

INTRODUCCIÓN

Los sistemas de información surgen para cubrir una necesidad de la organización derivada de su ambiente interno o externo. En este caso, el análisis de la situación actual se enfoca en los componentes que conforman el ambiente interno como son: personas, procesos, datos e información. Las características de estos componentes se analizan para identificar los problemas o las oportunidades que se presentan a fin de resuelta en parte o en totalidad por un sistema de información.

Una vez definida la necesidad o identificado el problema se requiere definir el alcance del sistema el cual se hace por medio de la especificación de requerimientos, descripción del producto y un modelo de casos de uso.

2.1. Descripción de procesos.

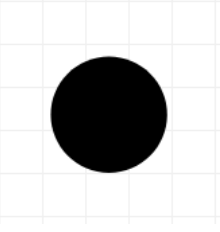
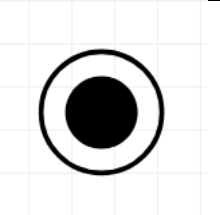
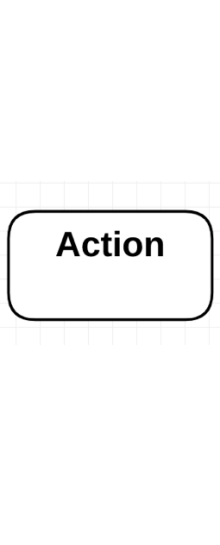
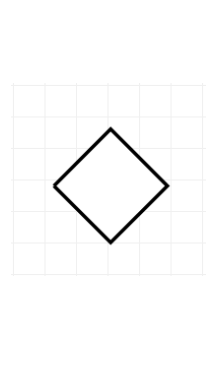
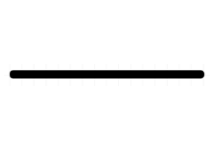
En México, un proceso se define como el conjunto de actividades que tienen como propósito obtener un resultado determinado, se caracteriza por tener una secuencia dada por la naturaleza del trabajo a realizar pudiendo ser ejecutada por un humano o tecnología. Cada una de las actividades requiere de un insumo que puede ser material, dato o energía y genera un subresultado que puede ser del mismo tipo o una combinación de ellos. A su vez, cada actividad la ejecuta una persona que asume un rol en la organización. Puede haber más características que conforman la actividad como la duración, el lugar donde se realiza, el receptor, entre otros.

En la administración de requerimientos los procesos son fundamentales para entender como se está trabajando actualmente y para describir como se quiere trabajar con el sistema de información propuesto.

2.1.1. *Diagrama de actividad.*

El diagrama de actividad es un diagrama de tipo de comportamiento definido por el Unified Model Language (UML) que permite especificar cómo están trabajando los procesos actuales o cómo un sistema logrará sus objetivos. Este tipo de diagrama muestra el flujo de trabajo de acciones de alto nivel encadenadas para representar un proceso actual o futuro. En este sentido, un diagrama de actividad representa lo que pasa o pasará en un proceso por medio de acciones. Para construir los diagramas de actividades se requiere tener conocimiento de las acciones que se llevan a cabo en el proceso o la especificación de los pasos a seguir en el proceso deseado.

Los símbolos que se ocupan en los diagramas de actividades son:

Símbolo	Nombre	Descripción
	Estado inicial	Representa el inicio del diagrama y sólo puede haber uno en el diagrama. No es necesario poner descripción, solo con el símbolo se sabe que inicia el proceso.
	Estado final	Representa el final del proceso. Cualquier acción puede terminar con el proceso. No es necesario poner descripción, solo con el símbolo se sabe que finaliza el proceso.
	Acción	Es una acción que procesa un insumo y lo transforma en un resultado. No hay que confundirlo con el término de actividad porque en UML una actividad es un proceso. Ejemplo: el sistema registra el currículum del alumno o el aspirante se postula en una vacante. Es decir, son acciones simples que muy difícilmente se pueden dividir en mas acciones. Dentro del rectángulo con esquinas redondeado se escribe el nombre de la acción que inicia con verbo.
	Decisión	Es una bifurcación que se presenta después de tomar una decisión. La condición puede generar diferentes secuencias de acciones: No es necesario poner descripción poner descripción, esta se pone en los flujos por medio de una condición de guarda, más adelante se ejemplifica como se realiza.
	Fork y join	Cuando se realizan acciones en forma paralela se ocupa el fork y el join que son barras gruesas. Cuando inician las dos o más acciones

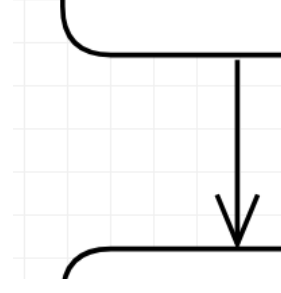
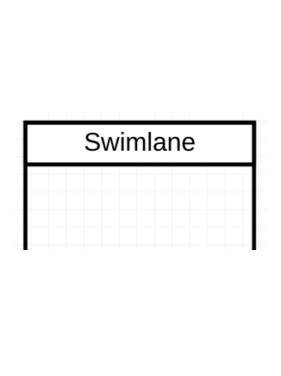
		la barra se denomina fork, cuando finalizan las acciones la barra se denomina join.
	Flujo de control	Es la transición de una acción a otra, es decir es cuando finaliza una acción y luego se cede el control a la siguiente acción. Los flujos de control se establecen entre acciones, decisiones, fork y join.
	Canal	Los canales delimitan las acciones que lleva a cabo un rol que ocupa una persona en la organización, por un actor o tipo de usuario del sistema o el sistema. Todas las acciones o decisiones que se pongan en el canal se entenderán que las ejecuta el rol, actor o sistema.

Figura 2.1 Símbolos que se ocupan en los diagramas de actividades.

Por ejemplo, supongamos que tenemos el siguiente proceso que corresponde a la forma actual en que se consultan las ofertas de trabajo.

1. El aspirante solicita la carpeta de ofertas de trabajo.
2. El servidor social solicita se registre en la bitácora de consulta de ofertas de trabajo.
3. El aspirante registra su nombre, fecha y hora en la bitácora.
4. El servidor social entrega la carpeta de ofertas de trabajo.
5. El aspirante busca ofertas de trabajo de acuerdo con su interés.
6. Si encuentra una oferta que le es atractiva anota el teléfono y correo electrónico.
7. El aspirante termina de revisar la carpeta.
8. El aspirante entrega la carpeta al servidor social.
9. El servidor social anota la hora de devolución en la bitácora.

El diagrama de actividades quedaría como se observa en la siguiente imagen, observa que cada una de las acciones está representada por un rectángulo. Cuando se presenta una decisión se ocupa el rombo donde encontrarás entre

corchetes las posibles opciones que se presentan en la condición, lo que está entre los corchetes de denomina condición de guarda.

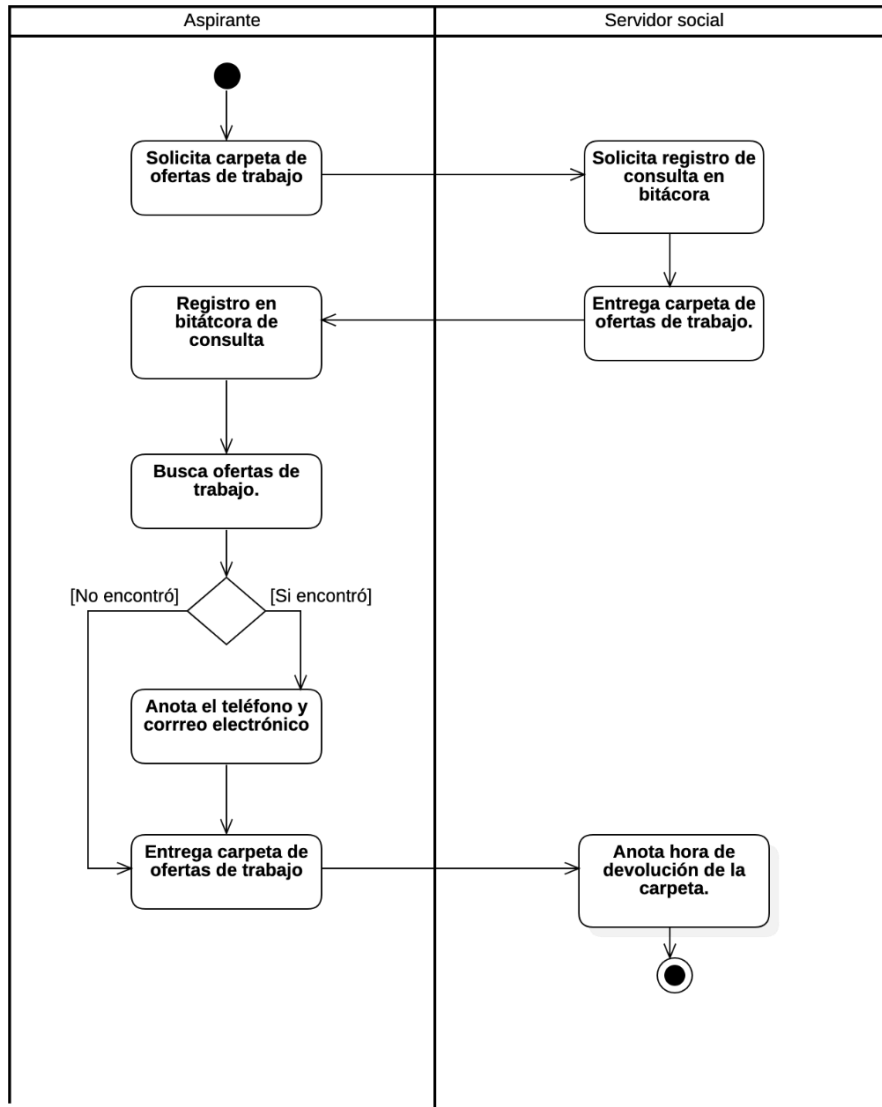


Figura 2.2 Diagramas de actividades

2.2. Identificación de los

stakeholders.

La identificación de stakeholders es un proceso periódico donde se reconocen las personas que se ven afectadas por el proyecto que tiene como propósito construir el sistema de información en cuestión. La identificación implica registrar la información relativa a: forma de contacto, intereses, grado de participación, expectativas, solicitante, control de cambios y cualquier otro que permita su gestión. Las técnicas que se pueden emplear para su identificación son:

- **Juicio de expertos.** Consultar a las personas que tengan conocimientos especializados en: la estructura de la organización y su política; los procesos y como la cultura organizacional influye en ellos; la relación que la organización establece con los clientes y proveedores; industria en la que se desenvuelve la organización; interacciones que se desarrollan en los equipos de trabajo formales e informales.
- **Grupos de enfoque.** Dirigir sesiones grupales con los stakeholders principales que permitan identificar al resto de stakeholders, se pueden tratar otros puntos en la sesión para obtener otro tipo de información o realización de alguna tarea.
- **Lluvia de ideas.** Identificar la mayor cantidad de ideas sin descartar alguna en un tiempo limitado. En el caso del proceso de identificación se busca encontrar la mayor cantidad de stakeholders por mínima que pueda ser su participación en el proyecto
- **Análisis de stakeholders.** Identificar cuáles son los intereses profesionales con respecto al proyecto y el sistema, conocimientos que tienen de la organización y en el uso de las tecnologías de la información.
- **Análisis de documentos.** Identificar los insumos, las actividades y las entradas que conforman los procesos en los manuales de organización, manuales de procedimientos, reportes, informes, contratos, entre otros.

2.3. Definición del alcance.

El alcance se define como el trabajo que se tiene que realizar en un proyecto para poder cumplir con los resultados esperados. En contexto de desarrollo de sistemas informáticos esto incluye los procesos que se requiere automatizar para procesar los datos y generar la información. Cuando se define el alcance de un sistema se esta hablando de los subsistemas que lo conforman, características funcionales y no funcionales. A continuación, se explican diferentes instrumentos que permiten describir el alcance del sistema de información.

2.3.1. Documento de requerimientos.

El documento de requerimientos integra los componentes mínimos esenciales que delimitan el alcance del sistema conformado por necesidades y requerimientos. En la industria hay diversos estándares o propuestas, en esencia su contenido está conformado por tres secciones introducción, visión general y requisitos el cual se basa en el estándar IEE 830 y en el documento de visión del Open Unified Process que se explican a continuación:

Introducción. En esta sección se conforma de las siguientes secciones:

- **Propósito.** Se define el propósito del documento que es describir y especificar las necesidades o problemáticas que dan justificación a la existencia del sistema; la descripción del producto, incluido el sistema, esperados; y los requerimientos y las características de cada uno de los productos.
- **Ámbito del sistema.** Se emplea para dar nombre al sistema, una descripción general de lo que va a hacer, se describen los beneficios esperados alineados a los objetivos de la organización.
- **Glosario.** Lista de conceptos de negocio o técnicos que requieren conocer cada uno de los stakeholders que intervienen en el desarrollo del sistema de información. También, se pueden incluir acrónimos y

abreviaturas. Se puede omitir si ya existe otro documento que tenga la información y tiene que ser mencionado en la subsección de referencias.

- **Referencias.** En enlistan todos los documentos previos que se han creado y que es necesario conocer para entender este documento.

Visión general. Esta sección cuenta con los elementos que justifican la existencia del sistema y la descripción general del producto esperado.

- **Necesidades.** Se describen las necesidades organizacionales que dan origen al sistema. Ejemplo de ello tenemos: mejora en el servicio, agilización de procesos, reducción de costos, entre otros, en alguna o todas las áreas de la organización.
- **Problemática.** Se describe las problemáticas y sus causas señalando cuales de ellas serán atacadas por el sistema de información. Se sugiere emplear el siguiente formato para cada problema detectado:

El problema [descripción del problema] afecta [stakeholders afectados] impactando en [aspectos negativos].

- **Perspectiva del producto.** Se retoma el nombre del sistema de información y se describen las funciones principales. Si el sistema trabajará en coexistencia con otros productos, se o no sistemas de información, se tiene que especificar el tipo de relación. Se sugiere emplear el siguiente formato para describir el producto:

El sistema [nombre del sistema] es para el [nombre del cliente] quien [tiene la necesidad o quiere aprovechar la oportunidad]. El sistema es del tipo [web, móvil, responsiva, standalone, cliente-servidor u otra] que permitirá [principal beneficio].

- **Ambiente del usuario.** Se categorizan y describen los usuarios en función de sus conocimientos y experiencia en el uso de software.

Requerimientos. Esta sección enlista todos los requerimientos del sistema funcionales y no funcionales.

- **Funcionales.** Se enlistan todos los requerimientos funcionales con sus distintos atributos.
- **No funcionales.** Se enlistan todos los requerimientos no funcionales con sus distintos atributos.

2.3.2. *Descripción de la necesidad o problemática.*

Una necesidad la puede presentar la organización o un stakeholder o un grupo de stakeholders; en el contexto del desarrollo de sistemas información las necesidades se centran en la organización. En este caso, una necesidad se entiende como capacidades que no se tienen o se buscan mejorar para lograr un objetivo de negocio como se ha mencionado anteriormente: incrementar los ingresos, aumentar las ventas, incursionar en un segmento de mercado, mejorar los procesos de negocio, reducir el desperdicio, mejorar la atención a los clientes, agilizar los servicios, entre otros. Para

2.3.2.1. Herramientas de diagnóstico.

Las herramientas de diagnóstico permiten encontrar las causas a los problemas ya que uno de los propósitos de los sistemas de información es intervenir en ellas para eliminar o reducir los problemas. Existen dos herramientas que nos permiten especificar los problemas y sus causas.

2.3.2.1.1. Diagrama Causa-Efecto de Ishikawa.

El diagrama de causa-efecto, creado por Kaoru Ishikawa en el año 1943, también conocido como diagrama de pescado, establece categorías que agrupan las causas que dan origen a un problema. En la administración de requerimientos, el diagrama de ishikawa es una herramienta valiosa que permite identificar

cuáles son las causas que se atacarán por medio del sistema de información propuesto y cuales tendrán que ser atacadas por medio de otras soluciones. La estructura básica del diagrama la podemos ver en la siguiente imagen donde en la cabeza se coloca el problema y en el principio de cada línea diagonal las categorías.

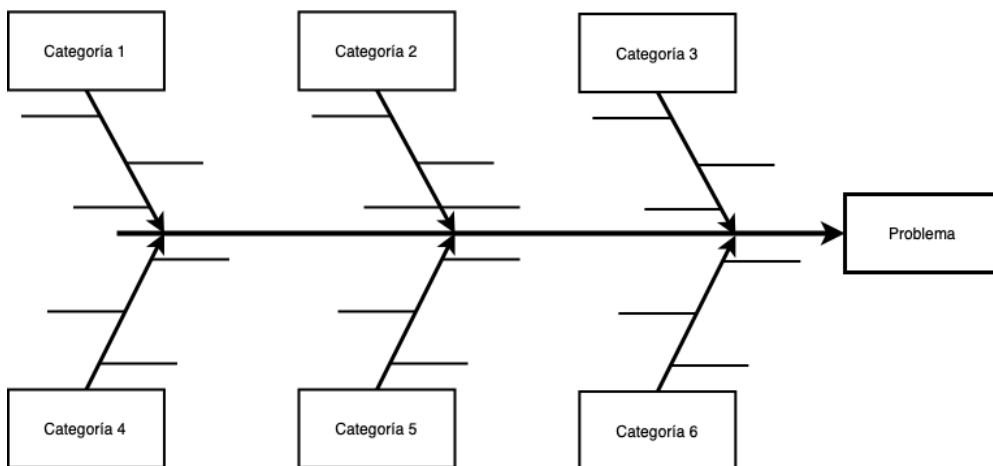


Figura 2.3 Estructura básica del diagrama

Las categorías comunes que se emplean son: personas, procesos, organización o tecnología, aunque se puede agregar todas las que sean necesarias. En cada categoría se escribe la causa o causas que dan origen al problema. No hay límite de causas.

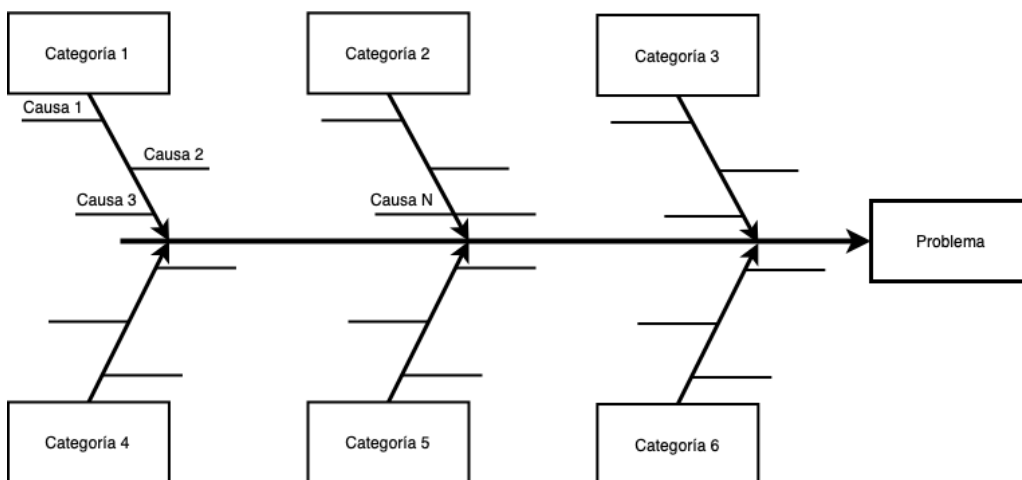


Figura 2.4 Estructura diagrama

Entonces, conforme se aplican cada una de las estrategias de identificación de requerimientos los stakeholders mencionan o dan indicios de los problemas y sus causas, conforme se van identificando es buena práctica que se vaya construyendo el diagrama de Ishikawa. Es importante que se identifiquen las causas que originan los problemas porque de ahí se surge el beneficio que se espera del sistema y que se espera se coloque en el documento de requerimientos.

2.3.2.1.2. Tormenta cerebral.

La tormenta cerebral o mindsigth es la capacidad de la mente humana para contemplarse a sí misma a fin de comprendernos, transformar nuestros esquemas de pensamiento y mejorar las relaciones con los demás. En esencia, permite conectar con las otras personas lo que a su vez permitirá identificar de forma clara y precisa las necesidades, problemas, expectativas y requerimientos.

2.3.3. *Descripción del producto.*

La descripción del producto consiste en especificar cada una de sus partes las cuales pueden ser sistemas, capacitación, manuales, infraestructura y cualquier otro elemento que se requiera para que el sistema de información funcione.

- **Sistemas.** El sistema que se va a crear se puede dividir en subsistemas si es muy grande, en este caso es necesario nombrar cada subsistema y describir su funcionamiento principal.
- **Capacitación.** La capacitación es un servicio que se debe proporcionar antes y después del desarrollo del sistema información a fin de que pueda ser utilizado como es esperado y se generen los beneficios esperados.
- **Manuales.** Los manuales son un apoyo para el uso del sistema, pueden ser para: el uso final del sistema; las operaciones de gestión de usuarios, privilegios, bitácoras, entre otros; o la generación de respaldos y su recuperación.

- **Infraestructura.** La infraestructura actual puede ser suficiente para soportar el sistema que se va a crear, pero si no es así será necesario instalar y configurar servidores, computadoras, servicios de terceros, equipos de comunicación, cableados, entre otros.

2.3.4. *El modelo de casos de uso.*

El modelo de casos de uso es un diagrama que busca identificar cuales son los procesos de negocio que se van a sistematizar. Este tipo de diagrama ocupa principalmente tres tipos de símbolos de UML: casos de uso, actores y asociaciones.

2.3.4.1. Casos de uso y actores.

Los casos de uso son procesos que se llevan de forma actual en la organización o que serán automatizados por el sistema de información que se va a construir. Aquí nos enfocamos en los casos de uso que delimitan el alcance del sistema ya que integran los requerimientos funcionales que fueron identificados previamente. En concreto, un caso de uso representa un conjunto de interacciones que se dan entre el sistema y el actor. Los casos de uso se representan por medio de un óvalo y se coloca su nombre abajo o dentro del símbolo.

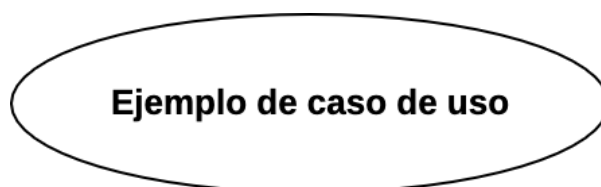
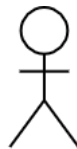


Figura 2.5 Ejemplo de caso de uso

Como se mencionó, el caso de uso es un proceso por lo que para nombrarlo se debe iniciar con un verbo preferentemente en infinitivo o en conjugado en primera

persona del presente. Ejemplo: Gestionar ofertas de trabajo o Gestión de ofertas de trabajo. Nótese que la primera letra inicia con mayúscula.

El actor es el rol que asume una persona, organización o sistema que interactúa con el sistema que se va a construir. En ocasiones los stakeholders indican cuales son los roles que se emplearán en el sistema derivados de los procesos de negocio actuales, en otras ocasiones el analista debe proponer nombres que de igual forma tengan relación con el proceso de negocio. El símbolo para representar el actor es:



Ejemplo actor

Figura 2.6 Ejemplo actor

El nombre del actor debe empezar con mayúscula, si está compuesto de dos o más palabras, a partir de la segunda palabra se escribe con minúsculas. Ejemplos de nombre de actor tenemos: Coordinador de la bolsa de trabajo, Aspirante, Reclutador.

¿Cómo se identifican los actores? Para identificar los actores necesitamos la lista de requerimientos e identificar el primer requerimiento funcional el cuál será un paso de todo un proceso por lo que será parte de un caso de uso; conforme a la descripción del requerimiento tendremos que nombrar el caso de uso al cuál pertenecerá. Ejemplo, teniendo el siguiente requerimiento ¿cuál es el caso de uso que se podría identificar?:

El sistema debe verificar que no se usen palabras altisonantes al momento de registrar ofertas de trabajo por parte de los Reclutadores. Las palabras deben ser definidas por el Jefe del Departamento de Bolsa de Trabajo en el sistema.

La primera operación “El sistema debe verificar que no se usen palabras altisonantes al momento de registrar ofertas de trabajo por parte de los Reclutadores.” Es muy precisa y está indicando que se hace cuando se **registra las ofertas de trabajo** por lo que se puede establecer un caso de uso denominado “Registrar ofertas de trabajo”. Lo interesante es la condición que dice “Las palabras deben ser definidas por el Jefe del Departamento de Bolsa de Trabajo en el sistema.” lo cual no forma parte del caso de uso “Registrar ofertas de trabajo” por lo que conviene crear otro caso de uso que le permita al actor “definir las palabras” un nombre apropiado podría ser “Gestionar palabras altisonantes”. Observa que se empleo el verbo gestionar en lugar de definir porque esto posibilita que se puedan registrar, editar, borrar o consultar las palabras altisonantes, a esto se le conoce como el patrón CRUD (Create, Read, Update, Delete) el cual indica que se puede hacer una o más de estas operaciones clásicas en el flujo del caso de uso. Por lo tanto, si empezamos hacer el modelo de casos de uso tendríamos el siguiente diagrama:



Figura 2.7 Inicio caso de uso

Nótese que en el caso de uso “Gestionar palabras altisonantes” se puso entre pico paréntesis dobles el patrón CRUD, a esto se le llama estereotipo el cual sirve para categorizar cualquier tipo de símbolo en UML. Con esto hemos identificado los casos de uso del requerimiento, lo que prosigue es identificar los

actores, conforme al requerimiento el rol que realiza las operaciones es el “Jefe del Departamento de Bolsa de Trabajo” por lo que el diagrama quedaría de la siguiente forma:

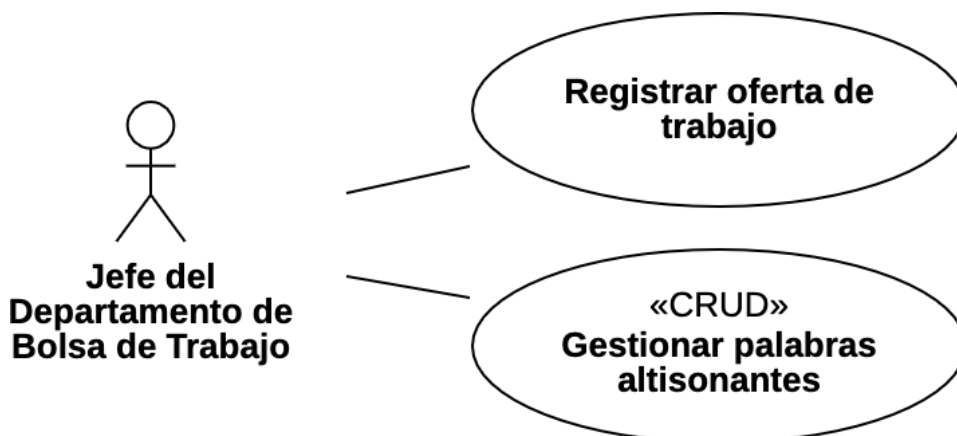


Figura 2.8 Figura caso de uso

Las líneas entre el actor y actores se denominan asociaciones y en este caso establecen una comunicación bidireccional por lo que no es necesario poner flechas, la asociación significa que se envía datos o información entre el actor y el caso de uso. Por último, hay que notar que el caso de uso “Registrar oferta de trabajo” refleja una operación de varias que tienen que ver con las ofertas de trabajo, considerando el siguiente requerimiento vamos a notar que el nombre del caso del uso no refleja la amplitud de todo el proceso:

El sistema debe permitir únicamente al Coordinador de BT y a los Reclutadores editar las ofertas de bolsa de trabajo de acuerdo con lo siguiente:

Reclutador. Podrá editar la información que compone la oferta de trabajo excepto cuando la oferta este aprobada.

Coordinador. Podrá editar únicamente el estado de la oferta de trabajo (Aprobado, Rechazado).

De igual forma en el anexo “Lista de requerimientos” encontrarás varios requerimientos que tienen que ver con la gestión de las ofertas de trabajo por lo que se hace necesario renombrar el caso de uso a lo que realmente va a hacer: “Gestionar ofertas de trabajo” y como te habrás dado cuenta también se cumple el patrón CRUD quedando el diagrama de la siguiente forma:

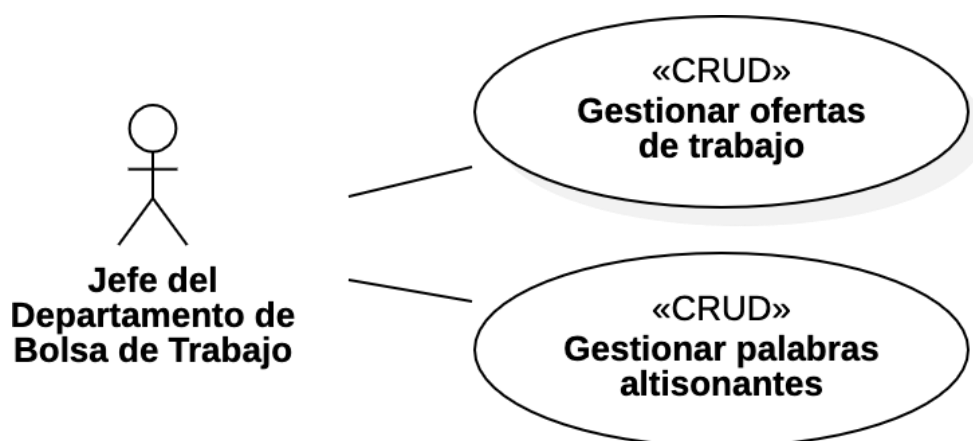


Figura 2.9 Caso de uso

2.3.4.2. Asociaciones: include y extend.

El proceso unificado es un proceso de desarrollo de sistemas informáticos ampliamente usado que establece buenas prácticas para el análisis y diseño de sistemas, fue creado por Philippe Kruchten en el que se establecen cuatro fases: concepción, elaboración, construcción y transición. En la fase de concepción es cuando pueden surgir las asociaciones tipo include y es en la fase de construcción y transición cuando pueden surgir las asociaciones extend o cuando el sistema está en producción y este requiere de una actualización por cuestiones de adaptación o especialización.

Tanto la asociación include o extend son asociaciones de dependencia lo que significa que un caso de uso requiere de la existencia de otro caso de uso para que puedan funcionar. Las asociaciones de dependencia no son exclusivas de los casos de uso, también se ocupan entre clases, el concepto de clase se

estudia en Informática por el momento basta con entender que es una abstracción de un conjunto de objetos que comparten características y comportamientos, como ejemplo se puede abstraer la clase “Estudiante” como el conjunto de personas que hacen algún tipo de estudio en cualquier nivel y que por lo menos tienen nombre y fecha de nacimiento.

La fase de concepción tiene como propósito delimitar el alcance del sistema lo cual se hace principalmente por medio de un documento de alcance del sistema, lista de requerimientos, un modelo de casos de uso y la especificación del 80% de los casos de uso. Gracias a que se cuenta con la especificación de los casos de uso se pueden identificar las asociaciones que incluye. La asociación se identifica cuando hay comportamientos que se repiten en dos o más especificaciones de casos de uso. Por ejemplo, supongamos que en el caso de uso “Registrar currículum” y en el caso de uso “Gestionar ofertas de trabajo” se repite el siguiente conjunto de interacciones:

1. El sistema solicita el usuario y contraseña. Y presenta la opción de ingresar al sistema.
2. El actor proporciona los datos solicitados y acepta enviar sus datos.
3. El sistema verifica que el usuario y contraseña sean correctos.
 - 3.1. Si los datos son correctos.
 - 3.1.1. El sistema da acceso al actor y presenta las opciones correspondientes al actor.
 - 3.1.2. El caso de uso continúa.
 - 3.2. Si los datos son incorrectos.
 - 3.2.1. El sistema notifica de esta situación al actor.
 - 3.2.2. El caso de uso inicia de nuevo.

Lo que se procede a hacer es sacar estas interacciones y crear una nueva especificación de caso de uso, como se puede apreciar las interacciones tienen como propósito dar acceso al sistema a los actores por lo que el nombre del caso de uso será “Acceder al sistema”. Por otro lado, en el modelo de casos de uso

se crea el nuevo caso de uso creando una asociación de dependencia tipo de incluye hacia los casos de uso de donde se sacaron las interacciones, en el ejemplo queda de la siguiente forma:

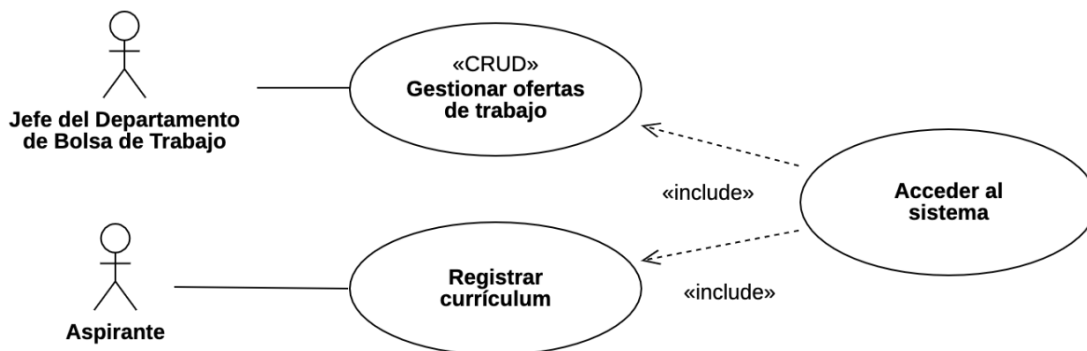


Figura 2.10 Ejemplo caso de uso

Las asociaciones de dependencia se denotan por medio de una línea recta formada por guiones y una flecha al final. Hacia donde apunta la flecha son los casos de los que depende el nuevo caso de uso.

Las asociaciones tipo extend surgen cuando se aprueba un requerimiento que amplía el funcionamiento de un caso de uso sin alterar su comportamiento original, pero el caso de uso ya fue implementado, es decir, se han implementado los programas y bases de datos que dan vida al caso de uso en el sistema. Por lo tanto, se recurre a este tipo de asociaciones; por ejemplo, supongamos que en el caso de uso “Gestionar ofertas de trabajo” aprueban el requerimiento donde piden una nueva opción que consiste en consultar las ofertas de trabajo bajo otras condiciones por parte del Aspirante sin que esto altere los otros flujos de consulta que ya existen. Bajo estas condiciones lo que se procede es actualizar la especificación de caso de uso y el modelo de casos de uso quedando el diagrama de la siguiente forma:

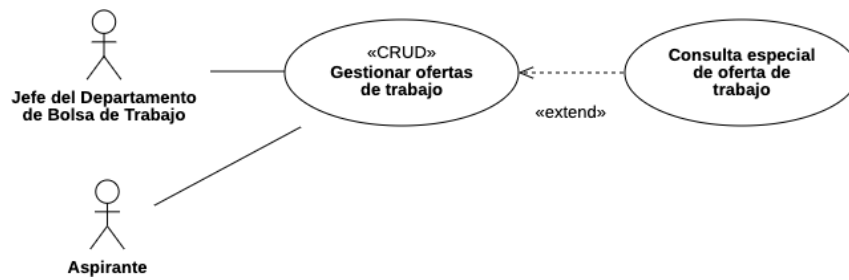


Figura 2.11 Diagrama caso de uso

Observa que el caso de uso que especializa depende del caso de uso original. Además, el ejemplo permite identificar que en el caso de uso “Gestionar ofertas de trabajo” intervienen dos actores lo cual es muy común dado que en los flujos pueden intervenir uno o más actores.

RESUMEN DE LA UNIDAD

Los procesos son parte de la materia prima para la creación de sistemas informáticos ya que es importante identificar y especificar las operaciones que los conforman para determinar cuales se van a sistematizar, eliminar o corregir. Una herramienta que nos permite especificar los procesos son los diagramas de actividad y otra herramienta que nos permite especificar los problemas en los procesos y sus causas son los diagramas de Ishikawa.

Una vez que se conocen los procesos y los problemas que existen se crea el documento de requerimientos en donde se concreta las necesidades que dan origen al sistema, la problemática, el producto esperado, y los requerimientos del sistema.

Por otro lado, se crea un modelo de casos de uso para especificar el alcance del sistema. De cada requerimiento surge uno o más casos de uso porque los requerimientos mencionan una o más operaciones que integran los casos de uso, esto nos recuerda que un caso de uso es un proceso de negocio que se busca sistematizar. Existe el patrón CRUD que indica que se agrega, edita, elimina o consulta el registro de un conjunto de datos. Los actores son los roles de las personas que interactúan con los casos de uso y en algunas ocasiones el analista los tiene que nombrar. Las asociaciones conectan los actores y los casos de uso de forma bidireccional.

Hay dos tipos de asociaciones: include y extend. La primera surge cuando un conjunto de interacciones se repite en dos o más caso de uso por lo que se retiran y se genera un nuevo caso generando una asociación de dependencia de tipo include. La segunda surge cuando se ha implementado un caso de uso y se ha solicitado un nuevo comportamiento que no afecta el comportamiento actual.

REFERENCIAS

Bahamonde, J. M., & Rossel, R. (2003). *Un Acercamiento a la Ingenieria de Requerimientos*.

Gottesdiener, E. (2005). *The Software Requirements: Memory Jogger: a Pocket Guide to Help Software and Business Teams Develop and Manage Requirements*. GOAL/QPC.