

**PROGRAMA DE ESTUDIOS**

# **DESARROLLO DE SISTEMAS DE INFORMACIÓN**

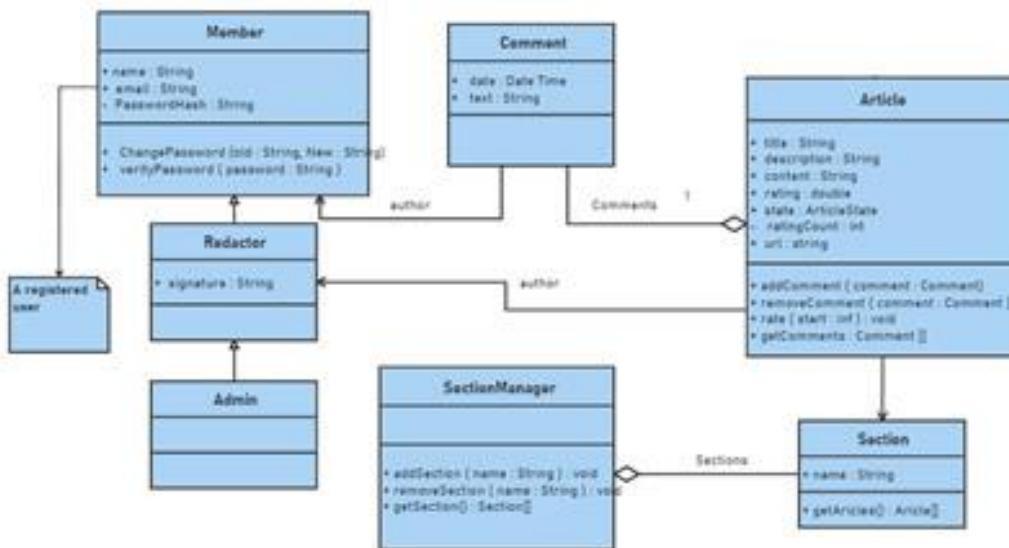
**ANÁLISIS Y DISEÑO  
DE SISTEMAS**

**Tema:**

**DESARROLLO DEL DIAGRAMA DE  
CLASES PARA UN SISTEMA DE  
GESTIÓN.**

## DESARROLLO DEL DIAGRAMA DE CLASES PARA UN SISTEMA DE GESTIÓN.

### ¿Qué es una clase?



En la programación orientada a objetos, **clases** son un plano definido por el usuario para un objeto. Traza las características, métodos o cualidades comunes a todos los objetos de cierto tipo. Asimismo, los objetos también son parte fundamental de la programación orientada a objetos que se crea en base al conjunto de reglas de una clase. Los objetos y las clases funcionan simultáneamente, y ninguno puede discutirse sin la referencia del otro.

### *¿Qué son los diagramas de clase?*

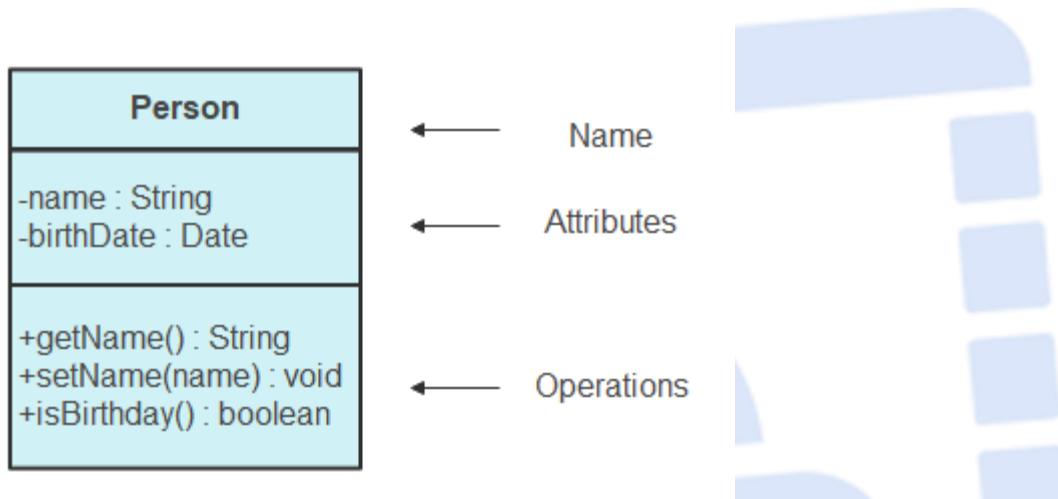
Basado en este concepto, los diagramas de clases se dibujan en **UML (lenguaje de modelado unificado)** que ilustran los atributos, relaciones y operaciones entre objetos. Los diagramas de clases son la base del software y son diagramas de estructura estática que funcionan según el principio de orientación a objetos. Ayudan en el modelado de datos, el modelado conceptual y la traducción de los modelos a códigos de programación.

El propósito principal de un diagrama de clases es diseñar y analizar la vista estática de una aplicación. Un diagrama de estructura UML define el conjunto de instrucciones de un sistema y trabaja para diseñar la representación de la realidad. Estos diagramas son especialmente útiles a la hora de generar códigos fuente. Son una excelente manera de demostrar las interrelaciones entre objetos debido a sus atributos y operaciones.

### *Notación de clase UML*

Los diagramas de clases se ilustran dibujando diagramas que se dividen en tres particiones.

- **Sección superior:** Esta sección es obligatoria en todos los diagramas de clases y muestra el nombre de la clase. El nombre siempre se escribe en negrita, alineado en el centro, con la primera letra en mayúsculas.
- **Sección intermedia:** La sección central contiene los atributos. Ayuda a describir las propiedades de la entidad que se está modelando. Los atributos se mantienen alineados a la izquierda, escritos en minúsculas. El tipo se menciona después del color.
- **Sección inferior:** La sección inferior presenta las operaciones. Las operaciones de la clase están escritas en una lista y describen la interacción de una clase con los objetos.



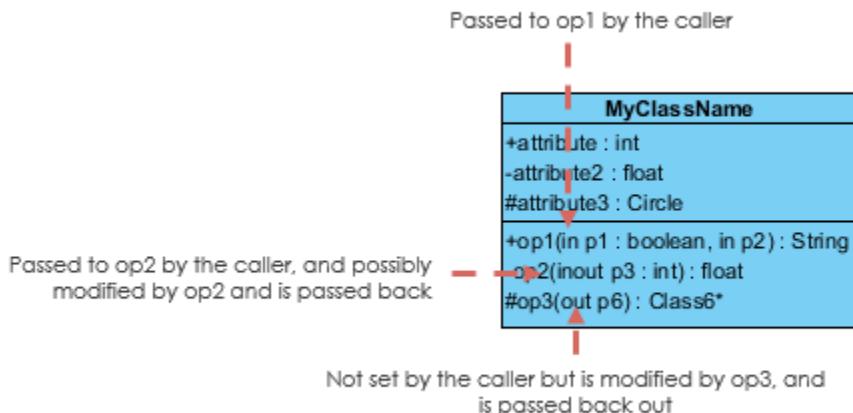
### Visibilidad

Todas las clases exhiben un nivel de modificador de acceso variable en función de su visibilidad. Los marcadores de visibilidad limitan a las personas que pueden acceder a la información en una clase en particular. A continuación, se presentan los marcadores de visibilidad y su información de acceso:

Marcador	Visibilidad	Descripción
+	Publico	Todas las clases pueden ver la información.
-	Privado	La información está oculta a todas las clases fuera de la partición.
#	Protegido	La clase secundaria puede acceder a la información heredada de la clase.

### Direccionalidad del parámetro

Para especificar la dirección relativa a la persona que llama, las operaciones se pueden etiquetar como 'in,' 'out,' o 'inout.' La direccionalidad del parámetro se especifica antes del nombre del parámetro.



## Relación entre clases

En un diagrama de clases, todas las clases están vinculadas entre sí mediante relaciones adecuadas. Estos enlaces ayudan al usuario a comprender a fondo la conexión entre diferentes entidades. Sin embargo, debido a las leves similitudes, muchos a menudo tienen problemas para comprender las diferentes relaciones del diagrama de clases.

### 1. Asociaciones

**Las asociaciones** se utilizan para representar los vínculos familiares y significan la relación estática entre clases. Conecta estructuralmente dos o más clasificadores y enumera sus atributos, propiedades y asociaciones. Las asociaciones están representadas por una línea sólida trazada entre los dos clasificadores.

Las asociaciones se dividen en cuatro tipos: asociación unidireccional, bidireccional, agregación y composición.

Asociación unidireccional:



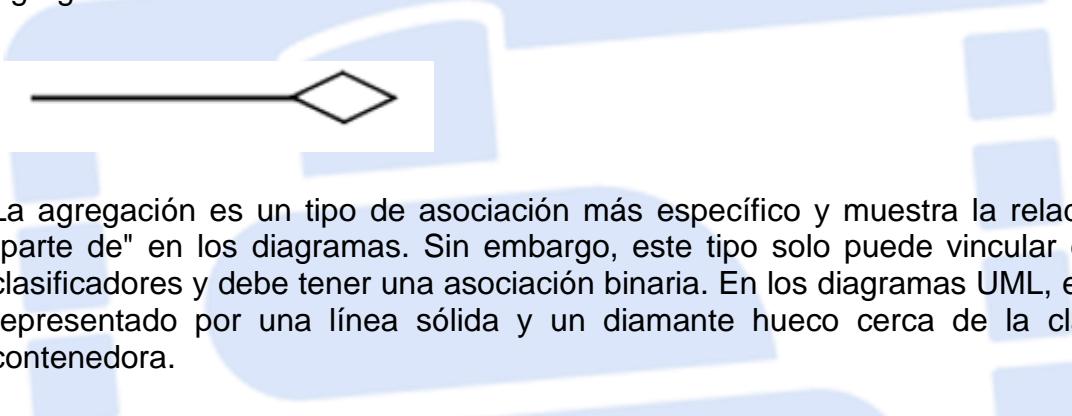
También conocida como asociación dirigida, este tipo de asociación se refiere a cuando un objeto contiene otro objeto en su campo. Esta relación significa el flujo de información entre dos clasificadores. La asociación está representada por una línea continua y una flecha que apunta hacia el clasificador de contenedores.

Asociación bidireccional:



Esta asociación se utiliza cuando dos clasificadores están estrechamente vinculados y pueden almacenarse entre sí en sus campos. Una línea continua representa la asociación. La asociación bidireccional es el tipo de asociación más frecuente en los diagramas UML.

Agregación:



La agregación es un tipo de asociación más específico y muestra la relación "parte de" en los diagramas. Sin embargo, este tipo solo puede vincular dos clasificadores y debe tener una asociación binaria. En los diagramas UML, está representado por una línea sólida y un diamante hueco cerca de la clase contenedora.

Composición:



Este tipo de relación se utiliza para representar la dependencia de los objetos de la entidad focal. El clasificador focal contiene objetos, pero los objetos contenidos también se eliminan si se elimina la clase focal. Las relaciones de composición están representadas por una línea sólida y una forma de diamante relleno dibujada cerca de la clase contenedora.

2. Generalización/Herencia:



En el modelado UML, la generalización se utiliza para representar las relaciones de la clase principal y la clase secundaria. Se puede ver una "especie de" relación entre los clasificadores y cómo una entidad se basa en la otra, heredando los atributos, las operaciones y las relaciones de los padres.

El modelo de padres puede tener muchas clases secundarias y, del mismo modo, una clase secundaria puede tener varios modelos principales. En un diagrama UML, las generalizaciones se muestran con una línea sólida, con una flecha vacía que apunta desde la clase secundaria a la clase principal.

### *3. Realización*



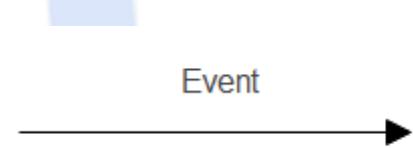
Es una relación que vincula dos elementos del modelo con un clasificador realizando/implementando el comportamiento de otro clasificador. La relación de realización ayuda a comprender cómo afecta la interfaz a la clase de implementación. La realización está representada por una línea discontinua con una flecha hueca.

### *4. Dependencia*



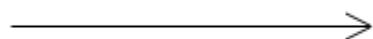
Este es un tipo de relación más débil que se usa cuando un objeto no está contenido en ningún campo. La relación de dependencia existirá y el cambio en una clase provocará cambios en la otra entidad. Está representado por una línea discontinua y una punta de flecha que apunta hacia la entidad dependiente.

### *5. Transición*



En el modelado UML, el estado de transición se usa para conectar dos compuestos / estados directamente. A los estados se les atribuye el mérito de ser la principal transición y la principal fuente de transición. Se utiliza una línea sólida con una flecha llena para representar el cambio de control entre dos clases. Sobre la flecha, se etiqueta el evento que causa esta transición.

### *6. Bordes de actividad*



El diagrama de actividades es un tipo esencial de diagrama UML que se utiliza para representar el flujo entre las diferentes actividades de un sistema. Ayudan a comprender la lógica detrás del algoritmo de un sistema. El flujo de actividad está representado por una línea sólida con una punta de flecha llena que apunta hacia la siguiente actividad en la secuencia.

#### FUENTE:

- <https://www.uml.org/>
- <https://www.codingdojo.la/2023/06/16/guia-del-ciclo-de-vida-del-desarrollo-de-software/>
- <https://aws.amazon.com/es/what-is/sdlc/>
- BURCH, John; GRUDNISKY, Gary. "Diseño de Sistemas de Información", Grupo Noriega editores.
- SENN, James A. "Análisis y diseño de sistemas de información", 2da. ed., McGraw-Hill.

#### *Beneficios de los diagramas de clases*

Los diagramas de clases ofrecen una serie de beneficios para toda organización. Usa los diagramas de clases UML para:

- Ilustrar modelos de datos para sistemas de información, sin importar qué tan simples o complejos sean.
- Comprender mejor la visión general de los esquemas de una aplicación.
- Expresar visualmente cualesquier necesidades específicas de un sistema y divulgar esa información en toda la empresa.
- Crear diagramas detallados que resalten cualquier código específico que será necesario programar e implementar en la estructura descrita.
- Ofrecer una descripción independiente de la implementación sobre los tipos empleados en un sistema que son posteriormente transferidos entre sus componentes.

¿Deseas crear un diagrama UML por tu cuenta? Prueba Lucidchart. Es rápido, sencillo y totalmente gratis.

### ***Componentes básicos de un diagrama de clases***

El diagrama de clases estándar está compuesto por tres partes:

- ***Sección superior:***

Contiene el nombre de la clase. Esta sección siempre es necesaria, ya sea que estés hablando del clasificador o de un objeto.

- ***Sección central:***

Contiene los atributos de la clase. Usa esta sección para describir cualidades de la clase. Esto solo es necesario al describir una instancia específica de una clase.

- ***Sección inferior:***

Incluye operaciones de clases (métodos). Esto está organizado en un formato de lista. Cada operación requiere su propia línea. Las operaciones describen cómo una clase puede interactuar con los datos.

### **Modificadores de acceso a miembros**

Todas las clases poseen diferentes niveles de acceso en función del modificador de acceso (visibilidad). A continuación te mostramos los niveles de acceso con sus símbolos correspondientes:

- Público (+)
- Privado (-)
- Protegido (#)
- Paquete (~)
- Derivado (/)
- Estático (subrayado)

### Alcance de los miembros

Hay dos alcances para los miembros: clasificadores e instancias.

Los clasificadores son miembros estáticos, mientras que las instancias son las instancias específicas de la clase. Si estás familiarizado con POO, esto no es nada nuevo.

### Componentes adicionales del diagrama de clases

En función del contexto, las clases de un diagrama de clases pueden representar los objetos principales, las interacciones en la aplicación o las clases que se programarán. Para responder la pregunta "¿Qué es un diagrama de clases en UML?", primero deberías comprender su composición básica.

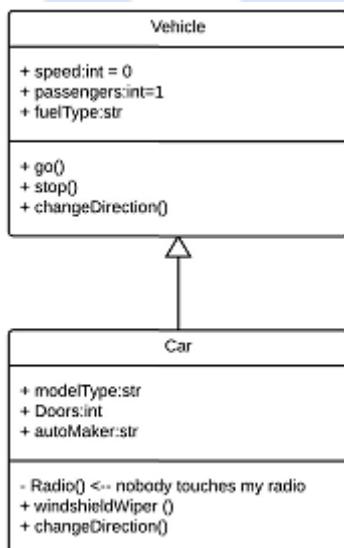
- Clases: Una plantilla para crear objetos e implementar un comportamiento en un sistema. En UML, una clase representa un objeto o un conjunto de objetos que comparte una estructura y un comportamiento comunes. Se representan con un rectángulo que incluye filas del nombre de la clase, sus atributos y sus operaciones. Al dibujar una clase en un diagrama de clases, solo se debe cumplimentar la fila superior. Las otras son opcionales y se usan si deseas agregar más detalles.
  - Nombre: La primera fila en una figura de clase.
  - Atributos: La segunda fila en una figura de clase. Cada atributo de una clase está ubicado en una línea separada.
  - Métodos: La tercera fila en una figura de clase. También conocidos como "operaciones", los métodos se organizan en un formato de lista donde cada operación posee su propia línea.
- Señales: Símbolos que representan comunicaciones unidireccionales y asincrónicas entre objetos activos.
- Tipos de datosClasificadores que definen valores de datos. Los tipos de datos pueden modelar tanto enumeraciones como tipos primitivos.
- Paquetes: Figuras diseñadas para organizar clasificadores relacionados en un diagrama. Se simbolizan con una figura de un gran rectángulo con pestañas.
- Interfaces: Una recopilación de firmas de operaciones o de definiciones de atributo que define un conjunto uniforme de comportamientos. Las interfaces son similares a una clase, excepto por que una clase puede tener una instancia de su tipo, y una interfaz debe poseer, como mínimo, una clase para implementarla.

- Enumeraciones: Representaciones de tipos de datos definidos por el usuario. Una enumeración incluye grupos de identificadores que representan valores de la enumeración.
- Objetos: Instancias de una clase o clases. Los objetos se pueden agregar a un diagrama de clases para representar instancias prototípicas o concretas.
- Artefactos: Elementos modelo que representan las entidades concretas de un sistema de software, como documentos, bases de datos, archivos ejecutables, componentes de software y más.

### Interacciones

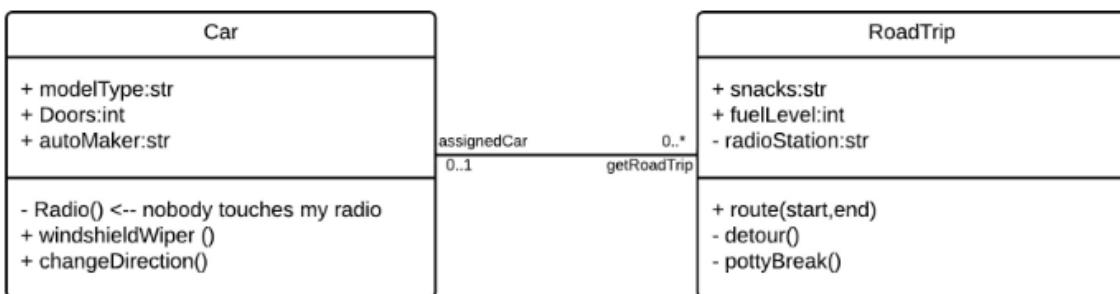
El término "interacciones" se refiere a múltiples relaciones y enlaces que pueden existir en diagramas de objetos y de clases. Algunas de las interacciones más comunes incluyen:

- Herencia: El proceso en el que una subclase o clase derivada recibe la funcionalidad de una superclase o clase principal, también se conoce como "generalización". Se simboliza mediante una línea de conexión recta con una punta de flecha cerrada que señala a la superclase.



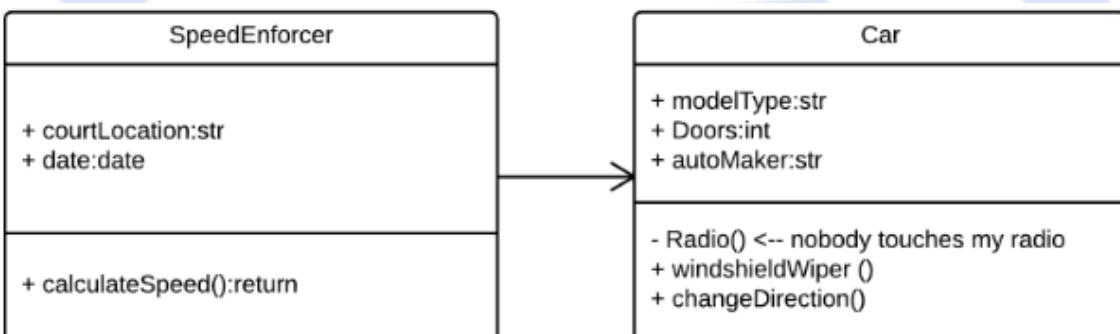
En este ejemplo, el objeto "Auto" heredaría todos los atributos (velocidad, números de pasajeros, combustible) y los métodos (arrancar(), frenar(), cambiarDirección()) de la clase principal ("Vehículo"), además de los atributos específicos (tipo de modelo, número de puertas, fabricante del auto) y métodos de su propia clase (Radio(), limpiaparabrisas(), aireacondicionado/calefacción()). La herencia se muestra en un diagrama de clases por medio de una línea continua con una flecha cerrada y vacía.

- Asociación bidireccional: La relación predeterminada entre dos clases. Ambas clases están conscientes una de la otra y de la relación que tienen entre sí. Esta asociación se representa mediante una línea recta entre dos clases.



En el ejemplo anterior, la clase Auto y la clase Viaje están interrelacionadas. En un extremo de la línea, el Auto recibe la asociación de "autoAsignado" con el valor de multiplicidad de 0..1, de modo que cuando la instancia de Viaje existe, puede tener una instancia de Auto asociada a ella o no tener instancias de Autos asociadas a ella. En este caso, una clase CasaRodante separada con un valor de multiplicidad de 0..\* es necesaria para demostrar que un Viaje puede tener múltiples instancias de Autos asociadas a ella. Dado que una instancia de Auto podría tener múltiples asociaciones "iniciarViaje", en otras palabras, un auto podría realizar múltiples viajes, el valor de multiplicidad se establece en 0..\*

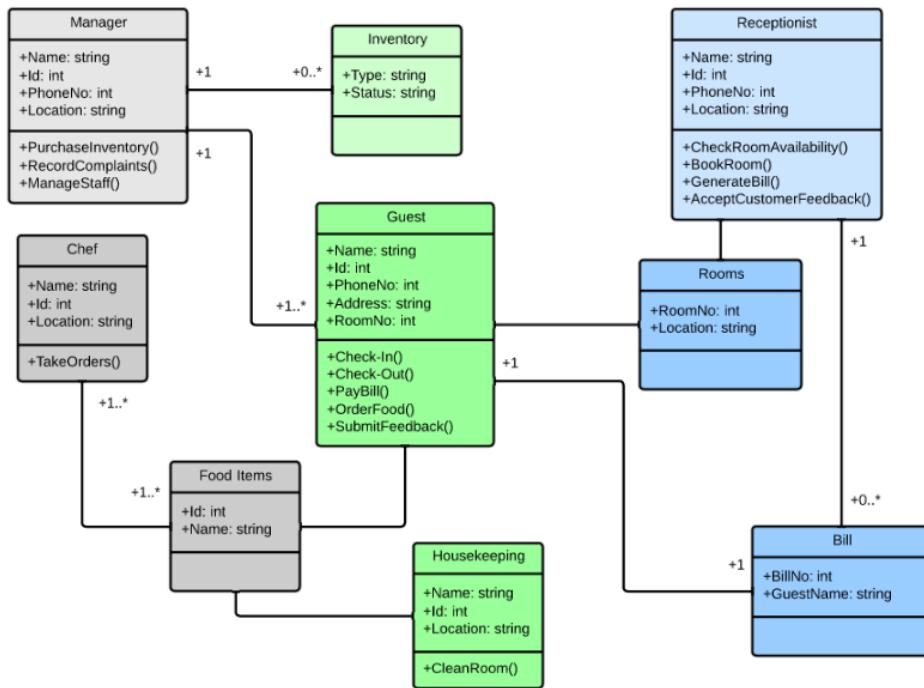
- Asociación unidireccional: Una relación un poco menos común entre dos clases. Una clase está consciente de la otra e interactúa con ella. La asociación unidireccional se dibuja con una línea de conexión recta que señala una punta de flecha abierta desde la clase "knowing" a la clase "known".



### EJEMPLOS:

Un diagrama de clases puede mostrar las relaciones entre cada objeto en un sistema administrativo hotelero, incluidas la información de huéspedes, las responsabilidades del personal y la ocupación por habitación. El siguiente

ejemplo proporciona un panorama útil del sistema administrativo hotelero. Inicia un diagrama de clases haciendo clic en la plantilla siguiente.



Los ATM son aparentemente simples. Aunque los clientes solo necesitan oprimir algunos botones para recibir efectivo, hay muchas capas de seguridad que un ATM seguro y efectivo debe pasar para evitar fraude y brindar valor a los clientes bancarios. Las diversas partes humanas e inanimadas de un sistema de ATM son ilustradas por este diagrama sencillo de leer. Cada clase tiene su título y los atributos se detallan debajo. Puedes editar, guardar y compartir este diagrama abriendo el documento y registrándote a una cuenta gratuita de Lucidchart.

