

75.10 – Técnicas de Diseño

Trabajo Práctico – Nikoli Games

Iteración 1

Enunciado

Se deberá implementar un motor de generación de juegos que permita prototipar juegos similares a los de la revista Nikoli (<https://en.wikipedia.org/wiki/Nikoli>).

Estos se caracterizan por ocurrir en una grilla, en la que el jugador analiza la información dada, y mediante conceptos matemáticos/lógicos básicos deduce la solución y la completa en la grilla con formas simples (líneas, números y sombreado de celdas).

Funcionalidad a Implementar Iteración 1

- Leer estructura del juego de un archivo.
- Mostrar el tablero.
- Permitir al usuario ingresar, jugada a jugada, su solución.
- Validar que cada jugada sea compatible con las reglas de los juegos.
- Se deberán implementar dos juegos que validan sus soluciones:
 - Kakuro
 - Sudoku

La idea no es implementar estos juegos en particular, sino diseñar un modelo que permita fácilmente la creación de los mismos y de nuevas variantes a futuro.

Los requerimientos pueden (y van a) cambiar en cualquier momento.

Restricciones

- Trabajo Práctico en grupos de 6 alumnos implementado en java.
- Se deben utilizar las mismas herramientas que en el TP0.
- Todas las clases del sistema deben estar justificadas.
- Todas las clases deben llevar un comentario con las responsabilidades de la misma.
- El uso de herencia debe estar justificado. Se debe explicar claramente el porqué de su conveniencia por sobre otras opciones.
- Se debe tener una cobertura completa del código por tests.

Criterios de Corrección

- Cumplimiento de las restricciones
- Documentación entregada
- Diseño del modelo
- Diseño del código
- Test unitarios
- **No se aceptarán TPs:**
 - **Con warnings.**
 - **Que no compilen en Travis-CI.**
 - **Con issues abiertos.**
 - **Que no se puedan utilizar.**
- **Que el TP no fue aceptado, significa que tiene esa entrega directamente desaprobada.**

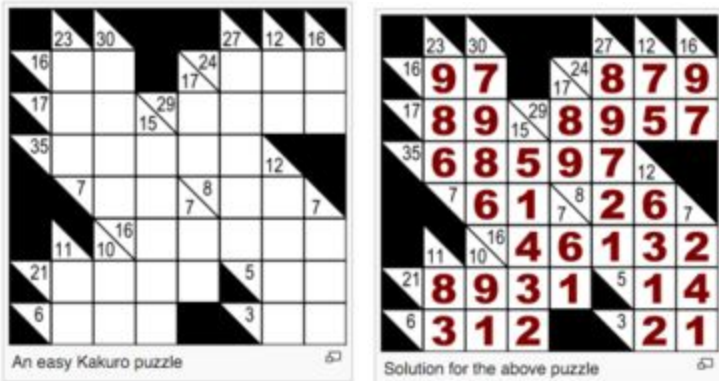
Se tendrán en cuenta también la completitud del tp, la correctitud, distribución de responsabilidades, aplicación y uso de criterios y principios de buen diseño, buen uso del repositorio y uso de buenas prácticas en general.

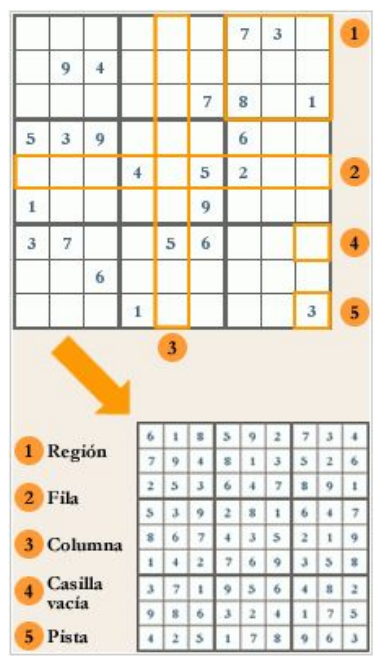
Aclaraciones

- Para las consultas generales se va a utilizar el channel **#tp-grupal-channel**.
- Cada grupo va a tener su channel para consultas particulares y comunicación con su ayudante.
- Se debe usar el mismo toolset que para el TP0 (java 8, IntelliJ, gradle, pmd, cpd, checkstyle, findbugs).
- Se supone que todos están al tanto de las notificaciones en Slack. Esto incluye cambios de alcance, cambios en los requerimientos, restricciones adicionales, etc.
- Se podrán modificar la configuración del toolset sólo bajo aprobación de los ayudantes (#quejas-tooling).
- Cada grupo tendrá un repositorio en GitHub sobre el cual deberá trabajar en grupo.
- Se debe integrar el repositorio con Travis-CI.
- La documentación se realizará en la misma Wiki del repositorio en GitHub.
- El readme.md deberá tener un resumen con una breve reseña del propósito del proyecto y una explicación de como usarlo, y link a la wiki.
- En la fecha de entrega se realizará una demo del TP a uno o dos ayudantes. Esto implica que el TP se debe poder utilizar dentro de las restricciones de la entrega.
- No hay re-entrega. Es responsabilidad del grupo realizar las consultas durante las 2-3 semanas disponibles para realizar el TP.
- Si un alumno no puede concurrir a la demo, **deberá avisar con anticipación** y se coordinará una nueva fecha para que dicho alumno haga la presentación y defensa de la entrega.

- Durante la demo y posterior corrección se cargarán issues en GitHub que deben estar solucionados para la siguiente entrega.

Juegos para esta Iteración

KAKURO	REGLAS
 <p>An easy Kakuro puzzle</p> <p>Solution for the above puzzle</p>	<p>Hay que completar las casillas vacías con números del 1 al 9, teniendo en cuenta que:</p> <p>Cada fila o columna debe sumar lo indicado, respectivamente, a la izquierda o arriba de la misma.</p> <p>No se puede repetir un mismo número en una suma ya sea por fila o columna.</p>

SUDOKU	REGLAS
 <p>1 Región</p> <p>2 Fila</p> <p>3 Columna</p> <p>4 Casilla vacía</p> <p>5 Pista</p>	<p>Hay que completar las casillas vacías con un solo número del 1 al 9, teniendo en cuenta que:</p> <p>En una misma fila no puede haber números repetidos.</p> <p>En una misma columna no puede haber números repetidos.</p> <p>En una misma región no puede haber números repetidos.</p>

Herramientas a utilizar

- Java \geq 1.8
- Gradle \geq 2.6
- JUnit \geq 4.11
- Git -> Bitbucket
- CheckStyle / PMD / Findbugs

Cronograma tentativo (puede ir adaptándose)

8 de Septiembre	Armado de grupos
15 de Septiembre	Publicación 1 ^{ra} Entrega
22 de Septiembre	
29 de Septiembre	Primera Entrega - Publicación 2 ^{da} Entrega
6 de Octubre	Entrega de notas 1 ^{ra} Entrega
13 de Octubre	
20 de Octubre	Segunda Entrega - Publicación 3 ^{ra} Entrega
27 de Octubre	Entrega de notas 2 ^{da} Entrega
3 de Noviembre	
10 de Noviembre	Tercera Entrega
17 de Noviembre	Revision
24 de Noviembre	Publicación de notas finales