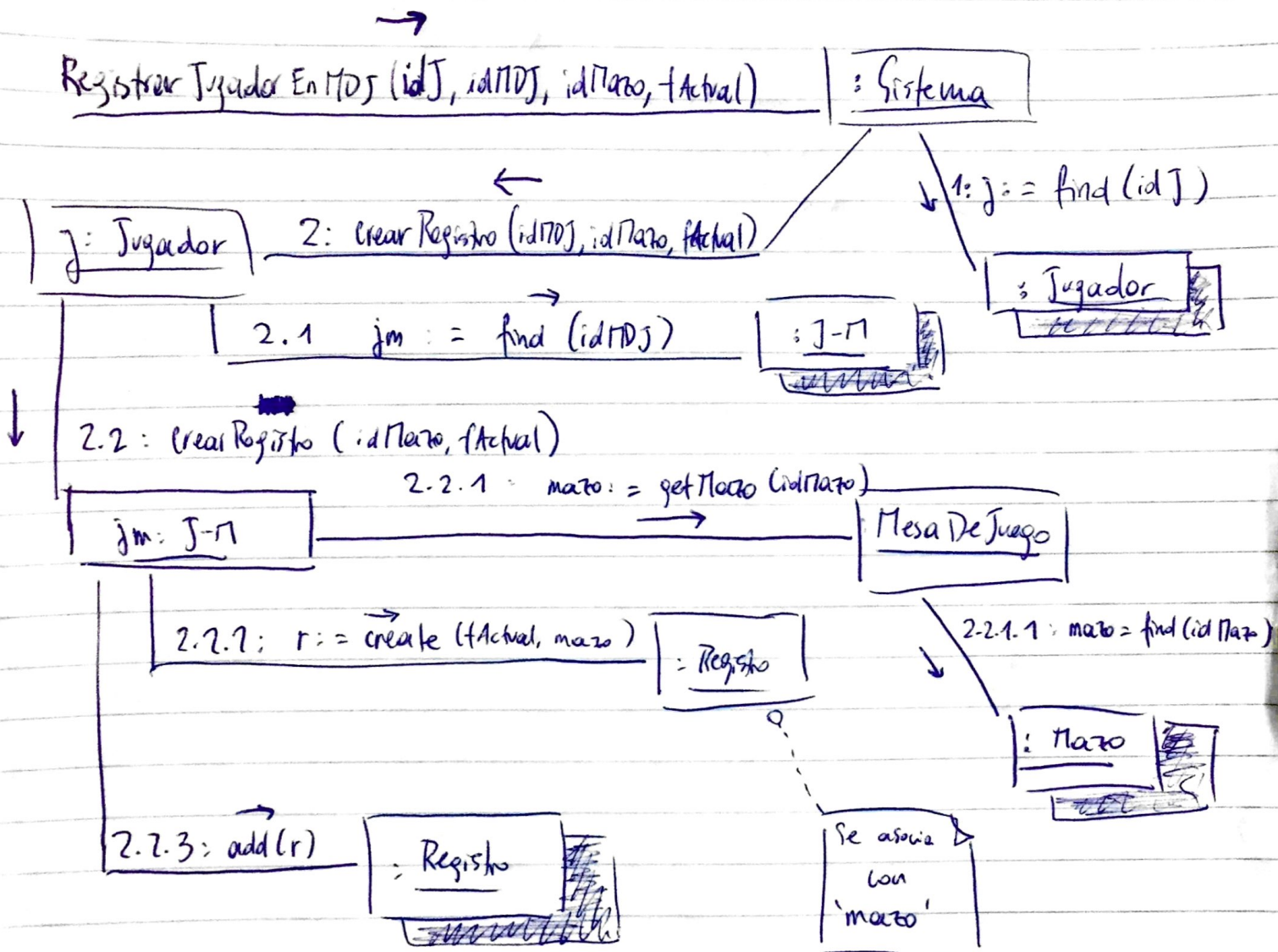


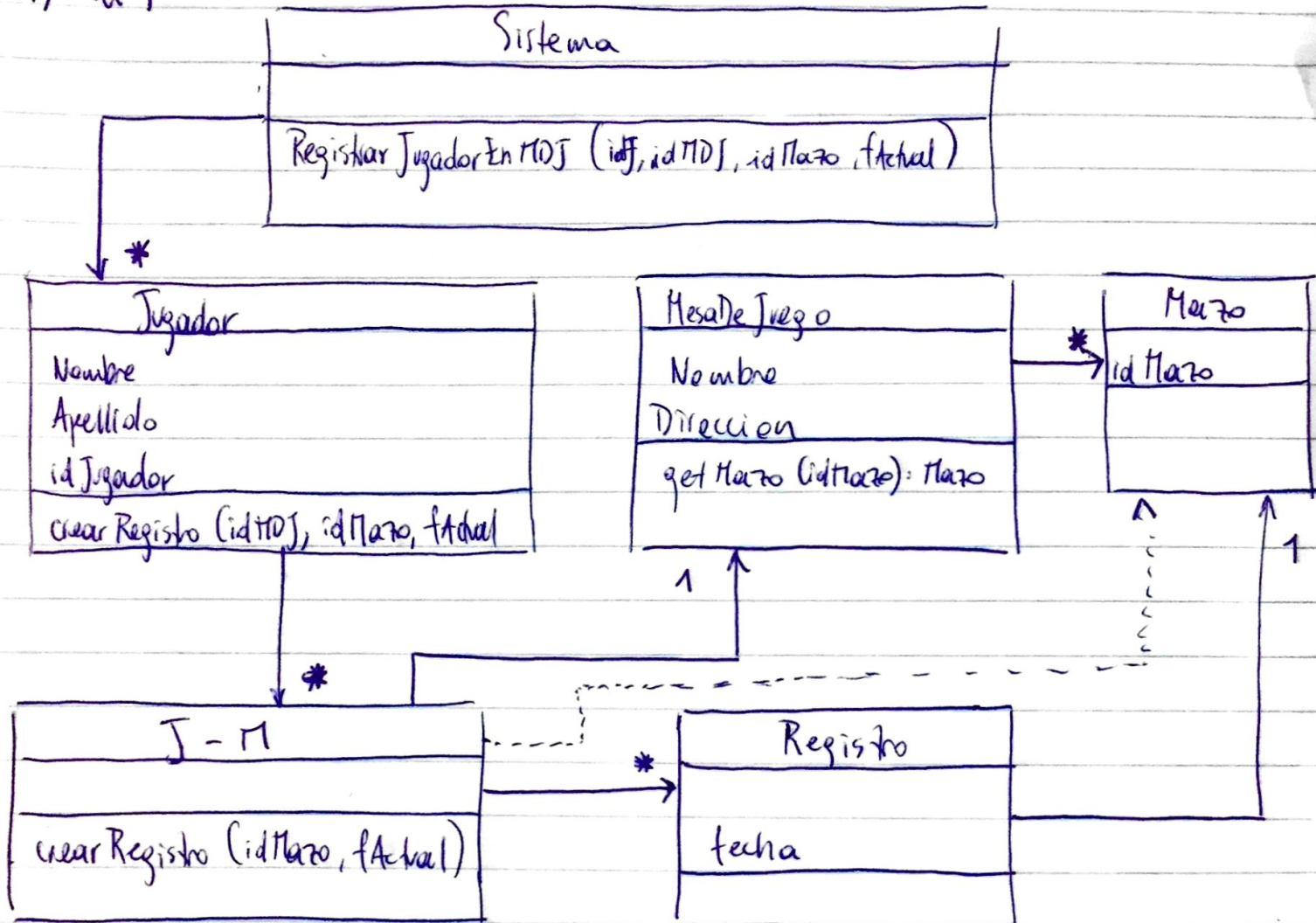
1)

i)



Para esta solución se consideró que Jugador almacena los tipos asociados J-M con la clave que tiene su MesaDeJuego asociada. Otra solución es utilizar colección sin clave, y en este caso habrá que recorrer todos los J-M hasta encontrar el que corresponda.

1) ii)



2) i)

```
Class Controlador {
```

```
private:
```

```
static ControladorTema * instance;
```

```
ControladorTema ();
```

```
IDictionary * temas;
```

```
public:
```

```
static ControladorTema * getInstance();
```

```
void agregarTema (string, string);
```

```
void agregarComentarioTijp (int, int, string);
```

```
void eliminarComentario (int, int);
```

```
}
```

```
class Tema: ICollectionable {
```

```
private:
```

```
int id;
```

```
string titulo;
```

```
string correoUsuarioCreador;
```

```
IDictionary * comentarios;
```

```
public:
```

```
Tema (int, string, string);
```

```
void agregarComentario (string);
```

```
~Tema ();
```

```
void eliminarComentario (int);
```

```
void agregarComentarioTijp (int, string);
```

```
}
```



```
Class Apote : ICollectible {
```

```
private:
```

```
    int id;
```

```
public:
```

```
    Apote (int, Tema *);
```

```
    virtual ~ Apote ();
```

```
}
```

```
class Comentario : public Apote {
```

```
private:
```

```
    string comentario;
```

```
    IDictionary * hijos;
```

```
public:
```

```
    Comentario (int, string);
```

```
    ~ Comentario ();
```

```
}
```

2) ii)

ControladorTema :: eliminarComentario (Int idTema, Int idComentario) {

 IKey * key = new KeyInt (idTema);

 Tema * t = (Tema *) temas → find (key);

 if (t != null) {

 t → borrarComentario (idComentario);

 delete (key);

 }

 else {

 delete (key);

 throw invalidArgument ("No existe tema");

 }

}

ControladorTema :: agregarTema (String titulo, String cuerpo) {

 Int idTema = Util → getId();

 Tema * t = new Tema (idTema, titulo, cuerpo);

 IKey * key = new KeyInt (idTema);

 temas → add (t, key);

 delete (key);

}

```
ControladorTema :: agregarComentario (int idTema, int idComentario, String comentario) {
```

```
    Key * key = new KeyInt (idTema);
```

```
    Tema * t = (Tema *) temas -> find (key);
```

```
    if (t != NULL) {
```

```
        t -> agregarComentario (idComentario, comentario);
```

```
        delete key;
```

```
    }
```

```
    else {
```

```
        delete key;
```

```
        throw invalid_argument ("No existe tema");
```

```
    }
```

```
}
```