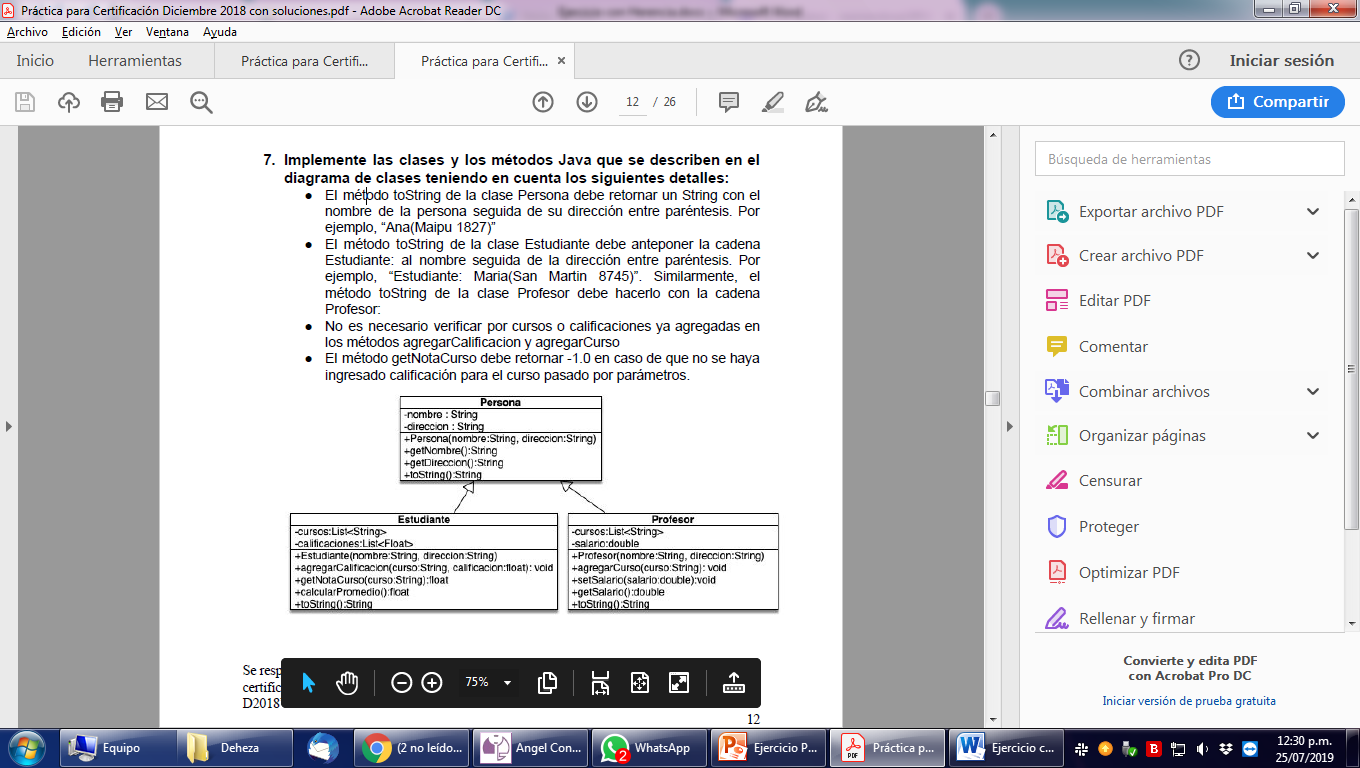
**Implemente las clases y los métodos Java que se describen en el diagrama de clases teniendo en cuenta los siguientes detalles:**

* El método toString de la clase Persona debe retornar un String con el nombre de la persona seguida de su dirección entre paréntesis. Por ejemplo, “Ana(Maipu 1827)”
* El método toString de la clase Estudiante debe anteponer la cadena Estudiante: al nombre seguida de la dirección entre paréntesis. Por ejemplo, “Estudiante: Maria(San Martin 8745)”. Similarmente, el método toString de la clase Profesor debe hacerlo con la cadena Profesor:
* No es necesario verificar por cursos o calificaciones ya agregadas en los métodos agregarCalificacion y agregarCurso
* El método getNotaCurso debe retornar -1.0 en caso de que no se haya ingresado calificación para el curso pasado por parámetros.



public class Persona{

private String nombre ;

private String direccion ;

public Persona(String nombre, String direccion){

this . nombre = nombre;

this . direccion = direccion;

}

public String getNombre(){

return nombre ;

}

public String getDireccion(){

return direccion ;

}

@Override

public String toString(){

return nombre + "(" + direccion + ")" ;

}

}

public class Estudiante extends Persona{

private List<String> cursos ;

private List<Float> calificaciones ;

public Estudiante(String nombre, String direccion){

super (nombre, direccion);

cursos = new ArrayList<>();

calificaciones = new ArrayList<>();

}

@Override

public String toString(){

return "Estudiante: " + super .toString();

}

public void agregarCalificacion(String curso, float calificacion){

cursos .add(curso);

calificaciones .add(calificacion);

}

public float getNotaCurso(String curso){

Float nota=-1f;

if ( cursos .contains(curso)){

nota= calificaciones .get( cursos .indexOf(curso));

}

return nota;

}

public float calcularPromedio(){

Float sum=0f;

Iterator<Float> it= calificaciones .iterator();

while (it.hasNext()){

Float calificacion = it.next();

sum=sum+calificacion;

}

return sum/ cursos .size();

}

}

public class Profesor extends Persona{

private List<String> cursos ;

private double salario ;

public Profesor(String nombre, String direccion){

super (nombre, direccion);

cursos = new ArrayList<>();

}

@Override

public String toString(){

return "Profesor: " + super .toString();

}

public void agregarCurso(String curso){

cursos .add(curso);

}

public void setSalario( double salario){

this . salario = salario;

}

public double getSalario(){

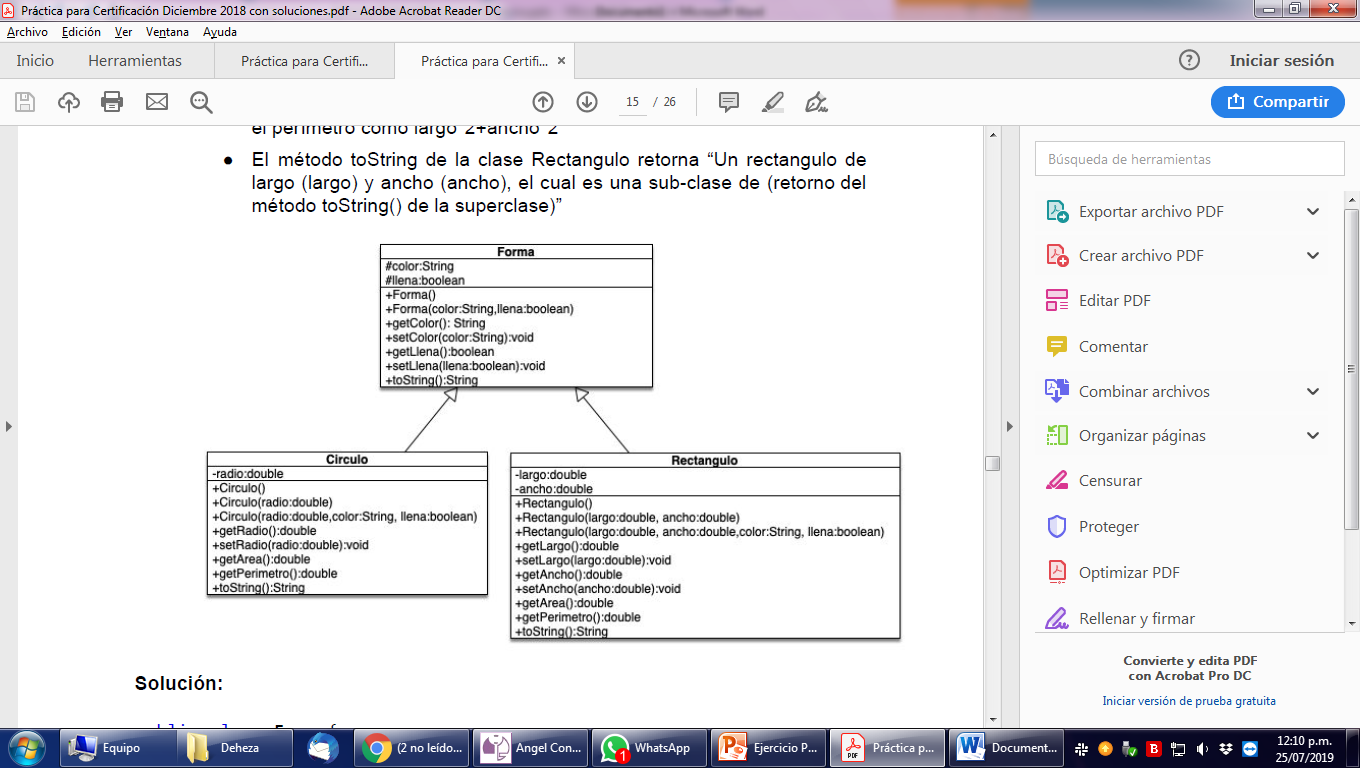
return salario ;

}

}

**Se está trabajando en una aplicación para visualizar figuras geométricas. Implemente las clases y los métodos Java que se describen en el diagrama de clases teniendo en cuenta los siguientes detalles:**

* La clase Forma tiene dos constructores: uno sin argumentos que inicializa color a “verde” y llena a true; y un constructor que inicializa los atributos con los valores dados.
* El método toString de la clase Forma retorna “Forma[color=(color)]”. Dónde (color) será el almacenado en la variable de instancia.
* La clase Circulo tiene tres constructores. El constructor sin parámetros inicializa radio en 1.
* Recuerde que el área de un círculo es igual a Pi\*radio 2 y el perímetro como 2\* Pi \*radio. El valor de Pi puede obtener como Math.PI del paquete java.lang.
* El método toString de la clase Circulo retorna “Un círculo de radio (radio), el cual es una sub-clase de (retorno del método toString() de la superclase)”
* La clase Rectangulo tiene tres constructores. El constructor sin parámetros inicializa ancho y largo en 1.
* Recuerde que el área de un rectángulo se calcula como largo\*ancho y el perímetro como largo\*2+ancho\*2
* El método toString de la clase Rectangulo retorna “Un rectangulo de largo (largo) y ancho (ancho), el cual es una sub-clase de (retorno del método toString() de la superclase)”



public class **Forma** {

protected String color ;

protected boolean llena ;

public **Forma** (){

color = "verde" ;

llena = true ;

}

public **Forma** (String color, boolean llena){

this . color = color;

this . llena = llena;

}

@Override

public String **toString** (){

return "Forma[color=" + color + "]" ;

}

public void **setColor** (String color){

this . color = color;

}

public String **getColor** (){

return color ;

}

public boolean **getLlena** (){

return llena ;

}

public void **serLlena** ( boolean llena){

this . llena = llena;

}

}

public class **Circulo** extends Forma{

private double radio ;

public **Circulo** ( double radio,String color, boolean llena){

super (color,llena);

this . radio = radio;

}

public **Circulo** ( double radio){

super ();

this . radio = radio;

}

public **Circulo** (){

super ();

this . radio = 1;

}

public double **getRadio** (){

return radio ;

}

@Override

public String **toString** (){

return "Un circulo de radio" + radio + ", el cual es una

sub-clase de " + super .toString();

}

public double **getArea** (){

return Math. *PI* \* radio \* radio ;

}

public double **getPerimetro** (){

return 2\*Math. *PI* \* radio ;

}

}

public class **Rectangulo** extends Forma{

private double largo ;

private double ancho ;

public **Rectangulo** ( double largo, double ancho, String

color, boolean llena){

super (color,llena);

this . largo = largo;

this . ancho = ancho;

}

public **Rectangulo** ( double largo, double ancho){

super ();

this . largo = largo;

this . ancho = ancho;

}

public **Rectangulo** (){

super ();

this . largo = 1;

this . ancho = 1;

}

public String **toString** (){

return "Un Rectangulo de largo " + largo + "y ancho

" + ancho + " que es una sub-clase de " + super .toString();

}

public double **getArea** (){

return largo \* ancho ;

}

public double **getPerimetro** (){

return largo \*2 + ancho \*2;

}

public void **setLargo** ( double largo){

this . largo = largo;

}

public double **getLargo** (){

return largo ;

}

public void **setAncho** ( double ancho){

this . ancho = ancho;

}

public double **getAncho** (){

return ancho ;

}

}