Fundación universitaria Konrad Lorenz

Andrés Peña Mantilla cód:

Samuel Villanueva cód.: 506122009

David Guavita cód.: 5061632025

Jhon Jairo Chaparro Ceballes cód: 506152042

Ingeniería de software II

Bogotá D,C

**Tabla de contenido**

[Historias de usuarios y Diagramas de casos de uso del software CRM 4](#_Toc478159153)

[1. Requerimientos funcionales 4](#_Toc478159154)

[*1.1.* *Requerimientos de creación de contactos* 4](#_Toc478159155)

[1.1.1. Caso de uso Creación de contactos 4](#_Toc478159156)

[*1.2.* *Historia de usuarios de crear de contactos* 4](#_Toc478159157)

[*1.3.* *Historia de usuarios de editar de contactos* 5](#_Toc478159158)

[*1.4.* *Historia de usuarios de leer de contactos* 5](#_Toc478159159)

[2. Requerimientos de Administración y gerencia de contactos 6](#_Toc478159160)

[*2.1.* *Caso de uso Creación de creación de contactos* 6](#_Toc478159161)

[*2.2.* *Historia de usuarios creación vendedor* 6](#_Toc478159162)

[*2.3.* *Historia de usuarios eliminar vendedor* 7](#_Toc478159163)

[*2.4.* *Historia de usuarios editar vendedor* 7](#_Toc478159164)

[3. Requerimientos de Control y Gestión de oportunidades 8](#_Toc478159165)

[*3.1.* *Diagrama de Caso de uso Gestión de Catálogo de productos* 8](#_Toc478159166)

[*3.2.* *Historia de usuarios creación de oportunidades* 8](#_Toc478159167)

[*3.3.* *Historia de usuarios edición de oportunidades* 9](#_Toc478159168)

[*3.4.* *Historia de usuarios leer oportunidades* 9](#_Toc478159169)

[4. Requerimientos de Gestión de Catálogo de productos 10](#_Toc478159170)

[*4.1.* *Diagrama de Caso de uso Gestión de Catálogo de productos* 10](#_Toc478159171)

[*4.2.* *Historia de usuarios creación de producto* 10](#_Toc478159172)

[*4.3.* *Historia de usuarios edición de producto* 11](#_Toc478159173)

[*4.4.* *Historia de usuarios eliminación de producto* 11](#_Toc478159174)

[*4.5.* *Historia de usuarios listar producto* 12](#_Toc478159175)

[5. Requerimientos de Administración y control de actividades comerciales 13](#_Toc478159176)

[*5.1.* *Diagrama de Caso de uso Creación de Administración y control de actividades comerciales* 13](#_Toc478159177)

[6. Requerimientos de Control de Clientes por línea de productos 16](#_Toc478159178)

[*6.1.* *Diagrama de Caso de uso Control de Clientes por línea de productos* 16](#_Toc478159179)

[DIAGRAMA DE CLASES DE ENTIDADES (ANEXO 2) 20](#_Toc478159180)

[21](#_Toc478159181)

[DIAGRAMAS DE SECUENCIA (ANEXO 3) 22](#_Toc478159182)

[DIAGRAMA DE DESPLIEGUE (ANEXO 3) 24](#_Toc478159183)

[GLOSARIO DE TÉRMINOS 26](#_Toc478159184)

[(ANEXO 4) 26](#_Toc478159185)

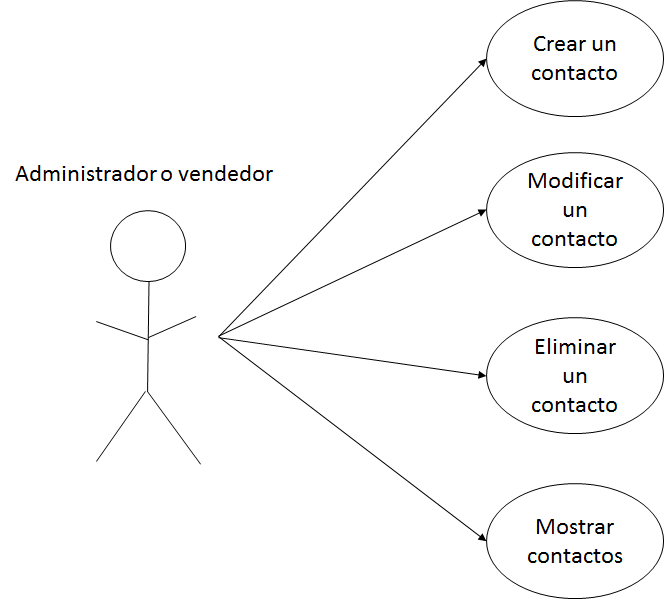
# Historias de usuarios y Diagramas de casos de uso del software CRM

A continuación se presenta la documentación de los casos de uso del sistema de información CRM

# Requerimientos funcionales

## *Requerimientos de creación de contactos*

## Caso de uso Creación de contactos



## *Historia de usuarios de crear de contactos*

|  |
| --- |
| **Nombre: CU1** Registrar contactos |
| **Actor(es):** Adinistrador o vendedor |
| **Descripción:** Se crean cuentas de contactos |
| **Precondiciones:** Tener una sesión iniciada como administrador |
| **Flujo Normal:**  El actor ingresa al módulo del contacto.   1. aparece un formulario web 2. el administrador sumista los datos requeridos por el sistema 3. el usuario guarda la información 4. En la tabla crea un nuevo registro con los datos del contacto. |
| **Flujo Alternativo:Si el usuario no coloca los campos requeridos el sistema el avisa de edicho error** |
| **Post Condición:**  **1.** Contacto creado |

## *Historia de usuarios de editar de contactos*

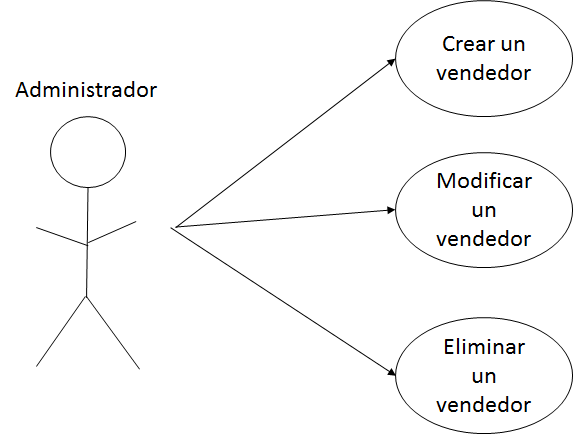
|  |
| --- |
| **Nombre: CU2** Editar contactos |
| **Actor(es):** Administrador o vendedor |
| **Descripción:** Se modifican atributos de los contactos excepto el PK. |
| **Precondiciones:**  **1.** El contacto debe existir. |
| **Flujo Normal:**   1. El actor ingresa al módulo del contacto. 2. En la tabla busca el contacto por código. 3. Edita y modifica el registro de la tabla. |
| **Flujo Alternativo:** |
| **Post Condición:**  **1.** Contacto modificado |

## *Historia de usuarios de leer de contactos*

|  |
| --- |
| **Nombre: CU3** Mostrar los contactos |
| **Actor(es):** Vendedor o Administrador |
| **Descripción:** El sistema muestra los contactos existentes |
| **Precondiciones:**  1. Deben existir contactos |
| **Flujo Normal:**  El usuario ingresa al módulo vendedor  **2.** El sistema muestra los contactos respectivos de cada vendedor. |
| **Flujo Alternativo:**  El usuario ingresa al módulo administrador.  **2.** El sistema muestra los contactos de toda la empresa. |
| **Post Condición:**  **1.** Contactos mostrados |

# Requerimientos de Administración y gerencia de contactos

## *Caso de uso Creación de creación de contactos*



## *Historia de usuarios creación vendedor*

|  |
| --- |
| **Nombre:CU4** Creación de Vendedores |
| **Actor(es):** Administrador |
| **Descripción:** Se crean cuentas de vendedores para ingreso al sistema por su respectivo rol, directamente en la tabla de usuarios. |
| **Precondiciones:** |
| **Flujo Normal:**  El actor ingresa al módulo de administración.  En la tabla usuarios crea un nuevo registro con los datos del vendedor.  **3.** Se le asigna usuario y clave. |
| **Flujo Alternativo:** |
| **Post Condición:**  **1.** Cuenta creada. |

## *Historia de usuarios eliminar vendedor*

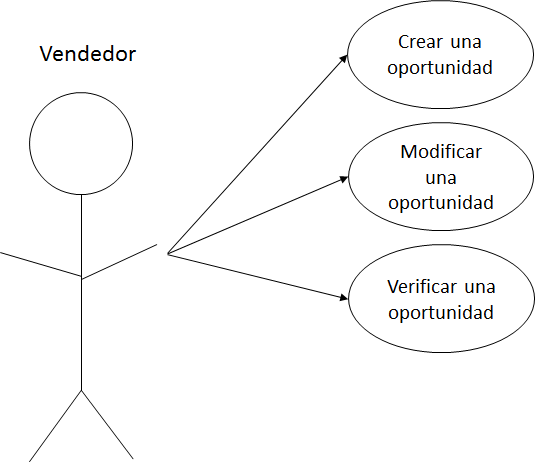
|  |
| --- |
| **Nombre: CU5** Elimina registro de vendedor |
| **Actor(es):** Administrador |
| **Descripción:** Se elimina el registro del vendedor determinado en la tabla usuario. |
| **Precondiciones:**  1. El vendedor debe existir. |
| **Flujo Normal:**  El actor ingresa al módulo de administración.  En la tabla usuarios busca el vendedor por código.  **3.** Selecciona y elimina el registro de la tabla. |
| **Flujo Alternativo:**  **1.** El actor desiste de la eliminación. |
| **Post Condición:**  **1.** Cuenta eliminada. |

## *Historia de usuarios editar vendedor*

|  |
| --- |
| **Nombre: CU6** Modificar registro de vendedor |
| **Actor(es):** Administrador |
| **Descripción:** Se modifican atributos en la tabla usuario a excepción del campo PK. |
| **Precondiciones:**  **1.** El vendedor debe existir |
| **Flujo Normal:**  El actor ingresa al módulo de administración.  En la tabla usuarios busca el vendedor por código.  **3.** Edita y modifica el registro de la tabla. |
| **Flujo Alternativo:** |
| **Post Condición:**  **1.** Cuenta modificada |

# Requerimientos de Control y Gestión de oportunidades

## *Diagrama de Caso de uso Gestión de Catálogo de productos*



## *Historia de usuarios creación de oportunidades*

|  |
| --- |
| **Nombre: CU7**Registrar oportunidades |
| **Actor(es):** Vendedor |
| **Descripción:** Se crean las oportunidades directamente en la tabla de seguimiento. |
| **Precondiciones:** |
| **Flujo Normal:**   1. El actor ingresa al módulo creación de oportunidades. 2. Llena la información requerida por el sistema 3. Guarda la información del sistema 4. En la tabla crea un nuevo registro con los datos de la oportunidad. |
| **Flujo Alternativo:** si el usuario ingresa mal algunos de los campos el sistema le mostrara el campo que tiene el errorpara que sea corregido |
| **Post Condición:**  **1.** tener una sesión activa en la plataforma |

## *Historia de usuarios edición de oportunidades*

|  |
| --- |
| **Nombre: CU8** Modificación oportunidades |
| **Actor(es):** Vendedor |
| **Descripción:** Se crean las oportunidades directamente en la tabla de seguimientos. |
| **Precondiciones:** |
| **Flujo Normal:**   1. El actor ingresa al módulo de oportunidades. 2. Selecciona la oportunidad a modificar 3. Realiza los cambios necesarios en la oportunidad 4. Guarda la oportunidad con las actualizaciones requeridas 5. En la tabla crea un nuevo registro con los datos de la oportunidad. |
| **Flujo Alternativo:** si el usuario ingresa mal algunos de los campos el sistema le mostrara el campo que tiene el errorpara que sea corregido |
| **Post Condición:**  **1.** tener una sesión activa en la plataforma |

## *Historia de usuarios leer oportunidades*

|  |
| --- |
| **Nombre: CU9** Mostrar oportunidades |
| **Actor(es):** Vendedor o Administrador |
| **Descripción:** El sistema muestra el catálogo de productos. |
| **Precondiciones:**  1. tener oportunidades registradas |
| **Flujo Normal:**   1. El usuario ingresa al módulo de oportunidades 2. Selecciona ver las oportunidades 3. El sistema lista las oportunidades del vendedor |
| **Flujo Alternativo:** N/A |
| **Post Condición:**  **1.** Catálogo de productos mostrado |

# Requerimientos de Gestión de Catálogo de productos

## *Diagrama de Caso de uso Gestión de Catálogo de productos*



## *Historia de usuarios creación de producto*

|  |
| --- |
| **Nombre: CU18** Registrar productos |
| **Actor(es):** Administrador o vendedor |
| **Descripción:** Se crean un producto directamente en la tabla |
| **Precondiciones:** |
| **Flujo Normal:**  El actor ingresa al módulo del vendedor.  **2.** En la tabla crea un nuevo registro con los datos del producto. |
| **Flujo Alternativo:** |
| **Post Condición:**  **1.** Producto creado |

## *Historia de usuarios edición de producto*

|  |
| --- |
| **Nombre: CU19** Editar productos |
| **Actor(es):** Administrador o vendedor |
| **Descripción:** Se modifican atributos del producto excepto el PK. |
| **Precondiciones:**  **1.** El producto debe existir. |
| **Flujo Normal:**  El actor ingresa al módulo del vendedor.  En la tabla busca el producto por código.  **3.** Edita y modifica el registro de la tabla. |
| **Flujo Alternativo:** |
| **Post Condición:**  **1.** Producto modificado |

## *Historia de usuarios eliminación de producto*

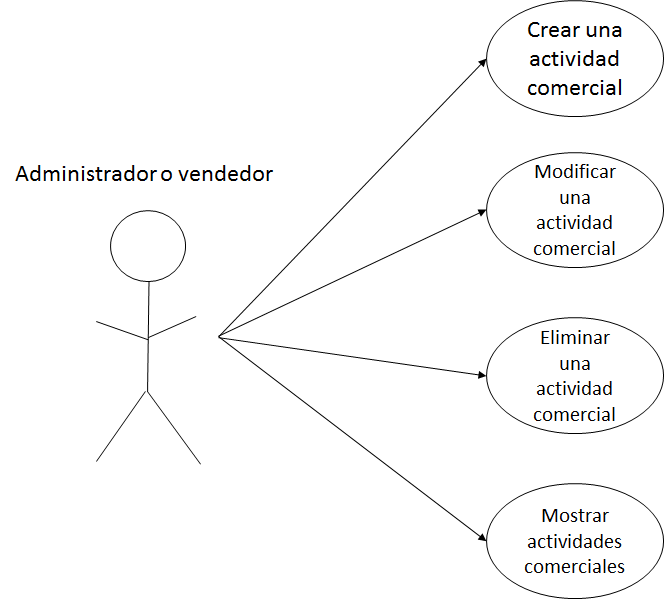
|  |
| --- |
| **Nombre: CU20** Eliminar un producto |
| **Actor(es):** Administrador o vendedor |
| **Descripción:** Se elimina el registro del producto determinado. |
| **Precondiciones:**  **1.** El producto debe existir. |
| **Flujo Normal:**  El actor ingresa al módulo del vendedor.  En la tabla busca el producto por código.  **3.** Selecciona y elimina el registro de la tabla. |
| **Flujo Alternativo:**  **1.** El actor desiste de la eliminación. |
| **Post Condición:**  **1.** El producto se ha eliminado |

## *Historia de usuarios listar producto*

|  |
| --- |
| **Nombre: CU21** Mostrar catálogo de productos |
| **Actor(es):** Vendedor o Administrador |
| **Descripción:** El sistema muestra el catálogo de productos. |
| **Precondiciones:**  1. Deben existir productos. |
| **Flujo Normal:**  El usuario ingresa al módulo vendedor  **2.** El sistema muestra el catálogo de productos. |
| **Flujo Alternativo:** |
| **Post Condición:**  **1.** Catálogo de productos mostrado |

# Requerimientos de Administración y control de actividades comerciales

## *Diagrama de Caso de uso Creación de Administración y control de actividades comerciales*



|  |
| --- |
| **Nombre: CU8** Registrar actividades comerciales |
| **Actor(es):** Administrador o vendedor |
| **Descripción:** Se crean actividades comerciales directamente en la tabla |
| **Precondiciones:** |
| **Flujo Normal:**  El actor ingresa al módulo del vendedor.  **2.** En la tabla crea un nuevo registro con los datos de la actividad comercial. |
| **Flujo Alternativo:** |
| **Post Condición:**  **1.** Actividad comercial creada |

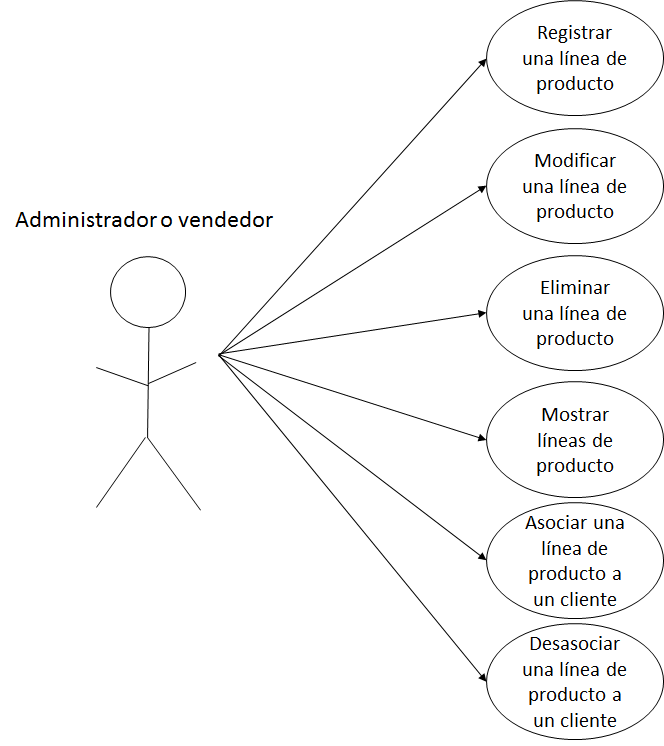
|  |
| --- |
| **Nombre: CU9** Editar actividades comerciales |
| **Actor(es):** Administrador o vendedor |
| **Descripción:** Se modifican atributos de la actividad comercial excepto el PK. |
| **Precondiciones:**  **1.** La actividad comercial debe existir. |
| **Flujo Normal:**  El actor ingresa al módulo del vendedor.  En la tabla busca la actividad comercial por código.  **3.** Edita y modifica el registro de la tabla. |
| **Flujo Alternativo:** |
| **Post Condición:**  **1.** Actividad comercial modificada |

|  |
| --- |
| **Nombre: CU10** Eliminar una actividad comercial |
| **Actor(es):** Administrador o vendedor |
| **Descripción:** Se elimina el registro de la actividad comercial determinado. |
| **Precondiciones:**  **1.** La actividad comercial debe existir. |
| **Flujo Normal:**  El actor ingresa al módulo del vendedor.  En la tabla busca la actividad comercial por código.  **3.** Selecciona y elimina el registro de la tabla. |
| **Flujo Alternativo:**  **1.** El actor desiste de la eliminación. |
| **Post Condición:**  **1.** Actividad comercial eliminada |

|  |
| --- |
| **Nombre: CU11** Mostrar las actividades comerciales |
| **Actor(es):** Vendedor o Administrador |
| **Descripción:** El sistema muestra las actividades comerciales existentes |
| **Precondiciones:**  1. Deben existir actividades comerciales. |
| **Flujo Normal:**  El usuario ingresa al módulo vendedor  **2.** El sistema muestra las actividades comerciales. |
| **Flujo Alternativo:** |
| **Post Condición:**  **1.** Actividades comerciales mostradas |

# Requerimientos de Control de Clientes por línea de productos

## *Diagrama de Caso de uso Control de Clientes por línea de productos*



|  |
| --- |
| **Nombre: CU12** Registrar líneas de productos |
| **Actor(es):** Administrador o vendedor |
| **Descripción:** Se crean líneas de producto directamente en la tabla de . |
| **Precondiciones:** |
| **Flujo Normal:**  El actor ingresa al módulo del vendedor.  **2.** En la tabla crea un nuevo registro con los datos de la línea de producto. |
| **Flujo Alternativo:** |
| **Post Condición:**  Línea de producto creada |

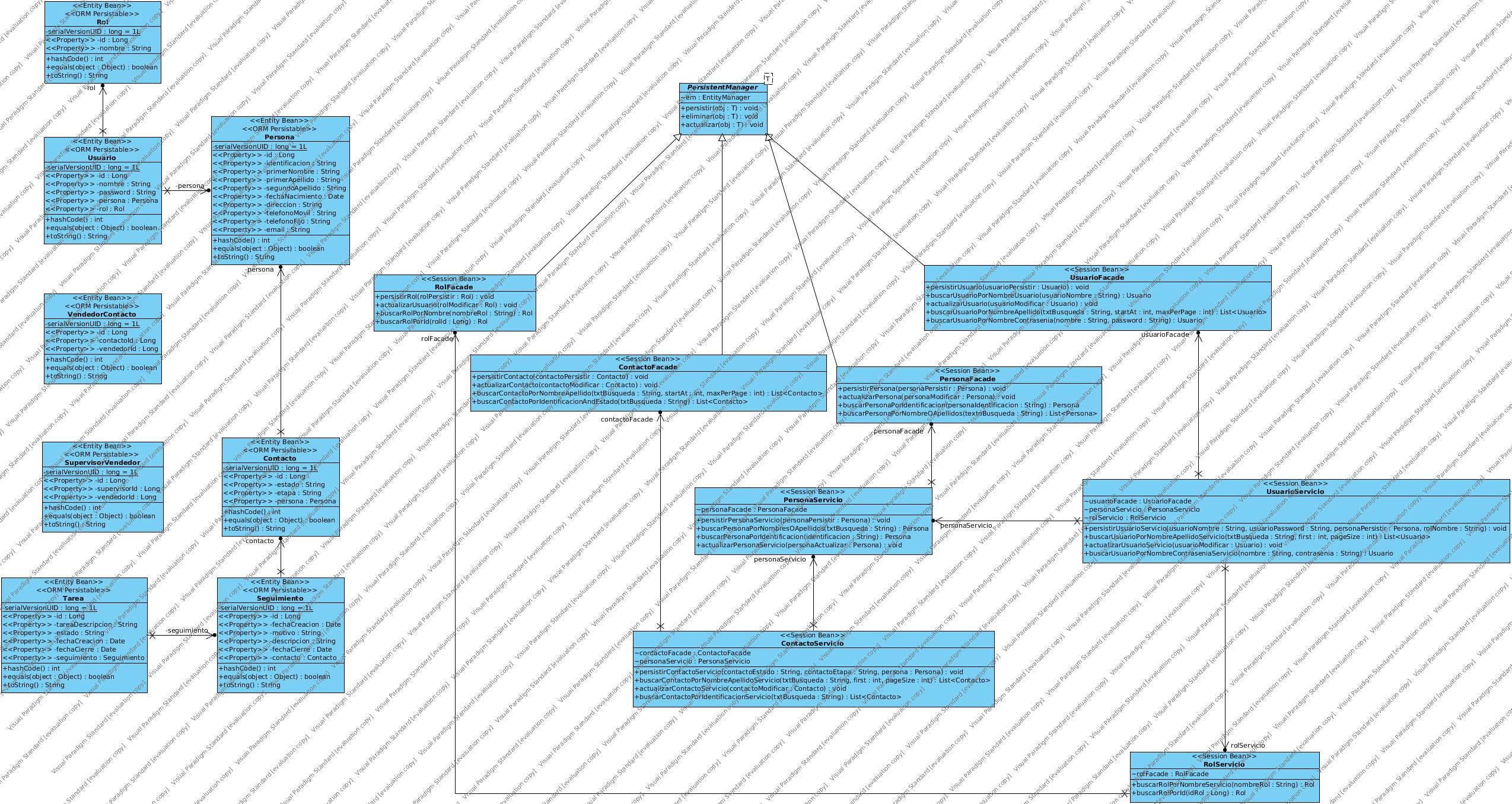
|  |
| --- |
| **Nombre: CU13** Editar líneas de productos |
| **Actor(es):** Administrador o vendedor |
| **Descripción:** Se modifican atributos de la línea de producto excepto el PK. |
| **Precondiciones:**  **1.** La línea de producto debe existir. |
| **Flujo Normal:**  El actor ingresa al módulo del vendedor.  En la tabla busca la línea de producto por código.  **3.** Edita y modifica el registro de la tabla. |
| **Flujo Alternativo:** |
| **Post Condición:**  **1.** Línea de producto modificada |
| **Nombre: CU14** Eliminar una línea de producto |
| **Actor(es):** Administrador o vendedor |
| **Descripción:** Se elimina el registro de la línea de producto determinado. |
| **Precondiciones:**  **1.** La línea de producto debe existir. |
| **Flujo Normal:**  El actor ingresa al módulo del vendedor.  En la tabla busca la línea de producto por código.  **3.** Selecciona y elimina el registro de la tabla. |
| **Flujo Alternativo:**  **1.** El actor desiste de la eliminación. |
| **Post Condición:**  **1.** Línea de producto eliminada |

|  |
| --- |
| **Nombre: CU15** Mostrar las líneas de productos |
| **Actor(es):** Vendedor o Administrador |
| **Descripción:** El sistema muestra las líneas de productos existentes |
| **Precondiciones:**  1. Deben existir líneas de productos. |
| **Flujo Normal:**  El usuario ingresa al módulo vendedor  **2.** El sistema muestra las líneas de productos. |
| **Flujo Alternativo:** |
| **Post Condición:**  **1.** Actividades comerciales mostradas |

|  |
| --- |
| **Nombre: CU16** Asociar una línea de producto a un cliente |
| **Actor(es):** Vendedor |
| **Descripción:** El vendedor asocia una línea de producto por el requerimiento de un cliente a su usuario. |
| **Precondiciones:**  El cliente tenga cuenta.  **2.** La línea de producto exista. |
| **Flujo Normal:**  El actor ingresa al módulo vendedor.  El actor asocia al cliente con una línea del producto.  **3.** El sistema adhiere la línea del producto al usuario requerido. |
| **Flujo Alternativo:** |
| **Post Condición:**  **1.** El cliente tiene asociado una línea de producto. |

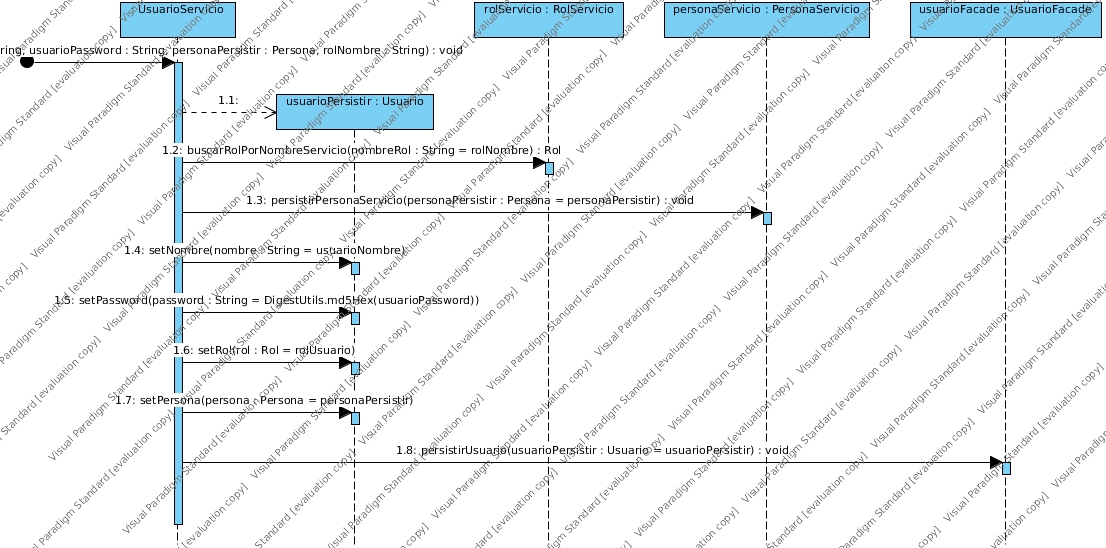
|  |
| --- |
| **Nombre: CU17** Eliminar asociación de una línea de producto a un cliente |
| **Actor(es):** Vendedor |
| **Descripción:** El vendedor desasocia una línea de producto por el requerimiento de un cliente a su usuario. |
| **Precondiciones:**  **1.** El cliente cuente con la asociación. |
| **Flujo Normal:**  El actor ingresa al módulo vendedor.  El actor desasocia al cliente con una línea del producto.  **3.** El sistema elimina la línea del producto al usuario requerido. |
| **Flujo Alternativo:** |
| **Post Condición:**  **1.** El cliente no tiene asociado una línea de producto requerida. |

# DIAGRAMA DE CLASES DE ENTIDADES (ANEXO 2)

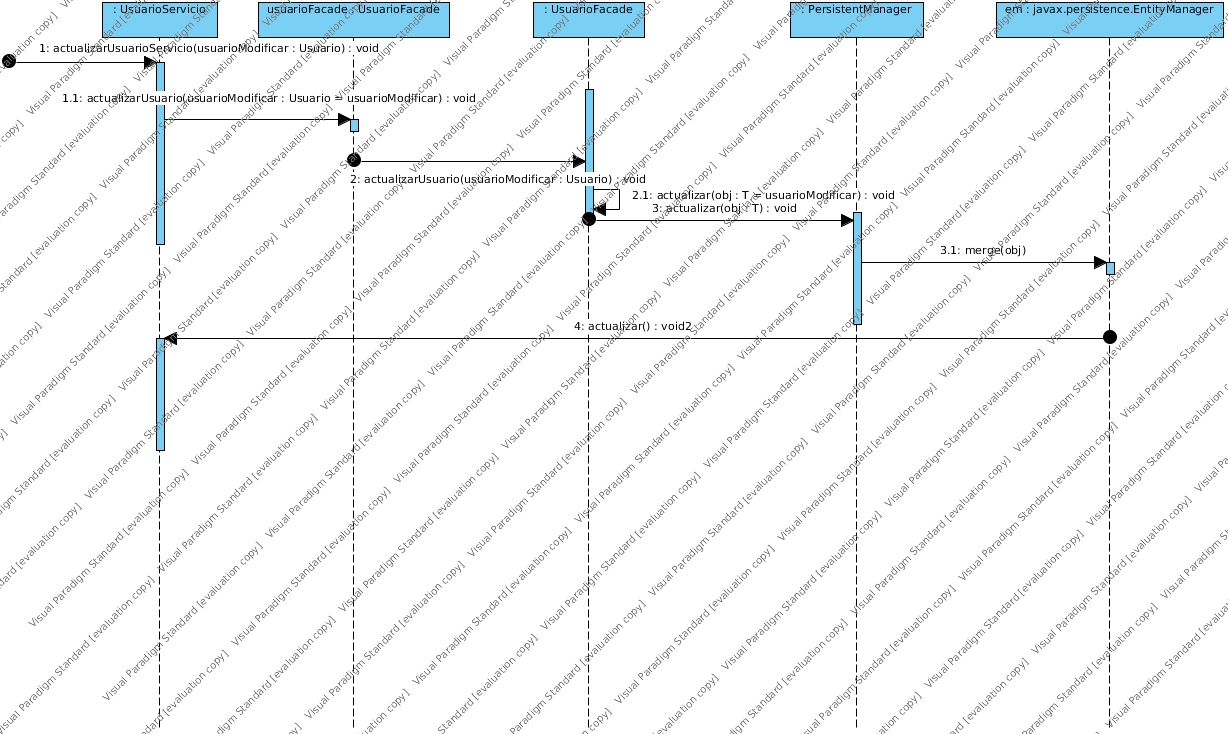


# DIAGRAMAS DE SECUENCIA (ANEXO 3)

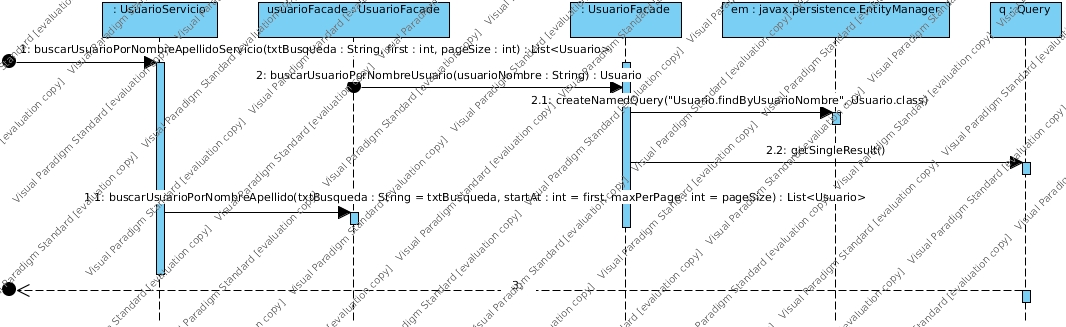
1. Diagrama secuencia de Creación de contactos



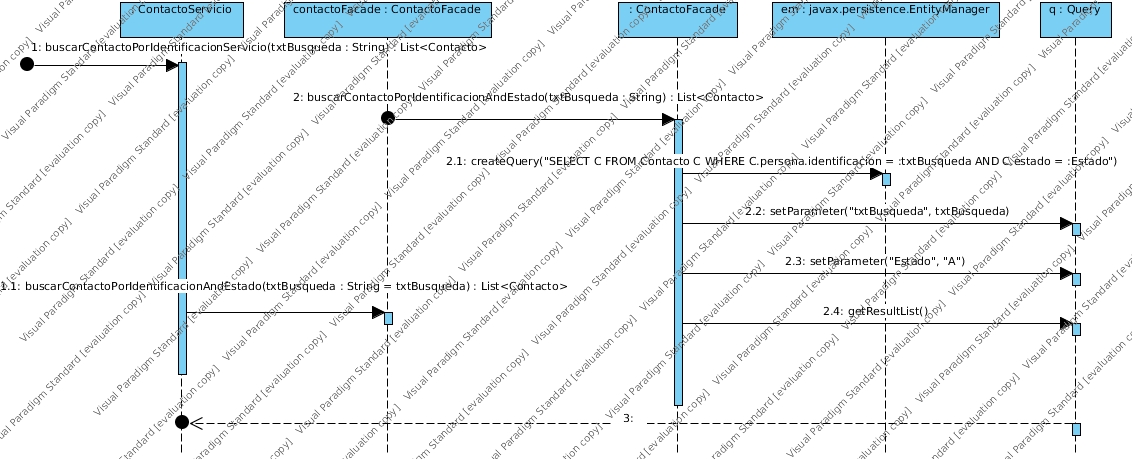
1. Diagrama secuencia de Actualización de contactos



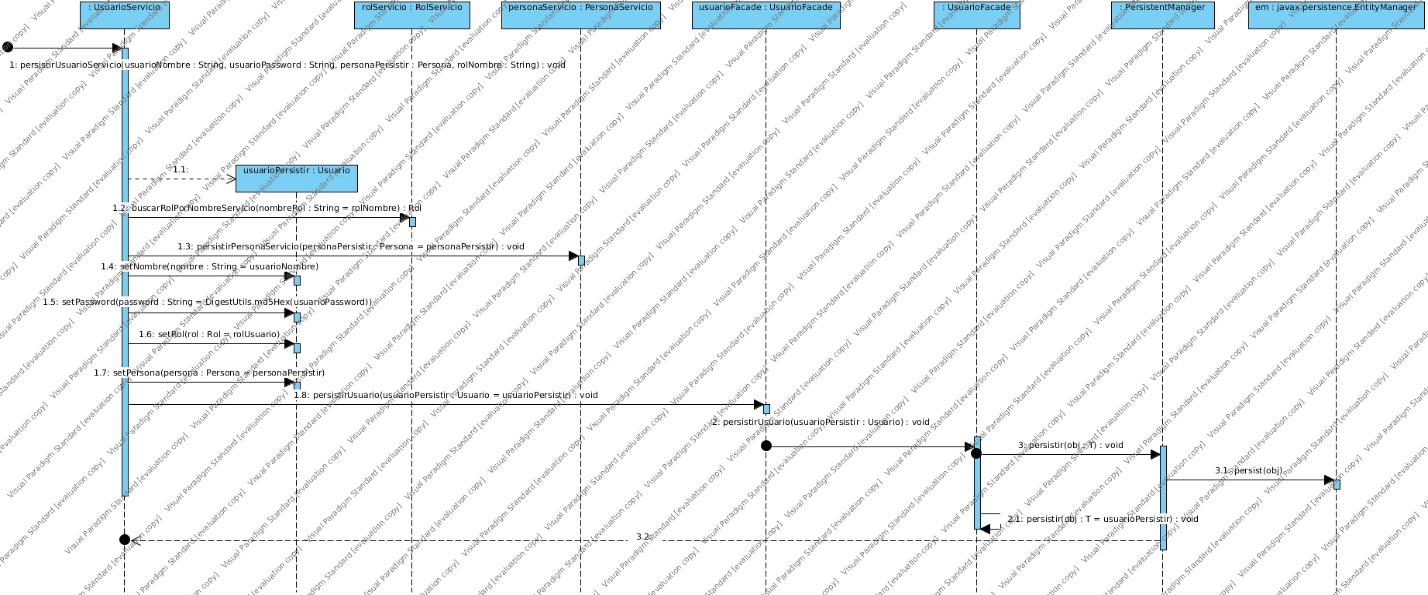
1. Diagrama secuencia de búsqueda de contactos por nombre



1. Diagrama secuencia de búsqueda de contactos por identificador



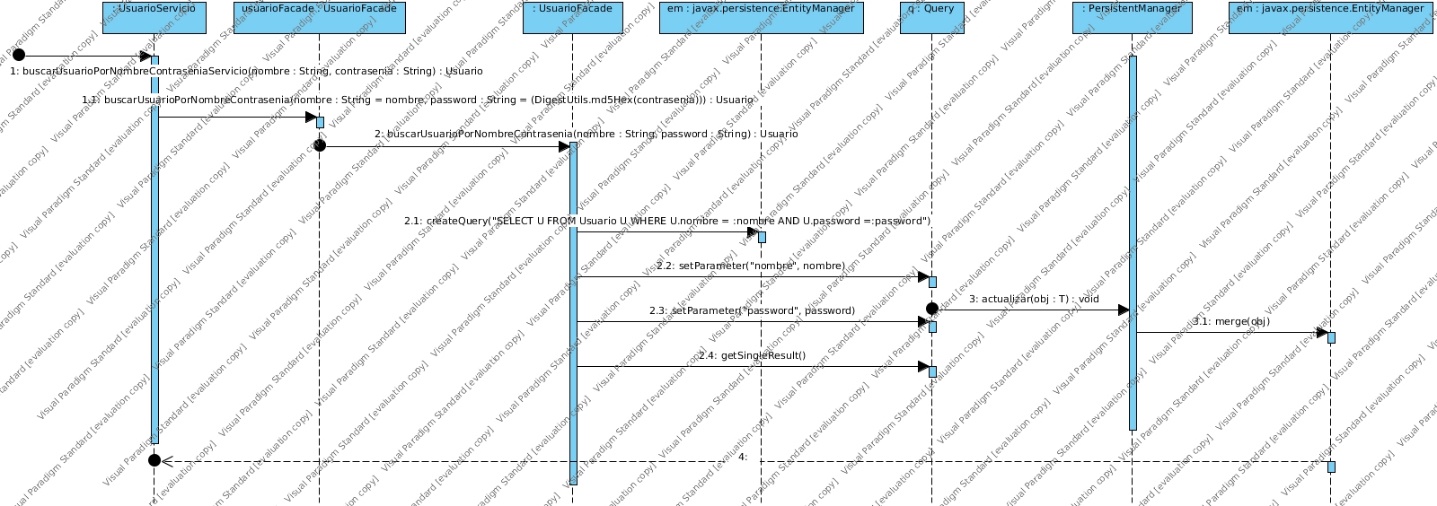
1. Diagrama secuencia de Creación de Usuario



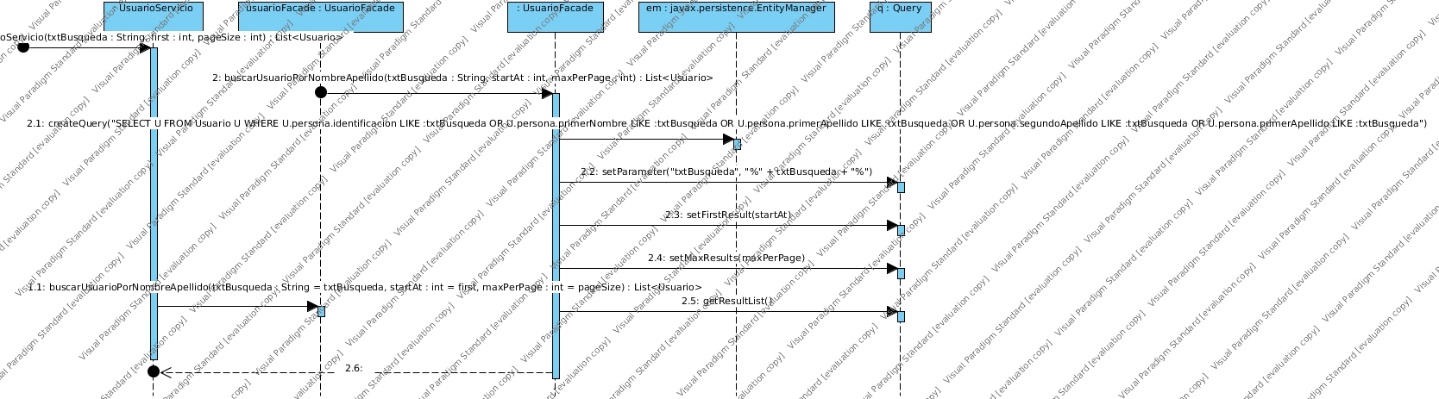
1. Diagrama secuencia de Edición de usuario



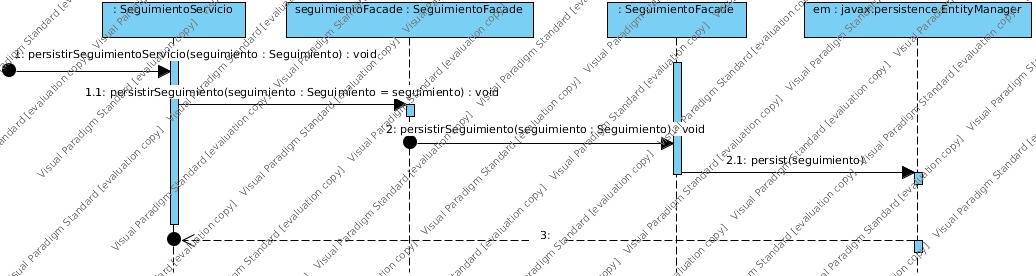
1. Diagrama secuencia de Búsqueda por Contraseña usuario



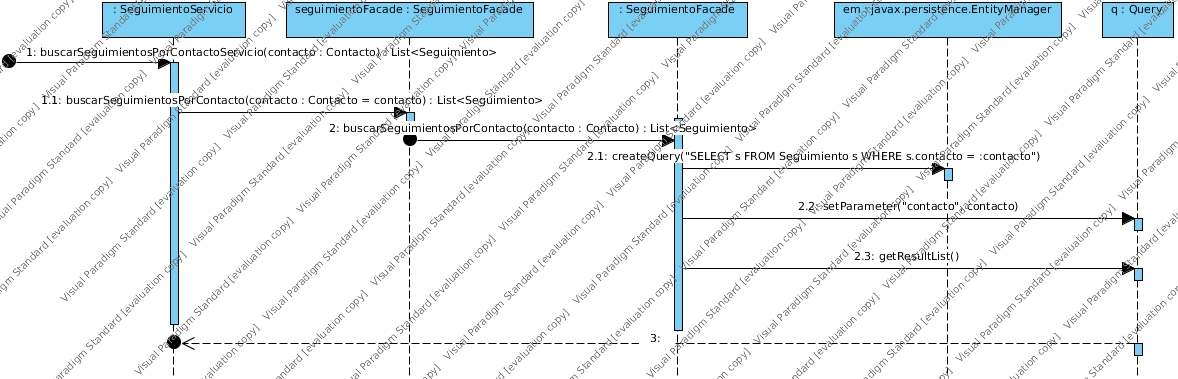
1. Diagrama secuencia de Búsqueda por nombre usuario



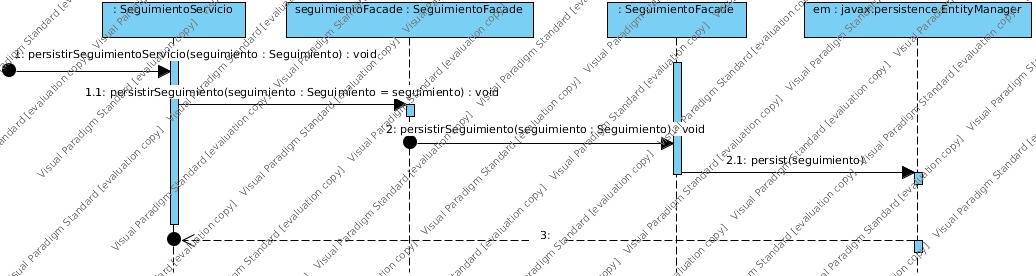
1. Diagrama secuencia de Creación de oportunidades



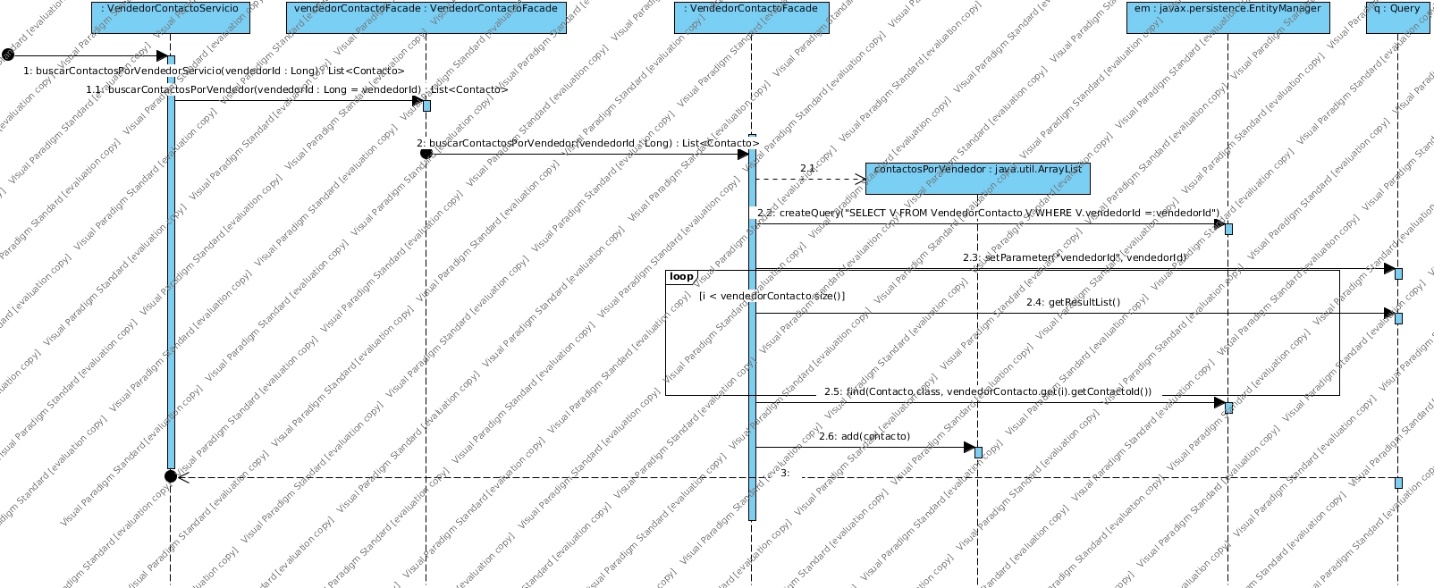
1. Diagrama secuencia de consulta de oportunidades



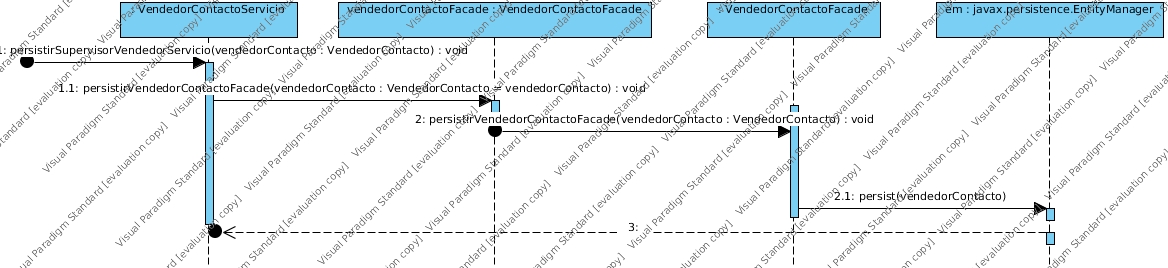
1. Diagrama secuencia de edición de oportunidades



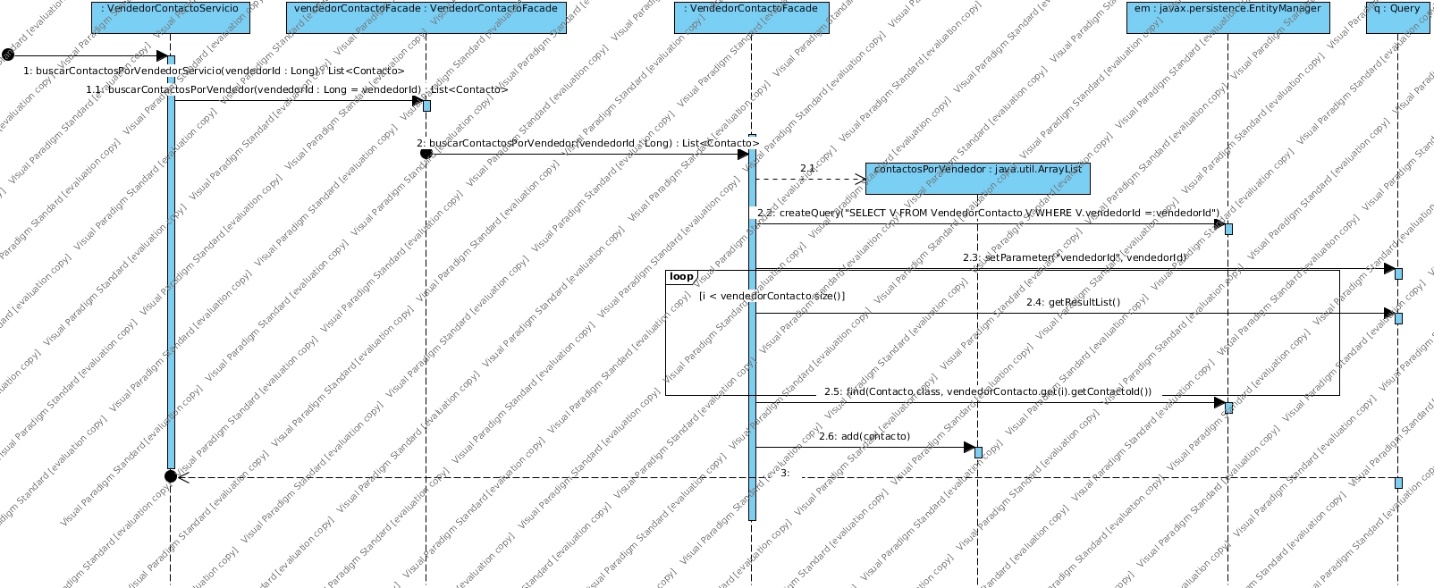
1. Diagrama secuencia de creación de asignación de oportunidades



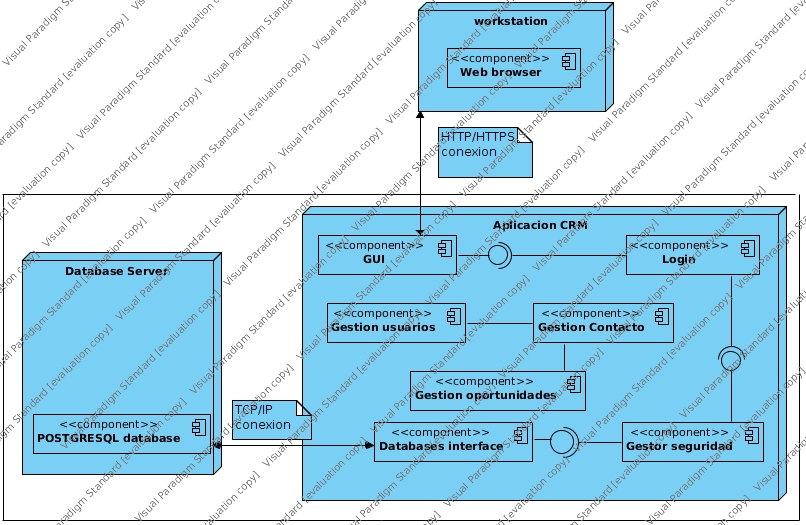
1. Diagrama secuencia de edición de asignación de oportunidades



1. Diagrama secuencia de búsqueda de asignación de oportunidades



# DIAGRAMA DE DESPLIEGUE (ANEXO 3)



# CASOS DE PRUEBA

# (ANEXO 4)

# GLOSARIO DE TÉRMINOS

# (ANEXO 5)

El glosario es la inclusión de los términos poco conocidos dentro de los cuales se menciona el proceso de desarrollo para el SISTEMA DE INFORMACION CRM a continuación se presentará el respectivo glosario.

|  |  |
| --- | --- |
|  |  |
| **APIs** | Interfaz de programación de aplicaciones o API (del inglés ApplicationProgramming Interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas (también denominadas vulgarmente "librerías"). |
| **Arquitectura DE SOFTWARE** | Es el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad,escalabilidad, portabilidad, y disponibilidad. |
|  |  |
| **B**  **Base de Datos**  **C** | Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, y por ende se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos. |
| **Casos de uso** | En un caso de uso, los flujos de eventos se refieren a los pasos que alternativamente van realizando los actores y el sistema en el contexto del requisito funcional capturado en el caso. Dichos pasos por claridad, se separan en el flujo principal y los flujos alternativos; de forma tal que en el flujo principal representamos el día feliz, donde todo ocurre sin problemas y en los flujos alternativos lidiamos con las situaciones de error y el comportamiento esperado del sistema en respuesta a dichos errores. |
| **D** |  |
| **Diagrama de actividad** | Es un Flujo de Trabajo. Tiene un punto de inicio y otro de fin, representados por los puntos de arriba y abajo, respectivamente. En medio, las actividades se indican con cuadros conectados con líneas, siguiendo el orden de los sucesos. Los pequeños rombos representan decisiones que modifican el flujo: aquí, según si el programa está completo o no, se repite el ciclo entero. Por ultimo, los cuadros azules representan documentos o artefactos del mundo real, que son intercambiados como parte del flujo de actividad. |
| **Diagrama de clases** | El diagrama de clases puede ser utilizado para presentar la vista estática del modelo de dominio, del modelo de diseño, o bien, del detalle de la implementación de un sistema en un lenguaje de programación orientado a objeto, como Eiffel, Java o C++. Para todos estos usos, lo que se desea es expresar las unidades en que el código se organiza -las clases- así como algunas características de estas, notablemente como dije antes, sus relaciones, atributos y métodos. |
| **Diagrama de estado** | Los diagramas de estado muestran el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación en respuesta a eventos (por ejemplo, mensajes recibidos, tiempo rebasado o errores), junto con sus respuestas y acciones. También ilustran qué eventos pueden cambiar el estado de los objetos de la clase. Normalmente contienen: estados y transiciones. Como los estados y las transiciones incluyen, a su vez, eventos, acciones y actividades. |
| **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. |
| **E** |  |
| **Editor de texto** | Es un programa que permite crear y modificar archivos digitales compuestos únicamente por texto sin formato, conocidos comúnmente como archivos de texto o texto plano. El programa lee el archivo e interpreta los bytes leídos según el código de caracteres que usa el editor. Hoy en día es comúnmente de 7- ó 8-bits en ASCII o UTF-8, rara vez EBCDIC. |
| **F** |  |
| **Framework** | Framework define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar. En el desarrollo de software, un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto. |
| **G**  **Glassfish** | GlassFish es un [servidor de aplicaciones](https://es.wikipedia.org/wiki/Servidor_de_aplicaciones) de software libre desarrollado por [Sun Microsystems](https://es.wikipedia.org/wiki/Sun_Microsystems" \o "Sun Microsystems), compañía adquirida por [Oracle Corporation](https://es.wikipedia.org/wiki/Oracle_Corporation), que implementa las tecnologías definidas en la plataforma [Java EE](https://es.wikipedia.org/wiki/Java_EE) y permite ejecutar aplicaciones que siguen esta especificación. Es gratuito, de [código libre](https://es.wikipedia.org/wiki/C%C3%B3digo_libre) y se distribuye bajo un licenciamiento dual a través de la licencia [CDDL](https://es.wikipedia.org/wiki/Common_Development_and_Distribution_License) y la [GNU GPL](https://es.wikipedia.org/wiki/Licencia_p%C3%BAblica_general_de_GNU). La versión comercial es denominada Oracle GlassFish Enterprise Server (antes Sun GlassFish Enterprise Server).  GlassFish está basado en el código fuente donado por Sun y [Oracle Corporation](https://es.wikipedia.org/wiki/Oracle_Corporation); este último proporcionó el módulo de persistencia TopLink.[1](https://es.wikipedia.org/wiki/GlassFish#cite_note-1) GlassFish tiene como base al servidor *Sun Java System Application Server* de [Oracle Corporation](https://es.wikipedia.org/wiki/Oracle_Corporation), un derivado de [Apache Tomcat](https://es.wikipedia.org/wiki/Apache_Tomcat), y que usa un componente adicional llamado Grizzly que usa Java NIO para escalabilidad y velocidad. |
|  |  |
| **H** |  |
| **HTML** | siglas de HyperTextMarkupLanguage («lenguaje de marcado de hipertexto»), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas porcorchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML. |
| **I** |  |
| **Internet** | Es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial. Sus orígenes se remontan a 1969, cuando se estableció la primera conexión de computadoras, conocida comoARPANET, entre tres universidades en California y una en Utah, Estados Unidos. |
| **J** |  |
| **Java**  **Javascript** | Es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "*write once, run anywhere*"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra  Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador. Aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo. |
| **JEE**  **JSF**  **L**  **Librería** | Es una plataforma de programación parte de la Plataforma Java para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java. Permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. La plataforma Java EE está definida por una especificación. Similar a otras especificaciones del Java Community Process, Java EE es también considerada informalmente como un estándar debido a que los proveedores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son conformes a Java EE; estandarizado por The Java Community Process / JCP.  Es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL (acrónimo de XML-based User-interface Language, lenguaje basado en XML para la interfaz de usuario).  Es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.  A diferencia de un programa ejecutable, el comportamiento que implementa una biblioteca no espera ser utilizada de forma autónoma (un programa sí: tiene un punto de entrada principal), sino que su fin es ser utilizada por otros programas, independientes y de forma simultánea. |
| **Licencia GNU GPL** | Es una licencia creada por la Free Software Foundation en 1989 (la primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios. |
| **M** |  |
| **Mapa del sitio** | Es una lista de las páginas de un sitio web accesibles por parte de los buscadores y los usuarios. Puede ser tanto un documento en cualquier formato usado como herramienta de planificación para el diseño de una web como una página que lista las páginas de una web (ya realizada), organizadas comúnmente de forma jerárquica. Esto ayuda a los visitantes y a los bots de los motores de búsqueda a hallar las páginas de un sitio web. |
| **Microsoft Windows** | Es el nombre de una familia de sistemas operativos desarrollados por Microsoft desde 1981, año en que el proyecto se denominaba «Interface Manager». |
| **Modelado del dominio** | Es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción, en la tarea construcción del modelo de dominio, presentado como uno o más diagramas de clases y que contiene, no conceptos propios de un sistema de software sino de la propia realidad física. |
| **MVC**  **MySQL** | Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón de llamada y retorno MVC (según CMU), se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista. |
| **N**  **Navegador**  **P** | Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.1 MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual.  Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.  Es un software que permite el acceso a Internet, interpretando la información de distintos tipos de archivos y sitios web para que éstos puedan ser visualizados.  La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Además, permite visitar páginas web y hacer actividades en ella, es decir, podemos enlazar un sitio con otro, imprimir, enviar y recibir correo, entre otras funcionalidades más |
| **PDF**  **Programación orientada a objetos**  **Primefaces** | Es un formato de almacenamiento para documentos digitales independiente de plataformas de software o hardware. Este formato es de tipo compuesto (imagen vectorial, mapa de bits y texto).  Fue inicialmente desarrollado por la empresa Adobe Systems, oficialmente lanzado como un estándar abierto el 1 de julio de 2008 y publicado por la Organización Internacional de Estandarización (ISO) como ISO 32000-1.  La programación orientada a objetos o POO (OOP según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción,polimorfismo y encapsulamiento. Su uso se popularizó a principios de la década de los años 1990. En la actualidad, existe variedad de lenguajes de programación que soportan la orientación a objetos. |
| **Programación WEB** | PrimeFaces es una librería de componentes para JavaServer Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. Primefaces está bajo la licencia de Apache License V2. Una de las ventajas de utilizar Primefaces, es que permite la integración con otros componentes.  Es el proceso de diseñar, codificar, depurar y mantener el código fuente de programas computacionales. El código fuente es escrito en un lenguaje de programación. El propósito de la programación es crear programas que exhiban un comportamiento deseado. El proceso de escribir código requiere frecuentemente conocimientos en varias áreas distintas, además del dominio del lenguaje a utilizar, algoritmos especializados y lógica formal. Programar no involucra necesariamente otras tareas tales como el análisis y diseño de la aplicación (pero sí el diseño del código), aunque sí suelen estar fusionadas en el desarrollo de pequeñas aplicaciones. |
| **R** |  |
| **RIA** | Las rich Internet applications, o RIA (en español "aplicaciones de Internet enriquecidas"), son aplicaciones web que tienen la mayoría de las características de las aplicaciones de escritorio tradicionales. Estas aplicaciones utilizan un navegador web estandarizado para ejecutarse y por medio de complementos o mediante una máquina virtual se agregan las características adicionales. |
| **RUP** | El Proceso Unificado de Rational (RationalUnifiedProcess en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. |
| **S** |  |
| **Servidor** | En informática, un servidor es una computadora que, formando parte de una red, provee servicios a otras computadoras denominadas clientes. También se suele denominar con la palabra servidor a: Una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de una computadora y los servicios de aplicaciones, que realizan tareas en beneficio directo del usuario final. Este es el significado original del término. Es posible que un ordenador cumpla simultáneamente las funciones de cliente y de servidor. Una computadora en la que se ejecuta un programa que realiza alguna tarea en beneficio de otras aplicaciones llamadas clientes, tanto si se trata de un ordenador central (mainframe), un miniordenador, una computadora personal, una PDA o un sistema embebido; sin embargo, hay computadoras destinadas únicamente a proveer los servicios de estos programas: estos son los servidores por antonomasia. |
| **Sitios web** | Es una colección de páginas web relacionadas y comunes a un dominio de Internet o subdominio en la World Wide Web en Internet. Una página web es un documento HTML/XHTML que es accesible generalmente mediante el protocolo HTTP de Internet. Todos los sitios web públicamente accesibles constituyen una gigantesca World Wide Web de información (un gigantesco entramado de recursos de alcance mundial). A las páginas de un sitio web se accede frecuentemente a través de un URL raíz común llamado portada, que normalmente reside en el mismo servidor físico. Los URL organizan las páginas en una jerarquía, aunque los hiperenlaces entre ellas controlan más particularmente cómo el lector percibe la estructura general y cómo el tráfico web fluye entre las diferentes partes de los sitios. |
| **Software** | Equipamiento lógico o soporte lógico de un sistema informático, comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados hardware. |
| **Software libre** | (en inglés free software, aunque esta denominación también se confunde a veces con "gratis" por la ambigüedad del término "free" en el idioma inglés, por lo que también se usa "libre software" y "logical libre") es la denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, modificado, y redistribuido libremente. Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar el software y distribuirlo modificado. |
| **Software no libre**  **SQL** | Es cualquier programa informático en el que el usuario tiene limitaciones para usarlo, modificarlo o redistribuirlo (esto último con o sin modificaciones). Para la Fundación para el Software Libre (FSF) este concepto se aplica a cualquier software que no es libre o que sólo lo es parcialmente (semi libre), sea porque su uso, redistribución o modificación está prohibida, o requiere permiso expreso del titular del software.  Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ellas. |
| **U** |  |
| **URL** | El URL es la cadena de caracteres con la cual se asigna una dirección única a cada uno de los recursos de información disponibles en la Internet. Existe un URL único para cada página de cada uno de los documentos de la World Wide Web, para todos los elementos de Gopher y todos los grupos de debate USENET, y así sucesivamente. |
| **UML**  **W** | Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados |
| **Web** | En informática, la World Wide Web (WWW) o Red informática mundial es un sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de Internet. Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces. |
| **X** |  |
| **XML** | Siglas en inglés de eXtensible Markup Language ('lenguaje de marcas extensible'), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium(W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades, de ahí que se le denomine metalenguaje. |
| **XHTML** | Es básicamente HTML expresado como XML válido. Es más estricto a nivel técnico, pero esto permite que posteriormente sea más fácil al hacer cambios o buscar errores entre otros. En su versión 1.0, XHTML es solamente la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML. Su objetivo es avanzar en el proyecto del World Wide Web Consortium de lograr una web semántica, donde la información, y la forma de presentarla estén claramente separadas. La versión 1.1 es similar, pero parte a la especificación en módulos. En sucesivas versiones la W3C planea romper con los tags clásicos traídos de HTML. |