

Práctica 01

Análisis de Requerimientos

1. Objetivo General

Conocer y aplicar métodos para la especificación y análisis de los requerimientos de una organización, para su posterior implementación en una base de datos.

2. Objetivos Secundarios

- Estructuración del método apropiado de trabajo que se implementará durante la extracción de los requisitos.
- Definición y análisis de los procesos y el flujo de trabajo de la organización.

3. Introducción

Generar código o modelar soluciones que atiendan las necesidades de una persona u organización, se convierte en un proceso mecánico cuando se tiene la información adecuada. La cuestión es saber responder ¿qué codificar o qué modelar?, sobre todo cuando no se tienen todos los datos necesarios del contexto de las necesidades o problemas a resolver.

Para ser capaces de responder a estas preguntas, una acción razonable es entrevistarse con los usuarios del sistema, que son los actores que día con día realizan actividades bien definidas dentro de la organización y en consecuencia pueden expresar de mejor manera sus necesidades, de esta manera es posible recabar una gran parte de los detalles del accionar de la empresa.

En la realidad, y en la mayoría de los casos, esta fuente de información está conformada por múltiples niveles de usuarios, causando que no todos ellos puedan visualizar de manera global el flujo de trabajo dentro del cual se desarrollan. En consecuencia, se pueden omitir detalles relevantes para la organización alejando de la realidad al posible modelo que se pretende proponer como solución.

En lo que respecta a las bases de datos el rol encargado de proponer soluciones es el Diseñador de Bases de Datos (DBD), éste en ocasiones es un agente externo a la organización cuyos procesos pretende modelar, generando con esto dos grandes dificultades:

- El DBD ignora cuáles son las expectativas del nuevo sistema, es decir, desconoce los procesos de la empresa que se pretenden automatizar
- El DBD al no ser el usuario final del sistema, no recibe retroalimentación alguna de este grupo de usuarios y por lo tanto no puede ajustar su modelo al mismo tiempo que lo prueba, es decir, no puede aplicarse el método de ensayo y error.

Para enfrentar éstas dificultades, en la Figura 1.1 se muestra el orden propuesto para el flujo de información y retroalimentación para el análisis de los requerimientos. Al ser un flujo cíclico, la repetición de los pasos puede extenderse hasta niveles donde el costo aumente haciéndolo no factible. Es por ello, que se pretende minimizar el número de iteraciones a través de los distintos niveles de implementación del sistema completo.

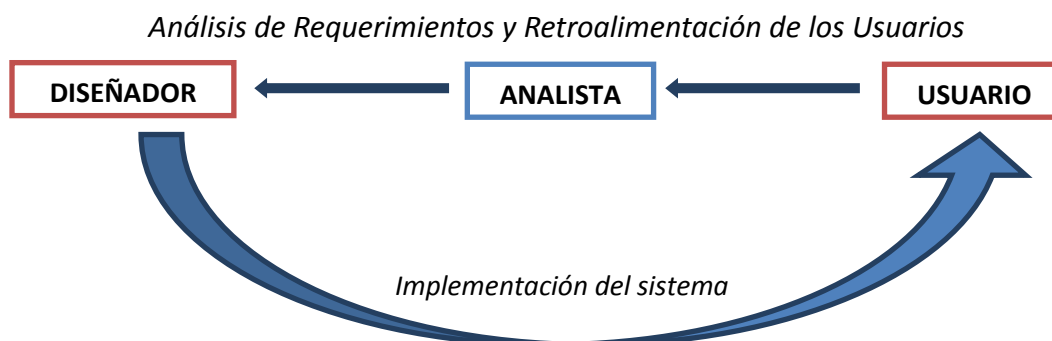


Figura 1.1 - Gráfica Cíclica del análisis y retroalimentación.

El primer paso dentro del desarrollo de una base de datos, que automatice los procesos de una organización, es difícilmente considerado por la mayoría de los DBD; es por ello que esta práctica presenta un método que puede ayudar a obtener un buen entendimiento de las reglas y las características del negocio que se intenta modelar con el fin de proveer una mejor solución. Es importante notar que no todas las tareas o supuestos que se extraigan en el análisis de requerimientos son candidatos para automatizarse, ya sea por su naturaleza, políticas de la empresa, entre otras razones.

3.1. Método de identificación de requisitos

A continuación se enlistan cuatro actividades para realizar el análisis de requerimientos:

3.1.1. Enumerar los requerimientos candidatos

En esta actividad se identificarán las necesidades expresadas por la organización, mismas que serán tratadas como requerimientos de alto nivel, es decir, aún no se especificarán a detalle. La lista que resulta de esta actividad tiene una estructura que permite, posteriormente, añadir diferentes tipos de información con el fin de proveer un panorama más amplio de las cualidades de cada requerimiento. En la Figura 1.2 se muestra la estructura general de esta lista de requerimientos.

Las actividades a seguir para lograr la primera etapa son:

1. Listar las ideas generales de lo que debe y no debe hacer el sistema sin distinción de posibles requerimientos o casos en específico.
2. Asignar un nombre corto y una breve explicación de cada idea general listada.

3. Agregar valores a cada elemento de la nueva lista. Esto parametrizará la tabla y conllevará a una mejor toma de decisiones, estos valores pueden ser:
 - a. Estado: Propuesto, Aprobado, Incluido, Validado, etc.
 - b. Costo estimado de implementación: Costo-Tiempo, Costo-Persona, Costo-Recurso, etc.
 - c. Prioridad: Crítico, Importante, Secundario, etc.
 - d. Nivel de riesgo asociado a la implementación: Grave, Significativo, Ordinario, etc.

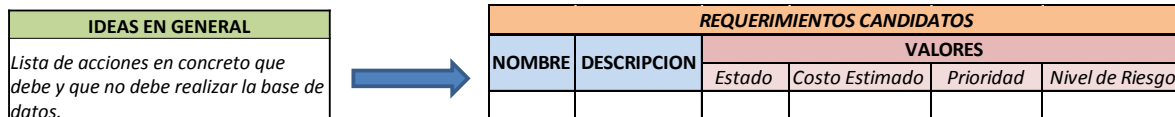


Figura 1.2 - Construcción de la tabla de Requisitos Candidatos.

3.1.2. Comprensión del contexto del sistema

Para lograr una comprensión del contexto del sistema que se pretende modelar, es necesario conseguir un entendimiento global del problema, es decir, encontrar todas las relaciones posibles entre los diferentes actores del sistema (trabajadores, clientes, proveedores, etc.) y las actividades que cada uno debe realizar.

Esta información se puede representar mediante un diagrama que modele la acción de cada actor, estableciendo así un modelo del flujo de trabajo de la organización en cuestión.

Se propone entonces trabajar con diagramas de Actividad. Estos diagramas se componen de los siguientes elementos: Roles, Actividades, Carriles, Flujos y Objetos. Con el objetivo de observar a detalle las necesidades y requerimientos del sistema, a cada actor del sistema se le clasifica según las actividades y responsabilidades que conlleva su puesto dentro de la organización.

Por ejemplo: tres personas distintas trabajan como cajeros en un banco, dos de ellos solo pueden realizar transacciones hasta 100 unidades monetarias (por políticas de la empresa, ya que llevan poco tiempo laborando en la misma), el tercero puede realizar todas las transacciones que se requieren. Además, si alguno de los dos primeros requiere realizar una transacción que no se le permita (mayor a 100 unidades monetarias), deberá pedir la autorización del cajero número que sí cuenta con este privilegio. Así, aunque los tres son cajeros no se podrían clasificar de la misma manera, debido a que sus responsabilidades no son las mismas. De esta manera se tendrían que modelar dos diferentes tipos de rol para los cajeros: “cajero junior” con privilegios limitados y “cajero senior” con todos los privilegios otorgados. Cada uno de éstos roles contará con sus respectivas actividades.

Otro ejemplo podría ser el siguiente: Existe una pizzería que solo reparte a domicilio, es decir, no se puede comer en el local. En ella intervienen 4 grupos de actores: Telefonista, Cocinero, Repartidor y Cliente. Las acciones que realiza cada uno están representadas en la Tabla 1.1.

Tabla 1.1 - Roles y sus Actividades.

Rol	Acciones			
Cliente	Realizar llamada	Proporcionar datos propios	Proporcionar datos de la Orden	Pagar la orden
Telefonista	Tomar llamada	Capturar los datos del cliente	Capturar la orden del cliente	Mandar la orden a Cocina
Cocinero	Recibir la Orden del Telefonista		Cocinar la orden	
Repartidor	Recibir la Orden del Cocinero		Entregar la orden al cliente	

El modelado del negocio resultante se representa como un diagrama de Actividades. La Figura 1.3 muestra dicho diagrama de manera global.

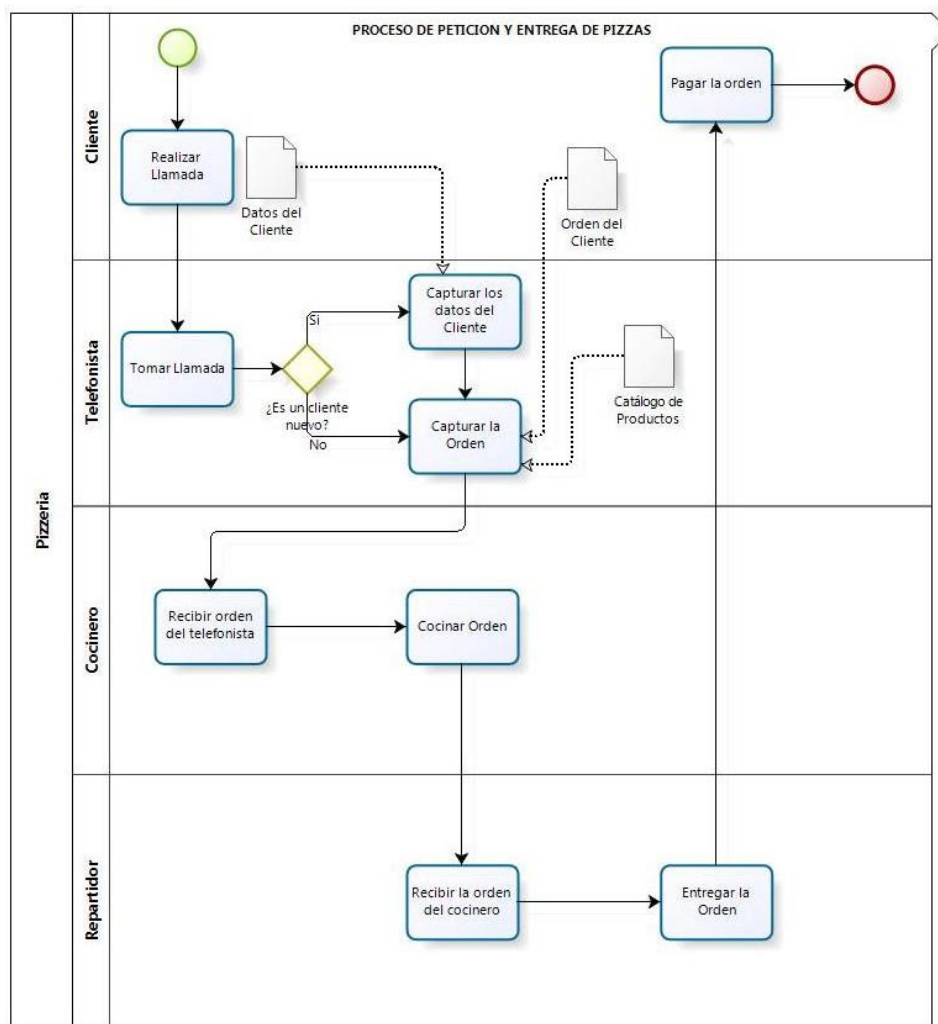


Figura 1.3 - Diagrama de Actividades que modela el flujo de negocio de la Pizzería.

Con este diagrama podemos comprender el contexto en el que se desarrolla el negocio y así tener más información acerca de las necesidades de la organización.

El programa que se utilizó para crear este diagrama fue Bizagi Process Modeler en la versión Express. Este es un software libre que permite modelar procesos a través de diagramas de actividades utilizando notación BPMN. Bizagi puede descargarse de la siguiente liga: <http://www.bizagi.com/>.

3.1.3. Captura de requerimientos funcionales.

Con los puntos 3.1.1 y 3.1.2 realizados, se debe proceder a una especialización y detallado de los requerimientos que se han extraído, obteniendo así los requerimientos funcionales de la solución.

Los requerimientos funcionales son aquellas características o acciones que definen de manera **única** el flujo de trabajo de la organización.

En el ejemplo de la pizzería, podemos observar que existen actores y entidades imprescindibles para el buen funcionamiento del negocio, como lo son: *Cliente*, *Repartidor*, *Telefonista*, *Cocinero*, *Orden*, *Domicilio* y *Pago*. Con la identificación de estos elementos se puede modelar por completo y bajo todos los supuestos la empresa de pizzas a domicilio.

3.1.4. Captura de requerimientos no funcionales.

Los requerimientos no funcionales pueden ser clasificados en dos diferentes grupos:

1.- Requerimientos no funcionales *asociados* a requerimientos funcionales:

Son requerimientos o reglas de negocio (por ejemplo: políticas de la empresa) que definen cada acción del flujo de trabajo, es decir, características propias de cada requerimiento funcional. Por ejemplo: el requisito *Orden*, no puede tener a más de dos *Telefonistas* actuando a la vez para evitar duplicidad de órdenes o errores en la captura. Este es un ejemplo de límite de conexión de usuarios para el requerimiento *Orden*. Así mismo se pueden tener requerimientos de sostenibilidad, fiabilidad, tiempo de respuesta, rendimiento, extensibilidad, etc.

2.- Requerimientos no funcionales *no asociados* a requerimientos funcionales:

Estos requerimientos no dependen de alguna actividad específica del flujo de trabajo, es decir, que no detallan o describen el sistema. Sin embargo, son requerimientos que deben tomarse en cuenta para la implementación del nuevo sistema, como por ejemplo: que plataforma usar, los recursos con los que se cuentan, etc. Para el caso de la pizzería, esta cuenta con 5 computadoras: 1 para cocina, 1 como servidor, 3 para los telefonistas. Las 5 computadoras trabajan con un sistema operativo definido y no se puede cambiar. Podemos observar que estas son características que no intervienen en el accionar del negocio, pero que podrían ser de importancia para la implementación de la base de datos.

4. Ejercicios

(NOTA: Resuelve los siguientes ejercicios en relación al proyecto que realizarás durante el curso, en dado caso que no tengas un proyecto, utiliza la información en el apéndice NFL-ONEFA parte 01 al final de esta práctica para realizarlos)

1. Construye el **Listado de Supuestos** o de Ideas Generales del proyecto.
2. Genera la *Lista de Requerimientos Candidatos* con los parámetros necesarios a partir de la información de la *Lista de Ideas Generales*.
3. Identifica y clasifica por roles a los diversos actores del así como las diversas actividades en las que interviene cada uno.
4. Realiza la tabla de *Actividades por Rol* y genera el **Diagrama de Actividades** asociado.
5. Identifica y construye la *Lista de los Requerimientos Funcionales*.
6. Identifica y construye la *Lista de los Requerimientos No Funcionales* inherentes a los requerimientos funcionales. Identifica y construye la *Lista de los Requerimientos No Funcionales* no inherentes a los requerimientos funcionales.

Entregables requeridos para prácticas subsecuentes:

- Listado de Supuestos del proyecto
- Diagrama de Actividades

5. Apéndice NFL-ONEFA parte 01

La motivación de realizar una base de datos referente a este tema, nace de la idea de llevar un control preciso de los elementos participantes en una liga de futbol americano, sobre todo enfocada a una liga estudiantil como es la ONEFA o a una profesional como la NFL.

La mayor parte de la información contenida en esta base de datos hace referencia a la liga más importante de futbol americano, la NFL. Esta decisión se tomó con el fin de mostrar un modelo eficaz y útil de una realidad bastante compleja como lo es esta liga.

Modelaremos los elementos más importantes de este entorno, como lo son los jugadores y entrenadores, los equipos y sus estadios, las ciudades, los partidos y las relaciones que pueden aparecer entre éstos.

Los supuestos que se tomarán en cuenta son los siguientes:

- La base de datos almacenará información de jugadores, equipos, estadios, ciudades, entrenadores, partidos y de las relaciones jugador-equipo, entrenador-equipo, equipo-estadio, equipo-ciudad.
- Los datos almacenados corresponden a una sola temporada.
- Un jugador puede estar relacionado con dos o más equipos debido a que fue cambiado.
- Un jugador sólo puede tener una especialización, y tomará alguno de los siguientes valores: Receiver, Runner, Quarterback, Punter, Línea Ofensiva, Kicker o Defensiva.
- Un jugador sólo puede tener una posición dentro de su especialización.
- Un jugador se puede especializar en línea ofensiva, donde su posición quedará definida por alguna de las siguientes: C, LS, OG, OL, OT.
- Los jugadores especializados en línea ofensiva tienen como estadística los pancakes (derribar a un rival sin caer) y toman valores mayores o iguales que cero.
- Un jugador se puede especializar en defensiva, donde su posición quedará definida por alguna de las siguientes: DB, DE, DL, DT, LB.
- Los jugadores especializados en defensiva tienen como estadísticas a las siguientes: tackles, sacks, ff, ints mayores o iguales que cero. Además los sacks (atrapar a un QB rival) tienen que ser menores o iguales que tackles.
- Un jugador se puede especializar en kicker, donde su posición quedará definida por K.
- Los jugadores especializados en kicker tienen como estadísticas a las siguientes: fga (goles de campo intentados), fgm (goles de campo anotados), pct (porcentaje de efectividad), blocked (goles de campo bloqueados) y longest (gol de campo más largo anotado en su carrera), todos mayores o iguales que cero. Además fga es mayor o igual que fgm y longest no puede ser mayor que 117.
- Un jugador se puede especializar en punter, donde su posición quedará definida por P.
- Los jugadores especializados en punter tienen como estadísticas a las siguientes: punts (patadas realizadas), avrg (promedio de yardas por patada), net (yardas netas totales), longest (patada más larga realizada en su carrera) e in20 (patadas

colocadas dentro de la yarda 20 rival) todas mayores o iguales que cero. Además punts tiene que ser mayor o igual que in20. Avrg y longest tienen que ser menores que 100.

- Un jugador se puede especializar en quarterback, donde su posición quedará definida por QB.
- Los jugadores especializados en quarterback tienen como estadísticas a las siguientes: att (pases lanzados), comp (pases completados), tds (pases de anotación), ints (pases interceptados) mayores o iguales que cero, yds (cantidad total de yardas obtenidas) sin restricción.
- Los jugadores quarterback son calificados mediante el rating, este valor se obtiene de la siguiente manera:
$$\frac{[(8.4 * yds) + (330 * tds) + (100 * comp) - (200 * int)]}{att}$$
- Un jugador se puede especializar en runner, donde su posición quedará definida por RB o FB.
- Los jugadores especializados en runner tienen como estadísticas a las siguientes: carries (número de veces que acarrea el balón), yds (cantidad total de yardas obtenidas), avrg (promedio de yardas por acarreo) y tds (acarreos de anotación) mayores o iguales que cero, yds sin restricción. Además carries tiene que ser mayor o igual que tds.
- Los jugadores especializados en receiver tienen como estadísticas a las siguientes: rec (número de veces que atrapó el balón), yds (cantidad total de yardas obtenidas), avrg (promedio de yardas por recepción) y tds (atrapadas de anotación) mayores o iguales que cero, yds sin restricción. Además tds debe ser menor o igual que rec.
- Un jugador tiene una categoría, misma que se define a través de su edad de la siguiente manera: Infantil (de 8 a 15 años), Juvenil (de 15 a 17 años), Intermedia (de 17 a 19 años), Liga Mayor (de 19 a 24 años) y Profesional (25 años en adelante).
- Los equipos serán identificados por un acrónimo de tres letras.
- Cada equipo tiene un sobrenombre o *nickname*.
- Cada equipo pertenece a alguna división dentro de su conferencia, pudiendo ser alguna de las siguientes: Norte, Sur, Este, Oeste.
- Cada equipo pertenece a una conferencia dentro de su liga, pudiendo ser alguna de las siguientes: Americana o Nacional.
- El récord de los equipos está compuesto por tres valores: ganados, empatados y perdidos, todos mayores o iguales que cero, menores o iguales que 16 y la suma de los tres es 16.
- Cada equipo tiene un año de fundación entre 1900 y 2013.
- Un entrenador tiene un nombre y apellido.
- Cada entrenador tiene un cargo designado, el cargo puede ser alguno de los siguientes: HC, OC, DC, SC.
- Una ciudad tiene un nombre y un estado asociados.
- El estado es un acrónimo de dos letras.
- Cada ciudad tiene un clima preponderante asignado, y puede tomar los siguientes valores: Templado, Frío, Caluroso.
- Cada ciudad tiene un número de habitantes, mayor o igual que cero.
- Cada ciudad tiene un índice de crecimiento.
- Cada ciudad pertenece a una región, pudiendo tomar alguno de los siguientes valores: Suroeste, Sureste, Medioeste, Mediooeste, Noreste, Noroeste, Central.

- Cada ciudad tiene un ingreso promedio por habitante.
- Los estadios tienen un nombre asignado.
- Cada estadio tiene una capacidad asignada mayor o igual que cero.
- Los estadios pueden tener dos tipos de grama: Artificial o Natural.
- Cada estadio tiene un grado de condición, puede tomar alguno de los siguientes valores: Excelente, Buena, Regular, Mala, Pésima.
- Cada estadio tiene un año de construcción asignado, debe ser mayor que 1800 y menor que 2013.
- Cada estadio está construido en una ciudad, la cual puede no ser a la que pertenece el equipo dueño del estadio.
- Cada estadio está localizado en un estado, siendo éste un acrónimo de dos letras.
- Un jugador relacionado con un equipo tiene un número asignado, el número de juegos jugados e iniciados mayores o iguales que 0. Además número es mayor que 0 y menor que 100, juegos jugados e iniciados son menores que 16.
- Juegos jugados es mayor o igual a juegos iniciados.
- El año de adquisición está determinado por el año en que el jugador fue contratado por el equipo.
- Cada equipo está relacionado con una ciudad.
- El apoyo de una ciudad a un equipo es medido entre 0 y 5, siendo 5 un mayor apoyo.
- Cada equipo está relacionado con un estadio.
- Cada equipo tiene asignado cero o más entrenadores.
- Cada entrenador es contratado en algún año por algún equipo.
- Cada partido es jugado por dos equipos.
- Un partido tiene asignado un equipo local y uno visitante.
- El marcador está compuesto por los puntos del local y del visitante, ambos están entre 0 y 100.
- Cada partido se jugó en alguna fecha entre 1 y 21.

6. Información complementaria

Al diseñar una base de datos es importante contar con el software indicado para realizar las diversas tareas que se necesitan. En particular en esta sección se presentan algunas herramientas alternativas para la realización de diagramas BPMN.

Este apartado presenta tres programas para diseñar diagramas BPMN.

6.1. Bonita Soft.

Bonita Soft es un software de código libre que permite modelar diagramas BPMN, el software permite ser instalado bajo Linux o Windows, ver Tabla 1.

Tabla 1. Ficha informativa de Bonita Soft

Ficha de la Herramienta	
Nombre:	Bonita Soft
Versión revisada:	6.3.3
Página de descarga:	http://es.bonitasoft.com/
Diagramas soportados:	Modelado de negocios/Motor/Diseñador de formularios/Creador de documentos de salida.
Requerimientos adicionales de instalación:	Registro de usuario con cuenta de correo antes de iniciar el Software
Licenciamiento:	Freeware
Entorno de diagramación:	Software
Formatos para exportar diagramas:	GIF, BMP, JPEG, JPG, SVG, PNG y PDF
Facilidad de uso:	Sencillo.
Comentarios adicionales:	-

6.2. Microsoft Visio

Microsoft Visio, es un Software que permite el modelado de diagramas BPMN, entre otros diseños, su instalación es bajo Windows, ver Tabla 2.

Tabla 2. Ficha informativa de Microsoft Visio

Ficha de la Herramienta	
Nombre:	Microsoft Visio
Versión revisada:	2013
Página de descarga:	http://office.microsoft.com/es-es/microsoft-visio-software-avanzado-de-diagramas-FX103472299.aspx
Diagramas soportados:	UML/Bases de Datos/Modelado de negocios/Ingeniería/Arquitectura
Requerimientos adicionales de instalación:	Se instala como parte de la suite de Microsoft Office
Licenciamiento:	Trial(60días), Costo \$7799 MXN
Entorno de diagramación:	Software
Formatos para exportar diagramas:	JPG/PDF/RTF/PDF/PPT/GIF/TIFF/DWG/OTROS
Facilidad de uso:	Sencillo.
Comentarios adicionales:	-

6.3. Altova UModel

Altova UModel es también un software de diseño para modelar diagramas BPMN, ver Tabla 3.

Tabla 3. Ficha informativa Altova UModel

Ficha de la Herramienta	
Nombre:	Altova UModel
Versión revisada:	2014
Página de descarga:	http://www.altova.com/es/umodel.html
Diagramas soportados:	BPMN, UML, Base de Datos SysML y XML
Requerimientos adicionales de instalación:	Se debe registrar el usuario en la página para iniciar el producto como prueba
Licenciamiento:	Trial (30 días), Costo \$150 Euros
Entorno de diagramación:	Software
Formatos para exportar diagramas:	PNG, XML además de generar código Java, C# o Visual Basic .NET
Facilidad de uso:	Sencillo.
Comentarios adicionales:	-