

---

# Trabajo obligatorio no presencial: Una solución distribuida al problema del viajante mediante algoritmos genéticos

---

Programación de Sistemas Concurrentes y Distribuidos  
Grado de Ingeniería Informática  
Escuela de Ingeniería y Arquitectura  
Universidad de Zaragoza

26 de noviembre de 2019

## 1. Objetivos

Los objetivos de este trabajo no presencial son los siguientes:

- Profundizar en los modelos de comunicación síncrona/asíncrona para procesos distribuidos.
- Asentar los conocimientos adquiridos para la programación de aplicaciones concurrentes y distribuidas en C++.
- Desarrollar una aplicación distribuida cliente/servidor.
- Trabajar en equipo.

## 2. Descripción del problema

El *problema del viajante de comercio* (TSP en sus siglas en inglés) es un caso prototípico de problema de optimización combinatoria. El objetivo es encontrar el camino más corto que debe recorrer un viajante para, partiendo de una ciudad dada, visitar un conjunto de ciudades intermedias y volver a la ciudad de partida, sin pasar dos veces por la misma. El TSP tiene numerosas aplicaciones en dominios diferentes (planificación de rutas en flotas, secuenciación de genomas, fabricación de circuitos electrónicos, ...).

Se trata de un problema duro desde el punto de vista del coste computacional, para el que se han ofrecido diferentes soluciones, tanto exactas como aproximadas (capaces de encontrar aproximadas a la solución óptima en tiempos razonables).

Uno de los posibles enfoques se basa en el uso de *algoritmos genéticos* (GA en sus siglas en inglés). Los GA pertenecen a la familia de los algoritmos *bio-inspirados*: algoritmos que buscan soluciones tratando de imitar el comportamiento de seres vivos. Los genéticos se basan en el concepto de

selección natural. La idea es bastante sencilla. Se parte de una población inicial de soluciones. El algoritmo, de manera iterativa, va a calcular la siguiente generación de candidatos mediante operaciones que seleccionan, combinan y mutan las mejores soluciones de la presente generación, y así sucesivamente, hasta que se llegue a una solución aceptable o se agote el número preestablecido de generaciones.

En las siguientes referencias podéis encontrar información detallada sobre los algoritmos genéticos.

- <http://www.aic.uniovi.es/ssii/tutorial/Introduccion.htm>
- <http://sabia.tic.udc.es/mgestal/cv/aaggtutorial/aagg.html>
- <http://oaji.net/articles/2014/489-1393931306.pdf>

### 3. Descripción del trabajo a realizar

En es trabajo debéis llevar a cabo una implementación distribuida de un algoritmo genético para resolver el TSP, siguiendo uno de los esquemas que se proponen más adelante en esta sección.

En la dirección siguiente <https://people.sc.fsu.edu/~jburkardt/datasets/cities/cities.html> hay un buen número de datos de prueba para el TSP. Deberéis trabaja con los datos *UK12* y *USCA312* para probar con problemas de diferentes magnitudes.

### 4. Esquemas arquitecturales para el diseño de la solución

En esta sección se presentan tres esquemas arquitecturales posibles para la implementación de la solución. Cada equipo deberá elegir uno de estos esquemas y, posteriormente, diseñar su sistema final en base a sus pautas de alto nivel.

#### 4.1. Arquitectura para un sistema concurrente

El primer esquema corresponde con un sistema concurrente, compuesto por un conjunto de procesos que se ejecutan en un único recurso de cómputo. La figura 1 muestra los procesos involucrados en la solución. Un proceso *GA control* implementa las tareas iniciales (lectura de parámetros de entrada y creación y evaluación de la población inicial) y el control del algoritmo genético (esquema de iteración y evaluación de la convergencia). En cada sucesiva generación delega en un conjunto de procesos especializados las operaciones específicas de selección, cruce y mutación. La información relativa a la población actual se encapsula en un monitor para gestionar correctamente el acceso concurrente a los datos por parte de los procesos.

Además, hay un proceso *estadístico* especializado en recopilar información sobre la ejecución del algoritmo y su rendimiento. Estos datos son guardados en un monitor para favorecer que procesos *informadores* puedan ir ofreciendo *feedback* continuo sobre lo que está sucediendo durante la ejecución del sistema.

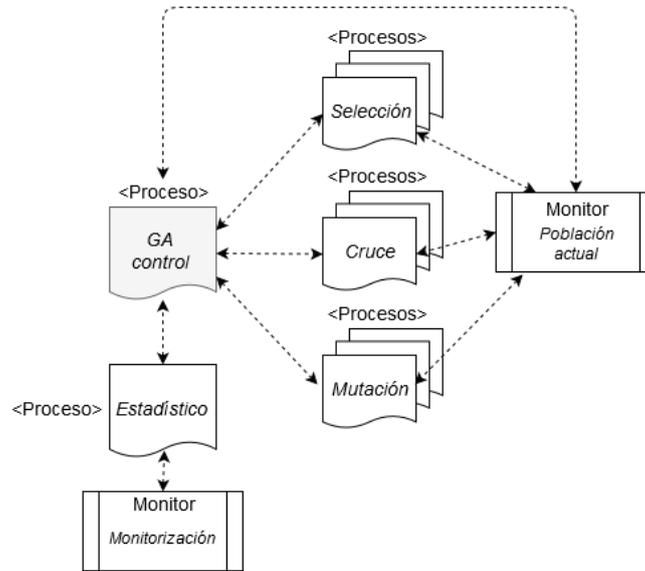


Figura 1: Esquema concurrente

## 4.2. Arquitectura para un sistema distribuido básico

El segundo esquema propone una arquitectura distribuida básica. Como se muestra en la figura 2, el sistema consta de dos nodos de cómputo. En uno de los nodos se ejecutará el proceso *GA control* que implementa el núcleo del algoritmo genético y el proceso *estadístico* que monitoriza su ejecución. En el otro nodo se ejecutará un servidor que ofrece las tres operaciones específicas que requiere un genético: la selección, el cruce y la mutación de una población. Cada vez que el proceso de control requiera evaluar una de estas operaciones, invocará al servidor remoto indicándole en la petición la operación a ejecutar y la población de entrada. Internamente, el comportamiento del servidor será el siguiente: almacenará la población de entrada en un monitor local ( $P_{IN}$ ), creará un conjunto de procesos especializados en la operación solicitada, esperará a que estos evalúen la población, y devolverá la población resultante ( $P_{OUT}$ ) como resultado al proceso de control.

La comunicación entre el proceso de control y el servidor será de naturaleza síncrona, utilizando las librerías proporcionadas durante las prácticas en el laboratorio.

## 4.3. Arquitectura para un sistema distribuido

El último esquema presenta una arquitectura distribuida que consta de un conjunto de servidores de operaciones, conforme se representa en la figura 3. Por simplicidad, podéis suponer que hay 3 servidores de operaciones. La ventaja de este esquema con respecto al anterior es que permite paralelizar la ejecución de las operaciones de selección, cruce y mutación.

Cuando el proceso de control tenga que aplicar una de estas tres operaciones sobre la población actual, se comportará de la siguiente manera: dividirá la

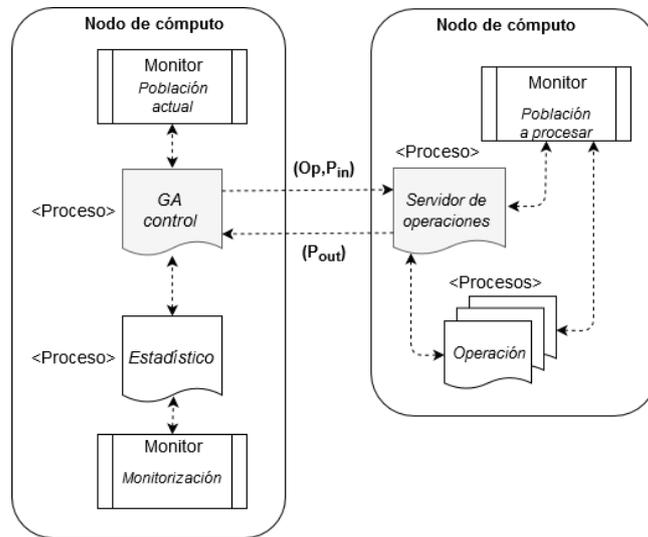


Figura 2: Esquema cliente-servidor

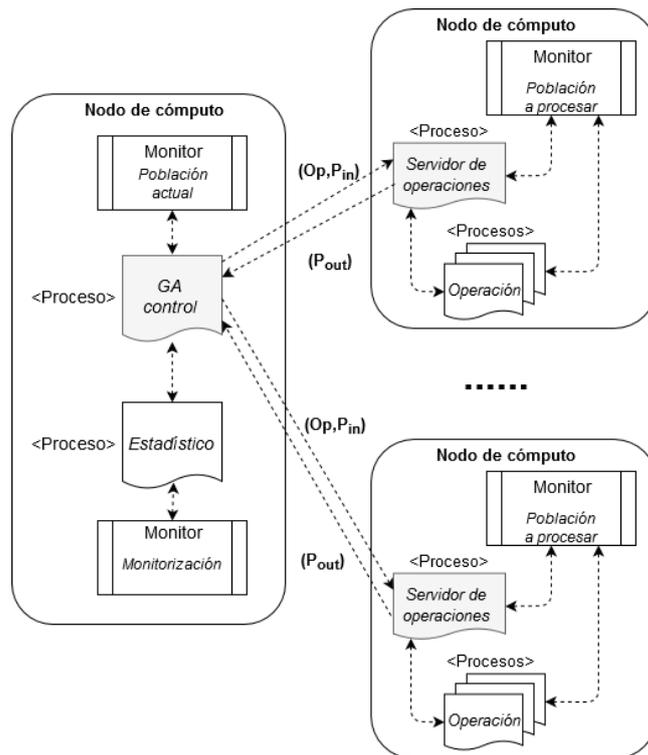


Figura 3: Esquema distribuido

población actual en sub-poblaciones (tantas como servidores de operaciones), enviará una sub-población a cada servidor para que aplique la operación correspondiente, esperará a que todos los servidores le respondan y, finalmente, fusionará las respuestas para crear la población que resulta de ejecutar la operación concreta. Las estrategias de división y fusión de poblaciones pueden tener una influencia significativa en la calidad de los resultados. Por tanto, es importante reflexionar sobre su diseño e, incluso, probar diferentes alternativas.

La comunicación entre los distintos procesos involucrados en la solución será distribuida y síncrona.

## 5. Instrucciones para el desarrollo y la entrega del trabajo

Es importante que os leáis con detenimiento esta sección. Incluye una panorámica general de las actividades e instrucciones a tener en cuenta para el desarrollo del proyecto e información sobre los hitos y fechas de interés.

### 5.1. Organización inicial del trabajo

La realización de este proyecto es una actividad en grupo. En la página Web de la asignatura se publicaron las instrucciones y los plazos para la creación de los grupos de trabajo. Para poder entregar y defender el proyecto realizado es obligatorio haber completado este trámite previo.

### 5.2. Elección del esquema arquitectural

El enunciado propone tres posibles esquemas arquitecturales de complejidad creciente, desde una solución concurrente hasta una distribuida con varios servidores. Dado que la carga de trabajo de cada solución es diferente, la nota máxima a la que se puede aspirar también varía. De esta forma, aquellos equipos que programen el esquema concurrente podrán aspirar a una nota máxima en el proyecto de 7 puntos sobre 10. En el caso de los esquemas distribuidos su puntuación máxima es de 8.5 puntos (un sólo servidor de operaciones) y 10 puntos sobre 10 (múltiples servidores), respectivamente.

En la primera sesión de seguimiento con el tutor, cada equipo deberá notificar el esquema arquitectural que va a diseñar y programar. No obstante, esta decisión puede variar en el transcurso del proyecto (por ejemplo, un equipo puede decidir programar en primer lugar una solución concurrente y, posteriormente, implementar una versión distribuida).

### 5.3. Material a entregar como resultado

Esta primera versión del enunciado se centra en describir el sistema objetivo a desarrollar. En próximas fechas se publicará una versión complementaria donde se describa con precisión todo el material a entregar como resultado del proyecto. También se publicarán en la página Web una serie de plantillas para elaborar la documentación final del proyecto.

A nivel de resumen, queremos anticipar que cada grupo deberá entregar como mínimo el siguiente material:

- El código fuente del sistema desarrollado y los correspondientes ficheros `Makefile` para su compilación y ejecución.
- La memoria técnica del proyecto, donde se describa el análisis y diseño de la solución implementada. Esta memoria también contendrá un análisis de rendimiento y prestaciones del sistema final.
- La memoria organizativa del proyecto, donde se detalle la organización del equipo, el plan de trabajo planificado/real y los esfuerzos dedicados al proyecto.
- Un documento de pruebas que recoja el plan de pruebas realizado para garantizar el correcto funcionamiento del sistema.
- La presentación que se va a realizar para la defensa del proyecto.

En la página Web se publicarán instrucciones precisas que guíen a los grupos en la elaboración de los distintos documentos y las correspondientes plantillas.

#### 5.4. Gestión de la dedicación

Como parte de la documentación a entregar se pide un informe de dedicación al proyecto. Este informe debe detallar las horas que cada estudiante ha dedicado a las distintas tareas del proyecto (participación en reuniones, programación, realización de pruebas, elaboración de la documentación, etc.). Esta información no tiene ningún tipo de consecuencia a nivel de evaluación y calificación, simplemente es una práctica habitual en el desarrollo de proyectos *software*. El objetivo es conocer el esfuerzo que ha costado a los estudiantes realizar el trabajo y valorar objetivamente si corresponde con el planificado en la ficha de la asignatura. Por tanto, solicitamos sinceridad en los datos que se entreguen. Nos ayudará a mejorar la asignatura.

Para facilitaros la labor os proporcionamos un documento `Excel` para la gestión de los esfuerzos individuales. Cada vez que trabajéis en el proyecto debéis apuntar las horas dedicadas, especificando la fecha y hora, la tarea realizada, y el tiempo invertido en fracciones de 15 minutos (es decir, si dedico 12 minutos, apuntaría 15 minutos de trabajo; o si dedico 37 minutos, apuntaría 45 minutos de trabajo). En el documento también se proporciona un listado de tareas posibles.

Por otro lado, está la cuestión de cómo computar las horas que se trabaja en equipo. Si por ejemplo se reúnen tres estudiantes para realizar una tarea durante 50 minutos, cada estudiante deberá apuntar en su dedicación individual que participó en la tarea durante 1 hora. Es decir, el esfuerzo computable es 50 minutos por el número de participantes. Esta regla se aplicará a cualquier tarea en grupo.

Finalmente, queremos recordaros que debéis ser disciplinados a la hora de registrar la información de esfuerzos. Una buena costumbre es apuntar la información nada más finalizar cada tarea.

#### 5.5. Procedimiento y fechas de entrega del proyecto

La fecha límite para la entrega del material del proyecto será el día 14 de enero del 2020, a las 23:59h. Las instrucciones y condiciones de entrega serán

publicadas con suficiente antelación. Esta entrega supone el primer paso en el proceso de evaluación de la actividad.

El segundo paso consiste en la evaluación presencial de los proyectos. Esta evaluación incluye una breve presentación de los resultados y una demostración en el laboratorio de prácticas del sistema implementado. Una vez entregados los trabajos, se publicarán con antelación suficiente las fechas y horas concretas en las que se realizará esta evaluación presencial. La asistencia a esta prueba es obligatoria para todos los integrantes del grupo. Los integrantes que no asistan serán calificados como *suspense*. La fecha de la defensa será los días 16 y 17 de enero del 2020. También se publicarán instrucciones sobre el desarrollo de esta prueba.