Tema 1 – reglas de estilo de código

Para comprender mejor las reglas de estilo de código, el estudiante preparará una presentación de 10 minutos acerca de tres reglas de estilo código con las que esté de acuerdo y tres con las que no esté de acuerdo, y dará una justificación. Además, presentará una herramienta que verifique estas una regla de estilo aplicada a una tarea que haya realizado previamente.

Tema 2 – Análisis estático de código

Para comprender mejor las capacidades y los beneficios de las herramientas de análisis estático, el estudiante seleccionará una herramienta de análisis estático bien establecida (por ejemplo, sonarqube) con documentación en línea y preparará una presentación de 10 minutos sobre dos características importantes de la herramienta y como la aplicado a una tarea que haya realizado previamente.

Tema 3 – Comentarios y Documentación

Para poner en práctica las mejores prácticas sobre agregar comentarios al código y generar documentación automáticamente a partir del código (algo similar a Doxigen), el estudiante agregará comentarios a un código creado en alguna tarea y utilizará una herramienta para generar documentación automática. Además, preparará una presentación de 10 minutos explicando lo realizado.

Tema 4 – Versionamiento de Código

Para comprender mejor las reglas y beneficios del Versionamiento de código, el estudiante preparará una presentación de 10 minutos explicando dos modelos de Versionamiento, mostrando ejemplo y contrastando ventajas y desventajas entre ellos (Por ejemplo, revise en detalle el artículo de Martin Fowler que aborda esta temática).

Tema 5 – Build

Para comprender mejor como generar un “build” del código, el estudiante automatizará el build de una tarea de programación realizada previamente mediante github actions (por ejemplo, para crear una imagen de Docker de su aplicación). Además, preparará una presentación de 10 minutos explicando lo realizado.

Presentación: