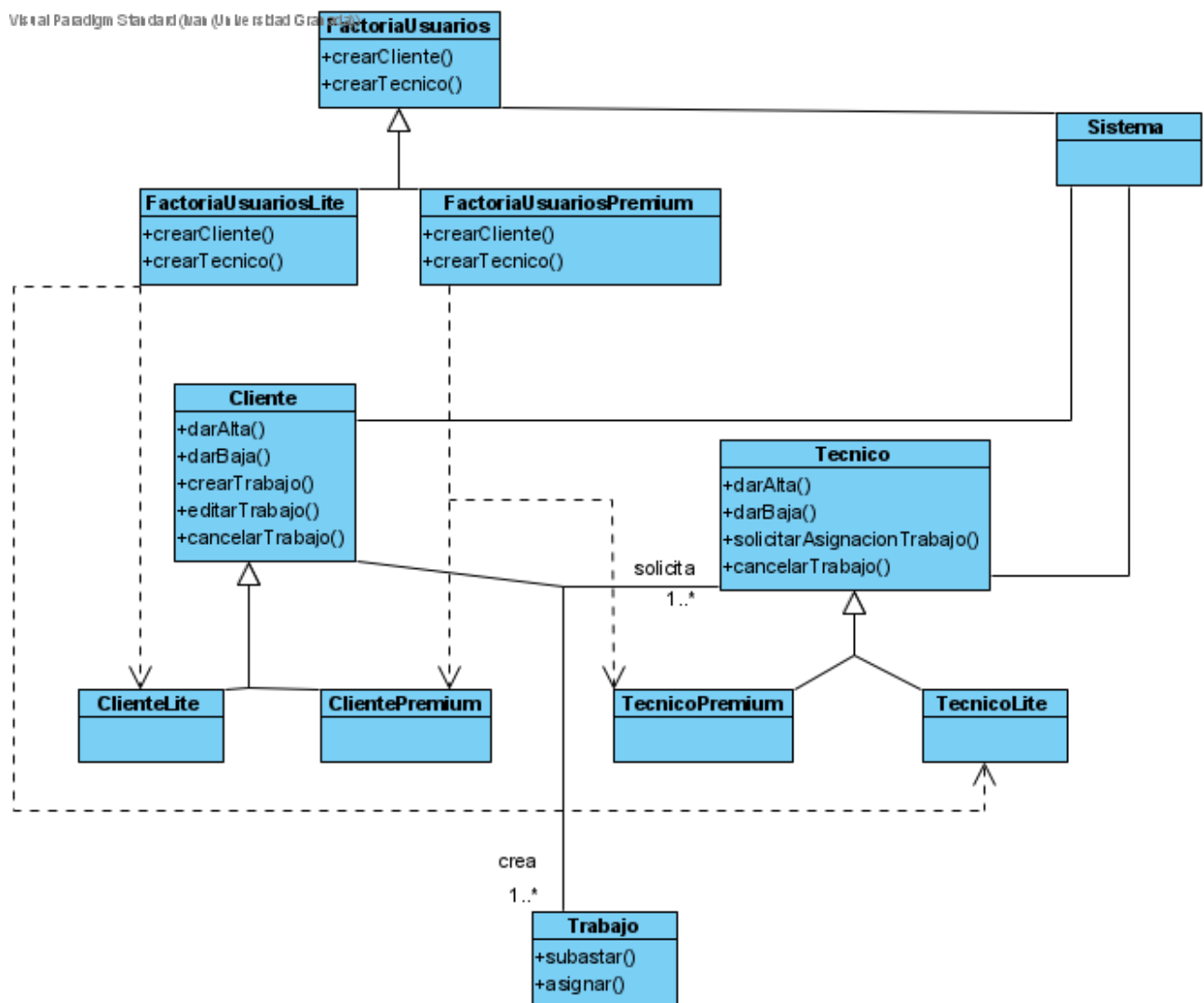


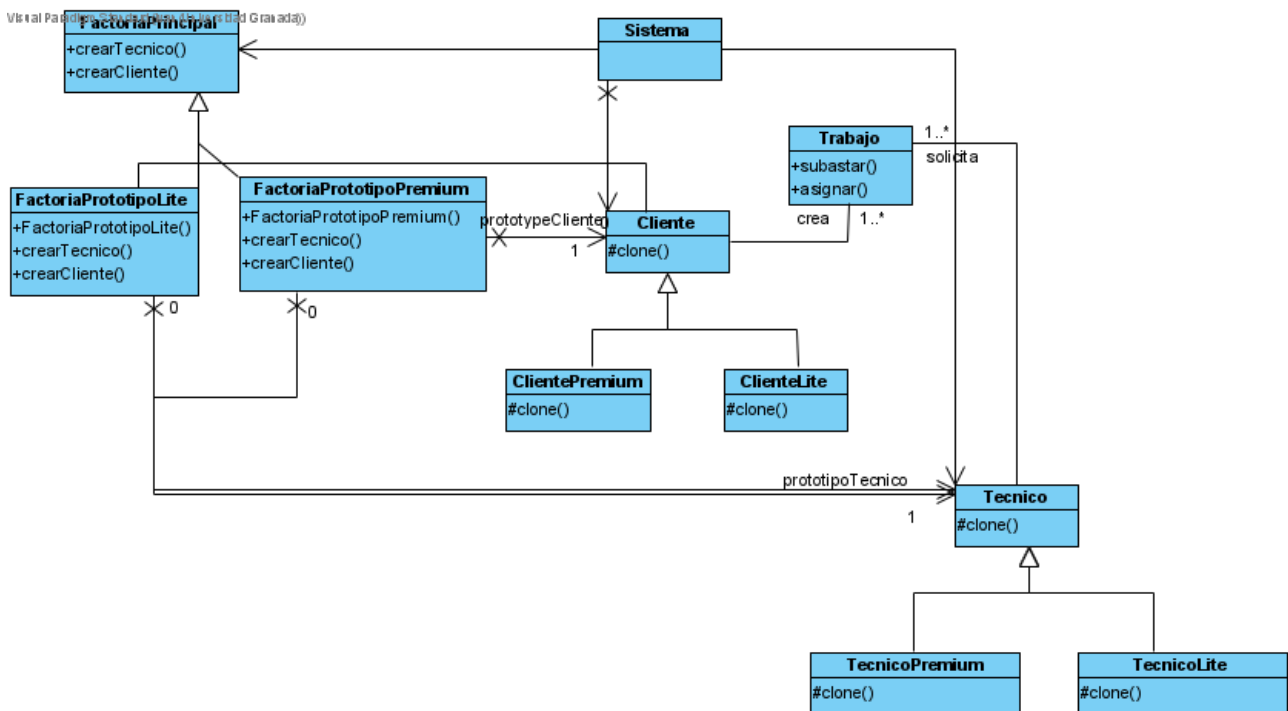
## Práctica 1. Sesión 2: Aplicación de patrones Factoría Abstracta, Método Factoría y Prototipo en el caso SOSEMADO.

### El diseño original

Para el programa basado en los patrones Factoría Abstracta y Método Factoría, se diseñó el siguiente diagrama



De manera análoga, para el programa basado en los patrones Factoría Abstracta y Prototipo:



## La implementación

En Java:

- La implementación general de las factorías se hizo siguiendo los diseños previos.
- Para integrar las hebras, se hizo un “bot autopujador”, el cual es lanzado al crear un trabajo, con el objetivo de ir leyendo el precio y bajar un euro cada 2 segundos.
- La interfaz de texto se hizo en una clase aparte, con todo lo referente a los menús y funciones entre la vista y el modelo.

En Ruby:

- La implementación general de las factorías se hizo siguiendo los diseños previos, aunque la clase clone() debería ser de clase (self.clone()).
- En **DSPrototype::Sistema.main()** se ha implementado un supuesto asistido por el usuario para probar lo principal.

## El resultado

En la versión de Ruby, el programa nos da un supuesto donde, a través de una factoría, creamos un cliente con un trabajo y 2 técnicos que pujarán por él.

```

Windows PowerShell
PS C:\Users\Ivan\Desktop\Clase\DS\Practicas\P1\S2\DS_Sesion2-Prototype> ruby .\Main.rb
Escoja el tipo de factoria (0 lite, 1 premium): 1
la factoria es de tipo premium

Vamos a añadir un nuevo cliente, introduzca el nombre del nuevo cliente: Juan
el nuevo cliente es: Cliente > [Nombre : Juan ; Importancia : 2.0 ; Trabajos creados : [] ]

Vamos a añadir un nuevo técnico, introduzca el nombre del nuevo técnico: Pepe
el nuevo tecnico es:
Tecnico > [Nombre : Pepe ; Valoración : 3.0 ; ¿Es en activo? : false ; Trabajos pedidos : [] ]

El cliente Juan va a añadir un nuevo trabajo, introduzca la descripción del trabajo: Poner grifo
Introduzca el precio máximo que pagaría usted por dicho trabajo: 200
El nuevo trabajo es: Poner grifo

Vamos a añadir un técnico pujador, introduzca el nombre del técnico pujador: Oscar
el tecnico pujador es:
Tecnico > [Nombre : Oscar ; Valoración : 3.0 ; ¿Es en activo? : false ; Trabajos pedidos : [] ]

Vamos a sacar un trabajo a subasta
Se comienza a subastar el trabajo Poner grifo
El tecnico Oscar va a hacer una oferta, introduzca la oferta: 190
Recibiendo puja del tecnico Oscar su oferta es la mejor: 190.0

El tecnico Pepe va a hacer una oferta, introduzca la oferta 180
Recibiendo puja del tecnico Oscar su oferta es la mejor: 180.0

Vamos a cerrar la subasta
El tecnico Oscar ha ganado la puja
PS C:\Users\Ivan\Desktop\Clase\DS\Practicas\P1\S2\DS_Sesion2-Prototype>

```

En la versión de Java, la CLI luce con varios menús menús para poder ir manejando las cosas del sistema.

```

*Menu del sistema creado*
Trabajo ---> {Descripción: Arreglar lavabo, Presupuesto: 200(
Trabajo ---> {Descripción: Alicatar, Presupuesto: 5000.0}
Elija una opción
1) Línea lite
2) Línea Premium
1
Elija una opción
1) Añadir cliente
2) Añadir tecnico
3) Seleccionar cliente
4) Seleccionar tecnico
5) Mostrar trabajos en subasta
6) Atras
1
Introduzca el nombre del cliente: Juan
Cliente añadido exitosamente
Elija una opción
1) Añadir cliente
2) Añadir tecnico
3) Seleccionar cliente
4) Seleccionar tecnico
5) Mostrar trabajos en subasta
6) Atras
2
Introduzca el nombre del tecnico: Oscar
Tecnico añadido exitosamente
Elija una opción
1) Añadir cliente
2) Añadir tecnico
3) Seleccionar cliente
4) Seleccionar tecnico
5) Mostrar trabajos en subasta
6) Atras
3
Elija una opción
Los clientes son:
0) Cliente(trabajos=[], nombre=Juan, importancia=1.0)
Elija una opción
1) Añadir trabajo
2) Seleccionar trabajo
3) Atras
1
Introduzca descripción del trabajo: Poner grifo
Introduzca precio maximo del trabajo: 200
Elija una opción
1) Añadir trabajo
2) Seleccionar trabajo
3) Atras
Elija una opción
1) Abrir subasta trabajo
2) Cerrar subasta trabajo
3) Atras
El autoPujador ha pujado.
2
Finaliza la subasta
Elija una opción
1) Abrir subasta trabajo
2) Cerrar subasta trabajo
3) Atiga*autovujador na pujado.
El autoPujador ha pujado.
1
Elija una opción
1) Mostrar trabajo actual
2) Solicitar/pujar por un nuevo trabajo
3) Atras
El autoPujador ha pujado.
2
Elija una opción
Los trabajos en subasta son:
0) Trabajo ---> {Descripción: Poner grifo, Presupuesto: 119.0}
El autoPujador ha pujado.
0
Trabajo ---> {Descripción: Poner grifo, Presupuesto: 118.0}
Introduce la cantidad que vayas a pujar (menor que la actual): El autoPujador ha pujado.
El autoPujador ha pujado.
110

```

Adjuntamos también el diagrama generado al revertir el código de Java a Visual Paradigm. (En Ruby no se podía hacer *Reverse Code*)

