

## GUÍA DE TRABAJO # 1

### DEFINICIÓN DE REQUISITOS FUNCIONALES Y MODELO DEL MUNDO DEL PROBLEMA

#### 1. IDENTIFICACIÓN DE LA ASIGNATURA

<b>Asignatura:</b>	Lógica de Programación y Laboratorio			
<b>Código:</b>	000506001		<b>Área:</b>	Básicas de la ingeniería
<b>Plan de Estudios:</b>	1		<b>Semestre:</b>	1
<b>Créditos: 5</b>	<b>TPS: 6</b>	<b>TIS: 9</b>	<b>TPT: 96</b>	<b>TIT: 144</b>
<b>Trabajo Independiente:</b>	<b>Trabajo Presencial:</b>			
Trabajo Teórico		Trabajo Práctico	X	Trabajo Teórico
				Trabajo Práctico
				X

#### 2. TEMA Y COMPETENCIA:

<b>Contenido temático:</b>	Definición de requisitos funcionales y modelo del mundo del problema
<b>Competencia por desarrollar:</b>	Genera soluciones algorítmicas, mediante el análisis lógico, basado en el uso de estructuras estáticas de datos y en el manejo deductivo de cada una de ellas, para implementarlas en un lenguaje de programación de alto nivel.
<b>Resultado de aprendizaje:</b>	Diseño de una solución algorítmica completa y su software, fundamentado en el paradigma objetual y herramientas de programación básicas.

#### 3. RECURSOS REQUERIDOS

- Equipo de cómputo con conexión a Internet
- Procesador de textos como Word o Notepad++

#### 4. MARCO TEÓRICO

##### 4.1 PROCESO PARA EL DESARROLLO DE SOFTWARE

Normalmente, la construcción de un software parte de las necesidades de información de un cliente que no están satisfechas de forma adecuada. En otras ocasiones, el desarrollo de un software se da para explotar una oportunidad o un tema de interés. Sin importar cuál de los anteriores casos de origen al desarrollo de un software, el cliente deberá comunicarse con un miembro del equipo de desarrollo para explicar los detalles de la situación. Una vez se tiene esta información inicial, los miembros del equipo llevarán a cabo las diferentes etapas del desarrollo de software, que es su forma “más simple” pueden enunciarse como: análisis, diseño y construcción de la solución. Se espera que el resultado de la ejecución de esas etapas sea una solución al problema, la cual en principio debe ser un “producto software”, el cual incluye, entre muchas cosas, un programa para ser ejecutado por el usuario. A esto se le conoce como el proceso para la solución de un problema usado un computador y se muestra en la **Figura 1**.



**Figura 1.** Solución de un problema usando el computador

Cada una de las fases del proceso de desarrollo de software tiene un objetivo y su resultado será un artefacto que hace parte del software que se entregará al usuario. Veamos la descripción de cada fase del proceso de desarrollo.

- El **Análisis del Problema** tiene como objetivo entender y especificar el problema que se quiere resolver. Para eso el grupo de desarrollo deben realizar actividades como la identificación y documentación de los requisitos funcionales y la descripción de las reglas de negocio. Además, se deben identificar las entidades del mundo del problema y reconocer los requisitos no funcionales. Como de esta fase se obtiene el documento de Especificación del problema el cual contiene la lista de requerimientos, las reglas de negocio y los objetos a modelar del mundo del problema.
- El **Diseño de la Solución** tiene como objetivo describir y detallar, textual o gráficamente, las características que tendrá la solución. Como resultado, en esta fase se obtienen los planos para la construcción del software, que en nuestro caso son la descripción detallada de los requerimientos, el diagrama de clases del mundo del problema y los algoritmos asociados a esas clases.
- La **Construcción de la Solución** se centra en implementar la solución en un lenguaje de programación, a partir del diseño. Es decir, en esta fase se hace una traducción de los modelos y de los algoritmos a un lenguaje de programación.
- Las **Pruebas** son la última fase del proceso básico de desarrollo de software y tienen como objetivo validar y verificar el funcionamiento de la solución. Es decir, en esta fase se determina si el software hace lo que debe hacer y si lo hace de forma correcta.

#### Para Recordar ...

Existe una diferencia entre Algoritmos, Programas y Software:

- Un **Algoritmo** es una secuencia ordenada y finita de pasos bien definidos para alcanzar un objetivo previamente establecido.
- Un **Programa** es la implementación de un algoritmo en un lenguaje de programación.
- Un **Software** está constituido por algoritmos, diseños, modelos, programas, documentos de ayuda y datos, todos asociados a un proceso de desarrollo de software.



#### 4.1.1 ANÁLISIS DEL PROBLEMA Y CONSTRUCCIÓN DE LA ESPECIFICACIÓN DEL PROBLEMA

Siempre que se va a solucionar un problema debemos partir del hecho de que un programador NO PUEDE resolver un problema que no entiende. Por esta razón, la primera etapa en todo proceso de construcción de software consiste en tratar de entender el problema que tiene el cliente y expresar toda la información de manera tal que cualquier otra persona del equipo de desarrollo pueda entenderla sin ambigüedad.



Para la especificación del problema se deben definir: los requerimientos funcionales, los requerimientos no funcionales, las reglas de negocio e identificar las entidades del mundo del problema o del dominio del problema. Estas actividades se muestran en la **Figura 2**.



Figura 2. Fase de Análisis del problema y sus actividades

##### 4.1.1.1 ESPECIFICACIÓN DE LOS REQUERIMIENTOS FUNCIONALES

La especificación de los requerimientos funcionales es una de las tareas más críticas en el proceso de desarrollo de software ya que estos **definen de manera explícita** el comportamiento que debe tener una solución de software. Por lo tanto, una especificación incorrecta conllevará a una solución incorrecta del problema.

En general, los requisitos funcionales:

- Definen cuál debe ser el comportamiento de la solución y qué información manejar
- Proporcionan una descripción detallada que permite a cualquier miembro del equipo desarrollar el algoritmo e implementar la solución asociada a cada requerimiento
- Permiten determinar si la solución será aceptada o no por los clientes, esto porque los requerimientos funcionales definen la información relacionada con las necesidades de negocio del cliente

Cuando se define un requerimiento funcional (o requisito funcional) se deben tener en cuenta las siguientes características. Todo requisito funcional debe ser:

- Necesario**. Un requisito es necesario si su omisión provoca una deficiencia en el sistema a construir y, además, su capacidad, características físicas o factor de calidad no pueden ser reemplazados por otras capacidades del producto o del proceso

- **Conciso.** Un requisito es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarla en un futuro
- **Completo.** Un requisito está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión
- **Consistente.** Un requisito es consistente si no es contradictorio con otro requisito

### Para Recordar ...

Un requerimiento funcional determina una **funcionalidad** que el sistema debe tener como parte de la solución del problema del cliente. Al tratarse de una funcionalidad, la forma de nombrar los requerimientos es usando un **verbo en infinitivo + un complemento**, por ejemplo, el sistema debe permitir:



- Registrar un estudiante
- Buscar un empleado por su documento de identidad
- Generar un reporte de las ventas realizadas en el mes

Tras identificar la lista de los requisitos funcionales que debe tener un software, estos deben documentarse para que cualquier miembro del equipo de desarrollo pueda saber con exactitud que debe realizar el sistema bien sea para desarrollar el algoritmo del requerimiento para implementarlo en un lenguaje de programación. Un formato que se puede usar por cada requerimiento identificado es el siguiente.

**Tabla 1.** Formato para la descripción de cada requerimiento

<b>Nombre</b>	Busca asociar un identificador para cada uno de los requerimientos que nos posibilite hacer referencia a él fácilmente, es recomendable que el nombre se especifique usando un <b>verbo en infinitivo</b>
<b>Resumen</b>	Un párrafo que explica que hace el requerimiento
<b>Entradas</b>	<ul style="list-style-type: none"> <li>• Las entradas corresponden a la información que debe ser suministrada por el usuario para solucionar el requerimiento.</li> <li>• Un requerimiento puede tener cero más entradas.</li> <li>• Cada entrada debe asociársele un nombre nemotécnico</li> </ul>
<b>Resultado</b>	Permite conocer que resultado debe obtenerse, es decir que cambios surgen en el sistema y que información se le muestra al usuario.

**Nota:** Recuerde que esta tabla debe diligenciarse por CADA uno de los requerimientos identificados.

### 4.1.1.2 REQUERIMIENTOS NO FUNCIONALES

Estos requerimientos no se refieren directamente a las funciones específicas del sistema, sino que hacen referencia a ciertas propiedades emergentes del sistema como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento, entre otras. Es decir, los requerimientos no funcionales especifican ciertas restricciones del sistema como la capacidad de los dispositivos de entrada/salida o el sistema operativo en el que correrá el software. Estas restricciones **surgen de la necesidad del cliente**, debido a las restricciones en el presupuesto, a las políticas de la organización, a la necesidad de interoperabilidad con otros sistemas de software o hardware o a factores externos como los reglamentos de seguridad, las políticas de privacidad, entre otros.

**Nota:** Estos requisitos son muy importantes para la definición de un sistema, sin embargo, no se tratarán en esta asignatura

### 4.1.1.3 REGLAS DE NEGOCIO

Las reglas de negocio están concebidas como reglas que sirven para definir o restringir alguna acción en los procesos del negocio u organización, por ejemplo, reglas de negocio en dos organizaciones diferentes podrían ser las siguientes:

- Para un sistema de ventas: “Un cliente al que le facturamos más de \$10.000.000 al año es un cliente de tipo A”, “A los clientes de tipo A se les aplica un descuento del 10% en pedidos superiores a 3.000.000”.
- Para un sistema de matrículas: “Un estudiante solo puede matricular máximo 18 créditos en un semestre”, “Una asignatura se aprueba cuando su nota final es mayor o igual a 3.0”.

Especificamente en este curso nosotros usamos las reglas de negocio para describir y detallar cómo el software debe llevar a cabo los procesos asociados a los requerimientos funcionales y como realizar ciertos cálculos, para indicar cuáles pueden ser las restricciones que tienen las entradas de los requerimientos requerimiento y para definir las relaciones que puede haber entre los objetos de diferentes clases.

Existen entonces cuatro tipos de reglas de negocio que consideraremos:

- **Reglas de Cálculo:** definen un conjunto de pasos o expresiones matemáticas que permiten calcular el valor de un término
- **Reglas de estructura y dominio de datos:** las cuales están asociadas a los tipos de datos y rangos de valores de cierta información en los procesos
- **Reglas de estructura y relaciones:** son reglas que permiten definir las entidades sobre las cuales se requiere cierta información y las relaciones existentes entre las entidades
- **Reglas de flujo:** Determinan y limitan como fluye la información a través de un proceso

#### 4.1.1.4 MUNDO DEL PROBLEMA

El objetivo del análisis del mundo del problema es entender el mundo en el que ocurre el problema y recopilar toda la información necesaria para que el programador pueda escribir el programa. Esta actividad está basada en un proceso de observación en el que se deben considerar todos los objetos que intervienen en la solución del problema, incluso aquellos que no están enunciados de forma explícita. Para hacer la definición del mundo del problema se lleva a cabo las siguientes actividades:

- **Identificar las entidades:** esta actividad tiene como objetivo identificar las entidades del mundo que intervienen en el problema. Esas entidades (u objetos) pueden ser concretas (como una persona, un vehículo o una mascota) o abstractos (como una cuenta bancaria, una asignatura o un registro de venta).
- **Reconocer los atributos:** una vez identificadas las entidades del mundo, se debe analizar cuáles son los atributos, propiedades o características de esas entidades que son relevantes para el problema. A cada característica que vayamos encontrando se le debe asociar un nombre significativo y una descripción del conjunto de valores que dicha característica puede tomar. En este caso las **reglas de estructura y dominio de datos** darán la base de la información que debe ser relacionada con los atributos de las entidades.
- **Establecer las relaciones entre las entidades:** en esta actividad, debemos tratar de identificar las relaciones que existen entre las distintas entidades del mundo y asignarles un nombre. En este caso las **Reglas de estructura y relaciones** darán la base de las relaciones a definir entre las entidades.

##### En Resumen ...

El Análisis del Problema es la primera etapa del proceso de desarrollo de software y su salida es un documento denominado Especificación del Problema. Este documento debe contener:

- **La descripción de los requerimientos funcionales**, las cuales hacen referencia a las funcionalidades asociadas al problema del cliente.
- **La definición de las reglas de negocio**, las cuales especifican la forma de realizar los cálculos asociados al negocio, las entradas con sus restricciones de dominio, las reglas de flujo que determinan los caminos de los procesos y las asociaciones entre las entidades del problema.
- **La identificación de las entidades del mundo del problema**, las cuales representan las clases que debemos programar, junto con sus atributos.



## 5. PROCEDIMIENTO

Consideré el siguiente enunciado a partir del cual se exemplificarán los elementos antes descritos. Lea detenidamente y mientras lo hace trate de identificar por sí mismo los requisitos funcionales, las entidades del mundo del problema, sus atributos, y las reglas de negocio asociadas al problema presentado.

### Ejemplo: Análisis y Especificación de un Problema

El propietario de la librería “*El Rincon del Vago*” quiere automatizar su sistema de control de inventario. Para ello le ha contratado a usted para que desarrolle el sistema. Después de algunas reuniones con el propietario para determinar las funciones que el sistema debe proporcionar a la librería, se plasman las siguientes condiciones que debe reunir el sistema de control de inventario:

- Cuando la librería recibe un envío de libros, el inventario deberá actualizarse bien sea mediante el incremento del número de libros disponibles o bien mediante la entrada de un nuevo libro cuando este no está en el inventario.
- Cada vez que un libro sea vendido, el inventario deberá actualizarse.
- El propietario quiere tener un listado completo del inventario, ordenado alfabéticamente por el título de los libros.
- El propietario también quiere que al buscar un libro se muestre la información completa de ese libro.



En las reuniones han acordado la información referente a cada libro que se guardará en el inventario, la cual es: título del libro, nombre del autor, precio, editorial, ISBN, la fecha de copyright, el número de ejemplares disponibles en el inventario.

### 5.1 IDENTIFICAR Y DESCRIBIR LOS REQUISITOS FUNCIONALES:

Existen diferentes formas para identificar los requerimientos. Una de manera para hacerlo es la siguiente:

1. Lea el texto, busque y resalte los verbos que estén relacionados con los procesos de la organización y el problema del cliente. El reconocer los verbos en el texto facilita identificar las acciones o funcionalidades que se deben implementar en el software, lo que deriva en los requerimientos funcionales.
2. Una vez identificados los verbos, renómbrellos usando una estructura tipo: **verbo infinitivo + complemento**. Analice y determine si cada una de las estructuras resultantes apunta a una funcionalidad del programa para solucionar el problema. Con esto puede generar la lista de requerimientos funcionales.
3. Detalle cada uno de los requerimientos funcionales.

Realicemos esta actividad en el ejemplo.

El propietario de la librería “*El Rincon del Vago*” quiere **automatizar** su sistema de control de inventario. Para ello le ha contratado a usted para que desarrolle el sistema. Después de algunas reuniones con el propietario para determinar las funciones que el sistema debe proporcionar a la librería, se plasman las siguientes condiciones que debe reunir el sistema de control de inventario:

Actualizar inventario por compra ↗

- Cuando la librería **recibe** un envío de libros, el inventario deberá **actualizarse** bien sea mediante el incremento del número de libros disponibles o bien **registrando** la entrada de un nuevo libro cuando este no está en el inventario. ↗ **Guardar Libro** ↗ **Actualizar inventario por venta**
- Cada vez que un libro sea vendido, el inventario deberá **actualizarse**.
- El propietario quiere que se **genere un listado** completo del inventario, ordenado alfabéticamente por el título de los libros. ↗ **Generar listado ordenado**
- El propietario también quiere que al **buscar** un libro se muestre la información completa de ese libro. ↗ **Buscar libro**

En las reuniones han acordado la información referente a cada libro que se guardará en el inventario, la cual es: título del libro, el autor, precio, editorial, ISBN, la fecha de copyright, el número de ejemplares disponibles en el inventario. Es importante resaltar que del autor se debe almacenar su nombre y su apellido.

Tras resaltar los verbos debemos analizar cuáles de ellos hacen referencia al negocio y al problema específico del cliente. En este sentido, se debe tener presente que el fin mismo de todo el sistema es **automatizar** el inventario. Ese es el problema, y por eso no lo consideramos un requerimiento específico como tal. Por otro lado, **recibir** los libros de un envío de un proveedor no hace parte de las funcionalidades del sistema, es por esa razón que no lo consideramos un requerimiento.

A partir de esto tenemos que la lista de requerimientos funcionales es la siguiente:

- Guardar Libro
- Actualizar Inventario por Compra
- Actualizar Inventario por Venta
- Generar Listado Ordenado de Libros
- Buscar Libro

Ahora debemos describir cada uno de estos requerimientos, por lo que debemos indicar de manera resumida en qué consiste la funcionalidad de cada requerimiento, cuáles son las entradas (o datos) que el usuario debe entregar al sistema para que este se pueda ejecutar y cuáles son los resultados del requerimiento cuando este termina de ejecutarse. Hagamos la descripción de los requerimientos identificados utilizando el formato de descripción presentado.

<b>Nombre</b>	<b>RF-1. Guardar Libro</b>
<b>Resumen</b>	Permite registrar (o almacenar) en el sistema los datos de un libro nuevo
<b>Entradas</b>	La información requerida para registrar un libro es: el título del libro, el autor, precio, editorial, ISBN, fecha de copyright, número de ejemplares disponibles
<b>Resultado</b>	<ul style="list-style-type: none"> <li>▪ Se ha almacenado un nuevo libro en el sistema y se muestra en pantalla un mensaje indicando que el libro se ha guardado</li> </ul>

<b>Nombre</b>	<b>RF-2. Actualizar Inventario por Compra</b>
<b>Resumen</b>	Permite aumentar la cantidad de ejemplares disponibles de un libro del inventario
<b>Entradas</b>	La información para actualizar el inventario por compra es: El título del libro y la cantidad de ejemplares comprados
<b>Resultado</b>	<ul style="list-style-type: none"> <li>▪ Se ha sumado al número de ejemplares de un libro la cantidad de ejemplares comprados</li> <li>▪ Se muestra un mensaje en la pantalla que indica que se hizo la actualización</li> </ul>

<b>Nombre</b>	<b>RF-3. Actualizar Inventario por Venta</b>
<b>Resumen</b>	Permite registrar en el sistema el egreso de ejemplares del inventario, en el momento en el cual la librería realiza una venta
<b>Entradas</b>	La información para actualizar el inventario por compra es: Título del libro, cantidad de ejemplares vendidos
<b>Resultado</b>	<ul style="list-style-type: none"> <li>▪ Se han restado el número de ejemplares vendidos al número de ejemplares en existencia de un libro</li> <li>▪ Se muestra un mensaje indicando que se hizo la actualización o un mensaje de error si la cantidad vendida es mayor al número de ejemplares disponibles</li> </ul>

<b>Nombre</b>	<b>RF-4. Generar Listado Ordenado de Libros</b>
<b>Resumen</b>	Permite que se genere un listado de libros en el inventario de la librería. Dicho listado debe estar ordenado alfabéticamente por el título de los libros
<b>Entradas</b>	

<b>Nombre</b>	<b>RF-5. Buscar Libro</b>
<b>Resumen</b>	Permite consultar la información completa de un libro en el inventario de la librería usando el nombre del libro
<b>Entradas</b>	Título del libro a buscar
<b>Resultado</b>	<ul style="list-style-type: none"> <li>▪ Se muestra en la pantalla la información completa del libro: el título del libro, nombre del autor, precio, editorial, ISBN, fecha de copyright, número de ejemplares disponibles</li> <li>▪ Si el libro buscado no está en el inventario se mostrará un mensaje en pantalla indicando que el libro no se encontró en el inventario</li> </ul>

**Nota:** observe que el requerimiento RF-4 no tienen ninguna entrada. Esto sucede porque para generar el listado el sistema no necesita que el usuario ingrese ningún dato. Esta es la razón por la cual la celda para las entradas está vacía.

## 5.2 DEFINIR LAS REGLAS DE NEGOCIO

Como se indicó antes, una regla de negocio es una declaración que define o limita algún aspecto del negocio. Su intención es valorar, controlar o influenciar el comportamiento del negocio.

Para definir las reglas de negocio se debe leer el enunciado identificando lo siguiente:

- Situaciones que obliguen a la realización de cálculos matemáticos. Estas son las reglas de cálculo y se deben escribir las fórmulas que se deben tener en cuenta.
- Situaciones que requieran almacenar o conocer una información específica. Esto permite definir las reglas de estructura y dominio de datos.
- Eventos que pueden generar la ocurrencia de otros eventos. Estas son las reglas de flujo y normalmente se identifican porque tienen una estructura del siguiente tipo:

**Si <ocurre este evento> entonces <algo sucede>**

Cuando se identifican estas reglas se debe crear una tabla que debe indicar el requerimiento al que corresponde la regla, la descripción de la regla y el tipo de regla.

Reglas Asociadas a:	Descripción de la Regla de Negocio:	Tipo de Regla:
Actualizar Inventario por Compra	<ul style="list-style-type: none"><li>▪ Para actualizar el número de ejemplares disponibles de un libro por un envío de libros recibidos: <math display="block">\text{cantidadDisponible} = \text{cantidadDisponible} + \text{cantidadComprada}</math></li></ul>	Cálculo
Actualizar Inventario por Venta	<ul style="list-style-type: none"><li>▪ Para actualizar el número de ejemplares disponibles de un libro por una venta realizada: <math display="block">\text{cantidadDisponible} = \text{cantidadDisponible} - \text{cantidadVendida}</math></li><li>▪ Si cantidadVendida es mayor a cantidadDisponible se genera un error en la actualización</li></ul>	Cálculo y flujo
Guardar libro	<ul style="list-style-type: none"><li>▪ La información para guardar un libro es: el título, autor, precio, editorial, ISBN, fecha de copyright y número de ejemplares disponibles.</li><li>▪ El precio y el número de ejemplares disponibles deben ser números enteros positivos</li><li>▪ El formato de la fecha del copyright es aaaa-mm-dd</li><li>▪ El título, autor y editorial deben ser cadenas de texto que no pueden estar vacías</li><li>▪ La información para guardar del autor es el nombre y apellido que son cadenas de texto</li></ul>	Estructura y dominio de datos
Inventario	El inventario es el conjunto de varios libros	Estructura y relaciones

## 5.3 IDENTIFICAR LAS ENTIDADES DEL MUNDO DEL PROBLEMA

Las entidades del mundo del problema son los objetos (reales o imaginarios, personas, animales, cosas o ideas) que son significativos en el negocio y que están DIRECTAMENTE relacionados con el problema del cliente. Las entidades se reconocen porque son los objetos sobre los cuales se requiere conocer o almacenar algún tipo de información.

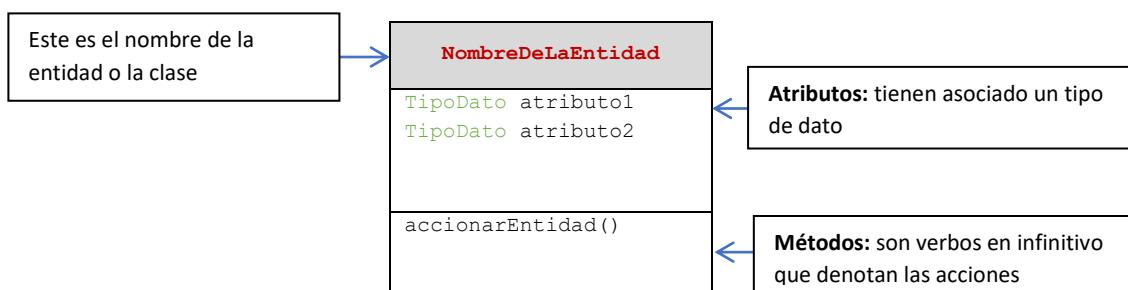
Un proceso básico para la identificación de las entidades del mundo del problema es el siguiente:

1. Identifique los sustantivos (o entidades) en el texto. Es decir, identifique los objetos de los cuales se desea almacenar o conocer información.
2. Identifique los atributos propios de cada una de las entidades y descríbalos. Es decir, identifique cuáles son los datos que se requiere conocer o almacenar de cada una de las entidades. Tenga presente que las **reglas de**

**estructura y dominio de datos** son el insumo principal para este proceso. Use el siguiente formato para describir las entidades:

Entidades	Atributos
<b>Nombre de la entidad:</b> descripción de la entidad	<ul style="list-style-type: none"> <li>■ <b>Nombre del atributo:</b> descripción de lo que es el atributo, su tipo de dato y las restricciones sobre sus valores.</li> </ul>

3. Identifique qué entidad es la responsable de llevar a cabo cada uno de los requerimientos funcionales. Esto nos permitirá establecer cuáles son los métodos (o funciones) que debe tener cada entidad (o clase).
4. Realice la representación gráfica de las diferentes entidades, teniendo presente como se relacionan entre ellas. Aquí las **Reglas de estructura y relaciones** son importantes porque son las que nos ayudan a definir las relaciones que existen entre las diferentes entidades identificadas. Por otro lado, la representación gráfica de las entidades es bastante simple y tiene una estructura compuesta por tres partes:



Realicemos este proceso en el ejemplo planteado. Partimos resaltando los sustantivos relacionados con el problema.

El propietario de la librería “*El Rincon del Vago*” quiere automatizar su sistema de control de inventario. Para ello le ha contratado a usted para que desarrolle el sistema. Después de algunas reuniones con el propietario para determinar las funciones que el sistema debe proporcionar a la librería, se plasman las siguientes condiciones que debe reunir el sistema de control de inventario:

- Cuando la librería recibe un envío de **libros**, el inventario deberá actualizarse bien sea mediante el incremento del número de libros disponibles o bien registrando la entrada de un nuevo libro cuando este no está en el inventario.
- Cada vez que un libro sea vendido, el inventario deberá actualizarse.
- El **propietario** quiere que se genere un listado completo del inventario, ordenado alfabéticamente por el título de los libros.
- El propietario también quiere que al buscar un libro se muestre la información completa de ese libro.

En las reuniones han acordado la información referente a cada libro que se guardará en el inventario, la cual es: título del libro, el **autor**, precio, editorial, ISBN, la fecha de copyright, el número de ejemplares disponibles en el inventario. Es importante resaltar que del autor se debe almacenar su nombre y su apellido.

A partir de la lectura se identificaron 5 entidades: Librería, Inventario, Libro, Autor y Propietario. En el problema específico no se requiere almacenar ni conocer la información de la librería y el propietario, por lo que son entidades que no participan de la solución y, por tanto, no son importantes en este problema. Ahora describamos las entidades encontradas y los atributos que se requieren, de acuerdo con el problema:

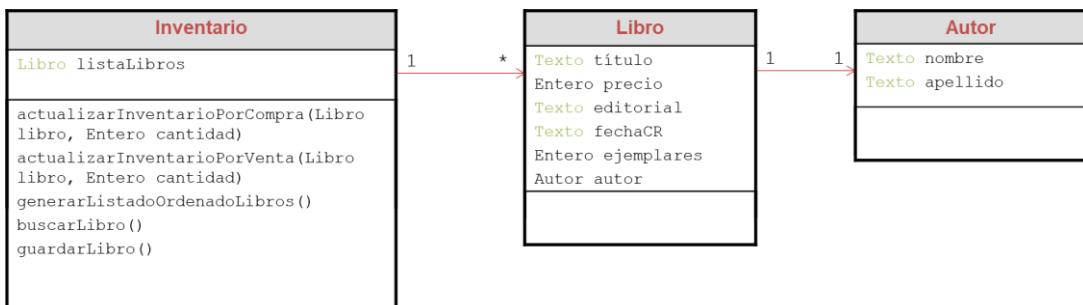
Entidades	Atributos
<b>Libro:</b> esta entidad hace referencia a un libro de los que se venden en la librería	<ul style="list-style-type: none"> <li>■ título: es el nombre del libro. Debe ser una cadena de texto no vacía</li> <li>■ precio: es el valor en pesos de venta del libro. Debe ser un número entero positivo.</li> <li>■ editorial: es la casa editorial que imprime y comercializa el libro. Debe ser una cadena de texto no vacía.</li> <li>■ fechaCR: es la fecha de impresión del libro y copyrigth. Debe ser una cadena de texto no vacía con el formato aaaa-mm-dd.</li> <li>■ ejemplares: es un número que indica cuál es el número de ejemplares disponibles de un libro</li> <li>■ autor: hace referencia al autor del libro y es una entidad</li> </ul>
<b>Autor:</b> esta entidad almacena la información de los autores de los libros	<ul style="list-style-type: none"> <li>■ nombre: es el nombre del autor de un libro. Debe ser una cadena de texto no vacía.</li> <li>■ apellido: es el apellido del autor de un libro: Debe ser una cadena de texto no vacía.</li> </ul>
<b>Inventario:</b> representa los libros disponibles en la librería para la venta	<ul style="list-style-type: none"> <li>■ listaLibros: este atributo es el que debe almacenar la lista completa de libros que hay en la librería. Debe ser un arreglo de entidades tipo libro.</li> </ul>

### Para Recordar ...

- Las **Entidades del Mundo** del problema son las clases que implementaremos en el lenguaje de programación orientado por objetos.
- Toda entidad tiene un nombre en el que cada palabra que lo compone inicia en mayúscula, por ejemplo, `EsteEsElNombreDeUnaEntidad`.
- Los atributos son los datos que requerimos manipular o almacenar de una entidad. Todo atributo tiene un nombre que inicia en minúscula, pero la primera letra de toda palabra después de la primera inicia en mayúscula así: `esteEsUnAtributo`.



Finalmente, hacemos la representación gráfica de las entidades y las conectamos de acuerdo con las relaciones que estas tienen. A esto es a lo que se le llama construir el [Diagrama de Clases](#).



## 5.4 DESARROLLO EN EL PROYECTO INTEGRADOR

Ahora debes poner en práctica lo descrito en esta guía de trabajo. Para ello, selecciona uno de los casos del proyecto integrador con tu equipo de trabajo y hagan lo siguiente:

1. Definir y detallar los requisitos funcionales
2. Definir las reglas de negocio y clasifíquelas por su tipo
3. Identifique las entidades del mundo del problema y sus atributos y descríbalos
4. Construir el modelo de mundo del problema o diagrama de clases
- 5.

## 6. RÚBRICA DE EVALUACIÓN

Resultado de aprendizaje	Indicador de logro	Rango de valoración
Diseño de una solución algorítmica completa y su software, fundamentado en el paradigma objetual y herramientas de programación básicas.	Identifica los requisitos funcionales de la situación problemática	0.0 - 0.5
	Detalla de forma completa los requisitos funcionales	0.0 - 1.5
	Identifica y detalla las reglas de negocio del sistema, clasificándolas de forma adecuada	0.0 - 1.5
	Identifica las entidades del modelo del mundo del problema, las describe y construye el diagrama de clases	0.0 - 1.5

## 7. REFERENCIAS BIBLIOGRÁFICAS

Pressman, R. S. (2010). Ingeniería del Software-Un enfoque práctico. 7<sup>a</sup> Edición McGraw-Hil.

Villalobos, J. A., & Casallas, R. (2006). Fundamentos de programación: aprendizaje activo basado en casos: un enfoque moderno usando Java, UML, Objetos y Eclipse (No. Sirsi) i9789702608462).

<b>Elaborado Por:</b>	Diana Patricia Bedoya Ruiz Carlos Andrés Mera Banguero
<b>Versión:</b>	2.0
<b>Fecha</b>	Julio de 2022
<b>Aprobado Por:</b>	Comité Curricular Departamento de Sistemas de Información Acta 02 del 7 de febrero de 2023