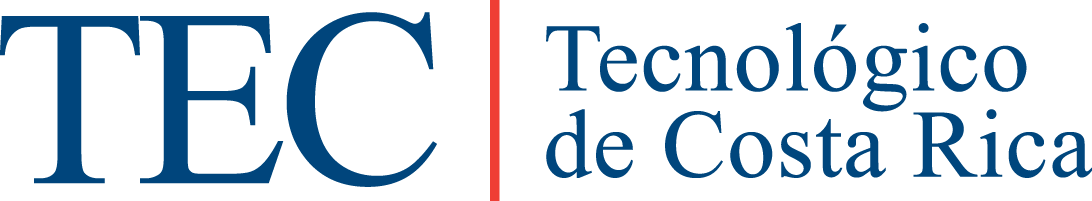
**Data Structures & Algorithms Library**

Instituto Tecnológico de Costa Rica

Área de Ingeniería en Computadores

Algoritmos y Estructuras de Datos I (CE 1103)

Segundo Semestre 2017

Valor 10%

**Objetivo General**

* Implementar las estructuras de datos y algoritmos vistos en clase.

**Objetivos Específicos**

* Implementar estructuras de datos lineales y las operaciones principales
* Implementar estructuras de datos jerárquicas y las operaciones principales
* Implementar estructuras de datos generales y las operaciones principales
* Implementar algoritmos de búsqueda
* Implementar algoritmos de ordenamiento
* Implementar benchmark entre los algoritmos de búsqueda y ordenamiento

**Descripción del Problema**

* Se deberá implementar una biblioteca reutilizable y genérica que contenga las estructuras de datos vistas en clases y algoritmos de búsqueda y ordenamiento. Se deben utilizar *generics* para que las estructuras soporten cualquier tipo de dato.
* En Java se deberán crear tres jars llamados org.tec.datastructures, org.tec.algo.sort y org.tec.algo.search.
* En C# se deberán crear tres DLLs llamados org.tec.datastructures, org.tec.algo.sort y org.tec.algo.search.
* Deberá implementar en C# .NET y Java las siguientes estructuras de datos lineales junto con las operaciones más comunes:
  + Listas simples
  + Listas dobles
  + Listas circulares
  + Colas
  + Pilas
* Deberá implementar en C# .NET y Java las siguientes estructuras de datos jerárquicas junto con las operaciones más comunes:
  + Árboles binarios
  + Árboles B
  + Árboles Splay
  + Árboles AVL
* Deberá implementar en C# .NET y Java las estructuras de datos generales (grafos) y los algoritmos más comunes.
* Implementar en C# .NET y Java los algoritmos de búsqueda y ordenamiento vistos en clase.
* Implementar en C# .NET y Java un benchmark de manera que puedan probarse todos los algoritmos de búsqueda o todos los algoritmos de ordenamiento sobre un mismo conjunto de elementos y generar gráficos que muestren el rendimiento de cada uno de los algoritmos sobre ese conjunto de elementos. Para esto se deben generar casos de prueba sobre conjuntos de elementos de diferente tamaño y en diferentes situaciones de orden de sus elementos.
* Cada uno de los algoritmos solicitados, tiene un “comportamiento teórico” según lo visto en análisis de algoritmos. Junto a la gráfica del benchmark, sobreponga la gráfica del comportamiento teórico para probar que la teoría refleja o no la realidad del algoritmo.

**Entregables**

1. Jars y DLLs indicados en la sección anterior
2. Pruebas unitarias automáticas de cada operación y algoritmo implementado. Se deberá cubrir al menos el 80% del código con dichas pruebas.
3. Documento con las evidencias de casos de prueba para el benchmark de los algoritmos de ordenamiento y búsqueda e incluir conclusiones obtenidos a partir de este benchmark. Por ejemplo cuando se debe usar uno u otro algoritmo, cuando no se debe usar, y otros.

**Aspectos operativos y evaluación**:

1. **La entrega es según cronograma.**
2. El trabajo se realizará en parejas o individualmente.
3. Cualquier indicio de copia será calificado con una nota de 0 y será procesado de acuerdo al reglamento.