

Fundamentos de Ingeniería de Software

Profesor: Raimundo Vega/Juan Pablo Salazar

Ingeniería Civil Informática.

Universidad Austral de Chile

**Todo Transportes Contratos**

**Integrantes:** José Luis Acuña Oyarce.

Luis Barrientos Fajardo.

Kurt Poehler Widemann.

**Fecha**: 18 de julio de 2014.

**Índice:**

[1. Introducción: 3](#_Toc400574841)

[1.1. Estructura del documento: 4](#_Toc400574842)

[2. Contratos: 4](#_Toc400574843)

[2.1. Realizar atención: 4](#_Toc400574844)

[**2.1.1. Ingreso Cliente al sistema:** 4](#_Toc400574845)

[2.2. Matricular cliente: 5](#_Toc400574846)

[**2.2.1. Ingresar documentos: Registro Estudiante:** 5](#_Toc400574847)

[**2.2.2. Ingresar horario de Clase Teorica:** 6](#_Toc400574848)

[**2.2.3. Ingresar horario de Clase Práctica:** 6](#_Toc400574849)

[**2.2.4. Ingresar Pago de matrícula:** 7](#_Toc400574850)

[2.3. Examen visual: 7](#_Toc400574851)

[**2.3.1. Ingresar resultado del examen:** 7](#_Toc400574852)

[**2.3.2. Ingresar datos oftalmológicos:** 8](#_Toc400574853)

[2.4. Registro de clase: 8](#_Toc400574854)

[**2.4.1. Ingresar datos de Asistencia y Contenido:** 8](#_Toc400574855)

[2.5. Realizar Evaluación: 9](#_Toc400574856)

[**2.5.1. Ingresar resultados de exámenes preliminares:** 9](#_Toc400574857)

[**2.5.2. Ingresar resultado de exámenes finales:** 9](#_Toc400574858)

[2.6. Encuesta Calidad: 10](#_Toc400574859)

[**2.6.1. Ingresar resultados de la encuesta:** 10](#_Toc400574860)

[2.7. Clase Práctica: 10](#_Toc400574861)

[**2.7.1. Revisar Deuda del Estudiante:** 10](#_Toc400574862)

[**2.7.2. Ingresar Datos de la Clase Práctica:** 11](#_Toc400574863)

[**2.7.3. Ingresar Datos de date sobre clase práctica:** 12](#_Toc400574864)

[**2.7.4. Ingresar Confirmación del Estudiante:** 12](#_Toc400574865)

[2.8. Examen Práctico: 13](#_Toc400574866)

[**2.8.1. Ingresar Resultados del examen práctico:** 13](#_Toc400574867)

[2.9. Clase cambio rueda: 14](#_Toc400574868)

[**2.9.1. Ingresar Fecha de la Clase de cambio de rueda:** 14](#_Toc400574869)

[**2.9.2. Ingresar Datos Clase Rueda:** 14](#_Toc400574870)

[2.10. Práctica Psicotécnico: 15](#_Toc400574871)

[**2.10.1. Ingresar Fecha Examen Psicotécnico:** 15](#_Toc400574872)

[**2.10.2. Ingresar Resultados Examen Psicotécnico:** 15](#_Toc400574873)

[2.11. Creación de horarios: 16](#_Toc400574874)

[**2.11.1. Ingresar Datos de los módulos:** 16](#_Toc400574875)

[**2.11.2. Solicitud Carta Gantt:** 17](#_Toc400574876)

[**2.11.3. Solicitud Horario Práctico:** 17](#_Toc400574877)

[2.12. Petición vehiculo y examen municipal: 18](#_Toc400574878)

[**2.12.1. Verificar aprobación del estudiante en el examen práctico:** 18](#_Toc400574879)

[**2.12.2. Ingresar Datos del Préstamo:** 18](#_Toc400574880)

[**2.12.3. Ingresar Resultados del examen municipal:** 19](#_Toc400574881)

[2.13. Revisión: 20](#_Toc400574882)

[**2.13.1. Mostrar datos estadísticos:** 20](#_Toc400574883)

[2.14. Generar Registro de Vehículos: 20](#_Toc400574884)

[**2.14.1. Ingreso datos de los vehículos:** 20](#_Toc400574885)

[2.15. Gestión de Flota: 21](#_Toc400574886)

[**2.15.1. Solicitud de estado de vehículo:** 21](#_Toc400574887)

[2.16. Añadir Usuario: 22](#_Toc400574888)

[**2.16.1. Añadir nombre de usuario y contraseña:** 22](#_Toc400574889)

[**2.16.2. Ingreso nuevo usuario en el sistema:** 23](#_Toc400574890)

[2.17. Cambiar Contraseña de Usuario: 23](#_Toc400574891)

[**2.17.1. Cambiar Contraseña de Usuario:** 23](#_Toc400574892)

**1. Introducción:**

En esta sección del trabajo se pretende mostrar los contratos de trabajo que son los documentos que contribuyen a definir el comportamiento de un sistema en términos de los cambios de estado que éste experimenta cuando se invoca una operación.

El objetivo de hacer estos contratos nos permitirá establecer un diseño más formal del trabajo siendo éstos la base del diseño del sistema entonces se debe de hacer de una manera más específica las funciones del sistema en contraste a los casos de uso que solo nombraban las tareas en su curso normal de los eventos. Además como ya se dio cuenta el lector de este archivo, definimos estos contratos en base a los casos de uso del sistema, específicamente sus tareas mostradas en el curso normal de los eventos, pero especificando que es lo que hará cada tarea, pero también no se mostrará cómo se hará dicha tarea en particular. Esto con el motivo de que el desarrollo de las tareas se mostrará en fases siguientes del proceso de desarrollo del producto software.

**1.1. Estructura del documento:**

En la siguiente sección de este documento se pretende mostrar cada caso de uso definido previamente, en cada uno de ellos se presentará uno o más contratos los cuáles indicarán que hace cada una de las tareas que ingresan desde los actores al sistema, se presentará su estructura que es general para cada contrato y es así:

Estructura del contrato:

* **Nombre:** Nombre del contrato.
* **Responsabilidades:** Descripción informal de su funcionamiento
* **Tipo:** Nombre del tipo de contrato.
* **Referencias Cruzadas:** Referencia a las funciones del sistema.
* **Notas:** Son referencias del diseño del sistema en base a la operación del contrato.
* **Excepciones:** Problemas surgidos, nacen de los casos de uso.
* **Salida:** Salidas relacionadas con el sistema, sin interfaz o nada que salga del sistema.
* **Precondiciones:** Estado del sistema antes de comenzar la operación.
* **Poscondiciones:** Estado del sistema después de la operación.

Con esto se ha comprobado que el contrato no es algo simple de realizar, se necesita un gran conocimiento de lo que debe hacer el sistema además de trabajar en relación tanto con los casos de uso como sus diagramas de procesos (análisis) para poder mostrar que procesos se deben de hacer contratos y cuáles no.

# **2. Contratos:**

## **2.1. Realizar atención:**

### **2.1.1. Ingreso Cliente al sistema:**

|  |  |
| --- | --- |
| **Nombre:** | Ingresar\_cliente(nombre: cadena, fecha\_consulta: date) |
| **Responsabilidades:** | Registrar a un cliente potencial en el sistema. |
| **Tipo:** | Sistema. |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ01, REQ02, REQ03, REQ04. * Caso de Uso: Realizar atención. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | Cliente no fue registrado, debido a inconsistencias en los datos. Se reporta el error indicando volver a ingresar los datos. |
| **Salida:** | * Registra\_cliente. |
| **Precondiciones:** | * Funcionario identificado como usuario**.** * Cliente no debe estar registrado. |
| **Poscondiciones:** | * Si se trata de un nuevo cliente entonces se crea la instancia *Cliente* (creación de instancia). * Se asignó un *nombre* a *Cliente.nombre* (modificación de atributos). * Se asoció una instancia *Cliente* a la *ListaCliente* basada en la correspondencia de la fecha\_consulta (asociación formada). |

## **2.2. Matricular cliente:**

### **2.2.1. Ingresar documentos: Registro Estudiante:**

|  |  |
| --- | --- |
| **Nombre:** | Ingresar\_documentos\_registro (nombre: cadena, Rut: cadena, fecha\_ingreso: date, id\_carnet: bool, foto: numero, menor\_edad: bool, cert\_est: bool, tipo: cadena) |
| **Responsabilidades:** | Ingresar documentos y registrar al estudiante al sistema. |
| **Tipo:** | Sistema. |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ01, REQ02, REQ03, REQ04. * Caso de Uso: Matricular cliente. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | * Faltan documentos esenciales. Se reporta error, y se añade en estado de moroso. * Estudiante no fue registrado, debido a inconsistencias en los datos. Se reporta el error indicando volver al paso anterior para repetir el proceso. |
| **Salida:** | * Procesar\_Documentos * Registrar\_Estudiante. |
| **Precondiciones:** | * Funcionario identificado como usuario**.** * Cliente no debe estar registrado. |
| **Poscondiciones:** | * Si se trata de un nuevo estudiante entonces se crea la instancia *Estudiante* basada en la correspondencia del *id\_carnet, foto, menor\_edad, cert\_est* (creación de instancia). * Se asoció una instancia *Estudiante* a la *ListaEstudiante* basada en la correspondencia de la *fecha\_ingreso* (asociación formada). |

### **2.2.2. Ingresar horario de Clase Teorica:**

|  |  |
| --- | --- |
| **Nombre:** | Ingresar\_horario\_teorico (h\_teorico: date, Estudiante: Estudiante) |
| **Responsabilidades:** | Ingresar el horario de la clase teorica al sistema. |
| **Tipo:** | Sistema. |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ34, REQ35, REQ36, REQ37, REQ49. * Caso de Uso: Matricular cliente. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | Horario entregado no coincide con los horarios actuales de clases. Se reporta el error, indicando la entrega de un nuevo horario, el proceso se detiene. |
| **Salida:** | * Procesa\_horario\_clase\_teorica. * Asociar\_clase\_teorica. |
| **Precondiciones:** | * Funcionario identificado como usuario**.** * Cliente debe estar registrado como Estudiante. |
| **Poscondiciones:** | * Se asoció una instancia *Estudiante* a la *ClaseTeorica* basada en la correspondencia de la *h\_teorico* (asociación formada). |

### **2.2.3. Ingresar horario de Clase Práctica:**

|  |  |
| --- | --- |
| **Nombre:** | Ingresar\_horario\_practico (h\_practico: date, Estudiante: Estudiante) |
| **Responsabilidades:** | Ingresar el horario de la clase práctica al sistema. |
| **Tipo:** | Sistema. |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ34, REQ35, REQ36, REQ37, REQ49, REQ50, REQ51. * Caso de Uso: Matricular cliente. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | Horario entregado no coincide con los horarios actuales de clases. Se reporta el error, indicando la entrega de un nuevo horario, el proceso se detiene. |
| **Salida:** | * Procesa\_horario\_clase\_practica. * Asociar\_clase\_practica. |
| **Precondiciones:** | * Funcionario identificado como usuario**.** * Cliente debe estar registrado como Estudiante. |
| **Poscondiciones:** | * Se asoció una instancia *Estudiante* a la *ClasePractica* basada en la correspondencia de la *h\_practico* (asociación formada). |

### **2.2.4. Ingresar Pago de matrícula:**

|  |  |
| --- | --- |
| **Nombre:** | Ingresar\_pago (pago\_matricula: numero, Estudiante: Estudiante). |
| **Responsabilidades:** | Ingresar pago de matrícula al sistema. |
| **Tipo:** | Sistema: |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ05, REQ06, REQ07, REQ08, REQ58, REQ59, REQ60, REQ61. * Caso de Uso: Matricular cliente. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | Pago no aceptado por ser menos que la cantidad mínima requerida. Se reporta el error, indicando la entrega de una cantidad suficiente igual o superior a la mínima requerida. |
| **Salida:** | Procesar\_pago. |
| **Precondiciones:** | * Funcionario identificado como usuario**.** |
| **Poscondiciones:** | * Si se trata de un nuevo ficha económica del estudiante entonces se crea la instancia *E\_Financiero\_Estudiante* (creación de instancia). * Se asoció una instancia *Estudiante* a *E\_Financiero\_Estudiante* basada en la correspondencia de *Pago* (asociación formada). |

## **2.3. Examen visual:**

### **2.3.1. Ingresar resultado del examen:**

|  |  |
| --- | --- |
| **Nombre:** | Ingresar\_resultado\_examen (apobado:bool) |
| **Responsabilidades:** | Ingresar resultado de Examen Visual al Sistema. |
| **Tipo:** | Sistema. |
| **Referencias Cruzadas:** | * REQ01, REQ02,REQ04 * Caso de Uso: Examen visual. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | N/A. |
| **Salida:** | Estado del examen. |
| **Precondiciones:** | * Funcionario identificado como usuario**.** * Cliente debe estar registrado como Estudiante. |
| **Poscondiciones:** | * Se crea la instancia *Documento* con *tipo: EXAMEN\_VISUAL* y *estado: estado* como argumentos (creación de instancia). * Se asoció instancia *Documento* a *Lista\_Documentos basada en la correspondencia de Documento.* |

### **2.3.2. Ingresar datos oftalmológicos:**

|  |  |
| --- | --- |
| **Nombre:** | Ingresar\_examen\_oft(aprobado: cadena, detalles: cadena) |
| **Responsabilidades:** | Actualizar en Sistema el estado de aprobación del examen visual y registrar detalles del examen oftalmológico. |
| **Tipo:** | Sistema. |
| **Referencias Cruzadas:** | * REQ01, REQ03 * Caso de Uso: Examen Visual. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | N/A. |
| **Salida:** | Estado del examen. |
| **Precondiciones:** | * Funcionario identificado como usuario**.** * Cliente debe estar registrado como Estudiante. |
| **Poscondiciones:** | * Se crea la instancia *Documento* con *tipo: EXAMEN\_OFT,*  *estado: estado y detalles: detalles* como argumentos (creación de instancia). * Se asoció instancia *Documento* a *Lista\_Documentos* basada en la correspondencia de *Documento.* |

## **2.4. Registro de clase:**

### **2.4.1. Ingresar datos de Asistencia y Contenido:**

|  |  |
| --- | --- |
| **Nombre:** | Ingresar\_Asistencia( profesor: cadena, fecha: date, modulo: cadena, curso: cadena, alumnos: bool[] ) |
| **Responsabilidades:** | Ingresar al sistema la asistencia diaria de cada módulo. |
| **Tipo:** | Sistema. |
| **Referencias Cruzadas:** | * REQ01, REQ02, REQ04, REQ28, REQ29, REQ31 * Caso de Uso: Registro de Clase. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | N/A |
| **Salida:** | Mensaje de transacción exitosa. |
| **Precondiciones:** | * Funcionario identificado como usuario**.** * Clientes deben estar registrados como Estudiantes. * Relator debe estar registrado como tal. * Debe existir curso. * Relator debe estar asociado al módulo. * Estudiantes deben estar asociados al módulo. * Date debe pertenecer al rango de duración del módulo. |
| **Poscondiciones:** | * Si es primer ingreso, crea instancia de *Asistencia*. * Asocia *Profesor*, *Modulo*, *Curso*, y cada *Estudiante* con *Asistencia* basada en la correspondencia de *fecha*. |

## **2.5. Realizar Evaluación:**

### **2.5.1. Ingresar resultados de exámenes preliminares:**

|  |  |
| --- | --- |
| **Nombre:** | Ingresar\_EXT (date: date, Estudiante: cadena, curso: cadena, aprobado: bool, tipo: numero) |
| **Responsabilidades:** | Ingresar resultado de examen teórico preliminar da cada Estudiante en el Sistema. |
| **Tipo:** | Sistema. |
| **Referencias Cruzadas:** | * REQ01, REQ02, REQ03, REQ22, REQ28 * Caso de Uso: Realizar evaluación. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | N/A |
| **Salida:** | Ex\_v0 |
| **Precondiciones:** | * Funcionario identificado como usuario**.** * Clientes deben estar registrados como Estudiantes. * Debe existir curso. * Estudiantes deben estar asociados al módulo. * Date debe pertenecer al rango de duración del módulo. * Tipo = 1: preliminar. |
| **Poscondiciones:** | * Se crea instancia de *Examen* con *Tipo* como parámetro. * Se asocia *Estudiante*, *Curso* y *Aprobado* con *Examen* basada en la correspondencia de *Date*. |

### **2.5.2. Ingresar resultado de exámenes finales:**

|  |  |
| --- | --- |
| **Nombre:** | Ingresar\_EXT(fecha: date, Estudiante: cadena, curso: cadena, aprobado: bool, tipo: numero) |
| **Responsabilidades:** | Ingresar resultado de examen teórico final da cada Estudiante en el Sistema. |
| **Tipo:** | Sistema. |
| **Referencias Cruzadas:** | * REQ01, REQ02, REQ03, REQ22, REQ28 * Caso de Uso: Realizar evaluación. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | N/A |
| **Salida:** | Ex\_v0 |
| **Precondiciones:** | * Funcionario identificado como usuario**.** * Clientes deben estar registrados como Estudiantes. * Debe existir curso. * Estudiantes deben estar asociados al módulo. * Date debe pertenecer al rango de duración del módulo. * Tipo = 0: final. |
| **Poscondiciones:** | * Se crea instancia de *Examen* con *Tipo* como parámetro. * Se asocia *Estudiante*, *Curso* y *Aprobado* con *Examen* basada en la correspondencia de *Date*. |

## **2.6. Encuesta Calidad:**

### **2.6.1. Ingresar resultados de la encuesta:**

|  |  |
| --- | --- |
| **Nombre:** | Ingresar\_Resultados\_Encuesta (estudiante: cadena, res: bool) |
| **Responsabilidades:** | Ingresar respuestas al sistema para poder procesarlas. |
| **Tipo:** | Sistema: |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ09, REQ10, REQ11, REQ12. * Caso de Uso: Encuesta Calidad 01. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | Sistema no registra los datos debido a inconsistencias en el ingreso de estos. Se reporta error, sugiere el reingreso de los datos. |
| **Salida:** | Procesa\_Respuestas |
| **Precondiciones:** | * Funcionario Académico y de Coordinación identificado como Usuario**.** * Estudiante matriculado. * Clases Teóricas realizadas. |
| **Poscondiciones:** | * Si se trata de una nueva encuesta se crea *Encuesta* (creación de instancias). * Se asoció una instancia *estudiante* a *Encuesta* basada en la correspondencia de *res* (asociación formada). |

## **2.7. Clase Práctica:**

### **2.7.1. Revisar Deuda del Estudiante:**

|  |  |
| --- | --- |
| **Nombre:** | Revisar\_Deuda\_Estudiante (estudiante: cadena ) |
| **Responsabilidades:** | Verificar si el estudiante posee alguna deuda sin pagar. |
| **Tipo:** | Sistema: |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ05, REQ06, REQ58, REQ59, REQ60, REQ61. * Caso de Uso: Clase Práctica. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | Sistema indica al Estudiante que posee deudas sin pagar o documentos aún no entregados. Se reporta error especificando si es una deuda o un documento faltante, solicitando una solución por parte del Estudiante, mientras no se solucione, se paraliza el proceso. |
| **Salida:** | Muestra\_Situacion\_Financiera |
| **Precondiciones:** | * Funcionario Académico y de Coordinación identificado como Usuario**.** * Instructor identificado en el Sistema como tal. * Estudiante matriculado. * Clases Teóricas realizadas. |
| **Poscondiciones:** | * Si no se ha creado una situación financiera entonces se crea una instancia *Situacion\_Financiera* (creación de instancia). * De haber una deuda asociada al *estudiante* se crea la instancia *deuda* (creación de instancias). * Se asocia el *estudiante* con *Situacion\_Financiera* basada en la correspondencia de *deuda* (asociación formada). |

### **2.7.2. Ingresar Datos de la Clase Práctica:**

|  |  |
| --- | --- |
| **Nombre:** | Ingreso\_Datos\_Clase\_Practica (instructor: cadena, estudiante: cadena, datos\_practica: bool) |
| **Responsabilidades:** | Ingresar los datos asociados a la clase práctica |
| **Tipo:** | Sistema: |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ50, REQ51, REQ52, REQ53. * Caso de Uso: Clase Práctica. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | Sistema no registra los datos debido a inconsistencias en el ingreso de estos. Se reporta error, sugiere el reingreso de los datos. |
| **Salida:** | Procesa\_Datos\_Clase\_Practica |
| **Precondiciones:** | * Funcionario Académico y de Coordinación identificado como Usuario**.** * Instructor identificado en el Sistema como tal. * Estudiante matriculado. * Clases Teóricas realizadas. |
| **Poscondiciones:** | * Se crea la clase práctica, si es que no existe, la cual se llamará *ClasePractica* (Creación de instancia). * Se asocia *instructor* y *estudiante* con *ClasePractica* basada en la correspondencia de *datos\_practica* (Asociación formada). |

### **2.7.3. Ingresar Datos de date sobre clase práctica:**

|  |  |
| --- | --- |
| **Nombre:** | Ingreso\_Datos\_Solicitados (Contenido: bool, hora\_clase: tiempo, fecha\_clase: date ) |
| **Responsabilidades:** | Ingresar los datos de la hora y date asociados a la clase práctica. |
| **Tipo:** | Sistema: |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ50, REQ51, REQ52, REQ53. * Caso de Uso: Clase Práctica. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | Sistema no registra los datos debido a inconsistencias en el ingreso de estos. Se reporta error, sugiere el reingreso de los datos. |
| **Salida:** | Procesa\_Datos\_Solicitados |
| **Precondiciones:** | * Funcionario Académico y de Coordinación identificado como Usuario**.** * Instructor identificado en el Sistema como tal. * Estudiante matriculado. * Clases Teóricas realizadas. |
| **Poscondiciones:** | * Se asocia *hora\_clase* y *fecha\_clase* con *ClasePractica* basada en la correspondencia de *Contenido* (Asociación formada). |

### **2.7.4. Ingresar Confirmación del Estudiante:**

|  |  |
| --- | --- |
| **Nombre:** | Ingreso\_Confirmacion (estudiante: cadena, Contenido: bool, Confirmacion: bool) |
|  | Ingresar confirmación del estudiante de que tomó la clase práctica y está listo para rendir examen. |
| **Tipo:** | Sistema: |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ50, REQ51, REQ52, REQ53. * Caso de Uso: Clase Práctica. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | Sistema rechaza confirmación del Estudiante. Se reporta error, sugiere el reingreso de la confirmación. |
| **Salida:** | Procesar\_Confirmacion |
| **Precondiciones:** | * Funcionario Académico y de Coordinación identificado como Usuario**.** * Instructor identificado en el Sistema como tal. * Estudiante matriculado. * Clases Teóricas realizadas. |
| **Poscondiciones:** | * Se asocia *nombre\_estudiante* y *Contenido* con *ClasePractica* basada en la correspondencia de *Confirmación* (Asociación formada). |

## **2.8. Examen Práctico:**

### **2.8.1. Ingresar Resultados del examen práctico:**

|  |  |
| --- | --- |
| **Nombre:** | Ingresar\_Resultados\_Examen\_Practico (Estudiante: cadena, Confirmacion: bool) |
| **Responsabilidades:** | Ingresar los resultados del examen práctico para asegurar al estudiante su aprobación o reprobación. |
| **Tipo:** | Sistema: |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ22, REQ23, REQ24, REQ25, REQ26, REQ27, REQ54, REQ55. * Caso de Uso: Examen Práctico. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | Sistema indica que el Estudiante no aprobó el examen práctico. Se reporta error, se avisa al Estudiante para que realiza un examen de reemplazo, el cual será evaluado e ingresado al Sistema de reprobar también debe rehacer el examen. |
| **Salida:** | Procesar\_Resultados\_Examenes |
| **Precondiciones:** | * Funcionario Académico y de Coordinación identificado como Usuario**.** * Instructor identificado en el Sistema como tal. * Estudiante matriculado. * Clases Teóricas realizadas. |
| **Poscondiciones:** | * Si no existe un examen práctico previo se crea la instancia *ExamenPractico* (creación de instancia). * Se asoció una instancia *estudiante* a *ExamenPractico* basada en la correspondencia de *res\_01* (asociación formada). |

## **2.9. Clase cambio rueda:**

### **2.9.1. Ingresar Fecha de la Clase de cambio de rueda:**

|  |  |
| --- | --- |
| **Nombre:** | Ingreso\_Fecha\_Clase (estudiante: cadena, Hora\_rueda: time, Fecha\_rueda: date) |
| **Responsabilidades:** | Ingresar la fecha de la clase de cambio de rueda en el sistema. |
| **Tipo:** | Sistema: |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ50, REQ51, REQ52, REQ53. * Caso de Uso: Clase cambio rueda. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | * El Sistema no realiza el registro debido a inconsistencias en el ingreso de datos. Se reporta el error, se sugiere el reingreso de los datos. |
| **Salida:** | * Registra\_Horario\_Clase * Elimina\_Horario |
| **Precondiciones:** | * Funcionario Académico y de Coordinación identificado como Usuario**.** * Instructor identificado en el Sistema como tal. * Estudiante debe estar matriculado. * Clases prácticas realizadas |
| **Poscondiciones:** | * Si no existe una clase de cambio de rueda previo se crea la instancia *Clase\_CRueda* (creación de instancia). * Se asoció una instancia *estudiante* a *Clase\_CRueda* basada en la correspondencia de *Hora\_rueda* y *Fecha\_rueda*  (asociación formada). |

### **2.9.2. Ingresar Datos Clase Rueda:**

|  |  |
| --- | --- |
| **Nombre:** | Ingreso\_Datos\_Clase\_Rueda (datos\_rueda: bool) |
| **Responsabilidades:** | Ingresar la fecha de la clase de cambio de rueda en el sistema. |
| **Tipo:** | Sistema: |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ50, REQ51, REQ52, REQ53. * Caso de Uso: Clase cambio rueda. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | * Estudiante no pertenece a clase B. Fin de la operación. * El Sistema no realiza el registro debido a inconsistencias en el ingreso de datos. Se reporta el error, se sugiere el reingreso de los datos. |
| **Salida:** | * Registra\_Datos |
| **Precondiciones:** | * Funcionario Académico y de Coordinación identificado como Usuario**.** * Instructor identificado en el Sistema como tal. * Estudiante debe estar matriculado. * Clases prácticas realizadas |
| **Poscondiciones:** | * Se asignó un *datos\_rueda* a *Clase\_CRueda.datos\_rueda* (modificación de atributos). |

## **2.10. Práctica Psicotécnico:**

### **2.10.1. Ingresar Fecha Examen Psicotécnico:**

|  |  |
| --- | --- |
| **Nombre:** | Ingreso\_Fecha\_Psicotecnico (estudiante: cadena, Hora\_pt: time, Fecha\_pt: date) |
| **Responsabilidades:** | Ingresar la fecha del examen psicotécnico dada por el estudiante. |
| **Tipo:** | Sistema: |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ54, REQ55. * Caso de Uso: Práctica Psicotécnico. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | * Fecha no disponible para la prueba Se reporta error. Se exige una nueva fecha. |
| **Salida:** | * Registra\_Fecha\_Psicotecnico * Elimina\_Horario |
| **Precondiciones:** | * Funcionario Académico y de Coordinación identificado como Usuario**.** * Estudiante debe estar matriculado. * Clases prácticas realizadas. |
| **Poscondiciones:** | * Si no existe un examen psicotécnico se crea la instancia *Examen\_PT* (creación de instancia). * Se asoció una instancia *estudiante* a *Examen\_PT* basada en la correspondencia de *Hora\_pt* y *Fecha\_pt*  (asociación formada). |

### **2.10.2. Ingresar Resultados Examen Psicotécnico:**

|  |  |
| --- | --- |
| **Nombre:** | Ingresar\_Resultados\_Psicotecnico (res\_pt: bool) |
| **Responsabilidades:** | Ingresar los resultados del examen psicotécnico dada por el estudiante en el sistema. |
| **Tipo:** | Sistema: |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ22, REQ23, REQ24, REQ25, REQ26, REQ27. * Caso de Uso: Práctica Psicotécnico. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | * Sistema indica que el Estudiante no aprobó el examen psicotécnico. Se reporta error, se avisa al Estudiante para que realiza un examen de reemplazo, el cual será evaluado e ingresado al Sistema de reprobar también debe rehacer el examen. |
| **Salida:** | * Procesa\_Resultados\_Psicotecnicos |
| **Precondiciones:** | * Funcionario Académico y de Coordinación identificado como Usuario**.** * Estudiante debe estar matriculado. * Clases prácticas realizadas. |
| **Poscondiciones:** | * Se asignó un *res\_pt* a *Examen\_PT.*res\_pt (modificación de atributos). |

## **2.11. Creación de horarios:**

### **2.11.1. Ingresar Datos de los módulos:**

|  |  |
| --- | --- |
| **Nombre:** | Registrar\_Datos (Datos\_Teorico: bool, Datos\_Practico: bool,  alumnos: bool[]) |
| **Responsabilidades:** | Ingresar los datos de los módulos de las clases para la creación de horarios y la carta Gantt. |
| **Tipo:** | Sistema. |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ34, REQ35, REQ36, REQ37. * Caso de Uso: Creación de horarios. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | * El Sistema no realiza el registro debido a inconsistencias en el ingreso de datos. Se reporta el error, se sugiere el reingreso de los datos. |
| **Salida:** | * Genera\_Carta\_Gantt * Genera\_Tablas\_Horario\_Estudiante |
| **Precondiciones:** | * Funcionario Académico y de Coordinación identificado como Usuario**.** |
| **Poscondiciones:** | * Si no existe una Carta Gantt se crea la instancia *Carta\_gantt* (creación de instancia). * Se asignó a *Carta\_gantt* un *Datos\_Teorico* y *Datos\_Practico* (modificación de atributos) * Si no existe una Tabla de estudiante se crea la instancia *Tabla\_estudiante* (creación de instancia). * Se asoció una instancia *alumnos* a *Tabla\_estudiante* basada en la correspondencia de *Datos\_Teorico* y *Datos\_Practico* (asociación formada). |

### **2.11.2. Solicitud Carta Gantt:**

|  |  |
| --- | --- |
| **Nombre:** | Solicitud\_Carta\_Gantt (C\_Gantt: bool) |
| **Responsabilidades:** | Solicitar una carta Gantt por parte del Funcionario Administrativo |
| **Tipo:** | Sistema |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ36, REQ37. * Caso de Uso: Creación de horarios. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** |  |
| **Salida:** | * Mostrar\_Carta\_Gantt |
| **Precondiciones:** | * Funcionario Académico y de Coordinación identificado como Usuario**.** |
| **Poscondiciones:** | * Se asoció una instancia *Carta\_gantt* un *C\_Gantt* (asociación formada). |

### **2.11.3. Solicitud Horario Práctico:**

|  |  |
| --- | --- |
| **Nombre:** | Solicitud\_Horario\_Practico (estudiante: cadena) |
| **Responsabilidades:** | Solicitar un horario práctico para un estudiante por parte del Funcionario Administrativo |
| **Tipo:** | Sistema |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ36, REQ37. * Caso de Uso: Creación de horarios. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** |  |
| **Salida:** | * Mostrar\_Carta\_Gantt |
| **Precondiciones:** | * Funcionario Académico y de Coordinación identificado como Usuario**.** |
| **Poscondiciones:** | * Se asoció una instancia *estudiante* a *Tabla\_estudiante* (asociación formada). |

## **2.12. Petición vehiculo y examen municipal:**

### **2.12.1. Verificar aprobación del estudiante en el examen práctico:**

|  |  |
| --- | --- |
| **Nombre:** | Verifica\_Aprobacion\_Examen\_Practico (Estudiante: Estudiante) |
| **Responsabilidades:** | El Funcionario Académico y de Coordinación debe de verificar si el estudiante que tomará el examen municipal tiene aprobado antes el examen práctico. |
| **Tipo:** | Sistema. |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ22, REQ23, REQ24, REQ25, REQ26, REQ27. * Caso de Uso: Petición vehiculo y examen municipal |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | * Sistema indica que el Estudiante no aprobó el examen práctico. Se reporta error, se avisa al Estudiante para que realiza un examen de reemplazo, el cual será evaluado e ingresado al Sistema de reprobar también debe rehacer el examen. |
| **Salida:** | * Procesa\_Examen\_Practico |
| **Precondiciones:** | * Funcionario Académico y de Coordinación identificado como Usuario**.** * Estudiante debe estar matriculado. * Clases prácticas realizadas. * Examen práctico aprobado. |
| **Poscondiciones:** | * Se asoció una instancia *estudiante* a *ExamenPractico* (asociación formada). * Si no existe un Examen Municipal se crea la instancia *Examen\_Municipal* (creación de instancia). |

### **2.12.2. Ingresar Datos del Préstamo:**

|  |  |
| --- | --- |
| **Nombre:** | Ingresar\_Datos\_Vehiculo (matricula: cadena, Estudiante: Estudiante, fecha\_examen: Fecha) |
| **Responsabilidades:** | Ingresar los datos concretos del préstamo para el examen municipal. |
| **Tipo:** | Sistema |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ13, REQ14, REQ15, REQ16, REQ17, REQ18, REQ19, REQ20, REQ21, REQ45, REQ46, REQ47, REQ48, REQ49,REQ50, REQ51, REQ52, REQ53,REQ54, REQ55. * Caso de Uso: Petición vehiculo y examen municipal. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | * El Sistema no realiza el registro debido a inconsistencias en el ingreso de datos. Se reporta el error, se sugiere el reingreso de los datos. |
| **Salida:** | * Procesar\_Prestamo |
| **Precondiciones:** | * Funcionario Académico y de Coordinación identificado como Usuario**.** * Estudiante debe estar matriculado. * Clases prácticas realizadas. * Examen práctico aprobado. |
| **Poscondiciones:** | * Si no existe un Préstamo se crea la instancia *Prestamo* (creación de instancia). * Se asignó a *Prestamo*  un *matricula* y *Estudiante* y *Fecha\_examen*  (modificación de atributos) * Se asoció una instancia *Prestamo* a *Examen\_Municipal* basada en la correspondencia de *Estudiante* y *Fecha\_examen* (asociación formada). |

### **2.12.3. Ingresar Resultados del examen municipal:**

|  |  |
| --- | --- |
| **Nombre:** | Ingresar\_Resultados\_Examen\_Municipal (res\_municipal:boolean) |
| **Responsabilidades:** | Ingresar los resultados del examen municipal para su evaluación |
| **Tipo:** | Sistema |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ21,REQ22, REQ23, REQ24, REQ25, REQ26, REQ27. * Caso de Uso: Petición vehiculo y examen municipal. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | * Sistema indica que el Estudiante no aprobó el examen municipal. Se reporta error, se avisa al Estudiante que debe tomar el curso de reforzamiento (con todo lo que eso implica) para que realice un examen de reemplazo, el cual será evaluado e ingresado al Sistema y poder obtener el certificado, si aprueba. De no ser así entonces se termina el proceso. |
| **Salida:** | * Procesar\_Resultados\_Municipal |
| **Precondiciones:** | * Funcionario Académico y de Coordinación identificado como Usuario**.** * Estudiante debe estar matriculado. * Clases prácticas realizadas. * Examen práctico aprobado. |
| **Poscondiciones:** | * Se asoció una instancia *Res\_municipal* a *Examen\_Municipal* basada en la correspondencia de *Estudiante* y *Fecha\_examen* (asociación formada). |

## **2.13. Revisión:**

### **2.13.1. Mostrar datos estadísticos:**

|  |  |
| --- | --- |
| **Nombre:** | Solicita\_Datos\_Estadisticos (D\_est: bool) |
| **Responsabilidades:** | Solicitar datos estadísticos con el objetivo de presentar información estadística de la empresa |
| **Tipo:** | Sistema |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ07, REQ08. * Caso de Uso: Revisión. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | * El Sistema no realiza el registro debido a inconsistencias en el ingreso de datos. Se reporta el error, se sugiere el reingreso de los datos. |
| **Salida:** | * Muestra\_Datos\_Estadisticos * Muestra\_Graficos |
| **Precondiciones:** | * Dirección identificada como Usuario**.** |
| **Poscondiciones:** | * Si no existe una tabla de datos estadísticos se crea la instancia *Tabla\_datosEst* (creación de instancia). * Se asignó a *Tabla\_datosEst*  un *D\_est* (modificación de atributos) * Si no existe una grafico estadístico se crea la instancia *Grafico\_est* (creación de instancia). * Se asoció una instancia *Grafico\_est* a *Tabla\_datosEst* basada en la correspondencia de *D\_est* (asociación formada). |

## **2.14. Generar Registro de Vehículos:**

### **2.14.1. Ingreso datos de los vehículos:**

|  |  |
| --- | --- |
| **Nombre:** | Ingreso\_Datos\_Vehiculos (matricula: cadena, marca: cadena,  fecha\_ingreso: date) |
| **Responsabilidades:** | Ingresar los datos de los vehículos usado por la empresa en sus clases o sus exámenes prácticos. |
| **Tipo:** | Sistema |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ13, REQ14, REQ16, REQ18, REQ21. * Caso de Uso: Generar Registro de Vehículos. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | * El Sistema no realiza el registro debido a inconsistencias en el ingreso de datos. Se reporta el error, se sugiere el reingreso de los datos. |
| **Salida:** | * Procesa\_Datos\_Vehiculo * Almacena\_Datos\_Vehiculo |
| **Precondiciones:** | * Funcionario de Gestión de Flota identificado como Usuario**.** |
| **Poscondiciones:** | * Si no existe una tabla vehículos se crea la instancia *Tabla\_Vehículos* (creación de instancia). * Si no existe un vehículo se crea la instancia *Vehículo* basadoen la correspondencia de *matrícula, marca, fecha\_ingreso* (creación de instancia). * Se asoció una instancia *Vehículo* a *Tabla\_Vehículos* basada en la correspondencia de *matrícula, marca, fecha\_ingreso* (asociación formada). |

## **2.15. Gestión de Flota:**

### **2.15.1. Solicitud de estado de vehículo:**

|  |  |
| --- | --- |
| **Nombre:** | Solicita\_Estado\_Vehiculo (Vehiculo: Vehiculo, estado: cadena) |
| **Responsabilidades:** | Solicitar el estado actual del vehículo determinando si está o no en buenas condiciones. |
| **Tipo:** | Sistema |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ13, REQ14, REQ15, REQ16, REQ17. * Caso de Uso: Gestión de Flota. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | * El Sistema muestra a un vehículo en mal estado. Se reporta el error, se sugiere verificar si el daño es reparable, de no ser así se procede a su inmediato reemplazo. * El Sistema muestra que el Vehículo tiene un daño reversible. De ser reparable el daño se procede a llevar el auto a reparaciones, mientras eso dure un vehículo de repuesto deberá de tomar su lugar en el registro. |
| **Salida:** | * Muestra\_Estado\_Vehiculo |
| **Precondiciones:** | * Funcionario de Gestión de Flota identificado como Usuario**.** |
| **Poscondiciones:** | * Se asignó a *Vehiculo.estado* el valor *estado* (modificación de atributos) |

## **2.16. Añadir Usuario:**

### **2.16.1. Añadir nombre de usuario y contraseña:**

|  |  |
| --- | --- |
| **Nombre:** | Añadir\_usuario (nombre\_func: cadena, nombre\_usuario: cadena, contraseña: cadena, id\_usuario: numero) |
| **Responsabilidades:** | Creación de un usuario que pueda ingresar al sistema. |
| **Tipo:** | Sistema |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ62. * Caso de Uso: Añadir Usuario. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | * El nombre de usuario entregado está siendo ocupado por otro usuario en este momento Se reporta el error, se procede a volver a la ventana de ingreso de nuevo usuario. |
| **Salida:** | * Procesar\_usuario |
| **Precondiciones:** | * Dirección identificado como Usuario**.** |
| **Poscondiciones:** | * Si no existe un Usuario creado se crea la instancia *Usuario* con *nombre\_func,* *nombre\_usuario* y *contraseña* como parámetros (creación de instancia). * Se asignó a *Usuario.id\_usuario* el valor *id\_usuario* (modificación de atributos). |

### **2.16.2. Ingreso nuevo usuario en el sistema:**

|  |  |
| --- | --- |
| **Nombre:** | Ingresar\_usuario (nombre\_usuario: cadena, id\_usuario: numero, permisos: bool) |
| **Responsabilidades:** | Asegurar que el usuario está en el sistema y agregar los permisos adecuados. |
| **Tipo:** | Sistema |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ63. * Caso de Uso: Añadir Usuario. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | * El usuario creado no queda registrado en la base de datos Se reporta el error, se procede a verificar los pasos anteriores verificando cuál fue el error, procediendo a repetir los pasos anteriores. |
| **Salida:** | * Agregar\_Permisos\_Usuario. * Asegurar\_Usuario. |
| **Precondiciones:** | * Dirección identificado como Usuario**.** |
| **Poscondiciones:** | * Se asignó a *Usuario.permisos* el valor *permisos* (modificación de atributos). * Si no existe una lista de usuario se crea *Tipo\_Usuario* (creación de instancia). * Se asocia *Usuario* con *Tipo\_Usuario* mediante *id\_usuario, nombre\_usuario* (asociación formada). |

## **2.17. Cambiar Contraseña de Usuario:**

### **2.17.1. Cambiar Contraseña de Usuario:**

|  |  |
| --- | --- |
| **Nombre:** | Cambiar\_contraseña (nombre\_usuario: cadena, vieja\_contraseña: cadena, nueva\_contraseña: cadena) |
| **Responsabilidades:** | Asegurar que al usuario tenga una nueva contraseña diferente a la anterior. |
| **Tipo:** | Sistema |
| **Referencias Cruzadas:** | * Funciones del sistema: REQ64. * Caso de Uso: Cambiar Contraseña de Usuario. |
| **Notas:** | Utilizar un acceso rápido a la base de datos y a la interfaz gráfica. |
| **Excepciones:** | * La contraseña está escrita de manera inconsistente, el sistema no puede procesar el texto escrito. Se reporta el error, se procede a volver a la ventana de ingreso de nueva contraseña. |
| **Salida:** | * Aceptar\_Contraseña. |
| **Precondiciones:** | * Dirección identificado como Usuario**.** |
| **Poscondiciones:** | * Se asignó a *Usuario.nombre\_usuario* el valor *nombre\_usuario* (modificación de atributos). * Se asignó a *Usuario.contraseña* el valor *vieja\_contraseña* (modificación de atributos). * Se asignó a *Usuario.contraseña* el valor *nueva\_contraseña* (modificación de atributos). |