	Calidad, Medición y Estimación de Producto y Proceso Software	4º Grado en Ingeniería Informática Itinerario de Ingeniería del Software
	Examen Final de Teoría Convocatoria de febrero	5/2/2016

## Ejercicio 1: ¿Cuánto esfuerzo requiere diseñar el siguiente programa?

### Análisis de Punto Función con IFPUG, ISBSG ( 2,5 puntos)

Deseamos crear un sistema que simule una variante del juego de los dados, usando un cliente basado en Java Swing. Nuestros analistas han estudiado los requisitos del sistema y han elaborado el correspondiente Diagrama de Casos de Uso y de Clases y especificado brevemente las siguientes funcionalidades:

#### Documento de Visión:

Nuestro sistema deberá implementar la siguiente variante del juego. Juegan 2 jugadores además de la banca, rol que asume el propio sistema. Cada jugador parte con un saldo inicial de 1000€. Antes de cada jugada, cada jugador deberá realizar una apuesta por un importe que no supere su saldo.

Una vez los dos jugadores han apostado, tiran sus dados:

- El que saque 2,3 o 4, pierde lo apostado directamente.
- El que saque 7, gana el doble de lo apostado directamente.
- En otro caso, la banca también tirará los dados, comparándose su resultado con el de los jugadores. El jugador gana lo apostado si supera la puntuación de la banca y pierde lo apostado en caso de empate o menor puntuación.

Tras la jugada, el sistema muestra el resultado, quien gana y pierde y actualiza los saldos. Además, se almacenan y visualizan los resultados de las tiradas para posterior consulta.

Nuestros analistas han modelado la solución, diseñando los siguientes diagramas UML:

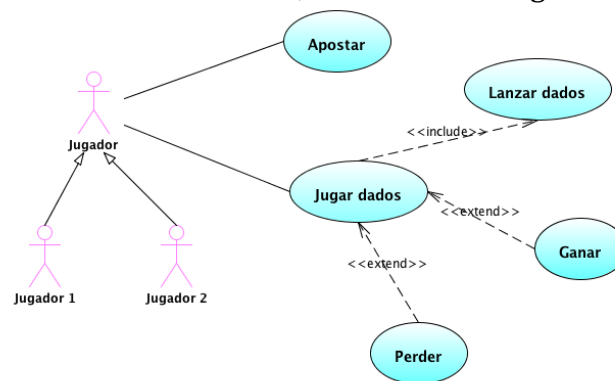


Ilustración 1. Diagrama de casos de uso

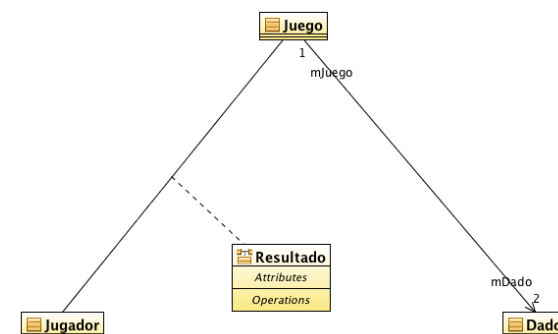


Ilustración 2. Diagrama de clases sin detallar

### Prototipo de la interfaz de usuario:

A los dos usuarios se les muestran 3 formularios (Ilust.3). Ellos interactúan con los formularios jfrmJugador1 y 2, pudiendo apostar y realizar jugada. La información de la apuesta y el resultado de los dados se muestra en el Tablero principal (jfrmPrincipal). Una vez los dos han tirado, si ha sido necesaria se actualiza los dados obtenidos por la banca y el resultado. También se actualiza un elemento JTextArea con una línea por jugada con el resultado.

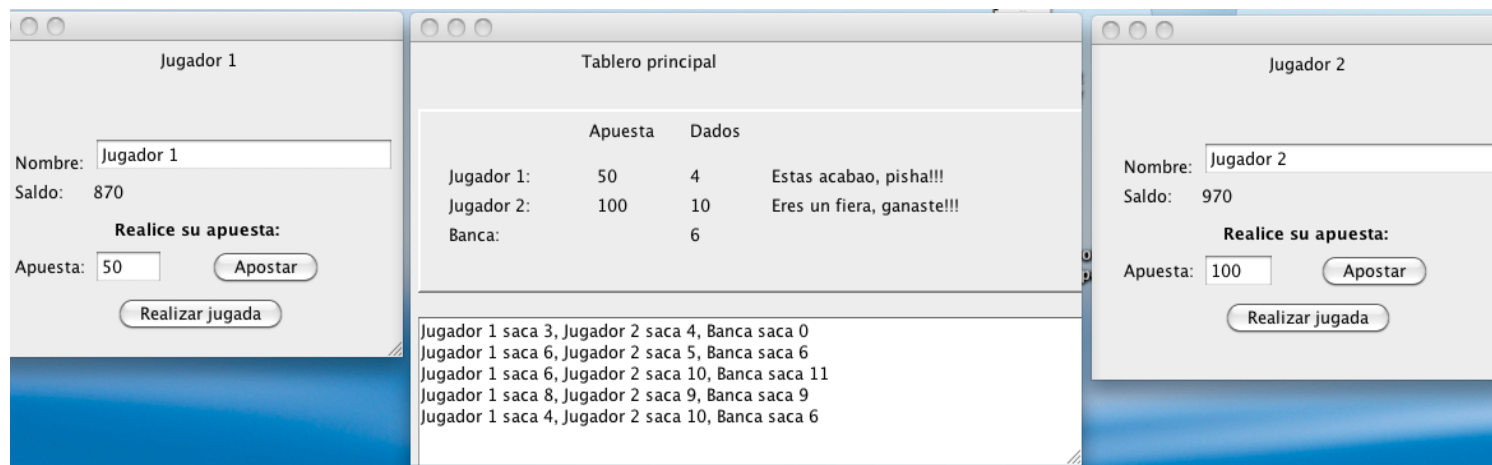


Ilustración 3. Formularios de la aplicación (jfrmJugador1, jfrmPrincipal y jfrmJugador2)

- a) Identifique las transacciones, su tipo, ficheros y datos elementales implicados que aparecen en el sistema y que habrán de recogerse en la tabla de puntos-función no ajustados. **(0,75 puntos)**

Transacción/Grupo Lógico	Tipo de Componente (EE, SE, GLDI, GLDIZ Y CE)	Número de ficheros y datos elementales	Lista de datos elementales

- b) Rellene la tabla de cálculo del total de puntos-función no ajustados: **(0.75 puntos)**

Descripción	Sencilla	Media	Compleja	Total PF
Nº Entradas externas	x 3	x 4	x 6	
Nº Salidas externas	x 4	x 5	x 7	
Nº Grupos Lógicos de Datos Internos	x 7	x 10	x 15	
Nº Grupos Lógicos de Datos de Interfaz	x 5	x 7	x 10	
Nº de Consultas Externas	x 3	x 4	x 6	
Total Puntos Función No Ajustados				

*\*Las tablas de cálculo de complejidad se encuentran en la última hoja del examen*

Las características particulares de este proyecto se consideran sencillas, por lo que la influencia será escasa y positiva. En concreto el ajuste reducirá en un 15% el tamaño obtenido inicialmente.

- c) Calcule el tamaño funcional necesario en la aplicación original en Punto-Función Ajustado. (0,5 puntos)

$$FA = 0.65 + (0.01 * SVA)$$

$$PFA = PFNA * FA$$

- d) ¿Qué **duración** requerirá el desarrollo de la modificación propuesta? Indique el resultado en **semanas**.

Use para ello las tablas ISBSG, sabiendo que el proyecto es una aplicación java monousuario. (0,5 puntos)

	Características	C	E
1	PC	0,503	0,409
2	Multi	0,679	0,341
3	4GL	0,578	0,393
4	Nuevo	0,739	0,359
5	PC-4GL	0,348	0,471
6	Multi-4GL	0,366	0,451
7	PC-4GL-Nuevo	0,250	0,515
8	Multi-4GL-Nuevo	0,240	0,518

para el resto, C=0,411 y E=0,328.

$$\text{Duración} = C \cdot \text{Tamaño}^E$$

## Ejercicio 2: Métricas Orientadas a Objeto y de Complejidad Ciclomática (2,5 puntos)

Una vez diseñada e implementada nuestra solución (de la que se entrega código en el anexo 1), deseamos calcular en el modelo la complejidad inherente a cada clase. Para ello se pide que calcule la métrica **Métodos ponderados por clase (WMC, weighted methods per class)**, teniendo en cuenta que la complejidad de cada método se obtendrá de la complejidad ciclomática de McCabe.

$$WMC = \sum_{i=1}^n c_i$$

- o **Importante, calcule la complejidad de cada método de la manera más sencilla**

Clase	Nº métodos	WMC
Controlador		
Juego		
Jugador		
Dado		
Resultado		

#### Ejercicio 4: Estimación con COCOMO.

(2 puntos)

Deseamos hacer una estimación usando el Modelo Constructivo de Costes II desarrollado por Barry Boehm. Para ello nos dicen que tanto los Factores de Escala como los Multiplicadores de Esfuerzo toman valores nominales salvo:

RCPX High	PDIF High	FCIL High	RESL High
-----------	-----------	-----------	-----------

a) Con los datos de tamaño del ejercicio 1 y con los datos aportados, **calcule el esfuerzo** en personas-mes necesario para el proyecto. (1 punto)

b) Proporcionalmente, **qué factor de los que no son nominales ha tenido más influencia** sobre la estimación del esfuerzo de proyecto, **RESL** o la unión de **RCPX+PDIF+FCIL**. Calcúlelo matemáticamente y razone por qué ocurre ello. (1 punto)

Fórmulas y tablas de ayuda:

$$PM_{nominal} = A * (Size)^B$$

$A = 2.94$

$$PM_{estimado} = PM_{nominal} \cdot \prod_{i=1}^7 EM_i$$

$$B = 0.91 + 0.01 \cdot \sum_{i=1}^5 W_i$$

Driver Cost	
RCPX	Fiabilidad y complejidad del producto
RUSE	Reutilización requerida
PDIF	Dificultad de la plataforma
PERS	Aptitud del personal
PREX	Experiencia del personal
FCIL	Facilidades de apoyo
SCED	Agenda

Factor de Escala Wj	
PREC	Precedencia
FLEX	Flexibilidad en el desarrollo
RESL	Dificultad de la plataforma
TEAM	Aptitud del personal
PMAT	Experiencia del personal

Tabla de conversión de Punto Función a Líneas de Código	
Lenguaje	LOC/PF
Java	53
C++	55
C	128

#### Ejercicio 4. Los gráficos en SCRUM (3 puntos).

Nos encontramos en medio de un proyecto que sigue la metodología Scrum. En el siguiente gráfico se recoge el seguimiento del desarrollo seguido.

SPRINT			INICIO		DURACIÓN															
1			1-feb-16		10															

## Ficheros o Grupos Lógicos de Datos Internos

		Tipos de datos elementales		
		1 a 19	20 a 50	51 ó más
Tipos de Registros	1	S	S	M
	2 a 5	S	M	C
	6 ó más	M	C	C

## Ficheros o Grupos Lógicos de Datos Externos

		Tipos de datos elementales		
		1 a 19	20 a 50	51 ó más
Tipos de Registros	1	S	S	M
	2 a 5	S	M	C
	6 ó más	M	C	C

## Entradas externas

		Tipos de datos elementales		
		1 a 4	5 a 15	16 ó más
Ficheros Referenciados	0 ó 1	S	S	M
	2	S	M	C
	3 ó más	M	C	C

## Salidas externas

		Tipos de datos elementales		
		1 a 5	6 a 19	20 ó más
Ficheros Referenciados	0 ó 1	S	S	M
	2 ó 3	S	M	C
	4 ó más	M	C	C

**Scalefactor Parameters**

	VLO	LO	NOM	HI	VHI	XHI
PREC	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	5.07	4.05	3.04	2.03	1.01	0.00
RESL	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	7.80	6.24	4.68	3.12	1.56	0.00

OK Cancel Help

**Early Design Parameters**

	XLO	VLO	LO	NOM	HI	VHI	XHI
RCPX	0.73	0.81	0.98	1.00	1.30	1.74	2.38
RUSE	XXXX	XXXX	0.95	1.00	1.07	1.15	1.24
PDIF	XXXX	XXXX	0.87	1.00	1.29	1.81	2.61
PERS	2.12	1.62	1.26	1.00	0.83	0.63	0.50
PREX	1.59	1.33	1.12	1.00	0.87	0.71	0.62
FCIL	1.43	1.30	1.10	1.00	0.87	0.73	0.62
SCED	XXXX	1.43	1.14	1.00	1.00	1.00	XXXX
USR1	XXXX	1.00	1.00	1.00	1.00	1.00	XXXX
USR2	XXXX	1.00	1.00	1.00	1.00	1.00	XXXX

OK Cancel Help