



Universidad Politécnica
de Madrid

**Escuela Técnica Superior de
Ingenieros Informáticos**



Grado en Grado en Ingeniería Informática

Trabajo Fin de Grado

Conjuntos Indexados como Alternativa al Array: Implementación y Evaluación.

Autor: Carlos Gaspar Pozo Serrano
Tutor(a): Santiago Tapia Fernández

Madrid, 03-2023

1 Descripción general del trabajo

Aunque la mayor parte de las estructuras clásicas de programación son de sobra conocidas, los estudios sobre su utilidad y eficiencia se establecieron con equipos informáticos cuyas características están muy lejos de las actuales. En ese sentido pueden ser especialmente interesantes los casos en los que se pueda intercambiar rendimiento por mayor uso de memoria.

El array es una estructura básica de programación de uso frecuente, sus características más notables respecto del rendimiento son que usa eficazmente la memoria (los elementos son contiguos) y que el tiempo de acceso a un elemento mediante el índice tiene complejidad $O(1)$. Mientras que en su debilidad está que las inserciones o eliminaciones de elementos en índices arbitrarios son $O(n)$.

El objetivo es estudiar implementaciones alternativas al Array, especialmente si el problema admite que la estructura de datos a utilizar tenga una interfaz que se corresponda con un conjunto indexado.

Los objetivos a conseguir son:

1. Analizar las interfaces genéricas de un array (por su uso, se tomará como referencia la `list` de python) y la de un conjunto indexado.
2. Diseñar e implementar alternativas al array que no usen almacenamiento contiguo de todos sus elementos con interfaz tipo conjunto indexado.
3. Diseñar e implementar un conjunto de pruebas automáticas para evaluar la validez y rendimiento de las estructuras de datos diseñadas.
4. Analizar el rendimiento de estructuras de datos alternativas a los arrays cuya interfaz se corresponde con un conjunto indexado.

2 Lista de tareas

1. Formación previa y preparación del entorno del trabajo.
2. Análisis del problema y planteamiento de alternativas.
3. Diseño y arquitectura del software.
4. Diseño e Implementación de las estructuras de datos.
5. Validación, evaluación y comparación de rendimientos.
6. Generación de documentación.
7. Escritura de la memoria.

3 Diagrama de Gantt

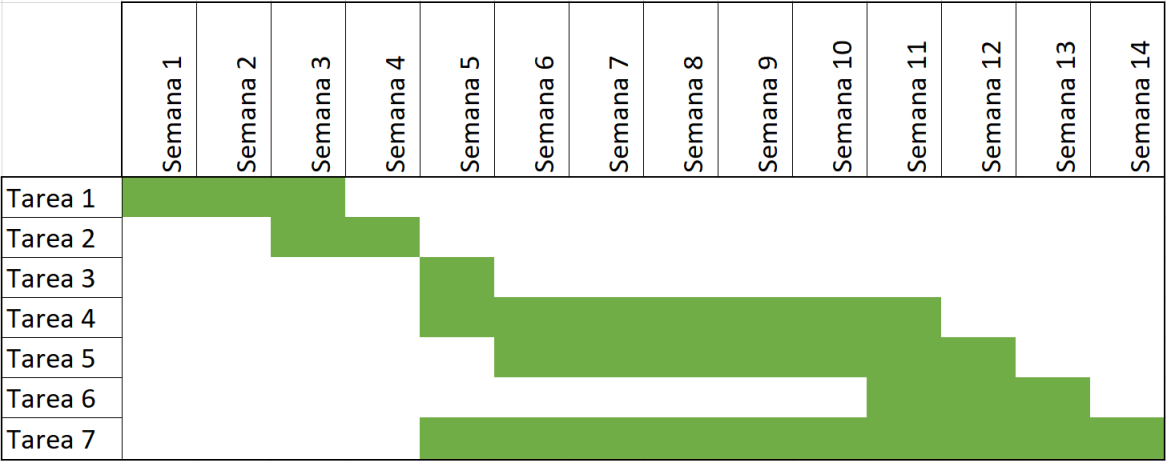


Figure 3.1: Diagrama de Gantt

4 Copia de la propuesta del tutor

Planes para los que la propuesta es válida:	10II, 10ID/II
ID de la oferta	6647
Título del trabajo	Conjuntos Indexados como Alternativa al Array: Implementación y Evaluación.
Tipo de trabajo	
Resumen general del trabajo	<p>Aunque la mayor parte de las estructuras clásicas de programación son de sobra conocidas, los estudios sobre su utilidad y eficiencia se establecieron con equipos informáticos cuyas características están muy lejos de las actuales. En ese sentido pueden ser especialmente interesantes los casos en los que se pueda intercambiar rendimiento por mayor uso de memoria.</p> <p>El array es una estructura básica de programación de uso frecuente, sus características más notables respecto del rendimiento son que usa eficazmente la memoria (los elementos son contiguos) y que el tiempo de acceso a un elemento mediante el índice tiene complejidad $O(1)$. Mientras que en su debe está que las inserciones o eliminaciones de elementos en índices arbitrarios son $O(n)$.</p> <p>El objetivo es estudiar implementaciones alternativas al Array, especialmente si el problema admite que la estructura de datos a utilizar tenga una interfaz que se corresponda con un conjunto indexado.</p>
Lista de objetivos concretos	<ul style="list-style-type: none"> * Analizar las interfaces genéricas de un array (por su uso, se tomará como referencia la list de python) y la de un conjunto indexado. * Diseñar e implementar alternativas al array que no usen almacenamiento contiguo de todos sus elementos con interfaz tipo conjunto indexado. * Diseñar e implementar un conjunto de pruebas automáticas para evaluar la validez y rendimiento de las estructuras de datos diseñadas. * Analizar el rendimiento de estructuras de datos alternativas a los arrays cuya interfaz se corresponde con un conjunto indexado.
Desglose de la dedicación total del trabajo en horas (324 horas en los Grados)	<ul style="list-style-type: none"> * Formación previa y preparación del entorno del trabajo (25 h). * Análisis del problema y planteamiento de alternativas (20 h). * Diseño y arquitectura del software (5 h). * Diseño e Implementación de las estructuras de datos (100 h) * Validación, evaluación y comparación de rendimientos (130 h). * Generación de documentación (30 h). * Escritura de la memoria (14 h).
Conocimientos previos recomendados para hacer el trabajo	<p>Es imprescindible motivación y ganas de aprender.</p> <p>Es imprescindible que el candidato tenga interés en hacer trabajo de programación (en sentido amplio, desde el diseño hasta la implementación). Y capacidad para ello.</p> <p>Es imprescindible que el candidato tenga competencia alta en C. Es conveniente que tenga nociones de python ó C++.</p>

Figure 4.1: Propuesta de trabajo del tutor