

Trabajo Práctico

Estructuras de Datos

Universidad Nacional de Quilmes

Segundo cuatrimestre 2019

El TP consta de 4 etapas, las cuales se corresponden con la implementación de distintas estructuras vistas en la materia: un registro, un set, un map, y una heap.

Cada etapa consiste en escribir partes del código que se encuentran incompletas. Es decir, sólo tenés que completar las partes que explícitamente digan **COMPLETAR**. Podés crear subtarefas, pero no podés modificar ninguna de las estructuras e interfaces dadas. Tampoco hace falta agregar ningún archivo.

La entrega consistirá en subir estos cambios a los repositorios creados para cada uno, semana a semana.

Para chequear la calidad del código, se implementaron un test automatizados que verifican que la implementación cumple con algunas propiedades deseadas.

Si las entregas se realizan en tiempo (antes de la fecha límite de finalización del mismo) y forma (pasan los test), el TP estará aprobado. La fecha final de entrega del TP es el **5 de diciembre**.

Forma de evaluación

Evaluaremos fundamentalmente los siguientes aspectos:

1. Autoría del trabajo. En clase tendrás que ser capaz de demostrar que entendés tu trabajo. Está bien buscar ideas en internet o consultar con tus compañeros, pero tenés que *demostrar* que entendés lo que estás entregando. Si no podemos comprobar durante la clase que el trabajo que estás entregando es de tu propiedad, te tomaremos un ejercicio escrito adicional, trivial, para demostrar que entendiste de qué trata tu código. En dicho caso, la aprobación del TP estará sujeta a la aprobación de ese ejercicio.
2. El proyecto debe compilar correctamente. Si esto no sucede, el trabajo se considera desaprobado. Consejo: **compilá todo el tiempo tu código**. No avances con otras cuestiones hasta que no hayas logrado esto, y pedí ayuda en lo que respecta al lenguaje y las herramientas que usamos.
3. La implementación correcta de la funcionalidad pedida. Recordá que podés pedir ayuda tanto como sea necesario. Pero dicho esto, la implementación debe **ejecutar correctamente** y **debe pasar todos los tests** provistos para cada etapa.
4. Buenas prácticas. Que tu implementación compile y funcione correctamente no es garantía de que hayas utilizado los conceptos vistos en la materia. Es muy importante que, por ejemplo, respetes barreras de abstracción, indiques los invariantes de representación, precondiciones, etc. Un trabajo con graves faltas en este sentido no estará aprobado con buena nota, e incluso puede llegar a estar desaprobado, si es que no cumple con los lineamientos de la materia.