

INFORME

Simulador de contagios

Presentado a:

Ing. Martín M. Miceli

Ing. Martin F. Bustos Menas

Presentado por:

Rueda Horacio

Severini Montanari Alejo

Vega Cuevas Silvia Jimena

Wortley Agustina Daniela

Dia de presentacion:

25 de Junio, 2020

INDICE

1. Dedicación de esfuerzo
2. Errores cometidos
3. Lecciones aprendidas
4. Bibliografía

1. DEDICACION DE ESFUERZO

Para la realización de los Trabajo Prácticos propuestos se comenzó por la creación de un grupo de whatsapp entre los miembros del grupo.

Cada integrante creó una cuenta en la plataforma Discord, y se creó un canal de conversación, esta herramienta está destinada a crear grupos de chat para diferentes finalidades, y cuenta con diferentes funcionalidades, permitiendo chatear por texto, por voz y realizar videoconferencias.

Además se utilizó Word online, como editor de texto, que nos permitió trabajar en conjunto en tiempo real

En la elaboración del Trabajo Practico 1 y 2 se realizaron todas las tareas en conjunto, por medio de las plataformas ya mencionadas

Se estima que el tiempo en horas, para la realización del **Trabajo practico 1** fue en total de aproximadamente **18 horas** en total.

Invirtiéndose unas 8 horas para el Plan de Configuración y 7 horas para el Documento de Requerimientos. Para la Matriz de trazabilidad y casos de pruebas unas 4 horas más.

Se estima que el tiempo en horas, para la realización del **Trabajo practico 2** fue en total de aproximadamente **15 horas** en total.

Invirtiéndose unas 6 horas para el Documento de Arquitectura y 9 horas para el Documento de Diseño, incluyendo allí, la realización de Test Unitarios.

Para la **Entrega Final** y realización del software si se dividieron las tareas, se coordinaron los horarios de trabajo para tener la posibilidad de agotar cualquier duda en ese mismo momento a través de una puesta en común en conferencia con el objetivo de poder avanzar más rápido.

Las tareas fueron realizadas según la siguiente división:

Horacio Rueda:

- Implementación de código tests de sistema – 12 horas
- Tests de sistema – 8 horas

Severini Montanari Alejo

- Implementación de código – 10 horas
- Implementación de patrones de diseño en código – 4 horas
- Verificación de pruebas realizadas – 2 horas

Vega Cuevas Silvia Jimena

- Implementación de código – 10 horas
- Implementación de patrones de diseño en código – 4 horas
- Verificación de pruebas realizadas – 2 horas

Wortley Agustina

- Implementación de código – 2 horas
- Implementacion de patrones de diseño en codigo – 4 horas
- Informe – 2 horas
- Notas de Entrega – 2 horas
- Readme – 1 hora
- Presentacion – 3 horas

2 ERRORES COMETIDOS

- No se pensó desde un principio como incorporar correctamente a nuestro software la utilización de los patrones de diseño, lo que generó que al realizar las implementaciones del Strategy y del Observer se genere una incertidumbre muy grande, y trabas para poder continuar con proyecto
- No todos los integrantes del grupo conocíamos la implementación de interfaces gráficas en Java lo que generó una brecha de conocimientos y teniendo que dividir las tareas acorde a las capacidades.
- La entrega del Trabajo Practico 2 se realizó a último momento, y se subestimó el tiempo que se necesitaba para realizar los diagramas de secuencias y actividades requeridos para el Documento de Diseño, teniendo que descuidar algunos detalles.
- La codificación del software llevó mucho más tiempo del estimado, ya que se requerían gráficos y otros elementos correspondientes a la programación orientada a objetos que no eran del todo conocidos.

3 LECCIONES APRENDIDAS

- Utilización de Github como una potente plataforma de desarrollo colaborativo, y todas las herramientas que incorpora
- La importancia del control de versiones, y de las herramientas que se utilizan para ello
- Necesidad de realizar integración continua, para ahorrar tiempo y esfuerzo
- Aplicación de Metodologías ágiles y sus prácticas en la producción de nuestro software
- Importancia del uso del patrón de arquitectura MVC (Model-View-Controller), para encapsular módulos y favorecer a la hora de hacer correcciones en métodos y clases
- Implementación de patrones de diseño Observer y Strategy para la solución de problemas comunes en el desarrollo de software.
- Realización de diagramas UML y como expresar a través de ellos nuestro software
- Redacción de distintos documentos de Ingeniería de Software, y la importancia que tienen estos para creación de software confiable y de calidad, basándose en las diferentes metodologías y técnicas aprendidas