Proyecto 1:

Una empresa constructora tiene varias obras en ejecución y 8 grupos de obreros que trabajan en las obras. De cada obra se conoce el nombre y dni del propietario, código interno (se lo asigna el sistema), tipo de obra (construcción, remodelación, ampliación, etc.), estado de avance (porcentaje), el jefe de obra y el costo. De cada obrero se almacena su nombre y apellido, dni, nro legajo, sueldo y cargo (capataz, albañil, peón, plomero, electricista, etc.). El jefe de obra ***es un obrero*** responsable de una obra por lo cual cobra una bonificación especial a parte de su sueldo y dirige un grupo de obreros que trabajan en dicha obra. De cada grupo de obreros se conoce el código de obra en la queestán trabajando (0 en caso de no estar asignado a ninguna obra) y los integrantes.

Se deberá desarrollar una aplicación, utilizando las clases que considere necesarias, utilizando herencia cuando corresponda. La aplicación debe proveer, mediante un menú, las siguientes funcionalidades:

- Contratar un obrero nuevo (se agrega a la empresa y a un grupo)
- Eliminar un obrero (se elimina de la empresa y de su grupo)
- Contratar a un jefe de obra (se asigna a una obra existente) y se le asocia un grupo de obreros libre. Si no existe ningún grupo libre se debe levantar una excepción que indique lo sucedido.
- Submenú de impresión: listado de obreros, de obras en ejecución, de obras finalizadas y de jefes, porcentaje de obras de remodelación sin finalizar.
- Modificar el estado de avance de una obra. Si el estado de avance llega al 100% la obra debe darse por finalizada, se elimina del listado de obras en ejecución y se guarda en el de obras finalizadas.
- Dar de baja a un jefe (se elimina de la empresa, se desvincula de la obra y se libera el grupo de obreros asignado)

Desarrollo:

En cada proyecto se espera que:

- Se haga el diagrama de clases UML
- Se diseñe e implemente cada una de las clases completas: variables de instancia, constructores, propiedades y métodos de instancia básicos **(es decir, con los métodos completos aún cuando en la aplicación no se los utilice)**
- Se haga la aplicación donde se crean y cargan los objetos intervinientes (se puede usar carga desde archivo de texto, NO es obligatorio). Por ejemplo: si el proyecto es sobre un supermercado que tiene productos y proveedores, deberán crear el supermercado, cargarle algunos productos y algunos proveedores para luego poder dar respuesta a las opciones del menú solicitado. Si el enunciado no lo solicita, no deben simular atención de clientes y proveedores, ni ventas, ni reposiciones, etc.. (para acotar la longitud del desarrollo)

- Se presenta **el menú de opciones** solicitado en el problema y se implementan las funciones que dan respuesta a esos ítems. En la mayoría de los casos, deben desarrollar funciones del Main que invocan a los métodos de las clases diseñadas. En menor escala, se invocan desde el menú los métodos ya implementados en las clases. La idea es que **NO pongan adentro de las clases** como métodos todos los requerimientos del menú porque simplifican la solución y personalizan las clases en base al enunciado (y conceptualmente es erróneo!). La aplicación debe plantear la lógica de resolución del problema haciendo uso de las clases intervinientes.
- Deben usar herencia que se verá reflejado en el código y en el diagrama UML.
- Es obligación que implementen **al menos un manejador de excepciones,** que se pide en forma explícita en cada proyecto. Si desean agregar otros, no hay problema.