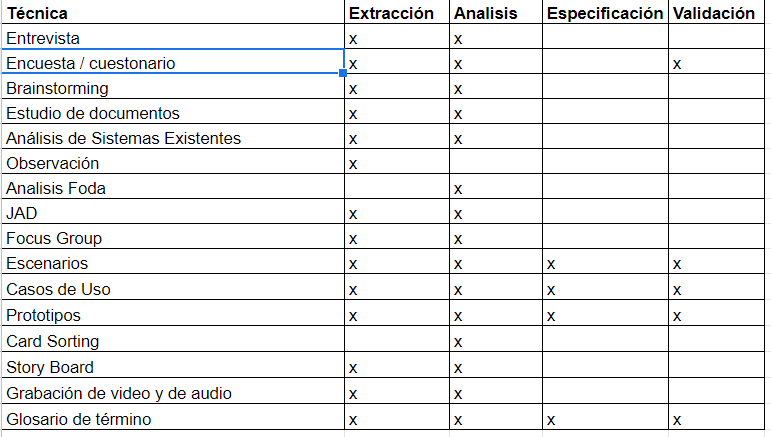
**Técnicas en el análisis de sistemas**

En todo proceso de desarrollo, la comunicación con los stakeholder es un factor importante, debido que son las fuentes de información del contexto, problemas, debilidades, fortaleza de la empresa. Sin embargo, no se puede aplicar todas las técnicas de elicitación.



**Empresa Trans-Turis**

**Objetivo**: Implementar un Sistema para venta de boletos para la empresa Trans-Turis

**Lista de requisitos del sistema**

|  |  |  |
| --- | --- | --- |
| Código | Descripción | Prioridad |
| RF01 | El Sistema debería registrar los pasajes vendido en la lista de pasajeros de cada turno | Alta |
| RF02 | El sistema debe imprimir el boleto entregado a cada pasajero | Alta |
| RF03 | Debe evitar la venta duplicada de asientos en un mismo turno especifico | Alta |
| RF04 | El sistema deberá permitir hacer la reservación de pasaje para un turno específico | Alta |
| RF05 | El sistema deberá controlar la capacidad máxima de cupos de cada bus de la empresa | Alta |
| RF06 | El sistema debe permitir al Vendedor registrar a un pasajero cuando solicite un pasaje |  |
| RF07 | Debe verificar si el pasajero está registrado para vender un pasaje |  |

El sistema permitirá elegir al vendedor un asiento disponible para venderlo

El sistema permitirá cargar las ciudades de destinos dependiendo de la selección de la ciudad de origen

El sistema debe permitir realizar el pago a través de transferencia bancaria, pago por paypal o tarjeta de crédito

El sistema permitir realizar descuentos del costo del pasaje a menores de edad, de tercera edad, y personas con discapacidad.

El sistema permitirá al vendedor realizar búsqueda de rutas

El sistema permitirá al vendedor realizar la compra de pasajes de ida

El sistema permitirá visualizar el número de la unidad en función de la hora o turno asignado

El sistema permitirá buscar al pasajero de acuerdo a un criterio de búsqueda (cedula, apellidos)

El sistema permitirá notificar al pasajero a través de su correo electrónico que la compra del boleto se ha realizado

El sistema permitirá calcular el valor de los pasajes en función al número de boletos adquiridos

El sistema permitirá al vendedor realizar búsqueda de rutas en función a una fecha seleccionada

El sistema permitirá visualizar los horarios o turnos disponibles dependiendo de la ruta seleccionada

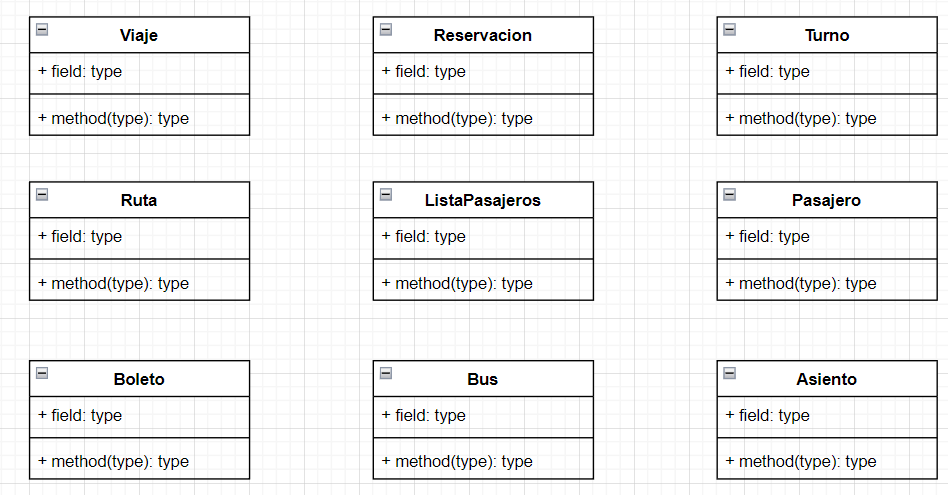
* El sistema debe ser multiusuario
* El sistema debe ser de fácil uso
* El sistema deber estar disponible las 24/7

Lista de frases sustantivas

1. Eliminar sinónimos, duplicados
2. Elementos que no tengan significados para el dominio
3. Cambiar plurales por singulares
4. Eliminar conceptos que no pertenecen al dominio
5. Eliminar conceptos que se consideren atributos
6. Eliminar acciones
7. Eliminar reportes o elemento físicos

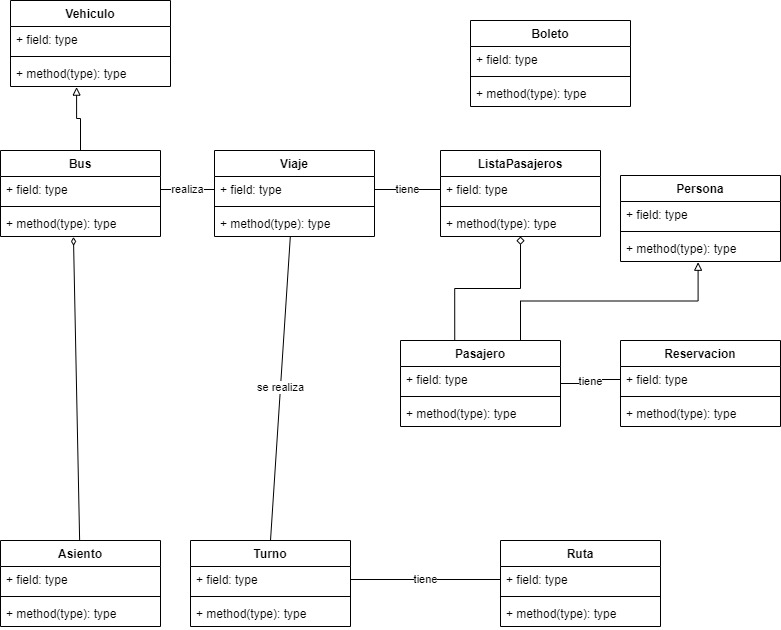
* ~~Sistema~~
* Pasaje~~s~~
* Lista de pasajeros
* Boleto
* ~~ticket~~
* Bus
* Reservación
* Pasajero
* ~~Ventas~~
* Asiento~~s~~
* Turno
* ~~Reporte~~
* ~~Capacidad máxima~~
* Cupo~~s~~
* ~~empresa~~

Modelo del dominio



Encontrar las relaciones

* Un turno tiene una ruta
* Un pasajero tiene una reservación
* Un bus realiza viaje
* Un viaje tiene una lista de pasajeros
* Un bus es un vehículo
* Un pasajero es una persona

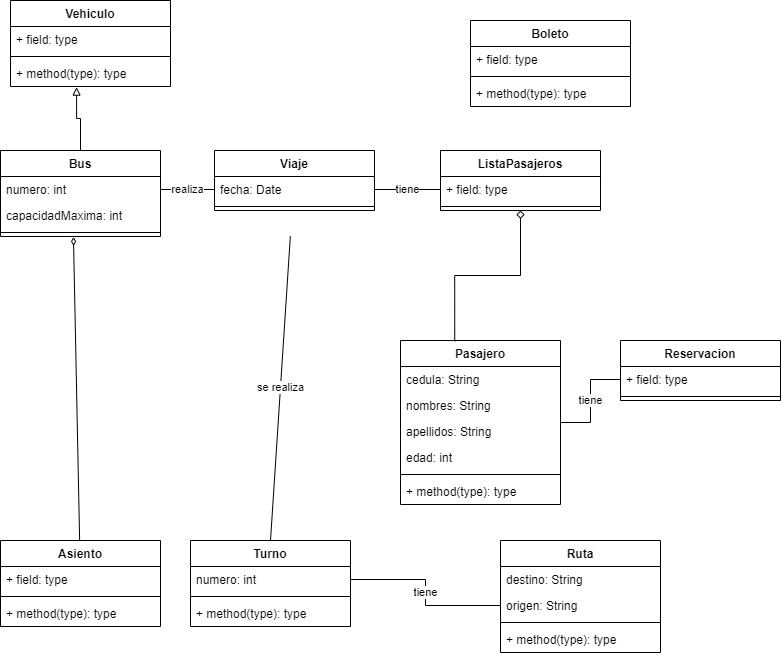


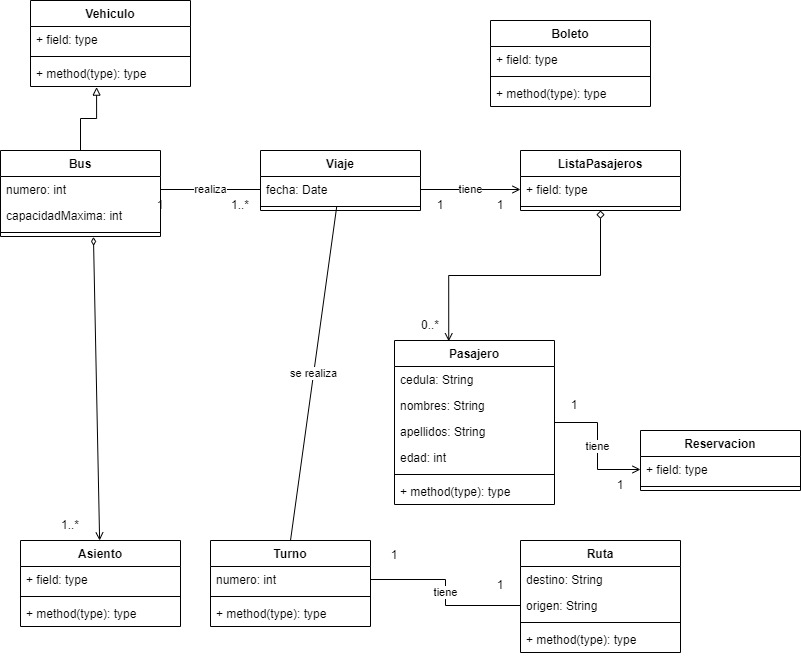
Identificar los atributos

Quién es el pasajero

René Guamán, 40 años

Karen Gonzaga, 20 años





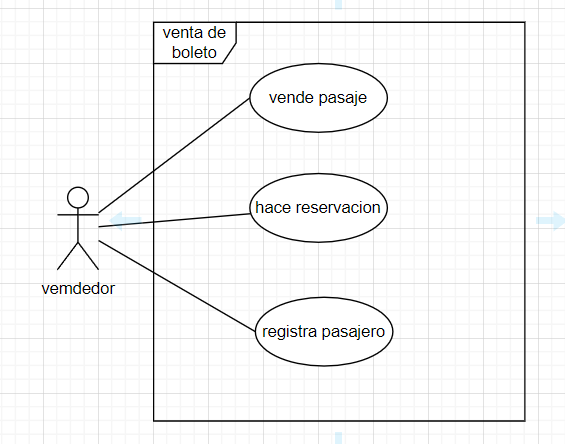
Determinar los Casos de Uso

|  |
| --- |
| Descripción |
| ~~El Sistema debería~~ **~~registrar los pasajes vendido~~** ~~en la lista de pasajeros de cada turno~~ |
| ~~El sistema debe~~ **~~imprimir el boleto~~** ~~entregado a cada pasajero~~ |
| ~~Debe~~ **~~evitar la venta duplicada~~** ~~de asientos en un mismo turno especifico~~ |
| ~~El sistema deberá permitir~~ **~~hacer la reservación~~** ~~de pasaje para un turno específico~~ |
| El sistema deberá **controlar la capacidad máxima de cupos** de cada bus de la empresa |
| ~~El sistema debe~~ **~~registrar a un pasajero~~** ~~cuando solicite un pasaje~~ |
| ~~Debe~~ **~~verificar si el pasajero está registrado~~** ~~para vender un pasaje~~ |

Tabla de UC

|  |  |  |  |
| --- | --- | --- | --- |
| Descripción | UC | ID | Actores |
| Debe **verificar si el pasajero está registrado** para vender un pasaje | Vende pasaje | UC03 | Vendedor |
| El sistema debe **imprimir el boleto** entregado a cada pasajero |
| Debe **evitar la venta duplicada** de asientos en un mismo turno especifico |
| El Sistema debería **registrar los pasajes vendido** en la lista de pasajeros de cada turno |
| El sistema deberá permitir **hacer la reservación** de pasaje para un turno específico | Hace Reservación | UC02 | Vendedor |
| El sistema debe **registrar a un pasajero** cuando solicite un pasaje | Registra pasajero | UC01 | Vendedor |

Diagrama de Use Case



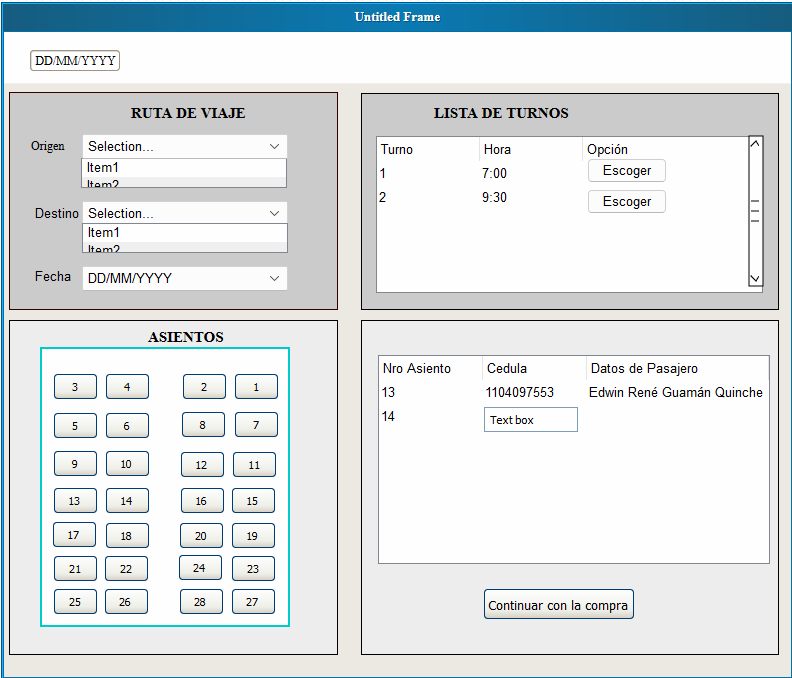
Narración UC Registra Pasajero



Figure 1 Pantalla REGISTRA PASAJERO

|  |  |
| --- | --- |
| Autor/es: René Guamán Fecha: 21 de julio de 2022  Versión 1.0 | |
| Nombre de caso de Uso | Registra Pasajero |
| ID de Caso de Uso | UC01 |
| Prioridad | Alta |
| Fuente / Referencia / requisitos | FR06 |
| Actor primario | Vendedor |
| Actor secundario |  |
| Descripción | El use case registra los datos de un nuevo pasajero o cliente |
| Precondición | El actor se haya autenticado en el sistema  El actor haya ingresa a la plantilla REGISTRA PASAJEROS |
|  |  |
| **Curso Normal de Eventos** | |
| **Acciones de Vendedor** | **Acciones del Sistema** |
| 1. Ingresa los datos del pasajero (cédula, apellidos y nombres, fecha de nacimiento) en la pantalla REGISTRA PASAJERO | 1. Calcula la edad del pasajero |
|  | 1. Clasifica el tipo de pasajero (normal, tercera edad, menor de edad) |
| 1. Selecciona el tipo de discapacidad |  |
| 1. Selecciona la botón **registrar** | 1. Registra los datos del pasajero |
|  | 1. Muestra un mensaje de confirmación “Pasajero registrado |
|  |  |
|  |  |
|  |  |
| **Cursos Alternativos** | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Postcondición |  |

Utilización de la técnica de Análisis de sistemas existentes



El sistema permitirá elegir al vendedor un asiento disponible para venderlo

El sistema permitirá cargar las ciudades de destinos dependiendo de la selección de la ciudad de origen

El sistema debe permitir realizar el pago a través de transferencia bancaria, pago por paypal o tarjeta de crédito

El sistema permitir realizar descuentos del costo del pasaje a menores de edad, de tercera edad, y personas con discapacidad.

El sistema permitirá al vendedor realizar búsqueda de rutas

El sistema permitirá al vendedor realizar la compra de pasajes de ida

El sistema permitirá visualizar el número de la unidad en función de la hora o turno asignado

El sistema permitirá buscar al pasajero de acuerdo a un criterio de búsqueda (cedula, apellidos)

El sistema permitirá notificar al pasajero a través de su correo electrónico que la compra del boleto se ha realizado

El sistema permitirá calcular el valor de los pasajes en función al número de boletos adquiridos

El sistema permitirá al vendedor realizar búsqueda de rutas en función a una fecha seleccionada

El sistema permitirá visualizar los horarios o turnos disponibles dependiendo de la ruta seleccionada

|  |  |
| --- | --- |
| Autor/es: René Guamán Fecha: 27 de julio de 2022  Versión 1.0 | |
| Nombre de caso de Uso | Vende Pasajero |
| ID de Caso de Uso | UC0xxx |
| Prioridad | Alta |
| Fuente / Referencia / requisitos | xxxxxxx |
| Actor primario | Vendedor |
| Actor secundario |  |
| Descripción | El use case empieza al realizar búsquedas de rutas según su origen y destino y desplegar los turnos de acuerdo con la fecha seleccionada. Además, se selecciona un pasaje para asignar a un pasajero. |
| Precondición | El actor se haya autenticado en el sistema  El actor haya ingresa a la pantalla VENTA DE BOLETO y carga la fecha actual |
|  |  |
| **Curso Normal de Eventos** | |
| **Acciones de Vendedor** | **Acciones del Sistema** |
| 1. Selecciona o ingresa una ciudad de origen | 1. busca las ciudades de destino de acuerdo con el origen seleccionado |
|  | 1. carga la lista de ciudades destino en la lista de despliegue **Destino** |
| 1. Selecciona la ciudad de destino |  |
| 1. Selecciona la **fecha** | 1. Busca los turnos de acuerdo a la ruta y la fecha seleccionada |
|  | 1. Carga la lista de turnos en la tabla LISTA DE TURNOS |
| 1. Selecciona la opción **Escoger** | 1. Despliega la distribución de los asientos del bus en la sección ASIENTOS |
|  | 1. Pinta los asientos vendidos de color rojo, los asientos reservados de color verde, y sin color los asientos disponibles para la venta en la sección ASIENTOS |
| 1. Selecciona los asientos | 1. Carga los asiento en la tabla LISTA DE PASAJEROS |
| 1. Ingresa la cédula en el campo de texto cédula de la tabla DATOS PASAJEROS | 1. Busca al pasajero por medio del criterio de búsqueda (cédula) |
|  | 1. Carga los datos del pasajero en la tupla datos pasajeros de la tabla |
| 1. Selecciona la opción Continuar con la compra | 1. El Use Case continua con el Caso de Uso Pagar |
|  |  |
|  |  |
| **Cursos Alternativos** | |
| 1. **Pasajero no registrado** |  |
|  | A.15 muestra la interfaz de registrar un pasajero nuevo |
| ~~A.16. ingresa la cédula, nombres y apellidos~~ |  |
| ~~A.17. selecciona la opción guardar~~ |  |
| A.16. Se invoca al UC registra pasajero |  |
| A.17. El UC retorna al paso 15 del curso normal de eventos |  |
| 1. Ssssssss |  |
| Postcondición |  |

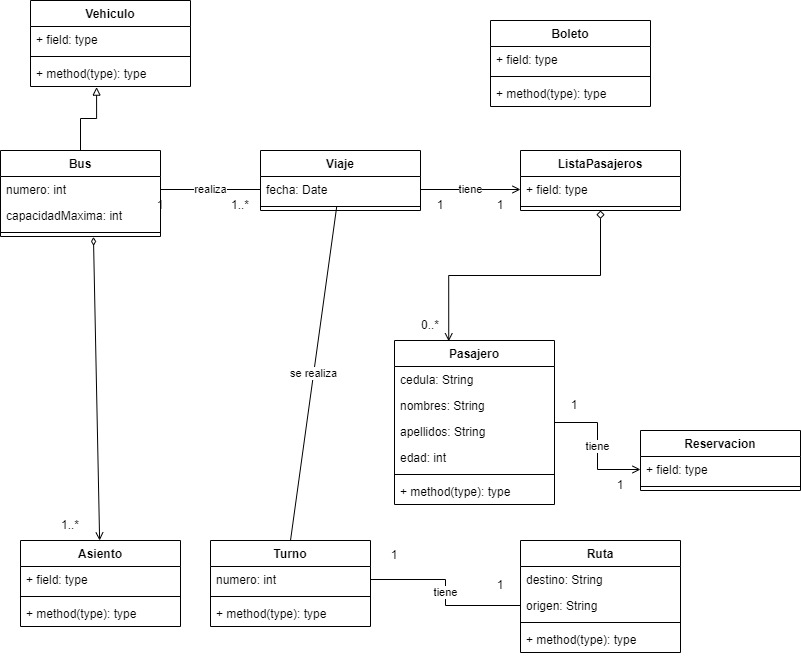
Matriz de Trazabilidad

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Caso de Uso | RF01 | RF02 | RF03 | RF04 | RF05 | RF06 | RF07 |
| Vende boleto | X | X | X |  |  |  | X |
| Hace reservación |  |  |  | X |  |  |  |
| Registra pasajero |  |  |  |  |  | X |  |

Glosario de Términos

|  |  |
| --- | --- |
| GLOSARIO DE TÉRMINOS | |
| Boleto | Documento de pago entregado al usuario o cliente para que lo presente al controlador / ayudante durante el viaje |
| Viaje |  |
| Turno |  |
| Ruta |  |
| Asiento |  |
| ~~Bus~~ |  |

Diagrama del modelo de dominio



La técnica de análisis de Sistemas Existentes permite al equipo de analistas indagar, estudiar comprender cómo otros desarrolladores han resuelto un problema específico a través de un software. Esta técnica permite añadir o adaptar otras funcionalidades, con el objetivo de comprender, mejorar, ajustar y/o su comportamiento.

