

EVALUACION	Obligatorio	GRUPO	TODOS	FECHA	Setiembre 2016
MATERIA	Ingeniería de Software – Taller				
CARRERA	Analista en Tecnologías de Información / Analista Programador				
CONDICIONES	<p>- Puntaje máximo total: 10 - Fecha máxima entrega: 03/11/2016 - Horario de Coordinación adjunta para entregas: <i>de lunes a jueves de 8:00 a 12:30 y de 17:30 a 19:00 hs.</i></p> <p><u>IMPORTANTE</u> - Es un trabajo individual. - Inscribirse (sacar la “<i>boleta de entrega</i>”). - Etiquetar documentación y los dos CDs con nombre, n° estudiante, carrera, grupo, materia y docente</p>				

Completar los siguientes ejercicios cumpliendo con las siguientes pautas:

Sobre el testing de la solución

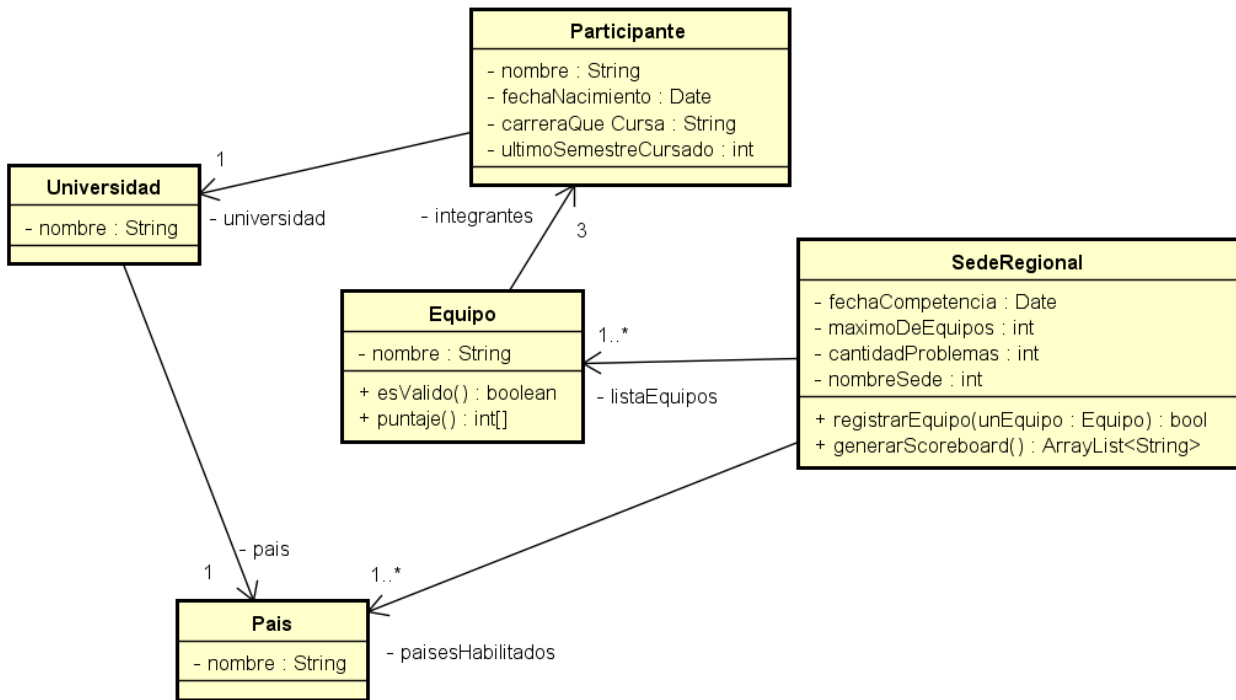
- 1) El paquete de la solución debe estar probado con JUnit.

Sobre la entrega y corrección

- 1) Para cada ejercicio de deberá entregar todo el código fuente (proyecto) en carpetas separadas
- 2) Los trabajos se entregan en bedelía con boleta, una semana luego de la finalización del taller (ver calendario de instancias del semestre).
- 3) De cada uno de los ejercicios se espera:
 - a) Que la implementación cumpla con todas las reglas definidas
 - b) Que cada regla cuente como mínimo con un caso de prueba codificado.

Ejercicio 1 (5 puntos)

Como parte del proceso de diseño de una aplicación para la gestión de un campeonato mundial de programación, se construyó el siguiente modelo.



El campeonato se realiza en dos fases. La primera fase consiste en competencias en Sedes Regionales en donde participan equipos de 3 integrantes.

Cada vez que un equipo quiere registrarse en la sede regional se realiza una validación para determinar si sus integrantes cumplen con las reglas establecidas. Si el proceso de validación es exitoso el equipo queda registrado en la competencia regional.

Las reglas de validación de un equipo y sus integrantes son las siguientes:

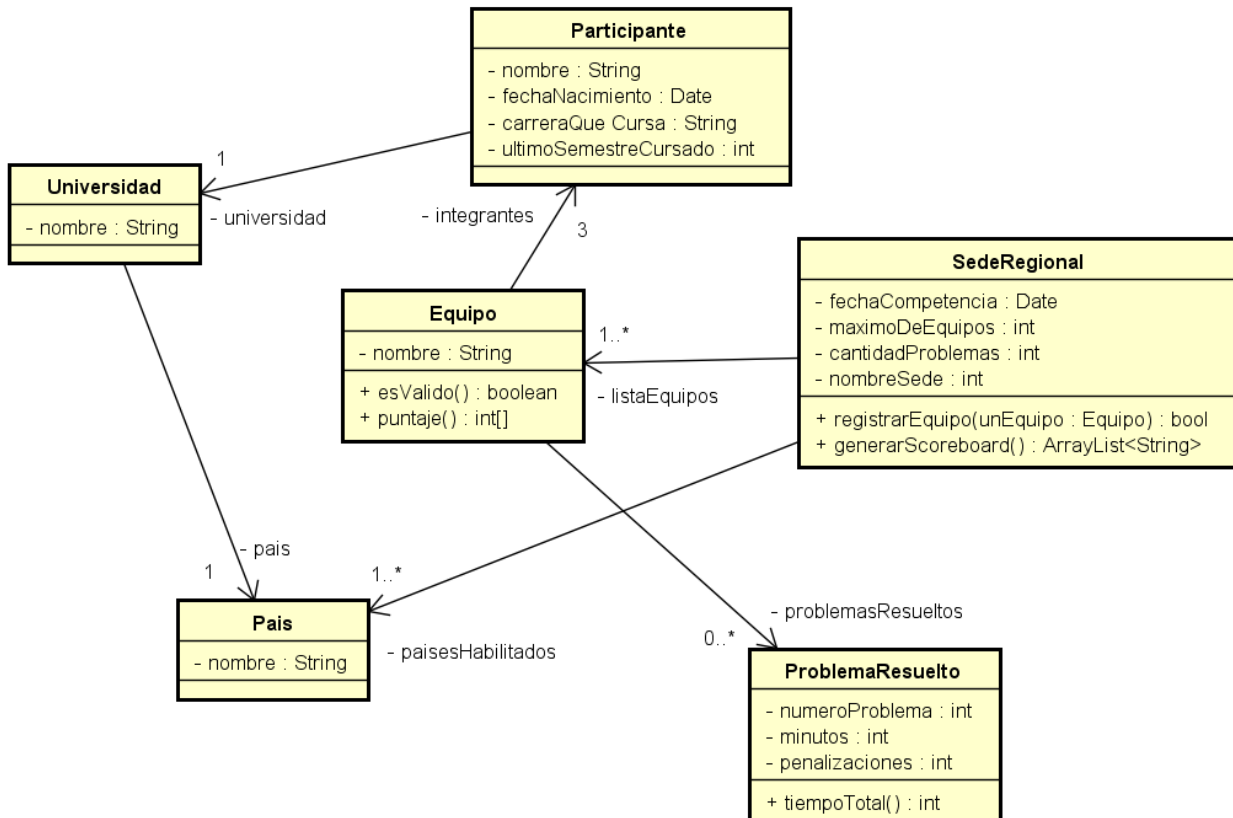
- 1- Se debe generar una excepción con el mensaje "Equipo no está formado por 3 integrantes" si la lista de integrantes no contiene exactamente 3 participantes distintos.
- 2- El nombre de cada participante no puede ser vacío y debe contener al menos 2 palabras de más de 4 caracteres.
- 3- El nombre de la carrera que cursa no puede ser vacío y debe por lo menos tener 5 caracteres
- 4- La edad de cada integrante del equipo debe ser como máximo de 20 años a la fecha de la competencia y cada integrante por lo menos debe tener 16 años de edad.
- 5- Todos los integrantes del equipo deben pertenecer a la misma Universidad.
- 6- En ningún caso los integrantes pueden haber cursado más de ocho semestres de la carrera y al menos deben haber cursado un semestre.
- 7- El nombre del equipo no puede ser vacío y debe tener entre 10 y 20 caracteres (inclusive)
- 8- La Universidad a donde asisten los integrantes del equipo debe ser de un país que esté dentro de los países habilitados a participar en esa Sede Regional.

Se Pide:

- 1) Implementar estas clases en Java. Todos los atributos deben tener métodos de acceso y modificación y constructor con parámetros.
- 2) Escribir el método esValido de la clase Equipo para que cumpla con las reglas descriptas.
- 3) Escribir los casos de prueba que considere necesarios para probar el funcionamiento del método esValido de la clase Equipo con un nivel aceptable de cobertura.
- 4) Escribir el código del método registrarEquipo de la clase SedeRegional. El equipo se puede registrar si es válido y no se ha superado la cantidad máxima de equipos permitidos en esa sede.
- 5) Escribir los casos de prueba que considere necesarios para probar el funcionamiento del método registrarEquipo de la clase SedeRegional con un nivel aceptable de cobertura.

Ejercicio 2 (3 puntos)

Ya en la competencia, se registra cada uno de los ejercicios resueltos correctamente por el equipo. Al momento del registro se almacena el número de problema resuelto, la cantidad de minutos que demoró en resolverlo (siempre se toma el tiempo desde el inicio de la competencia) y la cantidad de penalizaciones que tiene (número de veces que el ejercicio fue entregado previamente con una solución incorrecta). El modelo ajustado es el siguiente:



El tiempo total de un ejercicio resuelto correctamente es la cantidad de minutos más los minutos totales de penalización (se computan 20 minutos por cada penalización).

Se pide:

- 1) Implementar la clase ProblemaResuelto
- 2) Implementar el método tiempoTotal de la clase ProblemaResuelto según la lógica descripta.
- 3) Implementar el método puntaje de la clase Equipo. El método retorna un array de dos elementos en donde en la primera posición se almacena la cantidad de ejercicios resueltos y en la segunda posición el tiempo total (suma de los tiempos de cada ejercicio resuelto correctamente incluyendo los minutos de penalización)
- 4) Escribir los casos de prueba que considere necesarios para probar el funcionamiento del método puntaje de la clase Equipo con un nivel aceptable de cobertura.

Ejercicio 3 (2 puntos)

Durante la competencia la sede cuenta con un scoreboard que se despliega en varias plataformas (web, mobile, desktop). Para su correcto despliegue se genera un ArrayList de Strings, conteniendo cada una de las líneas de texto a desplegar en orden. Para ello, se desea agregar en la clase SedeRegional un método que genere dicho ArrayList, según las siguientes reglas:

- La primera línea debe contener el texto "Scoreboard de la Competencia – Sede XX", donde XX es el nombre de la sede regional.
- La segunda línea debe contener el texto "Fecha y hora de la competencia: dd/MM/yyyy HH:mm" donde dd/MM/yyyy HH:mm representan la fecha y hora de inicio de la competencia
- La tercera línea debe tener el texto "Posiciones"
- Las siguientes n líneas deberán tener la información de cada equipo (uno por línea) ordenada por posición actual en la competencia. Gana el equipo que tenga la mayor cantidad de problemas resueltos. A misma cantidad de problemas resueltos gana quien tenga menor cantidad total de minutos, y si vuelve a empatar gana el equipo se registró antes en la competencia. Cada línea tendrá el formato "<Nombre de Equipo> <Cantidad de Problemas Resueltos> <Tiempo Total>", y se debe generar una línea para cada equipo que esté compitiendo.
- Finalmente, se agrega una línea con el formato "Ultima Actualización: dd/MM/yyyy HH:mm" donde dd/MM/yyyy HH:mm representan la fecha y hora de generado el ArrayList (fecha del sistema).

Se Pide:

- 1) Escribir el método generarScoreboard() en la clase SedeRegional que devuelva un ArrayList de Strings según la especificación dada.
- 2) Escribir los casos de prueba que considere necesarios para probar el funcionamiento de dicho método.