

**Asignatura:** Programación Orientada a Objetos

**NRC:** 2858

**Integrantes:**

Cortez Tamayo Jonathan Andrés

Coyago Arce Eunice Pamela

Cují Revelo Milton Roberto

Garcés Mosquera Christian Marce

Guerra López Luciana Noemi

**Tema:** Reconocimiento de placas

**Contenido**

[**Objetivo general** 3](#_Toc61972310)

[**Objetivos específicos** 3](#_Toc61972311)

[**Teoría** 3](#_Toc61972312)

[**Glosario** 3](#_Toc61972313)

[**Definición** 3](#_Toc61972314)

[**Problema** 4](#_Toc61972315)

[**Solución** 4](#_Toc61972316)

[**Alcance** 4](#_Toc61972317)

[**Diagrama de casos de uso** 4](#_Toc61972318)

[**Descripción de casos de uso** 5](#_Toc61972319)

[**Diagrama UML** 9](#_Toc61972320)

[**Descripción diagrama UML** 9](#_Toc61972321)

# **Objetivo general**

* Desarrollar un sistema de visión artificial, utilizando programación orientada a objetos, para identificar el texto de una imagen.

# **Objetivos específicos**

* Definir el proceso de una imagen, conociendo sus parámetros y dimensiones, para reconocer y establecer sus medidas y contenido.
* Distinguir las imágenes ingresadas, con la carga o subida de las imágenes, para conocer el tipo de tratamiento de la imagen
* Descomponer la imagen ingresada, con el reconocimiento el uso de reconocimiento de texto para conocer el texto que la imagen ingresada o a ser tratada.

# **Teoría**

## **Glosario**

* Programación Orientada a Objetos: Manipula los datos de entrada y salida mediante objetos específicos.
* Librería: Implementación y codificación de un lenguaje de programación.
* Clase: conjunto de variables.
* Variable: Valores u operaciones a tomar o procesar.
* Función Programable: Algoritmo principal que permite resolver una tarea específica.
* Algoritmo: Conjunto de instrucciones.
* Objeto: Elementos de información caracterizadas por datos reales.
* Método: acciones que el objeto puede realizar por sí mismo.
* Atributo: Variables relacionadas al estado de un objeto.
* Evento o Comportamiento: Acciones Realizables.
* Mensajes: Transmisión de datos.
* Instancia: Llamada de un objeto.
* String: Tipo de dato el cual interpreta una cadena de texto.

## **Definición**

El sistema establece un tratamiento de imagen el cual podrá ser tratada y dimensionada, para poder entender el texto que esta compone tomando de caso de ejemplo una placa de auto, la cual podremos determinar que placa es y este texto poderlo tratar y el lenguaje de programación lo pueda entender y pasara a un string.

## **Problema**

Se conoce una imagen la cual no puede ser interpretada por la maquina y su texto no es perceptible por el lenguaje de programación y este poder utilizar o ser tratado como una cadena de texto.

## **Solución**

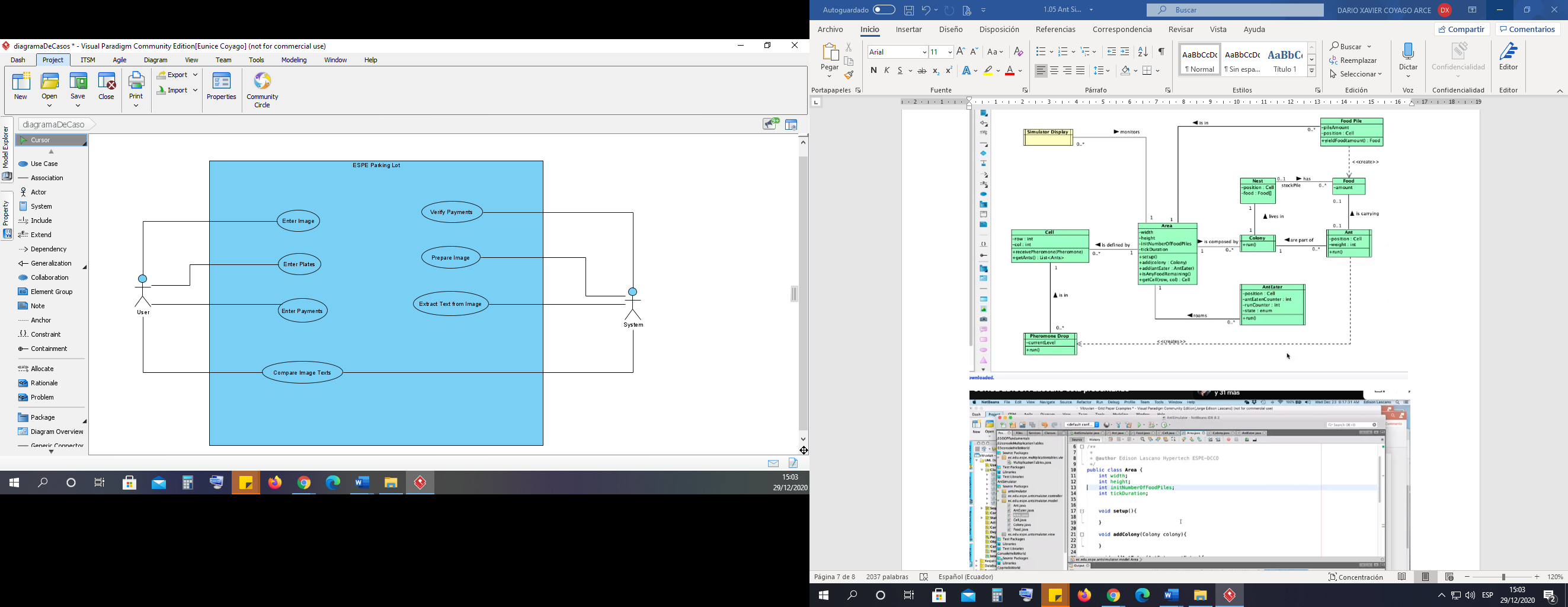
Es establecer un sistema en el cual contemple el uso de programación orientada a objetos, para lograr determinar o leer el texto que una imagen posee en caso de una imagen de las placas de un auto esto determinara el texto para poder ser tratado e interpretado por la máquina.

## **Alcance**

El sistema permitirá el ingreso de una imagen para que este pueda tratar la imagen y reconocer el texto que ella posee delimitando que la imagen será ingresada manualmente por el usuario.

# **Diagrama de casos de uso**





# **Descripción de casos de uso**

|  |  |
| --- | --- |
| **Nombre:** | Ingresar imagen |
| **Autor**: | Cortez Tamayo Jonathan Andrés  Coyago Arce Eunice Pamela  Cují Revelo Milton Roberto  Garcés Mosquera Christian Marce  Guerra López Luciana Noemi |
| **Fecha**: | 19/01/2021 |
| **Descripción**:  Permite que el usuario ingrese una imagen al sistema | |
| **Actores**:  Usuario | |
| **Precondiciones**:  El actor debe ingresar al sistema | |
| **Flujo Normal**:   1. El actor ingreso correctamente al sistema 2. El actor carga la imagen a ser tratada | |
| **Flujo Alternativo**:  El sistema comprueba si la imagen corresponde al formato establecido caso contrario, el sistema emitirá un mensaje de error “IMAGEN NO CORRESPONDE”. | |
| **Postcondiciones**:  El actor a cargado la imagen correctamente | |

|  |  |
| --- | --- |
| **Nombre:** | Ingresar placas |
| **Autor**: | Cortez Tamayo Jonathan Andrés  Coyago Arce Eunice Pamela  Cují Revelo Milton Roberto  Garcés Mosquera Christian Marce  Guerra López Luciana Noemi |
| **Fecha**: | 19/01/2021 |
| **Descripción**:  Permite que el usuario ingresar datos de las placas | |
| **Actores**:  Usuario | |
| **Precondiciones**:  El actor debe ingresar al sistema | |
| **Flujo Normal**:   1. El actor ingreso correctamente al sistema 2. El actor carga los datos de las placas | |
| **Flujo Alternativo**:  El sistema comprueba si los datos están correctos caso contrario emitirá un mensaje de error “DATOS INGRESADOS SON INCORRECTOS” | |
| **Postcondiciones**:  El actor a cargado correctamente los datos de las placas | |

|  |  |
| --- | --- |
| **Nombre:** | Ingresar pagos |
| **Autor**: | Cortez Tamayo Jonathan Andrés  Coyago Arce Eunice Pamela  Cují Revelo Milton Roberto  Garcés Mosquera Christian Marce  Guerra López Luciana Noemi |
| **Fecha**: | 19/01/2021 |
| **Descripción**:  Permite que el usuario ingresar los pagos correspondientes a las placas | |
| **Actores**:  Usuario | |
| **Precondiciones**:  El actor debe ingresar al sistema | |
| **Flujo Normal**:   1. El actor ingreso correctamente al sistema 2. El actor carga los datos de pagos por placa | |
| **Flujo Alternativo**:  El sistema comprueba si los datos están correctos caso contrario emitirá un mensaje de error “DATOS INGRESADOS SON INCORRECTOS” | |
| **Postcondiciones**:  El actor a cargado correctamente los pagos de las placas | |

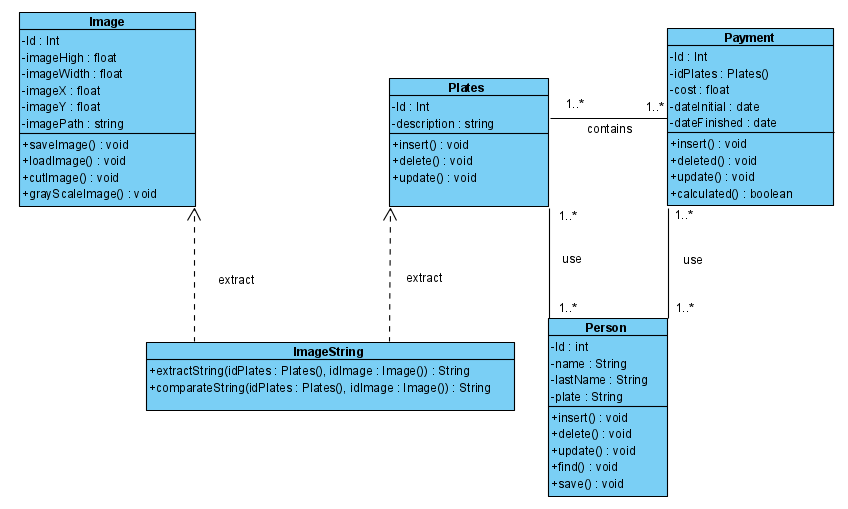
|  |  |
| --- | --- |
| **Nombre:** | Verificar pagos |
| **Autor**: | Cortez Tamayo Jonathan Andrés  Coyago Arce Eunice Pamela  Cují Revelo Milton Roberto  Garcés Mosquera Christian Marce  Guerra López Luciana Noemi |
| **Fecha**: | 19/01/2021 |
| **Descripción**:  Permite que el sistema verifique si los pagos están realizados | |
| **Actores**:  Sistema | |
| **Precondiciones**:  Los datos de pagos deben estar ingresados | |
| **Flujo Normal**:   1. El sistema buscara la existencia de las placas 2. El sistema verificara si esta con los pagos adecuados | |
| **Flujo Alternativo**:  El sistema verificara si los pagos y las placas están correctas caso contrario emitirá un mensaje de error “PLACAS INXISTENTE O PAGOS NO ESTAN AL DÍA” | |
| **Postcondiciones**:  El sistema da paso correcto de los pagos en las placas | |

|  |  |
| --- | --- |
| **Nombre:** | Preparar imagen |
| **Autor**: | Cortez Tamayo Jonathan Andrés  Coyago Arce Eunice Pamela  Cují Revelo Milton Roberto  Garcés Mosquera Christian Marce  Guerra López Luciana Noemi |
| **Fecha**: | 19/01/2021 |
| **Descripción**:  Permite que el sistema trate la imagen estableciendo las dimensiones deseadas y colores necesarios | |
| **Actores**:  Sistema | |
| **Precondiciones**:  La imagen debe estar previamente cargada | |
| **Flujo Normal**:   1. El sistema recorta la imagen a las dimensiones deseadas dependiendo de la imagen cargada 2. El sistema cambiara el tono de colores a escalas grises | |
| **Flujo Alternativo**:  El sistema analiza el tratamiento de imagen y si las dimensiones sobresalen del estándar establecido emitirá un error de “DIMENSIONES NO ADECUADAS”. | |
| **Postcondiciones**:  El sistema creara la imagen en escalas grises | |

|  |  |
| --- | --- |
| **Nombre:** | Extraer texto de imagen |
| **Autor**: | Cortez Tamayo Jonathan Andrés  Coyago Arce Eunice Pamela  Cují Revelo Milton Roberto  Garcés Mosquera Christian Marce  Guerra López Luciana Noemi |
| **Fecha**: | 19/01/2021 |
| **Descripción**:  El sistema extraerá el texto de la imagen en una cadena de texto | |
| **Actores**:  Usuario | |
| **Precondiciones**:  El sistema contiene la imagen en escalas grises para la extracción del texto | |
| **Flujo Normal**:   1. El sistema extraerá correctamente la cadena de texto que posee la imagen | |
| **Flujo Alternativo**:  El sistema comprueba si la imagen esta tratada adecuadamente y en las dimensiones casos contrario, el texto no será el adecuado. | |
| **Postcondiciones**:  La cadena de texto se imprimirá y podrá ser visible tanto para el sistema como para el usuario. | |

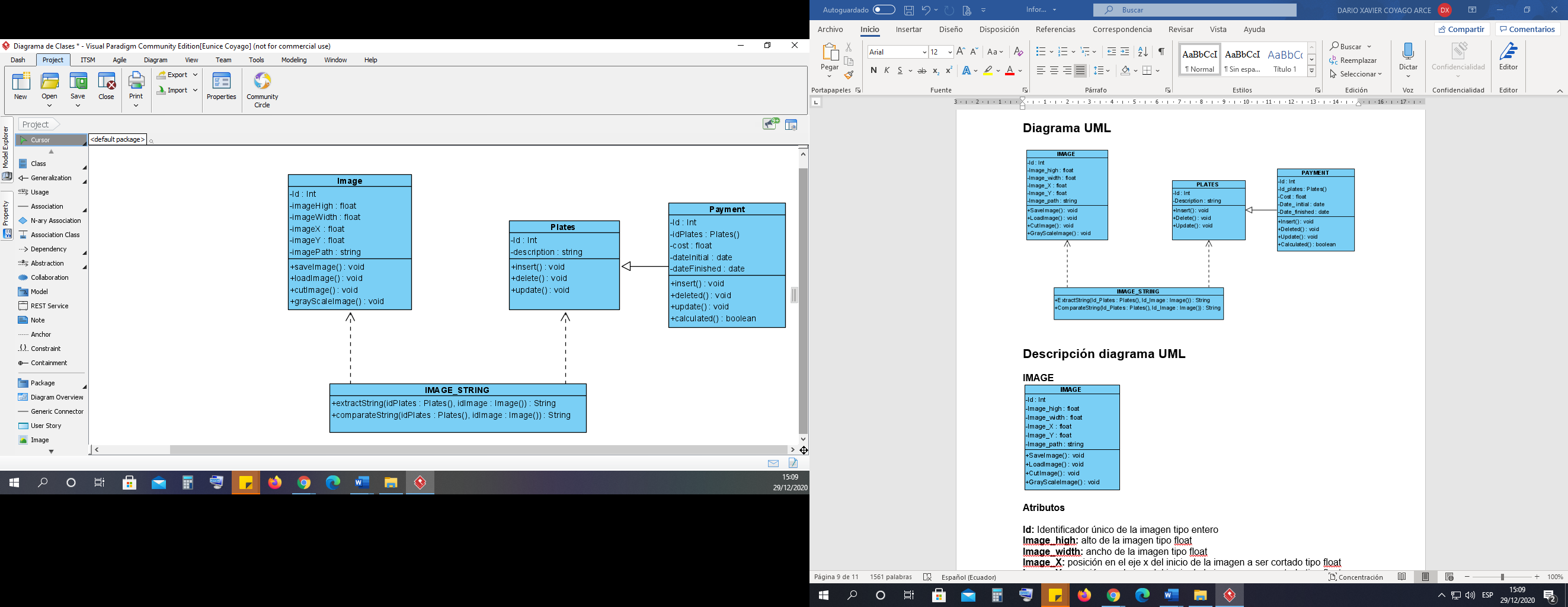
|  |  |
| --- | --- |
| **Nombre:** | Comprar texto de imagen |
| **Autor**: | Cortez Tamayo Jonathan Andrés  Coyago Arce Eunice Pamela  Cují Revelo Milton Roberto  Garcés Mosquera Christian Marce  Guerra López Luciana Noemi |
| **Fecha**: | 19/01/2021 |
| **Descripción**:  Permite que el actor pueda comprar el texto de la imagen tratada los datos de autos ingresados | |
| **Actores**:  Usuario  Sistema | |
| **Precondiciones**:  El sistema debe haber preparado la imagen | |
| **Flujo Normal**:   1. El actor podrá observar la cadena de texto de la imagen ingresada 2. El sistema comprara si el texto existe en los datos de autos | |
| **Flujo Alternativo**:  El actor observará que la cadena de texto y esta si no existe emitirá un mensaje de error “PLACA NO EXISTE” | |
| **Postcondiciones**:  El actor podrá comprar la cadena extraída con los datos de autos. | |

# **Diagrama UML**



# **Descripción diagrama UML**

**Image**



**Atributos**

**Id:** Identificador único de la imagen tipo entero.

**imageHigh:** alto de la imagen tipo float.

**imageWidth:** ancho de la imagen tipo float.

**imageX:** posición en el eje x del inicio de la imagen a ser cortado tipo float.

**imageY:** posición en el eje y del inicio de la imagen a ser cortado tipo float.

**imagePath:** ubicación de donde se encuentra la imagen tipo string.

**Métodos**

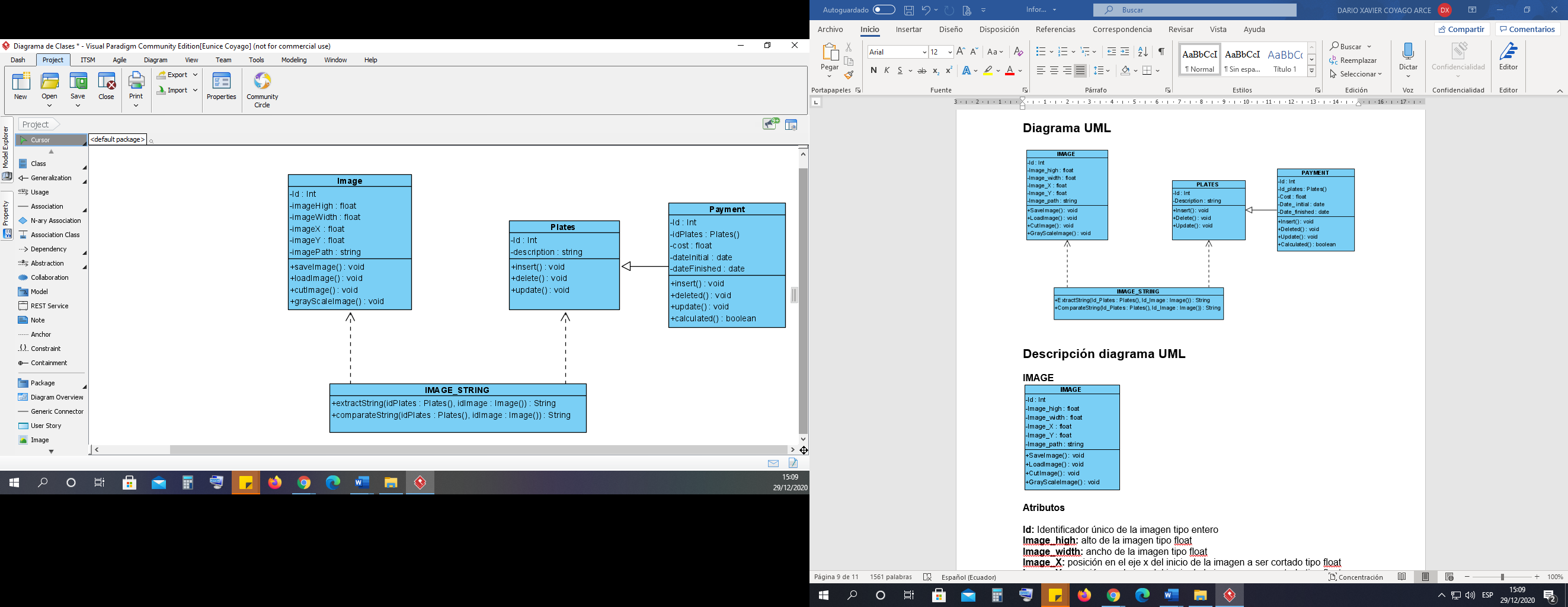
**saveImage():** Método void para guardar la imagen.

**loadImage():** Método void de carga de la imagen a ser tratada.

**cutImage():** Método void para cortar la imagen.

**grayScaleImage():** Método void para el cambio en escalas grises de la imagen.

**Payment**



**Atributos**

**Id:** identificador único del registro tipo int.

**idPlates:** identificador perteneciente a la clase plates tipo int.

**cost:** variable que representa al pago efectuado tipo float.

**dateIntial:** variable que determina la fecha de ingreso del pago tipo float.

**dateFinished:** variable que determina la fecha de finalización del pago tipo float.

**Métodos**

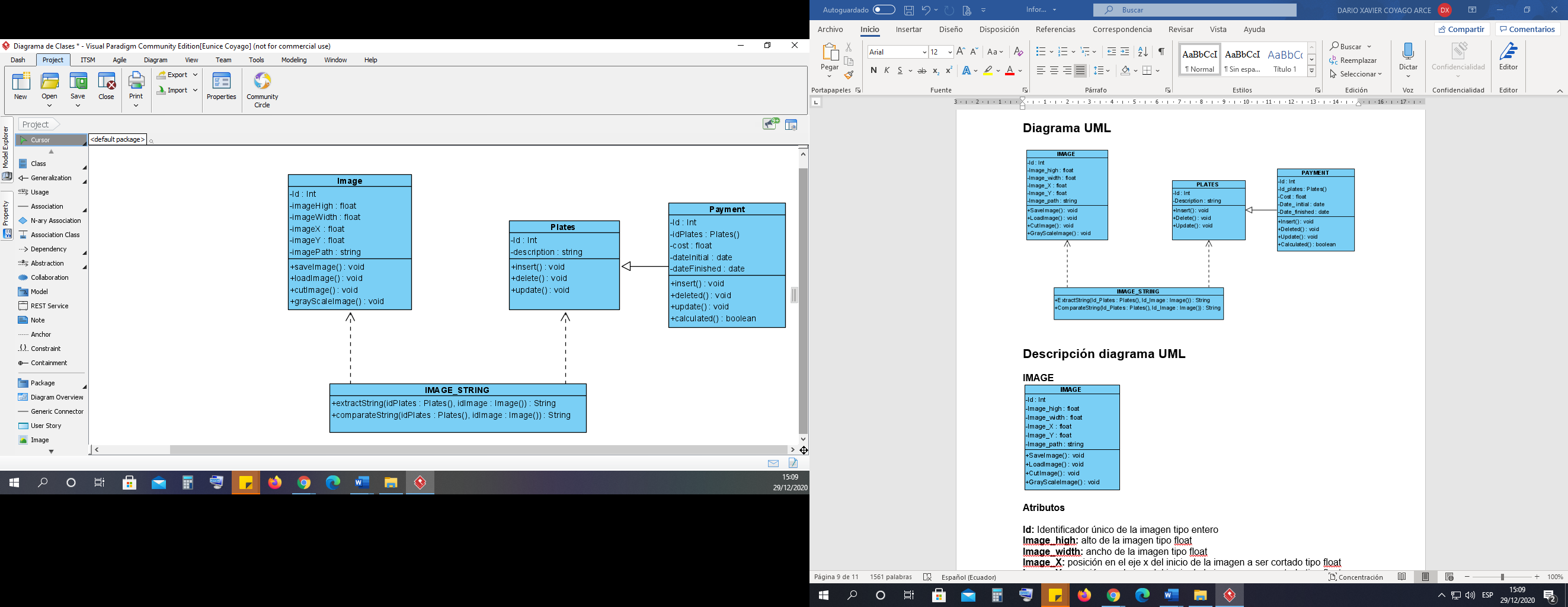
**insert():** Método void en el cual permite el ingreso de un pago.

**deleted():** Método void en el cual permite el eliminar de un pago.

**update():** Método void en el cual permite el actualizar de un pago.

**calculated():**Método que devuelve un booleano en el cual permite generar y calcular el pago efectuado y determina si se realizó o no dicho pago.

**Plates**



**Atributos**

**Id:** identificador único del registro tipo int.

**description:** variable que permite describir o almacenar la placa de tipo string.

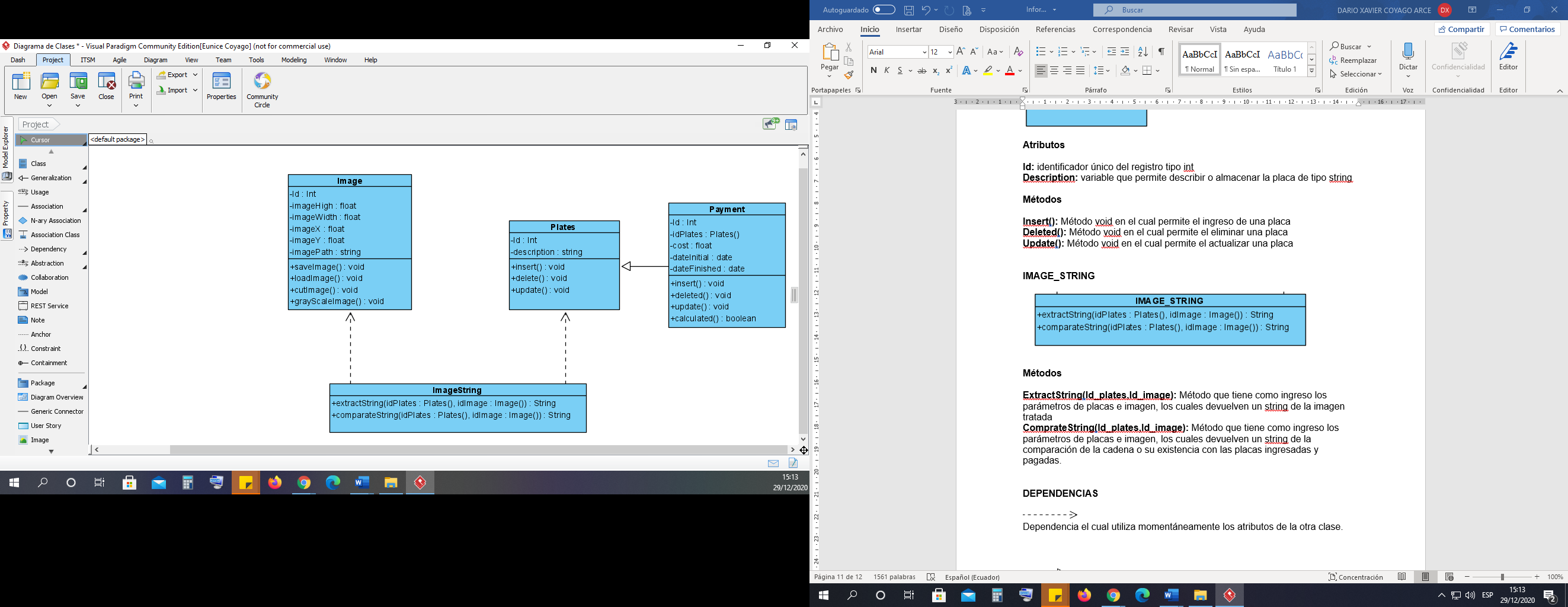
**Métodos**

**insert():** Método void en el cual permite el ingreso de una placa.

**deleted():** Método void en el cual permite el eliminar una placa.

**update():** Método void en el cual permite el actualizar una placa.

**ImageString**

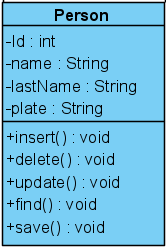


**Métodos**

**extractString(idPlates,idImage):** Método que tiene como ingreso los parámetros de placas e imagen, los cuales devuelven un string de la imagen tratada.

**ComprateString(idPlates,idImage):** Método que tiene como ingreso los parámetros de placas e imagen, los cuales devuelven un string de la comparación de la cadena o su existencia con las placas ingresadas y pagadas.

**Person**



**Atributos**

**Id:** identificador único de tipo int.

**name:** variable que permite guardar el nombre de una persona de tipo string.

**lastName:** variable que permite guardar el apellido de una persona de tipo string.

**plate:** variable que permite introducir la serie de una placa de tipo string.

**Métodos**

**insert():** Método void en el cual permite el ingreso de una placa o datos de la persona.

**deleted():** Método void en el cual permite el eliminar una placa o datos de la persona.

**update():** Método void en el cual permite el actualizar una placa o datos de la persona.

**find():** Método void en el que permite la entrada de una placa o datos de la persona.

**save():** Método void que permite guardar los datos de una placa o una persona.

**DEPENDENCIA**



Dependencia el cual utiliza momentáneamente los atributos de la otra clase.



Generalización para determinar que requiere elementos de la otra clase.