

# Entrega 2 Practica sabelotodo

---

**Sergio Botero Uribe 200710001010**

**14/05/2007**

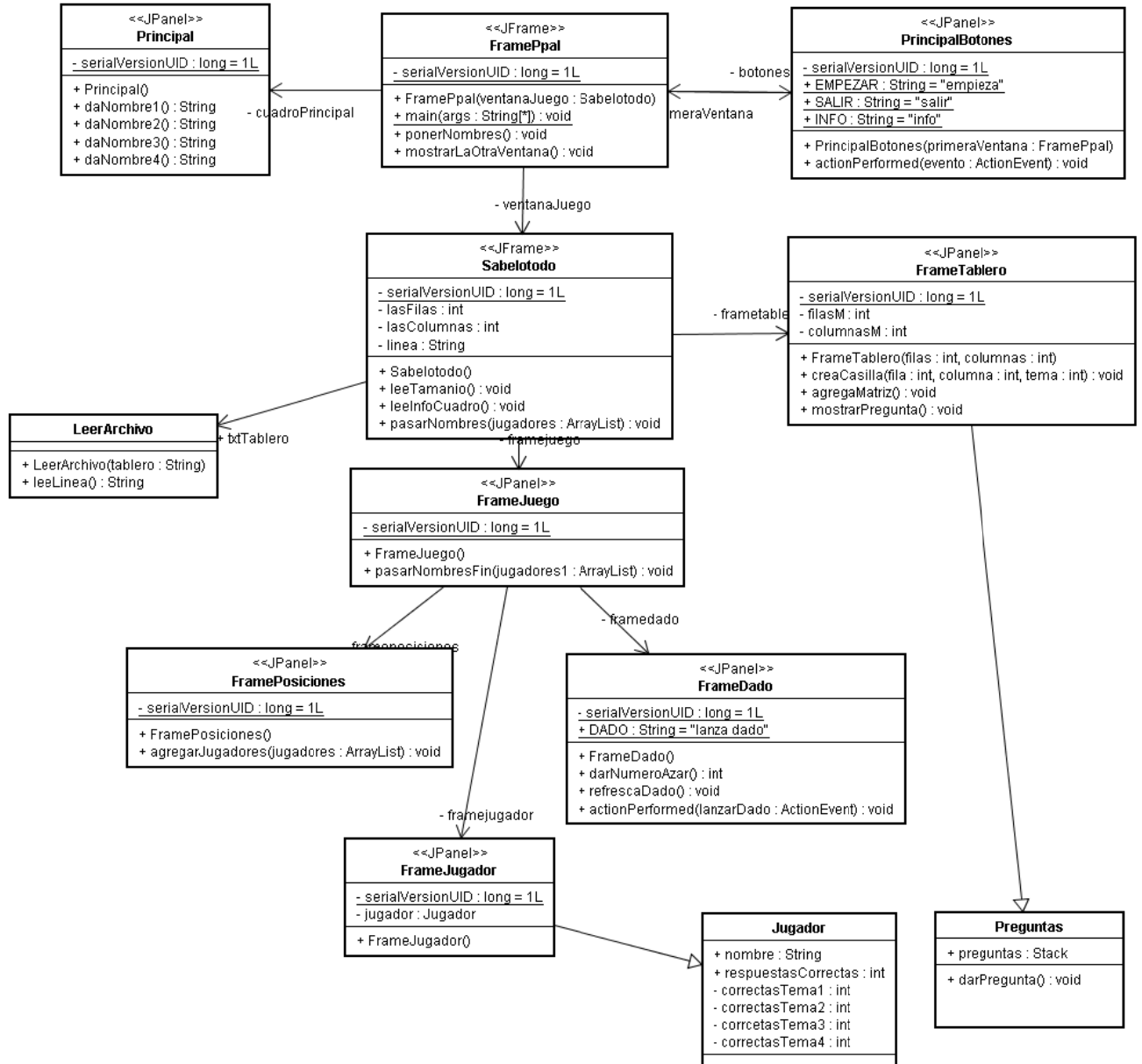
**Helmuth Trefftz Gómez**

Fundamentos de programación

Segunda entrega "sabelotodo".

# 1. Segunda versión del diagrama de clases.

Diagrama de Clases – en JUDE-



## 2. Diagrama de secuencia de operaciones principales.

Diagrama de secuencia de contestar pregunta.

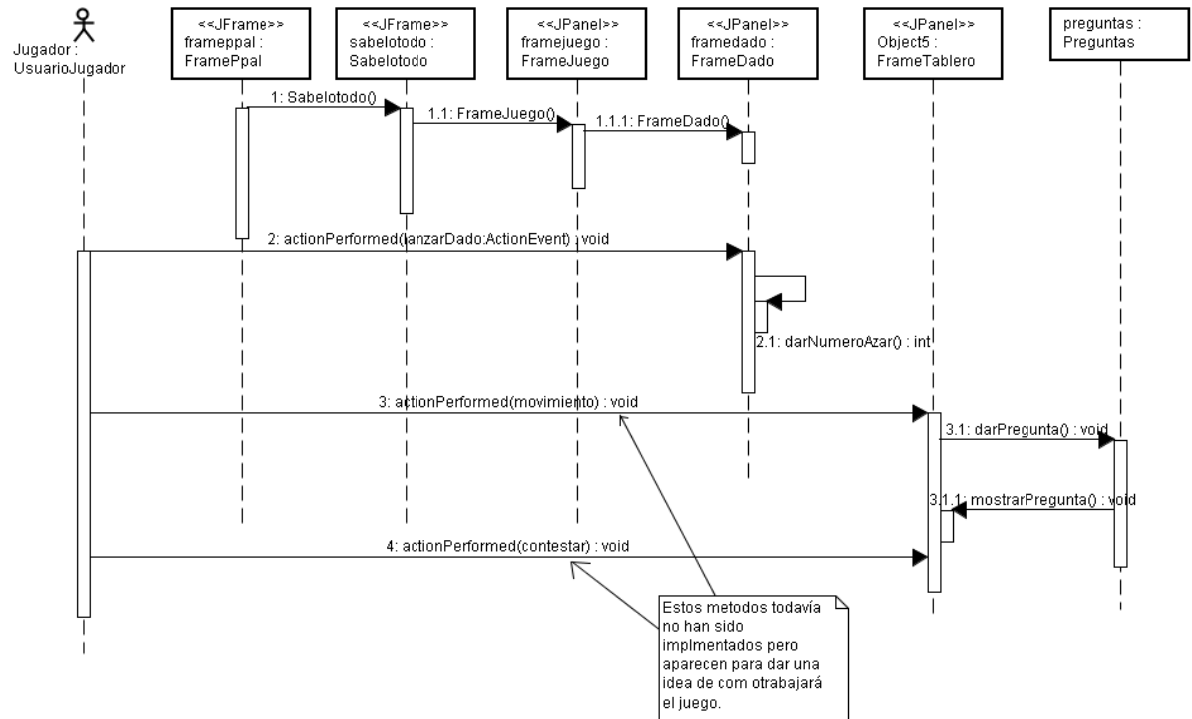


Diagrama de secuencia de mover ficha.

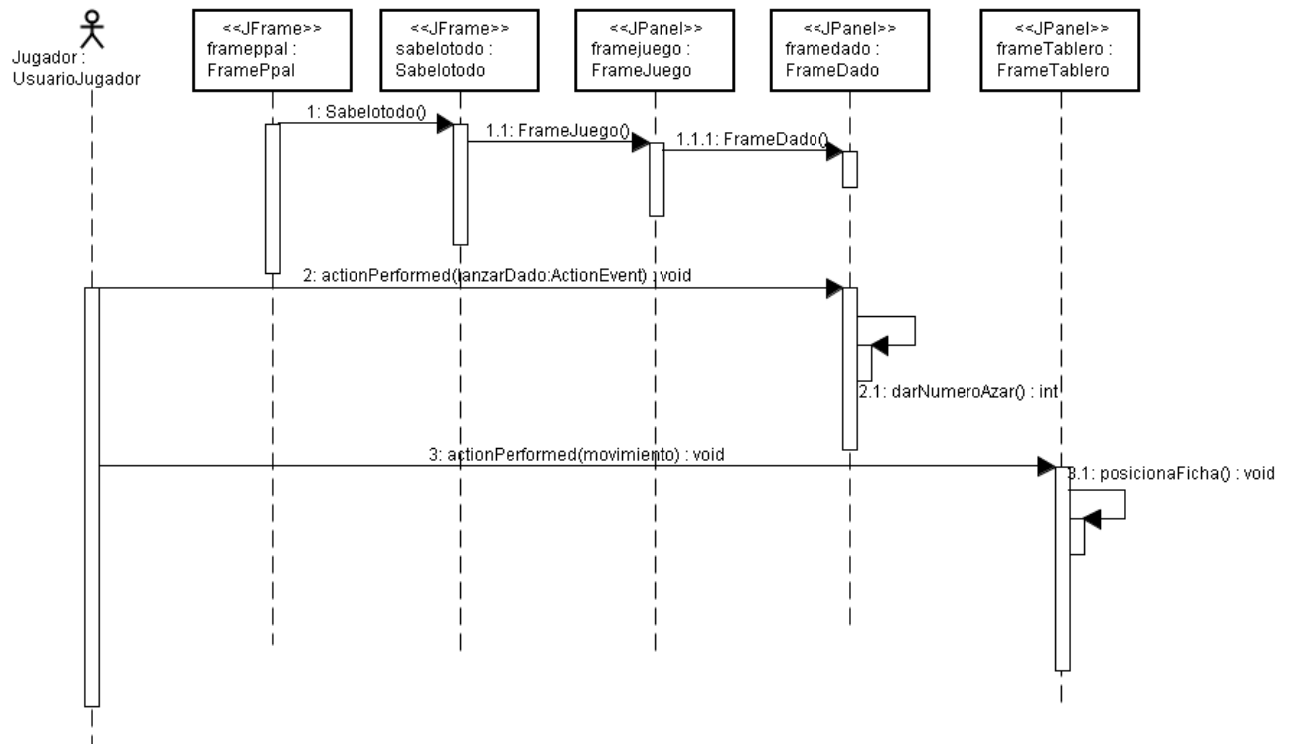
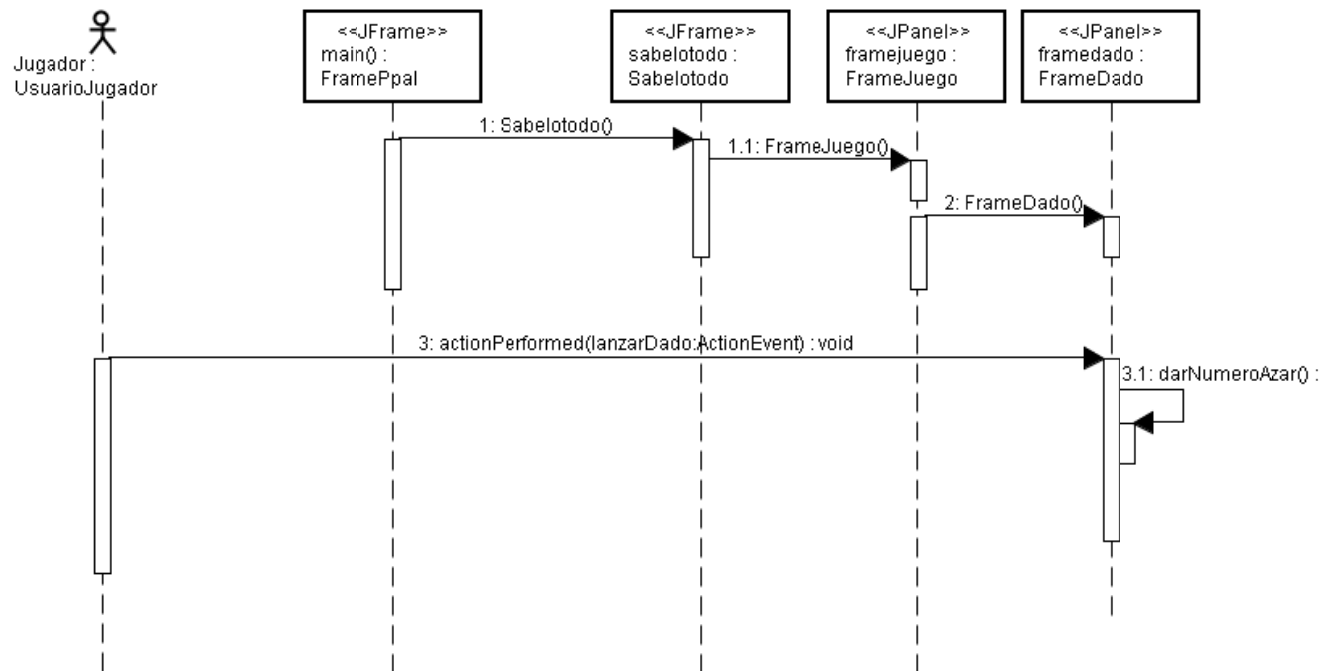


Diagrama de secuencias de tirar dado.

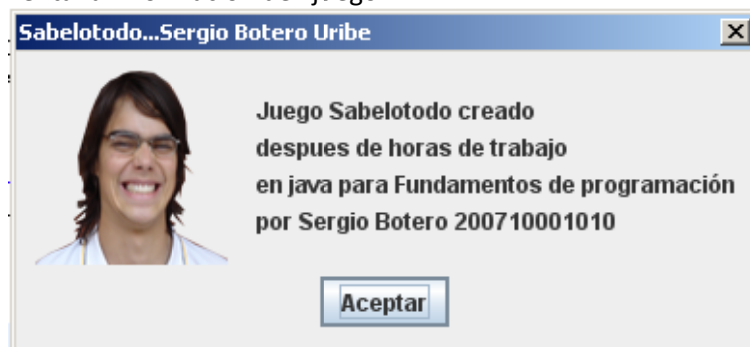


### 3. Interfaz gráfica

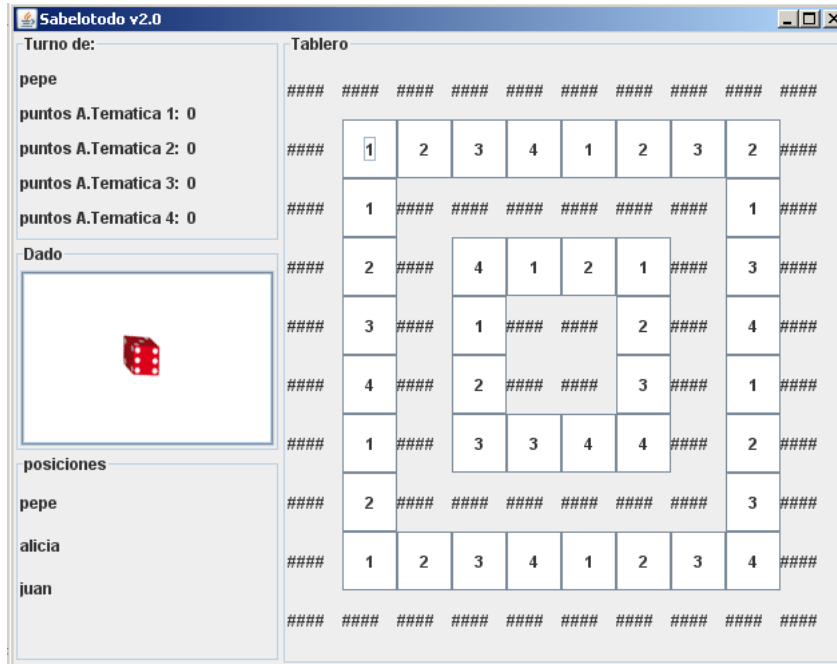
Pantalla principal



Ventana Información del juego



## Vista del Tablero



### 4. Códigos de métodos no triviales.

Hasta este momento parece que van a ser necesarios ciertos métodos que deben ser implementados en el mundo del problema, algunos pueden ser:

```
moverFicha();
refrescarPosicion();
mostrarPregunta();
```

Y su código a grandes rasgos sería probablemente así:

Para: moverFicha();// se manejaría todo desde una botonera con flechas

```
1 switch(boton){
    Case(1):
        Si (laCasillaArriba != null){
            jugador.modificarCasilla(fila, columna);
        }
        Break;
    Case(2):
        Si (laCasillaIzq != null){
            jugador.modificarCasilla(fila, columna);
        }
        Break;
    Case(3):
        Si (laCasillaAbajo != null){
            jugador.modificarCasilla(fila, columna);
        }
        Break;
    Case(4):
        Si (laCasillaDer != null){
```

```

        jugador.modificarCasilla(fila, columna);
    }
    Break;

}
movidasRestantes -- //así se podría mover el jugador usando una botonera, obviamente habría que
agregarle restricciones a cosas como devolverse o que se deje de mover cuando terminen los movimientos.

```

Para: refrescarPosicion(); este método busca tener actualizada la lista de posiciones del juego, el jugador que va de primero con más posibilidades de ganar, y su puntaje en general.

```

puntosActuales;
puntosJ1

```

```

Si (puntosJ1 > puntos actuales) entonces
    // este método no tengo muy claro como se va a implementar pero la idea es que los labels con los
    nombres de los jugadores y su puntaje esté variando de posición conforme avanza el juego.
    Agrega(puntosJ1)           //al Layout
    Agrega(puntosActuales)
    puntosAgrega=puntosJ1
repetir;                       // repetir para crear el ciclo que servirá a su vez de método de refresco.

```

Para: MostrarPregunta(); //las preguntas van a estar almacenadas en una pila, e irán saliendo conforme se llega a una casilla marcada con un numero de 1 a 4 que representa cada pila.

```

NumeroCasilla = Obtener tema casilla
Si (numeroCasilla ==1) entonces
    pilaTema1 = sacarPregunta
Sino(numeroCasilla==2) entonces
    pilaTema2 = sacarPregunta
Si (numeroCasilla ==3) entonces
    pilaTema3 = sacarPregunta
Sino(numeroCasilla==4) entonces
    pilaTema4= sacarPregunta

```

las preguntas serían mostradas en JOptionPane que den la posibilidad de una opción multiple.

5. Diseño de las pruebas a ser ejecutadas.
  1. Inicialización del Sabelotodo

No.	Descripción de la prueba	Datos de entrada	Resultado Esperado
1.1	Busca el archivo del tablero, la carga y crea el tablero.	El documento de texto con las especificaciones del tablero.	Se carga el tablero con las dimensiones y las casillas con las características definidas. Se termina la ejecución del programa si el archivo no está.
1.2	Variaciones del archivo de texto con las propiedades del tablero.	Que el número de filas y/o columnas sean 0 o	Mensajes de error por cada caso y terminado la

		valores negativos.	ejecución del programa.
1.3	Variación en la información para cada casilla.	Un número mayor de columnas o un número de tema no válido.	Mensaje de error reportando lo sucedido y terminando la ejecución del juego.
1.4	Determinar el número de jugadores y sus nombres.	Nombre de cada jugador.	Nombre de los jugadores en el panel de posiciones, iniciando con el jugador 1 en la primera posición por defecto.
1.5	Juego de un solo jugador.	Nombre de un solo jugador.	Mensaje aclarando que se necesitan mínimo dos jugadores para iniciar el juego.

## 2. Ejecución del juego

No.	Descripción de la prueba	Datos de entrada	Resultado Esperado
2.1	Lanzamiento del dado.	Ningún dato, pero se recibe la opción de un botón.	Se genera un número al azar entre el 1 y el 6.
2.2	Movimiento en casilla.	Al igual que el dado se genera en una acción y es donde se escoge la casilla donde se quiere mover.	El movimiento de la ficha en el lugar deseado.
2.3	Sacar una pregunta en pantalla y luego probar como adiciona el puntaje obtenido.	Ninguno, la pregunta aparece después de haberse parado en una casilla.	Que se cambie el puntaje del jugador que acaba de responder.
2.4	Revisar el manejo de turnos	Ninguno, ver cómo queda el tablero preparado para un nuevo jugador.	Que el dado esté otra vez en posición para ser lanzado y que la información del jugador que está en turno aparezca en pantalla.

## 6. Documentación del proceso de software.

Nombre del Rol	Descripción de la actividad	Tiempo estimado	Tiempo utilizado	Observaciones
Analista	Construcción del diagrama de clases en JUDE.	1.5 horas	1 hora aprox.	Instalación de la herramienta.
Analista	Construcción de los diagramas de secuencias en JUDE.	1.5 horas	2 horas 15 min. Aprox.	Tomó más tiempo del esperado ya que tuve que volver a ver la presentación y revisar los ejemplos.
Programador	Creación de la interfaz gráfica de juego.	4 horas	4 horas	El tiempo utilizado puede no ser el realmente usado ya que la construcción de la interfaz fue un proceso de varias sesiones de

				diferente duración que no quedó documentada.
Analista	Preparación del documento para la entrega (zip, pdf, impreso etc.).	15 minutos.	20 minutos.	Se tuvo problemas con la impresora. Y no tenía grapadora.