

ÍNDICE

Índice	1
I Clase 1: Presentation	2
1. Objetivos	2
2. Notas importantes	2
3. Código fuente del software	2
3.1. Qué implica un producto de software	3
4. Puntos importantes a tener en cuenta	3
5. Requisitos de aprobación	3
6. Aspectos sobre las tareas	3
7. Espacio de diálogo	3
8. La agenda del curso	4
9. Algunas palabras sobre ética	4
9.1. StackOverflow	4
9.2. El feedback	4
10. Material	4
11. Productividad	4
Referencias	5



CLASE 1: PRESENTATION

Clase basada en notas de la docente Nancy Hitschfeld K. en colaboración con material del mismo curso dictado por Alexandre Bergel en semestres anteriores. [1]. Los apuntes del curso tienen lugar en el semestre de Primavera 2022 bajo las cátedras del docente Matias Toro.

1 Objetivos

- Adquirir comodidad con la [programación orientada a objetos](#); Aprender a [pensar con objetos](#)
- Aprender el lenguaje de programación *Java*: Su sintaxis, idiomas, patrones y estilos.
- Aprender los [elementos esenciales de la librería de clases java](#) y **aprender cómo aprender** acerca de otras secciones de la librería cuando se necesiten.

Debemos recordar que este curso no es un curso de cómo aprender a programar ni cómo hacer software. Este curso es un curso [introdutorio](#) de Metodologías de programación.

2 Notas importantes

Tenemos que saber que al final de este semestre **no** nos habremos convertido en ingenieros de software experimentados o siquiera un buen programador (años de práctica y cursos complementarios son necesarios).

Pero al fin de este semestre usted, probablemente, podrá realizar tareas pequeñas (en tiempo, esfuerzo y recursos) y usted no debería sentir vergüenza del código que usted escribe.

3 Código fuente del software

Para desarrollar un software, se requiere naturalmente desarrollar aspectos funcionales (distinguir método de función).

Functional aspects: What you have learnt so far (e.g., writing functions, algorithms, structure ideas)

Making software extendable, maintainable: What this ramo CC3002 is all about (e.g., writing well engineered code, unit testing, methodology, design pattern)

3.1 Qué implica un producto de software

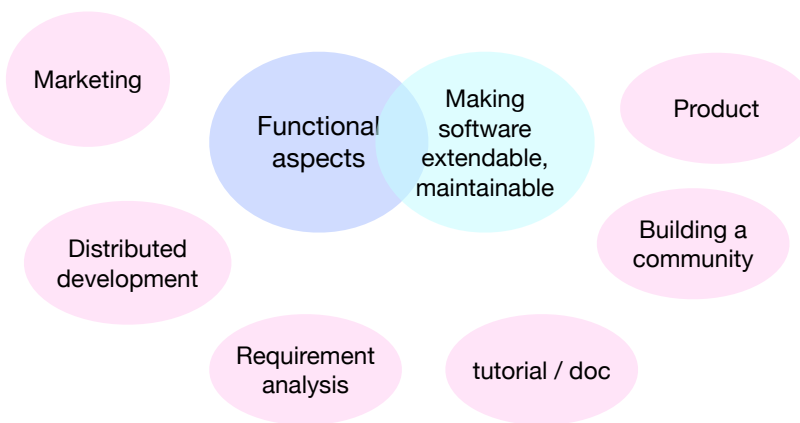


Figura 1.1: Diversas ramas de qué implica un producto de software

4 Puntos importantes a tener en cuenta

No es suficiente escribir código que solo funciona

Es igual o incluso más importante leer y escribir código bien; no solamente código que funciona, sino que necesitamos código legible, mantenible, reutilizable, rápido y eficiente.

La práctica es la clave

- Solo leer diapositivas y libros no es suficiente
- Es como querer aprender buen Francés solo de un libro (No óptimo)

5 Requisitos de aprobación

Disponible en [Material docente](#)

6 Aspectos sobre las tareas

Es esencial la discusión sobre las tareas con el equipo auxiliar. Se ha visto una gran cantidad de estudiantes que pierden puntos porque no hablan sobre su/s tarea/s.

7 Espacio de diálogo

Foro de **U-Cursos**

- Todas las preguntas relacionadas a la cátedra, programación, ingeniería de software y tareas son bienvenidas
- Se debe pensar dos veces antes de publicar en el foro algún post que pueda ser malinterpretado
- Los post que se malinterpretan pueden ser devastadores (se puede herir gente)

No postear en el foro:

- Antes de almuerzo
- Después de mucho trabajo en una tarea
- Si queda poco tiempo para hacer la tarea
- En medio de una batalla feroz con Git

8 La agenda del curso

- 1) Objetos, clases, tipos y polimorfismos.
- 2) Testeo y desarrollo iterativo de una aplicación simple
- 3) Metodologías y patrones de diseño de software
- 4) Ética en el desarrollo de software
- 5) Tópicos avanzados de Java de programación

9 Algunas palabras sobre ética

- Se deben respetar las fechas
- No se debe tomar autoría de código no propio

9.1 StackOverflow

Es un sitio de preguntas y respuestas para programadores; Se permite hacer preguntas en el sitio y buscar respuestas.

9.2 El feedback

- La retroalimentación (feedback) es una poderosa herramienta pedagógica
- Contribuye al logro de aprendizajes
- Los ejercicios serán acompañados con una clara solución
- Se entregará tan pronto como sea posible y en anticipación a la entrega de una siguiente tarea
- Los resultados de su tarea serán complementados con feedback

10 Material

Todo el software y librerías que se usen este semestre serán de código abierto (no se requiere pago alguno). Sin embargo, **no existe** un solo libro que cubra todo el material del curso.

11 Productividad

Se recomienda fuertemente la organización de calendarios en aspectos de planificación a largo plazo tanto como organización a corto plazo (e.g Pomodoro)

Es esencial que usted piense con cuidado sobre su productividad.

REFERENCIAS

- [1] Hitschfeld, Nancy y Alexandre Bergel: *01-Presentation.pdf*. https://www.u-cursos.cl/ingenieria/2022/1/CC3002/1/material_docente/, 2022.