Clases:

```
using System;
using System.Collections;
using System.Collections.Generic;

namespace Tp_Integrador
{
    8 referencias
    public class EmpresaConstructora
    {
        private ArrayList lista_Obras_Finalizadas;
        private ArrayList lista_Obras;
        private ArrayList lista_Grupos;
        1 referencia
        public EmpresaConstructora()
        {
            lista_Obras_Finalizadas = new ArrayList();
            lista_Obras = new ArrayList();
            lista_Grupos = new ArrayList();
        }

        1 referencia
        public ArrayList Lista_Obras_Finalizadas
        {get {return lista_Obras_Finalizadas;}}

        3 referencias
        public ArrayList Lista_Obras
        {get {return lista_Obras;} }

        5 referencias
        public ArrayList Lista_Grupos
        {get {return lista_Grupos;}}

        1 referencia
        public void Agregar_Grupo(Grupo g)
        {
            lista_Grupos.Add(g);
        }

        0 referencias
        public void Eliminar_Grupo(Grupo g1)
        {
            lista_Grupos.Remove(g1);
        }

        1 referencia
        public void Agregar_Obra(Obra b)
        {
            lista_Obras.Add(b);
        }

        0 referencias
        public void Eliminar_Obra(Obra c)
        {
            lista_Obras.Remove(c);
            }
        1 referencia
        public void Agregar_Obra_Finalizada(Obra d)
        {
            lista_Obras_Finalizadas.Add(d);
            }
        0 referencias
        public void Eliminar_Obra_Finalizada(Obra e)
        {
            lista_Obras_Finalizadas.Remove(e);
            }

    }
```

```
public class Obra
{
    private string nombre_Propietario;
    private int dNI_Propietario;
    private int codigo_Obra;
    private string tipo_Obra;
    private int dias_Restantes;
    private int estado_Avance;
    private JefeObra jefe_de_obra;

    1 referencia
    public Obra(string nombrePropietario, int dniPropietario, int codigoObra, string tipoObra, int diasRestantes, int estadoAvance, JefeObra jefe_a)
    {
        this.nombre_Propietario = nombrePropietario;
        this.dNI_Propietario = dniPropietario;
        this.codigo_Obra = codigoObra;
        this.tipo_Obra = tipoObra;
        this.dias_Restantes = diasRestantes;
        this.estado_Avance = estadoAvance;
        this.jefe_de_obra = jefe_a;
    }
    0 referencias
    public string Nombre_Propietario
    {
        set { nombre_Propietario = value;}
        get { return nombre_Propietario;}
    }
    0 referencias
    public int DNI_Propietario
    {
        set { dNI_Propietario = value;}
        get { return dNI_Propietario;}
    }
    3 referencias
    public int Codigo_Obra
    {
        set { codigo_Obra = value;}
        get { return codigo_Obra;}
    }
    0 referencias
    public string Tipo_Obra
    {
        set { tipo_Obra = value;}
        get { return tipo_Obra;}
    }
    0 referencias
    public int Dias_Restantes
    {
        set { dias_Restantes = value;}
        get { return dias_Restantes;}
    }
    2 referencias
    public int Estado_Avance
    {
        set { estado_Avance = value;}
        get { return estado_Avance;}
    }
    1 referencia
    public JefeObra Jefe_de_obra
    {
        set { jefe_de_obra = value;}
        get { return jefe_de_obra;}
    }


}
```

```
public class Grupo
{
    private int numero_Grupo;
    private ArrayList lista_Obreros;
    private int codigo_Obra_A_Cargo;
    1 referencia
    public Grupo(int num_gru)
    {
        this.codigo_Obra_A_Cargo = 0;
        lista_Obreros = new ArrayList();
        numero_Grupo = num_gru;
    }

    1 referencia
    public void Añadir_Obrero(Obrero b1)
    { lista_Obreros.Add(b1); }


    2 referencias
    public void Eliminar_Obrero(Obrero a1)
    { lista_Obreros.Remove(a1); }

    3 referencias
    public int Numero_Grupo
    {
        set { numero_Grupo = value;}
        get { return numero_Grupo;}
    }
    1 referencia
    public int Codigo_Obra_A_Cargo
    {
        set { codigo_Obra_A_Cargo = value;}
        get { return codigo_Obra_A_Cargo;}
    }
    3 referencias
    public ArrayList Lista_Obreros
    {
        get {return lista_Obreros;}
    }
}
```

```
public class Obrero
{
    protected string nombreApellido;
    protected int dNI;
    protected int legajo;
    protected int sueldo;
    protected string cargo;
    2 referencias
    public Obrero(string nombreApellido, int dni, int legajo, int sueldo, string cargo)
    {
        this.nombreApellido = nombreApellido;
        this.dNI = dni;
        this.legajo = legajo;
        this.sueldo = sueldo;

        this.cargo = cargo;
    }
    2 referencias
    public string NombreApellido
    {
        set { nombreApellido = value; }
        get { return nombreApellido; }
    }

    1 referencia
    public int DNI
    {
        set { dNI = value; }
        get { return dNI; }
    }

    3 referencias
    public int Legajo
    {
        set { legajo = value; }
        get { return legajo; }
    }

    1 referencia
    public int Sueldo
    {
        set { sueldo = value; }
        get { return sueldo; }
    }

    1 referencia
    public string Cargo
    {
        set { cargo = value; }
        get { return cargo; }
    }

}
```

```
9 referencias
public class JefeObra : Obrero
{
    private int codGrupoACargo;
    private int bonif_Jefe;

    1 referencia
    public JefeObra(int dni, string nombre_Apellido, int legajo, int sueldo, string cargo, int grupoACargo, int bonifExtra): base(nombre_Apellido, dni, legajo, sueldo, cargo)
    {
        bonif_Jefe = bonifExtra;
        codGrupoACargo = grupoACargo;
    }

    0 referencias
    public int CodGrupoACargo
    {
        set { codGrupoACargo = value; }
        get { return codGrupoACargo; }
    }
    0 referencias
    public int Bonif_Jefe
    {
        set { bonif_Jefe = value; }
        get { return bonif_Jefe; }
    }
}
1 referencia
public class ErrorDeCargaDeObrero : Exception
{
    public string razon;
    0 referencias
    public ErrorDeCargaDeObrero(string razon)
    {
        this.razon = razon;
    }
}
1 referencia
public class Error_Submenu : Exception
{
    public string razon;
    0 referencias
    public Error_Submenu(string razon)
    {
        this.razon = razon;
    }
}
13 referencias
public class ErrorDeCarga : Exception
{
    public string motivo;
    6 referencias
    public ErrorDeCarga(string motivo)
    {
        this.motivo = motivo;
    }
}
```

Main:

```
    0 referencias
    class Program
    {
0 referencias
public static void Main(string[] args)
    {
        EmpresaConstructora empresa1 = new EmpresaConstructora();
             Grupo g;
            for (int j=1; j<9;j++)
            {
            g = new Grupo(j);
            empresa1.Agregar_Grupo(g);
            }

        bool exit = false;
        string sigue;
        while (!exit)
        {
            Console.WriteLine("Elija una opción: ");
            Console.WriteLine("1- Contratar un obrero nuevo");
            Console.WriteLine("2- Eliminar obrero");
            Console.WriteLine("3 - Submenú de impresión");
            Console.WriteLine("4- Crear obra y asignarle un jefe");
            Console.WriteLine("5 - Modificar el estado de avance de una obra. ");
            Console.WriteLine("6- Dar de baja un jefe ");
            Console.WriteLine("7- Asignar obra a grupo");
            Console.WriteLine("8- Salir ");
            sigue = Console.ReadLine();
            switch (sigue)
            {
                case "1":
                        AñadirObrero(empresa1);
                        Console.WriteLine("Presiona cualquier tecla para continuar...");
                             break;

                case "2":
                        EliminarObrero(empresa1);
                        Console.WriteLine("Presiona cualquier tecla para continuar...");
                             break;
```

```
        case "7":
                int num_g;
                int nue_codigo_o;
                Console.WriteLine("Ingrese numero de grupo y codigo de obra a asignar: ");
                try
                {
                    num_g = int.Parse(Console.ReadLine());
                    nue_codigo_o = int.Parse(Console.ReadLine());
                    if ((nue_codigo_o < 0) || (nue_codigo_o > 8))
                       throw new ErrorDeCarga("El grupo no existe");
                    else
                        AsignarObra(empresa1,num_g, nue_codigo_o);
                }
                catch (ErrorDeCarga a5)
                {
                    Console.WriteLine(a5.motivo);
                }
                Console.WriteLine("Presiona cualquier tecla para continuar...");
                Console.ReadKey(true);
                break;

        case "8":
                exit = true;
                break;

        default:
                Console.WriteLine("Ingrese un valor correcto para el menú.");
                break;

        }
    }
```

```
    case "3":
            int submenu = 0;
            Console.WriteLine("1-Ver listado de obreros");
            Console.WriteLine("2-Ver obras con más del 50% de avance");
            Console.WriteLine("3-Ver obras finalizadas");
            Console.WriteLine("4-Ver jefes de obras");
            Console.WriteLine("0-Salir");
            try
            {
                submenu = int.Parse(Console.ReadLine());
                if ((submenu < 0) || (submenu > 4))
                    throw new ErrorDeCarga("Ha fallado el ingreso de un dato");
                switch (submenu)
                {
                    case 1:
                        ArrayList Lista_Submenu_1 = new ArrayList();
                        Lista_Submenu_1 = empresa1.Lista_Grupos;
                        foreach (Grupo g9 in Lista_Submenu_1)
                        {
                            foreach (Obrero o0 in g9.Lista_Obreros)
                            {
                                Console.WriteLine("{0} {1} {2} {3} {4}", o0.NombreApellido, o0.DNI, o0.Legajo, o0.Sueldo, o0.Cargo);
                            }
                        }
                        break;
                    case 2:
                        ArrayList Lista_Submenu_2 = new ArrayList();
                        Lista_Submenu_2 = empresa1.Lista_Obras;
                        foreach (Obra o21 in Lista_Submenu_2)
                        {
                            if (o21.Estado_Avance >= 50)
                                Console.WriteLine("Los códigos de obras con más del 50% de avance son: {0}", o21.Codigo_Obra);
                        }
                        break;
                    case 3:
                        ArrayList Lista_Submenu_3 = new ArrayList();
                        Lista_Submenu_3 = empresa1.Lista_Obras_Finalizadas;
                        foreach (Obra o12 in Lista_Submenu_3)
                        {
                            Console.WriteLine("Los códigos de obras finalizadas son {0}", o12.Codigo_Obra);
                        }
                        break;
                    case 4:
                        ArrayList Lista_Submenu_4 = new ArrayList();
                        Lista_Submenu_4 = empresa1.Lista_Obras;
                        foreach (Obra Ob in Lista_Submenu_4)
                        {
                            Console.WriteLine("Los jefes de obra son {0}", Ob.Jefe_de_obra.NombreApellido);
                        }
                        break;
                    default:
                        Console.WriteLine("Ingrese un término adecuado al menú.");
                        Console.ReadKey(true);
                        break;
```

```
case "5":
        Console.WriteLine("Ingrese el código de obra que desea modificar y el nuevo estado de avance:");
        try
        {
            int cod_obra_a_modificar = int.Parse(Console.ReadLine());
            int nuevo_estado_avance = int.Parse(Console.ReadLine());
            if ((nuevo_estado_avance < 0)|| (nuevo_estado_avance > 100))
             throw new ErrorDeCarga("Ingrese un estado de avance válido");
            else
            {    ArrayList Caso_5_Lista= new ArrayList();
        Caso_5_Lista= empresa1.Lista_Obras;
        foreach (Obra o13 in Caso_5_Lista){
            if (cod_obra_a_modificar == o13.Codigo_Obra)
            { o13.Estado_Avance=nuevo_estado_avance;
                if (nuevo_estado_avance==100)
                    empresa1.Agregar_Obra_Finalizada(o13);
                break;}
          }
        }
        }
        catch (ErrorDeCarga a7)
        {
            Console.WriteLine(a7.motivo);
        }
        Console.WriteLine("Presiona cualquier tecla para continuar...");
        Console.ReadKey(true);
        break;

    case "6":
        int legajo_a_borrar;
        Console.WriteLine("Ingrese el numero de legajo del jefe a eliminar: ");
        legajo_a_borrar = int.Parse(Console.ReadLine());
        EliminarJefe(empresa1,legajo_a_borrar);
        Console.WriteLine("Presiona cualquier tecla para continuar...");
        break;
```

```
1 referencia
public static void AñadirObrero(EmpresaConstructora emp)
{
    int dec;
    string nomyape;
    int dni;
    int legajo;
    int sueldo;
    string cargo;
    Console.WriteLine("Ingrese nombre y apellido del obrero: ");
    nomyape = Console.ReadLine();
    Console.WriteLine("Ingrese el DNI: ");
    dni = int.Parse(Console.ReadLine());
    Console.WriteLine("Ingrese el legajo: ");
    legajo = int.Parse(Console.ReadLine());
    Console.WriteLine("Ingrese el sueldo: ");
    sueldo = int.Parse(Console.ReadLine());
    Console.WriteLine("Ingrese el cargo: ");
    cargo = Console.ReadLine();
    Obrero Obr0 = new Obrero(nomyape, dni, legajo, sueldo, cargo);
    Console.WriteLine("Ingrese a que grupo quiere que ingrese: ");
         try
    {
        dec = int.Parse(Console.ReadLine());
        if ((dec < 0) || (dec > 8))
            throw new ErrorDeCarga("Ingrese un valor de grupo válido");
        else
        {       foreach (Grupo g in emp.Lista_Grupos)
          {
    if  (g.Numero_Grupo == dec)
        {  g.Añadir_Obrero(Obr0);
          break;}
      }
    }
   }
    catch (ErrorDeCarga a5)
    {
        Console.WriteLine(a5.motivo);
    }

    Console.ReadKey(true);

}
```

```
1 referencia
public static void EliminarObrero(EmpresaConstructora emp)
        {
            int legajo_a;
            int num_g_a;

            Console.WriteLine("Ingrese el número de grupo en el que se encuentra el obrero: ");
            try
            {
                num_g_a = int.Parse(Console.ReadLine());
                if (num_g_a < 0 || num_g_a > 8)
                    throw new ErrorDeCarga("El grupo no existe");
                else
                {
                    Console.WriteLine("Ingrese el legajo: ");
                    legajo_a = int.Parse(Console.ReadLine());

                    foreach (Grupo g in emp.Lista_Grupos)
                    {
                        if (g.Numero_Grupo == num_g_a)
                        {
                            Obrero obreroAEliminar = null;
                            foreach (Obrero ob in g.Lista_Obreros)
                            {
                                if (ob.Legajo == legajo_a)
                                {
                                    obreroAEliminar = ob;
                                    break;
                                }
                            }
                            if (obreroAEliminar != null)
                            {
                                g.Eliminar_Obrero(obreroAEliminar);
                                Console.WriteLine("Obrero eliminado correctamente.");
                            }
                            else
                            {
                                Console.WriteLine("No se encontró el obrero con legajo {0} en el grupo {1}.", legajo_a, num_g_a);
                            }
                        }
                    }
                }
            }
            catch (ErrorDeCarga a6)
            {
                Console.WriteLine(a6.motivo);
            }
        }
```

```
1 referencia
public static JefeObra AgregarJefe()
{
    string nombreyape;
    int dni;
    int nlegajo;
    int sueldoj;
    string cargo;
    int bonifExtra;
    int grupo_a_cargo;
    Console.WriteLine("Ingrese nombre y apellido del jefe: ");
    nombreyape = Console.ReadLine();
    Console.WriteLine("Ingrese el DNI:");
    dni = int.Parse(Console.ReadLine());
    Console.WriteLine("Ingrese el legajo:");
    nlegajo = int.Parse(Console.ReadLine());
    Console.WriteLine("Ingrese el sueldo:");
    sueldoj = int.Parse(Console.ReadLine());
    Console.WriteLine("Ingrese el cargo:");
    cargo = Console.ReadLine();
    Console.WriteLine("Ingrese la bonificacion del jefe de obra:");
    bonifExtra = int.Parse(Console.ReadLine());
    Console.WriteLine("Ingrese grupo a cargo del jefe de obra: ");
    grupo_a_cargo = int.Parse(Console.ReadLine());
    JefeObra obrero_jefe = new JefeObra(dni, nombreyape, nlegajo,  sueldoj, cargo,  grupo_a_cargo, bonifExtra);
    Console.ReadKey(true);
    return obrero_jefe;
}
1 referencia
public static void EliminarJefe(EmpresaConstructora emp, int legajo_a_borrar)
{
    ArrayList Lista_de_gruposl2= new ArrayList();
    Lista_de_gruposl2 = emp.Lista_Grupos;
    foreach (Grupo g22 in Lista_de_gruposl2)
    {
    foreach (Obrero job in g22.Lista_Obreros)
        {
        if (job.Legajo == legajo_a_borrar)
        {
            g22.Eliminar_Obrero(job);
            break;
        }
        }
    }
    Console.ReadKey(true);
}
```

```
}
1 referencia
public static void AsignarObra(EmpresaConstructora emp, int num_g, int nuevo_cod_o)
{
    ArrayList Lista_grupos_7= new ArrayList();
    Lista_grupos_7 = emp.Lista_Grupos;
    foreach (Grupo g7 in Lista_grupos_7)
    {
        if (num_g == g7.Numero_Grupo)
        {
            g7.Codigo_Obra_A_Cargo = nuevo_cod_o;
            break;
        }
    }
    Console.WriteLine("El producto se modifico con éxito");
    Console.ReadKey(true);


}
```

```
1 referencia
public static void AgregarObra(EmpresaConstructora emp, JefeObra Jefe_a)
{
    string NombrePropietario;
    int dniPropietario;
    int codigoObra;
    string tipoObra;
    int diasRestantes;
    int estadoAvance;

    Console.WriteLine("Ingrese nombre, dni del propietario, código de obra, tipo de obra, días restantes de la misma y el estado de avance en un número entre el 0 y el 100");
    NombrePropietario = Console.ReadLine();
    dniPropietario = int.Parse(Console.ReadLine());
    codigoObra = int.Parse(Console.ReadLine());
    tipoObra = Console.ReadLine();
    diasRestantes = int.Parse(Console.ReadLine());
    try
    {
        estadoAvance = int.Parse(Console.ReadLine());
        if ((estadoAvance < 0)|| (estadoAvance > 100))
         throw new ErrorDeCarga("Ingrese un estado de avance válido");
        else
        { Obra Obra1 = new Obra(NombrePropietario,  dniPropietario, codigoObra,  tipoObra,  diasRestantes, estadoAvance, Jefe_a);
    emp.Agregar_Obra(Obra1);
    }
    }
    catch (ErrorDeCarga A6)
    {
        Console.WriteLine(A6.motivo);
    }

    Console.ReadKey();

}
```