

Análisis y diseño de una aplicación web para gestionar la función empresarial de compras.

Analysis and design of a web application to manage the purchasing business function.

Mauricio Díaz Cabrera* (1).
Tecnológico Nacional de México, I.T. de Acapulco.
mm19320004@acapulco.tecnm.mx.

Rafael Hernández Reyna (2). Tecnológico Nacional de México, I. T. de Acapulco, rafael.hr@acapulco.tecnm.mx.

Mercedes Hernández de la Cruz (3). Tecnológico Nacional de México, I.T. de Chilpancingo,
mercedes.hd@chilpancingo.tecnm.mx

Miriam Martínez Arroyo (4). Tecnológico Nacional de México, I.T. de Acapulco, miriam.ma@acapulco.tecnm.mx.

*corresponding author.

Artículo recibido en enero 12, 2021; aceptado en febrero 26, 2021.

Resumen.

En este artículo se presenta el desarrollo de una aplicación web de compras, basado en la metodología de modelado conocida como proceso unificado de desarrollo o RUP, la cual es orientada a objetos y tiene como base los casos de uso para todas las fases del desarrollo. Se presentan los artefactos resultantes de cada fase dentro de un ciclo de la metodología hasta alcanzar un producto de interfaces visuales que representan los requerimientos funcionales del sistema, para lograrlo; se toma como ejemplo un caso de uso de una funcionalidad principal, el cual representa la manera en la que serán desarrolladas las demás funcionalidades que conforman el sistema. A pesar que la metodología RUP es uno de los procesos más generales para el desarrollo de software, queda demostrado un resultado satisfactorio en su aplicación para el desarrollo orientado a la web.

Palabras clave: Proceso unificado de desarrollo, RUP, metodologías de desarrollo.

Abstract.

This article presents the development of a purchasing web application, based on the modeling methodology known as rational unified process or RUP, which is object-oriented and based on "use cases" for all phases of the development. The artifacts resulting from each phase are presented within a cycle of the methodology until reach a product with visual interfaces that represent the functional requirements of the system, to achieve it; an example is taken form a use case of a main functionality, which represents the way in which other functionalities that make up the system will be developed. Although the RUP methodology is one of the most general processes for software development, a satisfactory result has been demonstrated in its application for web-oriented development.

Keywords: Rational unified process, RUP, development methodologies.

1. Introducción.

Actualmente, el desarrollo de software profesional ha crecido en volumen y complejidad, cada vez existe una mayor exigencia en la calidad y productividad del desarrollo, sin embargo; surgen complicaciones en el proceso que afectan la creación del producto.

De acuerdo a González (González, 2004), las fallas de los proyectos de software se deben a: una planificación irreal, mala calidad en el trabajo, personal inadecuado y cambios no controlados.

Para poder mejorar, la industria ha concentrado sus esfuerzos mayormente en el proceso y no en las herramientas de desarrollo. La solución construida ha evolucionado a través de los años, mediante estándares y metodologías que proponen fuertes cimientos y permiten que los sistemas de software tomen fuerza y cubran un mayor espectro dentro de las actividades de la sociedad.

Las metodologías surgen como guías para realizar acciones propias de un desarrollo, permiten identificar que hacer y cómo reaccionar ante algún evento antes, durante y después del desarrollo.

Este documento tiene como objetivo principal demostrar un proceso de desarrollo que asegure las mejores prácticas, tomando como referencia el desarrollo de una aplicación Web de compras. El cual se estructura de la siguiente manera: primero mostrando la literatura consultada, en donde se propone un marco teórico con los conceptos de la metodología RUP y los diagramas UML que se utilizaran después en el desarrollo; enseguida, se ofrece un recorrido a través de las fases del desarrollo y los artefactos que se crean en cada una de estas, para finalmente; detallar las conclusiones del desarrollo.

2. Métodos.

El Proceso Unificado de Desarrollo.

Según (Torossi, 2004), el proceso de unificado es “un conjunto de actividades necesarias para transformar los requisitos del usuario en un sistema de software”; es un marco genérico que puede especializarse para una variedad de sistemas, áreas de aplicación, tipos de organizaciones y diferentes tamaños de proyectos. Entre sus principales características están:

Dirigido por casos de uso.

Cualquier sistema informático es desarrollado para cubrir las necesidades de los usuarios. Un usuario no solamente son personas sino la interacción con otros sistemas.

Los casos de uso, definen los objetivos de cada iteración, se emplean para modelar los requerimientos funcionales de un sistema y, por otra parte, son una guía el proceso de desarrollo.

Centrado en la arquitectura.

La arquitectura, deberá incluir aspectos dinámicos y estáticos del sistema, es descrita por las vistas del sistema bajo construcción.

Los casos de uso no están solos, van de la mano con la arquitectura del sistema, cada uno recibe la influencia del otro, a medida que los casos de uso son llevados a cabo, se descubre más de la arquitectura.

Iterativo e incremental.

La complejidad, implica muchas horas de trabajo y esfuerzo; debido a esto, es funcional dividir el proyecto de software en pequeñas partes. Cada mini proyecto resulta en un incremento.

Cada iteración se debe llevar a cabo de manera controlada, es decir, deben ser seleccionadas y ejecutadas de manera proyectada. Cada iteración realiza las actividades específicas del ciclo de vida del desarrollo.

Entre algunos beneficios de esta perspectiva iterativa se encuentran:

- Controlar la iteración, disminuye el riesgo a los costos de los proyectos de un incremento.
- Disminuye demoras, considerando los riesgos más significativos en primer lugar.
- Agiliza el desarrollo.
- Muestra un enfoque más real, pues reconoce que no es posible definir los requisitos por completo desde el inicio.

El ciclo de vida. El Proceso Unificado de Desarrollo es reproducido mediante ciclos. Un ciclo representa una versión del sistema.

Fases (IBM, 1998).

Las cuatro fases que constituyen un ciclo son:

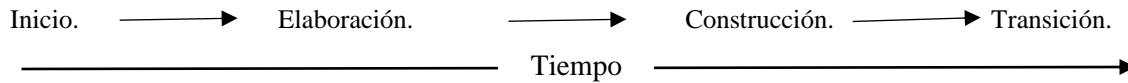


Figura 1. Línea de tiempo de las fases de la metodología RUP. (Torossi, 2004)

Tabla 1. Definición de las fases del proceso unificado de desarrollo. (Torossi, 2004)

Inicio	Definir el alcance del proyecto.
Elaboración	Planificar el proyecto, elaborar una arquitectura base, captura de requisitos.
Construcción	Construir el sistema.
Transición.	Transición a los usuarios, correcciones.

Cada fase se subdivide en iteraciones; la Figura 1 ejemplifica como cada iteración se desarrolla en secuencia de un conjunto de disciplinas o flujos de trabajo. (Torossi, 2004)

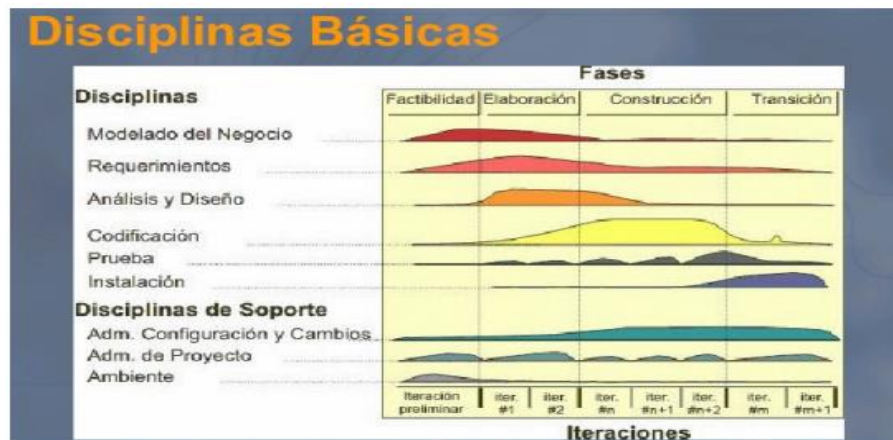


Figura 2. Disciplinas básicas del proceso unificado. (Torossi, 2004)

UML.

El lenguaje unificado de modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, hojear, configurar, mantener y controlar la información sobre tales sistemas.

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. La estructura estática define los tipos de objetos importantes para un sistema y para su implementación, así como las relaciones entre objetos. El comportamiento dinámico define la historia de los objetos en el tiempo y la comunicación entre objetos para cumplir sus objetivos.

¿Qué hay en un modelo?

En un modelo hay dos aspectos importantes: información semántica y presentación visual (notación).

La semántica capta el significado de una aplicación como una red de construcciones lógicas (clases, asociaciones, estados).

La notación muestra la información semántica de un modo que pueda ser considerada hojeada y corregida por los seres humanos. (James Rumbaugh, 2000)

Tipos de diagramas UML.

Al día de hoy, en la versión 2.5.1 de UML, existen dos clasificaciones de diagramas: los diagramas estructurales y los diagramas de comportamiento.

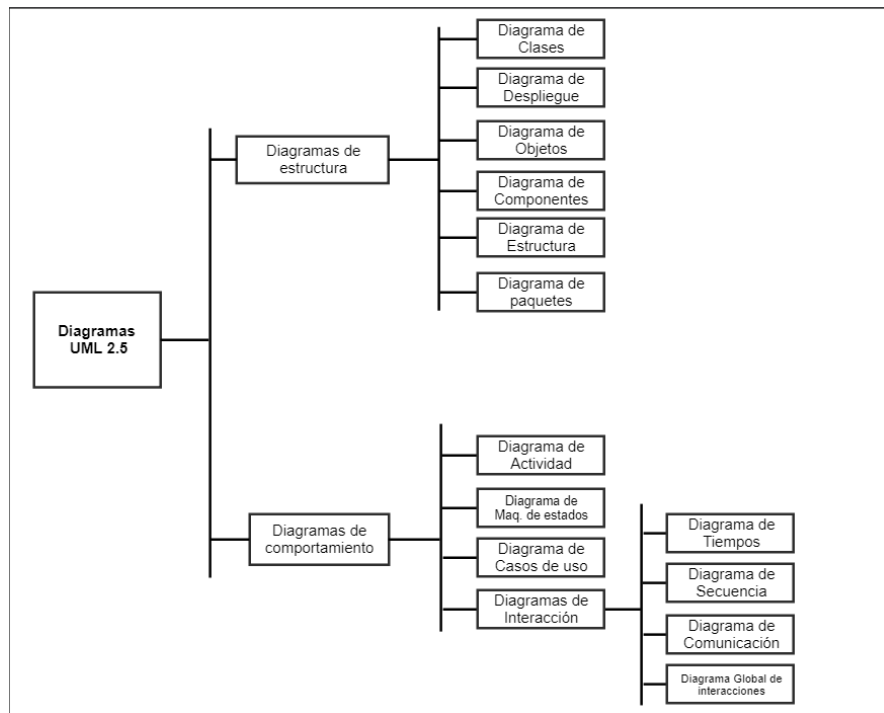


Figura 3. Clasificación de los diagramas UML.(Framework, 2020).

Diagrama de contexto.

El diagrama de contexto es un diagrama perteneciente al grupo de diagramas de flujo de datos, también conocido como diagrama de nivel 0, representa el nivel más alto de abstracción para la comprensión de un sistema. Consiste en terminadores, flujos de datos y flujos de control y un solo proceso el cual suele ser el nombre del sistema completo. Los terminadores o entidades externas se representan por medio de rectángulos y se comunican por medio de flechas que representan el flujo, los terminadores no se comunican entre sí.(Yourdon, 1993).

3. Desarrollo.

En este desarrollo se tomará como guía la metodología RUP propuesta en el apartado anterior, la cual irá a través de las diferentes fases del ciclo de desarrollo.

Fase de inicio.

La fase de inicio es la fase en donde se enuncia una descripción del producto final y presentan el proceso de negocio; se observa la posible arquitectura del sistema y se describe el plan del proyecto. También, se reconocen los casos de uso más críticos para reconocer las principales funciones del sistema, es aquí donde se producen los siguientes artefactos:

- Visión general del sistema.
- Esbozo del modelo de contexto.
- Lista de Actores.
- Esbozo del modelo de negocio.
- Esbozo del modelo de casos de uso.

Siguiendo el orden de los artefactos en esta fase se presenta la visión general del sistema.

Visión general del sistema.

Este proyecto consiste en el desarrollo de un sistema Web de compras, el cual posibilitará que la información sea accesible y procesable desde computadoras personales que cuenten con un sistema operativo Linux y/o Windows, mediante navegadores web.

Modelo de contexto.

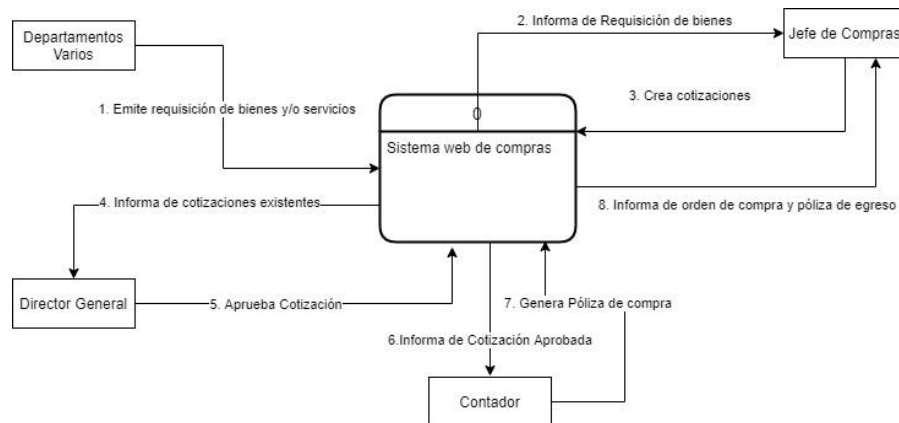


Figura 4. Diagrama de Contexto. Sistema de Compras. Fuente: elaboración propia.

La figura 4 describe el flujo de datos del sistema de compras de manera contextual, en donde en primer lugar los diferentes departamentos de la empresa emiten una requisición de bienes al sistema, el sistema informa al jefe de compras de las requisiciones y este las convierte en cotizaciones, el sistema envía las cotizaciones al director general para su aprobación, una vez aprobadas las cotizaciones el sistema informa al contador de la empresa para que genere una póliza de compra, después el sistema informa al jefe de departamento de compra de la emisión de una orden de compra junto con la póliza de egresos que respalda la compra.

Listado de actores.

Los actores son usuarios que utilizarán la aplicación web, cada actor representa un rol dentro de la misma. La siguiente tabla, muestra los actores identificados y la descripción correspondiente.

Tabla 2. Listado de actores de la aplicación web de compras. Fuente: elaboración propia.

No.	Actor.	Descripción.
1	Administrador de la aplicación.	Encargado administrar los parámetros básicos de la aplicación y del correcto funcionamiento por ejemplo la gestión de los usuarios del sistema.
2	Jefe de Dpartamento (varios).	Se encarga de realizar las requisiciones de los bienes o servicios.
3	Jefe del Departamento de Compras.	Encargado de crear la(s) cotización(es), a partir de una requisición en particular y dar seguimiento al proceso hasta que la compra se cierre.
4	Director General.	Este usuario se encarga de ajustar las cotizaciones, de acuerdo al presupuesto de cada departamento solicitante. Tiene la facultad de rechazar o aprobar una cotización.

5	Contador General.	Usuario que se encarga de crear un documento que respalde el gasto que se genera por una cotización aprobada para su compra.
---	-------------------	--

Diagrama de usuarios.

El diagrama de usuarios representa la jerarquización de los usuarios de acuerdo con los requerimientos de información y acceso al contenido de la aplicación. (Ver Figura 5).

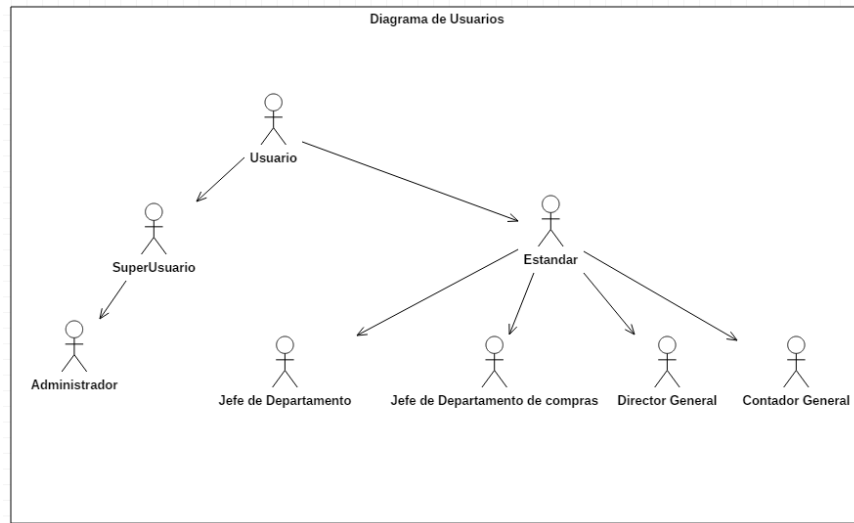


Figura 5. Diagrama de usuarios de la aplicación web de compras. Fuente: elaboración propia.

Modelado de negocio.

El objetivo de modelar el flujo de negocio del sistema es el representar los roles y los procesos bajo la responsabilidad de cada rol dentro de la empresa. En la figura 6, se presenta el diagrama de actividades UML, del proceso de compras propuesto.

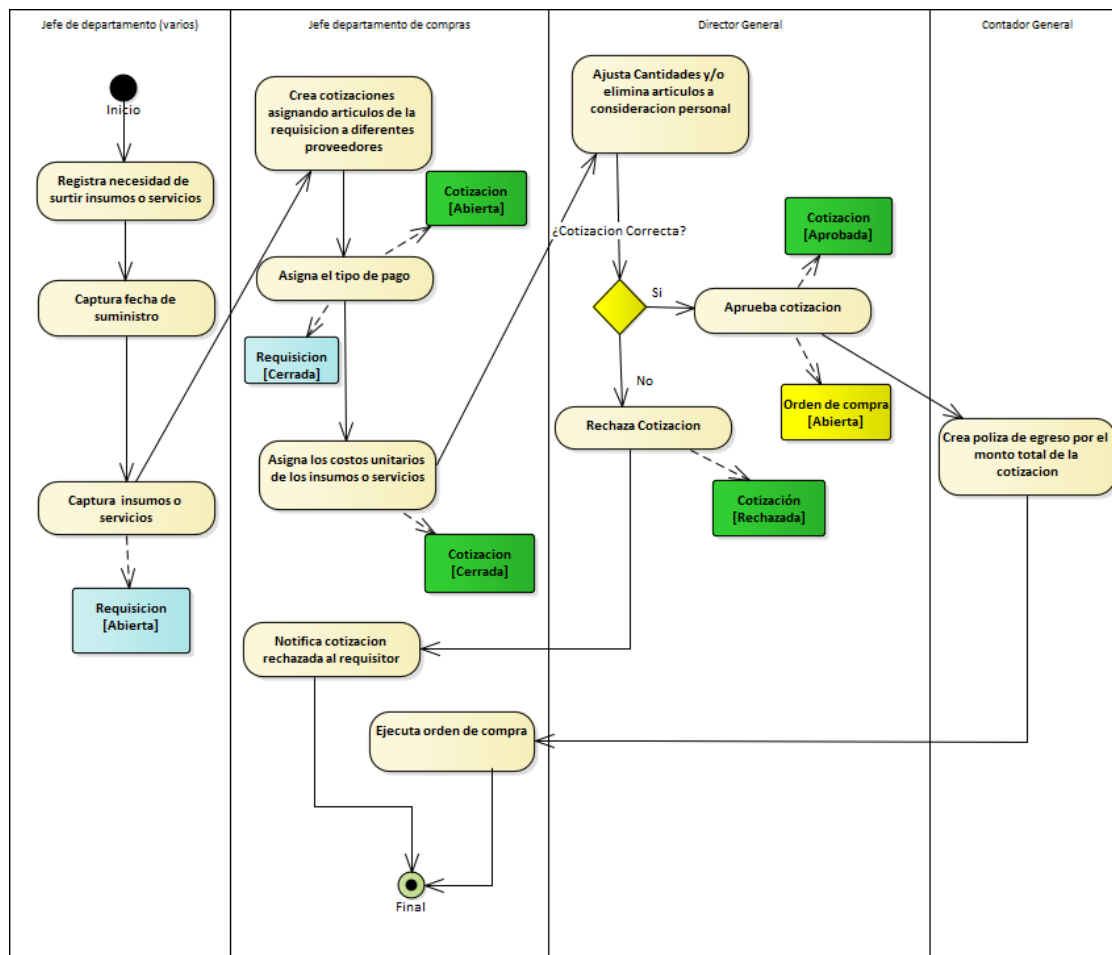


Figura 6. Modelado de negocio propuesto. Fuente: elaboración propia.

Modelo de casos de uso.

Para poder modelar los casos de uso, es necesario conocer los requerimientos funcionales del sistema, los cuales se enlistan de la siguiente manera:

1. **Requisito 1: Crear requisición de compra.** Los usuarios jefes de departamento (varios), deben tener la posibilidad de crear una requisición de compra y poder seleccionar dentro de la misma, los productos mediante una búsqueda por número de producto o por nombre del producto.
2. **Requisito 2: Crear cotización de compra.** El usuario jefe de departamento de compras, deberá recibir las requisiciones de compra con un estado “abierto” para poder crear una cotización de compra en donde tendrá la posibilidad de asignar el proveedor, el tipo de pago y la selección de los productos mediante una búsqueda por nombre y por número de producto.
3. **Requisito 3: Aprobar o rechazar cotización de compra.** El usuario director general, deberá recibir una lista con las cotizaciones de compra con un estado “abierto”, para poder seleccionar la cotización y hacer una revisión particular de la misma, el usuario podrá modificar las cantidades unitarias solicitadas, así como eliminar productos de la cotización para finalmente aprobar la cotización o en su defecto rechazarla en su totalidad.
4. **Requisito 4. Creación de póliza (documento) de egreso.** El usuario contador general, deberá recibir una lista de las cotizaciones de compra con un estado “aprobada”, para poder seleccionar la cotización y crear un documento que funcione como comprobante de gastos por el total de la cotización aprobada.

5. **Requisito 5. Consultas, altas, bajas y modificaciones de usuarios.** El usuario administrador del sistema, deberá contar con un menú en donde pueda consultar, inhabilitar, modificar e ingresar los usuarios del sistema mediante formularios.
6. **Requisito 6. Consultas, altas, bajas y modificaciones de productos.** El usuario administrador del sistema y el usuario jefe del departamento de compras, deberán contar con un menú en donde puedan consultar, inhabilitar, modificar e ingresar los productos al catálogo de productos del sistema mediante formularios

Diagrama de casos de uso.

Este tipo de diagrama es la representación más simple de la interacción entre un actor con el sistema (en donde un actor puede ser un usuario o un sistema); consta de actor(es), casos de uso, herencia y comunicación. La Figura 7, expone el diagrama de casos de uso, con los casos de uso que se desprenden de las historias de usuario, cada historia de usuario representa un requisito funcional de la aplicación, lo permite tener una mejor comprensión del sistema.







Figura 7. Diagrama de casos de uso de la aplicación web de compras. Fuente: elaboración propia.

Para comprender el diagrama representado en la ilustración 8, a continuación, se detallará como ejemplo del desarrollo el escenario del caso de uso “crear requisición” en conjunto con su diagrama de actividades de casos de uso, en ellos será posible apreciar la interacción entre el usuario que ejecuta el caso de uso y la aplicación o sistema. (Ver tabla 3 y figura 8).

Crear Requisición.

Tabla 3. Escenario caso de uso crear requisición. Fuente: elaboración propia.

Escenario.
<p> Básico. Básico 1</p> <ol style="list-style-type: none"> 1. El actor selecciona la opción crear requisición. 2. El sistema muestra los tipos de requisición. 3. El actor selecciona el tipo de requisición. 4. El sistema muestra folio de requisición y datos de la cabecera (nombre de quien solicita, empresa, departamento, fecha de requisición). 5. El sistema solicita fecha de suministro. 6. El actor ingresa fecha de suministro. 7. El actor selecciona la opción agregar artículo. 8. El sistema muestra ventana de búsqueda con 2 opciones búsqueda por nombre y búsqueda por número de artículo. 9. El actor selecciona tipo de búsqueda. 10. El actor ingresa información de búsqueda. 11. El sistema muestra información del artículo. 12. El sistema solicita cantidad del artículo. 13. El actor ingresa cantidad del artículo. 14. El actor repite proceso hasta completar lista de artículos. 15. El sistema muestra lista de artículos del pedido. 16. El actor guarda la operación. 17. El sistema almacena pedido de requisición con estado "Abierta". 18. El sistema muestra mensaje de operación guardada exitosamente.
<p> Alternativo. Alternativo 1</p> <ol style="list-style-type: none"> 1. La solicitud no contiene algún artículo <ol style="list-style-type: none"> 1.1 El sistema solicita al actor agregar por lo menos un artículo o cancelar la requisición. 1.2 El actor corrige seleccionando al menos un producto. 1.3 Pasa al punto 7 del evento principal.
<p> Alternativo. Alternativo 2</p> <ol style="list-style-type: none"> 2 El sistema no encuentra información del artículo. <ol style="list-style-type: none"> 2.1 El usuario regresa al punto 9 del evento primario.
<p> Alternativo. Alternativo 3</p> <ol style="list-style-type: none"> 3. El sistema muestra información incorrecta. <ol style="list-style-type: none"> 3.1 El actor cancela la operación. 3.2 El sistema descarta la información y regresa al tablero de control.
Restricciones.

Escenario.
<ul style="list-style-type: none"> Pre-condición. El actor debe mostrar las credenciales de usuario. <p>El actor debe mostrar las credenciales de usuario.</p>
<ul style="list-style-type: none"> Pre-condición. Debe de Existir un catálogo de productos. <p>Debe de Existir un catálogo de productos.</p>
<ul style="list-style-type: none"> Post-condición. Se genera documento de requisición con estado "Abierta".

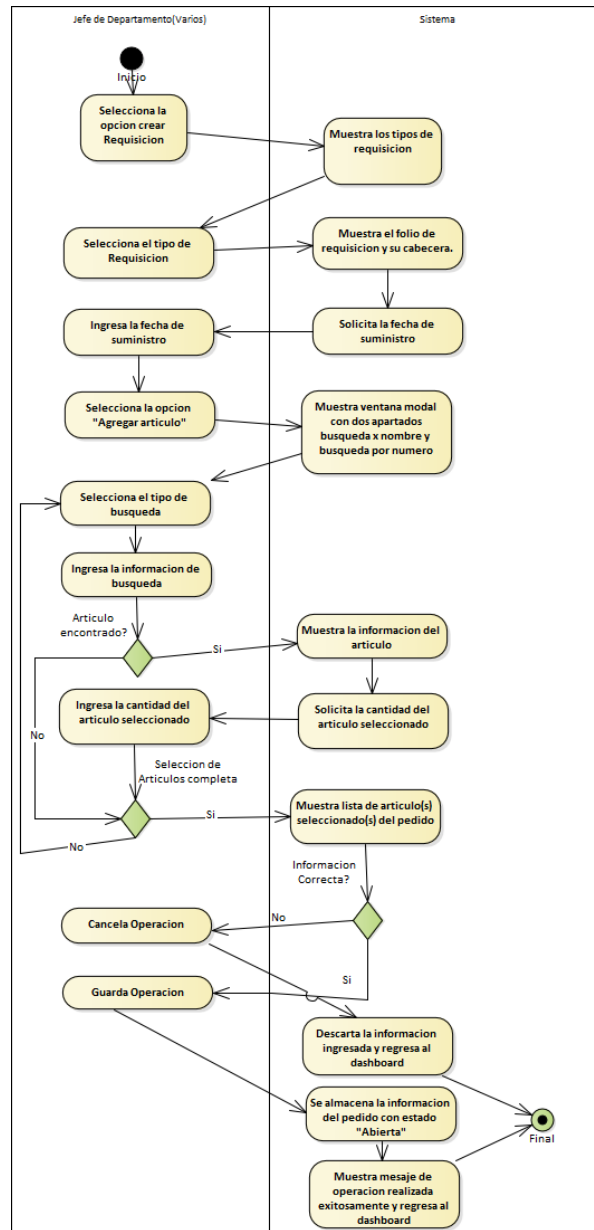


Figura 8. Diagrama de Actividades del caso de uso crear requisición. Fuente: elaboración propia.

Para complementar el caso de uso crear requisición, se utiliza el diagrama de estados, el cual representa la secuencia de estados por los que el objeto requisición pasa.

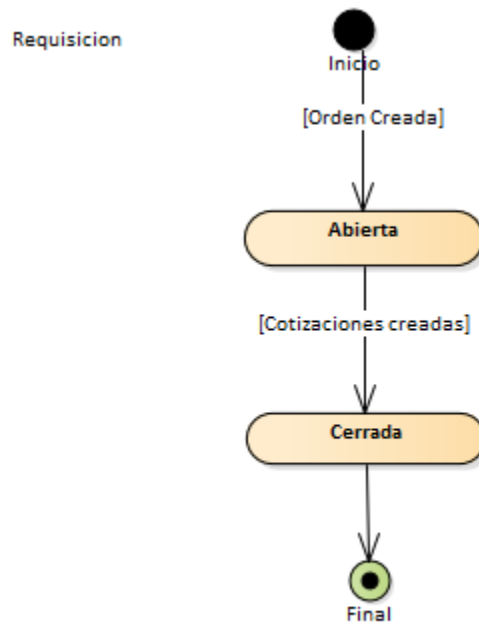


Figura 9. Diagrama de estados requisición. Fuente elaboración propia.

La figura 9 muestra, los estados por los que pasa la requisición, cuando la requisición es creada, esta inicia con un estado “abierta”, después cuando la(s) cotización(nes) que se desprenden de la misma se completan, la requisición toma el estado de “cerrada”.

De esta manera concluimos con la fase de inicio, para continuar con la fase de elaboración.

Fase de elaboración.

Dentro de esta fase se diseña la arquitectura del sistema mediante el siguiente artefacto:

- Modelo de clases.

Diagrama de clases.

El diagrama de clases muestra la estructura del sistema, es decir, expresa las clases o u objetos que comparten características en común y la relacionan entre sí.

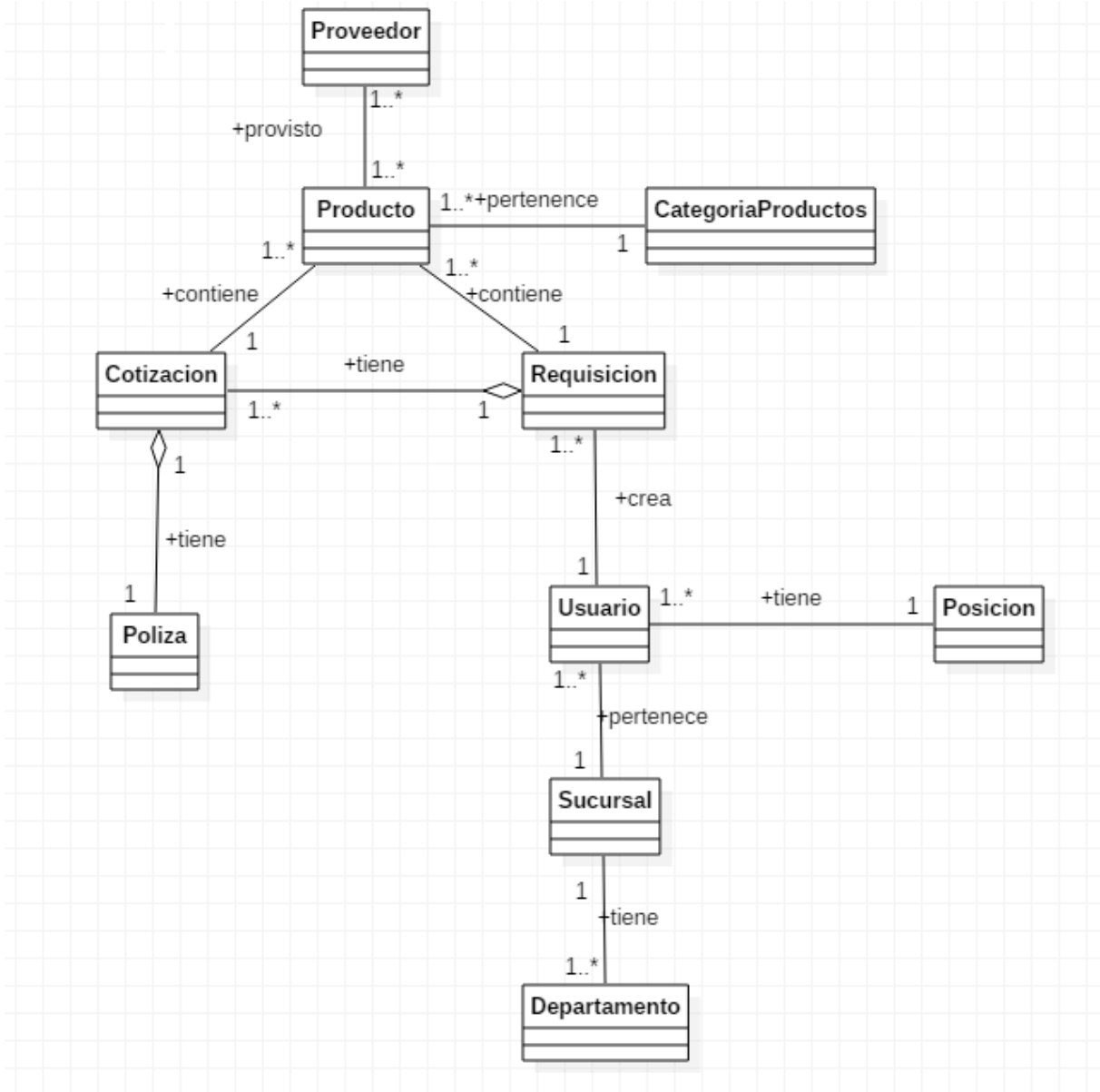


Figura 10. Diagrama de clases de la aplicación Web de compras. Fuente: elaboración propia.

La figura 10 muestra las clases que representan la aplicación, las cuales giran en torno a la clase principal Usuario. Cada usuario pertenece a una sucursal que tiene uno o más departamentos. Uno o más usuarios pueden tener una posición, además de crear una o más requisiciones. Una requisición puede tener una o más cotizaciones, una cotización puede tener una póliza de egreso; además, una requisición y una cotización pueden tener uno o muchos productos, uno o más productos pueden ser provistos por uno o muchos proveedores, finalmente uno o más productos pertenecen a una categoría de productos.

Arquitectura.

Para definir la arquitectura del software a construir, es necesario tomar en cuenta los requerimientos y las clases a utilizar, para este proyecto se utilizará el patrón modelo-vista-controlador en donde se distinguen tres

responsabilidades: la lógica del negocio- el modelo, la gestión de peticiones del usuario – el controlador y la presentación – la vista. (Figura 11)

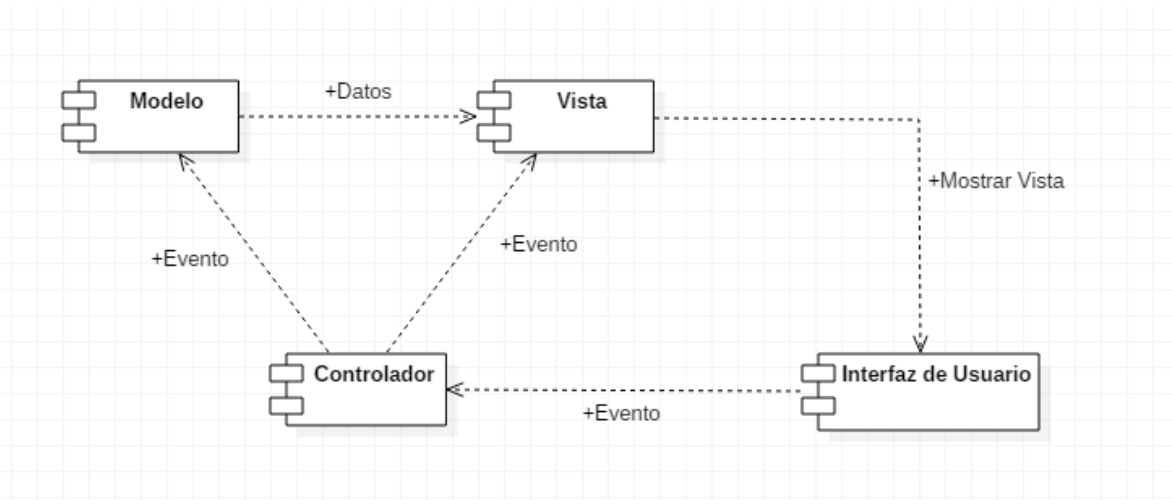


Figura 11. Patrón Modelo-Vista-Controlador. Fuente: elaboración propia.

Fase de construcción.

Esta es la fase se crea el producto final, añadiendo el software a la arquitectura. La arquitectura crece hasta convertirse en un sistema completo. Es aquí cuando los componentes faltantes y características del sistema son desarrolladas e integradas dentro del producto y todas características son enteramente probadas. La implementación de cada capa de la arquitectura Modelo-Vista-Controlador, se transforma en librerías de software, las cuales finalmente se visualizan como interfaces.

En la figura 12, se aprecia la interfaz de usuario “crear requisición”, correspondiente al caso de uso “crear requisición”, cuenta con una cabecera de información correspondiente a los datos del usuario que crea la requisición, botones para agregar artículos, cancelar y realizar la operación.

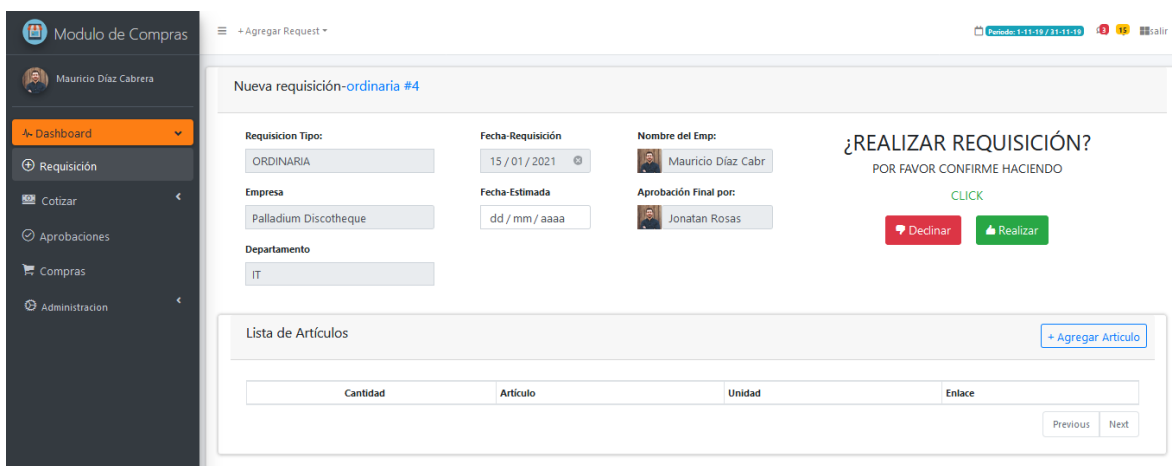


Figura 12. Interfaz de usuario Crear Requisición. Fuente: elaboración propia.

La Figura 13, muestra la interfaz de usuario de los casos de uso “búsqueda por nombre y búsqueda por número”, que extienden del caso de uso “crear requisición”.

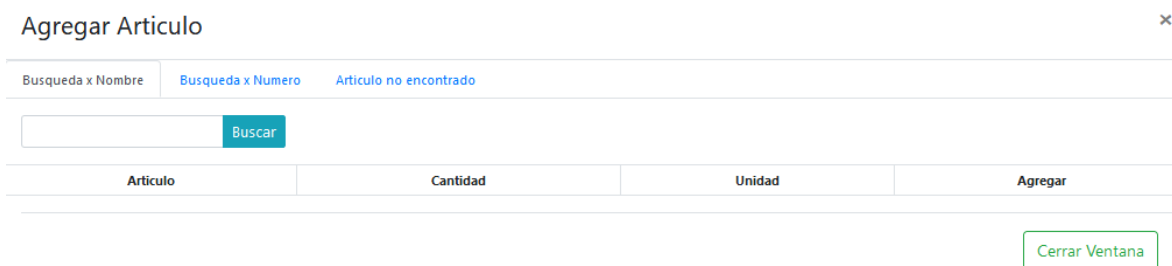


Figura 13. Interfaz buscar articulo con las opciones de búsqueda por nombre y por número.

Modelo de despliegue.

Para conocer la manera en que se va a implementar la aplicación, se utiliza el modelo de despliegue, el cual representa la disposición en físico de cada artefacto de software en los nodos.

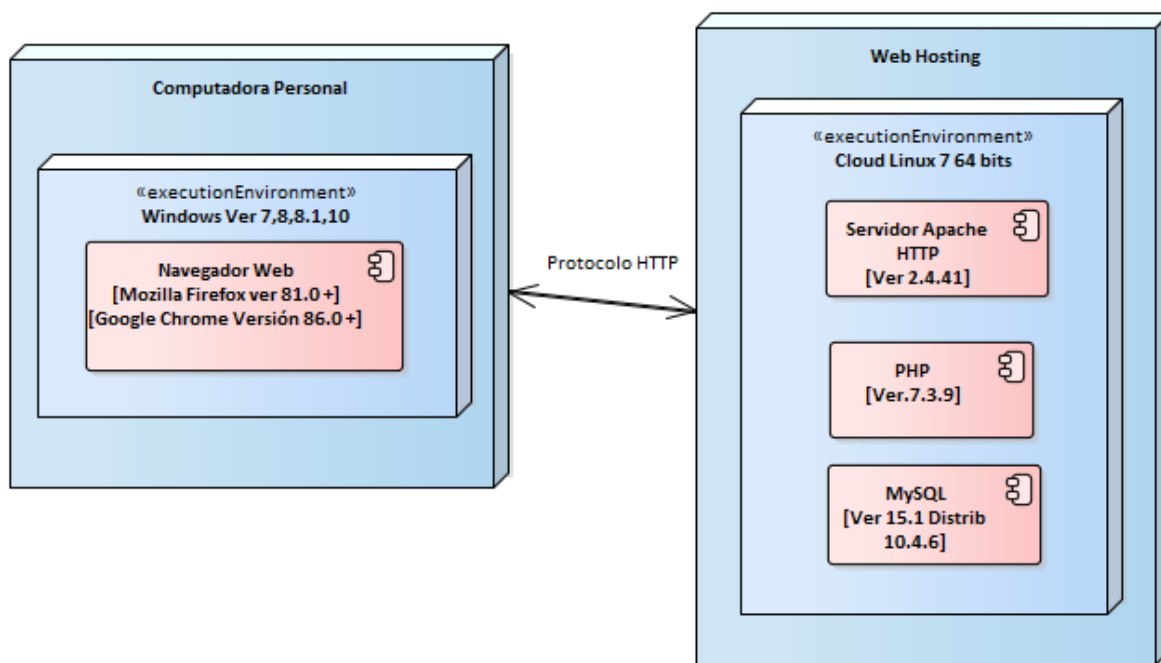


Figura 14. Diagrama de despliegue de la aplicación. Fuente: elaboración propia.

En la figura 14, se expone el diagrama de despliegue de la aplicación, en donde una computadora personal con sistema operativo Windows desde su versión 7 hasta su versión 10, establece una conexión a través de los navegadores Mozilla Firefox versión 81.0 o posterior y/o Google Chrome versión 86.0 o posterior, mediante el protocolo HTTP a un servidor Web Apache HTTP versión 2.4.41 y a un servidor de base de datos MySQL ver. 15.1, Dist. 10.4.6 alojados en un web Hosting con un sistema operativo CloudLinux 7 de 64 bits.

Conclusiones.

La propuesta del desarrollo de un sistema informático de compras reafirma el papel importante de las tecnologías de la información en el ámbito de las compras empresariales, ya que estas aportan nuevos métodos para el acceso y administración de dicha función. Una aplicación Web de compras facilitará el trabajo rutinario a la empresa y cumplirá satisfactoriamente mayores requerimientos de usuarios cada vez más involucrados con los avances tecnológicos. El principal objetivo de este documento ha sido la de utilizar la metodología RUP para el desarrollo de la aplicación, en donde los requisitos funcionales se transforman en casos de uso, que son la base de la aplicación y a través de estos se formula el inicio y el final de cada ciclo del desarrollo y de cada fase de la metodología. Por otra parte, UML es un lenguaje práctico para organizar, examinar la información de sistemas que permitan su mejor comprensión e implementación; así también como, explorar económicamente múltiples soluciones y dominar sistemas complejos.

En este documento se modelaron algunos diagramas con el fin de demostrar la comprensión de los objetos de un sistema de compras, el comportamiento y la comunicación entre los mismos, lográndolo de manera exitosa, lo cual demuestra que el Lenguaje Unificado de Modelado mantiene la capacidad de modelar un sistema eficientemente. Para poder exponer las fases de la metodología, se tomó como ejemplo un solo caso de uso para demostrar el desarrollo de la aplicación, desde su concepción como requisito hasta el despliegue de su interfaz de usuario codificada; con lo cual queda demostrado que la metodología del Proceso Unificado de Desarrollo se adapta perfectamente para desarrollar aplicaciones con orientación Web.

Créditos.

Tecnológico Nacional de México / I. T. de Acapulco – Maestría en Sistemas Computacionales

Agradecimientos.

Un sincero agradecimiento al Tecnológico Nacional de México campus Acapulco y campus Chilpancingo, en especial la División de Estudios de Posgrado e Investigación, dirigida por el Dr. Eduardo de la Cruz Gámez, por su apoyo y la excelente asistencia técnica que fue brindada para la redacción de este artículo. También se extiende un agradecimiento al Consejo Nacional de Ciencia y Tecnología por su soporte para este desarrollo.

Referencias Bibliográficas.

Framework, M. P. o. G. (2020). Diagramas UML. Retrieved from <https://diagramasuml.com/>

González, A. H. (2004). *Aplicación del Proceso Unificado de Desarrollo a proyectos de software.*

IBM. (1998). *Rational Unified Process: Best Practices for Software development Teams.* In.

James Rumbaugh, I. J., Grady Booch. (2000). *El lenguaje Unificado de Modelado Manual de Referencia* (P. E. S.A. Ed.). España: Addison Wesley.

Torossi, A. U. S. G. (2004). *El Proceso Unificado de Desarrollo de Software.* In. Retrieved from <http://dsc.itmorelia.edu.mx/~jcolivares/courses/pm10a/rup.pdf>

Yourdon, E. (1993). *Modern Structured Analysis.* Mexico: Prentice-Hall Hispanoamericana.

Información de Autores.



Mauricio Díaz Cabrera es Ingeniero por el Tecnológico Nacional de México campus Acapulco; actualmente se desempeña como estudiante de la Maestría en Sistemas Computacionales.



Rafael Hernández Reyna es Maestro en Tecnologías de la Información; actualmente se desempeña como docente de la Maestría en Sistemas Computacionales en el Tecnológico Nacional de México campus Acapulco en la línea de investigación: Desarrollo de Sistemas Inteligentes.



Mercedes Hernández de la Cruz es Doctora en Educación con Tecnologías del Aprendizaje y del Conocimiento; actualmente se desempeña como docente de la carrera de Ingeniería Informática en el Tecnológico Nacional de México, Campus Chilpancingo.



Miriam Martínez Arroyo es Doctora en Ciencias Computacionales; actualmente se desempeña como docente de la Maestría en Sistemas Computacionales en el Tecnológico Nacional de México campus Acapulco en la línea de investigación: Desarrollo de Sistemas Inteligentes. Cuenta con perfil PROMEP, miembro del padrón de investigadores del estado de Guerrero.