

# Manual de desarrollador de aplicación MATE

Escrito por Nicolás Vargas C. (nlvargas@uc.cl) el 21 de Marzo del 2022

En este documento se detallan las principales características técnicas de la aplicación con el objetivo de ayudar a la continuidad de este proyecto y para solucionar posibles errores que puedan ocurrir en el futuro.

**Link a la aplicación:** <https://my49ptyg5m.execute-api.us-east-2.amazonaws.com/dev>

## 1. Stack de la aplicación

El backend de la aplicación fue desarrollado utilizando el framework para aplicaciones web Django (Python) y el frontend en ReactJS (JavaScript).

El modelo de optimización que conforma los grupos se resuelve a través de la librería *Gurobi*

Por ahora la aplicación no requiere de ningún tipo de autenticación ni registro de datos por lo que no se utilizan bases de datos.

## 2. Infraestructura de la aplicación



### 2.1. AWS

La aplicación se encuentra alojada en AWS utilizando los servicios *Lambda AWS* y *S3*. El primero de estos servicios, *Lambda AWS*, permite realizar hasta un millón de requests al mes de forma gratuita y sin administrar

servidores (serverless). Mientras que *S3* sirve para almacenar los archivos estáticos (frontend) de la aplicación. Este último servicio también es gratuito hasta los 5GB de almacenamiento (actualmente la aplicación solo utiliza 4MB).

Si se desea desplegar algún cambio en la aplicación deben seguir el siguiente tutorial y solicitar un archivo de acceso a *nlvargas@uc.cl* para poder realizar cambios en los servicios de AWS.

## 2.2. Cluster

El cluster de cómputo es el encargado de correr el modelo de optimización y posteriormente enviar los resultados por email al usuario. Si se desea cambiar algo en alguna de estas funcionalidades deberá acceder al cluster siguiendo estas instrucciones y con las siguientes credenciales:

Usuario	nlvargas
Contraseña	Q2Wd6lPfMy9B

## 3. Obtención de la solución

El proceso a través del cual se obtiene la solución se corre en el Clúster de cómputo y lo podemos separar en 3 partes:

### 3.1. Preprocesamiento

Primero, a partir del Excel que contiene la información de los atributos, disponibilidades y preferencias de cada alumno se realiza una clasificación por "tipo" de alumno. Vale decir, dos alumnos que tienen los mismos atributos, disponibilidades y preferencias serán considerados como el mismo tipo de alumno por el modelo. De esta forma, reducimos la cantidad de variables del modelo de optimización y eliminamos soluciones simétricas.

### 3.2. Modelo de optimización

Luego de preprocesar los datos se define y resuelve el siguiente problema de optimización:

#### Conjuntos

Tipos de alumnos:  $I$

Tipos de alumnos de los cuales no se saben sus preferencias:  $NA \subseteq I$

Características:  $R$

Atributos:  $A$

Grupos:  $G$

Temas:  $T$

Módulos:  $M$

Grupos con el tema  $t \in T$ :  $G_t \subseteq G$

Grupos del módulo  $m \in M$ :  $G_m \subseteq G$

Grupos con el tema  $t \in T$  del módulo  $m \in M$ :  $G_{tm} \subseteq G$

## Parámetros

$N_{grupos}$  : Número de grupos a conformar

$R_i$  : Cantidad de alumnos del tipo  $i \in I$

$Q_{min}$  : Cantidad mínima de alumnos por grupo

$Q_{max}$  : Cantidad máxima de alumnos por grupo

$LR_r$  : Cantidad mínima de alumnos con la característica  $r \in R$  por grupo

$UR_r$  : Cantidad máxima de alumnos con la característica  $r \in R$  por grupo

$LT_t$  : Cantidad mínima de grupos con el tema  $t \in T$

$UT_t$  : Cantidad máxima de grupos con el tema  $t \in T$

$C_m$  : Capacidad máxima del módulo  $m \in M$

$D_{im}$  : 1 si el alumno  $i \in I$  está disponible en el módulo  $n \in M$

$N_i$  : Disponibilidad total de los alumnos del tipo  $i \in I$  (ie.  $\sum_{m \in M} D_{im}$ )

$A_{ir}$  : 1 si los alumnos del tipo  $i \in I$  poseen el atributo  $r \in R$

$P_{ig}$  : Prioridad en la que los alumnos del tipo  $i \in I$  ubican al grupo  $g \in G$

$FD_{tm}$  : 1 si el tema  $t \in T$  solo puede estar en el modulo  $m \in M$

$SD$  : 1 si los grupos de un mismo tema deben quedar asignados en el mismo módulo

$UP$  : Cantidad máxima de temas a utilizar

$M$  : Número muy grande

## Variables

$y_{ig}$ : Cantidad de alumnos del tipo  $i \in I$  asignados al grupo  $g \in G$

$w_g$ : 1 si el grupo  $g \in G$  tiene algún alumno asignado, y 0 e.o.c.

$z_i$ : Prioridad que los alumnos del tipo  $i \in I$  le dan en el que fueron asignados

$z_{max}$ : Prioridad obtenida por los alumnos que quedaron en la peor prioridad

$m_g$ : Cantidad de alumnos en el grupo  $g \in G$  de los cuales no se saben sus preferencias , y 0 e.o.c.

$m_{max}$ : Cantidad de alumnos de los cuales no se saben sus preferencias en el grupo que tiene más alumnos de los cuales no se saben sus preferencias

$o_t$ : 1 si se conforma algún grupo con el tema  $t \in T$ , y 0 e.o.c.

$u_{tm}$ : 1 si el tema  $t \in T$  queda asignado en el módulo  $m \in M$ , y 0 e.o.c.

$q_{gr}$ : número de alumnos con la característica  $r \in R$  en el grupo  $g \in G$

$p_{gr}$ : 1 si no hay alumnos con la característica  $r \in R$  en el grupo  $g \in G$ , y 0 e.o.c.

## Restricciones

- (a) Todos los alumnos deben ser asignados a algún grupo

$$\sum_{g \in G} y_{ig} = R_i \quad \forall i \in I$$

- (b) Activación de  $w$

$$y_{ig} \leq w_g Q_{max} \quad \forall i \in I, \forall g \in G$$

- (c) Cantidad de grupos

$$\sum_{g \in G} w_g = N_{grupos}$$

- (d) Cantidad mínima y máxima de alumnos por grupo

$$Q_{min} w_g \leq \sum_{i \in I} y_{ig} \leq Q_{max} w_g \quad \forall g \in G$$

- (e) Cantidad mínima y máxima de grupos para cada tema

$$LT_t \leq \sum_{g \in G_t} w_g \leq UT_t \quad \forall t \in T$$

- (f) Activación de  $o$

$$u_{pd} \leq o_t \quad \forall t \in T, \forall m \in M$$

- (g) Cantidad máxima de preferencias a usar

$$\sum_{t \in T} o_t \leq UP$$

- (h) No puede haber mas de un alumno que no contesta por grupo

$$\sum_{i \in NA} y_{ig} \leq 1 + m_g \quad \forall g \in G$$

- (i) Activación de  $m_{max}$

$$m_g \leq m_{max} \quad \forall g \in G$$

- (j) Prioridad que los alumnos de cada tipo le dan en el que fueron asignados

$$z_i = \sum_{g \in G} y_{ig} \quad \forall i \in I$$

- (k) Activación de  $z_{max}$

$$z_i \leq z_{max} \quad \forall i \in I$$

- (l) Cantidad mínima y máxima de alumnos con el atributo  $r$

$$LR_r w_g \leq \sum_{i \in I} y_{ig} A_{ir} \leq UR_r w_g \quad \forall g \in G, \forall r \in R$$

- (m) Restricciones disjuntas para  $r$

$$Q_{gr} = \sum_{i \in I} y_{ig} A_{ir} \quad \forall g \in G, \forall r \in R$$

$$LR_r(w_g - p_{gr}) \leq Q_{gr} \leq UR_r(1 - p_{gr}) \quad \forall g \in G, \forall r \in R$$

(n) Capacidad máxima de cada modulo

$$\sum_{i \in I} \sum_{g \in G_m} y_{ig} \leq C_m \quad \forall m \in M$$

(ñ) Disponibilidad de cada tipo de alumno

$$\sum_{g \in G_m} y_{ig} \leq D_{im} \quad \forall i \in I, \forall m \in M$$

(o) Si  $SD = 1$ , entonces todos los grupos de un mismo tema deben quedar asignados en el mismo módulo

$$\sum_{g \in G_{tm}} w_g \leq N_{grupos} u_{tm} \quad \forall t \in T, \forall m \in M$$

$$\sum_{m \in M} u_{tm} \leq 1 + M(1 - SD) \quad \forall t \in T, \forall m \in M$$

(p) No asignar el tema  $t$  en módulos en los que no puede estar

$$u_{tm} = 0 \quad \forall t \in T, m \in M : FD_{tm} = 0 :$$

(q) Naturaleza de las variables

$$y, z, z_{max}, m, m_{max}, q \in \mathbb{Z}^+ \\ w, p, u, o \in \{0, 1\}$$

**Función objetivo**

$$\min \sum_{i \in I} N_i z_i + 1000 z_{max} + 1000 m_{max}$$

### 3.3. Postprocesamiento

Finalmente tomamos la solución del modelo, asignamos los alumnos a los grupos según el tipo y enviamos la solución por email al usuario

## 4. Código fuente

El repositorio de Github con las instrucciones para correr localmente la aplicación y el código fuente es el siguiente: <https://github.com/nlvargas/MATE>