Práctica ORB 2. Invocación remota a n objetos de una clase

Objetivos específicos que persigue la realización del trabajo:

- Transformación de un programa que realiza una llamada a objetos locales de una misma clase a otro programa que realiza la llamada de forma transparente a los mismos objetos que residen ahora en un servidor remoto.
- Construcción de un servidor que soporte un número no limitado de instancias de una determinada clase.

Fecha	Grupo	Tiempo de resolución	

Apellidos y nombre

Calificación

Enunciado

Sea el ejemplo del servicio de la Agenda remota, visto en el practica anterior. Ahora se desea crear un servidor que soporte de forma remota la creación y el acceso a **cualquier número** de objetos de la clase Agenda. Por ejemplo, el cliente puede ser el siguiente:

```
/*****************************************************************/
// FICHERO: ClienteAgendas
/*****************************************************************/
public class ClienteAgendas{

	public static void main(String[] args) {

		Agenda agendaTelefonica = new Agenda ();
		Agenda guiaClaves = new Agenda ();
		agendaTelefonica.asociar("Juan", 66756677);
		guiaClaves.asociar ("Moodle", 23323);
		agendaTelefonica.asociar("Pepe", 644454456);
		System.out.println("Telefono Juan = " + agendaTelefonica.obtener("Juan"));
		System.out.println("Clave Moodle = " + guiaClaves.obtener("Moodle"));
	}
}
```

Para que tanto el proxy del cliente como servidor puedan distinguir sobre el objeto sobre el que se realiza la invocación, es necesario asignar un identificador a cada objeto (`objId`). Este identificador se creará cuando el servidor reciba la petición de creación de un nuevo objeto Agenda y se devolverá al proxy del cliente como resultado de la creación.

Por otra parte, tras la creación del objeto, el servidor debe guardar la tupla <objId, objeto> para poder acceder posteriormente a éste. Use una tabla hash (`ObjsTable`) para almacenar dichas tuplas.

Cuando el proxy del cliente invoque las operaciones asociar u obtener sobre un determinado objeto deberá incluir el identificador `objId` en sus peticiones.

Se pide:

1) Defina los diferentes tipos de mensajes a intercambiar y su formato (campos y tipo de los campos) para las operaciones crear agenda, asociar un contacto y obtener un contacto.
2) El proxy de Agenda. (Clase Agenda que usa para el cliente).
3) Explique brevemente qué estructura de datos usa el servidor para soportar cualquier número **no limitado** de objetos de tipo Agenda. Considere que el primer identificador de objeto (ObjId) es 0.
4) El código del servidor, comentado adecuadamente.