PagoMercado

Ejercicio segundo tramo

# Enunciado

La plataforma PagoMercado quiere fortalecer su posición en nuestro país a través de su nuevo servicio de pago para tiendas físicas (con venta en punto y online). Para hacer esto, necesita trabajar en su sistema de adhesión de comercios a través de empresas reclutadoras, que debe ser capaz de:

1) Contar con las entidades Reclutadora, Adhesión y Comercio (que bien puede ser ComercioOnline o ComercioPresencial).

1. ToString se debe sobrescribir sólo en Empresa y en Adhesión.
2. Encapsular los atributos necesarios aplicando sólo los getters y setters (en los casos que corresponda)

2) Poder adherir comercios presenciales (los comercios online no se pueden adherir, ya que tienen su propia plataforma de pagos). (Alta Adhesión)

1. En caso que el comercio ya exista en la lista de adhesiones, se debe arrojar una AdhesionExistenteException.
2. Solo se puede agregar un comercio a la lista (\_adhesiones) una vez.
3. En caso que se quiera forzar la adhesión de un comercio online, se deberá arrojar una AdhesionNoPermitidaException.
4. Se debe autogenerar el número de orden.

3) Poder listar las adhesiones con un {FORMATO}. (Listar Adhesiones)

1. En caso que no haya adhesiones mostrar “No hay comercios adheridos”.

4) Poder consultar POR CUIT si un comercio está adherido o no. (Consultar Adhesión)

1. En caso de que no exista, mostrar “No existe el comercio”.

5) Poder eliminar una adhesión POR CUIT de la empresa.

Siempre debe validarse que la empresa exista previamente, sino, debe lanzarse la excepción EmpresaNoExistenteException.

1. En caso que el comercio no esté adherido se debe arrojar una AdhesionNoExistenteException.

Formatos para ToString (display)

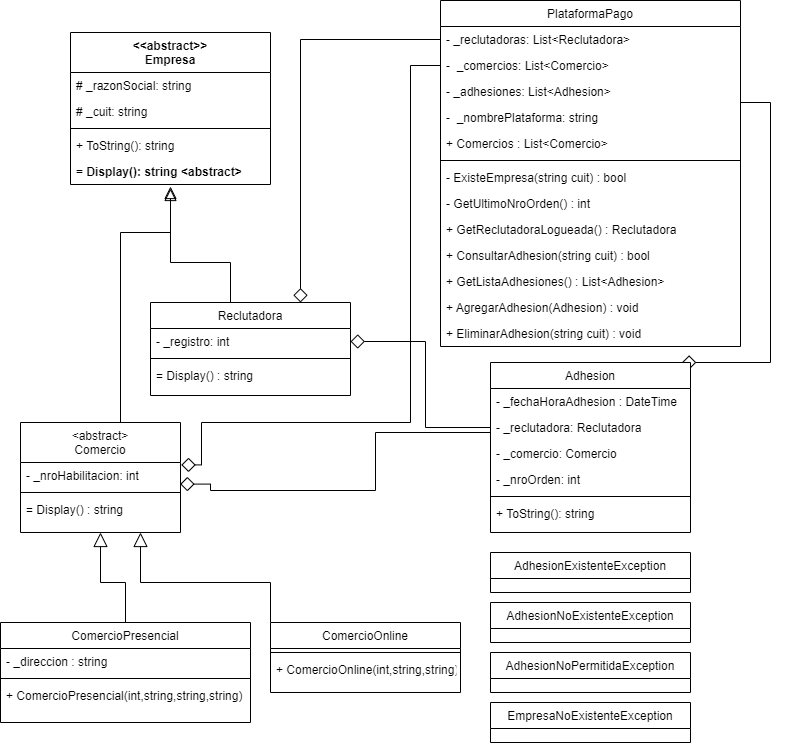
1. Reclutadora “REGISTRO - RAZONSOCIAL”
2. Comercio “RAZONSOCIAL (CUIT) - NROHABILITACION”
3. Adhesión "NROORDEN {COMERCIO FORMATEADO} está adherido a PagoMercado desde FECHAHORA"

# Consideraciones generales

Se deben respetar las visibilidades del diagrama

* Público - Privado # Protegido = Internal
* El proyecto principal deber ser de consola
* Se debe desarrollar utilizando almenos dos capas
* Deben realizarse validación de las entradas de usuario
* Manejo apropiado de excepciones
* La consola se debe mantener activa hasta que el usuario desee finalizar el programa ingresando X
* Se debe resolver en menos de 2 horas
* Debe compilar sin errores
* No debe terminar por excepción no controlada
* El prefijo del nombre de los proyectos debe ser Parcial1.APELLIDO.NombreProyecto
* Se entrega la solución entera (comprimida) en la dirección de drive previamente asignada
* Es válido agregar los constructores que sean necesarios

# Modelo



# Template de la clase Program (para completar)

public class Program

{

private static PlataformaPago \_plataforma;

static Program()

{

\_plataforma = new PlataformaPago(“PagoMercado”);

}

static void Main(string[] args)

{

Reclutadora reclutadoraActiva = \_plataforma.GetReclutadoraLogueada();

DesplegarOpcionesMenu();

string opcionMenu = ""; // pedir el valor

switch (opcionMenu)

{

case "1":

ListarAdhesiones();

break;

case "2":

ConsultarAdhesion();

break;

case "3":

AltaAdhesion(reclutadoraActiva);

break;

case "4":

EliminarAdhesion();

break;

case "X":

// SALIR

break;

default:

break;

}

}

static void DesplegarOpcionesMenu()

{

Console.WriteLine("1) Listar Adhesiones");

Console.WriteLine("2) Consultar Adhesion");

Console.WriteLine("3) Alta Adhesion");

Console.WriteLine("4) Eliminar Adhesion");

Console.WriteLine("X: Terminar");

}

static void ListarAdhesiones()

{

// mostrar adhesiones

}

static void ConsultarAdhesion()

{

// ingreso cuit

// consultar adhesion

}

static void AltaAdhesion(Reclutadora p)

{

// Listar comercios

// usuario selecciona el cuit que desea agregar

// Agregar adhesión

}

static void EliminarAdhesion()

{

// Listar comercios

// usuario selecciona el cuit que desea eliminar

// Eliminar adhesión

}

}

## Datos de Inicialización

// iniciar PlataformaPago con los siguientes datos

public PlataformaPago()

{

// incializar listas

\_comercios.Add(new ComercioPresencial(5001, "30-40123456-8", "Juarez SRL", "Moldes 123"));

\_comercios.Add(new ComercioPresencial(5002, "30-40854451-1", "Moda SA", "Vidal 456"));

\_comercios.Add(new ComercioPresencial(5003, "30-40853352-9", "Ronda SA", "Vidt 887"));

\_comercios.Add(new ComercioPresencial(5004, "30-20663323-7", "Andina Inc.", "Ugarte 2020"));

\_comercios.Add(new ComercioOnline(4040, "30-18512524-3", "TodoVenta.com"));

\_comercios.Add(new ComercioOnline(4041, "30-35112685-4", "AlqInmuebles.com.ar"));

\_comercios.Add(new ComercioOnline(4042, "30-28512571-2", "DeCarpinteros.com"));

\_reclutadoras.Add(new Reclutadora(101, "Agencia Ramos SRL", "30-55123456-1"));

}