprar productos	40
7	Administrar la información de la cuenta	13
8	Conocer el listado de los pedidos realizados	13
9	Conocer el listado de los productos en el carrito de compras	20
10	Pagar las compras	40

Tabla N° 3 Definición Pila del producto
Elaborado por: Gina Ulloa

2.3.5.2 Fase 2: Planificación del sprint

Esta fase es importante para Scrum porque de ello depende el éxito del sprint, el objetivo es brindar información suficiente y relevante al equipo de desarrollo, una de las ventajas de la planificación de los sprints es la fecha de revisión porque allí se presentan prototipos demostrativos; previo a llevar a cabo la reunión de planificación del Sprint, es necesario tener definido la pila del producto con la priorización correcta ya que esto nos permitirá posteriormente realizar la estimación correcta de los Sprints.

Para el desarrollo del presente trabajo de investigación el equipo de desarrollo de este proyecto fue desarrollado por una sola persona la misma tuvo los roles de Product Owner, Scrum Master y Scrum Team. La primera reunión de planificación permitió que el equipo Scrum construya los sprints necesarios, así como también realizar las estimaciones iniciales y verificar la importancia de cada requerimiento establecido por el cliente. La estimación inicial se realizó mediante el planning poker que es una técnica ágil basada en el consenso.

Para la realización del presente trabajo de investigación, el equipo de desarrollo tuvo los siguientes roles:

- Sra. Bertha Campoverde – Cliente / Stakeholder,
- Ing. Mauricio Rea – Facilitador / Scrum Master y
- Gina Ulloa – Dueño del producto / Product Owner y Equipo de Scrum / Scrum Team.

La primera reunión de planificación permitió que el equipo Scrum construya los sprints necesarios, así como también realizar las estimaciones iniciales y verificar la importancia de cada requerimiento establecido por el cliente. La estimación inicial se realizó mediante el planning poker que es una técnica ágil basada en el consenso.

La pila del producto se muestra en la siguiente tabla:

PILA DEL PRODUCTO			
ID	Descripción de requerimiento	Importancia	Estimación Inicial
1	Administrar productos	80	40
2	Visualizar detalles de productos	50	13
3	Administrar los pedidos de los usuarios	50	20
4	Administrar el stock de los productos	20	13
5	Registrarse en el sistema	50	13
6	Comprar productos	80	40
7	Administrar la información de la cuenta	20	13
8	Conocer el listado de los pedidos realizados	20	13
9	Conocer el listado de los productos en el carrito de compras	50	20
10	Pagar las compras	80	40

Tabla N° 4 Planificación del Sprint
Elaborado por: Gina Ulloa

Primera reunión de planificación de Sprint (SPRINT 1):

- El dueño del producto indica la meta del sprint y realiza un resumen de la Pila del Producto.
- El equipo Scrum da las estimaciones de tiempo.
- El dueño del producto actualiza la importancia de los requerimientos.
- Se depuran los elementos de la pila del producto.
- El equipo selecciona las historias que se incluirán en el Sprint.
- Se realizan cálculos para verificar si es factible realizarlas.
- Se continúa dividiendo las historias en tareas.

El sprint planning o reunión de planificación es crítica, seguramente es el suceso más importante en Scrum, ya que si está mal realizada puede llevar a vulnerar un sprint completo, el propósito de esta reunión es brindar al equipo de scrum la información necesaria para que sea capaz de trabajar durante el tiempo que dure el sprint y facilitar al

dueño del producto los entregables en la fecha fijada para su revisión y retrospectiva de ser el caso.

La tabla muestra la pila de Sprints definidos en la primera reunión de planificación de sprints:

Item	Sprint	Responsable	Historia de Usuario	Tareas	Días asignados
1	Sprint 1	Desarrollador		Realizar el análisis y diseño del proyecto	15
				Realizar el modelamiento de la base de datos	
2	Sprint 2	Desarrollador	HU06	Crear la pantalla inicial del proyecto	15
				Presentar los productos que tienen descuentos en la página principal, ordenados ascendentemente	
3	Sprint 3	Desarrollador	HU01	Crear la pantalla de administración de productos con los procesos y servicios necesarios	15
4	Sprint 4	Desarrollador	HU02	Crear la pantalla de visualización del detalle de productos	15
5	Sprint 5	Desarrollador	HU03	Crear la pantalla de administración de pedidos de los usuarios con los procesos y servicios necesarios	15
6	Sprint 6	Desarrollador	HU04	Crear la pantalla de Reporte del stock de los productos del catálogo con los procesos y servicios necesarios	15
7	Sprint 7	Desarrollador	HU06	Crear la pantalla de visualización de productos para el usuario	15
				Permitir al usuario añadir los productos al carrito de compras	
8	Sprint 8	Desarrollador	HU05 HU07	Crear la pantalla de logueo, registro de usuario y administración de su información con los procesos y servicios necesarios	15
9	Sprint 9	Desarrollador	HU08	Crear la pantalla de visualización de los pedidos realizados por el usuario	15
10	Sprint 10	Desarrollador	HU09	Crear la pantalla para la administración del carrito de compras del usuario con los procesos y servicios necesarios	15
11	Sprint 11	Desarrollador	HU10	Crear la opción de pagos de las compras realizadas por el usuario con los procesos y servicios necesarios	15

Tabla N° 5 Sprint backlog definidos en la primera reunión de planificación
Elaborado por: Gina Ulloa

2.3.5.3 Fase 3: Scrum diario

a. Comunicación de sprint backlogs

En esta fase se deberá definir un sitio y una hora para la entrega del scrum diario, este scrum diario define por donde van a empezar a trabajar los desarrolladores, aquí participan el Scrum Master y el equipo Scrum para verificar y evaluar los avances de los sprints y así de esta manera evitar retrasos en la culminación del proyecto se deberá cumplir con los siguientes procesos:

- Crear documentos entregables
- Ejecutar el scrum diario
- Mantener la lista de pendientes del proyecto

b. Presentación de avances con el cuadro Burndown

El gráfico que se muestra a continuación indica el comportamiento del desarrollo del sprint backlog 1, el cual tiende a decrecer porque las reuniones permiten avanzar rápidamente en el desarrollo y la minimización de retrasos, este tuvo una duración de 15 días hasta el siguiente sprint que igual tiene una duración de 15 días.

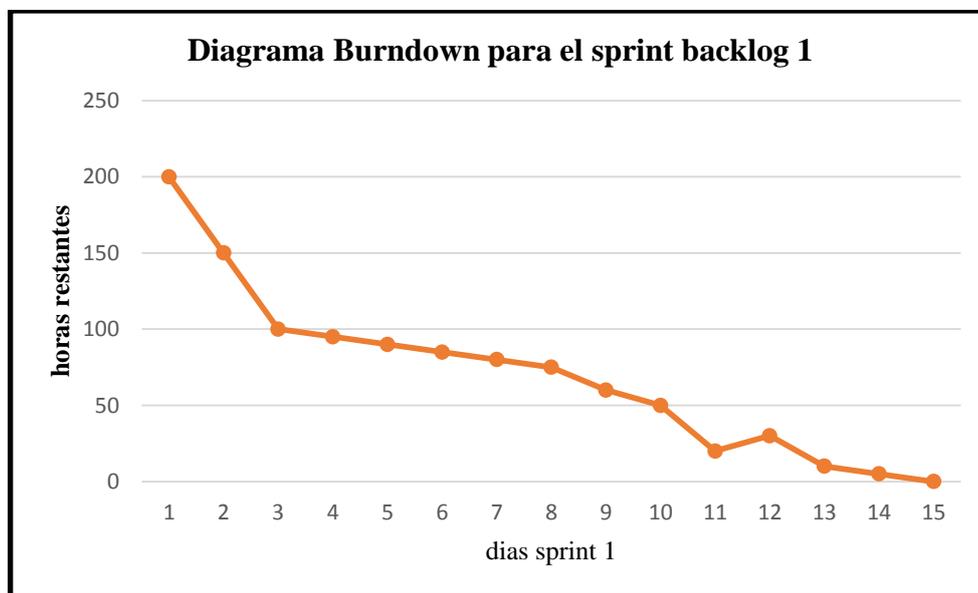


Gráfico N° 2 Diagrama Burndown para el sprint backlog 1
Elaborado por: Gina Ulloa

En el siguiente grafico se puede observar el comportamiento del sprint 11, la gráfica lineal tiende al decrecimiento, el cual tuvo una duración de 15 días.

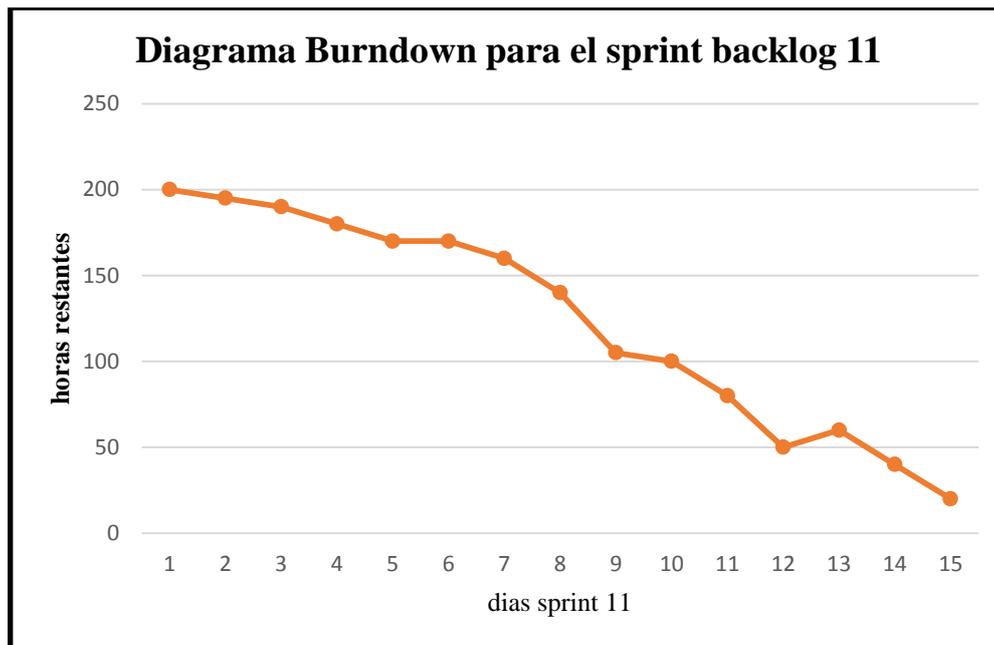


Gráfico N° 3 Diagrama Burndown para el sprint backlog 11
Elaborado por: Gina Ulloa

2.3.5.4 Fase 4: Revisión del sprint

En esta etapa se involucra al equipo de desarrollo scrum, scrum master, dueño del proyecto para presentar un demo del sprint, el objetivo es presentar las nuevas funcionalidades implementadas en el proyecto, es necesario realizar una lista de comprobación del demo en la cual debe contener de manera clara el objetivo del sprint, la demo debe ser sencilla donde se muestre el código funcionando.

a. Planificación de entregas

Los documentos entregables de cada uno de los sprints están basados en el historial de usuario inicial, para el sprint 1 se tenía planificado realizar el análisis y diseño del proyecto y el modelamiento de la base de datos, lo cual se culminó con satisfacción y en los tiempos acordados.

Para el sprint 2 se tiene como objetivos cumplir:

- Crear la pantalla inicial del proyecto
- Presentar los productos que tienen descuentos en la página principal ordenados ascendentemente.

Los cuales se culminaron satisfactoriamente y en los tiempos acordados.



Figura N° 31 Pantalla principal del proyecto
Elaborado por: Gina Ulloa

Para el sprint 11 se tiene como objetivo cumplir:

- Crear la opción de pagos de las compras realizadas por el usuario con los procesos y servicios necesarios.

Aquí ya están todos los sprints completos los cuales se culminaron satisfactoriamente y en los tiempos acordados.

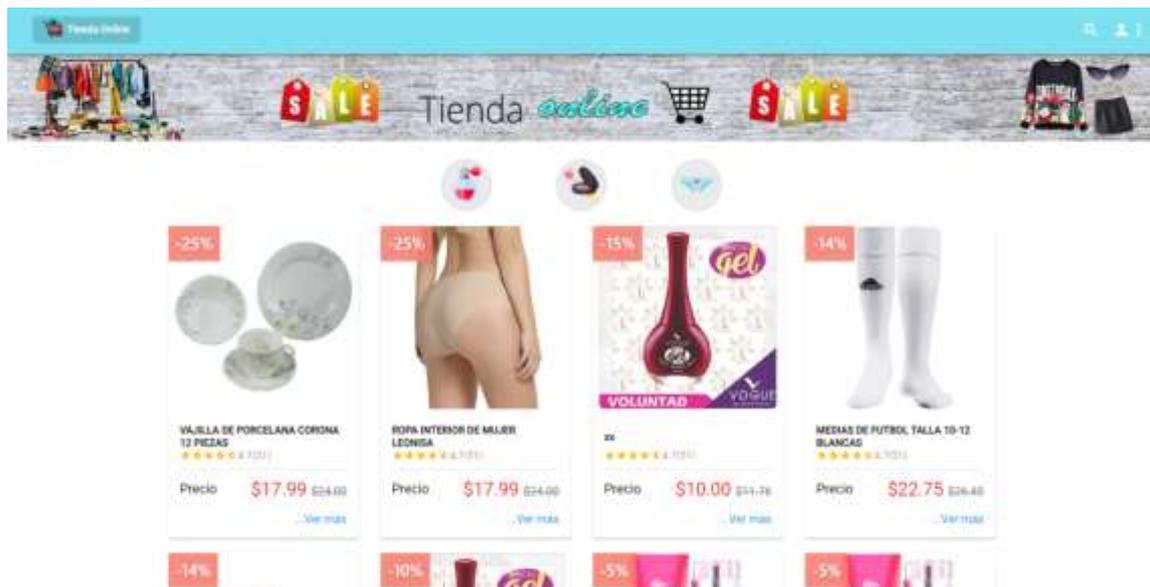


Figura N° 32 Pantalla final del proyecto finalizado
Elaborado por: Gina Ulloa

2.3.5.5 Fase 5: Retrospectiva del sprint

En esta última etapa de la metodología se deberá realizar la retrospectiva de cada uno de los sprints planteados y definidos en la fase de planificación cuando el dueño del producto o cliente determine que el entregable proporcionable no cumple con los requerimientos iniciales propuestos, para el desarrollo del presente trabajo de investigación se cumplió a cabalidad todos y cada uno de los sprints propuestos y desarrollados.

Finalmente, se presentará un resumen del estado final del historial de usuario donde refleja claramente que se ha realizado la aplicación correcta de la metodología Scrum, cada una de las fases fue de gran relevancia en el desarrollo del proyecto:

HISTORIAL DE USUARIO					
ID	Nombre	Importancia	Estim. inicial	Comprobación	Estado final
1	Ingreso al proyecto	Alto	5	Ejecutar el proyecto	Culminado
2	Ingreso a la pantalla de visualización del detalle de productos	Alto	3	Dar clic en el botón Productos	Culminado
3	Ingreso a la pantalla de administración de pedidos de los usuarios con los procesos y servicios necesarios	Alto	3	Dar clic en el botón Verificar	Culminado
4	Ingreso a la pantalla de reporte de stock de productos del catálogo con los procesos y servicios necesario	Alto	5	Dar clic en el botón Stock	Culminado
5	Ingreso a la pantalla de visualización de productos para el usuario	Alto	3	Dar clic en el botón Tienda Online	Culminado
6	Permitir al usuario añadir los productos al carrito de compras	Alto	8	Dar clic en el botón agregar	Culminado
7	Ingreso a la pantalla de visualización de los pedidos realizados por el usuario	Alto	3	Dar clic en el botón Mis Compras	Culminado
8	Ingreso a la administración del carrito de compras del usuario con los procesos y servicios necesarios	Alto	3	Dar clic en el botón Carrito de Compras	Culminado
9	Ingreso a los pagos de las compras realizadas por el usuario con los procesos y servicios necesarios	Alto	5	Dar clic en el botón Pago con PayPal	Culminado

Tabla N° 6 Estado final de las tareas del historial de usuario del proyecto

Fuente: Gina Ulloa

2.4 ESTABILIZACIÓN DEL PROYECTO

La estabilización del proyecto comienza con las pruebas beta del proyecto ya construido y termina esta fase cuando el sistema haya pasado todas las pruebas requeridas por las partes interesadas, el objetivo de realizar estas pruebas es medir el nivel de persistencia que tiene el software frente a las necesidades reales del cliente y de esta manera brindar la confianza y seguridad del proyecto.

En esta fase de estabilización se pretende integrar cada una de las funcionalidades del proyecto al aplicar algunos tipos de pruebas para analizar de manera minuciosa las falencias y corregir los errores que surgen durante su ejecución con la finalidad de brindar al cliente un producto operacionalmente seguro y confiable.

2.4.1 Pruebas de software para estabilización del proyecto

La elaboración de un proyecto de software implica muchos conocimientos, talento, experiencia y dedicación; para que el proyecto cumpla con el objetivo planteado es necesario que se apliquen pruebas funcionales de software para evitar fallas y errores antes de que el proyecto salga a producción, a continuación se describirá las pruebas de software más relevantes aplicadas en la presente investigación:

- **Pruebas unitarias.-** Estas pruebas permiten verificar la funcionalidad del código del sistema de acuerdo a las especificaciones planteadas y también permite verificar la lógica de los módulos del proyecto, las pruebas unitarias permiten al proyecto realizar lo siguiente: particionar los módulos y aplicar pruebas unitarias a cada módulo, definir los casos de prueba, considerar las rutinas de excepción, el manejo de parámetros y validaciones.
- **Pruebas de integración.-** Estas pruebas de integración o pruebas de componentes permiten identificar falencias cuando los módulos trabajan en conjunto, este tipo de pruebas permiten identificar errores producido por la combinación de programas, verificar las interfaces externas, la aplicación de esta prueba permite probar la integración total del sistema.
- **Pruebas de carga.-** Permiten verificar el tiempo de respuesta del sistema al trabajar bajo grandes cantidades de información, estas pruebas miden el comportamiento del

sistema bajo diferentes condiciones de carga de información en diferentes intervalos de tiempo.

- **Pruebas de estrés.-** Estas pruebas permiten evaluar los límites del sistema sometándolo a condiciones extremas como grandes cantidades de datos o gran afluencia de usuarios simultáneos, permite verificar las funcionalidades del sistema bajo las siguientes condiciones: memoria baja, máximo número de usuarios, múltiples usuarios interactuando en el sistema realizando la misma transacción al mismo tiempo.

2.5 IMPLEMENTACIÓN DEL PROYECTO

Implementar mejoras tecnológicas y automatización de procesos tradicionales en una empresa no es una tarea sencilla ya que pueden existir personas que se resistan al cambio por el nivel de dificultad o facilidad que implique la gestión de un proyecto de software, en la implementación del proyecto pueden existir modificaciones o nuevos requerimientos que pueden surgir a última hora en el proyecto ya desarrollado, una vez implementado el software en la empresa se procederá a la respectiva migración de datos, ingreso de información, capacitar a los usuarios y entregar la respectiva documentación de actas de entregas, manuales de usuario, manuales técnicos que brindarán soporte y seguridad al usuario.

Una vez entregado todos los requisitos, se deberá plantear un plan de pruebas constantes en el software con la finalidad de determinar si el software continúa cumpliendo con las necesidades y objetivos por los que fue creado, también se deberá sugerir un plan de soporte con la finalidad de brindar a la empresa seguridad y confianza a la hora de sacar provecho a las soluciones tecnológicas implementadas y así de esta manera lograr el éxito pretendido con el software.

2.7 MANUAL DE USUARIO

APLICACIÓN DE COMERCIO ELECTRÓNICO PARA EL BAZAR NOVEDADES GUAYAS

- **Pantalla Principal:** Presenta al usuario la pantalla principal donde se encuentran los productos con descuentos ordenados ascendentemente.

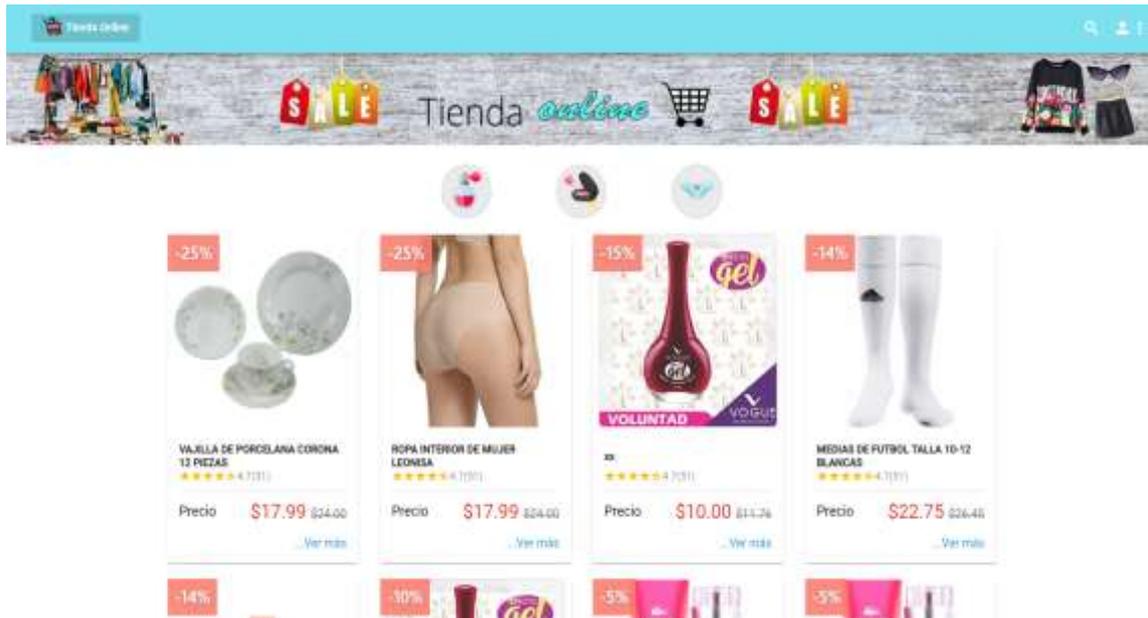


Figura N° 33 Página principal de la tienda Online
Elaborado por: Gina Ulloa

- Si da clic en la Opción  esta permitirá volver a visualizar el menú principal.
- Si da clic en la Opción  esta le permitirá filtrar los productos por la categoría perfumes.
- Si escoge la Opción  esta permite filtrar los productos por la categoría cosméticos.
- Si escoge la Opción  esta permite filtrar los productos por la categoría lencería.

- La Opción  permite ejecutar la búsqueda de los productos que se desea buscar para realizar la compra.
- La Opción  permite loguearse en el sistema.
- Si escoge la Opción [...Ver más](#) le permitirá ver la descripción del artículo seleccionado



Figura N° 34 Descripción de productos de la tienda Online
Elaborado por: Gina Ulloa

- En la Opción  le permitirá agregar el artículo al carrito de compra del cliente logueado
- La Opción  le permitirá al usuario volver a la pantalla previa sin agregar al carrito de compras
- En la Opción  le permitirá al usuario navegar por las páginas del catálogo, si existe más de una página las opciones siguiente y atrás se habilitarán
- **Pantalla de logueo:** Presenta la información necesaria para ingresar al sistema, se debe ingresar un email válido y una contraseña máximo de 10 caracteres



Figura N° 35 Pantalla de logueo para ingresar a la tienda Online
Elaborado por: Gina Ulloa

- En la Opción **Crear nueva cuenta** le permitirá al usuario registrarse en el sistema si no tiene una cuenta ya registrada
- **Pantalla de Registro:** Presenta la información necesaria para registrarse en el sistema, todos los campos registrados deben ser llenados obligatoriamente.



Figura N° 36 Pantalla de registro de un nuevo cliente
Elaborado por: Gina Ulloa

- En la Opción **Ingresar** le permitirá al usuario regresar a la pantalla de logueo.

ADMINISTRADOR

- En esta pantalla principal de Administrador: Presentará las opciones para el Administrador del sistema.

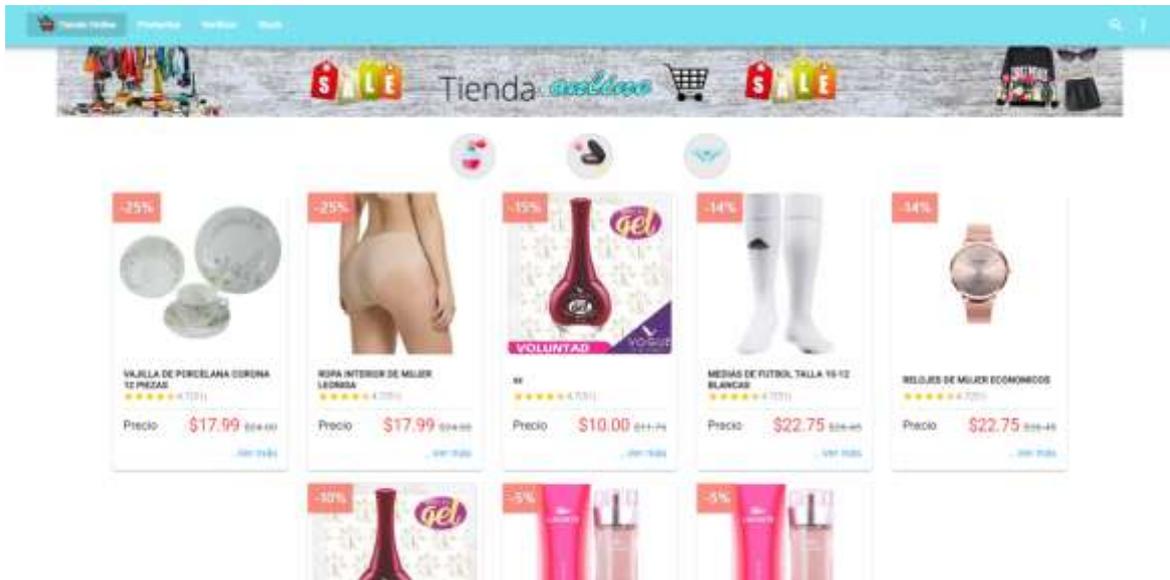
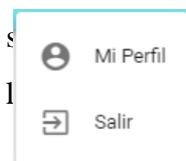


Figura N° 37 Página principal con las opciones del rol Administrador
Elaborado por: Gina Ulloa

- En la Opción **Productos** le permitirá administrar los productos del catálogo.
- Si escoge la Opción **Verificar** le permitirá verificar los pedidos enviados por los clientes desde su carrito de compras.
- En la Opción **Stock** le permitirá al administrador visualizar un reporte con el stock de los productos del catálogo.
- La Opción **<< Monstrando 1 - 1 de 8 items. >>** le permitirá navegar por las páginas del catálogo, si existe más de una página las opciones siguiente y atrás se habilitarán.
- Si el administrador escoge la Opción **⋮** le permitirá realizar dos acciones en el sistema, administración la información del usuario logueado y salir del sistema.
- **Pantalla de Administración de Información:** Aquí el usuario podrá visualizar la información para su edición de ser el caso.



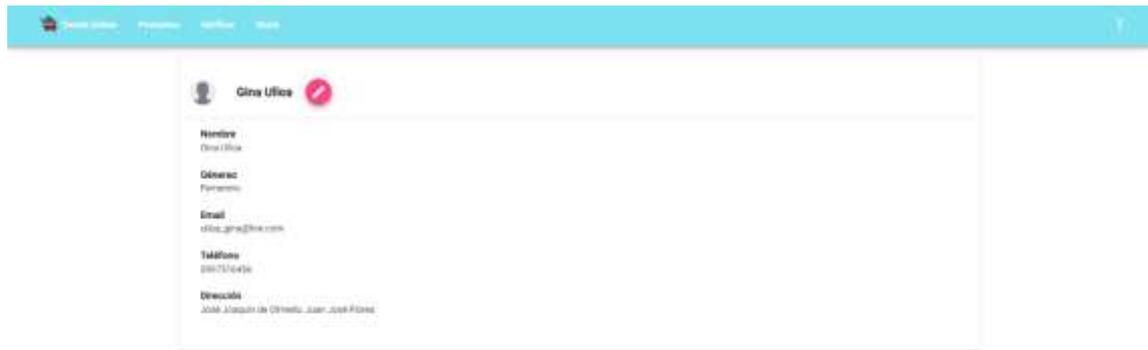


Figura N° 38 Página de administración de información
Elaborado por: Gina Ulloa

- Opción  le permitirá al usuario editar su información.
- **Pantalla de Edición de Información:** Aquí se presentará la información del usuario para su edición



Figura N° 39 Página de edición de información
Elaborado por: Gina Ulloa

- En la Opción  le permitirá cancelar la edición de la información del usuario.
- En la Opción  le permitirá actualizar la información del usuario.
- **Pantalla de Gestión de productos:** En esta pantalla se podrá visualizar una lista de los productos registrados en el sistema y las opciones para su administración.

ID	Nombre	Descripcion	Categoria	Precio	Stock	Acciones
304	Producto 1	Producto 1	COSMETICO	14	0	
303	Producto 2	Producto 2	LENCERA	16	9	
302	Producto 1	Producto 1	LENCERA	18	9	
301	yc	asasa	COSMETICO	12	0	
229	yo-estado	si - LANCHEOS I - GANCHOS I	COSMETICO	180	10	
228	XI	si	COSMETICO	16	15	
81	BONA INTERIOR DE MUJER LEONISA	BONA INTERIOR DE MUJER LEONISA	LENCERA	1739	25	
67	FORTUNA DE MUJER SIBAL MUK	FORTUNA DE MUJER SIBAL MUK	FORTUNA	96		
36	RELJES DE MUJER ECONOMICO	RELJES DE MUJER ECONOMICO	ACCESORIOS	2270	14	
33	MEDIAS DE FUTBOL TALLA 10-12 BLANCO	MEDIAS DE FUTBOL TALLA 10-12 BLANCO	ROPA	2270	14	

Figura N° 40 Página de gestión de productos
Elaborado por: Gina Ulloa

- Si da clic en la Opción le permitirá agregar nuevos productos al catálogo.
- Si escoge la Opción le permitirá buscar productos dentro de la pantalla de Gestión de productos.
- En la Opción se podrá editar la información del producto.
- En la Opción se podrá eliminar el producto.
- En la Opción le permitirá navegar por las páginas, si existe más de una página se habilitarán las opciones siguiente, anterior, último y primero, se podrá escoger el número de productos que se mostrarán por página.
- **Pantalla de Ingreso de Productos:** Aquí podrá visualizar la información necesaria para el ingreso de nuevos productos al catálogo, se presentan con * los campos obligatorios.

Gestión de Productos

Nombre del Producto *

Descripción del Producto *

URL Imagen del Producto *

Categoría * Proveedor *

Precio de Compra * Precio de Venta *

Cantidad de Productos Descuento del Producto

Save Cancel

Figura N° 41 Página de ingreso de productos
Elaborado por: Gina Ulloa

- Si escoge la Opción Save le permitirá guardar la información ingresada del producto.
- Con la Opción Cancel le permitirá regresar a la pantalla anterior sin guardar.
- **Pantalla de Edición de Productos:** Aquí el administrador podrá visualizar la información del producto seleccionado para su edición.

Gestión de Productos

304

Nombre del Producto *

Producto 3

Descripción del Producto *

Producto 3

URL Imagen del Producto *

<https://i.liniu.com/p/77d4a37f55595168119aab8be1bbe0a9-product.jpg>

Categoría Proveedor

COSMETICOS Papeleria Popular

Precio de Compra * Precio de venta *

10,5 14

Cantidad de Productos Descuento del Producto

-1 0

Save Cancel

Figura N° 42 Página de edición de productos
Elaborado por: Gina Ulloa

- Si escoge la Opción Save le permitirá guardar la información del producto.

- La Opción  **Cancel** le permitirá regresar a la pantalla anterior sin guardar cambios.
- **Pantalla de Verificación de Pedidos:** Aquí se presentará la información de los pedidos enviados por los clientes para su respectiva aprobación.



Fecha Compra	Nr. Compra	Total	Estado	Detalle
16-10-2019	10010001	\$17.99	EN PÉNDICE	
16-10-2019	10010011	\$17.99	EN PÉNDICE	
24-01-2019	1001000	\$84.74	EN PÉNDICE	
24-01-2019	1001000	\$100.00	EN PÉNDICE	

Figura N° 43 Página de edición de productos
Elaborado por: Gina Ulloa

- Si escoge la Opción  le permitirá visualizar el detalle del pedido.
- Con la Opción **Items per page: 7**  **1 - 4 of 4**  le permitirá navegar por las páginas, si existe más de una página se habilitarán las opciones siguiente, anterior, último y primero, se podrá escoger el número de productos que se mostrarán por página.
- **Pantalla de Detalle del Pedido:** Aquí podrá visualizar la información del detalle del pedido seleccionado



Imagen	Producto	Cantidad	Precio	Total
	VAJILLA DE PORCELANA CORONA 12 PIEZAS	1	\$17.99	\$17.99
		1 Unidades	Total	\$17.99

Figura N° 44 Página de edición de productos
Elaborado por: Gina Ulloa

- La Opción  le permitirá cancelar el pedido seleccionado.

- Si escoge la Opción **ENVIAR PEDIDO** le permitirá aprobar el pedido, el mismo que le aparecerá al cliente en la opción Mis Compras en estado Por Pagar.
- Si escoge la Opción **Regresar** le permitirá regresar a la pantalla anterior.
- **Pantalla de Stock de Productos:** Presenta un reporte de los productos del catálogo centrándose en el stock actual.

Para ello se presentará de color ● cuando el stock es igual o menor a 5 unidades, de color ● cuando el stock es menor o igual 25 y de color ● cuando el stock es mayor a 25, así de esta manera el administrador podrá realizar un control de los productos.

Imagen	Nombre	Descripción	Categoría	Precio	Stock
	Producto 1	Producto 1	COSMETICOS	14	● 1
	PERFUMES DE ALEJOT YAMAL MUJER	PERFUMES DE ALEJOT YAMAL MUJER	PERFUMES	36	● 2
	Producto 1	Producto 1	LEUCERA	18	● 18
	Producto 2	Producto 2	LEUCERA	18	● 18
	W1	WAAAA	COSMETICOS	12	● 18
	VAJILLA DE PORCELANA COPINA 12 PZAS	VAJILLA DE PORCELANA COPINA 12 PZAS	BAJAS	17.99	● 77
	ROPA INTERIOR DE MUJER LEUCERA	ROPA INTERIOR DE MUJER LEUCERA	LEUCERA	17.99	● 77
	W1 2 wWwW	W1 -caracteristica 1 -caracteristica 2	COSMETICOS	100	● 407
	W1	W1	COSMETICOS	10	● 498
	MEZMA DE FUTRES, TALLA 16-12 BLANCO	MEZMA DE FUTRES, TALLA 16-12 BLANCO	ROPA	2275	● 495

Figura N° 45 Página de stock de productos
Elaborado por: Gina Ulloa

- Si escoge la Opción **Q Buscar Producto** le permitirá buscar un producto dentro del reporte del stock de productos.
- Si escoge la Opción **Items per page: 7** **1 - 4 of 4** **|< < > >|** le permitirá navegar por las páginas, si existe más de una página se habilitarán las opciones siguiente, anterior, último y primero, se podrá escoger el número de productos que se mostrarán por página.

USUARIO COMPRADOR

- **Pantalla Principal Usuario Comprador:** Aquí podrá visualizar las opciones para el usuario del sistema.

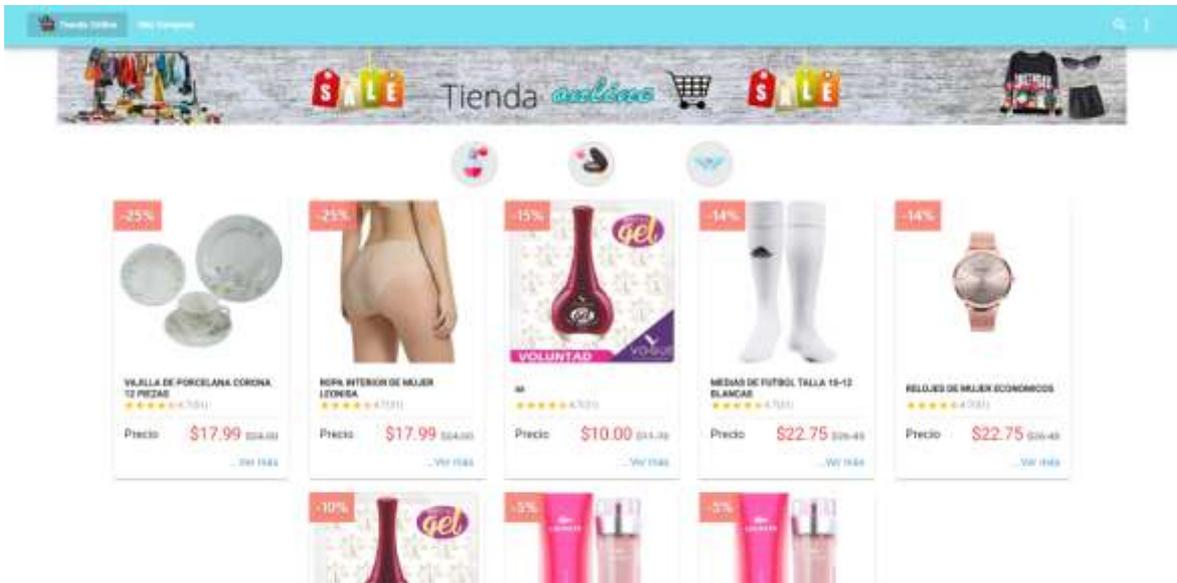


Figura N° 46 Página principal para el usuario comprador
Elaborado por: Gina Ulloa

- Si escoge la Opción **Mis Compras** le permitirá administrar las compras registradas por el usuario.
- Con la Opción **Buscar** le permitirá buscar dentro del catálogo de productos.
- En la Opción **Carrito** le permitirá visualizar el carrito de compras del usuario.
- Si da clic en la Opción **Menú** el usuario podrá realizar dos acciones en el sistema, administración de la información del usuario logueado y salir del sistema.

- **Pantalla de Compras:** Aquí podrá visualizar la información de las compras aprobadas por el administrador, el estado de las compras puede ser Pagado o Por Pagar.

Mis Compras

Fecha Compra	Nº Compra	Total	Detalle	Estado
25-01-2019	10010013	\$14.00	POR PAGAR	100
25-01-2019	10010014	\$14.00	PAGADO	100
25-01-2019	10010015	\$17.99	PAGADO	100
23-01-2019	10010012	\$100.00	POR PAGAR	100
23-01-2019	10010011	\$22.75	POR PAGAR	100
19-01-2019	10010010	\$30.74	POR PAGAR	100
17-01-2019	10010009	\$17.99	POR PAGAR	100

Figura N° 47 Página de stock de productos
Elaborado por: Gina Ulloa

- En la Opción **Items per page: 7** **1 - 4 of 4** **<< < > >>** le permitirá navegar por las páginas, si existe más de una página se habilitarán las opciones siguiente, anterior, último y primero, se podrá escoger el número de productos que se mostrarán por página.
- **Pantalla de Carrito de Compras:** Presenta la información del Carrito de Compras del Cliente.

Productos Agregados al Carrito

Imagen	Producto	Cantidad	Precio	Total
	MEGADE FUTERA TALLA 30 (2) BLANCAS	1	\$22.75	\$22.75
		1 Unidades	Total	\$22.75

CANCELAR **ENVIAR PEDIDO**

Figura N° 48 Página de carrito de compras
Elaborado por: Gina Ulloa

- Si el usuario escoge la Opción **CANCELAR** se le permitirá cancelar el pedido del carrito de compras.
- Si da clic sobre la Opción **ENVIAR PEDIDO** le permitirá enviar el pedido para la aprobación del administrador del sistema.

- **Pantalla de Administración de Información:** Aquí se presentará la información del usuario logueado para su edición.

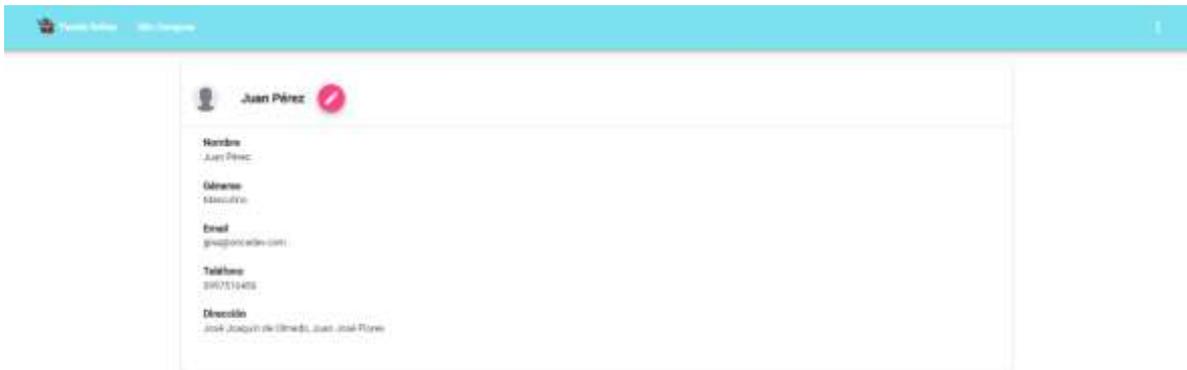


Figura N° 49 Página de administración de información
Elaborado por: Gina Ulloa

- Si escoge la Opción  le permitirá editar la información del usuario.
- **Pantalla de Edición de Información:** Podrá visualizar la información del usuario para su edición.



Figura N° 50 Página de edición de información
Elaborado por: Gina Ulloa

- Si escoge la Opción  le permitirá cancelar la edición de la información del usuario.
- Si escoge la Opción  le permitirá actualizar la información del usuario.

CAPÍTULO III

RESULTADOS

3.1 VALIDACIÓN

Para el presente trabajo de investigación ha sido necesario demostrar la teoría con la práctica mediante la aplicación de una encuesta, la obtención de resultados permitió realizar el análisis para la respectiva validación de resultados, lo que se pretendió conseguir fue validar la investigación desarrollada, cuyos objetivos fueron los siguientes:

- Determinar el nivel de conocimiento de la herramienta Maven y sus aplicaciones entre los profesionales del sector del software.
- Validación de los resultados mediante la tabulación y análisis de datos.
- Obtener criterios de los profesionales del software sobre las herramientas de gestión de desarrollo de software.

3.1.1 Tabulación y análisis de resultados de la encuesta aplicada

ENCUESTA DE SATISFACCIÓN DE USUARIOS

1. ¿Usted ha utilizado alguna herramienta para gestionar sus proyectos de desarrollo de software?

Pregunta	Cantidad	Porcentaje
Si	28	93.33%
No	2	6.67%
Total	30	100%

Tabla N° 7 Herramienta para gestión de proyectos de software
Elaborado por: Gina Ulloa



Gráfico N° 4 Herramienta para gestión de proyectos de software
Elaborado por: Gina Ulloa

ANÁLISIS

Se puede evidenciar en la figura que casi la totalidad de los encuestados supieron manifestar que si han utilizado alguna herramienta de gestión de software, lo que indica que en el ámbito laboral la mayoría de desarrolladores usan gestores para sus proyectos ya que permite minimizar tiempos de configuración, implementación y despliegue.

2. ¿Cree usted que es necesario utilizar una herramienta para gestionar sus proyectos de desarrollo de software?

Pregunta	Cantidad	Porcentaje
Si	29	96.67%
No	1	3.33%
Total	30	100%

Tabla N° 8 Uso de herramienta para gestión de proyectos de software
Elaborado por: Gina Ulloa



Gráfico N° 5 Uso de herramienta para gestión de proyectos de software
Elaborado por: Gina Ulloa

ANÁLISIS

La figura muestra que la mayoría de los encuestados manifestaron que si es necesario utilizar herramientas para la gestión de desarrollo de software, esto permite evidenciar que los desarrolladores de software conocen los beneficios de la utilización de gestores de proyectos, en realidad esto se ha convertido en una norma para el desarrollo de proyectos ya que permite una agilización y permite centrarse en lo realmente interesante de los proyectos.

3. ¿Usted ha utilizado Maven?

Pregunta	Cantidad	Porcentaje
Si	24	80%
No	6	20%
Total	30	100%

Tabla N° 9 Uso de herramienta Maven
Elaborado por: Gina Ulloa

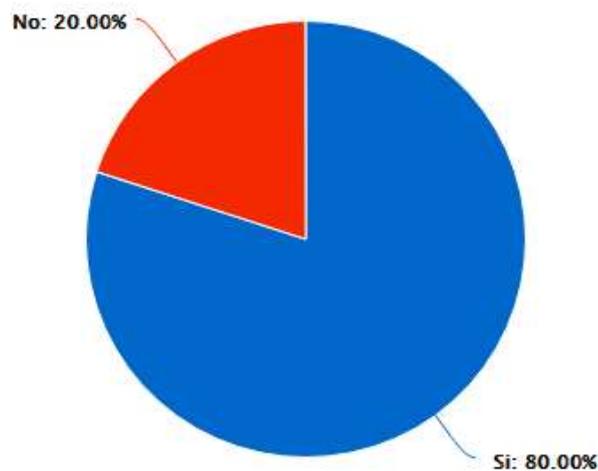


Gráfico N° 6 Uso de herramienta Maven
Elaborado por: Gina Ulloa

ANÁLISIS

Se puede evidenciar que casi la totalidad de los encuestados manifestaron que si han utilizado la herramienta de desarrollo Maven en la elaboración de sus proyectos, aunque un 20% indica que no, una de las razones puede ser la utilización de otro lenguaje de desarrollo en los proyectos ya que Maven es un gestor de proyectos Java.

4. ¿Cree usted que cualquier programador sin ningún tipo de experiencia puede utilizar Maven?

Pregunta	Cantidad	Porcentaje
Si	14	46.67%
No	16	53.33%
Total	30	100%

Tabla N° 10 Uso de herramienta Maven sin experiencia
Elaborado por: Gina Ulloa

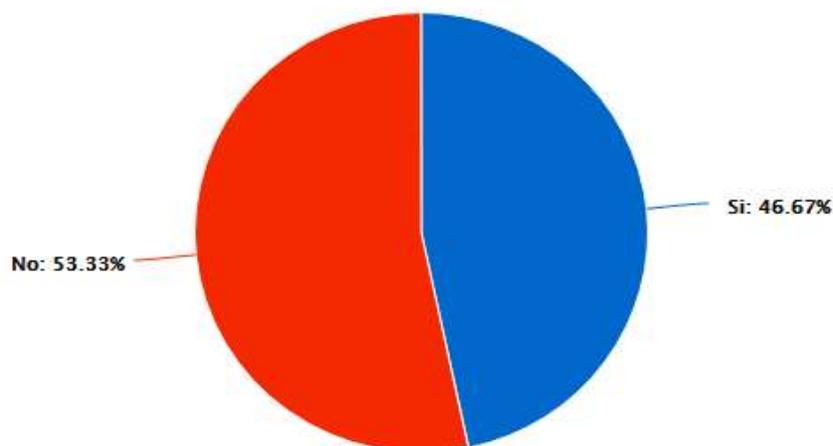


Gráfico N° 7 Uso de herramienta Maven sin experiencia
Elaborado por: Gina Ulloa

ANÁLISIS

Se puede evidenciar que no existe mucha diferencia en las respuestas de los encuestados, más de la mitad respondieron que no es necesario ningún tipo de experiencia para la utilización de la herramienta de desarrollo Maven, es decir, que para empezar a desarrollar en esta herramienta si es necesario tener conocimientos básicos de programación para realizar configuraciones y adaptaciones según los requerimientos y el tamaño del proyecto.

5. ¿Cree usted que se ha reducido el tiempo en el desarrollo de sus proyectos de software con Maven?

Pregunta	Cantidad	Porcentaje
Si	25	89.29%
No	3	10.71%
Ninguno	2	0.0%
Total	30	100%

Tabla N° 11 Reducción de tiempos con el uso de la herramienta Maven
Elaborado por: Gina Ulloa

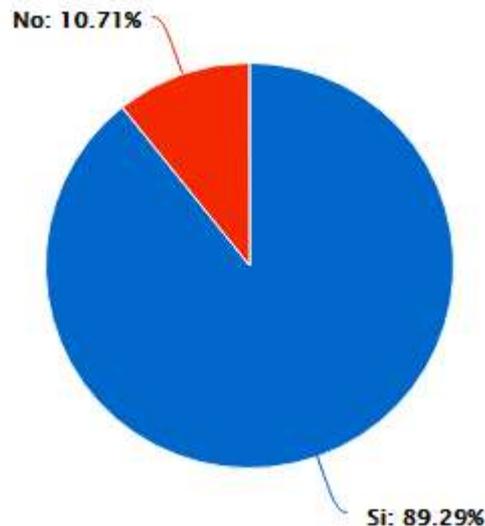


Gráfico N° 8 Reducción de tiempos con el uso de la herramienta Maven
Elaborado por: Gina Ulloa

ANÁLISIS

La figura muestra que la mayoría de los encuestados respondieron que si existe una reducción de tiempos en el desarrollo de aplicaciones informáticas con el uso de la herramienta Maven, ya que uno de los principales beneficios es el uso de arquetipos que le permiten al programador tener una estructura de carpetas definidas lo que a la larga y con grupos numerosos de desarrolladores resultaría ser una gran ventaja, es decir, que el programador al momento de seleccionar alguna herramienta de gestión de proyectos Java deberían optar por Maven cuando quieran trabajar en un entorno estandarizado.

6. ¿El gestor de proyectos Maven se adapta a las necesidades su empresa?

Pregunta	Cantidad	Porcentaje
Nada	3	3.58%
Parcialmente	17	35.71%
Totalmente	10	60.71%
Total	30	100%

Tabla N° 12 Gestor de proyectos Maven
Elaborado por: Gina Ulloa

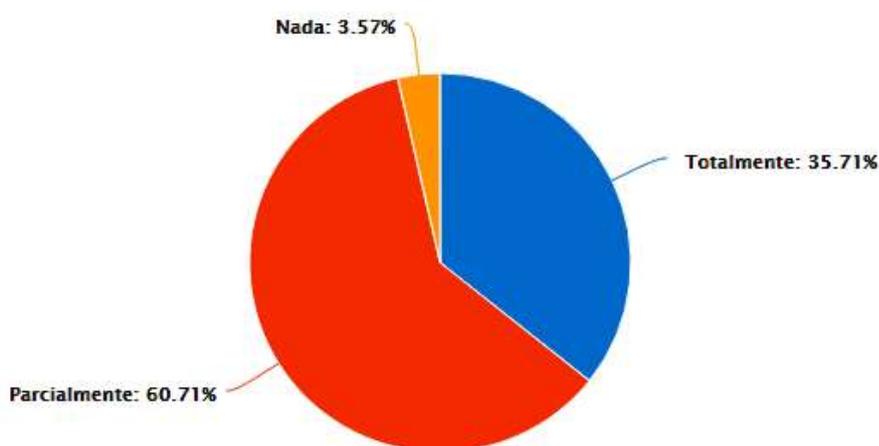


Gráfico N° 9 Gestor de proyectos Maven
Elaborado por: Gina Ulloa

ANÁLISIS

Se puede evidenciar que un gran porcentaje de los encuestados respondieron que el gestor de proyectos Maven se adapta parcialmente a las necesidades de la empresa, es decir, que Maven sería una opción acertada para el programador al momento de seleccionar una herramienta de gestión de proyectos Java ya que la mayoría de IDEs se pueden integrar con Maven lo que facilita la diversificación de los entornos de programación, pero en lo referente a lenguajes tiene una limitante, en este caso al ser una empresa que también desarrolla proyectos en C#, Javascript entre otros se hace evidente que Maven no podría satisfacer totalmente sus necesidades.

7. ¿Cree usted que Maven es mejor que otros gestores de proyectos de desarrollo de software?

Pregunta	Cantidad	Porcentaje
Si	18	64.29%
No	10	35.71%
Tal vez	2	0.0%
Total	30	100%

Tabla N° 13 Maven como mejor gestor de proyectos
Elaborado por: Gina Ulloa

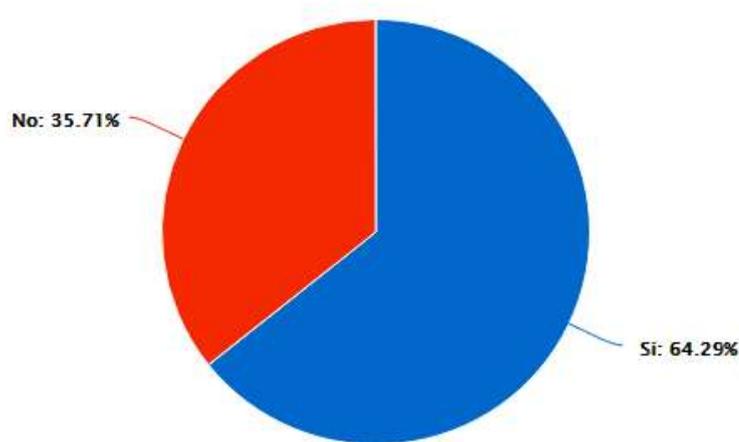


Gráfico N° 10 Maven como mejor gestor de proyectos
Elaborado por: Gina Ulloa

ANÁLISIS

De acuerdo a la figura la respuesta se encuentra dividida entre los encuestados ya que manifestaron que Maven es el mejor gestor de proyectos de software pero una desventaja importante es que es para proyectos Java, existen otros gestores para otros lenguajes de programación los cuales pueden resultar ser los mejores para los lenguajes específicos para los que fueron creados.

8. ¿Le gustaría incluir a Maven en sus proyectos de desarrollo de software?

Pregunta	Cantidad	Porcentaje
A veces	15	50.00%
Nunca	1	3.33%
Siempre	14	46.67%
Total	30	100%

Tabla N° 14 Maven como gestor de proyectos
Elaborado por: Gina Ulloa

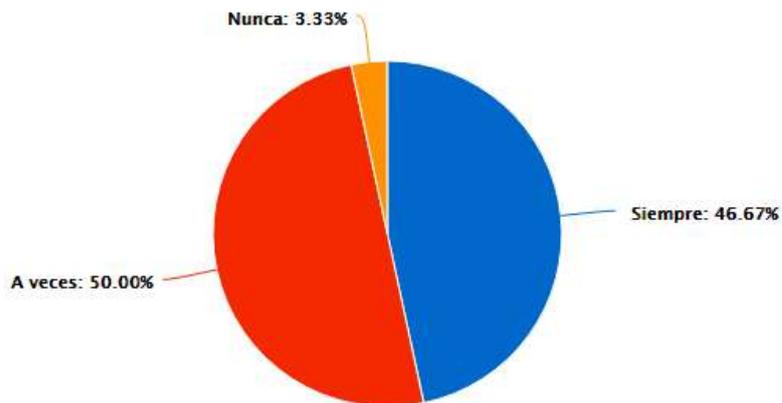


Gráfico N° 11 Maven como mejor gestor de proyectos
Elaborado por: Gina Ulloa

ANÁLISIS

Como se puede evidenciar la mitad de los encuestados manifestaron que Maven sería una buena alternativa a la hora de incluirla como herramienta de gestión de proyectos de desarrollo de software, ya que es una herramienta eficiente para las empresas desarrolladoras, es de vital importancia dar a conocer esta herramienta a los programadores por su entorno amigable y facilidad de manejo, pero para desarrollos en lenguajes diferentes a Java resulta casi imposible poder incluirla.

9. En General, ¿Cuál es su grado de satisfacción al utilizar Maven?

Indique en una escala de 1 a 10, donde 10 es completamente satisfecho y 1 es completamente insatisfecho.

Pregunta	Cantidad	Porcentaje
1	0	0.00%
2	1	3.57%
3	1	3.57%
4	0	0.00%
5	1	3.57%
6	0	0.00%
7	5	14.29%
8	13	46.43%
9	5	17.86%
10	4	10.71%
Total	30	100%

Tabla N° 15 Grado de satisfacción de Maven
Elaborado por: Gina Ulloa

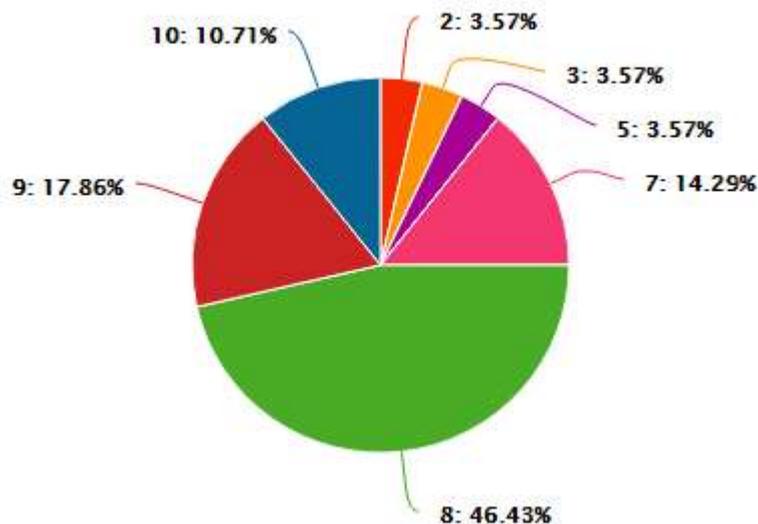


Gráfico N° 12 Grado de satisfacción de Maven
Elaborado por: Gina Ulloa

ANÁLISIS

De acuerdo a los resultados visualizados, más del 80% de los encuestados manifestaron que tienen un buen grado de satisfacción de la herramienta Maven, ya que facilita el trabajo a los desarrolladores Java en todas las fases del proyecto reduciendo los tiempos de entrega, trabajo innecesario durante la construcción del proyecto, facilidad a la hora de la compilación y despliegue, entre un sin número de ventajas.

10. ¿Qué otros gestores de proyectos conoce usted?

Pregunta	Cantidad	Porcentaje
Make	3	13.79%
Gradle	6	48.28%
NuGet	9	55.17%
Ant	7	51.72%
Mantri	1	6.90%
Bower	2	10.34%
Otro	2	10.34%
Total	30	100%

Tabla N° 16 Gestores de proyectos conocidos
Elaborado por: Gina Ulloa

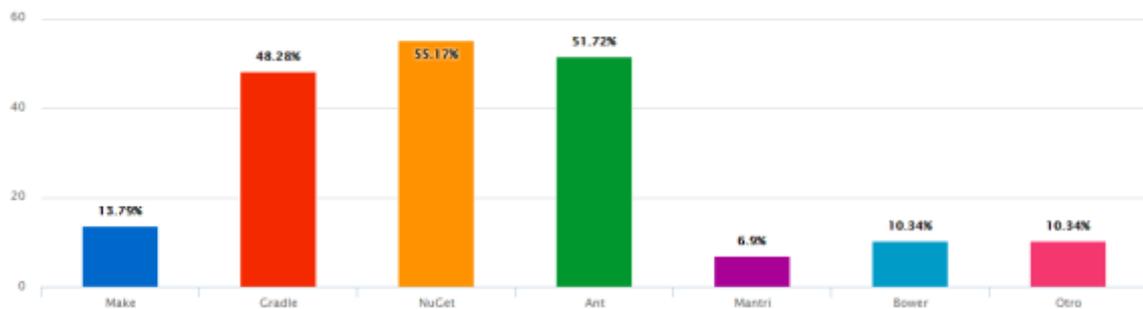


Gráfico N° 13 Gestores de proyectos conocidos
Elaborado por: Gina Ulloa

ANÁLISIS

En la figura se puede observar que los encuestados mencionaron muchos otros gestores de proyectos entre los más conocidos están Gradle, NuGet, Ant, que de igual manera ayudan a las empresas en la gestión de desarrollo de software.

11. ¿Cuál de éstos gestores ha utilizado?

Pregunta	Cantidad	Porcentaje
Make	3	10.34%
Gradle	7	44.83%
NuGet	9	48.28%
Ant	5	41.38%
Mantri	1	3.45%
Bower	2	6.90%
Otro	3	10.34%
Total	30	100%

Tabla N° 17 Gestores de proyectos utilizados
Elaborado por: Gina Ulloa

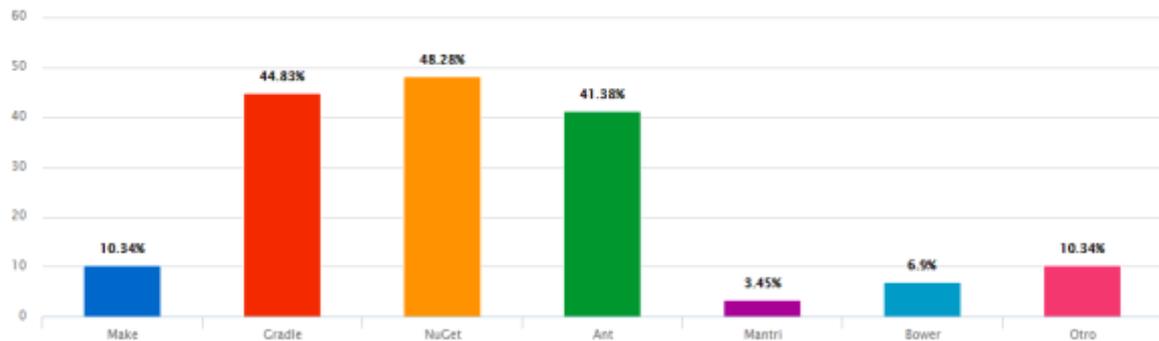


Gráfico N° 14 Gestores de proyectos utilizados
Elaborado por: Gina Ulloa

ANÁLISIS

Como se puede evidenciar en la figura los encuestados no solo conocen si no también han usado otros gestores de proyectos como Gradle, NuGet y Ant, y aun así Maven sigue siendo una excelente herramienta a la hora de elegir gestor de proyectos Java.

12. ¿En qué tipo de proyectos ha utilizado Maven?

Pregunta	Cantidad	Porcentaje
Apps de escritorio	8	29.63%
Web	14	85.19%
Móviles	6	22.22%
Otro	2	7.41%
Total	30	100%

Tabla N° 18 Tipos de proyectos generados con Maven
Elaborado por: Gina Ulloa

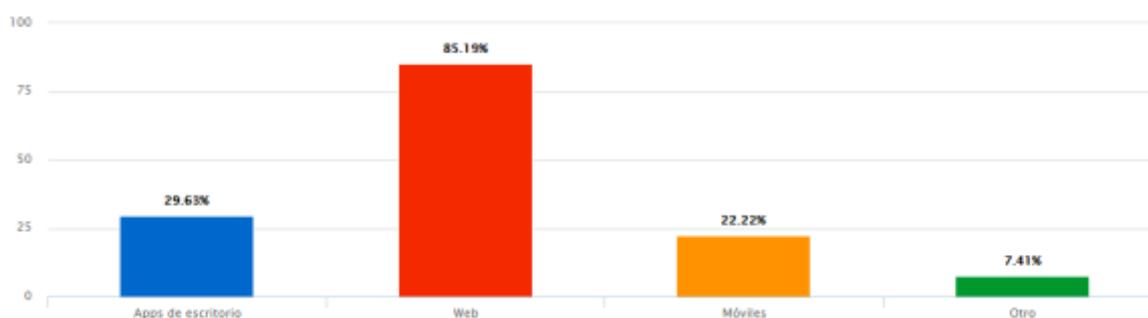


Gráfico N° 15 Tipos de proyectos generados con Maven
Elaborado por: Gina Ulloa

ANÁLISIS

De acuerdo a los resultados que indica la figura los encuestados manifestaron que Maven ha sido utilizada la mayor parte en aplicaciones Web, luego están las aplicaciones de escritorio y también están las aplicaciones móviles, es decir, Maven que es una herramienta de gestión de software completa que permite al programador elegir qué tipo de aplicaciones quiere desarrollar.

13. ¿Cree usted que es importante incluir este tema en el pensum de preparación académica?

Pregunta	Cantidad	Porcentaje
Si	29	96.67%
No	1	3.33%
Total	30	100%

Tabla N° 19 Maven como tema de estudio en la preparación académica
Elaborado por: Gina Ulloa

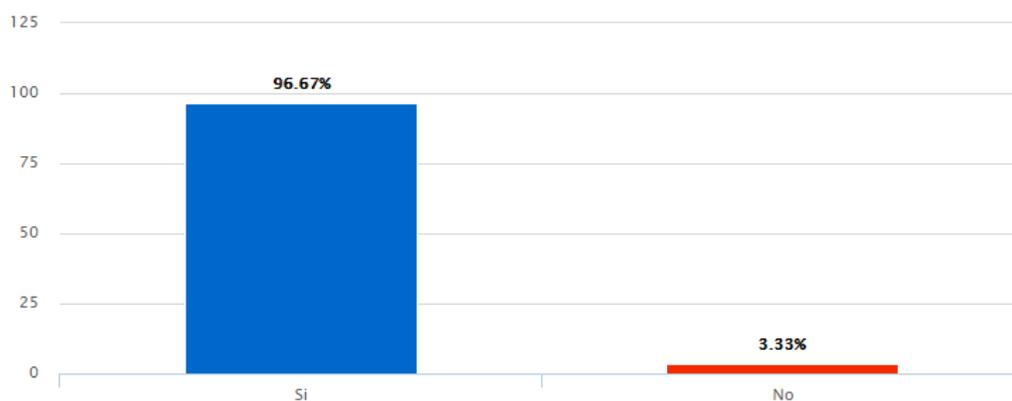


Gráfico N° 16 Maven como tema de estudio en la preparación académica
Elaborado por: Gina Ulloa

ANÁLISIS

Casi la totalidad de los encuestados manifestaron que si sería una buena opción incluir estos temas como parte del pensum de preparación académica, ya que ayudan al estudiante a investigar sobre temas actuales, estar al día en estas herramientas de desarrollo y ser competitivos en el mercado laboral.

CONCLUSIONES

- Mediante el desarrollo de la presente investigación se ha podido estudiar a tres herramientas importantes como son: Maven, Spring MVC y Spring Security, el punto fuerte de estas herramientas es el desarrollo web, están en auge y son una tendencia en el mercado actual, lo que le permite al estudiante estar actualizado en sus conocimientos.
- Se ha podido determinar que Maven es una herramienta completa que permite el desarrollo de aplicaciones de escritorio, móviles y web, permitiendo reducir el tiempo tanto en la construcción con el uso de artefactos que permiten trabajar con un entorno estandarizado, como en el desarrollo con la gestión de dependencias, simplificando la compilación de cualquier tipo de proyecto Java, así como también en pruebas y despliegue.
- Se generó documentación técnica del estudio la misma que permitirá a los interesados conocer sobre estas herramientas para que sean aplicadas en sus desarrollos.
- Se aplicó la metodología ágil Scrum en el desarrollo del proyecto lo que permitió una planificación en cada iteración y entregas continuas de los avances del proyecto, así como una fácil adaptación a cambios que el cliente solicitó.
- Con la aplicación de la encuesta se pudo determinar que existen otras herramientas de gestión de proyectos en las que se encuentra Maven que es una excelente herramienta de gestión de proyectos Java.
- Con el desarrollo del aplicativo de comercio electrónico se comprobó que el gestor Maven se adapta totalmente en la generación de proyectos MVC.

RECOMENDACIONES

- Se sugiere a los profesores de la Carrera de Sistemas incluir en su pensum de estudios, el estudio de la herramienta Maven ya que ha demostrado ser una excelente herramienta de gestión de proyectos.
- Se recomienda el uso de la herramienta Maven porque es una potente herramienta que permite la configuración rápida de los proyectos Java a desarrollarse, facilita la gestión de las dependencias necesarias, promueve la modularidad del código, entre muchas otras ventajas importantes durante todo el ciclo de vida del proyecto.
- Es importante que se impulse la investigación de herramientas innovadoras y actuales para que el estudiante tenga las bases necesarias y sepa defenderse en el mundo laboral que actualmente es muy exigente.
- Se recomienda la investigación de otros Gestores de proyectos como Gradle que también es utilizado en muchas empresas en la actualidad.
- Se recomienda que la microempresa Novedades Guayas cancele el costo del hosting para publicar la aplicación.

BIBLIOGRAFÍA

- Alberto Largo, M. E. (2005). *Guía Técnica para Evaluación de Software*. Pereira: Grupo Planeta.
- Apache Software Foundation. (04 de Febrero de 2019). *Apache Maven*. Obtenido de <http://maven.apache.org/index.html>.
- Concalves L. (25 de Enero de 2019). *Qué es la Metodología Scrum, todo lo que necesitas saber*. Obtenido de <https://luis-goncalves.com/es/que-es-la-metodologia-scrum/>.
- Esquivel, E. (2016). *Aplicación web de gestión de una academia con Spring Framework*. Sevilla: Universidad de Sevilla.
- Ganeshman, A. (2016). *Spring MVC: Beginner's Guide*. Birmingham: Packt Publishing.
- ISO. (21 de Agosto de 2018). *Organisation Internationale de Normalisation*. Obtenido de <https://www.iso.org/iso-9001-quality-management.html>.
- Johnson, R. (23 de Agosto de 2014). *Spring Framework Reference Documentation*. Obtenido de <https://docs.spring.io/spring/docs/4.0.x/spring-framework-reference/htmlsingle/#mvc-features>.
- Ken Schwaber, J. S. (15 de Noviembre de 2017). *La Guía de Scrum TM*. Obtenido de <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf#zoom=100>.
- MavenRepository. (21 de Agosto de 2006). *What's New in Maven*. Obtenido de <http://www.mvnrepository.com>.
- Mularien, P. (2010). *Spring Security*. Birmingham: Packt Publishing.
- Pivotal. (12 de Enero de 2018). *Spring Security*. Obtenido de <https://spring.io/projects/spring-security>.
- Satpathy, T. (2016). *Una Guía para el Conocimiento de Scrum*. Arizona: ScrumStudy TM.
- Sonatype. (2008). *Maven: The Definitive Guide*. USA: O'Reilly Media Books.
- Scrum.org. (12 de Enero de 2019). *What is Scrum*. Obtenido de <https://www.scrum.org/resources/what-is-scrum>.

ANEXOS

ANEXO 1: MANUAL TÉCNICO

1. Herramientas utilizadas para el desarrollo

- **Maven**

Es una herramienta de Gestión de proyectos Java utilizada durante todo el ciclo de vida del proyecto desde su construcción, además permite la administración de dependencias de una manera sencilla, mediante la configuración de un archivo xml. Además, utiliza plugins que permiten ampliar las funcionalidades básicas de Maven.

- **Angular 7**

Es un framework de código abierto para desarrollo de aplicaciones web que permite crear aplicaciones de una sola página SPA, además permite separar entre el front-end y back-end, se orienta más a objetos y con una arquitectura basada en componentes.

- **Postgresql 10**

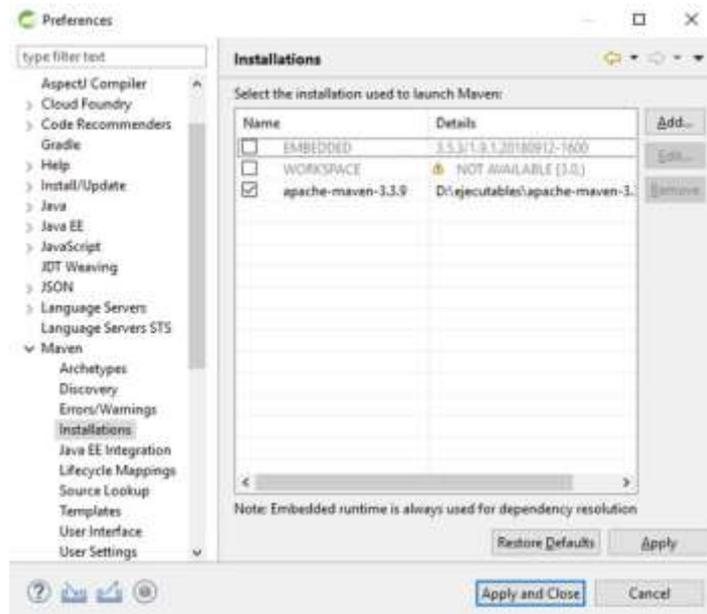
Es un servidor de base de datos el cual presenta varias características como el soporte a varios tipos de datos, herencia entre tablas, copias de seguridad, facilidad en su administración, entre otras, lo cual la hace una potente base de datos.

- **Tomcat**

Es un contenedor web usado para proyectos Java con soporte de servlets y JSP, que permite fácilmente desplegar las aplicaciones.

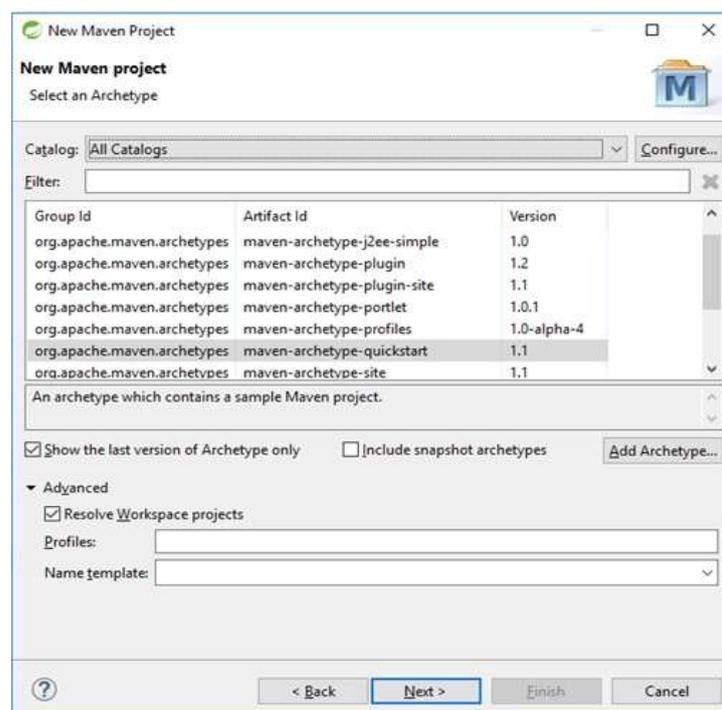
2. Configuración de Maven

Previamente en la documentación técnica de Maven se indicó la manera de como instalar la herramienta en el equipo. Para la configuración de Maven en el IDE de desarrollo Spring Tool Suite es necesario ir al menú “**Windows > Preferences**”, en donde se mostrará la siguiente pantalla:

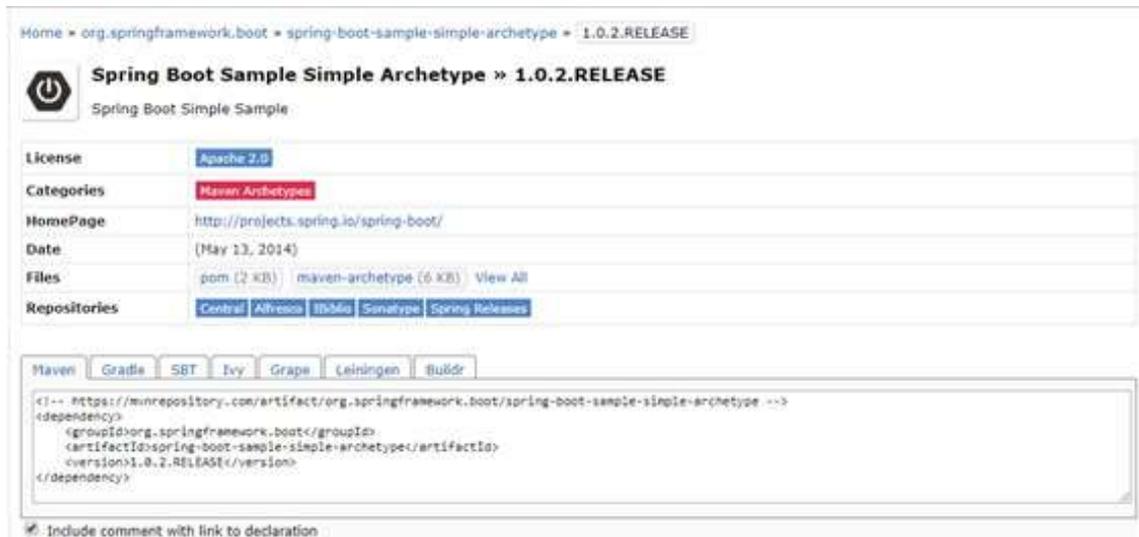


En esta parte deberá elegir “**Maven > Installations**”, en donde se añadirá el **path** donde se encuentra ubicado Maven dentro de su equipo, a continuación, seleccionar el elemento que acaba de añadir y clic en **Apply and Close**.

Para el desarrollo de la aplicación se utilizó la plantilla **spring-boot-sample-simple-archetype** la cual no se encuentra en los arquetipos por defecto que incluye Maven por lo que se la añadirá haciendo clic en **Add Archetype**.



En el repositorio central de Maven se puede buscar el tipo de arquetipo que se desea utilizar y este le indicará la información para agregarlo:



The screenshot shows the Maven Central repository page for the artifact `org.springframework.boot:spring-boot-sample-simple-archetype:1.0.2.RELEASE`. The page includes the following information:

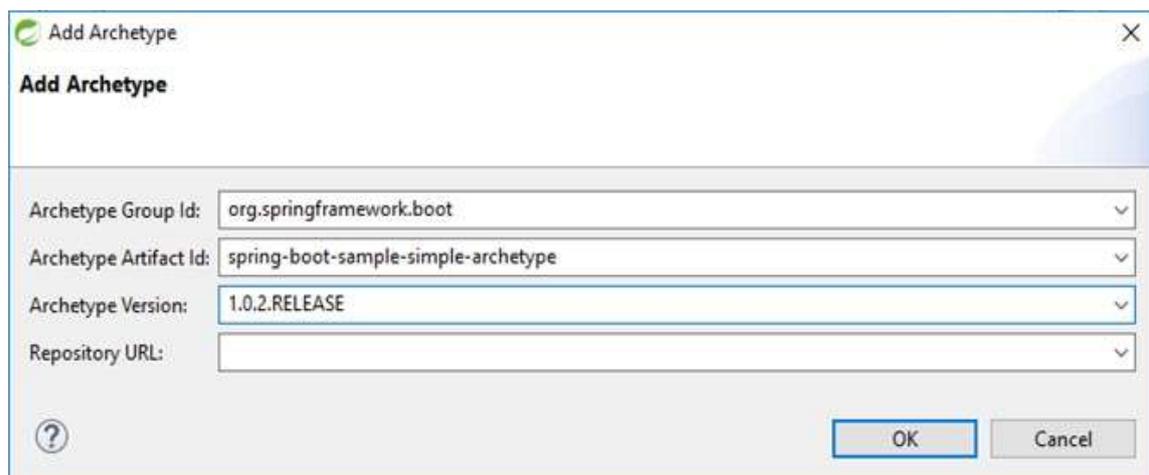
- License:** Apache 2.0
- Categories:** Maven Archetypes
- HomePage:** <http://projects.spring.io/spring-boot/>
- Date:** (May 13, 2014)
- Files:** pom (2 KB), maven-archetype (6 KB), View All
- Repositories:** Central, Alfresco, IBMId, Sonatype, Spring Releases

Below the metadata, there are tabs for different build systems: Maven, Gradle, SBT, Ivy, Grapes, Leiningen, and Buildr. The Maven tab is selected, and it displays the following XML snippet:

```
<!-- https://mavenrepository.com/artifact/org.springframework.boot/spring-boot-sample-simple-archetype -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-sample-simple-archetype</artifactId>
  <version>1.0.2.RELEASE</version>
</dependency>
```

There is a checkbox labeled "Include comment with link to declaration" which is checked.

Seguidamente se colocará esta información en la ventana del IDE a desarrollarse y se podrá visualizar que el artefacto se ha cargado.

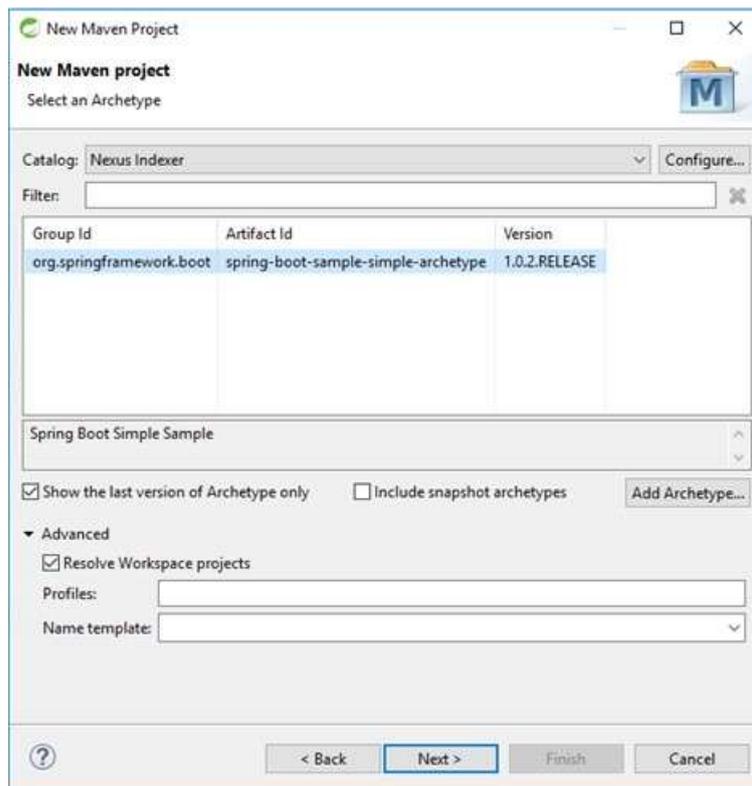


The screenshot shows the "Add Archetype" dialog box in an IDE. The dialog has a title bar with a question mark icon and a close button. The main content area is titled "Add Archetype" and contains four input fields:

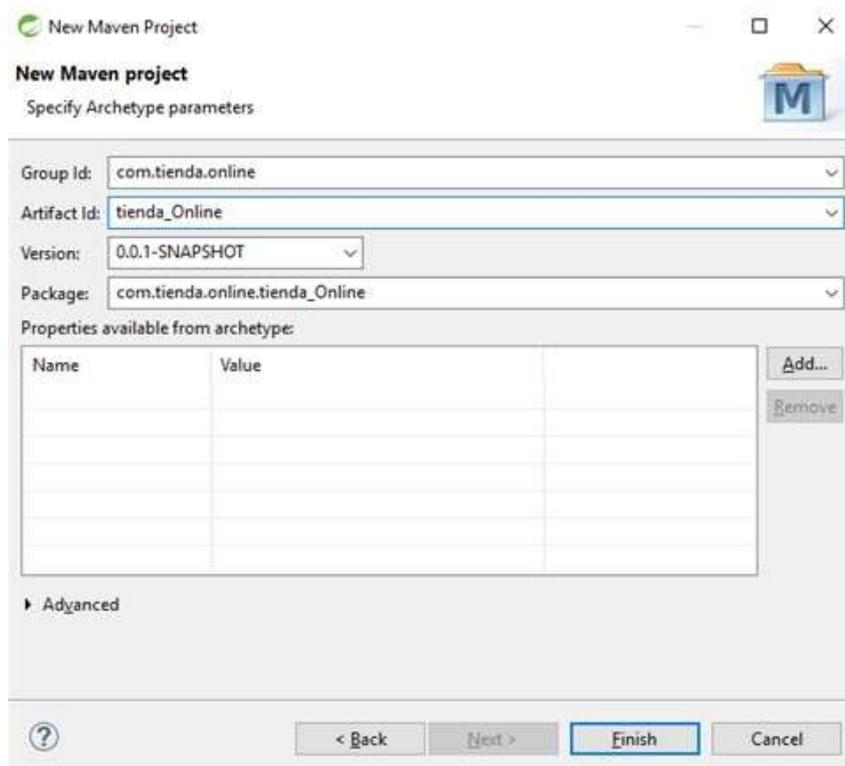
- Archetype Group Id:** org.springframework.boot
- Archetype Artifact Id:** spring-boot-sample-simple-archetype
- Archetype Version:** 1.0.2.RELEASE
- Repository URL:** (empty)

At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

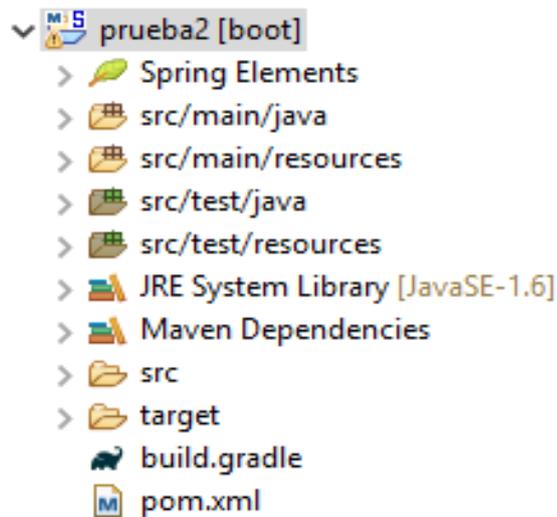
A continuación se desplegará una ventana donde indica el arquetipo que se ha descargado, se deberá seleccionar y dar clic en siguiente, como indica la figura.



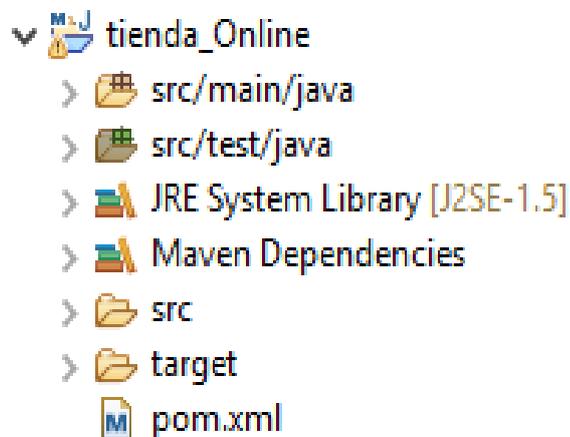
A continuación, deberá ingresar la información del Group Id, Artific Id y Version del proyecto a desarrollarse.



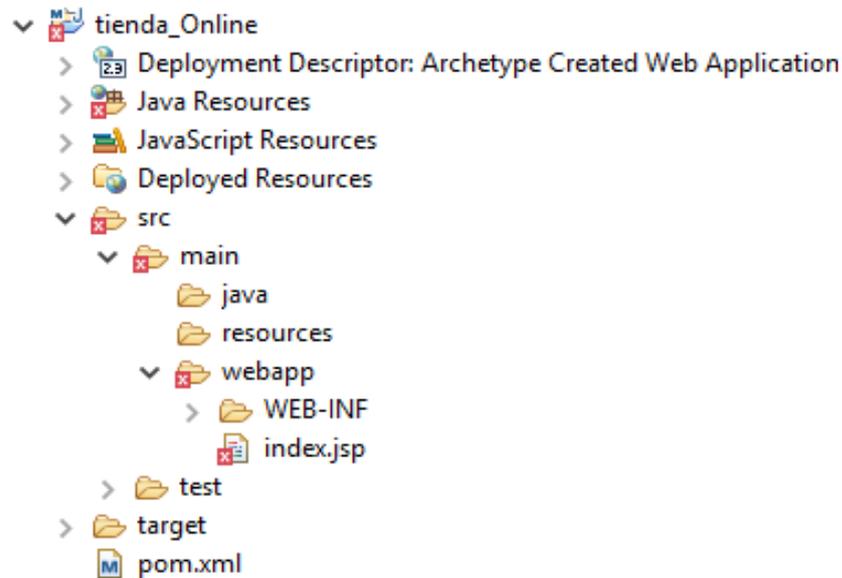
Una vez finalizada la creación del proyecto se podrá visualizar la estructura estándar que Maven crea para el proyecto dependiendo del arquetipo seleccionado.



Para un proyecto jar generado con el arquetipo estándar Maven-archetype-quickstart de Maven la estructura será la siguiente:



Para un proyecto war generado desde el arquetipo Maven-archetype-webapp la estructura sería la siguiente:



Además, cada arquetipo genera un archivo pom.xml básico, el cual deberá ser modificado según los requerimientos del proyecto.

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4
5   <groupId>utn.edu.ec</groupId>
6   <artifactId>prueba1</artifactId>
7   <version>0.0.1-SNAPSHOT</version>
8   <packaging>jar</packaging>
9
10  <name>prueba1</name>
11  <url>http://maven.apache.org</url>
12
13  <properties>
14    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15  </properties>
16
17  <dependencies>
18    <dependency>
19      <groupId>junit</groupId>
20      <artifactId>junit</artifactId>
21      <version>3.8.1</version>
22      <scope>test</scope>
23    </dependency>
24  </dependencies>
25  <reporting>
26    <plugins>
27      <plugin>
28        <groupId>org.apache.maven.plugins</groupId>
29        <artifactId>maven-site-plugin</artifactId>
30        <version>3.7.1</version>
31      </plugin>
32      <plugin>
33        <groupId>org.apache.maven.plugins</groupId>
34        <artifactId>maven-project-info-reports-plugin</artifactId>
35        <version>3.0.0</version>
36      </plugin>
37    </plugins>
38  </reporting>
39 </project>
40
```

El archivo pom.xml del proyecto quedaría de la siguiente manera:

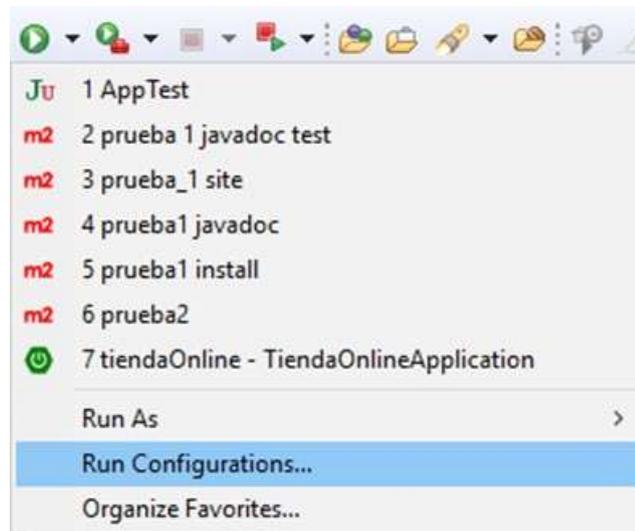
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6
7   <groupId>com.tienda.online</groupId>
8   <artifactId>tiendaOnline</artifactId>
9   <version>0.0.1-SNAPSHOT</version>
10  <packaging>jar</packaging>
11
12  <name>tiendaOnline</name>
13  <description>Proyecto inicial, Tienda Online</description>
14
15  <parent>
16    <groupId>org.springframework.boot</groupId>
17    <artifactId>spring-boot-starter-parent</artifactId>
18    <version>2.0.3.RELEASE</version>
19    <relativePath /> <!-- lookup parent from repository -->
20  </parent>
21
22  <properties>
23    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
24    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
25    <java.version>1.8</java.version>
26  </properties>
27
28  <dependencies>
29    <dependency>
30      <groupId>org.springframework.boot</groupId>
31      <artifactId>spring-boot-starter-data-jpa</artifactId>
32    </dependency>
33    <dependency>
34      <groupId>org.springframework.boot</groupId>
35      <artifactId>spring-boot-starter-security</artifactId>
36    </dependency>
37    <dependency>
38      <groupId>org.springframework.boot</groupId>
39      <artifactId>spring-boot-starter-web</artifactId>
40    </dependency>
41
42    <dependency>
43      <groupId>org.postgresql</groupId>
44      <artifactId>postgresql</artifactId>
45      <scope>runtime</scope>
46    </dependency>
47
48    <dependency>
49      <groupId>org.springframework.boot</groupId>
50      <artifactId>spring-boot-starter-mail</artifactId>
51    </dependency>
52
53    <dependency>
54      <groupId>org.springframework.boot</groupId>
55      <artifactId>spring-boot-starter-thymeleaf</artifactId>
56    </dependency>
57
58    <!-- generat randomico -->
59    <dependency>
60      <groupId>org.apache.commons</groupId>
61      <artifactId>commons-lang</artifactId>
62    </dependency>
63
64    <dependency>
65      <groupId>org.springframework.boot</groupId>
66      <artifactId>spring-boot-starter-test</artifactId>
67      <scope>test</scope>
68    </dependency>
69    <dependency>
70      <groupId>org.springframework.security</groupId>
71      <artifactId>spring-security-test</artifactId>
72      <scope>test</scope>
73    </dependency>
74
75
76    <dependency>
77      <groupId>io.jsonwebtoken</groupId>
78      <artifactId>jjwt</artifactId>
79      <version>0.7.0</version>
80    </dependency>
81
82  </dependencies>
83
84  <build>
85    <plugins>
86      <plugin>
87        <groupId>org.springframework.boot</groupId>
88        <artifactId>spring-boot-maven-plugin</artifactId>
89      </plugin>
90    </plugins>
91  </build>
92
93 </project>
94
95
96
97
```

Cabe recalcar que la creación de un proyecto también se puede realizar mediante la ejecución de comandos de Maven en la consola como se indicó en la documentación técnica. Ahora se especificará los objetivos o **goals** principales utilizados en el proyecto, los mismos que se detallan a continuación:

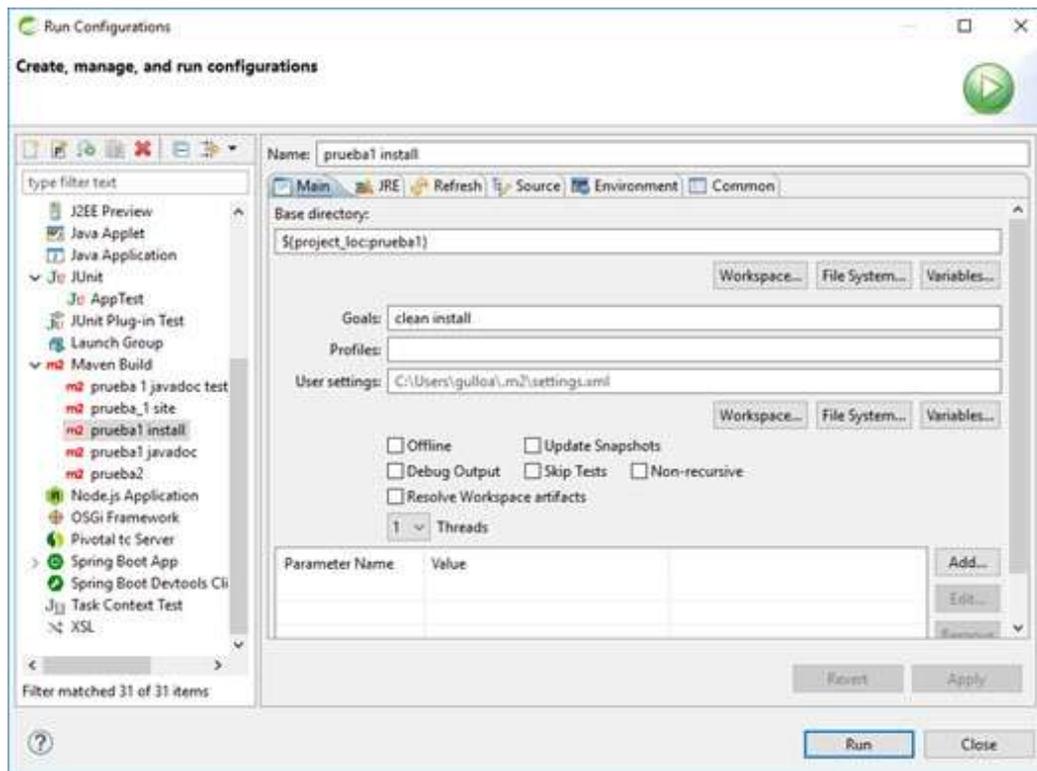
- **Clean:** permite borrar el directorio de salida, en este caso el target que es común para todos los proyectos independientemente del arquetipo seleccionado.
- **Install:** permite guardar el proyecto en el repositorio creado.

Como se mencionó en la documentación técnica de Maven al ejecutar un **goals** se ejecutarán todos sus antecesores del ciclo de vida, en este caso al ejecutar el comando **mvn clean install** se ejecutarán los objetivos **compile, test y package**.

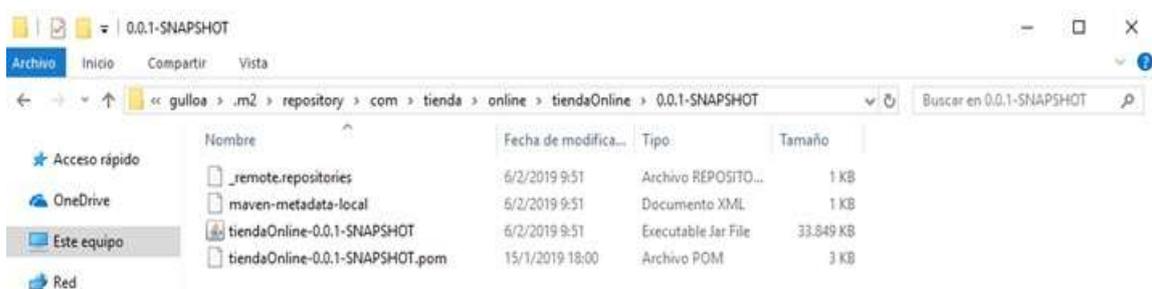
El IDE facilita la configuración de los **goals** del proyecto para mantenerlos listos para su ejecución, de la siguiente manera:



A continuación deberá escoger “**Run Configurations...**”, se creará una configuración nueva y se deberá indicar el nombre de la configuración, escoger el proyecto e indicar los **goals** que se desea ejecutar.



Esta configuración estará disponible cada vez que se requiera ejecutarla, ya no es necesario volver a configurar. Ahora se podrá verificar donde se cargó el proyecto compilado y empaquetado dentro del repositorio local donde se almacenan todas las dependencias de los proyectos, este directorio se puede cambiar en el archivo **settings.xml**, por defecto se encuentra en la carpeta **C:/Users/\${usuario}/.m2**.

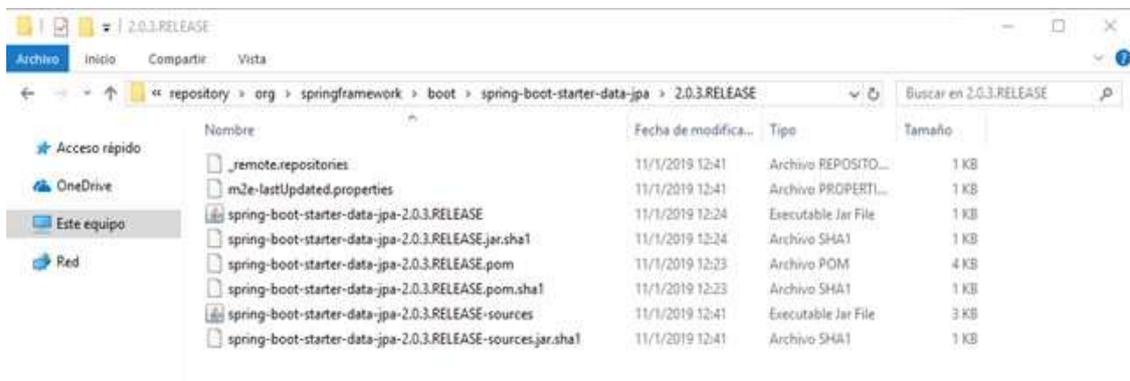


El directorio del proyecto jar se crea en una estructura de carpetas igual al group Id especificado, en este caso es **<groupId>com.tienda.online</groupId>**, el nombre del proyecto empaquetado viene dado por el **<artifactId>-<version>** indicados en el pom, de ahí la ventaja del versionamiento con Maven.

Además, se podrá validar las dependencias que Maven descargó desde su repositorio central al repositorio local:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
  <version>2.0.3.RELEASE</version>
</dependency>
```

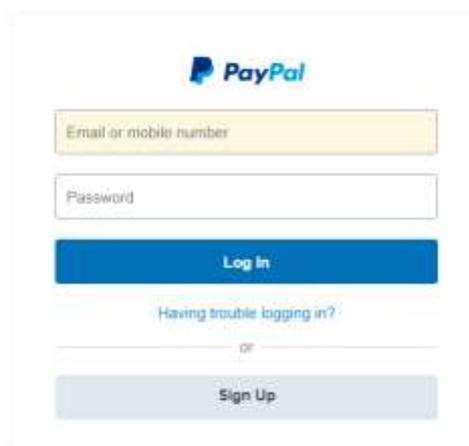
Esta es una de las dependencias que fueron utilizadas en el proyecto y en el repositorio local, se encuentra en la ruta de su groupId como se indicó anteriormente.



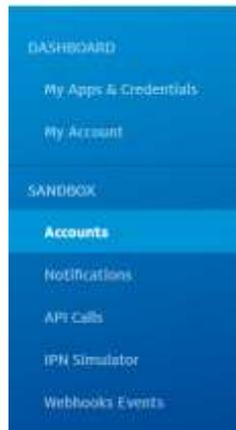
El archivo de configuración de Maven settings.xml que se utilizó es el que trae por defecto Maven, no se realizó ninguna configuración adicional.

3. Configuración botón PayPal

La configuración para pagos que se utilizó en la aplicación es el botón PayPal, para su implementación es necesario tener una cuenta PayPal, a continuación, ingresamos a la página developer de PayPal: <https://developer.paypal.com/> y nos logueamos:



Una vez que ingresamos seleccionamos en el menú izquierdo Accounts para crear las cuentas de prueba que utilizaremos para la aplicación:

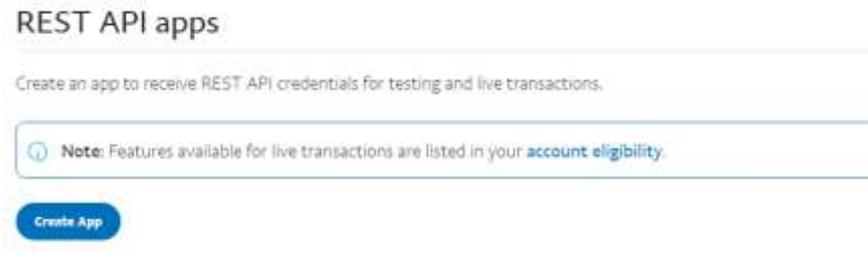


Es necesario crear una cuenta business y una cuenta personal para realizar todo el flujo de prueba, la creación de la cuenta BUSINESS será así:

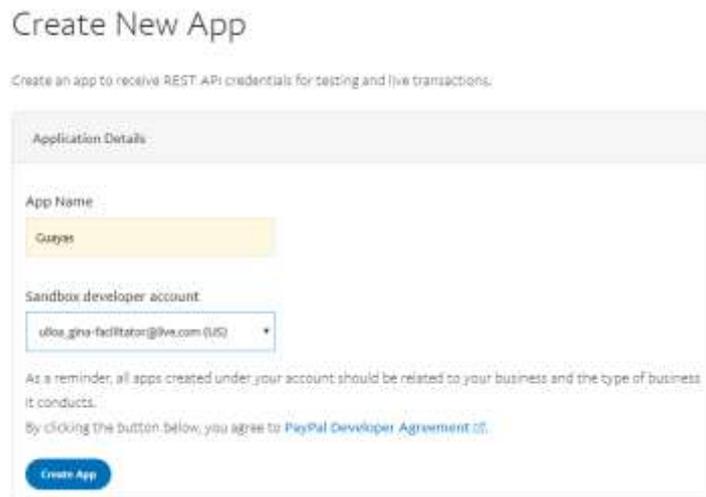
A screenshot of the PayPal Developer 'Create Sandbox Account' page. The page has a blue sidebar on the left with the 'Accounts' menu item selected. The main content area is white and contains the following sections:

- Alert:** Active service disruptions may be affecting Production performance. For details, see [PayPal Status](#).
- Create Sandbox Account:** You can also create a personal or business sandbox account on [sandbox.paypal.com](#).
- Account Details:**
 - Country:** A dropdown menu showing 'United States'.
 - Note:** Select the country that will be used for live payments or the country that you're developing the live application for.
 - Account Type:** Radio buttons for 'Personal (Buyer Account)' and 'Business (Merchant Account)'. 'Business' is selected.
 - Email Address:** A text input field containing 'ofice.gira.fed@hater@live.com'.
 - password:** A text input field for a password (8-30 characters).
 - First Name (optional):** A text input field containing 'Gira'.
 - Last Name (optional):** A text input field containing 'ofice'.
- Payment Methods:**
 - PayPal Balance:** A text input field with '100' and a currency dropdown set to 'USD'.
 - Bank Verified Account:** Radio buttons for 'Yes' and 'No'. 'No' is selected.
 - Credit Cards:** A dropdown menu showing 'Visa'.
 - Note (optional):** A text input field.
- Create Account:** A blue button at the bottom.

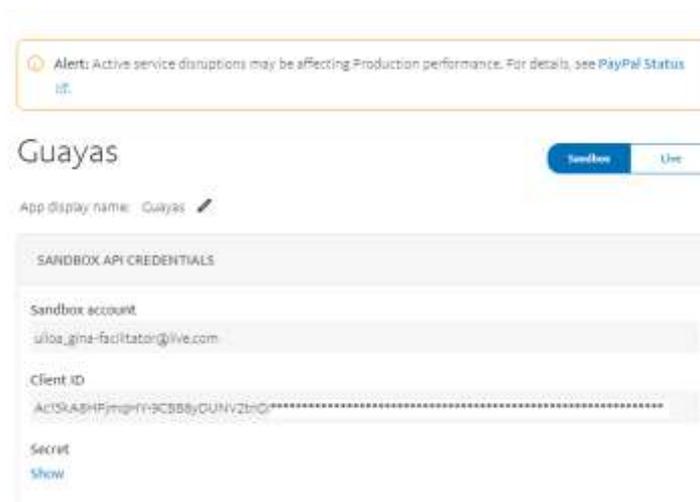
Esta cuenta se vinculará con nuestra aplicación, para esto vamos a configurar una app, como la aplicación utiliza servicios REST la crearemos en la opción **REST API apps**:



En la opción **Sandbox Developer Account** vamos a escoger el usuario anteriormente creado:



A continuación, nos muestra la información que será configurada en nuestra aplicación mediante el **Client ID**, en esta pantalla se encuentra dos opciones de configuración **Sandbox** asociada a nuestra cuenta de prueba y **Live** asociada a nuestra cuenta real.



Alert: Active service disruptions may be affecting Production performance. For details, see [PayPal Status](#) [\(E\)](#).

Guayas

App display name: Guayas 

LIVE API CREDENTIALS

Note that not all features are available for live transactions. Features available for live transactions are listed in your account eligibility.

PayPal account:
alice_gina@live.com

Client ID:
ARSU0bvAgj_de76VpCof8F0Xhu70t_NWyr8gtZ-*****

Secret:
[Show](#)

Ahora crearemos la cuenta **PERSONAL** de la siguiente manera:

PayPal Developer Docs API Support Search Close

Alert: Active service disruptions may be affecting Production performance. For details, see [PayPal Status](#) [\(E\)](#).

Create Sandbox Account

You can also create a personal or business sandbox account on [sandbox.paypal.com](#) [\(E\)](#).

Account Details

Country
United States

Note: Select the country that will be used for live payments or the country that you're developing the live application for.

Account Type
 Personal (Buyer Account)
 Business (Merchant Account)

First Name (optional)

Last Name (optional)

Payment Methods

PayPal Balance 00 USD

Bank Verified Account
 Yes No

Select Payment Card
 Discover PayPal

Credit Card

Notes (optional)

[Create Account](#)

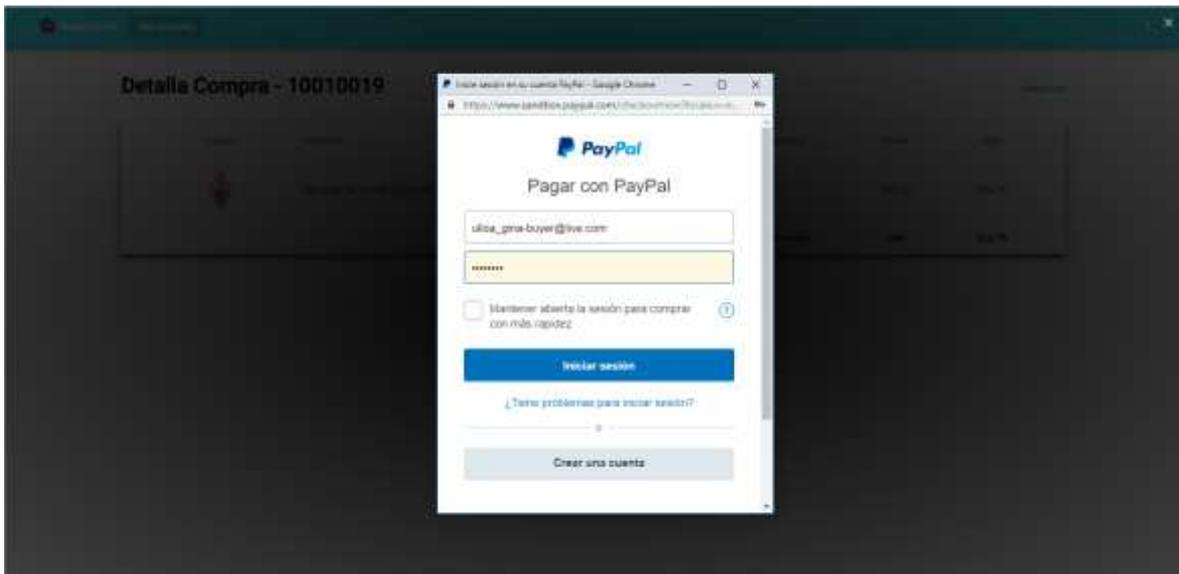
Además, en la aplicación es necesario crear el botón de pago con PayPal, el código necesario para su creación depende de la tecnología usada en cada aplicación:

```
57 paypalConfig = {
58   env: 'sandbox', // sandbox | production
59   client: {
60     sandbox: 'Ac15kABHPjmqHY-9CBB8yDUNV2tnDx2TaZ_*****',
61     production: 'AR5U0bvAqI_de76VpGXsfBFXkhu705_*****'
62   },
63   commit: true,
64   payment: (data, actions) => {
65     return actions.payment.create({
66       payment: {
67         transactions: [
68           { amount: { total: this.pedidoSelected.total, currency: 'USD' } }
69         ]
70       }
71     });
72   },
73   onAuthorize: (data, actions) => {
74     return actions.payment.execute().then((payment) => {
75       //Do something when payment is successful.
76       if (payment) {
77         this.pedidoSelected.payId = payment.id;
78         this.pedidoSelected.estado = "PAGADO";
79         this.pedidoSelected.formaPago = "PAYPAL";
80         this.pedidoSelected.payerInfo = payment.payer.payer_info.email;
81
82         this.pedidoService.pagoPaypalCompra(this.pedidoSelected).subscribe(resp => {
83           if (resp) {
84             this.pedidoSelected = resp;
85             this.openSnackBar("Su pedido se ha realizado con éxito", "OK", .);
86             this.getPedidos();
87             this.regresarCompra()
88           }
89         });
90       }
91     });
92   }
93 };
```

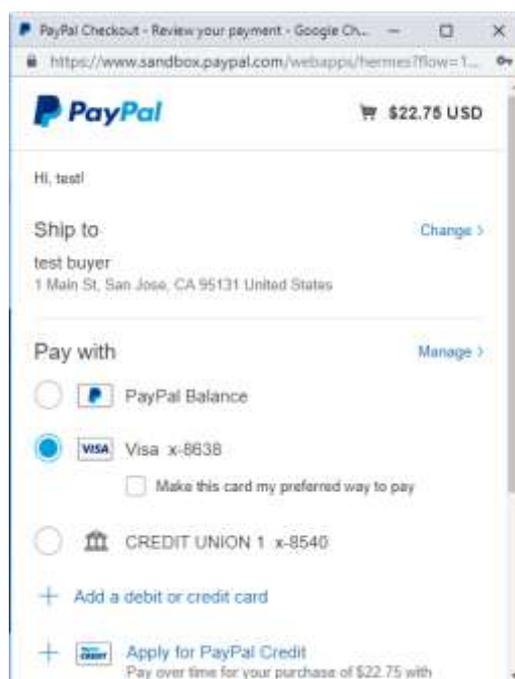
Para verificar que los pagos se estén realizando correctamente vamos a realizar una compra:



En este caso la compra se realizó por el valor de USD 22.75, una vez presionamos el botón PayPal nos pedirá loguearnos con nuestra cuenta, en este caso será la cuenta personal que creamos anteriormente:



Una vez logueados poder visualizar el monto de la compra, como es un usuario para pruebas se muestran algunas opciones de pago, en este caso seleccionamos que el pago se realizará con la tarjeta Visa y seleccionamos **Pay Now**.



Como el pago se realizó correctamente la aplicación muestra la compra en estado **PAGADO**.

Tests Online | Mis Compras

Mis Compras

Fecha Compra	Id Compra	Total	Estado	Detalle
16/02/2019	10010019	\$22.75	PAGADO	
15/02/2019	10010018	\$58.00	PAGADO	
14/02/2019	10010017	\$58.00	PAGADO	
14/02/2019	10010016	\$18.00	PAGADO	
03/01/2019	10010015	\$14.00	PAGADO	
03/01/2019	10010014	\$14.00	PAGADO	
24/01/2019	10010013	\$17.99	PAGADO	

1 - 7 de 7 | 1 - 7 de 7 | 11 | 12 | 13

Ahora validamos la transacción con nuestros usuarios de prueba, para esto nos logueamos en <https://www.sandbox.paypal.com> con el usuario **PERSONAL**.



ulloa_gina-buyer@live.com

.....

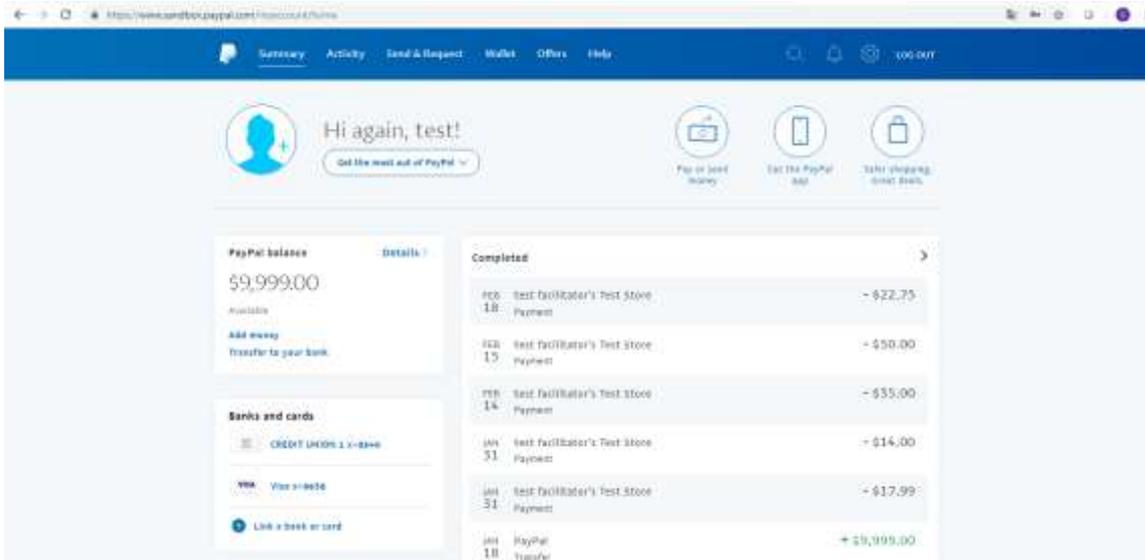
Log In

Having trouble logging in?

or

Sign Up

Verificamos que se realizó una transacción por el monto de la compra.



Summary | Activity | Send & Request | Wallet | Offers | Help

Hi again, test!

PayPal balance: **\$9,999.00** Available

Completed transactions:

Date	Description	Amount
Feb 18	test facilitator's test store Payment	-\$22.75
Feb 15	test facilitator's test store Payment	-\$58.00
Feb 14	test facilitator's test store Payment	-\$58.00
Jan 31	test facilitator's test store Payment	-\$14.00
Jan 31	test facilitator's test store Payment	-\$14.00
Jan 18	test facilitator's test store Payment	-\$17.99
Jan 18	PayPal Transfer	+\$9,999.00

Este es el detalle del pago, con lo que validamos que el pago se realizó a la cuenta **BUSINESS** configurada en nuestra aplicación.

Completed >

FEB 18 test facilitator's Test Store Payment - \$22.75

Ship to test buyer 1 Main St San Jose, CA 95131 United States	Seller info ✕ test facilitator's Test Store ulloa_gina-facilitador@live.com
Transaction ID OG417077TR789914P	Purchase details Purchase amount \$22.75
	Total \$22.75

 [Print details](#)

Need help?
If there's a problem, make sure to contact the seller through PayPal by **August 17, 2019**. You may be [eligible for purchase protection](#).

 [Report a problem](#)

Ahora validamos la transacción con el usuario **BUSINNES**:



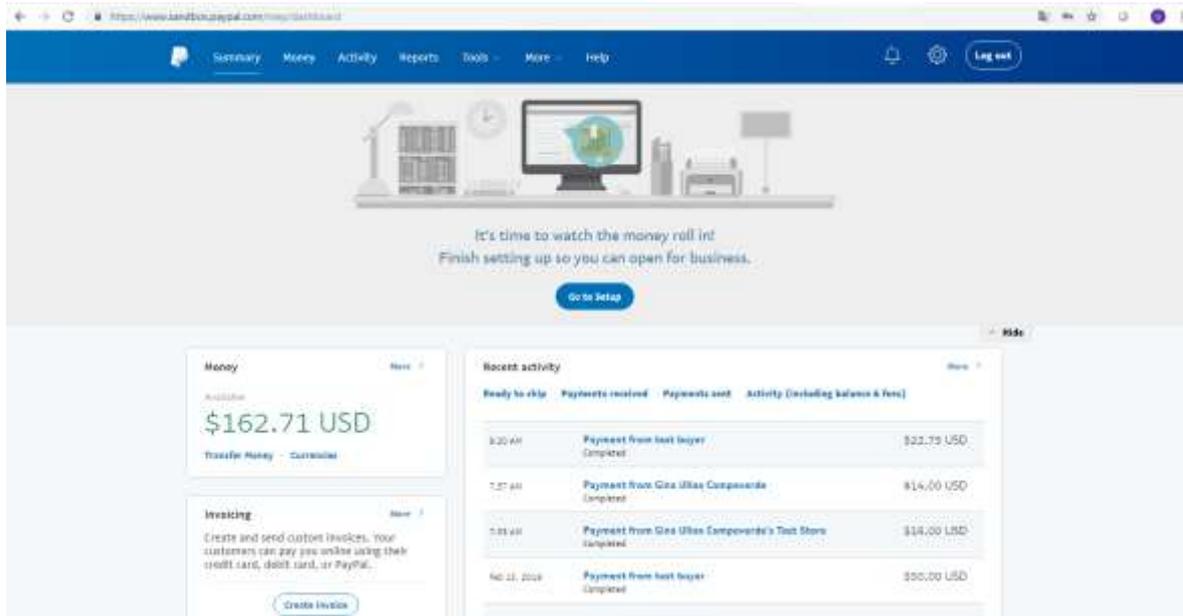
[Log In](#)

[Having trouble logging in?](#)

or

[Sign Up](#)

Verificamos que se realizó una transacción por el monto de la compra:



En el detalle se puede verificar que el pago fue realizado por nuestro usuario PERSONAL.

Paid by test buyer
The sender of this payment is **Verified**
ulloa_gina-buyer@live.com

Payment Sent to ulloa_gina-facilitator@live.com

ANEXO 2: ENCUESTA DE SATISFACCIÓN DE USUARIOS

Gracias por contestar esta encuesta, no tardará más de unos minutos en responder. Su identidad será confidencial y los resultados de esta encuesta se utilizará para fines de este estudio, sus respuestas son muy importantes.

1. Nombre de su empresa

.....
.....

2. ¿Usted ha utilizado alguna herramienta para gestionar sus proyectos de desarrollo de software?

Si

No

3. ¿Cree usted que es necesario utilizar una herramienta para gestionar sus proyectos de desarrollo de software?

Si

No

4. ¿Usted ha utilizado Maven?

Si

No

5. ¿Cree usted que cualquier programador sin ningún tipo de experiencia puede utilizar Maven?

Si

No

6. ¿Cree usted que se ha reducido el tiempo en el desarrollo de sus proyectos de software con Maven?

Si

No

7. ¿El gestor de proyectos Maven se adapta a las necesidades su empresa?

- Totalmente
- Parcialmente
- Nada

8. ¿Cree usted que Maven es mejor que otros gestores de proyectos de desarrollo de software?

- Si
- No

9. ¿Le gustaría incluir a Maven en sus proyectos de desarrollo de software?

- Siempre
- A veces
- Nunca

10. En General, ¿Cuál es su grado de satisfacción al utilizar Maven?

Indique en una escala de 1 a 10, donde 10 es completamente satisfecho y 1 es completamente insatisfecho.

Muy bajo Medio Alto

1 2 3 4 5 6 7 8 9 10

11. ¿Qué otros gestores de proyectos conoce usted?

- Make
- Gradle
- NuGet
- Ant
- Mantri
- Bower
- Otro (¿cuál?)

.....

.....

12. ¿Cuál de éstos gestores ha utilizado?

Make

Gradle

NuGet

Ant

Mantri

Bower

Otro (¿cuál?)

.....
.....

13. ¿En qué tipo de proyectos ha utilizado Maven?

Apps de escritorio

Web

Móviles

Otro (¿cuál?)

.....
.....

14. ¿Cree usted que es importante incluir este tema en el pensum de preparación académica?

Si

No

Porque

.....
.....

15. Indique la mayor ventaja y desventaja de utilizar Maven

Ventaja:.....
.....
.....

Desventaja:.....

.....

.....

ANEXO 3: CERTIFICACIÓN DE LA EMPRESA



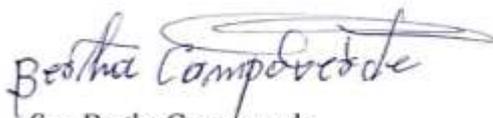
Ibarra, 13 de febrero del 2019

Ingeniero
Pedro Granda
DIRECTOR CISIC
Presente.-

Por medio del presente me permito dirigirme a usted para informarle que Bazar y Novedades Guayas, después de haber facilitado información para la realización del trabajo de investigación titulado: "Estudio de la herramienta Maven como gestor de proyectos Spring MVC con el caso de uso aplicación para comercio electrónico", desarrollado por la estudiante Gina Ulloa, acepta y da por culminado el proyecto de ventas por catálogo online.

Es todo cuanto puedo certificar, particular que pongo en su conocimiento.

Atentamente,



Sra. Bertha Campoverde

Gerente – Propietaria Bazar y Novedades Guayas