



# **INGENIERÍA DE SOFTWARE**

## **Proyecto Carpintería**

### **Módulo Compra**

**Docente:** Hugo Senda

**Carrera:** Lic. En Análisis De Sistemas Informático.

**Curso:** 4° Año

**Semestre:** 7°

#### **Integrantes:**

- Romina Patiño
- Alejandro Villalba
- Patricia Fernández
- Francisco Almada
- Giovani Giménez

**2020**

# INDICE

<b>METODOLOGÍA DE DESARROLLO .....</b>	<b>6</b>
<b>METODOLOGÍA SCRUM .....</b>	<b>6</b>
PLANIFICACIÓN DE LA ITERACIÓN .....	7
EJECUCIÓN DE LA ITERACIÓN.....	7
<b>PLANIFICACIÓN.....</b>	<b>8</b>
PROBLEMÁTICA.....	8
OBJETIVOS.....	9
<i>Objetivo general:</i> .....	9
<i>Objetivos específicos:</i> .....	9
LIMITACIONES Y ALCANCE DEL TEMA .....	9
<i>Alcances:</i> .....	9
<i>Limitaciones:</i> .....	9
<b>FACTIBILIDAD.....</b>	<b>10</b>
TÉCNICA.....	10
OPERATIVA .....	10
ECONÓMICA .....	10
<b>REQUERIMIENTOS MODELO DE NEGOCIOS.....</b>	<b>10</b>
IDENTIFICAR ACTIVIDAD DEL SISTEMA: .....	10
<b>DIAGRAMA CASO DE USO - MODELADO DE NEGOCIO (CUMN).....</b>	<b>11</b>
<b>TABLA DEL CASO DE USO DEL MODELO DE NEGOCIO.....</b>	<b>11</b>
<b>DIAGRAMA DE CLASES(CUMN) .....</b>	<b>13</b>
<b>DIAGRAMA DE COMPONENTES(CUMN).....</b>	<b>14</b>
<b>DIAGRAMA DE OBJETOS(CUMN) .....</b>	<b>15</b>
<b>DIAGRAMA DE DESPLIEGUE(CUMN) .....</b>	<b>16</b>
<b>DIAGRAMA DE PAQUETES(CUMN) .....</b>	<b>17</b>
<b>DIAGRAMA DE ACTIVIDADES(CUMN) .....</b>	<b>18</b>
<b>DIAGRAMA DE ESTADOS (CUMN) .....</b>	<b>19</b>
<b>DIAGRAMA DE SECUENCIA (CUMN).....</b>	<b>21</b>
<b>DIAGRAMA DE COLABORACION (CUMN).....</b>	<b>21</b>

<b>REQUERIMIENTOS MODELO DE REQUISITOS .....</b>	<b>23</b>
<b>IDENTIFICAR LOS CU A REALIZAR: .....</b>	<b>23</b>
<b>DIAGRAMAS DEL SUB MODULO INGRESAR AL MÓDULO COMPRAS .....</b>	<b>23</b>
DIAGRAMA DE ACTIVIDADES.....	23
DIAGRAMA DE CASOS DE USO .....	24
DIAGRAMA DE ESTADOS.....	25
DIAGRAMA DE SECUENCIA.....	26
DIAGRAMA DE COLABORACIÓN .....	26
<b>DIAGRAMAS DEL SUB MODULO NUEVO INGRESO .....</b>	<b>27</b>
DIAGRAMAS DE COMPARTIMIENTO .....	27
<i>Diagrama de Actividades</i> .....	27
<i>Diagrama de casos de uso</i> .....	28
<i>Diagrama de estados</i> .....	30
DIAGRAMAS DE INTERACCIÓN.....	31
<i>Diagrama de secuencia</i> .....	31
<i>Diagrama de colaboración</i> .....	31
<b>DIAGRAMAS DEL SUB MODULO ORDEN DE COMPRA .....</b>	<b>32</b>
DIAGRAMAS DE COMPARTIMIENTO .....	32
<i>Diagrama de Actividades</i> .....	32
<i>Diagrama de casos de uso</i> .....	33
<i>Diagrama de estados</i> .....	35
DIAGRAMAS DE INTERACCIÓN.....	36
<i>Diagrama de secuencia</i> .....	36
<i>Diagrama de colaboración</i> .....	37
<b>DIAGRAMAS DEL SUB MODULO NUEVO PROVEEDOR.....</b>	<b>38</b>
DIAGRAMAS DE COMPARTIMIENTO .....	38
<i>Diagrama de Actividades</i> .....	38
<i>Diagrama de casos de uso</i> .....	39
<i>Diagrama de estados</i> .....	41
DIAGRAMAS DE INTERACCIÓN.....	42
<i>Diagrama de secuencia</i> .....	42
<i>Diagrama de colaboración</i> .....	42
<b>ESPECIFICACIONES DEL SISTEMA.....</b>	<b>43</b>
MÓDULO COMPRA.....	43

Ámbito del proyecto a elaborar.....	43
<b>CALENDARIZACIÓN DE TAREAS CON RESPECTO AL TIEMPO .....</b>	<b>45</b>
<b>ANÁLISIS DE RIESGO .....</b>	<b>46</b>
INTRODUCCIÓN.....	46
PROPÓSITO DEL PLAN DE RIESGO.....	47
VISIÓN GENERAL .....	47
Objetivos .....	47
Prioridades de aversión al riesgo .....	47
ORGANIZACIÓN.....	47
Gestión .....	47
Responsabilidades.....	48
<b>ANÁLISIS Y GESTIÓN DE RIESGOS.....</b>	<b>48</b>
IDENTIFICACIÓN .....	48
CLASIFICACIÓN (O TAXONOMÍA) DE LOS RIESGOS – FUENTES .....	49
DECLARACIÓN DE LOS RIESGOS .....	53
ESTIMACIÓN DEL IMPACTO.....	62
EXPOSICIÓN AL RIESGO.....	63
<b>GESTIÓN DE LOS RIESGOS.....</b>	<b>65</b>
LÍNEAS DE ACCIÓN .....	66
Riesgo RI-03: Adelantar fecha de entrega.....	66
Plan de Acción del Riesgo RI-03.....	67
Plan de Contingencia del Riesgo RI-03 .....	67
Riesgo RI-06 Errores en la definición de requisitos.....	67
Plan de Acción del Riesgo RI-06.....	68
Plan de Contingencia del Riesgo RI-06 .....	69
Riesgo: RI-11 Aparición de errores por la reutilización de software preexistente. ....	69
Plan de Acción del Riesgo RI-11.....	70
Plan de Contingencia del Riesgo RI-11 .....	70
Riesgo: RI-18 Dificultades con herramientas y lenguajes nuevos de programación. ....	71
Plan de Acción del Riesgo RI-18.....	72
Plan de Contingencia del Riesgo RI-18 .....	72
Riesgo: RI-20 Despido o renuncia o abandono de miembros del equipo de desarrollo.....	73
Plan de Acción del Riesgo RI-20.....	73
Plan de Contingencia del Riesgo RI-20 .....	74
Riesgo: RI-23 Pérdida de documentos y archivos del programa. ....	74
Plan de Acción del Riesgo RI-23.....	75

<i>Plan de Contingencia del Riesgo RI-23 .....</i>	<i>75</i>
<i>Riesgo: RI-27: Paro de actividades por Pandemia. ....</i>	<i>76</i>
<i>Plan de Acción del Riesgo RI-27.....</i>	<i>76</i>
<i>Plan de Contingencia del Riesgo RI-23 .....</i>	<i>77</i>
<b>DISEÑO DE INTERFACES .....</b>	<b>78</b>
LOGIN.....	78
PROVEEDORES.....	79
NUEVO PROVEEDOR .....	81
<b>PLAN DE CONFIGURACION.....</b>	<b>82</b>
CONTROL DE LA DOCUMENTACIÓN.....	82
<i>Control de la Configuración.....</i>	<i>82</i>
<i>Histórico de Versiones. ....</i>	<i>82</i>
<i>Histórico de Cambios. ....</i>	<i>82</i>
INTRODUCCIÓN.....	83
<i>Propósito .....</i>	<i>83</i>
<i>Alcance .....</i>	<i>83</i>
<i>Documentos Referenciados.....</i>	<i>84</i>
<i>Acrónimos.....</i>	<i>84</i>
<i>Definiciones .....</i>	<i>84</i>
<i>Organización .....</i>	<i>84</i>
CCB .....	85
<i>Presidente CCB .....</i>	<i>85</i>
<i>Administrador de la Herramienta de Gestión de la Configuración.....</i>	<i>85</i>
<i>Herramientas.....</i>	<i>86</i>
<i>Procedimientos.....</i>	<i>86</i>
<i>Identificar elementos de configuración .....</i>	<i>86</i>
<i>Controlar los elementos de configuración.....</i>	<i>90</i>
AUDITORÍAS DE LÍNEA BASE .....	91
<i>Las auditorías de líneas base:.....</i>	<i>91</i>
<b>NOMENCLATURA PARA LOS ARCHIVOS DEL SISTEMA .....</b>	<b>94</b>
<b>NOMENCLATURA PARA LOS CÓDIGOS Y BASE DE DATOS .....</b>	<b>95</b>
NOMBRES DE VARIABLES.....	95
NOMBRES DE CLASES .....	96
NOMBRES DE LOS ARCHIVOS.....	96
NOMBRES DE LAS CARPETAS .....	96
CORCHETES E INDENTACIÓN .....	96

PASOS PARA HACER FUNCIONAR EL SISTEMA .....97

## METODOLOGÍA DE DESARROLLO

La metodología de un proyecto consiste en todos los pasos que se recogen a la hora de la planificación y gestión de un proyecto. Esta suele recoger desde la gestión de recursos hasta la coordinación del equipo de trabajo, o incluso la relación con todos los interesados en los resultados del proyecto.

En este caso como grupo escogimos la metodología **Scrum**. Con esta metodología, el cliente se entusiasma y se compromete con el proyecto dado que lo ve crecer parte por parte. Asimismo, le permite en cualquier momento realinear el software con los objetivos de negocio de su empresa, ya que puede introducir cambios funcionales o de prioridad en el inicio de cada nueva iteración sin ningún problema.

## METODOLOGÍA SCRUM

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativa mente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

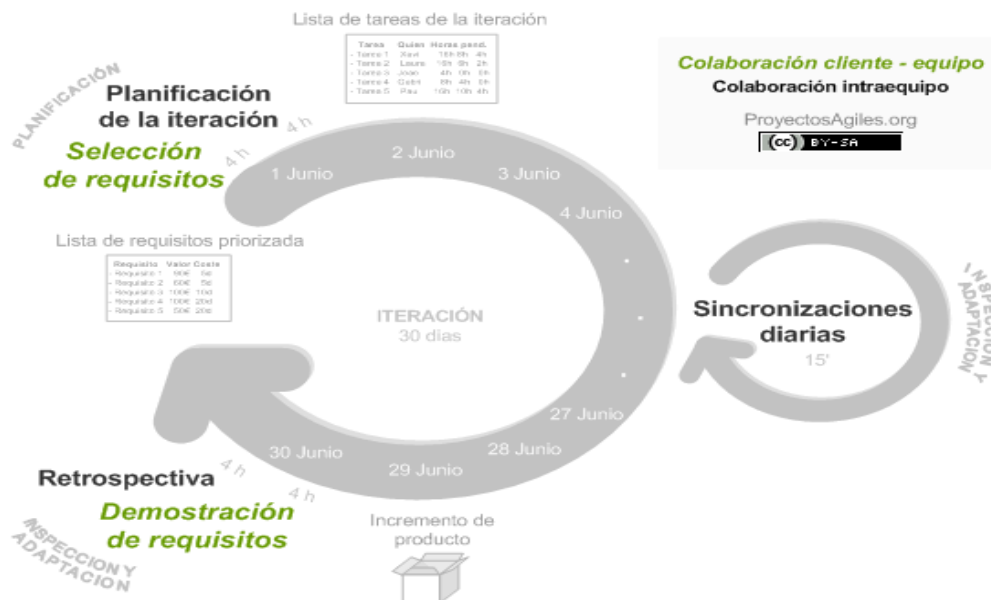
En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto.

### **El proceso**

En Scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de feedback de producto real y reflexión).

Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.



El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente (Product Owner) prioriza los objetivos balanceando el valor que le aportan respecto a su coste (que el equipo estima considerando la Definición de Hecho) y quedan repartidos en iteraciones y entregas.

Las actividades que se llevan a cabo en Scrum son las siguientes (*los tiempos indicados son para iteraciones de 2 semanas*):

## Planificación de la iteración

El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes: 9 Selección de requisitos (2 horas). El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que prevé que podrá completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.

Planificación de la iteración (2 horas). El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos seleccionados. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se autoasignan las tareas, se autoorganizan para trabajar incluso en parejas (o grupos mayores) con el fin de compartir conocimiento (creando un equipo más resiliente) o para resolver juntos objetivos especialmente complejos.

## Ejecución de la iteración

Cada día el equipo realiza una reunión de sincronización (15 minutos), normalmente delante de un tablero físico o pizarra (Scrum Taskboard). El equipo inspecciona el trabajo que el resto está



realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con la previsión de objetivos a mostrar al final de la iteración. En la reunión cada miembro del equipo responde a tres preguntas:

- ¿Qué he hecho desde la última reunión de sincronización para ayudar al equipo a cumplir su objetivo?
- ¿Qué voy a hacer a partir de este momento para ayudar al equipo a cumplir su objetivo?
- ¿Qué impedimentos tengo o voy a tener que nos impidan conseguir nuestro objetivo?

Durante la iteración el Facilitador (Scrum Master) se encarga de que el equipo pueda mantener el foco para cumplir con sus objetivos.

Elimina los obstáculos que el equipo no puede resolver por sí mismo.

Protege al equipo de interrupciones externas que puedan afectar el objetivo de la iteración o su productividad.

Durante la iteración, el cliente junto con el equipo refina la lista de requisitos (para prepararlos para las siguientes iteraciones) y, si es necesario, cambian o replanifican los objetivos del proyecto (10%-15% del tiempo de la iteración) con el objetivo de maximizar la utilidad de lo que se desarrolla y el retorno de inversión.

#### Inspección y adaptación

El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:

Revisión (demostración) (1,5 horas). El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, re planificando el proyecto.

Retrospectiva (1,5 horas). El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargará de eliminar o escalar los obstáculos identificados que estén más allá del ámbito de acción del equipo.

## PLANIFICACIÓN

### Problemática

Las dificultades a las que se enfrentan las PYMES del rubro de la carpintería son varias, principalmente porque, en el proceso de compra, venta o control de inventario, al hacerlo de manera

manual pueden surgir errores e incluso pérdidas económicas debido a que los procesos no son realizados de manera estructurada y controlada. Por este motivo, la creación de un sistema que automatice dicho problema será una de las mejores soluciones, específicamente en la parte de adquisiciones de parte de la empresa, evitando sobre compras o en su defecto, stock sobrante disponible.

## **OBJETIVOS**

### ***Objetivo general:***

- Desarrollar e implementar un módulo informático que soporte los diferentes procesos de gestión de compra para una carpintería.

### ***Objetivos específicos:***

- Recabar información sobre el proceso de gestión de compra en una carpintería.
- Determinar los de carpintería para el módulo de compras.

## **Limitaciones y Alcance del Tema.**

### ***Alcances:***

- Implementación del módulo compras
- Gestionar órdenes de compra
- Implementación del Submódulos Ingreso - Nuevo ingreso
- Implementación del Submódulos Proveedor - Nuevo proveedor
- Implementación del Submódulos Proveedor - Orden de compra
- Imprimir listado de todos los proveedores.
- Generar reporte cuenta a proveedor.

### ***Limitaciones:***

- No se contemplan más servicios de los ya mencionados anteriormente.
- No se contemplan el listado de todas las compras.

# **FACTIBILIDAD**

## **Técnica**

Debido a que se trata de un sistema y un módulo pequeño, pero sumamente importante el funcionamiento de estas empresas, su complejidad es relativamente baja y adquirir hardware costosos es innecesario, motivo por el cual un equipo pequeño común funcionarían adecuadamente con el sistema que pretendemos realizar. En cuanto a la necesidad de mano de obra, los integrantes del equipo cuentan con conocimientos suficientes para el desarrollo y cumplimiento del software.

## **Operativa**

Es viable operacional mente ya que podría ser utilizado fácilmente por la persona que esté a cargo de manipular el sistema (administrador-secretarios) con un mínimo de instrucciones que serán dadas por el mismo sistema o por una persona capacitada para su uso.

## **Económica**

En el estudio de la factibilidad económica, determinamos que es viable ya que el presupuesto de costos de los recursos técnicos, humanos y materiales tanto para el desarrollo como para la implantación del módulo dentro del sistema no son muy costosos. Con las computadoras existentes ya es posible manipular el sistema lo que lo hace más adquisitivo económicamente.

Dentro de nuestros recursos humanos contamos con un líder de proyecto (Analista), dos diseñadores de base de datos y 3 programadores. Cada uno de ellos cuenta con sus materiales, pero adquiriremos 2 Notebook Acer I3 53-33N7, con sistema operativo Windows 10 más con fines del desarrollo de este proyecto.

# **REQUERIMIENTOS MODELO DE NEGOCIOS**

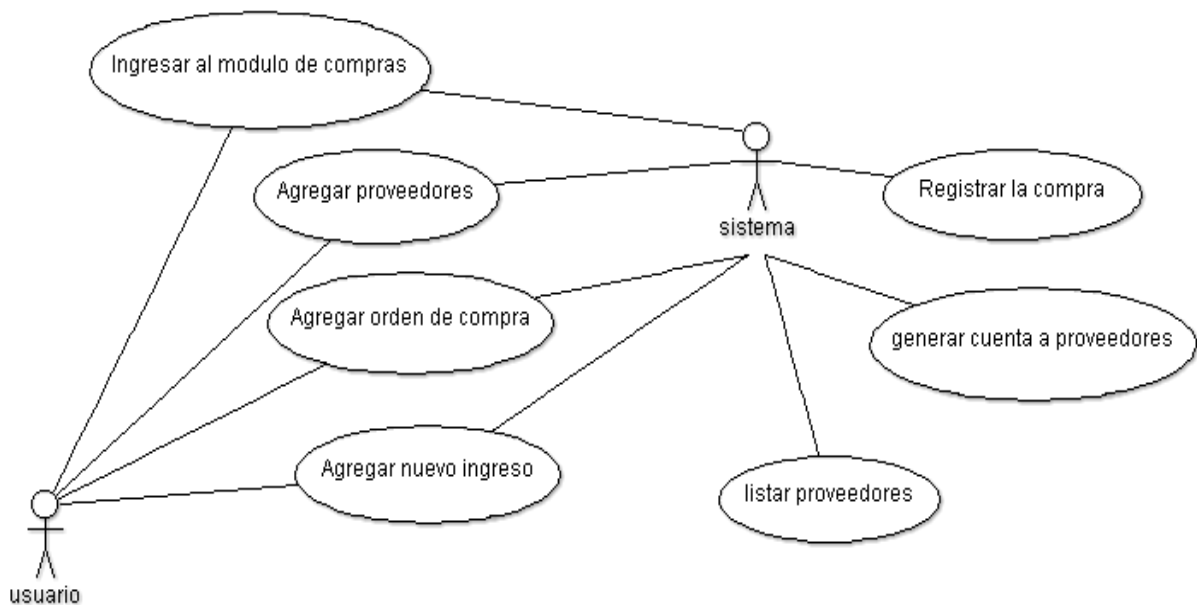
## **Identificar actividad del sistema:**

- Ingresar al módulo compras
- Submódulos Ingreso - Nuevo ingreso
- Submódulos Proveedor - Nuevo proveedor
- Submódulos Proveedor - Orden de compra

- Imprimir listado de todas las compras.
- Imprimir listado de proveedores.

El usuario podrá registrar los proveedores de la empresa y se encargara de realizar la orden de compra donde especificara las condiciones de pago, productos a comprar, fecha y lugar de Entrega, etc. que es remitida al proveedor. El Proveedor envía los productos y quien lo recepciona es el área de Almacén donde registra una nota de ingreso o de registro de compra. Cuando el usuario quisiera saber el listado de todas las compras realizadas con sus respectivos proveedores tendrá la opción de hacerlo de acuerdo al nombre del proveedor o fecha de compra en formato pdf.

### DIAGRAMA CASO DE USO - MODELADO DE NEGOCIO (CUMN)



### TABLA DEL CASO DE USO DEL MODELO DE NEGOCIO

<b>CU-1-MN</b>	<b>Modulo compras</b>
<b>Fecha</b>	<b>18/05/2020</b>
<b>Versión</b>	<b>1.0</b>
<b>Autor</b>	<b>Patricia Fernández</b>
<b>Actores</b>	<b>Usuario</b>
<b>Objetivos Asociados</b>	Ingresar al módulo compras para gestionar proveedores, órdenes de compra y compras.

<b>Descripción</b>	Se presenta este caso de uso cuando un usuario desea ingresar al sistema para gestionar compras y ordenes de compras.	
<b>Precondición</b>	1. Usuario ingresa al módulo compras	
<b>Flujo Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario ingresa al módulo compras
	2	El usuario gestiona proveedores
	3	El usuario gestiona órdenes de compra
	4	El usuario agrega la nueva compra al sistema
	5	El sistema registra la compra
	6	El sistema genera cuenta a proveedores
	7	El sistema lista todos los proveedores
Flujo alternativo	<b>Paso</b>	<b>Acción</b>
	2,3,4	El sistema valida los datos, si los datos no son correctos el sistema emite un mensaje de denegación para que el personal corrobore los datos.
	2,3,4	El sistema valida los datos, si los datos son correctos emite un mensaje de aprobación.
	5	El sistema registra la compra y emite un mensaje de aprobación.
Post condición	6	El usuario gestiona una nueva compra.
		Las pistas de auditoria se ven registradas por cada acción.
<b>Rendimiento</b>	Paso	Acción
	5	2 segundos
<b>Importancia</b>	Alta	
<b>Comentarios</b>		

## DIAGRAMA DE CLASES(CUMN)

Un **diagrama de clases** es una herramienta para comunicar el diseño de un programa que se creó para orientar objetos y que permite modelar relaciones entre diferentes entidades.

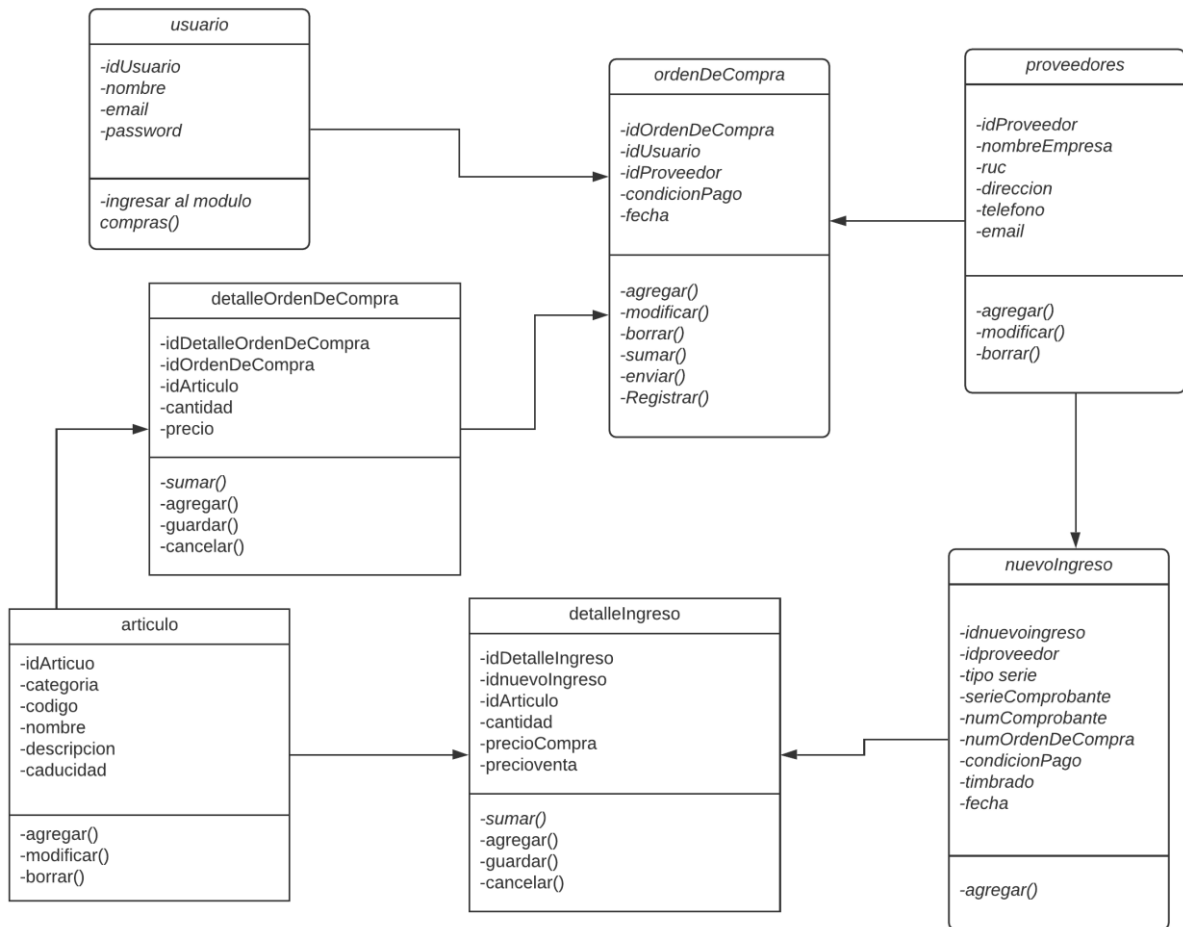
En UML, una clase representa un objeto o un conjunto de objetos que comparte una estructura y un comportamiento comunes. Se representan con un rectángulo que incluye filas del nombre de la clase, sus atributos y sus operaciones.

**Nombre:** La primera fila en una figura de clase.

**Atributos:** La segunda fila en una figura de clase. Cada atributo de una clase está ubicado en una línea separada. Si es abstracta, se escribe en letra cursiva o también se utiliza un estereotipo como < > arriba del nombre de la clase.

**Métodos:** La tercera fila en una figura de clase. También conocidos como "operaciones", es decir, los mensajes que puede entender. Los métodos se organizan en un formato de lista donde cada operación posee su propia línea.

## Diagrama de clases

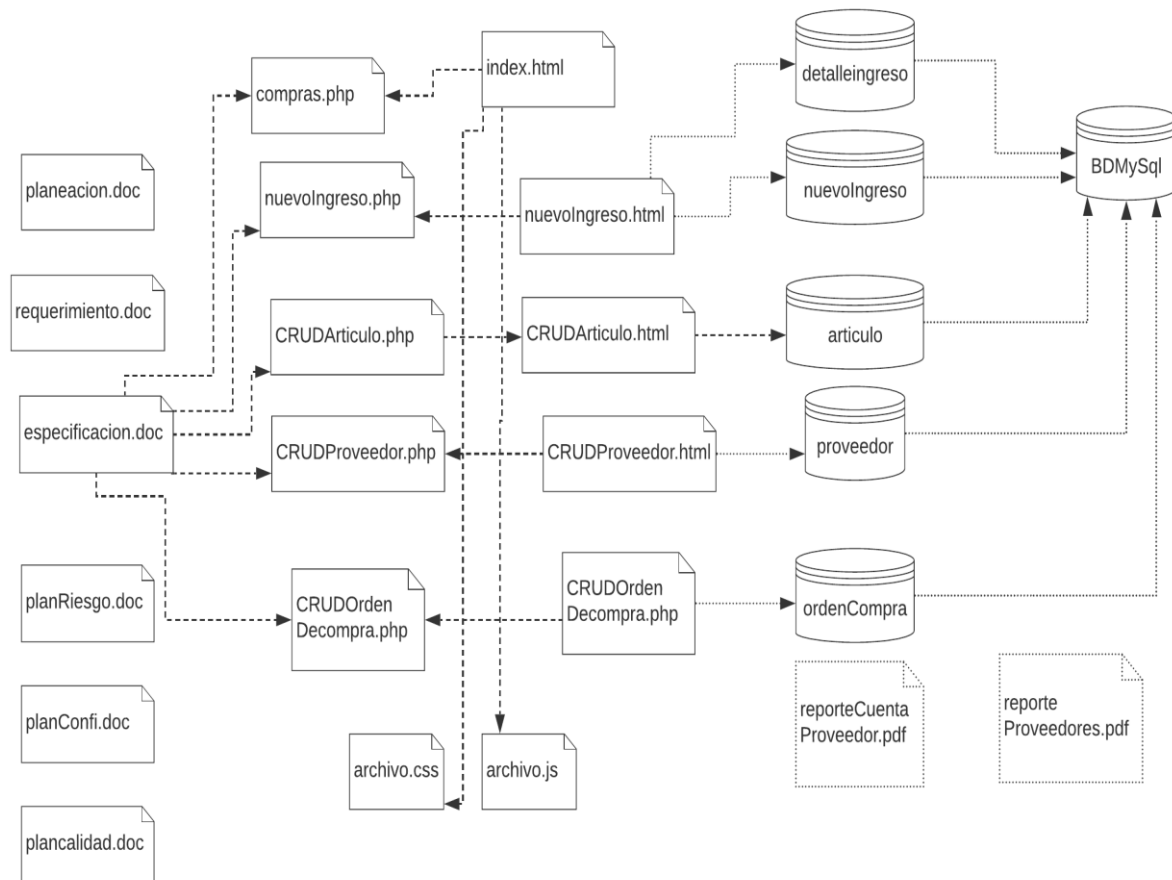


## DIAGRAMA DE COMPONENTES(CUMN)

Un diagrama de componentes es un diagrama tipo del Lenguaje Unificado de Modelado.

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software, pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.

## DIAGRAMA DE COMPONENTES



## DIAGRAMA DE OBJETOS(CUMN)

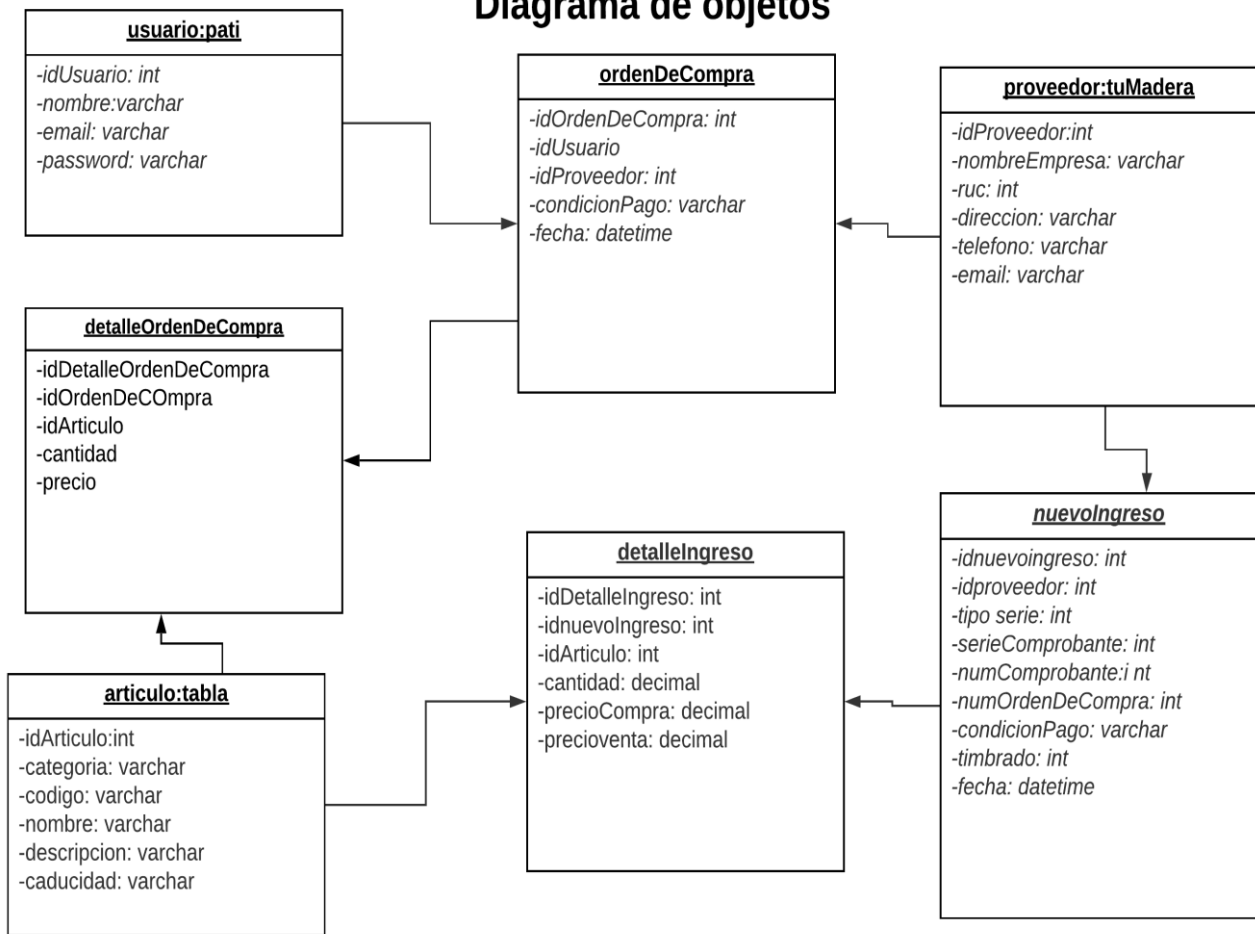
Los diagramas de objetos modelan las instancias de elementos contenidos en los diagramas de clases. Un diagrama de objetos muestra un conjunto de objetos y sus relaciones en un momento concreto.

En general los diagramas de objetos se utilizan para modelar estructuras de objetos, lo que implica tomar una instantánea de los objetos de un sistema en un cierto momento. Un diagrama de objetos representa una escena estática dentro de la historia representada por un diagrama de interacción.

Los diagramas de objetos se utilizan para visualizar, especificar, construir y documentar la existencia de ciertas instancias en el sistema, junto a las relaciones entre ellas.



## Diagrama de objetos

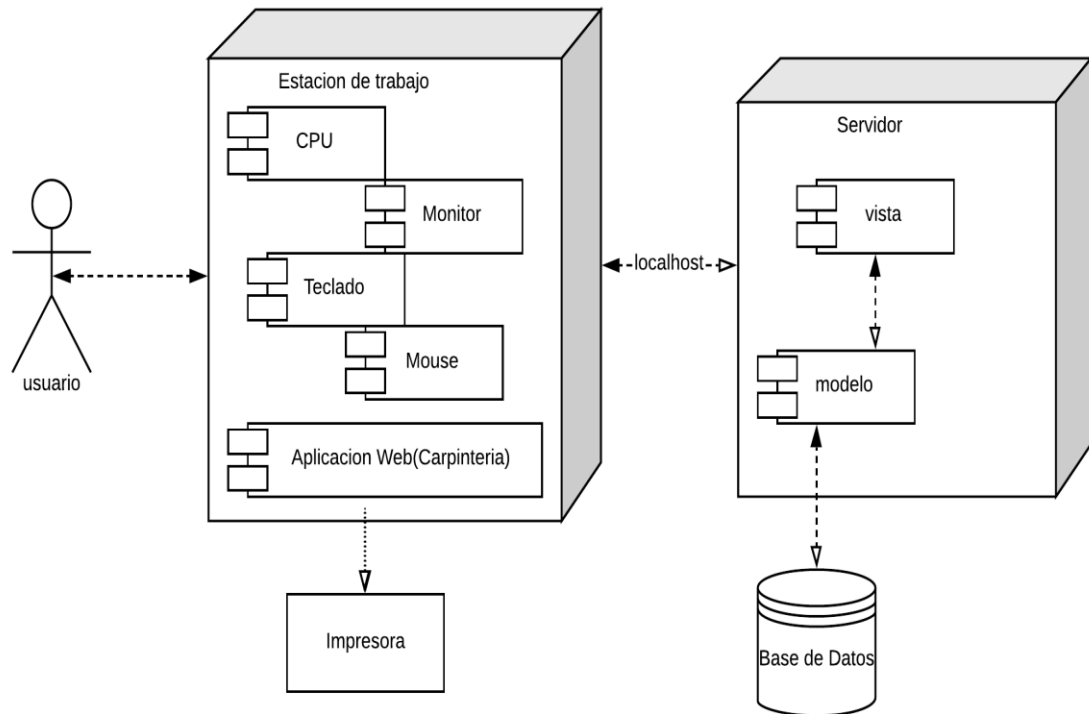


## DIAGRAMA DE DESPLIEGUE(CUMN)

Diagrama de Despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos.

Los diagramas de despliegue son los complementos de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema. Describen la topología del sistema, la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. Los diagramas de despliegue representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP.

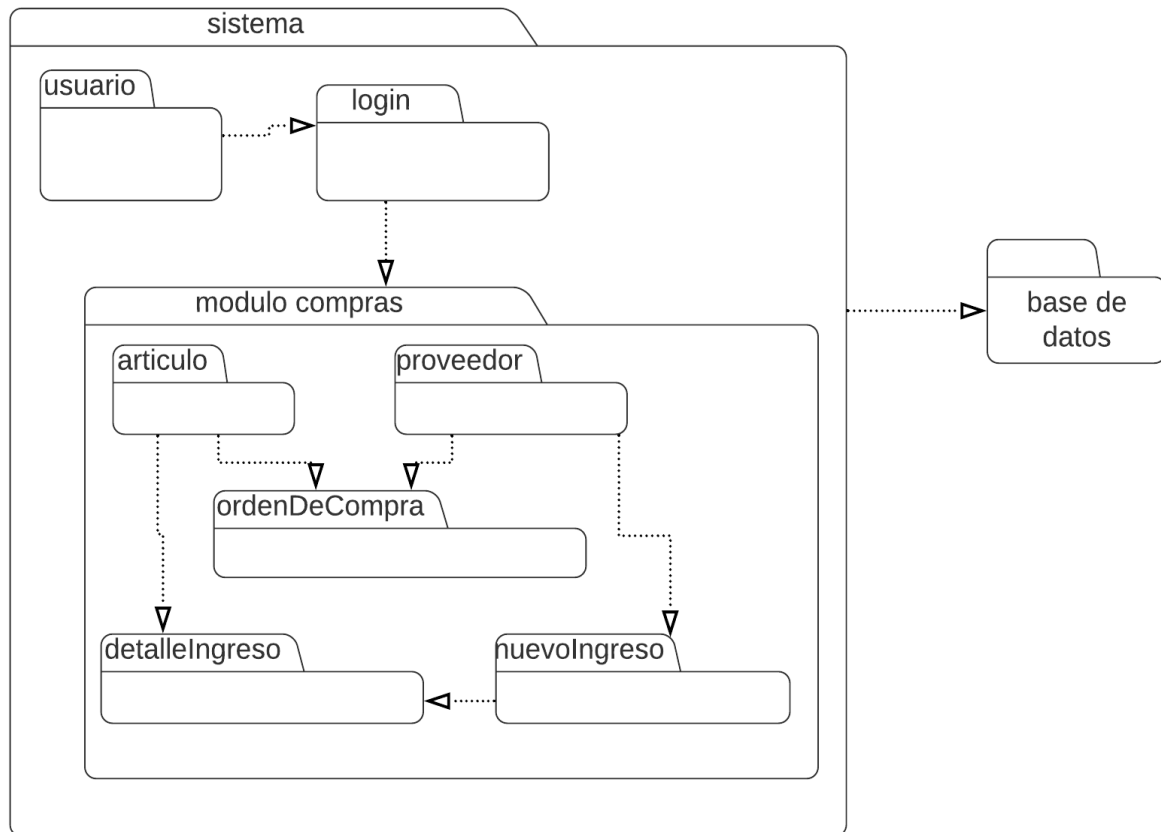
## Diagrama de despliegue



## DIAGRAMA DE PAQUETES(CUMN)

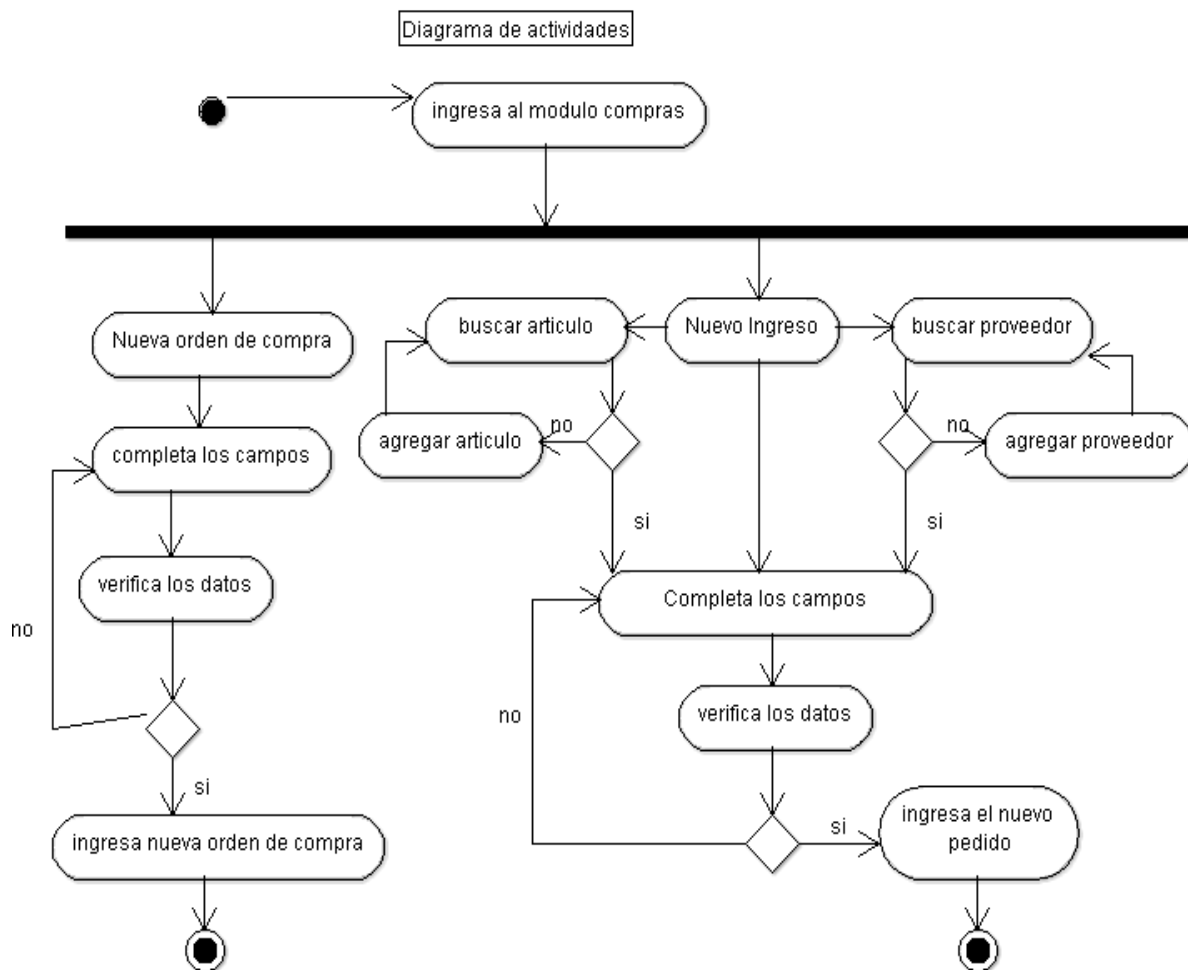
Muestra cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema. Cada paquete puede asignarse a un individuo o a un equipo, y las dependencias entre ellos pueden indicar el orden de desarrollo requerido.

# Diagrama de paquetes



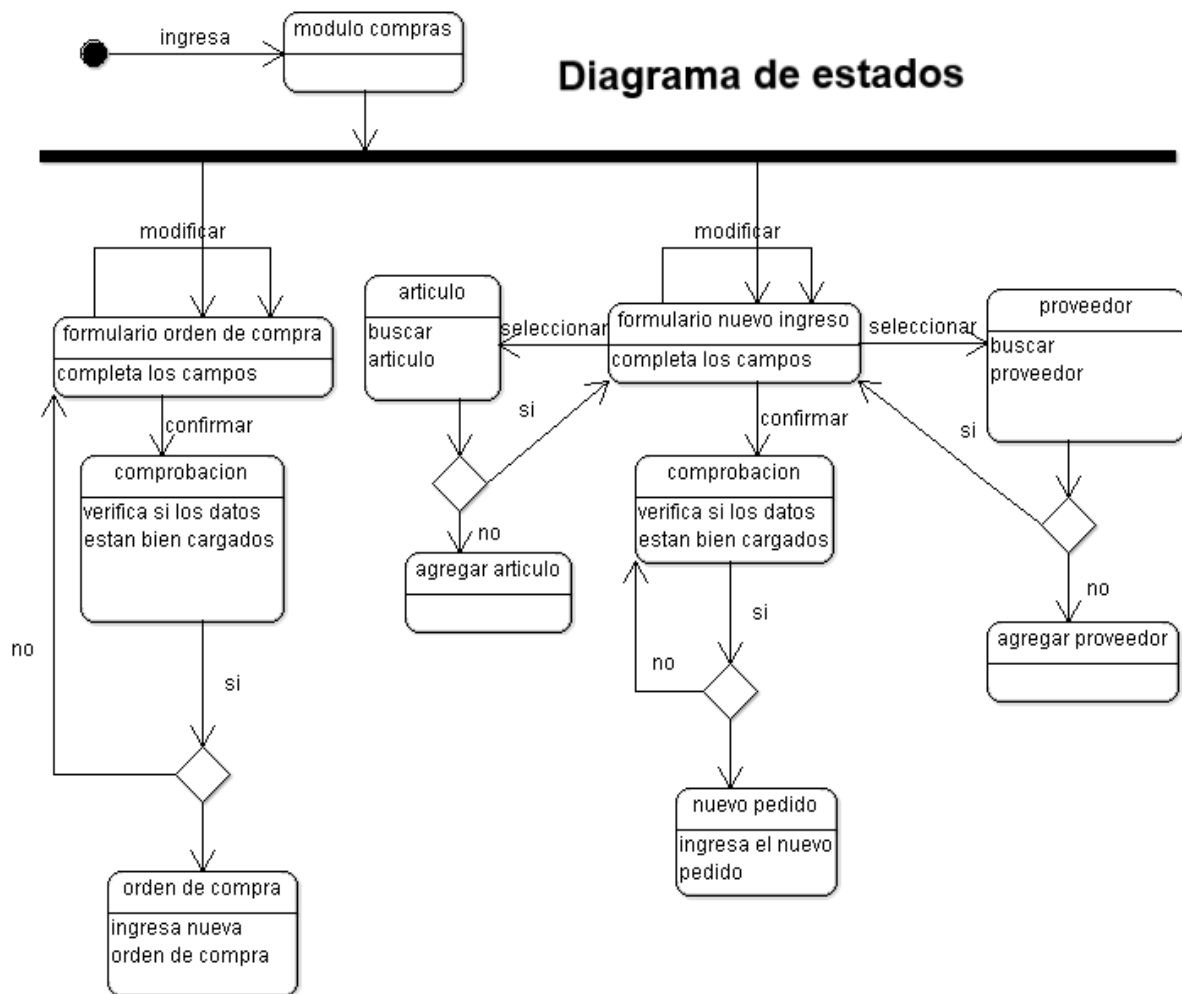
## DIAGRAMA DE ACTIVIDADES(CUMN)

Un diagrama de actividades muestra el flujo de actividades, siendo una actividad una ejecución general entre los objetos que se está ejecutando en un momento dado dentro de una máquina de estados, el resultado de una actividad es una acción que producen un cambio en el estado del sistema o la devolución de un valor. Las acciones incluyen llamadas a otras operaciones, envío de señales, creación o destrucción de objetos o simples cálculos. Gráficamente un diagrama de actividades será un conjunto de arcos y nodos. Desde un punto de vista conceptual, el diagrama de actividades muestra cómo fluye el control de unas clases a otras con la finalidad de culminar con un flujo de control total que se corresponde con la consecución de un proceso más complejo. Por este motivo, en un diagrama de actividades aparecerán acciones y actividades correspondientes a distintas clases. Colaborando todas ellas para conseguir un mismo fin. Ejemplo: Hacer un pedido.



## DIAGRAMA DE ESTADOS (CUMN)

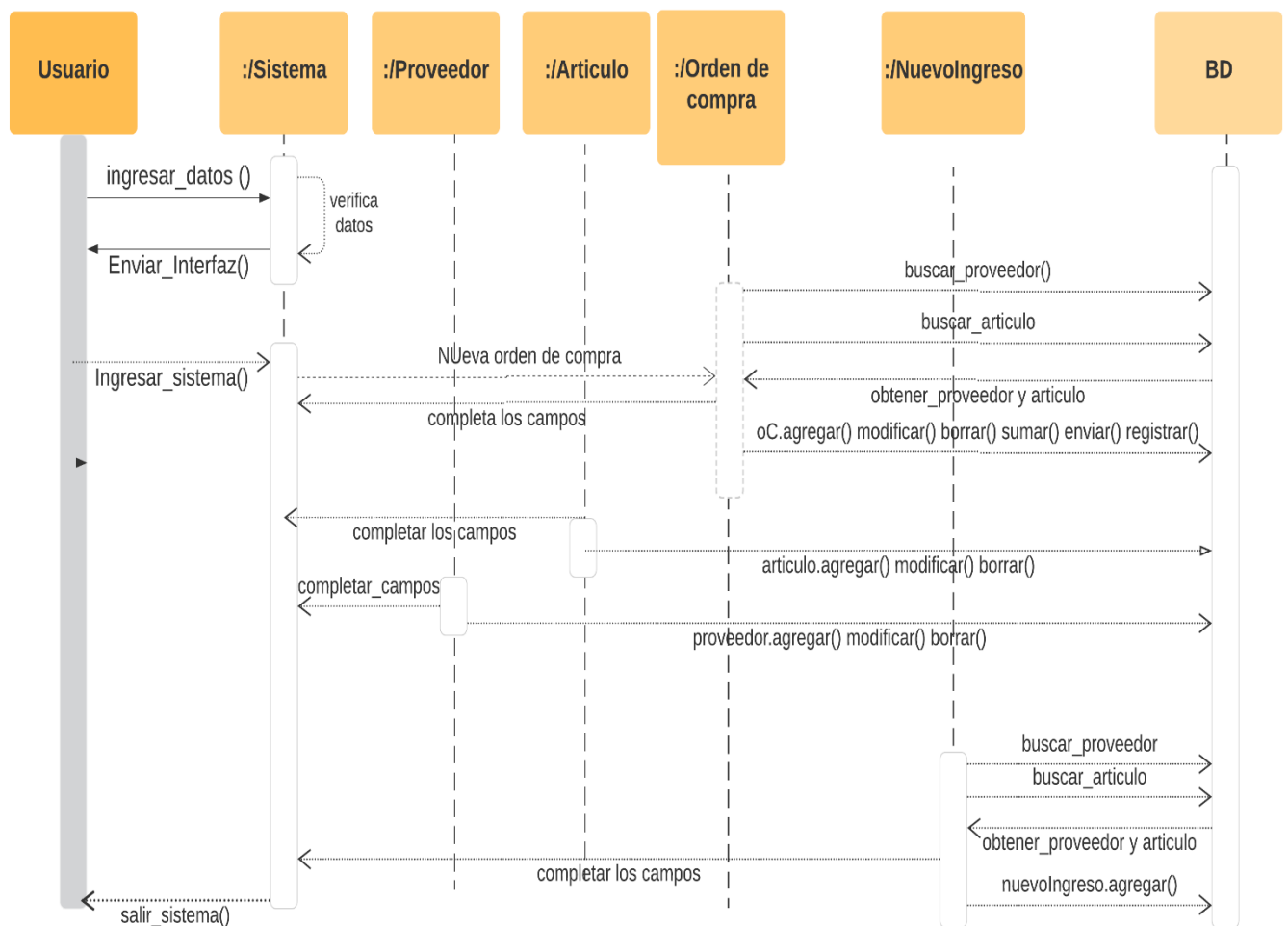
Una máquina de estados es cualquier dispositivo que almacena el estado de un objeto en un momento dado y puede cambiar el estado o causar otras acciones según la entrada que reciba. Estados se refiere a las diferentes combinaciones de información que un objeto puede mantener, no la forma en que el objeto se comporta. Para comprender los diferentes estados de un objeto, podrías visualizar todos los estados posibles y mostrar cómo un objeto llega a cada estado, y puedes hacerlo con un diagrama de estados UML.



## DIAGRAMA DE SECUENCIA (CUMN)

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Mientras que el Diagrama de casos de uso permite el modelado de una vista *business* del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos.

### Diagrama de secuencia



## DIAGRAMA DE COLABORACION (CUMN)

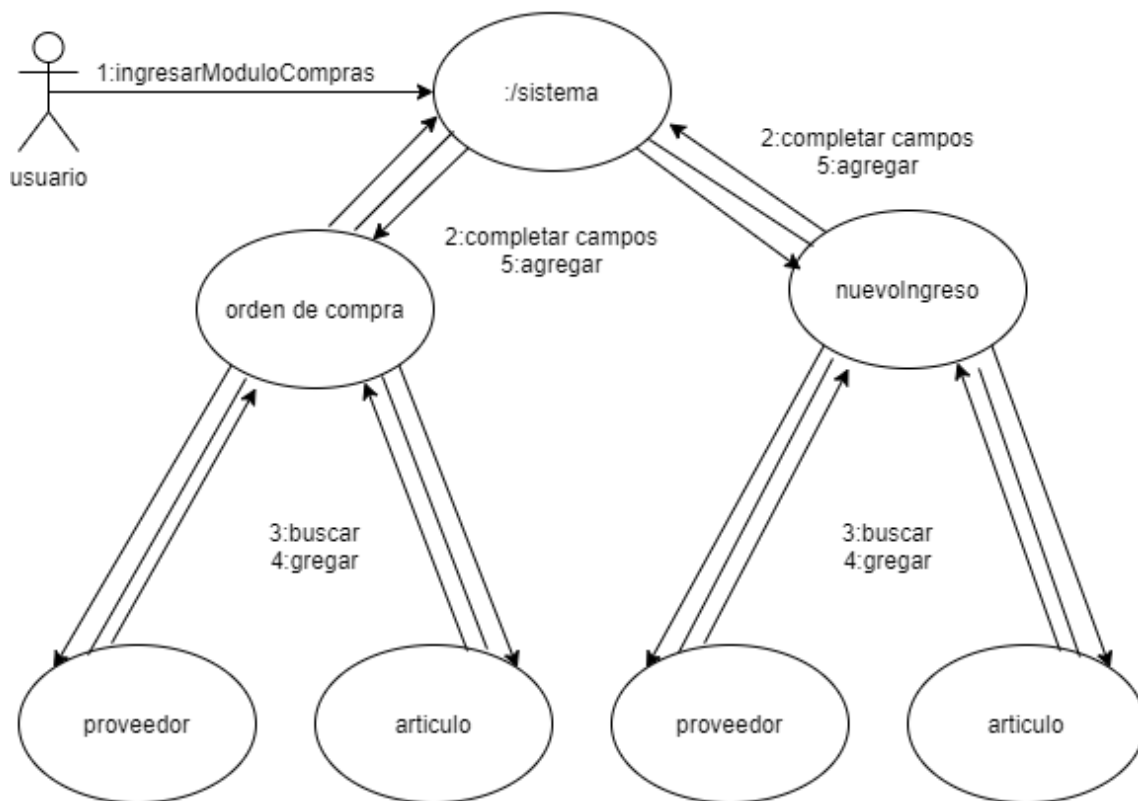
Un diagrama de colaboración en las versiones de UML 1.x es esencialmente un diagrama que muestra interacciones organizadas alrededor de los roles. A diferencia de los diagramas de secuencia, los diagramas de colaboración, también llamados diagramas de comunicación, muestran explícitamente las relaciones de los roles. Por otra parte, un diagrama de comunicación no muestra

el tiempo como una dimensión aparte, por lo que resulta necesario etiquetar con números de secuencia tanto la secuencia de mensajes como los hilos concurrentes.

Muestra cómo las instancias específicas de las clases trabajan juntas para conseguir un objetivo común.

Implementa las asociaciones del diagrama de clases mediante el paso de mensajes de un objeto a otro. Dicha implementación es llamada "enlace".

### Diagrama de colaboración



## REQUERIMIENTOS MODELO DE REQUISITOS

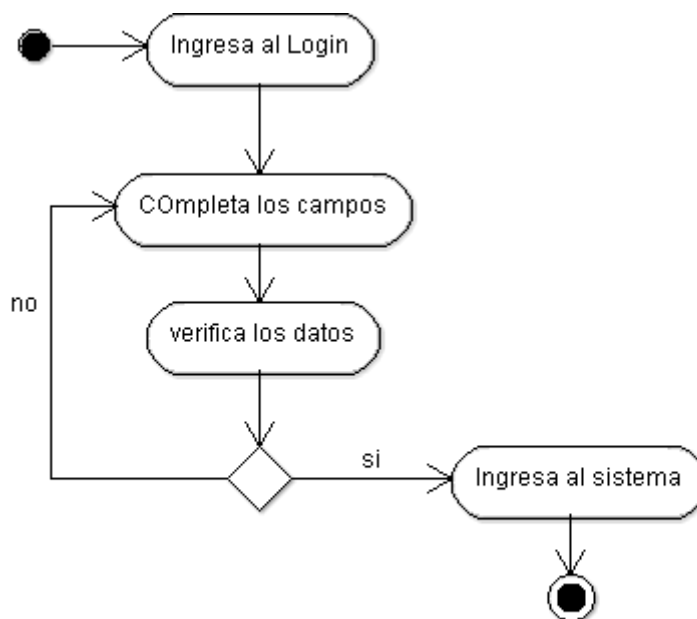
- Ingresar al módulo compras
- Submódulos Ingreso - Nuevo ingreso
- Submódulos Proveedor - Nuevo proveedor
- Submódulos Proveedor - Orden de compra
- Imprimir listado de todas las compras.
- Imprimir proveedores.

### IDENTIFICAR LOS CU A REALIZAR:

- Ingresar al módulo compras
- Nuevo ingreso
- Nuevo proveedor
- Realizar orden de compra

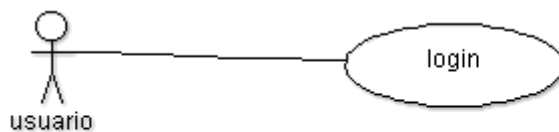
## DIAGRAMAS DEL SUB MODULO INGRESAR AL MÓDULO COMPRAS

### Diagrama de Actividades





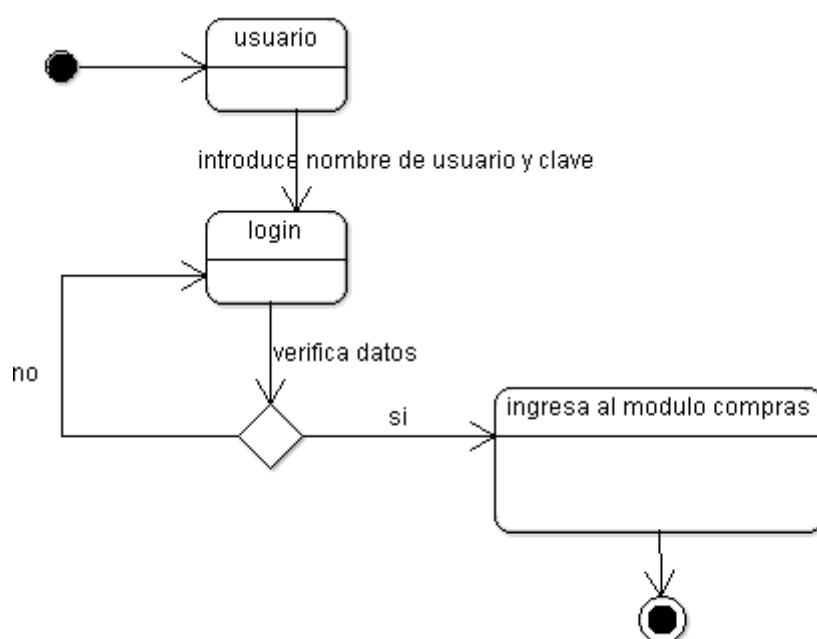
## Diagrama de casos de uso



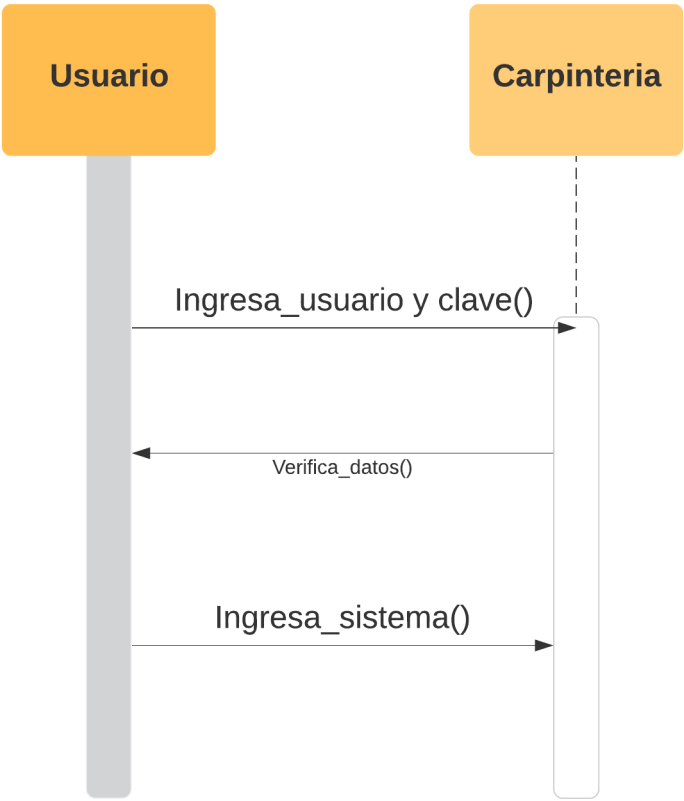
<b>CU-1</b>	<b>Ingresar al módulo compras</b>	
<b>Fecha</b>	<b>08/05/2020</b>	
<b>Versión</b>	<b>1.0</b>	
<b>Autor</b>	<b>Patricia Fernández</b>	
<b>Actores</b>	<b>Usuario</b>	
<b>Objetivos Asociados</b>	Ingresar al sistema para realizar las compras	
<b>Descripción</b>	Se presenta este caso de uso cuando un usuario desea ingresar al sistema	
<b>Precondición</b>	1. Usuario posee clave y usuario valido para ingresar al sistema	
<b>Flujo Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario no registrado accede al sitio de login
	1	El usuario introduce su nombre de usuario y clave, presiona aceptar.
	2	El sistema recepciona los datos y valida.
<b>Flujo alternativo</b>	3	El sistema redirecciona al módulo de compras del sistema
	<b>Paso</b>	<b>Acción</b>
	1	El sistema advierte que los campos deber ser obligatorios.
	2	El sistema emite un mensaje que dice “Usuario y/o Clave invalida. Comuniquese con el administrador del sistema”

	3	El sistema emite un mensaje de error indicando que debe Comunicar al administrador del sistema.
Post condición		El usuario registrado se convierte en un usuario valido
		Las pistas de auditoria se ven registradas por cada acción.
<b>Rendimiento</b>	Paso	Acción
	3	2 segundos
<b>Frecuencia esperada</b>	Un usuario puede intentar ingresar al sistema tres veces	
<b>Importancia</b>	alta	
<b>Comentarios</b>		

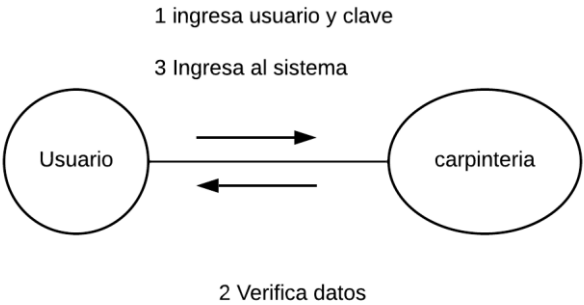
### Diagrama de estados



**Diagrama de secuencia**



**Diagrama de colaboración**



## DIAGRAMAS DEL SUB MODULO NUEVO INGRESO

### Diagramas de Compartimiento

#### *Diagrama de Actividades*

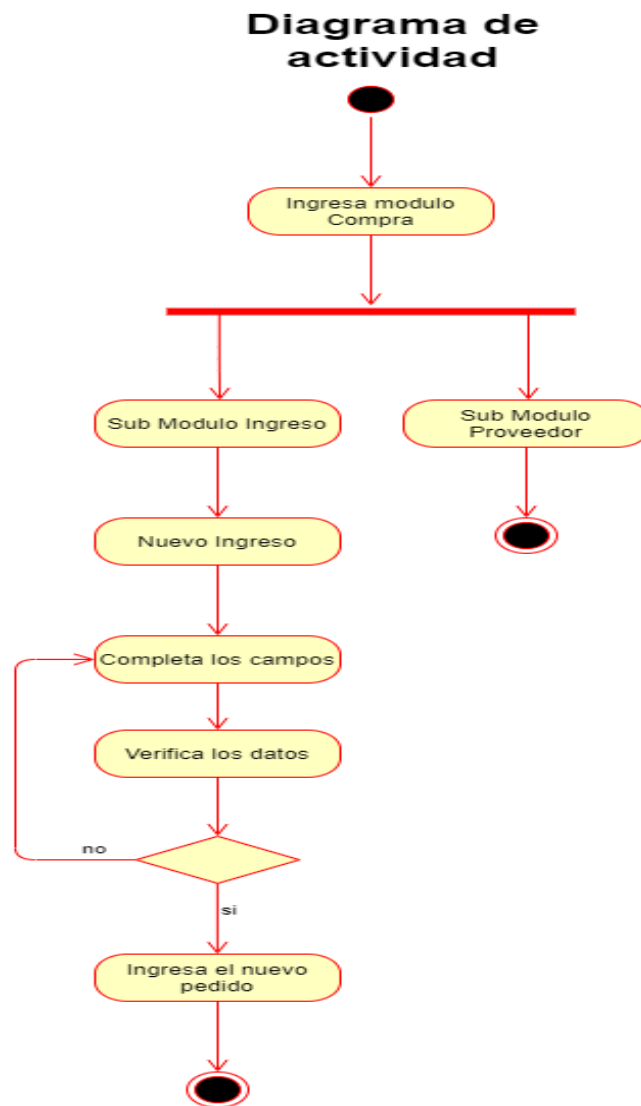
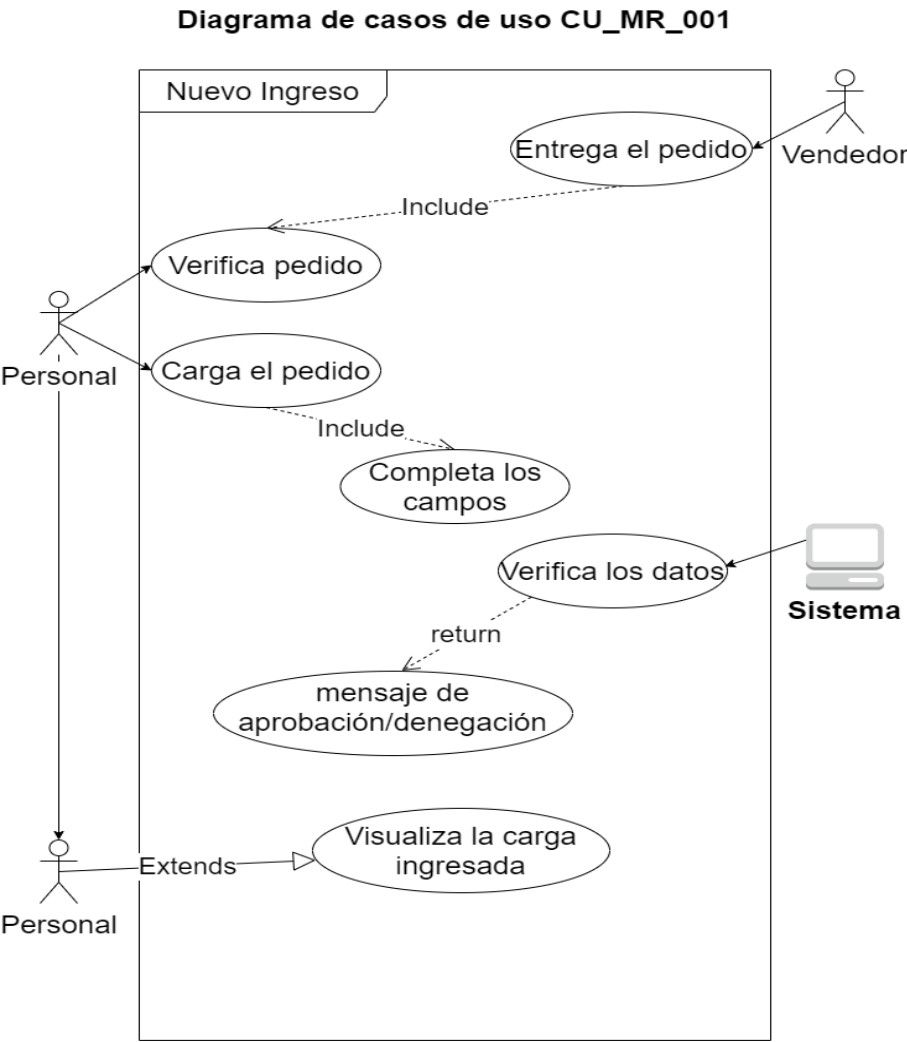
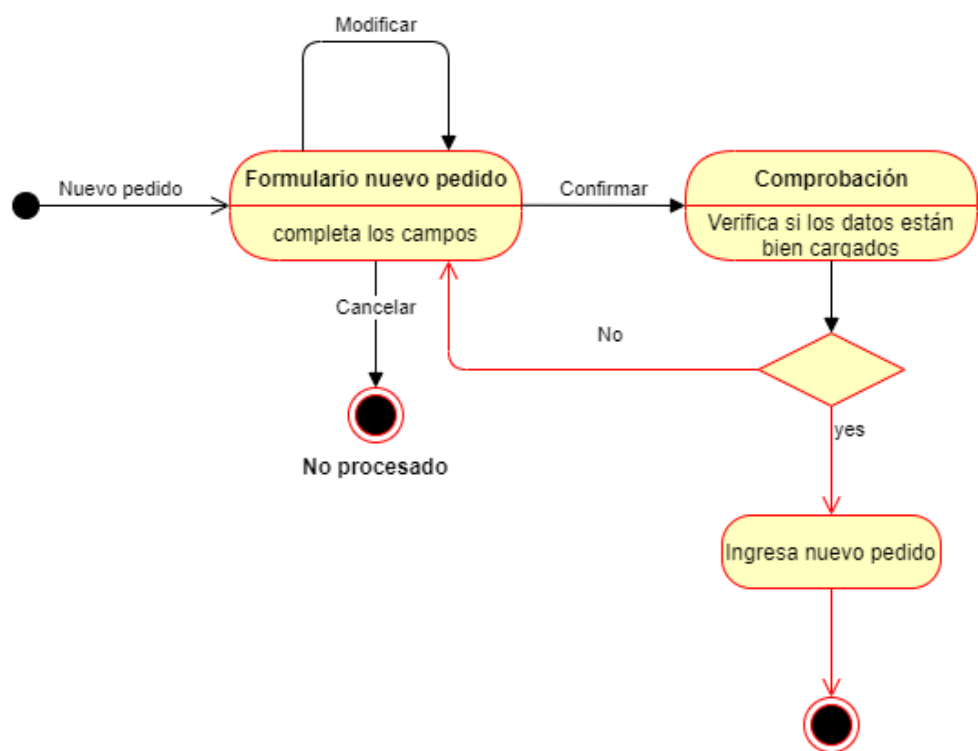


Diagrama de casos de uso



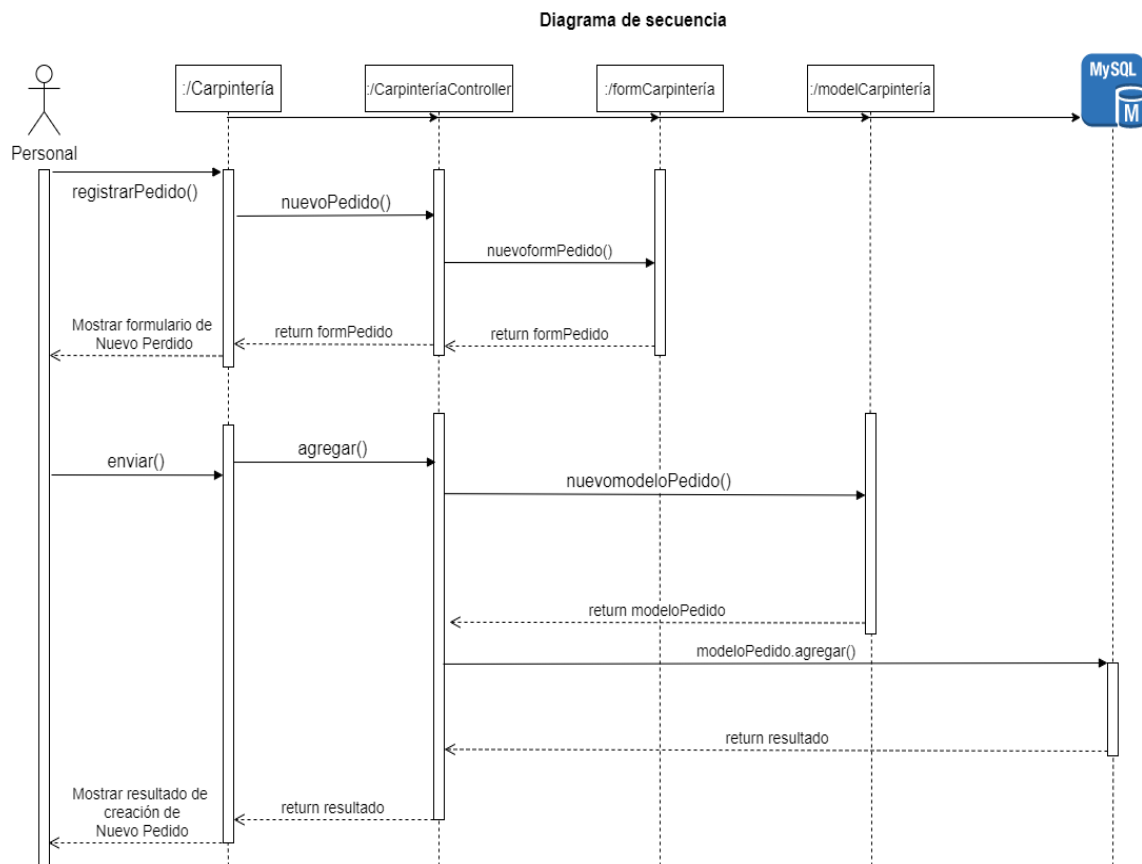
<b>CU_MR_001</b>	<b>Nuevo Ingreso</b>	
<b>Fecha</b>	<b>16/05/20</b>	
<b>Versión</b>	<b>1.0</b>	
<b>Autor</b>	<b>Alejandro Villalba</b>	
<b>Actores</b>	<b>Personal (usuario)</b>	
<b>Objetivos Asociados</b>	Ingresar al sub modulo Nuevo ingreso para agregar un pedido.	
<b>Descripción</b>	Se presenta este caso de uso cuando el personal desea ingresar un nuevo pedido de compra de algún artículo.	
<b>Precondición</b>	1. Personal debe de contener la factura legal del pedido.	
<b>Flujo Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El personal recibe la mercadería del proveedor
	2	El personal verifica la mercadería
	3	El personal ingresa al sub modulo y procede a la carga de la factura
	4	El personal rellena los campos y envía el formulario cargado
Flujo alternativo	<b>Paso</b>	<b>Acción</b>
	4.1	El sistema advierte que los campos deber ser obligatorios.
	4.2	El sistema validad los datos, si los datos no son correctos el sistema emite un mensaje de denegación para que el personal corrobore los datos.
	4.3	El sistema validad los datos, si los datos son correctos emite un mensaje de aprobación.
Post condición	1	El sistema registra el nuevo pedido
	2	Las pistas de auditoria se ven registradas por cada acción.
<b>Importancia</b>	alta	
<b>Comentarios</b>		

# Diagrama de Estado



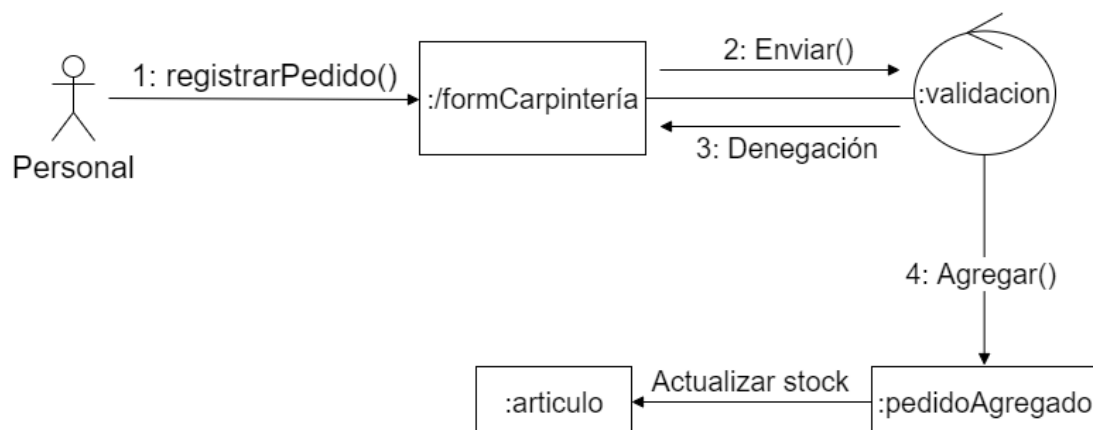
## Diagramas de Interacción

### Diagrama de secuencia



### Diagrama de colaboración

#### Diagrama de Colaboración





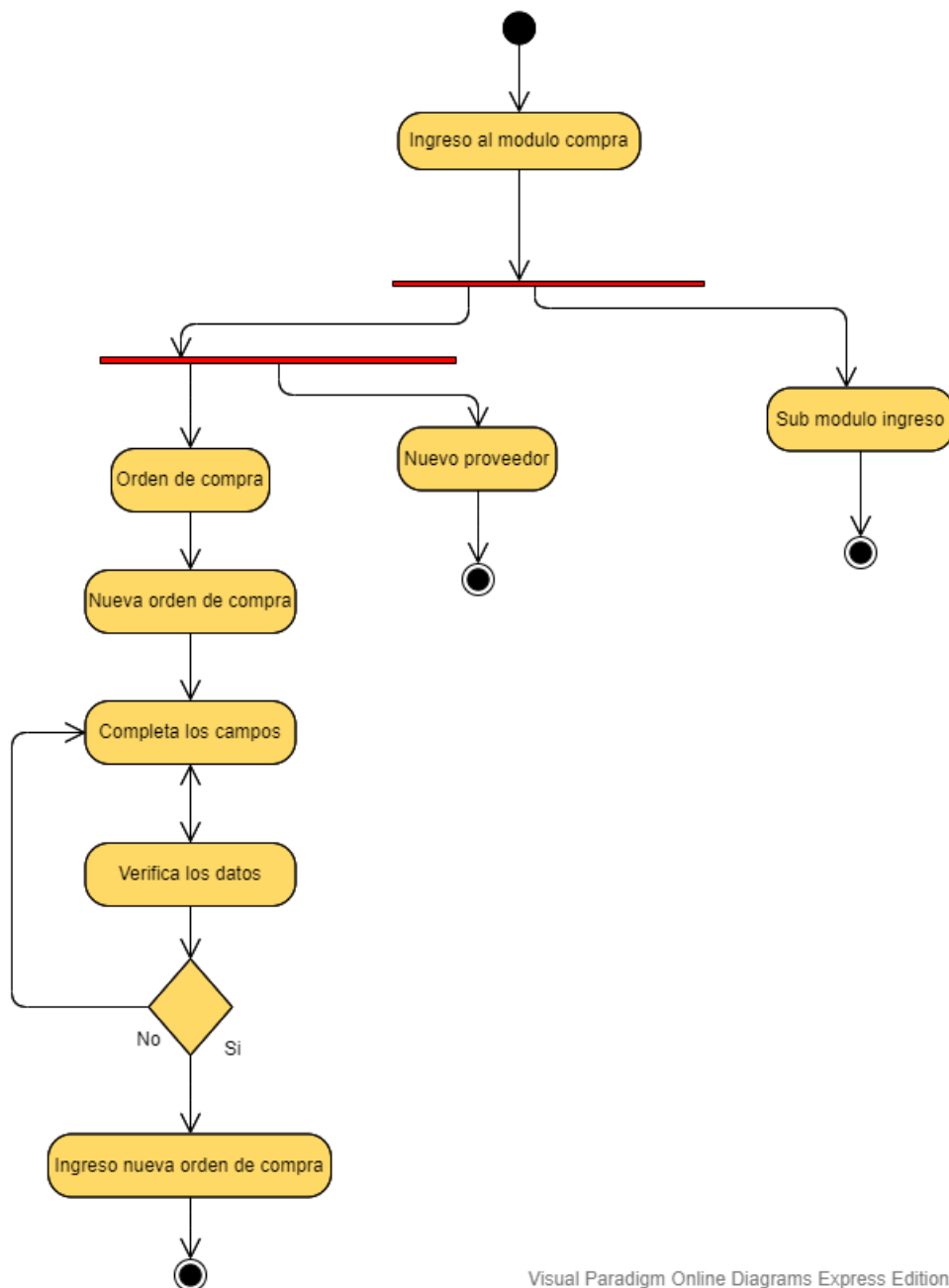
## DIAGRAMAS DEL SUB MODULO ORDEN DE COMPRA

### Diagramas de Compartimiento

#### Diagrama de Actividades

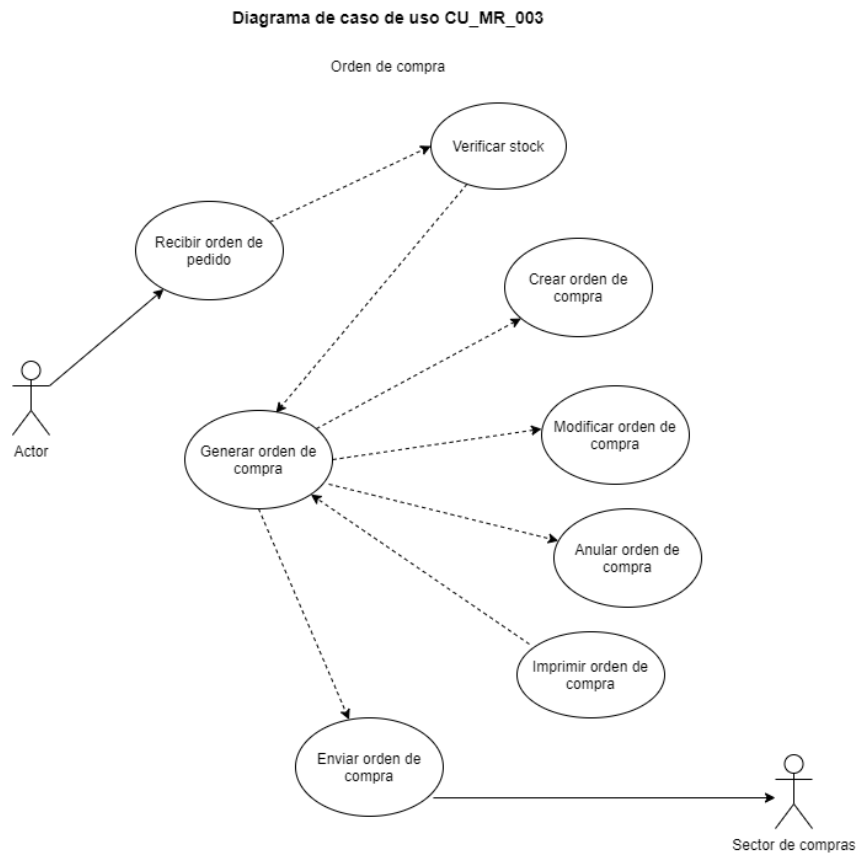
Visual Paradigm Online Diagrams Express Edition

Diagrama de  
Actividades



Visual Paradigm Online Diagrams Express Edition

## Diagrama de casos de uso

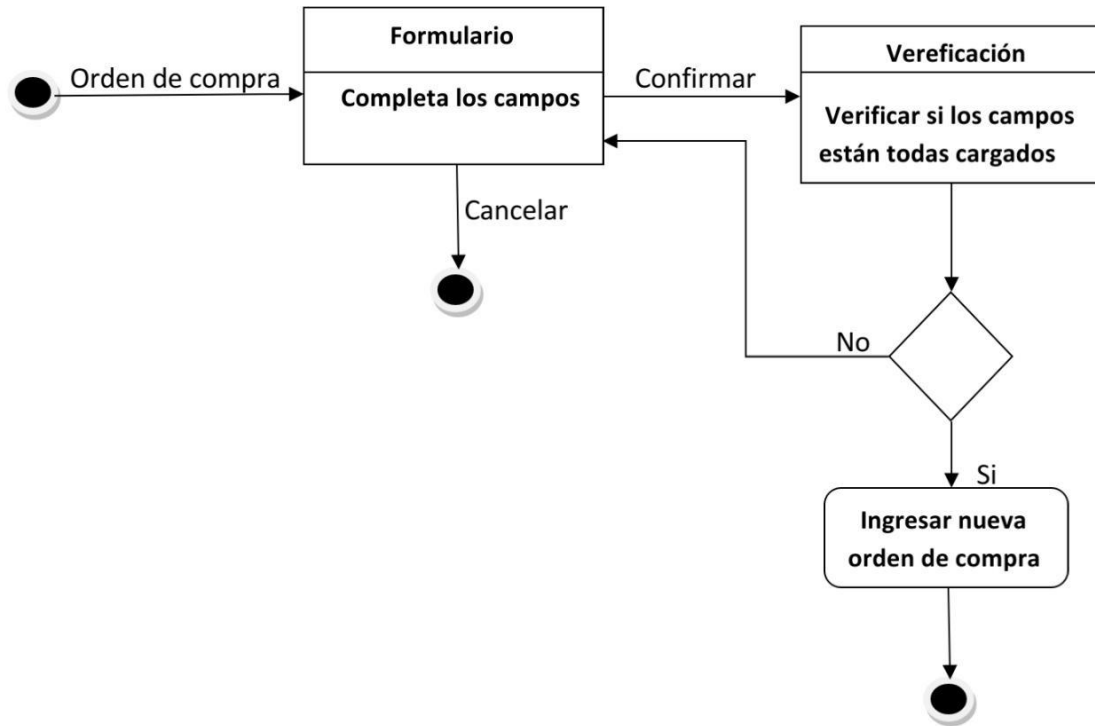


<b>CU_MR_003</b>	<b>Nueva orden de compra</b>	
<b>Fecha</b>	<b>16/05/20</b>	
<b>Versión</b>	<b>1.0</b>	
<b>Autor</b>	<b>Francisco Almada</b>	
<b>Actores</b>	<b>Personal (usuario)</b>	
<b>Objetivos Asociados</b>	Ingresar al sub modulo Orden de compra para agregar una nueva compra.	
<b>Descripción</b>	Se presenta este caso de uso cuando el personal desea ingresar una nueva orden de compra.	
<b>Precondición</b>	1. 1. Personal debe de agregar orden de compra e imprimir el reporte	
<b>Flujo Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El personal recibe la orden de compra

	2	El personal verifica la orden
	3	El personal ingresa al sub modulo y procede a la carga de la factura
	4	El personal rellena los campos y envía el formulario cargado
Flujo alternativo	<b>Paso</b>	<b>Acción</b>
	4.1	El sistema advierte que los campos deber ser obligatorios.
	4.2	El sistema validad los datos, si los datos no son correctos el sistema emite un mensaje de denegación para que el personal corrobore los datos.
	4.3	El sistema validad los datos, si los datos son correctos emite un mensaje de aprobación.
Post condición	1	El sistema registra la nueva orden de compra
	2	El sistema puede imprimir un reporte de sus compras o también lo descarga en un formato .pdf.
<b>Importancia</b>	alta	
<b>Comentarios</b>		

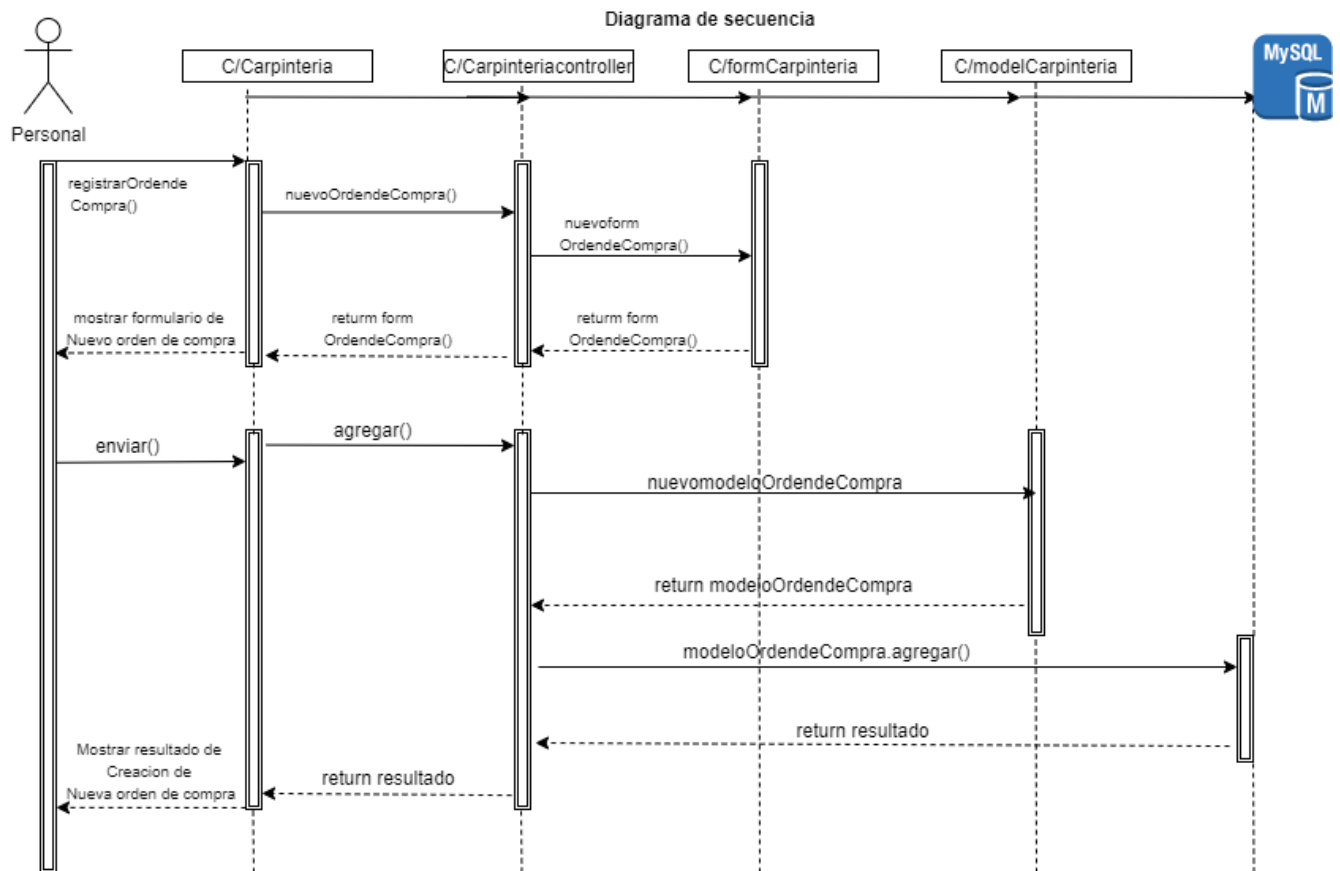
## Diagrama de estados

### 3. Diagrama de estados



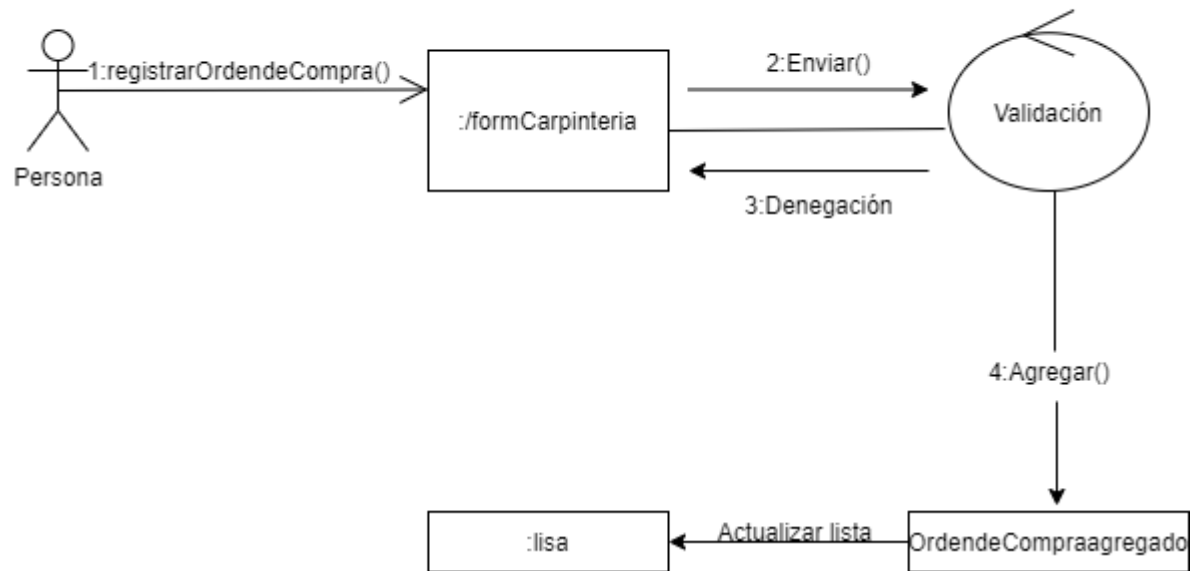
## Diagramas de Interacción

### Diagrama de secuencia



## Diagrama de colaboración

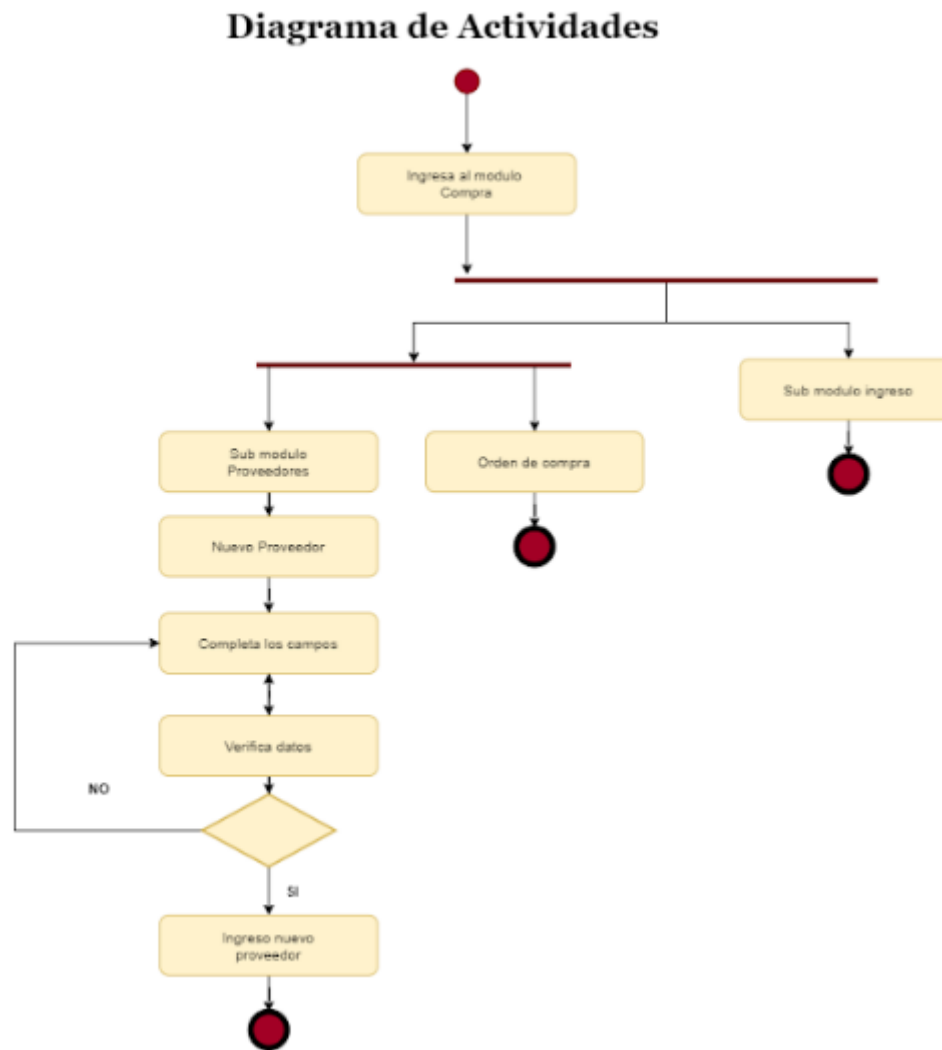
### Diagrama de colaboración



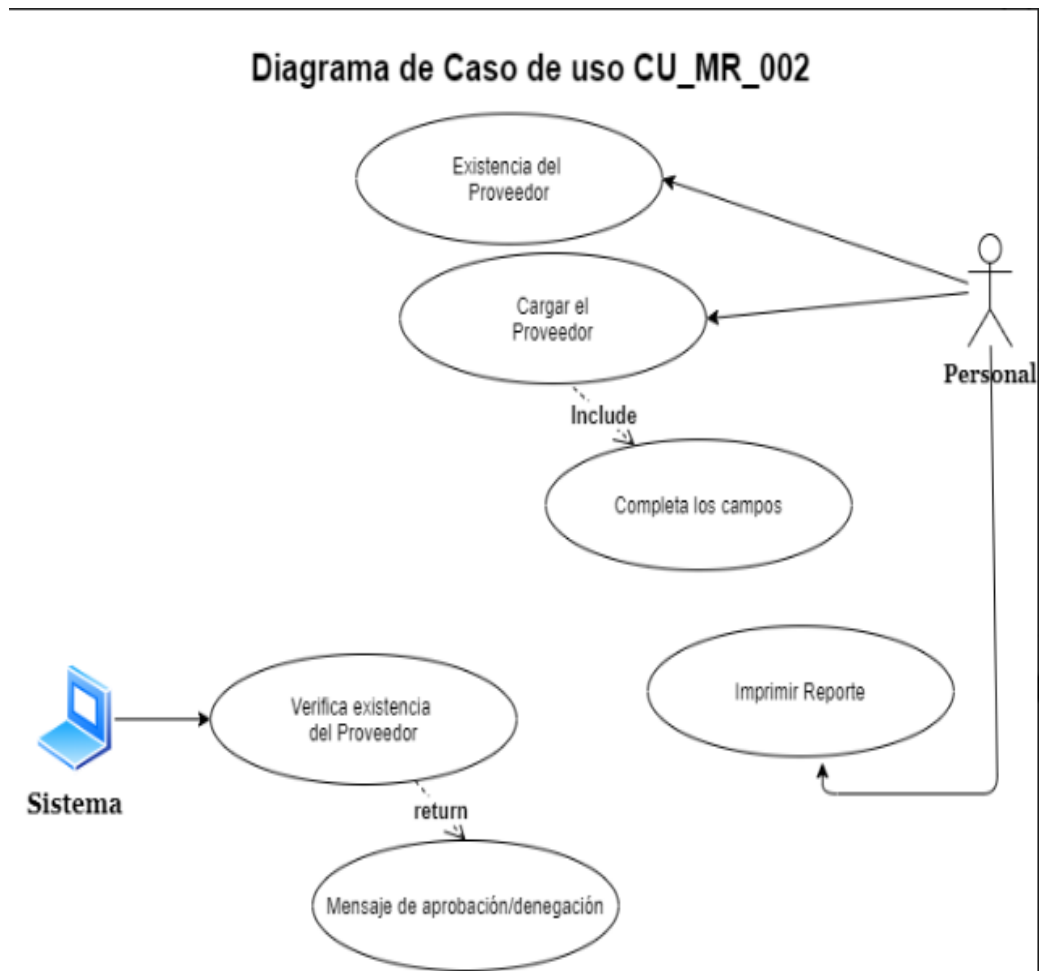
## DIAGRAMAS DEL SUB MODULO NUEVO PROVEEDOR

### Diagramas de Compartimiento

#### *Diagrama de Actividades*



*Diagrama de casos de uso*

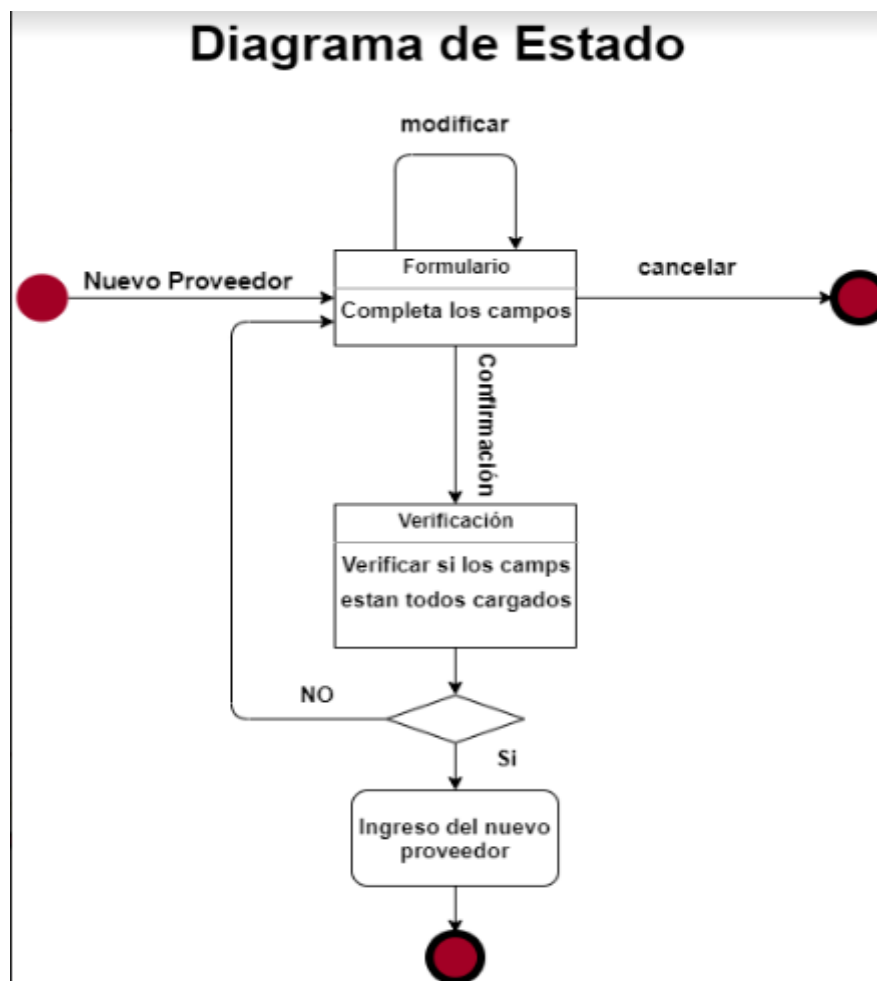




<b>CU_MR_002</b>	<b>Nuevo Proveedor</b>	
<b>Fecha</b>	<b>16/05/20</b>	
<b>Versión</b>	<b>1.0</b>	
<b>Autor</b>	<b>Giovani Giménez</b>	
<b>Actores</b>	<b>Personal (usuario)</b>	
<b>Objetivos Asociados</b>	Ingresar al sub modulo Nuevo proveedor para agregar proveedores.	
<b>Descripción</b>	Se presenta este caso de uso cuando el personal desea ingresar a un nuevo proveedor.	
<b>Precondición</b>	1. Personal debe de agregar proveedores e imprimir el reporte del proveedor.	
<b>Flujo Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El personal ingresa al sistema y verifica la lista de sus proveedores o agrega.
	2	El personal verifica a sus proveedores en la lista
	3	El personal ingresa al sub modulo y procede a la carga de sus proveedores
	4	El personal rellena los campos y envía el formulario cargado y aparece en la lista
<b>Flujo alternativo</b>	<b>Paso</b>	<b>Acción</b>
	4.1	El sistema advierte que los campos deber ser obligatorios.
	4.2	El sistema validad los datos, si los datos no son correctos el sistema emite un mensaje de denegación para que el personal corrobore los datos.
	4.3	El sistema validad los datos, si los datos son correctos emite un mensaje de aprobación.
<b>Post condición</b>	1	El sistema registra al nuevo proveedor.

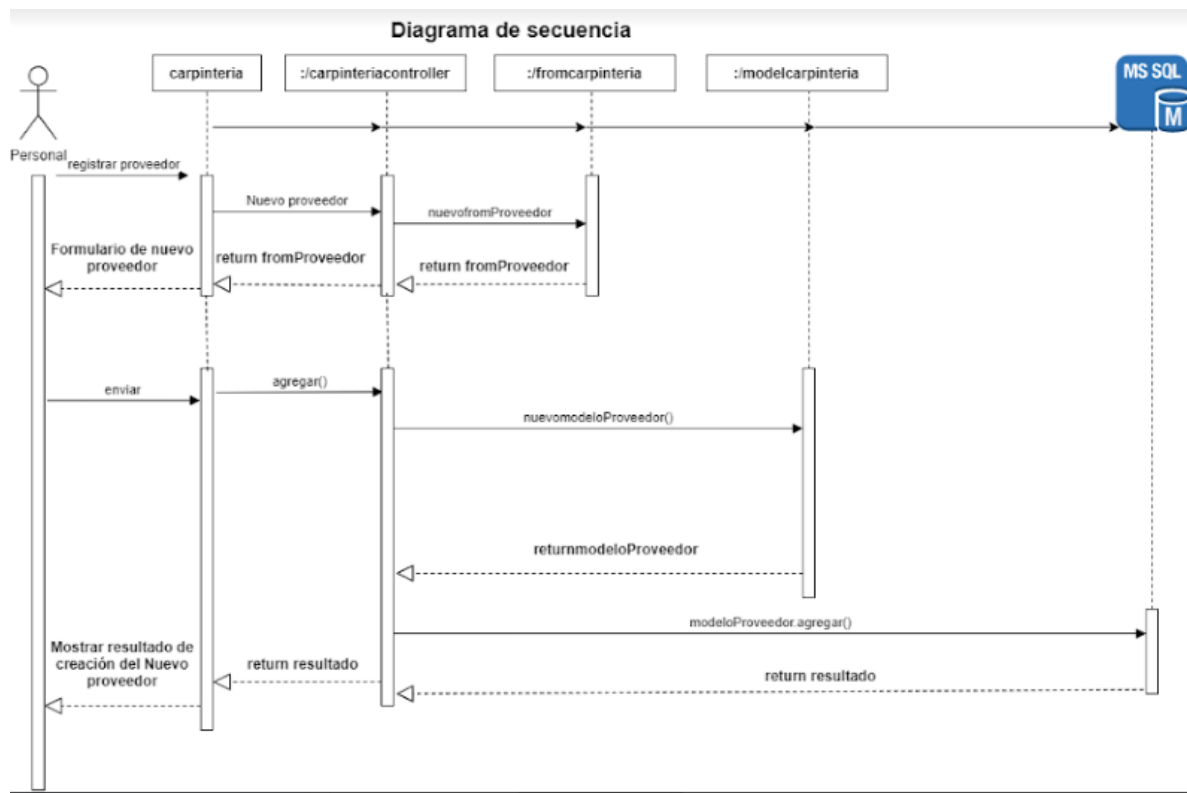
	2	El sistema puede imprimir un reporte de sus proveedores o también lo descarga en un formato .pdf.
<b>Importancia</b>	Alta	
<b>Comentarios</b>		

### Diagrama de estados

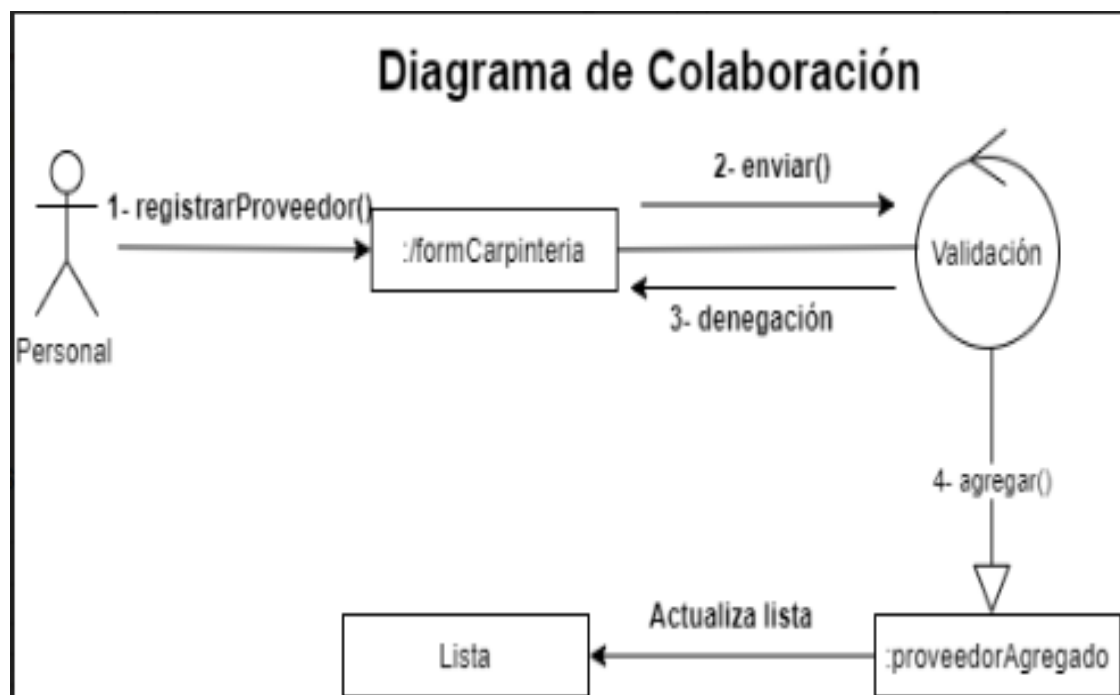


## Diagramas de Interacción

### Diagrama de secuencia



### Diagrama de colaboración



# ESPECIFICACIONES DEL SISTEMA

## MÓDULO COMPRA

### *Ámbito del proyecto a elaborar.*

Durante el desarrollo de la materia Ingeniería de Software, pretendemos elaborar un software de gestión para una carpintería, en este caso ficticia. El sistema contará con varios módulos. El módulo almacén, el módulo compra, el módulo venta y el módulo manufacturing. El mismo será desarrollado por todos los integrantes del curso, en lenguaje PHP, con un framework Laravel 6.13 (última versión), y una base de datos MySQL.

El alcance del presente proyecto considera solo el mercado local, y se enfocará en las necesidades de pequeñas y medianas empresas del rubro de la carpintería, ubicadas en la ciudad de encarnación, con el objetivo de proporcionar soluciones informáticas para los mismos.

El módulo el que nos toca desarrollar como grupo será el **módulo compra**. Este módulo se encarga de todos los procesos que forman parte del aprovisionamiento de artículos o materias primas. El módulo de compra incluye, para un control interno, el listado de proveedores, y actualización de existencias. El módulo compra se integra a su vez contará con 2 sub módulos principales: **sub módulo ingreso** y **sub módulo proveedor**. Este módulo emitirá un **reporte de compra** y un **reporte de proveedor**. Este módulo nos permite obtener el control de la parte operativa referente a la compras o adquisiciones (gastos) de una carpintería, con la finalidad de proporcionar información oportuna a la administración de la empresa.

Este módulo contiene a su vez, los siguientes sub módulos:

**Módulo ingreso:** Nos aparece una ventana donde nos va a permitir dar el respectivo mantenimiento de los pedidos o ingresos de artículos donde podremos agregar un nuevo pedido, editarlo, eliminarlo permitiendo también buscar los respectivos pedidos por su número de comprobante además tendremos la opción de visualizar todos los pedidos registrados.

Nuevo ingreso: Tendremos que seleccionar el proveedor del artículo, el tipo serie y número de comprobante elegimos el artículo que queremos hacer el pedido además la

cantidad el valor de compra y el valor de venta una vez que presionemos el botón de agregar el sistema se encargará de sumar el subtotal de todo el pedido.

Modulo Proveedor: Esta opción nos permite registrar los proveedores de la empresa, donde podemos registrar la siguiente información: Razón Social, Dirección, Ciudad, RUC, Teléfono, etc.

Nuevo proveedor: Permite agregar los detalles de cada uno de los proveedores que ingresan al sistema. Incluye Nombre de la Empresa, RUC, dirección, teléfono, Email.

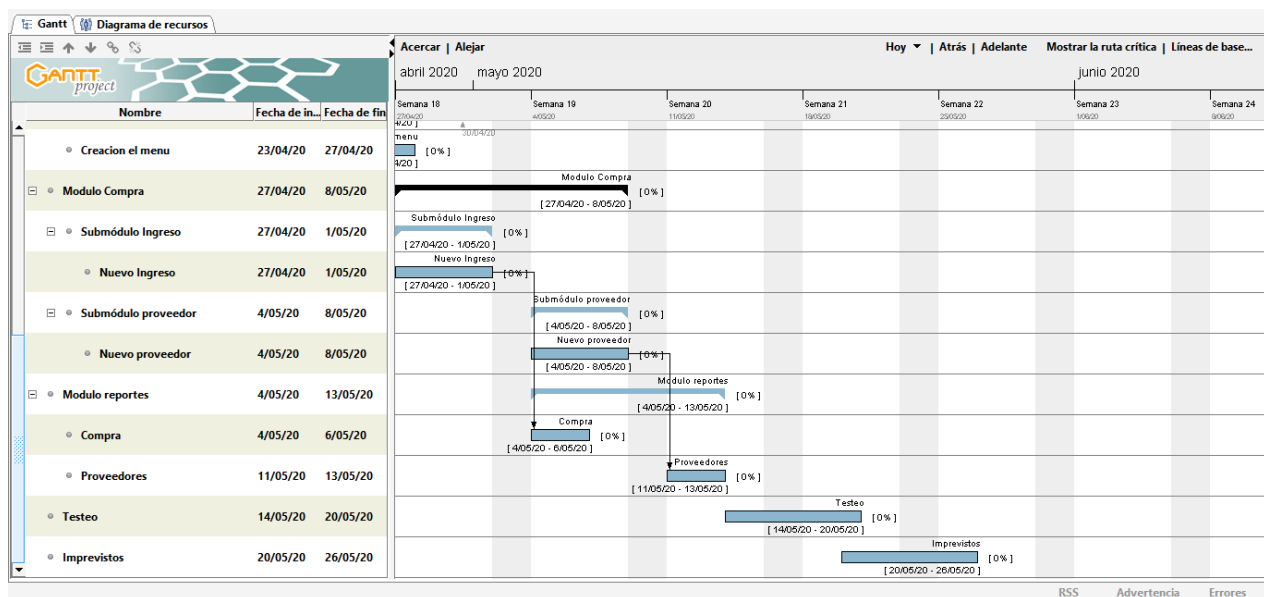
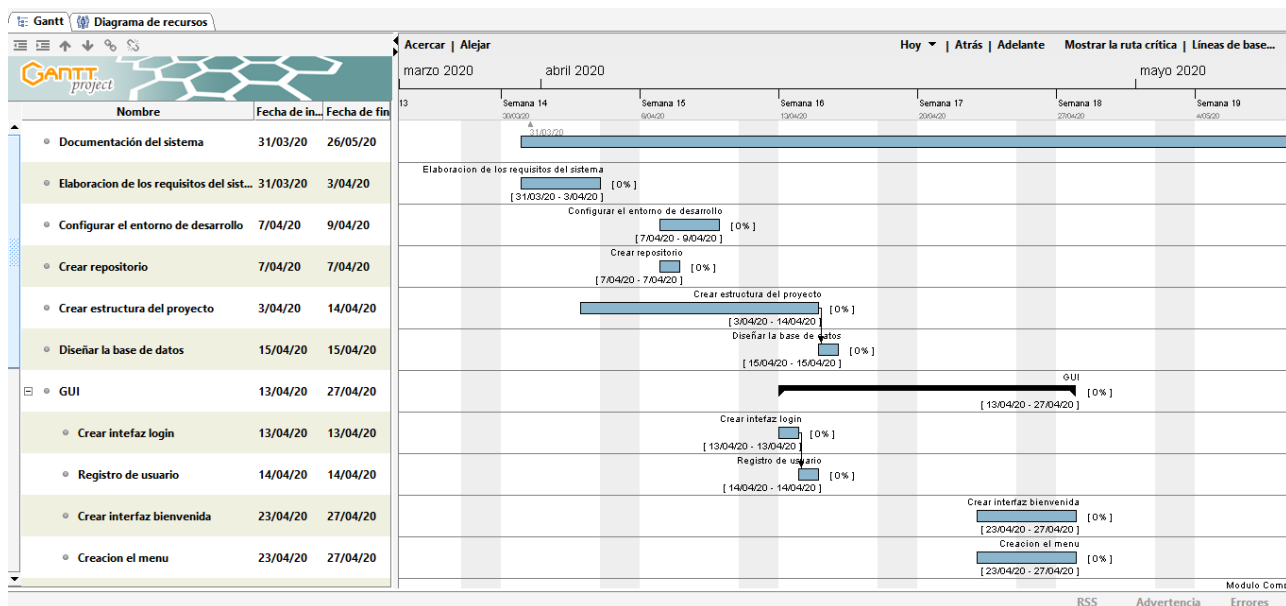
Orden de Compra: A través de la opción de registro de orden de compra generamos dicho documento donde debemos ingresar el proveedor, condición de pago, fecha, los productos, etc. Nos permite llevar un control de las órdenes de compra pendientes de ingresar al almacén.

Reportes:

Compra: Listado de todas las compras realizadas en el siguiente formato, fecha y hora de ingreso, nombre del proveedor, número de comprobante, total de la compra y estado.

Proveedores: Listado de todos los proveedores registrados.

# CALENDARIZACIÓN DE TAREAS CON RESPECTO AL TIEMPO



# ANÁLISIS DE RIESGO

## INTRODUCCIÓN

Tomar decisiones informadas mediante la evaluación consciente de lo que puede salir mal, así como la probabilidad y la gravedad del impacto, es el núcleo de la gestión de riesgos. Tomar decisiones informadas implica la evaluación de las compensaciones asociadas con todas las opciones de política de riesgo y mitigación en términos de sus costos, beneficios y riesgos, y la evaluación del impacto de la corriente decisiones sobre opciones futuras. Este proceso de gestión de riesgos encarna la identificación, análisis, planificación, seguimiento, control y comunicación de riesgos.

Un riesgo es una variable del proyecto que pone en peligro o impide el éxito del mismo. Es la “probabilidad de que un proyecto experimente sucesos no deseables, como retrasos en las fechas, excesos de costes, o la cancelación directa”. Se han producido amplios debates sobre la definición adecuada para riesgo de software, y hay acuerdo común en que el riesgo siempre implica dos características: Incertidumbre: el acontecimiento que caracteriza al riesgo puede o no puede ocurrir; por ejemplo, no hay riesgos de un 100 por ciento de probabilidad; Pérdida: Si el riesgo se convierte en una realidad, ocurrirán consecuencias no deseadas o pérdidas.

Es por esta razón que, para que un proyecto de desarrollo pueda llevarse a cabo dentro de los tiempos establecidos y los costos previstos, esos riesgos deben ser identificados y controlados, es decir se debe realizar un adecuado “Análisis y Gestión de Riesgos”. Mediante el análisis de los riesgos podremos identificar las amenazas que acechan a los distintos componentes pertenecientes o relacionados con el Sistema de información (conocidos como ‘activos’); para determinar la vulnerabilidad del sistema ante esas amenazas y para estimar el impacto o grado de perjuicio que una seguridad insuficiente puede tener para la organización, obteniendo cierto conocimiento del riesgo que se corre.

La gestión de los riesgos, basada en los resultados obtenidos en el análisis anterior, permite seleccionar e implantar las medidas o ‘salvaguardas’ de seguridad adecuadas para conocer, prevenir, impedir, reducir o controlar los riesgos identificados y así reducir al mínimo su potencialidad o sus posibles perjuicios.

## **Propósito del plan de Riesgo**

El propósito del presente plan es identificar los riesgos que se puedan presentar en el desarrollo del proyecto, analizarlos, calcular la exposición al riesgo (combinando información de su probabilidad de ocurrencia con la magnitud de su impacto), en base a ello poder priorizarlos, para establecer estrategias de control y resolución y ejercer una correcta supervisión de los mismos.

## **VISIÓN GENERAL**

### ***Objetivos***

El objetivo del presente plan de riesgo es asegurar que el proyecto se concrete dentro de los plazos previstos y observando todos los requerimientos fijados por los futuros usuarios. Permitiendo, al mismo tiempo, la detección oportuna de los problemas técnicos que se puedan presentar y la ejecución de un adecuado control y gestión de los cambios que se vayan presentando durante el desarrollo.

### ***Prioridades de aversión al riesgo***

Se definen para el presente plan las siguientes prioridades de aversión al riesgo:

Volatilidad de requisitos.

Cumplimiento de requisitos.

Cumplimiento de la planificación temporal.

Problemas de personal.

Problemas tecnológicos.

Problemas económicos.

## **ORGANIZACIÓN**

### ***Gestión***

La gestión del presente plan deberá ser llevada a cabo en forma directa por el Gestor de Riesgos del Proyecto con el apoyo de todos los personales con sus respectivos roles.



### ***Responsabilidades***

Del Gestor de Riesgos del proyecto:

Gestionar el presente plan.

Adoptar las medidas necesarias tendientes a evitar retrasos en la planificación realizada.

Tener un plan de contingencia para los diferentes tipos de problemas que se puedan presentar.

Del Líder de Proyecto:

Evaluar con el equipo de gestión de configuración los cambios solicitados en el caso que se presenten.

Ordenar al equipo de desarrollo la implementación de los cambios aprobados.

Supervisar el cumplimiento de la planificación de desarrollo del proyecto.

Del Gestor de configuración:

Interactuar con los futuros usuarios del Sistema para discutir posibles solicitudes de cambio y variación de los requisitos ya establecidos.

Comprobar que el producto satisfaga los requerimientos establecidos.

Evaluar con el equipo de gestión de configuración los cambios solicitados en el caso que se presenten.

De los documentadores del proyecto:

Interactuar con el equipo de trabajo para detectar tempranamente problemas técnicos o de personal.

Gestionar los informes de incidencia.

## **ANÁLISIS Y GESTIÓN DE RIESGOS**

### **Identificación**

La identificación del riesgo es un intento sistemático para especificar las amenazas al plan del proyecto. Los riesgos deben ser identificados y expresados en forma clara e inequívoca para que el equipo de desarrollo pueda llegar a un consenso y continuar hacia la fase de análisis y prioridad. El objetivo principal de la identificación de riesgos es la elaboración de una lista de los riesgos con los que el equipo deberá enfrentarse. Esta lista debe ser lo más extensa y detallada posible y deberá cubrir todas las áreas del proyecto, para ello se hará uso la clasificación de riesgos.

Los programas de adquisición, desarrollo e implementación continúan sufriendo grandes sobrecostos, retrasos en los horarios y bajo rendimiento técnico. Generalmente, esto es el resultado de no tratar apropiadamente y con incertidumbre en la adquisición y desarrollo de sistemas complejos de software. Las comunidades de adquisición y desarrollo, tanto gubernamentales e industriales, carecen de una forma sistemática de identificar, comunicar y resolver la incertidumbre técnica. A menudo, la atención se centra en los síntomas de sobrecostos y horarios retrasados en lugar de las causas fundamentales en la adquisición y desarrollo de productos. De hecho, todas las áreas en el desarrollo de sistemas son fuentes potenciales de riesgos de software ya que implica tecnología, hardware, software, personas, costos y horarios.

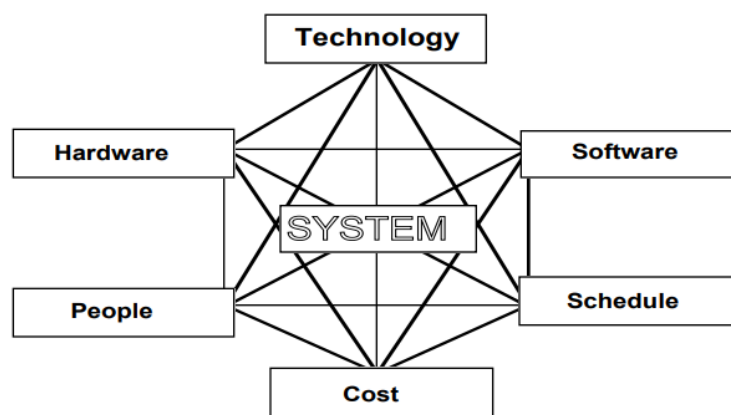


Imagen 1 Riesgos dentro de un contexto de sistema

### **Clasificación (o Taxonomía) de los Riesgos – Fuentes**

La clasificación de los riesgos, también denominadas taxonomías de riesgos- puede servir de ayuda para elaborar un enfoque coherente, reproducible y medible. Las listas de clasificación permiten al equipo pensar con mayor amplitud sobre los riesgos que pueden afectar al proyecto porque se dispone de una lista de áreas del proyecto susceptibles de esconder riesgos. Existen muchas taxonomías o clasificaciones para los riesgos de proyectos generales de desarrollo de software. Para el presente trabajo se ha escogido la clasificación propuesta por el Software Risk Management (SRM) desarrollado por el Software Engineering Institute.

La Taxonomía de riesgos sigue el ciclo de vida del desarrollo de software y proporciona un marco para organizar datos e información. El método de identificación basado en

taxonomía proporciona a la organización que desarrolla software un proceso sistemático de entrevista con el que identificar fuentes de riesgo.

El constructor de taxonomía consiste en un cuestionario basado en taxonomía y un proceso para su aplicación. La taxonomía organiza los riesgos de desarrollo de software en tres niveles: clase, elemento y atributo. El cuestionario consta de preguntas bajo cada atributo taxonómico que están diseñadas para generar el rango de riesgos y preocupaciones que potencialmente afectan el producto de software. El proceso de solicitud está diseñado de tal manera que el cuestionario pueda usarse de manera práctica y eficiente, de acuerdo con el objetivo de dar a conocer los riesgos del proyecto. Tanto el cuestionario como el proceso de solicitud se han desarrollado utilizando una amplia experiencia y múltiples pruebas de campo.

La siguiente tabla muestra una clasificación de alto nivel de las fuentes de riesgo de los proyectos siguiendo la taxonomía propuesta por la metodología SRM organizadas en tres niveles: clases, elementos y atributos.

ID	Elemento	Riesgo	Fuente
<b>RI-01</b>	Planificación	Errores en la estimación de presupuesto	Líder de Proyecto
<b>RI-02</b>	Planificación	Errores en la estimación de la calendarización	Líder del proyecto
<b>RI-03</b>	Planificación	Adelantar fecha de entrega	Usuario/ Líder del proyecto
<b>RI-04</b>	Requerimientos	Formulación de nuevos requisitos	Usuarios/ Documentadores
<b>RI-05</b>	Requerimientos	Eliminación de requisitos aprobados	Usuarios/ Documentadores
<b>RI-06</b>	Requerimientos	Errores en la definición de requisitos.	Usuarios/ Documentadores
<b>RI-07</b>	Requerimientos	Necesidades de los usuarios no recogidas de forma completa	Usuarios/ Documentadores
<b>RI-08</b>	Diseño	Existencia de funciones que presentan dificultades técnicas	Líder de Proyecto Desarrolladores
<b>RI-09</b>	Diseño	Diseño inadecuado para la realización del proceso de pruebas	Tester
<b>RI-10</b>	Diseño	Problemas con la definición de la interfase de usuario	Líder de Proyecto Desarrolladores
<b>RI-11</b>	Diseño	Aparición de errores por la reutilización de software preexistente	Líder de Proyecto Desarrolladores
<b>RI-12</b>	Gestión de Configuración	Solicitud de cambios no controlados o incorrectamente evaluados.	Usuarios Gestor de configuración
<b>RI-13</b>	Usuarios	Poca Comunicación con futuros usuarios	Líder de Proyecto Desarrolladores
<b>RI-14</b>	Usuarios	Ausencia de usuarios finales para validar el producto	Líder de Proyecto Desarrolladores

<b>RI-15</b>	Proceso	Falta de definición de formatos y ejemplos de documentos para todas las entregas definidas como parte del proceso del software	Líder de Proyecto/ Gestor de configuración
<b>RI-16</b>	Proceso	No documentar todos los resultados de las revisiones técnicas, incluyendo los errores encontrados y recursos empleados	Gestor de configuración
<b>RI-17</b>	Proceso	No definir reglas específicas para la documentación del código fuente.	Gestor de configuración
<b>RI-18</b>	Tecnología	Dificultades con herramientas y lenguajes nuevos de programación	Líder de Proyecto Desarrolladores
<b>RI-19</b>	Tecnología	Problemas con los equipos de hardware para el desarrollo	Líder de Proyecto Desarrolladores
<b>RI-20</b>	Equipo de Trabajo	Despido o renuncia o abandono de miembros del equipo de desarrollo.	Líder de Proyecto/ Desarrolladores
<b>RI-21</b>	Equipo de Trabajo	Dificultad de comunicación del equipo.	Líder de Proyecto Desarrolladores
<b>RI-22</b>	Equipo de Trabajo	Inexperiencia del equipo con la metodología de desarrollo	Líder de Proyecto Desarrolladores
<b>RI-23</b>	Equipo de Trabajo	Perdida de documentos y archivos del proyecto	Líder el proyecto/ Gestor de configuración
<b>RI-24</b>	Equipo de trabajo	Reposo a causa de problemas de salud	Líder de Proyecto/ Desarrolladores
<b>RI-25</b>	Equipo de trabajo	Permiso de ausencias laborales a causa de lutos	Líder de Proyecto/ Desarrolladores
<b>RI-26</b>	Equipo de trabajo	Problemas financieros	Líder de Proyecto/ Desarrolladores
<b>RI-27</b>	Equipo de trabajo	Paro de actividades por causa de Pandemia.	Líder de Proyecto/ Desarrolladores

## Declaración de los Riesgos

Las definiciones genéricas de un riesgo no hacen desaparecer la incertidumbre y dan lugar a distintas interpretaciones del riesgo. Las definiciones que no dejan lugar a dudas permiten a los equipos:

Asegurarse de que todos los miembros del equipo comprenden el riesgo de la misma forma.

Comprender la causa o causas del riesgo y la relación con los problemas que puedan surgir.

Disponer de una base para realizar un análisis formal y cuantitativo y planear los esfuerzos.

A continuación, se definen en forma más precisa los riesgos identificados anteriormente siguiendo un proceso de declaración en dos partes (condición – consecuencia). La primera parte de la declaración de riesgo se denomina **condición** y describe una situación o atributo del proyecto existente que el equipo prevé que puede resultar en una pérdida en el proyecto o en una reducción de beneficios. La segunda parte de la declaración de riesgo se denomina **consecuencia** y describe el atributo o situación no deseable del proyecto. Además, se incluyen los **efectos** que tendrían estos riesgos de no controlarse debidamente.

### **RI-01 Errores en la estimación del presupuesto.**

Condición: Errores en los cálculos, no estimar bien los factores que influyen en el cálculo (archivos, funciones, etc.) O el caso contrario, la sobre estimación.

Consecuencia: No disponer de los recursos necesarios para terminar el proyecto a tiempo, sobrecarga de tareas al personal.

Efecto: No contar con el presupuesto para desarrollar todas las funciones requeridas, baja calidad del producto.

### **RI-02 Errores en la estimación de la calendarización.**

Condición: Errores en los cálculos de estimaciones de fechas para el desarrollo del proyecto, no estimar bien los factores que influyen en el desarrollo (archivos, funciones, etc.).

Consecuencia: No disponer del tiempo necesario para terminar el proyecto a tiempo, sobrecarga de tareas al personal.

Efecto: No contar con el tiempo necesario para desarrollar todas las funciones requeridas, baja calidad del producto.

### **RI-03 Adelantar fecha de entrega.**

Condición: Se requiere que el sistema sea entregado antes de la fecha de entrega que fue fijada al iniciar su desarrollo.

Consecuencia: No disponer del tiempo necesario para terminar el proyecto a tiempo, sobrecarga de tareas al personal.

Efecto: No contar con el tiempo necesario para desarrollar todas las funciones requeridas, baja calidad del producto.

### **RI-04 Formulación de nuevos requisitos.**

Condición: Este riesgo tiene que ver con la volatilidad de los requerimientos de los usuarios. Al estar el producto enmarcado en estándares internacionales constantemente actualizados, este riesgo se reduce notablemente ya que con ello se tienen en cuenta los requisitos primordiales y de gran costo para el proyecto.

Consecuencia: Mayor tiempo de desarrollo, el proyecto puede sufrir retrasos y será más crítico cuando el desarrollo esté más avanzado.

Efecto: No terminar el producto en los plazos establecidos, carecer de las funcionalidades requeridas por los usuarios, el producto podría volverse inconsistente ya que no se han considerado de un comienzo estos requisitos y puede traer problemas para integrarlos.

### **RI-05 Eliminación de requisitos aprobados.**

Condición: Existen muchas causas por las cuales puede ser factible eliminar requisitos que en algún momento se los consideraba necesarios. Entre ellas se puede citar: relevamiento pobre de las necesidades de los futuros usuarios (lo que puede llevar a creer que son necesarias ciertas funciones que en realidad no lo son para los usuarios o la política de la empresa), ausencia de un estudio de antecedentes, o porque ya no es un requisito de los usuarios.

Consecuencia: Si se ha avanzado considerablemente sobre estos requisitos, puede significar un desperdicio de los recursos disponibles para el proyecto (humanos, financieros, etc.)

Efecto: El producto podría no contar con las funciones verdaderamente requeridas por los usuarios en un tiempo considerable, ya que se han empleado recursos en el

desarrollo de funciones que en realidad no beneficia a los usuarios y, posiblemente postergar tareas que eran vitales.

#### **RI-06 Errores en la definición de requisitos.**

Condición: Falta de experiencia del personal del proyecto, poca documentación para el análisis de los requisitos, etc.

Consecuencia: Desperdicio de recursos, más aún en fases avanzadas del proyecto. Clientes insatisfechos.

Efecto: Baja calidad del producto, ya que puede no ser detectado a tiempo, gastos innecesarios de recursos ya sea por el desarrollo de los mismos o por las correcciones que se deberían realizar al producto para enmendar errores.

#### **RI-07 Necesidades de las clientes no recogidas de forma completa.**

Condición: El equipo de desarrollo no se ha contactado con suficientes usuarios del sistema, de modo que no tiene una visión global de cuáles son los principales requisitos que demandan los usuarios finales.

Consecuencia: Pérdida de tiempo en el desarrollo de necesidades que podrían no ser las más importantes para los usuarios.

Efecto: Desarrollo de un producto incompleto, que no satisface de forma plena las necesidades específicas de los usuarios finales, con lo que se desestima su uso.

#### **RI-08 Existencia de funciones que presentan dificultades técnicas**

Condición: Errores en la determinación de ciertos factores en alguna de las etapas de desarrollo (análisis, diseño) son frecuentemente la causa de dificultades técnicas, poca experiencia en el lenguaje de programación utilizado, falta de experiencia en sistemas expertos y su tecnología de soporte, no considerar las limitaciones de la plataforma utilizada.

Consecuencia: Retraso del proyecto debido al tiempo necesario para superar estas dificultades, lo que también puede llevar a la inconsistencia de todo el sistema, con la posibilidad de tener que contratar personal adicional (altamente calificado en el tema) para solucionar el problema.

Efecto: En caso de no ser controlado el producto puede presentar serios defectos con lo cual puede producir el rechazo de los usuarios.

#### **RI-09 Diseño inadecuado para la realización del proceso de pruebas**



Condición: Escaso conocimiento de las funciones del producto por parte de los diseñadores del proceso de prueba. Falta de experiencia de los mismos.

Consecuencia: Esfuerzo adicional para la depuración del producto. No eliminación de la mayoría de los errores más importantes.

Efecto: Lanzar un producto con muchos defectos, retraso en la terminación de una versión estable del mismo.

#### **RI-10 Problemas con la definición de la interfase de usuario.**

Condición: Escasa experiencia del equipo de desarrolladores con aplicaciones afines para definir las interfaces adecuadas para los usuarios.

Consecuencia: Errores de diseño y retraso en la implementación del sistema.

Efecto: Desarrollar un producto con poca funcionalidad, dificultad en el manejo del sistema o no poder aprovechar todos los beneficios del mismo.

#### **RI-11 Aparición de errores por la reutilización de software preexistente**

Condición: Incompatibilidad del software, el software preexistente no es soportado completamente por la arquitectura diseñada, manejo de distintos formatos de datos por los diversos componentes (preexistentes y desarrollados por el equipo).

Consecuencia: Pérdida de tiempo en la solución de los errores, en la adaptación del software. Probabilidad de desarrollar ciertos módulos desde cero.

Efecto: Mal funcionamiento y bajo rendimiento del software en su conjunto, falta de funciones requeridas por los usuarios.

#### **RI-12 Solicitud de cambios no controlados o incorrectamente evaluados.**

Condición: Los cambios a solicitud del usuario o bien del equipo de desarrollo surgirán de manera inevitable.

Consecuencia: Complicaciones a la hora de gestionar los cambios, con posibilidades de incompatibilidades o inconsistencias.

Efecto: Si no se aplica un debido proceso de Gestión de Configuración del Software es posible desarrollar un producto con muchos defectos.

#### **RI-13 Poca comunicación con los futuros usuarios.**

Condición: El equipo de desarrollo no ha contactado con suficientes usuarios finales para definir sus necesidades.

Consecuencia: No tener una visión global de cuáles son los principales requisitos que demandan los usuarios finales, resistencia de los usuarios a implementar ciertas funciones.

Efecto: Desarrollo de un producto incompleto, que no satisface de forma plena las necesidades específicas de los usuarios finales, con lo que se puede desestimar su uso por gran parte de los futuros usuarios, ya que no sienten que el producto se adapta medianamente a su entorno.

#### **RI-14 Ausencia de usuarios finales para validar el producto.**

Condición: El equipo de desarrollo no cuenta con usuarios finales para validar los requisitos definidos, no contar con un grupo usuarios de un entorno real para la validación de los requisitos.

Consecuencia: Desarrollar un producto que posea funciones que no son necesarias para la mayoría de los usuarios y en contraste carencia de otras.

Efecto: Desarrollo de un producto incompleto, que no satisface de forma plena las necesidades específicas de los usuarios finales. O conformado por funciones que no le sirve al usuario.

#### **RI-15 Falta de definición de formatos y ejemplos de documentos para todas las entregas definidas como parte del proceso del software.**

Consecuencia: Dificultad para armar la carpeta del proyecto, puede llegar a ser muy difícil la utilización de dichos documentos o imposible. Ya que el equipo de desarrollo trabaja en forma distante, y a la vez existen sub-equipos.

#### **RI-16 No documentar todos los resultados de las revisiones técnicas, incluyendo los errores encontrados y recursos empleados**

Consecuencia: Es muy importante documentar el resultado de las revisiones (y datos relacionados) y ponerlos a disposición de todo el equipo de trabajo. Ya que puede servir, en el caso de los errores, por ejemplo: para no cometerlos nuevamente, y para utilizar esta información en proyectos similares.

#### **RI-17 No definir reglas específicas para la documentación del código fuente.**

Condición: Este riesgo se origina al no definir un estándar de programación.

Consecuencia: Dificultades para leer, comprender, detectar errores, realizar modificaciones del código fuente.

Efecto: Lanzar un producto con muchos defectos y baja calidad.

### **RI-18 Dificultades con herramientas y lenguajes nuevos de programación.**

Condición: Escaso conocimiento y experiencia de algunos de los integrantes del proyecto sobre las herramientas utilizadas y los lenguajes de programación.

Consecuencia: Mayor tiempo de desarrollo para investigar sobre el tema, o en la capacitación del personal.

Efecto: Lanzar un producto con baja calidad, con muchos defectos.

### **RI-19 Problemas con los equipos de hardware para el desarrollo.**

Condición: Equipos con los que cuentan los personales empiezan a fallar, a volverse lento, se descomponen o ya no funcionan.

Consecuencia: Invertir en reparaciones o nuevos equipos de hardware para el desarrollo.

### **RI-20 Despido o renuncia o abandono de miembros del equipo de desarrollo.**

Condición: Este riesgo se podría presentar como una consecuencia de RI-01, pero no necesariamente, también por la falta de presupuesto, por falta de tiempo, u otros motivos.

Consecuencia: Una de las consecuencias más evidentes es la demora del proyecto.

También se puede dar la ausencia de personal calificado para tareas específicas con lo que las mismas no se podrán realizar hasta contar con el personal requerido.

Efecto: Que el software desarrollado tenga carencias, ya sea técnicas o funcionales. O mala calidad del mismo en caso de no utilizar el personal calificado para llevar a cabo determinadas tareas.

### **RI-21 Dificultad de comunicación del equipo.**

Condición: El equipo de desarrollo se encuentra inhabilitado para reunirse físicamente debido a disposiciones del Ministerio de Salud Pública y Bienestar Social para la lucha contra el Covid-19.

Consecuencia: Dificultades de comunicación, la comunicación podría no ser lo suficientemente fluida o podría sufrir alteraciones en cuanto a tiempos.

Efecto: Errores de comunicación, retrasos en entregas de avances, posibilidad de avanzar en el desarrollo de una actividad sin la validación requerida.

### **RI-22 Inexperiencia del equipo con la Metodología de desarrollo.**

Condición: Algunos miembros del equipo de desarrollo pueden no contar con la experiencia suficiente en cuanto al uso de la Metodología.

Consecuencia: Retraso en el desarrollo de actividades definidas por la metodología, no aprovechar por completo las prestaciones de la metodología.

Efecto: Retraso en la entrega del producto final.

#### **RI-23 Pérdida de documentos y archivos del proyecto.**

Condición: puede darse el caso en que el disco de una computadora se queme y se borren todos los archivos debido a un corte de luz, o un virus lo destruya todo.

Consecuencia: Retraso y pérdida de documentos y archivos pertenecientes al desarrollo de actividades definidas.

Efecto: Retraso en la entrega del producto.

#### **RI-24 Reposo a causa de problemas de salud.**

Condición: puede darse el caso en el que el personal este con problemas de salud y requiera de reposo para su recuperación.

Consecuencia: Retraso en las tareas asignadas al personal o sin poder terminarse.

Efecto: Retraso en la entrega del producto.

#### **RI-25 Permiso de ausencias laborales a causa de lutos.**

Condición: puede darse el caso en el que un familiar cercano de algun miembro del equipo de desarrollo ha fallecido y el personal requiera de días de ausencia para darle luto al familiar.

Consecuencia: Retraso en las tareas asignadas al personal.

Efecto: Retraso en la entrega del producto.

#### **RI-26 Problemas financieros.**

Condición: no se cuentan con fondos necesarios para seguir llevando a cabo el desarrollo del proyecto.

Consecuencia: No poder pagar a los personales que llevan a cabo el desarrollo del proyecto.

Efecto: Retraso en la entrega del producto, baja calidad del producto.

#### **RI-27 Paro de actividades por causa de Pandemia.**

Condición: Aparece una enfermedad sin cura conocida, se declara Pandemia mundial y la OMS declara restricciones para evitar aglomeración de personas a nivel mundial, afectando el sistema de trabajo de muchas organizaciones a nivel mundial.

Consecuencia: Los equipos de trabajo no pueden reunirse físicamente, y las reuniones y trabajos proyectados deben realizarse de manera remota.

Efecto: Retraso en la entrega del producto, baja calidad del producto.

## ANÁLISIS Y PRIORIDAD DE LOS RIESGOS.

La meta principal del análisis de riesgos consiste en establecer las prioridades de los elementos de la lista de riesgos y determinar cuál de ellos justifica la reserva de recursos para el planeamiento. Por otro lado, la asignación de prioridades a los riesgos permitirá tratar en primer lugar los riesgos más importantes del proyecto.

### Estimación de la probabilidad.

La probabilidad del riesgo es una medida que calcula la probabilidad de que la situación descrita en el apartado de consecuencias de los riesgos de la declaración de riesgos llegue a producirse de verdad. Para cuantificar la incertidumbre acerca de la ocurrencia de los riesgos se emplearán las categorizaciones expresadas en lenguaje natural, en base a un rango de probabilidades establecido, como se muestra en la siguiente tabla:

Rango de probabilidad	Promedio para el calculo	Expresión de lenguaje natural	Valor numérico
de 1% a 10%	5 %	Baja	1
de 11 % a 25%	18 %	Poco probable	2
de 26% a 55%	40 %	Media	3
de 56% a 80%	68 %	Altamente probable	4
de 81% a 99%	90 %	Casi seguro	5

En la siguiente tabla se expresan los riesgos identificados para el proyecto con las probabilidades estimadas subjetivamente para cada uno de ellos.

ID	Riesgo	Expresión	Probabilidad
RI-01	Errores en la estimación del presupuesto	Poco probable	20%
RI-02	Errores en la estimación de la calendarización	Poco probable	18%
RI-03	Adelantar fecha de entrega	Altamente Probable	68%
RI-04	Formulación de nuevos requisitos	Poco probable	18%

<b>RI-05</b>	Eliminación de requisitos aprobados	Poco probable	18%
<b>RI-06</b>	Errores en la definición de requisitos.	Media	40%
<b>RI-07</b>	Necesidades de las clientes no recogidas de forma completa	Media	40%
<b>RI-08</b>	Existencia de funciones que presentan dificultades técnicas	Media	40%
<b>RI-09</b>	Diseño inadecuado para la realización del proceso de pruebas	Baja	5%
<b>RI-10</b>	Problemas con la definición de la interfase de usuario	Baja	5%
<b>RI-11</b>	Aparición de errores por la reutilización de software preexistente	Media	40%
<b>RI-12</b>	Solicitud de cambios no controlados o incorrectamente evaluados.	Poco Probable	18%
<b>RI-13</b>	Poca Comunicación con futuros usuarios	Poco Probable	18%
<b>RI-14</b>	Ausencia de usuarios finales para validar el producto	Baja	5%
<b>RI-15</b>	Falta de definición de formatos y ejemplos de documentos para todas las entregas definidas como parte del proceso del software	Media	40%
<b>RI-16</b>	No documentar todos los resultados de las revisiones técnicas, incluyendo los errores encontrados y recursos empleados	Media	40%
<b>RI-17</b>	No definir reglas específicas para la documentación del código fuente.	Media	40%
<b>RI-18</b>	Dificultades con herramientas y lenguajes nuevos de programación	Media	40%
<b>RI-19</b>	Problemas con los equipos de hardware para el desarrollo	Media	40%

<b>RI-20</b>	Despido o renuncia o abandono de miembros del equipo de desarrollo.	Altamente Probable	68%
<b>RI-21</b>	Dificultad de comunicación por la distribución del personal.	Media	40%
<b>RI-22</b>	Inexperiencia del equipo con la metodología de desarrollo	Media	40%
<b>RI-23</b>	Perdida de documentos y archivos del proyecto	Poco Probable	18%
<b>RI-24</b>	Reposo a causa de problemas de salud	Poco Probable	18%
<b>RI-25</b>	Permiso de ausencias laborales a causa de lutos	Baja	10%
<b>RI-26</b>	Problemas financieros	Media	40%
<b>RI-27</b>	Paro de actividades por causa de Pandemia.	Altamente Probable	100%

### Estimación del impacto.

El impacto del riesgo calcula la gravedad de los efectos adversos, la magnitud de una pérdida o el costo potencial de la oportunidad si el riesgo llega a producirse dentro del proyecto. Para el presente análisis se empleará la escala de medición subjetiva expresada en la siguiente tabla.

Criterio	Retraso en la planificación	Valor numérico
<b>Insignificante</b>	1 semana	1
<b>Marginal</b>	2 semanas	2
<b>Medio</b>	1 mes	3
<b>Crítico</b>	2 meses	4
<b>Catastrófico</b>	Más de 2 meses	5

En la siguiente tabla se definen el impacto que producirían la ocurrencia de los riesgos citados anteriormente:

Riesgo	Impacto	Riesgo	Impacto
<b>RI-01</b>	Marginal	RI-14	Critico
<b>RI-02</b>	Marginal	RI-15	Medio
<b>RI-03</b>	Medio	RI-16	Medio
<b>RI-04</b>	Crítico	RI-17	Medio
<b>RI-05</b>	Medio	RI-18	Medio
<b>RI-06</b>	Crítico	RI-19	Medio
<b>RI-07</b>	Media	RI-20	Critico
<b>RI-08</b>	Medio	RI-21	Media
<b>RI-09</b>	Marginal	RI-22	Critico
<b>RI-10</b>	Medio	RI-23	Catastrófico
<b>RI-11</b>	Medio	RI-24	Marginal
<b>RI-12</b>	Marginal	RI-25	Marginal
<b>RI-13</b>	Critico	RI-26	Medio
		RI-27	Crítico

### Exposición al riesgo.

La exposición al riesgo calcula la amenaza general que supone el riesgo combinando la información que expresa la probabilidad de una pérdida real con información que indica la magnitud de la pérdida potencial en un único valor numérico. Para el presente estudio la exposición al riesgo se calcula multiplicando la probabilidad de riesgo por el impacto. Luego se utilizará la magnitud de la exposición al riesgo para clasificar los riesgos.

Magnitud de exposición al riesgo	
<b>Aprox. 1</b>	Bajo riesgo.
<b>Aprox. 2</b>	Riesgo medio.
<b>Aprox. 3</b>	Alto riesgo.



ID	Riesgo	Probabilidad	Impacto	Exposición
<b>RI-01</b>	Errores en la estimación del presupuesto	5%	2	0.10
<b>RI-02</b>	Errores en la estimación de la calendarización	5%	2	0.10
<b>RI-03</b>	Adelantar fecha de entrega	68%	3	<b>2.04</b>
<b>RI-04</b>	Formulación de nuevos requisitos	18%	4	0.72
<b>RI-05</b>	Eliminación de requisitos aprobados	18%	3	0.54
<b>RI-06</b>	Errores en la definición de requisitos.	40%	4	<b>1.60</b>
<b>RI-07</b>	Necesidades de las clientes no recogidas de forma completa	40%	3	1.20
<b>RI-08</b>	Existencia de funciones que presentan dificultades técnicas	40%	3	1.20
<b>RI-09</b>	Diseño inadecuado para la realización del proceso de pruebas	5%	2	0.10
<b>RI-10</b>	Problemas con la definición de la interfase de usuario	5%	3	0.15
<b>RI-11</b>	Aparición de errores por la reutilización de software preexistente	40%	4	<b>1.60</b>
<b>RI-12</b>	Solicitud de cambios no controlados o incorrectamente evaluados.	18%	2	0.36
<b>RI-13</b>	Poca Comunicación con futuros usuarios	18%	4	0.72
<b>RI-14</b>	Ausencia de usuarios finales para validar el producto	5%	4	0.20
<b>RI-15</b>	Falta de definición de formatos y ejemplos de documentos para todas las entregas definidas como parte del proceso del software	40%	3	1.20
<b>RI-16</b>	No documentar todos los resultados de las revisiones técnicas, incluyendo	40%	3	1.20

	los errores encontrados y recursos empleados			
<b>RI-17</b>	No definir reglas específicas para la documentación del código fuente.	40%	3	1.20
<b>RI-18</b>	Dificultades con herramientas y lenguajes nuevos de programación	40%	4	<b>1.60</b>
<b>RI-19</b>	Problemas con los equipos de hardware para el desarrollo	40%	3	1.20
<b>RI-20</b>	Despido o renuncia o abandono de miembros del equipo de desarrollo.	68%	4	<b>2.72</b>
<b>RI-21</b>	Dificultad de comunicación por la distribución del personal.	18%	4	0.72
<b>RI-22</b>	Inexperiencia del equipo con la metodología de desarrollo	40%	3	1.20
<b>RI-23</b>	Perdida de documentos y archivos del proyecto	18%	5	0.90
<b>RI-24</b>	Reposo a causa de problemas de salud	18%	2	0.36
<b>RI-25</b>	Permiso de ausencias laborales a causa de lutos	18%	2	0.36
<b>RI-26</b>	Problemas financieros	40%	3	1.20
<b>RI-27</b>	Paro de actividades por Pandemia.	100%	3	<b>3.00</b>

## GESTIÓN DE LOS RIESGOS

La lista de riesgos presentada anteriormente servirá como punto de partida al proceso de administración de riesgos. La gestión de riesgos es una actividad continua, esta lista es susceptible de presentar modificaciones a medida que avance el proyecto, y como consecuencia de ello deberá ser actualizada a lo largo de todo el ciclo de análisis, planeamiento y supervisión de los riesgos.

## Líneas de Acción

Para ejercer una adecuada gestión y supervisión de los riesgos mencionados anteriormente, se elaborará un Plan de Acción y Un Plan de Contingencias para cada uno de ellos.

El **Plan de Acción** será utilizado para minimizar los riesgos mediante acciones preventivas. La probabilidad que un riesgo ocurra, así como el impacto que el mismo podría ocasionar en el proyecto pueden ser mitigados encarando los problemas en forma proactiva.

El **Plan de Contingencia**, por el contrario, intenta implementar respuestas rápidas para mitigar los efectos en el caso que los riesgos se concreten, es decir reducir el impacto de los mismos mediante una reacción planeada. Este plan, además definirá ciertos indicadores que permitirán poner en marcha las acciones previstas, es decir, en caso que se verifiquen ciertos disparadores se adoptarán las medidas indicadas.

***Riesgo RI-03: Adelantar fecha de entrega.***

### **Aspectos a considerar:**

**Importancia del riesgo:** Es probable que se adelante la fecha de entrega ya que ha sucedido con otros proyectos que se estuvieron desarrollando

#### **Información necesaria para seguir el estado del riesgo:**

Calendarización de sistema

Documento de especificaciones de requisitos de software.

**Responsables:** el responsable es el líder del proyecto junto con los futuros usuarios del sistema.

**Recursos para realizar las actividades de control de riesgo:** para realizar las actividades de control del riesgo no se necesitan recursos económicos extras, pero si herramientas y contacto con los futuros usuarios para analizar las fechas de entrega.

### ***Plan de Acción del Riesgo RI-03***

Contactarse con posibles usuarios del sistema para dejar escrito y firmado la fecha de entrega del proyecto.

Realizar reuniones con el grupo de desarrollo para exponer claramente la calendarización.

### ***Plan de Contingencia del Riesgo RI-03***

Disparador: Adelanto de la fecha de entrega del proyecto:

Interrogar a quien posee dudas ante la fecha de entrega del proyecto

Realizar una reunión entre usuarios y desarrolladores para constatar cuál es la nueva fecha de entrega del proyecto y a su vez:

Asignar a los desarrolladores más carga de tareas ajustando la calendarización del proyecto con el objetivo de terminar el desarrollo del software en la nueva fecha establecida.

### ***Riesgo RI-06 Errores en la definición de requisitos.***

#### **Aspectos a considerar:**

**Importancia del riesgo:** Es probable que en primera instancia no se interpreten debidamente los verdaderos requisitos de los futuros usuarios del sistema. Si este riesgo se verifica podría generarse graves inconvenientes en la implementación del sistema.

**Información necesaria para seguir el estado del riesgo:**

Documento de estudio de viabilidad de sistema.

Documento de especificaciones de requisitos de software.

**Responsables:** el responsable es el líder del proyecto junto con personal encargado de los requisitos.

**Recursos para realizar las actividades de control de riesgo:** para realizar las actividades de control del riesgo no se necesitan recursos económicos extras, pero si herramientas y contacto con los futuros usuarios para analizar los recursos.

#### ***Plan de Acción del Riesgo RI-06***

Contactarse con posibles clientes y usuarios del sistema.

Realizar las consultas y entrevistas necesarias con usuarios para lograr un relevamiento completo de sus necesidades.

Analizar detenidamente cada uno de los requisitos para comprobar la existencia de requerimientos encubiertos o mimetizados con otros requerimientos ya identificados.

Desarrollar y mantener un catálogo de los requisitos identificados donde aparezcan detalles sobre cada uno de ellos, el estado de desarrollo en el que se encuentra, como así también un informe de excepciones.

Realizar reuniones con el grupo de desarrollo para exponer claramente los requisitos identificados.

Construir prototipos para lograr que los usuarios, clientes y desarrolladores conozcan plenamente el producto y validen las características y funcionalidades del mismo antes de iniciar la fase del diseño definitivo del primer prototipo operacional.

Implementar planes de prueba y cruzar información para corroborar que el producto en desarrollo cumple todos y cada uno de los requisitos.

## ***Plan de Contingencia del Riesgo RI-06***

**Disparador:** Confusión en la interpretación funcional de los prototipos desarrollados:  
Ante diferencias de interpretación de las funcionalidades del prototipo en desarrollo se deberá:

Interrogar a quien posee dudas acerca de las funcionalidades del producto.

Realizar una reunión entre usuarios y desarrolladores para constatar cuál es la correcta visión por parte del usuario de la función donde se encontraron diferencias.

En caso de necesidad de implementar cambios proceder como lo establece el Plan de Gestión de Configuración.

## ***Riesgo: RI-11 Aparición de errores por la reutilización de software preexistente.***

### **Aspectos a considerar:**

**Importancia del riesgo:** para mejorar la eficiencia y economizar costos es aconsejable reutilizar software preexistente. Lo que frecuentemente se utiliza son módulos con determinadas funciones que son utilizadas en el desarrollo, también se puede aprovechar programas que se los puede integrar con el sistema. Pero puede que el software a reutilizar no se integre de manera consistente, lo que puede ser crítico.

**Información necesaria para seguir el estado del riesgo:** la información que se necesita para mantener al riesgo bajo control será la documentación que apoye la construcción de los módulos a reutilizar y la definición de los requisitos del sistema a desarrollar.

**Responsables:** El responsable es el líder del proyecto y el personal afectado a tareas relacionadas con el proyecto.

**Recursos para realizar las actividades de control de riesgo:** Para controlar este recurso no son necesarios recursos económicos, pero si es necesaria la documentación antes mencionada y la participación de cierto personal.

### ***Plan de Acción del Riesgo RI-11***

Para mantener controlado RI-11 deberán adoptar las siguientes medidas de prevención y seguimiento.

Antes de utilizar el software preexistente para implementar funciones al sistema se deberá analizar detenidamente la documentación de diseño del componente.

Seguidamente, se deberá estudiar y analizar si el software a reutilizar cumple con las especificaciones de diseño del sistema.

Se determinarán también los ajustes y modificaciones necesarios para adaptar el módulo a los .

Con esta información el encargado de la integración del módulo, precederá a consultar al líder del proyecto la conveniencia o no de la reutilización del software mencionado.

En caso de que se decida implementarlo se deberá realizar los ajustes necesarios y la documentación que será incluida a la del sistema, tomando como base la del software original.

### ***Plan de Contingencia del Riesgo RI-11***

Disparador: Aparición de errores en las etapas de implementación o pruebas:

Se informará al líder de desarrollo quien se encargará de llevar adelante el control del caso, con la ayuda del equipo de desarrollo.

Realizar pruebas del sistema sin la implementación del módulo reutilizado para comprobar si efectivamente es éste el que produce los errores.

En el caso de que se compruebe este hecho se deberá evaluar si es conveniente modificar el software reutilizado para que se adapte al sistema, sin generar errores.

***Riesgo: RI-18 Dificultades con herramientas y lenguajes nuevos de programación.***

**Aspectos a considerar:**

**Importancia del riesgo:** La falta de conocimientos por parte del personal son importantes ya que pueden demorar el desarrollo en los tiempos estimados.

**Información necesaria para seguir el estado del riesgo:**

Bases y conocimientos del personal.

Antecedentes de trabajo del personal.

Manejo del lenguaje de programación y base de datos por parte del personal.

**Responsables:** El responsable de realizar las actividades de control del riesgo es el líder del proyecto.

**Recursos para realizar las actividades de control de riesgo:** para realizar un adecuado control de este riesgo se necesitará personal capacitado para validar las funciones desde el punto de vista técnico. Este personal deberá tener amplios conocimientos en cuanto a Sistemas de Bases de Datos, lenguajes de programación, Sistemas expertos, etc.



### ***Plan de Acción del Riesgo RI-18***

Para mantener controlado el riesgo RI-18 se deberá adoptar las siguientes medidas de prevención:

Antes de integrar nuevos personales para el desarrollo del software, se deberá analizar la capacitación y el conocimiento en cuanto a lenguajes de programación, base de datos, sistemas expertos, etc. Para determinar su integración y que no presente dificultades o perjudique otras partes del sistema.

Una vez realizada la etapa anterior se deberá comprobar que el personal se encuentra lo suficientemente capacitado para llevar a cabo el desarrollo del software.

Con esta información el líder del proyecto decidirá la conveniencia o no de la integración del nuevo personal.

Si se decide integrar al nuevo personal, se deberá comunicar a los demás desarrollares para que le pongan al tanto del sistema y asignarle sus tareas de acuerdo a las necesidades del proyecto.

### ***Plan de Contingencia del Riesgo RI-18***

Disparador: Dificultades con herramientas y lenguajes nuevos de programación:

Informar al líder del proyecto quien se encargará de tomar el control del caso llevando adelante, con la participación del equipo de desarrolladores para llevar adelante las siguientes acciones:

Elaborar cursos prácticos para el manejo de herramientas y lenguajes de programación con el fin de que el personal tenga los conocimientos necesarios lo más pronto posible.

***Riesgo: RI-20 Despido o renuncia o abandono de miembros del equipo de desarrollo.***

**Aspectos a considerar:**

**Importancia del riesgo:** La ausencia de personal calificado son importantes ya que pueden demorar el desarrollo en los tiempos estimados.

**Información necesaria para seguir el estado del riesgo:**

Condiciones económicas, personales y sociales del personal.

Manejo del lenguaje de programación y base de datos por parte del personal.

**Responsables:** El responsable de realizar las actividades de control del riesgo es el líder del proyecto.

**Recursos para realizar las actividades de control de riesgo:** para realizar un adecuado control de este riesgo se necesitará conocer las condiciones económicas y personales del personal.

***Plan de Acción del Riesgo RI-20***

Para mantener controlado el riesgo RI-20 se deberá adoptar las siguientes medidas de prevención:

Tener un informe acerca del estado económico, personal, y social de los personales.

## ***Plan de Contingencia del Riesgo RI-20***

**Disparador:** Despido o renuncia del personal en las etapas de desarrollo:

Informar al líder del proyecto quien se encargará de tomar el control del caso llevando adelante, con la participación del equipo de desarrolladores para llevar adelante las siguientes acciones:

Se dividirán las tareas en partes iguales a los demás desarrolladores lo que correspondería al personal ausente con el fin de terminar el desarrollo de software en el tiempo estimado.

## ***Riesgo: RI-23 Perdida de documentos y archivos del programa.***

### **Aspectos a considerar:**

**Importancia del riesgo:** las pérdidas de documentos son importantes ya que puede atrasar catastróficamente el desarrollo de software.

**Información necesaria para seguir el estado del riesgo:**

Documentos y archivos del desarrollo del software.

**Responsables:** El responsable es el líder de proyecto.

**Recursos para realizar las actividades de control de riesgo:** para tratar este riesgo se necesitan disponer de herramientas para guardar los archivos. Se puede dedicar un servidor, el cual servirá de repositorio de archivos.

### ***Plan de Acción del Riesgo RI-23***

Para mantener controlado el riesgo RI-23 se deberán adoptar las siguientes medidas de prevención y seguimiento.

El líder de proyecto deberá realizar reuniones con todos los integrantes del grupo de trabajo, a fin de dilucidar los posibles problemas relacionados con la protección de archivos.

Establecer un mecanismo para llevar adelante las tareas grupales en forma virtual.

Seleccionar herramientas que se adapten a las necesidades.

Utilizar un repositorio con el fin de guardar archivos.

### ***Plan de Contingencia del Riesgo RI-23***

Disparador: Pérdida de archivos del desarrollo del software.

El líder de proyecto deberá realizar una reunión con el grupo responsable, para discutir la manera en que se puede adelantar el proceso de desarrollo y realizar de nuevo todos los archivos perdidos.

***Riesgo: RI-27: Paro de actividades por Pandemia.***

**Aspectos a considerar:**

**Importancia del riesgo:** Debido a lineamientos emitidos por el MSPYBS, se prohíbe la aglomeración de personas, causando así la imposibilidad de reunir al equipo de manera presencial.

**Información necesaria para seguir el estado del riesgo:**

Documentos y archivos del desarrollo del software.

**Responsables:** El responsable de realizar las actividades de control del riesgo es el líder del proyecto.

**Recursos para realizar las actividades de control de riesgo: :** Para controlar este recurso no son necesarios recursos económicos, pero si es necesaria la documentación antes mencionada y la participación de todo el equipo de desarrollo.

***Plan de Acción del Riesgo RI-27***

Para mantener controlado el riesgo RI-27 se deberán adoptar las siguientes medidas de prevención y seguimiento.

El líder de proyecto deberá realizar reuniones de manera remota utilizando las herramientas que sean necesarias con todos los integrantes del grupo de trabajo, a fin de esclarecer todos los lineamientos del trabajo y mantener la comunicación fluida como si el trabajo fuera de manera presencial.

Establecer un mecanismo para llevar adelante las tareas grupales en forma virtual.  
Seleccionar herramientas que se adapten a las necesidades. (Ej. Zoom, Google Meet, etc.)

Utilizar un repositorio con el fin de guardar archivos. (Ej. Git)

***Plan de Contingencia del Riesgo RI-23***

Disparador: Paro de actividades laborales y académicas presenciales a nivel mundial.

El líder de proyecto deberá realizar reuniones regulares con el grupo de trabajo, para discutir la manera en que se puede mantener el trabajo de manera fluida e ininterrumpida y evitar los atrasos de la manera más óptima posible.

# DISEÑO DE INTERFACES

## Login

El usuario podrá entrar al sistema ingresando su correo electrónico y contraseña de usuario.

The image shows a web interface for a login form. On the left is a dark vertical sidebar with icons: two circles, a hand cursor, a house, a group of people, and a grid. At the top of the sidebar is a search bar with the text 'Search' and a magnifying glass icon. The main content area is light gray and contains a white login form titled 'Iniciar Sesión'. The form has two input fields: 'Correo Electronico' and 'Contraseña'. Below the fields are two blue buttons: 'Aceptar' and 'Cancelar'. At the bottom of the page, there is a footer with the text 'Modulo Compra' on the left and 'Version 1.0' on the right.

Search

Iniciar Sesión

Correo Electronico

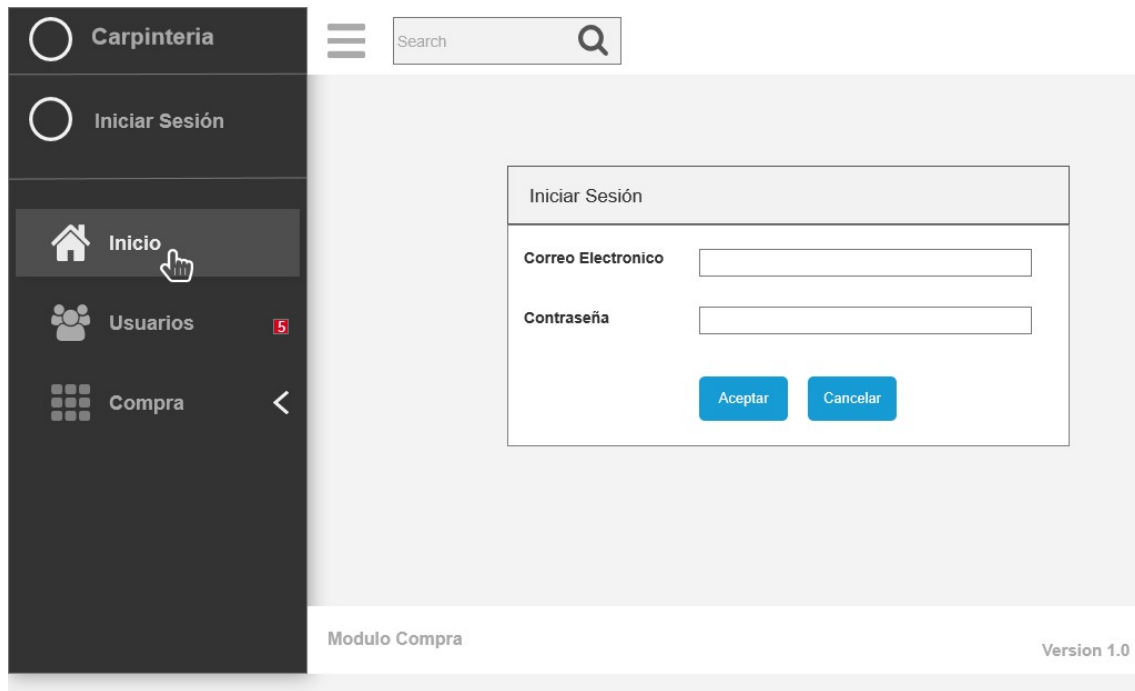
Contraseña

Aceptar Cancelar

Modulo Compra

Version 1.0

En la parte izquierda se encuentra el menú de opciones de Inicio, Usuarios, Compra, Proveedores, Reporte de proveedores.



## Proveedores

Nos aparece una ventana donde nos va a permitir dar el respectivo mantenimiento de los proveedores de los artículos donde podremos agregar un nuevo proveedor, editarlo, eliminarlo permitiendo también buscar los respectivos proveedores por su nombre, además tendremos la opción de visualizar en formato pdf todos los proveedores registrados.



Carpinteria

Alejandro

Cerrar Sesión

Inicio

Usuarios5

Compra

Nuevo Proveedor

Orden de Compra

Nuevo Ingreso

Search

Q

Listado de Proveedores

Nuevo

Nombre	Ruc	Telefono	Dirección	Email	Opciones
					<div>Editar</div> <div>Eliminar</div>
					<div>Editar</div> <div>Eliminar</div>
					<div>Editar</div> <div>Eliminar</div>
					<div>Editar</div> <div>Eliminar</div>
					<div>Editar</div> <div>Eliminar</div>

«

1

2

3

4

»

Modulo Compra

Version 1.0

Carpinteria

Alejandro

Cerrar Sesión

Inicio

Usuarios5

Compra

Nuevo Proveedor

Orden de Compra

Nuevo Ingreso

Search

Q

Listado de Proveedores

Agregar Proveedor

Nombre	Ruc	Telefono	Dirección	Email	Opciones
Universidad Aut	80050672-3	071205454	Padre Kreusser	repcion@una	<div>Editar</div> <div>Eliminar</div>
					<div>Editar</div> <div>Eliminar</div>
					<div>Editar</div> <div>Eliminar</div>
					<div>Editar</div> <div>Eliminar</div>
					<div>Editar</div> <div>Eliminar</div>

«

1

2

3

4

»

Modulo Compra

Version 1.0

## Nuevo Proveedor

Nos aparece una ventana donde nos va a permitir registrar un nuevo proveedor al sistema ingresando los respectivos campos de nombre, ruc, dirección, teléfono, correo electrónico.

The screenshot shows a web application interface with a dark sidebar on the left and a light gray main content area. The sidebar contains a menu with the following items: 'Carpinteria' (with a circle icon), 'Alejandro' (with a circle icon and 'Cerrar Sesión' below it), 'Inicio' (with a house icon and a blue background), 'Usuarios' (with a group of people icon and a red badge with the number '5'), 'Compra' (with a grid icon and a dropdown arrow), 'Nuevo Proveedor' (with a circle icon and highlighted), 'Orden de Compra' (with a circle icon), and 'Nuevo Ingreso' (with a circle icon). The main content area has a search bar at the top with the text 'Search' and a magnifying glass icon. Below the search bar is the title 'Nuevo Proveedor'. The form contains the following fields: 'Nombre' with the value 'Universidad Autónoma de Encarnación', 'Ruc' with the value '80050672-3', 'Dirección' with the value 'Padre Kreusser entre Independencia Nacional y Hd', 'Telefono' with the value '071205454', and 'Correo Electronico' with the value 'recepcion@unae.edu.py'. At the bottom of the form are two buttons: 'Guardar' (blue) and 'Cancelar' (red). A mouse cursor is pointing at the 'Guardar' button. At the bottom of the page, there is a footer with the text 'Modulo Compra' on the left and 'Version 1.0' on the right.

**Nuevo Proveedor**

**Nombre**

Universidad Autónoma de Encarnación

**Ruc**

80050672-3

**Dirección**

Padre Kreusser entre Independencia Nacional y Hd

**Telefono**

071205454

**Correo Electronico**

recepcion@unae.edu.py

**Guardar** **Cancelar**

Modulo Compra Version 1.0

# PLAN DE CONFIGURACION

## Control de la documentación

### *Control de la Configuración.*

Título:	
Referencia:	Ninguna
Autores:	Alejandro Villalba Patricia Fernández
Fecha:	14-05-20

### *Histórico de Versiones.*

Versión	Fecha	Estado	Responsable	Nombre de Archivo
1.0	17-05-20	Corregido	Alejandro Villalba	Plan_de_configuracion1.0
2.0	19-05-20	Revisión	Patricia Fernández	Plan_de_configuracion2.0
3.0	23-06-20	Corregido	Patricia Fernández	Plan_de_configuracion3.0

### *Histórico de Cambios.*

Versión	Fecha	Cambios
1.0	17-05-20	Primera Versión
2.0	19-05-20	Segunda Versión
3.0	23-06-20	Tercera Versión

## **Introducción**

### ***Propósito***

El objetivo de este Plan de Gestión de la Configuración (CM), es proporcionar una visión general de la organización, actividades, tareas en general, y los objetivos de Gestión de la Configuración. Se aborda la identificación de los elementos de configuración (CI), control de cambios y las auditorías de configuración en un alto nivel; se proporcionan detalles adicionales sobre las actividades de CM, técnicas y herramientas en los procedimientos de relacionados a CM.

### ***Alcance***

El plan de gestión de la configuración debe involucrar todas las fases del ciclo de vida del software. El documento permitirá mostrar los estándares de etiquetación de los productos de trabajo. Así mismo esclarecerá el tipo de nomenclatura utilizada para el control de las versiones de los documentos que se encuentran dentro de los elementos de la gestión.

Por otra parte, cualquier stakeholder podrá presentar cualquiera de los siguientes tipos de peticiones de cambio sobre el sistema, para el control de cambios:

Petición de cambios en los requerimientos (adiciones, supresiones, modificaciones, aplazamientos) en el software actualmente en desarrollo

Informes de los problemas en la producción corriente o sistemas de pruebas beta

Petición de mejoras en los sistemas actuales de producción

Petición de nuevos proyectos de desarrollo

Este proceso de control de cambio se aplica a los productos de línea base creados o gestionados por los miembros del sistema, incluyendo:

El software que se ha lanzado a la producción o se encuentra en versión beta

Requisitos de las especificaciones del sistema

Grupo de procedimientos y procesos

Usuarios y documentación técnica

Las siguientes clases de productos de trabajo están exentos de este proceso de control de cambios:

Los productos de trabajo que están todavía en desarrollo, a excepción de cambios en los requerimientos solicitados en nuevos proyectos.

### ***Documentos Referenciados***

Archivo	Nombre
Proceso de Gestión de la Configuración	Plan_de_configuracion.docx

### ***Acrónimos***

- CCB Configuration Control Board
- CM Control Management
- GCS Gestión de la Configuración del Software
- ECS Elementos de la Configuración de Software
- CMO Configuration Management Office
- CI Configuration Item

### ***Definiciones***

**Línea Base** Un documento o producto oficial aprobado que sirve como punto de partida para futuras versiones.

**Configuration Control Board** Revisa y aprueba los cambios sugeridos a un producto.

**Petición de Cambio** Una solicitud que alguien ha presentado al sistema de control de cambio que describe un problema de software, una mejora solicitada, una propuesta de cambio en los requisitos de un producto en fase de desarrollo, o un nuevo proyecto que se propone.

**Stakeholder** Persona que directa o indirectamente se ve afectada por el sistema y que puede afectar el proyecto.

**Configuration Item** Los elementos que son puestos bajo el control de gestión de la configuración.

### ***Organización***

- Sistema de Gestión de la Configuración

- Git, el Sistema de control de versiones, es una herramienta que se utiliza para almacenar todas las versiones del software y dar seguimiento de los cambios y líneas de base del proyecto.
- Github es un servicio para alojamiento de repositorios de software gestionados por el sistema de control de versiones Git.
- Personal, Roles y Responsabilidades

### ***CCB***

La CCB, está integrada por el Desarrollador Líder (Alejandro Villalba), está en cuestión tomará decisiones sobre los principales cambios necesarios. Cualquier cambio en los requisitos deberá ser aprobado por el CCB.

Revisar todas las peticiones de cambio y proporcionar los datos necesarios para determinar la disposición, tal como se describe en la Sección 3.4.

Asignar las peticiones de cambio aprobadas una fecha de implementación y un equipo.

Asegurar que las acciones tomadas por las peticiones de cambio son de manera oportuna.

### ***Presidente CCB***

El presidente de la CCB es el Administrador del Proyecto (Romina Patiño).

Dirigir las reuniones CCB.

Asignar/aprobar la disposición de cada petición de cambio y la asignación de la implementación de la petición de cambio aprobada.

Asegurarse de que la acción se toma en las solicitudes de cambio de manera oportuna.

### ***Administrador de la Herramienta de Gestión de la Configuración***

Es el encargado de administrar el sistema de gestión de la configuración (Patricia Fernández), introducir las líneas base, otorgar permisos y administración de usuarios.

#### **CMO (Patricia Fernández)**

Mantener el Plan de CM

Identificar los elementos de configuración (CI) y documentar sus características.

Controlar y facilitar los cambios a las características de un CI.

Realizar auditorías para verificar el cumplimiento del Plan CM.

Realizar auditorías para verificar la preparación para la liberación.  
Administrar la base de datos de administración de configuración.  
Informar a la CCB el estado de aprobación de todos los cambios propuestos y el estado de ejecución de todos los cambios aprobados.  
Trabajar con el Presidente y los miembros de CCB para programar reuniones periódicas de CCB, y preparar la agenda para cada reunión.

### ***Herramientas***

Las siguientes herramientas son usadas para administrar las líneas base del proyecto:  
Para gestionar los requisitos, y código fuente. La herramienta utilizada para la gestión de configuración de repositorios es Git.

### ***Procedimientos***

Estimación de tiempo para identificación de Elementos  
Con base al ERS del proyecto, el CMO, determino que el tiempo estimado para la identificación de los elementos tomara un total de 1 semana a partir del día martes 5 de mayo.

### ***Identificar elementos de configuración***

La identificación de los elementos de configuración (CI) consiste en determinar los componentes del sistema y documentos, que se pondrán bajo la identificación exclusiva de un identificador único, y hacerlas accesibles al equipo. Un esquema de correcta identificación, otorga a cada elemento trazabilidad entre el elemento y su información de estado.

Cada elemento se identificará de la siguiente manera

Proyecto\_TipoDeDocumento\_NombreDelElemento\_Versión (*MDC\_CMP\_2.3.1*), por ejemplo:

***Proyecto***                      *MDC (Modulo De Compra)*

***Tipo de Documento***    *PP (Planes del Proyecto)*

***Nombre del Elemento****Plan\_de\_configuracion*

***Versión del Elemento*** *1.0 (Primera Version)*

A continuación, se presentan los elementos que se pondrán bajo la gestión de la configuración:

- Diseño
- Código Fuente
- Herramientas
- Documentación del Sistema
- Procedimientos de Prueba
- Resultados de Prueba
- Establecer un sistema de administración de configuración
- Para seleccionar el sistema que servirá como gestor de la configuración, se tomara en cuenta los siguientes puntos:
  - Que la versión del software no sea de prueba o de paga.
  - Permita administrar a los usuarios que tendrán acceso al sistema
  - Permita otorgar permisos a los usuarios que accederán al sistema
  - Que sea un sistema fácil de usar
  - Que no sea un plugin de un ambiente de desarrollo (IDE).
  - Que se pueda utilizar en distintos sistemas operativos
  - Que permita solucionar los conflictos que surjan de una manera eficaz

Sistema	Prueba	Usuarios	Permisos	Usabilidad	Plugin	MultiOS	Resolución de Conflictos	Total
Git	No	Si	Si	Si	No	Si	Si	10
Assembla	No	Si	Si	Si	Si	Si	No	8
GoogleCode	No	Si	Si	Si	Si	Si	No	8

*Tabla 1. Comparación de sistemas de gestión de la configuración*

Con respecto a la Tabla 1. Comparación de sistemas de gestión de la configuración, se puede ver que el sistema que se usara es Git, ya que es el más completo según el análisis que se realizó.

Crear o liberar las líneas base

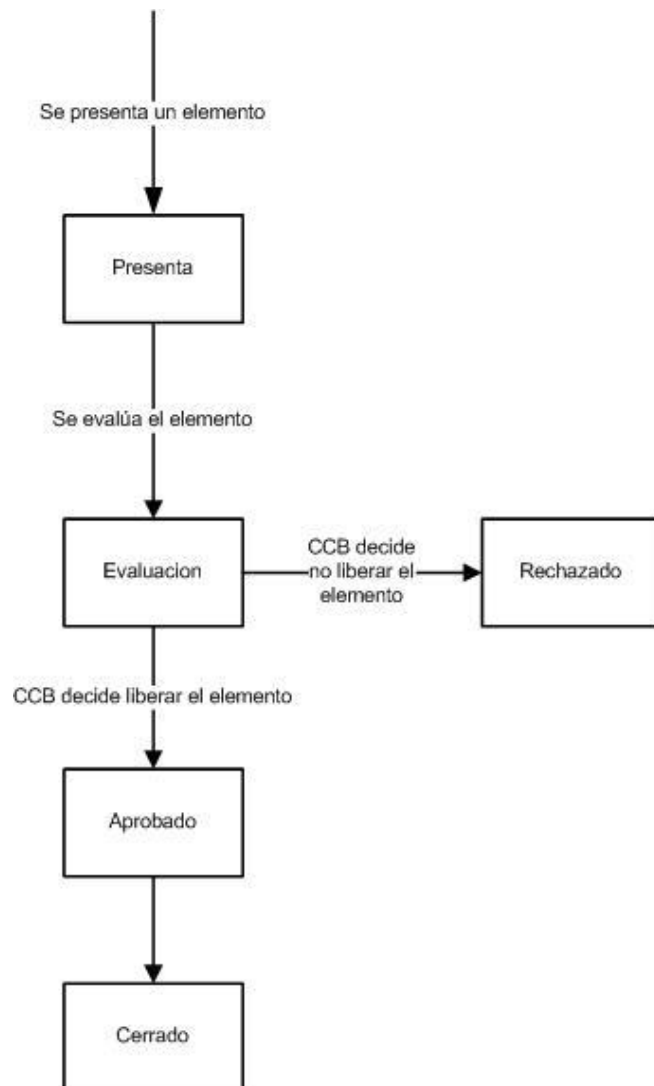
Para la creación o liberación de una línea base se debe seguir el proceso que se presenta en la Ilustración 1. Proceso de liberación de líneas base, la persona interesada en la creación o liberación de líneas base debe obtener la autorización de la CCB, cualquier



integrante del equipo puede presentar una propuesta para creación o liberación de línea base pero esta solo puede ser autorizada por la CCB, esta propuesta debe de realizarse a través del formato correspondiente.

Si la petición de liberación fue aprobada por el CCB, el administrador de la Herramienta de Gestión de la Configuración deberá de liberar los elementos de línea base que se le soliciten y deberá de registrar la salida de dichos elementos.

El CCB informará que el conjunto actual de líneas base esté disponible a los interesados.



*Ilustración 1. Proceso de liberación de líneas base*

## Seguir las peticiones de cambio

Cualquier cambio que afecta a los requisitos de línea base debe ser presentado a la CCB como una petición de cambio. La Ilustración 2. Proceso de petición de cambio muestra el diagrama de flujo típico de una solicitud de cambio. El CCB debe analizar y considerar cada petición de cambio. Si la solicitud es rechazada, puede ser redefinida y volver a presentarla. Si se acepta, los requisitos serán de nuevo línea base y todos los potencialmente afectados por el cambio serán informados. Esta petición debe de realizarse a través del formato correspondiente.

Cada petición de cambio presentada que sea aceptada se debe revisar o asignar una prioridad, y proporcionar una evaluación de impacto (una estimación aproximada del nivel de esfuerzo requerido para la ejecución, y el impacto de otras actividades actuales y previstas).

Se asigna una prioridad a cada petición de cambio cuando se recibe. La prioridad de una petición de cambio es asignada, ya sea por el autor o por el desarrollador líder. El desarrollador líder tiene la autoridad para modificar la prioridad de cualquier petición de cambio. Un cambio necesario tan pronto como sea posible, sin pasar por el calendario de lanzamientos regulares, se le asigna una prioridad urgente. Todos los cambios, independientemente de su prioridad, siguen el mismo proceso de aprobación.

Al recibir una petición de cambio, el CCB evalúa el cambio, en caso de ser necesario se pone en contacto con el autor del cambio, procesa la solicitud del cambio, y recomienda un calendario para la aplicación de los cambios aprobados.

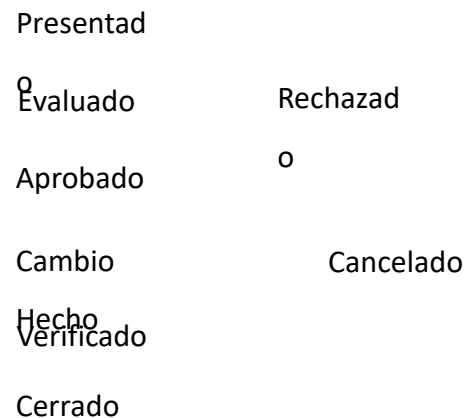


Ilustración 2. Proceso de petición de cambio

### ***Controlar los elementos de configuración***

La CMO recoge los datos necesarios para producir informes útiles para el CCB, y el administrador del proyecto.

Para la gestión del cambio, la CMO recoge información de identificación correspondiente a cada petición de cambio recibida y su estado en la base de datos de peticiones de cambio, tal como se define en el procedimiento de peticiones de cambio. Después la CMO prepara sus respectivos informes.

Para el estado de los elementos de configuración, la CMO recoge información de identificación correspondiente a cada elemento de configuración controlada, es decir, la versión actual, el historial de revisiones, el subsistema asociado. Al final de cada entrega, los elementos de configuración se actualizan como se define en el procedimiento de identificación de elementos bajo configuración. La CMO prepara informes conforme a lo solicitado sobre el estado de CI, que detalla las nuevas solicitudes de cambio, las solicitudes de cambio recientemente aprobadas, y las solicitudes de cambio cerradas. Se preparan informes de liberación por la CMO para que se ingresen en los informes de la liberación.

### **Realizar auditorías de configuración**

Las auditorías de configuración consisten en revisiones donde se compara el proceso de CM o una configuración de un producto con los requisitos para determinar si estos se cumplen.

Con base a lo anterior, se identifican las siguientes fechas para las auditorías de la gestión de la configuración:

- 12/05/2020
- 02/06/2020

### **Auditorías de línea base**

La CMO trabaja con el desarrollador líder para llevar a cabo una auditoría a las líneas base por cada liberación. Esta auditoría de base verifica el contenido de la versión completa, y todos los cambios que deberían realizarse han sido verificados por el equipo. Las revisiones de configuración se realizarán periódicamente para verificar la exactitud de la información de los elementos de configuración. El objetivo de la revisión es para verificar que todos los componentes del sistema han sido correctamente identificados y que todos los cambios en él han sido debidamente administrados. Las revisiones periódicas de configuración también permitirán evaluar la eficacia del proceso de CM y para identificar posibles modificaciones. El CMO seguirá de cerca la configuración de los elementos de configuración y trabajara con los involucrados para asegurar que los procedimientos de gestión de configuración se utilizan adecuadamente. Antes de las liberaciones principales del software, el Jefe de Desarrollo llevará a cabo una auditoría de la configuración mediante la realización de una construcción completa del software desde cero, y volverá a correr el conjunto de pruebas.

A continuación, se presentan los tres diferentes tipos de auditorías a líneas base para llevar a cabo:

#### ***Las auditorías de líneas base:***

Una auditoría de línea base se llevará a cabo por la CMO cuando cada línea base se halla establecido y se esté listo para pasar a la siguiente fase del ciclo de vida. Los resultados de esta auditoría se documentarán y se proveen al equipo de desarrollo.

Las auditorías de construcción de código serán llevadas a cabo por la CMO cuando una línea base construida esté lista para avanzar a la fase de pruebas. Esta auditoría verificara el contenido de la construcción en comparación con el contenido previsto en la planeación. Los resultados de esta auditoría se documentarán y se suministra con el equipo de pruebas para que conozcan que la aplicación está lista para ser probada. Cerca de la conclusión del proyecto, una auditoría funcional de configuración (FCA) se llevará a cabo por la CMO. La FCA valida que un producto de trabajo cumple con sus requisitos de rendimiento. La CMO analizará la:

Matriz de trazabilidad de Requisitos

Hojas de certificación y evidencia objetiva de que cada requerimiento se validó

Cerca de la conclusión del proyecto, una auditoría física de configuración (PCA) se llevará a cabo por la CMO. El PCA valida la coherencia entre el producto y su documentación técnica. La CMO analizara:

Lista de elementos a ser inspeccionados (inventario)

Registro de estado (inventario actualizado)

Herramientas y Técnicas de Implementación del Software

Nombre	Característica	Uso
<b>Git</b>	Es un sistema de control de versiones distribuido.	Control de versiones
<b>Github</b>	Es un servicio para alojamiento de repositorios de software gestionados por el sistema de control de versiones Git.	Servidor de versiones
<b>Scrum</b>	Se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto.	Metodología de desarrollo

<b>Espiral</b>	Las actividades de este modelo se conforman en una espiral, en la que cada bucle o iteración representa un conjunto de actividades.	Modelo de proceso
<b>Php</b>	Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.	Desarrollo del software
<b>Node.js</b>	Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos.	Librería para generar aplicaciones web
<b>Laravel</b>	Es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5. Su objetivo es desarrollar aplicaciones con código PHP de forma elegante y simple.	Framework
<b>Draw.io</b>	Es una herramienta de creación y edición de diagramas libre que permite la integración con diversas plataformas	Diagramas en general
<b>ArgoUml</b>	Es una aplicación de diagramado de UML escrita en Java	Diagramas en general

<b>GanttProject</b>	Es una aplicación que permite organizar y planificar proyectos a través de diagramas Gantt, como así también la carga horaria de los integrantes dentro del mismo.	Cronograma de actividades
<b>MySQL</b>	Es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual	Gestión de bases
<b>MySQL Workbench</b>	Es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL.	Modelar diagramas de Entidad-Relación para bases de datos MySQL.
<b>Atom</b>	Es un editor de código fuente de código abierto para macOS, Linux, y Windows con soporte para múltiples plug-in escritos en Node.js y control de versiones Git integrado, desarrollado por GitHub	Editor de código
<b>Xampp</b>	Es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl.	Sistema de gestión de bases de datos MySQL

## NOMENCLATURA PARA LOS ARCHIVOS DEL SISTEMA

El establecimiento de unas normas de nomenclatura comunes para toda la documentación gestionada es un aspecto fundamental dentro del proceso de gestión de

la configuración. Con estas normas se intentará establecer de forma lógica la forma de nombrar cualquier entregable documental atendiendo a las siguientes directrices:

Los nombres de los documentos deben tener una estructura común: Esto ayuda a la identificación rápida de los archivos y a la comprensión de su contenido, ya que se identifica cada parte del nombre con su significado.

Ejemplo: En este caso, cada documento parece utilizar una norma, lo que hace difícil identificar cada dato.

Acta\_20100301.doc

Subvenciones\_ACT\_Marzo2010.doc

Prestaciones\_Acta\_Seguimiento.doc

Deben poderse identificar de forma unívoca: Si todos los documentos existentes se ubicaran en una misma carpeta, se deberían cumplir que ningún archivo sobrescribiría a otro: lo cual quiere decir que no habría dos archivos con el mismo nombre.

Ejemplo: “Acta\_20100301.doc” si se celebran dos reuniones de distintos proyectos el mismo día, se repetiría este documento.

Deben ser fáciles de identificar: Hay que buscar un término medio entre un simple código y una descripción detallada para el nombre de los archivos.

Ejemplo: “ANEGDP\_023X23c\_KM.doc”

No deben ser excesivamente largos: No se debe tender a escribir el contenido del documento en su nombre.

Ejemplo: “Análisis de los cambios solicitados por el usuario para introducir la nueva propuesta de resolución.doc”

Deben incluir el número de versión como parte del archivo.

## **NOMENCLATURA PARA LOS CÓDIGOS Y BASE DE DATOS**

### **Nombres de variables**

Se utilizarán nombres significativos.

Los nombres deben estar en minúsculas, excepto la primera letra de cada palabra a partir de la segunda.



Se utilizará notación camello.

Ejemplo:

```
nuevoProveedor
```

### **Nombres de clases**

Cuando se crea una clase esta siempre debe empezar con mayúscula y continuar con notación camello.

Ejemplo:

```
class CarpinteriaAdmin
```

### **Nombres de los archivos**

Todo en minúscula excepto la primera letra de cada palabra a partir de la segunda.

Todos los programas deben tener la extensión .php preferiblemente.

Ejemplo:

```
carpinteriaIngreso.php
```

### **Nombres de las carpetas**

Todo en minúscula excepto la primera letra de cada palabra a partir de la segunda.

Ejemplo:

```
carpinteria
```

### **Corchetes e indentación**

La indentación en los programas (no sólo en los desarrollados en PHP, sino en casi todos los lenguajes de programación) es un buen hábito para la mayor comprensión de los bloques de código que se van anidando, y que representan un número de instrucciones que llevan a cabo un fin común (en la gran mayoría de las ocasiones).

### Ejemplo de indentación apropiada:

```
<?php
$producto = "Camisa";
function descubriendo($producto)
{
    if ($producto == "Camiseta"){
        echo "El producto es una Camiseta";
        echo "Adiós";
    }
    elseif ($producto == "Camisa"){
        echo "El producto es una Camisa";
        echo "Adiós";
    }
    else
        echo "No hay producto";
    }
    echo "El producto guardado es: ";
    descubriendo($producto);
}
?>
```

## **PASOS PARA HACER FUNCIONAR EL SISTEMA**

El proyecto se encuentra alojado en el repositorio remoto de git bajo el nombre carpintería si desea clonar y hacer funcionar el sistema debe de seguir los pasos detallados a continuación.

1. Primero se copia la URL del repositorio remoto que deseas clonar
2. Abrir una ventana de terminal e ingresar a la carpeta donde deseas descargar el proyecto y Clonar Proyecto:

```
git clone https://github.com/AlejandroVillalba/Carpinteria.git
```

3. Ingresar a la carpeta clonada e Instalar las dependencias del proyecto:

```
composer install
```

4. Copiar el archivo .env.example.env y cambiarle el nombre:

```
copy .env.example .env
```

5. Generar la clave de la aplicación:

```
php artisan key:generate
```

6. Instalar librería laravel/ui para la autenticación de usuario.

```
composer require laravel/ui
```

7. Instalar librería Dompdf dentro del directorio del proyecto ejecutamos la línea de comandos:

```
composer require barryvdh/laravel-dompdf
```

8. En el PhpMyAdmin:

Crear la base de datos “carpinteriadb” y en el archivo “.env” agregar la conexión a dicha base de datos con la siguiente línea

```
“DB_DATABASE=carpinteriadb”
```

9. Migrar las tablas de la base de datos:

```
php artisan migrate
```

10. En el PhpMyAdmin:

Ubicarse en la base de datos “carpinteriadb” y en el apartado de SQL insertar las tablas de auditorías con los trigger correspondiente en el siguiente orden, por cada bloque ejecutar el código:

```
CREATE TABLE proveedor_au_in (

id_proveedorAu int PRIMARY KEY AUTO_INCREMENT,
copia_nombreEmpresa varchar(255),
copia_ruc varchar(255),
copia_direccion varchar(255),
copia_telefono varchar(255),
copia_email varchar(255),
usuario varchar(255),
insertado datetime
)
```

```
CREATE TRIGGER PROVEEDOR_AI AFTER INSERT ON proveedor FOR EACH
ROW
INSERT INTO proveedor_au_in
(copia_direccion,copia_email,copia_nombreEmpresa,copia_ruc,copia_telefono,
insertado, usuario)
VALUES(new.direccion, new.email, new.nombreEmpresa, new.ruc, new.telefono,
now(), CURRENT_USER())
```

```
CREATE TABLE proveedor_au_up (

id_proveedorAu int PRIMARY KEY AUTO_INCREMENT,
Anterior_nombreEmpresa varchar(255),
Anterior_ruc varchar(255),
Anterior_direccion varchar(255),
Anterior_telefono varchar(255),
Anterior_email varchar(255),

Nuevo_nombreEmpresa varchar(255),
Nuevo_ruc varchar(255),
Nuevo_direccion varchar(255),
Nuevo_telefono varchar(255),
Nuevo_email varchar(255),
usuario varchar(255),
modificado datetime
)
```

```

CREATE TRIGGER AUTUALIZA_PROVEEDOR_BU BEFORE UPDATE ON
proveedor FOR EACH ROW
INSERT INTO proveedor_au_up
(Anterior_direccion,Anterior_email,Anterior_nombreEmpresa,Anterior_ruc,Anterior_telefono,

Nuevo_direccion,Nuevo_email,Nuevo_nombreEmpresa,Nuevo_ruc,Nuevo_telefono,
modificado, usuario)
VALUES(old.direccion, old.email, old.nombreEmpresa, old.ruc, old.telefono,
new.direccion, new.email, new.nombreEmpresa, new.ruc, new.telefono,
now(), CURRENT_USER())

```

```

CREATE TABLE proveedor_au_del (

id_proveedorAu int PRIMARY KEY AUTO_INCREMENT,
Eliminado_nombreEmpresa varchar(255),
Eliminado_ruc varchar(255),
Eliminado_direccion varchar(255),
Eliminado_telefono varchar(255),
Eliminado_email varchar(255),
usuario varchar(255),
eliminado datetime
)

```

```

CREATE TRIGGER ELIMINADO_PROVEEDOR_AD AFTER DELETE ON
proveedor FOR EACH ROW
INSERT INTO proveedor_au_del (Eliminado_direccion, Eliminado_email,
Eliminado_nombreEmpresa, Eliminado_ruc, Eliminado_telefono, eliminado, usuario)
VALUES(old.direccion, old.email, old.nombreEmpresa, old.ruc, old.telefono, now(),
CURRENT_USER())

```

11. Antes de registrar un usuario en el PhpMyAdmin en la tabla de Role crear los siguientes valores:

- Como Primer Id: Crear el Rol “administrador” en la file de **name**
- Como Segundo Id: Crear el Rol “moderador” en la file de **name**

12. Iniciar el servidor xampp y ejecutar la línea de comando para iniciar:

```
php artisan serve
```

13. Crear un usuario por defecto el sistema te asignara el rol de **administrador** y tendrás acceso a todo el sistema.