



5	3			7			
6			1	9	5		
	9	8				6	
8				6			3
4			8		3		1
7				2			6
	6					2	8
			4	1	9		5
				8		7	9

Paradigmas de Programación (EIF-400) Spec: Sudoku y FP-OOP

DR. CARLOS LORÍA-SÁENZ

LIBERADO: 2 DE AGOSTO 2017

EIF400 /UNA

Introducción

2

- ▶ Especificar los alcances y entregables el proyecto programado sobre Sudoku
- ▶ Definir lo esperado en distintos aspectos
- ▶ Establecer mecanismos de evaluación

Resumen de productos

3

- ▶ P1: Una SPA demostrativa desarrollada y construida usando un stack de desarrollo apropiado a los objetivos del curso y el mercado
- ▶ P2: Un reporte escrito formal sobre un tema computacional interesante en algoritmos combinatorios: SUDOKU

Objetivos Generales

4

- ▶ Poner en práctica técnicas de programación funcional-ooop aplicada en un caso de uso de programación Web
- ▶ Experimentar con herramientas, paradigmas y tendencias acordes con el curso
- ▶ Fomentar la investigación temáticas relacionadas al tema de paradigmas
- ▶ Ayudar al estudiante a ampliar su visión sobre temas modernos en desarrollo de aplicaciones

Problema

- ▶ Desarrollar una aplicación (app) cliente-servidor bajo un modelo SPA (single page application) que usando las herramientas y enfoques que se le piden en esta especificación le permita al usuario:
 - ▶ Generar, en cliente o servidor, instancias de sudokus de distinta complejidad usando algoritmos eficientes
 - ▶ Tratar de resolver un sudoku generado de manera manual o automática a solicitud del usuario mostrando en forma apropiada la solución
 - ▶ Salvar (persistir) el estado de la interacción local o remotamente para uso futuro según el usuario así lo demande

Consideraciones iniciales

6

- ▶ La tarea requiere nociones básicas de programación Web las que el profesor facilitará en los demos, en clase o en consulta extra
- ▶ Se explicará en clase/consulta lo necesario para arrancar asumiendo trabajo en equipo
- ▶ Se estima que con unas 2-3 horas de explicación se puede trabajar en el proyecto

Stack de Desarrollo

Esperado: M-E-(JS | A)-N

► En cliente

- HTML5, CSS3 y JS (ES6). El CSS no está restringido en el problema planteado puede usar cualquier herramienta a propia elección (por ejemplo Bootstrap).
- Alternativamente y valiéndolo extra puede usar Angular 4+ Typescript

► En server

- Node.js >=ES6 (N), Express.js (E), Mongoose/MongoDB (M)
- Alternativamente y valiéndolo extra Typescript (TS)
- Patrones funcionales y Promesas nativas ES6 en ambos lados, generadores/iteradores, asincronía, concurrencia
- Local Storage, Web-workers de HTML5/Node en los casos que corresponda

Requerimientos de organización

- ▶ Debe hacerse en los grupos previamente inscritos según se pidió al inicio del semestre
- ▶ No se acepta de otra forma
- ▶ No se aceptan nuevos grupos después del 18-agosto. SIN EXCEPCIÓN.

Requerimientos diseño e implementación

- ▶ Eliminar lo más que se pueda todo patrón imperativo y estado mutable del JS usando FP en el cliente y el server. Eliminar callbacks por medio de promesas
- ▶ Usar arquitecturas estándar en cliente y server.
- ▶ En concordancia con SPA la interacción cliente/server bidireccional sea solo vía AJAX/JSON salvo en la página inicial (index.html) contra un API Rest
- ▶ Estilo Rest (sin view en server excepto el inicial index.html)
- ▶ Maximizar el uso de estilo ES6/TS en lugares que lo permitan

Evaluación y Valor App

10

- ▶ Demo en clase: 60% (unos 15 minutos por grupo)
- ▶ Mejor traerlo en su propia laptop por si acaso haya problemas de setup en el lab. Si no logran la demo por razones ajenas al profesor reciben cero
- ▶ Revisión de código 40%. Se revisará que se emplean técnicas FP-OOP y herramientas según lo solicitado
- ▶ Este proyecto vale al menos un 50% del porcentaje dedicado a proyectos (revisable a favor del estudiante, eventualmente)
- ▶ Se puede otorgar hasta un 30% extra sobre la nota si se añaden más funcionalidades o herramientas (previamente discutidas con el profesor). Sólo aplica si la aplicación tiene lo mínimo obligatorio

SCM y Construcción y publicación

11

- ▶ Se espera que la aplicación pueda ser construida y corrida usando una herramienta apropiada desde una consola
- ▶ En todo caso: La construcción y corrida no deben depender de un IDE.
- ▶ Opcional y muy preferiblemente usar un repositorio de manejo de fuentes versionados (SCM) como Git/Github o comparable
- ▶ Opcionalmente si la aplicación corre sobre un PaaS como [Heroku](#).

Evaluación y valor reporte

12

- ▶ Junto con la app se debe entregar un reporte escrito sobre el tema de algoritmos para generar y resolver sudokus según se ocupa en la app
- ▶ El reporte debe cumplir con una estructura formal: portada, resumen, introducción, descripción de algoritmos usados, conclusiones, referencias. Usar un template estándar como IEEE. Si usa LaTeX suma extras.
- ▶ Se espera máximo 12 páginas.
- ▶ Tiene el valor dentro del 10% del rubro “investigación” del curso (ver carta al estudiante)
- ▶ Se califica calidad subjetivamente según apreciación del profesor.

Entregables

13

- ▶ El proyecto en un `.rar` (o equivalente) llamado `sudokus_AAA`, donde AAA son los autores. Se enviará por correo al profesor. El Subject siendo “`sudokus_AAA`” (AAA como antes) y en el correo de nuevo vienen los autores y grupos
- ▶ Dentro del `rar` viene un directorio (carpeta) `sudokus` (con el proyecto) y un `README.txt` con los nombres de los autores, su horario.
- ▶ Incumplimiento de estos requisitos anula la revisión que hará el profesor (pierden su 60%). Sin derecho a reclamo.
- ▶ Un reporte formal escrito sobre “*Algoritmos para generación y solución eficiente de Sudokus*” (pdf)

Fecha y hora de entrega

14

- ▶ El día exacto de entrega se da a conocer oportunamente en clase y por los medios usuales en el curso: esperado semana 10 (25-28 setiembre)
- ▶ Ese día de entrega (mismo de la demo) se envía el entregable por el medio requerido y se entrega el reporte escrito en la hora asignada. Esta dependa de la constitución del grupo: normalmente la hora de matrícula de la mayoría de miembros
- ▶ Entregas posteriores e injustificadas podrán recibir pérdida de puntos de hasta 5 puntos por minuto de anticipo o atraso hasta llegar a cero.
- ▶ Es obligación entregar el reporte por escrito. Sino no se recibe el proyecto.