## Universidad Tecnológica Metropolitana.

Departamento de Computación e Informática. Análisis de Algoritmos.

Profesor: Sebastián Salazar Molina.

# Proyecto

08 de agosto del 2020

## VISIÓN GENERAL

Para todas las instituciones de educación superior, la programación horaria de exámenes, es una tarea difícil de resolver debido a la multiplicidad de requerimientos concurrentes que debe satisfacer una solución.

#### **OBJETIVOS**

- 1. Investigar y usar algoritmos de voraces.
- 2. Investigar y usar algoritmos probabilísticos.
- 3. Mejorar el uso de archivos.

#### **ESPECIFICACIONES**

La Universidad Tecnológica Metropolitana ha dispuesto una semana (5 días hábiles, de lunes a viernes) para realizar pruebas y evaluaciones. Se le ha solicitado a usted realizar una aplicación piloto, con un conjunto reducido de carreras y algunas aulas acotadas, si esta experiencia funciona la Universidad implementará esta solución de forma permanente.

#### Salas de clase.

Para esta prueba se han disponibilizado las siguientes salas de clases:

- M2-201
- M2-202
- M2-203
- M2-204
- M2-205

## **Bloques Horarios.**

Para esta semana de pruebas, se ha definido un conjunto de períodos especiales, para facilitar las evaluaciones, estos bloques son:

- **Período 1** de 08:00 a 09:25 horas (Mañana).
- **Período 2** de 09:30 a 10:55 horas (Mañana).
- **Período 3** de 11:00 a 12:25 horas (Mañana).
- **Período 4** de 12:30 a 13:55 horas (Mañana).
- **Período 5** de 14:00 a 15:25 horas (Tarde).
- **Período 6** de 15:30 a 16:55 horas (Tarde).
- **Período 7** de 17:00 a 18:25 horas (Tarde).
- **Período 8** de 18:30 a 20:00 horas (Tarde).

### **Asignaturas**

Se adjunta un archivo CSV, con la siguiente estructura (la primera fila indica la etiqueta de las columnas):

- **Semestre**, número entero mayor que 0 (cero), que indica el semestre que corresponde la asignatura para la carrera.
- Carrera, es el código de la carrera, para la prueba de concepto los valores posibles son:
  - o 21030.
  - 0 21041.
  - o 21049.
  - o 21071.
- Asignatura, es el nombre del ramo, se define que si para dos (o más) carreras tienen el mismo nombre de asignatura, son la misma asignatura.

#### Restricciones.

Las siguientes son las restricciones mínimas que la aplicación debe cumplir.

- Cada filas del CSV corresponde a una sección, la cuál usará una sala de clases.
- Asignaturas de un mismo nombre, deben rendirse en paralelo en el mismo período y día, pero en aulas distintas.
- No pueden haber dos o más pruebas de primer o segundo semestre en la mañana o tarde del mismo día.
- Cada prueba se rendirá en un período por completo en un día específico (Lunes, Martes, Miércoles, Jueves o Viernes).

# Valoración especial.

Se valorará de mejor manera aquellas agrupaciones que minimicen las ventanas horarias de los estudiantes.

## **Aplicación**

# Implementación.

La implementación debe realizarse usando el compilador GCC (versión 9.3) incluido en Ubuntu 20.04 LTS. Se valora el uso de OpenMP.

Se solicita desarrollar una aplicación C/C++, que utilizando un archivo de entrada (descrito anteriormente). debe entregar como salida un conjunto de archivos CSV (de texto plano), en donde cada archivo debe corresponder a una sala de clases (en total 5 archivos).

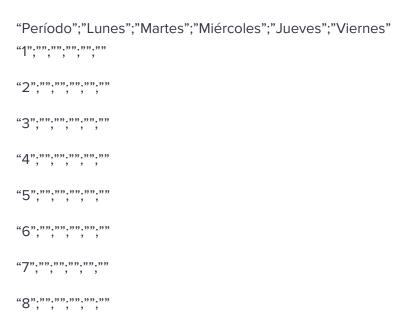
- Cada archivo debe disponer de 9 filas y 6 columnas.
- La primera fila corresponde a las etiquetas de las columnas y deben ser:
  - Período
  - Lunes
  - Martes
  - Miércoles
  - Jueves
  - Viernes
- La primera columna debe contener los períodos correspondientes:
  - Período
  - 0 1
  - 0 2
  - 0 3

  - 0 5
  - 0 6
  - 0 7
  - 0 8

## **Ejemplo Visual**

Período	Lunes	Martes	Miércoles	Jueves	Viernes
1					
2					
3					
4					
5					
6					
7					
8					

#### **Ejemplo CSV**



**Nota**: En cada celda se debe colocar la **Carrera - Asignatura** (Ejemplo: "21041 - Inglés I") que rinde prueba en dicha aula.

# Equipo.

El proyecto deberá ser realizado en forma grupal, con los equipos (de tres integrantes) conformados en clases.

#### Informe.

Se debe presentar un informe con la siguiente estructura:

- Portada.
- Índice.
- Glosario de símbolos o términos técnicos, en caso de que fuese necesario.
- Debe tener una introducción.
- Desarrollo, debe indicar la forma de resolver el problema, además de entregar la complejidad de los algoritmos usados y métricas de rendimiento.
- Debe tener conclusiones.
- Los anexos que se necesiten.
- La bibliografía consultada.
- No debe superar las 15 páginas.

# Código.

El código debe ser entregado a más tardar, el 08/08/2020 hasta las 23:59:59.999 horas de Santiago de Chile, este proyecto debe estar respaldado en un repositorio personal github.

# **EVALUACIÓN**

#### Documentación.

Parte de la evaluación consiste en la documentación de las funciones. Que debe ser clara, concisa y descriptiva de lo que el código realiza.

# Código

El código debe ser claro, fácil de leer, ordenado y cumplir con buenas prácticas de programación, se inspeccionará el código fuente.

#### Resultados.

Un criterio de evaluación que se tomará en consideración: el tiempo de ejecución de la tarea. Menos es mejor. La evaluación es porcentual.