

**OBLIGATORIO (PARTE 2)**

<b>EVALUACIÓN</b>	OBLIGATORIO 2	<b>GRUPO</b>	TODOS	<b>FECHA</b>	Noviembre 2017
<b>MATERIA</b>	Diseño y Desarrollo de Aplicaciones				
<b>CARRERA</b>	Analista Programador				
<b>CONDICIÓN</b>	Entrega 11/12/2017				

**El objetivo del presente trabajo obligatorio es modelar e implementar la segunda versión prototipo de un juego de buscaminas en red.**

En esta segunda versión el desarrollo se focalizará en:

1. Distribución del sistema
2. Persistencia de la información.
3. Time-out para jugadores

**1 -) DISTRIBUCIÓN DEL SISTEMA**

Se deberá dividir el sistema en 2 aplicaciones.

- Una aplicación “servidor” donde se implementará la lógica del sistema.
- Una aplicación “cliente/jugador” con la interfaz de usuario para los casos de uso del Jugador.
- Una aplicación “administrador/Consultas” con la interfaz de usuario para los casos de uso de Monitoreo y visualización de partidas.

El caso de uso “Crear una interfaz para un jugador” queda obsoleto en esta versión.

Las aplicaciones deberán inter-conectarse utilizando la tecnología RMI del lenguaje Java.

**2) PERSISTENCIA**

La siguiente información deberá almacenarse en una base de datos MySQL.

- Usuarios.

**3) TIME-OUT PARA JUGADORES**

El jugador tendrá un tiempo límite para realizar seleccionar un casillero y para aceptar una apuesta. Si pasado este tiempo no hay actividad, el sistema asume que el jugador no desea seguir participando y lo retira del juego.

**NOTA: El time-out debe programarse en el servidor, no en el cliente, de forma de mantener la división lógica.**

---

### **Notas**

- Las posibles omisiones, ambigüedades o contradicciones que surjan del estudio de los requerimientos detallados en este documento serán analizadas y corregidas en clase durante el curso.

### **Se pide entregar**

**\*Implementación del sistema en Java con interfaz de usuario grafica cumpliendo con todos los requerimientos funcionales y de diseño solicitados.**

#### **\*Diagramas de Clases:**

-Un diagrama de clases conceptual modelando el dominio del problema.

-Un diagrama de clases de diseño (o mas de uno si lo considera apropiado) que incluya a todas las entidades que participan en la solución.

#### **\*Auto Evaluación:**

Descripción breve de aquellos requerimientos funcionales o de diseño que faltan o no funcionan correctamente y/o pueden ser mejorados y el motivo.

Datos de prueba: Listado impreso con el nombre de usuario, contraseña y saldo de los jugadores y administradores pre-cargados en el sistema.

#### **Datos:**

Respaldo de la base de datos (estructura e información)

### **Entrega**

Se deberán entregar dos copias del sistema (incluyendo el código fuente) y la documentación en formato impreso.

Los diagramas UML solo se corregirán siempre y cuando hayan sido entregados en papel.

Es obligatorio utilizar estricta notación UML en todos los modelos.

Listado impreso con el nombre de usuario y contraseña de los jugadores pre-definidos en el sistema.